

Diplomarbeit

Simulation eines nichtlinearen Modells für einen
magnetgelagerten Mehrscheibenrotor

ausgeführt am Institut für
Maschinendynamik und Meßtechnik
der technischen Universität Wien

unter Anleitung von

o. Univ. Prof. Dr. techn. Helmut Springer

durch

Peter Wurmsdobler

Matr. Nr.: 8626655
A-4762 St. Willibald
Oberantlang 6

Wien, April 1992

Vorwort

Als Student war es mein Wunsch, in einer Diplomarbeit mehrere Gebiete des Maschinenbaus zu integrieren. Dem konnte mit einer Simulation eines Rotors auf Magnetlagern entsprochen werden. So bot das relativ aktuelle Forschungsgebiet "Magnetgelagerte Rotoren" die Möglichkeit, Maschinendynamik, Regelungstechnik und Elemente der Elektrotechnik zu verbinden.

Die Simulation wurde mit dem programmierbaren Softwarepaket MATLAB [8] durchgeführt. Die Arbeit selbst entstand unter Einbindung aller in AutoCAD erzeugten Graphiken mit L^AT_EX [9], und wurde an einem PostScript-Laserdrucker ausgegeben.

An dieser Stelle sei allen gedankt, die mir bei dieser Arbeit durch Anregungen und konstruktive Diskussionen geholfen haben, vor allem dem Institutsvorstand Prof. Helmut Springer, sowie den Assistenten des Instituts für Maschinendynamik und Meßtechnik. Ich will aber auch allen Freunden, und besonders denen, die mir in Ermangelung eines eigenen Computers ihren Rechner zu Verfügung stellten, oder in zahlreichen Gesprächen neue Aspekte einbrachten, meinen aufrichtigen Dank aussprechen.

Inhaltsverzeichnis

1	Einleitung	3
1.1	Problemstellung	3
1.2	Formulierung des Simulationsmodells und Zergliederung in Teilmodelle . . .	4
2	Modellbildung des mechanischen Teilsystems	7
2.1	Erstellen der Systemmatrizen	8
2.2	Kondensation des Modells	13
2.3	Dynamische Analyse	17
3	Modell des Magnetlagers	35
3.1	Hysteresemodell des magnetischen Werkstoffes	35
3.2	Differentialgleichungen des magnetischen Kreises mit Luftspalt	37
3.3	Magnetlagergeometrie und Lagersteifigkeiten	39
4	Linearisiertes Modell als Grundlage für die Auslegung eines Reglers	46
4.1	Linearisiertes Modell für einen Betriebspunkt	46
4.2	Reglerkonzepte und Stabilität	50
5	Simulation des nichtlinearen Gesamtsystems	63
5.1	Zustandsform des gesamten Systems	63
5.2	Simulationsergebnisse	65
6	Zusammenfassung und Ausblick	81
A	Anhang	84
A.1	Simulationsprogramm mit Input-File	85
A.2	Hilfsprogramme	102

Kapitel 1

Einleitung

1.1 Problemstellung

Dem Titel gerecht werdend, wird in der vorliegenden Diplomarbeit ein nichtlineares Modell eines Mehrscheiben-Rotors auf Magnetlagern erstellt und dessen dynamisches Verhalten mit einem passenden Computerprogramm simuliert. Die Ergebnisse sind als theoretische Vorarbeit für ein am Institut geplantes Projekt gedacht, in dessen Rahmen ein Versuchsstand aufgebaut werden soll.

Arbeiten über Magnetlager gibt es mittlerweile genug. Allerdings wird in den meisten nur deren lineares Verhalten um einen Arbeitspunkt untersucht, oder ein nichtlineares Ein- oder Zweifreiheitsgradmodell eines Magnetlagers mit einer für den Rotor repräsentativen Punktmasse. Dies ist zumeist ausreichend, aber jedenfalls notwendig, um einen Regler auszulegen und die Stabilität des geschlossenen Systems zu untersuchen. Hier soll ein FE-Modell eines Mehrscheibenrotors mit einem nichtlinearen Modell des Magnetlagers mit Digitalregler zu einem nichtlinearen Gesamtmodell mit mehreren Freiheitsgraden kombiniert werden.

Das nichtlineare Modell des Magnetlagers selbst bildet die eine Grundlage für eine Simulation des Rotor-Magnetlager-Systems. Dieses Modell wird zum Großteil einer Diplomarbeit über das nichtlineare Verhalten eines ferromagnetischen Werkstoffes [1] und einer Arbeit von Springer [2] entnommen.

Als weiterer Teil dient ein FE-Modell für die Rotorwelle, auf der diskrete Scheiben angebracht sein können. Zumal eine FE-Modellierung meist einen großen rechnerischen Aufwand darstellt, wird eine statische, oder eine modale Kondensation durchgeführt, wodurch die Dimension der Matrizen stark reduziert werden kann, und damit auch der Aufwand für die numerische Simulation.

Da aktive Magnetlager nur durch einen Regler stabilisiert werden können, ergibt sich als dritter Teil der Modellierung die Auslegung und Einbindung eines digitalen Reglers in die Simulation eines kontinuierlichen Systems. Wie bereits erwähnt, muß dafür das System um einen Arbeitspunkt linearisiert werden. Es kann natürlich auch das nichtlineare Verhalten des Magnetlagers mit diesem Regler simuliert werden, um so die Grenze der Verwendbarkeit des Reglers zu bestimmen.

Als Synthese dieser Teilmodelle entsteht ein nichtlineares Differentialgleichungssystem erster Ordnung, welches in der Folge mit einem Integrationsprogramm numerisch bearbeitet wird. Da die Gleichungen in Matrixschreibweise formuliert werden, und lineare Anteile wesentliche Bestandteile der Gleichungen sind, eignet sich ein für Matrizenoperationen geschaffenes Programm am besten. Die Wahl fiel auf MATLAB. Als noch jüngerer, offener, das heißt noch programmierbares, Softwarepaket hat es zwar den Nachteil über weniger Bibliotheken zu verfügen, steht aber als Programmiersprache doch schon wesentlich höher als die meisten gängigen Simulationssprachen.

1.2 Formulierung des Simulationsmodells und Zergliederung in Teilmodelle

Bevor das Simulationsmodell genauer definiert wird, soll schematisch eine mögliche Ausführungsform eines Mehrscheibenrotors auf Magnetlagern, dargestellt werden.

(Abb. 1.1)

Es handelt sich hierbei im wesentlichen um einen n -Scheiben-Rotor, der auf zwei Magnetlagern gelagert ist, und von einer Seite angetrieben wird. Das zu erstellende Modell soll jedoch so allgemein sein, daß es die Anzahl der Scheiben und Lager als freie Parameter enthält und auch die Position der Lager wählbar ist. Somit wird es beispielsweise möglich sein, einen Rotor mit einer Scheibe außerhalb der Lager in Hinblick auf den größeren Einfluß der Kreiselwirkung der Rotorscheiben zu untersuchen, ohne die Programmstruktur zu verändern.

Die Anordnung zerfällt in folgende für die Simulation relevanten Teile:

- Elastischer Rotor,
- Magnetlager mit integrierter Meßeinrichtung,
- Regler samt Verstärkungseinheit.

Die Anordnung dieser Glieder läßt sich übersichtlich in einem Blockschaltbild darstellen.

(Abb. 1.2)

Im obigen Blockschaltbild liegt die Nichtlinearität nur im Magnetlager, das zusätzlich noch als ein mit dem Lagevektor des Rotors moduliertes Glied betrachtet werden kann. Sowohl der Regler als auch der Rotor werden als lineare Glieder beschrieben. Da nicht der ganze Zustandsvektor gemessen werden kann, enthält der Vektor \mathbf{X} bloß die Verschiebungen der Lagerzapfen in der Ebene normal zur Wellenachse. Diese Verschiebungen werden dann negativ rückgekoppelt und dem Regler zugeführt, der als Stellgrößen die Steuerspannungen für die Magnetlager erzeugt.

In den folgenden Kapiteln werden die einzelnen Teilsysteme besprochen, um ein gesamtes Modell für die Simulation bilden zu können.

Kapitel 2

Modellbildung des mechanischen Teilsystems

Für das Teilsystem elastischer Rotor wird ein lineares Modell betrachtet, und es werden gewisse Vereinfachungen vorgenommen:

- Entkoppelung von Biege und Torsionsschwingungen, d.h. Koppeleffekte zwischen Biegung und Torsion werden vernachlässigt. Dies ist in Hinblick auf einen zu erwartenden hohen Gleichlauf des Antriebes und eines guten Auswuchtzustandes des Rotors gerechtfertigt. Im System treten außerdem keine statischen und dynamischen Axialkräfte auf.
- Die Welle wird durch finite zylindrische Biegeelemente des Bernoulli-Euler-Typs diskretisiert. Die Biegelinie, auf der die Wellenmasse konzentriert gedacht ist, wird durch Ansatzfunktionen angenähert.
- Rotorscheiben werden durch starre dünne Scheiben, deren translatorische und rotatorische Trägheit, sowie deren Kreiselwirkung berücksichtigt wird, in den Modellknoten ersetzt.
- Zur Untersuchung des dynamischen Verhaltens des Rotors werden die geregelten Magnetlager zunächst durch Ersatzsteifigkeiten und Ersatzdämpfungen in ihren Betriebspunkten modelliert. Kupplungen können als biegenachgiebige Lager an den Enden des Rotors betrachtet werden.
- Als externe Erregerkräfte sind nur die Unwucht und Schwerkraft von Bedeutung.

Die allgemeine Form der linearisierten Bewegungsgleichungen für ein zwangserregtes Rotor-Wellen-Lager Modell ohne zirkulatorische Lagekräfte lautet:

$$\mathbf{M} \ddot{\mathbf{x}} + (\mathbf{C} + \mathbf{G}) \dot{\mathbf{x}} + \mathbf{K} \mathbf{x} = \mathbf{f}(t) \quad (2.1)$$

$\mathbf{M} = \mathbf{M}^T$	Trägheitsmatrix
$\mathbf{C} = \mathbf{C}^T$	Dämpfungsmatrix
$\mathbf{G} = -\mathbf{G}^T$	Gyroskopiematrix
$\mathbf{K} = \mathbf{K}^T$	Steifigkeitsmatrix
\mathbf{x}	Lagevektor der Freiheitsgrade
$\mathbf{f}(t)$	Externer Erregervektor

In der weiteren Folge geht es darum, diese Matrizen aus den Daten des Rotors, (Abmessungen der Welle und die Gestalt der Scheiben), sowie Informationen über Lager und Kupplung, aufzubauen und das entstehende System gegebenenfalls zu kondensieren und numerisch auszuwerten [5, 6].

2.1 Erstellen der Systemmatrizen

Ein Rotor hat beispielsweise die Gestalt nach Abb. 2.1 . Das kontinuierliche System wird, um einer Simulation zugänglich zu sein in folgender Weise diskretisiert. Es besteht aus n Knoten, die die massebehaftete elastische Welle in $n - 1$ FE des Bernoulli-Euler-Typs unterteilen. Auf den Knoten können nun Lager, Scheiben oder Kupplungen, oder eine sinnvolle Kombination aus diesen Komponenten angebracht werden.

Knotenbestimmte Matrizen

In einem Koordinatensystem läßt sich ein Knoten, auf dem z.B. eine Scheibe sitzt, gemäß Abb. 2.2 darstellen:

Jedem der Knoten k werden 4 Freiheitsgrade der Biegedeformation zugeordnet, und die Verschiebungen und Verdrehungen zu einem lokalen Lagevektor zusammengefaßt, der den jeweiligen Knoten charakterisiert. (Der Index k wird nicht extra an die einzelnen Knotenvariablen angefügt)

Der lokale Lagevektor ist definiert zu

$$\mathbf{q} = (v, w, \beta, \gamma)^T$$

und der Vektor der virtuellen Verschiebungen:

$$\delta\mathbf{q} = (\delta v, \delta w, \delta\beta, \delta\gamma)^T$$

Die Verschiebungen v, w sollen in Relation zu einer charakteristischen Länge klein sein und $\beta \ll 1, \gamma \ll 1$. Unter Vernachlässigung der Axial- und Torsionsschwingungen ($\Omega = \text{const.}$) folgt für die Geschwindigkeiten mit $d/dt = (\cdot)$:

$$\dot{\mathbf{r}}_S \doteq \begin{pmatrix} 0 \\ \dot{v} \\ \dot{w} \end{pmatrix}, \quad \dot{\boldsymbol{\omega}} \doteq \begin{pmatrix} \Omega \\ \dot{\beta} + \Omega\gamma \\ \dot{\gamma} - \Omega\beta \end{pmatrix}$$

und für die virtuellen Verschiebungen bzw. Verdrehungen

$$\delta\mathbf{r}_S \doteq \begin{pmatrix} 0 \\ \delta v \\ \delta w \end{pmatrix}, \quad \delta\boldsymbol{\alpha} \doteq \begin{pmatrix} 0 \\ \delta\beta \\ \delta\gamma \end{pmatrix}.$$

Mit dem Impuls, bzw. dem Drall einer Scheibe

$$\vec{\mathbf{p}}_S \doteq \begin{pmatrix} 0 \\ m\dot{v} \\ m\dot{w} \end{pmatrix}, \quad \vec{\mathbf{L}}_S \doteq \begin{pmatrix} I_p\Omega \\ I_a\dot{\beta} + I_p\Omega\gamma \\ I_a\dot{\gamma} - I_p\Omega\beta \end{pmatrix}$$

läßt sich die virtuelle Arbeit der Trägheitskräfte einer Scheibe anschreiben:

$$\delta W_T = -\delta\mathbf{r}_S \cdot \dot{\vec{\mathbf{p}}}_S - \delta\boldsymbol{\alpha} \cdot \dot{\vec{\mathbf{L}}}_S \quad (2.2)$$

Unter Verwendung des Lagevektors \mathbf{q} und dessen virtueller Variation $\delta\mathbf{q}$ kann die Gleichung 2.2 in der Form geschrieben werden:

$$\delta W_T = -\delta\mathbf{q}^T \mathbf{M}_R \ddot{\mathbf{q}} - \delta\mathbf{q}^T \mathbf{G}_R \dot{\mathbf{q}} \quad (2.3)$$

mit den Matrizen

$$\mathbf{M}_R = \begin{pmatrix} m & 0 & & \\ 0 & m & & \\ & & I_a & 0 \\ & & 0 & I_a \end{pmatrix} \quad \text{und}$$

$$\mathbf{G}_R = \begin{pmatrix} 0 & 0 & & \\ 0 & 0 & & \\ & & 0 & I_p \\ & & 0 & -I_p \end{pmatrix} \Omega$$

Die Massenmatrix \mathbf{M}_R ist eine Diagonalmatrix, die Gyroskopiematrix \mathbf{G}_R ist schief-symmetrisch und berücksichtigt die Kreiselwirkung der Rotorscheiben.

