



RUHR-UNIVERSITÄT BOCHUM

Peter Hilgers

Der Einsatz eines Mikrorechners
zur hybriden Optimierung und
Schwingungsanalyse

Heft Nr. 14



Mitteilungen
aus dem
Institut für Mechanik

Herrn Peter Grundmann mit freundlichen Grüßen
überreicht 6. 11. 78 Peter Hilgers

Institut für Mechanik

RUHR-UNIVERSITÄT BOCHUM

Peter Hilgers

Der Einsatz eines Mikrorechners zur hybriden
Optimierung und Schwingungsanalyse

Mitteilungen aus dem Institut für Mechanik Nr. 14

Juli 1978

Herausgeber:

Institut für Mechanik der Ruhr-Universität Bochum

Die vorliegende Arbeit entstand während meiner Tätigkeit am Institut für Mechanik der Ruhr-Universität Bochum und wurde von der dortigen Abteilung für Maschinenbau als Dissertation angenommen.

Herrn Prof. Dr.-Ing. H. Waller danke ich besonders für die Anregung zu dieser Arbeit und die stete, vielfältige Unterstützung, Herrn Prof. Dr. techn. K.-H. Fasol für die Übernahme des zweiten Referates.

Herrn Prof. Dr.-Ing. O. Mahrenholtz von der TU Hannover danke ich für die Erlaubnis, Meßeinrichtungen seines Lehrstuhls zu verwenden, dabei waren mir die Ratschläge von Herrn Dr.-Ing. B. Dirr hilfreich.

Mein Dank gilt ebenso den Herren H.D. Dreier und H.J. Michel, die bei der Programm- und Schaltungsentwicklung mitgearbeitet haben, Herr N. Wiegmann hat in kurzer Zeit das Manuskript mit großer Sorgfalt geschrieben.

Referenten:

Prof. Dr.-Ing. H. Waller

Prof. Dr. techn. K.-H. Fasol

Tag der Einreichung: 24.4.1978

Tag der mündlichen Prüfung: 9.6.1978

© 1978 Dipl. Ing. P. Hilgers

Spitzwegstr. 1, 435 Recklinghausen

Alle Rechte vorbehalten. Auch die fotomechanische Vervielfältigung des Werkes (Fotokopie, Mikrokopie) oder von Teilen daraus bedarf der vorherigen Zustimmung des Autors.

Zusammenfassung

Der Einsatz hybrider Rechenanlagen zur Lösung von Optimierungsaufgaben wird behandelt. Nach einem kurzen Überblick über verschiedene Optimierungsmethoden wird der Aufbau eines speziellen Hybridrechners aus einem Analogrechner und einem Mikrorechner dargestellt. Seine Nützlichkeit wird bei der Lösung verschiedener Optimierungsprobleme der Mechanik und Regelungstechnik unter Beweis gestellt. Dabei zeigt sich, daß bei der Optimierung dynamischer Systeme hybride Rechenanlagen rein digitalen vielfach überlegen sind, der Grund dafür liegt in der hohen analogen Rechengeschwindigkeit. Mit der Schwingungsmessung und Identifikation von Schwingungsparametern wird eine Aufgabe behandelt, bei der in besonderem Maße Echtzeitbedingungen zu beachten sind.

Summary

The use of hybrid computers for the solution of optimization problems is treated. After a short survey of optimization methods the development of a special hybrid computer from an analog- and a microcomputer is presented. Evidences for its usefulness are given with the solution of optimization problems from mechanics and automatic control. It shows that because of the high analog computing speed for the optimization of dynamic systems hybrid computers often are superior to pure digital ones. With vibration measurement and identification of vibrational parameters a task is treated, for which real time conditions are especially important.

<u>Inhaltsverzeichnis</u>	Seite
Häufig verwendete Bezeichnungen	5
<u>1. Einleitung</u>	6
<u>2. Klassifizierung der Optimierungsprobleme und -methoden</u>	8
2.1 Die Problemklassen	8
2.2 Methoden zur Parameteroptimierung	9
2.3 Methoden zur Funktionaloptimierung	11
<u>3. Iterative Optimierungsmethoden</u>	12
3.1 Methoden zur Parameteroptimierung ohne Berücksichtigung von Nebenbedingungen	12
3.1.1 Gradientenmethoden	13
3.1.2 Direkte Methoden	17
3.1.2.1 Methode von Hooke-Jeeves	18
3.1.2.2 Zufallsmethoden	18
3.2 Die Behandlung von Nebenbedingungen	20
3.3 Vergleichende Wertung	21
3.4 Methoden zur Funktionaloptimierung	22
<u>4. Hybride Optimierung</u>	24
4.1 Bisherige Entwicklung	24
4.2 Realisierung von hybriden Optimierungs- verfahren mit einem Mikrorechner	25
4.2.1 Beschreibung des vorhandenen Analog- rechners	25
4.2.2 Beschreibung des gewählten Mikro- rechners	26
4.2.2.1 Allgemeines	26
4.2.2.2 Auswahl des Mikrorechners	26
4.2.2.3 Der Mikroprozessor IMP 16 C	27
4.2.2.4 Der Mikrocomputer IMP 16 P	29

	Seite
4.2.3 Aspekte der rein digitalen Daten- verarbeitung mit dem Mikrorechner	30
4.2.3.1 Compiler, Assembler, Lader	30
4.2.3.2 Adressierung	31
4.2.3.3 Zahlendarstellung	33
4.2.3.4 Rechenzeiten	35
4.2.3.5 Allgemein benötigte Hilfspro- gramme	38
4.2.3.6 Programmierung eines Zufalls- generators	39
4.2.4 Schaltungen und Programme zur hy- briden Datenverarbeitung	40
4.2.4.1 Analoge Rechenarten und Anfor- derungen an die Rechnerkopplung	40
4.2.4.2 Synchronisierung von Analog- und Digitalrechner	42
4.2.4.3 Analog/Digitalwandlung	44
4.2.4.4 Digital/Analogwandlung	46
4.2.5 Zur Programmierung der Optimie- rungsverfahren	48
4.2.6 Beurteilung des entwickelten Re- chensystems	50
<u>5. Anwendungen des entwickelten Rechensystems</u> <u>zur hybriden Optimierung</u>	52
5.1 Allgemeine Hinweise und analoge Rechen- technik	52
5.2 Anwendungen zur Parameteroptimierung	53
5.2.1 Optimierung eines nichtlinearen Regelkreises	53
5.2.2 Suboptimaler Entwurf eines Zustands- reglers	58
5.2.2.1 Die Problemstellung	58
5.2.2.2 Der Einsatz hybrider Optimie- rungsverfahren	60
5.2.2.3 Anwendung beim Entwurf aktiver Schwingungsdämpfer für Eisen- bahnwagen	61

	Seite
5.3 Anwendungen zur Funktionaloptimierung	70
5.3.1 Optimale Steuerung eines nicht-linearen Systems	70
5.3.2 Optimaler Entwurf dynamisch beanspruchter Stäbe	74
5.3.2.1 Die maßgebenden Differentialgleichungen	75
5.3.2.2 Lösung durch hybride Optimierung	78
5.3.2.3 Ergebnisse	81
5.4 Erfahrungen aus der hybriden Optimierung	84
<u>6. Der Einsatz des Mikrorechners zur Schwingungsanalyse mit der Fourier- und Laplace-Transformation</u>	86
6.1 Gemeinsamkeiten hybrider Optimierung und digitaler Meßwertverarbeitung	86
6.2 Die Fourier-Transformation	86
6.2.1 Die kontinuierliche Fourier-Transformation	87
6.2.2 Die diskrete Fourier-Transformation als Spezialfall der kontinuierlichen	90
6.3 Die systematischen Fehler der diskreten Fourier-Transformation	92
6.3.1 Der aus der Zeitdiskretisierung stammende Fehler (aliasing)	94
6.3.2 Der aus dem Abschneiden im Zeitbereich stammende Fehler (ripple)	96
6.4 Der Algorithmus der schnellen Fourier-Transformation	99
6.5 Die Ermittlung von Schwingungskenndaten aus der Fourier-Transformation von Meßsignalen	101
6.5.1 Abgrenzung des betrachteten Schwingungstyps	101
6.5.2 Analytische Fourier-Transformation von gedämpften harmonischen Schwingungen	102

	Seite
6.5.3 Die Bestimmung der Eigenfrequenzen	103
6.5.4 Die Bestimmung weiterer Parameter durch Optimierung	106
6.5.4.1 Prinzipielles Vorgehen	106
6.5.4.2 Vorgehen bei schwacher Dämpfung	109
6.6 Auswertung digital simulierter Schwin- gungen	110
6.7 Versuchsaufbau und -durchführung	116
6.8 Versuchsergebnisse	119
6.8.1 Untersuchung einer Verdichterschau- fel	119
6.8.2 Untersuchung einer Turbinenschaufel	121
6.9 Diskussion und mögliche Behandlung an- derer Schwingungsprobleme	126
6.10 Ausblick auf die Optimierung unter Echtzeitbedingungen	128
<u>7. Zusammenfassung</u>	130
<u>8. Literaturverzeichnis</u>	133
Anhang: Befehlssatz des Mikrorechners IMP 16 P	142

Häufig verwendete Bezeichnungen

$G(\underline{x})$: Gütefunktion bei Parameteroptimierungen

J : Gütefunktion bei Funktionaloptimierungen

n : Anzahl der Parameter

$2N$: Anzahl diskreter Punkte im Zeitbereich bei der
Fourier-Transformation

T : Meßzeit

\underline{x} : Parametervektor bei Parameteroptimierungen

$\underline{x}(t)$: Zustandsvektor

Vektoren sind durch kleine unterstrichene Buchstaben
gekennzeichnet, z.B. \underline{x} , Matrizen durch Großbuchstaben,
z.B. \underline{A} .

Die Fourier-Transformierte einer Zeitfunktion $f(t)$ ist
mit $F(\omega)$ bezeichnet.

1. Einleitung

Seit jeher versuchen Ingenieure, bei Konstruktionen die Festigkeit zu erhöhen, Gewicht zu sparen, Kennwerte genauer zu ermitteln oder den Wirkungsgrad eines Prozesses zu erhöhen, kurz, Maschinen und Verfahren zu verbessern. Eine fortgesetzte, systematische Suche nach immer besseren Lösungen wird als Optimierung bezeichnet, man kann also die oben genannten Aufgaben als spezielle Optimierungsprobleme auffassen. Diese vom Einzelfall abstrahierende Betrachtungsweise wird durch zwei sich gegenseitig bedingende Tendenzen gefördert:

1. Die Entwicklung der Rechenmaschinen, speziell der Digitalrechner, gekennzeichnet durch ständig kleiner werdende Volumina, höhere Arbeitsgeschwindigkeiten und günstigere Preis-Leistungsverhältnisse. Das jüngste, inzwischen zu einer gewissen Popularität gelangte Kind dieser Entwicklung ist der Mikrocomputer.
2. Die Entwicklung numerischer Methoden zur Parameter- und Funktionaloptimierung (auf diese Begriffe wird im zweiten Kapitel eingegangen).

Zunächst wurden Optimierungsmethoden hauptsächlich in der Luft- und Raumfahrt [1] und der Regelungstechnik [2] eingesetzt, der Anwendungsbereich weitet sich jedoch inzwischen ständig weiter aus, etwa auf die Konstruktionstechnik [3] oder Verfahrenstechnik [4].

Die vorliegende Arbeit wird vom Verfasser als ein Beitrag zur Anwendung von Optimierungsmethoden verstanden. Ihr Ziel ist es

- zu zeigen, wie durch die Koppelung analoger und digitaler Rechenelemente besonders bei der Optimierung dynamischer Systeme spezifische Vorteile auftreten, wie sie von keiner anderen Rechnerart erreicht werden,
- den Aufbau eines speziellen Hybridrechners aus einem Mikrocomputer und einem Analogrechner darzustellen,
- schließlich die vielfältigen Einsatzmöglichkeiten des entwickelten Rechensystems zu demonstrieren.

Im einzelnen werden in dieser Arbeit die folgenden Punkte behandelt:

- Klassifikation der verschiedenen Optimierungsprobleme und -methoden (Kap. 2).
- Darstellung ausgewählter Verfahren zur Parameter- und Funktionaloptimierung (Kap. 3).

Den Hauptteil der Arbeit bilden die folgenden Abschnitte über

- Grundprinzipien der hybriden Optimierung und Realisierung hybrider Optimierungsmethoden mit einem Mikrorechner (Kap. 4).
- Anwendungen des realisierten Rechensystems zur Parameter- und Funktionaloptimierung (Kap. 5).
- Einsatz des Mikrorechners zur digitalen Meßwertverarbeitung, dargestellt am Beispiel einer Schwingungsanalyse (Kap. 6).

2. Klassifizierung der Optimierungsprobleme und -methoden

2.1 Die Problemklassen

Die Auswahl eines aus bestimmten Gründen als besonders vorteilhaft angesehenen Systems oder einer Systemvariante wird als Optimierung bezeichnet. Optimierung setzt voraus, daß alternative Wahlmöglichkeiten bestehen und daß die Alternativen durch ein Gütemaß bewertet werden. Bei der mathematischen Behandlung von Optimierungsproblemen verwendet man in der Regel ein skalares Gütemaß, man spricht von einer Gütefunktion. Die Bezeichnungen Güteindex, Gütekriterium, Zielfunktion, Kriteriumsfunktion sind ebenfalls gebräuchlich.

Eine erste Einteilung der Optimierungsmethoden geschieht nach der Art der zu bestimmenden Werte. Sind einzelne, diskrete Werte gesucht, spricht man von Parameteroptimierung, sollen eine oder mehrere Funktionen einer oder mehrerer unabhängig Veränderlicher bestimmt werden, von Funktionaloptimierung. Anstelle von Parameteroptimierung findet man in der Literatur auch die Begriffe nichtlineare Programmierung, sofern der Zusammenhang zwischen Parametern und Gütefunktion nichtlinear ist, oder statische Optimierung. Letztere Bezeichnung ist nicht sehr glücklich gewählt, da eine Parameteroptimierung statischer und dynamischer Systeme gleichermaßen durchgeführt werden kann.

Zur Nomenklatur ist zu sagen, daß man bei Parameteroptimierungen die Parameter gewöhnlich in einem Spaltenvektor \underline{x} zusammenfaßt, die Zahl der Parameter wird im folgenden stets mit n bezeichnet, die Gütefunktion mit $G(\underline{x})$. Bei Funktionaloptimierungen ist auch die Verwendung von J für die Gütefunktion gebräuchlich.

Bei den meisten technischen Optimierungsaufgaben sind Nebenbedingungen zu beachten, z.B. können Verstärkungsfaktoren und Stellgrößen nicht beliebig groß gemacht, Grenzwerte dürfen nicht überschritten werden. Parametervektoren, die alle Nebenbedingungen erfüllen, nennt

man zulässig. Nebenbedingungen sind bei der Parameteroptimierung häufig durch Gleichungen und Ungleichungen

$$h_i(\underline{x}) = 0, i=1 \dots p < n \quad (2.1)$$

$$g_j(\underline{x}) \leq 0, i=1 \dots q \stackrel{\geq}{\leq} n \quad (2.2)$$

gegeben. Bei Funktionaloptimierungen sind Beschränkungen einer gesuchten Funktion $g(t)$ der Art

$$|g(t)| \leq a \quad (2.3)$$

typisch.

2.2 Methoden zur Parameteroptimierung

Je nach Fall ist das angestrebte Optimum ein Minimum oder ein Maximum. Oft spricht man nur von Minimierung, da jedes Maximum von $G(\underline{x})$ ein Minimum von $-G(\underline{x})$ ist und umgekehrt. Die Unterscheidung von lokalem und globalem Minimum ist in Abb. 2.1 für den eindimensionalen Fall dargestellt und gilt sinngemäß für mehr Dimensionen.

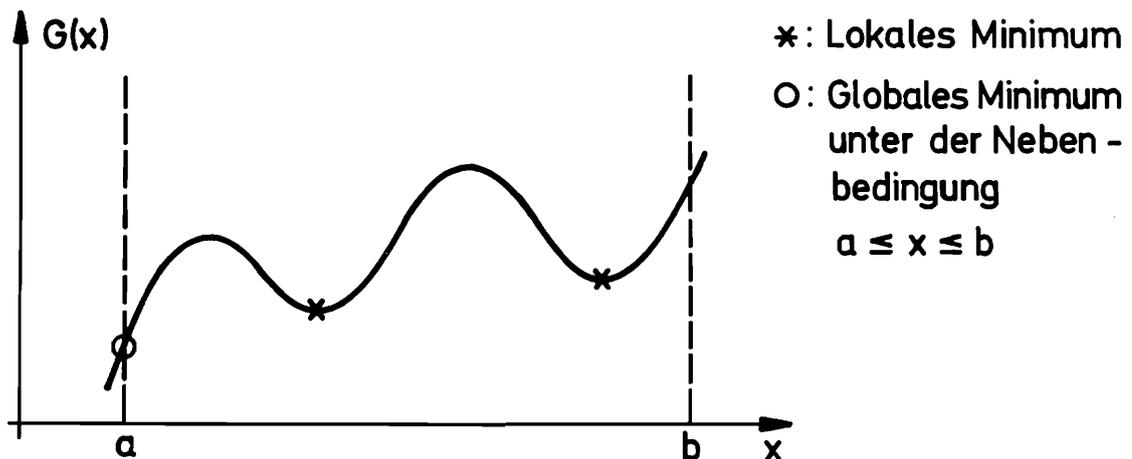


Abb. 2.1: Lokales und globales Minimum

Ist die Zielfunktion stetig und differenzierbar, so ist beim unbeschränkten Problem

$$\text{grad } G(\underline{x}) = \underline{0} \quad (2.4)$$

notwendige Bedingung für ein Minimum. Abgesehen von dem Fall, daß G linear oder quadratisch von \underline{x} abhängt, liefert (2.3) einen Satz nichtlinearer Gleichungen, der nicht ohne weiteres gelöst werden kann. Da Wendepunkte und relative Maxima ebenfalls Lösungen von (2.3) sind, müssen die zweiten Ableitungen der Gütefunktion herangezogen werden. In vielen Fällen, etwa wenn zu einem Parametervektor \underline{x} der zugehörige Wert $G(\underline{x})$ erst durch Lösen eines Differentialgleichungssystems bestimmt werden muß, ist eine analytische Gradientenbestimmung gar nicht möglich. Man ist daher gezwungen, zu iterativen Methoden des Typs

$$\underline{x}_{i+1} = \underline{x}_i + \Delta \underline{x}_i \quad i=0,1,2,\dots \quad (2.5)$$

überzugehen. \underline{x}_0 wird geschätzt oder geraten. Die verschiedenen Methoden unterscheiden sich in der Art, wie \underline{x}_i gebildet wird, sie werden in Abschnitt 3.1 behandelt.

Nebenbedingungen können auf verschiedene Art berücksichtigt werden. Zum Abbruch der Iteration benutzt man unterschiedliche Kriterien, etwa

$$|G(\underline{x}_{i+1})| < \epsilon_1 \quad (2.6)$$

wenn der optimale Wert der Gütefunktion von vornherein bekannt ist (Fehlerminimierung) oder sonst

$$|\text{grad } G(\underline{x}_i)| < \epsilon_2 \quad (2.7)$$

$$|\underline{x}_{i+1} - \underline{x}_i| < \epsilon_3 \quad (2.8)$$

2.3 Methoden zur Funktionaloptimierung

Viele Aufgaben der Funktionaloptimierung sind nach Röhe [5] Spezialfälle des allgemeinen Problems:

Minimiere

$$J\{\underline{x}(\underline{y}(t)), t\} = M(\underline{x}(t_b), t_b) + \int_{t_a}^{t_b} L(\underline{x}(\underline{y}(t)), t) dt \quad (2.9)$$

bezüglich eines Satzes unbekannter Funktionen

$\underline{y}(t) = |y_1(t), y_2(t), y_3(t) \dots y_n(t)|^T$. Der Zusammenhang zwischen den gesuchten Funktionen $\underline{y}(t)$ und den Zustandsvariablen $\underline{x}(t)$ des Problems ist gegeben durch

$$\dot{\underline{x}}(t) = f(\underline{x}, \underline{y}, t) \quad (2.10)$$

und entsprechende Anfangs- und Randbedingungen. In vielen Fällen ist die unabhängig Veränderliche die Zeit, es kann sich aber auch etwa um eine Längenkoordinate handeln. Die so gestellte Aufgabe gehört zur Variationsrechnung, für den Fall, daß $\underline{y}(t)$ keinen Nebenbedingungen der Art (2.3) unterliegt, stellen die Euler-Lagrange'schen Differentialgleichungen notwendige Bedingungen für $\underline{y}(t)$ dar. Ihre Lösung ist meist, da ein Randwertproblem vorliegt, nur iterativ möglich. Daher sind Methoden entwickelt worden, das Gütekriterium (2.10) direkt iterativ zu minimieren nach dem Prinzip

$$\underline{y}_{i+1}(t) = \underline{y}_i(t) + \underline{\Delta y}_i(t) \quad (2.11)$$

In Abschnitt 3.4 wird kurz auf diese Methoden eingegangen.

3. Iterative Optimierungsmethoden

3.1 Methoden zur Parameteroptimierung ohne Berücksichtigung von Nebenbedingungen

Die Literatur über iterative Methoden zur Parameteroptimierung ist nahezu unübersehbar, Schwefel's vergleichende Arbeit [6] enthält rund 1.000 Titel (im Literaturverzeichnis). Nahezu monatlich werden neue Verfahren oder Varianten vorgeschlagen. Schon allein daraus ist ersichtlich, daß es eine allgemein als beste anerkannte Strategie nicht gibt und vermutlich auch nicht geben wird. Im folgenden sollen einige der renommiertesten Methoden kurz dargestellt werden. Weitere Verfahren und zusätzliche Details zu den hier beschriebenen findet man in den Büchern von Aoki und Wilde [7, 8].

Die iterativen Methoden können eingeteilt werden in

- Methoden, die keine Ableitungen der Zielfunktion benötigen, sie werden auch direkte Methoden genannt,
- Methoden, die nur erste Ableitungen der Zielfunktion verwenden und
- Methoden, die auch zweite Ableitungen der Zielfunktion verwenden.

Häufig verläuft die Iteration nach dem Schema

$$x_{i+1} = x_i + \alpha_i s_i \quad (3.1)$$

dabei ist s_i eine vom Verfahren bestimmte Suchrichtung. Der Faktor α_i muß während des Iterationsschrittes so bestimmt werden, daß $G(x)$ beim Fortschreiten von x_i aus in Richtung s_i ein lokales Minimum annimmt. Die Bestimmung von α_i wird lineare Suche genannt, da bei dieser Suche x nur längs einer Geraden im Parameterraum variiert wird. Numerisch wird die lineare Suche entweder durch einen Einschachtelungsprozess oder ein Extrapolationsverfahren durchgeführt. Bei den Extrapolationsmethoden nimmt man an, daß sich der Verlauf der Gütefunktion in Richtung s_i durch eine Parabel zweiter oder

dritter Ordnung beschreiben läßt, deren Minimum kann aus den Funktionswerten an zwei oder drei Stützstellen berechnet werden.

Die quadratische Funktion

$$G(\underline{x}) = c + \underline{b}^T \underline{x} + \frac{1}{2} \underline{x}^T \underline{Q} \underline{x} \quad (3.2)$$

mit

$$\text{grad } G(\underline{x}) = \underline{b} + \underline{Q} \underline{x} \quad (3.3)$$

und \underline{Q} als der Hesse'schen Matrix der partiellen zweiten Ableitungen hat ein globales Minimum an der Stelle

$$\underline{x}^* = -\underline{Q}^{-1} \underline{b} \quad (3.4)$$

wenn \underline{Q} positiv definit ist.

Im Fall $n=2$ sind dann die Linien $G=\text{const.}$ konfokale Ellipsen in der x_1-x_2 Ebene. Da die Funktion (3.2) als eine nach dem zweiten Glied abgebrochene Taylorentwicklung allgemeiner Gütefunktionen um das Minimum angesehen werden kann, ist zu vermuten, daß (3.2) zumindest in der Nähe des Minimums eine gute Approximation für andere Gütefunktionen darstellt. Aus diesem Grund hat man sich bemüht, iterative Verfahren zu entwickeln, die für die Funktion (3.2) gute Konvergenzeigenschaften aufweisen. Diese Verfahren basieren auf der bekannten Gradientenmethode und sollen mit dieser zusammen dargestellt werden, die Behandlung der direkten Methoden erfolgt anschließend.

3.1.1 Gradientenmethoden

Klassisch ist die Methode des steilsten Anstieges (Anstieg bei Maximierung, Abstieg bei der hier betrachteten Minimierung). Bei ihr wird als Suchrichtung

$$\underline{s}_i = -\underline{g}_i := -\text{grad } G|_{\underline{x}_i} \quad (3.5)$$

gewählt, da im Punkt \underline{x}_i die Funktion G die stärkste Abnahme in negativer Gradientenrichtung aufweist. α_i aus (3.1) wird durch lineare Suche bestimmt. Die Gradientenmethode hat den Vorteil, daß sie häufig zu Beginn der Suche rasch, später jedoch langsam konvergiert, wie Abb. 3.1 für den Fall einer quadratischen Zielfunktion zweier Veränderlicher verdeutlicht.

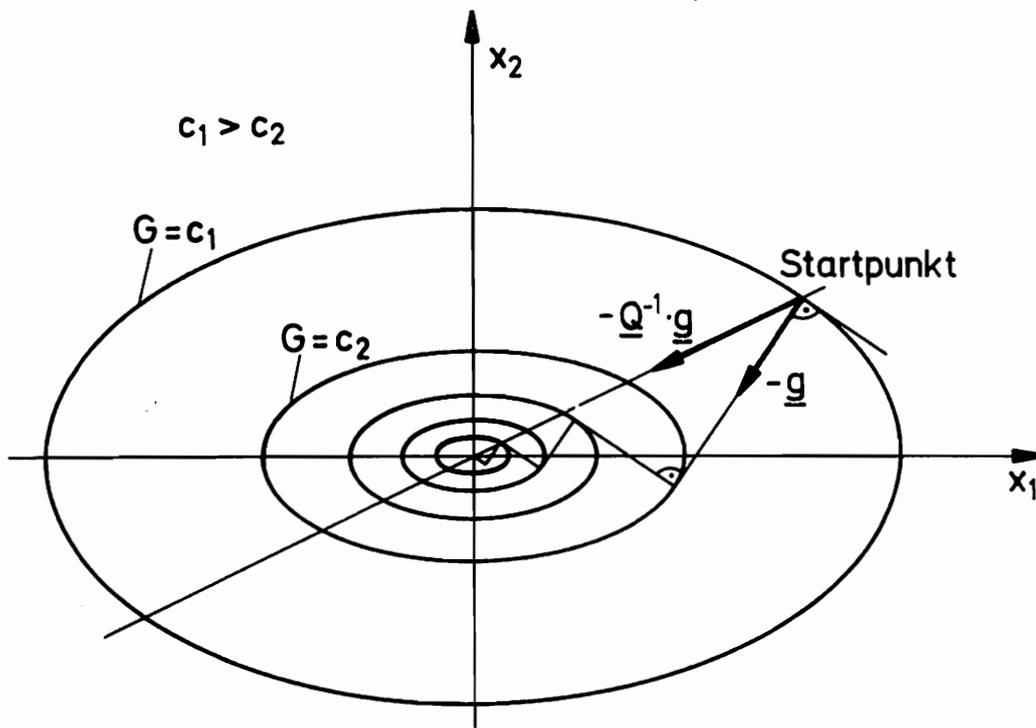


Abb. 3.1: Gradientenmethoden für $n = 2$

Die Suche könnte offensichtlich beschleunigt werden, wenn statt der Richtung $-\underline{g}$ eine direkt durch das Minimum laufende Richtung zur linearen Suche gewählt würde. Eine solche Richtung ist für quadratische Funktionen gegeben durch $-\underline{Q}^{-1} \underline{g}$, da unter Berücksichtigung von (3.3) gilt:

$$\begin{aligned} G(\underline{x}_i - \alpha_i \underline{Q}^{-1} \underline{g}_i) &= G(\underline{x}_i - \alpha_i \underline{Q}^{-1} \underline{b} - \alpha_i \underline{x}_i) \\ &= G(-\underline{Q}^{-1} \underline{b}) \quad \text{für } \alpha_i = 1 \end{aligned} \quad (3.6)$$

Das Minimum $-\underline{Q}^{-1} \underline{b}$ könnte also durch eine einzige lineare Suche bestimmt werden, wenn die Matrix \underline{Q} der partiellen zweiten Ableitungen bekannt wäre. Verfahren, die eine komplette Kenntnis der zweiten Ableitungen voraussetzen, werden Newton-Verfahren genannt. In vielen Fällen läßt sich jedoch der Gradient, und erst recht die Matrix der zweiten Ableitungen, nicht analytisch berechnen. Die numerische Annäherung durch finite Differenzen ist aufwendig und mit Ungenauigkeiten behaftet, sie kann deshalb nur für die Gradientenberechnung empfohlen werden. Die in Gl. (3.6) benötigte Matrizeninversion ist eine zusätzliche Fehlerquelle.

Im n -dimensionalen Vektorraum heißen zwei Vektoren \underline{x}_1 und \underline{x}_2 konjugiert bezüglich einer positiv definiten Matrix \underline{A} , oder \underline{A} -konjugiert, wenn gilt:

$$\underline{x}_1^T \underline{A} \underline{x}_2 = 0. \quad (3.7)$$

Die \underline{A} -Konjugiertheit ist eine Verallgemeinerung der Orthogonalität. Bei Orthogonalität ist \underline{A} gleich der Einheitsmatrix. Es kann bewiesen werden [7], daß das Minimum einer quadratischen Funktion (3.2) durch genau n lineare Suchen längs \underline{Q} -konjugierter Richtungen gefunden werden kann (\underline{Q} aus Gl. (3.2)). Die Verfahren der sogenannten konjugierten Gradienten bemühen sich, dieses günstige Verhalten auch ohne Kenntnis der Hesse'schen Matrix \underline{Q} und ohne Matrizeninversion zu erzielen.

Dazu wird versucht, die Matrix \underline{Q}^{-1} durch eine Folge von Matrizen \underline{B}_i unter Verwendung des Gradienten, nicht jedoch von zweiten Ableitungen, schrittweise anzunähern. Für den Fall der quadratischen Zielfunktion (3.2) ist

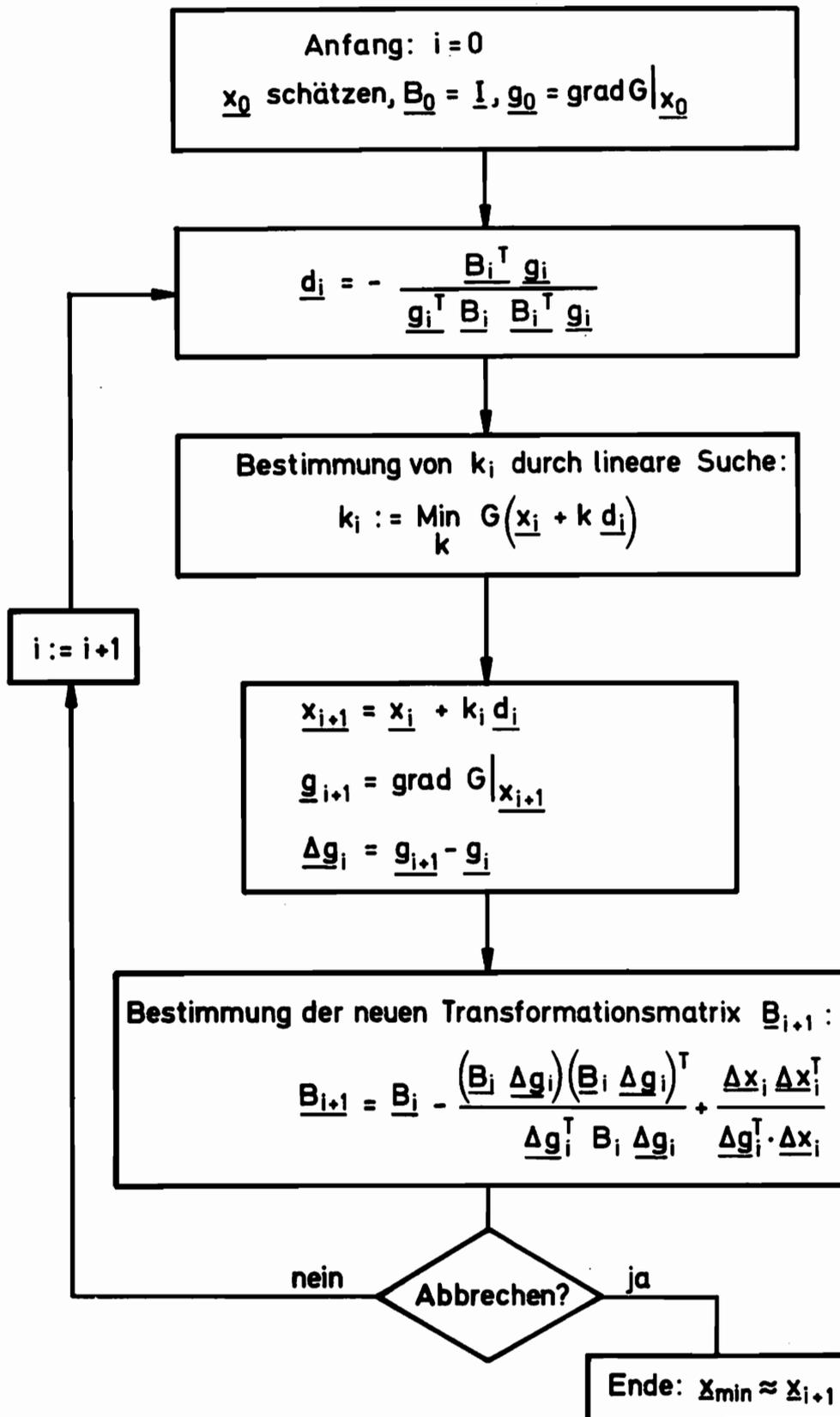


Abb. 3.2: Algorithmus von Davidon-Fletcher-Powell

$\underline{B}_n = \underline{Q}^{-1}$ und das Minimum wird nach genau n linearen Suchen gefunden. Abb. 3.2 enthält das Flußdiagramm für die nach Davidon-Fletcher-Powell benannte Variante des Verfahrens. Die darin verwendeten Abkürzungen sind:

$$\Delta \underline{g}_i = \underline{g}_{i+1} - \underline{g}_i \quad (3.8)$$

$$\Delta \underline{x}_i = \underline{k}_i \quad \underline{d}_i = \underline{x}_{i+1} - \underline{x}_i \quad (3.9)$$

Die Einheitsmatrix ist mit \underline{I} bezeichnet.

Numerische Testrechnungen haben ergeben, daß der Algorithmus nur dann das vorhergesagte günstige Verhalten zeigt, wenn die linearen Suchen sehr genau durchgeführt werden. Andernfalls kann es vorkommen, daß aufeinanderfolgende Suchrichtungen voneinander linear abhängig werden und somit nicht mehr der gesamte Parameterraum abgesucht wird. Die praktisch vorkommenden Zielfunktionen sind vielfach nicht quadratisch, dann kann auch in der Theorie das Minimum nicht nach n linearen Suchen erreicht werden. \underline{B}_n stellt in diesem Fall nur eine Näherung für die Inverse der Hesse'schen Matrix dar, und es müssen am Algorithmus Modifikationen vorgenommen werden. Steward [9] setzt dazu nach einer bestimmten Zahl von Iterationen \underline{B}_i wieder gleich der Einheitsmatrix. Eine neuere Variante von Davidon [10] ermöglicht es, das Minimum einer quadratischen Funktion auch ohne lineare Suchen in n Schritten zu finden.

3.1.2 Direkte Methoden

Von den direkten Methoden sollen die "pattern search" von Hooke-Jeeves [4] und die Zufallsmethode von Rechenberg [11] besprochen werden.

3.1.2.1 Methode von Hooke-Jeeves

Diese heuristische Methode ist gekennzeichnet durch zwei sich abwechselnde Zyklen: Tasten und Voranschreiten. Während der Tastphase wird die Umgebung eines bisher besten Punktes \underline{x} durch Variation jeweils nur eines Parameters untersucht. Wird so eine erfolgversprechende Richtung gefunden, erfolgt ein Schritt in diese Richtung. Die Schrittweite dieses Voranschreitens vergrößert sich dabei von Erfolg zu Erfolg, wodurch ein Trend (pattern search) berücksichtigt wird. Mißerfolge beim Voranschreiten in Trendrichtung führen zu einer Verkleinerung der Schrittweite, nach mehreren aufeinanderfolgenden Fehlschlägen wird eine neue Trendrichtung ermittelt. Lineare Suchen finden nicht statt. Abb. 3.3 veranschaulicht das Vorgehen bei zwei Variablen. Erfolgreiche Schritte sind ausgezogen, nicht erfolgreiche durch gestrichelte Linien gekennzeichnet. Man beachte die Vergrößerung der Schritte beim Voranschreiten.

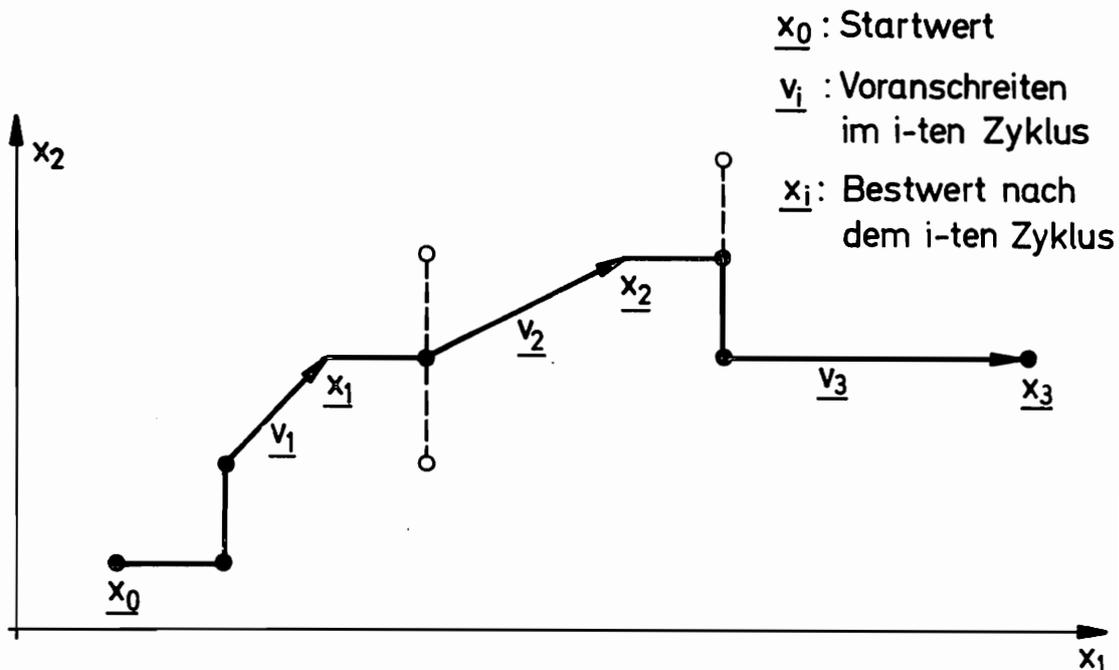


Abb. 3.3: Strategie von Hooke-Jeeves

3.1.2.2 Zufallsmethoden

Die Zufallsmethoden unterscheiden sich grundlegend von allen bisher besprochenen Verfahren. Während bei diesen versucht wird, die im Laufe der Iterationen gesammelte Information auszunutzen und im Sinn etwa einer Modellgütefunktion zu interpretieren, verzichten Zufallsverfahren weitgehend auf jedes Modell der Gütefunktion. Vor allem aus zwei Gründen haben solche Methoden Interesse gefunden: Die bisher behandelten Verfahren können mehr oder weniger versagen, wenn die Gütefunktion Ecken, Kanten oder stark gekrümmte, steile Täler aufweist. Diese anschaulichen Bezeichnungen sind als Verallgemeinerungen der im zweidimensionalen Fall möglichen Darstellung der Gütefunktion als Gebirge über der Parameterebene aufzufassen. Zum zweiten wächst bei Zufallsmethoden die zur Bestimmung des Minimums nötige Anzahl von Funktionswertberechnungen linear mit der Anzahl der Parameter, wie die Arbeiten von Rechenberg, Rastrigin und Schumer-Steiglitz ergeben haben [11, 12, 13]. Bei anderen Methoden ist mit überproportionalem Anstieg des Aufwandes zu rechnen, so daß für große n Zufallsmethoden vorteilhafter sein können.

Das von Rechenberg entwickelte Verfahren der Evolutionsstrategie ist eine bewußte Nachahmung von Prinzipien der biologischen Evolution. Bei jeder Iteration wird der bis dahin gefundene beste Wert \underline{x}_{best} nach der Vorschrift

$$\underline{x}_{Probe} = \underline{x}_{best} + \underline{\Delta x} \quad (3.10)$$

variiert. Führt die Variation zu einer Verringerung der Gütefunktion, wird \underline{x}_{Probe} als neuer Bestwert genommen, andernfalls eine neue Variation gebildet usw. $\underline{\Delta x}$ ist ein Vektor von Zufallszahlen. Jede einzelne seiner voneinander unabhängigen Komponenten ist normalverteilt mit dem Mittelwert 0 und der Streuung σ . Von entscheidender Bedeutung für schnelle Konvergenz ist die passende Wahl von

6. Aufgrund wahrscheinlichkeitstheoretischer Untersuchungen kommt Rechenberg zu dem Ergebnis, daß die Konvergenzgeschwindigkeit zum Minimum dann am größten wird, wenn die sogenannte 1/5 Erfolgsregel beachtet wird, sie lautet:

Ermittle während der Optimumssuche von Zeit zu Zeit die Erfolgswahrscheinlichkeit, d.h. das Verhältnis der Zahl der Erfolge zur Zahl der Versuche insgesamt. Ist das Verhältnis größer als 1/5, dann vergrößere die Streuung, ist es kleiner als 1/5, dann verringere sie.

