

# **GAFAS – Graphentheoretische Algorithmen zur Analyse und Synthese (nichtlinearer Systeme)**

Michael Spielmann und Torsten Wey

Forschungsbericht Nr. 13/97

Meß-, Steuer- und Regelungstechnik

**Übersicht:** Die Analyse von nichtlinearen Systemen und die Berechnung nichtlinearer Regelungsgesetze ist ohne die Verwendung von symbolverarbeitenden Computer-Algebra-Systemen (CAS) kaum vorstellbar. Je nach Komplexität und Größe des Systems werden aber sehr schnell die Rechnerleistungsgrenzen erreicht, die durch die zur Verfügung stehende Hardware gegeben ist. Bei Verwendung graphentheoretischer Algorithmen kann durch die Einführung generischer Größen eine drastische Reduzierung der Rechenzeiten erzielt werden. Eine Reihe solcher Algorithmen sind zu dem Software-Tool GAFAS für das CAS MAPLE® zusammengefügt. Die Handhabung dieses Programmpakets und die verwendeten Algorithmen werden in dem vorliegenden Bericht vorgestellt. GAFAS ist über das Internet auf der Fachgebiete Seite

<http://www.msrt.uni-duisburg.de/software/index.html>

erhältlich.

Gerhard-Mercator-Universität - GH Duisburg  
Meß-, Steuer- und Regelungstechnik  
Prof. Dr.-Ing. H. Schwarz

# Inhaltsverzeichnis

Nomenklatur	II
1 Einführende Übersicht	1
2 Graphentheoretische Grundlagen	3
3 GAFAS - Handhabung	7
3.1 Installation der Toolbox . . . . .	7
3.2 Verwendung in MAPLE <sup>®</sup> . . . . .	8
3.3 Beispiel QLS . . . . .	9
4 Implementierte Befehle	12
4.1 Strukturelles System – SSystem . . . . .	12
4.2 Struktureller Graph – SGraph . . . . .	13
4.3 Struktureller Rang – SRank . . . . .	14
4.4 Differenzengrad – (E)SRelativeDegree . . . . .	16
4.5 Nullstellen im Unendlichen – (E)SZerosAtInfinity . . . . .	18
4.6 Beobachtbarkeit – (E)SObservability . . . . .	27
5 Zusammenfassung und Ausblick	30
6 Literatur	31

# Nomenklatur

## Skalare Größen:

$i, j, k$	Laufindex
$l_P$	Länge des Pfades $P$
$m$	Anzahl der Systemeingänge
$n$	Systemordnung
$p$	Anzahl der Systemausgänge
$\rho$	Differentieller Rang
$\rho_k$	Dimension des Vektorraums $\mathcal{E}_k$
$\sigma_k$	Anzahl der NU mit Ordnung kleiner oder gleich k

## Vektoren, Vektorfelder und Matrizen:

$\mathbf{A}$	Systemmatrix
$\mathbf{a}(\mathbf{x}(t))$	Systemvektorfeld eines ALS
$B$	Nichtleere Teilmenge aus $\mathbb{R}$
$\mathbf{B}$	Eingangsmatrix
$\mathbf{B}(\mathbf{x}(t))$	Eingangsvektorfeld eines ALS
$C$	Ausgangsmatrix
$\mathbf{c}(\mathbf{x}(t))$	Ausgangsvektorfeld eines ALS
$\mathbf{d}$	Vektordifferenzengrad eines MIMO Systems
$\mathbf{d}^*$	Struktureller Vektordifferenzengrad eines MIMO Systems
$\mathbf{d}^{*e}$	Erweiterter struktureller Vektordifferenzengrad eines MIMO Systems
$\mathbf{f}(\mathbf{x}(t), u(t))$	Systemvektorfeld eines NS
$\mathbf{F}(s)$	Übertragungsmatrix
$\mathbf{h}(\mathbf{x}(t))$	Ausgangsvektorfeld
$\mathbf{J}_k$	Jacobi-Matrix
$\mathbf{M}^*$	Strukturmatrix von $\mathbf{M}$
$\mathbb{N}^+$	Menge der natürlichen Zahlen ohne Null
$n_k*$	Ordnung der NU
$Q$	Beobachtbarkeitsmatrix
$Q^{*e}$	Erweiterte strukturelle Beobachtbarkeitsmatrix
$\mathbb{R}$	Menge der reellen Zahlen
$\mathcal{U}$	Durch $d\bar{\mathbf{u}}$ aufgespannter Vektorraum
$\mathbf{u}(t)$	Eingangsvektor
$\mathbf{u}_0$	Arbeitspunkt im Eingangsraum

---

$\bar{\mathbf{u}}$	Abkürzende Schreibweise für $\{\mathbf{u}, \dot{\mathbf{u}}, \ddot{\mathbf{u}}\}$
$\mathbf{x}(t)$	Zustandsvektor
$\mathbf{x}_0$	Arbeitspunkt im Zustandsraum
$\mathbf{y}(t)$	Ausgangsvektor

**Mengen, Mengenelemente:**

$D$	Geordnetes Paar
$e$	Element der Kantenmenge
$E, E_{\mathcal{G}}$	Kantenmenge
$G$	Graph (geordnetes Paar, bestehend aus Kanten- und Knotenmenge)
$G'$	Kantenbewerteter Graph
$G^*$	Funktional kantenbewerteter Graph, Struktureller Graph
$P$	Pfad
$u$	Eingangsknoten
$v, v'$	Element der Knotenmenge
$V, V_{\mathcal{G}}$	Knotenmenge
$x$	Zustandsknoten
$y$	Ausgangsknoten
$\mathcal{E}$	Vektorraum über dem Körper $\bar{K}$
$\mathcal{E}_k$	durch $\{\mathrm{d}\mathbf{y}, \dots, \mathrm{d}\mathbf{y}^{(k)}\}$ aufgespannter Vektorraum
$\Gamma$	Graphen Menge

**Funktionen, Operatoren:**

diff. dim	Differentielle Dimension eines Vektorraums
dim	Dimension eines Vektorraums
$F$	Abbildung des Zustands- und Eingangsraums in $\mathbb{R}$
$\bar{K}$	Körper der rationalen Funktionen in $\mathbf{u}, \dot{\mathbf{u}}, \ddot{\mathbf{u}}, \dots$ , mit meromorphen Koeffizienten in $\mathbf{x}$
$K < \mathbf{u} >$	differentieller Körper aller rationalen Funktionen in den Variablen $\mathbf{u}^{(i)}$ mit Koeffizienten in $K$
$L$	Lie-Operator
$\min(\cdot)$	Kleinster Wert aus $(\cdot)$
$\mathrm{Rg}(\mathbf{A})$	Rang der Matrix $\mathbf{A}$
$\mathrm{span}_K \{\cdot\}$	der durch die Operanden aufgespannte $K$ -Vektorraum
$c w(e)$	Wert der Kante $e$
$\alpha(e)$	Funktionales Kantengewicht

$\beta(e)$	Abbildung der Kantenelemente in $\mathbb{R}$
$(\cdot)^T$	Transponierte einer Matrix
$\oplus$	Kronecker Summe

**Abkürzungen:**

ALS	Analytisches System mit linear eingehender Steuerung
QLS	Quadratisches System mit linear eingehender Steuerung
LS	Lineares System
MIMO	Mehrgrößensystem (Multi Input Multi Output)
NS	Nichtlineares System
NU	Nullstellen im Unendlichen
$\Sigma$	System

# 1 Einführende Übersicht

Die Untersuchungen nichtlinearer Systeme auf die für eine Reglerauslegung relevanten Systemeigenschaften sind im allgemeinen nur unter Zuhilfenahme von Computer-Algebra-Systemen, kurz CAS, möglich. Vor diesem Hintergrund sind bereits eine Reihe von Software-Tools entstanden, mit dem Ziel, die routinemäßig anfallenden Berechnungen zu automatisieren und damit auch einem größeren Personenkreis nichtlineare Beurteilungs- und Auslegungskriterien zugänglich zu machen. Genannt sei das ebenfalls in diesem Fachgebiet entstandene Tool NSAS – Nonlinear System Analysis and Synthesis Package (Lemmen u. a. 1995) – und das Software Paket NonLinCon (Essen und Jager 1992). Beide setzen die beispielsweise in (Isidori 1995) und (Schwarz 1991) zu findenden Definitionen für Systemeigenschaften und Gleichungen zur Reglerauslegung in entsprechende Algorithmen um.

GAFAS – Graph-theoretical Algorithms For Analysis and Synthesis (of nonlinear systems) – baut auf diesen Software-Paketen auf. Wie in verschiedenen Veröffentlichungen (Svaricek 1993, Wey 1996, Spielmann und Schwarz 1996, Lévine 1997) nachgewiesen, zeigt sich ein praktischer Nutzen bei der Verwendung graphentheoretisch basierter Algorithmen im Besonderen durch die Reduzierung von Rechenzeiten.

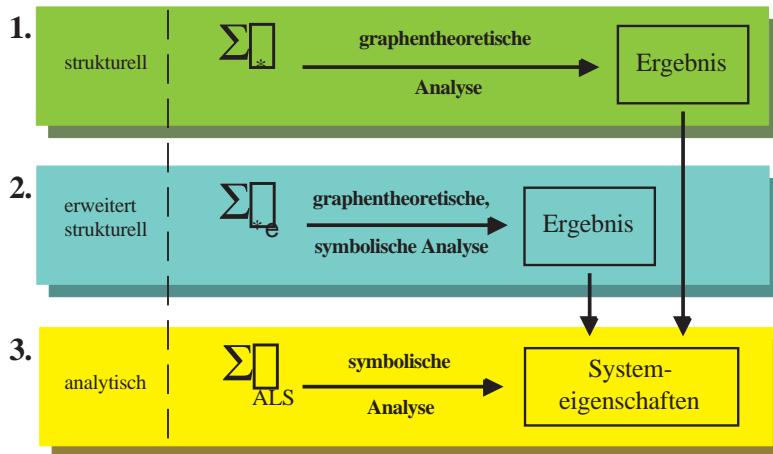


Bild 1.1: Vorgehensweise bei der Systemanalyse

Dieser Vorteil wird im wesentlichen aufgrund der in Bild 1.1 am Beispiel der Systemanalyse dargestellten Schritte erzielt. So kann bereits vielfach bei der strukturellen Analyse (Schritt 1) ein Ergebnis erzielt werden. Der dazu verwendete Strukturgraph enthält für diesen Zweck lediglich Informationen über Abhängigkeiten zwischen den Systemzuständen.

Führt diese strukturelle Untersuchung zu keinem brauchbaren Ergebnis, werden zusätzlich Gewichtungen zwischen den Zuständen berücksichtigt (extended structural analysis). Die

Gewichte können konstant oder Funktionen in den Systemzuständen sein.

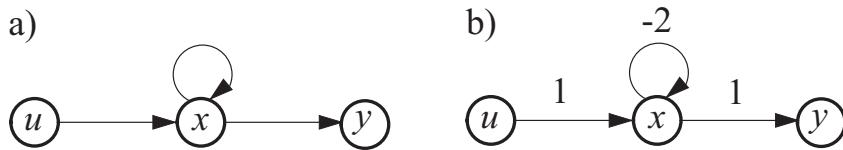
Wenn auch diese *erweiterte* strukturelle Untersuchung keinen Erfolg bringt, dann wird auf die herkömmliche, also analytische Vorgehensweise zurückgegriffen. Der sich daraus ergebende Nachteil, zwei zusätzliche Operationen durchgeführt zu haben kann vernachlässigt werden, da die strukturelle und die erweiterte strukturelle Analyse auf graphentheoretischen Methoden basieren und in der Regel schnelle Ergebnisse liefern. Des weiteren ist bei einer analytischen Untersuchung komplexer Systeme bzw. bei Systemen mit hoher Systemordnung die Wahrscheinlichkeit groß, an die Grenzen der heutigen Rechnerleistungen zu stoßen.

Nachdem im folgenden Abschnitt einige für das weitere Verständnis notwendige Grundlagen zur Graphentheorie behandelt werden, wird in Abschnitt 3 die Handhabung von GAFAS näher erläutert. Alle implementierten Befehle werden mit dem notwendigen theoretischen Hintergrund in Abschnitt 4 erörtert.

Zusammenfassung und Ausblick schließen den Bericht ab.