Sitzt auf einem Knoten eine Kupplung, oder ist an einem Knoten ein Lager angebracht, so wird dieses mit einer Steifigkeitsmatrix \mathbf{K}_L und einer Dämpfungsmatrix \mathbf{C}_L in die Modellierung miteinbezogen. Eine Kupplung wird so einerseits als nachgiebiges Zwischenlager betrachtet, andererseits wird sie als Rotorscheibe auch massenmäßig modelliert.

Für die Steifigkeitsmatrizen werden die Hauptsteifigkeiten in den jeweiligen Lagerhauptrichtungen, die mit den Inertialrichtungen y und z zusammenfallen sollen, herangezogen. Die Matrizen sind Diagonalmatrizen, in der die Steifigkeiten bzw. Dämpfungen die Stelle der jeweiligen Freiheitsgrade besetzen. Steifigkeit und Dämpfung ergeben sich aus den Strom- und Lagekoeffizienten des Magnetlagers und werden in Hinblick auf die statische Kondensation des Rotormodells bevorzugt behandelt.

$$\mathbf{K}_L = \begin{pmatrix} k_y & 0 & 0 & 0 \\ 0 & k_z & 0 & 0 \\ 0 & 0 & k_\beta & 0 \\ 0 & 0 & 0 & k_\gamma \end{pmatrix}$$

$$\mathbf{C}_L = \begin{pmatrix} c_y & 0 & 0 & 0 \\ 0 & c_z & 0 & 0 \\ 0 & 0 & c_\beta & 0 \\ 0 & 0 & 0 & c_\gamma \end{pmatrix}$$

Somit kann jedem Knoten k eine Massenmatrix $\mathbf{M}_R^{(k)}$, eine Gyroskopiatrix $\mathbf{G}_R^{(k)}$, eine Dämpfungsmatrix $\mathbf{C}_L^{(k)}$ und eine Steifigkeitsmatrix $\mathbf{K}_L^{(k)}$ zugeordnet werden, je nachdem ob sich dort Scheiben, Lager oder beides befinden.

Elementbestimmte Matrizen

Zwischen zwei Knoten befindet sich jeweils ein finites zylindrisches Bernoulli-Euler-Element (Abb. 2.3), dessen statische Biegelinie bei Abwesenheit von Feldlasten den Differentialgleichungen

$$EIv'''' = 0$$

$$EIw'''' = 0$$

genügt, wobei $(\)' = d/dx$ die Ableitung nach der axialen Ortskoordinate x bezeichnet.

Viermalige Integration ergibt ein Polynom dritten Grades für die Biegelinie, dessen Konstanten sich aus der Erfüllung der Randbedingungen (Knotenverschiebungen und -verdrehungen) ergeben. Diese Polynome (=Hermite-Polynome) werden als Ritz'sche Ansatzfunktionen näherungsweise auch für dynamische Deformationen der Welle verwendet und man erhält

$$\vec{\mathbf{u}} = \mathbf{N} \mathbf{u} \tag{2.4}$$

mit dem Verschiebungsfeldvektor eines **FE** für die Biegeverformung

$$\vec{\mathbf{u}}(x, t) = \begin{pmatrix} v(x, t) \\ w(x, t) \end{pmatrix},$$

der Formfunktionsmatrix eines **FE**

$$\mathbf{N}(x) = \begin{pmatrix} H_1(x) & 0 & 0 & H_2(x) & H_3(x) & 0 & 0 & H_4(x) \\ 0 & H_1(x) & -H_2(x) & 0 & 0 & H_3(x) & -H_4(x) & 0 \end{pmatrix}$$

und dem Spaltenvektor, der die zeitabhängigen Verschiebungen und Verdrehungen eines **FE** an den Knoten enthält

$$\mathbf{u}(t) = (v_1, w_1, \beta_1, \gamma_1, v_2, w_2, \beta_2, \gamma_2)^T.$$

Die Hermite'schen Ansatzfunktionen sind gegeben durch

$$\begin{aligned} H_1 &= 1 - 3\xi^2 + 2\xi^3 \\ H_2 &= \xi(1 - \xi)^2 l \\ H_3 &= 3\xi^2 - 2\xi^3 \\ H_4 &= -\xi^2(1 - \xi)l \end{aligned}$$

mit $\xi = x/l$, [5, 6].

Massenmatrix des finiten Wellen-Elements

Wenn die Kreiselwirkung der Wellenquerschnitte vernachlässigt wird, gilt für die virtuelle Arbeit der Trägheitskräfte:

$$\delta W_T = - \int_m \delta \ddot{\mathbf{u}} \cdot \ddot{\mathbf{u}} \, dm \quad (2.5)$$

Mit Gleichung 2.4 und

$$dm = \rho A l \, d\xi$$

ergibt sich

$$\delta W_T = -\delta \mathbf{u}^T \mathbf{M}_W \ddot{\mathbf{u}}$$

wobei

$$\mathbf{M}_W = A \rho l \int_0^1 \mathbf{N}^T \mathbf{N} \, d\xi$$

die Massenmatrix des Wellenelements bedeutet.

Steifigkeitsmatrix des finiten Wellen-Elements

Da ein Bernoulli-Euler-Element verwendet wird, sind die schubspannungsbedingten Deformationen vernachlässigt, und es gilt bei verschwindenden Axialkräften für die Biegeverzerungsenergie des Wellenelements

$$U = \frac{1}{2} \int_0^l EJ(v''^2 + w''^2) dx.$$

Die virtuelle Arbeit der elastischen Kräfte im Element ist

$$\delta W_{el} = -\delta U = - \int_0^1 l EJ \delta \bar{\mathbf{u}}'' \cdot \bar{\mathbf{u}}'' d\xi \quad (2.6)$$

oder mit der Annahme von Formfunktionen

$$\delta W_{el} = -\delta \mathbf{u}^T \mathbf{K}_W \mathbf{u}$$

mit der Steifigkeitsmatrix eines Wellenelementes

$$\mathbf{K}_W = l \int_0^1 EJ \mathbf{N}''^T \mathbf{N}'' d\xi.$$

Somit ergibt sich für ein Wellenelement e eine zugehörige Massenmatrix $\mathbf{M}_W^{(e)}$ und eine Steifigkeitsmatrix $\mathbf{K}_W^{(e)}$.

Zusammenbau der Systemmatrizen

Der globale Lagevektor zur Beschreibung der Deformationen des Rotors wird durch Hintereinanderreihen der einzelnen lokalen Verschiebungsvektoren der Knoten erhalten.

$$\mathbf{x} = (\mathbf{q}_1^T, \dots, \mathbf{q}_n^T)^T$$

Der globale Lastvektor hat analoge Gestalt:

$$\mathbf{f} = (\mathbf{f}_1^T, \dots, \mathbf{f}_n^T)^T.$$

Die globalen Matrizen werden nach folgendem Formalismus zusammengebaut:
(Koinzidenztransformation, siehe [6])

$$\begin{aligned}
\mathbf{M} &= \sum_{k=1}^n \mathbf{T}^{(k)\mathbf{T}} \mathbf{M}_R^{(k)} \mathbf{T}^{(k)} + \sum_{e=1}^{n-1} \mathbf{T}^{(e)\mathbf{T}} \mathbf{M}_W^{(e)} \mathbf{T}^{(e)} \\
\mathbf{C} &= \sum_{k=1}^n \mathbf{T}^{(k)\mathbf{T}} \mathbf{C}_L^{(k)} \mathbf{T}^{(k)} \\
\mathbf{G} &= \sum_{k=1}^n \mathbf{T}^{(k)\mathbf{T}} \mathbf{G}_R^{(k)} \mathbf{T}^{(k)} \\
\mathbf{K} &= \sum_{k=1}^n \mathbf{T}^{(k)\mathbf{T}} \mathbf{K}_L^{(k)} \mathbf{T}^{(k)} + \sum_{e=1}^{n-1} \mathbf{T}^{(e)\mathbf{T}} \mathbf{K}_W^{(e)} \mathbf{T}^{(e)}
\end{aligned}$$

Die Matrizen $\mathbf{T}^{(k)}$ und $\mathbf{T}^{(e)}$ ergeben sich aus der Koinzidenz zwischen globalen und lokalen Verschiebungen. Der lokale Lagevektor eines Knotens, sowie der Verschiebungsvektor eines FE, sind wiederum formal gegeben durch:

$$\begin{aligned}
\mathbf{q}^{(k)} &= \mathbf{T}^{(k)\mathbf{T}} \mathbf{x} \\
\mathbf{u}^{(e)} &= \mathbf{T}^{(e)\mathbf{T}} \mathbf{x}
\end{aligned}$$

Für den Knoten k ist die Koinzidenzmatrix

$$\mathbf{T}^{(k)} = (\mathbf{0}, \dots, \mathbf{E}, \dots, \mathbf{0})$$

mit der 4*4 Einheitsmatrix \mathbf{E} im k -ten Feld gegeben. Für das Finite Element e , begrenzt durch die Knoten k und $k + 1$ ergibt sich die Koinzidenzmatrix zu:

$$\mathbf{T}^{(e)} = \begin{pmatrix} \mathbf{T}^{(k)} \\ \mathbf{T}^{(k+1)} \end{pmatrix}$$

Da die Matrizen $\mathbf{T}^{(k)}$ und $\mathbf{T}^{(e)}$ viele Nullelemente enthalten, wird die programmtechnische Ausführung der obigen Zusammenbauvorschrift unter Vermeidung der Nullmultiplikationen ausgeführt.

2.2 Kondensation des Modells

Mit den im vorherigen Abschnitt bestimmten Matrizen ergibt sich beispielsweise für einen 20 Knoten-Rotor ein 80×80 -Gleichungssystem in Bandstruktur mit $HB = 8$. Es ist also von Nutzen, das System zu reduzieren, um den numerischen Aufwand für eine Simulation zu verringern. Dabei darf das System aber in seinem Eigenverhalten nicht wesentlich verändert werden. Dies geschieht in einer Kondensation des Modells mit n Freiheitsgraden auf ein Modell mit $m < n$ Freiheitsgraden, oder formal ausgedrückt:

$$\mathbf{x} = \mathbf{R} \mathbf{y} \quad \text{und} \quad \delta \mathbf{x}^T = \delta \mathbf{y}^T \mathbf{R}^T$$

wobei \mathbf{y} den Vektor der neuen Freiheitsgrade bezeichnet und \mathbf{R} die $m \times n$ Reduktionsmatrix. Die d'Alembertsche Form der Bewegungsgleichung 2.1 lautet:

$$\delta \mathbf{x}^T \left(\mathbf{M} \ddot{\mathbf{x}} + (\mathbf{C} + \mathbf{G}) \dot{\mathbf{x}} + \mathbf{K} \mathbf{x} - \mathbf{f}(t) \right) = 0$$

oder unter Zuhilfenahme obiger Transformation:

$$\delta \mathbf{y}^T \left(\mathbf{R}^T \mathbf{M} \mathbf{R} \ddot{\mathbf{y}} + (\mathbf{R}^T \mathbf{C} \mathbf{R} + \mathbf{R}^T \mathbf{G} \mathbf{R}) \dot{\mathbf{y}} + \mathbf{R}^T \mathbf{K} \mathbf{R} \mathbf{y} - \mathbf{R}^T \mathbf{f}(t) \right) = 0$$

Als Ergebnis entstehen $m \times m$ - Systemmatrizen, deren Symmetrien bzw. Antimetrien erhalten bleiben:

$$\begin{aligned} \mathbf{M}^* &= \mathbf{R}^T \mathbf{M} \mathbf{R} = \mathbf{M}^{*\text{T}} && \text{reduzierte Trägheitsmatrix} \\ \mathbf{C}^* &= \mathbf{R}^T \mathbf{C} \mathbf{R} = \mathbf{C}^{*\text{T}} && \text{reduzierte Dämpfungsmatrix} \\ \mathbf{G}^* &= \mathbf{R}^T \mathbf{G} \mathbf{R} = -\mathbf{G}^{*\text{T}} && \text{reduzierte Gyroskopiematrix} \\ \mathbf{K}^* &= \mathbf{R}^T \mathbf{K} \mathbf{R} = \mathbf{K}^{*\text{T}} && \text{reduzierte Steifigkeitsmatrix} \\ \mathbf{f}^* &= \mathbf{R}^T \mathbf{f} && \text{reduzierter externer Erregervektor} \end{aligned}$$

Die reduzierte Bewegungsgleichung mit nur mehr m Freiheitsgraden lautet nun :

$$\mathbf{M}^* \ddot{\mathbf{y}} + (\mathbf{C}^* + \mathbf{G}^*) \dot{\mathbf{y}} + \mathbf{K}^* \mathbf{y} = \mathbf{f}^*(t) \quad (2.7)$$

Es gilt jetzt, die Matrix \mathbf{R} entweder durch ein statisches oder modales Kondensationsverfahren, welche in den nachfolgenden Abschnitten beschrieben werden, zu bestimmen.

Statische Kondensation

(Kondensation von Freiheitsgraden nach der Methode von R.J. Guyan).

Im allgemeinen besteht ein Modell eines Rotor-Lager-System aus Knoten mit konzentrierten Massen (Scheiben) oder Lagerstellen, deren Freiheitsgrade von wesentlichem Interesse sind (=Vorzugskoordinaten), und Knoten mit weniger wichtigen Freiheitsgraden (=Nebenkoordinaten). Durch Einführen bestimmter Zwangsbedingungen können Nebenkoordinaten durch Vorzugskoordinaten ausgedrückt und so die Zahl der Freiheitsgrade reduziert werden.