Die Anwendung der 1/5 Erfolgsregel führt dazu, daß in der Nähe des Minimums die Streuung und damit die durchschnittliche Schrittweite immer mehr verkleinert wird, daraus kann ein sinnvolles Abbruchkriterium gewonnen werden.

3.2 Die Behandlung von Nebenbedingungen

Bei den Nebenbedingungen werden die Fälle der Nebenbedingungen in Gleichungsform

$$h_i(\underline{x}) = 0, \quad i = 1, \dots, p < n \quad (3.11)$$

und in Ungleichungsform

$$g_j(\underline{x}) \leq 0, \quad j = 1, \dots, q \leq n \quad (3.12)$$

unterschieden. Nebenbedingungen in Gleichungsform lassen sich mit Hilfe der Methode der Lagrange'schen Multiplikatoren behandeln [7]. Man geht aus von der erweiterten Zielfunktion

$$\hat{G}(\underline{x}, \underline{\lambda}) = G(\underline{x}) + \underline{\lambda}^T \underline{h}. \quad (3.13)$$

Dabei stellt \underline{h} die in einem Vektor zusammengefaßten Nebenbedingungen (3.11) und $\underline{\lambda}$ den $p \times 1$ Vektor der Lagrange'schen Multiplikatoren dar. Unter recht allgemeinen Voraussetzungen (s. Aoki [7]) stimmt das Minimum

von \hat{G} mit dem von G überein, man hat also das Problem in eine Optimierungsaufgabe ohne Nebenbedingungen überführt. Dafür ist allerdings der Parametervektor \underline{x} um die zusätzlichen Unbekannten $\underline{\lambda}$ erweitert worden.

Von den verschiedenen Methoden zur Berücksichtigung von Nebenbedingungen in Ungleichungsform soll nur das Verfahren der Straffunktionen angedeutet werden. Man bemüht sich wieder, das Problem mit Nebenbedingungen in ein solches ohne Nebenbedingungen, aber (nahezu) derselben Lösung umzuwandeln. Dazu wird, anschaulich gesprochen, eine bestimmte "Strafe" zur Gütefunktion addiert, wenn eine oder mehrere Nebenbedingungen verletzt werden. Eine Möglichkeit, Strafterme einzufügen, ist die Modifikation

$$\hat{G}(\underline{x}, k) = G(\underline{x}) - k \sum_{j=1}^q \frac{1}{g_j(\underline{x})}, \quad k > 0. \quad (3.14)$$

Nähert sich während des Iterationsprozesses der Punkt \underline{x} dem durch (3.12) als unzulässig deklarierten Gebiet, strebt das entsprechende $g_j(\underline{x})$ von der negativen Seite her gegen Null und der additive Term in (3.14) wird sehr groß. So wird das Suchverfahren von dieser Region ferngehalten, die Wahl des Faktors k ist allerdings nicht unproblematisch.

3.3 Vergleichende Wertung

Die vorangegangene Darstellung der verschiedenen Optimierungsverfahren führt zu der Frage, wie bei gegebenem Problem eine geeignete Optimierungsmethode ausgewählt werden kann. Ganz allgemein läßt sich sagen, daß in jedem Fall alle problemspezifischen Besonderheiten bei der Auswahl berücksichtigt werden sollten. Ist etwa bei der Minimierung von Fehlerquadraten zu hoffen, daß die Gütefunktion angenähert quadratisch ist, und kann der Gradient der Gütefunktion mit vertretbarem Aufwand - eventuell numerisch - bestimmt werden, kann zu einem Verfahren der konjugierten Gradienten

geraten werden. Ist der Gradient nicht verfügbar, käme als nächstes ein direktes Suchverfahren in Betracht. In Fällen großer Dimension des Problems, oder wenn etwa die Gütefunktionswerte mit Meßfehlern behaftet sind, ist ein robustes, aber langsames Zufallsverfahren zu empfehlen.

Keine dieser Methoden ist in der Lage, zwischen einem lokalen und dem globalen Extremum zu unterscheiden. Für die auf Gradientenauswertungen beruhenden Verfahren ist das schon eine Folge des Arbeitsprinzips, die direkten Methoden zeigen in der praktischen Erprobung kein besseres Verhalten. Ist man ausschließlich am globalen Extremum interessiert, bleibt nur die Möglichkeit, die Optimierung für verschiedene Startwerte zu wiederholen und die Ergebnisse miteinander zu vergleichen. Wie die später dargestellten Anwendungen jedoch zeigen werden, ist dieser prinzipielle Mangel der iterativen Optimierungsmethoden bei vielen Problemen von keiner großen praktischen Bedeutung.

3.4 Methoden zur Funktionaloptimierung

Ebenso wie zur Parameter- sind auch zur Funktionaloptimierung zahlreiche numerische Methoden entwickelt worden. Da Funktionaloptimierungen jedoch nicht im Mittelpunkt der hier dargestellten Arbeit standen, soll entsprechend den Rechenerfahrungen des Verfassers nur ein ausgewähltes Verfahren besprochen werden.

Korn und Kosako haben vorgeschlagen [14], die von der Parameteroptimierung her bekannten Zufallsmethoden auch für Funktionaloptimierungen einzusetzen. Eine gesuchte Funktion $y(t)$ wird dazu in äquidistanten Schritten Δt diskretisiert, die einzelnen Diskretisierungswerte $y(i \cdot \Delta t)$ ($i = 1, 2, \dots, n$) behandelt man dann als Parameter. Die konzeptionell einfache Methode ist für hybride Rechenanlagen bestimmt.

Im übrigen sei darauf verwiesen, daß eine weitgehende Analogie zwischen Parameter- und Funktionaloptimierungsmethoden besteht. Gradienten- und konjugierte Gradientenmethoden sind vom Vektorraum auf den Funktionenraum übertragen worden, Einzelheiten findet man bei Röhe[5] .

4. Hybride Optimierung

4.1 Bisherige Entwicklung

Seit dem ersten Auftreten hybrider Rechenanlagen etwa Mitte der 60er Jahre findet man vereinzelt in der Literatur Arbeiten über ihren Einsatz zur Optimierung dynamischer Systeme [15-23]. Ausgangspunkt ist in jedem Fall die einfache Grundidee, jedem Teil einer hybriden Rechenanlage den Aufgabenteil zuzuweisen, für den er besonders geeignet ist. Der Analogrechner löst dann die das Systemverhalten beschreibenden Differentialgleichungen und berechnet die Gütefunktion, der Digitalrechner wertet die Gütefunktion aus und bestimmt nach einer iterativen Optimierungsmethode neue Parameter- bzw. Funktionsvariationen. Abb. 4.1 verdeutlicht das Zusammenwirken beider Komponenten.

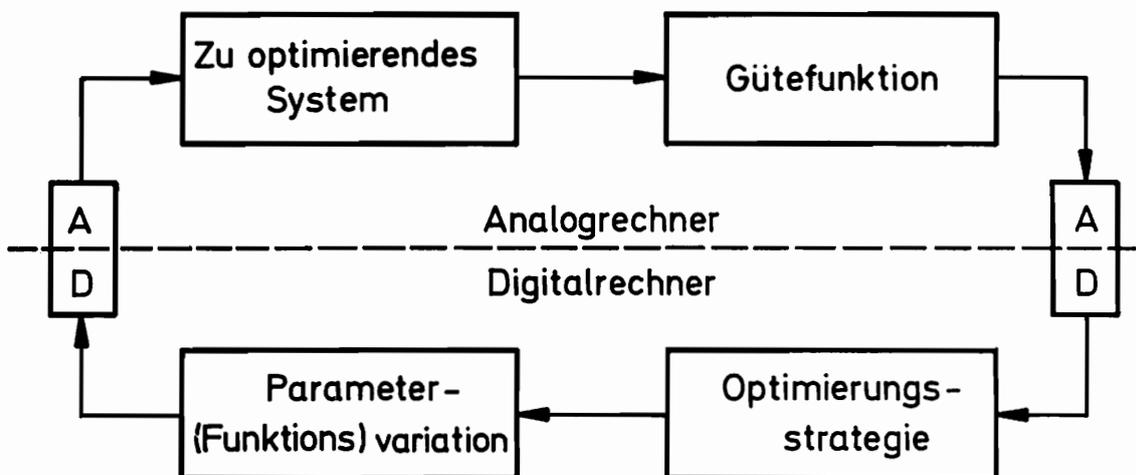


Abb. 4.1: Grundprinzip der hybriden Optimierung

Für den Datenaustausch zwischen beiden Rechnern werden Digital-Analog- und Analog-Digital-Wandler eingesetzt.

Die dem Verfasser bekanntgewordenen Anwendungen hybrider Rechenverfahren zur Optimierung konzentrieren sich auf die Regelungstechnik. Vielfach wurden die aus dem Pontryagin'schen Maximumprinzip resultierenden Differentialgleichungen und Randwertprobleme durch eine Optimierung gelöst, meist mit von industriellen Anbietern entwickelten, fertigen Hybridsystemen.

In frühen Arbeiten des Verfassers [24, 25] wurde gezeigt, wie systematisch logische Schaltungen für die Realisierung einfacher Optimierungsstrategien auf einem Analogrechner entwickelt werden können. Der Aufwand an analogen und logischen Bauelementen begrenzte die Anwendung meist auf Optimierungen von 2 bis 3 Parametern, auch die Behandlung von Nebenbedingungen war nur schwer möglich. Es stellte sich daher die Frage, ob durch den Einsatz neuer elektronischer Bauteile (Mikroprozessoren) diese Beschränkungen überwunden und darüber hinaus der hybriden Optimierung neue Anwendungen erschlossen werden können. Die folgenden Abschnitte beschäftigen sich mit der Beantwortung dieser Fragen.

4.2 Realisierung von hybriden Optimierungsverfahren mit einem Mikrorechner

4.2.1 Beschreibung des vorhandenen Analogrechners

Als Analogrechner wurde ein RA 770/775 der Firma Telefunken benutzt. In der gegebenen Ausbaustufe verfügt er über 60 als Integrierer oder Summierer schaltbare Verstärker, ca. 30 weitere Verstärker, 15 Parabelmultiplizierer und 12 Komparatoren. Die Verstärker sind in 10 Volt Technik ausgeführt.

Mit Hilfe von zusätzlich vorhandenen, auf einem Logiksteckbrett beliebig koppelbaren Logikelementen ist neben den üblichen Rechenarten eine frei programmierbare Rechnersteuerung möglich. Dabei können einzelne Integrierer unabhängig vom vorgegebenen Rechentakt gesteuert werden. Statussignale zeigen den Zustand des Rech-

ners an und können nach außen geleitet werden, umgekehrt ist es auch möglich, den Rechner durch ein äußeres Ereignis gesteuert anzuhalten oder in eine andere Betriebsart zu bringen.

4.2.2 Beschreibung des gewählten Mikrorechners

4.2.2.1 Allgemeines

Die Begriffe Mikroprozessor und Mikrocomputer bzw. -rechner sind im allgemeinen Sprachgebrauch nicht klar unterschieden. Man kann jedoch sagen, daß unter Mikroprozessor hauptsächlich eine durch wenige hochintegrierte Bausteine realisierte Rechnerzentraleinheit verstanden wird. Als Mikrorechner wird dagegen eine Maschine aus einer solchen Zentraleinheit und zusätzlichen Einrichtungen wie Speichern oder Anzeigen bezeichnet.

Die heute auf dem Markt angebotenen Mikrocomputer sind frei programmierbare Digitalrechner, sie unterscheiden sich in ihrem grundsätzlichen Konzept nicht von anderen, weitaus leistungsfähigeren Maschinen. Ihre Zentraleinheit besteht also aus einem Rechen- und einem Steuerwerk. Das Steuerwerk enthält die Mikroprogramme, die nötig sind, um einzelne Befehle des Befehlssatzes aus dem Speicher in die Zentraleinheit zu holen, zu entschlüsseln und im Rechenwerk auszuführen.

4.2.2.2 Auswahl des Mikrorechners

Der Mikrorechner IMP 16 P der Firma National Semiconductor wurde ausgewählt, weil er zum Zeitpunkt der Beschaffung (Mitte 1975) einer der wenigen Rechner war, die eine Wortlänge von 16 bit aufwies, und weil er mit einem Entwicklungssystem angeboten wurde. Die Wortlänge von 16 bit ist bei der Verarbeitung von analogen Daten vorteilhaft, da jeder analoge Wert nach Wandlung mit einem üblichen 12 bit Analog-Digital-Wandler in einem einzigen Speicherplatz abgelegt werden kann.

Außerdem bieten 16 bit Maschinen in der Regel einen leistungsfähigeren Befehlssatz als 8 bit Rechner. Die Entscheidung war richtig, da gerade das Entwicklungssystem - auf das unten näher eingegangen wird - für die Softwareerstellung unentbehrlich war.

4.2.2.3 Der Mikroprozessor IMP 16 C

Der IMP 16 C ist ein Mikroprozessor mit einer Befehls- und Datenwortlänge von 16 bit. Er bildet eine komplette Rechnerzentraleinheit und ist auf einer Steckkarte untergebracht. Seine wichtigsten Funktionselemente sind in Abb. 4.2 dargestellt. Die Zentraleinheit verfügt u.a. über 4 Register und einen Stack von 16 Worten Tiefe. Der Datenaustausch mit anderen Rechnerkomponenten oder der Peripherie erfolgt über 4 Datenbusse a 16 bit. Es sind dies

- der Adressbus (ADX in Abb. 4.2), auf ihm werden von der Zentraleinheit sowohl die Adressen der Speicherplätze als auch der peripheren Geräte übertragen,
- der Bus für die Ausgabe aus der Zentraleinheit in die Speicher bzw. die Peripherie (BDO),
- der Bus für die Eingabe vom Speicher in die Zentraleinheit (MDO),
- der Bus für die Eingabe von der Peripherie zur Zentraleinheit (SW).

Neben anderen sind diese Signale auf die Steckerleiste der Karte geführt.

Die Mikroprogramme des Steuerwerkes sind in sogenannten CROMs (Control Read Only Memory) untergebracht, das Rechenwerk besteht aus 4 RALU (Register And Logic Unit) genannten hochintegrierten Bausteinen. Jede RALU bearbeitet 4 bit eines 16 bit Wortes parallel. Diese Aufteilung ist für den Benutzer im übrigen unmerklich, man spricht von einem "bit sliced computer". Abb. 4.3 zeigt die Realisierung dieses Konzeptes. Zusätzlich sind auf der Karte noch die ROMs (Read Only Memory) der Fließkommaarithmetik (s.u.) untergebracht.

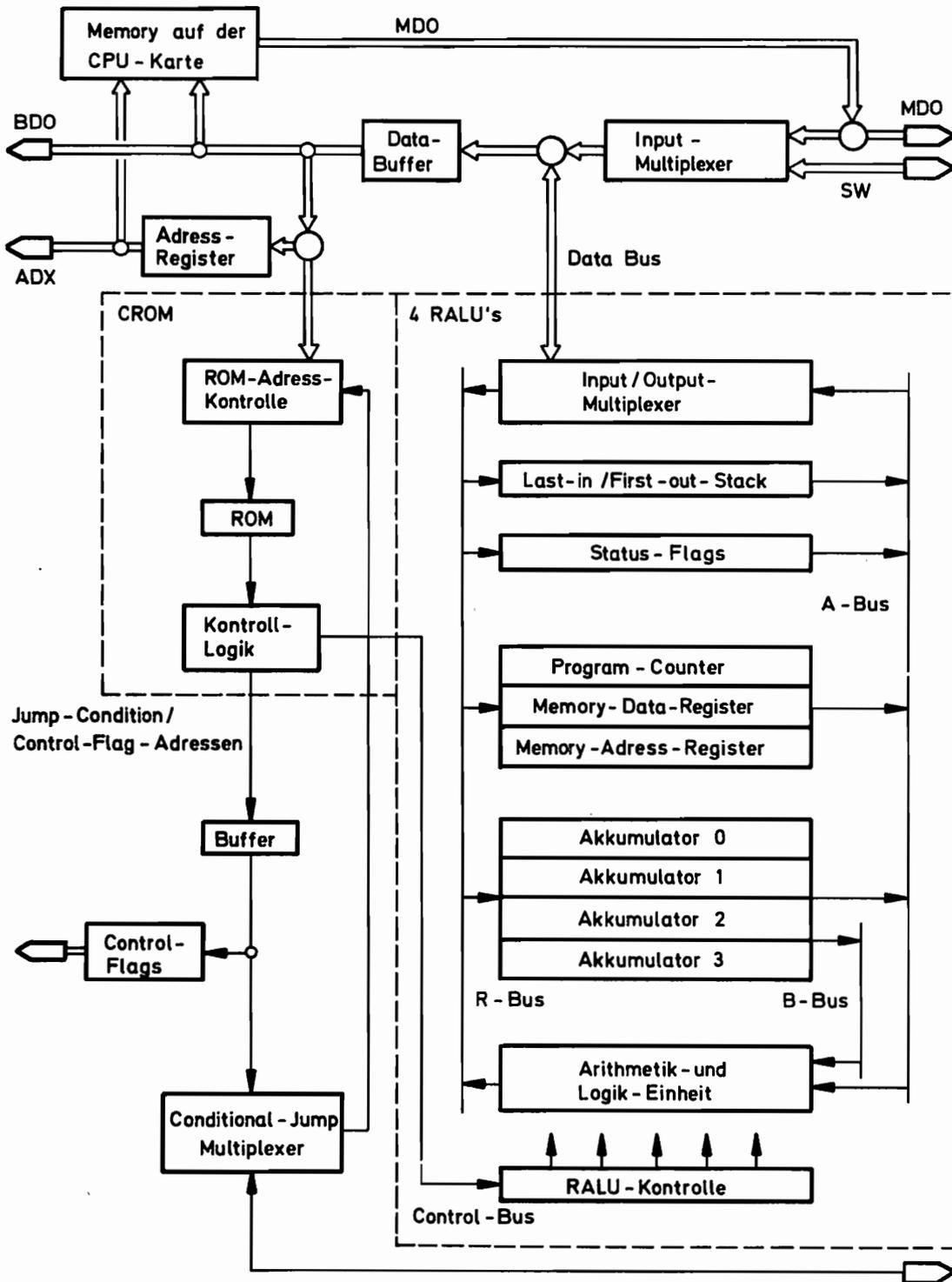


Abb. 4.2: Struktur des Mikroprozessors IMP 16 C nach [26]

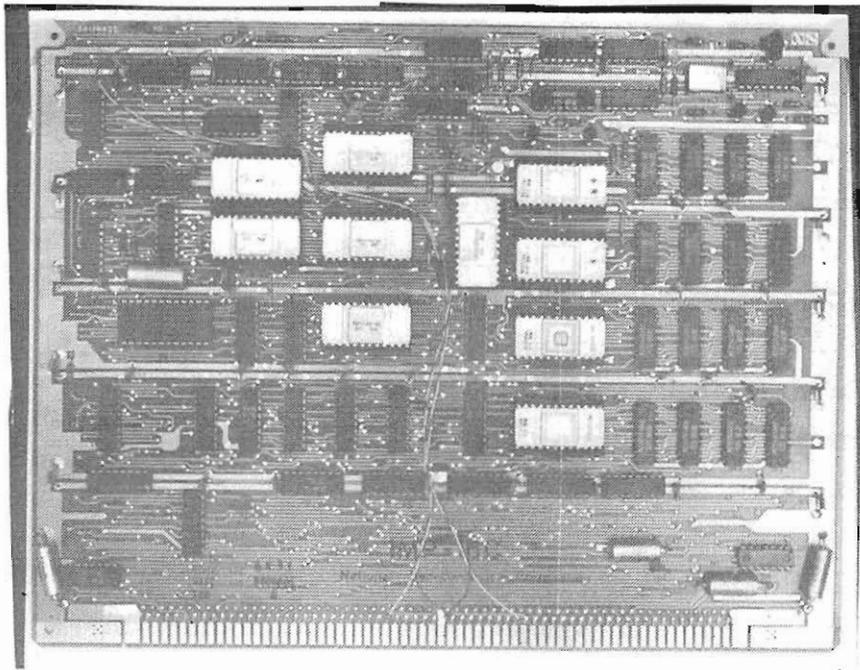


Abb. 4.3: Mikroprozessor IMP 16 C

4.2.2.4 Der Mikrocomputer IMP 16 P

Der Mikrocomputer IMP 16 P ist ein Rechensystem mit dem IMP 16 C als Zentraleinheit. Dazu kommen noch 2 Karten Schreib-Lese-Speicher (Grundausstattung) mit je 4 K Worten, ein Bedienungsfeld sowie das Interface zu einem Fernschreiber. Die vom Hersteller mitgelieferte Software umfaßt einen Assembler, einen Editor zur Bearbeitung von Quelltexten, einen Lader u.a. Auf Assembler und Lader wird unten noch eingegangen. Ein Absolutlader sowie die Programme zur Versorgung des Bedienungsfeldes sind in ROMs dauerhaft gespeichert. In einem Gehäuse von ca. 30 x 40 x 60 cm können außerdem auf zusätzlichen Einschubplätzen vom Benutzer gebaute Einheiten untergebracht werden.

Mit der beschriebenen hard- und softwareseitigen Ausstattung ist der IMP 16 P ein Entwicklungssystem zur Entwicklung und Erprobung von Programmen und Schaltungen für spezielle Mikrocomputeranwendungen. In einer solchen Anwendung würde man etwa nur noch die Zentraleinheit und einige RAM- und ROM-Speicher zusammen mit peripheren Ein-Ausgängen und einer Spannungsversorgung benötigen. Außerdem läßt sich der IMP 16 - wie hier geschehen - als Spezialrechner zur Meßwerterfassung, hybriden Optimierung etc. verwenden.

Durch Erweiterungen um

- einen schnellen Lochstreifenleser,
- einen kleinen Plattenspeicher mit 2 Laufwerken (Floppy Disk)

konnten die bei der Programmentwicklung nötigen Arbeitsschritte erheblich beschleunigt werden. Ferner gelang es, ein für andere Zwecke angeschafftes Sichtgerät der Firma Tectronix an den Mikrorechner anzuschließen, dadurch wurde auch die Erstellung von Zeichnungen leicht möglich.

4.2.3 Aspekte der rein digitalen Datenverarbeitung mit dem Mikrorechner

4.2.3.1 Compiler, Assembler, Lader

Die Entwicklung von Mikroprozessoren und -computern befindet sich noch stark im Fluß, laufend werden neue Produkte angeboten. Bis vor kurzem wurden dabei von den Herstellern keine höheren Programmiersprachen wie ALGOL oder FORTRAN auf Mikrorechnern implementiert. Das hat einen Grund darin, daß es schwierig ist, Sonderperipherie wie A/D Wandler, in Echtzeit arbeitende Statusmelder etc. in normales FORTRAN einzubeziehen. Nach Wissen des Verfassers sind auch die heute für Mikrorechner angebotenen FORTRAN-Compiler dazu nicht in der Lage. Die Programmierung erfolgte daher vollständig in Assemblersprache. Der komplette Befehlssatz ist im Anhang dokumentiert.

Von größter Bedeutung für eine schnelle Programmentwicklung und die Realisierung gut strukturierter Programme sind die Leistungsfähigkeit des Assemblers und des Laders. Der IMP 16 ist in dieser Hinsicht gut ausgestattet, beide Programme sollen kurz charakterisiert werden.

Der Assembler bietet die Möglichkeit, sogenannten relokalisierbaren Code zu erzeugen. Das bedeutet, daß der Programmierer einem Programm für die Assemblierung keinen festen Speicherbereich zuweisen muß, vielmehr wird der Speicherbereich erst während des Ladens festgelegt. So ist es möglich, Unterprogramme ohne Rücksicht auf die Speicherbelegung durch später zu entwickelnde Hauptprogramme zu schreiben. Ebenfalls von großer Bedeutung ist die Möglichkeit, in einem Assemblerprogramm Marken global für verschiedene Programme zu vereinbaren. Ist etwa die Startadresse eines Unterprogramms als global deklariert, so kann dieses Unterprogramm von einem anderen Programm aus aufgerufen werden, ohne daß während der Assemblierung der aktuelle Wert der globalen Startadresse bekannt wäre. Erst während des Ladens wird die aktuelle Adresse ermittelt und der Maschinencode für den Sprung in das Unterprogramm erzeugt.

Diese Eigenschaften des Assemblers und Laders ermöglichen es, die unten behandelten allgemeinen Hilfsprogramme, Programme zur hybriden Optimierung etc. als in sich geschlossene, universell einsetzbare Softwarepakete zu konzipieren. Ein Beispiel dazu bilden die in Abschnitt 4.2.5 behandelten allgemeinen Rahmenprogramme zur Optimierung.

4.2.3.2 Adressierung

Der IMP 16 benutzt zur Darstellung der meisten Befehle ein Wort. Das bietet die Möglichkeit, neben der Codierung des Befehlsnamens und einiger anderer Informationen, wofür 8 bit benötigt werden, $256=2^8$ Speicherplätze

mit den verbleibenden 8 bit direkt zu adressieren.

Abb. 4.4 zeigt das Befehlsformat für den "Lade" Befehl.

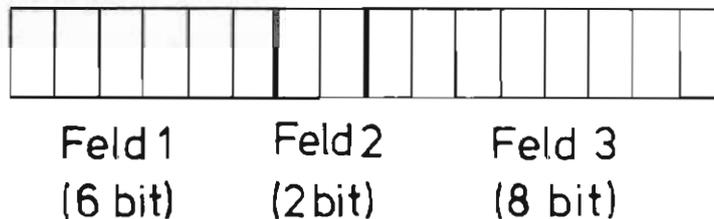


Abb. 4.4: Beispiel zum Befehlsformat

Feld 1 enthält den codierten Befehlsnamen, Feld 2 das angesprochene Register und die Adressierungsart (s.u.), Feld 3 dient zur Adressierung. Wegen der Möglichkeit, 256 Speicherplätze direkt zu adressieren, ist der gesamte Schreib-Lese-Speicher in 2 Bereiche geteilt, den sogenannten base-sector der ersten 256 Plätze und den alle anderen Plätze umfassenden top-sector. Damit sind folgende Adressierungsarten möglich:

I Direkte Adressierung

a Speicherplätze des base-sectors können von jedem anderen Speicherplatz aus adressiert werden. Feld 3 enthält die base-sector Adresse.

b Programmzähler - relative Adressierung

Der Programmzähler enthält während der Programmausführung stets die Adresse des nächsten auszuführenden Befehls. Diese am häufigsten verwendete Adressierungsart hat den Vorteil, daß man bei der Programmierung keine Rücksicht darauf zu nehmen braucht, in welchem Speicherbereich später das Programm während der Ausführung stehen wird, der Maschinencode wird im Speicher verschieblich. Feld 3 enthält die Differenz von Programmzähler und absoluter Adresse des angesprochenen Speicherplatzes.

c Indizierte Adressierung

Hierbei wird ein Register als Indexregister benutzt,

d.h. dieses Register enthält die benötigte Adresse. So ist eine direkte Adressierung des gesamten Speicherbereiches möglich.

Beispiel: LD 0, (3), d.h. Register 0 ist zu laden, Register 3 ist Indexregister. Enthält Register 3 die Zahl 1000, so wird Register 0 mit dem Inhalt der Speicherzelle 1000 geladen.

II Indirekte Adressierung

Indirekte Adressierung ist nur mit einigen Befehlen möglich (s. Anhang), sie kann dann mit jeder der drei oben beschriebenen Arten durchgeführt werden.

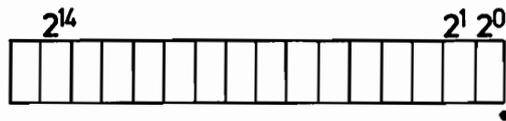
Beispiel: LD 0, @(3), indirekte indizierte Adressierung. Lade Register 0, Register 3 ist Indexregister. Der Inhalt der durch Register 3 adressierten Speicherzelle ist erst die Adresse des in Register 0 zu ladenden Operanden, nicht der Operand selbst. Enthält Register 3 die Zahl 1000 und die Speicherzelle 1000 die Zahl 2000, so wird Register 0 mit dem Inhalt von Speicherzelle 2000 geladen.

4.2.3.3 Zahlendarstellung

Zur Darstellung von Zahlen werden im Mikrorechner IMP 16 P sechs verschiedene Möglichkeiten benutzt. Im einzelnen sind dies ganze Zahlen (Integer), doppelt lange ganze Zahlen, echte Brüche (fractional), doppelt genaue echte Brüche, Festkomma- und Fließkommazahlen. Die Darstellung ist binär, für negative Werte wird jeweils das Zweierkomplement gebildet, bei Fließkommazahlen getrennt für Mantisse und Exponent. Abb. 4.5 zeigt die verschiedenen Formate der Zahlendarstellung mit dem jeweiligen Geltungsbereich und der erreichbaren Genauigkeit.

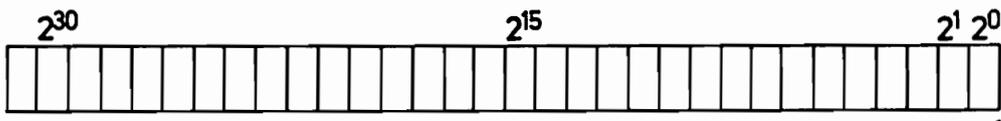
Während der Befehlssatz Befehle zur Addition, Subtraktion, Multiplikation und Division für einfache Integerzahlen und für Addition und Subtraktion doppelt genauer Integerzahlen enthält, müssen für die anderen Zahlendarstellungen diese Verknüpfungen durch Unterprogramme

Integer



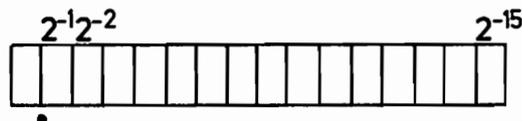
$-2^{15} = -32768 \leq x \leq 2^{15} - 1 = 32767$ Genauigkeit 1

Integer doppelt lang



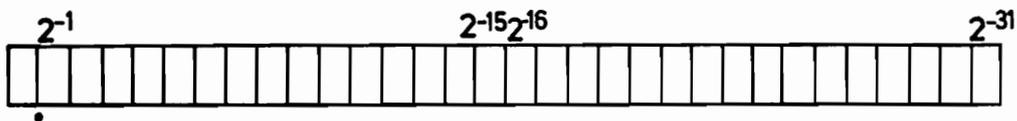
$-2^{31} \approx -2,14 \cdot 10^9 \leq x \leq 2^{31} - 1$ Genauigkeit 1

Fractional



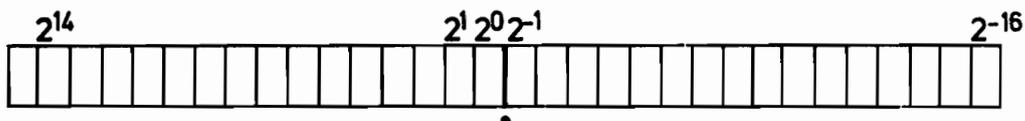
$-1 \leq x \leq 1 - 2^{-15}$ Genauigkeit $2^{-15} \approx 3 \cdot 10^{-5}$

Fractional doppelt genau



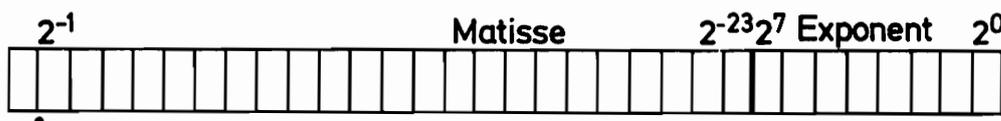
$-1 \leq x \leq 1 - 2^{-31}$ Genauigkeit $2^{-31} \approx 4,7 \cdot 10^{-10}$

Festkomma



$-2^{16} \leq x \leq 2^{15} + (1 - 2^{-16})$ Genauigkeit $2^{-16} \approx 1,5 \cdot 10^{-5}$

Fließkomma



$-2^{127} \approx -1,7 \cdot 10^{-38} \leq x \leq 2^{127} - 1$ Genauigkeit $1,2 \cdot 10^{-7} \cdot 2^{\text{Exponent}}$

Abb. 4.5: Möglichkeiten der Zahlendarstellung

realisiert werden. Speziell für die Fließkomma- und Fractionalarithmetik steht ein Programmpaket auf ROMs zur Verfügung. Wichtig ist in diesem Zusammenhang, daß die Deutung eines bestimmten Bitmusters z.B. als Fractionalzahl nicht aus dem Wert selbst hervorgeht, sondern ausschließlich durch den Programmierer bestimmt wird.

4.2.3.4 Rechenzeiten

Der IMP 16 hat eine Zykluszeit von 1,4µs. Die Rechenzeit für die Ausführung von Optimierungsprogrammen wird entscheidend von den Rechenzeiten für die arithmetischen Grundrechenarten bestimmt. Tab. 4.1 gibt eine Übersicht über typische Rechenzeiten bei Verknüpfung von Zahlen desselben Typs:

	+	-	*	÷
Integer	8	8	161	200
Integer doppelt genau	19	19	-	-
Fractional	8	19	356	506
Fractional doppelt genau	19	102	1 683	3 911
Festkomma	19	19	-	3 911
Fließkomma	938	1 221	7 768	10 080

Tab. 4.1 Rechenzeiten in µs

Bei fehlenden Zahlenangaben wurde für die entsprechende Operation kein Unterprogramm geschrieben. Aus der Tabelle geht hervor, daß insbesondere der Komfort der durch Unterprogramme realisierten Fließkommaarithmetik durch hohe Rechenzeiten erkauft werden muß. Ist die Anwendung zeitkritisch, so läßt sich in manchen Fällen Rechenzeit sparen, wenn man von vornherein über die Größenordnung

$\frac{a}{b}$	Integer	Integer doppelt lang	Fractional	Fractional doppelt genau	Festkomma
Integer	a < b: JSR @DIVID → Fractional	a < b: JSR @DPDIV → Frac. d. g.	DIV und schieben → Integer	JSR @DPDIV → Festkomma	a < b: JSR @DPDIV → Frac. d. g.
Integer doppelt lang	DIV → Integer	a < b: JSR @DPDIV → Frac. d. g.	/	/	/
Fractional	DIV → Fractional	/	a < b: JSR @DIVID → Fractional	a < b: JSR @DPDIV → Frac. d. g.	a < b: JSR @DPDIV → Frac. d. g.
Fractional doppelt genau	/	/	a < b: JSR @DPDIV → Frac. d. g.	a < b: JSR @DPDIV → Frac. d. g.	/
Festkomma	a < b: JSR @DPDIV → Frac. d. g.	/	DIV und schieben → Integer	JSR @DPDIV → Festkomma	a < b: JSR @DPDIV → Frac. d. g.

DIV: Assemblerbefehl

Tab. 4.2: Rechenzeitparende Möglichkeiten zur Division positiver Zahlen

DIVID: Unterprogramm für Fractionalzahlen

DPDIV: Unterprogramm für d. g. Fractionalzahlen

a · b a	Integer b	Integer doppelt lang	Fractional	Fractional doppelt genau	Festkomma
Integer	MPY → Int. d. l.	/	JSR @ MUL → Festkomma	JSR @ DPMUL → Festkomma	JSR @ DPMUL und schieben → Int. gerundet
Integer doppelt lang		/	JSR @ DPMUL → Int. d. l.	JSR @ DPMUL → Int. d. l.	/
Fractional			JSR @ MUL → Frac. d. g.	JSR @ DPMUL → Frac. d. g.	JSR @ DPMUL → Festkomma
Fractional doppelt genau		symmetrisch		JSR @ DPMUL → Frac. d. g.	JSR @ DPMUL → Festkomma
Festkomma					/

MPY: Assemblerbefehl

Tab. 4.3: Rechenzeitparende Möglichkeiten zur Multiplikation

MUL: Unterprogramm für Fractionalzahlen
 DPMUL: Unterprogramm für d. g. Fractionalzahlen

der Ergebnisse einzelner Operationen unterrichtet ist, etwa bei der Division positiver ganzer Zahlen mit einem Ergebnis kleiner als 1. Die Tabellen 4.2 und 4.3 zeigen für die Multiplikation und Division zeitsparende Möglichkeiten und den Zahlentyp des Ergebnisses. Dabei wurden nur Assemblerbefehle oder Unterprogramme des arithmetischen Unterprogramm Pakets berücksichtigt.

4.2.3.5 Allgemein benötigte Hilfsprogramme

Da die rechnerinterne Zahlendarstellung ausschließlich in binärer Form erfolgt, die peripheren Geräte dagegen eine BCD (Binary Coded Digital) Darstellung verwenden, wurden zur Ein-Ausgabe entsprechende Unterprogramme für Integer-, Fractional- und Fließkommazahlen geschrieben.

Für einige häufig benutzte mathematische Funktionen wie \sqrt{x} , e^x , $\sin x$, $\ln x$ wurden Unterprogramme entwickelt. Auf sie soll im einzelnen nicht eingegangen werden, da das Vorgehen stets nach folgendem Schema erfolgt: Der ganze zulässige Bereich des Arguments wird unterteilt in ein elementares Intervall, z.B. $[0, 1]$ für die Funktion e^x , und den übrigen Bereich. In dem elementaren Intervall wird die jeweilige Funktion angenähert durch ein Polynom oder eine rationale Funktion. Hart [28] gibt dazu Approximationen unterschiedlicher Genauigkeit an. Die Berechnung von Funktionswerten für Argumente außerhalb des elementaren Intervalls wird auf die Berechnung solcher aus dem Intervall zurückgeführt. Beispielsweise ist

$$e^x = 2^{\text{ld } e^x} = 2^y \quad \text{mit } y = \text{ld } e \cdot x$$

$$y \text{ wird zerlegt: } y = n + \xi, \quad n \text{ ganz, } 0 \leq \xi < 1 \quad (4.1)$$

$$\text{damit: } e^x = 2^{n+\xi} = 2^n \cdot 2^\xi$$

Es braucht nur die Funktion 2^{ξ} für $0 \leq \xi \leq 1$ approximiert zu werden, die Multiplikation mit 2^n wird durch eine Addition von n zum binären Exponenten des Fließkommagergebnisses ausgeführt.

Da die meisten Algorithmen zur Parameteroptimierung in vektorieller Schreibweise formuliert sind, wurde ein Programmpaket zur Matrizenrechnung mit Fließkommazahlen geschrieben. Zu den Hilfsprogrammen sind ferner Programme zur graphischen Darstellung von Funktionsverläufen auf dem Telectronix Sichtgerät zu zählen.

4.2.3.6 Programmierung eines Zufallszahlengenerators

Für die Realisierung der in Abschnitt 3.1.2.2 dargestellten Zufallsmethode werden normalverteilte Zufallszahlen benötigt. Da normalverteilte Zufallszahlen aus zwischen 0 und 1 gleichverteilten erzeugt werden können, wird zunächst auf diese eingegangen.

Grundsätzlich ist zu sagen, daß alle Methoden zur Erzeugung von Zufallszahlen mit einem Digitalrechner vollständig determinierte, periodische Zahlenfolgen liefern, größere Mengen der so erzeugten Zahlen haben jedoch die gewünschten statistischen Eigenschaften. Nach Knuth [29] sind die bei weitem besten Methoden, gleichverteilte Zufallszahlen zu erhalten, Spezialfälle des Algorithmus:

Wähle natürliche Zahlen m , $x_0 < m$, $a < m$, $c < m$. Erzeuge x_{n+1} aus x_n durch die Vorschrift

$$\begin{aligned} x_{n+1} &= (a x_n + c) \bmod m \quad \text{und} \\ u_{n+1} &= \frac{x_{n+1}}{m} \end{aligned} \quad (4.2)$$

Darin bedeutet $x \bmod m$ den bei der Division x/m entstehenden Rest. Die so erzeugten u_n sind dann bei passender Wahl von a , c , m zwischen 0 und 1 gleichverteilt. Die Wahl $m = 32\,749$, größte Primzahl kleiner

$2^{15} - 1$, $a = 23\ 107$, $c=0$ führt zu einem Generator mit der größtmöglichen Periode (nämlich m) und einer zufriedenstellenden Verteilungsfunktion. Die Güte der Verteilungsfunktion wurde mit dem χ^2 -Test überprüft. Um Rechenzeit zu sparen, wurde die Division x_{n+1}/m nicht ausgeführt und x_{n+1} als Fractionalzahl aufgefaßt. Dieses Vorgehen entspricht einer Division $x_{n+1}/2^{15}$ und erscheint berechtigt, da $m/2^{15} = 0,9995$ ist, die Division durch m also kaum andere Ergebnisse brächte. Unangenehme Konsequenzen dieser Vereinfachung wurden nicht beobachtet.

Normalverteilte Zufallszahlen werden aus gleichverteilten nach folgender Vorschrift gewonnen:

Es seien r_i , $i = 1 \dots 5$ zwischen 0 und 1 gleichverteilte Zufallszahlen, dann ist

$$g = \sum_{i=1}^5 (r_i - 2,5) \sqrt{2,4} \quad (4.3)$$

näherungsweise normalverteilt mit Mittelwert 0 und Streuung 1. Der hier ebenfalls durchgeführte statistische Test ergab, daß die Hypothese "Normalverteilung" nicht aufgegeben zu werden braucht.

4.2.4 Schaltungen und Programme zur hybriden Datenverarbeitung

4.2.4.1 Analoge Rechenarten und Anforderungen an die Rechnerkopplung

Bei der Kopplung von Analog- und Digitalrechner stellt sich die Frage, wie der in Abb. 4.1 angedeutete Datenaustausch zwischen beiden Rechnern zeitlich geordnet ablaufen soll. Ein Lösungsweg folgt aus dem iterativen Charakter der numerischen Optimierungsmethoden: Im Laufe einer hybriden Optimierung dynamischer Systeme müssen ja unter Variation von Parametern oder Eingangsfunktionen die Systemdifferentialgleichungen immer wieder gelöst werden. Zur wiederholten Integration ist an jedem Analog-

rechner die Betriebsart "repetierendes Rechnen" vorgesehen, in ihr wiederholt sich ein Zyklus aus 3 Phasen. In der Anfangswertphase nehmen die Integrierer die vorgeschriebenen Anfangswerte an, in der Rechenphase werden die Differentialgleichungen gelöst, und in der Haltphase halten die Integrierer den am Ende der Rechenphase erreichten Wert. Die Dauer der einzelnen Phasen ist einstellbar, beim benutzten Analogrechner zwischen 1 ms und 100 s. Abb. 4.6 verdeutlicht den Rechenablauf.

