

*İhtiyarla Şismana*

Fachgruppe für Automatik

Prof. Dr. M. Mansour/Prof. Dr. W. Schaufelberger

Zürich, 22. Oktober 1980

Physikstrasse 3  
Telefon 01 326211

Postadresse:  
Fachgruppe für Automatik  
ETH-Zentrum  
CH-8092 Zürich

DA.7

Wintersemester 1980/81

D I P L O M A R B E I T

für

Herrn Selim Hacisalihzade  
Herrn Luis von Achenbach

SYNTHESE VON SUBOPTIMALEN ROBUSTEN, LINEAREN SYSTEMEN

A. Beschreibung

Bei linearen Systemen mit variablen Parametern kann die Sensitivität der Systemeigenschaften auf Parameteränderungen durch Rückführung verändert werden. Die Sensitivität der Trajektorie des optimalen dynamischen Systems wird dabei auf die Sensitivität von Eigenwerten des geschlossenen Systems zurückgeführt.

Eine neue Methode bietet die Möglichkeit der Synthese von robusten Systemen, indem die vorgegebenen Schranken für die Systemtrajektorie für alle Parameter aus einem Gebiet des Parameterraumes angehalten werden können.

Ziel dieser Diplomarbeit ist, diese Methode zu studieren und ihre Brauchbarkeit für die Synthese von robusten, suboptimalen Systemen zu untersuchen.

Ein Programmpaket soll entwickelt und an Hand einiger Beispiele erprobt werden. Die erhaltenen Resultate werden durch Systemsimulation getestet und diskutiert.

B. Aufgabenstellung

- B1. Man studiere die Regelung von linearen Systemen mit variablen Parametern [2],[3] und gebe eine einführende Uebersicht über die Methoden für den Entwurf von robusten Regelsystemen.
- B2. Man studiere die Methode in [1] und fasse sie kurz zusammen.
- B3. Ein Algorithmus für die Methode in [1] soll entwickelt und an einem digitalen Rechner programmiert werden.
- B4. Das in B3 erhaltene Programmpaket soll an einigen Beispielen [4] angewendet werden.
- B5. Man verifiziere das Verhalten des suboptimalen robusten Systems durch Systemsimulation und diskutiere die Resultate.
- B6. Man simuliere das offene System und vergleiche es mit dem Verhalten des robusten suboptimalen sowie des optimalen Systems.
- B7. Man schreibe einen ausführlichen Bericht. Dieser soll eine Zusammenfassung mit den folgenden Abschnitten enthalten:
1. Problemstellung
  2. Lösungsmethoden
  3. Ergebnisse
  4. Ungelöste Probleme

Jeder der beiden Herren soll die Hälfte des Berichtes verfassen, diese bezeichnen und unterschreiben. Von dieser Vorschrift darf unter keinen Umständen abgegangen werden.

Das Datum für die Präsentation der Arbeit wird zu einem späteren Zeitpunkt bekannt gegeben.

Das ausgefüllte Registraturblatt und die weiße Kontrollkarte müssen zusammen mit dem Bericht (2 Exemplare) im Sekretariat ETL I23 abgegeben werden.

Zuständig: Dipl.Ing. M.Djordjevic  
Dipl.Ing. F.Wiederkehr

  
Prof.Dr.M.Mansour

#### Literatur

- [1] R.K.Krishnan, S.Brzezowski: Design of robust linear regulator with prescribed trajectory insensitivity to parameter variations. IEEE Trans.on Automatic Control, Vol.AC-23, pp.474-478, 1978
- [2] Horisberger, P.H., Belanger, R.P.: Regulators for linear, time invariant plants with uncertain parameters. IEEE Trans. on Automatic Control, vol.AC-21, pp.705-708, 1976
- [3] E.Kreindler: Closed-loop sensitivity reduction of linear optimal control systems. IEEE Trans.on Automatic Control, vol.AC-13, pp.254-262 1968
- [4] G.Lekkas u.a.: Identifikation dynamischer Systeme durch nichtlinearen Modellabgleich. Interner Bericht der Fachgruppe für Automatik,Nr.77-04

Ausgabe der Diplomarbeit: Montag, den 27.Oktober 1980

Abgabe der Diplomarbeit: Montag, den 22.Dezember 1980, 17.00 Uhr

## INHALTSVERZEICHNIS

1. EINLEITUNG .....	1
1.1. Problemstellung und Motivation .....	1
1.2. Vorgehen .....	3
2. EINIGE METHODEN FUER DEN ENTWURF VON ROBUSTEN REGELSYSTEMEN .....	5
2.1. Einleitung .....	5
2.2. Mathematische Grundlagen der Sensitivitätstheorie ..	6
2.2.1 Herleitung des Begriffs der Sensitivität .....	6
2.2.2 Anwendung auf die Theorie der linearen Regelsysteme .....	7
2.3. Methoden mit differentiellen Zustands- sensitivitätsterme in der Zielfunktion .....	9
2.3.1 Einleitung .....	9
2.3.2 Die Methode von Kreindler .....	11
2.3.3 Die Methode von Rao und Soudack .....	13
2.3.4 Die Methode von Fleming und Newmann .....	14
2.4. Die Methode der maximal zulässigen Systemmatrixänderungen .....	16
3. DIE METHODE VON KRISHNAN UND BRZEZOWSKI .....	19
3.1. Einleitung .....	19
3.2. Beschreibung der Methode .....	20
3.3. Der Entwurfsalgorithmus .....	23
3.4. Zusammenfassung .....	25
3.5. Einige Vereinfachungen und Erläuterungen zur beschriebenen Methode .....	25
4. PROGRAMMIERUNG DER VON KRISHNAN UND BRZEZOWSKI ANGEGEBENEN METHODE .....	35

4.1. Einleitung .....	35
4.2. Beschreibung des Hauptprogrammes und der Subroutinen .....	38
4.3. Flussdiagramme .....	51
4.4. Das Programm ROBUST .....	53
5. ANWENDUNG DES PROGRAMMPAKETES AN DREI BEISPIELEN .....	65
5.1. "Pitch-axis control system" eines Flugzeuges .....	65
5.1.1 Systembeschreibung .....	65
5.1.2 Das offene System .....	67
5.1.3 Das suboptimale, robuste System .....	70
5.1.4 Das optimal-geregelte System .....	82
5.1.5 Die $\lambda_{\text{krit}}(\rho)$ -Kurve .....	89
5.2. Regelung eines elektromechanischen Systems .....	90
5.2.1 Systembeschreibung .....	90
5.2.2 Das offene System .....	94
5.2.3 Das suboptimale, robuste System .....	99
5.2.4 Das optimal-geregelte System .....	107
5.2.5 Die $\lambda_{\text{krit}}(\rho)$ -Kurve .....	112
5.3. Regelung einer hydraulischen Werkzeugmaschine .....	113
5.3.1 Systembeschreibung .....	113
5.3.2 Das offene System .....	117
5.3.3 Das suboptimale, robuste System .....	121
5.3.4 Das optimal-geregelte System .....	146
6. SCHLUSSBEMERKUNGEN UND UNGELOESTE PROBLEME .....	154
LITERATURVERZEICHNIS .....	156

## 1. EINLEITUNG

### 1.1. Problemstellung und Motivation

Die meisten Regelsysteme - seien es technische, biologische oder kombinierte - hängen von gewissen Parametern ab, die sich zeitlich ändern können. Es ist bekannt, dass eine Rückführung u.a. eine Sensitivitätsverminderung des Ausgang-/Eingangverhältnis, bezüglich Änderungen der Systemparameter gegenüber deren Nominalwerten zur Folge hat. Kreindler gibt in [13] Ungleichungen an, die diese Eigenschaft der Empfindlichkeitsverringernng eines geschlossenen linearen optimalen Systems beweisen.

Eine Methode den Regelkreis immer optimal, bezüglich einer gegebenen Zielfunktion, trotz Parametervariationen, zu gestalten, ist die der adaptiven Regelung. Bei dieser Regelungsart werden die aktuellen Werte der Parameter der Regelstrecke - durch Identifikation - bestimmt und für diese Werte stellt sich der Regler automatisch neu ein. Üblicherweise besteht also ein adaptives Regelsystem aus einer sogenannten primitiven Schleife (bestehend aus Regler und Strecke) und einer adaptiven Schleife (bestehend aus Identifikationsstufe, Entscheidungsstufe und Modifikationsstufe). Siehe Fig. 1.1.

Die adaptive Regelung ist im allgemeinen nichtlinear. Es gibt eine Reihe theoretischer Publikationen und Bücher über die adaptiven Systeme, dagegen sind sehr wenige Berichte, betreffend deren praktischen Anwendungen zu finden. Ein Grund dafür ist sicher die Tatsache, dass diese Systeme relativ kompliziert sind.

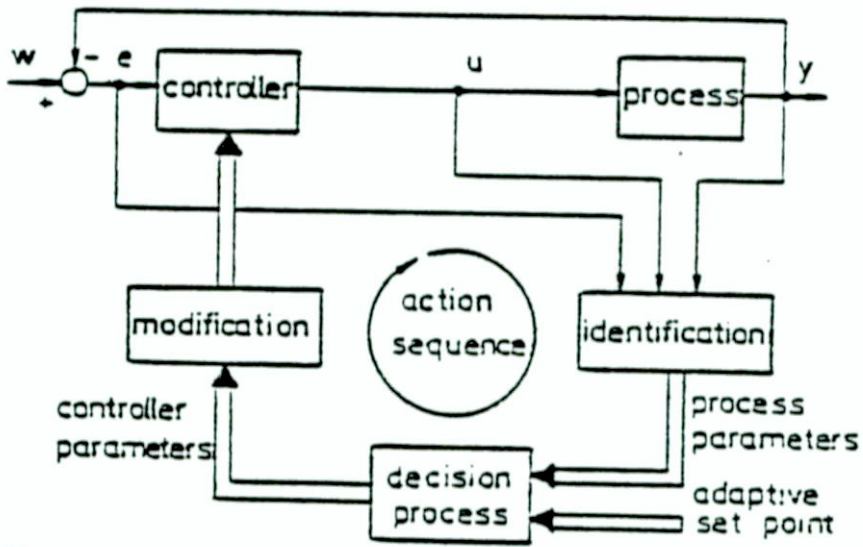


Fig. 1.1

Einen Ausweg bieten die robusten Regelsysteme. Die Idee der robusten Regelung besteht darin, einen konstanten Regler nach gewissen Kriterien so zu entwerfen, dass bei Änderungen der Parameter, das Systemverhalten nicht mehr als einen erlaubten Mass von dem erwünschten Verhalten abweicht. Diese Entwurfsmöglichkeit stellt also "nur" eine suboptimale Regelung dar. Siehe Fig. 1.2.

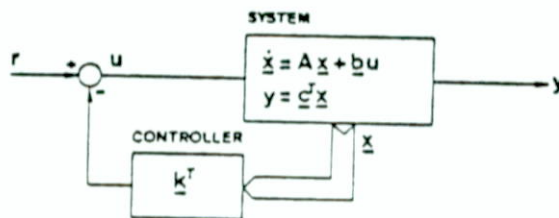


Fig. 1.2

Diese Arbeit befasst sich mit der Synthese von suboptimalen, robusten, linearen Systemen.

## 1.2. Vorgehen

Im nächsten Kapitel sind verschiedene Methoden zusammengefasst worden, die der robusten Reglersynthese dienen. Es war unumgänglich für das Verständnis der Zusammenhänge der angegebenen Methoden einige mathematische Beziehungen zu bringen.

In Kapitel 3 wurde die Methode von K.R.Krishnan und S.Brzezowski näher beschrieben. Sie beruht auf die Idee die Steuerenergie mit einem Parameter  $\lambda$  schrittweise zu erhöhen, bis eine "gewünschte Robustheit" erreicht wird. Diese Methode wird der eigentliche Gegenstand unserer Arbeit sein.

In Kapitel 4 wurde ein Programmpaket angegeben, der sowohl die Berechnung der konstanten Rückführungsmatrix erlaubt, wie auch eine Verkleinerung des Robustheitsgebietes gestattet für den Fall, dass die Steuerenergie einen unzulässig grossen Wert erreicht.

In Kapitel 5 wurden schliesslich mit Hilfe des erhaltenen Programmes ROBUST die Ergebnisse des in [7] angegebenen Beispielles bestätigt und darüber hinaus das Programm an zwei weiteren Systeme :

- a) ein elektromechanisches System
- b) eine hydraulische Werkzeugmaschine

angewandt. Jedesmal wurde das Verhalten des suboptimalen robusten Systems durch Systemsimulation mit dem Verhalten des offenen, sowie auch des optimalen Systems miteinander verglichen.

In Kapitel 6 wurde schliesslich eine Beurteilung der Methode [7] vorgenommen und auf einige ungelöste Probleme hingewiesen.

Wir möchten auch diesmal an dieser Stelle die Gelegenheit benützen Herrn Prof. Dr. M. Mansour und dem ganzen Institut für Automatik zu danken, dass wir diese Arbeit durchführen konnten. Unser Dank gilt besonders den Herren M. Djordjevic und F. Wiederkehr für ihre grosse Hilfsbereitschaft und tatkräftige Unterstützung.

Zürich, den 22. Dezember 1980

Selim Hacısalihzade

Luis von Achenbach

## 2. EINIGE METHODEN FÜR DEN ENTWURF VON ROBUSTEN REGELSYSTEMEN

### 2.1. Einleitung

Eine sehr häufig verwendete Methode für den Reglerentwurf, ist die der Polfestlegung. Durch die Rückführung aller Zustandsgrößen des Prozesses können die Pole des geschlossenen Systems beliebig festgelegt werden (falls das System steuer- und beobachtbar ist), was gegenüber traditionellen P-, PI-, und PID-Reglern – die i.a. nur die Lage der dominanten Pole beeinflussen können – wesentliche Vorteile bringen kann. Man wählt also eine Polkonfiguration und simuliert dann das geregelte System. Ist der Verlauf der Zustände nicht zufriedenstellend, so werden die Pole verschoben und das System wird nochmals simuliert. Dieser Vorgang muss so lange wiederholt werden, bis zum Erreichen der gewünschten Zustandsverläufe.

Damit ist jedoch die Frage nach der Zustandssensitivität nicht zufriedenstellend gelöst, da die Polverschiebung in sich, keine geeignete Beschreibung für die aktuellen Zustandsänderungen liefert. In unserer Semesterarbeit (AIE 8398) haben wir bei der Regelung verschiedener Systeme u.a. diese Methode der Polfestlegung verwendet, um nach dem von Itschner [1] angegebenen Verfahren, Stabilitätsgebiete im Parameterraum zu finden. Die Abhängigkeit der Systemmatrix von den Parametern konnte beliebig sein, die gefundenen Gebiete waren jedoch nur Ausschnitte des grösst-möglichen Stabilitätsgebietes. Es sei hier nur am Rande erwähnt, dass Algorithmen existieren [2], die es erlauben das grösste Polyheder im Parameter-raum zu finden, innerhalb dessen das System stabil bleibt, mit der Einschränkung jedoch, dass die Elemente der Systemmatrix multilineare Funktionen der Parameter sind. Dieses Verfahren ist aber relativ anspruchsvoll.

In diesem Kapitel geht es darum, verschiedene andere Methoden darzustellen, die der robusten Regler-Synthese dienen, die

also die Zustandssensitivität reduzieren. Die Zustandssensitivität gibt ein Mass für die Abweichung der Trajektorie von der optimalen Trajektorie, bei Veränderung der Parameter eines gegebenen Systems und bei gleichen Anfangsbedingungen.

## 2.2. Mathematische Grundlagen der Sensitivitätstheorie

### 2.2.1 Herleitung des Begriffs der Sensitivität

Gegeben sei eine beliebige Funktion  $F(x)$ , deren sämtliche Ableitungen  $F^{(k)}(x_0)$  an der Stelle  $x_0$  existieren. Der Wert dieser Funktion an der Stelle  $x_0 + \Delta x$  gibt uns die Taylor-Reihe

$$\begin{aligned} F(x_0 + \Delta x) &= F(x_0) + \frac{\Delta x}{1!} F'(x_0) + \dots + \frac{(\Delta x)^n}{n!} F^{(n)}(x_0) + \dots \\ &= F(x_0) + \Delta F(x_0) \end{aligned}$$

mit

$$\Delta F(x_0) = \sum_{i=1}^{\infty} \frac{(\Delta x)^i}{i!} F^{(i)}(x_0)$$

Ist die Aenderung  $\Delta x$  sehr klein, so können die Glieder höherer Ordnung vernachlässigt werden und wir schreiben im folgenden  $dx$  statt  $\Delta x$  und  $dF$  statt  $\Delta F$ . Dividieren wir nun beide Seiten des letzten Ausdruckes durch  $F(x_0)$ , so erhalten wir die relative Aenderung oder Variation  $V_x^F$  der Funktion  $F(x)$  (da  $x_0$  beliebig ist, schreiben wir  $x$  anstelle von  $x_0$ )

$$\frac{dF(x)}{F(x)} = V_x^F = \frac{F'(x) dx}{F(x)} = \frac{F'(x) dx}{F(x)} \cdot \frac{x}{x} = S_x^F \cdot \frac{dx}{x}$$

Die Grösse  $S_x^F$  heisst relative Sensitivität der Funktion  $F$  bezüglich des Parameters  $x$ . Sie gibt die relative Aenderung von  $F(x)$  in Funktion der relativen Aenderung von  $x$  an :

$$S_x^F = \frac{dF}{dx} \cdot \frac{x}{F(x)} = \frac{dF(x)/F(x)}{dx/x}$$

Ist die Funktion  $F$  von mehreren Parametern abhängig, so muss zur Bestimmung der relativen Sensitivität von  $F$  bezüglich irgend eines  $x_i$  die partielle Ableitung gebildet werden

$$S_{x_i}^F(x_1, x_2, \dots, x_n) = \frac{\partial F}{\partial x_i} \cdot \frac{x_i}{F}$$

Die Variation von  $F$  für den Fall, dass alle Parameter  $x_i$  sich ändern, erhalten wir dann als Summe der einzelnen Variationen

$$\frac{dF}{F} = \sum_{i=1}^n V_{x_i}^F = \sum_{i=1}^n S_{x_i}^F \cdot \frac{dx_i}{x_i}$$

## 2.2.2 Anwendung auf die Theorie der linearen Regelsysteme

### A) Empfindlichkeit im Bildbereich

Es sei die Uebertragungsfunktion  $G(s)$  eines Parameterabhängigen Systems gegeben. Die Sensitivität gibt ein Mass für die Aenderung von  $G(s)$ , falls einer der Parameter von seinem Nominalwert abweicht.

Die Sensitivität der Uebertragungsfunktion  $G(s, \alpha)$  bezüglich  $\alpha$  an der Stelle  $\alpha = \alpha_{\text{nom}}$  ist gegeben durch

$$S_{\alpha_{\text{nom}}}^{G(s, \alpha)} = \lim_{\alpha \rightarrow \alpha_{\text{nom}}} \frac{\frac{G(s, \alpha) - G(s, \alpha_{\text{nom}})}{G(s, \alpha_{\text{nom}})}}{\frac{\alpha - \alpha_{\text{nom}}}{\alpha_{\text{nom}}}} = \frac{\partial G(s, \alpha)}{\partial \alpha} \Big|_{\alpha_{\text{nom}}} \cdot \frac{\alpha_{\text{nom}}}{G(s, \alpha_{\text{nom}})}$$

B) Empfindlichkeit im Zeitbereich

Es sei das folgende SISO-System n-ter Ordnung gegeben

$$\begin{aligned} \dot{\underline{x}} &= A\underline{x} + \underline{b}u & \text{mit } \underline{x}(0) &= \underline{x}_0 \\ y &= \underline{c}'\underline{x} \end{aligned}$$

welches einen veränderlichen Parameter  $\alpha$  enthalte. Das Sensitivitätsmodell erhält man durch die Ableitung des gegebenen Systems nach  $\alpha$

$$\begin{aligned} \underline{\sigma} &= A\underline{\sigma} + \frac{\partial A}{\partial \alpha} \underline{x} + \frac{\partial \underline{b}}{\partial \alpha} u \\ w &= \underline{c}'\underline{\sigma} + \frac{\partial \underline{c}'}{\partial \alpha} \underline{x} \end{aligned}$$

wobei

$$\underline{\sigma} = \frac{\partial \underline{x}}{\partial \alpha} \quad \text{und} \quad w = \frac{\partial y}{\partial \alpha}$$

## 2.3. Methoden mit differentiellen Zustandssensitivitätsterme in der Zielfunktion

### 2.3.1 Einleitung

Die Aufgabe einen optimalen Rückführungssteuervektor  $\underline{u}^*(t)$  für das System n-ter Ordnung

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \quad , \quad \underline{x}(0) = \underline{x}_0 \quad (2.1)$$

zu finden, welcher die Zielfunktion

$$J = \int_0^{\infty} (\underline{x}'Q\underline{x} + \underline{u}'R\underline{u})dt \quad (2.2)$$

minimiert, ist wohl bekannt. Was geschieht nun, wenn die A- und B-Matrix Funktionen von einem skalaren, zeitinvarianten<sup>†</sup> Parameter  $\alpha$  sind? Der für einen gewissen Nennwert von  $\alpha$  berechneten Steuervektor  $\underline{u}^*$  ist nicht mehr optimal für  $\alpha \neq \alpha_{\text{nom}}$  und folglich treten hier Abweichungen des Trajektorienverlaufes gegenüber dem erwünschten Verlauf (siehe Einleitung).

Mehrere Forscher haben versucht die Zustandssensitivität bezüglich Systemparameteränderungen zu vermindern, indem sie Algorithmen verwendeten, welche auf die Idee beruhen, einen quadratischen Zustandssensitivitätsterm in den Integrand der Zielfunktion einzusetzen, und eine lineare Rückführung als Ansatz zu nehmen, die aus Zustands- und Zustandssensitivitätsterme besteht.

Die partielle Ableitung von (2.1) nach  $\alpha$  ergibt

$$\dot{\underline{\sigma}} = A_{\alpha}\underline{x} + A\underline{\sigma} + B_{\alpha}\underline{u} + B \frac{\partial \underline{u}}{\partial \alpha} \quad , \quad \underline{\sigma}(0) = 0 \quad (2.3)$$

†) Obwohl die Parameter sich zeitlich ändern, setzt man voraus, dass diese Änderungen langsam geschieht, verglichen mit der Dynamik des Systems.

mit

$$\begin{aligned}\dot{\underline{\sigma}} &= \frac{\partial \underline{x}}{\partial \alpha} : \text{Zustandssensitivitätsvektor} \\ A_{\alpha} &= \frac{\partial A}{\partial \alpha} \\ B_{\alpha} &= \frac{\partial B}{\partial \alpha}\end{aligned}$$

Indem man für die lineare Rückführung den Ansatz

$$\underline{u} = K_1 \underline{x} + K_2 \underline{\sigma} \quad (2.4)$$

macht, so dass

$$\frac{\partial \underline{u}}{\partial \alpha} = K_1 \underline{\sigma} + K_2 \frac{\partial \underline{\sigma}}{\partial \alpha}$$

kann (2.3) neu geschrieben werden als

$$\begin{aligned}\dot{\underline{\sigma}} &= (A_{\alpha} + B_{\alpha} K_1) \underline{x} + (A + BK_1 + B_{\alpha} K_2) \underline{\sigma} + BK_2 \frac{\partial \underline{\sigma}}{\partial \alpha} \\ &\text{mit } \underline{\sigma}(0) = 0\end{aligned} \quad (2.5)$$

Vernachlässigt man den Term mit  $\frac{\partial \underline{\sigma}}{\partial \alpha}$ , so ergibt die Lösung von (2.5) den approximativen Zustandssensitivitätsvektor, den wir mit  $\underline{\rho}(t)$  bezeichnen wollen. Anstatt (2.4) benützen wir nun für den Steuervektor den modifizierten Ansatz

$$\underline{u} = K_1 \underline{x} + K_2 \underline{\rho} \quad (2.6)$$

Das System (2.1) lässt sich nun, mit diesem Ansatz für  $\underline{u}(t)$ , schreiben als

$$\dot{\underline{x}} = (A + BK_1) \underline{x} + BK_2 \underline{\rho}, \quad \underline{x}(0) = \underline{x}_0 \quad (2.7)$$

wobei  $\underline{\rho}$  die Lösung der folgenden Differentialgleichung ist

$$\dot{\underline{\rho}} = (A_{\alpha} + B_{\alpha}K_1) \Big|_{\alpha = \alpha_{\text{nom}}} \underline{x} + (A + BK_1 + B_{\alpha}K_2) \Big|_{\alpha = \alpha_{\text{nom}}} \underline{\rho}, \quad \underline{\rho}(0) = 0 \quad (2.8)$$

Wie schon erwähnt, ist die Lösung von (2.8) nur eine Approximation der Trajektoriensensitivität,  $\underline{\sigma}(t)$ , die gegeben ist durch

$$\dot{\underline{\sigma}} = (A_{\alpha} + B_{\alpha}K_1) \underline{x} + B_{\alpha}K_2 \underline{\rho} + (A + BK_1) \underline{\sigma} + BK_2 \frac{\partial \underline{\rho}}{\partial \alpha} \quad (2.9)$$

mit  $\underline{\sigma}(0) = 0$

wobei

$$\frac{\partial \underline{\rho}}{\partial \alpha} = (A_{\alpha} + B_{\alpha}K_1) \Big|_{\alpha = \alpha_{\text{nom}}} \frac{\partial \underline{\rho}}{\partial \alpha} + (A + BK_1 + B_{\alpha}K_2) \Big|_{\alpha = \alpha_{\text{nom}}} \frac{\partial \underline{\rho}}{\partial \alpha} \quad (2.10)$$

mit  $\frac{\partial \underline{\rho}}{\partial \alpha}(0) = 0$

### 2.3.2 Die Methode von Kreindler

Kreindler schlägt vor [3] das folgende System 2n-ter Ordnung zu betrachten

$$\dot{\underline{v}} = A_1 \underline{v} + B_1 \underline{u}, \quad \underline{v}(0) = \underline{v}_0 \quad (2.11)$$

wobei

$$\underline{v} = \begin{bmatrix} \underline{x}' \\ \underline{\rho}' \end{bmatrix}, \quad A_1 = \begin{bmatrix} A & 0 \\ A_{\alpha} & A + BK_1 \end{bmatrix} \quad \text{und} \quad B_1 = \begin{bmatrix} B \\ B_{\alpha} \end{bmatrix}$$

Der Steuervektor  $\underline{u}$  ist

$$\underline{u} = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \underline{v}$$

Die Zielfunktion (2.2) wird folgendermassen modifiziert

$$J_1 = \int_0^{\infty} (\underline{x}' Q_1 \underline{x} + \underline{p}' Q_2 \underline{p} + \underline{u}' R \underline{u}) dt$$

( $Q_1$  und  $Q_2$  sind  $n \times n$  symmetrische positiv-definite Matrizen).  
 $J_1$  kann umgeschrieben werden mit

$$\bar{Q} = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}$$

zu

$$J_1 = \int_0^{\infty} (\underline{v}' \bar{Q} \underline{v} + \underline{u}' R \underline{u}) dt \quad (2.12)$$

Falls  $A_1$  unabhängig wäre von  $K_1$ , so könnte der Steuervektor, der die Zielfunktion (2.12) minimiert, für irgend eine Anfangsbedingung angegeben werden, mit Hilfe der klassischen linearen Regelsystemtheorie, als

$$\underline{u} = K \underline{v} = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \underline{v}$$

mit

$$K = -R^{-1} B_1' P \quad (2.13)$$

wobei  $P$  die symmetrische positiv-definite Lösung der Matrix-Riccati-Gleichung

$$P A_1 + A_1' P - P B_1 R^{-1} B_1' P + \bar{Q} = 0 \quad (2.14)$$

ist.

Durch diesen Tatbestand motiviert, wählt Kreindler für sei-

ne Rückführungsmatrizen  $K_1$  und  $K_2$  diejenigen Werte, welche die Gleichungen (2.13) und (2.14) erfüllen, unbekümmert der Tatsache, dass da  $A_1$  von  $K_1$  abhängig ist, solch eine Wahl von  $K_1$  und  $K_2$  i.a.  $J_1$  nicht minimiert. Trotzdem scheint die Methode ziemlich erfolgreich zu sein, um die Zustandssensitivitäten zu reduzieren. Ausserdem hat sie den Vorteil, dass der Entwurf unabhängig ist von der Wahl der Anfangsbedingungen.

### 2.3.3 Die Methode von Rao und Soudack

Diese Methode [4] erlaubt eine bessere Approximation von  $\underline{g}$  durch  $\underline{p}$ . Dafür wird das Trajektoriensensitivitätsmodell, gegeben in (2.8), ersetzt durch

$$\dot{\underline{p}} = C\underline{x} + D\underline{p} \quad , \quad \underline{p}(0) = 0 \quad (2.15)$$

wobei

$$\frac{d\underline{p}}{d\alpha} = C\underline{g} + D\frac{d\underline{p}}{d\alpha} \quad , \quad \frac{d\underline{p}}{d\alpha}(0) = 0$$

Das System 2n-ter Ordnung wird zu

$$\dot{\underline{v}} = A_1\underline{v} + B_1\underline{u} \quad , \quad \underline{v}(0) = \underline{v}_0 \quad (2.16)$$

mit

$$A_1 = \begin{bmatrix} A & O \\ C & D \end{bmatrix} \quad \text{und} \quad B_1 = \begin{bmatrix} B \\ O \end{bmatrix}$$

Für irgend eine anfängliche Wahl von C und D wird nun die Zielfunktion  $J_1$  (2.12) mit dem K aus den Gleichungen (2.13) und (2.14) minimiert, wobei  $A_1$  durch (2.16) gegeben ist. Für dieses optimale K werden nun die Parameter in C und D modifiziert

indem man den Wert von

$$J_2 = \int_0^{\infty} \|\underline{\rho}(t) - \underline{\sigma}(t)\|_e^2 dt$$

mit dem Gradientenverfahren verkleinert. Die neuen Matrizen C und D eingesetzt in  $A_1$  erlauben nun eine neue Berechnung von K. Dies wird so lange fortgesetzt, bis  $J_2$  einen gewünschten Wert erreicht hat.

Der Nachteil dieser Methode ist, dass die Iteration nicht immer konvergiert und dass die Berechnungen von dem Anfangszustandsvektor  $\underline{x}_0$  abhängen.

#### 2.3.4 Die Methode von Fleming und Newmann

Diese Methode [5] verwendet die Gleichungen (2.8) bis (2.10) in einer anderen Art, um ein System  $4n$ -ter Ordnung darzustellen, in welchem  $K_1$  und  $K_2$  so gewählt werden, dass die Zielfunktion

$$J_3 = \int_0^{\infty} (\underline{x}'Q_1\underline{x} + \underline{\sigma}'Q_2\underline{\sigma} + \underline{u}'R\underline{u})dt \quad (2.17)$$

minimiert wird (für  $\alpha = \alpha_{nom}$ ). Wie man sieht wird hier die exakte Trajektoriensensitivität,  $\underline{\sigma}(t)$ , in die Zielfunktion eingesetzt, es wird jedoch aus praktischen Gründen weiterhin mit  $\underline{\rho}(t)$  in Gleichung (2.6) gearbeitet.

Die neue Systemgleichung ist

$$\dot{\underline{v}} = \bar{A}\underline{v} \quad , \quad \underline{v}(0) = \underline{v}_0 \quad (2.18)$$

wobei

$$\underline{v} = \left[ \underline{x}' \quad \underline{\rho}' \quad \underline{\sigma}' \quad \frac{\partial \underline{\rho}'}{\partial \alpha} \right]'$$

und

$$\bar{A} = \begin{bmatrix} A + BK_1 & BK_2 & 0 & 0 \\ A_\alpha + B_\alpha K_1 & A + BK_1 + B_\alpha K_2 & 0 & 0 \\ A_\alpha + B_\alpha K_1 & B_\alpha K_2 & A + BK_1 & BK_2 \\ 0 & 0 & A_\alpha + B_\alpha K_1 & A + BK_1 + B_\alpha K_2 \end{bmatrix}$$

Die Zielfunktion (2.17) kann nun neu geschrieben werden als

$$J_3 = \int_0^{\infty} \underline{v}' (\bar{Q} + E_1' K' R K E_1) \underline{v} dt \quad (2.19)$$

mit

$$\bar{Q} = \begin{bmatrix} Q_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & Q_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad E_1 = \begin{bmatrix} I_{2n} & 0_{2n} \end{bmatrix}, \quad K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$$

Die Aufgabe besteht nun darin, ein  $K$  zu wählen, welches die Zielfunktion (2.19) minimiert für das gegebene System (2.18). Es werden verschiedene Methoden angegeben, um  $K$  zu berechnen, es sei jedoch in dieser kurzen Uebersicht auf deren Beschreibung verzichtet.

Nachteil dieser Methode ist, dass sie auch von den Anfangsbedingungen abhängt, sie erreicht jedoch eine gute Sensitivitätsverminderung und hat noch den Vorteil, dass sie leicht modifizierbar ist für den Fall, dass nicht alle Zustände messbar sind.

#### 2.4. Die Methode der maximal zulässigen Systemmatrixänderungen

Die Autoren Toda, Patel und Sridhar [6] haben einige neue Erkenntnisse gebracht, bezüglich der Stabilität eines optimalen Regelsystemes bei Modellierungsfehler und/oder Systemparameteränderungen. Diese Ergebnisse drücken sich als Einschränkungen für die Änderungen in der Systemmatrix aus. Eine Beziehung ist gegeben zwischen dem zugelassenen Fehler bzw. Änderungen in der Systemmatrix und deren dominanten Eigenwerte.

Betrachten wir wiederum den optimalen Zustandsrückführungsentwurf für ein lineares, zeitinvariantes System gegeben durch

$$\dot{\underline{x}} = A\underline{x} + B\underline{u}$$

mit der zu minimierenden quadratischen Zielfunktion

$$J = \int_0^{\infty} (\underline{x}'Q\underline{x} + \underline{u}'R\underline{u}) dt \quad , \quad Q > 0 \text{ und } R > 0$$

Es wird angenommen, dass das System steuerbar ist. Wenn die Systemparameter entweder unbekannt sind, oder sich zeitlich ändern, schlagen die Autoren folgendes neue Modell vor

$$\dot{\underline{x}} = A_m\underline{x} + B\underline{u}$$

wobei

$$A_m = A + E$$

E ist der Modellierungsfehler, oder die Änderung der Systemmatrix A bei Parametervariationen.

Die Minimierung der Zielfunktion  $J$  mit diesem neuen Systemmodell liefert

$$\underline{u} = -R^{-1}B'P \underline{x}$$

wobei  $P$  die Lösung der algebraischen Matrix-Riccati-Gleichung

$$A_m'P + PA_m - PBR^{-1}B'P + Q = 0$$

ist.

Das geschlossene System

$$\dot{\underline{x}} = (A - BR^{-1}B'P) \underline{x} \quad (2.20)$$

ist sicher stabil, falls  $E = 0$ , d.h. wenn es keine Modellierungsfehler, bzw. keine Systemparameteränderungen gibt. Da dies meistens nicht der Fall ist, ist man daran interessiert zu wissen, wie gross  $E$  sein darf ohne die Stabilität des geschlossenen Kreises (2.20) zu gefährden.

Die Autoren formulieren diese Bedingung für  $E$  in einem Satz der besagt, dass wenn

$$\|E\|_s \leq \frac{1}{2\|P\|_s\|T^{-1}\|_s} = \frac{\text{Min } \lambda(T)}{2\text{Min } \lambda(P)} \quad (2.21)$$

dann ist das System (2.20) stabil.

$\|\cdot\|_s$  : spektrale Norm

$\lambda(\cdot)$  : Eigenwert

$$T \equiv 2Q + A_m'P + PA_m$$

Dieser Satz wird durch zwei Korollare ergänzt. (Siehe die Ungleichungen (2.22) und (2.23)).

$$\|E\|_e \leq \frac{1}{2\|P\|_s \|T^{-1}\|_s} \quad (2.22)$$

wobei

$$\|E\|_e = \left\{ \sum_{i=1}^n \sum_{j=1}^n |e_{ij}|^2 \right\}^{1/2}$$

die Euklidische Norm der Matrix E ist und deren (i,j)-tes Element mit  $e_{ij}$  bezeichnet wird.

Diese Aussage gibt eine gesamthafte Einschränkung für die Systemmatrixänderungen E. Man kann aber auch die Bedingung elementenweise formulieren

$$|e_{ij}| \leq \epsilon = \frac{1}{2n} \frac{1}{\|P\|_s \|T^{-1}\|_s} \quad \forall i, j=1, \dots, n \quad (2.23)$$

wobei n die Systemordnung ist. Sind (2.22) und/oder (2.23) erfüllt, dann ist das geschlossene System (2.20) stabil.

Die hier gemachten Aussagen und formulierten Einschränkungen für E stellen in sich keine eigentliche Entwurfsmethode dar, sie können jedoch eine grosse Hilfe bei der Wahl der Polfestlegung sein. Falls die erwünschte Grösse des Stabilitätsgebietes im Parameterraum nicht erreicht ist, kann man natürlich die Gewichtungsmatrix Q grösser wählen. In Kapitel 3 wird über diese Möglichkeit ausführlich berichtet.

### 3. DIE METHODE VON KRISHNAN UND BRZEZOWSKI

Die beiden Autoren K.R.Krishnan und S.Brzezowski geben in [7] eine neue Methode an für den Entwurf von robusten linearen Reglern mit vorgegebener Zustandsinsensitivität bezüglich Parameteränderungen im System, die hier kurz zusammengefasst wird.

#### 3.1. Einleitung

Gegeben sei ein lineares, zeitinvariantes System, dessen Parameter sich ändern können, innerhalb eines bekannten Gebietes. Die im folgenden skizzierte Entwurfsmethode führt zur Berechnung einer konstanten Rückführungsmatrix, welche eine vorgegebene Insensitivität des geschlossenen Systemzustandsverlaufes bezüglich Parameteränderungen bewirkt. Der Unterschied dieser Methode gegenüber den angegebenen in Kapitel 2 besteht darin, dass sie die tatsächlichen Zustandsänderungen für ein gegebenes Parametervariationsgebiet vor Augen hat, anstatt mit differentiellen Sensitivitäten zu arbeiten. Zudem eignen sich jene Methoden nicht für grosse Parameteränderungen.