## 2 Graphentheoretische Grundlagen

Obwohl die Graphentheorie bereits über 250 Jahre in unzähligen Bereichen erfolgreich angewendet wird (Sachs und Stiebitz 1987), ist sie im Laufe der Zeit immer wieder neu entdeckt worden. Dies hängt damit zusammen, daß den Elementen, aus denen ein Graph besteht (Bild 2.1), unterschiedliche Bedeutungen zugeordnet werden können. Bei technischen Systemen beschreibt der hier gewählte Graph die Abhängigkeiten der Zustandsänderungen von den Systemzuständen. Mit den Informationen lassen sich strukturelle Untersuchungen durchführen, welche erste Hinweise auf Eigenschaften geben, die alle Systeme mit gleicher Struktur gemeinsam haben. Die Hinzunahme einer Gewichtung dieser Abhängigkeiten (Bild 2.1b) wird erforderlich, wenn anhand des Strukturgraphen allein keine abschließenden Aussagen getroffen werden können.



**Bild 2.1:** Gerichteter Graph: a) Strukturgraph  
b) bewerteter Graph

Bei den hier betrachteten gerichteten Graphen haben die Kanten eine Pfeilspitze, und damit eine festgelegte Orientierung. Einen Überblick über die unterschiedlichen Definitionen für Graphen ist u. a. in (Gondran und Minoux 1986) und (Lemmen 1996) zu finden.

Üblicherweise werden die Kanten und Knoten zu einer Kantenmenge  $E$  und einer Knotenmenge  $V$  zusammengefaßt:

**Definition 2.1** (Wagner und Bodendiek 1989)

Ein geordnetes Paar  $D = (V, E)$  ist ein endlicher, gerichteter Graph genau dann, wenn  $V$  eine endliche Menge ist und  $E$  eine Teilmenge des kartesischen Produkts  $V \times V = \{(v, v') \mid v, v' \in V\}$  mit  $E \subseteq V \times V$  bedeutet und  $E$  außerdem die Bedingung erfüllt, daß aus  $(v, v') \in E$  stets  $(v', v) \notin E$  gilt.  $\square$

Eine mögliche graphische Darstellung für ein nichtlineares System<sup>1</sup>

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) &= \mathbf{h}(\mathbf{x}(t))\end{aligned}\tag{2.1}$$

mit  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{u} \in \mathbb{R}^m$  und  $\mathbf{y} \in \mathbb{R}^p$

---

<sup>1</sup> Die Zeitabhängigkeit der Zustände und Ein-/Ausgänge wird in diesem Abschnitt nicht weiter angegeben.

als gerichteter Graph resultiert aus Anwendung der in (Reinschke 1988) zu findenden Bildungsregeln:

1. Der Graph hat für die Eingänge, die Zustände und für die Ausgänge je einen Knoten mit entsprechender Bezeichnung:

$$V = \{x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m, y_1, y_2, \dots, y_p\} \quad .$$

2. Wenn die Zustandsvariable  $x_j$  in  $f_i(\mathbf{x}, \mathbf{u})$  enthalten ist, dann existiert eine Kante vom Zustandsknoten  $x_j$  zum Zustandsknoten  $x_i$ :

$$(x_j, x_i) \in E \quad .$$

3. Wenn die Eingangsvariable  $u_j$  in  $f_i(\mathbf{x}, \mathbf{u})$  enthalten ist, dann existiert eine Kante vom Eingangsknoten  $u_j$  zum Zustandsknoten  $x_i$ :

$$(u_j, x_i) \in E \quad .$$

4. Wenn die Zustandsvariable  $x_j$  in  $h_i(\mathbf{x})$  enthalten ist, dann existiert eine Kante vom Zustandsknoten  $x_j$  zum Ausgangsknoten  $y_i$ :

$$(x_j, y_i) \in E \quad .$$

Werden die linguistischen Bildungsregeln, die die Elemente der Kantenmenge festlegen, durch partielles Ableiten der Funktionen  $\mathbf{f}(\mathbf{x}, \mathbf{u})$  und  $\mathbf{h}(\mathbf{x})$  nach den Zuständen  $\mathbf{x}$  und Eingängen  $\mathbf{u}$  ersetzt

$$\begin{aligned} E = & \left\{ (x_j, x_i) \mid \frac{\partial f_i(\mathbf{x}, \mathbf{u})}{\partial x_j} \neq 0, \quad i, j = 1 \dots n \right\} \\ \cup & \left\{ (u_j, x_i) \mid \frac{\partial f_i(\mathbf{x}, \mathbf{u})}{\partial u_j} \neq 0, \quad i = 1 \dots n; j = 1 \dots m \right\} \\ \cup & \left\{ (x_j, y_i) \mid \frac{\partial h_i(\mathbf{x})}{\partial x_j} \neq 0, \quad i = 1 \dots p; j = 1 \dots n \right\} \quad , \end{aligned} \quad (2.2)$$

dann besteht die Möglichkeit, neben einer Ja/Nein–Entscheidung bezüglich der Existenz von Kanten, eine Gewichtung (Bewertung) dieser Kanten vorzunehmen.

**Definition 2.2** (Wagner und Bodendiek 1990)

Ist  $G = (V, E)$  ein Graph aus  $\Gamma$ ,  $B$  eine nichtleere Teilmenge von  $\mathbb{R}$  und  $\beta : E \rightarrow B$  eine Abbildung, die jeder Kante von  $G$  eine reelle Zahl aus  $B$  zuordnet, so heißt das Quadrupel  $G' = (V, E, \beta, B)$  kantenbewerteter Graph. Ist  $e \in E$ , so heißt  $\beta(e)$  Wert der Kante  $e$ .  $\square$

Die reellen Zahlen werden für ein technisches System so gewählt, daß sie identisch mit den partiellen Ableitungen (vgl. Gl. (2.2)) an einem Arbeitspunkt sind. Bei einem linearen System in Zustandsraumdarstellung sind die Kantengewichte identisch mit den Elementen der Systemmatrix  $\mathbf{A}$ , der Eingangsmatrix  $\mathbf{B}$  und der Ausgangsmatrix  $\mathbf{C}$ . Durch das Einsetzen eines Arbeitspunktes können Kanten  $e$  mit Gewicht  $w(e) = 0$  auftreten, also Kanten verschwinden.

**Definition 2.3** (Spielmann 1997)

Für ein System mit einer Zustandsraumdarstellung nach Gl. (2.1) resultiert der Wert  $\beta(e)$  einer Kante  $e \in E$  für einen Punkt  $(\mathbf{x}_0, \mathbf{u}_0)$  aus:

$$\beta(e) = \begin{cases} \frac{\partial f_i(\mathbf{x}, \mathbf{u})}{\partial x_j} \Big|_{(\mathbf{x}_0, \mathbf{u}_0)}, & \text{für } e = (x_j, x_i) \in E; i, j = 1 \dots n \\ \frac{\partial f_i(\mathbf{x}, \mathbf{u})}{\partial u_j} \Big|_{(\mathbf{x}_0, \mathbf{u}_0)}, & \text{für } e = (u_j, x_i) \in E; i = 1 \dots n; j = 1 \dots m \\ \frac{\partial h_i(\mathbf{x})}{\partial x_j} \Big|_{(\mathbf{x}_0, \mathbf{u}_0)}, & \text{für } e = (x_j, y_i) \in E; i = 1 \dots p; j = 1 \dots n \end{cases} . \quad (2.3)$$

□

Bei der Betrachtung nichtlinearer Systeme ist die Beschränkung auf einen Arbeitspunkt in der Regel unerwünscht. Damit verbunden ist der Übergang von reellwertigen Kantengewichten zu funktionalen Kantengewichten. Definition 2.2 ist entsprechend zu erweitern:

**Definition 2.4** (Spielmann 1997)

Ist  $G = (V, E)$  ein Graph aus  $\Gamma$ ,  $F : (X, U) \rightarrow \mathbb{R}$  eine Abbildung des Zustandsraums  $X \subseteq \mathbb{R}^n$  und des Eingangsraums  $U \subseteq \mathbb{R}^m$  in  $\mathbb{R}$ , sowie  $\alpha : E \rightarrow F$  eine Abbildung, die jeder Kante von  $G$  eine Funktion aus  $F$  zuordnet, so heißt das Quadrupel  $G^* = (V, E, \alpha, F)$  funktional kantenbewerteter Graph. Ist  $e \in E$ , so heißt  $\alpha(e)$  Wert der Kante  $e$ . □

Das funktionale Kantengewicht  $\alpha(e)$  kann analog zu Definition 2.3 bestimmt werden:

**Definition 2.5** (Spielmann 1997)

Für ein System mit einer Zustandsraumdarstellung nach Gl. (2.1) ist

$$\alpha(e) = \begin{cases} \frac{\partial f_i(\mathbf{x}, \mathbf{u})}{\partial x_j}, & \text{für } e = (x_j, x_i) \in E; i, j = 1 \dots n \\ \frac{\partial f_i(\mathbf{x}, \mathbf{u})}{\partial u_j}, & \text{für } e = (u_j, x_i) \in E; i = 1 \dots n; j = 1 \dots m \\ \frac{\partial h_i(\mathbf{x})}{\partial x_j}, & \text{für } e = (x_j, y_i) \in E; i = 1 \dots p; j = 1 \dots n \end{cases} . \quad (2.4)$$

□

Wie ein Vergleich der Gln. (2.3) und (2.4) zeigt, gilt der Zusammenhang

$$\beta(e) = \alpha(e) \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} . \quad (2.5)$$

Voraussetzung für die Existenz der partiellen Ableitungen ist die stetige Differenzierbarkeit der Funktionen  $\mathbf{f}(\mathbf{x}, \mathbf{u})$  und  $\mathbf{h}(\mathbf{x})$  in Gl. (2.1). Diese wird für die im weiteren ausschließlich betrachtete, technisch relevante Klasse der analytischen Systeme mit linear eingehender Steuerung (Schwarz 1991, Isidori 1995), kurz ALS, erfüllt. Viele Ergebnisse, die anhand der behandelten Graphen gewonnen werden können, lassen sich oftmals auf allgemeine nichtlineare Systeme erweitern. Dies trifft insbesondere dann zu, wenn nicht-differenzierbare Funktionen solchen Kanten zugeordnet werden, die keine Verwendung bei der Systemanalyse oder Reglersynthese finden (Spielmann u. a. 1996).

In der Regel wird vor der expliziten Berechnung der Kanten mit sogenannten generischen Größen gearbeitet (Wey 1996, Spielmann 1996). Dies führt zu einer wesentlichen Reduzierung der Rechenzeit, da nur die tatsächlich benötigten Kantengewichte bestimmt werden und zuvor mit konstanten Größen gerechnet werden kann.

### 3 GAFAS - Handhabung

#### 3.1 Installation der Toolbox

GAFAS ist über die Internet-Adresse

<http://www.msrt.uni-duisburg.de/software/index.html>

verfügbar (Bild 3.1) und kann von dort direkt als `gafas.zip`-Datei geladen werden.