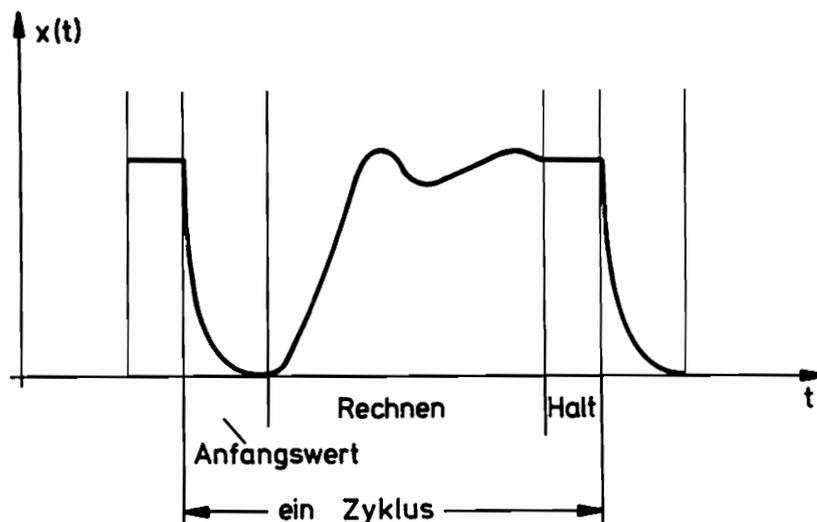


Abb. 4.6: Repetierendes Rechnen

Bei Wahl der Betriebsart repetierendes Rechnen müssen zur Parameteroptimierung während der Anfangswertphase die jeweils neuen Parameterwerte auf den Analogrechner gegeben werden, damit in der anschließenden Rechenphase die Gütefunktion zu diesen Parametern bestimmt werden kann. Zur Funktionaloptimierung müssen diskretisierte Funktionsverläufe während der Rechenphase vom Digitalrechner ausgegeben werden, in beiden Fällen wird der Wert der Gütefunktion in der Haltphase übernommen. Es liegt also der Fall vor, daß der Analogrechner den zeitlichen Ablauf bestimmt, der Digitalrechner hat zu folgen. Der führende Rechner wird auch als master bezeichnet, der folgende als slave.

Wenn auch das beschriebene Verhalten als Regelfall anzusehen ist, so sollte doch die Rechnerkoppelung auch den umgekehrten Fall - Digitalrechner master, Analogrechner slave - ermöglichen. Man denke beispielsweise an Optimierungsaufgaben, bei denen ein dynamisches System in kürzest möglicher Zeit in einen gewünschten Zustand gebracht werden soll. Hier ließe sich die Dauer der Rechenphase gar nicht im voraus festlegen und es wäre wünschenswert, abhängig von bestimmten Bedingungen die analoge Rechenphase vom Digitalrechner aus abbrechen zu können.

Es ergeben sich also folgende Forderungen an das hybride Rechensystem:

1. Synchronisierung der analogen und digitalen Rechnungen mit beliebigem master-slave Verhältnis.
2. Übernahme einzelner analoger Werte durch den Digitalrechner zu bestimmten Zeiten und Abtastung kontinuierlicher Funktionen.
3. Ausgabe einzelner Werte auf den Analogrechner zu bestimmten Zeiten und Ausgabe diskretisierter Zeitfunktionen.

In den folgenden Abschnitten ist dargestellt, wie sich diese Forderungen erfüllen lassen.

4.2.4.2 Synchronisierung von Analog- und Digitalrechner

Für die Synchronisierung der analogen und digitalen Rechnungen wurde ein Interface gebaut und für seine Benutzung verschiedene Unterprogramme geschrieben. Das Interface vermittelt den Datenverkehr zwischen Mikrorechner und digitalem Programmierfeld des Analogrechners. Die Informationen fließen dabei in beiden Richtungen. Während die Verbindungen zum Mikrorechner fest sind, ist die Kopplung zum digitalen Programmierfeld vom Benutzer durch einfache Steckverbindungen herzustellen. Abb. 4.7 zeigt die Frontplatte des Interfaces.

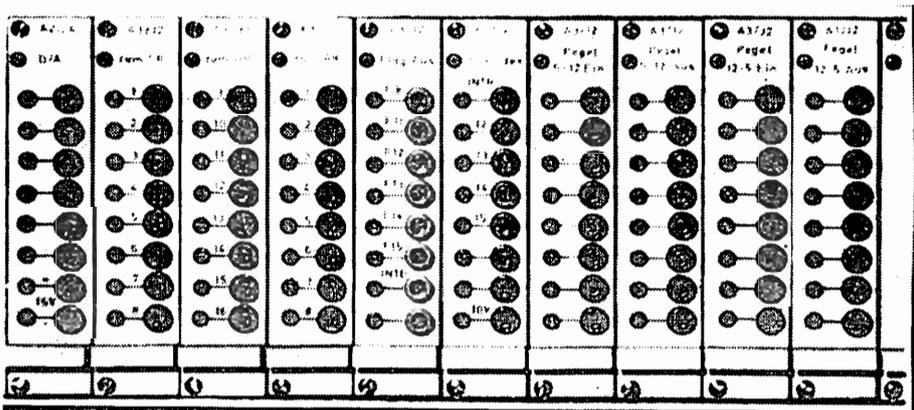


Abb. 4.7: Interface zur Rechnersynchronisierung

Funktionsmäßig gliedert es sich in 4 Teile:

1. Lesen und Beeinflussen des Analogrechnerstatus
Das digitale Programmierfeld wird vom Mikrorechner adressiert wie ein peripheres Gerät. So ist es möglich, durch RIN und ROUT Befehle die dem Interface durch Steckverbindungen zugeführten Statusinformationen des Analogrechners zu lesen bzw. den Betrieb des Analogrechners vom Digitalrechner aus zu steuern. Entsprechende Unterprogramme liefern die softwareseitige Unterstützung.
2. Ausgabe von Flags
Der IMP 16 P verfügt über 6 aus der Zentraleinheit herausgeführte, vom Programm her zu setzende Flags. Sie stehen am Interface für beliebige Steuerungen zur Verfügung.
3. Sprungbedingungen
Von den möglichen Bedingungen, unter denen bedingte Sprünge ausgeführt werden können, beziehen sich 4 auf den Status spezieller Eingänge des Mikrorechners, sie sind auf das Interface geführt.
4. Interrupteingang
Der ebenfalls vorhandene Interrupteingang ergänzt die Möglichkeiten zur Steuerung.

Damit kann der Digitalrechner z.B. auf einen bestimmten Status des Analogrechners warten, indem er permanent das entsprechende Statussignal einliest, bis das erwartete Ereignis eingetroffen ist.

4.2.4.3 Analog / Digital Wandlung

Zur A/D Wandlung wird ein 12 bit Wandler der Firma Data Translation eingesetzt. Er ist mit einem Multiplexer für 32 Kanäle auf einer speziell für den IMP 16 konzipierten Steckkarte untergebracht. Jede Wandlung wird entweder programmgesteuert oder durch einen externen Takt ausgelöst. Die Programmsteuerung kommt vor allem in Frage zur Wandlung der Gütefunktionswerte, die externe Triggerung für die Abtastung von Meßsignalen. Besonders bei der externen Triggerung sind einige Zeitbedingungen zu beachten.

Betrachtet wird dazu der Fall, daß ein Signal in äquidistanten Zeitabständen abgetastet werden soll. Die Grundbedingung für einwandfreie Resultate ist, daß die Taktperiode größer sein muß als die für die A/D Wandlung und Abspeicherung der digitalisierten Werte benötigte Zeit. Der verwendete Wandler benötigt pro Wandlung 10 μ s. Ein schnelles Programmstück für die Abtastung und Abspeicherung einer vorgegebenen Anzahl von Meßwerten lautet:

```
SCHLEIFE: RIN 2      ; Status des Wandlers lesen
          BOC 3, READ; Sprung wenn Wandlung fertig
          JMP SCHLEIFE

READ     : RIN      ; Datum lesen
          ST 0, (2) ; abspeichern
          AISZ 2,2  ; Speicheradresse erhöhen
          DSZ ZAEHL ; Zähler erniedrigen
          JMP SCHLEIFE
```

Für ein einmaliges Durchlaufen dieser Schleife ergibt sich eine Rechenzeit von 59,3 μ s, so daß unter Berücksichtigung des Shannon'schen Abtasttheorems Schwingungen mit Frequenzen von max. 8,4 kHz abgetastet werden können.

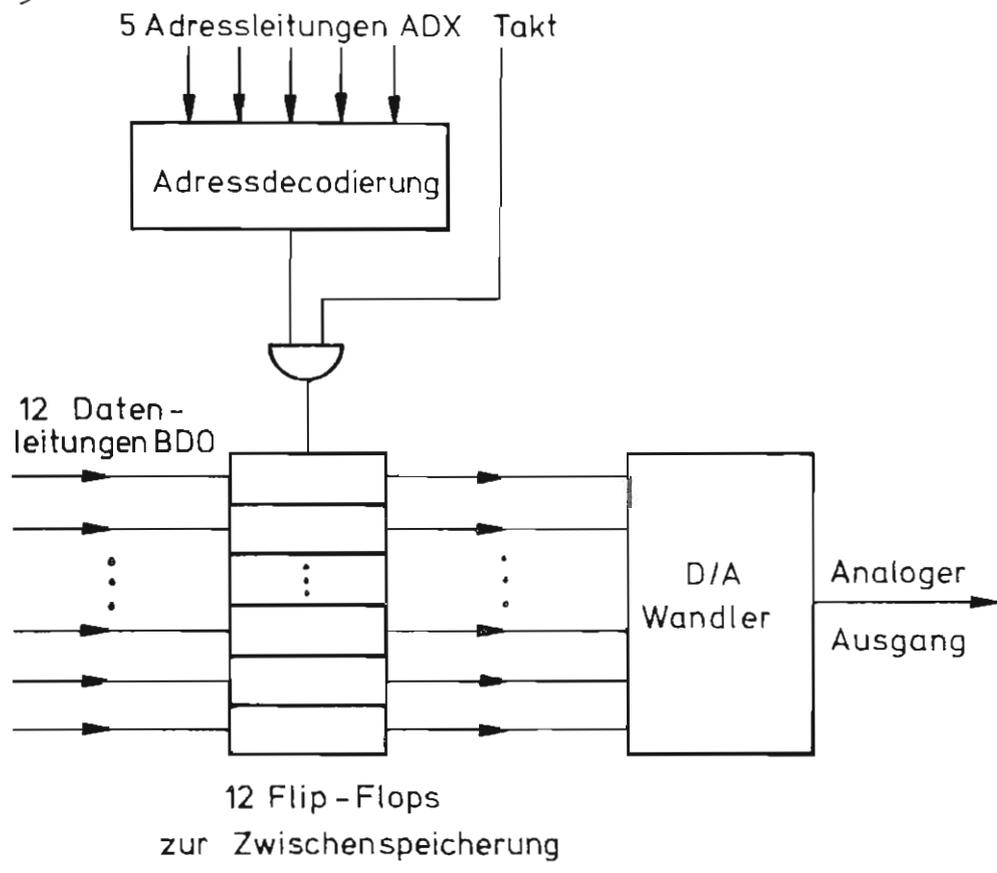


Abb. 4.8: Blockschaltbild zur D/A-Wandlung

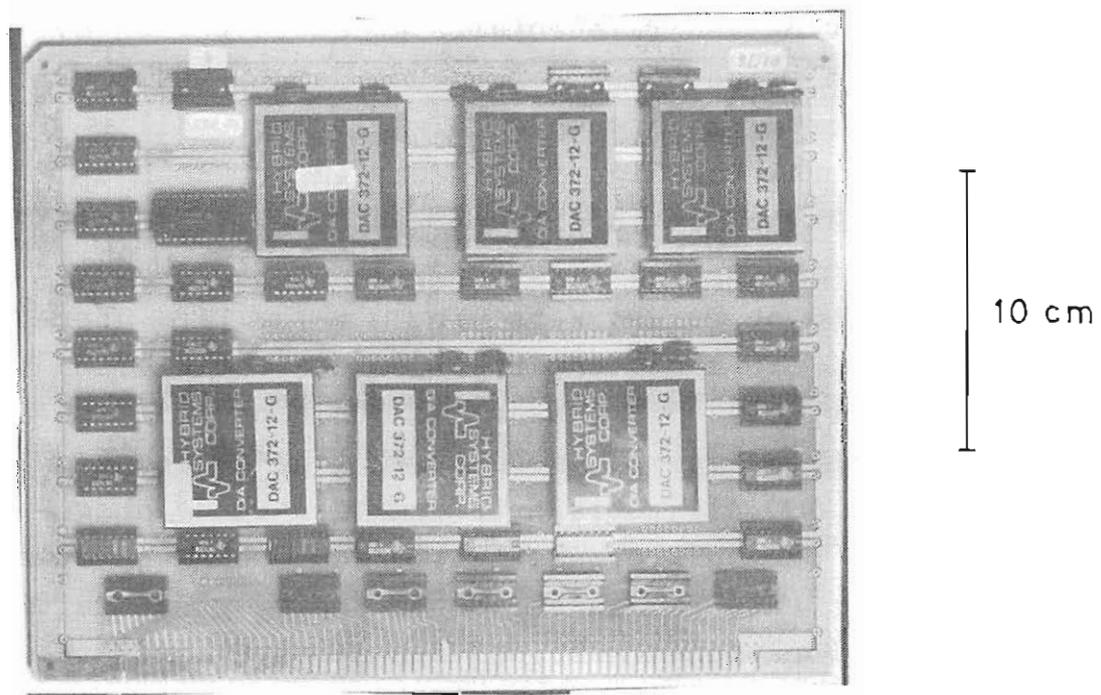


Abb. 4.9: D/A-Wandler Karte

4.2.4.4 Digital / Analog Wandlung

Zur Ausgabe von Parametern und diskretisierten Zeitfunktionen wurde eine Steckeinheit mit 6 unabhängig voneinander arbeitenden 12 bit D/A Wandlern gebaut. Die Abb. 4.8 und 4.9 zeigen ein Schaltbild und ein Photo der Karte.

Da die zur D/A Wandlung erforderliche Zeit von $5 \mu\text{s}$ in jedem Fall kürzer ist als die von einem ROUT Befehl benötigte, ist diese Wandlung nicht zeitkritisch. Jeder Wandler behält seinen Wert bei, bis ihm ein neuer zugewiesen wird.

Zur Funktionaloptimierung ist es nötig, eine oder mehrere in n Werte diskretisierte Zeitfunktionen während der analogen Rechenphase auf den Analogrechner auszugeben. Dauert die analoge Rechenphase T_R Sekunden, so ist alle T_R / n Sekunden die Ausgabe der nächsten Werte nötig.

Es ist wünschenswert, im Ausgabeprogramm nicht explizit auf T_R Bezug zu nehmen, da das Unterprogramm sonst für jede Änderung von T_R ebenfalls geändert werden müßte. Dies ist möglich unter Verwendung des am Analogrechner zur Verfügung stehenden Signals zur Oszillographensteuerung. Das Signal verläuft in der Rechenphase - unabhängig von ihrer Dauer - linear von 0 bis 1. Man kann nun dieses Signal laufend mit dem A/D Wandler abtasten und, wenn es den Wert $0/n, 1/n, \dots, n-1/n$ überschritten hat, die nächsten Werte der diskretisierten Funktionen ausgeben. Mit dieser Methode können in analogen Rechenphasen von 20 ms Dauer oder länger ohne weiteres etwa 100 diskretisierte Funktionswerte ausgegeben werden. Die auftretende Zeitverschiebung von etwa $30 \mu\text{s}$ zwischen entsprechenden Werten zweier gleichzeitig ausgegebener Funktionen kann bei den meisten Anwendungen toleriert werden.

Ist eine noch schnellere Ausgabe erwünscht, müssen andere Methoden, etwa programmierte Wartezeiten des Digitalrechners, verwendet werden. Eine solche Warteschleife

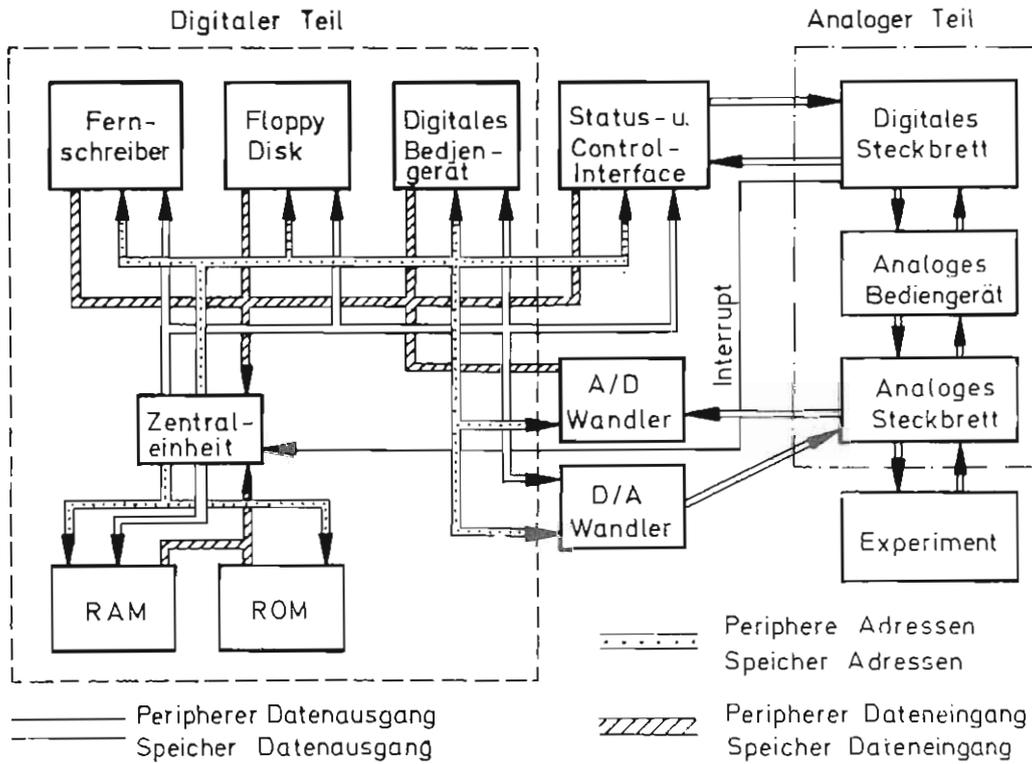


Abb. 4.10: Gesamtstruktur des hybriden Rechensystems

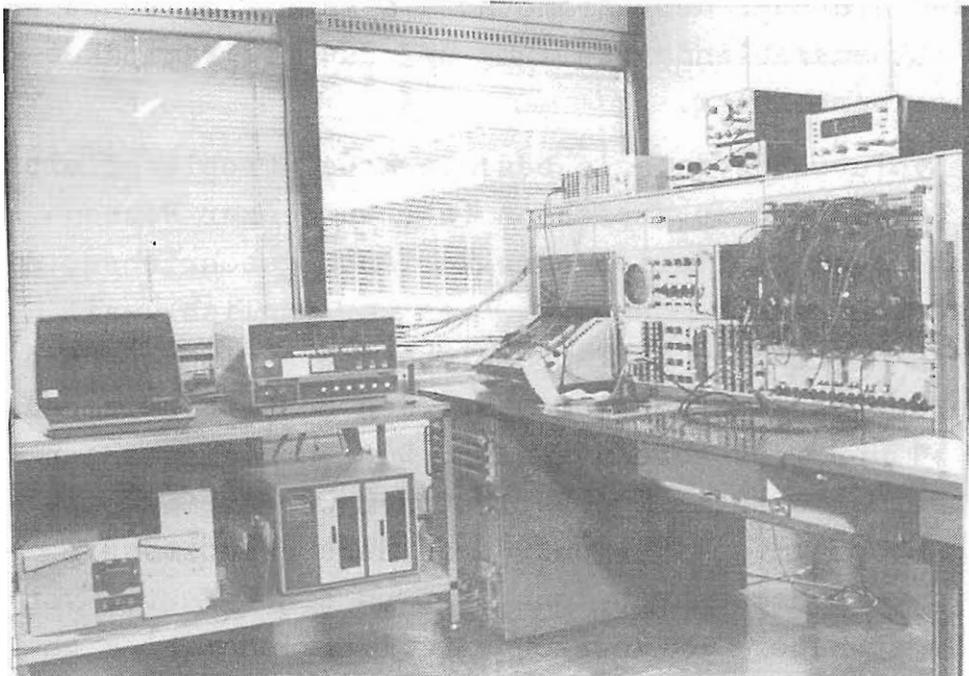


Abb. 4.11: Gesamtansicht des hybriden Rechensystems

wird in Abschnitt 5.3.1 behandelt. Die Flexibilität der oben beschriebenen Methode läßt sich so jedoch nicht erreichen.

Die zur hybriden Optimierung benötigten Geräte sind nun beschrieben, die Abbildungen 4.10,11 sollen ihr Zusammenwirken verdeutlichen.

4.2.5 Zur Programmierung der Optimierungsverfahren

Da es keine Optimierungsmethode gibt, die für alle Probleme gleich gut geeignet wäre, wurden für die Verfahren

- Methode des steilsten Abstieges (Einfache Gradientenmethode),
- Konjugierte Gradientenmethode ohne lineare Suche nach Davidon [10] ,
- Pattern search nach Hooke-Jeeves,
- Evolutionsstrategie nach Rechenberg

Unterprogramme geschrieben. Nach den Ausführungen von Abschnitt 3.1 braucht hier auf die einzelnen Methoden nicht mehr eingegangen zu werden. Alle Unterprogramme enthalten die Zahl der Parameter als Variable, benutzen die Fließkommazahlendarstellung und das Programmpaket zur Matrizenrechnung.

Da die Evolutionsstrategie besonders bei Problemen mit vielen Parametern zu empfehlen ist, wurde zur Rechenzeitverkürzung zusätzlich eine Variante geschrieben, die nur Fractionalzahlen benutzt. Dies war möglich, weil wegen des begrenzten Zahlenbereichs des Analogrechners alle Parameter und Funktionen auf den Bereich zwischen -1 und +1 normiert werden müssen. Die Bereichsgrenzen überschreitende zufällige Variationen werden im Optimierungsprogramm erkannt und durch neue ersetzt. Die rein digitale Rechenzeit konnte so um einen Faktor von etwa 20 gegenüber der Version mit Fließkommazahlen gesenkt werden.

Um trotz der umständlichen Assemblerprogrammierung zu einer nutzerfreundlichen Handhabung der Verfahren zu

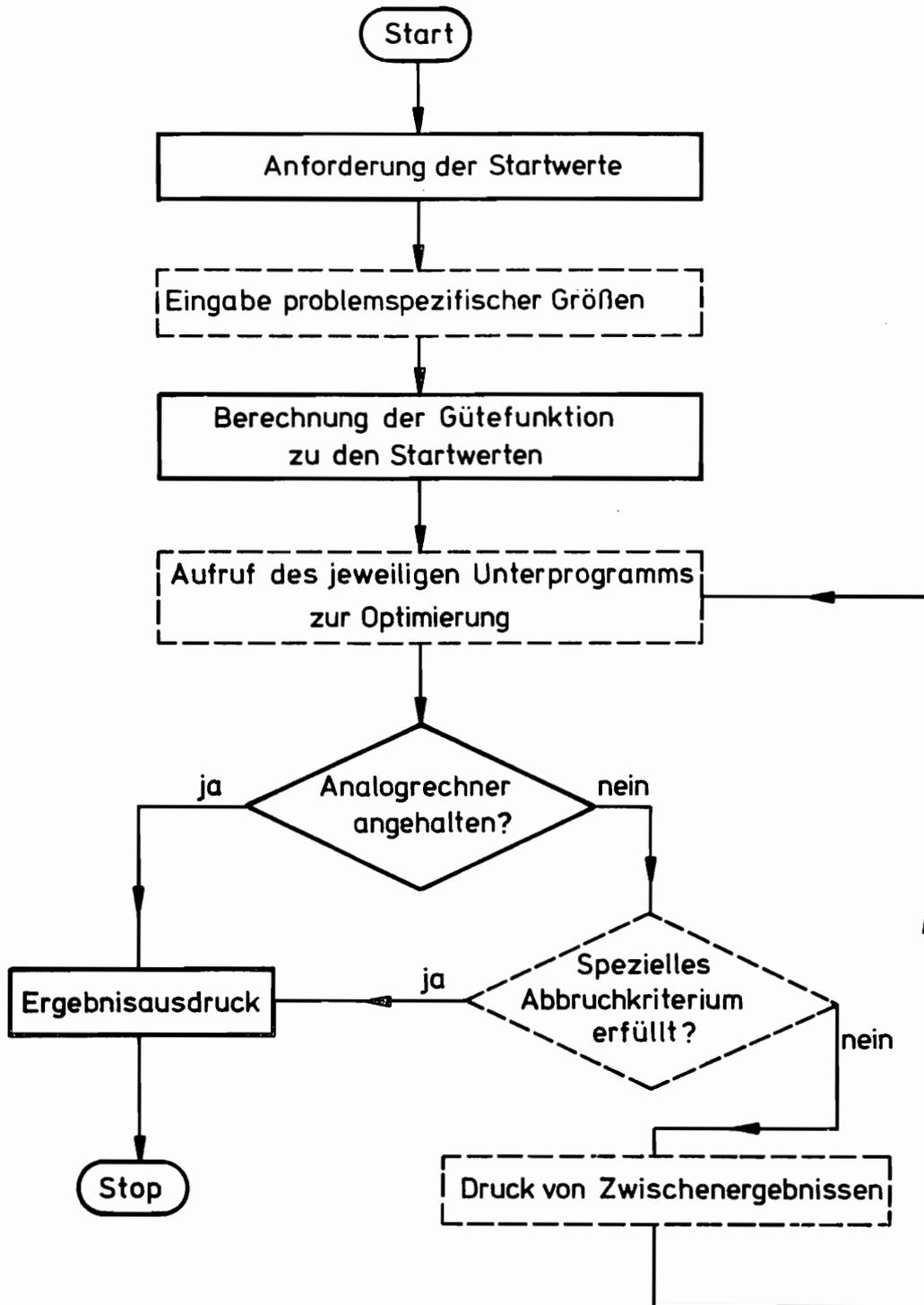


Abb. 4.12: Flußdiagramm zu den Rahmenprogrammen

kommen, wurde für alle Methoden gemeinsam je ein Rahmenprogramm zur Parameter- und zur Funktionaloptimierung entwickelt. Die Rahmenprogramme steuern den Ablauf der Optimierung von der Eingabe der Startwerte bis zur Ergebnisprotokollierung. An vorbereiteten Stellen kann der Benutzer spezielle Abschnitte zur Eingabe oder ein spezielles Abbruchkriterium in die Rahmenprogramme einfügen.

Innerhalb der Rahmenprogramme wird überprüft, ob der Analogrechner vom Benutzer angehalten worden ist, in diesem Fall erfolgt ein Abbruch der Optimierung und die Ergebnisse werden ausgedruckt. So ist eine einfache und praktische Möglichkeit geschaffen, die Optimierung zu beenden. Abb. 4.12 enthält ein vereinfachtes Flußdiagramm für die Rahmenprogramme. Vom Benutzer wahlweise einzusetzende Teile sind gestrichelt dargestellt, der Aufruf der gewählten Optimierungsmethode muß in jedem Fall eingefügt werden.

Durch die Rahmenprogramme konnte eine klare Trennung zwischen den allgemein verwendbaren Optimierungsunterprogrammen und nur im jeweiligen Anwendungsfall benötigten Programmteilen erreicht werden. Zur Lösung eines speziellen Problems ist es dann nur erforderlich, nach vorgegebenen Konventionen ein Unterprogramm zur Berechnung der Gütefunktion zu schreiben und in das Rahmenprogramm nach Bedarf spezielle Blöcke an vorbereiteten Plätzen einzufügen. Anschließend müssen alle Programme zusammen geladen werden.

4.2.6 Beurteilung des entwickelten Rechensystems

Zur Beurteilung des realisierten Rechensystems ist ein Vergleich mit einem in der Literatur [30] beschriebenen Hybridrechner, bestehend aus einem Analogrechner Telefunken RA 770 und einem Digitalrechner CIIC 90-40, nützlich.

Man erkennt schnell, daß beide Rechensysteme in ihren

wesentlichen Hardwarestrukturen zur D/A-, A/D Wandlung und zur Rechnersynchronisierung übereinstimmen.

Unterschiede bestehen in der Softwarestruktur, speziell bei der Behandlung der für das Hybridrechnen typischen Echtzeitbedingungen. Der in [30] dargestellte Hybridrechner benutzt dafür die Technik der Programmunterbrechungen. Beim Eintritt eines bestimmten äußeren Ereignisses wird dabei die Abarbeitung des gerade behandelten Programms im Digitalrechner unterbrochen und ein für den Unterbrechungsfall bereitgestelltes Programm vorgezogen. Das Umschalten von einem Programm zum anderen während des Betriebes bringt viele Probleme bezüglich der sicheren Fortsetzung des unterbrochenen Programms mit sich. Die in Abschnitt 4.2.4.2 behandelte Methode der "Quasi-Echtzeitprogrammierung" [30], bei der äußere Ereignisse nur nach einer programmierten Abfrage auf den Programmablauf einwirken können, ist demgegenüber wesentlich unkomplizierter und nur geringfügig langsamer als eine Programmunterbrechung.

Weiterhin verfügt der Mikrorechner nicht über ein Betriebssystem zur Steuerung und Überwachung aller digitalen Rechenprogramme. Da jedoch nach Ansicht des Verfassers die gleichzeitige Bearbeitung mehrerer Programme (Multiprogramming) die Grenzen der sinnvollen Nutzung eines heutigen Mikrorechners überschreitet, fällt dieser Mangel nicht sehr ins Gewicht.

Insgesamt gesehen ist also das realisierte Rechensystem mit seinen Schaltungen und Programmpaketen in wesentlichen Punkten einem konventionellen Hybridrechner ebenbürtig.

5. Anwendungen des entwickelten Rechensystems zur hybriden Optimierung

5.1 Allgemeine Hinweise und analoge Rechentechnik

In diesem Abschnitt sollen Anwendungen der oben beschriebenen Schaltungen und Programme vorgestellt werden. Einzelheiten der digitalen Programmierung sind nur soweit betrachtet, wie es sich nicht um Spezialfälle der besprochenen Rahmenprogramme handelt.

Bezüglich der analogen Rechentechnik wird auf die Literatur [30, 31] verwiesen. Bei Ammon [32] findet man eine Sammlung vorbereiteter Schaltbilder, u.a. für häufig vorkommende Übertragungsglieder der Regelungstechnik. Auf die beim Analogrechnen gelegentlich problematische Normierung soll an dieser Stelle pauschal für alle Anwendungen eingegangen werden. Es wurde jeweils zunächst nur eine Zeitskalierung vorgenommen, um kurze Rechenzeiten zu erhalten. Falls einzelne Rechenelemente dann schlecht ausgesteuert waren oder übersteuerten, erfolgte eine zusätzliche Amplitudennormierung nach der empfehlenswerten, übersichtlichen Matrizenmethode von Petersen und Vetter [33].

Bei allen Beispielen wurde die analoge Rechenart des repetierenden Rechnens gewählt. Die jeweils angegebenen Rechenzeiten beziehen sich auf den Analogrechner. Es kann vorkommen, daß bei einem Problem der Digitalrechner mit der Auswertung des in der analogen Haltephase übernommenen Gütefunktionswertes erst nach Beginn der nächsten Rechenphase fertig ist (s. Abschnitt 4.2.4.1). In diesem Fall muß der Digitalrechner bis zum Beginn des nächsten Zyklus des repetierenden Rechnens mit der Ausgabe von Parametern bzw. Funktionen warten. Die so hervorgerufene Verlängerung der Rechenzeit war jedoch stets geringfügig, so daß sie bei den Angaben zur Rechenzeit vernachlässigt worden ist.

5.2 Anwendungen zur Parameteroptimierung

5.2.1 Optimierung eines nichtlinearen Regelkreises

Betrachtet wird der in Abb. 5.1 dargestellte Regelkreis. Ein Begrenzer ist in Serie geschaltet mit einem linearen Übertragungsglied 3. Ordnung.

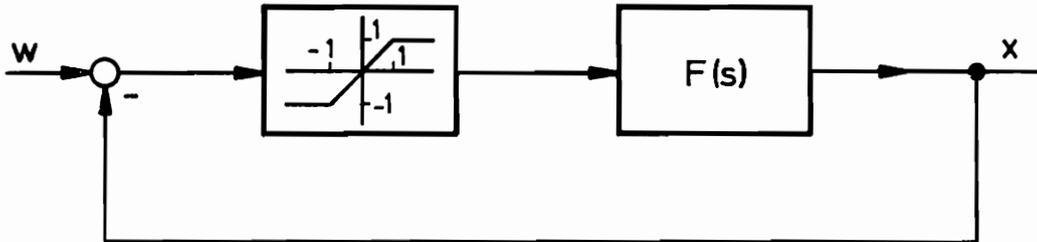


Abb. 5.1: Struktur des Regelkreises

Die Übertragungsfunktion des linearen Teils lautet

$$F(s) = \frac{33}{s(s+2)(s+3)} \quad (5.1)$$

Das nach Abb. 5.1 rückgekoppelte System hat einen Grenzzyklus (Abb. 5.2). Durch das Einführen eines Kompensationsgliedes in den Regelkreis soll die Stabilität des geschlossenen Kreises herbeigeführt und eine optimale Systemantwort für einen Sollwertsprung gewährleistet werden.

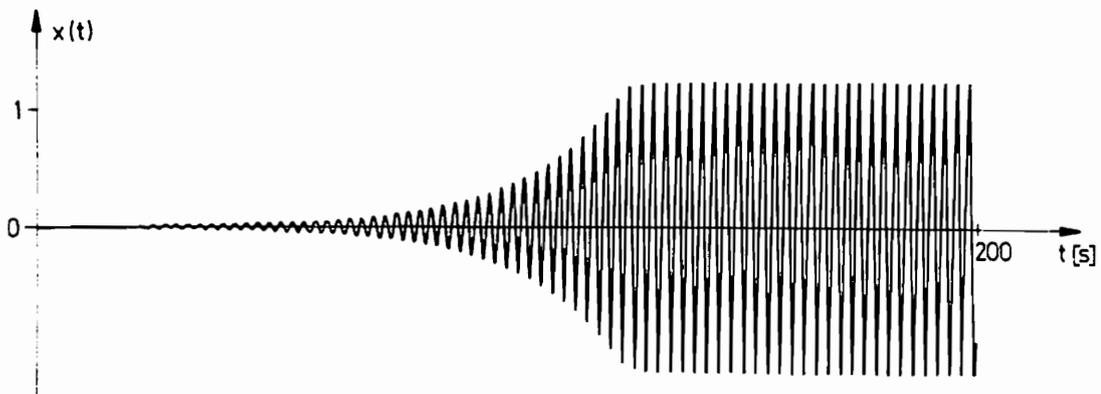


Abb. 5.2: Grenzzyklus des Kreises ohne Kompensator

Es wurde ein Kompensator des Typs

$$F_K(s) = K \frac{1 + T_1 s}{1 + T_2 s} \quad (5.2)$$

gewählt, Abb. 5.3 zeigt das Blockschaltbild des Regelkreises mit Kompensator.

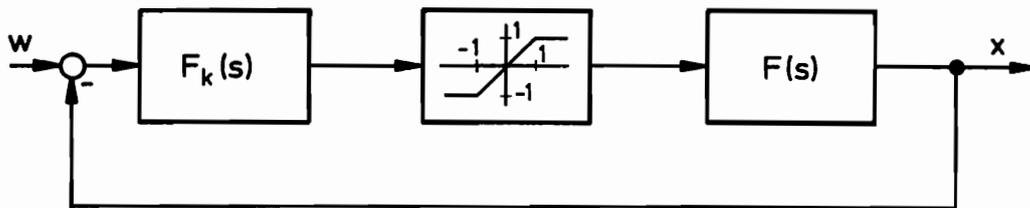


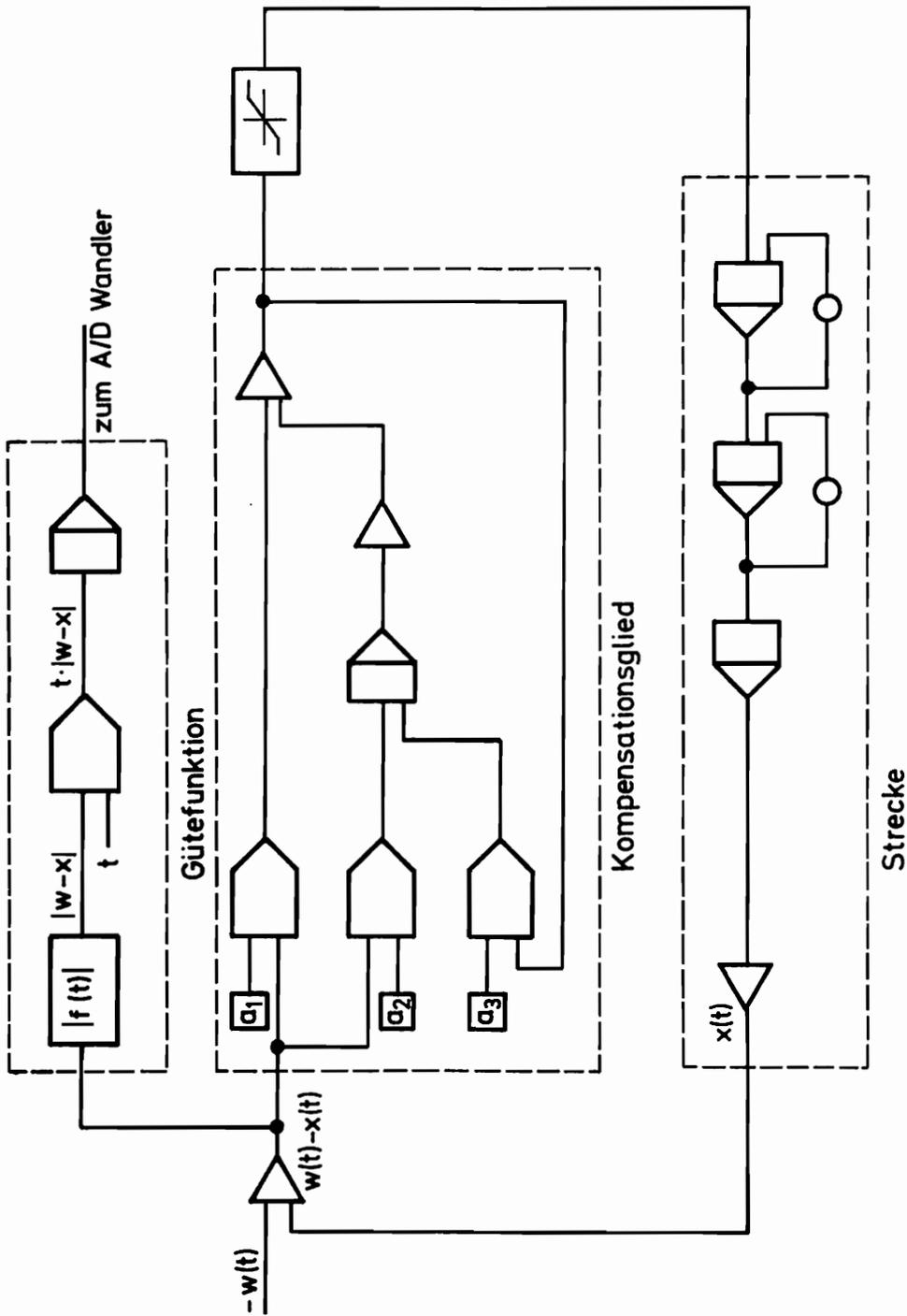
Abb. 5.3: Regelkreis mit Kompensator

Als Gütekriterium für die Optimierung des Kompensators wurde das ITAE (Integrated time multiplied absolute error) Kriterium verwendet, es ist also das Minimum von

$$G = \int_0^{T_e} t |x(t) - w(t)| dt \quad (5.3)$$

bezüglich der Parameter K , T_1 und T_2 zu bestimmen. Der Regelkreis wurde auf dem Analogrechner, der auch die Berechnung der Gütefunktion übernahm, nachgebildet. Abb. 5.4 enthält den Schaltplan. Die drei gesuchten Parameter wurden über D/A-Wandler und Multiplizierer in die Anologschaltung eingespeist. Nach Beobachtung des Systemverhaltens erfolgte die Festsetzung der oberen Integrationsgrenze zu $T_e = 20$ s. Optimiert wurde nach der ableitungsfreien Methode von Hooke-Jeeves.

Einen Hinweis auf mögliche Startwerte für die Optimierung gibt der Kompensatorentwurf für lineare Systeme [34]: Bei der Verwendung eines phasenhebenden Korrekturgliedes ($T_1 > T_2$) sollte der Pol des Korrekturgliedes in der komplexen Zahlenebene links von den



a_1, a_2, a_3 vom D/A-Wandler $a_1 = \frac{1}{T_1} \cdot K$ $a_2 = \frac{K}{T_2}$ $a_3 = \frac{1}{T_2}$

Abb. 5.4: Analogrechenplan zur Optimierung

Polen des offenen Kreises, hier 0, -2, -3 liegen. Mit der Wahl $T_1 = 0.125$, $T_2 = 0.1$, $K = 1$ lag ein phasenanehebendes Glied mit einem Pol bei -10 vor. Tab. 5.1 zeigt typische Ergebnisse der Optimierung. Die Resultate wurden nach 50 bis 100 Iterationen erreicht. Durch Zeitraffung um den Faktor 100 ergab sich eine analoge Rechenzeit von 0,2 s je Iteration, so daß die gesamte Optimierung nach spätestens 20 s abgeschlossen war.