Die Frage nach der Sensitivität wird hier gelöst, indem man sie als eine klassische Optimierungsaufgabe mit quadratischer Zielfunktion und einer Nebenbedingung stellt.

#### 3.2. Beschreibung der Methode

Das gegebene lineare, zeitinvariante System n-ter Ordnung sei

$$\dot{\underline{x}} = A(\underline{\alpha})\underline{x} + B(\underline{\alpha})\underline{u} \quad , \quad \underline{\alpha} \in \Omega \quad (3.1)$$

dessen Parameter Werte innerhalb eines gegebenen Gebietes  $\Omega$  im Parameterraum annehmen dürfen, wobei

- $\underline{x}$  : n-dimensionaler Zustandsvektor
- $\underline{u}$  : r-dimensionaler Steuervektor
- $\underline{\alpha}$  : m-dimensionaler Parametervektor
- $A(\underline{\alpha})$  : (nxn)-Systemmatrix
- $B(\underline{\alpha})$  : (nxr)-Steuermatrix

Gesucht ist eine konstante (rxn) Rückführungsmatrix L, der Form

$$\underline{u} = -L\underline{x} \tag{3.2}$$

so dass die resultierende Antwort des geschlossenen Systems, bis zu einem gewissen Ausmass, insensitiv bezüglich Parameteränderungen in  $\Omega$  ist. Das Insensitivitätskriterium wird mit Hilfe einer Integralbedingung formuliert

$$\begin{aligned} \text{Max}_{\underline{\alpha} \in \Omega} \left\{ \int_0^{\infty} \underline{x}' Q \underline{x} dt \right\} &\leq \rho \|\underline{x}(0)\|^2 \\ \|\underline{x}(0)\| &= \text{const} \end{aligned} \tag{3.3}$$

wobei

- Q : gegebene (nxn) positiv definite oder semidefinite Gewichtungsmatrix
- $\rho$  : gegebener positiver Wert, als Mass für die erlaubte Trajektorienänderung
- $\|\underline{x}(0)\|$  : Norm der Anfangsbedingungen

Die zu minimierende Zielfunktion ist

$$\int_0^{\infty} \underline{u}' R \underline{u} dt \tag{3.4}$$

für  $\underline{\alpha} = \underline{\alpha}_{\text{nom}}$ , wobei R eine gegebene (rxr) positiv-definite Gewichtungsmatrix darstellt.

Unter Berücksichtigung der Nebenbedingung (3.3) kann für

$\underline{\alpha} = \underline{\alpha}_{\text{nom}}$  die modifizierte Zielfunktion gebildet werden

$$J(\underline{\alpha}_{\text{nom}}, \lambda) = \int_0^{\infty} \{ \lambda(\underline{x}' Q \underline{x}) + \underline{u}' R \underline{u} \} dt \quad (3.5)$$

Betrachten wir nun das System (3.1) bei  $\underline{\alpha} = \underline{\alpha}_{\text{nom}}$

$$\dot{\underline{x}} = \bar{A} \underline{x} + \bar{B} \underline{u} \quad (3.6)$$

mit

$$\bar{A} = A(\underline{\alpha}_{\text{nom}}) \quad \text{und} \quad \bar{B} = B(\underline{\alpha}_{\text{nom}})$$

Die klassische Optimierungstheorie liefert uns die Rückführungsmatrix L für das System (3.6) mit der Zielfunktion (3.5)

$$L(\underline{\alpha}_{\text{nom}}, \lambda) = R^{-1} \bar{B}' K(\underline{\alpha}_{\text{nom}}, \lambda) \quad (3.7)$$

wobei K die (nxn) positiv-definite Lösung der Matrix-Riccati-Gleichung

$$K \bar{A} + \bar{A}' K + \lambda Q - K \bar{B} R^{-1} \bar{B}' K = 0 \quad (3.8)$$

ist.

Für das geschlossene System resultiert

$$\dot{\underline{x}} = F(\underline{\alpha}, \underline{\alpha}_{\text{nom}}, \lambda) \underline{x} \quad , \quad \underline{\alpha} \in \Omega \quad (3.9)$$

mit

$$\begin{aligned} F(\underline{\alpha}, \underline{\alpha}_{\text{nom}}, \lambda) &= A(\underline{\alpha}) - B(\underline{\alpha}) L(\underline{\alpha}_{\text{nom}}, \lambda) \\ &= A(\underline{\alpha}) - B(\underline{\alpha}) R^{-1} \bar{B}' K(\underline{\alpha}_{\text{nom}}, \lambda) \end{aligned} \quad (3.10)$$

Siehe dazu Fig. 3.1

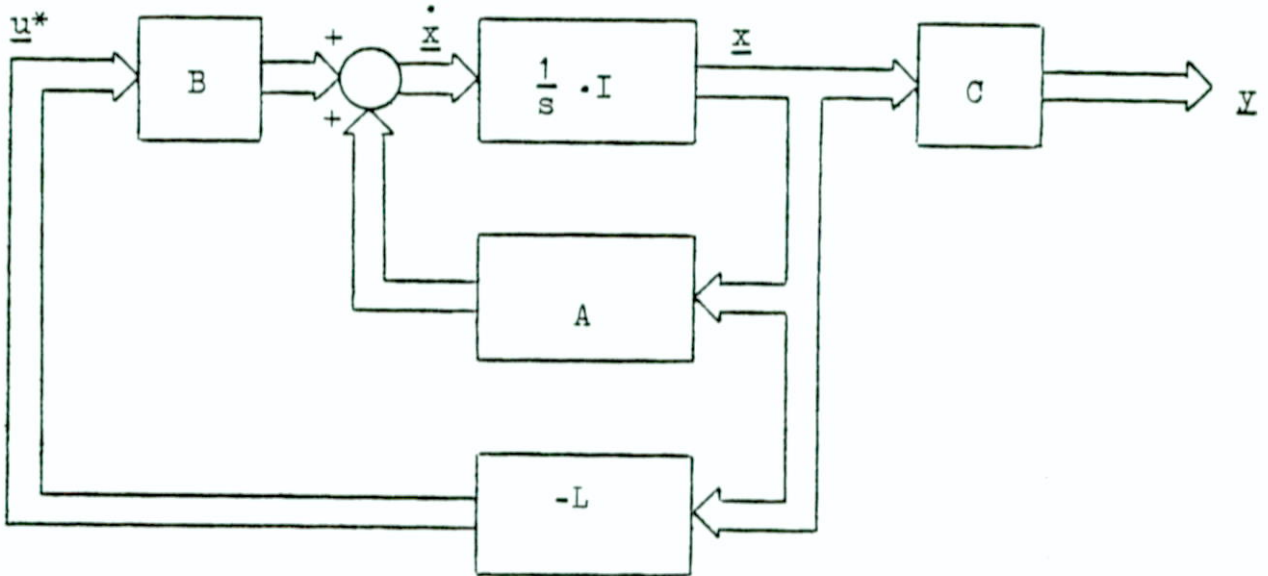


Fig. 3.1

Für  $\underline{\alpha}_{nom}$  ist die Matrix  $F$  bekanntlich immer stabil. Es wird angenommen, dass ein  $\underline{\alpha}_{nom} \in \Omega$  existiert für welchen ein  $\lambda$  gefunden werden kann, so dass  $F(\underline{\alpha}, \underline{\alpha}_{nom}, \lambda)$  für alle  $\underline{\alpha} \in \Omega$  stabil ist. Siehe [2].

Die Frage die noch offen geblieben ist, ist, ob das in (3.3) angegebene Insensitivitätskriterium für alle  $\underline{\alpha} \in \Omega$  erfüllt wird. Im folgenden wird (3.3) zu einer einfacheren Beziehung umgeformt.

Sei

$$V(\underline{x}) = \underline{x}' P \underline{x}$$

eine positiv-definite quadratische Form. In Abschnitt 3.5.,

bei der Berechnung der Zielfunktion wird bewiesen, dass wenn P die Lösung der Ljapunov-Matrix-Gleichung

$$F'P + PF = -Q \quad (3.11)$$

ist, dann gilt

$$\int_0^{\infty} \underline{x}' Q \underline{x} \, dt = \underline{x}'(0) P \underline{x}(0) \quad (3.12)$$

Die Ungleichung

$$\underline{x}'(0) P \underline{x}(0) \leq \|P\|_s \|\underline{x}(0)\|^2 = \mu_{\max}(P) \|\underline{x}(0)\|^2$$

wobei  $\mu_{\max}(P)$  den grössten Eigenwert der positiv-definiten Matrix P bezeichnet, führt zu

$$\underset{\underline{\alpha} \in \Omega}{\text{Max}} \{ \mu_{\max}(P) \} \leq \rho \quad (3.13)$$

(3.13) stellt eine hinreichende Bedingung für die Erfüllung der Ungleichung (3.3) dar.

### 3.3. Der Entwurfsalgorithmus

- 1) Die Nominalwerte der Parameter sind entweder gegeben oder werden sonst ausgesucht ( $\underline{\alpha}_{\text{nom}} \in \Omega$ ).
- 2) Einen Anfangswert für  $\lambda$  wird gewählt.
- 3) Die Matrix  $K(\underline{\alpha}_{\text{nom}}, \lambda)$  wird aus (3.8) berechnet.
- 4) Aus (3.10) erhält man die Matrix  $F(\underline{\alpha}, \underline{\alpha}_{\text{nom}}, \lambda)$  des geschlossenen Systems, die nach Stabilität geprüft wird für alle  $\underline{\alpha} \in \Omega$ . Sollte F instabil sein für ein gewisses  $\underline{\alpha}$ , so wird  $\lambda$  erhöht und zu Punkt 3) zurück gegangen. Falls kein  $\lambda$  bis zu einem gegebenen

Maximalwert  $\lambda_{\max}$  den Test der Stabilität erfüllt, dann muss ein anderes  $\underline{\alpha}_{\text{nom}} \in \Omega$  gewählt werden und es wird zu Punkt 2) zurückgesprungen. Kann kein  $\underline{\alpha}_{\text{nom}} \in \Omega$  und  $\lambda$  gefunden werden für die  $F$  stabil ist für alle  $\underline{\alpha} \in \Omega$ , so muss gesagt werden, dass dieser Algorithmus untauglich für dieses spezielle Problem ist.

- 5) In der Annahme, dass 4) erfolgreich war, wird nun  $P(\underline{\alpha}, \underline{\alpha}_{\text{nom}}, \lambda)$  aus (3.11) berechnet. Schliesslich muss noch die Ungleichung (3.13) getestet werden. Ist (3.13) nicht erfüllt, dann wird  $\lambda$  erhöht und zu Punkt 3) gesprungen.

In [7] wird anschliessend am Beispiel eines Flugzeuges die beschriebene Methode angewendet. Darüber wird ausführlich in Kapitel 5 berichtet, wo die Ergebnisse von Krishnan und Brzezowski u.a. verifiziert werden.

### 3.4. Zusammenfassung

Für das gegebene parameterabhängige System (3.1) wurde eine Methode vorgestellt, welche die Berechnung einer konstanten und linearen Rückführungsmatrix L erlaubt, so dass die Zielfunktion (3.4) minimal wird für  $\underline{\alpha} = \underline{\alpha}_{\text{nom}}$ , unter der Nebenbedingung (3.3). Diese Nebenbedingung wurde zu der für praktische Zwecke einfacheren Beziehung (3.13) umgeformt.

Die Berechnung von L setzt voraus die Kenntniss einer skalaren Grösse  $\lambda_{\text{krit}}$  - die Elemente von L sind Funktionen von  $\underline{\alpha}_{\text{nom}}$  und  $\lambda_{\text{krit}}$  -, so dass der eigentliche Entwurfsalgorithmus auf das Suchen von  $\lambda_{\text{krit}}$  beruht.  $\lambda_{\text{krit}}$  ist der kleinste  $\lambda$ -Wert für welchen die Steuerenergie (3.4) minimal wird für  $\underline{\alpha} = \underline{\alpha}_{\text{nom}}$ , unter der Nebenbedingung (3.3) bzw. (3.13).

### 3.5. Einige Vereinfachungen und Erläuterungen zur beschriebenen Methode

A) In punkt 4) des Abschnittes 3.3. wird geprüft, ob die Eigenwerte der Matrix F des geschlossenen Systems in der linken Halbebene liegen - damit F stabil - und in Punkt 5) werden anschliessend die Eigenwerte der Matrix P berechnet, um sie dem Test (3.13) zu unterziehen. Aus dem folgendem bekannten Satz der linearen Regelsysteme wird sofort ersichtlich, dass es genügt nur die Eigenwerte von P zu berechnen, die nun die doppelte Ungleichung

$$0 < \mu_i(P) \leq \rho \quad , \quad i = 1, \dots, n \quad (3.14)$$

erfüllen müssen, wobei hiermit auch automatisch F nach Stabilität untersucht wird. (Linke Seite der Ungleichung).

SATZ

Der Ursprung des linearen Systems

$$\dot{\underline{x}} = F\underline{x}$$

ist global asymptotisch stabil dann und nur dann, wenn für eine beliebige gewählte symmetrische positiv-definite Matrix  $Q$  die Lösung  $P$  der Gleichung

$$F'P + PF = -Q$$

positiv-definit ist. Siehe [8] .

Es sei hier bemerkt, dass obige Ljapunov-Gleichung dann und nur dann eine eindeutige Lösung  $P$  hat, wenn  $F$  keinen symmetrischen Eigenwertpaar besitzt. Dies ist natürlich immer erfüllt, falls  $F$  stabil ist. Hat jedoch  $F$  mindestens einen symmetrischen Eigenwertpaar, dann können zwei Fälle auftreten, je nach der Wahl von  $Q$  : entweder ist die Ljapunov-Gleichung widersprüchlich, oder sie besitzt unendlich viele Lösungen gegeben durch

$$P = P_0 + P_1$$

wobei  $P_0$  eine partikuläre Lösung der Ljapunov-Gleichung darstellt und  $P_1$  die allgemeine Lösung der homogenen Gleichung (d.h. wenn  $Q = 0$ ) ist [9] .

B) Im beschriebenen Algorithmus, siehe Abschnitt 3.3., wird  $\lambda$  erhöht, falls  $F$  instabil bzw. die Eigenwerte von  $P$  den Test (3.13) für irgend ein  $\underline{\alpha} \in \Omega$  nicht erfüllen. Dies führte uns zu dem Gedanken, dass für einen genügend grossen  $\lambda$ -Wert beide Bedingungen immer erfüllt werden könnten was wir auch praktisch in den untersuchten Beispielen bestätigt haben. Weiter-

hin wird in dem Artikel von Krishnan und Brzezowski implizit angenommen, dass, wenn die Bedingungen für ein gewisses  $\lambda_{\text{krit}}$  erfüllt werden, dann werden sie auch für jeden grösseren  $\lambda$ -Wert erfüllt und auch dies konnten wir in der Praxis an allen behandelten Beispielen bestätigen. Es wird im folgenden ein Versuch vorgenommen, diese Hypothese analytisch zu beweisen :

Voraussetzung :  $F(\underline{\alpha}, \underline{\alpha}_{\text{nom}}, \lambda)$  ist stabil  $\forall \underline{\alpha} \in \Omega$

Behauptung :  $F(\underline{\alpha}, \underline{\alpha}_{\text{nom}}, \lambda^*)$  ist stabil  $\forall \underline{\alpha} \in \Omega$ , falls  $\lambda^* > \lambda$

Beweis :

Gegeben seien

$$\text{System} \quad : \quad \dot{\underline{x}} = A(\underline{\alpha})\underline{x} + B(\underline{\alpha})\underline{u}$$

$$\text{Zielfunktion} : \quad J(\underline{\alpha}_{\text{nom}}, \lambda) = \int_0^{\infty} \{ \lambda(\underline{x}'Q\underline{x}) + \underline{u}'R\underline{u} \} dt$$

Die Matrix des geschlossenen Systems

$$F(\underline{\alpha}, \underline{\alpha}_{\text{nom}}, \lambda) = A(\underline{\alpha}) - B(\underline{\alpha})R^{-1}\bar{B}'K(\underline{\alpha}_{\text{nom}}, \lambda)$$

ist nach Voraussetzung stabil, wobei  $K(\underline{\alpha}_{\text{nom}}, \lambda)$  die Lösung der Matrix-Riccati-Gleichung

$$K\bar{A} + \bar{A}'K + \lambda Q - K\bar{B}R^{-1}\bar{B}'K = 0$$

ist. (Es sei in Erinnerung gerufen, dass  $\bar{A} = A(\underline{\alpha}_{\text{nom}})$  und  $\bar{B} = B(\underline{\alpha}_{\text{nom}})$  ist).

Es werde nun  $\lambda$  um ein positives  $\delta\lambda$  erhöht

$$\lambda^* = \lambda + \delta\lambda \quad , \quad \delta\lambda > 0$$

Die sich daraus ergebende neue Matrix  $K(\underline{\alpha}_{nom}, \lambda^*)$  wurde mit  $K^*$  bezeichnet

$$K^* = K + \delta K$$

Das Einsetzen von  $K^*$  und  $\lambda^*$  in die Matrix-Riccati-Gleichung ergibt

$$\begin{aligned} (K + \delta K)\bar{A} + \bar{A}'(K + \delta K) + (\lambda + \delta\lambda)Q - \\ - (K + \delta K)\bar{B}R^{-1}\bar{B}'(K + \delta K) = 0 \end{aligned}$$

Die Terme ohne  $\delta K$  und  $\delta\lambda$  ergeben die ursprüngliche Riccati-Gleichung, d.h. sie verschwinden, und unter Vernachlässigung der Terme 2.Ordnung in  $\delta K$  erhält man

$$\delta K\bar{A} + \bar{A}'\delta K + \delta\lambda Q - \delta K\bar{B}R^{-1}\bar{B}'K - K\bar{B}R^{-1}\bar{B}'\delta K = 0$$

Nun wird  $\delta K$  ausgeklammert

$$\delta K(\bar{A} - \bar{B}R^{-1}\bar{B}'K) + (\bar{A}' - K\bar{B}R^{-1}\bar{B}')\delta K = -\delta\lambda Q$$

Mit

$$F(\underline{\alpha}_{nom}, \underline{\alpha}_{nom}, \lambda) = \bar{A} - \bar{B}R^{-1}\bar{B}'K = F_0$$

wird obige Gleichung zu

$$\delta K F_0 + F_0' \delta K = -\delta\lambda Q$$

Da

$$\left. \begin{array}{l} F_0 \text{ stabil} \\ Q > 0 \\ \delta\lambda > 0 \end{array} \right\} \implies \delta K > 0$$

Es sei  $F^*$  die Matrix des geschlossenen Systems bei  $\lambda^*$

$$F^* = A - BR^{-1}\bar{B}'(K + \delta K) = A - BR^{-1}\bar{B}'K - BR^{-1}\bar{B}'\delta K$$

Mit

$$\delta F = -BR^{-1}\bar{B}'\delta K$$

wird

$$F^* = F + \delta F$$

Man bilde nun

$$U = \delta KF^* + F^{*'}\delta K$$

Einsetzen von  $F^*$  und  $\delta F$  ergibt

$$\begin{aligned} U &= \delta KF + \delta K\delta F + F'\delta K + \delta F'\delta K \\ &= \delta KF - \delta KBR^{-1}\bar{B}'\delta K + F'\delta K + \delta K\bar{B}R^{-1}B'\delta K \end{aligned}$$

Und wiederum unter Vernachlässigung der Terme 2. Ordnung in  $\delta K$  erhält man

$$U = \delta KF + F'\delta K$$

Da aus Voraussetzung  $F$  stabil ist und da - wie oben gezeigt wurde -  $K$  positiv-definit ist folgt, dass  $U$  negativ-definit ist und daraus - aus der Definitionsgleichung von  $U$  -, dass  $F^* = F(\underline{\alpha}_{\text{nom}}, \underline{\alpha}_{\text{nom}}, \lambda^*)$  stabil ist. Q.E.D.

Für den Beweis wurde angenommen, dass  $\delta K$  genügend klein ist, so dass die Terme 2.Ordnung in  $\delta K$  vernachlässigt werden dürfen. Da jedoch keine brauchbare Beziehung zwischen  $\delta K$  und  $\delta \lambda$  gefunden werden konnte, ist dieser Beweis von relativer Bedeutung.

C) Weil eine Erhöhung von  $\lambda$  gleichbedeutend ist mit der Erhöhung der Steuergrösse  $\underline{u}(t)$  - was zur Sättigung des Systems führen kann - bzw. der Steuerenergie, siehe (3.5), gibt man ein maximal zulässige  $\lambda_{\text{max}}$  vor. Andererseits bewirkt ein zu kleines  $\lambda$  grosse Abweichungen der Zustandsverläufe gegenüber den vorgegebenen oder erwünschten Verläufen (oder anders gesagt "die Zustände werden langsam"). Dies ist der Grund, weswegen man auch ein minimales  $\lambda_{\text{min}}$  vorgibt. Die Suche nach  $\lambda_{\text{krit}}$  wird also aus physikalischen Gründen im Intervall  $[\lambda_{\text{min}}, \lambda_{\text{max}}]$  vorgenommen.

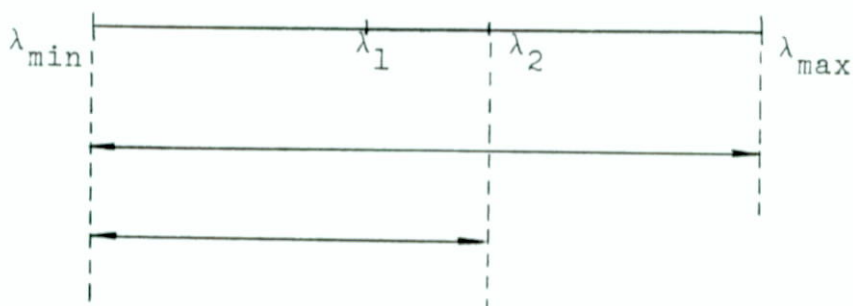
Aus den Erläuterungen in B) können nur die folgenden drei Fälle auftreten :

- a) die Bedingung (3.14) ist für kein  $\lambda \in [\lambda_{\text{min}}, \lambda_{\text{max}}]$  erfüllt
- b) die Bedingung (3.14) ist  $\forall \lambda \in [\lambda_{\text{min}}, \lambda_{\text{max}}]$  erfüllt
- c) die Bedingung (3.14) ist  $\forall \lambda \geq \lambda_{\text{krit}} \in [\lambda_{\text{min}}, \lambda_{\text{max}}]$  erfüllt

Eine systematische Suchmethode für  $\lambda_{\text{krit}} \in [\lambda_{\text{min}}, \lambda_{\text{max}}]$  ist unter diesen Voraussetzungen die "Golden Section" Methode. Voraussetzung nämlich für die Anwendbarkeit von "Golden Section" ist, dass nur ein Minimum im Suchintervall existiert, d.h. für unsere Aufgabe, dass es nur ein  $\lambda_{\text{krit}} \in [\lambda_{\text{min}}, \lambda_{\text{max}}]$  gibt und dies ist erfüllt, da - wie oben gesagt - nur die Fälle a), b) oder c) auftreten können. Im folgenden sei kurz gezeigt, wieso

auf die "Golden Section" Methode jedoch verzichtet wurde.

Dazu betrachte man folgende Skizze, die die Unterteilung des Intervalls nach dem 1. "Golden Section" Schritt zeigt :



Nehmen wir an Test (3.14) sei für  $\lambda_1$  erfolgreich. Der Suchbereich wird auf  $[\lambda_{\min}, \lambda_2]$  reduziert. Laut den oben gemachten Aussagen würde es jedoch genügen, nur im Intervall  $[\lambda_{\min}, \lambda_1]$  weiterzusuchen. "Golden Section" ist also zwar eine zulässige Methode, aber für unsere Aufgabe ungeeignet, weil sie zu langsam ist. (Abgesehen davon, dass man hier keine Zielfunktion minimiert und deswegen auch auf keine zwei Stützpunkte angewiesen ist). Eine schnellere Methode bietet uns die Bisektionsmethode, wo das Suchintervall jeweils halbiert wird.

D) Aus der Unmöglichkeit Punkt 4) und 5) des Entwurfsalgorithmus in 3.3. für *alle*  $\underline{a} \in \Omega$  durchzuführen, gibt man sich eine *endliche* Anzahl  $\underline{a} \in \Omega$  vor. Aus Plausibilitätsgründen und mit dem Einverständnis unserer Betreuer wurden die Ecken des - durch die Grenzen der einzelnen Parameter gebildeten - Hyperquaders gewählt. In [2] sind *hinreichende* Bedingungen erstellt worden, die auf dieses Vorgehen hinweisen, für den Fall der multilinearen Abhängigkeit der Systemmatrixelementen von den Parametern (siehe Kapitel 6).

E) Es wird hier eine analytische Methode angegeben, die es gestattet, die zu minimierende Zielfunktion (3.4) in einfacher Weise zu berechnen.

Gegeben sei das System (3.6)

$$\dot{\underline{x}} = \bar{A}\underline{x} + \bar{B}\underline{u}$$

Setzt man die gefundene suboptimale Regelung (3.2)

$$\underline{u} = -L\underline{x}$$

in (3.6) ein, so bekommt man für den geschlossenen Kreis

$$\dot{\underline{x}} = (\bar{A} - \bar{B}L)\underline{x} = F(\underline{\alpha}_{\text{nom}}, \lambda_{\text{krit}})\underline{x}$$

Die Lösung dieses Differentialgleichungssystem ist wohl bekannt

$$\underline{x} = \underline{x}_0 e^{Ft}$$

In der für  $\underline{\alpha} = \underline{\alpha}_{\text{nom}}$  zu minimierende Zielfunktion (3.4)

$$J = \int_0^{\infty} \underline{u}' R \underline{u} dt$$

werden nun  $\underline{u}$  und  $\underline{x}$  eingesetzt :

$$\begin{aligned} J &= \int_0^{\infty} \underline{u}' R \underline{u} dt = \int_0^{\infty} \underline{x}' L' R L \underline{x} dt \\ &= \underline{x}'_0 \left\{ \int_0^{\infty} e^{F' t} Q^* e^{Ft} dt \right\} \underline{x}'_0 \\ &\stackrel{!}{=} \underline{x}'_0 P^* \underline{x}_0 \end{aligned}$$

mit

$$Q^* = L'RL \quad \text{und} \quad P^* = \int_0^{\infty} e^{F't} Q^* e^{Ft} dt$$

### Berechnung von P\*

Man bilde die Funktion

$$\begin{aligned} & F'P^* + P^*F \\ F'P^* + P^*F &= F' \int_0^{\infty} e^{F't} Q^* e^{Ft} dt + \int_0^{\infty} e^{F't} Q^* e^{Ft} dt F \\ &= \int_0^{\infty} \{ (Fe^{Ft})' Q^* e^{Ft} + e^{F't} Q^* (Fe^{Ft}) \} dt \\ &= \int_0^{\infty} \{ e^{F't} Q^* e^{Ft} \}' dt \\ &= e^{F't} Q^* e^{Ft} \Big|_{\infty} - e^{F't} Q^* e^{Ft} \Big|_0 \end{aligned}$$

Der 1. Term ist gleich 0, wegen der Stabilität von F, der 2. Term ist gleich  $-Q^*$ , so dass

$$\begin{aligned} F'P^* + P^*F &= -Q^* \\ &= -L'RL \end{aligned}$$

Dabei sind  $F(\underline{\alpha}_{\text{nom}}, \lambda_{\text{krit}})$ ,  $L(\underline{\alpha}_{\text{nom}}, \lambda_{\text{krit}})$  und R bekannt. Die Lösung dieser Ljapunov-Matrix-Gleichung ergibt das gesuchte  $P^*$ .

Die Zielfunktion berechnet sich also nach der einfachen  
Vorschrift

$$J = \int_0^{\infty} \underline{u}' R \underline{u} \, dt = \underline{x}_0' P^*(\underline{\alpha}_{\text{nom}}, \lambda_{\text{krit}}) \underline{x}_0$$

anstelle der mühseligen Auswertung des Integrals mittels Simulation. Ganz analog folgt die Herleitung von (3.12). (Für einen leicht modifizierten Beweis siehe [10] ).

#### 4. PROGRAMMIERUNG DER VON KRISHNAN UND BRZEZOWSKI ANGEGEBENEN METHODE

##### 4.1. Einleitung

Das Programm ROBUST sucht  $\lambda_{\text{krit}}$  innerhalb den vorgegebenen Grenzen  $\lambda_{\text{min}}$  und  $\lambda_{\text{max}}$  und berechnet anschliessend die zu diesem  $\lambda$ -Wert zugehörige konstante Rückführungsmatrix  $L(\underline{\alpha}_{\text{nom}}, \lambda_{\text{krit}})$ . Falls  $\lambda_{\text{krit}} \notin [\lambda_{\text{min}}, \lambda_{\text{max}}]$ , so wird das gegebene Gebiet im Parameterraum verkleinert und das neue Stabilitäts- und Robustheitsgebiet für  $\lambda = \lambda_{\text{max}}$  gefunden. Weiterhin ist das Programm allgemein in der Lage, für jeden vorgegebenen  $\lambda$ -Wert das entsprechende grosstmögliche Stabilitäts- und Robustheitsgebiet anzugeben.

Das Programm wurde gestaltet für Systeme bis zu 10. Ordnung und für maximal 10 Parameter. Eine Erweiterung auf Systeme höherer Ordnung, bzw. für eine grössere Anzahl Parameter ist jedoch mit sehr kleinem Aufwand jederzeit möglich.

Die Angaben die der Benutzer machen muss teilen sich in zwei Gruppen auf :

##### a) Im Programm

Die Systemmatrixelemente müssen als multilineare Funktionen der Parameter angegeben werden. (Parameter, welche die Bedingung der Multilinearität verletzen, sollen als Konstante angenommen werden.)

$$A(1,1) = f_{11}(\alpha_1, \alpha_2, \dots, \alpha_m)$$

$$A(1,2) = f_{12}(\alpha_1, \alpha_2, \dots, \alpha_m)$$

⋮

$$A(n,n) = f_{nn}(\alpha_1, \alpha_2, \dots, \alpha_m) \quad ; n = 1, \dots, 10$$

mit

$$\underline{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_m) \quad ; m = 1, \dots, 10$$

b) Im Datenfeld

1. Datenkarte

N : Systemordnung oder Anzahl Zustandsvariablen  
NR : Anzahl Steuervariablen  
M : Anzahl Parameter  
NGEB : Die neuen Grenzen des Gebietes werden erwünscht  
(NGEB = 1), bzw. nicht erwünscht (NGEB = 0)  
SR : Man interessiert sich für das Stabilitätsgebiet  
(SR = 1), bzw. für das Robustheitsgebiet (SR = 0)  
LAMIN :  $\lambda_{\min}$   
LAMAX :  $\lambda_{\max}$   
RHO :  $\rho$  (Sensitivitätskoeffizient)

2. Datenkarte

Q : Positiv-definite oder semidefinite Gewichtungsmatrix  
(NxN)

3. Datenkarte

R : Positiv-definite Gewichtungsmatrix (NRxNR)

4. Datenkarte

AQ :  $\bar{A}$  (Systemmatrix bei  $\underline{\alpha}_{\text{nom}}$ )

5. Datenkarte

BQ :  $\bar{B}$  (Steuermatrix bei  $\underline{\alpha}_{\text{nom}}$ )

6. Datenkarte

ALMIN :  $\underline{\alpha}_{\min}$  (Vektor, der die minimalen Grenzen der Parameter angibt)

7. Datenkarte

ALMAX :  $\underline{\alpha}_{\max}$  (Vektor, der die maximalen Grenzen der Parameter angibt)

8. Datenkarte

ALMIG : Vektor, der die grössten zugelassenen minimalen Grenzen der Parameter bei Gebietsverkleinerung angibt

$$\underline{\alpha}_{\min} < \text{ALMIG} \leq \underline{\alpha}_{\text{nom}}$$

9. Datenkarte

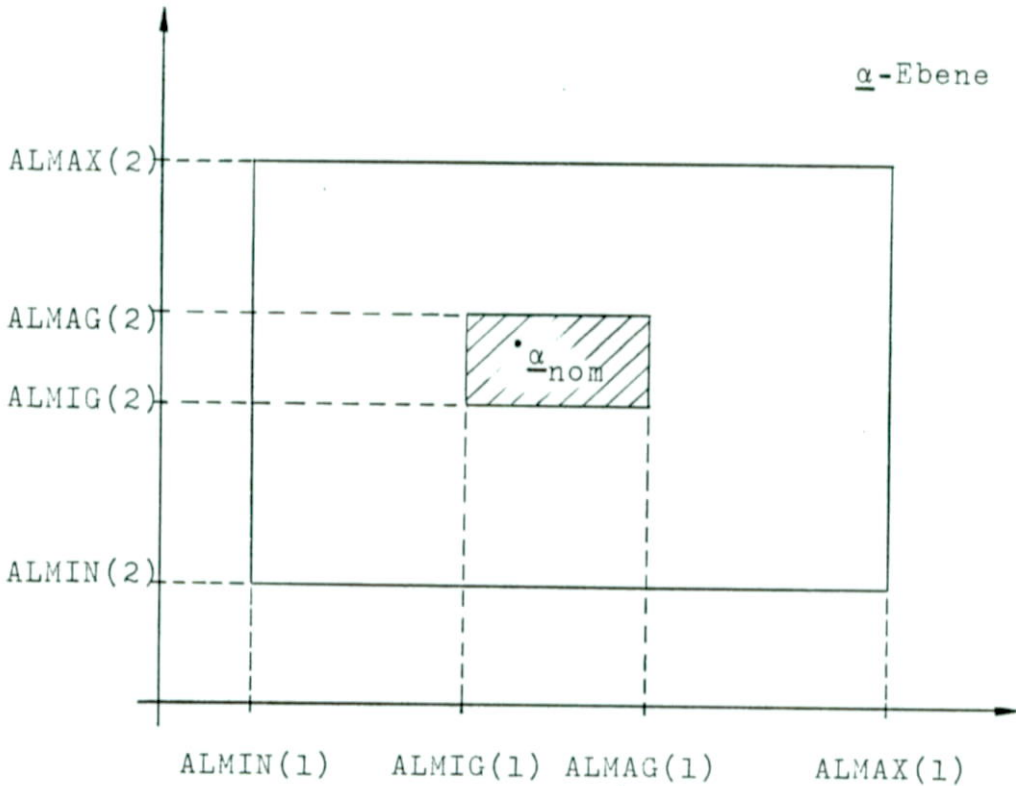
ALMAG : Vektor, der die kleinsten zugelassenen maximalen Grenzen der Parameter bei Gebietsverkleinerung angibt

$$\underline{\alpha}_{\text{nom}} \leq \text{ALMAG} < \underline{\alpha}_{\max}$$

Bemerkungen

1) Da man mit der Programmbibliothek des Institutes für Automatik arbeitet (AUTLIB), müssen alle Matrizen als Vektoren eingelesen werden (Storage Mode 0).

2) Um Verwechslungen mit den Abkürzungen der - in der 6., 7., 8. und 9. Datenkarte - eingeführten Parametervektoren zu vermeiden, siehe nachfolgende Skizze in der Parameterebene (M = 2) :



#### 4.2. Beschreibung des Hauptprogrammes und der Subroutinen

##### Hauptprogramm

Das Programm beginnt für  $\lambda = \lambda_{min}$  ( $LA = LAMIN$ ). Der zur 1. Ecke des Hyperquaders zugehöriger Parametervektor  $\underline{\alpha}$  (ALPHA) wird - aus den eingelesenen Grenzen  $\underline{\alpha}_{min}$  (ALMIN) und  $\underline{\alpha}_{max}$  (ALMAX) - berechnet und in der 1. Zeile der Tabelle 1 (TAB1) gespeichert. Die Subroutine ABMAT bestimmt nun die Matrizen  $A(\underline{\alpha})$  und  $B(\underline{\alpha})$  und das Bibliothekprogramm LCRFBI gibt die Lösung  $K_R(\underline{\alpha}_{nom}, \lambda)$  der Matrix-Riccati-Gleichung und die Rückführungsmatrix  $L(\underline{\alpha}_{nom}, \lambda)$  an. Anschliessend werden in FPMAT die Matrix  $F(\underline{\alpha}, \underline{\alpha}_{nom}, \lambda)$  des rückgekoppelten Systems und die Lösung  $P(\underline{\alpha}, \underline{\alpha}_{nom}, \lambda)$  der Ljapunov Gleichung - letzteres mit Hilfe der Bibliotheksubroutine ATXPXA - berechnet.

In der Subroutine TEST werden die Eigenwerte von P bestimmt - mit Hilfe der AUTLIB-Subroutine EIGVAL - und die doppelte Ungleichung (3.14) geprüft. Ist sie für alle Eigenwerte von P erfüllt, dann wird Testparameter  $TP = 1$  gesetzt, andernfalls wird  $TP = 0$ . Ist  $TP = 1$ , so wird der zur 2. Ecke des Quaders zugehöriger Parametervektor  $\underline{\alpha}$  berechnet, in der 2. Zeile der Tabelle 1 gespeichert und alle genannten Schritte für dieses neue  $\underline{\alpha}$  wiederholt. Dieser Vorgang wird so lange weitergeführt, bis, entweder alle Ecken durchlaufen worden sind und in diesem Falle stoppt das Programm mit der Meldung " $\lambda_{\min}$  wurde zu gross gewählt" ( $\lambda_{\text{krit}} < \lambda_{\min}$ ), oder der Test fällt für irgend eine Ecke  $\underline{\alpha}$  durch. Trifft letzteres zu -  $TP = 0$  -, so werden für das betreffende  $\underline{\alpha}$  die Matrizen  $K_R$ , F und P neu berechnet jetzt aber für  $\lambda = \lambda_{\max}$  (LA = LAMAX). Das heisst also, es wird ab dieser Ecke weitergefahren, unter Verwendung der in 3.5. Punkt B) gemachten Aussage, dass, wenn der Test bei den vorher untersuchten Ecken für  $\lambda_{\min}$  erfolgreich war, dann wird er auch dort bei  $\lambda_{\max}$  erfolgreich sein.