Es wird nun der globale Lagevektor \mathbf{x} mit seinen n_v Vorzugskoordinaten und seinen $n - n_v$ Nebenkoordinaten mit Hilfe einer Matrix \mathbf{A} derart umgeordnet, daß der entstehende neue Vektor \mathbf{y} in einen Bereich der Vorzugs- und in den der Nebenkoordinaten zerfällt.

$$\mathbf{y} = \mathbf{A} \mathbf{x} = (\mathbf{y}_v^T, \mathbf{y}_n^T)^T \quad (2.8)$$

Für die Umsortierungsmatrix, die nur aus Nullen und Einsen besteht und eine eigentlich orthogonale Transformation darstellt, gilt:

$$\mathbf{A}^{-1} = \mathbf{A}^T \quad \text{und} \quad \det \mathbf{A} = 1$$

Da es sich um eine statische Kondensation handelt, gewinnt man die Zwangsbedingung, welche auch für den dynamischen Fall verwendet wird, aus der Bewegungsgleichung 2.1 für die statische Durchbiegung. Diese reduziert sich unter Berücksichtigung der Umsortierung zu :

$$\mathbf{A} \mathbf{K} \mathbf{A}^T \mathbf{y} = \mathbf{A} \mathbf{f}$$

$$\begin{pmatrix} \mathbf{K}_{vv} & \mathbf{K}_{vn} \\ \mathbf{K}_{nv} & \mathbf{K}_{nn} \end{pmatrix} \mathbf{y} = \begin{pmatrix} \mathbf{f}_v \\ \mathbf{f}_n \end{pmatrix}$$

Der gesuchte Zusammenhang zwischen \mathbf{y}_v und \mathbf{y}_n läßt sich mit der Annahme, daß $\mathbf{f}_n = 0$ ist, unter der Bedingung $\det \mathbf{K}_{nn} \neq 0$, nach den Nebenkoordinaten aufgelöst, anschreiben:

$$\mathbf{y}_n = -\mathbf{K}_{nn}^{-1} \mathbf{K}_{nv} \mathbf{y}_v$$

Mit $\mathbf{y}_v = \mathbf{E}_{vv} \mathbf{y}_v$ und der Gleichung 2.8 erhält die Beziehung zwischen den globalen und den Vorzugskordinaten die Form:

$$\mathbf{x} = \mathbf{R} \mathbf{y}_v$$

mit der gesuchten statischen Reduktionsmatrix :

$$\mathbf{R} = \mathbf{A}^T \begin{pmatrix} \mathbf{E}_{vv} \\ -\mathbf{K}_{nn}^{-1} \mathbf{K}_{nv} \end{pmatrix}$$

Durch dieses Kondensationsverfahren kann zwar der numerische Aufwand beträchtlich verringert werden. Allerdings zeigt eine Eigenwertanalyse, daß noch immer hochfrequente Anteile im Modell enthalten sein können. Diese Tatsache bedingt die beschränkte Anwendbarkeit dieses Verfahrens für numerische Integrationsverfahren. Es können dann nur mehr implizite Integrationsverfahren verwendet werden, um den numerischen Aufwand in Grenzen zu halten. Deshalb ist ein Kondensationsverfahren vorzuziehen, das die höheren Eigenfrequenzen in gezielter Weise unterdrückt.

Modale Kondensation

(Mode-Superposition)

Die modale Kondensation unterscheidet sich grundsätzlich von der statischen Kondensation, bei der uninteressante Freiheitsgrade gestrichen werden. Bei einem modalen Kondensationsverfahren wird das Differentialgleichungssystem auf modale Koordinaten transformiert, wobei jeder Eigenschwingungsform eine sogenannte modale Koordinate zugeordnet wird. Bei der Kondensation des Systems geht man von der Annahme aus, daß die zu den höheren Eigenfrequenzen gehörenden Schwingungsformen vernachlässigt werden können und es für die Beschreibung des Systems vollkommen ausreichend ist, nur die ersten, zu den niederen Frequenzen gehörenden Eigenformen, zu berücksichtigen. Dies wird durch die Tatsache, daß die höchste Eigenfrequenz durch das steifste FE bestimmt wird, deren dazugehörige Eigenform aber keinen wesentlichen Einfluß auf das globale Verhalten haben kann, bekräftigt.

Als Basis für die Transformation werden die ungedämpften Eigenvektoren ohne Berücksichtigung der gyrokopischen Effekte verwendet. Diese ergeben sich aus der Differentialgleichungssystem

$$\mathbf{M} \ddot{\mathbf{x}} + \mathbf{K} \mathbf{x} = 0 \quad (2.9)$$

Ein harmonischen Ansatz führt auf das lineare Eigenwertproblem:

$$(\mathbf{M} \omega^2 - \mathbf{K}) \boldsymbol{\varphi} = 0$$

mit n Eigenwerten ω^2 und n dazugehörigen Eigenvektoren $\boldsymbol{\varphi}$, die zueinander verallgemeinert orthogonal sind und derart normiert werden, daß gilt

$$\begin{aligned} \boldsymbol{\Phi}^T \mathbf{M} \boldsymbol{\Phi} &= \mathbf{E} \\ \boldsymbol{\Phi}^T \mathbf{K} \boldsymbol{\Phi} &= \boldsymbol{\Omega}^2 \end{aligned}$$

Hier ist $\boldsymbol{\Omega}^2$ die Diagonalmatrix, in der die Eigenwerte nach steigenden Eigenfrequenzen ω_j geordnet sind:

$$\boldsymbol{\Omega}^2 = \begin{pmatrix} \omega_1^2 & & & \mathbf{0} \\ & \omega_2^2 & & \\ & & \ddots & \\ \mathbf{0} & & & \omega_n^2 \end{pmatrix}$$

und

$$\boldsymbol{\Phi} = (\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_n)$$

ist die entsprechend aus den Eigenvektoren aufgebaute Modalmatrix. Die Transformation auf modale Koordinaten \mathbf{y} lautet dann:

$$\mathbf{x} = \boldsymbol{\Phi} \mathbf{y}$$

Die Differentialgleichung des gesamten Systems 2.1 wird transformiert und erhält die neue Gestalt:

$$\ddot{\mathbf{y}} + \Phi^T (\mathbf{C} + \mathbf{G}) \Phi \dot{\mathbf{y}} + \Omega^2 \mathbf{y} = \Phi^T \mathbf{f} \quad (2.10)$$

Diese kann ebenso wie das ursprüngliche System einer Integration zugeführt werden. Allerdings muß zuerst das Eigenwertproblem gelöst werden, um eine leichtere Integration zu ermöglichen, da die modale Massenmatrix nunmehr eine Einheitsmatrix ist und die modale Steifigkeitsmatrix eine Diagonalmatrix. Die Dimension des Differentialgleichungssystems ist aber noch unverändert.

Wie am Beginn dieses Abschnittes erwähnt, werden bei einem modalen Kondensationsverfahren nur mehr die ersten p Eigenvektoren für die Integration herangezogen, deren Eigenfrequenzen unter einer gewissen oberen Schranke liegen. Das heißt, daß die Näherungslösung als Linearkombination der ersten p Eigenvektoren, gewichtet mit den Komponenten des p -dimensionalen Lösungsvektors \mathbf{y}^* , berechnet werden kann (Mode-Superposition), oder formal ausgedrückt :

$$\mathbf{x} = \mathbf{R} \mathbf{y}^*$$

mit der gesuchten Reduktionsmatrix

$$\mathbf{R} = (\varphi_1, \dots, \varphi_p)$$

2.3 Dynamische Analyse

Im Hinblick auf die Simulation des Rotors auf Magnetlagern ist es von Nutzen, zuvor eine dynamische Analyse durchzuführen. Es werden die Eigenwerte, die auch für das Integrationsverfahren wesentlich sind, und die dazugehörigen Eigenformen bestimmt. Um zu beurteilen, ob gewisse Eigenformen überhaupt gefährlich werden können, wird auch die stationäre Unwuchtlösung berechnet. Am Ende dieses Abschnittes werden die theoretischen Überlegungen für spezielle Rotoren numerisch ausgewertet.

Biegeeigenfrequenzen

Betrachten wir nur den homogenen Anteil der Gleichung 2.1 oder, um die Güte der Kondensation zu beurteilen, deren kondensierte Version.

$$\mathbf{M} \ddot{\mathbf{x}} + (\mathbf{C} + \mathbf{G}(\Omega)) \dot{\mathbf{x}} + \mathbf{K} \mathbf{x} = 0$$

Mit dem Ansatz

$$\mathbf{x} = \mathbf{X}e^{i\omega t}$$

lautet das Eigenwertproblem

$$\left(-\mathbf{M}\omega^2 + i\omega(\mathbf{C} + \mathbf{G}(\Omega)) + \mathbf{K}\right) \mathbf{X} = 0$$

mit

$$\det\left(-\mathbf{M}\omega^2 + i\omega(\mathbf{C} + \mathbf{G}(\Omega)) + \mathbf{K}\right) = 0.$$

Die Eigenwerte und die dazugehörigen Eigenvektoren können numerisch bestimmt werden. Um diese numerisch berechnen zu können, muß das System auf Zustandsform gebracht werden

$$\det(i\omega\mathbf{E} - \mathbf{A}) = 0$$

mit einer passenden Einheitsmatrix \mathbf{E} und der Zustandsmatrix

$$\mathbf{A} = \left(\begin{array}{c|c} \mathbf{0} & \mathbf{E} \\ \hline -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}(\mathbf{G} + \mathbf{C}) \end{array} \right).$$

Das Eigenwertproblem kann in dieser Form vom Programm MATLAB direkt mit einer eigenen Funktion gelöst werden.

Stationäre Unwuchtlösung

Für die stationäre Unwuchtlösung ist nur der Vektor der Unwuchten in den Knoten als inhomogener Anteil der Differentialgleichung von Bedeutung. Der lokale Erregervektor für einen Knoten wird durch die Unwuchtmassen $m^{(k)}$, dessen Schwerpunktsexentrität $e^{(k)}$, und dessen Phasenlage $\varphi^{(k)}$ bestimmt in der Form

$$\mathbf{f}^{(k)}(t) = \begin{pmatrix} m^{(k)}e^{(k)}\Omega^2 \cos(\Omega t - \varphi^{(k)}) \\ m^{(k)}e^{(k)}\Omega^2 \sin(\Omega t - \varphi^{(k)}) \\ 0 \\ 0 \end{pmatrix}$$

Hier wurden die dynamischen Unwuchten vernachlässigt. Der Unwuchtvektor wird gemäß Koinzidenztransformation zusammengestellt. Die stationäre Unwuchtlösung wird als harmonischer Ansatz

$$\mathbf{X} = \mathbf{X}_1 \sin \Omega t + \mathbf{X}_2 \cos \Omega t \tag{2.11}$$

formuliert, und der Unwuchtvektor in entsprechender Form

$$\mathbf{f} = \mathbf{f}_1 \sin \Omega t + \mathbf{f}_2 \cos \Omega t.$$

Damit erhält man aus 2.1 die Lösung

$$\begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{K} - \mathbf{M}\Omega^2 & | & -(\mathbf{C} + \mathbf{D})\Omega \\ \hline (\mathbf{C} + \mathbf{D})\Omega & | & \mathbf{K} - \mathbf{M}\Omega^2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{pmatrix}.$$

Gemäß Ansatz 2.11 erhält man daraus den Zeitverlauf der Unwuchtschwingungen.

Numerische Auswertung

Für die numerische Berechnung wurden Programme geschrieben, die aus den in einem File gespeicherten Rotordaten die Systemmatrizen generieren, weiters das System sowohl statisch als auch modal kondensieren und für beide Systeme die Eigenwerte und Eigenvektoren berechnen. Letzere können dann, ebenso wie die statische Durchbiegung und die stationär Unwuchtlösung, in einem Plot beurteilt werden. Für die modale Kondensation wurde die Grenzfrequenz bei etwa 150 Hz gewählt, da damit auch für ein explizites Integrationsverfahren die numerische Stabilität gesichert ist. Zur Untersuchung eines Rotors sind folgende Programme im Anhang enthalten:

- sysdatN** m-File mit der Nummer N , in dem die Rotordaten enthalten sind.
- rotttest** Analyse-Programm, die folgende Unterprogramme enthalten:
 - Statische Durchbiegung mit Berechnung der Randfaserspannungen,
 - Berechnung der Eigenwerte und -vektoren,
 - Testprogramm für die Kondensation.
- pw_ewo** Berechnung der Eigenwerte in Abhängigkeit von der Drehzahl.
- pw_ew** Berechnung der Eigenwerte bei Variation der Lagerparameter.
Diese können mit **pw_ewpl** graphisch angezeigt werden.

Zwei verschiedene Rotoren, die in jeweils zwei Magnetlagern gelagert sind, werden Gegenstand der Untersuchung sein. Für die Berechnung der stationären Unwuchtschwingung wurde für den jeweiligen Rotor ein Unwuchtzustand festgesetzt. Die Schwerpunktsexzentrizität wurde mit $10 \mu m$ angenommen. Für eine Scheibe mit $4 kg$ erzeugt diese bei der Nenndrehfrequenz 100 rad/s eine umlaufende Erregerkraft von 4 N . Die Phasenlage der Exzentrizitäten zueinander wurde willkürlich angenommen, aber so daß nicht alle Exzentrizitäten in einer Ebene liegen. Alle Eigenformen wurden für eine Nenndrehfrequenz von 100 rad/s berechnet. Es werden vier Verformungszustände des Rotors pro Umdrehung dargestellt. Die angegebene Dämpfung ergibt sich aus dem Realteil des komplexen Eigenwertes bezogen auf die ungedämpfte Eigenfrequenz.

Rotor 1

Rotordrehzahl:	100	rad/s
Rotormasse:	23.3	kg
Rotorlänge:	1.214	m
Wellendurchmesser:	25	mm
Lagerabstand:	1.034	m
Lagersteifigkeit	5E6	N/m
Dämpfungsfaktor:	0.5	Ns/m
Anzahl finiten Elemente:	22	-
Rotorscheiben:	3	-

Unwuchtzustand der Rotorscheiben:

Scheibe	Exzentrizität	Phasenlage
1	10 μm	0 rad
2	10 μm	1 rad
3	10 μm	2 rad

Der Rotor wird in Bild 2.4 schematisch dargestellt. Er ist außenseitig gelagert. Die Kupplung befindet sich auf der rechten Seite.