	Startwerte			Endwerte		
	T_1	T_2	K	T_1	T_2	K
I	0,125	0,1	1,0	0,213	0,043	0,174
II	0,056	0,143	1,28	0,220	0,077	0,385

Tab. 5.1: Optimierungsergebnisse

Die Gütefunktion hat mehrere lokale Minima, da jedoch beide erhaltenen Lösungen zu einem befriedigenden Systemverhalten führen, wurde nicht nach dem globalen Minimum gesucht. Interessant ist, daß auch bei der Wahl eines phasenabsenkenden Korrekturgliedes ($T_1 < T_2$) als Startvorgabe doch wieder ein phasenanehebender Kompensator als optimal gefunden wurde. Beide ermittelten Lösungen zeigen für unterschiedlich große Sollwert-sprünge bis auf einen entsprechenden Faktor dasselbe Ausgangsverhalten, ein Grenzzyklus tritt nicht auf. Aufgrund dieser Beobachtungen wurde auf eine weitere Stabilitätsanalyse verzichtet. Abb. 5.5 zeigt die Sprungantworten für die Start- und Ergebniswerte aus Tab. 5.1.

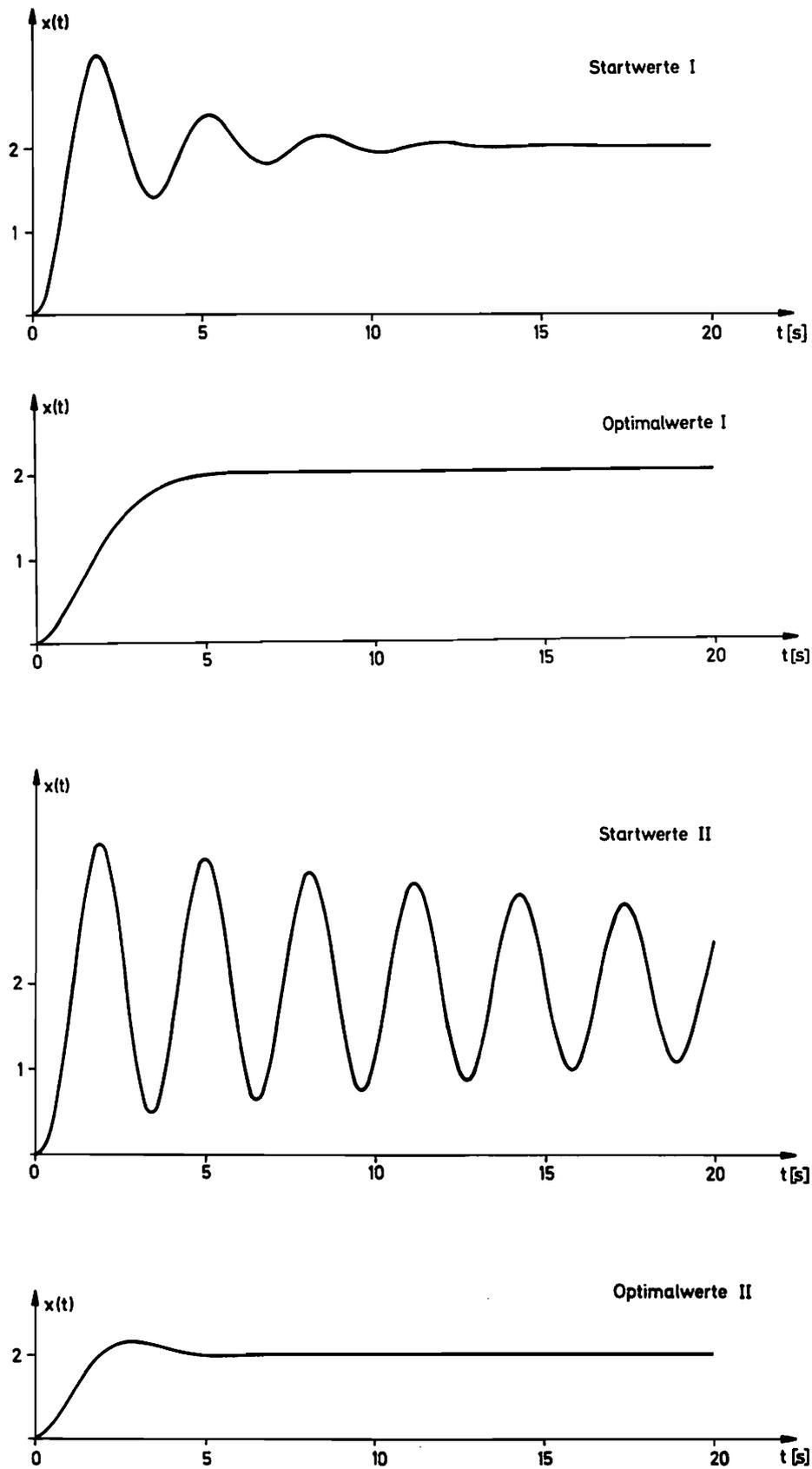


Abb. 5.5: Antworten auf einen Sollwertsprung
(Sprunghöhe 2)

5.2.2 Suboptimaler Entwurf eines Zustandsreglers

5.2.2.1 Die Problemstellung

Die Untersuchung von Mehrgrößenregelsystemen hat bei linearen Systemen zum Konzept der Zustandsvektorrückkoppelung geführt. Das Regelgesetz kann dabei sowohl bezüglich seiner Struktur als auch seiner Parameter optimal im Sinn eines bestimmten Gütekriteriums festgelegt werden. Die wichtigsten Resultate sollen nach Athans, Falb [2] für die Ausregelung von Störungen kurz dargestellt werden:

Gegeben sei das steuerbare, lineare zeitinvariante System

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{B} \underline{u}(t) \quad (5.4)$$

und das Gütekriterium

$$J = \int_0^{\infty} (\underline{x}^T \underline{Q} \underline{x} + \underline{u}^T \underline{R} \underline{u}) dt \quad (5.5)$$

Die Steuerfunktionen \underline{u} sollen unbeschränkt, \underline{Q} und \underline{R} positiv definite Matrizen sein. Dann existiert eine optimale Regelung, die das System von einem beliebigen Anfangszustand \underline{x}_0 in den Zustand $\underline{x} = \underline{0}$ überführt und das Gütekriterium minimiert. Die optimale Regelung ist gegeben durch

$$\underline{u}(t) = -\underline{R}^{-1} \underline{B}^T \hat{\underline{K}} \underline{x}(t) = \underline{K} \underline{x}(t) \quad (5.6)$$

Die konstante Matrix \underline{K} ergibt sich als Lösung der nicht-linearen algebraischen Matrixgleichung

$$-\hat{\underline{K}} \underline{A} - \underline{A}^T \hat{\underline{K}} + \hat{\underline{K}} \underline{B} \underline{R}^{-1} \underline{B}^T \hat{\underline{K}} - \underline{Q} = \underline{0} \quad (5.7)$$

Mit der so bestimmten Rückführung wird das Systemverhalten beschrieben durch

$$\dot{\underline{x}}(t) = \underline{G} \underline{x}(t) \quad , \quad \underline{x}(0) \text{ gegeben} \quad (5.8)$$

und

$$\underline{G} = \underline{A} - \underline{B} \underline{R}^{-1} \underline{B}^T \underline{K} \quad . \quad (5.9)$$

Die Eigenwerte der Matrix \underline{G} haben alle einen negativen Realteil, dies gilt auch, wenn Eigenwerte von \underline{A} einen positiven Realteil haben, d.h. instabile Systeme werden durch die Rückführung (5.6) stabilisiert.

Das Kontrollgesetz (5.6) hat folgende Vorzüge und Nachteile:

Vorzüge:

1. Das Kontrollgesetz hat eine einfache, lineare Struktur.
2. Die Reglerparameter lassen sich bei bekannten Systemmatrizen im voraus berechnen.
3. Das Systemverhalten ist optimal im Sinne des Gütekriteriums (5.5) für beliebige Anfangszustände \underline{x}_0 .

Nachteile:

1. Alle Zustandsgrößen müssen zurückgeführt werden.
2. Die Auswahl der Gewichtsmatrizen \underline{Q} und \underline{R} ist schwierig. Werden die Elemente von \underline{R} im Verhältnis zu den Elementen von \underline{Q} zu klein gewählt, erhält man unrealisierbare Verstärkungsfaktoren in der Rückführung.

Die Matrix \underline{K} muß man praktisch immer numerisch berechnen. Man erhält sie als stationäre Lösung einer (5.7) entsprechenden Matrizendifferentialgleichung für die Zustandsvektorrückführung zeitvarianter Systeme.

Der Hauptnachteil des Regelgesetzes (5.6) liegt in der Forderung nach der Rückführung - und damit Messung - aller Zustandsgrößen. In den meisten Fällen kann diese Forderung aus meßtechnischen Gründen oder wegen zu

hoher Kosten nicht erfüllt werden. Aus diesem Grund wird versucht, nicht gemessene Zustandsgrößen über einen Beobachter zu schätzen [35] oder optimale Rückführgesetze des Typs

$$\underline{u}(t) = -\underline{F} \underline{y}(t)$$

mit dem Vektor

$$\underline{y}(t) = \underline{C} \underline{x}(t)$$

der gemessenen Ausgangsgrößen $\underline{y}(t)$ zu finden. \underline{y} hat dabei weniger Komponenten als \underline{x} .

Die zur Ermittlung von optimalen Ausgangsvektorrückführungen von Levine, Athans [36] angegebene Methode erfordert numerische Berechnungen in einem Umfang, der weit über den zur Lösung von Gl. (5.7) nötigen hinausgeht. Im folgenden soll daher gezeigt werden, wie durch hybride Optimierung optimale Ausgangsvektorrückführungen bestimmt werden können.

5.2.2.2 Der Einsatz hybrider Optimierungsverfahren

Regelgesetze der Gestalt $\underline{u}(t) = -\underline{F}\underline{y}(t)$, die das Gütekriterium (5.5) minimieren, können als suboptimal bezeichnet werden. Sie sind einerseits durch einen schlechteren Wert des Gütekriteriums gekennzeichnet als der optimalen Rückführung (5.6) entspricht, andererseits werden nach Festlegung einer Kontrollstruktur, d.h. Festlegung, welche Größen rückgeführt werden sollen, die Parameter der Rückkoppelungsmatrix so bestimmt, daß das Gütekriterium einen möglichst geringen Wert annimmt.

Folgendes Vorgehen ist zweckmäßig:

1. Untersuchung der Steuerbarkeit des Systems
2. Festlegung der Gewichtsmatrizen \underline{Q} und \underline{R}
3. Festlegung einer Kontrollstruktur

4. Hybride Optimierung der Parameter der Rückführmatrix F
5. Beurteilung der erhaltenen Ergebnisse, gegebenenfalls Neufestsetzung der Gewichtsmatrizen und/oder Wahl einer anderen Kontrollstruktur und erneute Optimierung

Zur hybriden Optimierung wählt man einen Anfangszustand des Systems und ermittelt für diesen die suboptimalen Parameter. Wegen der Linearität des Systems und der Kontrollstruktur ist ein für einen beliebigen Anfangszustand \underline{x}_0 suboptimales Kontrollgesetz auch für jeden Anfangszustand $\alpha \underline{x}_0$ suboptimal. Anders ist die Situation bei zwei linear unabhängigen Anfangsvektoren, in diesem Fall braucht eine für den Anfangsvektor \underline{x}_{01} suboptimale Lösung nicht auch für den Anfangsvektor \underline{x}_{02} suboptimal zu sein. Um für beliebige Anfangswerte suboptimales Verhalten zu erzielen, genügt es, die Optimierung für p linear unabhängige Anfangsvektoren, etwa Einheitsvektoren, durchzuführen und die erhaltenen Ergebnisse zu mitteln. p ist dabei die Zahl der Zustandsgrößen des Systems.

5.2.2.3 Anwendungen beim Entwurf aktiver Schwingungsdämpfer für Eisenbahnwagen

Seit langem ist bekannt, daß Eisenbahnwagen selbsterregte Schwingungen in der horizontalen Ebene senkrecht zur Fahrtrichtung ausführen können. Ist die Geschwindigkeit des Wagens kleiner als eine bestimmte Grenzggeschwindigkeit v_{gr} , klingen etwa durch eine Unregelmäßigkeit des Gleises hervorgerufene Auslenkungen ab. Bei der Grenzggeschwindigkeit tritt eine stabile, Sinuslauf genannte Dauerschwingung auf (Abb. 5.6). Wird die Geschwindigkeit über die kritische hinaus erhöht, klingen die Schwingungen auf, bis die Spurkränze der Räder an die Schienen anschlagen. Wegen des auftretenden Verschleißes und der erhöhten Unfallgefahr ist diese Erscheinung unbedingt zu vermeiden.

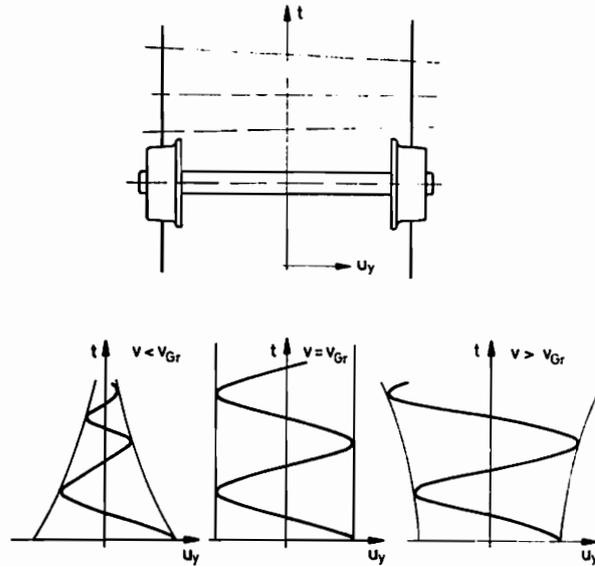


Abb. 5.6: Stabilitätsverhalten

Bei der Konstruktion von Eisenbahnwagen bemüht man sich, durch passende Dimensionierung von Federn und Dämpfern zwischen Radsätzen und Wagenkasten die kritische Geschwindigkeit über die Reisegeschwindigkeit zu legen [37 - 40]. Nach Sarma [41] ist jedoch für zukünftige Hochgeschwindigkeitszüge zu erwarten, daß durch passive Elemente allein keine Stabilisierung mehr erreicht werden kann, so daß eine aktive Dämpfung, also eine Schwingungsregelung, von Interesse wird.

Betrachtet wird das von Sarma [41, 42] angegebene mechanische Modell eines Drehgestells mit 2 Radsätzen. Der als starr angenommene Rahmen ist über Federn und viskose Dämpfer an den Wagenkasten gekoppelt. Verschiebungen der 2 Achsen gegenüber dem Drehgestell sind vernachlässigt. Bezüglich der zwischen Schiene und rollendem Rad wirkenden Kräfte wird auf die Literatur [40, 42] verwiesen. Das in Abb. 5.7 dargestellte Modell wurde von Sarma zum Entwurf von Reglern mit Zustandsbeschränkungen verwendet.

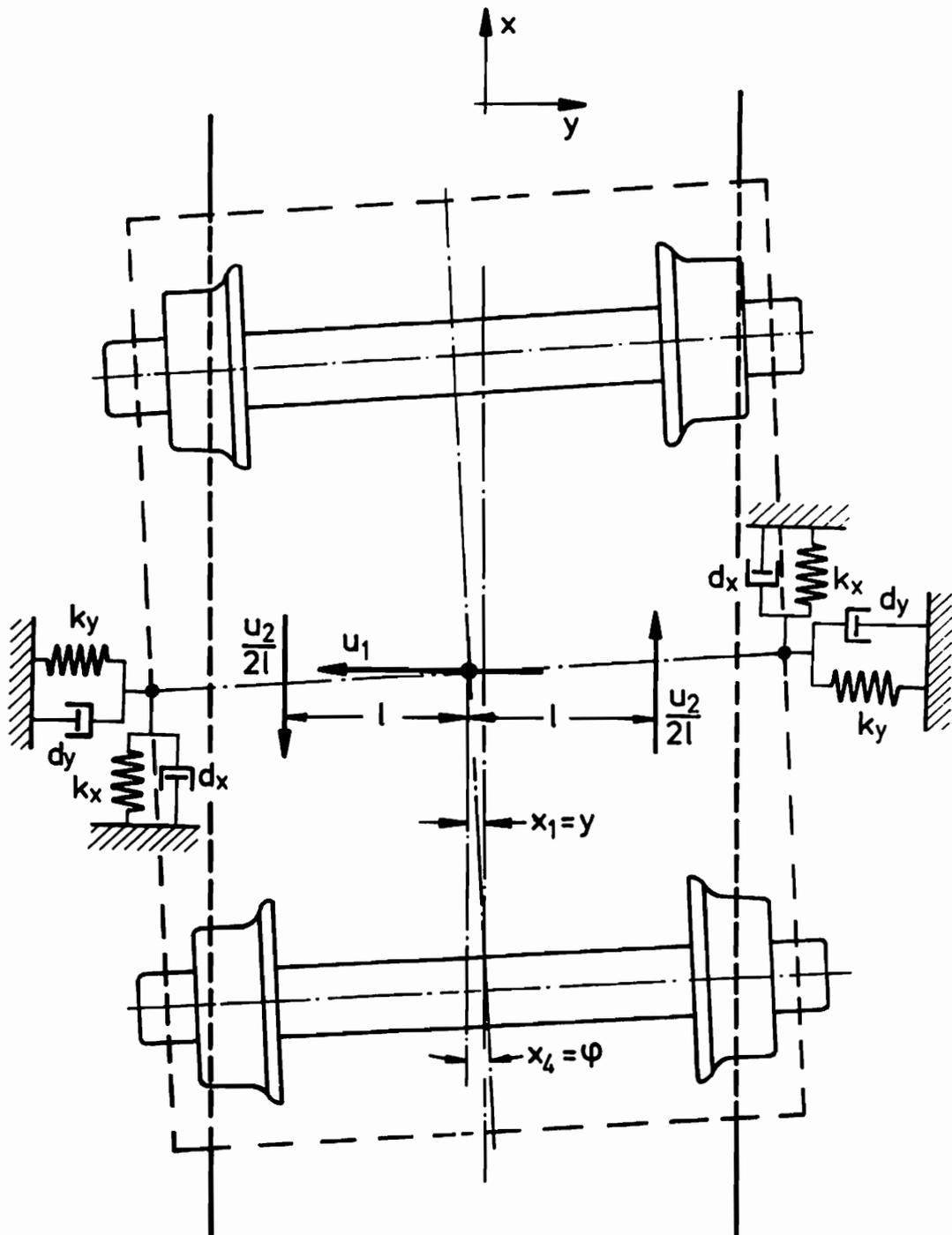


Abb. 5.7: Kopplung Drehgestell-Wagenkasten und Kontrolleingänge

Mit den Zustandsgrößen

x_1 = Querverschiebung des Drehgestellschwerpunktes

$x_2 = \dot{x}_1$

$x_3 = \dot{x}_4$

x_4 = Verdrehung des Drehgestells

und den Kontrolleingängen

u_1 : zusätzliche Querkraft zwischen Drehgestell und Wegen

u_2 : zusätzliches Drehmoment zwischen Drehgestell und Wagen

ergeben sich bei Fahrt auf gerader Bahn mit einer konstanten Geschwindigkeit des Drehgestells von ca. 450 km/h die normierten Zustandsdifferentialgleichungen des instabilen Systems

$$\begin{bmatrix} \dot{x}_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -89 & -91 & 0 & 6840 \\ -770 & 0 & -120 & -120 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (5.10)$$

Diese Gleichung wird als Ausgangspunkt für den suboptimalen Reglerentwurf benutzt. Die einzelnen Schritte sollen entsprechend dem im letzten Abschnitt angegebenen Vorgehen dargestellt werden:

1. Untersuchung der Steuerbarkeit

Das System ist steuerbar, da die Matrix

$$(\underline{B}, \underline{A} \underline{B}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & -91 & 0 \\ 0 & 1 & 0 & -120 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.11)$$

den Rang 4 hat. Das System ist außerdem normal, da es sich, wie man leicht nachrechnet, durch u_1 oder u_2 allein steuern läßt.

2. Festlegung der Gewichtsmatrizen \underline{Q} und \underline{R}

Es wurden die in [41] benutzten Matrizen

$$\underline{Q} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 0,01 & 0 & 0 \\ 0 & 0 & 0,01 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ und } \underline{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.12)$$

übernommen. Mit dieser Wahl ergibt sich die optimale Zustandsvektorrückführung nach Gl. (5.6) zu

$$\underline{u}(t) = \begin{bmatrix} -150 & -1,6 & -0,52 & -150 \\ 33 & -0,52 & -10 & -1300 \end{bmatrix} \underline{x}(t). \quad (5.13)$$

Die durch diese Rückführung benötigten Kräfte können nach [41] hydraulisch aufgebracht werden, zur Betätigung sind Leistungen von einigen kW erforderlich.

3. Festlegung einer Kontrollstruktur

Es wurden die Kontrollstrukturen

- a. $u_1 = \alpha_1 x_1$
 - b. $u_2 = \alpha_2 x_4$
 - c. $u_1 = \beta_1 x_1$, $u_2 = \beta_2 x_4$
 - d. $u_1 = \gamma_1 x_1 + \gamma_3 x_4$
 $u_2 = \gamma_4 x_1 + \gamma_2 x_4$
- (5.14)

untersucht.

4. Hybride Optimierung der Kontrollparameter

Zur hybriden Optimierung wurden das System, das Gütekriterium und die verschiedenen Kontrollstrukturen auf dem Analogrechner simuliert. Die entsprechenden Gleichungen können unmittelbar in einen Analogschaltplan umgesetzt werden, daher ist eine Abbildung des Schaltplanes hier unnötig. Die Parameter der Kontrollstrukturen wurden wieder mit der Methode von Hooke-Jeeves optimiert. Bei einer oberen Grenze des Integrationsintervalls von 2 s und einer Zeitraffung um den Faktor 10 erforderten die jeweils benötigten 50 bis 150 Iterationen eine Rechenzeit von 10 bis 30 s. Es erwies sich als zweckmäßig, den Ablauf der Optimierungen auf dem Oszillographen zu beobachten und die Rechnung durch Anhalten des Analogrechners zu beenden (s. Abschnitt 4.2.5).

Bei der Optimierung wurden für alle Kontrollstrukturen gleiche Anfangswerte für die Auslenkungen x_1 und x_4 gewählt und die Anfangsgeschwindigkeiten $\dot{x}_1 = x_2$, $\dot{x}_4 = x_3$ zu Null gesetzt. In der folgenden Übersicht sind die suboptimalen Lösungen für die verschiedenen Kontrollstrukturen enthalten, außerdem sind die jeweiligen Werte der Gütefunktion, bezogen auf den zu 1 gesetzten Gütefunktionswert der optimalen Lösung (5.13), angegeben.

Es ist jeweils

$$\underline{u}(t) = \underline{K} \underline{x}(t) \quad (5.15)$$

a. Kontrollstruktur $u_1 = \alpha_1 x_1$

$$\underline{K} = \begin{bmatrix} -1280 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

J = 2,0 bis 15 je nach Anfangswerten

b. Kontrollstruktur $u_2 = \alpha_2 x_4$

$$\underline{K} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2200 \end{bmatrix}$$

$J = 1,21 \text{ bis } 2,1$

c. Kontrollstruktur $u_1 = \beta_1 x_1$, $u_2 = \beta_2 x_4$

$$\underline{K} = \begin{bmatrix} -160 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1700 \end{bmatrix}$$

$J = 1,1 \text{ bis } 1,32$

d. Kontrollstruktur $u_1 = \gamma_1 x_1 + \gamma_3 x_4$, $u_2 = \gamma_4 x_1 + \gamma_2 x_4$

$$\underline{K} = \begin{bmatrix} -170 & 0 & 0 & -55 \\ 112 & 0 & 0 & -1700 \end{bmatrix}$$

$J = 1,004 \text{ bis } 1,1$

5. Beurteilung der Ergebnisse

Alle suboptimalen Lösungen sind in der Lage, das System zu stabilisieren. Abb. 5.8 ermöglicht einen Vergleich der optimalen Lösung nach Gl. (5.13) mit einigen der suboptimalen Lösungen. Die Kontrollstrukturen a und b erfordern zur Stabilisierung allerdings erheblich größere Verstärkungsfaktoren als die optimale Lösung (5.13). Für die suboptimalen Kontrollstrukturen c und d ist die Abhängigkeit von den Anfangswerten so gering, daß die wiederholte Optimierung mit linear unabhängigen Anfangsvektoren entfallen kann. Der Verlauf der Gütefunktion (Abb. 5.9) zeigt das der Anschauung entsprechende Ergebnis, daß der Wert der Gütefunktion am Ende des be-

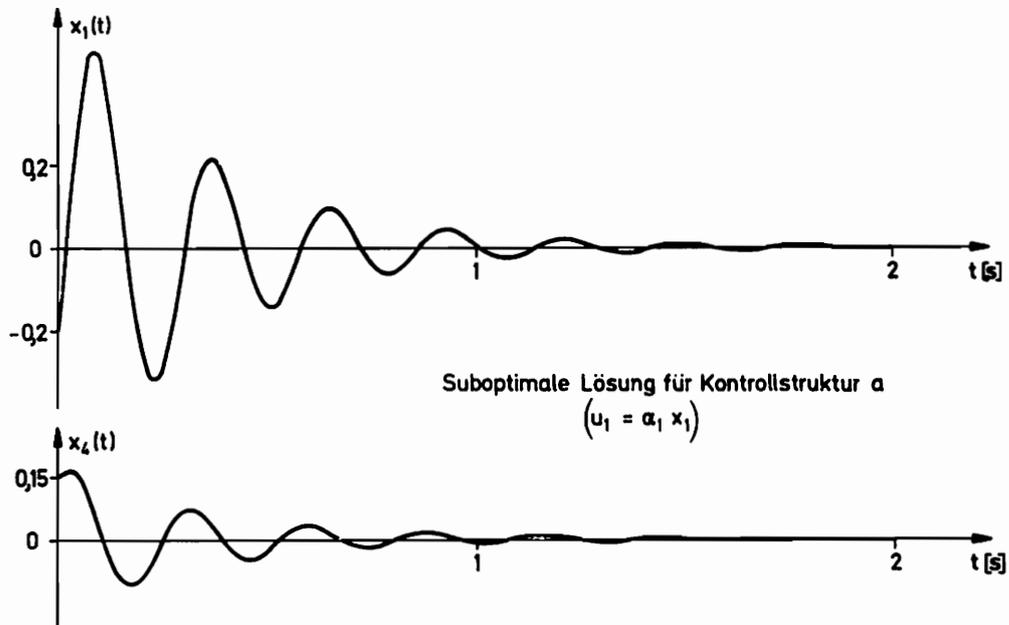
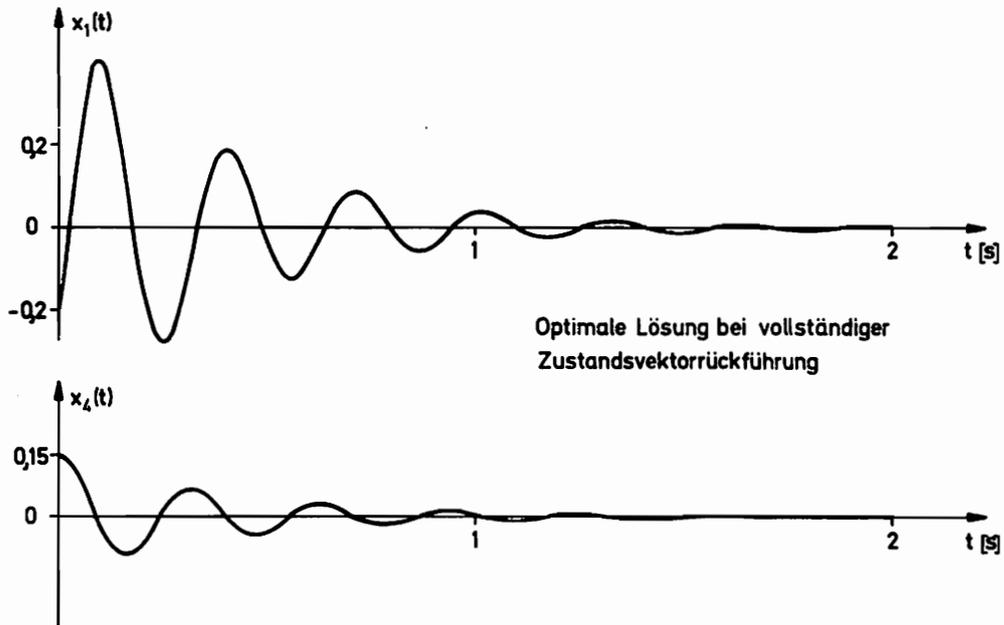


Abb. 5.8: Optimale und suboptimale Regelungen

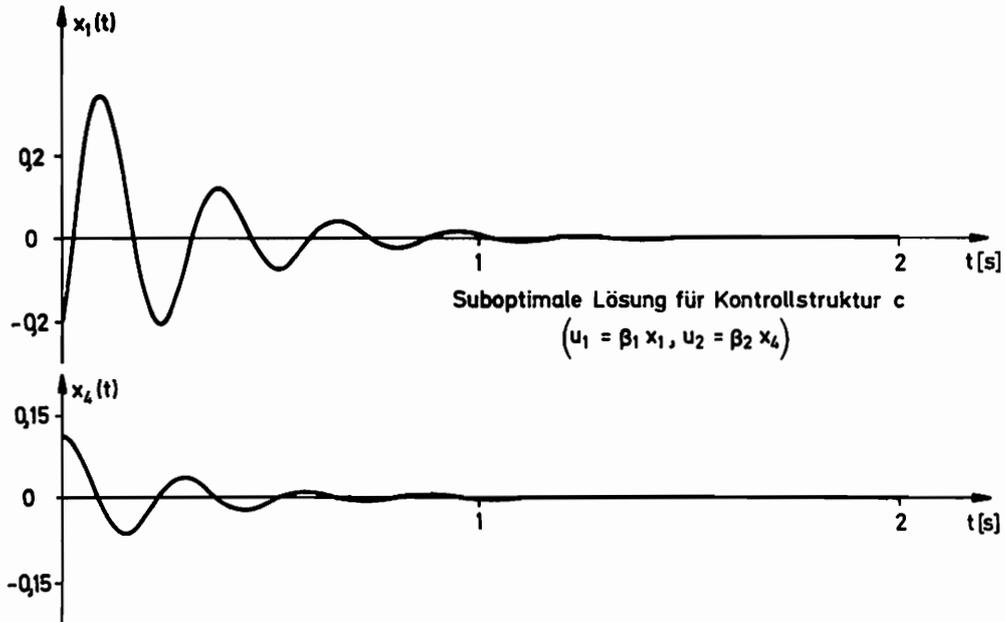


Abb. 5.8: Optimale und suboptimale Regelungen
(Fortsetzung)

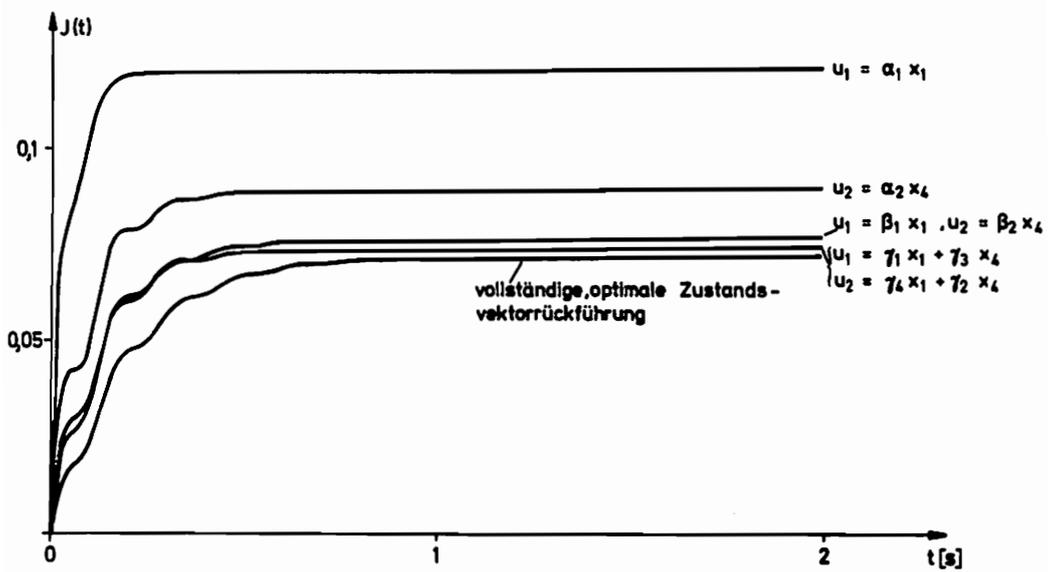


Abb. 5.9: Verlauf der Gütefunktion

trachteten Zeitintervalls einen umso geringeren Wert hat, je mehr die Reglerstruktur der Struktur der optimalen Zustandsvektorrückkoppelung angenähert wird. Alle untersuchten suboptimalen Regler haben den Vorteil, zu ihrer Realisierung keine Geschwindigkeitsmessungen zu benötigen.

Zusammenfassend läßt sich sagen, daß durch hybride Optimierung in der Lösung c ein suboptimaler Zustandsregler gefunden wurde, der im Vergleich zum optimalen eine wesentlich einfachere Struktur besitzt bei vergleichbar großen, realisierbaren Verstärkungsfaktoren und vergleichbarer Stabilität des geschlossenen Kreises.

5.3 Anwendungen zur Funktionaloptimierung

5.3.1 Optimale Steuerung eines nichtlinearen dynamischen Systems

Das hier untersuchte Beispiel ist von relativ geringem praktischen Interesse. Seine Untersuchung hat vor allem den Zweck, anhand eines konkreten Falles einen Test bezüglich folgender Fragen durchzuführen:

1. Ist das entwickelte Rechensystem auch bei sehr kurzen Rechenzeiten des Analogrechners arbeitsfähig?
2. Findet die Rechenberg'sche Evolutionsstrategie auch bei Problemen mit vielen Parametern nach einer vertretbaren Zahl von Iterationen das Optimum?
3. Sind Zufallsmethoden zur hybriden Funktionaloptimierung geeignet (Vorschlag von Korn und Kosako, s. Abschnitt 3.4)?

Folgendes, von Volz [22] angegebene und auf der vander-Pool'schen Differentialgleichung basierende Funktionaloptimierungsproblem wurde behandelt:

Bestimme $u(t)$ für $0 \leq t \leq T$, so, daß für das System

$$\begin{aligned} \dot{x}_1 &= (1 - x_1^2 - x_2^2) x_1 - x_2 + u(t) , & x_1(0) &= 1 \\ \dot{x}_2 &= x_1 & , & x_2(0) = 0 \end{aligned} \quad (5.16)$$

die Gütefunktion

$$J(u(t)) = \int_0^{T_e} ((1-K)u^2 + K(x_1^2 + x_2^2)) dt, \quad (5.17)$$
$$0 < K < 1,$$

minimal wird.

Für T_e wurde eine Zeit von 6 s gewählt, da das System (5.16) bei verschwindendem $u(t)$ einen stabilen Grenzzyklus der Amplitude 1 mit einer Periode von etwa 6 s hat (s. Abb. 5.11). Um zu möglichst geringen analogen Rechenzeiten zu kommen, wurde nicht die in Abschnitt 4.2.4.4 beschriebene Methode zur Ausgabe diskretisierter Zeitfunktionen $u(t)$ benutzt, diese Methode ist auf analoge Rechenphasen länger als 20 ms beschränkt, sondern mit programmierten Zeitverzögerungen des Digitalrechners gearbeitet. Bei programmierten Zeitverzögerungen werden Maschinenbefehle zu keinem anderen Zweck ausgeführt, als eine bestimmte Zeitspanne verstreichen zu lassen. Mit Kenntnis der Ausführungsdauer einzelner Befehle können mit wenigen Befehlen Zeitverzögerungen von einigen μ s bis zu einigen ms erzeugt werden (die Befehlsausführungszeiten sind im Anhang enthalten). Beispielsweise kann man mit den zwei Befehlen

SHL 0,M1

SHL 0,M2

die in Tab. 5.2 angegebenen Zeitverzögerungen erzielen [27]. Auf diese Weise war es möglich, mit einer Zeitraffung um den Faktor 1000 auf eine analoge Rechenzeit von nur 6 ms je Iteration zu kommen. Diesem Vorteil steht als Nachteil gegenüber, daß für jede Änderung der Rechenzeit oder Verfeinerung der Diskretisierung die Programmierung der Zeitverzögerung neu erfolgen muß. Zur Optimierung wurde die Evolutionsstrategie in der Version für Fractionalzahlen mit dem Rahmenprogramm zur Funktionaloptimierung benutzt.

<u>Gewünschte Zeitverzögerung in μs ($\pm 0,5\%$)</u>	<u>M1</u>	<u>M2</u>
100	20	1
150	30	3
200	39	6
250	37	20
500	116	0
1000	118	116

Tab. 5.2: Programmierte Zeitverzögerungen

Abb. 5.10 gibt den optimalen Verlauf von $u(i \cdot \Delta t)$ und die zugehörigen Funktionen $x_1(t)$, $x_2(t)$ für verschiedene Diskretisierungsstufen wieder. Die für eine Diskretisierung optimalen Werte $u(i \cdot \Delta t)$, $i = 1 \dots n$ wurden - auf zwei Stützstellen der nächstfeineren Diskretisierung verteilt - als Startwerte für die nächste Optimierung verwendet. So konnte auf jeder Diskretisierungsstufe bis $n = 32$ die Zahl der notwendigen Iterationen relativ niedrig - bis ca. 1000 - gehalten werden.

Aus den Abbildungen wird deutlich, daß $u(i \cdot \Delta t)$ eine immer weiter verfeinerte Approximation einer stetigen Funktion $u(t)$ darstellt. Die für $n = 32$ am Ende des Zeitintervalls auftretende, relativ große "Streuung" bei $u(i \cdot \Delta t)$ ist darauf zurückzuführen, daß Variationen von u in diesem Bereich keinen großen Einfluß auf die Gütefunktion mehr haben. Eine Erhöhung von n auf 64 brachte keine weitere Verbesserung der Gütefunktion, hier war vermutlich die Genauigkeitsgrenze des Analogrechners und des A/D-Wandlers erreicht.

Zur Beantwortung der oben aufgeführten drei Fragen ist zu sagen, daß sie alle, wenn auch mit unterschiedlichem Nachdruck, positiv beantwortet werden können. Der Funktionsnachweis der hybriden Rechenanlage für sehr hohe analoge Rechengeschwindigkeiten kann als voll erbracht gelten. Die Evolutionsstrategie hat, angewandt nach dem Vorschlag von Korn und Kosako zur hybriden Funktionaloptimierung, gutes Konvergenzverhalten gezeigt.