Für  $\lambda = \lambda_{\max}$  unterscheidet man auch zwei Fälle :

1) Der Test ist für alle Ecken erfüllt

oder

2) Der Test ist für irgend ein  $\underline{\alpha}$  nicht erfüllt

Trifft 2) zu, dann heisst dies, dass  $\lambda_{\text{krit}} > \lambda_{\max}$  ist und die Meldung " $\lambda_{\max}$  wurde zu klein gewählt" wird herausgedruckt. Falls nun das neue Stabilitäts- bzw. Robustheitsgebiet erwünscht wird ( in der 1. Datenkarte NGE = 1), wird die Subroutine GEBIET aufgerufen, die die Aufgabe übernimmt die betreffende Ecke  $\underline{\alpha}$  - mit Hilfe der Bisektionsmethode - so lange in Richtung von  $\underline{\alpha}_{\text{nom}}$  zu verschieben, bis der Test wieder erfolgreich ist. In diesem Falle (LL = 1) wird die entsprechende Zeile der Tabelle 1 mit dem neuen  $\underline{\alpha}$  überschrieben und es wird

zur nächsten Ecke gesprungen. Es kann aber passieren, dass das neue Stabilitäts- bzw. Robustheitsgebiet kleiner ist, als das durch ALMIG und ALMAG angegebene Minimalgebiet ( $LL = 0$ ); trifft dies zu, dann stoppt das Programm mit entsprechenden Meldungen. Nehmen wir aber an, dass  $LL = 1$  ist. Wenn bei irgend einer der noch zu untersuchenden Ecken der Test wieder erfolglos ist, dann wird nochmals die betreffende Ecke mit Subroutine GEBIET verschoben. (Die Variable NS bewirkt, dass die Meldung " $\lambda_{\max}$  wurde zu klein gewählt" nur einmal erscheint und nicht jedesmal, wenn GEBIET aufgerufen wird). Es seien nun alle Ecken durchlaufen worden. Dann sind wir in Fall 1).

Trifft also 1) zu, dann kann dies, wie wir gesehen haben, zweierlei bedeuten: Entweder ist bei irgend einer Ecke, oder bei mehreren, Subroutine GEBIET aufgerufen worden ( $KX = K$ : Abbruchkonstante in GEBIET), oder das Gebiet musste nicht verkleinert werden ( $KX = 0$ ).

Im ersten Fall werden die neuen Grenzen der Parameter ALMIN und ALMAX berechnet und zusammen mit diesen wird auch  $L(\underline{\alpha}_{\text{nom}}, \lambda_{\text{krit}})$  herausgegeben. Das Programm stoppt hier mit der Angabe des neuen Stabilitätsgebietes ( $SR = 1$ ) bzw. des neuen Robustheitsgebietes ( $SR = 0$ ) und der Rückführungsmatrix  $L$ . Die Berechnung der neuen Vektoren ALMIN und ALMAX geschieht mit Hilfe der Tabelle 2 (TAB2), die in Beziehung gebracht wird zur Tabelle 1 (TAB1), einer  $(2^M \times M)$ -Matrix, dessen Elementen — wie schon gesagt wurde — die Komponenten aller, zu den Ecken des Hyperquaders zugeordneten, Parametervektoren  $\underline{\alpha}$  enthält. Tabelle 2 besteht aus lauter "0" und "1" und ist sozusagen der Code für Tabelle 1, ist also auch eine  $(2^M \times M)$ -Matrix. (Es gibt  $2^M$  Ecken, also  $2^M$  Parametervektoren  $\underline{\alpha}$ , bei denen wir den Test (3.14) anwenden;  $M$  ist die Anzahl Komponenten in  $\underline{\alpha}$ ). Wie dies im einzelnen vor sich geht, sei an Hand eines einfachen fiktiven Beispiels auf Seite 43 gezeigt.

Im zweiten Fall ( $KX = 0$ ) sind wir sicher, dass

$$\lambda_{\min} < \lambda_{\text{krit}} < \lambda_{\max}$$

und die eigentliche Suche nach  $\lambda_{\text{krit}}$  kann, mit Hilfe der Bisektionsmethode, beginnen. Als Abbruchkriterium wählten wir

$$\frac{\Delta\lambda}{\lambda} < \epsilon$$

mit  $\epsilon = 10^{-6}$ . Die Bisektionsmethode beginnt bei  $\lambda = \lambda_{\min} + \Delta\lambda$  ( $LA = LAMIN + DELTA$ ), wobei anfänglich

$$\Delta\lambda = \frac{\lambda_{\max} - \lambda_{\min}}{2}$$

ist. Hier wird wieder Gebrauch von der Bemerkung in Punkt B) von 3.5. gemacht, d.h. für dieses  $\lambda$  prüft man den Test von der Ecke an, bei der dieser - bei  $\lambda = \lambda_{\min}$  - durchgefallen war. Diese Eckennummer wurde in IXX gespeichert; es werden also nur die Ecken von  $I = IXX + 1$  bis  $I = 2^M$  durchgelaufen. Ist für irgend einer dieser Ecken der Test nicht erfüllt, so wird  $\Delta\lambda$  halbiert, die betreffende Eckennummer in IX gespeichert und mit  $\lambda = \lambda + \Delta\lambda$  nochmals versucht. Dies wird so lange getan - wobei jedesmal  $\Delta\lambda$  halbiert wird - bis entweder das Abbruchkriterium erreicht wird, oder der Test einmal erfolgreich ist, in dessen Fall man  $\lambda$  in LAX abspeichert,  $\Delta\lambda$  nochmals halbiert und mit  $\lambda = \lambda - \Delta\lambda$  zur 1. Ecke gesprungen wird. Die Erhöhung bzw. Verkleinerung von  $\lambda$  um  $\Delta\lambda = \Delta\lambda/2$  wird so lange fortgesetzt, bis das Abbruchkriterium erfüllt ist. Dabei beachte man folgendes :

- a) man verkleinert  $\lambda$  um  $\Delta\lambda$ , falls für das betreffende  $\lambda$  der Test bei allen  $2^M$  Ecken erfolgreich war
- b) man vergrössert  $\lambda$  um  $\Delta\lambda$ , falls für das betreffende  $\lambda$  der Test bei irgend einer Ecke erfolglos blieb, und zwar vergrössert man  $\lambda$  so lange, bis diese Ecke "wieder geht" und springt dann mit diesem Wert zur nächsten Ecke, wo der Test wieder vorgenommen wird.

Das Erreichen des Abbruchkriteriums kann unter den folgenden zwei Umständen geschehen :

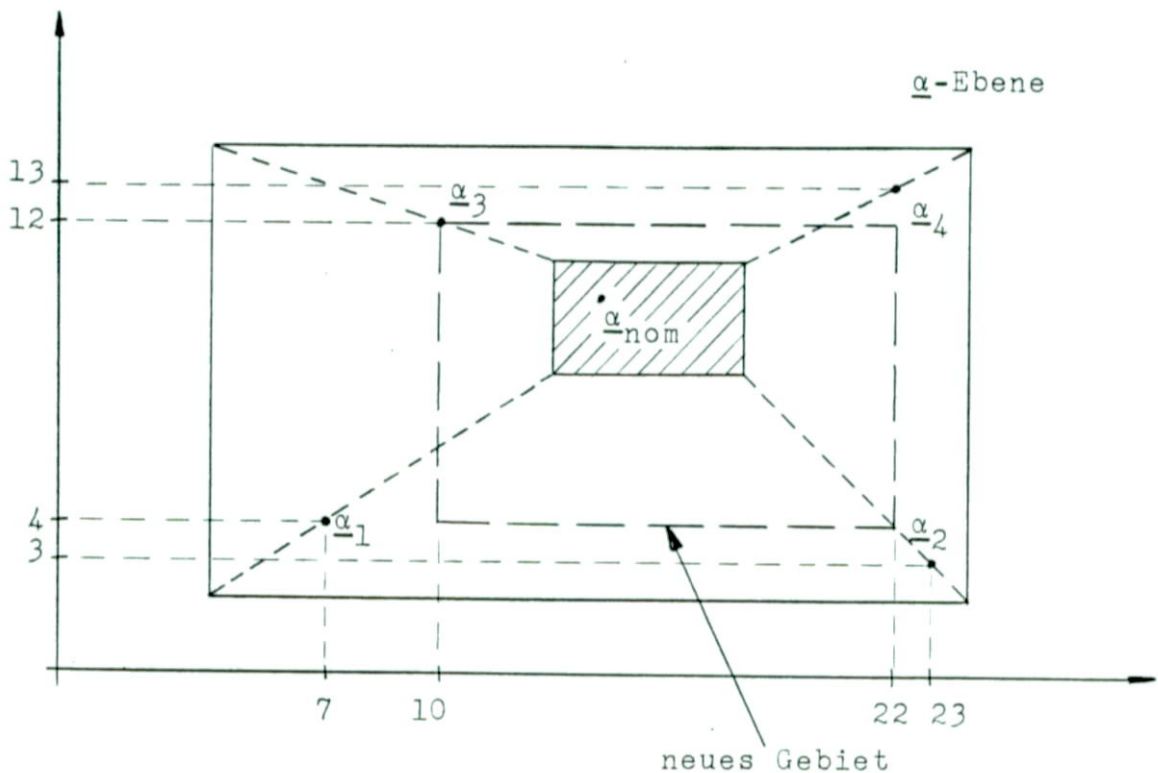
- 1) Das Abbruchkriterium wurde erreicht, nachdem alle Ecken durchgelaufen worden sind. In diesem Fall wird  $J = 1$  gesetzt und der Test wird - beginnend bei der 1. Ecke - nur für diejenigen Ecken vorgenommen, für die ein kleineres  $\lambda$ , als der aktuelle Wert benutzt worden war. Dies erreicht man mit dem IX, welches jedesmal um 1 reduziert wird. Erreicht IX den Wert 0, so hat man für das gefundene  $\lambda_{\text{krit}}$  alle Ecken geprüft;  $\lambda_{\text{krit}}$  und  $L(\underline{\alpha}_{\text{nom}}, \lambda_{\text{krit}})$  können herausgegeben werden und das Programm stoppt.
- 2) Das Abbruchkriterium wurde bei irgend einer Ecke erreicht, d.h., bevor alle Ecken durchlaufen worden sind. In diesem Fall wird 1) mit dem letzten  $\lambda$ -Wert, für welchen alle Ecken den Test bestanden haben ( $LA = LAX$ ), durchgenommen.

Dieser letzte Durchgang mit  $\lambda = \lambda_{\text{krit}}$  ( $J = 1$ ) wurde nur als Sicherheitskontrolle eingebaut. Falls  $\lambda$  in diesem Durchgang bei irgend einer noch zu prüfenden Ecke erhöht werden muss, dann muss die Meldung "Der Algorithmus versagt für dieses Beispiel" gedruckt werden und das Programm stoppt. Für die drei untersuchten Beispiele - und sicher für eine grosse Anzahl weiterer Systeme - war die Methode jedoch erfolgreich!

Bemerkung

Alle vier verwendeten Bibliothekprogramme :  
LCRFBI zur Berechnung der Lösung  $K_R$  der Matrix-Riccati-Gleichung  
und der Rückführungsmatrix des geschlossenen Systems L,  
ATXPXA zur Bestimmung der Lösung P der Ljapunov-Gleichung und,  
EIGVAL zur Berechnung der Eigenwerte von P, sowie auch  
MOUT zur Herausgabe von Matrizen oder Vektoren,  
sind von der Bibliothek des Instituts für Automatik, (AUTLIB).

Einfaches Beispiel-als Illustration-für die Berechnung der  
neuen Gebietsgrenzen ALMIN und ALMAX



Die durch die Subroutine GEBIET neu gefundenen Eckenvektoren sind in TAB1 gespeichert :

TAB1

$\underline{\alpha}_1$	7	4
$\underline{\alpha}_2$	23	3
$\underline{\alpha}_3$	10	12
$\underline{\alpha}_4$	22	13

Tabelle 2 entspricht dem Bildungsgesetz von den ALPHA-Vektoren, aus den gegebenen Grenzen ALMIN und ALMAX. Sie wird durch die Dezimal-Binär-Umwandlung gebildet, wobei LSB *links* und MSB *rechts* ist. Für den Fall  $M = 2$  wird TAB2 :

	$2^0$	$2^1$
0	0	0
1	1	0
2	0	1
3	1	1

In diesem 2-Dimensionalen Beispiel ( $M = 2$ ) wird :

$$MEE = 2^M = 4 \quad : \text{Anzahl Ecken}$$

$$MEF = MEE/2 = 2 \quad : \text{Anzahl 0 bzw. 1}$$

### Berechnung des neuen ALMIN-Vektors

$\underline{X}(\text{MEF})$  : Vektor, der die Komponenten des  $\underline{\alpha}$ -Vektors speichert, denen in TAB2 eine 0 zugeordnet ist; für jede Kolonne wird er überschrieben. In jedem  $\underline{X}$  wird die grösste Komponente ausgesucht; diese bilden zusammen die Koordinaten des neuen ALMIN-Vektors :

$$\text{Kolonne 1 : } \underline{X} = (7, 10)' \longrightarrow \text{ALMIN}(1) = 10$$

$$\text{Kolonne 2 : } \underline{X} = (4, 3)' \longrightarrow \text{ALMIN}(2) = 4$$

$$\implies \text{ALMIN}_{\text{neu}} = (10, 4)'$$

### Berechnung des neuen ALMAX-Vektors

$\underline{X}(\text{MEF})$  : Vektor, der die Komponenten des  $\underline{\alpha}$ -Vektors speichert, denen in TAB2 eine 1 zugeordnet ist; für jede Kolonne wird er überschrieben. In jedem  $\underline{X}$  wird die kleinste Komponente ausgesucht; diese bilden zusammen die Koordinaten des neuen ALMAX-Vektors :

$$\text{Kolonne 1 : } \underline{X} = (23, 22)' \longrightarrow \text{ALMAX}(1) = 22$$

$$\text{Kolonne 2 : } \underline{X} = (12, 13)' \longrightarrow \text{ALMAX}(2) = 12$$

$$\implies \text{ALMAX}_{\text{neu}} = (22, 12)'$$

### Subroutine ABMAT

Die Elemente der Systemmatrix A und der Steuermatrix B, die multilineare Funktionen der m Parameter sind, müssen vom Benutzer für jedes System neu geschrieben werden und zwar - wie anfangs schon erwähnt - werden die Matrizen als Vektoren dargestellt.

### Subroutine FPMAT

In dieser Subroutine werden die Matrix F des rückgekoppelten Systems

$$F = A - B L$$

mit Hilfe der beiden AUTLIB-Programme GMPRD und GMSUB, sowie auch die Matrix P, als Lösung der Ljapunov-Gleichung

$$F' P + P F = -Q \longrightarrow P$$

(mitATXPXA) berechnet.

### Subroutine TEST

Hier werden die Eigenwerte von P berechnet (mit EIGVAL). Anschliessend wird P nach positiv-Definitheit untersucht und schliesslich wird geprüft, ob die Eigenwerte von P das Zustandsinsensitivitätskriterium erfüllen. Es wird also die doppelte Ungleichung (3.14)

$$0 < \mu_i(P) \leq \rho \quad \text{für alle } i = 1, \dots, n$$

getestet.

### Subroutine GEBIET

Diese Subroutine wird dann aufgerufen, wenn für  $\lambda = \lambda_{\max}$ , die Ungleichung

$$0 < \mu_i(P) \leq \rho \quad \text{für alle } i = 1, \dots, n$$

bei irgend einer Ecke - charakterisiert durch den Vektor  $\underline{\alpha}$  - nicht erfüllt wird. Die Subroutine übernimmt dann die Aufgabe die betreffende Ecke in Richtung  $\underline{\alpha}_{nom}$  so lange zu verschieben, bis obige Ungleichung wieder erfüllt wird. Die neuen Ecken erlauben schliesslich die Berechnung der neuen Stabilitäts- bzw. Robustheitsgebiete. Letzteres geschieht im Hauptprogramm mit Hilfe von TAB1 und TAB2.

Bevor aber mit der Beschreibung der Subroutine begonnen wird, etwas zur Terminologie :

ALMIN }  
ALMAX } ursprüngliche Grenzen der Parameter  $\rightarrow$  ALPHA ( $\underline{\alpha}$ )

ALMIG }  
ALMAG } kleinste zugelassene Grenzen der Parameter  $\rightarrow$  ALPMI ( $\underline{\alpha}_{mi}$ )

(Siehe Skizze auf Seite 38).

Ganz analog wie im Hauptprogramm ALPHA aus ALMIN und ALMAX berechnet wurde, wird als erstes ALPMI aus ALMIG und ALMAG gewonnen. Erst dann kann - mit Hilfe der Bisektionsmethode - die Suche nach dem neuen  $\underline{\alpha}$  beginnen. Dazu wird der Hilfsvektor

$$\underline{hv} = \underline{\alpha}_{mi} - \underline{\alpha}$$

gebildet; und nach anschliessendem Halbieren von  $\underline{hv}$ , wird er dem Vektor  $\underline{\alpha}$  addiert. Die Matrizen  $A(\underline{\alpha})$ ,  $B(\underline{\alpha})$ ,  $F(\underline{\alpha})$  und  $P(\underline{\alpha})$  werden nun, mit Hilfe von ABMAT und FPMAT berechnet. Mit EIGVAL findet man die Eigenwerte von  $P$  ( $u_i\{P\}$ ), welche den folgenden Test zu erfüllen haben :

Ist  $SR = 1$  (Stabilitätsgebiet erwünscht; siehe 1. Datenkarte), dann wird nur die Ungleichung

$$\mu_i(P) > 0 \quad \text{für alle } i = 1, \dots, n$$

geprüft.

Ist  $SR = 0$  (Robustheitsgebiet erwünscht), dann muss zusätzlich noch

$$\mu_i(P) \leq \rho \quad \text{für alle } i = 1, \dots, n$$

getestet werden.

Unabhängig davon, ob  $SR = 1$  oder  $SR = 0$  ist, wird die eingeführte Testvariable

$$TP1 = \begin{cases} 1 & , \text{ falls der Test erfolgreich war} \\ 0 & , \text{ sonst.} \end{cases}$$

Nun wird der Hilfsvektor  $\underline{hv}$  nochmals halbiert.

Ist  $TP1 = 1$ , dann muss, nach vorheriger Abspeicherung des neuen Vektors  $\underline{\alpha}$  in  $\underline{\alpha}^*(ALPST)$  die Summe

$$\underline{\alpha} = \underline{\alpha} + \underline{hv}$$

gebildet werden.

Ist  $TP1 = 0$ , dann muss die Differenz

$$\underline{\alpha} = \underline{\alpha} - \underline{hv}$$

berechnet werden.

In beiden Fällen erhält man einen neuen  $\underline{\alpha}$ -Vektor mit dem nun nochmals die Matrizen  $A(\underline{\alpha})$ ,  $B(\underline{\alpha})$ ,  $F(\underline{\alpha})$  und  $P(\underline{\alpha})$  berechnet werden; die neuen Eigenwerte von  $P$  werden wiederum dem Test unter-

worfen. Dieser Vorgang wird so lange wiederholt (dabei wird jedesmal der Betrag von  $h_v$  um die Hälfte reduziert; die Richtung bleibt erhalten), bis man eine gegebene Anzahl Schritte — die mit der Laufvariable  $K$  gezählt wird — erreicht hat. Dann wird die Bisektionsmethode abgebrochen. Der neue  $\underline{\alpha}$  ist in  $\underline{\alpha}^*$  abgespeichert.

Falls nun der Test *immer* erfolglos blieb, wird  $LL = 0$ , war er jedoch *mindestens einmal* erfolgreich, dann wird  $LL = 1$ . Im Hauptprogramm wird dann nach  $LL$  gefragt. Im ersten Fall ( $LL=0$ ) wird die Meldung

"Das gesuchte Stabilitätsgebiet ist kleiner als das angegebene Minimalgebiet" ( $SR = 1$ )

oder

"Das gesuchte Robustheitsgebiet ist kleiner als das angegebene Minimalgebiet" ( $SR = 0$ )

herausgedruckt. Im zweiten Fall ( $LL=1$ ) wird mit  $\underline{\alpha} = \underline{\alpha}^*$  die entsprechende Zeile in Tabelle 1 überschrieben. Siehe das dazugehörige Flussdiagramm in 4.3. und nachfolgende Skizze für  $K = 3$ .

Bemerkung

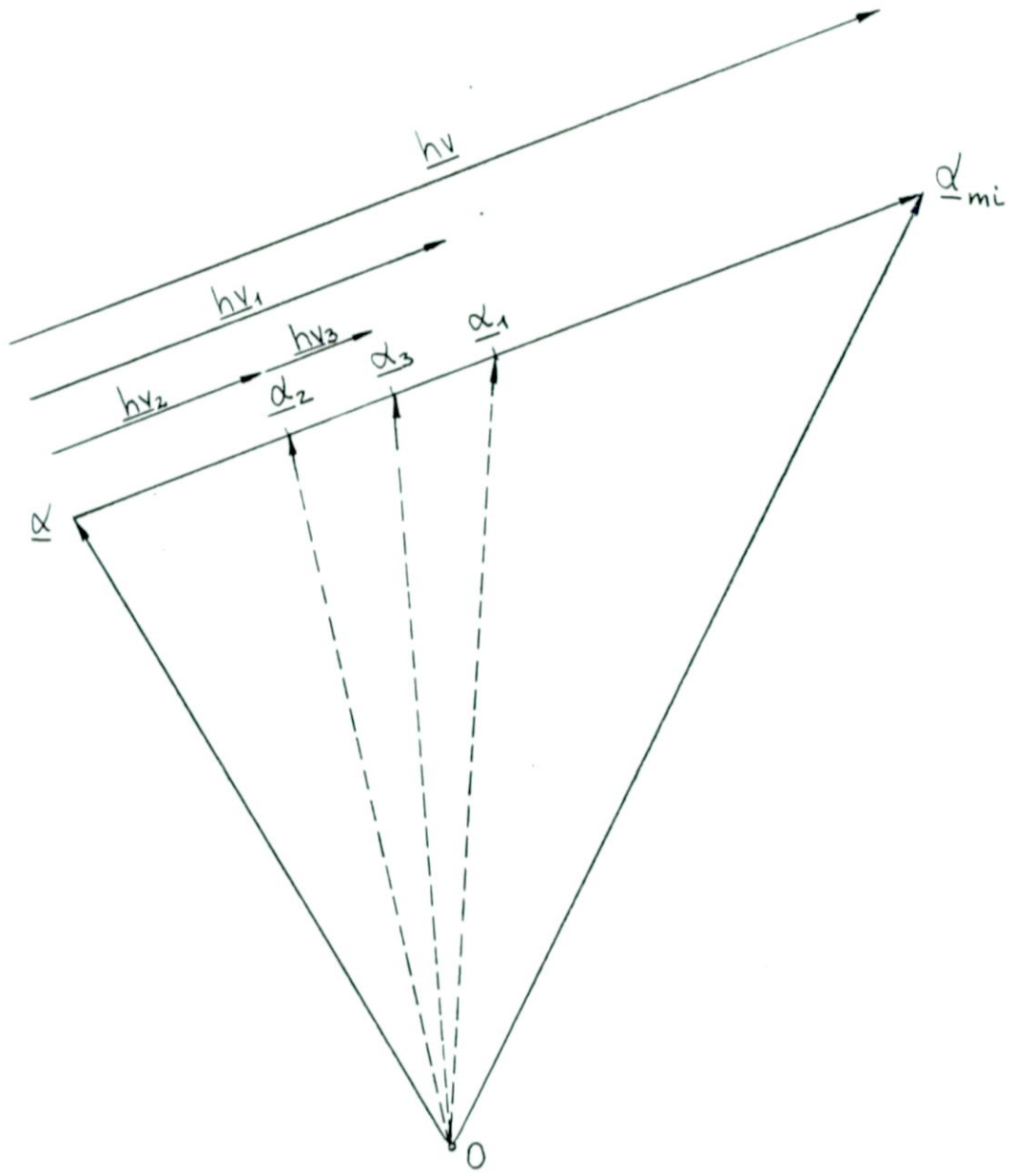
Alle Additionen und Subtraktionen der Vektoren werden mit den Bibliothekprogrammen von AUTLIB

GMADD

und

GMSUB

durchgeführt.



Koordinatenursprung

4.4. Das Programm Robust

```
PROGRAM ROBUST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
```

```
INTEGER TP,SR,TAB2
```

```
REAL    QO(100),R(100),AQ(100),BQ(100),KR(100),L(100),  
$       VR(400),VI(400),HH(400),H1(100),H2(100),H3(100),  
$       ER(20),EI(20),LH1(10),LH2(10),LA,LAX,LAMAX,LAMIN
```

```
DIMENSION I1(20),TAB2(1024,10),X(512)
```

```
COMMON /1/ ALPHA(10),ALMIN(10),ALMAX(10),ALPMI(10),ALMIG(10),  
$          ALMAG(10),ALPST(10),SR,K,M,I,LL
```

```
COMMON /2/ N,RHO,INDEX,TP,P(100),A(100),B(100),  
$          Q(100),L,NR,NQUA,IER,C(100),U(100),FF(121),  
$          BL(100),F(100),WR(10),WI(10),ZR(100),ZI(100),  
$          Z(100),FV1(10),IV1(10),TAB1(1024,10)
```

\*\*\*\*\*

- \* N : SYSTEMORDNUNG ODER ANZAHL ZUSTANDSVARIABLEN
- \* NR : ANZAHL STEUERVARIABLEN
- \* M : ANZAHL PARAMETER (DIMENSION VON ALPHA-VEKTOR)
- \* NGEB : NGEB=1 DIE NEUEN GRENZEN DES GEBIETES WERDEN ERWUENSCHT  
 : NGEB=0 DIE NEUEN GRENZEN DES GEBIETES WERDEN NICHT ERWUENSCHT
- \* SR : SR=1 DAS STABILITAETS GEBIET WIRD GESUCHT  
 : SR=0 DAS ROBUSTHEITS GEBIET WIRD GESUCHT
- \* LAMIN : KLEINSTES LAMBDA
- \* LAMAX : GROESSTES LAMBDA
- \* RHO : SENSITIVITAETSKOEFFIZIENT
- \* Q : POSITIV DEFINITE ODER SEMIDEFINITE GEWICHTUNGSMATRIX(N\*N)
- \* R : POSITIV DEFINITE GEWICHTUNGSMATRIX(NR\*NR)
- \* AQ : SYSTEMMATRIX BEI DEN NENNWERTEN DER PARAMETER(N\*N)
- \* BQ : STEUERMATRIX BEI DEN NENNWERTEN DER PARAMETER(N\*NR)
- \* ALMIN, :
- \* ALMAX : URSPRUENGLICHE GRENZEN DER PARAMETER
- \* ALPHA : DER AUS ALMIN UND ALMAX ZU BERECHENDER VEKTOR
- \* ALMIG, :
- \* ALMAG : KLEINSTE ZUGELASSENE GRENZEN DER PARAMETER BEI GEBIETSVERKLEINERUNG
- \* ALPMI : DER AUS ALMIG UND ALMAG ZU BERECHENDER VEKTOR

\*\*\*\*\*

```
C
100 READ(5,100) N,NR,M,SR,NGEB,LAMIN,LAMAX,RHO
    FORMAT(5I4,3F15.7)
    NQUA=N*N
    NRQUA=NR*NR
    NNR=N*NR
101 READ(5,101) (Q(I), I=1,NQUA)
    FORMAT(10F7.3)
    READ(5,102) (R(I), I=1,NRQUA)
    READ(5,102) (AQ(I), I=1,NQUA)
    READ(5,102) (BQ(I), I=1,NNR)
102 FORMAT(10F7.3)
    READ(5,103) (ALMIN(I), I=1,M)
    READ(5,103) (ALMAX(I), I=1,M)
    READ(5,103) (ALMIG(I), I=1,M)
    READ(5,103) (ALMAG(I), I=1,M)
103 FORMAT(10F9.5)
C
201 WRITE(6,201) N,NR,M,SR,NGEB,LAMIN,LAMAX,RHO
    FORMAT(1H1,20X,*SYSTEMORDNUNG : *,I4 ,/,21X,*ANZAHL STE
$UERVARIABLEN : *,I4 ,/,21X,*ANZAHL PARAMETER : *,I4 ,///
$,21X,*SR : *,I4,/,21X,*NGEB : *,I4,/,21X,*LAMIN : *,F7.2 ,/,21
$X,*LAMAX : *,F7.2,/,21X,*RHO : *,F7.2)
    CALL MOUT(20HQ-MATRIX ,Q,N,N,0)
    CALL MOUT(20HR-MATRIX ,R,NR,NR,0)
    CALL MOUT(20HA-QUER-MATRIX ,AQ,N,N,0)
    CALL MOUT(20HB-QUER-MATRIX ,BQ,N,NR,0)
    WRITE(6,200)
200 FORMAT(////////,*)
    CALL MOUT(20HALMIN-VEKTOR ,ALMIN,1,M,0)
    CALL MOUT(20HALMAX=ALPHA-QUER ,ALMAX,1,M,0)
    CALL MOUT(20HALMIG-VEKTOR ,ALMIG,1,M,0)
    CALL MOUT(20HALMAG-VEKTOR ,ALMAG,1,M,0)
```

EINIGE KONSTANTEN UND VARIABLEN WERDEN DEFINIERT.

```
I3=2**M-1
DELTA=(LAMAX-LAMIN)/2.
EPSI=0.0000001
LA=LAMIN
KX=0
NS=0
IX=0
IXX=0
J=0
I=-1
I=I+1
K=0

IF(IX.EQ.0.OR.J.NE.1) GOTO 3
IX=IX-1
CONTINUE
```

DER ZUR I-TEN ECKE DES HYPERQUADERS ZUGEHÖRIGER PARAMETERVEKTOR ALPHA WIRD BERECHNET.

```
M1=M+1
ME=2**(M-1)
I2=I
M1=M1-1
IF(M1.EQ.0) GOTO 6
ID=I2/ME
IF(ID.NE.0) GOTO 5
ALPHA(M1)=ALMIN(M1)
ME=ME/2
GOTO 4
ALPHA(M1)=ALMAX(M1)
I2=I2-ME
ME=ME/2
GOTO 4
CONTINUE
```

```
IF(NGEB.EQ.0) GOTO 51
```

BILDUNG VON TABELLE 1

```
II=I+1
DO 50 M1=1,M
TAB1(II,M1)=ALPHA(M1)
```

BERECHNUNG DER A-UND B-MATRIZEN FÜR DIE AKTUELLEN WERTE VON ALPHA UND LAMBDA.

```
CALL ABMAT
```

```
CONTINUE
```

```
MSQ=0
```

```
MSR=0
```

```
DO 8 II=1,NQUA
```

```
QO(II)=LA*Q(II)
```

```
CALL LCRFB1(AQ,BQ,QO,R,MSQ,MSR,N,NR,KR,L,ER,EI,VR,VI,I1,HH,H1,H2,  
$H3,LH1,LH2,INDER)
```

```
IF(INDER.NE.0) WRITE(6,202) INDER
```

```
FORMAT(///,20X,*INDER=*,I3)
```

BERECHNUNG DER MATRIX  $F=A-B*L$  DES RUECKGEKOPPELTEN SYSTEMS.  
BERECHNUNG DER MATRIX-P, ALS LOESUNG DER LJAPUNOV-MATRIX-  
GLEICHUNG  $F'*P+P*F=-Q$ , MIT HILFE DER AUTLIB-SUBROUTINE  
ATXPXA.

```
CALL FPMAT
```

```
IF(IER.EQ.0) GOTO 9
```

```
WRITE(6,203)
```

```
FORMAT(///,10X,*LJAPUNOV-GLEICHUNG UNLOESBAR*)
```

```
GOTO 23
```

```
CONTINUE
```

```
IF(LA.NE.LAMIN) GOTO 13
```

DIE UNGLEICHUNG

$0 < \text{EIGENWERTE VON } P \leq \text{RHO}$

WIRD FUER ALLE EIGENWERTE VON P GEPRUEFT(FALLS SIE ERFUELLT IST,WIRD TP=1,ANDERFALLS IST TP=0.)

CALL TEST

NUMMER=1

IF(INDEX.EQ.0) GOTO(11,14,17,25),NUMMER

WRITE(6,204)

204 . FORMAT(///,10X,\*NICHT ALLE EIGENWERTE VON P GEFUNDEN\*)

GOTO 23

1 IF(TP.EQ.1) GOTO 12

LA=LAMAX

GOTO 7

2 IXX=I

IF(I.LT.I3) GOTO 2

WRITE(6,205)

205 . FORMAT(///,10X,\*LAMBDA-MIN WURDE ZU GROSS GEWAEHLT\*)

GOTO 23

3 CONTINUE

IF(LA.NE.LAMAX) GOTO 16

CALL TEST

NUMMER=2

GOTO 10

4 IF(TP.EQ.1) GOTO 15

NS=NS+1

IF(NS.NE.1) GOTO 24

WRITE(6,206)

06 . FORMAT(///,10X,\*LAMBDA-MAX WURDE ZU KLEIN GEWAEHLT\*)

IF(NGEB.EQ.1) GOTO 24

GOTO 23

FALLS,FUER LAMBDA-MAX DER TEST ERFOLGLOS IST(TP=0),WIRD DAS VORGEGEBENE GEBIET IM PARAMETERRAUM VERKLEINERT.

4 CALL GEBIET

KX=K

NUMMER=4

GOTO 10

5 IF(LL.NE.0) GOTO 26

IF(SR.EQ.1) GOTO 27

WRITE(6,207)

07 . FORMAT(///,10X,\*DAS GESUCHTE ROBUSTHEITSGEBIET IST KLEINER ALS DAS \$ ANGELEGEBENE MINIMALGEBIET\*)

GOTO 23

7 WRITE(6,208)

08 . FORMAT(///,10X,\*DAS GESUCHTE STABILITAETS GEBIET IST KLEINER ALS DA \$S ANGELEGEBENE MINIMALGEBIET\*)

GOTO 23

6 DO 28 KK=1,M

8 ALPHA(KK)=ALPST(KK)

C UEBERSCHREIBUNG DER ZEILE I+1 DER TABELLE 1  
C:

II=I+1  
DO 29 KK=1,M  
29 TAB1(II, KK)=ALPHA(KK)  
C  
IF(I.LT.I3) GOTO 2  
GOTO 30  
15 CONTINUE  
IF(I.LT.I3) GOTO 2  
IF(KX.EQ.0) GOTO 47

C  
C  
C  
C BILDUNG VON TABELLE 2

30 MEE=2\*\*M  
DO 36 JJ=1,M  
J1=2\*\*(JJ-1)  
II=1  
31 KK=0  
32 TAB2(II, JJ)=0  
IF(II.EQ.MEE) GOTO 35  
II=II+1  
KK=KK+1  
IF(KK.EQ.J1) GOTO 33  
GOTO 32  
33 KK=0  
34 TAB2(II, JJ)=1  
IF(II.EQ.MEE) GOTO 35  
II=II+1  
KK=KK+1  
IF(KK.EQ.J1) GOTO 31  
GOTO 34  
5 CONTINUE  
6 CONTINUE

BILDUNG DER NEUEN GRENZEN DER PARAMETER ALMIN UND ALMAX.

MEF=MEE/2  
DO 40 JJ=1,M  
KK=0  
DO 38 II=1,MEE  
KK=KK+1  
IF(TAB2(II, JJ).EQ.0) GOTO 37  
KK=KK-1  
GOTO 38  
7 X(KK)=TAB1(II, JJ)  
8 CONTINUE  
  
XMAX=X(1)  
DO 39 KK=2,MEE  
IF(X(KK).LE.XMAX) GOTO 39  
XMAX=X(KK)  
9 CONTINUE  
ALMIN(JJ)=XMAX  
0 CONTINUE



```
NUMMER=3
GOTO 10
17 IF(TP.NE.1) GOTO 18
CONTINUE
IF(I.LT.I3) GOTO 19
LAX=LA
DELMIN=DELTA/LAMAX
IF(DELMIN.LT.EPSI) GOTO 20
DELTA=DELTA/2.
LA=LA-DELTA

GOTO 1
8 CONTINUE
DELMIN=DELTA/LAMAX
IF(DELMIN.LT.EPSI) GOTO 21
DELTA=DELTA/2.
IX=I
LA=LA+DELTA

GOTO 7
9 CONTINUE
IF(.NOT.(J.EQ.1.AND.IX.EQ.0)) GOTO 2
WRITE(6,215)
15 FORMAT(/////,1X,*****#)
WRITE(6,216) LA
16 FORMAT(1X,** LAMBDA-KRITISCH =#,F9.5,1X,**)
WRITE(6,217)
17 FORMAT(1X,*****#)

CALL MOUT(20HL-MATRIX ,L,NR,N,0)

GOTO 23

FUER DAS GEFUNDENE LAMBDA WIRD SCHLIESSLICH NOCH EIN UMLAUF GEMACHT-
ZUR KONTROLLE-DURCH ALLE ECKEN DES HYPERQUADERS.

0 J=1
GOTO 1
1 CONTINUE
IF(J.NE.0) GOTO 22
LA=LAX
GOTO 20
2 WRITE(6,218)
18 FORMAT(///,10X,*ALGORITHMUS VERSAGT FUER DIESES BEISPIEL*)
WRITE(6,219)
19 FORMAT(10X,*****#)
3 CONTINUE
STOP
END
```

SUBROUTINE ABMAT

DIE ELEMENTEN DER A- UND B-MATRIZEN, DIE FUNKTIONEN DER PARAMETER SIND, MUESSEN VOM BENUETZER FUER JEDES SYSTEM NEU GESCHRIEBEN WERDEN.