Rotor 2

Rotordrehzahl:	100	rad/s
Rotormasse:	22.7	kg
Rotorlänge:	1.228	m
Wellendurchmesser:	25	mm
Lagerabstand:	0.794	m
Lagersteifigkeit	5E6	N/m
Dämpfungsfaktor:	0.5	Ns/m
Anzahl finiten Elemente:	22	-
Rotorscheiben:	3	-

Unwuchtzustand der Rotorscheiben:

Scheibe	Exzentrizität	Phasenlage
1	10 μm	0 rad
2	10 μm	1 rad
3	10 μm	2 rad

Der Rotor wird in Bild 2.13 schematisch dargestellt. Die überhängende Scheibe befindet sich auf der linken Seite, die Kupplung auf der rechten Seite.

æ

Abbildung 2.5: Statische Durchbiegung des Rotors zufolge Eigengewicht

Abbildung 2.6: Differenz der statischen Durchbiegungen zwischen kondensiertem und unkondensiertem Rotormodell

Abbildung 2.7: Gedämpfte Eigenfrequenzen ω_e über Drehfrequenz ω_0 . x markiert die kritischen Drehzahlen.

Abbildung 2.8: Auslenkungen der stationären Unwuchtschwingung an den mit Scheiben 1,2 und 3 besetzten Knoten (Unwuchtverteilung siehe S. 21)

Abbildung 2.9: 1. Eigenform bei Nenndrehfrequenz, (U-Mode)

Abbildung 2.10: 2. Eigenform bei Nenndrehfrequenz, (S-Mode)

Abbildung 2.11: 3. Eigenform bei Nenndrehfrequenz, (W-Mode)

Abbildung 2.12: 4. Eigenform bei Nenndrehfrequenz

Abbildung 2.14: Statische Durchbiegung des Rotors zufolge Eigengewicht

Abbildung 2.15: Differenz der statischen Durchbiegungen zwischen kondensiertem und unkondensiertem Rotormodell

Abbildung 2.16: Gedämpfte Eigenfrequenzen ω_e über Drehfrequenz ω_0 . x markiert die kritischen Drehzahlen.

Abbildung 2.17: Auslenkungen der stationären Unwuchtschwingung an den mit Scheiben 1,2 und 3 besetzten Knoten. (Unwuchtverteilung siehe S. 26)

Abbildung 2.18: 1. Eigenform bei Nenndrehfrequenz

Kapitel 3

Modell des Magnetlagers

3.1 Hysteresemodell des magnetischen Werkstoffes

Magnetische Materialien zeigen ein nichtlineares Verhalten. Dieses ist einerseits durch die Existenz einer magnetischen Sättigung bestimmt, andererseits durch den Hystereseeffekt. Dies wird deutlich im Magnetisierungsschaubild Abb. 3.1.

Wird ein entmagnetisiertes ferromagnetisches Material einer magnetischen Feldstärke ausgesetzt, so erzeugt dieses eine magnetische Induktion, die entlang der Neukurve mit der Anfangspermeabilität μ_N ansteigt. Dies ist der Bereich der reversiblen Wandverschiebungen zwischen den Weiß'schen Bezirken (Rayleigh-Gebiet). Im nächsten, annähernd linearen Bereich der irreversiblen Wandverschiebungen, wird im ferromagnetischen Material durch geringe Feldstärkeänderung eine große Induktionsänderung hervorgerufen (Barkhausen-Sprünge). Wenn diese Wandverschiebungen abgeschlossen sind, beginnt der reversible Prozeß der Feldvektordrehungen innerhalb der Bezirke. Das ferromagnetische Material befindet sich in der Sättigung, die etwa bei 1.2 Tesla (Silikoneisen), oder bei neueren Materialien bei 2 Tesla liegt. Eine weitere Magnetisierung verläuft nur mehr mit einer sehr geringen Steigerung, nämlich mit μ_S . Dreht man die Richtung der Feldstärke um, so nimmt die magnetische Induktion in einem geringeren Maße ab, da für die irreversiblen Wandverschiebungen Energieaufwand nötig ist. Wird die Feldstärke gleich Null, verbleibt deshalb die Remanenzinduktion B_R . Um die magnetische Induktion wiederum zu beseitigen, muß in entgegengesetzter Richtung die Koerzitivfeldstärke H_C aufgewendet werden. Der entmagnetisierte Zustand wird in dieser Weise nicht mehr erreicht. Das bedeutet, daß die magnetische Feldstärke und die Induktion in einer nichtlinearen Weise verknüpft sind, die von der Vorgeschichte der Magnetisierung abhängt, in der Form

$$B = \Psi(H, t).$$

Wird weiters, wie das Bild 3.1 verdeutlicht, einem gewissen Magnetisierungszustand ein kleines Wechselfeld ($\Delta H, \Delta B$) überlagert, so ergeben sich nach einer geringen Drift kleine

Hystereseschleifen, sogenannte Lanzetten. Das obige Bild zeigt, daß die differentielle Permeabilität in diesen wesentlich kleiner sein kann als die Permeabilität am Beginn der Magnetisierung.

Diese Effekte sind für das dynamische Verhalten eines Magnetlagers von Bedeutung und vor allem dann, wenn das Magnetlager an die Grenzen seiner Tragfähigkeit stößt. Deshalb muß ein Simulationsmodell diese Effekte berücksichtigen, um Aussagen über das Verhalten speziell in jenen Bereichen machen zu können.

Zur Simulation des Verhaltens eines magnetischen Werkstoffes wird das Hysteresemodell von Hodgdon/Coleman (siehe [1]) verwendet. Dieses Modell basiert auf der Differentialgleichung erster Ordnung

$$\frac{dH}{dB} = \frac{\dot{H}(t)}{\dot{B}(t)} = \alpha \operatorname{sgn} \dot{B} [f(B) - H] + g(B) \quad (3.1)$$

wobei die magnetische Feldstärke H und die magnetische Induktion B stückweise einmal stetig differenzierbare Funktionen der Zeit sind. Durch geeignete Wahl des Parameters α und der Materialfunktionen f und g ist eine Simulation magnetisch harter als auch magnetisch weicher Materialien möglich.

Der Parameter α beeinflusst die Größe der Hystereseverluste. Setzt man in Gleichung 3.1 diesen Null, treten keine Hystereseeffekte auf und es ist $dH/dB = g(B)$ eine eindeutige Funktion von B .

Die Materialfunktionen müssen, qualitativ formuliert, gewissen Bedingungen genügen. Diese beinhalten die Forderung nach Stetigkeit des Hysterese Modells. Weiters muß das asymptotische Verhalten der Sättigung sichergestellt sein sowie die Tatsache, daß der mit dem Durchlaufen einer Hystereseschleife verbundene Energieverlust immer positiv ist. (siehe [1]) Für die Materialfunktionen werden gemäß [1] eine Kombination aus Tangens- und Exponentialfunktionen verwendet :

$$f = \begin{cases} A_1 \tan A_2 B, & \text{für } |B| \leq B^\# \\ A_1 \tan A_2 B^\# + (B - B^\#)/\mu_{cl}, & \text{für } B > B^\# \\ -A_1 \tan A_2 B^\# + (B + B^\#)/\mu_{cl}, & \text{für } B < -B^\# \end{cases}$$

$$g = \begin{cases} f'(B) [1 - A_3 \exp\left(\frac{-A_4 |B|}{B_{cl} - |B|}\right)], & \text{für } |B| \leq B_{cl} \\ f'(B), & \text{für } |B| > B_{cl} \end{cases}$$

Liegt nun die voll ausgesteuerte Hystereseschleife eines magnetischen Werkstoffes in Form einer gemessenen Kurve vor, werden die Parameter B_{cl} , H_{cl} , μ_{cl} , μ_s , μ_r , μ_c , B_r , und H_c entnommen, vgl. Abb. 3.2 .

B_{cl} ist die magnetische Induktion, H_{cl} die magnetische Feldstärke und μ_s die differentielle Permeabilität in jenem Punkt, in dem die Grenztrajektorien "scheinbar" schließen. μ_{cl} entspricht der Permeabilität für Werte der magnetischen Induktion mit $|B| > B^\#$.

Mit den empirisch gefundenen Parametern werden die Koeffizienten A_1 bis A_4 so bestimmt, daß die geometrischen Bedingungen in der gemessenen Kurve erfüllt werden [1]. Somit kann nach Einsetzen der nunmehr bestimmten Koeffizienten gemäß Gleichung 3.1 jedem Punkt (H, B) eine differentielle Permeabilität B' zugeordnet und die Gleichung selbst integriert werden. Um das hysteretische Verhalten eines magnetischen Materials testen zu können, ist es notwendig ein eigenes kleines Testprogramm zu schreiben und ein Hysterese-Modell zu integrieren. Dies geschieht mit dem Programm `hystest`. Für dieses und für die späteren Simulationen werden folgende Materialdaten verwendet (Silikoneisen ARNON):

A_1	=	22.7971	A/m
A_2	=	1.18639	1/T
A_3	=	-11.817	-
A_4	=	0.44265	-
μ_0	=	$4\pi E-7$	Vs/Am
μ_{cl}	=	$3E-5$	Vs/Am
α	=	10	1/T
$B^\#$	=	1.3	T
B_{cl}	=	1.3	T

3.2 Differentialgleichungen des magnetischen Kreises mit Luftspalt

Bevor die Kraftbeziehungen für ein Magnetlager aufgestellt werden, soll ein allgemeiner magnetischer Kreis mit Luftspalten besprochen werden. Die Ergebnisse können dann in einfacher Weise auf ein Modell des Magnetlagers übertragen werden.

Wie im Bild 3.3 zu sehen, besteht ein allgemeiner magnetischer Kreis aus n Luftspalten der Länge L_{gj} und n dazwischenliegenden magnetischen Kernen der Länge L_{cj} mit den jeweiligen Querschnittsflächen A_{gj} und A_{cj} entlang des magnetischen Pfades. Der hier auftretende magnetische Fluß wird mittels einer Spule mit N Windungen mit einer magnetomotorischen Kraft F erzeugt. Diese ist gegeben durch:

$$F = NI = \oint_L H(l) dl \quad (3.2)$$

mit H als der magnetischen Feldstärke entlang des magnetischen Pfades, Mit der obigen Diskretisierung folgt annähernd

$$NI \doteq \sum_{j=1}^n (H_{cj} L_{cj} + H_{gj} L_{gj}) . \quad (3.3)$$

Weiters wird angenommen, daß die magnetische Induktion im Querschnitt selbst annähernd konstant verteilt ist, mit

$$B_{cj} = \frac{\Phi}{A_{cj}}$$

$$B_{gj} = \frac{\Phi}{A_{gj}} .$$

Für die Beziehung zwischen der magnetischen Feldstärke und der magnetischen Induktion gilt im Luftspalt

$$B_{gj} = \mu_0 H_{gj}$$

und für das ferromagnetische Kernmaterial die im vorhergehenden Kapitel beschriebene Beziehung

$$B'_{cj}(t) = \frac{\dot{B}_{cj}(t)}{\dot{H}_{cj}(t)} \quad (3.4)$$

Ist eine elektromotorische Kraft $U(t)$, wie in Abbildung 3.3 gezeigt, die treibende Kraft für den magnetischen Fluß ist, so sind die beschreibenden Differentialgleichungen gegeben durch [2]

$$\frac{d\Phi}{dt} = \frac{u(t) - rI}{N} \quad (3.5)$$

$$\frac{dH_{cj}}{dt} = \frac{1}{A_{cj}B'_{cj}} \frac{d\Phi}{dt} \quad (3.6)$$

$$I = \frac{1}{N} \sum_{j=1}^n \left(H_{cj} L_{cj} + \Phi \frac{L_{gj}}{\mu_0 A_{gj}} \right) \quad (3.7)$$

Für ein System mit n Kernen und n Luftspalten ergeben sich $n + 1$ gewöhnliche, nichtlineare Differentialgleichungen für den Fluß Φ und den n Feldstärken mit n differentiellen Permeabilitäten B'_{cj} , die gemäß dem vorhergehenden Abschnitt bestimmt werden. Diese Gleichungen sind Bestandteil des für das System Rotor-Magnetlager zu simulierenden Differentialgleichungssystems. Die Implementierung wird im Kapitel 5 genauer beschrieben.

3.3 Magnetlagergeometrie und Lagersteifigkeiten

Die Geometrie des Magnetlagers ist für die Bestimmung der Aktuatorkräfte auf den Rotor von Bedeutung. Ein Magnetlager besteht nebst tragendem Gehäuse zumeist aus p Polschuhpaaren die über den Umfang verteilt sind und denen ein Positionswinkel β_k zur z-Achse zugeordnet ist. (Abb. 3.4)

Üblicherweise sind alle Polschuhpaare gleich gestaltet und symmetrisch. Auf jedem sitzt eine Spule mit N_k Windungen, die mit einer Spannung U_k bzw. einem Strom I_k betrieben wird. Der entstehende magnetische Fluß wird in einem magnetischen Pfad konzentriert gedacht. Dieser Pfad besteht aus der Pfadlänge L_R im Rotor und der im Statorteil L_S , mit den Querschnittsflächen A_R bzw. A_S sowie aus den mit der Zeit veränderlichen Luftspatllängen L_{gk1} und L_{gk2} mit der Querschnittsfläche A_{gk} . Jeder Polschuh hat in der Nullposition die nominelle Luftspatllänge g_0 . Bei einer im Verhältnis zu den Rotorabmessungen kleinen Auslenkung (y, z) kann eine linearisierte Abhängigkeit der Luftspatllänge von den Rotorverschiebungen angegeben werden in der Form

$$\begin{pmatrix} L_{gk1} \\ L_{gk2} \end{pmatrix} \doteq \begin{pmatrix} g_k \\ g_k \end{pmatrix} - \begin{pmatrix} \cos(\beta_k - \alpha_k) & \sin(\beta_k - \alpha_k) \\ \cos(\beta_k + \alpha_k) & \sin(\beta_k + \alpha_k) \end{pmatrix} \cdot \begin{pmatrix} y \\ z \end{pmatrix} \quad (3.8)$$

Von jedem Polschuh wirkt nun die anziehende Magnetkraft F_k normal auf die Rotoroberfläche im Winkel α_k zur Polschuhpaarsymmetrale mit :

$$F_{k1} = F_{k2} = F_k = \frac{\Phi_k^2}{2\mu_0 A_{gk}} \quad (3.9)$$

Sind jetzt p Polschuhpaare in einem Kreis um den Rotor angeordnet, so ergibt sich die resultierende Aktuatorkraft als Summe der einzelnen Magnetkräfte :

$$F_z = \sum_{k=1}^p \frac{\Phi_k^2}{A_{gk}\mu_0} \cos \alpha_k \cos \beta_k \quad (3.10)$$

$$F_y = \sum_{k=1}^p \frac{\Phi_k^2}{A_{gk}\mu_0} \cos \alpha_k \sin \beta_k . \quad (3.11)$$

Diese Kräfte sind Funktionen der magnetischen Flüsse Φ_k , die wiederum von der Geometrie, sprich Größe des Luftspaltes, somit der Rotorposition, und der Bestromung in den Spulen abhängig sind. Um einen linearen Regler auslegen, oder darauf aufbauend eine Eigenwertanalyse des Rotors durchführen zu können, ist es notwendig, in einem Betriebspunkt magnetische Steifigkeiten des Lagers anzugeben. Wegen der Positions- und Stromabhängigkeit der Aktuatorkräfte ergeben sich jeweils ein Lage bzw. ein Stromkoeffizient.