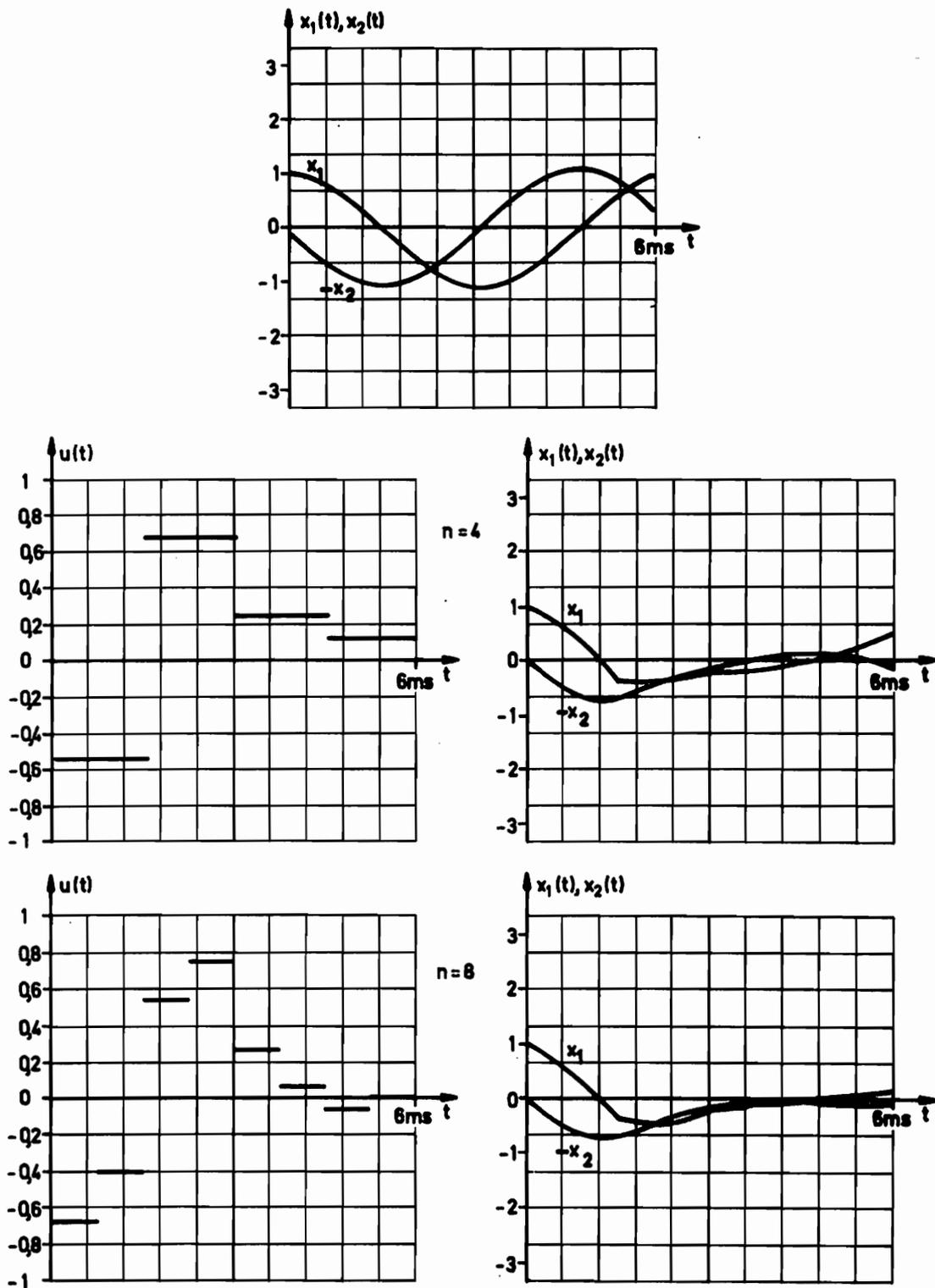


Abb. 5.10: Optimale Funktionsverläufe

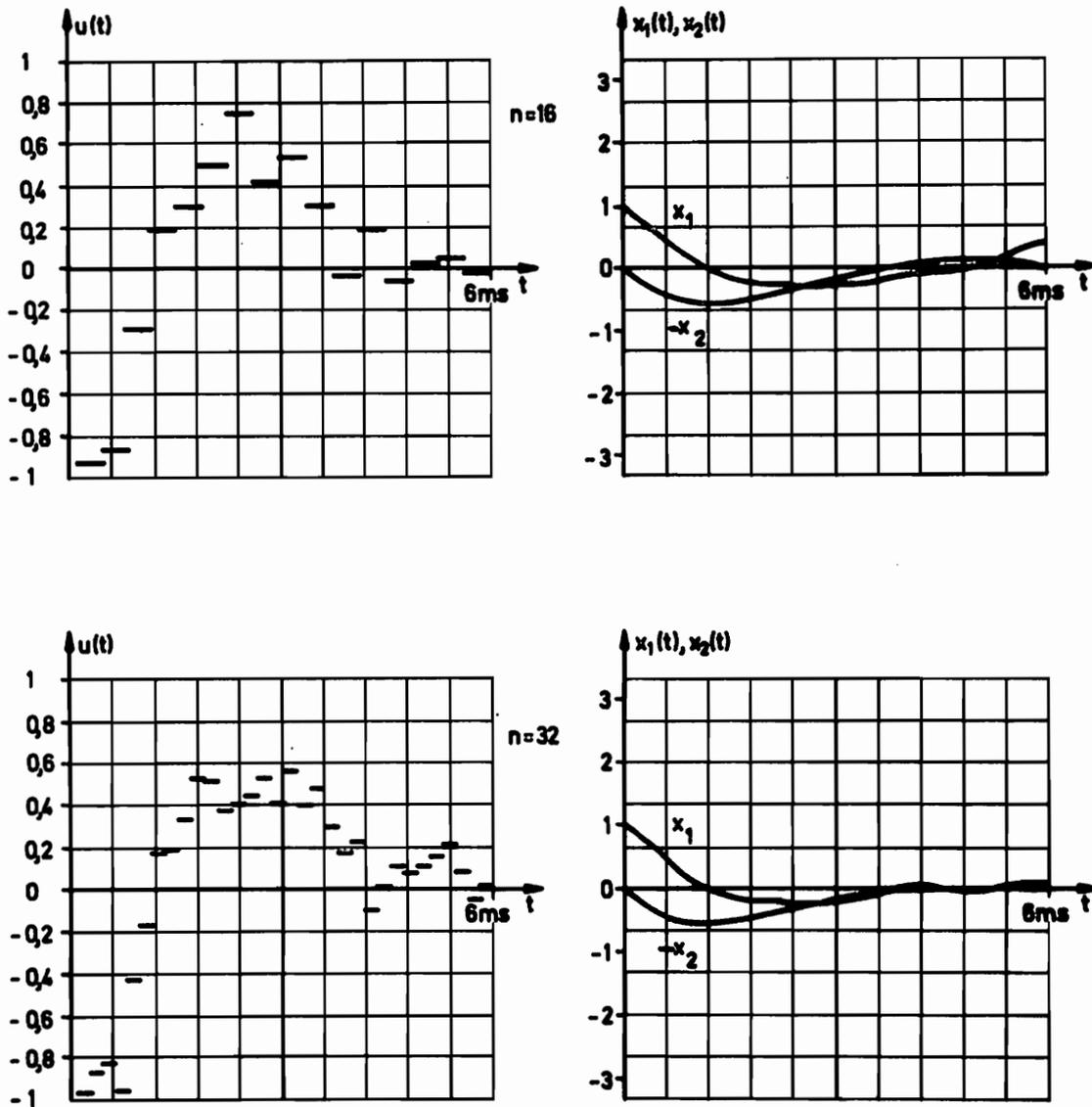


Abb. 5.10: Optimale Funktionsverläufe (Fortsetzung)

Aufgrund eines einzelnen Anwendungsfalles ist natürlich kein allgemeines Urteil möglich.

5.3.2 Optimaler Entwurf dynamisch beanspruchter Stäbe

Der optimale Entwurf einzelner Maschinenelemente oder ganzer Konstruktionen, speziell unter dynamischer Belastung, findet in den letzten Jahren zunehmendes Interesse. Am Institut für Mechanik der Ruhr-Universität Bochum sind dazu einige Arbeiten entstanden [43-46].

Aufgaben dieser Art zählen zu den Funktionaloptimierungen, gesucht wird eine das Teil kennzeichnende Funktion, etwa der Querschnittsverlauf eines Balkens oder die Wandstärkenverteilung einer Rotationschale, zur Optimierung eines bestimmten Kriteriums. Oft wird ein minimales Gewicht der Konstruktion angestrebt.

Hier soll die Frage untersucht werden, wie ein Stab so entworfen werden kann, daß bei fest vorgegebenem Volumen die erste Eigenfrequenz möglichst hoch liegt. Die Lastfälle der dynamischen Torsionsbeanspruchung und der dynamischen Beanspruchung in Längsrichtung werden behandelt.

5.3.2.1 Die maßgebenden Differentialgleichungen

Die Problemstellung soll zunächst präzisiert werden: Für den in Abb. 5.11 dargestellten Stab konstanter Dichte aus homogenem, isotropem Material ist der Querschnittsverlauf so zu bestimmen, daß für Torsions- bzw. Längsbeanspruchung die erste Schwingungseigenfrequenz möglichst hoch liegt. Dabei soll der Typ der Querschnittsfläche nicht verändert werden, und der Verlauf des Querschnitts in Abhängigkeit von x soll durch nur eine skalare Funktion eindeutig beschrieben werden können (z.B. Kreisquerschnitt mit variablem Radius r , Rechteckquerschnitt variabler Höhe und konstanter Breite).

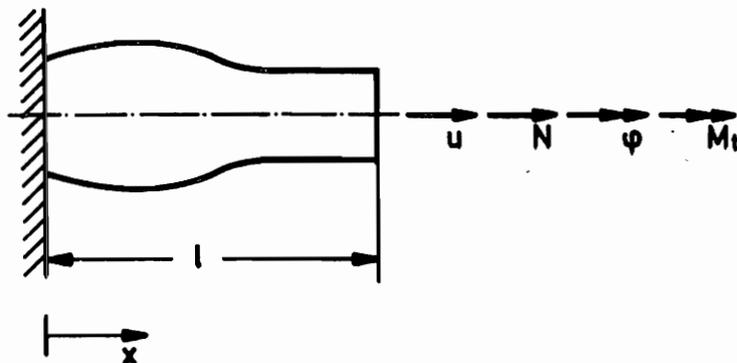


Abb. 5.11: Stab veränderlichen Querschnitts

Ferner sind folgende Nebenbedingungen zu beachten: Das Volumen und damit das Gewicht des Stabes sollen konstant sein. Um eine gewisse Mindestfestigkeit nicht zu unterschreiten, soll die Querschnittsfläche an keiner Stelle einen vorzugebenden Mindestwert A_{\min} unterschreiten.

Die das Problem beschreibenden Differentialgleichungen werden im folgenden parallel für Längs- und Torsionsschwingungen für Stäbe mit Kreisquerschnitt entwickelt, das Vorgehen läßt sich nach [43] jedoch leicht auf andere Querschnittsformen übertragen.

Für freie Schwingungen lauten die Bewegungsdifferentialgleichungen:

bei Torsionsschwingungen	bei Längsschwingungen
Gleichgewichtsbedingung	
$\frac{\partial M_T(x,t)}{\partial x} = g J_0(x) \frac{\partial^2 \varphi(x,t)}{\partial t^2}$	$\frac{\partial N(x,t)}{\partial x} = g A(x) \frac{\partial^2 u(x,t)}{\partial t^2}$ (5.18)

M_T : Torsionsmoment φ : Verdrehung J_0 : polares Flächen- trägheitsmoment g : Dichte	N : Normalkraft u : Verschiebung A : Querschnitts- fläche
---	--

Stoffgesetz

$$M_T(x,t) = G J_0(x) \frac{\partial \varphi(x,t)}{\partial x} \quad N(x,t) = E A(x) \frac{\partial u(x,t)}{\partial x} \quad (5.19)$$

Nach einem Bernoulli'schen Separationsansatz für die Orts- und Zeitlösung erhält man die hier nur interessierende Ortslösung aus

$$\frac{dM_T(x)}{dx} = -g \omega^2 J(x) \varphi(x) \quad \frac{dN(x)}{dx} = -g \omega^2 A(x) u(x)$$

$$\frac{d\varphi(x)}{dx} = \frac{M_T(x)}{G J_0(x)} \quad \frac{du(x)}{dx} = \frac{N(x)}{E A(x)} \quad (5.20)$$

darin ist ω eine Eigenfrequenz des Stabes, sie muß aus den Randbedingungen

$$\varphi(0) = 0, M_T(l) = 0 \quad u(0) = 0, N(l) = 0 \quad (5.21)$$

bestimmt werden.

Die Einführung einer beliebigen Bezugsfläche A ermöglicht die Darstellung

$$\begin{aligned} G J_0(x) &= c_1 \left(\frac{A(x)}{A_0} \right)^r & E A(x) &= c_2 \left(\frac{A(x)}{A_0} \right)^s \\ &= c_1 a^r(x) & &= c_2 a^s(x) \\ r &= 2 & s &= 1 \end{aligned} \quad (5.22)$$

Mit

$$\xi = \frac{x}{l} \quad (5.23)$$

und den Abkürzungen

$$\beta^2 = \frac{g \omega^2 l^2}{G} \quad \beta^2 = \frac{g \omega^2 l^2}{E} \quad (5.24)$$

$$y_1 = \frac{G A_0^2 \varphi}{l} \quad y_1 = \frac{E A_0^2 u}{l} \quad (5.25)$$

$$y_2 = M_T \quad y_2 = N \quad (5.26)$$

gewinnt man für beide Lastfälle die einheitliche Darstellung

$$\left\| \begin{array}{l} \frac{dy_1}{d\xi} = y_2(\xi) \frac{1}{a^i(\xi)} \\ \frac{dy_2}{d\xi} = -\beta^2 a^i(\xi) y_1(\xi) \end{array} \right. \quad \left. \begin{array}{l} y_1(0) = 0 \\ y_2(1) = 0 \end{array} \right\| \quad (5.27)$$

$$i = \begin{cases} 1 & \text{für Längsschwingungen} \\ 2 & \text{für Torsionsschwingungen.} \end{cases}$$

Es sei darauf verwiesen, daß für konstanten Querschnittsverlauf in beiden Fällen die Eigenfrequenzen unabhängig von der Größe der Querschnittsfläche sind, für die erste Eigenfrequenz ist dann jeweils $\beta = \pi/2 = 1,571$.

5.3.2.2 Lösung durch hybride Optimierung

Das so gestellte Problem wurde mit der Evolutionsstrategie in der Version für Fractionalzahlen und dem allgemeinen Rahmenprogramm für Funktionaloptimierungen gelöst. Infolge der hier auftretenden Nebenbedingungen wurde die Evolutionsstrategie durch einige Programmteile ergänzt. Die Nebenbedingung $A > A_{\min}$ ist derart behandelt, daß nicht zulässige Variationen durch neue ersetzt werden, solange, bis an jeder Stelle des Sta-
bes diese Nebenbedingung erfüllt ist. Für diese Variation der bis dahin gefundenen besten Lösung wird das Volumen ermittelt, es weicht in der Regel vom vorgeschriebenen Gesamtvolumen ab. Heuristisch werden dann alle Querschnittswerte so um den gleichen Betrag verändert, daß der aufgetretene Volumenfehler unter Berücksichtigung der ersten Nebenbedingung ausgeglichen wird.

Zu einer nun alle Nebenbedingungen erfüllenden Variation der bisherigen Lösung muß dann die erste Eigenfrequenz ermittelt werden, sie ist ja zu maximieren.

Während bei allen bisher behandelten Anwendungen sich die Gütefunktion nach einem einzigen analogen Rechenzyklus ergab, sind die Verhältnisse hier komplizierter, da zur Bestimmung des Gütefunktionswertes erst ein Randwertproblem gelöst werden muß. Dazu werden die Systemgleichungen (5.27) analog integriert mit festen Anfangswerten $y_1(0) = 0$ und $y_2(0) = \text{const.}$ Ein einfaches Unterprogramm zur linearen Suche verändert dann β^2 von 0 ausgehend solange, bis $|y_2(1)|$ zu Null wird. Abb. 5.12 enthält ein Flußdiagramm für die lineare Suche. Jede Eigenwertbestimmung erfordert etwa 5 bis 10 Rechenzyklen des Analogrechners.

Man kann hier von einer hierarchischen Optimierung sprechen, da bei jeder Querschnittsvariation für die übergeordnete Maximierung der ersten Eigenfrequenz diese Eigenfrequenzen erst durch eine untergeordnete Optimierung bestimmt werden. Es sei erwähnt, daß bei dem hier vorliegenden linearen System der Eigenwert auch aus den Endwerten $y_1(1)$, $y_2(1)$ berechnet werden könnte, wenn man (5.27) zweimal mit verschiedenen Anfangswerten $y_1(0)$, $y_2(0)$ löste. Der Lösungsgang entspräche dann dem Übertragungsverfahren der Mechanik.

Man beachte, daß durch die Minimierung von $|y_2(1)|$ zur Bestimmung von β^2 das Rahmenprogramm nicht geändert zu werden braucht. Die Eigenwertbestimmung wird vollständig von dem in jedem Anwendungsfall vom Nutzer zu schreibenden Unterprogramm zur Berechnung der Gütefunktion übernommen.

Für die analoge Auswertung der Systemgleichungen (5.27) wurde zunächst nur die Funktion $a^i(\xi)$ vom Mikrorechner auf den Analogrechner ausgegeben und die Funktion $1/a^i(\xi)$ durch analoge Division gebildet. Es stellte sich jedoch heraus, daß durch die bereichsweise unvermeidlich schlechte Aussteuerung der Divisionsschaltung bei der Eigenwertbestimmung Fehler bis zu 20% auftraten.

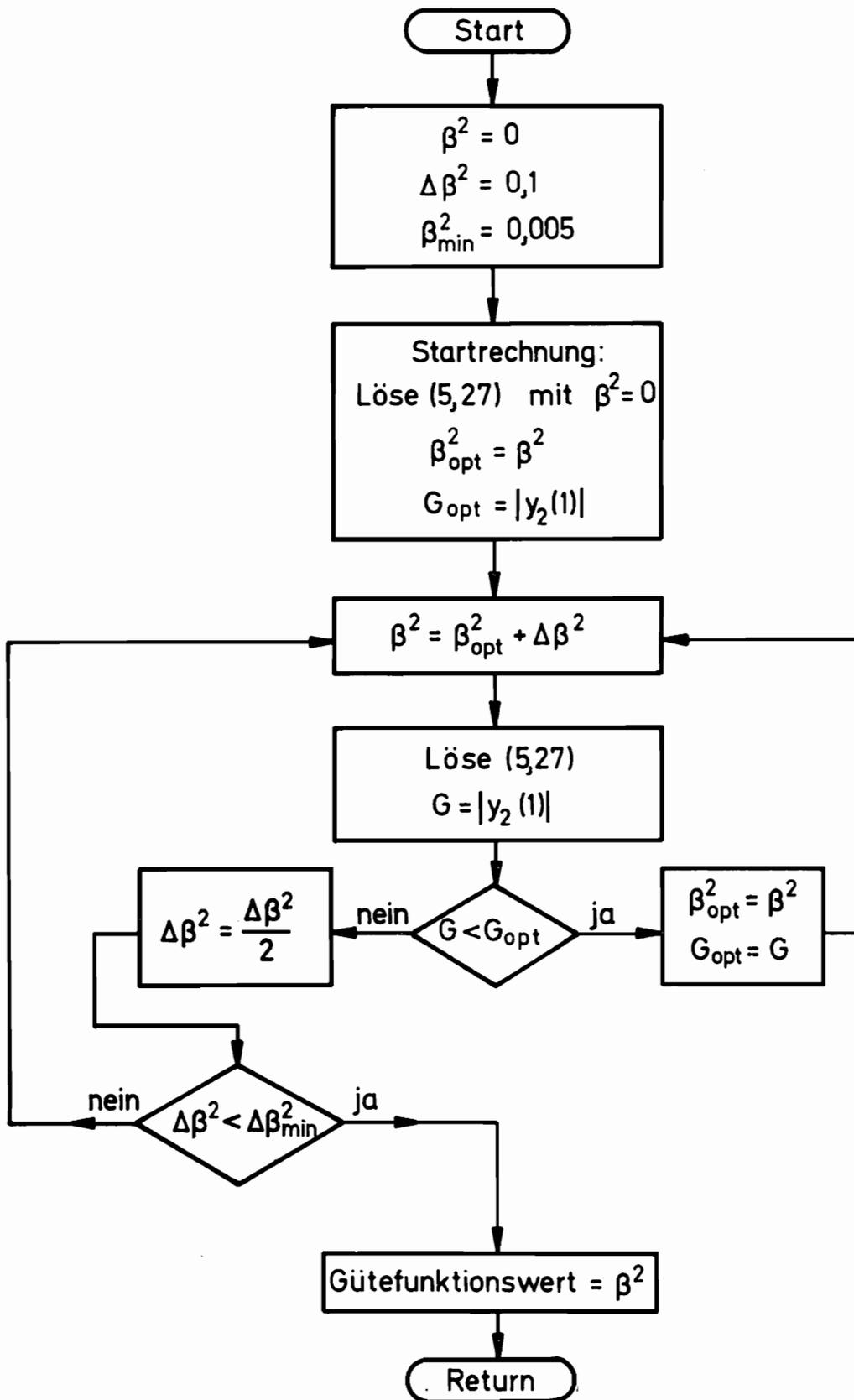


Abb. 5.12: Unterprogramm zur linearen Suche

Daher wurde dann auch die Funktion $1/a^i(\xi)$ digital berechnet und gleichzeitig mit der Funktion $a^i(\xi)$ nach der in Abschnitt 4.2.4.4 beschriebenen Methode ausgegeben. Abb. 5.13 zeigt das analoge Schaltbild.

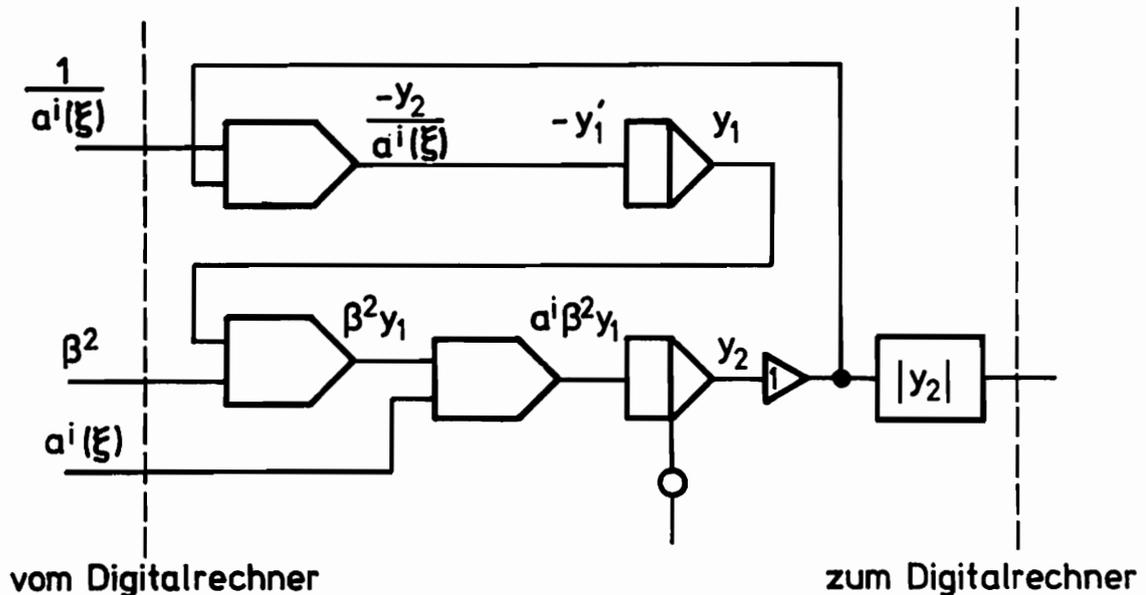


Abb. 5.13: Analogrechenplan zum Stabentwurf

Die mit dieser Schaltung ermittelten Eigenwerte wurden für den glatten und den einmal abgesetzten Stab mit analytisch durch das Übertragungsverfahren ermittelten Werten verglichen. Die Abweichungen in den Ergebnissen betragen für Längsschwingungen maximal 2%, für Torsionsschwingungen - wohl infolge schlechter Aussteuerung der Multiplizierer - maximal 3%. Zur analogen Integration wurde eine Zeitraffung um den Faktor 40 vorgenommen, so daß eine Integration 25 ms dauerte.

5.3.2.3 Ergebnisse

Im folgenden sollen die Ergebnisse für den Fall

$$A_{\min}/A_0 = 0,5 \quad V = |A_0 = \text{const} \quad (5.28)$$

diskutiert werden. Anschaulich gesprochen bedeuten diese

Bedingungen, daß das halbe Stabvolumen gleichmäßig über die Länge verteilt ist und die andere Hälfte beliebig angeordnet werden kann. Bezeichnet man den Radius des Stabes konstanten Querschnitts mit r_0 , so ist wegen (5.28)

$$r_{\min} = 0,707 r_0 , \quad (5.29)$$

allzu schroffe Querschnittsübergänge sind damit ausgeschlossen. Aus der Arbeit von Zander [46] geht hervor, daß in diesem Fall die überhaupt größtmögliche erste Eigenfrequenz bei $\beta_{\max} = 2,17$ für Längsschwingungen und $\beta_{\max} = 2,64$ bei Torsionsschwingungen liegt, bei konstantem Querschnitt ist jeweils $\beta = 1,57$. Zu den angegebenen Maximalwerten gehört ein stetig veränderlicher Querschnittsverlauf. Abb. 5.14 stellt die Ergebnisse für verschiedene Diskretisierungen bei Längsschwingungen dar.

Das - auch in Hinblick auf eine einfache Fertigung - wichtigste Ergebnis ist, daß in beiden Lastfällen schon mit 3 stückweise konstanten Querschnittswerten Eigenfrequenzen erzielt werden können, die im Rahmen der Meßgenauigkeit mit den überhaupt theoretisch größtmöglichen übereinstimmen. Die Optimierung wurde mit Diskretisierungen des Querschnittsverlaufs in 2 bis 20 Werte durchgeführt, in allen Fällen mit dem glatten Stab als Ausgangspunkt. Es wurden jeweils 50 bis 200 Iterationen benötigt, dabei traten ca. 10 bis 20 Erfolge auf. Nimmt man an, daß bei jeder Variation maximal 10 Rechenzyklen für die Eigenwertbestimmung benötigt werden, so kommt man auf eine Rechenzeit von höchstens $200 \cdot 10 \cdot 25 \text{ ms} = 50 \text{ s}$.

Für Torsionsbelastung kommt man zu ähnlichen Querschnittsverläufen (Abb. 5.15). Für beide Lastfälle wurden für $n > 3$ auch Lösungen gefunden, die von den hier dargestellten geringfügig abweichen. Dieses Phänomen ist zu erwarten, da die Eigenwerte ja im Rahmen der

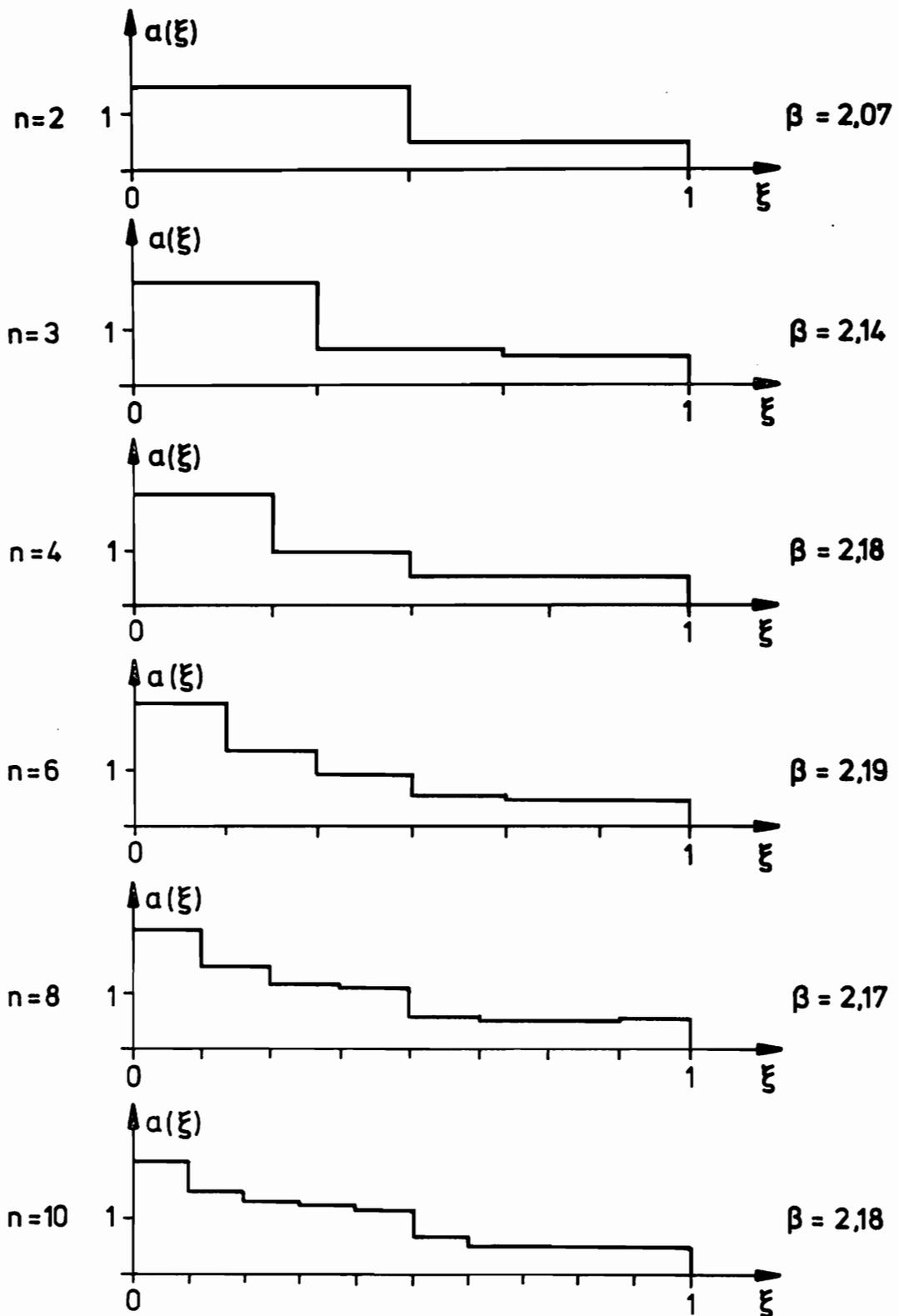


Abb. 5.14: Bei Längsschwingungen optimaler Querschnittsverlauf

Meßgenauigkeit mit den größtmöglichen Werten übereinstimmen und die Gütefunktion hier nicht mehr selektiv wirken kann.

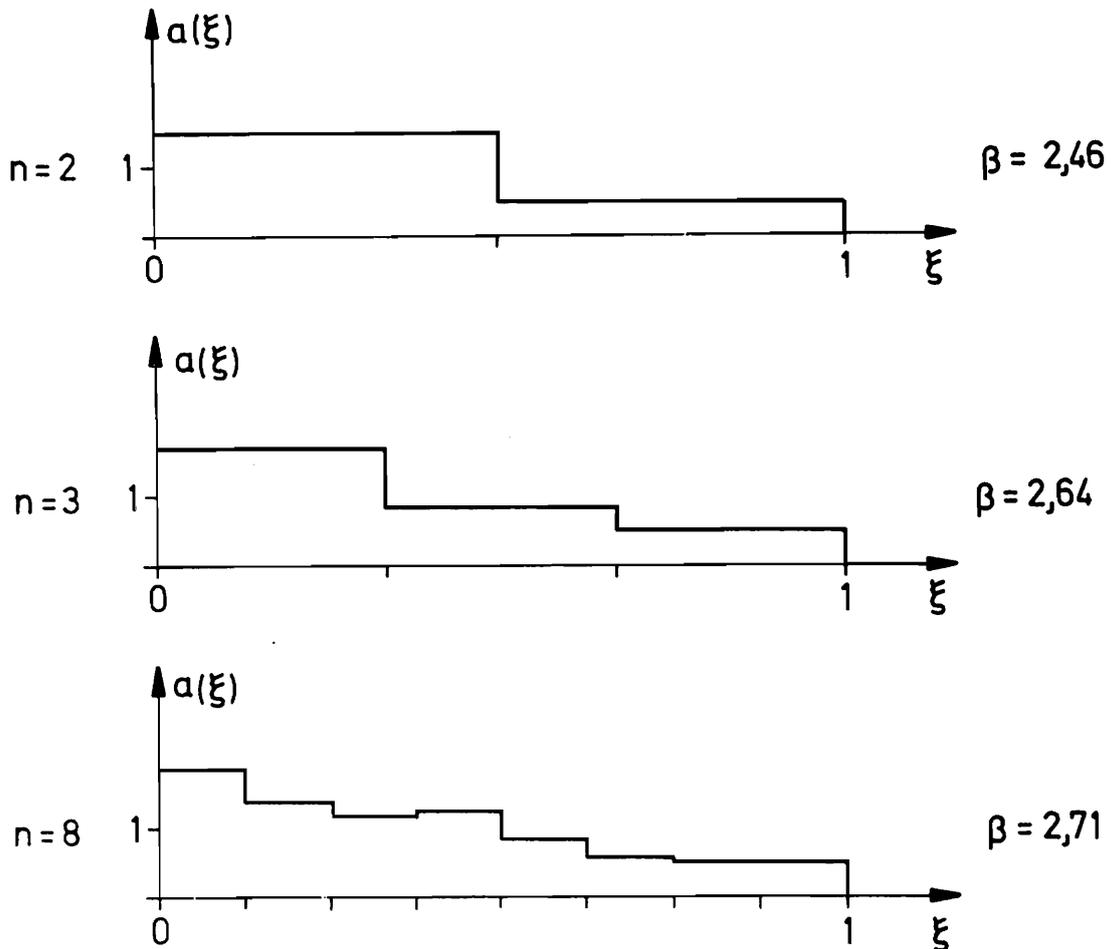


Abb. 5.15: Bei Torsionsschwingungen optimaler Querschnittsverlauf

5.4 Erfahrungen aus der hybriden Optimierung

Abschließend zu diesem Kapitel sollen einige bei der praktischen Arbeit mit dem entwickelten Hybridrechner gesammelte Erfahrungen diskutiert werden.

Ein wichtiges Ergebnis ist, daß Rechenzeiten bei der hybriden Optimierung bei weitem nicht eine so große Bedeutung haben wie bei der rein digitalen Lösung von

Optimierungsaufgaben. Das liegt zum einen an der Fähigkeit des Analogrechners, sehr schnell gewöhnliche Differentialgleichungen lösen zu können, zum andern kann der in unmittelbarem Kontakt zum Rechner stehende Benutzer die erhaltenen Ergebnisse sofort beurteilen und z.B. für die Optimierung andere Startwerte oder Schrittweiten wählen. Dabei spielt es dann praktisch keine Rolle, ob ein Optimierungslauf 10, 50 oder 100 s dauert.

Aus diesem Grund wurden auch bei keinem der vorgestellten Anwendungsfälle Vergleichsrechnungen durchgeführt, um etwa eine für diesen Spezialfall besonders rechenzeitparende Optimierungsmethode zu finden. Allerdings ist die Bedienung des Hybridrechners schwieriger als die eines reinen Digitalrechners, dieser Nachteil hat aber andererseits auch zur Folge, daß der Nutzer beim Aufbau der Analogrechenschaltung und der Kopplung an den Mikrorechner schnell einen guten Einblick in das dynamische Verhalten des simulierten Systems gewinnt.

Es war auch ein Ziel dieser Arbeit, die Anwendbarkeit hybrider Optimierungsmethoden zur Lösung von Problemen aus sehr unterschiedlichen Bereichen zu untersuchen. Die oben diskutierten Optimierungsprobleme demonstrieren die vielfältigen Einsatzmöglichkeiten des entwickelten Hybridrechners. Lediglich in einem Fall, nämlich bei der Lösung eines nichtlinearen Randwertproblems der Strömungsmechanik, wurden keine brauchbaren Resultate erzielt. Die Genauigkeit des Analogrechners reichte hier für die exponentiell auf- und abklingenden Lösungen der Differentialgleichungen nicht aus. Ähnliche Probleme sind auch von digitalen Rechnungen bekannt, etwa bei der Untersuchung von Biegeschwingungen bei Balken mit dem Verfahren der Übertragungsmatrizen. In keinem der betrachteten Fälle behinderten lokale Extrema das Auffinden einer optimalen Lösung.

6. Der Einsatz des Mikrorechners zur Schwingungsanalyse mit der Fourier- und Laplace-Transformation

6.1 Gemeinsamkeiten hybrider Optimierung und digitaler Meßwertverarbeitung

Die in Abschnitt 4.2 beschriebenen Schaltungen und Programme sind in Hinblick auf den Einsatz zur hybriden Optimierung entwickelt worden. Echtzeitbedingungen spielten dabei sowohl bei der Ein-Ausgabe abgetasteter Zeitfunktionen als auch für die Steuerung und Koordination der analogen und digitalen Rechnungen eine wichtige Rolle. Bei der Programmierung wurde darauf Wert gelegt, daß jeder spezielle Anwendungsfall möglichst weitgehend nur durch den Aufruf allgemein verwendbarer Unterprogramme abgedeckt werden kann.

Ähnliche Echtzeitbedingungen wie bei der hybriden Optimierung treten auch bei der digitalen Meßwertfassung und -verarbeitung rechnerüberwachter Versuche auf. Ein allgemeines Unterprogramm zur extern getriggerten Abtastung und Wandlung analoger Signale kann ein Signal vom Analogrechner ebenso verarbeiten wie eines aus einem elektronischen Meßgerät. Gleichzeitig mit den Voraussetzungen zur hybriden Optimierung sind also die Grundlagen für eine digitale Meßwertverarbeitung und Versuchssteuerung geschaffen worden. Der Analogrechner kann dabei zur analogen Datenvorverarbeitung - etwa Filterung - ebenfalls eingesetzt werden. Wie die Möglichkeiten des Mikrorechners und der entwickelten Schaltungen ohne irgendwelche hardwaremäßigen Modifikationen genutzt werden können, soll in diesem Kapitel beispielhaft anhand einer Schwingungsanalyse mit der Fourier- und Laplace-Transformation gezeigt werden.

6.2. Die Fourier-Transformation

Die Kenntnis der dynamischen Eigenschaften hochbeanspruchter Konstruktionen ist von ständig zunehmendem Interesse. Im Mittelpunkt steht dabei die Kenntnis der

Schwingungseigenfrequenzen, aber auch die Ermittlung von Dämpfungswerten und Eigenformen wird immer wichtiger. Auch zur Überprüfung numerischer Rechnungen, die etwa mit der Finite-Element-Methode ausgeführt wurden, sind experimentelle Untersuchungen wichtig. Eigenfrequenzen und -formen sind - im regelungstechnischen Sprachgebrauch ausgedrückt - Informationen im Frequenzbereich, während die bei Schwingungsmessungen auftretenden Signale Informationen im Zeitbereich darstellen.

Die Bedeutung der Fourier- und Laplace-Transformation für die Schwingungsanalyse liegt nun darin, daß mit ihnen Übergänge vom Zeit- in den Frequenzbereich und umgekehrt möglich sind. Da das entwickelte Verfahren zur Schwingungsanalyse auf der diskreten Fourier-Transformation aufbaut, werden zunächst die wichtigsten Eigenschaften der kontinuierlichen und diskreten Transformation dargestellt, soweit sie dem Verständnis der bei der Analyse möglichen Fehler dienen.

6.2.1 Die kontinuierliche Fourier-Transformation

Die Fourier-Transformation einer Zeitfunktion $f(t)$ ist definiert als

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt, \quad (6.1)$$

die inverse Transformation durch

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega. \quad (6.2)$$

In den Transformationsgleichungen hat ω die Bedeutung einer Kreisfrequenz. Beide Integrale existieren im üblichen Sinn nur, wenn die Bedingung

$$\int_{-\infty}^{\infty} |f(t)| dt < \infty \quad (6.3)$$

erfüllt ist. Das ist aber für viele wichtige Funktionen, z.B. $f(t) = \sin \omega t$, nicht der Fall. Diese Schwierigkeit wird behoben, wenn man neben den gewöhnlichen, reellwertigen Funktionen auch die bekannte Dirac'sche δ - Funktion, auch δ -Impuls genannt, in die Betrachtung miteinbezieht.

Die Benutzung von δ -Funktionen ist hier auch noch aus einem anderen Grund zweckmäßig: Die Gleichung

$$\int_{-\infty}^{\infty} \delta(t-t_0) x(t) dt = x(t_0) \quad (6.4)$$

bei der $x(t)$ eine beliebige, im Punkt t_0 stetige Funktion ist, stellt eine mathematische Beschreibung der Abtastung einer Funktion $x(t)$ zum Zeitpunkt t_0 dar. Abb. 6.1 enthält einige wichtige, später benötigte Fourier-Transformationen. Die Transformierte eines Rechteckimpulses bezeichnet man als Spaltsinus. Für gerade Funktionen $f(t)$ ist die Fourier-Transformierte rein reell, für ungerade rein imaginär. Ist $f(t)$ - wie hier immer - eine rein reelle Funktion, so ist der Realteil von $F(\omega)$ gerade und der Imaginärteil ungerade. $F(\omega)$ oder auch $|F(\omega)|$ wird vielfach Spektrum genannt.

Als Faltung zweier Funktionen $f_1(t)$ und $f_2(t)$ bezeichnet man die Operation

$$\int_{-\infty}^{\infty} f_1(\tau) \cdot f_2(t-\tau) d\tau .$$

Das Ergebnis der Faltung ist eine andere Funktion $g(t)$, man schreibt

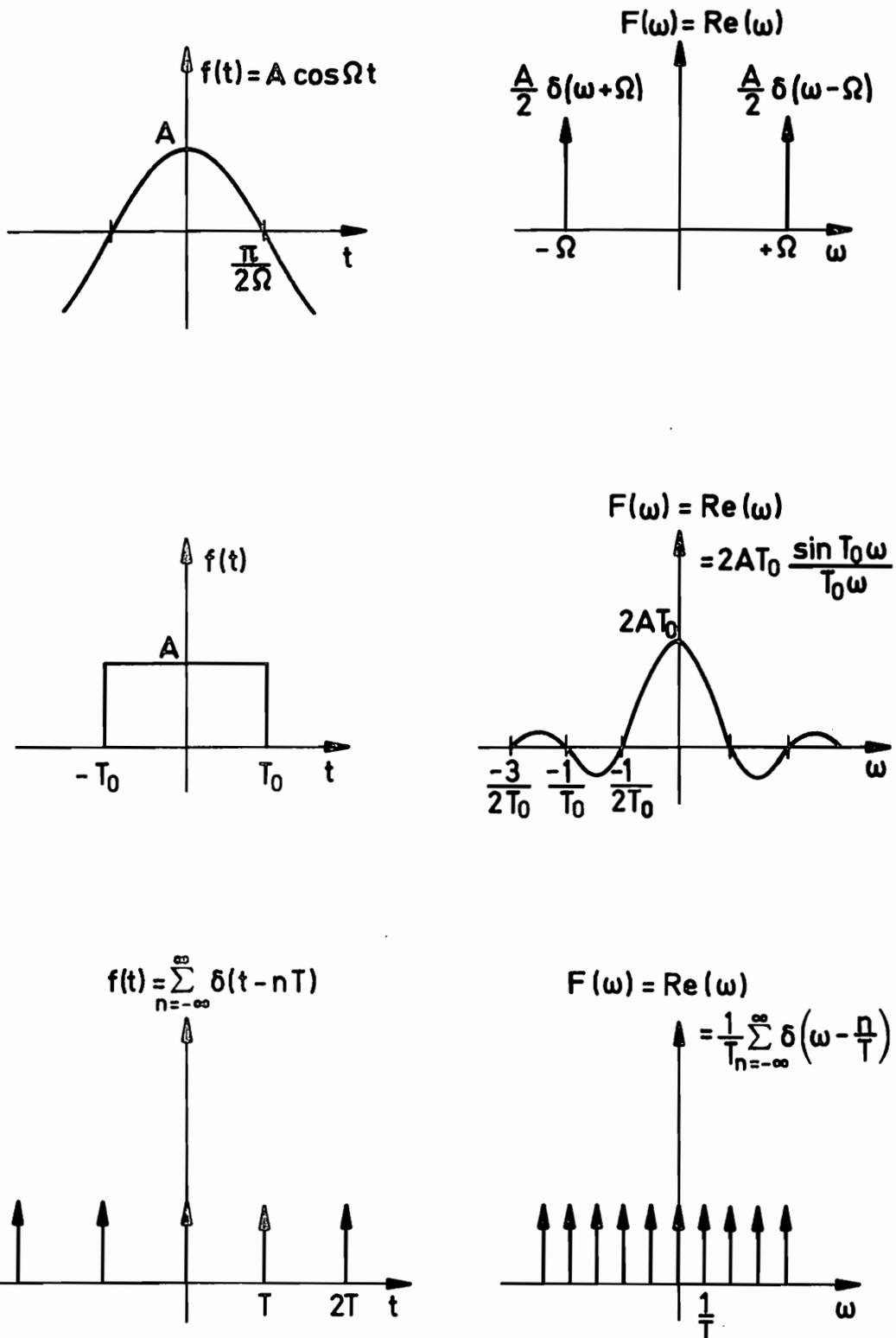


Abb. 6.1: Einige Fourier-Transformationen

$$g(t) = f_1(t) * f_2(t) \quad (6.5)$$

In dem speziellen Fall, daß $f_{1,2}(t)$ ein δ -Impuls oder eine Folge von δ -Impulsen ist, ergeben sich bei einer Faltung besonders einfache Ergebnisse, Abb. 6.2 zeigt ein Beispiel.