INTEGER TP  
COMMON /1/ ALPHA(10)  
COMMON /2/ N,RHO,INDEX,TP,P(100),A(100),B(100)

DIE N\*N SYSTEMMATRIX A

A(1)=ALPHA(1)  
A(2)=ALPHA(2)  
A(3)=.0  
A(4)=1.  
A(5)=ALPHA(3)  
A(6)=.0  
A(7)=ALPHA(4)  
A(8)=ALPHA(5)  
A(9)=ALPHA(6)

DIE N\*NR STEUERMATRIX B

B(1)=.0  
B(2)=.0  
B(3)=6.6667

RETURN  
END

SUBROUTINE FPMAT

DIE MATRIZEN F UND P WERDEN BERECHNET

```
INTEGER      TP
REAL         L(100)
COMMON /2/   N,RHO,INDEX,TP,P(100),A(100),B(100),
$           Q(100),L,NR,NQUA,IER,C(100),U(100),FF(121),
$           BL(100),F(100)
```

BERECHNUNG VON BL=B\*L

```
CALL GMPRD(B,L,BL,N,NR,N)
```

BERECHNUNG VON F=A-B\*L

```
CALL GMSUB(A,BL,F,N,N)
```

BERECHNUNG DER P-MATRIX, ALS LOESUNG DER LJAPUNOV-MATRIX-GLEICHUNG

$$A' * X + X * A = C \quad , \quad \text{WOBEI} \quad A = F \quad , \quad X = P \quad , \quad C = -Q$$

MIT HILFE DER AUTLIB SUBROUTINE ATXPXA.

```
DO 1 II=1,NQUA
C(II)=-Q(II)
```

DIE MATRIX F(N,N) MUSS FUER DEN INTERNEN GEBRAUCH IN DIESER AUTLIB SUBROUTINE UMGESCHRIEBEN WERDEN ZU FF(N+1,N+1).

```
EPS=1.E-20
NA=N+1
IIX=1
DO 2 II=1,NQUA,NA
NE=II+N-1
DO 2 JII=II,NE
FF(JII)=F(IIX)
IIX=IIX+1
```

```
CALL ATXPXA(FF,U,C,N,NA,N,N,EPS,IER)
```

```
IF(IER.NE.0) GOTO 4
DO 3 II=1,NQUA
P(II)=C(II)
CONTINUE
RETURN
END
```

SUBROUTINE TEST

MIT HILFE DER AUTLIB SUBROUTINE EIGVAL WERDEN DIE EIGENWERTE DER P-MATRIX BERECHNET UND DANN WIRD DIE P-MATRIX NACH IHRER POSITIV-DEFINITHEIT UNTERSUCHT. SCHLIESSLICH WIRD DAS ZUSTANDS-INSENSITIVITAETS KRITERIUM FUER ALLE EIGENWERTE VON P GEPRUEFT.

```
INTEGER    TP
REAL       L(100)
COMMON /2/ N,RHO,INDEX,TP,P(100),A(100),B(100),
$          Q(100),L,NR,NQUA,IER,C(100),U(100),FF(121),
$          BL(100),F(100),WR(10),WI(10),ZR(100),ZI(100),
$          Z(100),FV1(10),IV1(10)
```

```
CALL EIGVAL(N,P,WR,WI,ZR,ZI,Z,FV1,IV1,INDEX)
```

```
IF(INDEX.NE.0) GOTO 4
K=0
K=K+1
IF(K.GT.N) GOTO 2
IF(WR(K).GT.0.) GOTO 1
TP=0
GOTO 4
CONTINUE
```

```
K=0
TP=1
K=K+1
IF(K.GT.N) GOTO 4
IF(WR(K).LE.RHO) GOTO 3
TP=0
CONTINUE
RETURN
END
```

SUBROUTINE GEBIET

BERECHNUNG DES NEUEN PARAMETERVEKTORS ALPHA, MIT DEM DIE NEUEN GRENZEN DES STABILITAETS-BEZ. ROBUSTHEITSGEBIETES IM HAUPTPROGRAMM BESTIMMT WERDEN KOENNEN ,FUER LAMBDA = LAMBDA-MAX.

```
INTEGER    TP,SR,TP1
REAL       L(100),HV(10)
COMMON /1/ ALPHA(10),ALMIN(10),ALMAX(10),ALPMI(10),ALMIG(10),
$          ALMAG(10),ALPST(10),SR,K,M,I,LL
COMMON /2/ N,RHO,INDEX,TP,P(100),A(100),B(100),
$          Q(100),L,NR,NQUA,IER,C(100),U(100),FF(121),
$          BL(100),F(100),WR(10),WI(10),ZR(100),ZI(100),
$          Z(100),FV1(10),IV1(10),TAB1(1024,10)
```

```
LL=0
TP1=0
```

BERECHNUNG VON ALPMI-DES KLEINSTEN ZUGELASSENEN PARAMETERVEKTORS-AUS DEN VORGEGEBENEN MINIMALGRENZEN ALMIG UND ALMAG.

```
M1=M+1
ME=2**(M-1)
I2=I
M1=M1-1
IF(M1.EQ.0) GOTO 3
ID=I2/ME
IF(ID.NE.0) GOTO 2
ALPMI(M1)=ALMIG(M1)
ME=ME/2
GOTO 1
ALPMI(M1)=ALMAG(M1)
I2=I2-ME
ME=ME/2
GOTO 1
CONTINUE
```

VERSCHIEBUNG DER I-TEN ECKE DES HYPERQUADERS IN RICHTUNG DES NENNPARAMETERPUNKTES, MIT HILFE DER METHODE DER BISEKTION.

```
CALL GMSUB(ALPMI,ALPHA,HV,1,M)
```

```
K=K+1
IF(K.GT.30) GOTO 13
```

```
DO 5 KK=1,M
HV(KK)=0.5*HV(KK)
```

```
IF(TP1.NE.1) GOTO 7
LL=1
```

```
DO 6 KK=1,M
ALPST(KK)=ALPHA(KK)
```

```
CALL GMSUB(ALPHA,HV,ALPHA,1,M)
```

```
GOTO 8
```

```
C  
7 CALL GMADD(ALPHA,HV,ALPHA,1,M)  
C  
B CALL ABMAT  
C  
C CALL FPMAT  
C  
CALL EIGVAL(N,P,WR,WI,ZR,ZI,Z,FV1,IV1,INDEX)  
IF(INDEX.NE.0) GOTO 13  
KK=0  
9 KK=KK+1  
TP1=1  
IF(KK.GT.N) GOTO 10  
IF(WR(KK).GT.0.) GOTO 9  
TP1=0  
GOTO 12  
0 CONTINUE  
IF(SR.EQ.1) GOTO 12  
KK=0  
1 KK=KK+1  
IF(KK.GT.N) GOTO 12  
IF(WR(KK).LE.RHO) GOTO 11  
TP1=0  
2 CONTINUE  
GOTO 4  
3 CONTINUE  
RETURN  
END
```

## 5. ANWENDUNG DES PROGRAMMPAKETES AN DREI BEISPIELEN

### 5.1. "Pitch-axis control system" eines Flugzeuges

#### 5.1.1 Systembeschreibung

Die beiden Autoren Krishnan und Brzezowski bringen in ihrem Bericht ein Beispiel bei dem sie ihre Methode anwenden. Es handelt sich um ein Flugzeug, dessen "Pitch-axis" geregelt werden soll. Wir bestätigten ihre Ergebnisse mit unserem Programm ROBUST.

Das System ist 3. Ordnung und wird gegeben durch die Vektordifferentialgleichung

$$\dot{\underline{x}} = A\underline{x} + \underline{b}u \quad , \quad \underline{x}(0) = \underline{x}_0$$

wobei

$x_1$  : Einstellwinkel (angle of attack)

$x_2$  : Zeitliche Aenderung des Knickwinkels (rate of pitch angle)

$x_3$  : Höhenruderwinkel (elevator angle)

und

$u$  : Steuergrösse des Höhenruderstellgliedes (elevator actuator) bedeuten.

Die Parameterabhängigkeit wird verursacht durch die grossen Druckunterschiede, die ein Flugzeug bei der Aenderung der Flughöhe erfährt.

Die Systemmatrix A ist in diesem speziellen Beispiel der Form

$$A = \begin{bmatrix} a_{11} & 1 & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}$$

Der Hyperquader im 6-dim. Parameterraum wird aufgespannt, durch die Grenzen

$$-0.074 \leq a_{11} \leq -0.0016$$

$$-8.0 \leq a_{21} \leq -0.1569$$

$$-0.055 \leq a_{22} \leq -0.0015$$

$$-0.012 \leq a_{13} \leq -0.0002$$

$$-6.2 \leq a_{23} \leq -0.1131$$

$$-6.667 \leq a_{33} \leq -4.6667$$

wobei

$$\underline{\alpha}_{\text{nom}} = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{22} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} = \begin{bmatrix} -0.0016 \\ -0.1569 \\ -0.0015 \\ -0.0002 \\ -0.1131 \\ -4.6667 \end{bmatrix}_{\text{nom}}$$

ist.

Der Steuervektor  $\underline{b}$  ist konstant :

$$\underline{b} = ( 0 \quad 0 \quad 6.6667 )'$$

### 5.1.2 Das offene System

Das offene System wurde für die drei Systemmatrizen  $A_1$ ,  $A_2$  und  $A_3$  simuliert, wobei für die beiden Figuren 5.1 und 5.2 folgendes gilt :

$$\begin{aligned} \text{Kurve 1 : } \dot{\underline{x}} &= A_1 \underline{x} \quad \text{bei} \quad \underline{\alpha}_{\max} = \underline{\alpha}_{\text{nom}} \\ \text{Kurve 2 : } \dot{\underline{x}} &= A_2 \underline{x} \quad \text{bei} \quad \underline{\alpha}_1 \quad \text{mit} \quad \underline{\alpha}_{\min} < \underline{\alpha}_1 < \underline{\alpha}_{\max} \quad (*) \\ \text{Kurve 3 : } \dot{\underline{x}} &= A_3 \underline{x} \quad \text{bei} \quad \underline{\alpha}_{\min} \end{aligned}$$

jeweils mit der Anfangsbedingung  $\underline{x}(0) = ( 1 \ 0 \ 0 )'$ . (Der 3. Zustand  $x_3(t)$  hat die Differentialgleichung

$$\dot{x}_3 = a_{33}x_3$$

und da  $x_3(0) = 0$  folgt  $x_3(t) = 0$  für alle  $t \geq 0$ ).

Wie man aus den beiden Figuren erkennen kann, ändern sich sowohl die Amplitude, als auch die Frequenz der beiden Verläufe  $x_1(t)$  und  $x_2(t)$  merklich für die verschiedenen Parametervektoren.

Als Simulationssprache wurde DARE-PORTABLE (Version 4.1) verwendet. Sie besitzt bessere Darstellungsmöglichkeiten als CSMP und gegenüber RK 58 hat sie den Vorteil, dass man die Integrationsmethode - aus einer grossen Anzahl - selber wählen kann. Letzteres war unentbehrlich bei der Simulation der hydraulischen Werkzeugmaschine, siehe 5.3. Wie man in der nächsten Seite erkennen kann, ist diese Sprache auch sehr einfach zu programmieren.

DARE PORTABLE (VERSION 4.1)

```

$D1
*           SIMULATION DES OFFENEN SYSTEMS
*           *****
*
*           X1.=A11*X1+X2+A13*X3
*           X2.=A21*X1+A22*X2+A23*X3
*           X3.=A33*X3+6.6667*U
*           U=R1*X1+R2*X2+R3*X3
*
*           Y1.=B11*Y1+Y2+B13*Y3
*           Y2.=B21*Y1+B22*Y2+B23*Y3
*           Y3.=B33*Y3+6.6667*V
*           V=R1*Y1+R2*Y2+R3*Y3
*
*           Z1.=C11*Z1+Z2+C13*Z3
*           Z2.=C21*Z1+C22*Z2+C23*Z3
*           Z3.=C33*Z3+6.6667*W
*           W=R1*Z1+R2*Z2+R3*Z3
*
END
*
*
*           A11=-0.0016, A13=-0.0002, A21=-0.1569, A22=-0.0015
*           A23=-0.1131, A33=-4.6667, X1=1., X2=0., X3=0.
*
*           B11=-0.0378, B13=-0.0006, B21=-4.078, B22=-0.028
*           B23=-3.157, B33=-5.6667, Y1=1., Y2=0., Y3=0.
*
*           C11=-0.074, C13=-0.012, C21=-8.0, C22=-0.055
*           C23=-5.2, C33=-6.6667, Z1=1., Z2=0., Z3=0.
*
*
*           R1=0., R2=0., R3=0.
*
*           NPOINT=151
*           TMAX=22.
END
*
*
*           GRAPH,X3,Y3,Z3
*
*
END
```

Die Zeiteinheit bei allen Simulationen in diesem Kapitel ist die Sekunde.

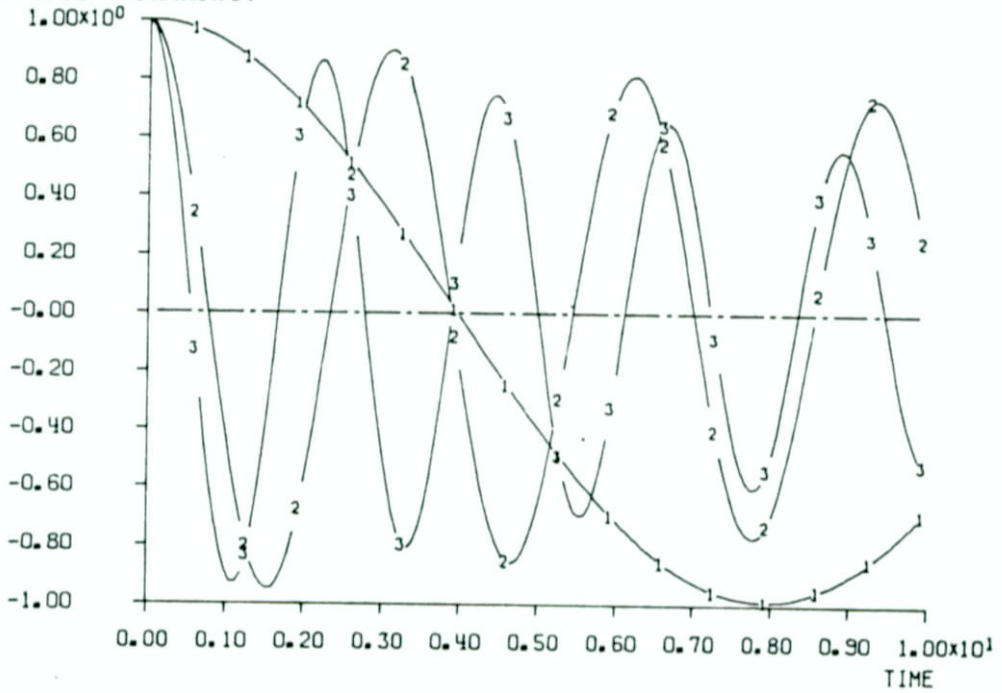


Fig. 5.1 Zustand  $x_1(t)$

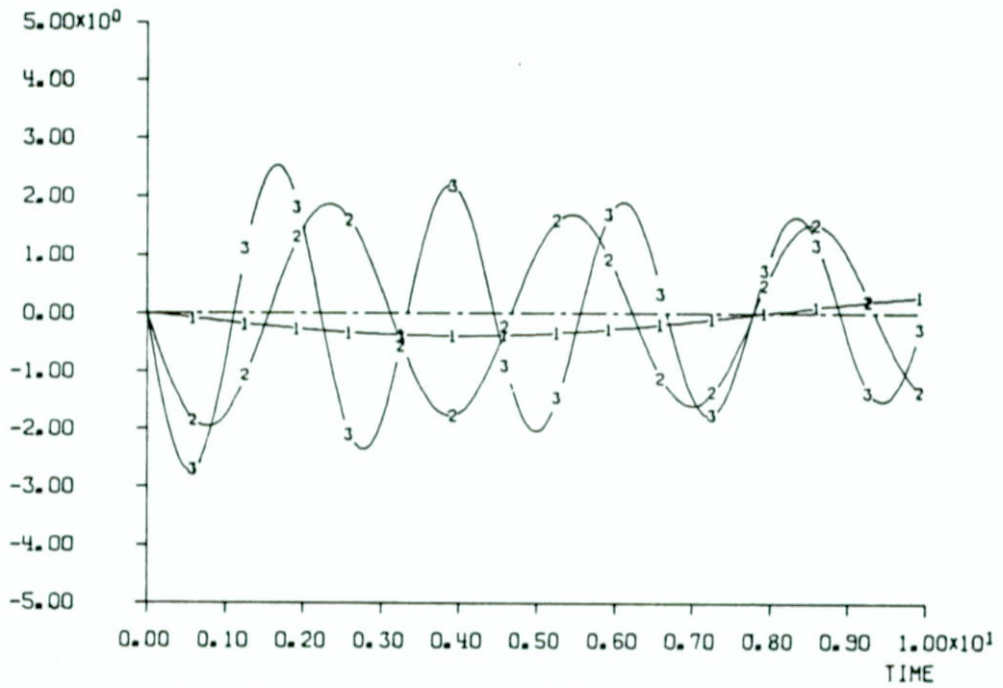


Fig. 5.2 Zustand  $x_2(t)$

Bemerkung : In [7] sind die Kurven 2 und 3 - bei Fig. 3 - vertauscht!

### 5.1.3 Das suboptimale, robuste System

Im Intervall  $[1, 100]$  wurde  $\lambda_{\text{krit}}$  gesucht. Es ergab sich ein Wert von

$$\lambda_{\text{krit}} = 8.9235$$

und die damit berechnete Rückführungsmatrix ist

$$L(\underline{\alpha}_{\text{nom}}, \lambda_{\text{krit}}) = ( -1.98 \quad -5.99 \quad 0.13 )'$$

für die auf der nächsten Seite angegebenen  $Q$ ,  $R$ ,  $\bar{A}$  und  $\bar{B}$  Matrizen.

#### Bemerkung

Da  $u = -Lx$  angenommen wurde, ist das Vorzeichen von  $L$  umgekehrt wie in [7].

Auf den nächsten Seiten sind auch die Simulationen zu finden. Für alle drei Figuren 5.3, 5.4 und 5.5 gilt :

$$\begin{array}{lll} \text{Kurve 1} & : & \dot{\underline{x}} = (A_1 - BL)\underline{x} \quad \text{bei} \quad \underline{\alpha}_{\text{max}} = \underline{\alpha}_{\text{nom}} \\ \text{Kurve 2} & : & \dot{\underline{x}} = (A_2 - BL)\underline{x} \quad \text{bei} \quad \underline{\alpha}_1 \text{ mit } \underline{\alpha}_{\text{min}} < \underline{\alpha}_1 < \underline{\alpha}_{\text{max}} \quad (**) \\ \text{Kurve 3} & : & \dot{\underline{x}} = (A_3 - BL)\underline{x} \quad \text{bei} \quad \underline{\alpha}_{\text{min}} \end{array}$$

und

$$\underline{x}(0) = ( 1 \quad 0 \quad 0 )'$$

Die Kurven unterscheiden sich leicht von denen in [7]; auch die  $L$  Matrix ist etwas verschieden. Der Grund dafür ist, dass wir mit einer grösseren Genauigkeit gearbeitet haben als Krishnan und Brzezowski. Als Beleg dafür vergleiche die Werte der beiden Zielfunktionen, die im Anschluss angegeben werden.

Berechnung von  $\lambda_{krit}$  und  $L(\alpha_{nom}, \lambda_{krit})$

ANZAHL STEUERVARIABLEN : 1  
 ANZAHL PARAMETER : 6

SR : 0  
 NGB : 0  
 LAMIN : 1.00  
 LAMAX : 100.00  
 RHO : 5.00

Q-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

.....  
 COLUMN 1 2 3

ROW 1 1.00000000 0. 0.  
 ROW 2 0. -500000000 0.  
 ROW 3 0. 0. 0.

R-MATRIX 1 ROWS 1 COLUMNS STORAGE MODE 0

.....  
 COLUMN 1

ROW 1 1.00000000

A-QUER-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

.....  
 COLUMN 1 2 3

ROW 1 -.160000000E-02 1.00000000 -.200000000E-03  
 ROW 2 -.156900000 -.150000000E-02 -.113100000  
 ROW 3 0. 0. -4.66700000

B-QUER-MATRIX 3 ROWS 1 COLUMNS STORAGE MODE 0

.....  
 COLUMN 1

ROW 1 0.  
 ROW 2 0.  
 ROW 3 6.66670000

ALMIIH-VEKTOR 1 ROWS 6 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5 6  
 ROW 1 --.740000000E-01 -8.000000000 --.550000000E-01 --.120000000E-01 -6.200000000 -6.667000000

ALMAX=ALPHA-OEER 1 ROWS 6 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5 6  
 ROW 1 --.160000000E-02 --.156900000 --.150000000E-02 --.200000000E-03 --.113100000 -4.667000000

ALMIG-VEKTOR 1 ROWS 6 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5 6  
 ROW 1 --.500000000E-01 -5.000000000 --.400000000E-01 --.600000000E-02 -1.200000000 -5.800000000

ALMAG-VEKTOR 1 ROWS 6 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5 6  
 ROW 1 --.160000000E-02 --.156900000 --.150000000E-02 --.200000000E-03 --.113100000 -4.667000000

\*\*\*\*\*  
 \* LAM90A-KRITISCH = 8.92348 \*  
 \*\*\*\*\*

L-PATFIX 1 ROWS 3 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3  
 ROW 1 -1.97792681 -5.99176332 .132709941

6 1\*  
 7  
 8 )  
 9  
 10  
 11 )  
 12 2\*  
 13  
 14 )  
 15  
 16  
 17 )  
 18 3\*  
 19  
 20 )  
 21  
 22  
 23 )  
 24 4\*  
 25  
 26 )  
 27  
 28  
 29 )  
 30 5\*  
 31  
 32 )  
 33  
 34 )  
 35 )  
 36 6\*  
 37  
 38 )  
 39  
 40 )  
 41 )  
 42 7\*  
 43  
 44 )  
 45 )  
 46 )  
 47 )  
 48 8\*  
 49  
 50 )  
 51 )  
 52 )  
 53 )  
 54 9\*  
 55 )  
 56 )  
 57 )  
 58 )  
 59 )  
 60 10\*  
 61 )  
 62 )  
 63 )  
 64 )  
 65 11\*  
 66 )  
 67 )

Simulationen

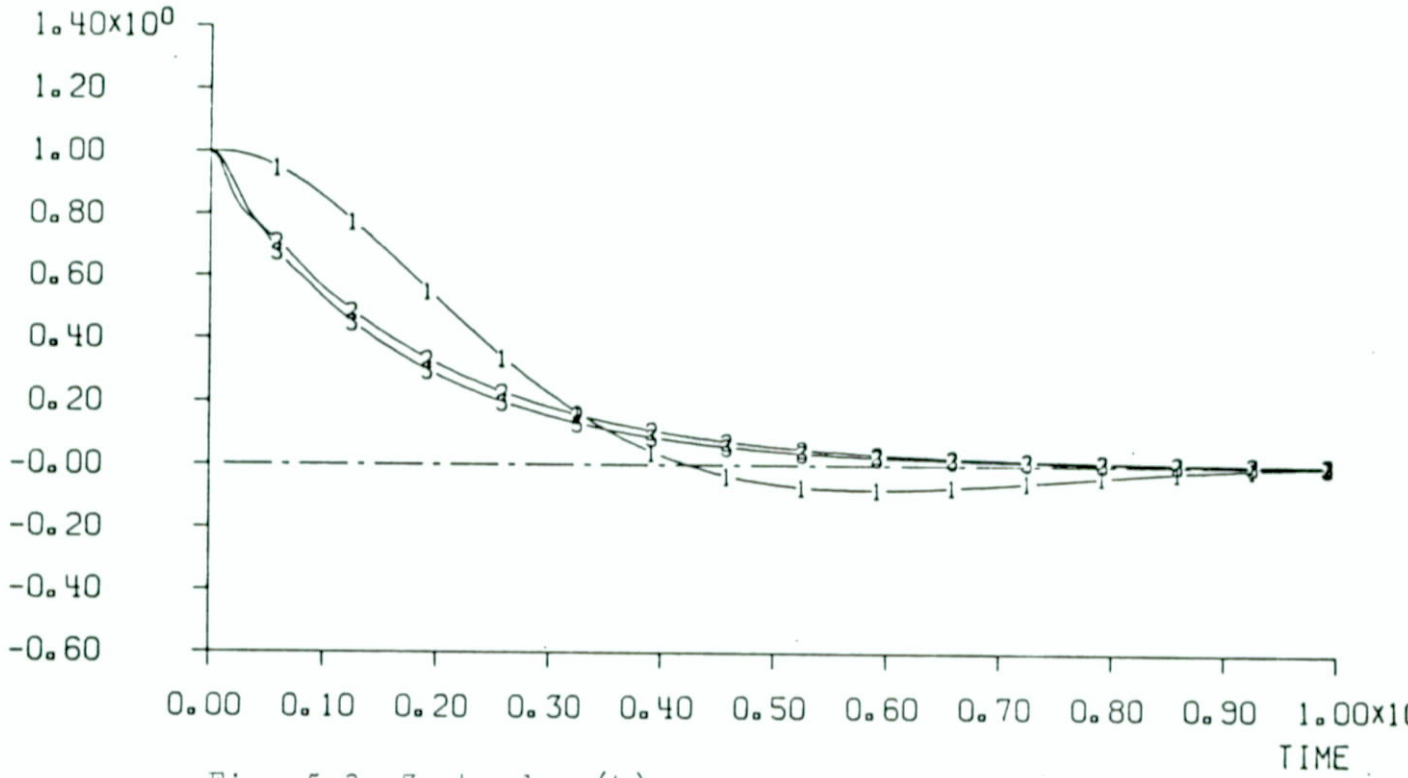


Fig. 5.3 Zustand  $x_1(t)$

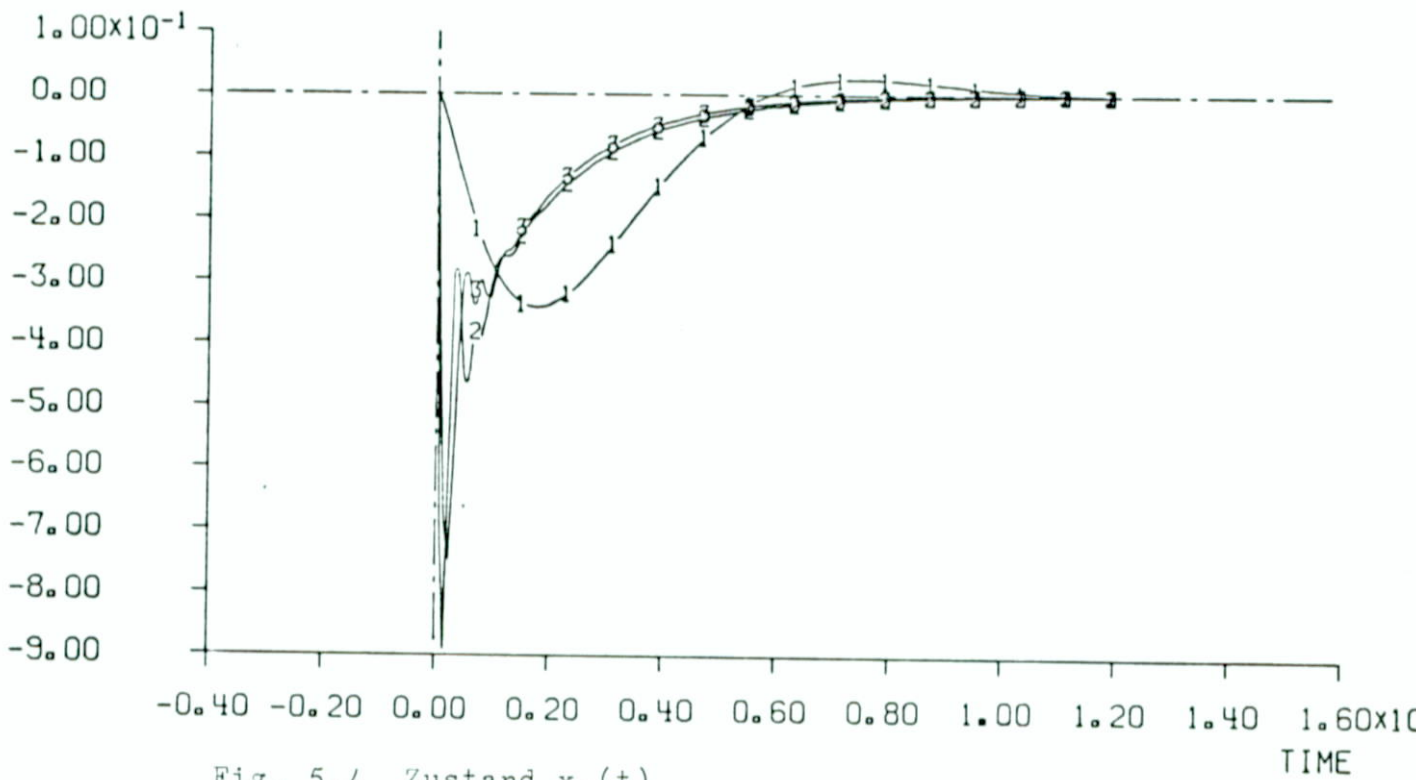


Fig. 5.4 Zustand  $x_2(t)$

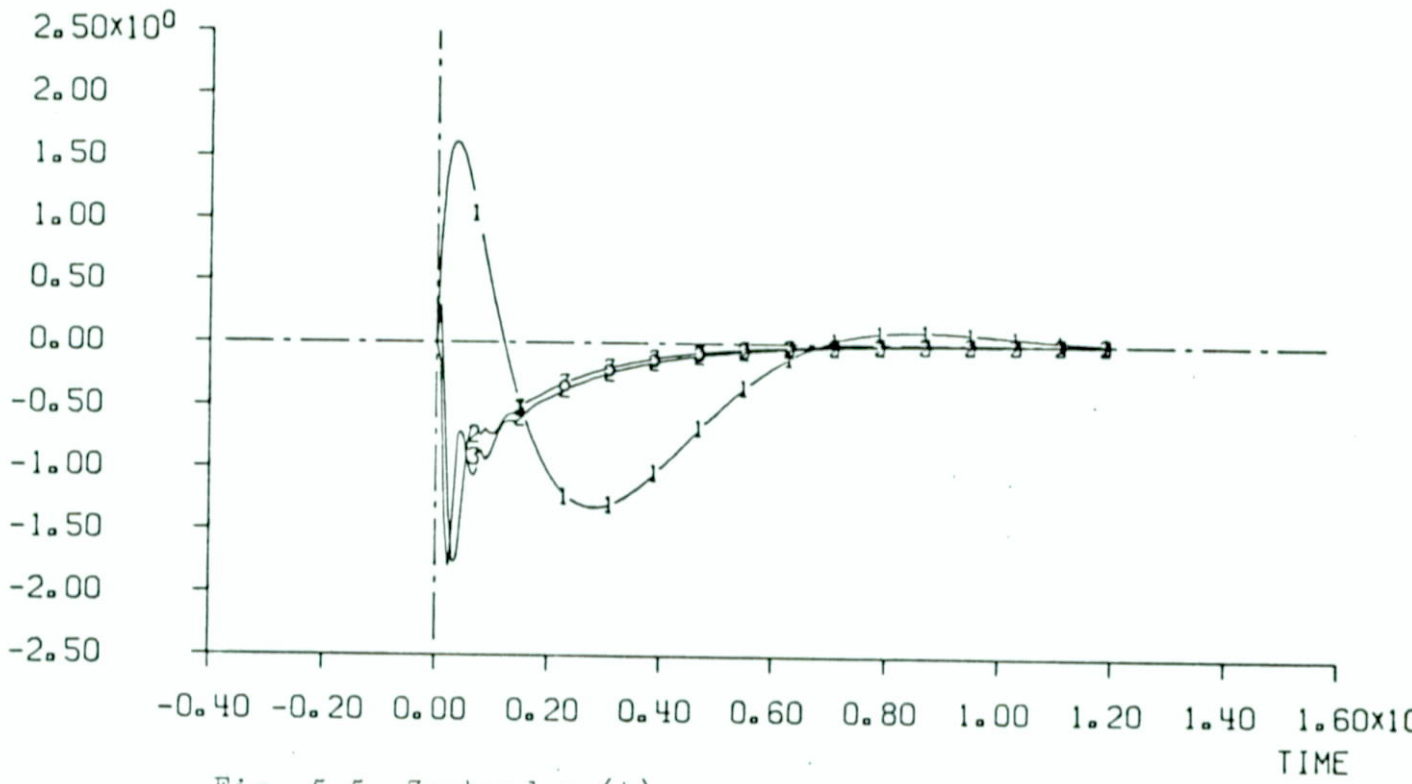


Fig. 5.5 Zustand  $x_3(t)$

Mit dem, auf den nächsten Seiten, angegebenen Programm ZF wurde die Zielfunktion

$$J = \int_0^{\infty} \underline{u}' R \underline{u} \, dt \quad \text{bei } \underline{\alpha} = \underline{\alpha}_{\text{nom}}$$

nach der in Kapitel 3 hergeleiteten Formel

$$J = \underline{x}'(0) P \underline{x}(0)$$

berechnet.

Für das in [7] angegebene L ist

$$J_1 = 3.36$$

Hingegen, wenn man das von uns berechnete L verwendet, erhält man

$$J_2 = 3.15$$

Aus  $J_1 > J_2$  folgt, dass unsere Berechnungen ( $\lambda_{\text{krit}}$  und L) genauer sind!

```
PROGRAM ZF(INPUT,OUTPUT,TAPES=INPUT,TAPE6=OUTPUT)
REAL      Q(100),R(100),AQ(100),BQ(100),L(100),
$         F(100),LT(100),QL(100),C(100),FF(121),
$         U(100),XO(10),XOT(10),PI(1),LTR(100),XOTC(100)

READ(5,100) N,NR
00  FORMAT(2I4)
    NQUA=N*N
    NRQUA=NR*NR
    NNR=N*NR
01  READ(5,101) (Q(I), I=1,NQUA)
    FORMAT(16F4.2)
02  READ(5,102) (R(I), I=1,NRQUA)
    FORMAT(10F7.3)
03  READ(5,103) (AQ(I), I=1,NQUA)
    FORMAT(10F8.3)
04  READ(5,104) (BQ(I), I=1,NNR)
    FORMAT(10F11.3)
05  READ(5,105) (L(I), I=1,NNR)
    FORMAT(5F16.8)
06  READ(5,106) (XO(I), I=1,N)
    FORMAT(10F7.3)

WRITE(6,200)
00  FORMAT(1H1,40X,*DAS ELEKTROMECHANISCHE SYSTEM MIT LAMBDA=1 (OPTIMA
$L)*)
WRITE(6,201)
01  FORMAT(41X,******)
WRITE(6,202) N
02  FORMAT(/,1X,*SYSTEMORDNUNG                =*,I3)
WRITE(6,203) NR
03  FORMAT(/,1X,*ANZAHL STEUERVARIABLEN =*,I3)
CALL MOUT(20HQ-MATRIX           ,Q,N,N,0)
CALL MOUT(20HR-MATRIX           ,R,NR,NR,0)
CALL MOUT(20HA-QUER-MATRIX      ,AQ,N,N,0)
CALL MOUT(20HB-QUER-MATRIX      ,BQ,N,NR,0)
CALL MOUT(20HL-MATRIX           ,L,NR,N,0)
CALL MOUT(20HXO-VEKTOR          ,XO,N,1,0)

DAS PRODUKT F=B-QUER*L WIRD GEBILDET.

CALL GMPRD(BQ,L,F,N,NR,N)

DIE SYSTEMMATRIX DES GESCHLOSSENEN KREISES
    F=A-B*L
WIRD BERECHNET FUER ALPHA-NOMINAL.

CALL GMSUB(AQ,F,F,N,N)
```

NUN WIRD  $Q=L'R*L$  GEBILDET

```
CALL GMTRA(L,LT,NR,N)
CALL GMPRD(LT,R,LTR,N,NR,NR)
CALL GMPRD(LTR,L,QL,N,NR,N)
```

BERECHNUNG DER C-MATRIX, ALS LOESUNG DER LJAPUNOV-MATRIX-GLEICHUNG  
MIT HILFE DER AUTLIB SUBROUTINE ATXPA.

```
DO 1 II=1,NQUA
C(II)=-QL(II)
```

DIE MATRIX  $F(N,N)$  MUSS FUER DEN INTERNEN GEBRAUCH IN DIESER AUTLIB  
SUBROUTINE UMGESCHRIEBEN WERDEN ZU  $FF(N+1,N+1)$ .

```
EPS=1.E-20
NA=N+1
IIX=1
DO 2 II=1,NQUA,NA
NE=II+N-1
DO 2 III=II,NE
FF(III)=F(IIX)
IIX=IIX+1

CALL ATXPA(FF,U,C,N,NA,N,N,EPS,IER)

IF(IER.NE.0) GOTO 3
```

DER WERT DER ZIELFUNKTION BEI DEN NENNWERTEN DER PARAMETER WIRD NACH

$$PI=XO'^*P^*XO$$

BERECHNET.

```
CALL GMTRA(XO,XOT,N,1)
CALL GMPRD(XOT,C,XOTC,1,N,N)
CALL GMPRD(XOTC,XO,PI,1,N,1)
WRITE(6,205) PI
05 FORMAT(////////,* PERFORMANCE INDEX =*,E12.5)
WRITE(6,206)
06 FORMAT(* ******)
GOTO 4
WRITE(6,207)
07 FORMAT(///,10X,*LJAPUNOV-GLEICHUNG UNLOESBAR*)
CONTINUE
STOP
END
```

Berechnung der Zielfunktion

SYSTEMORDNUNG = 3

ANZAHL STEUERVARIABLEN = 1

Q-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN 1 2 3

ROW 1 1.00000000 0. 0.  
ROW 2 0. 500000000 0.  
ROW 3 0. 0. 0.

R-MATRIX 1 ROWS 1 COLUMNS STORAGE MODE 0

COLUMN 1

ROW 1 1.00000000

A-QUER-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN 1 2 3

ROW 1 -.160000000E-02 1.00000000 -.200000000E-03  
ROW 2 -.156900000 -.150000000E-02 -.113100000  
ROW 3 0. 0. -4.66700000

B-QUER-MATRIX 3 ROWS 1 COLUMNS STORAGE MODE 0

COLUMN 1

ROW 1 0.  
ROW 2 0.  
ROW 3 6.66670000



SYSTEMORDNUNG = 3

ANZAHL STEUERVARIABLEN = 1

Q-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

.....

COLUMN 1 2 3

ROW 1 1.00000000 0. 0.  
ROW 2 0. .500000000 0.  
ROW 3 0. 0. 0.

R-MATRIX 1 ROWS 1 COLUMNS STORAGE MODE 0

.....

COLUMN 1

ROW 1 1.00000000

A-QUEP-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

.....

COLUMN 1 2 3

ROW 1 -.160000000E-02 1.00000000 -.200000000E-03  
ROW 2 -.156900000 -.150000000E-02 -.113100000  
ROW 3 0. 0. -4.66700000

B-QUEP-MATRIX 3 ROWS 1 COLUMNS STORAGE MODE 0

.....