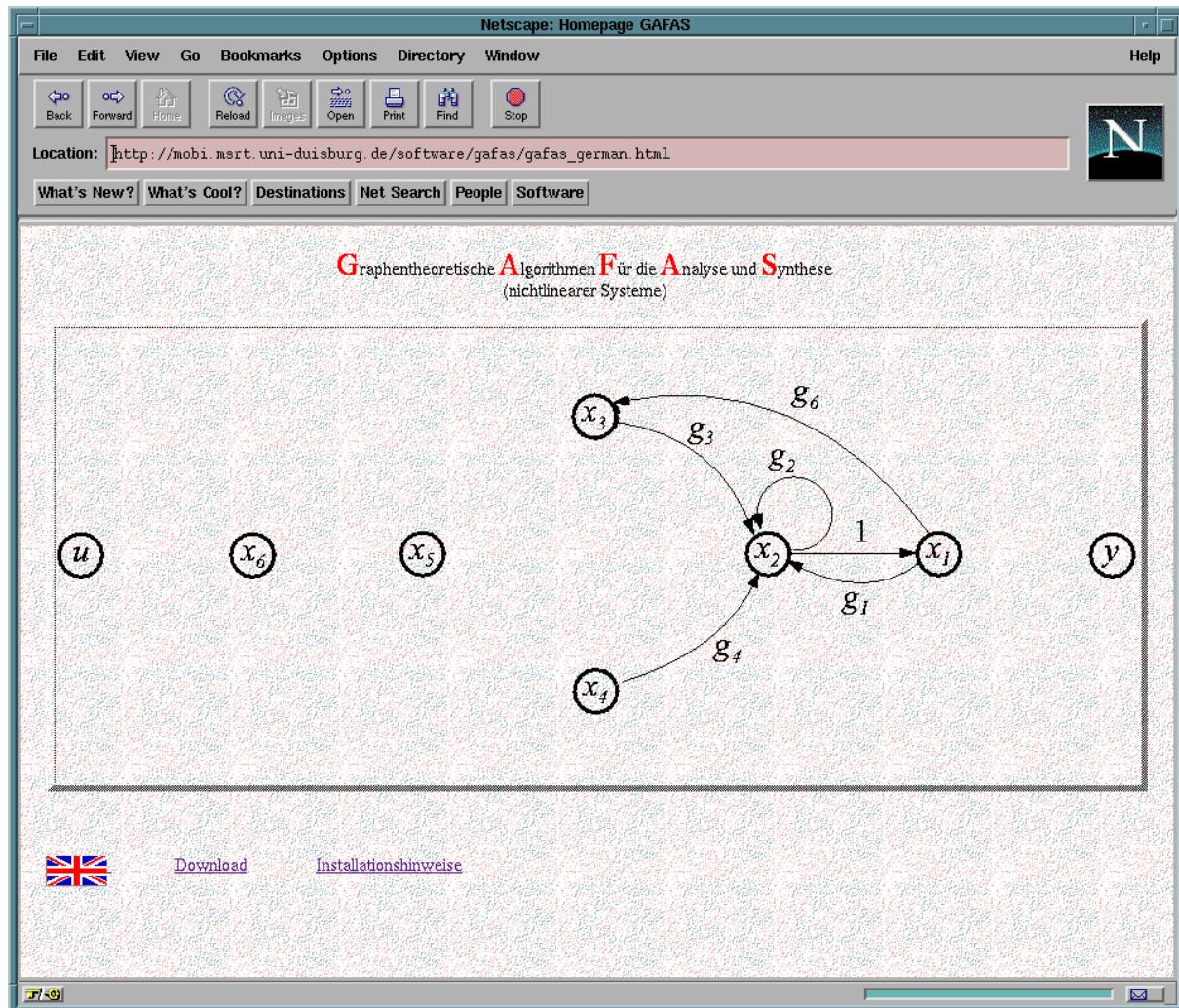


Bild 3.1: Gafas-Homepage

Das `.zip`-Archiv enthält die Dateien

- `gafas.main`
- `gafas.tools`
- `algebra.tools`

- nsas.main
- qls.als
- qls.mws

Die Datei `nsas.main` enthält das in der Einleitung erwähnte Tool NSAS, welches für den Fall, daß weder eine strukturelle noch eine erweitert strukturelle Untersuchung zu einem Ergebnis führt, angewendet wird. Vor dem ersten Aufruf innerhalb einer Maple-Session sollten mit einem Editor die Pfad-Variablen `gafas_dir` und `gafas_help` in der Datei `gafas.main` angepaßt werden.

### 3.2 Verwendung in MAPLE<sup>®</sup>

GAFAS nutzt die Vorteile des Computer Algebra Systems MAPLE<sup>®</sup>, insbesondere den dort bereits vorhandenen Variablentyp für einen Graphen. Nach dem Start einer neuen Maple-Sitzung wird die Tool-Box mit

```
read('gafas.main')
```

aus dem Verzeichnis, in der sich die Datei befindet, geladen. Anschließend stehen die in Tabelle 3.1 aufgeführten und in Abschnitt 4 näher erörterten Befehle zur Verfügung. Der in Bild 1.1 erkennbare Ansatz kann anhand der Befehls-Prefixe `S` für strukturell und `ES` für erweitert strukturell nachvollzogen werden. Befehle ohne Prefix bedeuten demnach die exakte analytische Berechnung, wie sie beispielsweise mit dem bereits erwähnten Software-Tool NSAS vorgenommen werden kann.

<code>SRelativeDegree()</code>	Relativer Grad (strukturell)
<code>ESRelativeDegree()</code>	Relativer Grad (erweitert strukturell)
<code>RelativeDegree()</code>	Relativer Grad <sup>2</sup>
<code>SZerosAtInfinity()</code>	Nullstellen im Unendlichen (strukturell)
<code>ESZerosAtInfinity()</code>	Nullstellen im Unendlichen (erweitert strukturell)
<code>SObservability()</code>	Beobachtbarkeit (strukturell)
<code>ESObservability()</code>	Beobachtbarkeit (erweitert strukturell)

**Tabelle 3.1:** In GAFAS implementierte Befehle

Für eigene Algorithmen können desweiteren die in Tabelle 3.2 genannten und von den GAFAS-Routinen verwendeten Befehle nützlich sein.

Auch diese Befehle werden in Abschnitt 4 näher beschrieben und an Beispielen demonstriert. Der größte Teil der in den Tabellen aufgeführten Befehle erwartet als Argument

---

<sup>2</sup> Dieser Befehl ist nicht in NASA implementiert, deshalb wurde der in GAFAS aufgenommen.

<code>SSystem()</code>	ALS $\Rightarrow$ strukturelles System
<code>SGraph()</code>	ALS $\Rightarrow$ Strukturgraph
<code>SRank()</code>	Struktureller Rang einer Matrix

**Tabelle 3.2:** Ergänzende Befehlsliste

eine Liste mit den funktionalen Vektoren und Matrizen eines analytischen Systems mit linear eingehender Steuerung (Gl. 3.1):

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{a}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{c}(\boldsymbol{x})\end{aligned}\tag{3.1}$$

Hilfreich ist es, das System in einer separaten Datei zu hinterlegen, wie das Beispiel in dem folgenden Unterabschnitt zeigt.

### 3.3 Beispiel QLS

Untersucht werden soll das mit Gl. (3.2) gegebene quadratische System mit linear einge- hender Steuerung:

Das System der Ordnung  $n = 12$  mit  $m = 2$  Eingängen und  $p = 2$  Ausgängen ist in der Darstellung als ALS gegeben und sollte zweckmäßigerweise in eine Datei wie folgt übernommen werden:

```
# quadratic system
n := 12:
m := 2:
p := 2:
x := linalg[vector](n, []):
a := linalg[vector](n,[
```

```

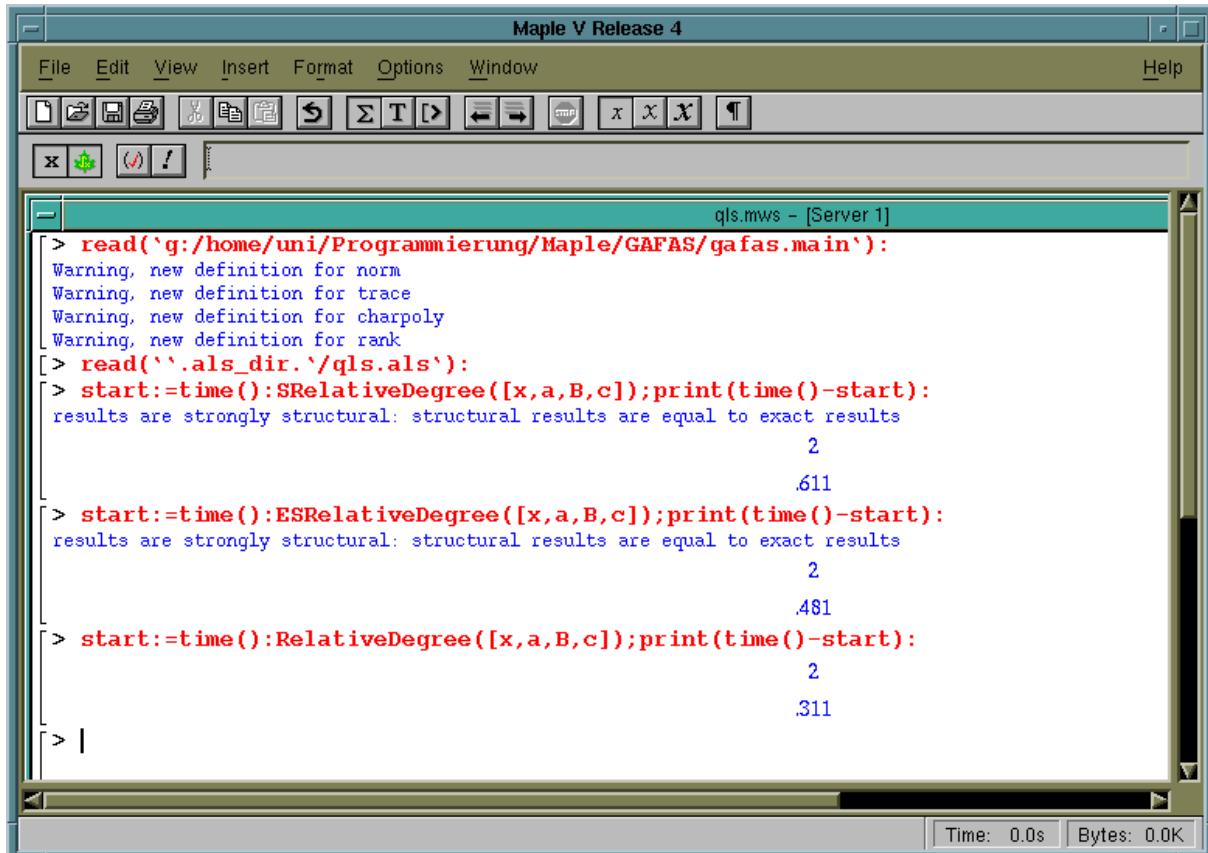
0,
x[1]^2,
x[2]^2,
x[3]^2,
0,
x[5]^2,
x[6]^2,
x[7]^2,
x[8]^2,
x[9]^2,
x[10]^2,
x[1]
]):

B := linalg[matrix](n,m,[
    1, 0,
    0, 0,
    0, 0,
    0, 0,
    0, 1,
    0, 0,
    0, 0,
    0, 0,
    0, 0,
    0, 0,
    0, 0,
    0, 0
]):

c := linalg[vector](p,[
    x[4],
    x[12]
]):
```

Die unterschiedliche Definition von `a`, `c` und `x` als Vektoren und `B` als Matrix im Falle eines Mehrgrößen-ALS muß auch für eigene Systeme eingehalten werden. Eine Definition der Vektoren als Matrizen, was programmtechnisch ebenso möglich wäre, führt zu Fehlern bei der Programmausführung.

Eingelesen werden die Systemgleichungen genau wie die Tool-Box selbst über einen `read()`-Befehl. Bild 3.2 zeigt, wie der Ablauf einer Maple-Sitzung mit dem als `qls.als` hinterlegten System aussehen kann.



The screenshot shows a Maple V Release 4 interface. The title bar reads "Maple V Release 4". The menu bar includes "File", "Edit", "View", "Insert", "Format", "Options", "Window", and "Help". Below the menu is a toolbar with various icons. The main window is titled "qls.mws - [Server 1]". It contains the following Maple session:

```
> read(`q:/home/uni/Programmierung/Maple/GAFAS/gafas.main`):
Warning, new definition for norm
Warning, new definition for trace
Warning, new definition for charpoly
Warning, new definition for rank
[> read(``.als_dir.`qls.als`):
[> start:=time():$RelativeDegree([x,a,B,c]);print(time()-start):
results are strongly structural: structural results are equal to exact results
2
.611
[> start:=time():ESRelativeDegree([x,a,B,c]);print(time()-start):
results are strongly structural: structural results are equal to exact results
2
.481
[> start:=time():RelativeDegree([x,a,B,c]);print(time()-start):
2
.311
[> |
```

The session shows three different methods being compared for their execution times: \$RelativeDegree, ESRelativeDegree, and RelativeDegree. The results indicate that the third method is significantly faster than the first two.