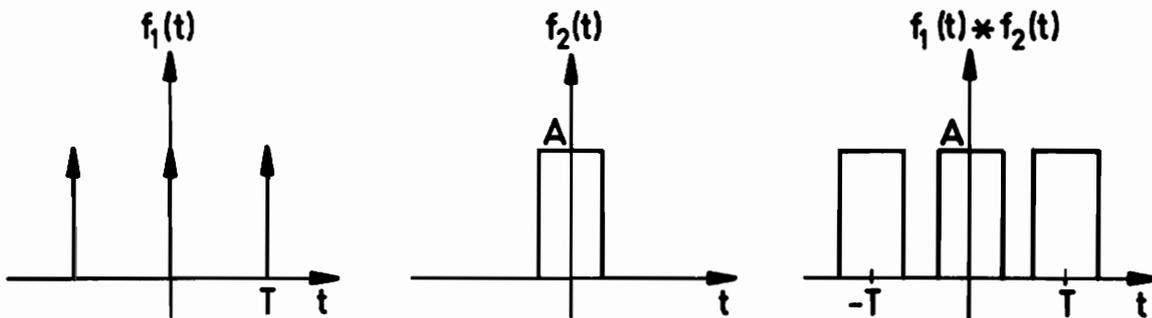


Abb. 6.2: Faltung mit einer Impulsfolge

Bei den späteren Fehlerbetrachtungen sind die Faltungssätze der Fourier-Transformation von Bedeutung, sie lauten:

$$\begin{aligned} f_1(t) * f_2(t) &\hat{=} F_1(\omega) \cdot F_2(\omega) \\ f_1(t) \cdot f_2(t) &\hat{=} F_1(\omega) * F_2(\omega) \end{aligned} \quad (6.6)$$

D.h., einer Multiplikation im Zeitbereich entspricht eine Faltung im Frequenzbereich und umgekehrt.

6.2.2 Die diskrete Fourier-Transformation als Spezialfall der kontinuierlichen

Die numerische Berechnung von Fourier-Transformierten mit einem Digitalrechner ist aus zwei Gründen durch direkte Auswertungen der Gleichungen (6.1) bzw. (6.2) nicht möglich: Wegen der prinzipiell endlichen Speichermöglichkeiten und Rechenzeiten ist die Integration von

$-\infty$ bis $+\infty$ nicht durchführbar, ferner können im Digitalrechner nur diskretisierte, keine kontinuierlichen Funktionsverläufe dargestellt werden. Nach Brigham [48] kann man die Definitionsgleichungen (6.1,2) in Hinblick auf digitale Rechnungen folgendermaßen modifizieren:

Es sei $X(\omega)$ die entsprechend (6.1) Transformierte von $x(t)$. Die Diskretisierung von $x(t)$ kann nach (6.4) als eine Modulation von $x(t)$ mit einer Folge von Dirac-Impulsen angesehen werden. Die Transformation dieses Produkts ist nach dem Faltungssatz (6.6) die Faltung von $X(\omega)$ mit der Transformierten der Impulsfolge, also mit einer Impulsfolge im Frequenzbereich (s. Abb. 6.1). Die Faltung erzeugt damit eine im Frequenzbereich periodische Funktion, wie Abb. 6.2 völlig analog für eine Faltung im Zeitbereich verdeutlicht. Erfolgt die Abtastung im Zeitbereich mit der Abtastrate Δt , so im Frequenzbereich das Spektrum periodisch mit $1/\Delta t$.

Die Berücksichtigung nur endlich vieler Werte im Zeitbereich läßt sich beschreiben als Multiplikation der abgetasteten Zeitfunktion mit einer "Fensterfunktion" nach Abb. 6.3:

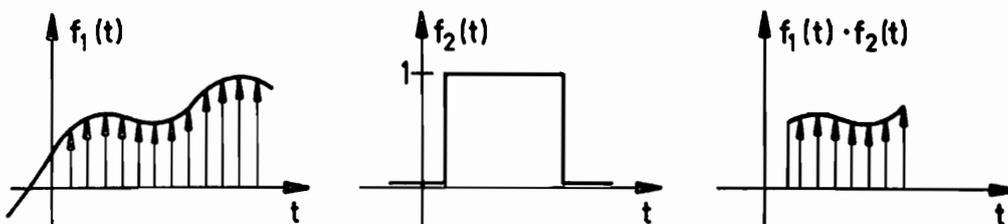


Abb. 6.3: Fensterfunktion im Zeitbereich

Der Multiplikation im Zeitbereich entspricht wieder eine Faltung im Frequenzbereich. Die Transformierte der Fensterfunktion $f_2(t)$ erhält man aus dem Spaltsinus (Abb. 6.1) unter Anwendung des Zeitverschiebungssatzes

$$f(t - t_0) \hat{=} F(\omega) e^{-j\omega t_0} . \quad (6.7)$$

Schließlich muß die Transformierte zur Darstellung im Digitalrechner noch im Frequenzbereich diskretisiert werden. Als Folge daraus ergibt sich bei der Rücktransformation in den Zeitbereich eine periodische Zeitfunktion, die Periode ist gerade gleich der Breite der Fensterfunktion im Zeitbereich (Abb. 6.3). Nach analytischer Ausführung dieser Operationen erhält man die Definitionsgleichungen der diskreten Fourier-Transformation

$$F_n = \frac{T}{2N} \sum_{k=0}^{2N-1} f\left(k \frac{T}{2N}\right) e^{-jnk \frac{\pi}{N}} \quad n = 0, 1, \dots, 2N-1 \quad (6.8)$$

und der inversen Transformation

$$f_k = \frac{1}{T} \sum_{n=0}^{2N-1} F\left(n \frac{2\pi}{T}\right) e^{jnk \frac{\pi}{N}} \quad k = 0, 1, \dots, 2N-1 \quad (6.9)$$

Dabei ist das Zeitintervall T in $2N$ äquidistante Zeitschritte Δt unterteilt, die Schrittweite im Frequenzbereich beträgt $\Delta\omega = 2\pi/T$. Für reelle Zeitfunktionen ergibt sich der Realteil der diskreten Fourier-Transformation als gerade, der Imaginärteil als ungerade Funktion. Da gleichzeitig Real- und Imaginärteil periodisch sind, folgt, daß auch an der Stelle $\omega = N \cdot 2\pi/T$ eine Symmetrie- bzw. Antimetrieachse vorliegt (Abb. 6.4).

6.3 Die systematischen Fehler der diskreten Fourier-Transformation

$F_1(\omega)$ sei die Fourier-Transformierte der Funktion $f(t)$ und $F_2(i \cdot \Delta\omega)$ die diskrete Fourier-Transformierte der abgetasteten und durch eine Fensterfunktion nach Abb. 6.3 "abgeschnittenen" Funktion $f(t)$. Die diskrete Transformation ist aus der kontinuierlichen entwickelt wor-

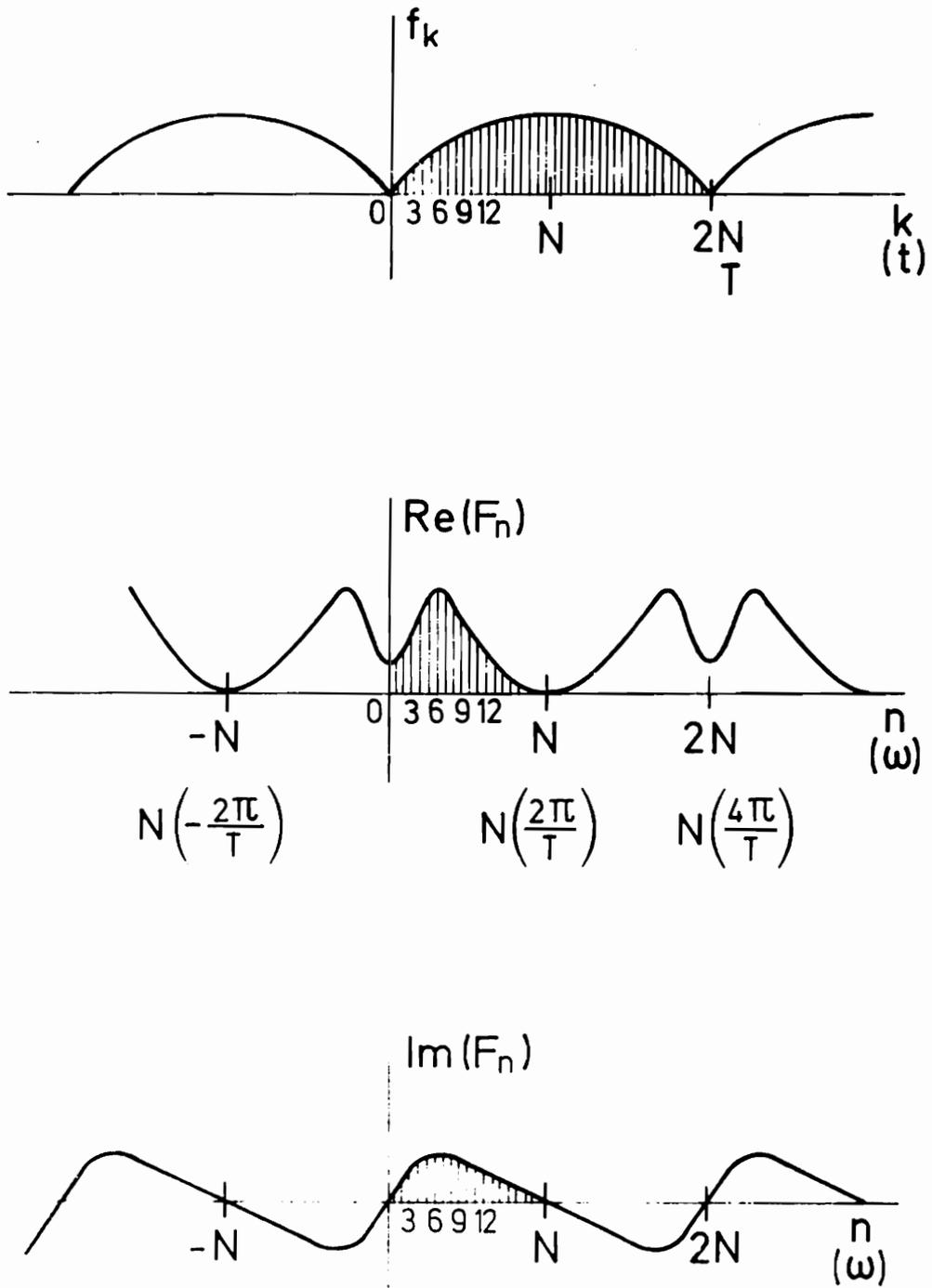


Abb. 6.4: Diskrete Fourier-Transformation nach [49]

den, damit Fourier-Transformationen mit einem Digitalrechner berechnet werden können. Es ist also unbedingt erwünscht, daß beide Transformierten übereinstimmen, also daß gilt

$$F_1(i \Delta \omega) = F_2(i \Delta \omega) . \quad (6.10)$$

Zwischen F_1 und F_2 können jedoch erhebliche Abweichungen auftreten. Dabei handelt es sich nicht etwa um Rundungsfehler, die bei der Auswertung der Gleichungen (6.8,9) auftreten, sondern um verfahrensbedingte, durch Diskretisierung und Abschneiden hervorgerufene systematische Fehler der diskreten Fourier-Transformation.

6.3.1 Der aus der Zeitdiskretisierung stammende Fehler (aliasing)

Dieser, in der englischsprachigen Literatur als aliasing bezeichnete Fehler tritt auf, wenn im betrachteten Zeitsignal hochfrequente Anteile vorhanden sind, die weniger als zweimal pro Periode abgetastet werden. Zur Erläuterung wird nach [50] ein Signal

$$x(t) = \sin(2\pi \tilde{f} t) \quad (6.11)$$

betrachtet, das mit einer Frequenz $f=1/\Delta t$ abgetastet wird. Aus Abb. 6.4 geht hervor, daß die höchste Frequenz, die im Spektrum noch Information liefert, $f=1/(2\cdot\Delta t)$ ist. Es sei $\tilde{f} > f$, dann läßt sich \tilde{f} darstellen als

$$\tilde{f} = a f = \frac{1}{2\Delta t} (n+p) \quad \begin{array}{l} a > 1, \\ a = n+p, \\ n \text{ ganzzahlig} \\ 0 \leq p < 1 \end{array} \quad (6.12)$$

Damit wird das abgetastete Signal zu

$$x(k \cdot \Delta t) = \sin(2\pi \tilde{f} k \Delta t) \quad (6.13)$$

und nach Einsetzen von (6.12) in (6.13)

$$\begin{aligned}
 x(k \cdot \Delta t) &= \sin(\pi k(n+p)) \\
 &= \sin \pi k n \cos \pi k p + \cos \pi k n \sin \pi k p \\
 &= \cos \pi k n \sin \pi k p \\
 &= [(-1)^n]^k \sin 2\pi p .
 \end{aligned} \tag{6.14}$$

Die Einführung von

$$f' = \frac{p}{2\Delta t}, \quad 0 \leq f' < f \tag{6.15}$$

führt zu

$$x(k \Delta t) = [(-1)^n]^k \sin 2\pi k \Delta t f' , \tag{6.16}$$

d.h. für gerade n und $t = k \cdot \Delta t$ ist

$$x(t) = \sin 2\pi t f' , \tag{6.17}$$

für ungerade n und $t = k \cdot \Delta t$ ist

$$\begin{aligned}
 x(t) &= (-1)^k \sin 2\pi k p = \cos \pi k \sin \pi k p \\
 &= \sin(2\pi t(f - f')) .
 \end{aligned} \tag{6.18}$$

Somit werden alle Frequenzen, die größer als f sind, auf Frequenzen zwischen 0 und f abgebildet. Es sei beispielsweise $f = 350$ Hz, $\tilde{f} = 1200$ Hz, damit $n = 3$, $p = 0.4286$, $f' = 150$ Hz. Die Frequenz 1200 Hz wird hier auf 200 Hz abgebildet (Abb. 6.5). Zur Vermeidung von aliasing gibt es mehrere Möglichkeiten. Tastet man das Zeitsignal so schnell ab, daß alle enthaltenen Frequenzen mindestens zweimal pro Periode erfaßt werden, tritt kein aliasing auf. Vielfach ist jedoch gar nicht genau bekannt, welche Frequenzen in dem Signal vorhanden sind. Dann kann man das Zeitsignal mit einem Tiefpass, dessen Eckfrequenz passend gewählt ist, etwa zu $1/(2\Delta t)$,

filtern und so die zu hohen Frequenzkomponenten ausschalten.

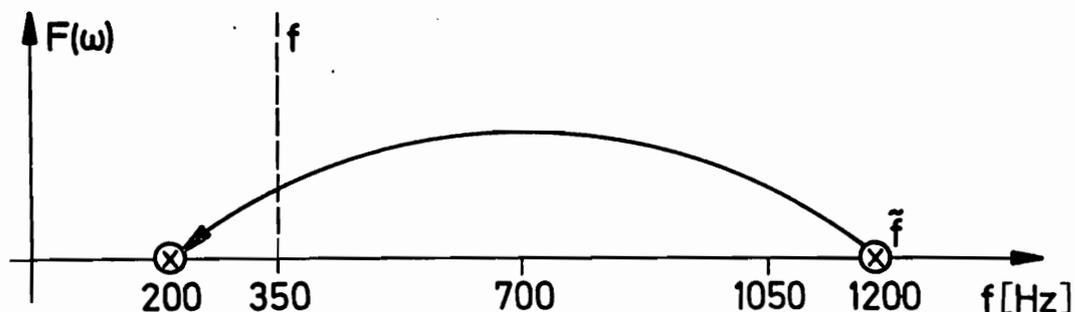


Abb. 6.5: Beispiel für aliasing

Bei Schwingungsmessungen sind häufig nur wenige Frequenzen im Zeitsignal vertreten, so daß das Spektrum aus einzelnen, deutlich voneinander getrennten Spitzen besteht (siehe z.B. Abb. 6.13). Man kann dann erkennen, ob eine Spitze an einer Stelle im Frequenzbereich eine Folge von aliasing ist, indem man die Messung und Transformation mit einer um 10 bis 20% variierten Abtastrate wiederholt. Lag aliasing vor, findet man aufgrund der beschriebenen Zusammenhänge die Spitze an einer völlig anderen Stelle im Frequenzbereich wieder.

6.3.2 Der aus dem Abschneiden im Zeitbereich stammende Fehler (ripple)

Dieser Fehler tritt auf, wenn durch das Abschneiden im Zeitbereich wesentliche Informationen über das Signal verlorengehen. Für die hier hauptsächlich interessierenden gedämpften harmonischen Schwingungen erfolgt unten eine quantitative Präzisierung dieser Aussage. Eine allgemeine mathematische Fehlerabschätzung ist dem Verfasser aus der Literatur nicht bekannt. Die englische Bezeichnung ripple, d.h. Gekräusel, stammt daher, daß durch diesen Fehler die diskrete Transformierte häufig gegenüber der kontinuierlichen aufgeraut

erscheint (siehe z.B. Abb. 6.13, erste Spitze).

Die grundsätzliche Fehlerursache soll anhand von Abb. 6.6 erläutert werden. Der Multiplikation von $f_1(t)$ mit der Fensterfunktion $f_2(t)$ entspricht im Frequenzbereich eine Faltung. Die Transformierte des Rechteckfensters $f_2(t)$ ist ein Spaltsinus. Je länger die Meßzeit T ist, desto mehr nähert der Spaltsinus einen δ -Impuls an.

Die Faltung von $F_1(\omega)$ mit einem δ -Impuls ergibt wieder $F_1(\omega)$. Also wird $F_1(\omega)$ durch die Faltung mit $F_2(\omega)$ umso weniger verfälscht, je größer die Meßzeit T ist. T läßt sich aber bei einer gegebenen Zahl $2N$ von diskreten Punkten im Zeitbereich und einem gewünschten Umfang des Frequenzspektrums nicht beliebig vergrößern ($f_{\max} = N/T$), der Fehler ist dann grundsätzlich nicht zu vermeiden.

Um zu quantitativen Aussagen zu kommen, wurden anhand der eine Messung gedämpfter Schwingungen gut repräsentierenden Funktion

$$x(t) = \begin{cases} e^{-bt} \sin 2\pi ft & \text{für } t > 0 \\ 0 & \text{für } t \leq 0 \end{cases} \quad (6.19)$$

Vergleiche der kontinuierlichen und der diskreten Transformierten durchgeführt. Die Funktion wurde für negative t zu Null gesetzt, um den Beginn eines Meßvorganges zum Zeitpunkt $t = 0$ zu simulieren. Für diese einfache Funktion läßt sich die Definitionsgleichung (6.1) der Fourier-Transformation analytisch auswerten (s. Abschnitt 6.5.2). Zur Berechnung der diskreten Transformierten wurde $x(t)$ an äquidistanten Punkten numerisch berechnet und die so erhaltenen Punkte nach Gl. (6.8) mit dem Programmsystem STO [51] transformiert.

Die Ergebnisse zahlreicher Vergleichsrechnungen lassen sich wie folgt zusammenfassen: Wird durch das Abschneiden im Zeitbereich ein Intervall der Dauer T geschaffen (Breite der Fensterfunktion), so ist für die Größe

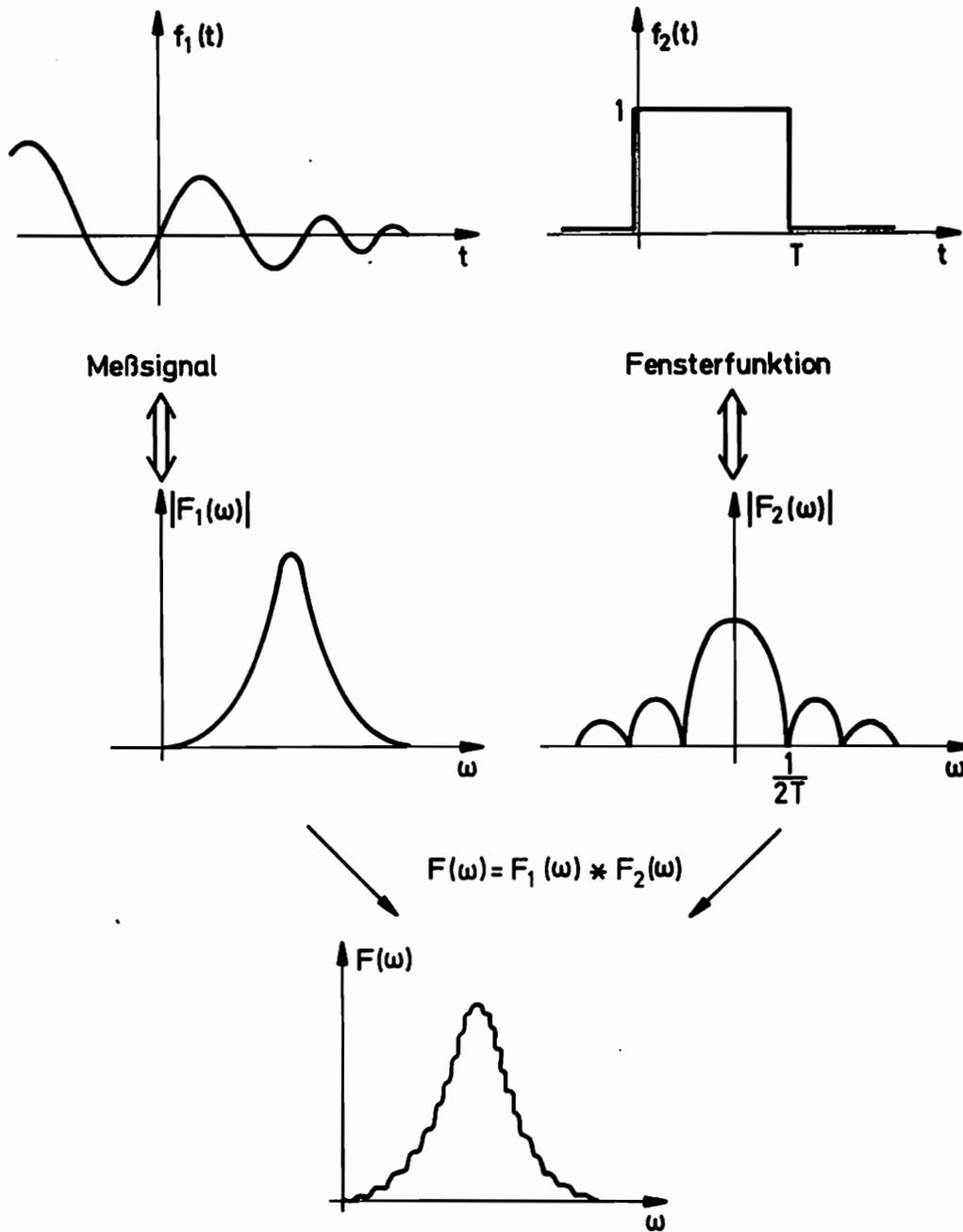


Abb. 6.6: Zum ripple-Fehler

des Fehlers nur der Wert von $\exp(-bT)$ maßgebend (b aus Gl. 6.19). Für

$$e^{-bT} < 0,1 \text{ bis } 0,15 \quad (6.20)$$

kann der Fehler vernachlässigt werden.

Abb. 6.7 zeigt, daß der hier untersuchte Fehler weit größer sein kann, als die Bezeichnung ripple vermuten läßt. Für $f = 315/s$, $T = 0,128 s$, $b = 1,28/s$, also $\exp(-bT) = 0.85$, ist jeweils der Imaginärteil der kontinuierlichen und der diskreten Transformierten dargestellt.

6.4 Der Algorithmus der schnellen Fourier-Transformation (FFT)

Zur numerischen Berechnung der diskreten Fourier-Transformation wurde in den 60er Jahren von Cooley und Tuckey der Algorithmus der "Fast Fourier Transform", kurz FFT, entwickelt. Mit ihm ist es möglich, die diskrete Transformation erheblich schneller auszuführen, als es durch direkte Auswertung der Gleichungen (6.8,9) möglich ist. Das Verfahren nützt bestimmte Eigenschaften der Funktion $\exp(jnk \pi / N)$ aus, Einzelheiten findet man in der Literatur [48]. Der Algorithmus in seiner hier verwendeten Form setzt voraus, daß die Zahl der diskreten Punkte im Zeitbereich eine ganzzahlige Potenz von 2 ist.

Die Berechnung der diskreten Fourier-Transformation geschieht hier mit einem von der Firma Plessey für den Mikrorechner IMP 16 P entwickelten Unterprogramm zur FFT [52]. Da die gemessene Zeitfunktion normalerweise über einen A/D Wandler abgetastet und im Rechner gespeichert wird, ist es sinnvoll, von einer einfach genauen Fractionaldarstellung im Zeitbereich auszugehen. Betrachtet man Gl. (6.8), so erkennt man, daß der Ausdruck

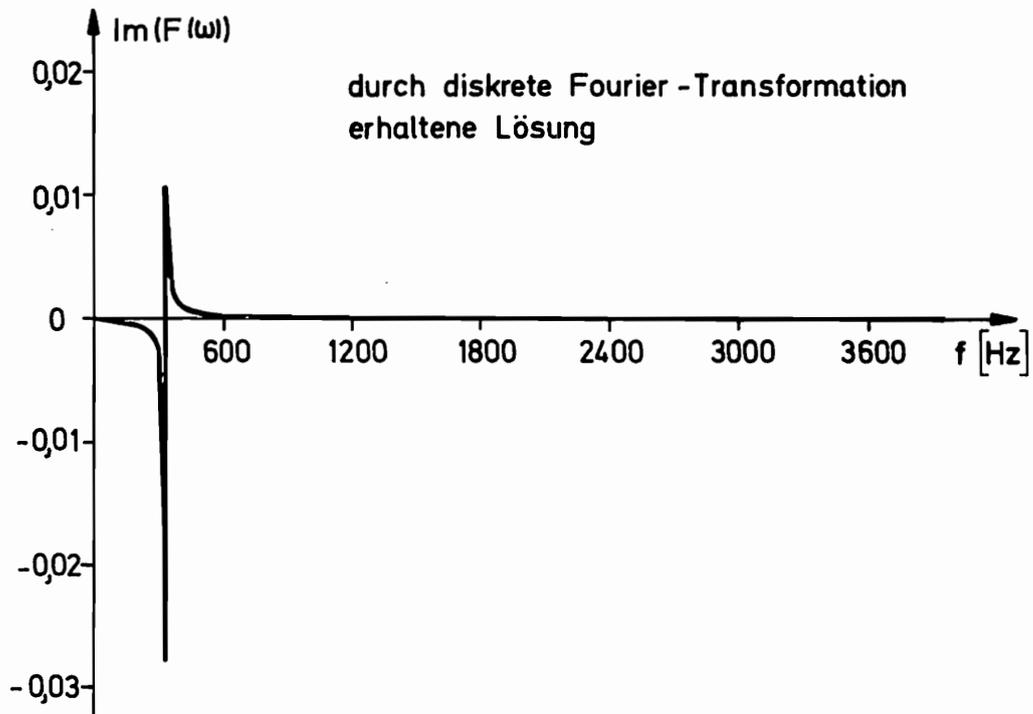
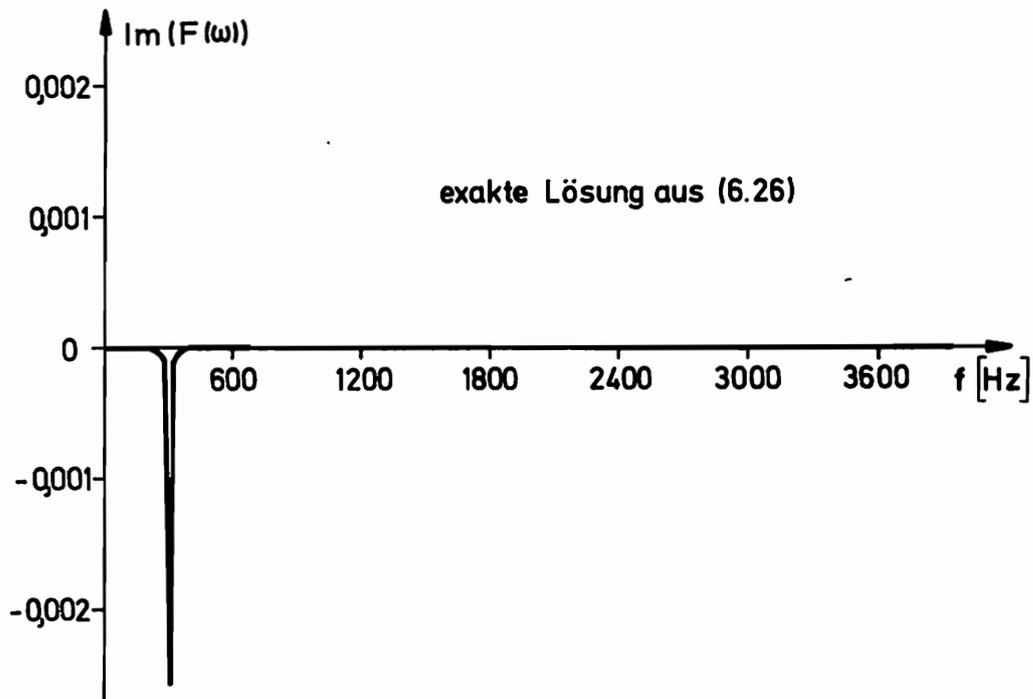


Abb. 6.7: Beispiel für ripple

$$\frac{1}{2N} \sum_{k=0}^{2N-1} f\left(k \frac{T}{2N}\right) e^{-jnk \frac{T}{N}} \quad (6.21)$$

dann in jedem Fall betragsmäßig kleiner als 1 ist. Durch geschickte Programmierung kann man erreichen, daß dies bei der Auswertung von (6.21) auch für alle Zwischenergebnisse gilt. Durch die so mögliche Beschränkung auf Fractionalzahlen lassen sich in erheblichem Umfang Rechenzeiten sparen, entsprechendes gilt auch für die Rücktransformation nach Gl. (6.9). Die dann noch benötigte Multiplikation mit T , bzw. $1/T$ bei der Rücktransformation, ist durch ein vom Nutzer zu schreibendes Programm auszuführen. Bei Verwendung von 1024 Datenpunkten im Zeitbereich ergibt sich eine Rechenzeit für die diskrete Fourier-Transformation von etwa 14 s.

6.5 Die Ermittlung von Schwingungsdaten aus der Fourier-Transformierten von Meßsignalen

6.5.1 Abgrenzung des betrachteten Schwingungstyps

Im folgenden werden durch eine stoßartige Belastung oder einen Ausschwingversuch hervorgerufene Schwingungen mechanischer Systeme betrachtet. Instationäre Schwingungen dieser Art sind im Zuge einer verfeinerten dynamischen Berechnung von zunehmendem Interesse, sie lassen sich auch experimentell relativ leicht realisieren.

Wird der Beginn der Messung auf einen Zeitpunkt nach dem Stoß gelegt, so handelt es sich um eine freie Schwingung und man kann auf eine Impulsmessung verzichten. Faßt man die Schwingung als Überlagerung mehrerer harmonischer Schwingungen in den Eigenfrequenzen der Struktur auf, so kann man unter der Annahme einer viskosen Dämpfung die Auslenkung beschreiben durch

$$\begin{aligned}x(t) &= \sum_{i=1}^k p_i e^{-b_i t} \sin(\Omega_i t + \varphi_i) \\ &= \sum_{i=1}^k e^{-b_i t} (A_i \sin \Omega_i t + B_i \cos \Omega_i t)\end{aligned}\tag{6.22}$$

für $t \geq 0$ und

$$x(t) = 0$$

für $t < 0$.

Es gilt

$$A_i = P_i \cos \varphi_i, \quad B_i = P_i \sin \varphi_i \tag{6.23}$$

Ob und wie weit die Annahmen des harmonischen Schwingungsverlaufs und der viskosen Dämpfung berechtigt sind, muß in jedem einzelnen Anwendungsfall untersucht werden.

6.5.2 Analytische Fourier-Transformation von gedämpften harmonischen Schwingungen

Für die einzelnen additiven Terme von (6.22) läßt sich die Fourier-Transformation (6.1) analytisch berechnen [53]. Die Transformierte der Summe (6.22) ist gleich der Summe der Einzeltransformierten.

Für

$$f(t) = \begin{cases} A e^{-bt} \sin \Omega t & \text{für } t \geq 0 \\ 0 & \text{für } t < 0 \end{cases} \tag{6.24}$$

ergeben sich der Real- und Imaginärteil der Transformierten zu

$$\operatorname{Resin}(A, b, \Omega, \omega) = A \left[\frac{0,5(\Omega + \omega)}{b^2 + (\Omega + \omega)^2} + \frac{0,5(\Omega + \omega)}{b^2 + (\Omega + \omega)^2} \right] \quad (6.25)$$

$$\operatorname{Imsin}(A, b, \Omega, \omega) = -A \left[0,5 b \left(\frac{1}{b^2 + (\Omega - \omega)^2} - \frac{1}{b^2 + (\Omega + \omega)^2} \right) \right]. \quad (6.26)$$

Entsprechend gilt für die gedämpfte cos-Funktion:

$$\operatorname{Recos}(A, b, \Omega, \omega) = A \left[0,5 b \left(\frac{1}{b^2 + (\Omega - \omega)^2} + \frac{1}{b^2 + (\Omega + \omega)^2} \right) \right] \quad (6.27)$$

$$\operatorname{Imcos}(A, b, \Omega, \omega) = -A \left[\frac{0,5(\Omega + \omega)}{b^2 + (\Omega + \omega)^2} - \frac{0,5(\Omega - \omega)}{b^2 + (\Omega - \omega)^2} \right] \quad (6.28)$$

Zur Auswertung der Gl. (6.25-28) wurden für den Mikrorechner Funktions-Unterprogramme geschrieben, die zu beliebigen A, b, Ω, ω die jeweiligen Funktionswerte liefern. Wegen der erforderlichen Genauigkeit und des großen möglichen Zahlenbereichs wurde die Fließkommazahlendarstellung verwendet.

6.5.3 Die Bestimmung der Eigenfrequenzen

$X(\omega)$ sei die Fourier-Transformierte der gedämpften Schwingung (6.22). Die Frequenzen Ω_i der gedämpften Schwingung sind dann unter bestimmten Bedingungen identisch mit den Frequenzen, bei denen $|X(\omega)|$ ein relatives Maximum hat.

Zur Begründung dieser Behauptung wird zunächst die Schwingung in nur einer Eigenfrequenz untersucht, es ist also

$$x_1(t) = \begin{cases} e^{-bt} (A \sin \Omega t + B \cos \Omega t) & \text{für } t \geq 0 \\ 0 & \text{für } t < 0 \end{cases} \quad (6.29)$$

Für die weitere Untersuchung ist es vorteilhaft, die Schwingungsdifferentialgleichung

$$m\ddot{x} + d\dot{x} + cx = f(t) \quad (6.30)$$

zu betrachten. Mit den Abkürzungen [54]

$$\begin{aligned} \Omega_0 &= \sqrt{\frac{c}{m}} & , & \text{ Eigenkreisfrequenz der ungedämpften Schwingung} \\ D &= \frac{d\Omega_0}{2c} & , & \text{ Lehr'sches Dämpfungsmaß} \end{aligned} \quad (6.31)$$

hat (6.30) für $t \geq 0$ und falls $D < 1$ ist die homogene Lösung

$$x_2(t) = e^{D\Omega_0 t} (C_1 \sin \Omega_0 \sqrt{1-D^2} t + C_2 \cos \Omega_0 \sqrt{1-D^2} t). \quad (6.32)$$

C_1 und C_2 ergeben sich aus den Anfangsbedingungen. Ein Vergleich von (6.29) und (6.32) ergibt

$$\begin{aligned} \Omega &= \Omega_0 \sqrt{1-D^2} \\ b &= D \Omega_0 \end{aligned} \quad (6.33)$$

Das durch (6.30) beschriebene Schwingungssystem führt freie Schwingungen des Typs (6.29) aus, wenn

$$f(t) = \delta(t) \quad (6.34)$$

ist. Es werden nun von beiden Seiten der Gleichung

$$m\ddot{x} + d\dot{x} + cx = \delta(0) \quad (6.35)$$

die Fourier-Transformierten gebildet. Da die Transformierte von $\delta(0)$ gleich 1 ist und einer Differentiation im Zeitbereich einer Multiplikation mit $j\omega$ im Frequenzbereich entspricht, erhält man

$$\begin{aligned} -m\omega^2 X(\omega) + dj\omega X(\omega) + c X(\omega) &= 1 \\ X(\omega) &= \frac{1}{c - m\omega^2 + dj} \end{aligned} \quad (6.36)$$

Für $|X(\omega)|$ ergibt sich nach [55]

$$|X(\omega)| = \frac{1/c}{\sqrt{\left(1 - \left(\frac{\omega}{\Omega_0}\right)^2\right)^2 + \left(2D\left(\frac{\omega}{\Omega_0}\right)\right)^2}} \quad (6.37)$$

Durch Differenzieren und Nullsetzen des Nenners folgt für das Maximum von $|X(\omega)|$

$$\omega_{\max} = \Omega_0 \sqrt{1 - 2D^2} \quad \text{für } D^2 \leq 0,5 \quad (6.38)$$

Es ist also

$$\frac{\Omega}{\omega_{\max}} = \sqrt{\frac{1 - D^2}{1 - 2D^2}} \quad (6.39)$$

Bei den meisten Schwingungssystemen ist $D < 0,1$. Für jede der drei unten behandelten, schon relativ stark gedämpften Schwingungskomponenten ist $D < 0,02$, so daß allgemein in sehr guter Näherung

$$\Omega = \omega_{\max} \quad (6.40)$$

gesetzt werden kann. In dem ungünstigen Fall $D = 0,1$ beträgt der durch diese Vereinfachung gemachte Fehler 5‰.

Liegt nun eine Schwingung in mehr als einer Eigenfrequenz vor, so entspricht jeder Eigenfrequenz ein relatives Maximum von $|X(\omega)|$, sofern die Eigenfrequenzen im Spektrum weit genug auseinanderliegen. Als Beispiel sei eine Schwingung

$$\begin{aligned} f(t) = & e^{-0,01t} (0,3 \sin 1/s t + 0,2 \cos 1/s t) \\ & + e^{-0,02t} (0,2 \sin \frac{2,9t}{s} + 0,2 \cos \frac{2,9t}{s}) \\ & + e^{-0,08t} (0,3 \sin \frac{5,1t}{s} + 0,5 \cos \frac{5,1t}{s}) \quad \text{für } t \geq 0 \end{aligned} \quad (6.41)$$

und

$$f(t) = 0 \quad \text{für } t < 0$$

betrachtet. Abb. 6.8 stellt $f(t)$ und $|F(\omega)|$ dar, ermittelt durch eine diskrete Fourier-Transformation mit $2N = 1024$, $T = 300$ s. Aus Abb. 6.8 wird deutlich, daß nur in einer gewissen Umgebung einer Frequenz $|F(\omega)|$ von Null unterschieden werden kann. Liegen zwei Eigenfrequenzen zu dicht beieinander, überlagern sich die jeweiligen "Einflußzonen", und eine sichere Eigenfrequenzbestimmung aus den relativen Maxima ist nicht möglich.