COLUMN 1

ROW 1 0.  
ROW 2 0.  
ROW 3 6.66670000

7  
8  
9  
10  
11  
12 2'  
13  
14  
15  
16  
17  
18 3'  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30 5'  
31  
32  
33  
34  
35  
36 6'  
37  
38  
39  
40  
41  
42 7'  
43  
44  
45  
46  
47  
48 8'  
49  
50  
51  
52  
53  
54 9'  
55  
56  
57  
58  
59  
60 10'

```

61'
7
8
9
10
11
12 2'
13
14
15
16
17 1'
18 3'
19
20
21
22
23
24 4'
25
26
27
28
29
30 5'
31
32
33
34
35
36 6'
37
38
39
40
41
42 7'
43
44
45
46
47
48 8'
49
50
51
52
53
54 9'
55
56
57
58
59
60 10'
61
62
63
64
65
66 11'
67

```

L-MATRIX  
.....  
1 ROWS      3 COLUMNS      STORAGE MODE 0  
.....  
COLUMN 1                      2                      3  
ROM 1 -2.13000000      -6.42000000      +1.00000000

```

X0-VEKTOR  

.....  

3 ROWS      1 COLUMNS      STORAGE MODE 0  

.....  

COLUMN 1  

ROM 1 1.00000000  

ROM 2 0.  

ROM 3 0.
```

```

PERFORMANCE INDEX = .33557E+01  

*****
```

#### 5.1.4 Das optimal-geregelte System

Mit der Subroutine GEBIET wurden die neuen Stabilitäts- bzw. Robustheitsgebiete für  $\lambda = 1$  gesucht. Dabei hat sich folgendes gezeigt : das ganze angegebene Gebiet ist ein Stabilitätsgebiet, dagegen konnte kein Robustheitsgebiet gefunden werden, da das System nicht einmal bei  $\underline{\alpha} = \underline{\alpha}_{nom}$  (für  $\lambda = 1$ ) robust ist.

Für  $\lambda = 1$  erhält man die folgende Rückführungsmatrix

L-MATRIX		1 ROWS	3 COLUMNS	STORAGE MODE 0
.....				
COLUMN	1	2	3	
ROW 1	-.340296710	-2.45340253	.571384520E-01	

#### Bemerkung

Das System ist für  $\lambda = 1$  *nur* für  $\underline{\alpha} = \underline{\alpha}_{nom}$  optimal. Für die beiden anderen Parameterpunkte  $\underline{\alpha}_1$  und  $\underline{\alpha}_{min}$  ist das System hingegen nicht optimal. Für alle drei Parameterpunkte  $\underline{\alpha}_{nom}$ ,  $\underline{\alpha}_1$  und  $\underline{\alpha}_{min}$  zeigt sich in den nachfolgenden Simulationen, dass das optimale System langsamer ist, als das suboptimale. Der "Preis" dafür ist die - im letzten Fall - höhere Steuerenergie.

Für die Simulationen gilt (\*\*), siehe 5.1.3.

Die Suche nach dem neuen Robustheitsgebiet

ANZAHL STEUERVARIABLEN # 1  
ANZAHL PARAMETER # 6

SR # 0  
NGEB # 1  
LAMIN # .50  
LAMAX # 1.00  
RHO # 5.00

Q-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

.....  
COLUMN 1 2 3

ROW 1 1.00000000 0. 0.  
ROW 2 0. -500000000 0.  
ROW 3 0. 0. 0.

R-MATRIX 1 ROWS 1 COLUMNS STORAGE MODE 0

.....  
COLUMN 1

ROW 1 1.00000000

A-QUER-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

.....  
COLUMN 1 2 3

ROW 1 --160000000E-02 1.00000000 --200000000E-03  
ROW 2 --156900000 --150000000E-02 --113100000  
ROW 3 0. 0. -4.66700000

B-QUER-MATRIX 3 ROWS 1 COLUMNS STORAGE MODE 0

.....  
COLUMN 1

ROW 1 0.  
ROW 2 0.  
ROW 3 6.66670000

```

7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70

```

```

ALMIII-VEKTOR      1 ROWS      6 COLUMNS      STORAGE MODE 0
.....
COLUMN 1          2          3          4          5          6
ROM 1  -.740000000E-01 -6.000000000  -.550000000E-01 -.120000000E-01 -6.200000000  -6.667000000

```

```

ALMAX=ALPHA-QUER      1 ROWS      6 COLUMNS      STORAGE MODE 0
.....
COLUMN 1          2          3          4          5          6
ROM 1  -.160000000E-02 -.156900000  -.150000000E-02 -.200000000E-03 -.113100000  -4.667000000

```

```

ALMIG-VEKTOR      1 ROWS      6 COLUMNS      STORAGE MODE 0
.....
COLUMN 1          2          3          4          5          6
ROM 1  -.160000000E-02 -.156900000  -.150000000E-02 -.200000000E-03 -.113100000  -4.667000000

```

```

ALMAG-VEKTOR      1 ROWS      6 COLUMNS      STORAGE MODE 0
.....
COLUMN 1          2          3          4          5          6
ROM 1  -.160000000E-02 -.156900000  -.150000000E-02 -.200000000E-03 -.113100000  -4.667000000

```

LAMBDA-MAX WURDE ZU KLEIN GEWAEHLT

DAS GESUCHTE FORBSTHEITSGEBIET IST KLEINER ALS DAS ANGEGEBENE MINIMALGEBIET

DA ALMIN = ALMAG = ALPHA-QUER, FOLGT ALPHA-QUER NICHT ROBUST

Simulationen

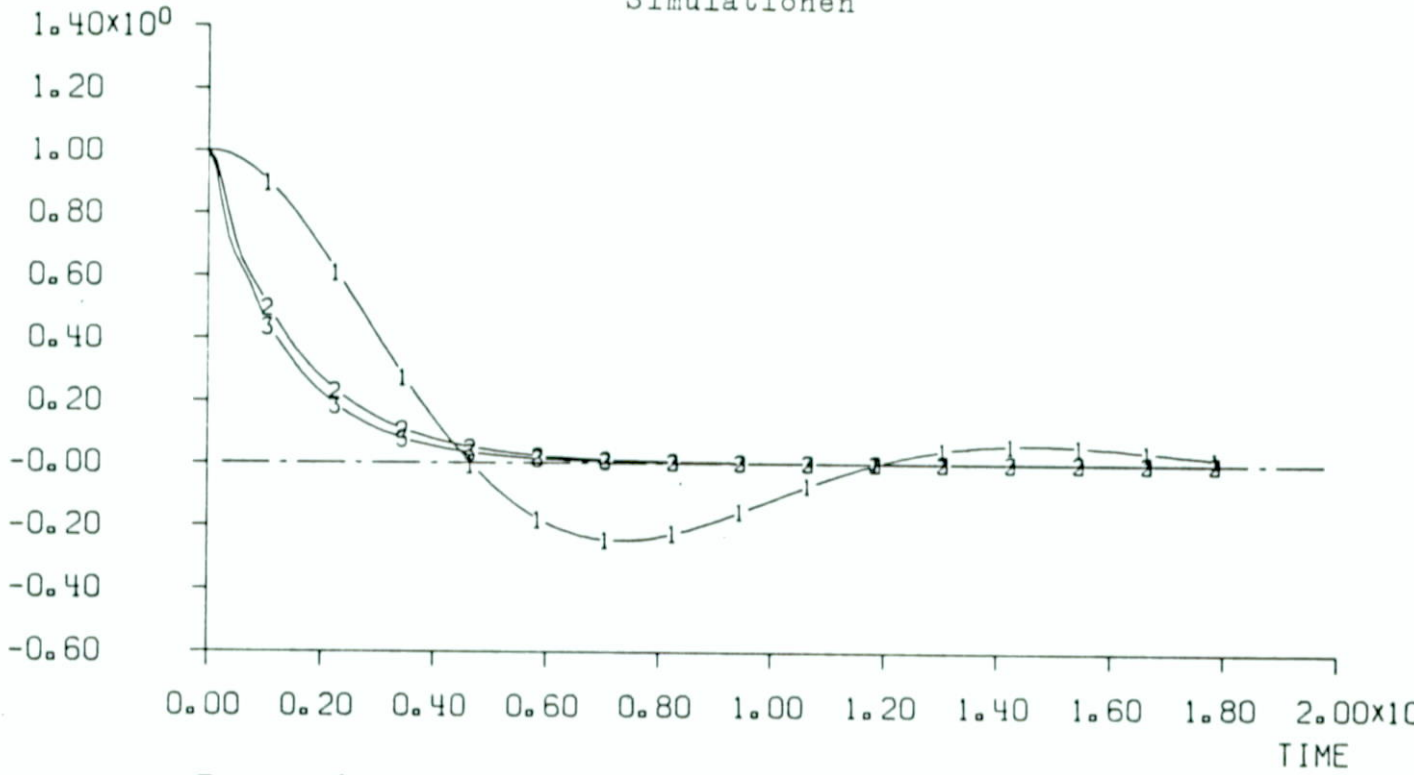


Fig. 5.6 Zustand  $x_1(t)$

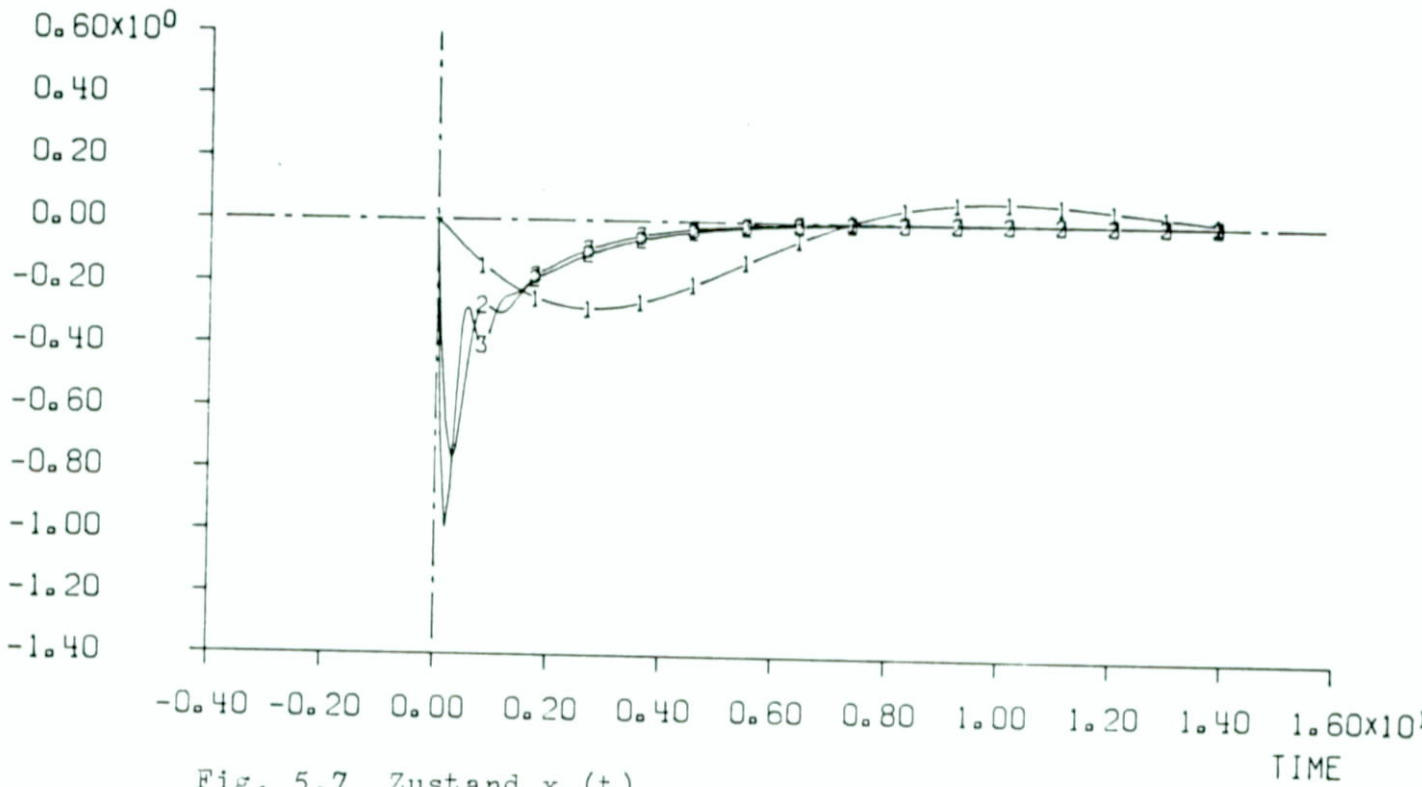


Fig. 5.7 Zustand  $x_2(t)$

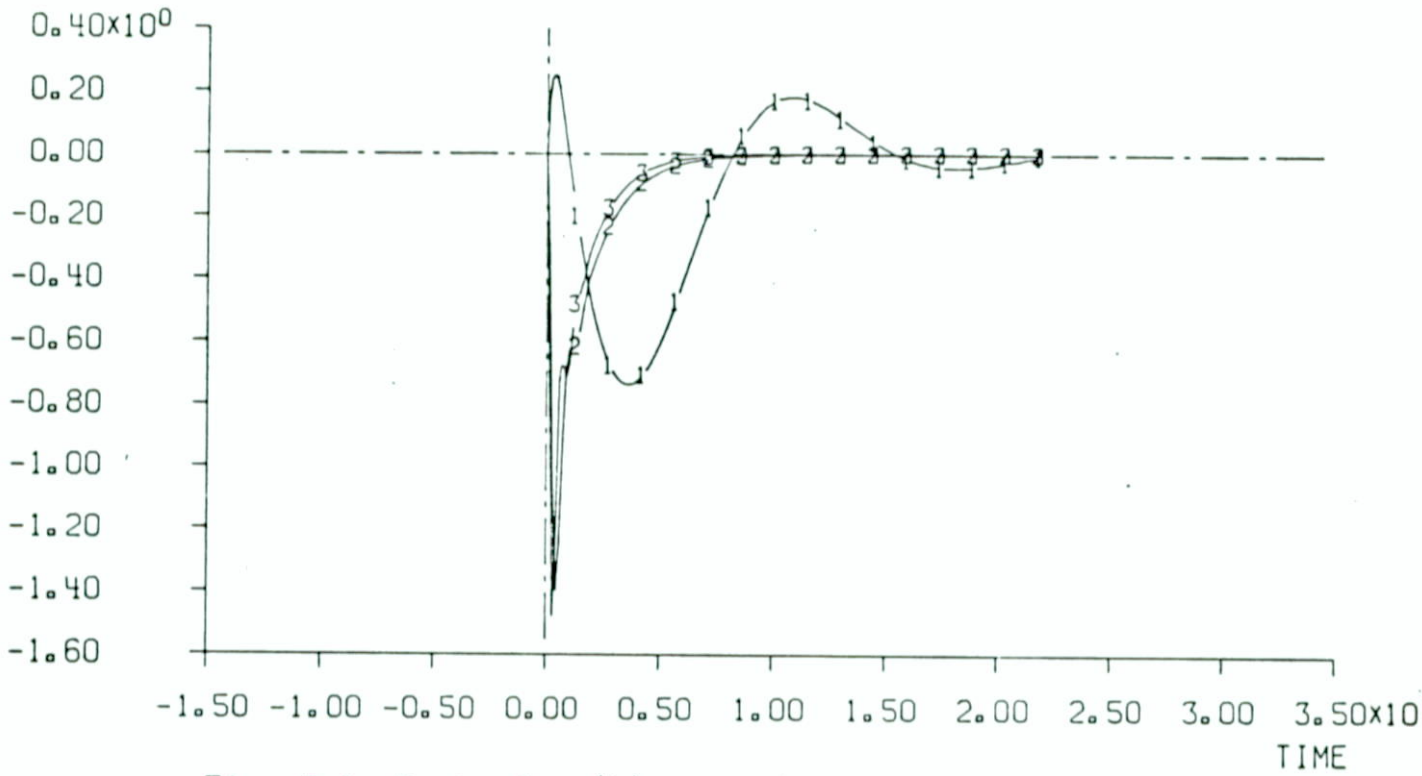


Fig. 5.8 Zustand  $x_3(t)$

Anschliessend wird die Zielfunktion für  $\lambda = 1$  berechnet. Man erhält einen Wert von

$$J = 1.00$$

welcher deutlich kleiner ist, als für  $\lambda = \lambda_{\text{krit}}$ .

Berechnung der Zielfunktion

SYSTEMORDNUNG = 3

ANZAHL STEUERVARIABLEN = 1

Q-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3  
 ROW 1 1.00000000 0. 0.  
 ROW 2 0. 500000000 0.  
 ROW 3 0. 0. 0.

R-MATRIX 1 ROWS 1 COLUMN STORAGE MODE 0  
 .....  
 COLUMN 1  
 ROW 1 1.00000000

A-QUER-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3  
 ROW 1 -.160000000E-02 1.00000000 -.200000000E-03  
 ROW 2 -.156900000 -.150000000E-02 -.113100000  
 ROW 3 0. 0. -4.66700000

B-QUER-MATRIX 3 ROWS 1 COLUMN STORAGE MODE 0  
 .....  
 COLUMN 1  
 ROW 1 0.  
 ROW 2 0.  
 ROW 3 6.66670000

6 1'  
7  
8  
9  
10  
11  
12 2'  
13  
14  
15  
16  
17  
18 3'  
19  
20  
21  
22  
23  
24 4'  
25  
26  
27  
28  
29  
30 5'  
31  
32  
33  
34  
35  
36 6'  
37  
38  
39  
40  
41  
42 7'  
43  
44  
45  
46  
47  
48 8'  
49  
50  
51  
52  
53  
54 9'  
55  
56  
57  
58  
59  
60 10'  
61  
62  
63  
64  
65  
66 11'  
67  
68

L-MATRIX  
.....

1 ROWS            3 COLUMNS    STORAGE MODE 0  
COLUMN    1            2            3  
ROW 1   -0.340296710   -2.45340253   .571384500E-01

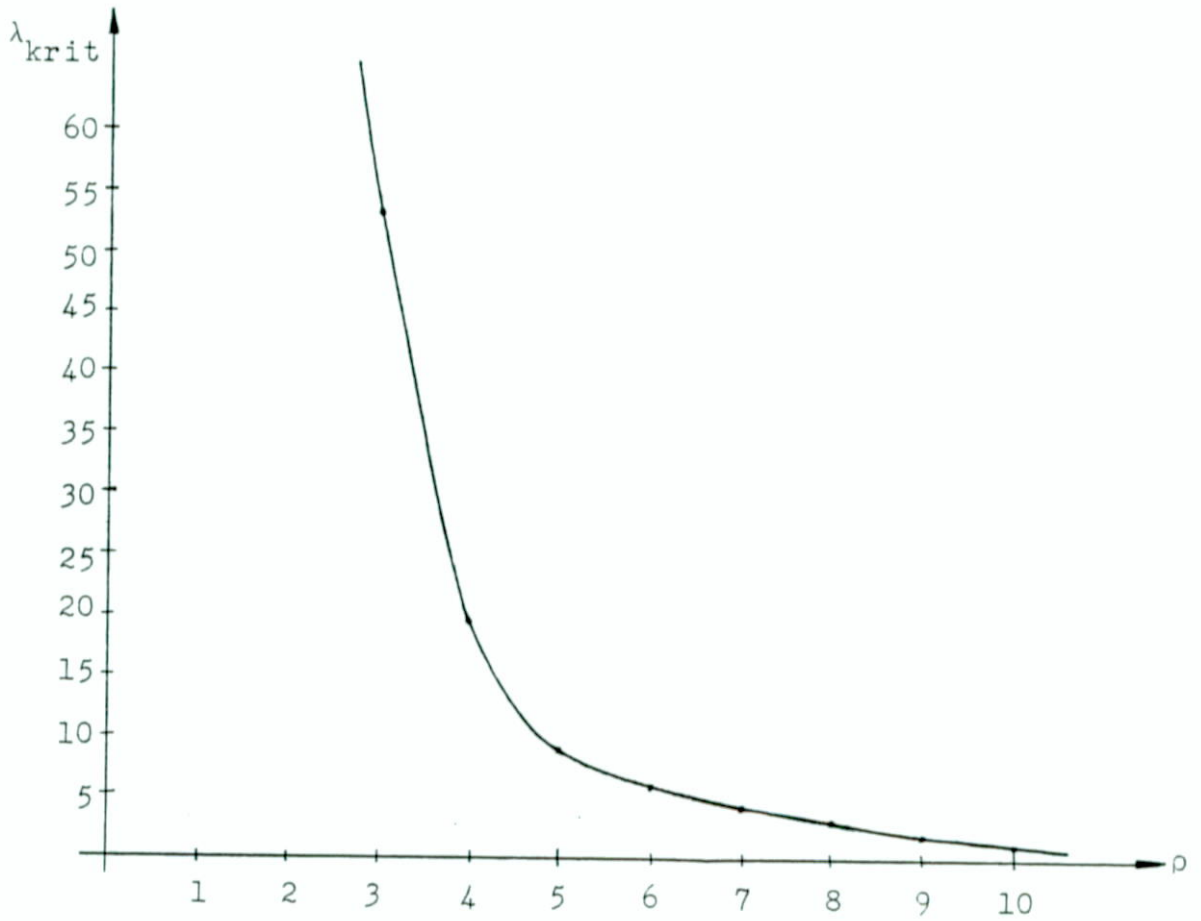
X0-VEKTOR  
.....

3 ROWS            1 COLUMNS    STORAGE MODE 0  
COLUMN    1  
ROW 1   1.000000000  
ROW 2   0.  
ROW 3   0.

PERFORMANCE INDEX = .10022E+01  
\*\*\*\*\*

|  
∞  
∞  
|

5.1.5 Die  $\lambda_{\text{krit}}(\rho)$ -Kurve



$\rho$	$\lambda_{\text{krit}}$	$\rho$	$\lambda_{\text{krit}}$
1	$>10^5$	7	4
2	256	8	3
3	53	9	2
4	19	10	1.5
5	8.92	50	0.06
6	6	100	0.02

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

## 5.2. Regelung eines elektromechanischen Systems

### 5.2.1 Systembeschreibung

Das zu regelnde System besteht aus : Motor, Getriebe, Feder und Schwungmasse. Siehe Fig. 5.9.

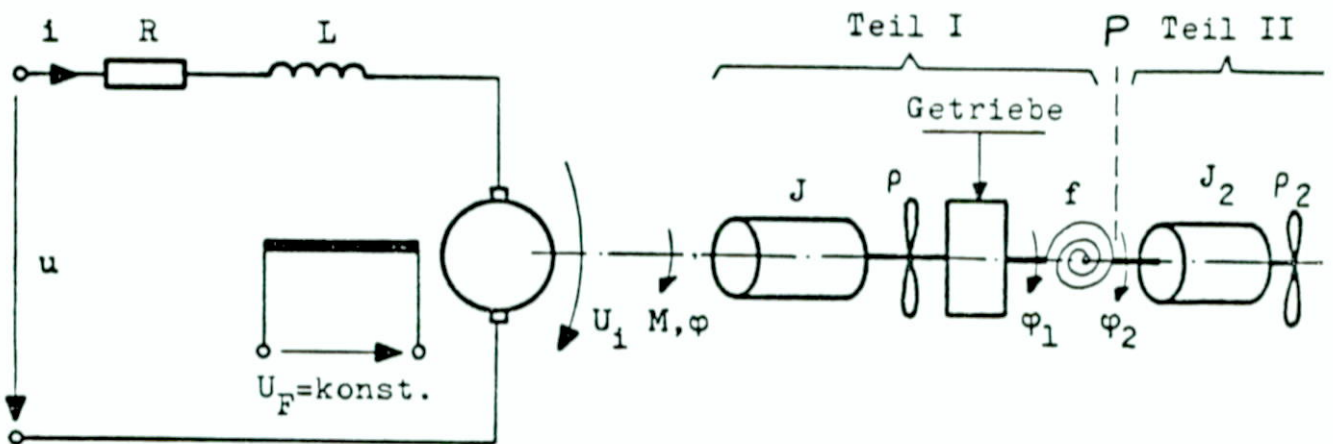


Fig. 5.9

Für die Modellierung werden folgende Annahmen getroffen :

- Die Feder sei linear im Arbeitsbereich (Hook'sches Gesetz)
- Das Getriebe habe kein Spiel
- Das Gleitreibungsmoment sei  $M = \rho_1 \dot{\phi}_1$
- Die Haftreibung, die Ankerinduktivität und die Ankerrückwirkung werden vernachlässigt

### Darstellung im Zustandsraum

$$\dot{\underline{x}} = \underline{A}\underline{x} + \underline{b}u \quad , \quad u : \text{Ankerspannung}$$

$$y = \underline{c}'\underline{x}$$

Mit den oben gemachten Annahmen und der Einführung der Zustandsvariablen

$$x_1 = \phi_1$$

$$x_2 = \dot{\phi}_1$$

$$x_3 = \phi_2$$

$$x_4 = \dot{\phi}_2$$

berechnet sich die Systemmatrix A und die Inputmatrix b zu :

Systemmatrix :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -f/J_1 & -\left\{\frac{(\psi\ddot{u})^2}{RJ_1} + \rho_1/J_1\right\} & f/J_1 & 0 \\ 0 & 0 & 0 & 1 \\ f/J_2 & 0 & -f/J_2 & -\rho_2/J_2 \end{bmatrix}$$

Steuervektor :

$$\underline{b} = \begin{bmatrix} 0 \\ \psi\ddot{u}/RJ_1 \\ 0 \\ 0 \end{bmatrix}$$

Messvektor :

$$\underline{c}' = ( 0 \quad 0 \quad 1.762947 \quad 0 )$$

(d.h. die Ausgangsgrösse  $y$  ist proportional zu  $\phi_2$ ).

Setzt man die Nennwerte der Parameter

$$\begin{aligned} f &= 0.4 \quad \text{Nm/rad} \\ J_1 &= 0.287 \quad \text{Kgm}^2 \\ \psi &= 0.06 \quad \text{Vs} \\ R &= 1.81 \quad \Omega \\ \rho_1 &= 4.6 \quad \text{Kgm}^2/\text{s} \\ J_2 &= 0.017 \quad \text{Kgm}^2 \\ \rho_2 &= 0.004 \quad \text{Kgm}^2/\text{s} \end{aligned}$$

in die Matrix  $A$  ein, so hat sie die folgende Eigenwerte :

$$\begin{aligned} \mu_1 &= -0.13443 + j4.85166 \\ \mu_2 &= -0.13443 - j4.85166 \\ \mu_3 &= -40.94281 \\ \mu_4 &= 0 \end{aligned}$$

Der Pol beim Nullpunkt bewirkt, dass das System instabil ist. Mit Hilfe einer Regelung möchte man dieses Uebel beseitigen.

Auf eine ausführlichere Beschreibung des Systems, im speziellen der physikalischen Zusammenhänge, wurde hier absichtlich verzichtet, da das behandelte Beispiel im Institut sehr bekannt ist und in mehreren Publikationen behandelt worden ist [11].

Da sowohl die Elemente der Systemmatrix  $A$  wie auch des Inputvektors  $\underline{b}$  multilineare Funktionen der zu variierenden Parametern sein müssen, werden  $J_1$ ,  $\psi$ ,  $R$  und  $J_2$  bei ihren Nominalwerten konstant gehalten, so dass sich nur die drei Parameter  $f$ ,  $\rho_1$  und  $\rho_2$  ändern. Die Systemmatrix  $A$  bekommt dann folgende neue Gestalt

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -f/0.287 & -24.949-\rho_1/0.287 & f/0.287 & 0 \\ 0 & 0 & 0 & 1 \\ f/0.017 & 0 & -f/0.017 & -\rho_2/0.017 \end{bmatrix}$$

und der Steuervektor  $\underline{b}$

$$\underline{b} = ( 0 \quad 6.93 \quad 0 \quad 0 )'$$

Die Grenzen der Parameter sind

$$0.2 \leq f \leq 0.6$$

$$3.5 \leq \rho_1 \leq 5.5$$

$$0.001 \leq \rho_2 \leq 0.007$$

mit

$$\underline{a}_{\text{nom}} = \begin{bmatrix} f \\ \rho_1 \\ \rho_2 \end{bmatrix} = \begin{bmatrix} 0.4 \\ 4.6 \\ 0.004 \end{bmatrix}$$

nom

### 5.2.2 Das offene System

In den nachfolgenden Simulationen bedeuten :

$$\text{Kurve 1 : } \dot{\underline{x}} = A_1 \underline{x} \quad \text{bei } \underline{\alpha}_{\text{nom}}$$

$$\text{Kurve 2 : } \dot{\underline{x}} = A_2 \underline{x} \quad \text{bei } \underline{\alpha}_{\text{min}}$$

$$\text{Kurve 3 : } \dot{\underline{x}} = A_3 \underline{x} \quad \text{bei } \underline{\alpha}_{\text{max}}$$

jeweils mit der Anfangsbedingungen

$$\underline{x}(0) = ( 1 \quad 1 \quad 1 \quad 1 )'$$

#### Bemerkungen

Aus darstellungstechnischen Gründen, lässt sich der Zustand  $x_1(t)$  nicht aufzeichnen, da der grösste und der kleinste Amplitudenwert sehr nahe bei 1 liegen.

Der Zustand  $x_4(t)$  wurde für die drei verschiedenen Parameterpunkte separat aufgezeichnet, weil sich sonst die Kurven überlappen würden.

Die Zustände  $x_1(t)$  und  $x_3(t)$  sind nur BIBO-stabil, - sie streben gegen 1 -, die Zustände  $x_2(t)$  und  $x_4(t)$  sind dagegen asymptotisch stabil isL.

DARE PORTABLE (VERSION 4.1)

```
SD1
*   DAS ELEKTROMECHANISCHE SYSTEM (OFFEN)
*   ****
*
*   X1.=X2
*   X2.=A21*X1+A22*X2-A21*X3+B2*U
*   X3.=X4
*   X4.=A41*X1-A41*X3+A44*X4
*
*   U=R1*X1+R2*X2+R3*X3+R4*X4
*
*
*   Y1.=Y2
*   Y2.=B21*Y1+B22*Y2-B21*Y3+B2*V
*   Y3.=Y4
*   Y4.=B41*Y1-B41*Y3+B44*Y4
*
*   V=R1*Y1+R2*Y2+R3*Y3+R4*Y4
*
*
*   Z1.=Z2
*   Z2.=C21*Z1+C22*Z2-C21*Z3+B2*W
*   Z3.=Z4
*   Z4.=C41*Z1-C41*Z3+C44*Z4
*
*   W=R1*Z1+R2*Z2+R3*Z3+R4*Z4
*
*
*   END
*
*   NENNWERTE :
*   A21=-1.3937, A22=-40.9769, A41=23.5294, A44=-0.2353, B2=6.93
*
*   MINIMALWERTE :
*   B21=-0.6369, B22=-37.1441, B41=11.7674, B44=-0.0588
*
*   MAXIMALWERTE :
*   C21=-2.0906, C22=-44.1128, C41=35.2941, C44=-0.1765
*
*
*   ANFANGSWERTE :
*   X1=1., X2=1., X3=1., X4=1.
*   Y1=1., Y2=1., Y3=1., Y4=1.
*   Z1=1., Z2=1., Z3=1., Z4=1.
*
*   RUECKFUEHRUNGSGEWICHTUNGEN :
*   R1=0.0, R2=0.0, R3=0.0, R4=0.0
*
*   NPOINT=151
*   TMAX=12.0
*
*
*   END
*
*   PLOT,X1
*
*
*   END
```

Simulationen

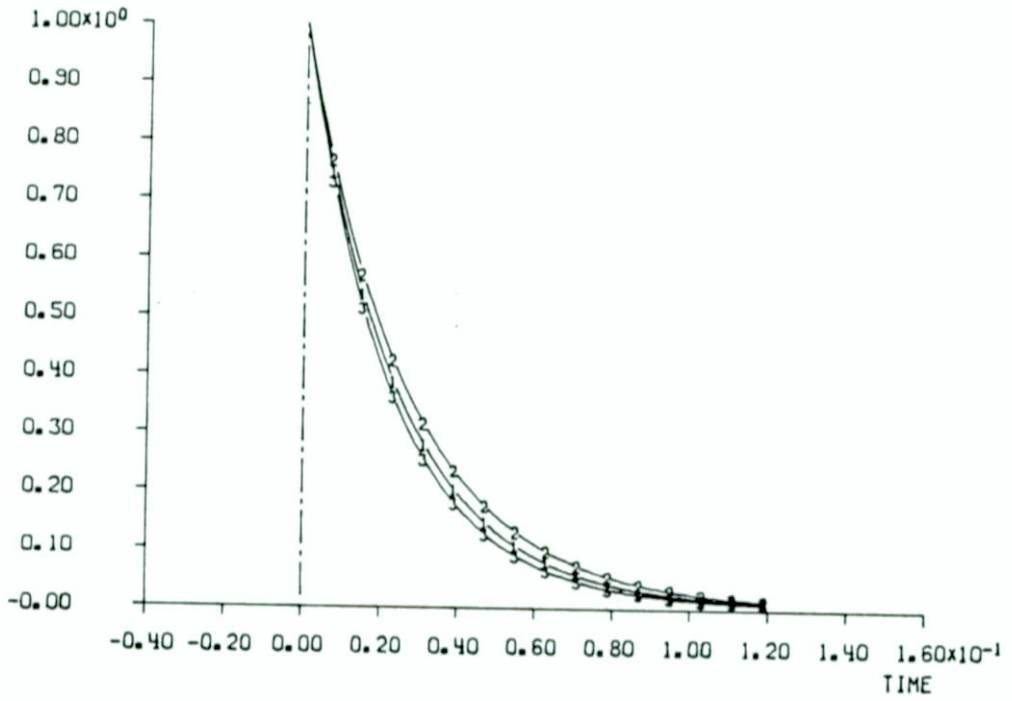


Fig. 5.10 Zustand  $x_2(t)$

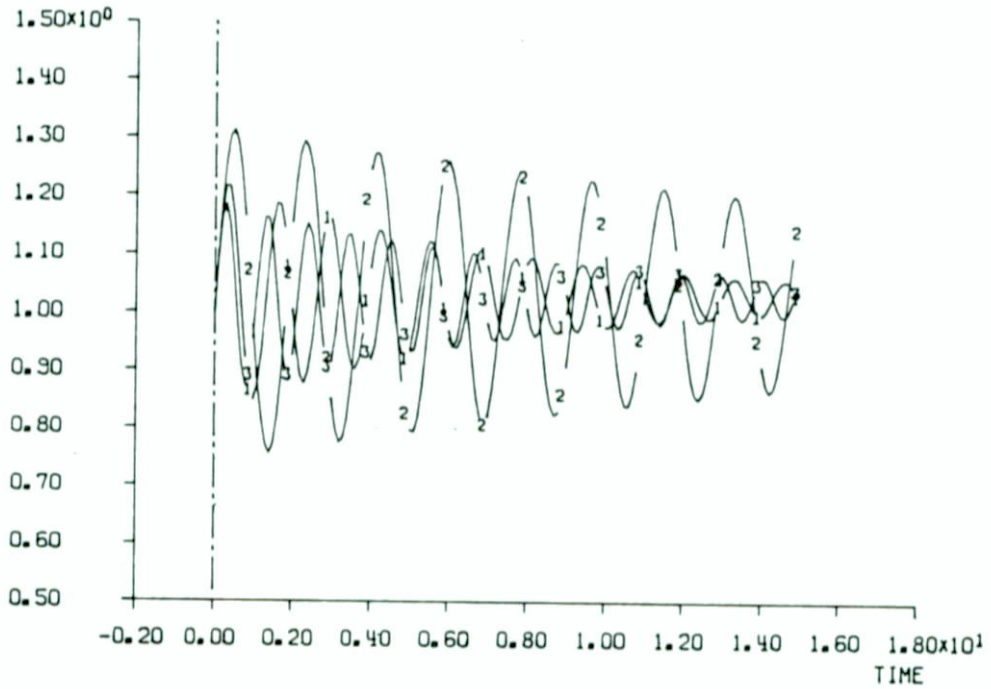


Fig. 5.11 Zustand  $x_3(t)$

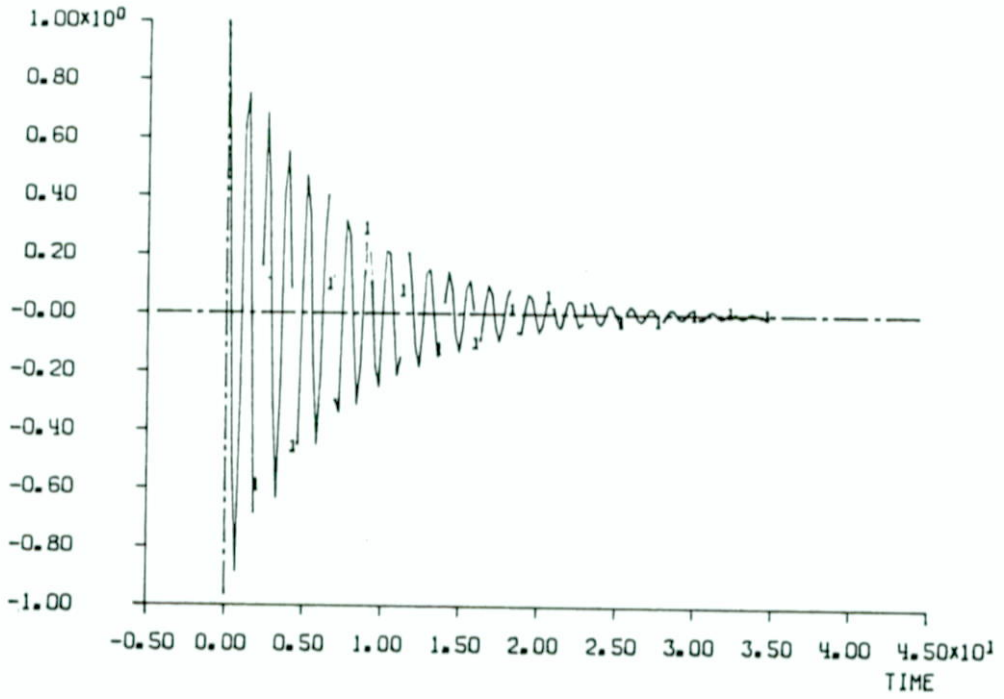


Fig. 5.12 Zustand  $x_4(t)$  mit  $\underline{\alpha}_{nom}$

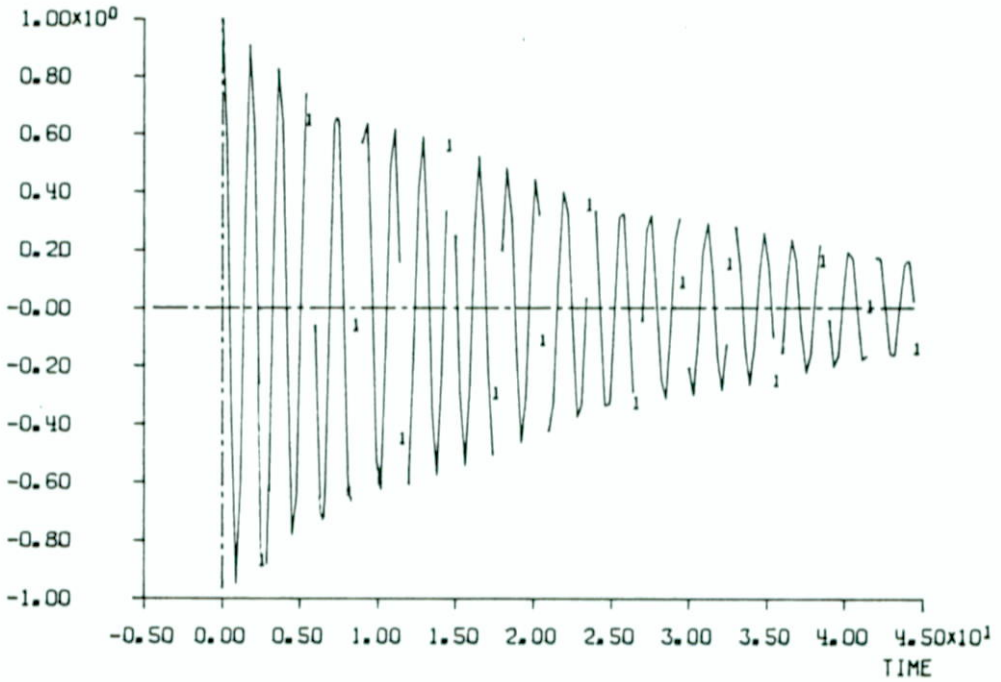


Fig. 5.13 Zustand  $x_4(t)$  mit  $\underline{\alpha}_{min}$

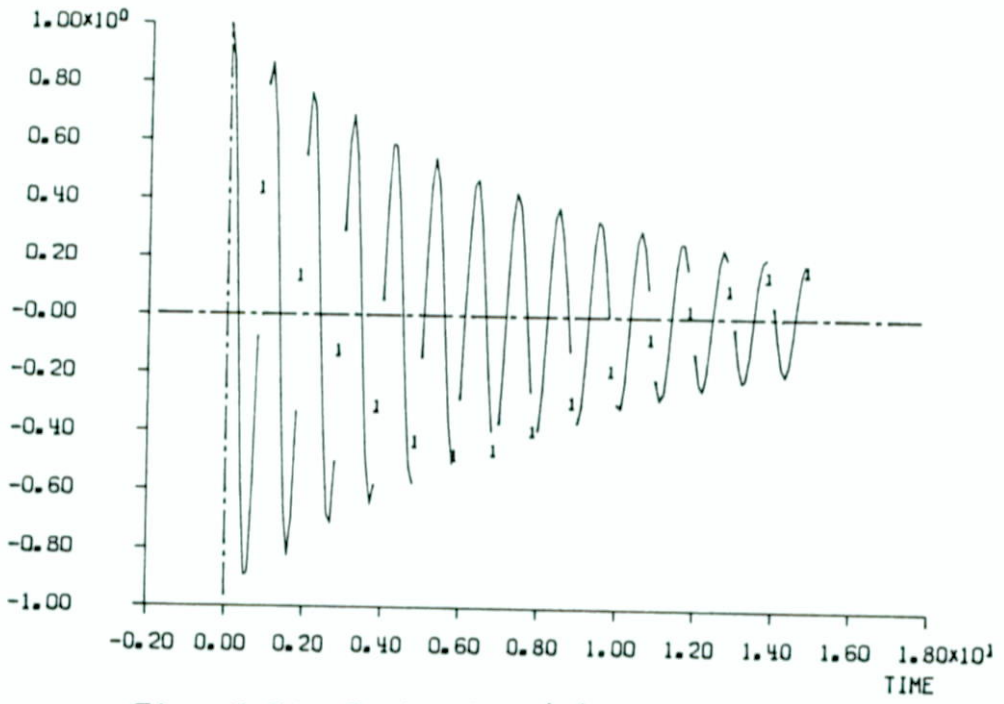


Fig. 5.14 Zustand  $x_4(t)$  bei  $\underline{\alpha}_{\max}$

### 5.2.3 Das suboptimale, robuste System

Das Programm ROBUST liefert die folgenden Werte

$$\lambda_{\text{krit}} = 14.168$$

und

$$L(\underline{\alpha}_{\text{nom}}, \lambda_{\text{krit}}) = ( 6.14 \quad 0.26 \quad -4.46 \quad 0.03 )'$$

für die auf der nächsten Seite angegebenen  $Q$ ,  $R$ ,  $\bar{A}$  und  $\bar{B}$  Matrizen.