Bild 3.2: Beispiel für die Anwendung von GAFAS

## 4 Implementierte Befehle

### 4.1 Strukturelles System – SSystem

Grundlage einer strukturellen Untersuchung, wie sie gemäß Schritt 1 in Bild 1.1 durchgeführt wird, ist ein *strukturelles System*. Bei diesem Systemmodell sind nur die wechselseitigen Abhängigkeiten der Zustände von Interesse. Physikalische Größen, die ebenfalls die Systemdynamik bestimmen, bleiben in dem Modellierungsansatz unberücksichtigt. Der Vorteil solcher Strukturmodelle liegt darin, daß Systemeigenschaften für eine ganze Klasse von strukturell äquivalenten Systemen Gültigkeit besitzen. Aus einem linearen System

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{c}^T \mathbf{x}(t)\end{aligned}\tag{4.1}$$

wird mit Anwendung der Definition

#### Definition 4.1 (Wend 1993)

Die Strukturmatrix  $\mathbf{M}^*$  ist von gleicher Dimension wie die zugehörige Zahlenmatrix  $\mathbf{M}$ . Für jedes nicht identisch verschwindende Element von  $M$  wird ein „\*“ an der entsprechenden Stelle von  $\mathbf{M}^*$  eingesetzt, jedem Nullelement von  $M$  eine „0“ oder eine Leerstelle in  $\mathbf{M}^*$  zugeordnet.  $\square$

das lineare strukturelle System

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}^*\mathbf{x}(t) + \mathbf{B}^*\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{c}^{*^T} \mathbf{x}(t)\end{aligned}\tag{4.2}$$

Wenn man beachtet, daß bei den hier betrachteten analytischen Systemen mit linear eingehender Steuerung (ALS) Vektorfelder und damit Funktionen in den Zustandsgrößen auftreten, ist eine Übertragung auf nichtlineare Systeme möglich.

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{a}(\mathbf{x}(t)) + \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{c}(\mathbf{x}(t))\end{aligned}\tag{4.3}$$

Die Struktur des Systems ist damit abhängig vom aktuellen Systemzustand. Elemente des Strukturmodells, die mit „\*“ markiert sind, können verschwinden. Besondere Beachtung verdient aus diesem Grunde der Definitionsbereich des Strukturmodells.

Mit dem GAFAS-Befehl `SSystem` werden aus einem ALS die Strukturmatrizen bestimmt, wie das folgende Beispiel demonstriert:

**Beispiel 4.1** Gegeben sei das nichtlineare Zustandsmodell

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} \sin x_2 \\ x_2^2 + e^{x_1} \\ \cos x_2 \end{bmatrix}}_{\mathbf{a}(\mathbf{x})} + \underbrace{\begin{bmatrix} 3x_2 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{b}(\mathbf{x})} u \quad , \quad (4.4)$$

$$y = \underbrace{x_3}_{c(\mathbf{x})} .$$

Zweckmäßigerweise werden die Vektorfelder  $\mathbf{a}(\mathbf{x})$ ,  $\mathbf{b}(\mathbf{x})$  und  $c(\mathbf{x})$ , wie im Abschnitt 3.2 gezeigt, in einer Datei hinterlegt. Den Aufruf und das Ergebnis zeigt der folgende Ausschnitt einer Maple-Sitzung:

```
> read('gafas.main'):
> read('beispiel.als'):
> S:=SSystem([x,a,B,c]);

S := AStructural, BStructural, CStructural
> print(S);


$$\begin{bmatrix} 0 & g_1 & 0 \\ g_2 & g_3 & 0 \\ 0 & g_4 & 0 \end{bmatrix}, \begin{bmatrix} g_5 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & g_6 \end{bmatrix}$$

```

#### Maple-Session 4.1: Anwendung von SSystem

Für eine erweiterte strukturelle Untersuchung werden anstelle des in Definition 4.1 erwähnten Zeichens „\*“ für ein von Null verschiedenes Element generische Größen  $g_i$  verwendet. Die generischen Größen sind zunächst nichts weiter als eine Möglichkeit, die Kanten voneinander zu unterscheiden.  $\square$

## 4.2 Struktureller Graph – SGraph

Für die Anwendung graphentheoretischer Algorithmen findet der bereits in Maple implementierte Objekttyp *graph* Anwendung. Ein Strukturgraph, der im wesentlichen die Verknüpfungen der Systemzustände, der Eingänge und der Ausgänge darstellt, wird mit dem von GAFAS zur Verfügung gestellten Befehl **SGraph** angelegt.

**Beispiel 4.2** Betrachtet wird wieder das mit Gl. (4.4) gegebene System. Bild 4.1 zeigt den zugehörigen (erweiterten) Strukturgraph durch Anwendung der in Abschnitt 2 aufgeführten Regeln zur Bildung eines Systemgraphen. Die Benennung der Kanten mit den generischen

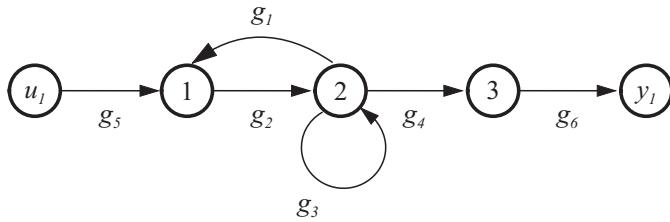


Bild 4.1: Erweiterter Strukturgraph

Größen  $g_1$  bis  $g_6$  kann den (erweiterten) Strukturmatrizen der Maple-Session 4.1 entnommen werden. Als Parameter erwartet der Befehl **SGraph** wiederum eine Liste mit dem Zustandsvektor  $\mathbf{x}$  und den Vektorfeldern  $\mathbf{a}(\mathbf{x})$ ,  $\mathbf{B}(\mathbf{x})$  und  $\mathbf{c}(\mathbf{x})$ . Wie die folgende Maple-Sitzung zeigt, ist das Ergebnis von **SGraph** ein Maple-Objekt vom Typ *Graph*. Zur weiteren Verarbeitung stellt Maple eine Reihe von Befehlen im Rahmen des Standard-Pakets *networks* zur Verfügung. Die in der Maple-Session 4.2 angewendeten **edges**, **vertices** und **ends** stellen nur eine geringe Auswahl dar.

```

> read('gafas.main'):
> read('beispiel.als'):
> G:=SGraph([x,a,B,c]);

G := GStructural
> edges(G);

{g1, g2, g3, g4, g5, g6}

> vertices(G);

{1, 2, 3, y1, u1}

> ends(G);

{[2, 3], [1, 2], [2, 1], [2, 2], [3, y1], [u1, 1]}

```

#### Maple-Session 4.2: Anwendung von **SGraph**

Die Zustandsknoten werden für das Maple-Objekt *Graph* von 1 bis  $n$  durchnumeriert, die Eingänge und Ausgänge erhalten neben der Bezeichnung  $u$  bzw.  $y$  einen Index zur Unterscheidung bei Mehrgrößensystemen. Die Kantengewichte werden mit **SGraph** nicht weiter berechnet und haben alle zunächst den Wert  $w(e_i) = 1$ .  $\square$

### 4.3 Struktureller Rang – SRank

Ausgehend von der Definition des Ranges einer Matrix

**Definition 4.2** (Bronstein und Semendjajew 1989)

Die Matrix  $\mathbf{A} \neq \mathbf{0}$  hat den *Rang*  $\text{Rg}(\mathbf{A}) = \varrho$  genau dann, wenn  $\mathbf{A}$  mindestens eine reguläre  $\varrho$ -reihige Untermatrix besitzt und alle höherreihigen Untermatrizen von  $\mathbf{A}$  singulär sind. Dabei heißt eine quadratische Matrix *regulär* bzw. *singulär* je nachdem, ob ihre Determinante von Null verschieden bzw. gleich Null ist.  $\square$

kann der Rang als die Maximalzahl der linear unabhängigen Zeilen- bzw. Spaltenvektoren der Matrix interpretiert werden. Übertragen auf eine Strukturmatrix  $\mathbf{A}^*$ , die keine festen Zahlenwerte enthält, wird der strukturelle Rang als ein oberer Wert für den Rang aller Matrizen mit gleicher Struktur definiert:

**Definition 4.3** (Wend 1993)

Der strukturelle Rang einer Strukturmatrix ist der maximal mögliche Rang, den die zulässigen Zahlenmatrizen  $\mathbf{A}$  mit der durch  $\mathbf{A}^*$  vorgegebenen Struktur annehmen können:

$$s\text{-Rang } \mathbf{A}^* = \max_{\mathbf{A} \in \mathbf{A}^*} \text{Rang } \mathbf{A} . \quad (4.5)$$

$\square$

Mit dem GAFAS-Befehl `SRank` kann der strukturelle Rang bestimmt werden, wie die folgende Maple-Sitzung zeigt:

```
> read('gafas.main'):
> A:=matrix(2,2,[1,3,2,6]);

A := 
$$\begin{bmatrix} 1 & 3 \\ 2 & 6 \end{bmatrix}$$


> SRank(A);

2

> linalg[rank](A);

1
```

#### Maple-Session 4.3: Anwendung von `SRank`

Der Vergleich mit dem gewöhnlichen Rangbefehl zeigt deutlich die erwähnte Eigenschaft des strukturellen Ranges, ein Supremum zu sein.

## 4.4 Differenzengrad – (E)SRelativeDegree

Der *relative Grad* oder auch *Differenzengrad* eines nichtlinearen Systems spielt bei der Linearisierung und auch Entkopplung eine entscheidende Rolle. Unter anderem kann anhand des Differenzengrads festgestellt werden, ob ein System eine Nulldynamik aufweist und mit welchen Regelgesetzen eine Linearisierung durchgeführt werden kann. Für ALS wird der Differenzengrad folgendermaßen festgelegt:

**Definition 4.4** (Schwarz 1991)

Ein Mehrgrößen-ALS der Form (3.1) hat den (Vektor-) Differenzengrad  $\mathbf{d}$  für alle  $\mathbf{x}$  in einer Umgebung von  $\mathbf{x}_0$ :

$$\mathbf{d} = \{d_1, d_2, \dots, d_m\} \quad (4.6)$$

mit<sup>3</sup>

$$d_i = \min\{r | L_{\mathbf{B}} L_a^{r-1} \mathbf{c}_i^T \mathbf{x} \neq \mathbf{0}\} \quad ; \quad i = 1, 2, \dots, m \quad . \quad (4.7)$$

□

Diese Definition macht Gebrauch von sogenannten *Lie-Operatoren*  $L$ , die in (Schwarz 1991, Isidori 1995) ausführlich beschrieben sind. Mit dem Programmmpaket NSAS – nonlinear systems analysis and synthesis – steht eine MAPLE-Anwendung zur Verfügung, die eine einfache Nutzung von Lie-Ableitungen und weiteren differentialgeometrischen Werkzeugen wie Lie-Klammern etc. ermöglicht (Lemmen u. a. 1995).

Anschaulich bedeutet Gl. (4.7), daß  $d_i$  gerade der Anzahl der zeitlichen Differentiationen von  $y_i$  entspricht, bis mindestens eine Eingangsgröße  $u_j$  explizit darin enthalten ist:

$$\begin{aligned} y_i^{(k)}(t) &= \frac{d^k y_i(t)}{dt^k} \neq f(\mathbf{u}(t)) \quad ; \quad k = 1, \dots, d_i - 1 \\ y_i^{(d_i)}(t) &= \frac{d^{d_i} y_i(t)}{dt^{d_i}} = f(\mathbf{u}(t)) \quad . \end{aligned} \quad (4.8)$$

Für  $\Sigma_{\text{ALS}}$  wird nun der Differenzengrad mit den in Tabelle 4.1 genannten Befehlen berechnet.

Befehl	Argumente	Resultat	Beschreibung
SRelativeDegree	$\Sigma_{\text{ALS}}$ ODER $\mathcal{G}^*$	$\mathbf{d}^*$ wahr/möglich für $\mathbf{d} = \mathbf{d}^*$	Gl. (4.7)
ESRelativeDegree	$\Sigma_{\text{ALS}}$ UND $\mathcal{G}^*$ $\mathcal{G}^*$ optional	$\mathbf{d}^*$ wahr/möglich für $\mathbf{d} = \mathbf{d}^{*e}$	

**Tabelle 4.1:** MAPLE-Befehle zur Bestimmung des Differenzengrads von  $\Sigma_{\text{ALS}}$

<sup>3</sup>  $\mathbf{c}_i^T$  in Gl. (4.7) bezeichnet die  $i$ -te Zeile der Matrix  $\mathbf{C}$ .