6.5.4 Die Bestimmung weiterer Parameter durch Optimierung

6.5.4.1 Prinzipielles Vorgehen

Die analytischen Ausdrücke (6.25 - 28) der Fourier-Transformationen gedämpfter Schwingungen sind Funktionen der Amplitude, Frequenz, Phase und Dämpfung. Hier soll die Frage untersucht werden, wie diese Größen aus Meßdaten und deren Transformation ermittelt werden können. Ein Lösungsweg läßt sich in seinen Grundzügen wie folgt charakterisieren:

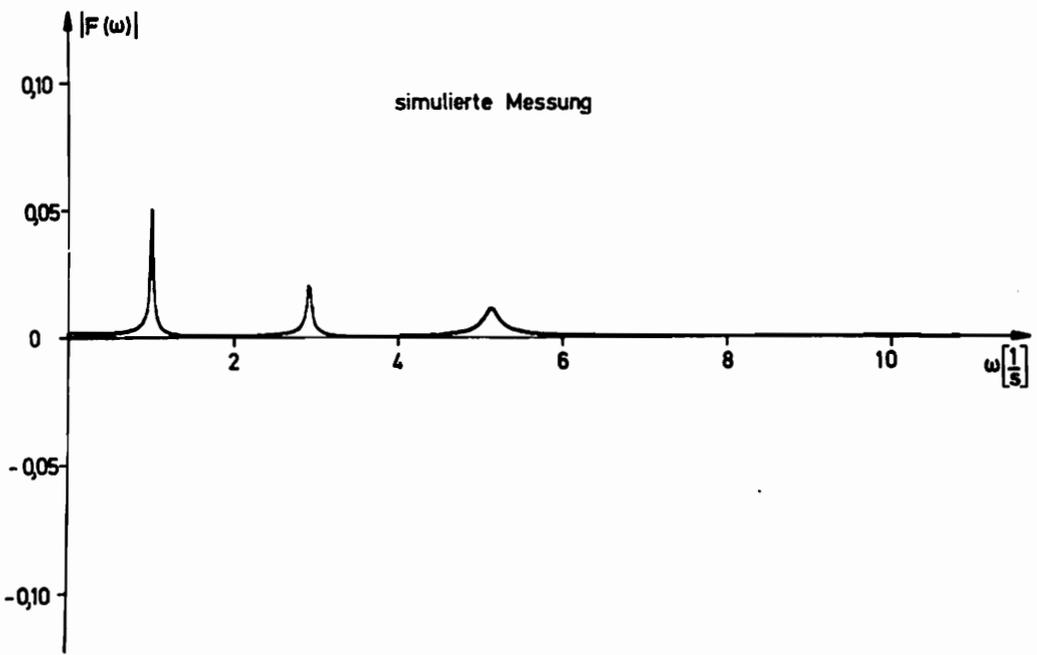
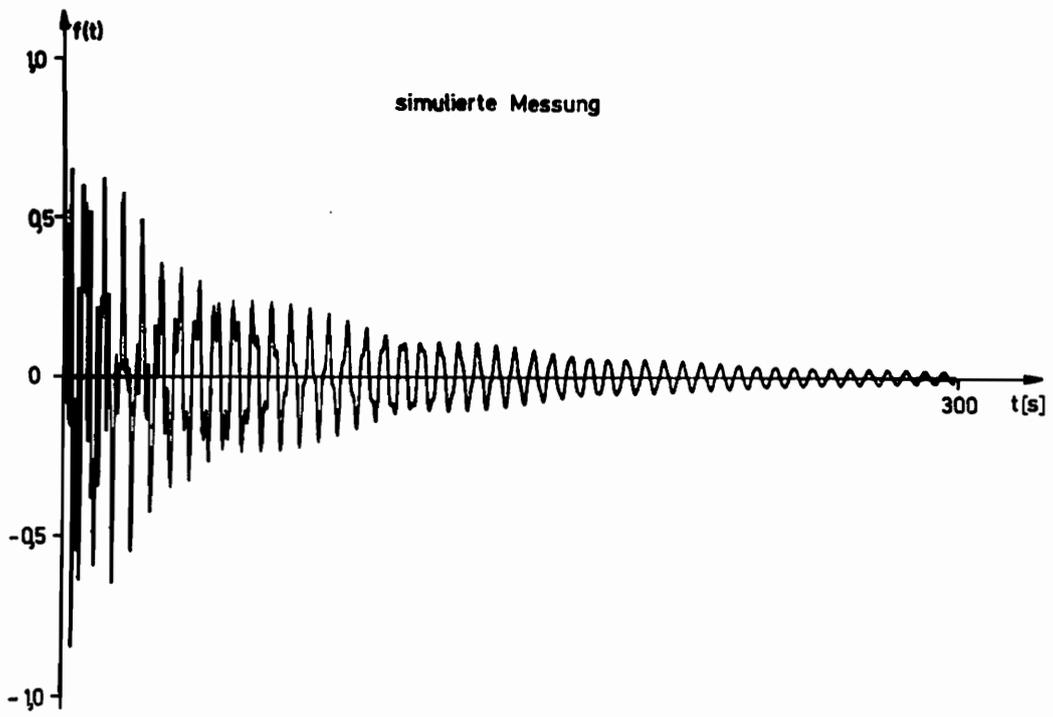


Abb. 6.8: Digital simulierte Schwingung

1. Schwingungsmessung, Abtastung und Transformation durch FFT
2. Bestimmung der Eigenfrequenzen aus den relativen Maxima des Betrages der Transformierten
3. Wahl einer Modellordnung k (Anzahl der berücksichtigten Eigenfrequenzen)
4. Bestimmung der A_i , B_i , b_i , $i = 1 \dots k$ so, daß die diskrete Fourier-Transformierte des Meßsignales und die analytische Fourier-Transformierte (Gl. 6.25-28) des Modells möglichst gut übereinstimmen.

Damit ist das Identifikationsproblem auf ein Parameteroptimierungsproblem zurückgeführt worden.

Als Kriterium für die Modellanpassung wurde die Summe der quadratischen Differenzen in Real- und Imaginärteil

$$G(A_i, B_i, b_i) = \sum_{i=1}^N \left(\text{Re} \left[F(\omega_i)_{\text{Modell}} - F(\omega_i)_{\text{Experiment}} \right]^2 \right) + \left(\text{Im} \left[F(\omega_i)_{\text{Modell}} - F(\omega_i)_{\text{Experiment}} \right]^2 \right) \quad (6.42)$$

gewählt. Sind die einzelnen Eigenfrequenzen im Spektrum deutlich voneinander getrennt, so ist es möglich, die Parameter für jede Eigenfrequenz einzeln anzupassen, dadurch verringert sich die Zahl der in einer Optimierung zu behandelnden Parameter erheblich. Ist man an der Phaseninformation nicht interessiert, so kann man obiges Kriterium ebenfalls verwenden, indem man den Realteil durch den absoluten Betrag ersetzt und den Imaginärteil zu 0 setzt, anstelle von A_i und B_i ist dann

$$P_i = \sqrt{A_i^2 + B_i^2} \quad (6.43)$$

gesuchter Parameter. In Gl. (6.25-28) können dann jeweils A_i und B_i unter Berücksichtigung von (6.43) beliebig gewählt werden. Je Eigenfrequenz sind so nur noch 2 Parameter (P_i und b_i) zu bestimmen.

Voraussetzung für eine sinnvolle Optimierung ist, daß Modellwerte und experimentell erhaltene Werte überhaupt übereinstimmen können, d.h. die oben diskutierten Fehler der diskreten Fourier-Transformation müssen ausgeschaltet sein. Aber selbst wenn diese Voraussetzung erfüllt ist, bleibt die Frage zu kären, ob mit dem oben eingeführten Gütekriterium überhaupt eine Optimierung möglich ist, da bei vielen Problemen der Einfluß einer Schwingungskomponente sich nur auf wenige Punkte im diskreten Frequenzspektrum erstreckt (ca. 20-40). Außerdem ist zu bedenken, daß durch die Diskretisierung im Frequenzbereich die häufig vorkommenden scharfen Spitzen nicht ganz genau abgebildet werden können (s. etwa Abb. 6.8). Um diesen Fragen nachzugehen, wurde anhand digital simulierter Schwingungen mit bekannten Parametern das Verhalten der Gütefunktion bei Parametervariationen untersucht. Die Ergebnisse sollen hier nicht im Detail wiedergegeben werden, es hat sich herausgestellt, daß eine Optimierung mit dem Gütekriterium (6.42) möglich ist.

6.5.4.2 Vorgehen bei schwacher Dämpfung

Ist das untersuchte System nur schwach gedämpft, so ist es häufig nicht möglich, die Meßzeit T so groß zu wählen, daß die Schwingung in T auf etwa 10 bis 15% der anfänglichen Auslenkung abgeklungen ist, wie ja in Abschnitt 6.3.2 für eine fehlerfreie Transformation gefordert wurde.

In solchen Fällen ist der Übergang zu einem Spezialfall der Laplace-Transformation zu empfehlen. Dazu ist die abgetastete Funktion im Zeitbereich mit $\exp(-\beta t)$ zu multiplizieren. Der Parameter β wird so gewählt, daß $\exp(-\beta T) \approx 0,1$ wird. Erst nach dieser Vorverarbeitung des Signals ist die FFT durchzuführen. Man kann hier von einem Spezialfall der Laplace-Transformation sprechen, da folgende Zusammenhänge bestehen [49] :

Für eine bei negativen t verschwindende Funktion $f(t)$ ist die Laplace-Transformation definiert durch

$$F(s) = \int_0^{\infty} f(t) e^{-st} dt \quad (6.44)$$

mit $s = \delta + j\omega$ als komplexer Veränderlicher. Wählt man speziell $\delta = \beta$, so ergibt sich aus (6.44) die Fourier-Transformation der Funktion

$$g(t) = f(t) e^{-\beta t} . \quad (6.45)$$

Am Optimierungsverfahren ändert sich dadurch nichts, man muß lediglich $\beta + b$; anstelle von b ; optimal bestimmen.

6.6 Auswertung digital simulierter Schwingungen

Zur Erprobung des Verfahrens wurden Optimierungsrechnungen für digital simulierte Schwingungen mit bekannten Parametern durchgeführt. Als Optimierungsmethode wurde die in Abschnitt 3.1.1 erwähnte konjugierte Gradientenmethode ohne lineare Suchen [10] benutzt. Vergleichsrechnungen hatten gezeigt, daß mit diesem Verfahren bei derselben Anzahl von Funktionswertbestimmungen größere Genauigkeit erzielt werden kann als mit der direkten Suchmethode von Hooke-Jeeves (s. Abschnitt 3.1.2.1).

Charakteristische Resultate sollen für die schon oben beschriebene Funktion (6.41) mit 3 Schwingungskomponenten und die Funktion

$$g(t) = \begin{cases} e^{\frac{-0,0012}{s} t} (0,3 \sin 1/s t + 0,2 \cos 1/s t) & \text{für } t \geq 0 \\ 0 & \text{für } t < 0 \end{cases} \quad (6.46)$$

diskutiert werden. Die Zeitfunktionen wurden so ausgewählt, daß bei einer "Meßzeit" von 300 s im ersten Fall

eine Transformation ohne ripple Fehler möglich ist, im zweiten dagegen dieser Fehler nur durch eine Laplace-Transformation vermieden werden kann.

Bei der Auswertung von Meßsignalen ist es manchmal möglich, aus dem Verlauf des Zeitsignals einzelne Parameter abzuschätzen. Dann kann man zur Ermittlung von Startwerten für die Optimierung so vorgehen:

1. Schätzung der Gesamtamplitude P_i einer Schwingungskomponente. Unter der Annahme, daß die sin- und cos-Anteile dieselbe Amplitude haben, ergeben sich die Schätzwerte

$$A_i \approx 0,7 P_i \quad , \quad B_i \approx 0,7 P_i \quad . \quad (6.47)$$

2. Schätzung, auf welchen Wert a eine Schwingung der Amplitude 1 im betrachteten Zeitintervall T abgekungen ist. Eine Näherung für b_i erhält man damit zu

$$b_i \approx - \frac{1}{T} \ln a_i \quad . \quad (6.48)$$

Startwerte, die nicht geschätzt werden können, müssen geraten werden.

In der folgenden Tabelle sind die Ergebnisse der Optimierung für die Funktion (6.41) zusammengefaßt. Für die Frequenzen wurden jeweils die wahren Werte zugrundegelegt. Diese Vereinfachung ist berechtigt, da sich die Frequenzen sehr genau bestimmen lassen (s. Abschnitt 6.5.3). Die Optimierung erfolgte getrennt für jede Eigenfrequenz. Für das Auffinden der optimalen Lösung wurden etwa 50 Berechnungen der Gütefunktion benötigt, darin ist die Annäherung des Gradienten durch finite Differenzen bereits enthalten. Werden in der Gütefunktion etwa 60 diskrete Frequenzpunkte berücksichtigt, so beträgt die Rechenzeit für eine Gütefunktionsbestimmung rund 5 s, damit nimmt die Identifikation eine Eigenfrequenz und ihrer Parameter ca. 5 min Rechenzeit in Anspruch. Tab. 6.1 zeigt, daß die

Amplituden A_i und B_i bis auf etwa 8% und die Dämpfungen b_i auf etwa 2% genau bestimmt werden können.

	<u>Startwerte</u>	<u>berechnete Optimalwerte</u>	<u>wahre Werte</u>
A_1	0,4	0,283	0,3
B_1	0,4	0,213	0,2
b_1	0,02	0,00995	0,01
A_2	0,3	0,188	0,2
B_2	0,3	0,186	0,2
b_2	0,04	0,0192	0,02
A_3	0,4	0,334	0,3
B_3	0,4	0,500	0,5
b_3	0,1	0,0803	0,08

Tab. 6.1: Optimierungsergebnisse I

Die Abb. 6.9,10 ermöglichen den Vergleich der optimalen Lösungen und der Sollwerte für Real- und Imaginärteil.

Zieht man zur Optimierung den absoluten Betrag des Spektrums heran, so ergibt sich folgendes Bild:

	<u>Startwerte</u>	<u>berechnete Optimalwerte</u>	<u>wahre Werte</u>
P_1	0,4	0,359	0,361
b_1	0,02	0,010	0,01
P_2	0,6	0,275	0,283
b_2	0,03	0,0197	0,02
P_3	0,4	0,567	0,583
b_3	0,05	0,0749	0,08

Tab. 6.2: Optimierungsergebnisse II

Abb. 6.11 stellt den Betrag der Transformierten für die optimale Lösung dar (die Sollkurve ist schon in Abb. 6.8 dargestellt worden).

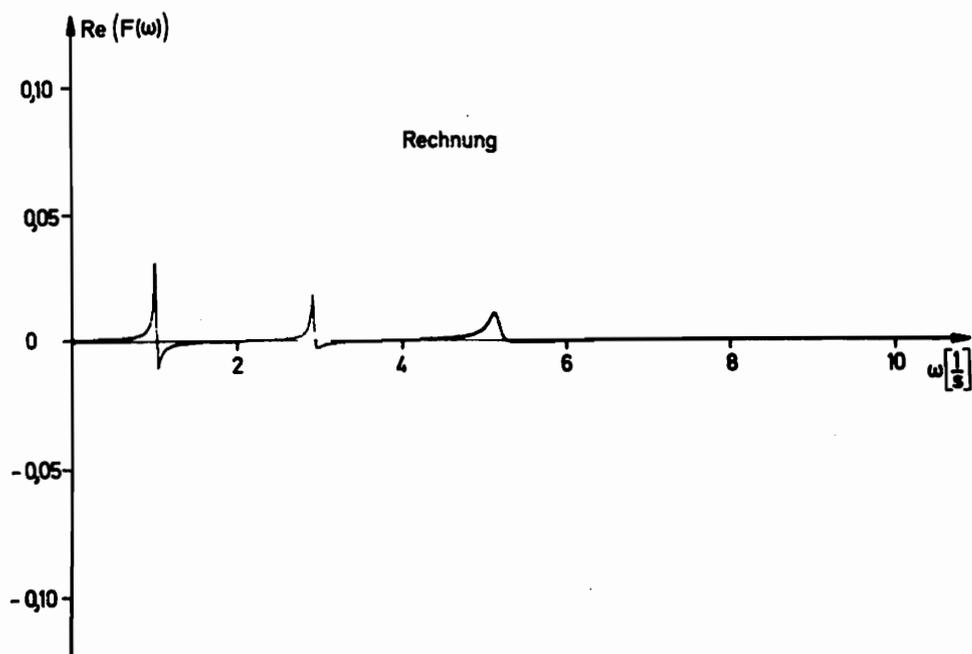
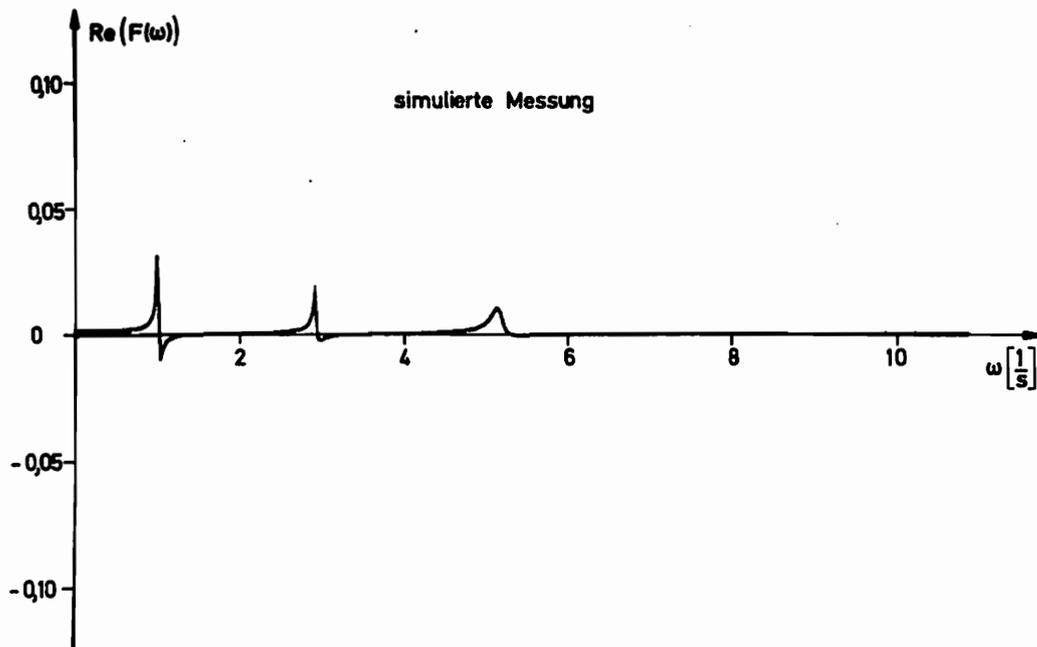


Abb. 6.9: Optimierungsergebnisse Realteil

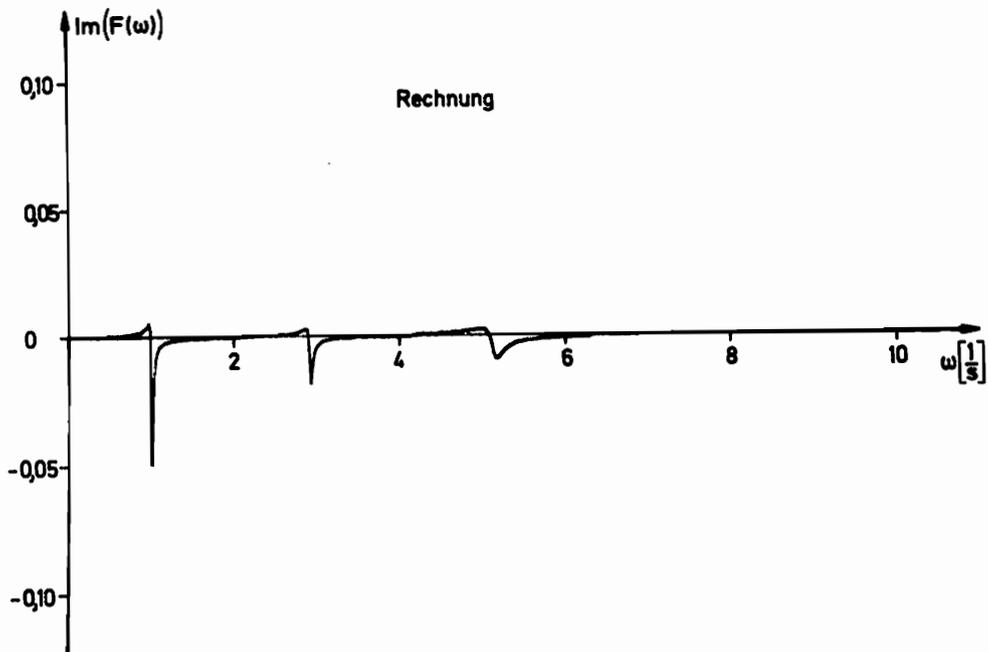
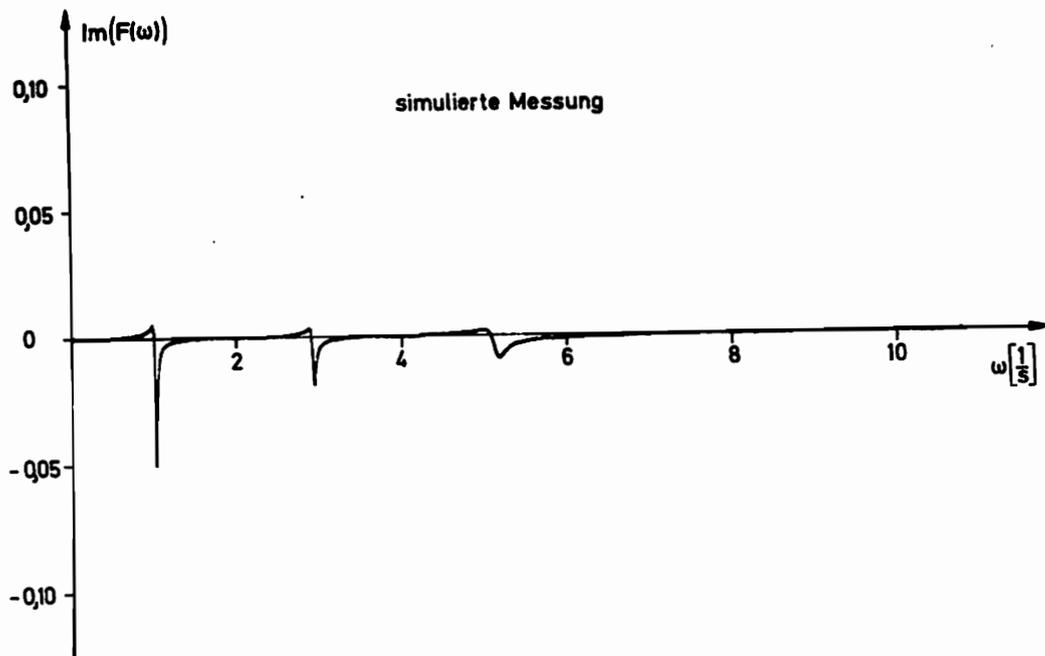


Abb. 6.10: Optimierungsergebnisse Imaginärteil

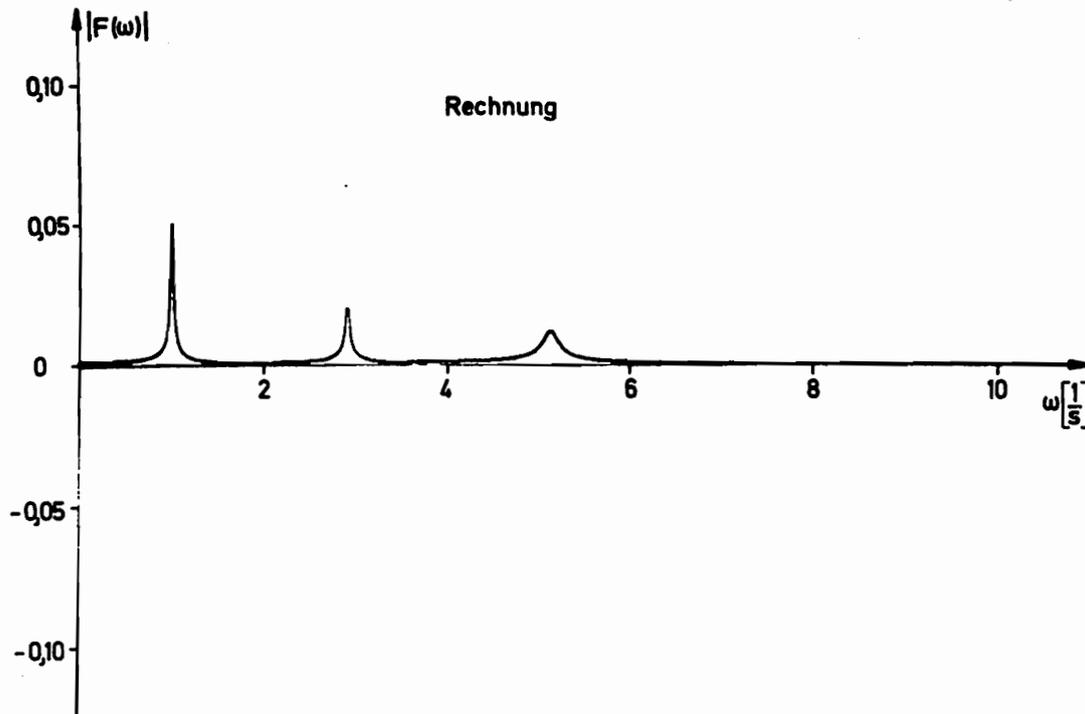


Abb. 6.11: Optimierungsergebnisse Betrag

Die Zahl der benötigten Gütefunktionsberechnungen lag hier bei etwa 30.

Da die Funktion $g(t)$ relativ schwach gedämpft ist (in der Meßzeit auf etwa 70% der Anfangsauslenkung), wurde eine Laplace-Transformation mit $\beta = 0,008$ durchgeführt. Tab. 6.3 zeigt die Resultate sowohl für eine Anpassung von Real- und Imaginärteil als auch für eine Anpassung des Betrages des komplexen Spektrums. Die negativen Resultate für b bzw. $b + \beta$ sind sinnvoll, da bei der Ermittlung des Betrages der Transformierten aus (6.25-28) nur gerade Potenzen von b auftreten, man hat also bei der Dämpfung den Betrag der Optimalwerte als Lösung aufzufassen.

Aus Tab. 6.3 wird deutlich, daß bei der Laplace-Transformation eine schwache Dämpfung nur relativ ungenau bestimmt werden kann, da bei der Identifikation immer $b + \beta$ ermittelt werden muß. Ein kleiner relativer Fehler bei der Bestimmung von $b + \beta$ bedeutet nämlich einen

umso größeren relativen Fehler bezüglich der allein interessierenden Größe b , je kleiner das Verhältnis b/β ist.

	Startwerte	Optimalwerte	wahre Werte
A	0,4	0,274	0,3
B	0,4	0,212	0,2
$b + \beta$	0,012	0,0089	0,0092
b	/	0,0009	0,0012
<hr/>			
P	0,2	0,342	0,361
$b + \beta$	0,005	-0,0091	-0,0092
b	/	-0,0011	-0,0012

Tab. 6.3: Optimierungsergebnisse III

6.7 Versuchsaufbau und -durchführung

Zur praktischen Erprobung des entwickelten Verfahrens wurden in Bochum Messungen an einer Turbinenschaufel und an der TU Hannover Messungen an einer Verdichtungs-schaufel durchgeführt. Abb. 6.12 zeigt den Bochumer Versuchsaufbau. Das zu untersuchende Objekt - etwa eine Turbinenschaufel (1) - wird in die in Beton eingegossene Haltevorrichtung eingespannt. Ein Stoß oder Schlag regt die Schaufel zu freien Schwingungen an, die so hervorgerufenen Beschleunigungen werden mit einem piezoelektrischen Beschleunigungsaufnehmer gemessen. Ein Ladungsverstärker (2) verstärkt das Signal und formt es in eine elektrische Spannung um. Dieses Signal wird auf einen Eingang des A/D Wandlers (3) gelegt. Ein Funktionsgenerator (4) erzeugt die variable Abtastfrequenz.

Die Abtastung erfolgt so, daß erst nach Überschreiten eines bestimmten, einstellbaren Schwellwertes eine Messung erfolgt. Auf diese Weise ist sichergestellt,

daß die Abtastung erst nach der schlagartigen Erregung beginnt und also wirklich nur die freien Schwingungen gemessen werden. Die digitalisierten Meßwerte werden im Mikrorechner (5) gespeichert und sofort nach der Messung auf dem Sichtgerät (6) dargestellt. So ist eine schnelle Kontrolle der Messungen möglich. Anschließend erfolgt die Fourier-Transformation und Optimierung.

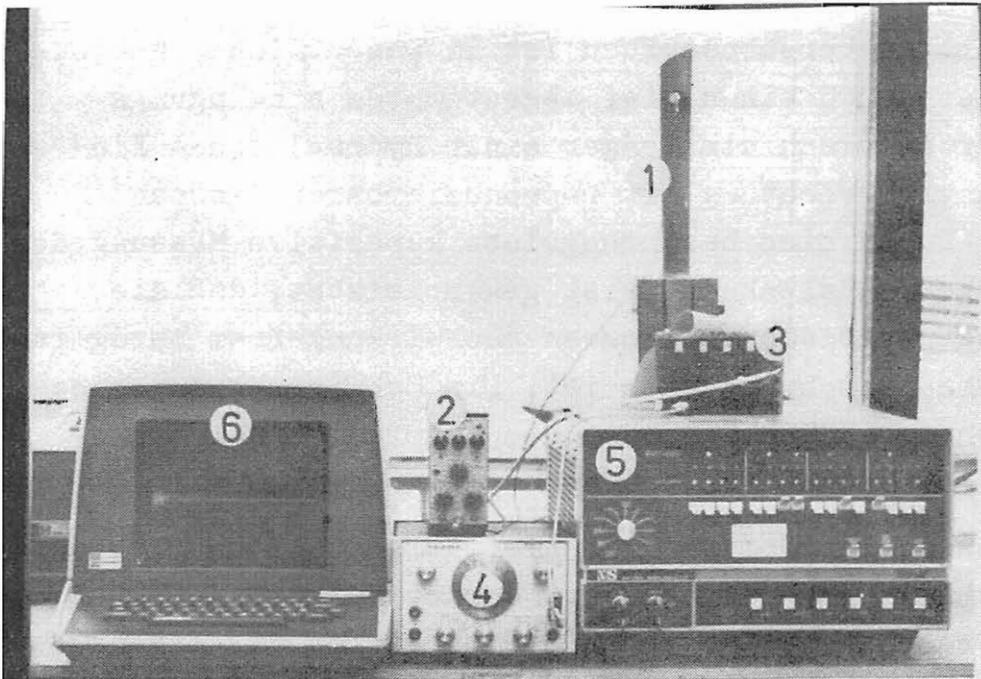


Abb. 6.12: Versuchsaufbau

Ähnlich wie bei der hybriden Optimierung werden die einzelnen digitalen Rechenschritte von einem Rahmenprogramm überwacht. Der Benutzer hat vielfältige Wahlmöglichkeiten, etwa zwischen Fourier- und Laplace-Transformation, Optimierung des komplexen Modellspektrums oder des absoluten Betrages. Weitere Unterprogramme bieten die Möglichkeit, Meßergebnisse auf einer Magnetplatte zu konservieren und später wieder abzurufen. Es ist bekannt, daß die Einspannung eines Objektes wesentlichen Einfluß auf die Lage der Eigenfrequenzen hat.

Da bei diesem Versuchsaufbau die Einspannkraft nicht reproduzierbar ist und außerdem die Erfassung der Schwingungsdämpfung durch den Beton allein kaum möglich ist, kann dieser einfache Versuchsaufbau nur zu einer prinzipiellen Durchführbarkeitsanalyse dienen. Aus diesem Grund wurden am Institut für Mechanik der TU Hannover Messungen an einer im Rahmen früherer Arbeiten [56] untersuchten Verdichterschaukel durchgeführt.

Der dortige Versuchsaufbau ist im wesentlichen gekennzeichnet durch einen tief abgestimmten schwingungs-isolierten Tisch als Träger einer hydraulischen Einspannungsvorrichtung mit reproduzierbarer Einspannkraft. Durch eine berührungslose kapazitive Messung der Schwingungsauslenkungen ist gewährleistet, daß die Messung praktisch vollkommen rückwirkungsfrei bezüglich der Schaukelschwingungen ist. Die Schwingungsanregung geschah durch Schläge oder den Aufprall einer Stahlkugel aus freiem Fall.

Ein besonderer Vorteil der hier dargestellten Methode der Schwingungsanalyse liegt darin, daß die Anregung nicht reproduzierbar zu sein braucht, da die Eigenfrequenzen und - im Rahmen einer linearen Theorie - die Dämpfungsparameter unabhängig vom Verlauf der Anregung sind. Es ist allerdings zu erwarten, daß dieselbe Anregung, an verschiedenen Stellen des Untersuchungsobjektes aufgebracht, die einzelnen Eigenfrequenzen unterschiedlich stark anregt. Eine Untersuchung der damit zusammenhängenden Fragen führt zur experimentellen Untersuchung von Schwingungseigenformen, dieser Komplex wird im Rahmen dieser Arbeit nicht behandelt.

Die vielseitigen Verwendungsmöglichkeiten des realisierten Mikrorechnersystems zur Versuchsauswertung kommen auch darin zum Ausdruck, daß soft- und hardwaremäßig keinerlei Anpassung an die speziellen Gegebenheiten des jeweiligen Versuchs nötig war. Eine ebenfalls durchgeführte Schwingungsmessung und -analyse an

einem Radialverdichterlaufrad soll hier nicht näher behandelt werden.

6.8 Versuchsergebnisse

6.8.1 Untersuchung einer Verdichterschaufel

Abb. 6.13 zeigt den Verlauf einer auf dem Hannover'schen Prüfstand gemessenen Schwingung und den Betrag ihrer Fourier-Transformierten. Das Zeitsignal ist in diesem und allen folgenden Fällen in 1024 Punkte diskretisiert. Da wegen der fehlenden und auch nicht benötigten Reproduzierbarkeit des erregenden Schlages die Größe der Schwingungsamplitude nur von untergeordnetem Interesse ist, enthalten die Abb. 6.13 ff den Ordinatenmaßstab der dimensionslosen rechnerinternen Darstellung.

Die Schwingung ist so schwach gedämpft, daß im betrachteten Zeitintervall praktisch kein Abklingen zu erkennen ist, deshalb wurde in Hinblick auf die Diskussion von Abschnitt 6.5.4.2 auf die Ermittlung von Amplituden und Dämpfungen verzichtet. Obwohl im Real- und Imaginärteil der Transformation erhebliche ripple Fehler auftreten müssen, lassen sich jedoch, wie aus Tab. 6.4 hervorgeht, die Eigenfrequenzen aus den Spitzen des Betrages des Spektrums sehr genau ermitteln. Zum Vergleich sind die von Vogt [56] rechnerisch nach einem Finite-Element-Verfahren ermittelten Eigenfrequenzen ebenfalls in die Tabelle aufgenommen worden. Bei den von Vogt durchgeführten Messungen wurde die Schaufel harmonisch erregt, und, nachdem der Einschwingungsvorgang abgeklungen war, die Schwingungsamplitude im Verhältnis zur Erregung bestimmt. Führt man das Verfahren für viele verschiedene Frequenzen durch, so erhält man mit dem Amplitudengang auch die Eigenfrequenzen - ein mühevolleres und zeitaufwendiges Verfahren im Vergleich zu dem hier vorgestellten, bei dem man die Eigenfrequenzen mit einem Schlag ermittelt.

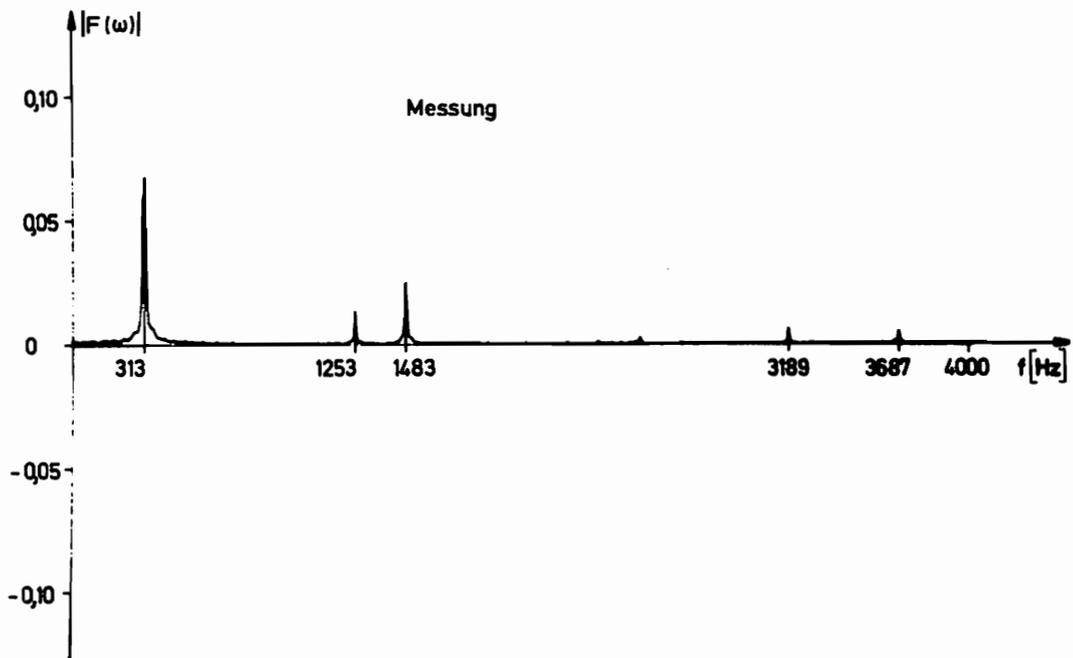
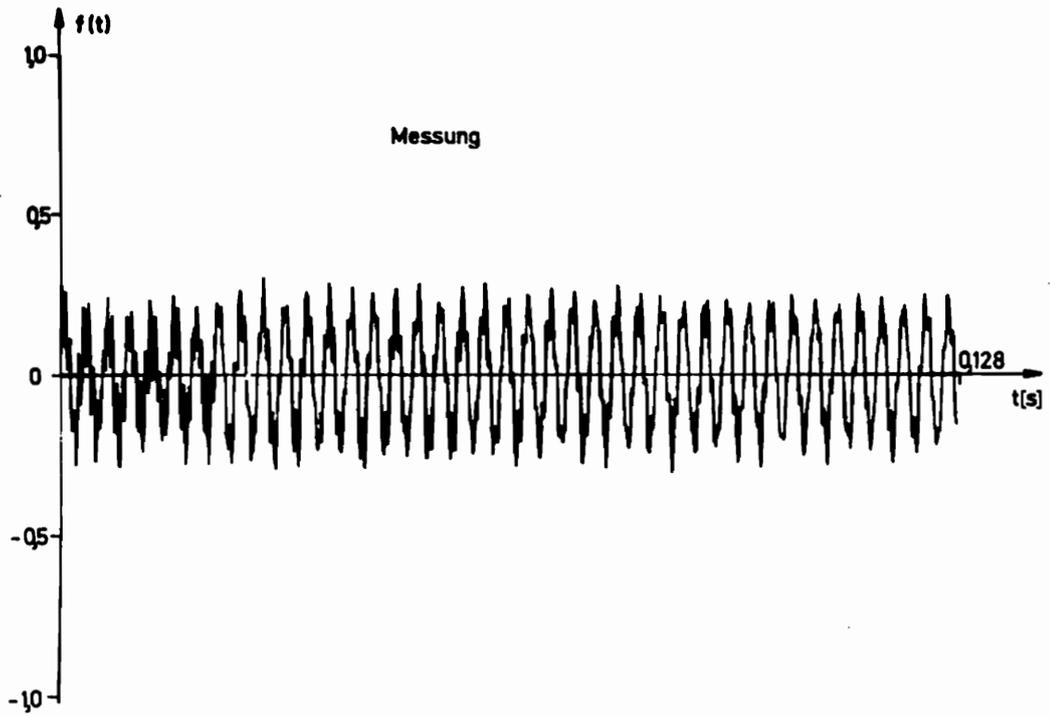


Abb. 6.13: Schwach gedämpfte Schwingung,
Eigenfrequenzen

	Rechnung Vogt	Messung Vogt	Messung Abb. 6.13
1. Biegung	307,5	315,5	313
1. Torsion	1201,6	1259,5	1253
2. Biegung	1407,7	1484,1	1483
2. Torsion	2961,4	3206,8	3189
3. Biegung	3495,4	3692,2	3687

Tab. 6.4: Eigenfrequenzen in Hz

6.8.2 Untersuchung einer Turbinenschaufel

Aus Abb. 6.14 ist oben ein typischer Schwingungsverlauf für den Bochumer Versuchsaufbau ersichtlich, die nächsten Abb. 6.15, 16, 17 zeigen jeweils oben den Real- und Imaginärteil und den absoluten Betrag des zugehörigen Spektrums. Für die erste, zweite und dritte Eigenfrequenz wurde eine Identifikation durchgeführt, die Ergebnisse sind in Tab. 6.5 zusammengefaßt.