Lagekoeffizient des Magnetlagers

Die Steifigkeitsmatrix des Magnetlagers ist formal gegeben durch :

$$\mathbf{K}_L = - \begin{pmatrix} \frac{\partial F_z}{\partial z} & \frac{\partial F_z}{\partial y} \\ \frac{\partial F_y}{\partial z} & \frac{\partial F_y}{\partial y} \end{pmatrix} \quad (3.12)$$

An ferromagnetischen Trennflächen können nur Zugkräfte erzeugt werden und nicht wie bei einer mechanischen Feder auch Druckkräfte. Diese Tatsache bewirkt eine Destabilisierung und kann nur durch einen Regler kompensiert werden. In Hinblick auf die Reglerauslegung ist es günstiger, die Steifigkeit als positive Zahl einzuführen, um aber der Natur der auslenkenden Kräfte gerecht zu werden, ist ein Minuszeichen davorzusetzen.

Die Kraftbeziehungen 3.10 partiell abgeleitet und in 3.12 eingesetzt, ergeben:

$$\frac{\partial(F_z, F_y^T)}{\partial(z, y)} = \sum_{k=1}^p \frac{2\Phi_k \cos \alpha_k}{A_{gk} \mu_0} \begin{pmatrix} \frac{\partial \Phi_k}{\partial z} \cos \beta_k & \frac{\partial \Phi_k}{\partial y} \cos \beta_k \\ \frac{\partial \Phi_k}{\partial z} \sin \beta_k & \frac{\partial \Phi_k}{\partial y} \sin \beta_k \end{pmatrix} \quad (3.13)$$

Die partiellen Ableitungen des magnetischen Flusses Φ_k nach den Ortskoordinaten sind gegeben durch (siehe [2])

$$\frac{\partial \Phi_k}{\partial(z, y)} = \frac{-\Phi_k}{A_{gk} \mu_0 R_{loc,k}} \frac{\partial(L_{gk1} + L_{gk2})}{\partial(z, y)} \quad (3.14)$$

wobei sich die Summe der Luftspalte mit der geometrischen Beziehung 3.8 ergibt :

$$L_{gk1} + L_{gk2} = 2(g_k - z \cos \alpha_k \cos \beta_k - y \cos \alpha_k \sin \beta_k)$$

und der lokale magnetische Widerstand $R_{loc,k}$ definiert ist durch

$$R_{loc,k} = \frac{L_R}{A_R B'_{Rk}} + \frac{L_S}{A_S B'_{Sk}} + \frac{L_{gk1} + L_{gk2}}{\mu_0 A_{gk}}. \quad (3.15)$$

B'_{Sk} und B'_{Rk} bezeichnen dabei die differentiellen Permeabilitäten im Rotor bzw. im Stator, die vom jeweiligen Magnetisierungszustand abhängig sind.

Alle diese Beziehungen in Gleichung 3.13 eingesetzt ergeben die gesuchte Aktuatorsteifigkeitsmatrix für die Lage

$$\mathbf{K}_L = - \sum_{k=1}^p \frac{4\Phi_k^2 \cos^2 \alpha_k}{(A_{gk} \mu_0)^2 R_{loc,k}} \begin{pmatrix} \cos^2 \beta_k & \sin \beta_k \cos \beta_k \\ \sin \beta_k \cos \beta_k & \sin^2 \beta_k \end{pmatrix}. \quad (3.16)$$

Stromkoeffizient des Magnetlagers

Wird ein einzelner Aktuator in einem Magnetlager mit einem Steuerstrom I_k angesteuert, so sind die Stromkoeffizienten der Luftspaltkräfte des Aktuators gegeben (vgl. [2]) durch

$$K_{SI_k,z} \equiv \frac{\partial F_z}{\partial I_k} = \frac{2\Phi_k N_k}{A_{gk}\mu_0 R_{loc,k}} \cos \alpha_k \cos \beta_k \quad (3.17)$$

$$K_{SI_k,y} \equiv \frac{\partial F_y}{\partial I_k} = \frac{2\Phi_k N_k}{A_{gk}\mu_0 R_{loc,k}} \cos \alpha_k \sin \beta_k \quad (3.18)$$

Da der Rotor im Magnetlager die zwei Freiheitsgrade z und y haben soll, kann man davon ausgehen, daß zwei Regler zwei Ströme erzeugen, die in einer gewissen Weise zusammengesetzt, den jeweiligen Aktuatorstrom ergeben, oder formal ausgedrückt

$$I_k = \psi_k(\bar{i}_z, \bar{i}_y).$$

Dies geschieht im Hinblick auf die Regelung des Magnetlagers mit zwei Reglern, je einen für einen Freiheitsgrad. Bei moderneren Magnetlagern wird aber jeder einzelne Aktuator geregelt. Die Koeffizientenmatrix für die Stromabhängigkeit ist mit obiger Annahme gegeben durch

$$\mathbf{K}_S = \begin{pmatrix} \frac{\partial F_z}{\partial i_z} & \frac{\partial F_z}{\partial i_y} \\ \frac{\partial F_y}{\partial i_z} & \frac{\partial F_y}{\partial i_y} \end{pmatrix} \quad (3.19)$$

und ergibt sich aus Gleichung 3.17 und der inneren Ableitungen für I_k über alle einzelnen Aktuatoren aufsummiert zu

$$\mathbf{K}_S = \sum_{k=1}^p \frac{2\Phi_k N_k \cos \alpha_k}{A_{gk}\mu_0 R_{loc,k}} \begin{pmatrix} \frac{\partial I_k}{\partial i_z} \cos \beta_k & \frac{\partial I_k}{\partial i_y} \cos \beta_k \\ \frac{\partial I_k}{\partial i_z} \sin \beta_k & \frac{\partial I_k}{\partial i_y} \sin \beta_k \end{pmatrix}. \quad (3.20)$$

æ

Kapitel 4

Linearisiertes Modell als Grundlage für die Auslegung eines Reglers

4.1 Linearisiertes Modell für einen Betriebspunkt

In den beiden vorgegangenen Kapiteln ging es darum, ein Modell des Rotors zu bilden, und ein für die Simulation verwendbares nichtlineares Modell des aktiven Magnetlagers aufzustellen. Eingangsgrößen des Rotors sind die Erregerkräfte und im Fall eines Magnetlagers die Lagerkräfte, Ausgangsgröße der Zustandsvektor. Eingangsgrößen des Magnetlagers sind die Steuerspannungen des Reglers über den Verstärker, Ausgangsgrößen sind die Lagerkräfte. Da das aktive Magnetlager von Natur aus instabil ist, bedarf es eines Reglers, um sein Verhalten um einen statischen Gleichgewichtszustand zu stabilisieren. Der Regler koppelt die Steuerspannungen und den Zustandsvektor, wodurch sich ein geschlossener Wirkungsablauf des Systems ergibt (vgl. Bild 1.2). Für ein Lager ist dieser im Bild 4.1 zu sehen.

Der Digitalregler, dem über einen A/D-Wandler diskrete Werte des Lagevektors zugeführt werden, errechnet die Stellgrößen, die über ein Verteilernetzwerk ψ , dem Sample-Hold-Glied H_0 und dem Leistungsverstärker die einzelnen Aktuatoren des Magnetlagers ansteuern. Das Magnetlager erzeugt in einem nichtlinearen Zusammenhang die Lagerkräfte, die gemeinsam mit äußeren Kräften auf den Rotor einwirken und die Rotorverschiebungen erzeugen.

Wie das Bild 4.1 zeigt, werden für die Regelung eines Lagers nur die Verschiebungen des Rotors in diesem Lager rückgeführt. Die beiden aktiven Magnetlager werden voneinander entkoppelt betrachtet. Dies entspricht nicht der Wirklichkeit, erleichtert aber die Reglerauslegung, da sonst ein multiple input-multiple output-System verwendet werden müßte. Der Rotor wird für die Reglerauslegung als Punktmasse im Lager aufgefaßt. Als Masse wird die halbe Rotormasse verwendet, da der Rotor beinahe symmetrisch ist. Der Einfluß dieser Abschätzung wird mit der Reglerauslegung später behandelt.

Um einen Regler mit geringem Aufwand analytisch auszulegen, ist es notwendig ein lineares Modell des Wirkungsablaufes zu verwenden. Das Magnetlager macht hier Schwierigkeiten. Gemäß dem vorhergehenden Kapitel ist es sinnvoll, lokale Kraftkoeffizienten einzuführen und zwar einen für die Lage und einen für den Spulenstrom. Da diese vom Arbeitspunkt

(statische Last) des Magnetlagers abhängen, werden gewisse, realistische Vereinfachungen vorgenommen, die die Gleichheit aller p Aktuatoren betreffen.

$$\begin{array}{ll}
\text{Flußpfadquerschnitte:} & A_R = A_S = A_{gk} = A \\
\text{Windungszahlen:} & N_k = N \\
\text{Nennluftspatllängen:} & g_k = g_0 \\
\text{Polschuhwinkel:} & \alpha_k = \alpha_A \\
\text{Flußpfadlänge:} & L_R + L_S = L_{tot} \\
\text{Magnetischer Fluß:} & \Phi_k = \Phi_0 \\
\text{Lokaler magn. Widerstand:} & R_{loc,k} = R_{loc}
\end{array}$$

Einen wesentlichen Einfluß auf den magnetischen Kreis hat der lokale magnetische Widerstand, der stark vom Belastungszustand des Lagers und der Größe des Luftspaltes abhängt. Für einen linearisierten magnetischen Kreis bei einem konstanten Luftspalt g_0 und einer konstanten Durchflutung Φ_0 wird der magnetische Widerstand zu

$$R_{loc} = \frac{1}{\mu_0 A} \left(\frac{L_{tot}}{\mu_r} + 2g_0 \right),$$

mit der relative Permeabilität μ_r , die sich aus der differentiellen Permeabilität B' für den Magnetisierungszustand $(B(t), H(t))$ entlang der anhysteretischen Kurve ergibt zu

$$\mu_r(\Phi) = \frac{1}{\mu_0 A_1 A_2} \cos A_2 \frac{\Phi}{A}.$$

Die Konstanten Größen A_1 und A_2 sind identisch mit den Parametern des Hysterese Modells aus Kapitel 3.

Wie Bild 4.2 für den magnetischen Widerstand erkennen läßt, ist für einen weiten Bereich unterhalb der Sättigung eine Linearisierung zulässig, für höhere Belastungen, d.h. kleines μ_r , müßte ein parameteradaptiver Regler in Betracht gezogen werden. Für den Bereich der Sättigung ergibt sich im Bild ein Plateau, da hier die Permeabilität fast gleich der Permeabilität im Vakuum ist.

Mit den obigen Annahmen vereinfachen sich die Formeln 3.16 für die Steifigkeiten zu:

$$\mathbf{K}_L = - \frac{4\Phi_0^2 \cos^2 \alpha_A}{(A\mu_0)^2 R_{loc}} \mathbf{G} \quad (4.1)$$

mit der Matrix \mathbf{G} , die der Geometrie des Magnetlagers Rechnung trägt :

$$\mathbf{G} = \sum_{k=1}^p \begin{pmatrix} \cos^2 \beta_k & \sin \beta_k \cos \beta_k \\ \sin \beta_k \cos \beta_k & \sin^2 \beta_k \end{pmatrix}$$

Die Funktion, die die Ansteuerung der einzelnen Aktuatoren des Magnetlagers in Abhängigkeit der Regelströme für die Freiheitsgrade z und y , i_z bzw. i_y , bewerkstelligt wird folgendermaßen gewählt:

$$\psi(i_z, i_y) = i_z \cos \beta_k + i_y \sin \beta_k$$

Mit den partiellen Ableitungen dieser Funktion nach i_z und i_y wird Gleichung 3.20 zu :

$$\mathbf{K}_S = \frac{2\Phi_0 N \cos \alpha_A}{A\mu_0 R_{loc}} \mathbf{G} \quad (4.2)$$

mit derselben Matrix \mathbf{G} wie in Gleichung 4.1. Meist wird die Geometrie so gewählt, daß die Teilung zwischen den Aktuatoren gleich groß ist und β_k zu einer arithmetischen Folge wird. Wenn ein Aktuator in die z -Achse gelegt wird, erhält \mathbf{G} die Diagonalform

$$\mathbf{G} = \begin{pmatrix} k_g & 0 \\ 0 & k_g \end{pmatrix}$$

mit dem Geometriefaktor, der das Zusammenwirken der p Aktuatoren berücksichtigt :

$$k_g = \sum_{k=1}^p \cos^2 \frac{2(k-1)\pi}{p} = \sum_{k=1}^p \sin^2 \frac{2(k-1)\pi}{p}$$

Für die Reglerauslegung ist der Arbeitspunkt des Magnetlagers, oder vielmehr der der einzelnen Aktuatoren von Bedeutung. Der Arbeitspunkt wird durch den magnetischen Fluß Φ festgelegt und bestimmt so die Steifigkeiten des Lagers. Die konstante Durchflutung wurde so gewählt, daß sich im ferromagnetischen Material eine Induktion von 0.5 Tesla einstellt. Es soll damit eine Reserve von ± 0.5 Tesla im annähernd linearen Bereich des lokalen magnetischen Widerstandes für die dynamische Beanspruchung des ferromagnetischen Materials zu Verfügung stehen. Die elektromagnetischen Konstanten wurden gleich denen in [1] gewählt. Die geometrischen Daten wurden für ein aktives Magnetlager mittlerer Größe geschätzt. Konkret werden folgende Daten für die Simulation verwendet:

Windungszahl einer Spule:	N	=	570	-
Nomin. Luftspaltlänge:	g_0	=	0.5	mm
Polschuhwinkel:	α_A	=	23	Grad
Flußfadlänge:	L_{tot}	=	100	mm
Flußfadquerschnitt:	A	=	100	mm ²
Positionswinkel:	β_k	=	0,90,180,270	Grad
Geometriefaktor:	k_g	=	2	-

Damit können die Kraftkoeffizienten der Lage und des Stromes als Funktion des magnetischen Flusses Φ berechnet werden. Die Bilder 4.3 und 4.4 zeigen die Kraftkoeffizienten in Abhängigkeit des magnetischen Flusses.