Entsprechend dem dreigeteilten Ansatz wird unterschieden zwischen dem strukturellen Differenzengrad  $\mathbf{d}^*$ , dem erweitert strukturellen Differenzengrad  $\mathbf{d}^{*e}$  und dem analytischen Differenzengrad  $\mathbf{d}$ . Der Algorithmus liefert neben dem Ergebnis  $\mathbf{d}^*$  und  $\mathbf{d}^{*e}$  die Information, ob das Resultat mit dem „exakten“ Differenzengrad  $\mathbf{d}$  gemäß Definition 4.4 übereinstimmt.

**Beispiel 4.3** Die analytische Berechnung des Differenzengrades für das System nach Gl. (4.4) durch Ableiten des Systemausgangs ergibt den Wert  $d = 3$ . Die Maple-Session 4.3 zeigt die alternative Vorgehensweise unter Verwendung von GAFAS-Befehlen:

```
> read('gafas.main'):
> read('beispiel.als'):
> G:=SGraph([x,a,B,c]);

G := GStructural
> SRelativeDegree(G);

results are strongly structural: structural results are equal to exact
results

3
> ESRelativeDegree([x,a,B,c],G);

results are strongly structural: structural results are equal to exact
results

3
```

#### Maple-Session 4.4: Anwendung von (E)SRelativeDegree

Mit dem Hinweis „Die strukturellen Ergebnisse entsprechen den exakten Ergebnissen“, nach der Anwendung von `SRelativeDegree`, erübrigkt sich die Anwendung von `ESRelativeDegree` und erfolgt hier nur zu Demonstrationszwecken. Dabei reduziert die optionale Übergabe des strukturellen Graphen die Rechenzeit, die dafür erforderlich ist. □

Der Algorithmus zur Bestimmung der strukturellen Differenzengrade basiert auf der Analyse des Systemgraphen. Es wird die Existenz des kürzesten Pfades zwischen den Eingangsgrößen  $\mathbf{u}$  – genauer den diesen zugeordneten Knoten – und dem Ausgangsknoten  $y_i$  geprüft und dessen Länge  $l_i$  ermittelt. Für den Differenzengrad  $d_i$  gilt dann

$$d_i = l_i - 2 \quad , \quad (4.9)$$

Der Subtrahend „-2“ resultiert aus den Verbindungskanten zwischen Eingängen und Zuständen bzw. Zuständen und Ausgängen, die im Gegensatz zu den übrigen Kanten keine Integration repräsentieren. Existiert nur ein kürzester Weg, so gilt  $\mathbf{d}^* = \mathbf{d}$ . Bei mehreren Pfaden

kürzester Länge müssen zusätzlich die Pfadgewichte untersucht werden. Addieren sich alle Pfadgewichte zu einem Wert ungleich Null, dann folgt daraus  $\mathbf{d}^{*e} = \mathbf{d}$ . Sollte auf diese Weise keine exakte Aussage möglich sein, so sind  $\mathbf{d}^*$  und  $\mathbf{d}^{*e}$  dennoch immer obere Abschätzungen für  $\mathbf{d}$ .

## 4.5 Nullstellen im Unendlichen – (E)SZerosAtInfinity

Weitere wesentliche Eigenschaften nichtlinearer Systeme lassen sich anhand der sogenannten *Nullstellenstruktur im Unendlichen* definieren. Es sei darauf hingewiesen, daß dieser Terminus für das Tupel aller Ordnungen der Nullstellen im Unendlichen steht. Der hierin enthaltene Begriff „Struktur“ wurde in Anlehnung an die in der Literatur übliche Benennung gewählt und beinhaltet die Invarianz dieser Kenngröße gegenüber einer Systemänderung durch bestimmte Transformationen und statische Zustandsrückführungen. Er ist in keiner Weise mit den hier untersuchten „strukturellen“ Methoden zu verwechseln. Die Nullstellenstruktur im Unendlichen, die im Falle von  $\Sigma_{LS}$  direkt anhand der Übertragungsmatrix  $\mathbf{F}(s)$  ermittelt wird, hat für nichtlineare Systeme keine derart anschauliche Bedeutung. Dennoch ist sie als eine für die Analyse nichtlinearer Systeme aussagekräftige Größe anzusehen.

### Definition

Als einer der ersten, der die Differentialalgebra im Bereich der nichtlinearen Regelungstheorie verwendete, führte Fliess (1986) eine *algebraische* Definition für die Nullstellen im Unendlichen (im folgenden mit NU bezeichnet) von nichtlinearen Systemen ein. Diese wird anhand von differentiellen Vektorräumen erstellt, welche aus den *Kähler-Differentialen* der Ein- und Ausgangsgrößen resultieren. Im Gegensatz zu differentialgeometrisch definierten NU verfügt die algebraische Darstellung über eine Reihe von Vorteilen. So ist sie z. B. *global* über dem Zustandsraum eines Systems gültig und in ihren Eigenschaften konsistent mit bekannten Eigenschaften der unendlichen Nullstellen linearer Systeme.

Geht man von einem  $\Sigma_{ALS}$ -Zustandsmodell der Form (4.3) aus, lassen sich zunächst die

zeitlichen Ableitungen der Ausgangsgrößen durch

$$\begin{aligned}
 \dot{\mathbf{y}}(t) &= \dot{\mathbf{y}}(\mathbf{x}, \mathbf{u}) \\
 &= \frac{\partial \mathbf{y}}{\partial \mathbf{x}} [\mathbf{a}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u}] \\
 \ddot{\mathbf{y}}(t) &= \ddot{\mathbf{y}}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}) \\
 &= \frac{\partial \dot{\mathbf{y}}}{\partial \mathbf{x}} [\mathbf{a}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u}] + \frac{\partial \dot{\mathbf{y}}}{\partial \mathbf{u}} \dot{\mathbf{u}} \\
 &\vdots \\
 \mathbf{y}^{(k+1)}(t) &= \mathbf{y}^{(k+1)}(\mathbf{x}, \mathbf{u}, \dots, \mathbf{u}^{(k)}) \\
 &= \frac{\partial \mathbf{y}^{(k)}}{\partial \mathbf{x}} [\mathbf{a}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u}] + \sum_{l=0}^{k-1} \frac{\partial \mathbf{y}^{(k)}}{\partial \mathbf{u}^{(l)}} \mathbf{u}^{(l+1)}
 \end{aligned} \tag{4.10}$$

allgemein beschreiben. Soll auf die Betrachtung von reinen Ein-/Ausgangsdarstellungen verzichtet und die zeitlichen Ableitungen der Ausgangsgrößen in der Form (4.10) differentialalgebraisch beschrieben werden, so reicht der Körper  $K\langle \mathbf{u} \rangle$  hierfür nicht aus. Denn in diesem sind formal keine Funktionen in  $\mathbf{x}(t)$  enthalten, die aber als Koeffizienten der Differentialgleichungen benötigt werden. Deshalb wird der oben eingeführte differentielle Körper  $\bar{K}$  verwendet, der neben  $K\langle \mathbf{u} \rangle$  Koeffizienten enthält, die meromorph in der Variablen  $\mathbf{x}(t)$  sind.

Über  $\bar{K}$  wird anschließend ein differentieller Vektorraum  $\mathcal{U}$  definiert. Dieser ist aufgespannt durch Kähler-Differentiale  $d\mathbf{u}_1, \dots, d\mathbf{u}_m$ . Die Projektionen der Ausgangsgrößen-Differentiale  $d\mathbf{y}^{(k)}$  auf  $\mathcal{U}$  spannen ihrerseits differentielle  $\bar{K}$ -Vektorräume auf, deren differentielle Dimensionen mit  $\sigma_k$  bezeichnet werden. Aus den  $\sigma_k$ , für die der Zusammenhang  $0 = \sigma_0 \leq \sigma_1 \leq \dots \leq \sigma_n \leq \min(m, p)$  immer erfüllt ist, lässt sich direkt die Nullstellenstruktur im Unendlichen eines Systems definieren zu

#### Definition 4.5

Die Differenz  $\sigma_{k+1} - \sigma_k$  mit  $\sigma_k = \text{diff. dim } \text{span}_{\bar{K}} \{d\mathbf{y}^{(k)}\}$  gibt die *Anzahl der Nullstellen im Unendlichen* der Ordnung  $k+1$  an. Das Maximum der ganzzahligen Größen  $\sigma_n$  entspricht der Gesamtanzahl der NU und stimmt mit dem differentiellen Rang  $\rho$  von  $\Sigma$  überein.  $\square$

Um die Auswertung der NU mit herkömmlichen Programmsystemen, d. h. mit klassischen mathematischen Ansätzen, zu ermöglichen, wird nun ein Übergang zur linearen Algebra vollzogen. Gemäß Gl. (4.10) sind sowohl der Ausgangsvektor  $\mathbf{y}$  eines  $\Sigma_{\text{ALS}}$  als auch dessen zeitliche Ableitung Funktionen in  $\mathbf{x}$  und  $\bar{\mathbf{u}}$ . Die Differentiale  $d\bar{\mathbf{y}}$  werden demzufolge in Abhängigkeit der Differentiale  $d\mathbf{x}$  und  $d\bar{\mathbf{u}}$  gebildet:

$$d\bar{y}_i^{(k)} = \left[ \frac{\partial y_i^{(k)}}{\partial \mathbf{u}} \dots \frac{\partial y_i^{(k)}}{\partial \mathbf{u}^{(n-1)}} \right] \begin{bmatrix} d\mathbf{u} \\ \vdots \\ d\mathbf{u}^{(n-1)} \end{bmatrix} + \frac{\partial y_i^{(k)}}{\partial \mathbf{x}} d\mathbf{x} \quad ; \quad i = 1, \dots, p . \tag{4.11}$$

Jedes Element von  $d\bar{\mathbf{y}}$  kann als Vektor im (nichtdifferentiellen) Vektorraum  $\mathcal{E} = \mathcal{U} \oplus \mathcal{X}$  interpretiert werden, wobei  $\mathcal{U}$  durch die  $d\bar{\mathbf{u}}$  und  $\mathcal{X}$  durch die  $d\mathbf{x}$  aufgespannt wird. Der von der Wahl des Zustandsvektors  $\mathbf{x}$  unabhängige differentielle Rang  $\rho = \text{diff. trg } K\langle \mathbf{y} \rangle / K$  eines Systems entspricht der Anzahl von differentiell linear unabhängigen Elementen  $d\mathbf{y}$  bzw. der differentiellen Dimension des durch  $d\mathbf{y}$  aufgespannten Vektorraums. Der Übergang auf (nichtdifferentiell) linear unabhängige Größen erfolgt durch die sukzessiv von  $d\bar{\mathbf{y}}$  aufgespannten Vektorräume

$$\begin{aligned}\mathcal{E}_0 &= \text{span}_{\bar{K}} \{d\mathbf{x}\} \\ \mathcal{E}_1 &= \text{span}_{\bar{K}} \{d\mathbf{x}, d\dot{\mathbf{y}}\} \\ &\vdots \\ \mathcal{E}_n &= \text{span}_{\bar{K}} \{d\mathbf{x}, d\dot{\mathbf{y}}, \dots, d\mathbf{y}^{(n)}\} \quad .\end{aligned}\tag{4.12}$$

Diese stellen Unterräume von  $\mathcal{E}$  dar:

$$\mathcal{E}_0 \subset \mathcal{E}_1 \subset \dots \subset \mathcal{E}_n \subset \mathcal{E} = \text{span}_{\bar{K}} \{d\mathbf{x}, d\bar{\mathbf{u}}\} \quad .\tag{4.13}$$

Die Differenzen ihrer (nichtdifferentiellen) Dimensionen  $\rho_k = \dim \mathcal{E}_k$  entsprechen einer monoton steigenden Folge, die gegen den differentiellen Rang  $\rho$  konvergiert und für  $k \geq n$  diesem entspricht:

$$\rho_{k+1} - \rho_k \geq \rho_k - \rho_{k-1} \quad \forall k > 0\tag{4.14}$$

$$\rho = \rho_{k+1} - \rho_k = \rho_n - \rho_{n-1} \quad \forall k \geq n \quad .\tag{4.15}$$

Eine an Definition 4.5 anknüpfende Bestimmung der NU resultiert dann aus

#### Definition 4.6

Die Anzahl  $\sigma_k$  der Nullstellen im Unendlichen der Ordnung kleiner oder gleich  $k$ ,  $k \geq 1$ , entspricht  $\sigma_k = \dim \mathcal{E}_k - \dim \mathcal{E}_{k-1}$ . Setzt man  $\sigma_{\leq 0} := 0$ , so ergibt sich die Anzahl der Nullstellen im Unendlichen einer Ordnung  $k$  aus  $\sigma_k - \sigma_{k-1}$ . Die Struktur im Unendlichen wird durch das geordnete Tupel  $\{n_1, n_2, \dots, n_{\sigma_n}\}$  der Indizes  $k$  beschrieben, für die  $\sigma_k - \sigma_{k-1} \neq 0$  gilt. Die Indizes werden jeweils  $(\sigma_k - \sigma_{k-1})$ -mal wiederholt.  $\square$

Die Tupel  $\{\rho_1, \dots, \rho_n\}$ ,  $\{\sigma_1, \dots, \sigma_n\}$  und  $\{n_1, \dots, n_{\sigma_n}\}$  beinhalten im Grunde identische Informationen über die Nullstellen im Unendlichen. In Abhängigkeit des jeweiligen Anwendungsfalls kann sich eine der Darstellungsformen jedoch als vorteilhafter erweisen als die anderen. Soweit möglich, wird im weiteren das Tupel der Ordnungen  $\{n_1, \dots, n_{\sigma_n}\}$  verwendet.