		Startwerte	optimale Werte
Erste Eigenfrequenz $f_1 = 207,4$ Hz	A_1	0,1	-0,214
	B_1	0,3	0,274
	b_1	10,0	17,8
Zweite Eigenfrequenz $f_2 = 449,3$ Hz	A_2	0,2	0,05
	B_2	0,3	0,164
	b_2	10,0	9,09
Dritte Eigenfrequenz $f_3 = 584,1$ Hz	A_3	0,0	-0,02
	B_3	-0,2	-0,186
	b_3	20,0	34,2

Tab. 6.5: Optimierungsergebnisse IV

Die Abb. 6.15, 16, 17 enthalten unten die sich aus den optimalen Parametern ergebenden Funktionsverläufe. Besonders aus einem Vergleich der gemessenen und der aus den Optimierungsergebnissen folgenden Zeitverläufe

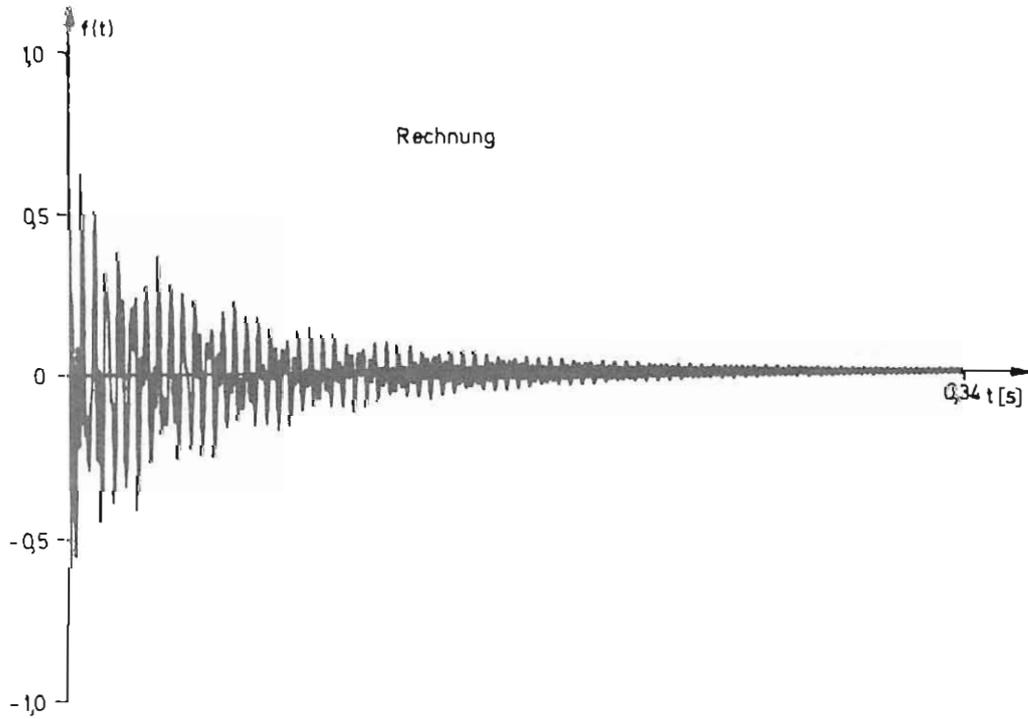
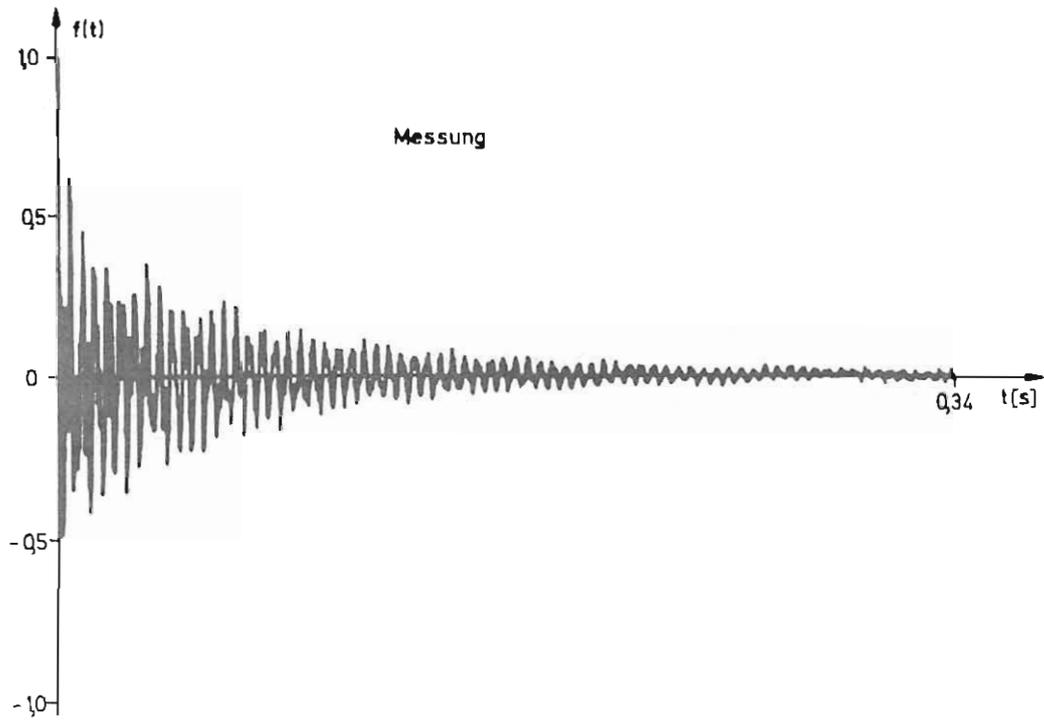


Abb. 6.14: Gemessene und identifizierte Schwingung

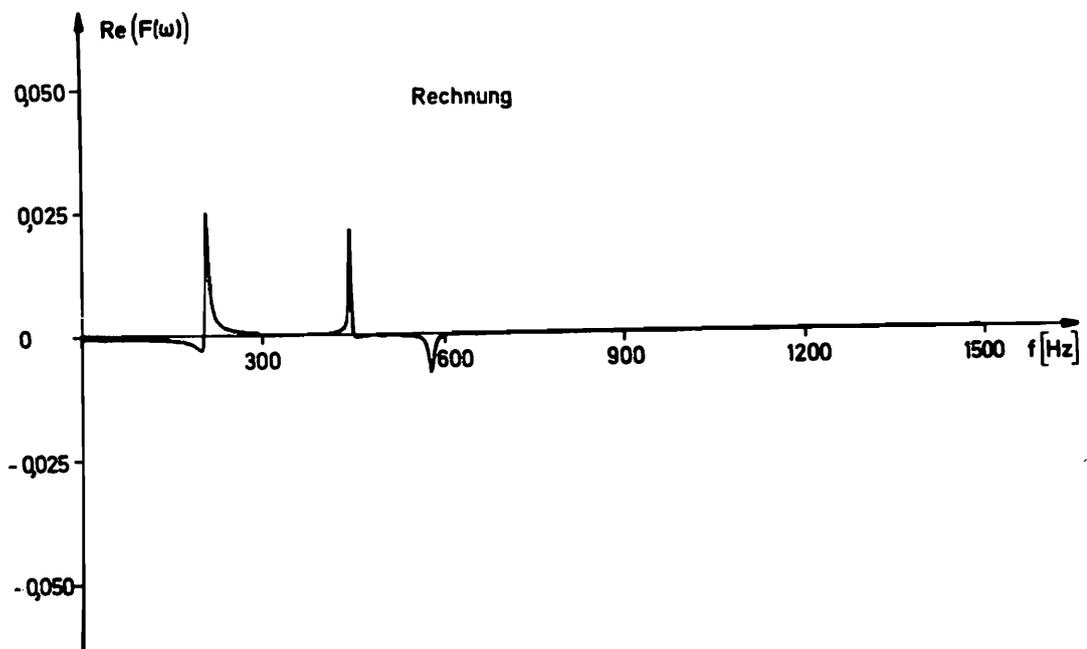
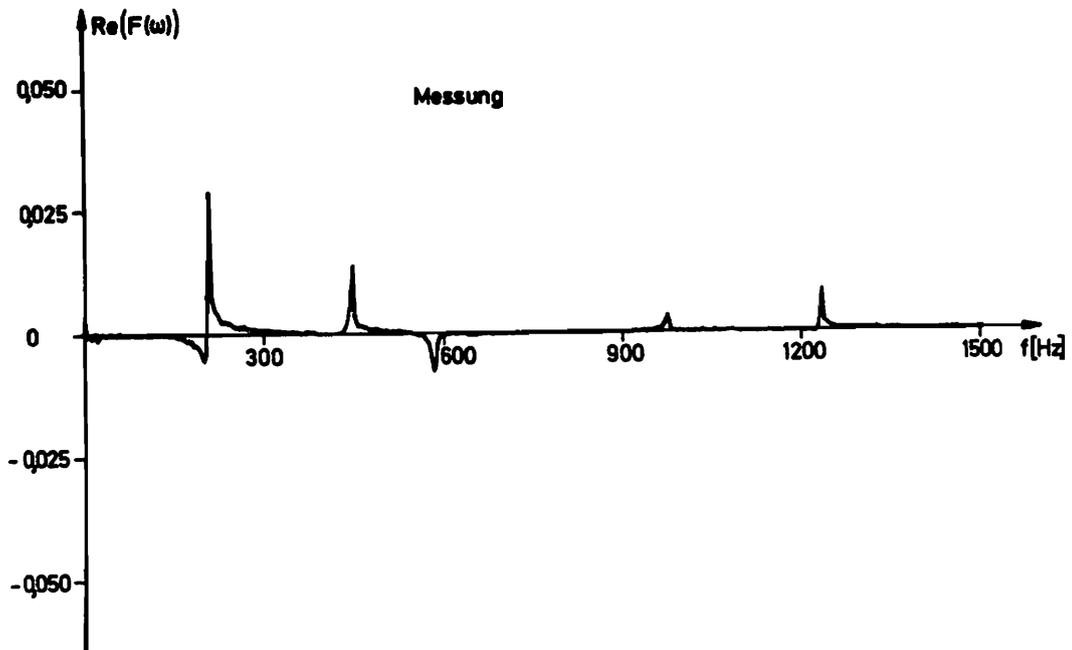


Abb. 6.15: Realteil aus Messung und Identifikation

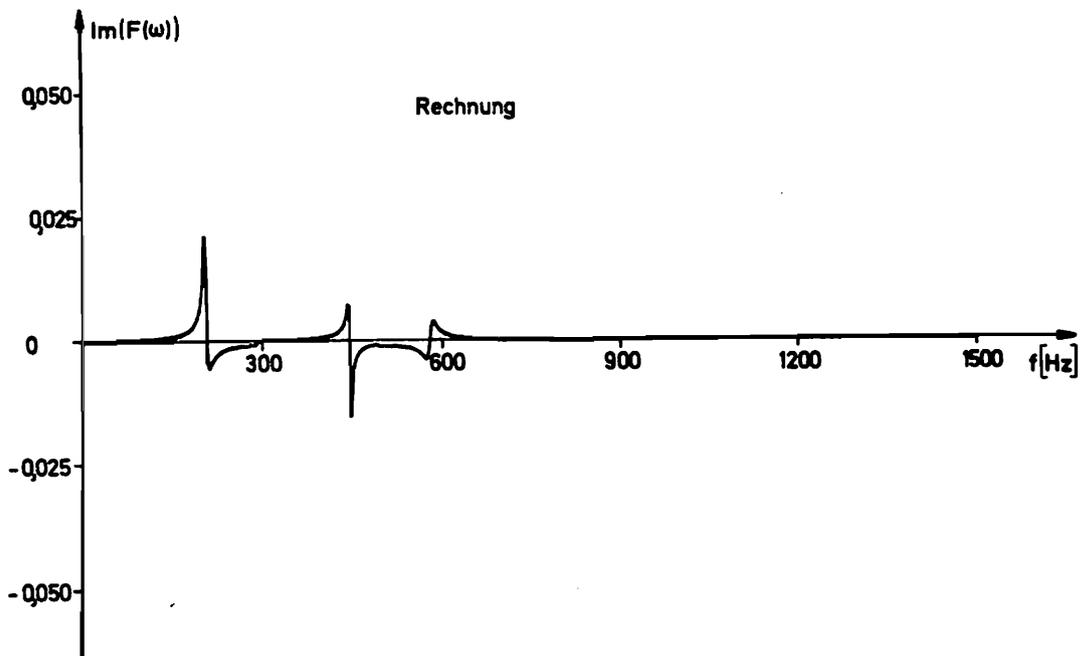
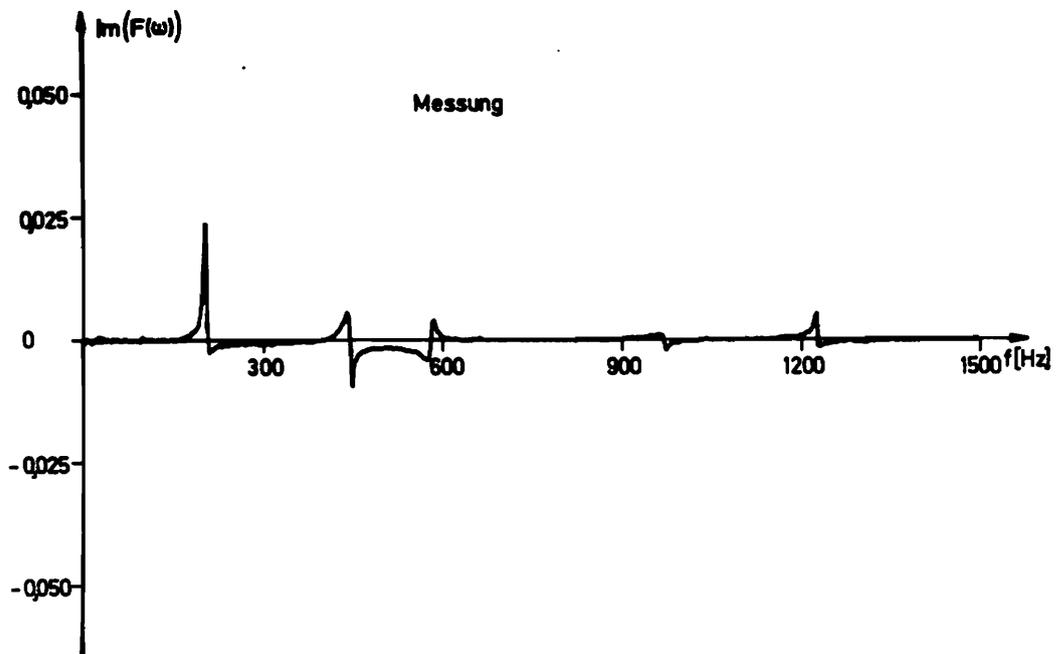


Abb. 6.16: Imaginärteil aus Messung und Identifikation

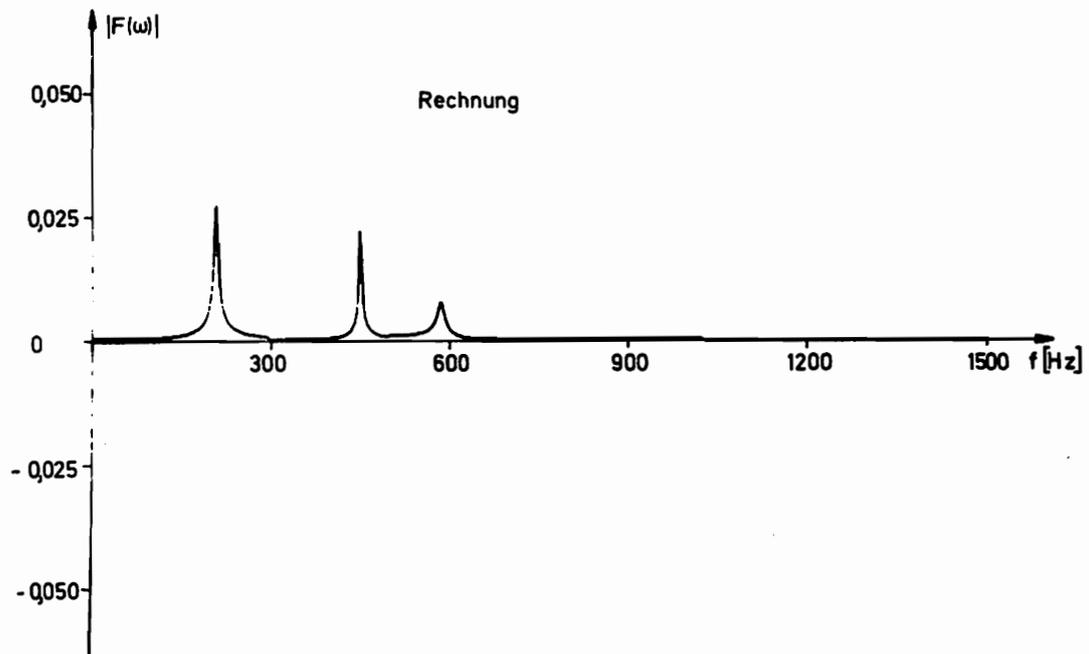
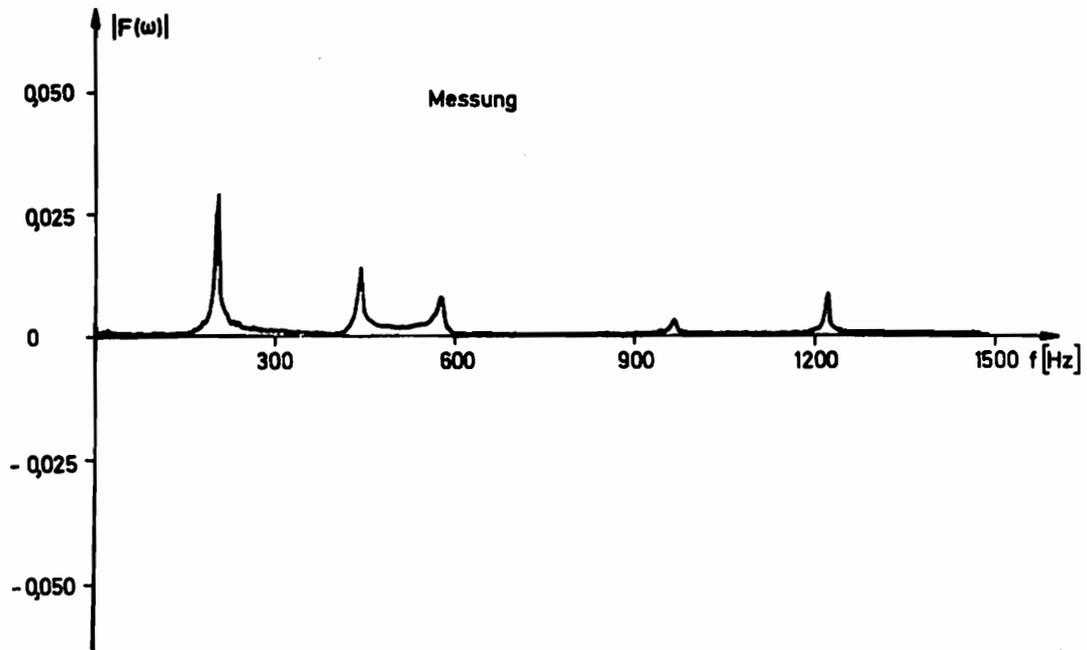


Abb. 6.17: $|F(\omega)|$ aus Messung und Identifikation

(Abb. 6.14) geht klar hervor, daß die wesentlichen Frequenz-, Amplituden-, Phasen- und Dämpfungscharakteristika der Schwingung durch die Fourier-Transformation und anschließende Optimierung zutreffend ermittelt werden konnten. Das bei der Optimierung vorausgesetzte viskose Dämpfungsgesetz erlaubt hier eine gute Beschreibung der realen Verhältnisse. Es soll aber nicht verschwiegen werden, daß für die zweite und dritte Eigenfrequenz je nach Startwerten auch erheblich andere Dämpfungen ermittelt wurden. Die Ursache dafür liegt vermutlich darin, daß eine zu geringe Zahl von diskreten Punkten im Frequenzbereich durch diese Eigenfrequenzen beeinflußt wird.

6.9 Diskussion und mögliche Behandlung anderer Schwingungsprobleme

Die Untersuchungen haben gezeigt, daß durch die Fourier-Transformation in freien Schwingungen auftretende Eigenfrequenzen gleichzeitig, schnell und genau ermittelt werden können. Die Bestimmung der zu den einzelnen Eigenfrequenzen gehörenden Amplituden und Dämpfungen gelingt mittels einer Optimierungsrechnung. In der Regel geben die Optimierungsergebnisse die vorhandenen Verhältnisse gut wieder, eine blind vertrauende Übernahme ist jedoch nicht ratsam, vielmehr sollten die Resultate der Optimierung stets auf Plausibilität geprüft werden.

Die zur Identifikation benötigte Rechenzeit liegt in einem vertretbaren Rahmen von etwa 5 min je Eigenfrequenz. Da davon ca. 90% auf die Unterprogramme zur Fließkommaarithmetik entfallen, kann man davon ausgehen, daß bei einem Rechner mit festverdrahteter Fließkommaarithmetik die Rechenzeit drastisch reduziert werden kann. Angesichts der rapiden Entwicklung auf dem Gebiet der Mikrorechner ist es durchaus möglich, daß in den nächsten Jahren Mikrorechner mit so weit fortgeschrittener Leistungsfähigkeit angeboten werden.

Es ist zu vermuten, daß die Genauigkeit der Identifizierung noch wesentlich gesteigert werden kann, wenn eine bandgespreizte Analyse, die sogenannte Zoom-FFT, durchgeführt wird. Bei der Zoom-FFT handelt es sich darum, das Zeitsignal derart zu transformieren, daß nicht der ganze Frequenzbereich von 0 Hz bis f_{max} abgedeckt wird, sondern vielmehr nur ein bestimmtes Frequenzband [57]. Dadurch läßt sich die Auflösung im Frequenzbereich wesentlich erhöhen, so daß bei der Berechnung der Gütefunktion erheblich mehr als die bisher typischen 50 Punkte berücksichtigt werden könnten.

Neben den in diesem Kapitel behandelten gedämpften harmonischen Schwingungen treten in der Praxis andere Schwingungsarten auf. Bei überkritisch gedämpften Systemen (Dämpfungsmaß $D > 1$) wird die freie Bewegung durch Exponentialfunktionen beschrieben. Da deren Fourier-Transformierten analytisch bestimmt werden können, erscheint auch in diesem Fall eine Bestimmung der Dämpfungsparameter überlagerter Bewegungskomponenten durch eine Optimierung im Frequenzbereich möglich.

Bei periodischen Vorgängen, wie sie etwa durch Unwuchterregung bei umlaufenden Maschinenteilen vorkommen, interessiert man sich für eine Fourier-Analyse der periodischen Signale. Das Programm zur FFT kann auch dafür eingesetzt werden, da sich durch eine diskrete Fourier-Transformation genau dann die Fourier-Koeffizienten der Analyse ergeben, wenn die Meßzeit T gleich einem Vielfachen der Periode ist. Mit einem Frequenzvervielfacher kann aus dem zu messenden periodischen Signal die Impulsfolge zur Abtastung einer oder mehrerer Perioden gewonnen werden.

Bei stochastischen Schwingungen sind die Autokorrelationsfunktion und das Leistungsspektrum wichtige, das Zeitverhalten kennzeichnende Funktionen. Zu ihrer Ermittlung aus Meßdaten wird nach [55] ebenfalls die Fourier-Transformation eingesetzt.

Die experimentelle Bestimmung von Schwingungseigenformen erfordert, daß gleichzeitig an verschiedenen Stellen der untersuchten Struktur die Antwort auf eine Erregung gemessen wird. Die Antworten müssen dann untereinander und zur Erregung in Beziehung gesetzt werden. Ein auf der Fourier-Transformation basierendes Verfahren dazu ist in der Literatur beschrieben [58] und wird in industriell hergestellten Meßwertverarbeitungsanlagen bereits verwendet. Über die erreichbare Genauigkeit liegen dem Verfasser keine Angaben vor.

6.10 Ausblick auf die Optimierung unter Echtzeitbedingungen

In dieser Arbeit ist die Optimierung dynamischer Systeme mit einem speziellen Hybridrechner behandelt worden, die Identifikation von Schwingungsparametern konnte ebenfalls durch eine Optimierungsrechnung erfolgen. Im ersten Fall wurden die Systeme auf dem Analogrechner mit unterschiedlicher Zeitskalierung simuliert, bei der Optimierung waren nur Zeitbedingungen des Rechners, nicht jedoch Echtzeitbedingungen des simulierten Systems zu beachten. Bei der Schwingungsanalyse spielten Echtzeitbedingungen nur bei der Messung, nicht bei der Optimierung eine Rolle.

In der Praxis gibt es daneben viele Optimierungsaufgaben, bei denen auch bei der Optimierung Echtzeitbedingungen wichtig sind. Beispielsweise soll bei einer adaptiven Regelung den Änderungen von Streckenparametern während des Betriebes so durch eine Anpassung von Reglerparametern entgegengewirkt werden, daß der Gesamtkreis sein günstiges dynamisches Verhalten behält. Bei solchen Problemen müssen die Reglerparameter in Echtzeit so schnell wie möglich angepasst werden. Für solche adaptiven Regelungen wird häufig ein Modellvergleichs-Verfahren benutzt [47], Abb. 6.18 zeigt das Prinzip.

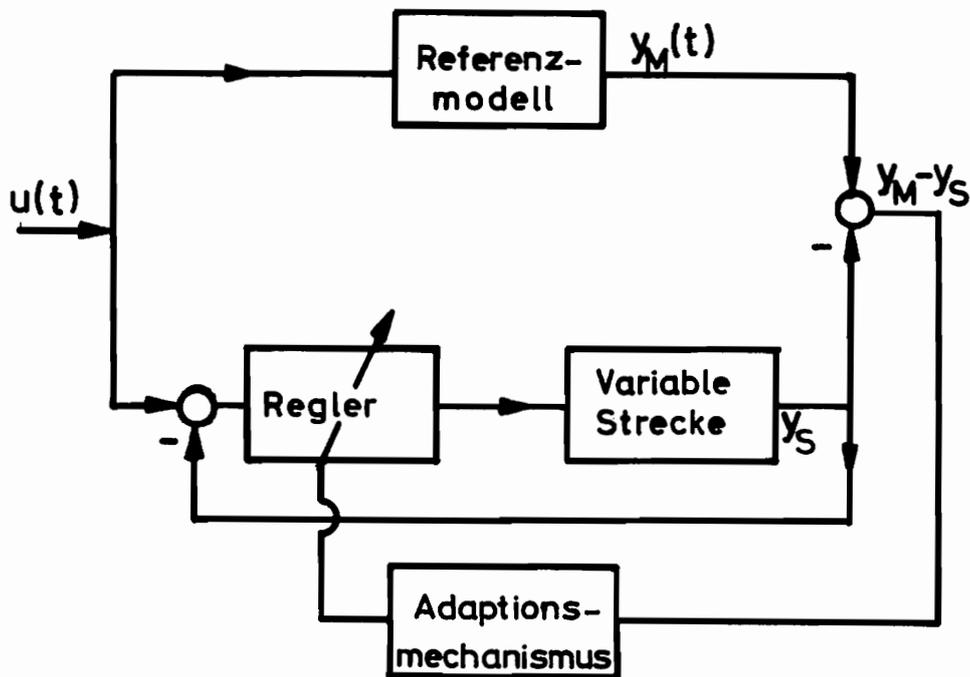


Abb. 6.18: Adaptiver Regelkreis mit Modellvergleich

Ein dynamisches Modell für Strecke und Regler liefert für jeden Eingang $u(t)$ das angestrebte Ausgangssignal $y_M(t)$. Das Modell kann in analoger Technik oder auch auf einem in Echtzeit arbeitenden Digitalrechner realisiert sein. Der Ausgang der veränderlichen Strecke wird mit dem gewünschten Ausgang verglichen und die Abweichung zwischen beiden als Fehler betrachtet. Durch eine Anpassung der Reglerparameter wird versucht, den Fehler zu minimieren. Für diese Anpassung kann z.B. ein Gradientenverfahren verwendet werden.

Optimierungsprobleme dieser Art waren nicht Gegenstand der hier dargestellten Arbeit, es soll jedoch darauf hingewiesen werden, daß ein Mikrorechner wie der hier vorgestellte gut die hardwaremäßige Basis für solche Optimierungen unter Echtzeitbedingungen sein kann.

7. Zusammenfassung

Aus dem Titel der vorliegenden Arbeit könnte der Leser den Eindruck gewinnen, hier liege eine merkwürdige Mischung moderner Rechentechnik, vertreten durch einen Mikrocomputer, mit altertümlicher und im Grunde überholter Analogrechentechnik vor. Eine solche Einschätzung kann begründet werden aus der Entwicklung der Rechentechnik in den letzten 20 Jahren.

Lange Zeit war der Analogrechner unbestritten führend bei der schnellen Lösung gewöhnlicher Differentialgleichungen. In Hinblick auf die erreichbare Genauigkeit und die Fähigkeit zu logischen Entscheidungen ist ihm der Digitalrechner jedoch überlegen. Deshalb wurden etwa ab 1962 Hybridrechner entwickelt, in denen die Vorzüge beider Rechnerarten vereint werden sollten. Die Kompliziertheit der Bedienung und die raschen Fortschritte in der rein digitalen Datenverarbeitung verhinderten jedoch, daß sich Hybridrechner allgemein durchsetzten, ihre Benutzung blieb stets eine Sache von Spezialisten.

Jedoch auch bei modernen Digitalrechnern stößt der Benutzer recht schnell an eine Kapazitätsgrenze, wenn er ein, möglicherweise nichtlineares, Differentialgleichungssystem viele hundert Mal mit unterschiedlichen Koeffizienten, Anfangs- oder Randbedingungen zu lösen hat. Die Notwendigkeit dazu ergibt sich schnell etwa bei der Optimierung dynamischer Systeme. Bei solchen Problemen stellen auch heute hybride Rechenverfahren eine echte Alternative zu rein digitalen dar.

Nach einem kurzen Abriß der verschiedenen Optimierungsmethoden - Parameteroptimierungen stehen dabei im Vordergrund - bildet daher die Beschreibung des Aufbaus eines Hybridrechners aus einem Mikrocomputer und einem Analogrechner den ersten Hauptteil dieser Arbeit. Es zeigt sich, daß das realisierte System industriell entwickelten in wesentlichen Punkten gleichwertig ist.

Anwendungen aus der Mechanik und Regelungstechnik unterstreichen im nächsten Teil die vielfältigen Einsatzmöglichkeiten des Hybridrechners zur Lösung von Optimierungsaufgaben. Die Probleme "Suboptimaler Zustandsreglerentwurf" und "Optimaler Entwurf eines dynamisch beanspruchten Stabes" stellen Aufgaben aus aktuellen Forschungsrichtungen dar, für die nach Wissen des Verfassers hybride Rechenverfahren bisher noch nicht eingesetzt worden sind.

Die bei den Anwendungen gewonnenen Erfahrungen zeigen, daß der Analogrechner durch seine Fähigkeit zur sehr schnellen Integration auch heute noch eine Existenzberechtigung hat. Seine Nachteile, besonders die geringe Genauigkeit, können durch unterstützenden Einsatz digitaler Methoden ausgeglichen werden. Ein Beispiel dafür ist die in Abschnitt 5.3.2.2 behandelte Eigenwertbestimmung. Als Folge der hohen analogen Rechengeschwindigkeit ergibt sich, daß bei der hybriden Optimierung dynamischer Systeme die theoretisch vorhergesagte Leistungsfähigkeit des jeweils benutzten Optimierungsverfahrens bei weitem nicht die Rolle spielt, die sie bei rein digitalen Rechnungen hat. Durch eine gut strukturierte Software wird die Benutzung des Rechensystems stark vereinfacht, sie bleibt aber schwieriger als bei konventionellen Analog- oder Digitalrechnern.

Viele technische Systeme sollen während des Betriebes optimiert werden, man denke etwa an adaptive Regelkreise oder Aufgaben der Prozessoptimierung, auch Identifikationsprobleme lassen sich in diesen Zusammenhang einordnen. Mit der Anwendung des entwickelten Rechensystems zur Schwingungsmessung und Identifikation von Schwingungsparametern wird eine Aufgabe aus diesem Problembereich behandelt. Die Parameteridentifikation geschieht dabei im Frequenzbereich. Zum Übergang vom Zeit- in den Frequenzbereich wird die diskrete Fourier-Transformation angewandt. Die dabei möglichen prinzipiellen Fehler finden besondere Beachtung.

Hybride Rechenanlagen sind zur Simulation und Erprobung von Echtzeitsystemen besonders gut geeignet, da sie durch ihr Arbeitsprinzip auf die hierbei typischen Zeitbedingungen besonders vorbereitet sind. Durch den beschriebenen Anwendungsfall wurde gezeigt, daß mit dem zugrundeliegenden Konzept freiprogrammierbare Rechen-systeme in Miniaturbauweise für die Optimierung technischer Systeme unter Echtzeitbedingungen entwickelbar sind.

8. Literaturverzeichnis

- [1] Leitmann, G.
(Editor) Optimization Techniques with
Applications to Aerospace
Systems
Academic Press, New York 1962
- [2] Athans, M.
and Falb, P. Optimal Control
McGraw Hill, New York 1966
- [3] Klein, B. Ein Beitrag zur rechnerunter-
stützten Analyse und Synthese
ebener Gelenkgetriebe unter
besonderer Berücksichtigung
mathematischer Optimierungs-
strategien und der Finite-
Element-Methode
Diss. Ruhr-Universität Bochum
1977
- [4] Hoffmann, U. und
Hofmann, H. Einführung in die Optimierung
mit Anwendungsbeispielen aus
dem Chemieingenieurwesen
Verlag Chemie, Weinheim 1971
- [5] Röhe, H. Anwendung iterativer Extrem-
wertsuchverfahren zur Berech-
nung optimaler Steuerfunkti-
onen an ausgewählten Beispie-
len der Regelungstechnik
Diss. TU Braunschweig 1973
- [6] Schwefel, P. Evolutionsstrategie und nume-
rische Optimierung
Diss. TU Berlin 1975
- [7] Aoki, M. Introduction to Optimization
Techniques
Macmillan, New York 1971

- [8] Wilde, D.G. Optimum Seeking Methods
Prentice Hall, Englewood
Cliffs 1964
- [9] Steward, G.W. A Modification of Davidon's
Minimization Method to Accept
Difference Approximations of
Derivatives
Journal of the ACM, Vol. 14
(1967), pp 72-83
- [10] Davidon, W.C. Optimally Conditioned Optimi-
zation Algorithms without Line
Searches
Math. Programming, Vol. 9
(1975), pp 1-30
- [11] Rechenberg, I. Evolutionsstrategie
Frommann-Holzboog Verlag,
Stuttgart 1973
- [12] Rastrigin, L.A. The Convergence of the Random
Search Method in the Extremal
Control of a Many Parameter
System
Automation and Remote Control,
Vol. 24 (1963), pp 1337-1342
- [13] Schumer, M.A. Adaptive Stepsize Random
and Steiglitz, K. Search
IEEE Transactions on Automatic
Control, Vol. 13 (1968),
pp 270-276
- [14] Korn, G.A. and A Proposed Hybrid Computer
Kosako, H. Method for Functional Optimi-
zation
IEEE Transactions on Computers,
Vol. 19 (1970), pp 149-153

- [15] Bekey, G.A. Optimization of Multiparameter Systems by Hybrid Computer Techniques
Simulation, Vol. 3 (1964), nos. 2, 3
- [16] Maybach, R. Solution of Optimal Control Problems on a High-Speed Hybrid Computer
Simulation, Vol. 7 (1966), pp 238-245
- [17] Bekey, G.A. and Gran, M.H. e.a. Parameter Optimization by Random Search Using Hybrid Computer Techniques
Proceedings Fall Joint Computer Conference, Spartan Books, Washington 1966, pp 191-200
- [18] Miura, T. and Tsuda, J. Solution of Optimal Control Problems Using Hybrid Computers
Electrical Engineering in Japan, 1966, pp 81-91
- [19] Steward, E.C. e.a. Study of a Global Search Algorithm for Optimal Control
Proceedings of the 5th AICA Conference, Lausanne, 1967
- [20] Bekey, G.A. and Karplus, W.J. Hybrid Computation
John Wiley, New York 1968
- [21] Gonzales, R.S. An Optimization Study on a Hybrid Computer
Annales de l'Association internationale pour le Calcul analogique
Vol. 12 (1970), pp 138-148

- [22] Volz, R. Example of Function Optimization via Hybrid Computation Simulation, Vol. 21 (1973), pp 43-48
- [23] Korn, G.A. and Andrews, M. Hybrid Computer Methods for Direct Functional Optimization IEEE Transactions on Computers, Vol. 24 (1975), pp 958-965
- [24] Hilgers, P. und Waller, H. Iterative Parameteroptimierung mit hybriden Analogrechnern Elektronische Rechenanlagen, Bd. 16 (1974), S. 108-116
- [25] Hilgers, P. und Waller, H. Die Synthese iterativer Optimierer und ihre Simulation mit dem hybriden Analogrechner Messen-Steuern-Regeln, Bd. 19 (1976), H. 2, S. 70-72, H. 3, S. 95-97
- [26] Reißerweber, B. Einführung in die Programmierung des Mikrocomputers IMP - 16 P
Manuskript des Instituts für Regelungs- und Steuerungstechnik, Universität Karlsruhe, 1976
- [27] National Semiconductor Corporation Handbücher zum Mikrorechner IMP 16 P
- [28] Hart, J.F. e.a. Computer Approximations John Wiley Inc. 1968

- [36] Levine, W. and Athans, M. On the Determination of the Optimal Constant Output Feedback Gains for Linear Multi-variable Systems
IEEE Transactions on Automatic Control
Vol. 15 (1970), pp 44-48
- [37] Gasch, R., Kik, W. und Knothe, K. Beeinflussung der Grenzgeschwindigkeit von Radsätzen durch Federung und Dämpfung in Längs- und Querrichtung
VDI Bericht 221, VDI Schwingungstagung 1974
- [38] Gasch, R., Kik, W. und Knothe, K. Passives Hilfssystem zur Erhöhung der Grenzgeschwindigkeit des stabilen Radsatzlaufes
VDI Bericht 221, VDI Schwingungstagung 1974
- [39] Kik, W. Einfluß konstruktiver Parameter auf die Laufstabilität eines Eisenbahnwagens mit Drehgestellen
VDI Bericht 269, VDI Schwingungstagung 1974
- [40] Knothe, K. Die dynamische Analyse von Mehrkörpersystemen zur Stabilitätsuntersuchung von Schienenfahrzeugen
VDI Bericht 269, VDI Schwingungstagung 1976

- [41] Sarma, G. Control Theory Applied to Railroad Car Dynamics Thesis, Polytechnic Institute of Brooklyn, 1970
- [42] Sarma, G. and Kozin, F. An Active Suspension System for the Lateral Dynamics of a High-Speed Wheel-Rail System Journal of Dynamic Measurement and Control (1971), pp 233-241
- [43] Mioduchowski, A. and Thermann, K. Minimum Weight Design of Structures with two Prescribed Frequencies Zeitschr. f. angew. Math. u. Mech., Bd. 54 (1974), S. 442-445
- [44] Meyers, A. Ein Beitrag zum optimalen Entwurf von schnelllaufenden Zentrifugenschalen Mitteilungen aus dem Institut für Mechanik Nr. 8 (1977), Ruhr-Universität Bochum
- [45] Ullenboom, W. Optimierung von Stäben unter nichtperiodischer dynamischer Belastung Mitteilungen aus dem Institut für Mechanik Nr. 11 (1977), Ruhr-Universität Bochum
- [46] Zander, G. Gewichtsminimaler Entwurf eines Balkens Studienarbeit am Institut für Mechanik, Ruhr-Universität Bochum 1974 (unveröffentlicht)

- [47] Landau, I. A Survey of Model Reference Adaptive Techniques - Theory and Applications
Automatica, Vol. 10 (1974), pp 353-379
- [48] Brigham, E.O. The Fast Fourier Transform
Prentice Hall, Englewood Cliffs, 1974
- [49] Waller, H. und Numerisches Berechnen von Krings, W. Schwingungs- und Kriechvorgängen mit der Laplace-Transformation
Ingenieur-Archiv Bd. 44 (1975), S. 335-346
- [50] Otnes, R.K. and Digital Time Series Analysis Enochson, L. J. Wiley, New York 1972
- [51] Krings, W. Beschreibung des Programmpaketes STO
Bericht, Arbeitsgruppe Numerisches Rechnen in der Mechanik und Simulationstechnik, Ruhr-Universität Bochum 1976 (unveröffentlicht)
- [52] Plessey Micro- IMP 16-P Fast Fourier Transform systems Package
Firmenunterlagen 1976
- [53] Oberhettinger, F. Tabellen zur Fourier-Transformation
Springer Verlag Berlin, 1957

- [54] Magnus, K. Schwingungen
Teubner Verlag, Stuttgart 1976
- [55] Bendat, J.S. and Random Data: Analysis and
Piersol, A.G. Measurement Procedures
J. Wiley, New York 1971
- [56] Vogt, H. Berechnung der Eigenfrequen-
zen und Eigenformen von Schau-
feldn thermischer Turbomaschi-
nen mit Hilfe eines finiten
Shallow-Shell-Elementes
Sonderforschungsbereich 61
Technische Hochschule Hannover
Bericht 7 zum Teilprojekt D 1,
1974
- [57] McKinney, H.W. Band-Selectable Fourier Ana-
lysis
Hewlett-Packard Journal
Vol. 26 (1975), No. 8,
pp 20-24
- [58] Richardson, M. Identification of the Modal
and Potter, R. Properties of an Elastic
Structure from Measured Trans-
fer Function Data
20th International Instrumen-
tation Symposium, Albuquerque,
New Mexico 1974

Anhang: Der Befehlssatz des Mikrorechners IMP 16 P

Zum besseren Verständnis der im Text enthaltenen Programmteile ist neben den mnemotechnischen Abkürzungen die englische Bezeichnung der Befehle angegeben. Die Übersicht enthält außerdem die Befehlsausführungszeiten in μ s.

1. Speicher Befehle

LD	Load	8,05
LD @	Load Indirect	8,58
ST	Store	9,275
ST @	Store Indirect	12,6
LDB	Load Byte	30,1
LLB	Load Left Byte	30,1
LRB	Load Right Byte	18,9
STB	Store Byte	36,05
SLB	Store Left Byte	36,05
SRB	Store Right Byte	26,25

2. Arithmetische Befehle

ADD	Add	8,05
SUB	Subtract	8,05
MPY	Multiply	149,975 bis 172,375
DIV	Divide	176,575 bis 224,175
DADD	Double Precision Add	18,9
DSUB	Double Precision Subtract	18,9

3. Logische Befehle

AND	Logical And	8,05
OR	Logical Or	8,05

4. Register Befehle

PUSH	Push onto Stack	4,8
PULL	Pull from Stack	4,8
XCHRS	Exchange Register and Stack	7,525
AISZ	Add Immediate, Skip if Zero	bei Sprung 7,53 sonst 6,2

CAI	Complement and Add Immediate	4,8
RADD	Register Add	4,8
RXCH	Register Exchange	11,725
RCPY	Register Copy	8,925
RXOR	Register Exclusive Or	8,925
RAND	Register And	8,925

5. Bit und Statusflag Befehle

PUSHF	Push Status Flags onto Stack	6,125
PULLF	Pull Status Flags from Stack	7,53
SETST	Set Status Flag	24,325 bis 50,925
CLRST	Clear Status Flag	24,325 bis 50,925
SETBIT	Set Bit	21,525 bis 48,125
CLTBIT	Clear Bit	21,525 bis 48,125
CMPBIT	Complement Bit	21,525 bis 48,125

6. Sprungbefehle

JMP	Jump Direct	4,8
JMP	Jump Indirect	8,05
JMPP	Jump through Pointer	11,375
JSR	Jump to Subroutine	6,2
JSR @	Jump to Subroutine Indirect	9,45
JSRI	Jump to Subroutine Implied	6,125
JSRP	Jump to Subroutine through Pointer	12,775
RTS	Return from Subroutine	6,125
RTI	Return from Interrupt	7,525
BOC	Branch On Condition	bei Sprung 7,525 sonst 6,125
ISZ	Increment and Skip if Zero	11,2 (Skip 12,6)
DSZ	Decrement and Skip if Zero	12,6 (14,0)
SKG	Skip if Greater	12,25 (15,05)
SKNE	Skip if Not Equal	9,45 (10,85)
SKAZ	Skip if And is Zero	9,45 (10,85)
SKSTF	Skip if Status Flag true	27,125 (55,125)
SKBIT	Skip if Bit true	27,125 (55,125)

7. Schiebe Befehle

SHL	Shift Left	10,325
SHR	Shift Right	10,325
ROL	Rotate Left	10,325
ROR	Rotate Right	10,325

8. Interrupt Befehle

ISCAN	Interrupt Skan	13,125 bis 112,5
JINT	Jump to Level 0 Interrupt, Indirect	10,85

9. Ein- Ausgabe, Control Flag und Halt Befehle

RIN	Register In	10,325
ROUT	Register Out	10,325
SFLG	Set Flag	6,125
PFLG	Pulse Flag	6,125
HALT	Halt	6,125

**Mitteilungen aus dem Institut für Mechanik
RUHR-UNIVERSITÄT BOCHUM
Nr. 14**