Es ist ersichtlich, daß der Stromkoeffizient fast bis in die Sättigung linear ansteigt, und der Lagekoeffizient quadratisch. Da der magnetische Widerstand in der Sättigung stark ansteigt,

und dieser in den Koeffizienten im Nenner auftaucht, sinken die Kraftkoeffizienten beträchtlich, und steigen nachher wieder linear an.

Wählt man einen Arbeitspunkt mit Φ_0 so ergibt sich ein bestimmter magnetischer Widerstand und daraus die entsprechenden Lage- und Stromsteifigkeiten. Für die rotationssymmetrische Anordnung der Aktuatoren kann der Reglerentwurf bei Vernachlässigung gyroskopischer Effekte für die zwei Freiheitsgrade aufgetrennt werden. Zumal die Steifigkeiten in beiden Richtungen gleich sind, braucht somit nur ein Regler repräsentativ für beide ausgelegt zu werden. Ein Blockschaltbild (Bild 4.5) zeigt den linearisierten geschlossenen Kreis für die z-Koordinate.

K_S und K_L bezeichnen die für beide Freiheitsgrade gleichen Kraftkoeffizienten des Stromes bzw. der Lage. Das Meßsystem und der Verstärker sollen proportionales Verhalten haben mit der Übertragungsfunktion:

$$G_V = K_V \quad \text{und} \quad G_M = K_M$$

Die Übertragungsfunktion der einem Lager zukommenden Punktmasse lautet

$$G_{Rot}(s) = \frac{1}{ms^2}.$$

Da der Regler als Spannungsregler gedacht ist, die Kraftkonstante des Stromes sich aber definitionsbemaß auf den Strom bezieht, wird die Wirkung des magnetischen Kreises in ein lineares Aktuatorglied zusammengefaßt. Für lineares Verhalten des ferromagnetischen Kernmaterials lautet das eindimensionale Durchflutungsgesetz

$$NI = \sum_j H_j l_j \doteq \Phi R_{loc}$$

mit

$$R_{loc} = \sum_j \frac{l_j}{A_j \mu_j}.$$

Das Induktionsgesetz für die Aktuatorspule liefert mit r dem ohmschen Widerstand der Spule

$$N\dot{\Phi} = u - ri = R_{loc}N^2(i).$$

oder

$$u = ri + \frac{N^2}{R_{loc}}(i).$$

und nach einer Laplace-Transformation

$$U(s) = \left(r + \frac{N^2}{R_{loc}} s \right) I(s).$$

Die Übertragungsfunktion erhält nach Umformung der obigen Gleichung die Form

$$G_A = \frac{I(s)}{U(s)} = \frac{K_{mk}}{s + \omega_{mk}}$$

mit

$$\begin{aligned} K_{mk} &= \frac{R_{loc}}{N^2} \quad \text{und} \\ \omega_{mk} &= \frac{R_{loc} r}{N^2}. \end{aligned}$$

G_D stellt die Übertragungsfunktion des digitalen Reglers dar, und H_0 das Halteglied nullter Ordnung mit

$$H_0 = \frac{1 - e^{Ts}}{s}$$

und T als der Abtastzeit des digitalen Systems. Die Konvertierung eines Analogsignals und weiters die Berechnung des Stellwertes im Computer, sowie die Konvertierung des digitalen Ausgangs in ein analoges Signal nehmen ein gewisse Zeit in Anspruch. Dies wird durch ein Totzeitglied mit ganzzahligem Vielfachen der Abtastzeit berücksichtigt. Im Laplace-Bereich hat dies die Form

$$G_{tot} = e^{-k_c Ts}$$

Obwohl die eigentliche Regelgröße die Lagekoordinate ist, ist für ein Magnetlager das Störungsverhalten von größerer Bedeutung. Die entsprechende Übertragungsfunktion soll der Ausgangspunkt für die Reglerauslegung sein.

$$H(s) = \frac{Z(s)}{F_{ext}(s)} = \frac{1}{m s^2 + K_S G_A K_V H_0 G_D G_{tot} G_M - K_L} \quad (4.3)$$

4.2 Reglerkonzepte und Stabilität

Denkt man sich das Magnetlager ersetzt durch ein viskoelastisches Strukturlager mit den wesentlichen Parametern für Steifigkeit und Dämpfung K^* beziehungsweise C^* , so ließe sich eine vergleichbare Übertragungsfunktion angeben :

$$H^*(s) = \frac{\frac{1}{m}}{s^2 + s\frac{C^*}{m} + \frac{K^*}{m}}$$

Dies entspricht einem PT2-Glied. Geht man davon aus, daß das geschlossene geregelte Magnetlager-Rotor-System ein dominantes Polpaar hat, so wäre der umgekehrte Weg denkbar, daß man aus der gewünschten Steifigkeit, bzw. Dämpfung ein dominantes Polpaar bestimmt und danach den Regler optimiert. Die Parameter des dominanten Polpaares ergeben sich aus der obigen Übertragungsfunktion zu

$$2\zeta\omega_n = \frac{C^*}{m} \quad \text{und} \quad \omega_n^2 = \frac{K^*}{m}.$$

In weiterer Folge soll es darum gehen, für dieses dominante Polpaar einen digitalen Regler mit Hilfe der Wurzelortskurven zu entwerfen. Für die Reglerauslegung mit den Wurzelortskurven muß das offene System bekannt sein. Man denkt sich das Blockschaltbild 4.5 vor dem Regler aufgeschnitten, und bildet die gesamte Streckenübertragungsfunktion (der Regler wird hier natürlich ausgeklammert). Diese wird mit Auflösung der Rückkopplung der Rotorposition zu

$$G_S(s) = \frac{K_{St}}{(s + \omega_{mk})(s^2 - \omega_L^2)}$$

mit der Polstelle, die sich aus der Lagersteifigkeit ergibt

$$\omega_L^2 = \frac{K_L}{m}$$

und der Verstärkung der offenen Regelstrecke :

$$K_{St} = \frac{K_{mk}K_VK_SK_M}{m}$$

Man sieht sofort die positive Polstelle des offenen Systems aufgrund der negativen Lagersteifigkeit, die für die Instabilität verantwortlich ist (siehe Kapitel 3). Der zweite Pol des Lagers liegt symmetrisch in der negativen Halbebene. Die Modellierung der Spule ergibt ein PT1-Glied mit einem weiteren Pol, der etwas niedriger liegt als der des Lagers.

Da ein digitaler Regler in der Wurzelortskurve ausgelegt werden soll, muß das System in den z-Bereich transformiert werden. Ein zeitkontinuierliches System wird üblicherweise durch ein Differentialgleichungssystem beschrieben, und der leichten Lösbarkeit wegen, Laplace-transformiert. Ein zeitdiskretes System wird durch Differenzgleichungen beschrieben und in den z-Bereich transformiert.

Liegt ein zeitdiskretes Signal $x(kT)$ mit einer Abtastperiode T vor, so ist die z-Transformierte definiert durch

$$X(z) = \mathcal{Z}[x(kT)] = \sum_{k=0}^{\infty} x(kT)z^{-k} .$$

Im z -Bereich gelten, wie im Laplace-Bereich, Sätze für die Transformation von Differentialgleichungen in den z -Bereich. Daraus kann die z -Übertragungsfunktion, die ebenso Pole und Nullstellen enthält, aus Differenzen- oder Differentialgleichungen gewonnen werden. Das kontinuierliche Modell liegt in dieser Arbeit im Laplace-Bereich vor. Es ist aber möglich, den Laplace-Bereich auf den z -Bereich abzubilden. Dabei wird die linke Halbebene des Laplace-Bereiches auf einen Einheitskreis abgebildet. Für diese Abbildung gibt es mehrere Möglichkeiten. Bei einer, der Pol-Nullstellentransformation, geht man von der Annahme aus, daß die Pole und Nullstellen in beiden Bereichen für das dynamische Verhalten maßgeblich sind (siehe [7]). Dabei wird jeder Pol im Laplace-Bereich auf einen Pol im z -Bereich abgebildet. Nullstellen im Unendlichen werden auf Nullstellen bei $z = -1$ abgebildet.

Pole und Nullstellen werden dabei wie folgt transformiert

$$z = e^{sT}$$

mit s als den Pol oder die Nullstelle im Laplace-Bereich, z als den Pol oder die Nullstelle im z -Bereich und der Abtastzeit T . Die Streckenverstärkung im digitalen Bereich ergibt sich aus der niederfrequenten Asymptote bei $s = 0$, bzw. $z = 1$ für den z -Bereich.

Somit ergibt sich die z -Transformierte der Strecke

$$G_S(s) \rightarrow G_{S,z}(z) = K_D \frac{(z + 1)^3}{(z - a)(z - b)(z - c)} .$$

mit den Polen des offenen Systems:

$$\begin{aligned} a &= e^{-\omega_{mk}T} \\ b &= e^{\omega_L T} \\ c &= e^{-\omega_L T} \end{aligned}$$

und der Verstärkung im z -Bereich:

$$K_D = -\frac{K_{St}}{8\omega_{mk}\omega_L^2}(1 - a)(1 - b)(1 - c) .$$

Die durch diverse Konversionen und die Rechenzeit entstehende Totzeit wird dem Rechner angelastet, das heißt schon im z -Bereich berücksichtigt, mit

$$G_{tot,z} = z^{-k_c} .$$

Alles in allem wird die Übertragungsfunktion des offenen Kreises zu

$$G_0(z) = G_{tot}(z) G_{S,z}(z).$$

Damit die Wurzelortskurven für obige Funktion gezeichnet werden können, müssen die Systemparameter bekannt sein. Diese ergaben sich z.T. aus den vorhergehenden Annahmen (Lage- und Stromkoeffizient, halbe Rotormasse des geplanten Rotors), oder aus eigenen Abschätzungen.

Abtastzeit	T	=	100	μs
Verzögerung	k_c	=	1	-
Nominelle Lagersteifigkeit	K^*	=	5E6	N/m
gewünschter Dämpfungsfaktor	ζ	=	0.5	Ns/m
Lagekoeffizient des Lagers	K_L	=	1.3E5	N/m
Stromkoeffizient des Lagers	K_S	=	1.1E2	N/A
Punktmasse	m	=	12	kg
Ohmscher Widerstand der Spule	r	=	10	Ohm
Magn. Widerstand des Kreises	R_{loc}	=	0.8E7	A/Vs
Spannungsverstärker	K_V	=	5	V/V
Meßglied	K_M	=	5000	V/m

G_0 lautet mit diesen Werten für eine reine Streckenverstärkung:

$$G_0(z) = K \frac{(z + 1)^3}{z^2(z - a)(z - b)(z - c)}$$

mit den Polen

$$\begin{aligned} a &= 0.9975 \\ b &= 1.0012 \\ c &= 0.9988 \end{aligned}$$

und der variablen Verstärkung K des offenen Systems. Wie zu erwarten war, liegt ein Pol des offenen Systems außerhalb des Einheitskreises und verursacht die Instabilität. Mit dieser konkreten Funktion kann eine Wurzelortskurve für obige Funktion gezeichnet werden (Bild 4.6 und Bild 4.7). Ebenso wird das gewünschte dominante Polpaar in die Wurzelortskurve eingetragen, die aus dem dominanten Polpaar in der s-Ebene berechnet werden:

$$\begin{aligned} z_{p_1} &= 0.9967 + i 0.0541 \\ z_{p_2} &= 0.9967 - i 0.0541 \end{aligned}$$

Das dominante Polpaar repräsentiert die dynamische Spezifikation des Systems.

Man kann kaum erwarten, daß ein reiner P-Regler das Verhalten des geschlossenen Systems stabilisiert. Ebenso wenig können die statischen und dynamischen Spezifikationen erfüllt werden. Als statische Spezifikation ist eine Positionsabweichung identisch Null erwünscht. Dies kann im Laplace-Bereich nur durch einen I-Anteil erreicht werden, und wäre für ein konventionelles Lager nicht möglich. Um die statischen und dynamischen Spezifikation zu erfüllen, wird ein Korrekturglied gewählt, welches einem PIDD-Regler mit doppeltem D-Verhalten entspricht. Es gelingt nämlich mit diesem Regler, alle Wurzelortskurven innerhalb des Einheitskreises zu halten. Das bedeutet, das lineare System ist unbedingt stabil. Dabei werden sowohl der instabile Pol als auch der langsame Pol des magnetischen Kreises kompensiert. Die Kompensation eines instabilen Poles mit einer positiven Nullstelle ist nur im digitalen Bereich möglich.

Damit ein Punkt in der komplexen Ebene Wurzelortskurvenpunkt ist, muß die Winkelbedingung erfüllt sein. Es wird von diesem Punkt aus je eine Linie zu den Polen und Nullstellen des offenen Systems gezeichnet. Dann werden die Winkel dieser Linien zu der realen Achse aufsummiert. Nullstellen zählen positiv, Pole negativ. Die Summe muß gleich $(2n - 1)\pi$ sein. Der dritte Pol ergibt sich somit aus der Winkelbedingung für die Wurzelortskurve. Die z-Übertragungsfunktion lautet dann für den Regler:

$$G_R = K_R \frac{(z - a)(z - b)(z - z_R)}{z^2(z - 1)}$$

Ebenso muß für das dominante Polpaar die Betragsbedingung, die die Verstärkung des offenen Systems für dieses Punktepaar bestimmt, erfüllt sein. Hiefür werden die Beträge der oben erwähnten Linien gebildet. Die Verstärkung K ergibt sich als Quotient der Nullstellenanteile mit den Polanteilen. Wird diese Verstärkung K in der komplexen Ebene bestimmt, so ergibt sich die Reglerverstärkung zu

$$K_R = \frac{K}{K_D}$$

mit der Verstärkung K_D der Strecke.