Im Falle von  $\Sigma_{\text{LS}}$  ist die Struktur im Unendlichen eng mit dem Begriff *Toepplitz-Matrizen*

verknüpft. Das nichtlineare Äquivalent der Toeplitz-Matrix sind die Jacobi-Matrizen

$$\begin{aligned} \mathbf{J}_k = \mathbf{J}_k(\mathbf{x}, \mathbf{u}, \dots, \mathbf{u}^{(k-1)}) &= \frac{\partial(\dot{\mathbf{y}}, \dots, \mathbf{y}^{(k)})}{\partial(\mathbf{u}, \dots, \mathbf{u}^{(k-1)})} ; \quad k = 1, \dots, n ; \quad \mathbf{J}_0 = \mathbf{0} \\ &= \begin{bmatrix} \frac{\partial \dot{\mathbf{y}}}{\partial \mathbf{u}} & \mathbf{0} \\ \frac{\partial \ddot{\mathbf{y}}}{\partial \mathbf{u}} & \frac{\partial \ddot{\mathbf{y}}}{\partial \dot{\mathbf{u}}} \\ \vdots & \ddots \\ \frac{\partial \mathbf{y}^{(k)}}{\partial \mathbf{u}} & \frac{\partial \mathbf{y}^{(k)}}{\partial \dot{\mathbf{u}}} \dots \frac{\partial \mathbf{y}^{(k)}}{\partial \mathbf{u}^{(k-1)}} \end{bmatrix} . \end{aligned} \quad (4.16)$$

Der rekursive Charakter der Bestimmungsgleichung bei wachsendem  $k$ , der vor allem für die spätere Implementierung als Algorithmus von Interesse ist, wird in der Form

$$\mathbf{J}_k = \begin{bmatrix} \mathbf{J}_{k-1} & \mathbf{0} \\ \mathbf{Y}_k & \end{bmatrix} \quad \text{mit} \quad \mathbf{Y}_k = \left[ \frac{\partial \mathbf{y}^{(k)}}{\partial \mathbf{u}} \frac{\partial \mathbf{y}^{(k)}}{\partial \dot{\mathbf{u}}} \dots \frac{\partial \mathbf{y}^{(k)}}{\partial \mathbf{u}^{(k-1)}} \right] \quad (4.17)$$

besonders deutlich. Ein Vergleich der Elemente von  $\mathbf{J}_k$  mit Gl. (4.11) führt zu einer direkten Beziehung zwischen der Dimension eines Vektorraums  $\mathcal{E}_k$  und dem Rang der Matrix  $\mathbf{J}_k$ , wenn  $\mathcal{E}_k$  folgendermaßen aufgespalten wird:

$$\begin{aligned} \mathcal{E}_k &= \text{span}_{\bar{K}} \left\{ \frac{\partial \dot{\mathbf{y}}}{\partial \mathbf{u}} \mathbf{d}\mathbf{u}, \dots, \sum_{l=0}^{k-1} \frac{\partial \mathbf{y}^{(k)}}{\partial \mathbf{u}^{(l)}} \mathbf{d}\mathbf{u}^{(l)} \right\} \oplus \text{span}_{\bar{K}} \{ \mathbf{d}\mathbf{x} \} \\ &= \mathcal{V}_k \oplus \text{span}_{\bar{K}} \{ \mathbf{d}\mathbf{x} \} . \end{aligned} \quad (4.18)$$

Dann können auch die Dimensionen derart zerlegt werden, daß die Zeilen von  $\mathbf{J}_k$  gerade den aufspannenden Vektoren entsprechen:

$$\begin{aligned} \dim \mathcal{E}_k &= \dim \text{span}_{\bar{K}} \left\{ \frac{\partial \dot{\mathbf{y}}}{\partial \mathbf{u}} \mathbf{d}\mathbf{u}, \dots, \sum_{l=0}^{k-1} \frac{\partial \mathbf{y}^{(k)}}{\partial \mathbf{u}^{(l)}} \mathbf{d}\mathbf{u}^{(l)} \right\} + \dim \text{span}_{\bar{K}} \{ \mathbf{d}\mathbf{x} \} \\ &= \text{rang } \mathbf{J}_k + n ; \quad k = 1, \dots, n . \end{aligned} \quad (4.19)$$

#### Beispiel 4.4

Die Matrizen

$$\mathbf{a}(\mathbf{x}) = \begin{bmatrix} 0 \\ x_1^2 \\ 0 \end{bmatrix} ; \quad \mathbf{B}(\mathbf{x}) = \begin{bmatrix} 1 & 0 \\ x_3 & 0 \\ 0 & 1 \end{bmatrix} ; \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.20)$$

beschreiben das gegebene  $\Sigma_{\text{ALS}}$ . Durch Anwendung von Definition 4.5 kann festgestellt werden, daß das System rechtsinvertierbar und damit entkoppelbar ist:

$$\begin{aligned} \dot{\mathbf{y}} &= \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ x_1^2 + x_3 u_1 \end{bmatrix} \\ \Rightarrow \sigma_1 &= \text{rang } \mathbf{J}_1 = \text{rang} \left[ \frac{\partial \dot{\mathbf{y}}}{\partial \mathbf{u}} \right] = \text{rang} \begin{bmatrix} 1 & 0 \\ x_3 & 0 \end{bmatrix} = 1 . \end{aligned} \quad (4.21)$$

Die Differenz  $\sigma_1 - \sigma_0$  erreicht noch nicht den größtmöglichen Rang, deshalb sind weitere Differentiationen notwendig:

$$\begin{aligned}\ddot{\mathbf{y}} &= \begin{bmatrix} \dot{u}_1 \\ 2x_1u_1 + u_2\dot{u}_1 + x_3\dot{u}_1 \end{bmatrix} \\ \Rightarrow \frac{\partial \ddot{\mathbf{y}}}{\partial \mathbf{u}} &= \begin{bmatrix} 0 & 0 \\ 2x_1 + u_2 & u_1 \end{bmatrix} ; \quad \frac{\partial \ddot{\mathbf{y}}}{\partial \dot{\mathbf{u}}} = \begin{bmatrix} 1 & 0 \\ x_3 & 0 \end{bmatrix} \quad (4.22) \\ \Rightarrow \sigma_2 &= \text{rang} \begin{bmatrix} \frac{\partial \ddot{\mathbf{y}}}{\partial \mathbf{u}} & \mathbf{0} \\ \frac{\partial \ddot{\mathbf{y}}}{\partial \dot{\mathbf{u}}} & \frac{\partial \ddot{\mathbf{y}}}{\partial \dot{\mathbf{u}}} \end{bmatrix} - \text{rang } \mathbf{J}_1 = 2 .\end{aligned}$$

Wegen Gl. (4.14) kann  $\sigma_3$  nicht kleiner als  $\sigma_2$  werden, so daß für den Rang  $\rho = \sigma_3 = \sigma_2 = 2$  gilt. Folglich ist das System entkoppelbar.  $\square$

Damit steht eine auf einfache Weise in symbolischen Programmiersprachen zu implementierende Beschreibungsform für die NU zur Verfügung. Für einen effizienten Algorithmus zur Ermittlung der Struktur im Unendlichen ist die Begrenzung

$$\begin{aligned}\sigma_k &= \dim \mathcal{E}_k - \dim \mathcal{E}_{k-1} \\ &= \text{rang } \mathbf{J}_k - \text{rang } \mathbf{J}_{k-1} \leq \min(m, p) \quad \forall k \in \mathbb{N}^+ \quad (4.23)\end{aligned}$$

von Interesse. Sobald ein  $\sigma_k$  in Gl. (4.23) erstmalig gleich dem Minimum der Anzahl von Ein- und Ausgängen ist, kann die Berechnung abgebrochen werden. Die weiteren Nullstellenanzahlen erfüllen dann  $\sigma_n = \dots = \sigma_k$ . Insbesondere bei Mehrgrößensystemen höherer Ordnungen kann der Einsatz des Abbruchkriteriums eine erhebliche Reduzierung des erforderlichen Rechenaufwandes bewirken.

Der Rechenaufwand steigt exponentiell mit  $k$  an, so daß bereits für Nullstellen-Ordnungen zwischen 6 und 10 die Rangbestimmung der Jacobi-Matrizen (4.16) kein Ergebnis liefert. In diesem Fall bietet sich die strukturelle Vorgehensweise an, die in MAPLE mit den in Tabelle 4.2 genannten Befehlen aufgerufen wird.

Befehl	Argumente	Resultat	Beschreibung
SZerosAtInfinity	$\Sigma_{\text{ALS}}$ ODER $\mathcal{G}^*$	$\{n_1^*, \dots, n_\rho^*\}$ wahr/möglich für $n_i = n_i^*$	Def. (4.6)
ESZerosAtInfinity	$\Sigma_{\text{ALS}}$ UND $\mathcal{G}^*$ $\mathcal{G}^*$ optional	$\{n_1^{*e}, \dots, n_\rho^{*e}\}$ wahr/möglich für $n_i = n_i^{*e}$	

**Tabelle 4.2:** MAPLE-Befehle zur Bestimmung der Nullstellen im Unendlichen von  $\Sigma_{\text{ALS}}$

### Algorithmus

Auch hier wird für die strukturelle Untersuchung der Systemgraph verwendet. Im Gegensatz zur Bestimmung des Differenzengrads muß jedoch nicht nur ein Ein-/Ausgangspfad

berechnet werden. Vielmehr werden mehrere Ein-/Ausgangswege gesucht, die *knotendisjunkt* sind, also keine gemeinsamen Knoten – und damit auch Kanten – aufweisen. Aus den Pfadlängen kann dann die Struktur im Unendlichen ermittelt werden. Einer der effizientesten Algorithmen zur sukzessiven Ermittlung solcher knotendisjunkter Pfade greift auf den *inkrementalen Graphen* zurück. Hierbei ist zu betonen, daß es sich um einen ausschließlich graphentheoretisch orientierten Algorithmus handelt, der in keiner Weise mit dem ursprünglichen regelungstechnischen Problem verknüpft ist.