Das System wurde anschliessend mit dieser Rückführungsmatrix simuliert, und zwar wieder an drei verschiedenen Parameterpunkte. Dabei gilt

$$\begin{array}{lll} \text{Kurve 1} & : & \dot{\underline{x}} = (A_1 - BL)\underline{x} \quad \text{bei} \quad \underline{\alpha}_{\text{nom}} \\ \text{Kurve 2} & : & \dot{\underline{x}} = (A_2 - BL)\underline{x} \quad \text{bei} \quad \underline{\alpha}_{\text{min}} \\ \text{Kurve 3} & : & \dot{\underline{x}} = (A_3 - BL)\underline{x} \quad \text{bei} \quad \underline{\alpha}_{\text{max}} \end{array}$$

für

$$\underline{x}(0) = ( 1 \quad 1 \quad 1 \quad 1 )'$$

Wie aus allen Kurven deutlich zu erkennen ist, wirken sich die Parameteränderungen bei den einzelnen Zustandsverläufen, sehr wenig aus. Für dieses System ist also die Methode besonders erfolgreich : das System wird sehr insensitiv, bezüglich Parameteränderungen.

Für die Zielfunktion wurde einen Wert von

$$J = 5.49$$

erhalten, wie im Abschluss zu sehen ist.

Berechnung von  $\lambda_{krit}$  und  $L(\alpha_{nom}, \lambda_{krit})$

SYSTEMORDNUNG : 4  
 ANZAHL STEUERVARIABLEN : 1  
 ANZAHL PARAMETER : 3

SR : 0  
 NGB : 0  
 LAMIN : 1.00  
 LAMAX : 20.00  
 RHO : 5.00

Q-MATRIX  
 .....  
 4 ROWS 4 COLUMNS STORAGE MODE 0

COLUMN	1	2	3	4
ROW 1	.100000000	0.	0.	0.
ROW 2	0.	.100000000	0.	0.
ROW 3	0.	0.	.100000000	0.
ROW 4	0.	0.	0.	.100000000

R-MATRIX  
 .....  
 1 ROWS 1 COLUMNS STORAGE MODE 0

COLUMN	1
ROW 1	1.000000000

A-QUER-MATRIX  
 .....  
 4 ROWS 4 COLUMNS STORAGE MODE 0

COLUMN	1	2	3	4
ROW 1	0.	1.000000000	0.	0.
ROW 2	-1.394000000	-40.976000000	1.394000000	0.
ROW 3	0.	0.	0.	1.000000000
ROW 4	23.529000000	0.	-23.529000000	-23500000000

B-QUER-MATRIX  
 .....  
 4 ROWS 1 COLUMNS STORAGE MODE 0

COLUMN	1
ROW 1	0.
ROW 2	5.930000000
ROW 3	0.

61\*  
 7  
 8  
 9  
 10  
 11  
 12\*  
 13  
 14  
 15  
 16  
 17  
 18\*  
 19  
 20  
 21  
 22  
 23  
 24\*  
 25  
 26  
 27  
 28  
 29  
 30\*  
 31  
 32  
 33  
 34  
 35  
 36\*  
 37  
 38  
 39  
 40  
 41  
 42\*  
 43  
 44  
 45  
 46  
 47  
 48\*  
 49  
 50  
 51  
 52  
 53  
 54\*  
 55  
 56  
 57  
 58  
 59  
 60\*  
 61  
 62  
 63  
 64  
 65  
 66\*  
 67

ALMIN-VEKTOR .....  
1 ROWS 3 COLUMNS STORAGE MODE 0  
COLUMN 1 2 3  
ROW 1 .200000000 3.50000000 .100000000E-02

ALMAX-VEKTOR .....  
1 ROWS 3 COLUMNS STORAGE MODE 0  
COLUMN 1 2 3  
ROW 1 .600000000 5.50000000 .700000000E-02

\*\*\*\*\*  
\* LAMBDA-KRITISCH = 14.16803 \*  
\*\*\*\*\*

L-MATRIX .....  
1 ROWS 4 COLUMNS STORAGE MODE 0  
COLUMN 1 2 3 4  
ROW 1 5.13872239 .263737974 -4.45539059 .290246443E-01

Simulationen

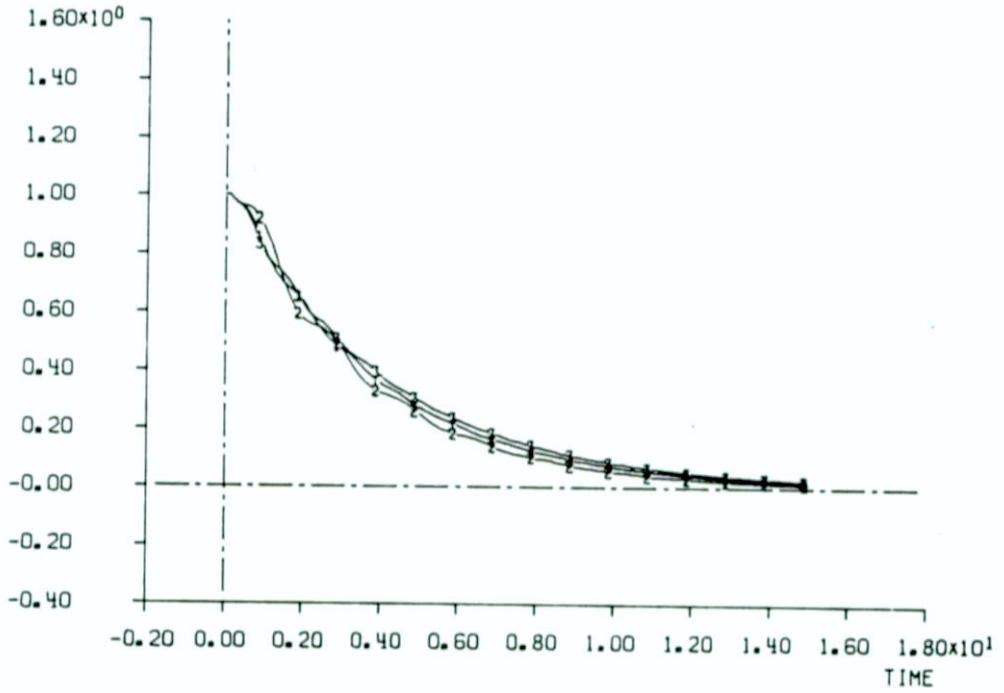


Fig. 5.15 Zustand  $x_1(t)$

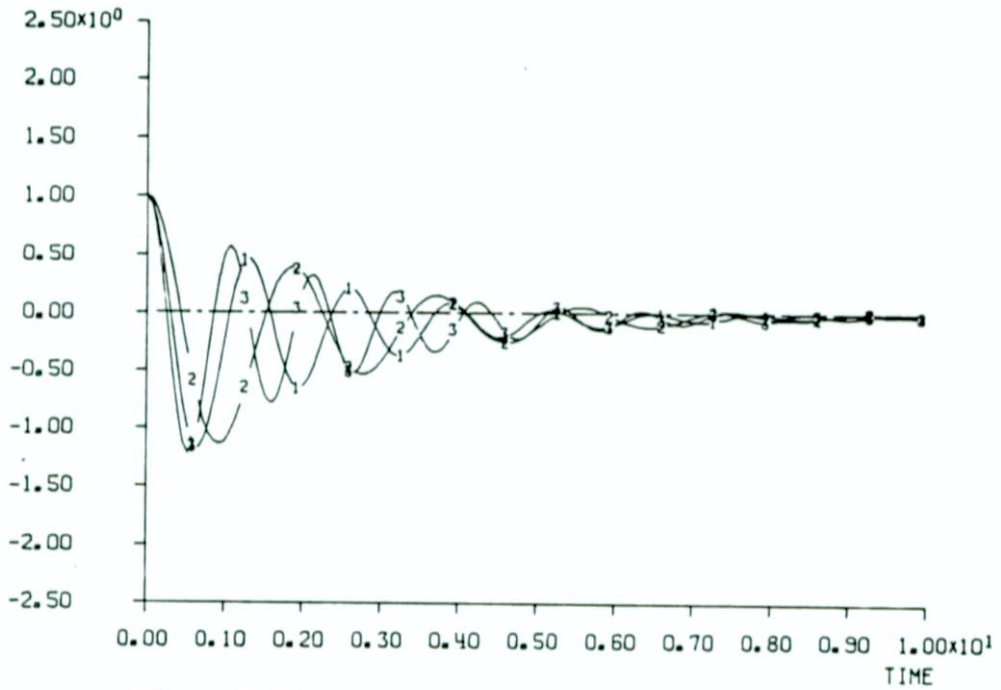


Fig. 5.16 Zustand  $x_4(t)$

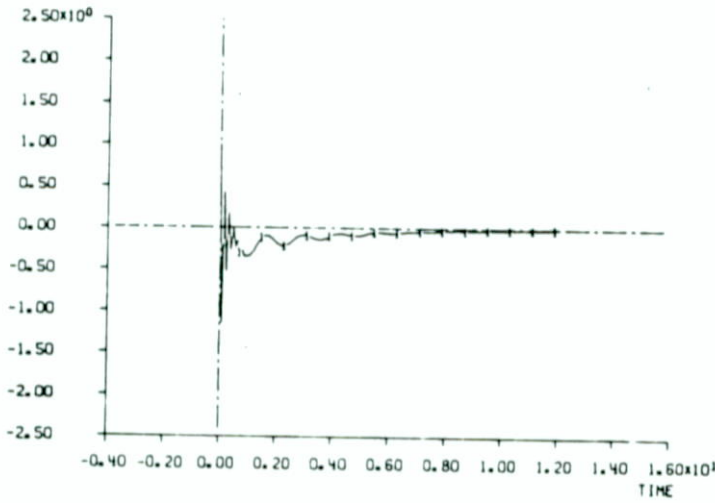


Fig. 5.17 Zustand  $x_2(t)$  bei  $\underline{\alpha}_{nom}$

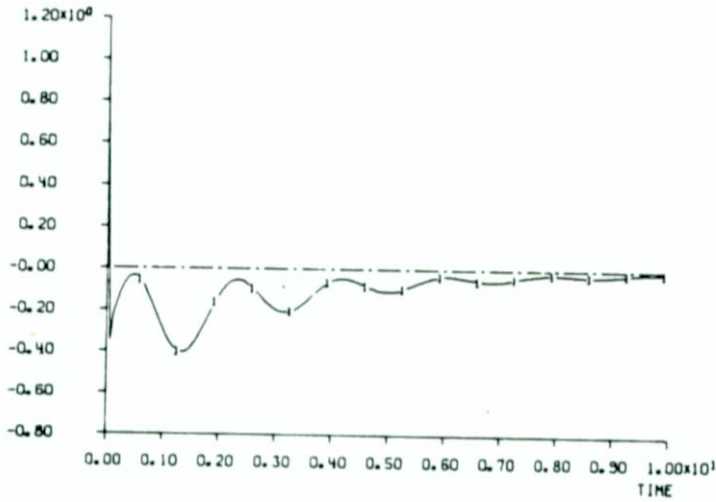


Fig. 5.18 Zustand  $x_2(t)$  bei  $\underline{\alpha}_{min}$

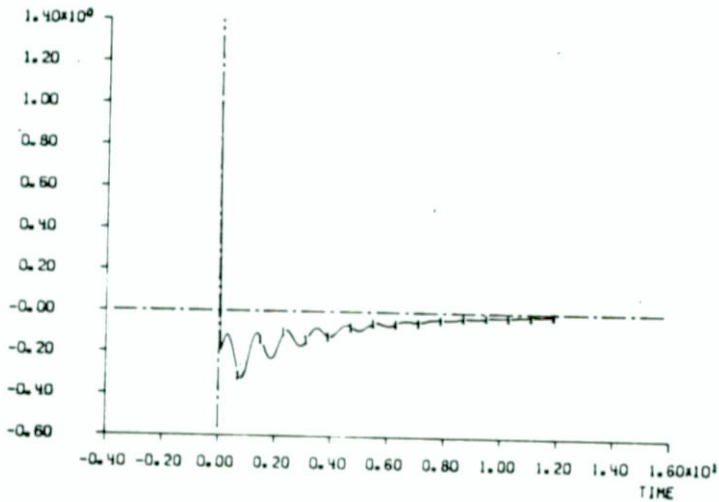


Fig. 5.19 Zustand  $x_2(t)$  bei  $\underline{\alpha}_{max}$

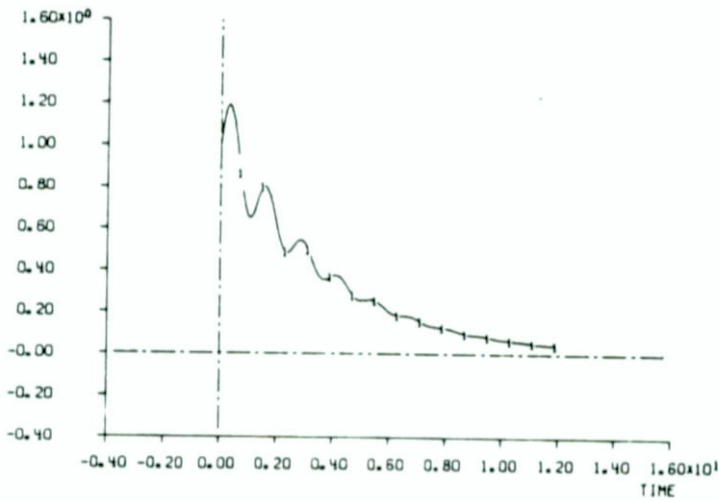


Fig. 5.20 Zustand  $x_3(t)$  bei  $\underline{\alpha}_{nom}$

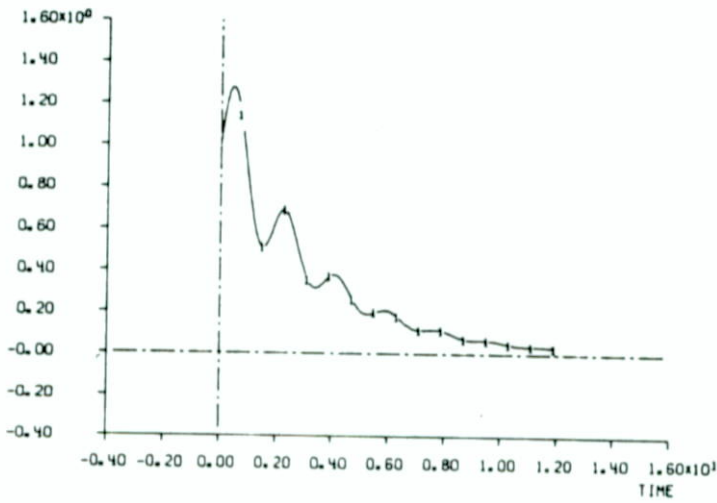


Fig. 5.21 Zustand  $x_3(t)$  bei  $\underline{\alpha}_{min}$

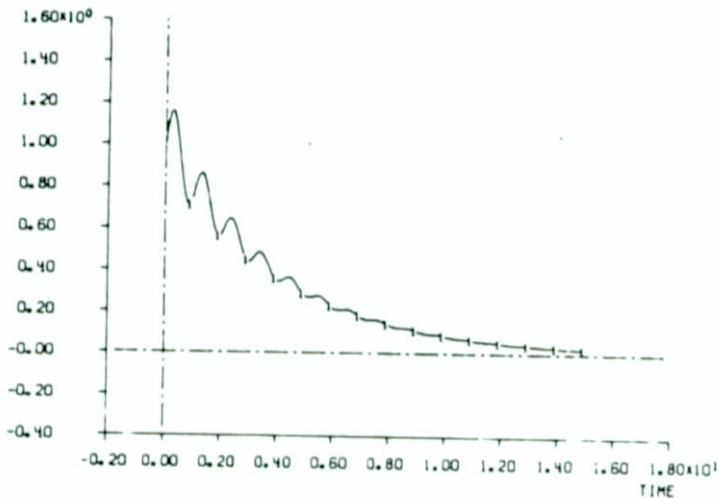


Fig. 5.22 Zustand  $x_3(t)$  bei  $\underline{\alpha}_{max}$

Berechnung der Zielfunktion

SYSTEMORDNUNG = 4

ANZAHL STEUERVARIABLEN = 1

Q-MATRIX  
 .....  
 COLUMN 1 2 3 4  
 ROW 1 .10000000 0. 0. 0.  
 ROW 2 0. .10000000 0. 0.  
 ROW 3 0. 0. .10000000 0.  
 ROW 4 0. 0. 0. .10000000

R-MATRIX  
 .....  
 COLUMN 1  
 ROW 1 1.00000000

A-QUER-MATRIX  
 .....  
 COLUMN 1 2 3 4  
 ROW 1 0. 1.00000000 0. 0.  
 ROW 2 -1.00000000 -40.97600000 1.39400000 0.  
 ROW 3 0. 0. 0. 1.00000000  
 ROW 4 23.52900000 0. -23.52900000 -235000000

B-QUER-MATRIX  
 .....  
 COLUMN 1  
 ROW 1 0.  
 ROW 2 6.93000000  
 ROW 3 0.  
 ROW 4 0.

DAS ELEKTROTECHNISCHE ZENTRUM MIT LANDBAU-14.100 \*\*\*\*\*

7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51

L-MATRIX  
 .....  
 1 ROWS      4 COLUMNS      STORAGE MODE 0  
 COLUMN      1                      2                      3                      4  
 ROW 1   6.13872239      .263737970      -4.45539059      .290246400E-01

X0-VEKTOR  
 .....  
 4 ROWS      1 COLUMNS      STORAGE MODE 0

COLUMN      1  
 ROW 1   1.00000000  
 ROW 2   1.00000000  
 ROW 3   1.00000000  
 ROW 4   1.00000000

PERFORMANCE INDEX = .54926E+01  
 \*\*\*\*\*

5 )  
 6 1\*  
 7  
 8 )  
 9  
 10  
 11 )  
 12 2\*  
 13  
 14 )  
 15  
 16  
 17 )  
 18 3\*  
 19  
 20 )  
 21  
 22  
 23 )  
 24 4\*  
 25  
 26 )  
 27  
 28  
 29 )  
 30 5\*  
 31  
 32 )  
 33  
 34  
 35  
 36 6\*  
 37  
 38  
 39 )  
 40  
 41  
 42 7\*  
 43  
 44 )  
 45  
 46  
 47  
 48 8\*  
 49  
 50  
 51  
 52  
 53  
 54 9\*  
 55  
 56  
 57  
 58  
 59  
 60 10\*  
 61  
 62  
 63  
 64  
 65  
 66 11\*  
 67

#### 5.2.4 Das optimal-geregelte System

Für  $\lambda = 1$  erhält man die folgende Rückführungsmatrix

```
L-MATRIX          1 ROWS      4 COLUMNS      STORAGE MODE  0
.....
  COLUMN      1          2          3          4
ROW  1  1.08223420    .347653820E-01  -.635020606    -.996691062E-02
```

Im übrigen gelten die selben Bemerkungen wie auf Seite 99. Das Vorgehen ist auch analog. Auch hier konnte kein Robustheitsgebiet für  $\lambda = 1$  gefunden werden, da für das System – sogar bei  $\underline{\alpha}_{nom}$  – die Insensitivitätsbedingung nicht erfüllt wird.

Für die Simulationen wurden auch die Anfangsbedingungen

$$\underline{x}(0) = ( 1 \ 1 \ 1 \ 1 )'$$

gewählt.

Simulationen

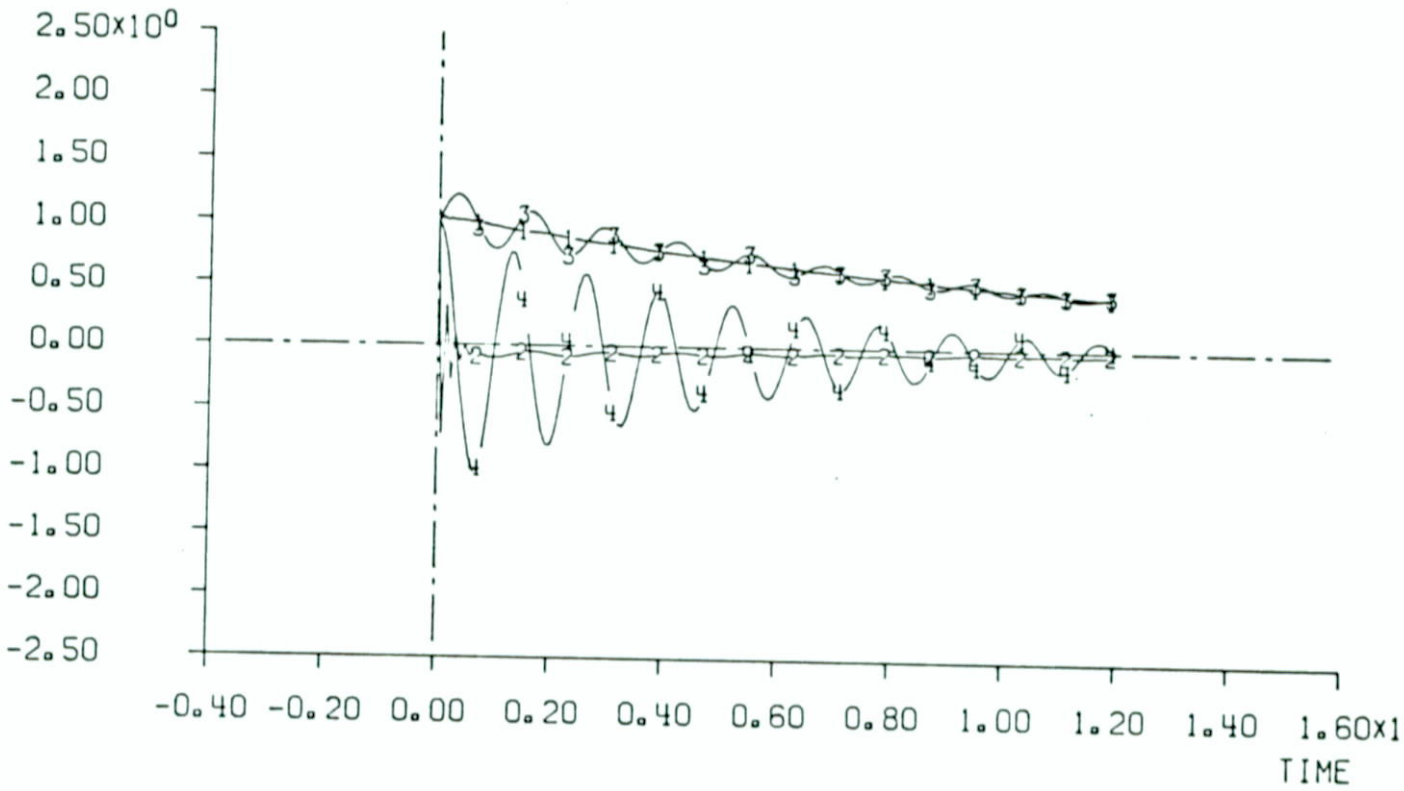


Fig. 5.23 Zustände  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$  und  $x_4(t)$   
bei  $\underline{\alpha}_{nom}$

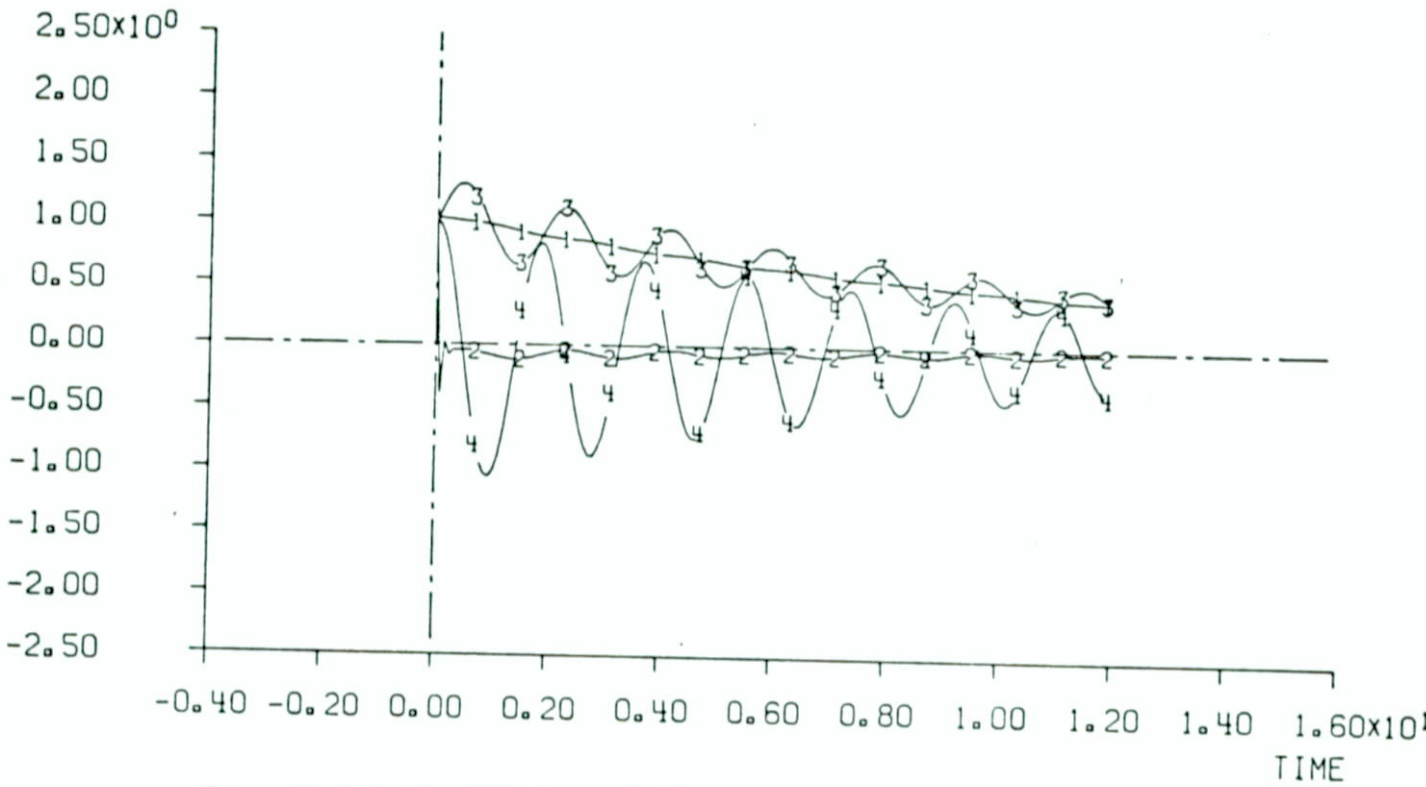


Fig. 5.24 Zustände  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$  und  $x_4(t)$   
bei  $\underline{\alpha}_{min}$

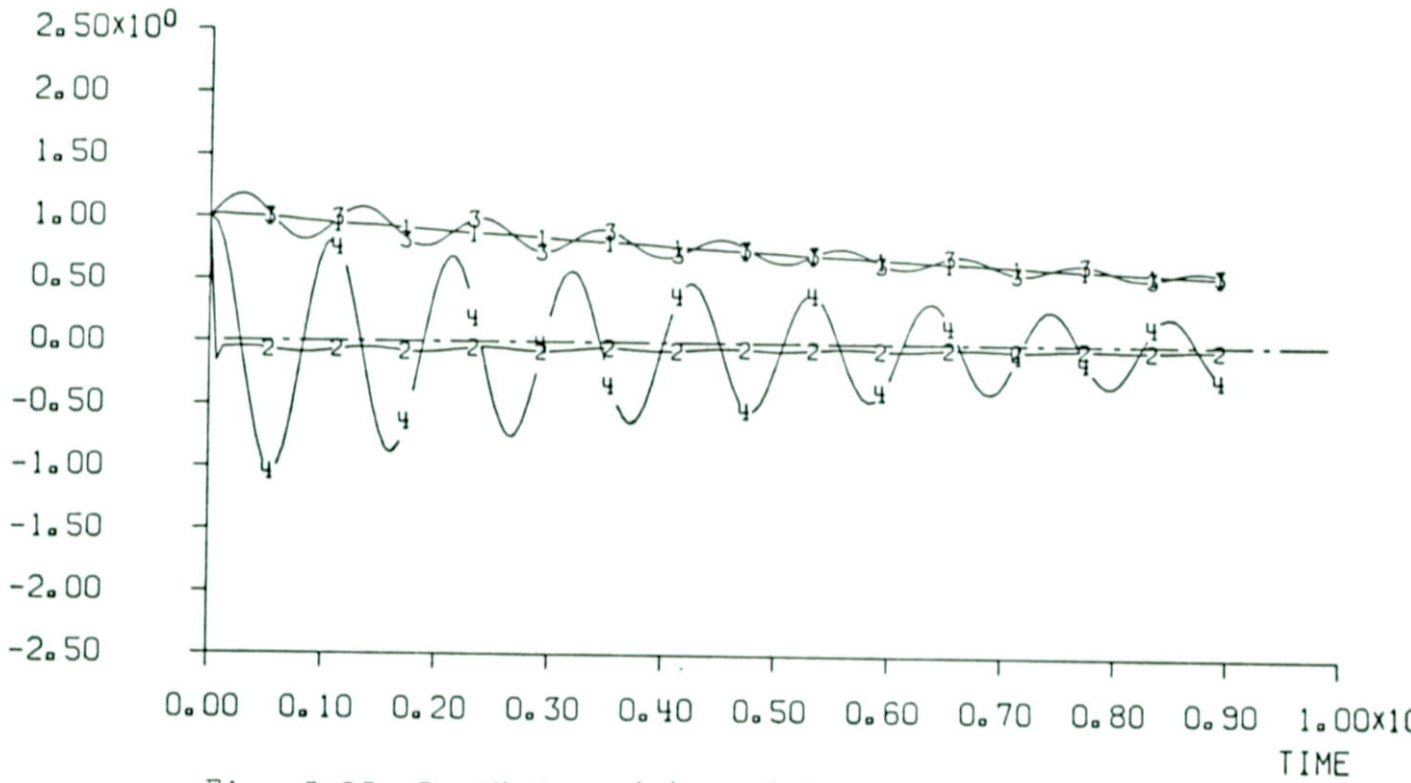


Fig. 5.25 Zustände  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$  und  $x_4(t)$   
bei  $\underline{\alpha}_{\max}$

Anschliessend wird die Zielfunktion für  $\lambda = 1$  berechnet. Es ergibt sich einen Wert von

$$J = 1.61$$

Man erinnere sich, dass beim suboptimalen System

$$J = 5.49$$

war.

Berechnung der Zielfunktion

SYSTEMORDNUNG = 4

ANZAHL STEUERVARIABLN = 1

Q-MATRIX  
 .....  
 4 ROWS 4 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4  
 ROW 1 -1.00000000 0. 0. 0.  
 ROW 2 0. -1.00000000 0. 0.  
 ROW 3 0. 0. -1.00000000 0.  
 ROW 4 0. 0. 0. -1.00000000

R-MATRIX  
 .....  
 1 ROWS 1 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1  
 ROW 1 1.00000000

A-QUER-MATRIX  
 .....  
 4 ROWS 4 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4  
 ROW 1 0. 1.00000000 0. 0.  
 ROW 2 -1.00000000 -40.9760000 1.39400000 0.  
 ROW 3 0. 0. 0. 1.00000000  
 ROW 4 23.5290000 0. -23.5290000 -235000000

B-QUER-MATRIX  
 .....  
 4 ROWS 1 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1  
 ROW 1 0.  
 ROW 2 6.93000000  
 ROW 3 0.  
 ROW 4 0.