Die Nullstelle und die Verstärkung ergeben sich mit den oben angenommenen Werten für dieses System zu

$$\begin{aligned} z_R &= 0.935 \\ K_R &= 5.29E4. \end{aligned}$$

Daraus wird die Gewichtsfolge für den digitalen Regler mit den Abtastwerten $x(k)$ der Position und der Stellgröße $u(k)$ berechnet. Diese ergibt sich aus der Rücktransformation der Reglerfunktion in den Zeitbereich zu

$$u_k = u_{k-1} + 5.29E7 (x(k) - 2.9213 x(k-1) + 2.8433 x(k-2) - 0.9220 x(k-3)).$$

Durch diesen Regler wird die Übertragungsfunktion für das offene System zu:

$$G_0(z) = K \frac{(z - z_{nr})(z + 1)^3}{z^3(z - 1)(z - c)}$$

und es kann wiederum die Wurzelortskurve gezeichnet werden, um so den Einfluß der Streckenverstärkung zu sehen (Bild 4.8 und Bild 4.9). Im Bild 4.9 geht eine Wurzelortskurve durch das dominante Polpaar hindurch.

Das lineare Verhalten kann leicht durch die Sprungantwort des geschlossenen Systems getestet werden. Im Bild 4.10 ist diese für die in der Simulation verwendeten Parameter aufgetragen. Wie erwartet, zeigt das geschlossene System PT2-Verhalten.

Im Nachhinein kann durch Variation der Verstärkung die Stabilität des ausgelegten Reglers untersucht werden, und so deren Auswirkung auf das dominante Polpaar und somit auf Steifigkeit und Dämpfung des Magnetlagers. Da die Punktmasse in die Verstärkung eingeht, kann deren Einfluß durch den Regler kompensiert werden. Eine geringere Masse wird das Lager bei gleichbleibender Reglerverstärkung versteifen. Dieser Einfluß zeigt sich in der Sprungantwort mit einer geringeren Anregelzeit.

Es ist daraus ersichtlich, wie leicht durch Änderung der Verstärkung ein gewünschtes Verhalten des Lagers eingestellt werden kann. Dieser Effekt macht wohl einen eindeutigen Vorteil eines Magnetlagers gegenüber konventionellen Lagern aus, da auch das Lagerverhalten während des Betriebes durch Parameteradaption geändert werden kann, und so eventuell kritische Drehzahlen leicht durchfahren werden können.

An dieser Stelle sei darauf hingewiesen, daß dieser Regler für ein lineares System und für einen gewissen Arbeitspunkt ausgelegt worden ist. Man kann nicht erwarten, daß der Regler für veränderte Systemparameter stabil bleibt, oder die dynamischen Spezifikationen beibehält. Dies wird eine Simulation des nichtlinearen Systems zeigen.

æ

Abbildung 4.2: Abhängigkeit des lokalen magnetischen Widerstandes vom magnetischen Fluß Φ

Abbildung 4.3: Kraftkoeffizient des Stromes eines aktiven Magnetlagers in Abhängigkeit des magnetischen Flußes

Abbildung 4.4: Kraftkoeffizient der Lage eines aktiven Magnetlagers in Abhängigkeit des magnetischen Flußes

Abbildung 4.6: Wurzelortskurve mit reiner Verstärkung

Abbildung 4.7: Detail aus der Wurzelortskurve mit reiner Verstärkung

Abbildung 4.8: Wurzelortskurve mit Regler

Abbildung 4.9: Detail der Wurzelortskurve mit Regler

Abbildung 4.10: Sprungantwort des Systems mit dem entworfenen Regler

Kapitel 5

Simulation des nichtlinearen Gesamtsystems

5.1 Zustandsform des gesamten Systems

Die in den vorhergehenden Kapiteln genau abgegrenzten und formulierten einzelnen Teilmodelle sollen nun in einem Gesamtmodell integriert werden.

Da die Lagerkräfte nichtlinear sind, werden sie als zustandsabhängige Erregung in die rechte Seite der Gleichung 2.1 eingeführt. Im Rotor selbst ist keine Dämpfung vorgesehen. Diese wird neben der Kupplung nur durch die Lagerkräfte in das System eingebracht. Das Differentialgleichungssystem lautet nun:

$$\mathbf{M} \ddot{\mathbf{x}} + (\mathbf{G} + \mathbf{C}_k) \dot{\mathbf{x}} + \mathbf{K} \mathbf{x} = \mathbf{f}(\mathbf{x}, \Phi, t) \quad (5.1)$$

Die Massenmatrix \mathbf{M} und die Gyroskopiematrix \mathbf{G} bleiben unverändert. In die Dämpfungsmatrix \mathbf{C}_k geht jetzt nur mehr die Dämpfung der Kupplung ein. Die Steifigkeitsmatrix enthält nur mehr die Rotor- und Kupplungssteifigkeiten. Das heißt, es treten auf der linken Seite keine Lagerkräfte auf. Um das System statisch kondensieren zu können, darf die Steifigkeitsmatrix nicht singulär werden, das heißt, es müssen Starrkörpermodes vorhanden sein. Dies kann nur der Fall sein, wenn die Steifigkeitsmatrix, mit der kondensiert wird, Nennsteifigkeiten der Lager enthält. So wird für die statische, bzw. für die modale Kondensation, bei der die Steifigkeitsmatrix ebenso für die Bestimmung der Eigenformen notwendig ist, eine Lager-Nennsteifigkeit verwendet. Diese ergibt sich aus den Strom- und Positionskoeffizienten durch Miteinbeziehen der Reglerparameter für einen bestimmten Gleichgewichtszustand (Nennbetriebspunkt). Bezeichnet K_P die Streckenverstärkung, so ist die Nennsteifigkeit gegeben durch :

$$K_e = K_S K_P - K_L$$

Der Reglerentwurf erfolgte nach einem dominanten Polpaar. Das bedeutet, daß das geschlossene System die Nennsteifigkeiten und Nenndämpfungen für das Lager als dominantes Pol-

paar enthält. Mit dieser Nennsteifigkeit, die für beide Freiheitsgrade und für alle Lager gelten soll, kann die Transformationsmatrix \mathbf{R} nach einem Kondensationsverfahren bestimmt werden. Mit $\mathbf{x} = \mathbf{R}\mathbf{y}$ wird das Differentialgleichungssystem 5.1 zu

$$\mathbf{M}^* \ddot{\mathbf{y}} + (\mathbf{G}^* + \mathbf{C}^*) \dot{\mathbf{y}} + \mathbf{K}^* \mathbf{y} = \mathbf{f}^*(\Phi, \mathbf{x}, t) \quad (5.2)$$

mit dem Erregervektor $\mathbf{f}^* = \mathbf{R}^T \mathbf{f}(\Phi, \mathbf{x}, t)$, der jetzt auch vom physikalischen Lagevektor \mathbf{x} und dem Vektor der magnetischen Flüsse Φ abhängt. \mathbf{f}^* läßt sich in einen externen und in einen nichtlinearen Anteil zerlegen,

$$\mathbf{f}^* = \mathbf{f}^*_{ext}(t) + \mathbf{f}^*_{lag}(\Phi, \mathbf{x}, t)$$

mit $\mathbf{f}^* \in \mathcal{R}^m$. Der Vektor der externen Erregungen \mathbf{f}^*_{ext} setzt sich zusammen aus der Unwuchtwirkung und der Wirkung der Schwerkraft an den Knoten des mechanischen Modells. Der Vektor der Lagerkräfte ergibt sich aus der nichtlinearen Wirkung der magnetischen Aktuatoren. Die Lagerkräfte greifen nur an jenen Knoten an, an denen Lager vorgesehen sind. Die Kräfte ergeben sich aus den Ergebnissen des dritten Kapitels.

Das so entstandene Differentialgleichungssystem zweiter Ordnung könnte schon als solches integriert werden. Damit dieses System aber einem in MATLAB implementierten Integrationsverfahren zugeführt werden kann, muß es auf Zustandsform gebracht werden, das heißt

$$\dot{\mathbf{z}} = \mathbf{A} \mathbf{z} + \mathbf{g} \quad (5.3)$$

mit dem Zustandsvektor der Lage

$$\mathbf{z} = \begin{pmatrix} \mathbf{y} \\ \dot{\mathbf{y}} \end{pmatrix},$$

dem neuen Erregervektor

$$\mathbf{g} = \begin{pmatrix} \mathbf{0} \\ \mathbf{f}^* \end{pmatrix},$$

und der linearen Zustandsmatrix

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{E} \\ -\mathbf{M}^{*-1} \mathbf{K}^* & -\mathbf{M}^{*-1} (\mathbf{G}^* + \mathbf{C}^*) \end{pmatrix}.$$

Neben den Gleichungen für das mechanische Teilsystem müssen noch die bestimmenden Differentialgleichungen für die magnetischen Flüsse aufgestellt werden. Diese ergeben sich aus den Gleichungen 3.5, 3.6 und 3.7, und zwar für jeden Aktuator jeweils eine Differentialgleichung für den magnetischen Fluß und eine Differentialgleichung für die magnetische Feldstärke. Für letztere ist jeweils die Bestimmung der differentiellen Permeabilität gemäß

dem vorgestellten Werkstoffmodell notwendig. So ergeben sich für jedes Lager mit den Annahmen für die Geometrie und den Vereinfachungen im Kapitel 3 insgesamt $2p$ nichtlineare Differentialgleichungen und eine zusätzliche nichtlineare Gleichung für die magnetomotorische Kraft der Form

$$\frac{d\Phi_k}{dt} = \frac{e_k(t) - rI_k}{N} \quad (5.4)$$

$$\frac{dH_k}{dt} = \frac{1}{AB'_k} \frac{d\Phi_k}{dt} \quad (5.5)$$

$$NI = H_k L_{tot} + \frac{\Phi_k}{AB'} 2(g_0 - z \cos \alpha \cos \beta_k - y \cos \alpha \sin \beta_k) \quad (5.6)$$

für $k = 1 \dots p$ mit p als der Anzahl der Aktuatoren für ein Lager. Die elektromotorische Kraft $e_k(t)$ für einen Aktuator eines Lagers ist gegeben durch:

$$e_k = K_V \psi_k(u_z(kT), u_y(kT)) + u_0$$

mit $k = 1 \dots p$. Die Spannung u_0 erzeugt den Nennfluß Φ_0 im Nennbetriebszustand. Die beiden Spannungen $u_z(kT)$ und $u_y(kT)$ ergeben sich aus der Gewichtsfolge des digitalen Reglers aus Kapitel 4. In dieser Arbeit ist für jeden Freiheitsgrad jeweils ein Regler vorgesehen. Eingangsgröße für diesen sind die negativ rückgekoppelten Verschiebungen in den Knoten der Lager. Durch die Abtastung, die A/D-Wandlung und die Rechenzeit entsteht eine Verzögerung, die mit dem Totzeitglied miteinbezogen wird. Dies ergibt die Gleichung für die Lagegröße z_L eines Lagers in z -Richtung:

$$z(k) = K_M z_L(k-1)$$

So muß für die Simulation ein Differentialgleichungssystem 1. Ordnung der Dimension $4p+2m$ integriert werden, wenn man davon ausgeht, daß nur zwei Magnetlager vorhanden sind. Zusätzlich muß bei jedem Integrationsschritt für jeden Aktuator die differentielle Permeabilität bestimmt werden, und zu jeder Abtastzeit des Reglers die dazugehörige Stellgröße.

5.2 Simulationsergebnisse

Mit den theoretischen Grundlagen der obigen Abschnitte soll das Verhalten zweier verschiedener Rotoren auf Magnetlagern simuliert werden. Die benötigten numerischen Werte werden hierfür in ein Eingabefile geschrieben, welches im Anhang angegeben ist. Dieses File wird dann dem Simulationsprogramm übergeben, indem es als m-File geladen wird. In diesem Simulationsprogramm werden die Systemmatrizen generiert und in der weiteren Folge kondensiert. Aus den entstehenden Matrizen wird zusammen mit den Daten für das Magnetlager (gemäß

Kapitel 4), und den für dieses Magnetlager bestimmten Reglerdaten das gesamte Differentialgleichungssystem erzeugt. Dieses wird dann einem Integrationsprogramm übergeben. Der Ablauf des Programmes ist im Struktogramm gemäß Abbildung 5.1 ersichtlich.

Die Systemdaten sind im Programm `sysdatN` mit der Nummer N enthalten. Wird das Simulationsprogramm `pw_sim2` aufgerufen, so wird nach dem Namen des zu simulierenden Rotors gefragt. Daraufhin werden die Systemdaten geladen, und es werden die globalen Daten berechnet. Die Berechnung der Systemmatrizen erfolgt unter Zuhilfenahme der Programme `pw_fema` für die Massenmatrix und `pw_fest` für die Steifigkeitsmatrix. Hiernach wird das System mit dem Programm `pw_mkond` modal kondensiert. Die rechte Seite der Differentialgleichungen steht im Programm `pw_sfun2`, das folgende Unterprogramme verwendet:

<code>pw_dyn</code>	Berechnung des Erregervektors zufolge Unwucht.
<code>pw_omega</code>	Berechnung der Drehzahl nach einem einfachen Beschleunigungsmodells mit der Annahme konstanter Winkelbeschleunigung bis zu einer vorgegebenen Maximaldrehzahl.
<code>pw_mag</code>	Berechnung der nichtlinearen Anteile der Magnetlagerkräfte.
<code>pw_BstrN</code>	Berechnung der differentiellen Permeabilität nach dem Modell N .
<code>pw_reg</code>	Berechnung der Stellgrößen nach den im Kapitel 4 bestimmten Reglerparametern.

Als Anfangsbedingung für das mechanische Teilsystem wird die statische Durchbiegung des Rotors verwendet. Der Rotor wird nach Integrationsbeginn mit einer Unwucht bei einer vorher festgelegten Drehfrequenz belastet. Die Anfangsbedingungen für die elektromagnetischen Aktuatoren werden so gesetzt, daß die Arbeitspunkte gemäß Kapitel 4 eingestellt sind.