#### Algorithmus 4.1

Ordnungen der NU mittels knotendisjunkter Ein-/Ausgangspfade bestimmen

1. Eingabe:

- Graph  $\mathcal{G} = [V_{\mathcal{G}}, E_{\mathcal{G}}]$

2. Setze:

- Eingangsknoten:  $U = \{u_1, \dots, u_m\} \in V_{\mathcal{G}}$
- Ausgangsknoten:  $Y = \{y_1, \dots, y_p\} \in V_{\mathcal{G}}$
- Zustandsknoten:  $X = \{1, 2, \dots, n\} \in V_{\mathcal{G}}$

3. Graphen  $\bar{\mathcal{G}}$  festlegen:

- Knotenmenge:

Ein-/Ausgangsknoten von  $\mathcal{G}$  übernehmen und Zustandsknoten verdoppeln:

$$\bar{V}_{\mathcal{G}} = U \cup Y \cup \{1', 2', \dots, n'\} \cup \{1'', 2'', \dots, n''\} = \bar{U} \cup \bar{Y} \cup \bar{X}$$

mit  $\bar{U} = U$ ,  $\bar{Y} = Y$  und  $\bar{X} = \bar{X}' \cup \bar{X}''$

- Kantenmenge:

$$\begin{aligned} \bar{E}_{\mathcal{G}} = & \{(i, j') | i \in \bar{U}, j' \in \bar{X}', (i, j) \in E_{\mathcal{G}}\} \\ & \cup \{(i'', j) | i'' \in \bar{X}'', j \in \bar{Y}, (i, j) \in E_{\mathcal{G}}\} \\ & \cup \{(i', i'') | i' \in \bar{X}', i'' \in \bar{X}''\} \end{aligned}$$

4. Initialisierung:  $k = 0$ ,  $L_0 = 0$

5. Kürzesten Ein-/Ausgangspfad  $\{P : u_i \rightarrow y_j | u_i \in \bar{U}, y_j \in \bar{Y}\}$  in  $\bar{\mathcal{G}}$  finden und streng strukturelles Verhalten prüfen

→ angepaßter Moore-Algorithmus (Wey 1996, S. 96)

6. Wenn  $P$  nicht existiert, dann gehe zu 13.

7.  $k := k + 1$

8. Länge  $l_P$  des Pfads  $P$  bestimmen, wobei doppelte Zustandsknoten Berücksichtigung finden:

$$l_P = \frac{l_{u_i y_j} + 1}{2}$$

9. Ein- und Ausgangsknoten des Pfads  $P$  nicht mehr für weitere knotendisjunkte Pfade verwenden:

$$\bar{U} := \bar{U} \setminus \{u_i\} \quad ; \quad \bar{Y} := \bar{Y} \setminus \{y_j\}$$

10. Ordnungen der Nullstellen im Unendlichen anhand der Länge  $l_P$  des Pfads  $P$  bestimmen:

$$L_k = L_{k-1} + l_P \quad ; \quad n_k^* = \begin{cases} L_1 - 1 & \text{für } k = 1 \\ L_k - k - \sum_{j=1}^{k-1} n_j^* & \text{für } k \geq 2 \end{cases}$$

11. Richtungen aller Kanten  $e \in P$  umkehren und deren Länge invertieren:

$$\{\bar{E}_G := \bar{E}_G \setminus \{(i, j)\} \cup \{(j, i)\}, l_{ji} = -l_{ij} \mid \forall (i, j) \in P\}$$

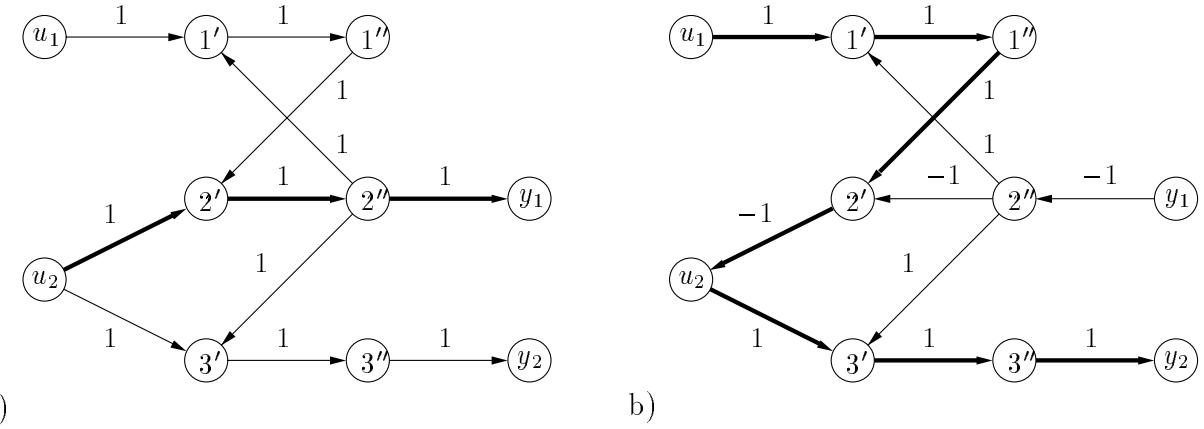
12. Wenn  $k < \min(m, p)$ , dann gehe zu 5.

13. Ausgabe:

- Ordnungen der Nullstellen im Unendlichen:

$$\{n_1^*, n_2^*, \dots, n_k^*\}$$

- Das System verhält sich (nicht) streng strukturell □



**Bild 4.2:** Kürzeste Ein-/Ausgangswege im inkrementalen Graph  $\bar{G}$

Den zentralen Gedanken, welcher auf einer Richtungsumkehr von Kanten in Verbindung mit der Invertierung ihrer Längen beruht (vgl. Punkt 11 in Algorithmus 4.1), verdeutlicht Bild 4.2. Diese Richtungsumkehr wird allein zur Abarbeitung des graphentheoretischen Algorithmus benötigt, besitzt aber *keine* physikalische Bedeutung für den ursprünglichen Systemgraphen bzw. das nichtlineare System. Für den Graphen wird zunächst der Ein-/Ausgangsweg  $u_2 \rightarrow 2' \rightarrow 2'' \rightarrow y_1$  (Bild 4.2a) mit kürzester Länge 3 gefunden. Durch Umkehr der verwendeten Kanten und Streichen von bereits verwendeten Ein- und Ausgangsknoten existiert im zweiten Schritt wieder ein kürzester Ein-/Ausgangsweg (Bild 4.2b)

$$u_1 \rightarrow 1' \rightarrow 1'' \rightarrow 2' \rightarrow u_2 \rightarrow 3' \rightarrow 3'' \rightarrow y_2 , \quad (4.24)$$

der aufgrund einer negativen Kantenlänge die Gesamtlänge 5 aufweist. Bei Berücksichtigung doppelter Zustandsknoten können aus den Längen dieser Wege die Ordnungen der NU zu  $\{1, 2\}$  berechnet werden. Zum Auffinden des jeweilig kürzesten Wegs zwischen einem der Ein- und einem der Ausgangsknoten wird ein speziell angepaßter Moore-Algorithmus (Gondran und Minoux 1986, S. 51) benutzt, der in der Lage ist, negative Kantenlängen zu berücksichtigen. Daß die Suche nach Wegen in  $\bar{\mathcal{G}}$  kein eindeutiges Problem darstellt, wird an der Existenz einer weiteren Kombination kürzester Ein-/Ausgangswege

$$u_2 \rightarrow 3' \rightarrow 3'' \rightarrow y_2 \quad \text{und} \quad u_1 \rightarrow 1' \rightarrow 1'' \rightarrow 2' \rightarrow 2'' \rightarrow y_1 \quad (4.25)$$

in Bild 4.2 deutlich. Die kürzesten Gesamtlängen dieser Wege sind jedoch identisch mit den zuvor bestimmten.

Das bisher aufgeführte Verfahren bestimmt Kenngrößen struktureller Natur, da keinerlei Kantenbewertungen Berücksichtigung finden. Allerdings ist ohne hohen Aufwand anhand des inkrementalen Graphen zu prüfen, ob mehrere Möglichkeiten für knotendisjunkte Wege bestehen und ob diesen dieselben Bewertungen zugeordnet sind. Denn es kann nur bei einer gegenseitigen Elimination von Wegbewertungen zur Abweichung von strukturellen und exakten Kenngrößen kommen. Zur Berücksichtigung dessen ist Algorithmus 4.1 eine Prüfung auf identische Wegbewertungen hinzuzufügen:

### Algorithmus 4.2

Ordnungen der NU mit knotendisjunkten Ein-/Ausgangswegen in  $\Sigma_{*e}$  bestimmen

Algorithmus 4.1, wobei Punkt 5 ersetzt wird durch

5. Kürzeste Ein-/Ausgangswege  $P_h$  in  $\bar{\mathcal{G}}$  finden und streng strukturelles Verhalten prüfen:
  - angepaßter Moore-Algorithmus (Wey 1996, S. 97)
- 5a. Wenn  $h > 1$ , dann Bewertungen aller Kanten  $\{(i, j) | (i, j) \in P_h; h \geq 1\}$  mittels Differenziationen 1. Ordnung bestimmen
- 5b. Wegbewertungen für alle  $P_h$  bestimmen und auf Identität prüfen. Falls identisch, dann Ausgabe von
  - $n_k^* \neq n_k$
  - ab Schritt  $k$  ist eine Übereinstimmung von strukturellen und exakten Ordnungen der NU nicht gewährleistet
- 5c. Weiterrechnen mit einem der Wege:

$$P = P_1$$

□

Zur Realisierung dieser Erweiterung ist ein Verfahren ähnlich dem Moore-Algorithmus notwendig, welches nicht nur einen, sondern alle Ein-/Ausgangswege einer kürzesten Länge ermittelt. Die Kantenbewertungen in Schritt 5a entsprechen für in ihrer Richtung umgekehrte Kanten dem negativen Kehrwert der ursprünglichen Bewertung.

Der Vorteil der strukturellen Vorgehensweise bei der Bestimmung der Nullstellen im Unendlichen kann schon für einfache Klassen nichtlinearer Systeme nachgewiesen werden. Betrachtet man z. B. einfache quadratische Systeme (QLS) mit den Matrizen

$$\begin{aligned}
 \dot{x}_1 &= u_1 \\
 \dot{x}_5 &= u_2 \\
 \Sigma_{\text{QLS}}^a \quad \dot{x}_i &= x_{i-1}^2 \quad \forall i = 2, 3, 4, 6, \dots, 12 \\
 \dot{x}_{12} &= x_1 \\
 y_1 &= x_4 \\
 y_2 &= x_{12}
 \end{aligned} \tag{4.26}$$

$$\Sigma_{\text{QLS}}^b = \Sigma_{\text{QLS}}^a \text{ mit } \dot{x}_{12} = x_1 ,$$

so ist  $\Sigma_{\text{QLS}}^b$  nicht beobachtbar und degeneriert, d. h. der Systemrang entspricht nicht der minimalen Anzahl 2 von Ein- und Ausgängen. Die Ordnungen der Nullstellen im Unendlichen ergeben sich damit zu  $\{2, \infty\}$ . Vergleicht man alle drei Ebenen – strukturelle, erweitert strukturelle und analytische –, so sind signifikante Unterschiede in den Rechenzeiten<sup>4</sup> der Algorithmen erkennbar (vgl. Tabelle 4.3). Der Aufruf und die Ergebnisse

**Tabelle 4.3:** Rechenzeiten für die Bestimmung der Nullstellen im Unendlichen

Algorithmus	System	
	$\Sigma_{\text{QLS}}^a$	$\Sigma_{\text{QLS}}^b$
SZerosAtInfinity	1.71 s	1.62 s
ESZerosAtInfinity	1.87 s	1.79 s
ZerosAtInfinity	8.26 s	66.25 s

der verschiedenen in GAFAS implementierten Algorithmen sind in der Maple-Session 4.5 wiedergegeben. Insbesondere stellt sowohl der strukturelle als auch der erweitert strukturelle Ansatz die Übereinstimmung der Ergebnisse mit den gemäß Definition 4.6 richtigen Resultaten fest.

---

<sup>4</sup> Alle Ergebnisse wurden auf einem Pentium 150MHz mit 80MB Hauptspeicher und MAPLE V RELEASE 4 unter Windows® NT4.0 erzielt.

```

> read('gafas.main'):
> read(`'als_dir.'/qls_b.als`);
> start:=time():SZerosAtInfinity([x,a,B,c]);print(time()-start):

results are strongly structural: structural results are equal to exact
results
[2, ∞]
1.622
> start:=time():ESZerosAtInfinity([x,a,B,c]);print(time()-start):

results are strongly structural: structural results are equal to exact
results
[2, ∞]
1.793
> start:=time():ZerosAtInfinity([x,a,B,c]);print(time()-start):
[2, ∞]
66.246

```

**Maple-Session 4.5:** Bestimmung der Nullstellen im Unendlichen mit GAFAS

## 4.6 Beobachtbarkeit – (E)SObervability

Bei der Entwicklung von Zustandsreglern spielt die Frage nach der Beobachtbarkeit der Zustände eine entscheidende Rolle. Sind alle Zustände des Systems beobachtbar, so lassen sich diese theoretisch mit Hilfe eines Beobachters aus den Eingangssignalen  $\mathbf{u}(t)$ , den Ausgangssignalen  $\mathbf{y}(t)$  und deren zeitlichen Ableitungen ermitteln. Dies führt zur Kostensenkung bei der Meßtechnik bzw. macht die Bestimmung von einigen Zuständen überhaupt erst möglich.