L-MATRIX  
 .....  
 1 ROWS 4 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4  
 ROW 1 1.08220000 .347650000E-01 -.635020000 -.996690000E-02

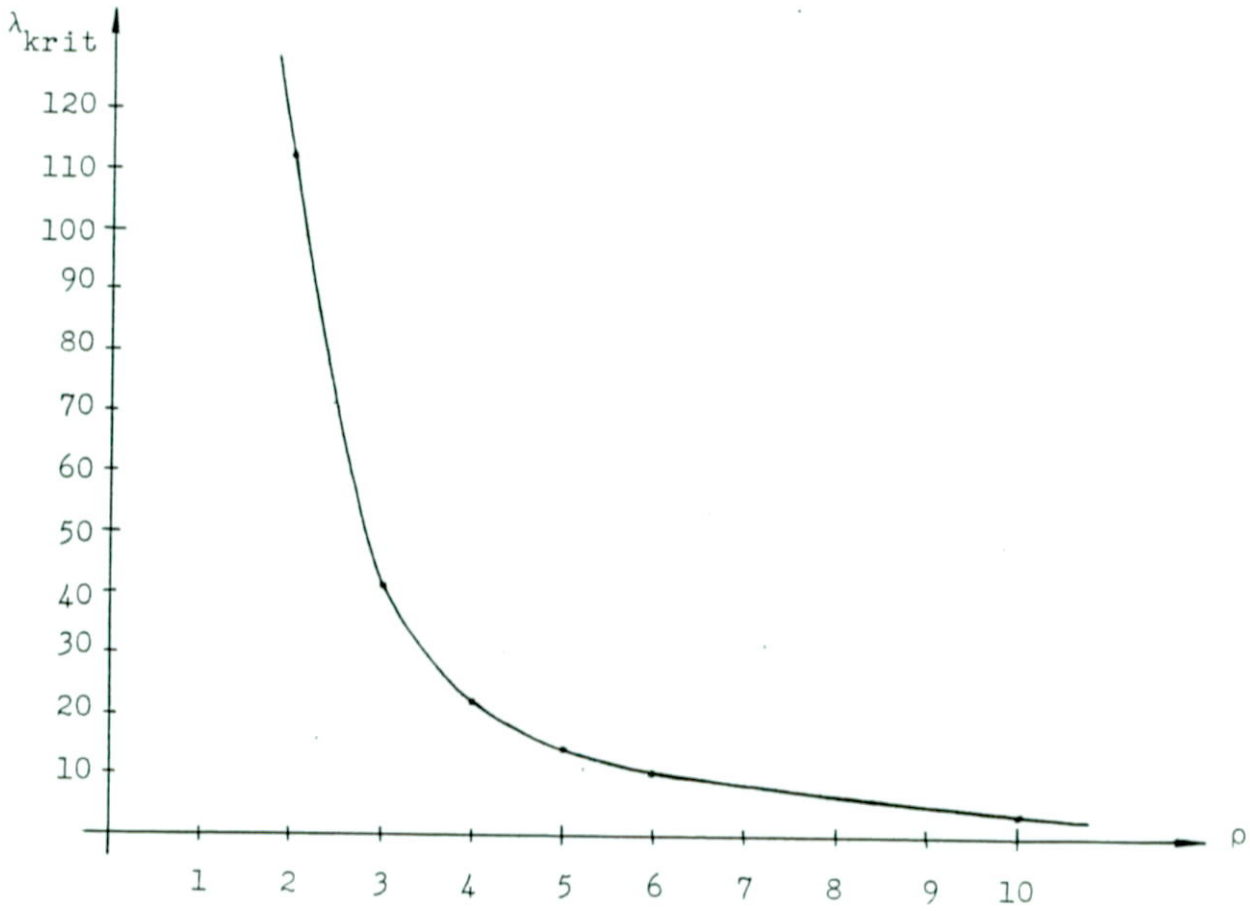
X0-VEKTOR  
 .....  
 4 ROWS 1 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1

ROW 1 1.00000000  
 ROW 2 1.00000000  
 ROW 3 1.00000000  
 ROW 4 1.00000000

PERFORMANCE INDEX = .16056E+01  
 \*\*\*\*\*

61\*  
 7  
 8 )  
 9  
 10  
 11 )  
 12 2\*  
 13  
 14 )  
 15  
 16  
 17 )  
 18 3\*  
 19  
 20 )  
 21  
 22  
 23 )  
 24 4\*  
 25  
 26 )  
 27  
 28  
 29 )  
 30 5\*  
 31  
 32 )  
 33  
 34  
 35 )  
 36 6\*  
 37  
 38 )  
 39  
 40  
 41 )  
 42 7\*  
 43  
 44 )  
 45  
 46  
 47 )  
 48 8\*  
 49  
 50 )  
 51  
 52  
 53 )  
 54 9\*  
 55  
 56 )  
 57  
 58  
 59 )  
 60 10\*  
 61  
 62 )  
 63  
 64  
 65 )  
 66 11\*  
 67

5.2.5 Die  $\lambda_{\text{krit}}(\rho)$ -Kurve



$\rho$	$\lambda_{\text{krit}}$	$\rho$	$\lambda_{\text{krit}}$
1	$5 \cdot 10^5$	6	10.0
2	112	10	3.7
3	41.2	50	0.004
4	22.3	100	0.0004
5	14.17		

$$Q = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}$$

### 5.3. Regelung einer hydraulischen Werkzeugmaschine

#### 5.3.1 Systembeschreibung

Die zu regelnde Werkzeugmaschine ist in Fig. 5.26 angegeben.

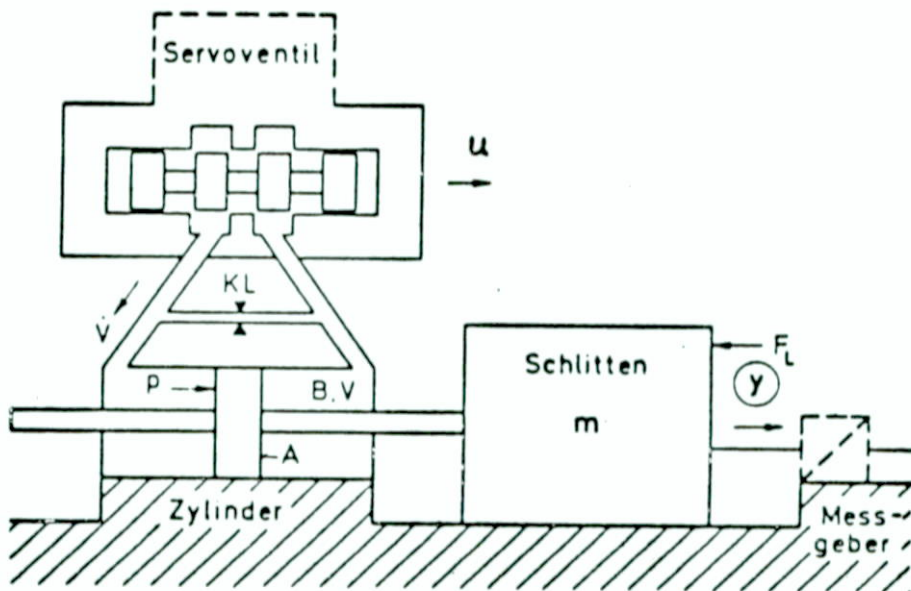


Fig. 5.26

#### Darstellung im Zustandsraum

Mit der Einführung der Zustandsvariablen

$$\underline{x} = \begin{bmatrix} y \\ \dot{y} \\ y \\ p \end{bmatrix}$$

wobei  $y$  die Lage des Schlitten angibt,  $\dot{y}$  dessen Geschwindigkeit und  $p$  der Druck am Zylinder, hat das linearisierte Mo-

dell folgende Zustandsdarstellung :

$$\begin{bmatrix} y \\ \dot{y} \\ p \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -(R + \frac{4R_c}{\pi})/m & A/m \\ 0 & -\frac{4B}{V} \cdot A & -\frac{4B}{V}(\sigma + K_L) \end{bmatrix} \cdot \begin{bmatrix} y \\ \dot{y} \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{4B}{V} \end{bmatrix} \cdot u + \begin{bmatrix} 0 \\ -1/m \\ 0 \end{bmatrix} \cdot F_L$$

Die unbekannte Kraft  $F_L$  wurde gleich 0 gesetzt, nach Absprache mit Herrn Dr. Senning. Die Systemordnung reduziert sich infolgedessen auf 2. (Klarheitshalber arbeiten wir jedoch mit einem System 3. Ordnung.)

Die Systemmatrix A hängt von acht Parametern ab : B, m, R,  $R_c$ ,  $K_L$ ,  $\sigma$ , V, A (für die Erklärung der einzelnen Parameter siehe [12]). Setzt man die Nennwerte der Parameter in die Matrix A ein, so ergeben sich für die Matrixelemente Werte, die sich um 15 Zehnerpotenzen unterscheiden. Um die dadurch bedingten rechentechnischen Schwierigkeiten zu umgehen, hat man eine Skalierung vorgenommen, die zu "vernünftigen" Werten führt.

Das skalierte System sieht wie folgt aus

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{b}u$$

mit

$$\underline{\tilde{x}} = \begin{bmatrix} \tilde{p} \\ \dot{y} \\ y \end{bmatrix} \quad \text{und} \quad \tilde{p} = 10^{-7} p$$

Berechnung der skalierten Systemmatrix  $\tilde{A}$  und  $\tilde{b}$  :

$$\begin{bmatrix} y \\ \dot{y} \\ p \end{bmatrix} = A \begin{bmatrix} y \\ \dot{y} \\ p \end{bmatrix} + \underline{b}u \xrightarrow{\substack{\tilde{p} = 10^{-7} p \\ \tilde{y} = \dot{y}}} \begin{bmatrix} \tilde{p} \\ \dot{y} \\ y \end{bmatrix} = \tilde{A} \begin{bmatrix} \tilde{p} \\ \dot{y} \\ y \end{bmatrix} + \tilde{\underline{b}}u$$

Das zu A und  $\underline{b}$  zugehörige Differentialgleichungssystem ist

$$\begin{aligned} \dot{y} &= \dot{y} \\ \ddot{y} &= -\frac{1}{m}(R + \frac{4}{\pi}R_c)\dot{y} + \frac{A}{m}p \\ \dot{p} &= -\frac{4BA}{V}\dot{y} - \frac{4B(\sigma + K_L)}{V}p - \frac{4B}{V}u \end{aligned}$$

Durch die Substitution

$$p = 10^7 \tilde{p}$$

und durch Vertauschung der Gleichungen erhält man das zu A und  $\underline{b}$  zugehörige Differentialgleichungssystem

$$\begin{aligned} \dot{\tilde{p}} &= -\frac{4B(\sigma + K_L)}{V}\tilde{p} - \frac{4BA}{V}\cdot 10^{-7}\dot{y} - \frac{4B}{V}\cdot 10^{-7}u \\ \ddot{y} &= \frac{A}{m}\cdot 10^7\tilde{p} - \frac{1}{m}(R + \frac{4}{\pi}R_c)\dot{y} \\ \dot{y} &= \dot{y} \end{aligned}$$

Wegen der erforderlichen Bedingung der Multilinearität, werden die Parameter B, V und m - bei ihren Nennwerten - konstant gehalten.

Die neue Systemmatrix  $\tilde{A}$  und der neue Steuervektor  $\tilde{b}$  sehen nun folgendermassen aus

$$\tilde{A} = \begin{bmatrix} -6.407 \cdot 10^{12} (\sigma + K_L) & -6.407 \cdot 10^5 A & 0 \\ \frac{A}{128.7} \cdot 10^7 & -\frac{1}{128.7} (R + \frac{4}{\pi} R_c) & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\tilde{b} = ( -640732 \quad 0 \quad 0 )'$$

mit den Grenzen

$$\begin{aligned} 10^{-14} &\leq \sigma \leq 1.5 \cdot 10^{-10} && [m^5/sN] \\ 10.3 \cdot 10^{-16} &\leq K_L \leq 3.5 \cdot 10^{-14} && [m^5/sN] \\ 10.5 \cdot 10^{-4} &\leq A \leq 11 \cdot 10^{-4} && [m^2] \\ 50 &\leq R \leq 5000 && [sN/m] \\ 0 &\leq R_c \leq 50 && [sN/m] \end{aligned}$$

und der Nominalwerten

$$\underline{a}_{nom} = \begin{bmatrix} \sigma \\ K_L \\ A \\ R \\ R_c \end{bmatrix} = \begin{bmatrix} 2.4 \cdot 10^{-12} \\ 2 \cdot 10^{-14} \\ 10.75 \cdot 10^{-4} \\ 150 \\ 10 \end{bmatrix}_{nom}$$

Setzt man in  $\tilde{A}$  auch diese Nennwerte ein und berechnet die Eigenwerte dieser Matrix, so erhält man

$$\mu_1 = 0$$

$$\mu_2 = -8.35 + j239.75$$

$$\mu_3 = -8.35 - j239.75$$

Die vorgenommene Skalierung ändert an dem Systemeigenschaften nichts, d.h.  $A$  ist kongruent zu  $\tilde{A}$  und folglich besitzen beide Matrizen die selben Eigenwerte. Dies bedeutet, dass das System instabil ist, wegen dem 1. Eigenwert.

### 5.3.2 Das offene System

Für dieses System musste die Integrationsmethode geändert werden : anstelle von Runge Kutta 4. Ordnung mit variabler Schrittlänge wie bisher, wurde GEAR verwendet. Der Grund dafür ist, dass das System "sehr steif" ist. Trotzdem blieben einige Integrationen erfolglos, z.B. Zustand  $x_3(t)$ . (Die Integration wird auch dadurch erschwert, dass dieser Zustand für  $\underline{\alpha} = \underline{\alpha}_{nom}$  instabil ist).

Die Verläufe von  $x_1(t)$  bzw.  $x_2(t)$  sind für die verschiedenen Parameterpunkte sehr unterschiedlich, wie leicht zu sehen ist. Daraus erkennt man die grosse Sensitivität des Systems bezüglich Parameteränderungen.

Mit einer Regelung möchte man einerseits die Instabilität beseitigen und andererseits die Sensitivität vermindern. Siehe dazu 5.3.3 und 5.3.4.



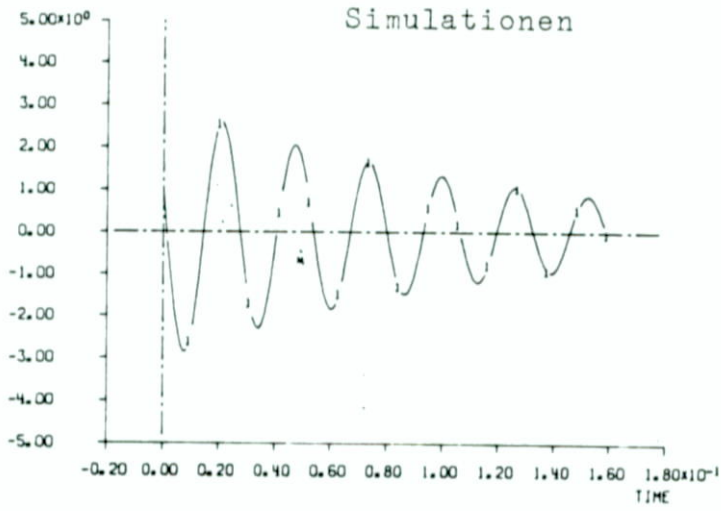


Fig. 5.27 Zustand  $x_1(t)$  bei  $\underline{\alpha}_{nom}$

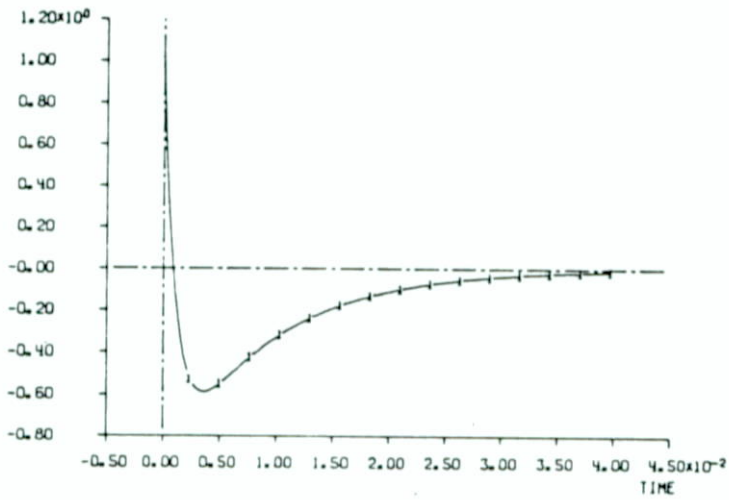


Fig. 5.28 Zustand  $x_1(t)$  bei  $\underline{\alpha}_{max}$

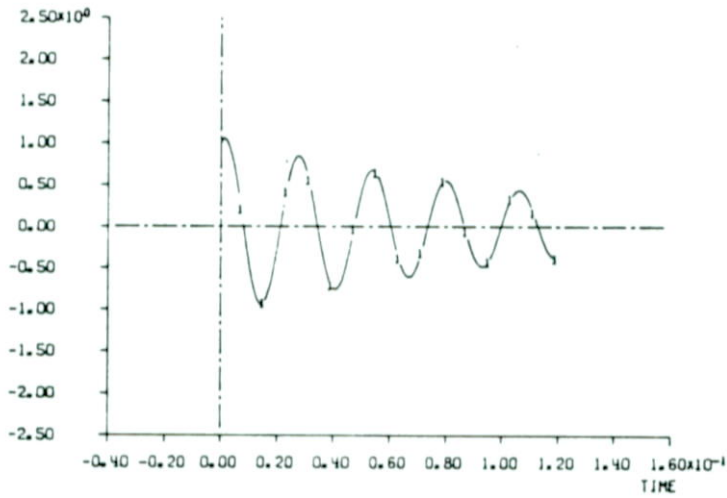


Fig. 5.29 Zustand  $x_2(t)$  bei  $\alpha_{nom}$

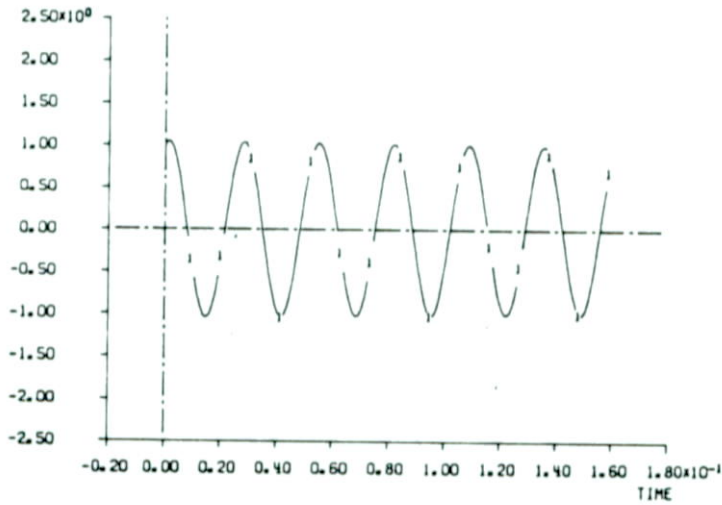


Fig. 5.30 Zustand  $x_2(t)$  bei  $\alpha_{min}$

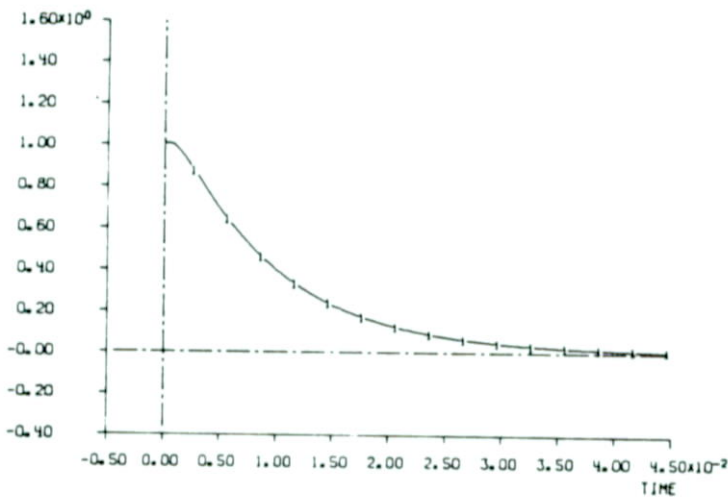


Fig. 5.31 Zustand  $x_2(t)$  bei  $\alpha_{max}$

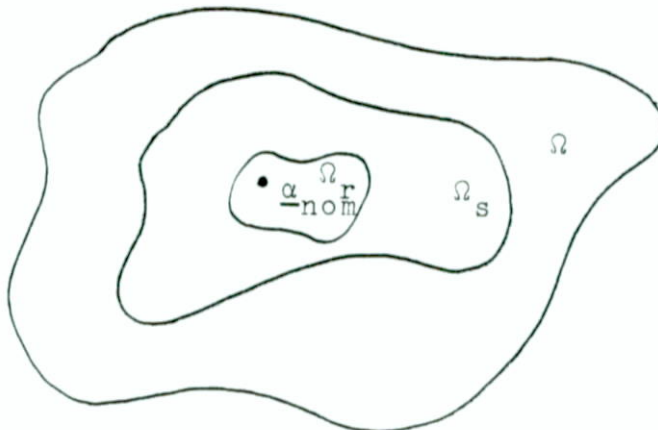
### 5.3.3 Das suboptimale, robuste System

Der Wert von  $\lambda_{\text{krit}}$  wurde anfänglich im Intervall  $[1, 100]$  gesucht. Da für diese Intervallgrenzen die Suche erfolglos blieb, wurde  $\lambda_{\text{max}}$  schrittweise bis  $10^6$  erhöht, obwohl dies aus physikalischen Gründen unbrauchbar wäre. Trotzdem konnte kein  $\lambda_{\text{krit}}$  gefunden werden, für die gegebenen Parametergrenzen.

Mit Hilfe der Subroutine GEBIET wurden nun für verschiedene  $\lambda$ -Werten (  $0.5, 10, 50, 10^3, 10^6$  ) die entsprechenden neuen Robustheitsgebiete gesucht und die dazugehörigen Rückführungsmatrizen berechnet.

Für jede L-Matrix wurde das geschlossene System simuliert. Da alle Verläufe voneinander kaum zu unterscheiden sind, sei hier nur die Simulation für  $\lambda = 1000$  gebracht ( für  $\lambda = 1$  siehe 5.3.4). Die Erklärung dazu ist die Tatsache, dass man für  $\lambda \in [1, 10^6]$  und  $\rho = 5.25$  an einer sehr steilen Flanke der  $\lambda_{\text{krit}}(\rho)$ -Kurve ist, d.h., dass grosse Änderungen in  $\lambda$  sich sehr wenig auf das Systemverhalten auswirken (obwohl die Zielfunktionswerte sich merklich voneinander unterscheiden). Aus dem gleichen Grund konnte man auch nicht die  $\lambda_{\text{krit}}(\rho)$ -Kurve aufnehmen.

Bei der Suche nach den neuen Stabilitäts- bzw. Robustheitsgebiete wird das gegebene Parametergebiet  $\Omega$  so lange verkleinert bis  $\Omega$  ganz in  $\Omega_s$  bzw.  $\Omega_r$  enthalten ist. Siehe folgende Skizze :



Dabei bedeuten

$\Omega$  : ursprüngliches Gebiet ,  $\underline{\alpha} \in \Omega$

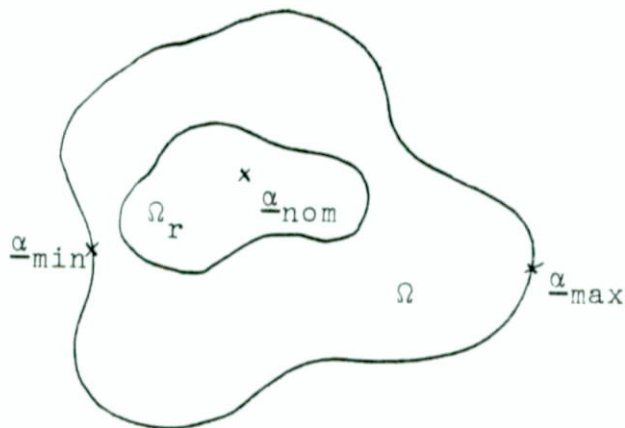
$\Omega_s$  : Stabilitätsgebiet

$\Omega_r$  : Robustheitsgebiet

Für dieses Beispiel, wie auch für die vorher untersuchten Systeme, ist  $\Omega$  eine Teilmenge von  $\Omega_s$ .

Bemerkung

Es sei zu beachten, dass die verschiedenen  $\lambda$ -Werte für die neuen Robustheitsgebiete die jeweiligen  $\lambda_{\text{krit}}$ -Werte sind. Die Simulationen wurden aber, wie immer, für die drei Parameterpunkte  $\underline{\alpha}_{\text{nom}}$ ,  $\underline{\alpha}_{\text{min}}$  und  $\underline{\alpha}_{\text{max}}$  gemacht. Letztere liegen aber jetzt ausserhalb der neuen Robustheitsgebiete.



Dies ist der Grund dafür, dass für  $\lambda = 1000$  und  $\lambda = 1$  der Zustandsverlauf von  $x_1(t)$  bei  $\underline{\alpha} = \underline{\alpha}_{\text{max}}$  wesentlich langsamer ist als bei  $\underline{\alpha}_{\text{min}}$  und  $\underline{\alpha}_{\text{nom}}$  (siehe Fig. 5.34 und Fig. 5.39).

Berechnung des Robustheitsgebietes  $\Omega_r$  und der Rückführungsmatrix L  
für  $\lambda = 0.5$

SYSTEMORDNUNG : 3  
ANZAHL STEUERVARIABLEN : 1  
ANZAHL PARAMETER : 5

SR : 0  
NGEB : 1  
LAMIN : .10  
LAMAX : .50  
RHO : 5.25

Q-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

.....

COLUMN 1 2 3

ROW 1 2.00000000 0. 0.  
ROW 2 0. 5.00000000 0.  
ROW 3 0. 0. 5.00000000

R-MATRIX 1 ROWS 1 COLUMNS STORAGE MODE 0

.....

COLUMN 1

ROW 1 1.00000000

A-QUER-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

.....

COLUMN 1 2 3

ROW 1 -15.5060000 0. 0.  
ROW 2 83.5280000 -1.26400000 0.  
ROW 3 0. 1.00000000 0.

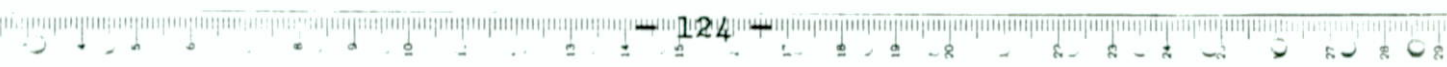
B-QUER-MATRIX 3 ROWS 1 COLUMNS STORAGE MODE 0

.....

COLUMN 1

ROW 1 -540732.000  
ROW 2 0.  
ROW 3 0.

5) 61\* 7 8) 9 10 11) 12\* 13 14 ) 15 16 17 ) 18 3\* 19 20 ) 21 22 23 ) 24 4\* 25 26 ) 27 28 29 ) 30 5\* 31 32 ) 33 34 35 36 6\* 37 38 ) 39 40 41 ) 42 7\* 43 44 ) 45 46 47 ) 48 8\* 49 50 ) 51 52 53 ) 54 9\* 55 56 57 58 59 60 10\* 61 62 63 64 65 66 11\* 67



```

7 ALMIN-VEKTJR      1 ROWS      5 COLUMNS      STORAGE MODE 0
8 .....
9
10 COLUMN 1      2      3      4      5
11
12 ROW 1  .10000000E-13  .10300000E-14  .10500000E-02  50.0000000  0.
13
14
15
16
17
18 ALMAX-VEKTJR      1 ROWS      5 COLUMNS      STORAGE MODE 0
19 .....
20
21 COLUMN 1      2      3      4      5
22
23 ROW 1  .15000000E-09  .35000000E-13  .11000000E-02  5000.00000  50.0000000
24
25
26
27
28
29
30
31 ALMIG-VEKTJR      1 ROWS      5 COLUMNS      STORAGE MODE 0
32 .....
33
34 COLUMN 1      2      3      4      5
35
36 ROW 1  .10000000E-12  .11000000E-14  .10600000E-02  100.000000  5.0000000
37
38
39
40
41
42 ALMAG-VEKTJR      1 ROWS      5 COLUMNS      STORAGE MODE 0
43 .....
44
45 COLUMN 1      2      3      4      5
46
47 ROW 1  .10000000E-10  .30000000E-13  .10900000E-02  1000.00000  40.0000000
48
49
50
51
52
53 ALPHA-QUER      1 ROWS      5 COLUMNS      STORAGE MODE 0
54 .....
55
56 COLUMN 1      2      3      4      5
57
58 ROW 1  .24000000E-11  .20000000E-13  .10750000E-02  150.000000  10.0000000
59
60
61
62
63
64
65
66
67
68
69
70

```

LAMBDA-MAX WURDE ZU KLEIN GEWAHLT

DAS NEUE ROBUSTHEITSGEBIET WIRD GEGEBEN DURCH :

ALMIN = .366432337E-13 .105075425E-14 .105296489E-02 .500000000E+02 .141059018E+01

ALMAX = .108491478E-09 .335175528E-13 .109747567E-02 .391404224E+04 .470351056E+02

L-MATRIX 1 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN 1 2 3

ROW 1 -1.00018135 -1.57692982 -1.58113883

61\* 7 8 ) 9 10 11 ) 12 2\* 13 14 ) 15 16 17 ) 18 3\* 19 20 ) 21 22 23 ) 24 4\* 25 26 ) 27 28 29 ) 30 5\* 31 32 ) 33 34 35 ) 36 6\* 37 38 ) 39 40 41 ) 42 7\* 43 44 ) 45 46 47 ) 48 8\* 49 50 ) 51 52 53 ) 54 9\* 55 56 ) 57 58 59 ) 60 10\* 61 62 63 64 65 ) 66 11\* 67

Berechnung des Robustheitsgebietes  $\Omega_r$  und der Rückführungsmatrix L

für  $\lambda = 10$

SYSTEMGRÖßUNG : 3  
ANZAHL STEUERVARIABLEN : 1  
ANZAHL PARAMETER : 5

SP : 0  
NGEB : 1  
LAMIN : 1.00  
LAMAX : 10.00  
RHO : 5.25

Q-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

.....  
COLUMN 1 2 3  
ROW 1 2.00000000 0. 0.  
ROW 2 0. 5.00000000 0.  
ROW 3 0. 0. 5.00000000

R-MATRIX 1 ROWS 1 COLUMNS STORAGE MODE 0

.....  
COLUMN 1  
ROW 1 1.00000000

A-QUER-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

.....  
COLUMN 1 2 3  
ROW 1 -15.5060000 -686.787000 0.  
ROW 2 83.5280000 -1.26400000 0.  
ROW 3 0. 1.00000000 0.

B-QUER-MATRIX 3 ROWS 1 COLUMNS STORAGE MODE 0

.....  
COLUMN 1  
ROW 1 -64.0732.000  
ROW 2 0.  
ROW 3 0.

5 )  
6 1 )  
7 )  
8 1 )  
9 )  
10 )  
11 )  
12 2 )  
13 )  
14 )  
15 )  
16 )  
17 )  
18 3 )  
19 )  
20 )  
21 )  
22 )  
23 )  
24 4 )  
25 )  
26 )  
27 )  
28 )  
29 )  
30 5 )  
31 )  
32 )  
33 )  
34 )  
35 )  
36 6 )  
37 )  
38 )  
39 )  
40 )  
41 )  
42 7 )  
43 )  
44 )  
45 )  
46 )  
47 )  
48 8 )  
49 )  
50 )  
51 )  
52 )  
53 )  
54 9 )  
55 )  
56 )  
57 )  
58 )  
59 )  
60 10 )  
61 )  
62 )  
63 )  
64 )  
65 )  
66 11 )  
67 )

ALPHI-VEKTOR 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .100000000E-13 .103000000E-14 .105000000E-02 50.0000000 0.

ALHAX-VEKTOR 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .150000000E-09 .350000000E-13 .110000000E-02 5000.00000 50.0000000

ALMIG-VEKTOR 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .100000000E-12 .110000000E-14 .106000000E-02 100.000000 5.0000000

ALMAG-VEKTOR 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .100000000E-10 .300000000E-13 .109000000E-02 1000.00000 40.0000000

ALPHA-QUER 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .240000000E-11 .200000000E-13 .107500000E-02 150.000000 10.0000000

LAMBDA-MAX WURDE ZU KLEIN GEWAHLT

5.7  
6.1\*  
7  
8.3  
9  
10  
11.0  
12.2\*  
13  
14.1  
15  
16  
17.3  
18.5\*  
19  
20.1  
21  
22  
23.3  
24.4\*  
25  
26.1  
27  
28  
29.1  
30.5\*  
31  
32.1  
33  
34  
35  
36.6\*  
37  
38.1  
39  
40  
41.1  
42.7\*  
43  
44.1  
45  
46  
47.1  
48.8\*  
49  
50  
51  
52  
53  
54.9\*  
55  
56.1  
57  
58  
59  
60.0\*  
61  
62.3  
63  
64  
65.0  
66.1\*  
67

DAS NEUE ROBUSTHEITSGEBIET WIRD GEGEBEN DURCH :

ALMIN = .366392079E-13 .105072648E-14 .105296093E-02 .500000000E+02 .140856503E+01

ALMAX = .106547027E-09 .335195367E-13 .109743054E-02 .301562936E+04 .470390734E+02

L-MATRIX 1 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN 1 2 3

ROW 1 -4.47231743 -7.05597778 -7.07105781

61  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67

ALPHI-VEKTOR 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .100000000E-13 .103000000E-14 .105000000E-02 50.0000000 0.

ALPHA-VEKTOR 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .150000000E-09 .350000000E-13 .110000000E-02 5000.00000 50.0000000

ALPHI-VEKTOR 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .100000000E-12 .110000000E-14 .106000000E-02 100.0000000 5.00000000

ALPHA-VEKTOR 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .100000000E-10 .300000000E-13 .109000000E-02 1000.00000 40.0000000

ALPHA-QUER 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .240000000E-11 .260000000E-13 .107500000E-02 150.0000000 10.0000000

LAMBDA-MAX WURDE ZU KLEIN GEWAFHLT

61  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68

DAS NEUE ROBUSTHEITSGEBIET WIRD GEgeben DURCH :

ALMIN = .366385704E-13 .105072206E-14 .105295029E-02 .500000000E+02 .140824262E+01

ALMAX = .108555371E-09 .335198525E-13 .109748131E-02 .381588202E+04 .470397050E+02

L-MATRIX 1 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN 1 2 3

ROW 1 -10.0001815 -15.7789754 -15.8113883

7  
8  
9  
10  
11  
12 2'  
13  
14  
15  
16  
17  
18 3'  
19  
20  
21  
22  
23  
24 4'  
25  
26  
27  
28  
29  
30 5'  
31  
32  
33  
34  
35  
36 6'  
37  
38  
39  
40  
41  
42 7'  
43  
44  
45  
46  
47  
48 8'  
49  
50  
51  
52  
53  
54 9'  
55  
56  
57  
58  
59  
60 10'  
61  
62  
63  
64  
65  
66 11'  
67  
68  
69

Berechnung des Robustheitsgebietes  $\Omega_r$  und der Rückführungsmatrix L für  $\lambda = 1000$

ANZAHL STEUERVARIABLEN I 1  
ANZAHL PARAMETER I 5

SR I 0  
NGFB I 1  
LAMIN I 1.00  
LAMAX I 1000.00  
RHC I 5.25

Q-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN 1 2 3

ROW 1 2.00000000 0. 0.  
ROW 2 0. 5.00000000 0.  
ROW 3 0. 0. 5.00000000

F-MATRIX 1 ROWS 1 COLUMNS STORAGE MODE 0

COLUMN 1

ROW 1 1.00000000

A-QUER-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN 1 2 3

ROW 1 -15.5060000 -688.787000 0.  
ROW 2 83.5280000 -1.26400000 0.  
ROW 3 0. 1.00000000 0.

B-QUER-MATRIX 3 ROWS 1 COLUMNS STORAGE MODE 0

COLUMN 1

ROW 1 -640732.000  
ROW 2 0.  
ROW 3 0.

10  
11  
12 2\*  
13  
14  
15  
16  
17  
18 3\*  
19  
20  
21  
22  
23  
24 4\*  
25  
26  
27  
28  
29  
30 5\*  
31  
32  
33  
34  
35  
36 6\*  
37  
38  
39  
40  
41  
42 7\*  
43  
44  
45  
46  
47  
48 8\*  
49  
50  
51  
52  
53  
54 9\*  
55  
56  
57  
58  
59  
60 10\*  
61  
62  
63  
64  
65  
66 11\*  
67  
68  
69

ALMII-VEKTOR  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .100000000E-13 .103000000E-14 .105000000E-02 50.0000000 0.

ALMAX-VEKTOR  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .150000000E-09 .350000000E-13 .110000000E-02 5000.00000 50.0000000

ALMIG-VEKTOR  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .100000000E-12 .110000000E-14 .105000000E-02 100.000000 5.00000000

ALMAG-VEKTOR  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .100000000E-10 .300000000E-13 .109000000E-02 1000.00000 40.0000000

ALPHA-QUER  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .240000000E-11 .200000000E-13 .107500000E-02 150.000000 10.0000000

LAMBDA-MAX WURDE ZU KLEIN GEMACHT

61  
 7  
 8  
 9  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48  
 49  
 50  
 51  
 52  
 53  
 54  
 55  
 56  
 57  
 58  
 59  
 60  
 61  
 62  
 63  
 64  
 65  
 66  
 67  
 68

DAS NEUE ROBUSTHEITSGEBIET WIRD GEGEBEN DURCH \*

ALMIN = .366381706E-13 .105071929E-14 .105295990E-02 .500000000E+02 .140804011E+01

ALMAX = .108561425E-09 .335200509E-13 .109748180E-02 .381604072E+04 .470401018E+02

L-MATRIX 1 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN 1 2 3

ROW 1 -44.7215411 -70.5694578 -70.7105781

61  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68

Berechnung des Robustheitsgebietes  $\Omega_r$  und der Rückführungsmatrix L  
für  $\lambda = 10^6$

ANZAHL STEUERVARIABLEN : 1  
ANZAHL PARAMETER : 5

S2 : 0  
NGEB : 1  
LAMIN : 1.00  
LAMAX : \*\*\*\*\*  
RHO : 5.25

Q-MATRIX  
.....

3 ROWS	3 COLUMNS	STORAGE MODE
COLUMN 1	2	3
ROW 1	2.00000000	0.
ROW 2	0.	0.
ROW 3	0.	5.00000000

R-MATRIX  
.....

1 ROWS	1 COLUMNS	STORAGE MODE
COLUMN 1		
ROW 1	1.00000000	

A-QUER-MATRIX  
.....

3 ROWS	3 COLUMNS	STORAGE MODE
COLUMN 1	2	3
ROW 1	-15.5060000	-688.787000
ROW 2	83.5280000	-1.26400000
ROW 3	0.	1.00000000

B-QUER-MATRIX  
.....

3 ROWS	1 COLUMNS	STORAGE MODE
COLUMN 1		
ROW 1	-640732.000	
ROW 2	0.	
ROW 3	0.	

ALMIN-VEKTJR            1 ROWS    5 COLUMNS    STORAGE MODE 0  
 .....  
 COLUMN    1            2            3            4            5  
 ROW 1   .100000000E-13   .103300000E-14   .105000000E-02   50.0000000   0.

ALMAX-VEKTJR            1 ROWS    5 COLUMNS    STORAGE MODE 0  
 .....  
 COLUMN    1            2            3            4            5  
 ROW 1   .150000000E-09   .350000000E-13   .110000000E-02   5000.00000   50.0000000

ALMIG-VEKTJR            1 ROWS    5 COLUMNS    STORAGE MODE 0  
 .....  
 COLUMN    1            2            3            4            5  
 ROW 1   .100000000E-12   .110000000E-14   .106000000E-02   100.000000   5.00000000

ALMAG-VEKTJR            1 ROWS    5 COLUMNS    STORAGE MODE 0  
 .....  
 COLUMN    1            2            3            4            5  
 ROW 1   .100000000E-10   .300000000E-13   .109000000E-02   1000.00000   40.0000000

ALPHA-QUER             1 ROWS    5 COLUMNS    STORAGE MODE 0  
 .....  
 COLUMN    1            2            3            4            5  
 ROW 1   .240000000E-11   .203000000E-13   .107500000E-02   150.000000   10.0000000

LAMBDA-MAX WURDE ZU KLEIN GEWAHLT

61  
60  
9  
10  
110  
12 2  
13  
14 )  
15  
16  
17 )  
18 3  
19  
20 )  
21  
22  
23 )  
24 4  
25  
26 )  
27  
28  
29  
30 5  
31  
32  
33  
34  
35  
36 6  
37  
38  
39  
40  
41  
42 7  
43  
44  
45  
46  
47  
48 8  
49  
50  
51  
52  
53  
54 9  
55  
56  
57  
58  
59  
60 10  
61  
62 )  
63  
64  
65 )  
66 11  
67  
68

DAS NEUE ROBUSTHEITSGEBIET WIRD GEGEBEN DURCH :

ALMIN = .366380591E-13 .105071851E-14 .105295979E-02 .500000000E+02 .140798364E+01

ALMAX = .108562974E-09 .335201062E-13 .109748194E-02 .381608498E+04 .470402125E+02

L-MATRIX 1 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN 1 2 3

ROW 1 -1414.21374 -2231.63514 -2236.06798

6.1  
7  
8  
9  
10  
11  
12.2  
13  
14  
15  
16  
17  
18.3  
19  
20  
21  
22  
23  
24.4  
25  
26  
27  
28  
29  
30.5  
31  
32  
33  
34  
35  
36.6  
37  
38  
39  
40  
41  
42.7  
43  
44  
45  
46  
47  
48.8  
49  
50  
51  
52  
53  
54.9  
55  
56  
57  
58  
59  
60.10  
61  
62  
63  
64  
65  
66.11  
67

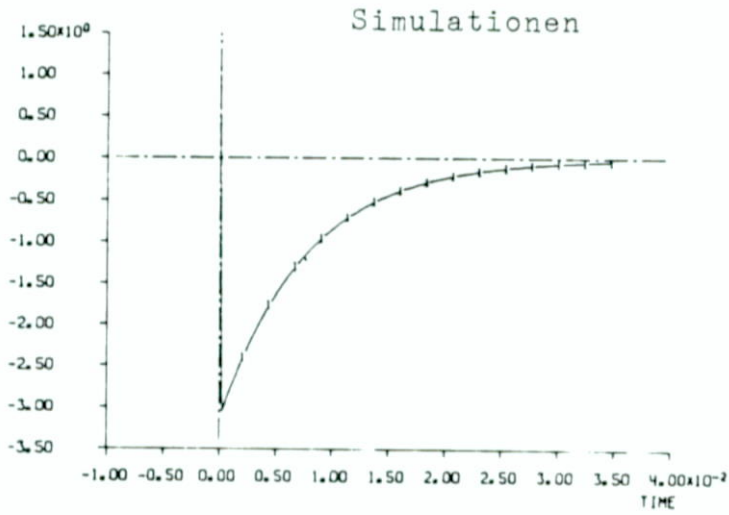


Fig. 5.32 Zustand  $x_1(t)$  bei  $\underline{\alpha}_{nom}$

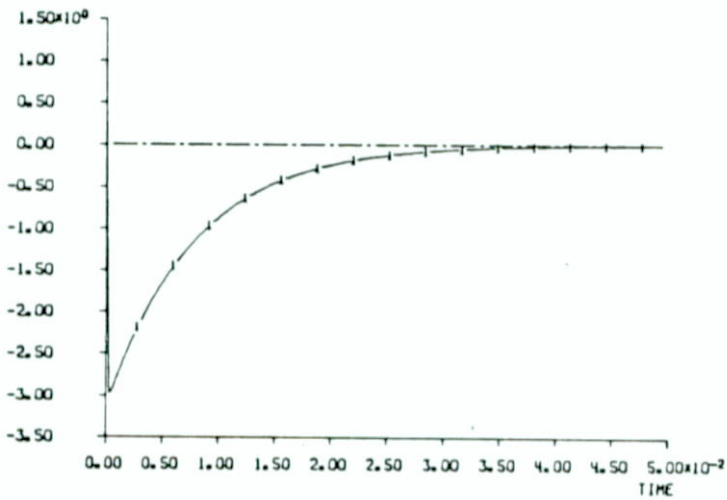


Fig. 5.33 Zustand  $x_1(t)$  bei  $\underline{\alpha}_{min}$

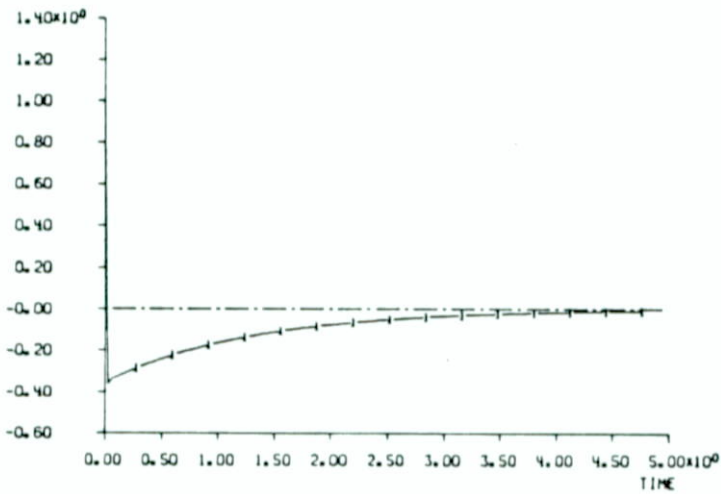


Fig. 5.34 Zustand  $x_1(t)$  bei  $\underline{\alpha}_{max}$

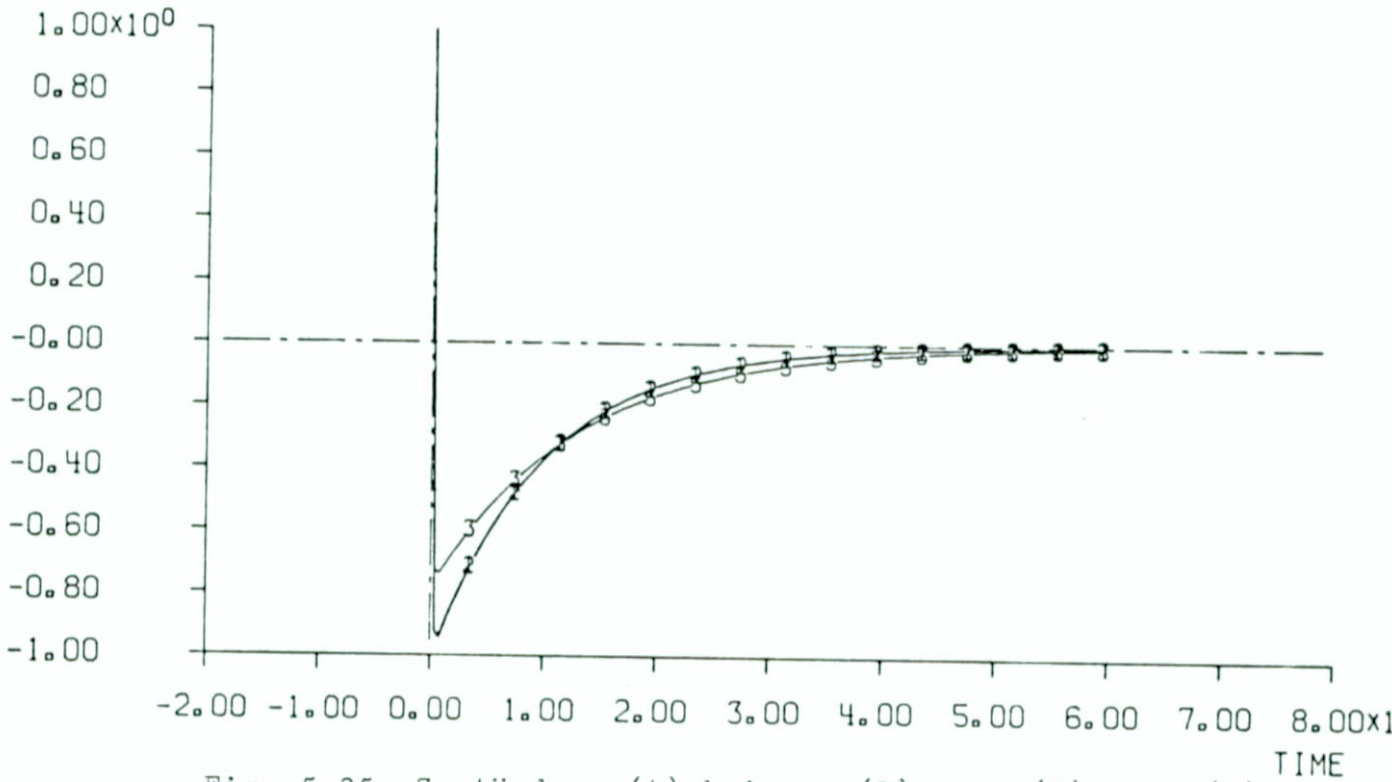


Fig. 5.35 Zustände  $x_2(t)$  bei  $\alpha_{nom}(1)$ ,  $\alpha_{min}(2)$ ,  $\alpha_{max}(3)$

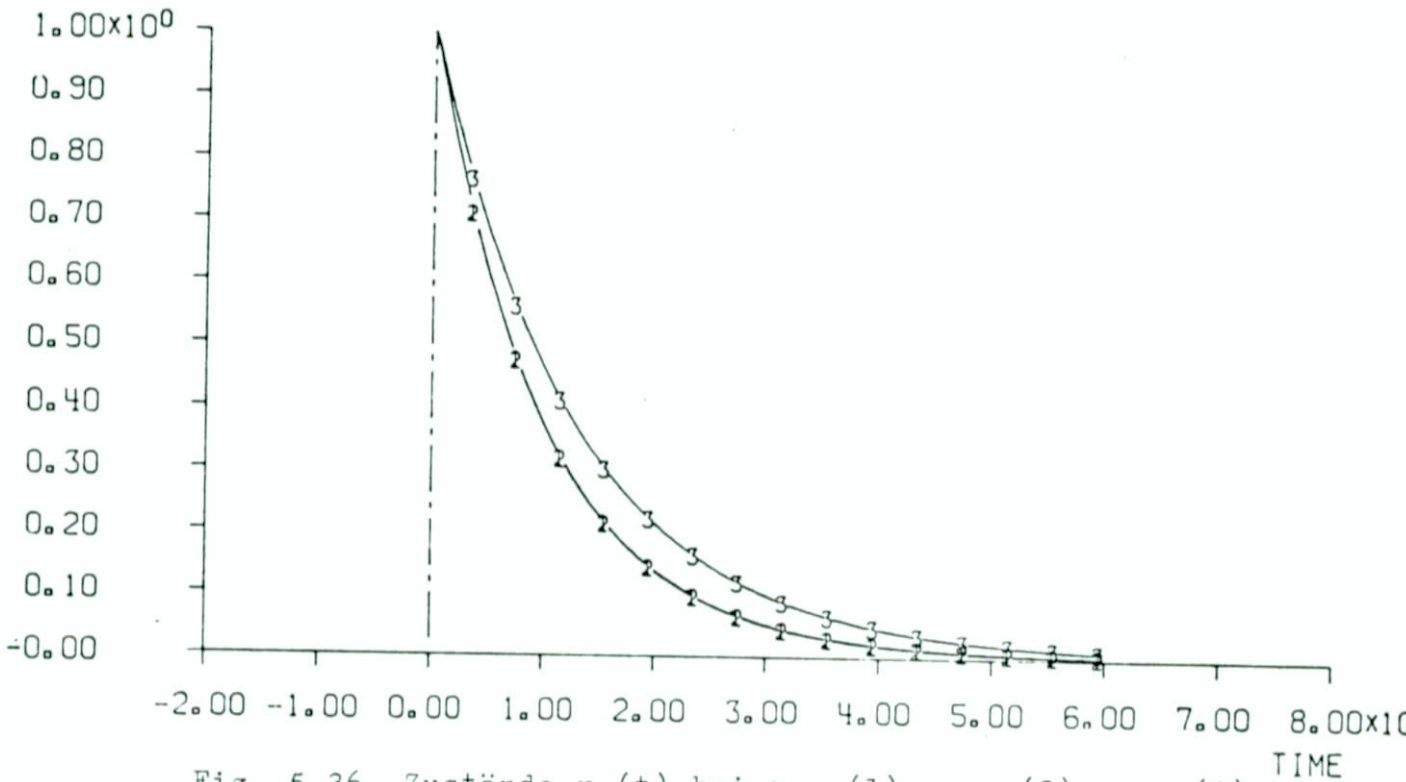


Fig. 5.36 Zustände  $x_3(t)$  bei  $\alpha_{nom}(1)$ ,  $\alpha_{min}(2)$ ,  $\alpha_{max}(3)$

Für die verschiedenen Werte von  $\lambda$ , bzw.  $L$ , wurden auch die Zielfunktionen berechnet. Die Werte sind in nachfolgender Tabelle zusammengefasst.

$\lambda$	J
0.5	0.653
10	0.711
50	0.788
1000	1.258

Auf den nächsten Seiten findet man die dazugehörigen Computer-Outputs.



$\lambda = 0.5$

```

L-MATRIX          1 ROWS      3 COLUMNS      STORAGE MODE 0
*****
COLUMN          1          2          3
ROW 1 -1.00010135 -1.57692982 -1.58113083

```

```

X0-VEKTOR        3 ROWS      1 COLUMNS      STORAGE MODE 0
*****

```

```

COLUMN          1
ROW 1  1.00000000
ROW 2  1.00000000
ROW 3  1.00000000

```

```

PERFORMANCE INDEX = .65312E+00
*****

```

$\lambda = 10$

L-MATRIX            1 ROWS    3 COLUMNS    STORAGE MODE 0

.....

COLUMN    1            2            3

ROW 1 -4.47231743    -7.05597778    -7.07106781

X0-VEKTOR            3 ROWS    1 COLUMNS    STORAGE MODE 0

.....

COLUMN    1

ROW 1 1.00000000  
ROW 2 1.00000000  
ROW 3 1.00000000

PERFORMANCE INDEX = .71146E+00  
\*\*\*\*\*

6.1  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67

$$\lambda = 50$$

L-MATRIX  
.....  
1 ROWS      3 COLUMNS      STORAGE MODE 0  
.....  
COLUMN    1                    2                    3  
ROW 1 -10.0001815      -15.7789754      -15.8113883

X0-VEKTOR  
.....  
3 ROWS      1 COLUMNS      STORAGE MODE 0  
.....  
COLUMN    1  
ROW 1 1.00000000  
ROW 2 1.00000000  
ROW 3 1.00000000

PERFORMANCE INDEX = .78794E+00  
\*\*\*\*\*

6.1\*  
7  
8  
9  
10  
11  
12 2'  
13  
14  
15  
16  
17  
18 3'  
19  
20  
21  
22  
23  
24 4'  
25  
26  
27  
28  
29  
30 5'  
31  
32  
33  
34  
35  
36 6'  
37  
38  
39  
40  
41  
42 7'  
43  
44  
45  
46  
47  
48 8'  
49  
50  
51  
52  
53  
54 9'  
55  
56  
57  
58  
59  
60 10'  
61  
62  
63  
64  
65  
66 11'  
67

$\lambda = 1000$

2-1000

L-MATRIX  
 .....  
 1 ROWS      3 COLUMNS      STORAGE MODE 0  
 .....  
 COLUMN      1                      2                      3  
 .....  
 ROW 1 -44.7215411      -70.5694578      -70.7106781

X0-VEKTOR  
 .....  
 3 ROWS      1 COLUMNS      STORAGE MODE 0  
 .....  
 COLUMN      1  
 .....  
 ROW 1 1.00000000  
 ROW 2 1.00000000  
 ROW 3 1.00000000

PERFORMANCE INDEX = .12578E+01  
 \*\*\*\*\*

51  
 61  
 7  
 8  
 9  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48  
 49  
 50  
 51  
 52  
 53  
 54  
 55  
 56  
 57  
 58  
 59  
 60  
 61  
 62  
 63  
 64  
 65  
 66  
 67

#### 5.3.4 Das optimal-geregelte System

Für  $\lambda = 1$  berechnet sich die Rückführungsmatrix L zu

$$L(\underline{\alpha}_{\text{nom}}, \lambda = 1) = ( -1.41 \quad -2.23 \quad -2.24 )'$$

Ausserdem wurde für  $\lambda = 1$  das entsprechende Robustheitsgebiet gefunden.

Für die Simulationen gelten die gleichen Aussagen, die in 5.3.3 gemacht worden sind, insbesondere die, die sich auf Fig. 5.39 bezieht.

Der Wert der Zielfunktion

$$J = 0.663$$

liegt natürlich zwischen den beiden berechneten Zielfunktionswerten für  $\lambda = 0.5$  und  $\lambda = 10$ .

Berechnung des Robustheitsgebietes  $\Omega_r$  und der Rückführungsmatrix L  
für  $\lambda = 1$

SYSTEMORDNUNG : 3  
ANZAHL STEUERVARIABLEN : 1  
ANZAHL PARAMETER : 5

SR : 0  
NGEB : 1  
LAMIN : .50  
LAMAX : 1.00  
RHO : 5.25

Q-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN 1 2 3

ROW 1 2.00000000 0. 0.  
ROW 2 0. 5.00000000 0.  
ROW 3 0. 0. 5.00000000

R-MATRIX 1 ROWS 1 COLUMNS STORAGE MODE 0

COLUMN 1  
ROW 1 1.00000000

A-QUER-MATRIX 3 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN 1 2 3

ROW 1 -15.5060000 0. 0.  
ROW 2 83.5260000 -1.26400000 0.  
ROW 3 0. 1.00000000 0.

B-QUER-MATRIX 3 ROWS 1 COLUMNS STORAGE MODE 0

COLUMN 1

ROW 1 -6.40732.000  
ROW 2 0.  
ROW 3 0.

6.1.  
 1 ALMIN-VEKTJR 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .100000000E-13 .103000000E-14 .105000000E-02 50.0000000 0.

18.3.  
 2 ALMAX-VEKTJR 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .150000000E-09 .350000000E-13 .110000000E-02 5000.00000 50.0000000

30.5.  
 3 ALMIG-VEKTJR 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .100000000E-12 .110000000E-14 .106000000E-02 100.000000 5.0000000

42.7.  
 4 ALMAG-VEKTJR 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .100000000E-10 .300000000E-13 .109000000E-02 1000.00000 40.0000000

60.0.  
 5 ALPHA-QUER 1 ROWS 5 COLUMNS STORAGE MODE 0  
 .....  
 COLUMN 1 2 3 4 5  
 ROW 1 .240000000E-11 .200000000E-13 .107500000E-02 150.00000 10.0000000

LAMBDA-MAX HURDE ZU KLEIN GEWAHLT

6.1.  
 7  
 8  
 9  
 10  
 11  
 12.2.  
 13  
 14  
 15  
 16  
 17  
 18.3.  
 19  
 20  
 21  
 22  
 23  
 24.4.  
 25  
 26  
 27  
 28  
 29  
 30.5.  
 31  
 32  
 33  
 34  
 35  
 36.6.  
 37  
 38  
 39  
 40  
 41  
 42.7.  
 43  
 44  
 45  
 46  
 47  
 48.8.  
 49  
 50  
 51  
 52  
 53  
 54.9.  
 55  
 56  
 57  
 58  
 59  
 60.0.  
 61  
 62  
 63  
 64  
 65  
 66.1.  
 67

DAS NEUE ROBUSTHEITSGEBIET WIRD GEGEBEN DURCH :

ALMIN = .366417105E-13 .105074378E-14 .105296346E-02 .500000000E+02 .140982618E+01

ALMAX = .108512435E-09 .335183012E-13 .109747751E-02 .381464099E+04 .470366025E+02

L-MATRIX 1 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN 1 2 3

ROW 1 -1.41439496 -2.23050083 -2.23606798

61  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68

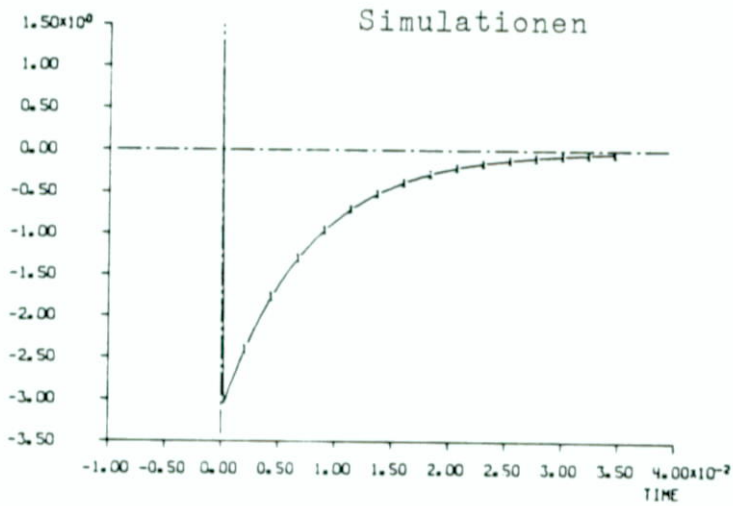


Fig. 5.37 Zustand  $x_1(t)$  bei  $\underline{\alpha}_{nom}$

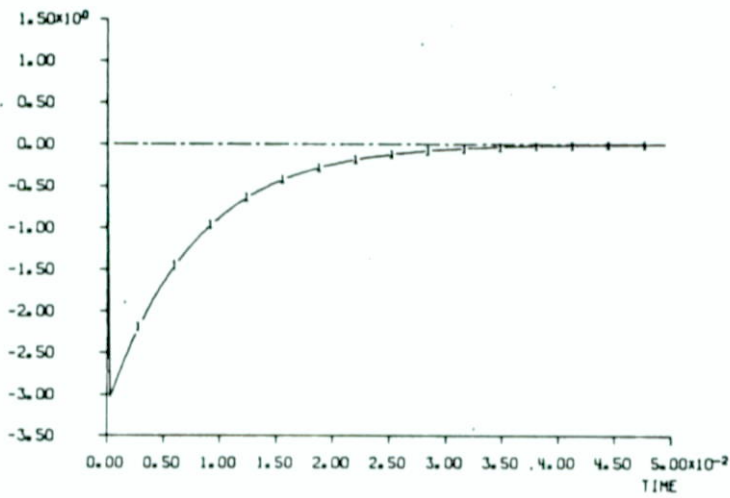


Fig. 5.38 Zustand  $x_1(t)$  bei  $\underline{\alpha}_{max}$

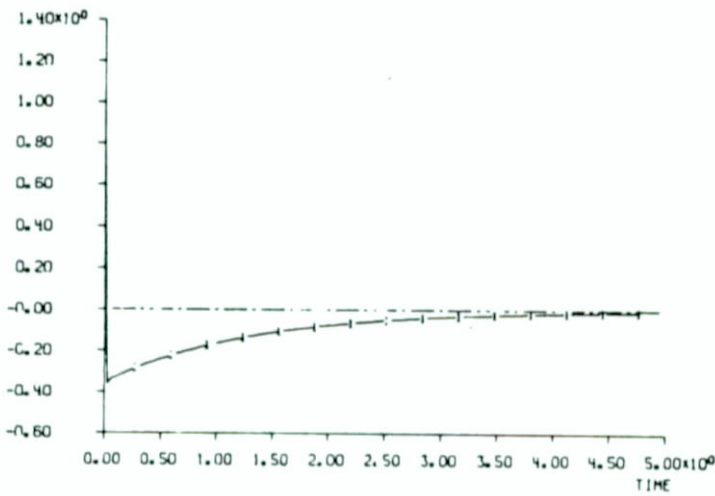


Fig. 5.39 Zustand  $x_1(t)$  bei  $\underline{\alpha}_{max}$

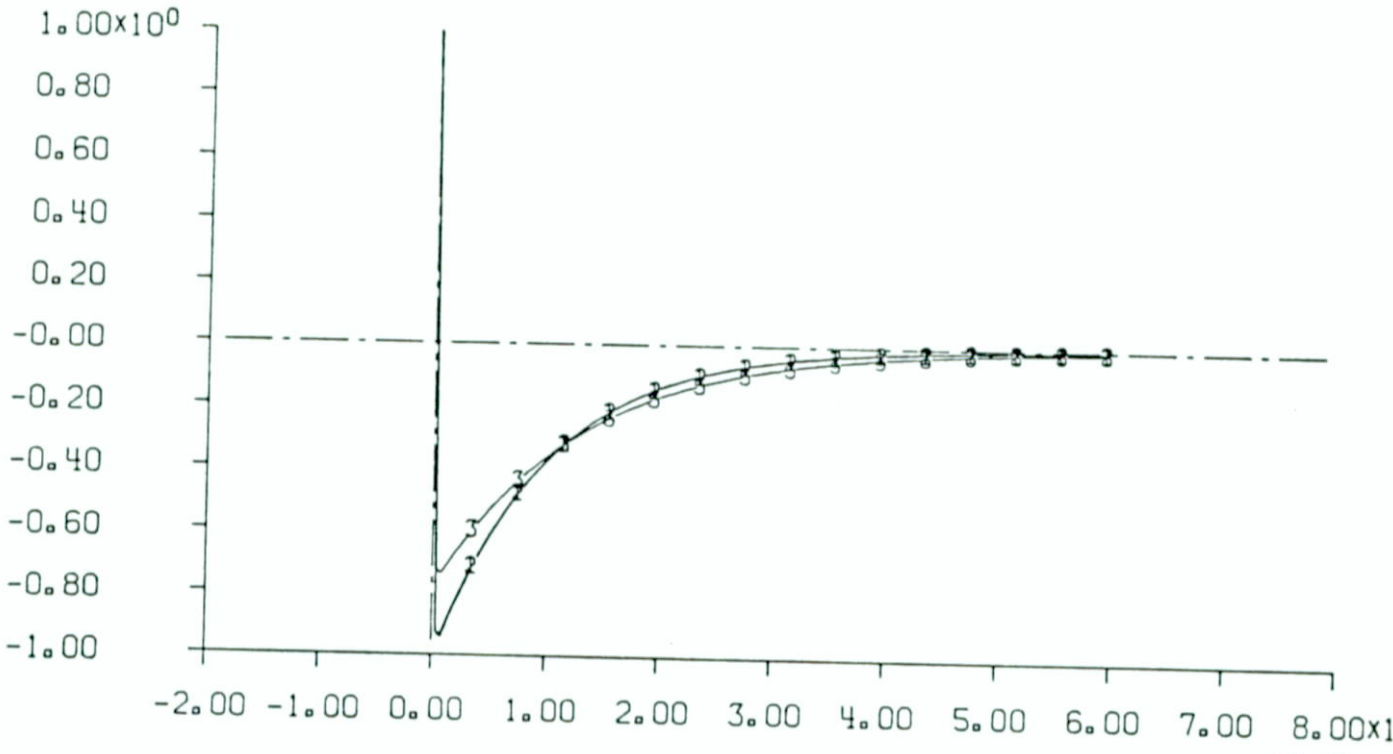


Fig. 5.40 Zustände  $x_2(t)$  für  $\alpha_{\text{nom}}(1)$ ,  $\alpha_{\text{min}}(2)$  und  $\alpha_{\text{max}}(3)$

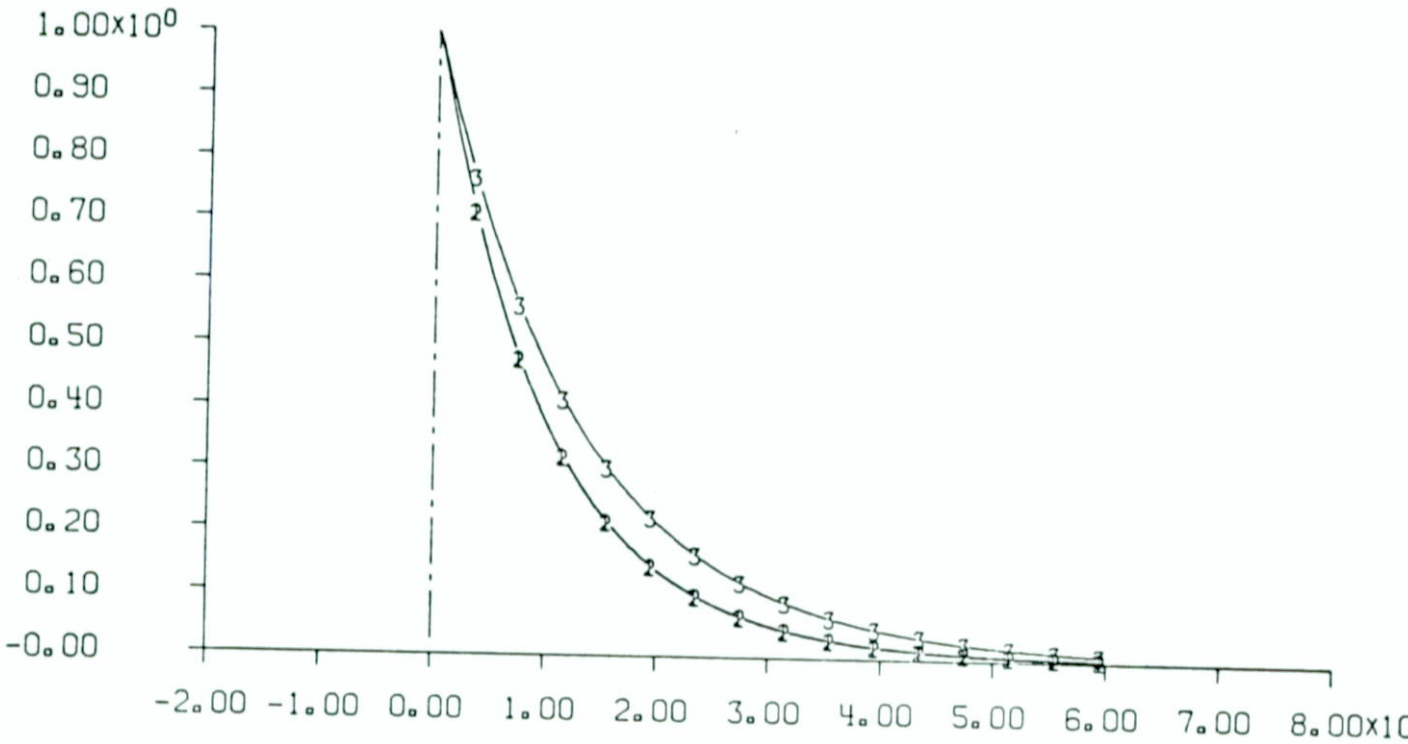


Fig. 5.41 Zustände  $x_3(t)$  für  $\alpha_{\text{nom}}(1)$ ,  $\alpha_{\text{min}}(2)$  und  $\alpha_{\text{max}}(3)$

Berechnung der Zielfunktion

SYSTEMORDNUNG = 3

ANZAHL STEUERVARIABLEN = 1

Q-MATRIX  
 .....  
 3 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN	1	2	3
ROW 1	2.00000000	0.	0.
ROW 2	0.	5.00000000	0.
ROW 3	0.	0.	5.00000000

R-MATRIX  
 .....  
 1 ROWS 1 COLUMNS STORAGE MODE 0

COLUMN	1
ROW 1	1.00000000

A-QUER-MATRIX  
 .....  
 3 ROWS 3 COLUMNS STORAGE MODE 0

COLUMN	1	2	3
ROW 1	-15.50000000	-688000000	1.00000000
ROW 2	60083.5000	.787000000	0.
ROW 3	280000.000	1.26400000	0.

B-QUER-MATRIX  
 .....  
 3 ROWS 1 COLUMNS STORAGE MODE 0

COLUMN	1
ROW 1	-640.732000
ROW 2	0.
ROW 3	0.



## 6. SCHLUSSBEMERKUNGEN UND UNGELOESTE PROBLEME

Die Methode [7], die in Kapitel 3 beschrieben, in Kapitel 4 programmiert und in Kapitel 5 angewendet wurde, stellt eine relativ einfache Möglichkeit dar, um robuste Regler zu entwerfen. Man erinnere sich an die Idee die dahinterstand: die Steuergrösse in einer quadratischen Zielfunktion  $J(\underline{a}_{\text{nom}}, \lambda)$  wurde schrittweise weniger und weniger gewichtet, bis ein sog. Insensitivitätskriterium erfüllt war, d.h. bis eine erwünschte Robustheit erreicht wurde. Je schärfer die Insensitivitätsbedingung, also je grösser die erreichte Robustheit, desto grösser die dazu aufzubringende Steuerenergie. Die  $\lambda_{\text{krit}}(\rho)$ -Kurven zeigten dies deutlich.

Die Vorteile dieser Methode gegenüber den in Kapitel 2 beschriebenen Verfahren, seien hier kurz zusammengebracht: erstens kommt diese Methode ohne differentiellen Sensitivitäten aus, sie arbeitet einzig mit einem skalaren und konstanten Sensitivitätskoeffizient  $\rho$ , zweitens ist sie auch für grosse Parameteränderungen anwendbar und schliesslich, wie anfänglich erwähnt, ist sie auch relativ einfach. Es wäre sehr interessant - vielleicht das Ziel einer nächsten Diplom- oder Semesterarbeit - die verschiedenen Algorithmen unter sich (Kreindler, Fleming und Newman, Rao und Soudack und die hier näher untersuchte Methode von Krishnan und Brzezowski) genauer zu vergleichen, z.B. hinsichtlich Systemantwort oder Trajektorien-sensitivität.

Andererseits muss gesagt werden, dass die Arbeit von Krishnan und Brzezowski nicht genügend theoretisch fundiert ist. Sie beruht auf den Beweis von Horisberger [2], der jedoch nur hinreichende - und keine notwendigen und hinreichenden - Bedingungen formuliert. Auf die Frage, ob es genügt nur an den Ecken des Hyperquaders im Parameterraum die Stabilitäts- und Robustheitsbedingungen zu prüfen sollte näher eingegangen wer-

den. Diese erforderlichen theoretischen Untersuchungen sind jedoch sehr anspruchsvoll, sie lagen sicherlich ausserhalb dem Rahmen unserer Arbeit. Womöglich existiert auch kein exakter Beweis für obiges Vorgehen. Auch wenn ein Gegenbeispiel gefunden werden kann, bei dem der Algorithmus nicht funktioniert, stellt [7] sicher eine zulässige und gut brauchbare Methode dar für viele Beispiele. Wir verwendeten die Methode bei Systemen bis zu 4. Ordnung und mit bis 6 Parametern und haben dabei gute Ergebnisse erzielt. Natürlich kann dies kein Beweis dafür sein, dass der Algorithmus immer "gut geht". Dazu das folgende Zitat von Newmann [5], in der am Anschluss seines Referates über seine Methode stattgefundenen Diskussion :

*"We have tended to treat examples that other people have treated. Obviously such an approach cannot be a proof that an algorithm always works. All we can say is that here is an alternative design procedure, we have tried it and it seems to work".*

Betreffend der Subroutine GEBIET sei darauf hingewiesen, dass anstelle der Verschiebung eines Eckpunktes des Hyperquaders in Richtung  $\underline{\alpha}_{nom}$ , mit  $\underline{\alpha}_{nom}$  fest, für den Fall, dass die Insensitivitätsbedingung verletzt wird, auch das Umgekehrte möglich wäre, nämlich die Verschiebung von  $\underline{\alpha}_{nom}$  in Richtung  $\underline{\alpha}$ . Die Autoren Krishnan und Brzezowski [7] empfehlen dieses Vorgehen, das jedoch zwei Nachteile besitzt : erstens ist meistens der Nominalwert der Parameter aus physikalischen Gründen gegeben und zweitens ist dies sehr arbeitsaufwendig. Falls nämlich  $\underline{\alpha}_{nom}$  in Richtung eines Eckpunktes verschoben wurde, kann es passieren, dass das Kriterium für die vorher untersuchte Ecke jetzt, mit dem neuen  $\underline{\alpha}_{nom}$ , nicht mehr erfüllt wird. Das heisst, dass für jede Verschiebung von  $\underline{\alpha}_{nom}$  jedesmal der Test ab der 1. Ecke nochmals durchzuführen ist. Es wäre jedoch interessant gewesen dieses praktisch durchzuführen. Dazu reichte uns leider nicht die Zeit.

LITERATURVERZEICHNIS

- [1] *B.Itschner* : Stabilitätsanalyse mit Hilfe idealer Ljapunov-Funktionen. Regelungstechnik, Heft 8, 1977.
- [2] *P.H.Horishberger, R.P.Belanger* : Regulators for linear, time invariant plants with uncertain parameters. IEEE Trans. on Automatic Control, Vol. AC-21, pp.705-708, 1976.
- [3] *E.Kreindler* : IEEE Trans. on Automatic Control, Vol. AC-14, pp. 206-207, 1969.
- [4] *S.G.Rao, A.C.Soudack* : IEEE Trans. on Automatic Control, Vol. AC-16, pp. 194-196, 1971.
- [5] *P.J.Fleming, M.M.Newmann* : Trajectory sensitivity reduction in the optimal linear regulator. Recent Mathematical Developments in Control, D.J.Bell Ed. New York, pp. 137-151, 1973.
- [6] *R.V.Patel, B.Snidhar, M.Toda* : Robustness in linear quadratic feedback design with application to an aircraft control problem. Proc. of the Asilomar Conf. on Circuits, Systems and Computers, Vol. 10, pp. 293-300, 1976.
- [7] *K.R.Krishnan, S.Brzezowski* : Design of robust linear regulators with prescribed trajectory insensitivity to parameter variations. IEEE Trans. on Automatic Control, Vol. AC-23, pp.474-478, 1978.
- [8] *M.Mansour* : Nichtlineare Regelsysteme. AMIV-Verlag, 1978.
- [9] *Gantmacher* : Matrix-Theory, Chelsea, Vol.1, p. 225, 1959.

- [10] *B.Porter* : Synthesis of Dynamic Systems. New York, Barnes and Noble, 1969.
- [11] *G.Lekkas u.a.* : Identifikation dynamischer Systeme durch nichtlinearen Modellabgleich. Interner Bericht der Fachgruppe für Automatik, No. 77-04.
- [12] *M.Senning* : Regelung einer hydraulischen Werkzeugmaschine. Fachgruppe für Automatik, 1980.
- [13] *E.Kreindler* : Closed-loop sensitivity reduction of linear optimal control systems. IEEE Trans. on Automatic Control, Vol. AC-13, pp.254-262, 1968.