Die Betrachtung der Beobachtbarkeit eines Systems führt auf die Auswertung eines Rangkriteriums (hier für Eingrößensysteme):

$$\text{Rang} \begin{bmatrix} \frac{\partial y}{\partial \mathbf{x}} \\ \frac{\partial \dot{y}}{\partial \mathbf{x}} \\ \vdots \\ \frac{\partial y^{(n-1)}}{\partial \mathbf{x}} \end{bmatrix} = n \quad . \quad (4.27)$$

Dabei wird überprüft, ob zur Berechnung der  $n$  Systemzustände  $n$  linear unabhängige Gleichungen zur Verfügung stehen. Ist das der Fall, so können die Gleichungen eindeutig

nach den gesuchten Zuständen aufgelöst werden. Im Falle eines linearen Systems

$$\sum_{\text{LS}} \begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \quad , \\ y(t) &= \mathbf{c}^T \mathbf{x}(t) \quad , \end{aligned} \quad (4.28)$$

mit  $\mathbf{x}(t) \in \mathbb{R}^n$ ;  $u(t), y(t) \in \mathbb{R}$

führt Gl. (4.27) zum Kalman-Kriterium für Eingrößensysteme:

**Definition 4.7** (Kalman u. a. 1969)

Ein lineares System, gegeben durch das Zustandsmodell gemäß Gl. (4.28), ist vollständig beobachtbar genau dann, wenn gilt

$$\text{Rang} \underbrace{\begin{bmatrix} \mathbf{c}^T \\ \mathbf{c}^T \mathbf{A} \\ \vdots \\ \mathbf{c}^T \mathbf{A}^{n-1} \end{bmatrix}}_Q = n \quad . \quad (4.29)$$

□

Wie in (Spielmann 1996) ausführlich hergeleitet, kann ein Element der Beobachtbarkeitsmatrix  $\mathbf{Q}$  graphentheoretisch durch Multiplikation von Pfadgewichten bestimmt werden. Die  $i$ -te Zeile von  $\mathbf{Q}$  enthält Pfade der Länge  $i$  zum Ausgangsknoten  $y$ . Dabei ist der Start(zustands)knoten durch die Spalte von  $\mathbf{Q}$  festgelegt. Für eine strukturelle Untersuchung reicht zur Bestimmung von  $q_{ij}$  die Information, ob ein Pfad der Länge  $l = i$  vom Zustandsknoten  $j$  zum Ausgang  $y$  existiert. Falls ein Pfad vorhanden ist, hat das Element  $q_{ij}$  den Wert „1“, ansonsten den Wert „0“.

Da die strukturelle Beobachtbarkeitsuntersuchung nur für den Fall, daß ein Rangdefekt auftritt, zu einem abschließendes Ergebnis gelangt, werden für eine erweiterte strukturelle Untersuchung die Kantengewichte explizit eingesetzt. Die resultierende Beobachtbarkeitsmatrix  $\mathbf{Q}^{*e}$  liefert dann das gewünschte Endergebnis. Insbesondere im Falle eines nicht vollständig beobachtbaren Systems ergibt sich ein großer Rechenzeitvorteil im Vergleich zur herkömmlichen analytischen Vorgehensweise, da bereits die strukturelle Untersuchung das Ergebnis liefert. Die Übertragung auf nichtlineare Systeme ist ebenso möglich, wobei die Rangbestimmung durch die in der Beobachtbarkeitsmatrix enthaltenen Veränderlichen aufwendiger ausfällt. Hierzu wurde, wie bereits bei NSAS von Lemmen u. a. (1995), auf die Funktion `extrank` aus der Arbeit von Essen und Jager (1992) zurückgegriffen. Ferner wird die in (Fliess und Glad 1993) dargestellte Erkenntnis, wonach ein nichtlineares System nur dann beobachtbar ist, wenn das zugehörige Tangentialsystem beobachtbar ist, herangezogen.

**Beispiel 4.5** Die Anwendung der zur Beobachtbarkeitsanalyse erforderlichen Befehle zeigt die folgende Maple-Session.

```

> read('gafas.main'):
> read('beispiel.als'):
> G:=SGraph([x,a,B,c]);


$$G := G_{Structural}$$

> SObservability(G);
true
> Erg:=ESObservability([x,a,B,c],G);
the system is structural observable

the tangent system is observable

Erg := Q, true
> print(Erg[1]);

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & -\sin(x_2) & 0 \\ -\sin(x_2) e^{x_1} & -2 \sin(x_2) x_2 & 0 \end{bmatrix}$$

```

#### Maple-Session 4.6: Anwendung von (E)SObservability

Liefert die strukturelle Untersuchung das Ergebnis, daß das System strukturell beobachtbar ist, ist eine erweiterte Untersuchung mit `ESObservability` erforderlich. Rückgabewert ist dabei neben einer ja/nein Entscheidung die Beobachtbarkeitsmatrix  $Q$ . □

## 5 Zusammenfassung und Ausblick

Innerhalb der drei Schritte *Modellbildung*, *Systemanalyse* und *Reglersynthese*, die beim konventionellen Reglerentwurf vorgenommen werden, bietet die in diesem Bericht vorgestellte Maple Toolbox GAFAS Hilfestellung bei der Analyse. Die auf graphentheoretischen Algorithmen basierenden Befehle erlauben durch einen ebenfalls dreigeteilten Ansatz eine systematische und durchsichtige Vorgehensweise. Zunächst kann durch eine strukturelle Untersuchung festgestellt werden, ob Systeme mit der gleichen Struktur wie das zu untersuchende System eine Eigenschaft erfüllen oder nicht. Bei genauerer Kenntnis der Systemzusammenhänge erfolgt in einem zweiten Schritt eine erweiterte strukturelle Untersuchung, bei der die Systemparameter, die als Gewichte in dem zugrundeliegenden Systemgraphen einfließen, Berücksichtigung finden. Die dadurch gewonnenen Aussagen beziehen sich nicht mehr auf eine Klasse von Systemen mit gleicher Struktur, sondern auf das zu untersuchende System. In wenigen Fällen kann die auf dem Tangentialsystem beruhende graphentheoretische Analyse nicht zu einem abschließenden Ergebnis gelangen. Dann ist in einem dritten Schritt die Anwendung herkömmlicher Analyseverfahren, z. B. durch Verwendung der ebenfalls im Fachgebiet erstellten Maple Toolbox NSAS, unumgänglich. Gerade bei komplexen nichtlinearen Systemen und bei Systemen hoher Ordnung (large-scale systems) entsteht ein großer Vorteil durch den dreigeteilten Ansatz, da Aussagen zu Systemeigenschaften oftmals bereits durch eine strukturelle Untersuchung, also beim ersten Schritt, gewonnen werden können.

Die erste Version von GAFAS bietet zwar bereits zu wesentlichen Fragestellungen wie Beobachtbarkeit, Entkoppelbarkeit und Nullstellenstruktur im Unendlichen Einsatzmöglichkeiten. Dennoch steht die Implementierung zu einigen anderen Eigenschaften, die für eine Reglerauslegung von Interesse sind, noch aus. Erwähnenswert sind hier die Steuerbarkeit, die exakte Linearisierbarkeit und damit verbunden die Stabilität der Nulldynamik.

Weiteres Potential besteht neben der Ausweitung auf den Bereich des Reglerentwurfs in der Anwendung komplexer graphentheoretischer Beschreibungen durch Mehrschicht-Graphen (Spielmann 1997) mit dem Ziel einer ausschließlich graphentheoretisch basierten Systemanalyse und Reglersynthese.

## 6 Literatur

- Bronstein, I. N. und K. A. Semendjajew.** 1989. *Taschenbuch der Mathematik*. Frankfurt am Main: Harri Deutsch.
- Essen, H. v. und B. d. Jager.** 1992. Analysis and design of nonlinear control systems with the symbolic computation system maple. *Proc. IFAC Nonlinear Control System Design Symposium*. Bordeaux/Frankreich. 2081–2085.
- Fliess, M.** 1986. Nonlinear control theory and differential algebra. *Modelling and Adaptive Control*, hg. von C. I. Byrnes und A. Kurszanski. *Lecture Notes in Control and Information Science*. 105. Berlin: Springer.
- Fliess, M. und S. T. Glad.** 1993. An algebraic approach to linear and nonlinear control. *Essays on Control: Perspectives in the Theory and its Applications*, hg. von H. L. Trentelmann und J. C. Willems. *Progress in Systems and Control Theory*. 14. 223–267. Boston/USA: Birkhäuser.
- Gondran, M. und M. Minoux.** 1986. *Graphs and algorithms*. Chichester/Großbritannien: Wiley-Interscience.
- Isidori, A.** 1995. *Nonlinear Control Systems*. 3. Auflage. Berlin: Springer.
- Kalman, R. E., P. L. Falb und M. A. Arbib.** 1969. *Topics in Mathematical System Theory*. New York/USA: McGraw-Hill.
- Lemmen, M.** 1996. *Systembeschreibende Graphen – Grundlagen* –. Forschungsbericht 10/96. MSRT. Universität Duisburg.
- Lemmen, M., T. Wey und M. Jelali.** 1995. *NSAS – ein Computer-Algebra-Paket zur Analyse und Synthese nichtlinearer Systeme*. Forschungsbericht 20/95. MSRT. Universität Duisburg.
- Lévine, J.** 1997. A Graph-Theoretic Approach to Input-Output Decoupling and Linearization. *Nonlinear Systems*, hg. von A. J. Fossard und D. Normand-Cyrot. 296–310. London/Großbritannien: Chapman & Hall.
- Reinschke, K.** 1988. *Multivariable Control – A Graph-theoretic Approach*. Berlin: Springer.
- Sachs, H. und M. Stiebitz.** 1987. 250 Jahre Graphentheorie. *NTM-Schriftenreihe Gesch. Naturwissenschaft, Technik, Med.* Leipzig 24(2). 37–62.
- Schwarz, H.** 1991. *Nichtlineare Regelungssysteme – Systemtheoretische Grundlagen*. München: Oldenbourg.

- Spielmann, M.** 1996. *Beobachtbarkeitsanalyse nichtlinearer Systeme mittels Digraphen*. Forschungsbericht 1/96. MSRT. Universität Duisburg.
- Spielmann, M.** 1997. *Mehrschicht-Graphen zur Analyse nichtlinearer Systeme und zur Generierung nichtlinearer Regelgesetze*. Forschungsbericht 2/97. MSRT. Universität Duisburg.
- Spielmann, M.** und **H. Schwarz**. 1996. A graph theoretical method for analyzing the observability of nonlinear systems. *CSC (Circuits, Systems and Computers)*, 15.-17.7.1996. Hellenic Naval Academy, Athen/Griechenland. 611–614.
- Spielmann, M., T. Wey** und **H. Schwarz**. 1996. Transformation of nonlinear systems into Isidori-Byrnes normal form via graph-theoretical methods. *Systems and Networks: Mathematical Theory and Applications (Proc. MTNS'96)*. Saint Louis/USA. (vorgestellter Beitrag).
- Svaricek, F.** 1993. Algebraische Methoden zur Analyse und Synthese nichtlinearer Regelungssysteme. *VDI/VDE-GMA Aussprachetag „Nichtlineare Regelung – Methoden, Werkzeuge, Anwendungen“*. VDI-Bericht. 1026. Düsseldorf: VDI.
- Wagner, K.** und **R. Bodendiek**. 1989. *Graphentheorie I. Anwendung auf Topologie, Gruppentheorie und Verbandstheorie*. Mannheim: Wissenschaftsverlag.
- Wagner, K.** und **R. Bodendiek**. 1990. *Graphentheorie II. Weitere Methoden, Massen-Graphen, Planarität und minimale Graphen*. Mannheim: Wissenschaftsverlag.
- Wend, H. D.** 1993. *Strukturelle Analyse linearer Regelungssysteme*. München: Oldenbourg.
- Wey, T.** 1996. *Ein graphentheoretischer Ansatz zur strukturellen Analyse und Synthese nichtlinearer Systeme*. VDI Fortschritt-Berichte. Reihe 8. 556. Düsseldorf: VDI.