Damit kann die Integration gestartet werden. Da es sich bei diesem Modell um ein steifes Differentialgleichungssystem handelt, ist es entweder notwendig, kleine Integrationsschritte zu wählen, wodurch man lange Rechenzeiten in Kauf nimmt, oder ein implizites Verfahren zu wählen. Es ergeben sich durch die kleinen Abtastzeiten des digitalen Reglers ohnehin kleine Integrationsintervalle. Deswegen wird ein explizites Verfahren gewählt, wobei die Ordnung möglichst niedrig sein muß, damit die Sprünge im Hysteresemodell noch aufgelöst werden können. Durch eine modale Kondensation können die höchsten auftretenden Eigenschwingungsdauern über den Abtastzeiten gehalten werden, was die Stabilität des Verfahrens garantiert.

Die errechneten Zustandsvektoren als Funktionen der Zeit werden auf die Files `simx.mat`, `sime.mat` (magnetische Größen) und `simu.mat` (Stellgrößen), abgelegt, und können anschließend mit dem Programm `pw_plot2` in einer Graphik anschaulich gemacht werden.

Für die im Kapitel 2 vorgestellten Rotoren wurden einige Integrationen durchgeführt, deren Ergebnisse im folgenden dargestellt sind.

Rotor 1

Die Daten dieses Rotors sind im Kapitel 2, S. 20 angegeben. Da die Unwucht bei der Drehfrequenz 100 rad/s sprunghaft aufgebracht wird, werden die Eigenformen angeregt. Diese müssen bei ausreichender Dämpfung rasch abklingen, und es bleiben nur mehr die erzwungenen Schwingungen. Da die Steifigkeit des Lagers in Relation zur Steifigkeit des elastischen Rotors sehr groß ist, liegen die Schwingungsknoten der ersten Eigenform nahezu in den Lagern selbst. Die Lagerdämpfungen sind somit praktisch nicht wirksam, und die Eigenschwingungen klingen nur sehr langsam ab. Dies erzeugt ein Übergangsverhalten mit einer Schwebung zwischen der Erregerfrequenz und der ersten Eigenfrequenz. Die Bilder 5.3 und 5.4 machen diesen Effekt deutlich. Bilder 5.5 und 5.6 zeigen den dazugehörigen Orbit im Knoten des Lagers 2 (Bild 5.2).

Repräsentativ für die Aktuatoren zeigt Bild 5.7 die magnetischen Zustandsgrößen und Bild 5.8 die Stellgrößen im oberen ($\beta_k = 90 \text{ Grad}$) Aktuator des Lagers 2. Da die Verstärkung sehr groß gewählt wurde, um die Lagersteifigkeit zu erreichen, sind die an der Spule angelegten Spannungen am Beginn sehr groß. Deshalb durchläuft das ferromagnetische Material in der B-H-Magnetisierungsebene zuerst eine große Schleife, bevor es in einen kleinen Grenzyklus hineinläuft.

Es sind in den Bildern der Orbits und der magnetischen Zustandsgrößen Unstetigkeiten zu sehen. Diese ergeben sich aus der Speicherung der errechneten Daten in größeren Intervallen als den der Abtastzeiten des digitalen Reglers.

Rotor 2

Die Daten dieses Rotors sind im Kapitel 2, S. 25 angegeben (Rotor mit überhängender Scheibe). Da die Unwucht bei der Drehfrequenz 100 rad/s aufgebracht wird, wird die erste Eigenformen dieses Rotors angeregt (Die 1.Eigenfrequenz liegt etwa bei 102 rad/s). Da die Steifigkeit des Lagers in Relation zur Steifigkeit des elastischen Rotors sehr groß ist, liegen die Schwingungsknoten der ersten Eigenform nahezu in den Lagern selbst. Die Lagerdämpfungen sind somit praktisch nicht wirksam, und die Unwuchtschwingung klingt stark auf. Die Bilder 5.10 und 5.11 zeigen die Unwuchtschwingung im Knoten der überhängenden Scheibe 1 (Bild 5.9). Bild 5.12 zeigt den dazugehörenden Orbit.

Repräsentativ für die Aktuatoren zeigt Bild 5.17 die magnetischen Zustandsgrößen und Bild 5.18 die Stellgrößen im oberen ($\beta_k = 90 \text{ Grad}$) Aktuator des Lagers 1. Die Bilder 5.13 und 5.14 zeigen die aufklingende Unwuchtschwingung im Knoten des Lagers 1. Bilder 5.15 und 5.16 zeigen den dazugehörenden Orbit.

Es sind in den Bildern der Orbits und der magnetischen Zustandsgrößen Unstetigkeiten zu sehen. Diese ergeben sich aus der Speicherung der errechneten Daten in größeren Intervallen als den der Abtastzeiten des digitalen Reglers.

æ

Abbildung 5.3: Zeitverlauf der Verschiebung in y-Richtung im Knoten des Lagers 2 gemäß Bild 5.2

Abbildung 5.4: Zeitverlauf der Verschiebung in z-Richtung im Knoten des Lagers 2 gemäß Bild 5.2

Abbildung 5.5: Orbit im Knoten des Lagers 2 gemäß Bild 5.2

Abbildung 5.6: Orbit im Knoten des Lagers 2 gemäß Bild 5.2

Abbildung 5.7: Magnetisierungspfad im Aktuatorwerkstoff des vertikalen Aktuators ($\beta_k = 90 \text{ Grad}$) des Lagers 2 gemäß Bild 5.2

Abbildung 5.8: Ausgangssignal des Reglers für den vertikalen Aktuator ($\beta_k = 90 \text{ Grad}$) des Lagers 2 gemäß Bild 5.2

Abbildung 5.10: Zeitverlauf der Verschiebung in y -Richtung im Knoten der Scheibe 1 gemäß Bild 5.9

Abbildung 5.11: Zeitverlauf der Verschiebung in z -Richtung im Knoten der Scheibe 1 gemäß Bild 5.9

Abbildung 5.12: Orbit im Knoten der Scheibe 1 gemäß Bild 5.9

Abbildung 5.13: Zeitverlauf der Verschiebung in y -Richtung im Knoten des Lagers 1 gemäß Bild 5.9

Abbildung 5.14: Zeitverlauf der Verschiebung in z-Richtung im Knoten des Lagers 1 gemäß Bild 5.9

Abbildung 5.15: Orbit im Knoten des Lagers 1 gemäß Bild 5.2

Abbildung 5.16: Orbit im Knoten des Lagers 1 gemäß Bild 5.2

Abbildung 5.17: Magnetisierungspfad im Aktuatorwerkstoff des vertikalen Aktuators ($\beta_k = 90 \text{ Grad}$) des Lagers 1 gemäß Bild 5.9

Abbildung 5.18: Ausgangssignal des Reglers für den vertikalen Aktuator ($\beta_k = 90 \text{ Grad}$) des Lagers 1 gemäß Bild 5.9

Kapitel 6

Zusammenfassung und Ausblick

Aufgabe dieser Diplomarbeit war die Simulation der Schwingungen eines Rotors auf Magnetlagern. Um dies auf einem Computer durchführen zu können, war es notwendig, ein elektromechanisches Modell zu bilden, das in seinem Umfang noch auf einem vorhandenen Digitalrechner zu implementieren ist und dennoch das vorliegende Problem hinreichend genau beschreibt. Die Komplexität des Systems ergab sich aus der FE-Behandlung des Rotors, sowie aus einem nichtlinearen Modells des Magnetlagers und eines dafür entworfenen digitalen Reglers. Es gelang durch modale Kondensation, die sich als für die Simulation günstiger erwies als eine statische Kondensation, bei der noch hochfrequente Anteile erhalten bleiben, das System mit einer akzeptablen Rechenzeit einer Simulation zugänglich zu machen.

Es stellt sich natürlich die Frage, ob ein so kompliziertes Modell überhaupt notwendig ist, und weiter, ob ein einfacheres Modell das Lager- und das Rotorverhalten wiedergeben kann. Nun war es aber nicht die Aufgabe ein linearisiertes Modell zu untersuchen, sondern vielmehr ein System, das für einen Arbeitspunkt ausgelegt worden war, vor allem was die Regelung betrifft, auf dessen Verhalten, wenn der Betriebspunkt weit vom Arbeitspunkt abweicht. Hier sollten die nichtlinearen Effekte des Magnetlagers, die zum Großteil durch die Sättigung des magnetischen Werkstoffes verursacht werden, zum Tragen kommen. Um diese Effekte zu studieren, ist es allerdings notwendig, ein umfassendes Modell zu betrachten, damit sie einer Untersuchung zugänglich sind. Dafür wurde im Rahmen dieser Diplomarbeit ein Simulationsprogramm, das beginnend mit der Datenaufbereitung, die Ausgabe von Zeitverläufen für die mechanischen und magnetischen Zustandsgrößen in graphischer Form ermöglicht.

Es steht jedem Anwender offen, für seinen speziellen Rotor und für seinen eigenen digitalen Regler Untersuchungen durchzuführen und die Integrationsergebnisse in einem dafür geschriebenen Programm zu beurteilen. So können Anfahrvorgänge oder das Durchfahren von Resonanzdrehzahlen leicht simuliert werden. Ein Programm dafür zu erstellen, war ein Teil der vorliegenden Diplomarbeit.

Als weiteres Ergebnis dieser Diplomarbeit liegt unter anderem ein Programmpaket vor, sowohl für die Untersuchung eines Rotors, als auch für die Reglerauslegung. Die darin enthaltenen Programme ergaben sich oft zwangsweise aus Voruntersuchungen, oder aus Prozeduren für das Integrationsprogramm, oder entsprangen einfach meinem Interesse an der Materie.

Offen geblieben sind sicher eine Reihe von Problemen, die Gegenstand einer weiteren Arbeit auf diesem Gebiet sein könnten, wie z. B.:

Regleroptimierung In der hier vorliegende Diplomarbeit wurde lediglich ein PID-Regler verwendet. Um die Qualitäten des Magnetlagers zu nutzen, könnte ein Regler entworfen werden, der sich adaptiv an den Betriebspunkt anpaßt, oder ein leichtes Durchfahren durch die Resonanzen zuläßt.

Reglerausfall Da die Stabilität des Systems von der Güte des Reglers abhängig ist, wäre es ein interessantes Problem, das Verhaltens eines schnell rotierenden Mehrscheibenrotors nach einem Reglerausfall zu simulieren.

Es tut sich somit eine Reihe von interessanten Aufgabenstellungen auf, die zum Teil mit den hier geschriebene Programmen unmittelbar behandelt werden könnten, oder wo es nur einer geringen Modifikation bedürfte. Die in MATLAB geschriebene TOOL-Box auszuwerten und zu nutzen, sowie für verschiedenste Regler Simulationen durchzuführen, könnte Aufgabe von weiteren Arbeiten sein.

Literaturverzeichnis

- [1] A. Haferl. “*Magnetisierungseffekte in aktiven Magnetlagern*”,
Diplomarbeit am Institut für Maschinendynamik und Meßtechnik, TU Wien, Juli 1991.
- [2] H. Springer. “*Nonlinearities and Hysteresis Effects in ferromagnetic circuits of electromagnetic bearings*”
Department of Mechanical and Aerospace Engineering, University of Virginia, 1989.
- [3] P. E. Allaire, D. W. Lewis, J. D. Knight. “*Aktive Vibration Control of a Single Mass Rotor on Flexible Supports*”
Department of Mechanical and Aerospace Engineering, University of Virginia, 1983.
- [4] F. J. Keith, R. D. Williams, P. E. Allaire. “*Digital Control System Design for Active Magnetic Bearings*”,
Department of Mechanical and Aerospace Engineering, University of Virginia, 1989.
- [5] H. Springer. “*Maschinendynamik*” Vorlesungsskriptum,
Institut für Maschinendynamik und Meßtechnik, TU Wien
- [6] F. G. Rammerstorfer. “*Finite Elemente und andere numerische Ingenieursmethoden*”
Vorlesungsskriptum,
Institut für Leicht- und Flugzeugbau, TU Wien.
- [7] H. P. Jörgl. “*Einführung in die digitale Regelung*” Vorlesungsskriptum,
Institut für Maschinen und Prozeßautomatisierung, TU Wien 1990.
- [8] D. R. Hill, Temple University. C. B. Moler, The Mathworks. “*Experiments in Computational Matrix Algebra*”,
The Rondu House/Birkhausen Mathematics Series, New York 1988.
- [9] H. Kopka. \LaTeX , “*Eine Einführung*” Addison-Wesley GmbH,
München 1988.

Anhang A

Anhang

Im Rahmen dieser Diplomarbeit wurden nicht nur für die eigentliche Simulation Programme in MATLAB geschrieben, sondern darüber hinaus ergab sich aus diversen Problemen wie das Bestimmen der Eigenwerte oder die Berechnung der Lage und Stromkoeffizienten eine Menge von für die Simulation dienlichen Programmen. So konnte eine eigene TOOL-Box für MATLAB geschaffen werden, die Programme zur Untersuchung von Rotoren auf Magnetlagern oder auf konventionellen Lagern beinhaltet.

Diese Programme gliedern sich in zwei Bereiche.

A1.: Programme, die nur unter Verwendung von globalen Variablen zusammen mit dem Simulationsprogramm arbeiten.

A2.: Programme, die als gewöhnliche m-Files in jeder Umgebung funktionieren.

Die Programme gestatten, wie dies in den meisten Programmiersprachen der Fall ist, Kommentare hinzuzufügen. Deshalb wird verzichtet, die Programme, die ohnehin fast wie ein Mathematikbuch zu lesen sind, noch extra zu erklären. Werden diese Programme innerhalb von MATLAB verwendet, so kann mit der Hilfe Funktion eine entsprechende Erklärung abgerufen werden. Sonst bleibt noch die Hoffnung, diese mittels Editor durchzugehen.

A.1 Simulationsprogramm mit Input-File

```
Daten-Inputfile:
%
%           sysdat1
%
% sysdat1.m ist ein m-File, in dem die Systemdaten eines Rotors
% mit Magnetlagern festgehalten werden.
% Die Eingabe der Systemparameter fuer einen Rotor ist nach Knoten
% geordnet.

% Materialdaten der Welle

rot(1) = 7.85E3;      % kg/m3      Dichte
rot(2) = 210E9;      % N/m2      E-Modul
rot(3) = 9.81;       % m/s2      Erdbeschleunigung

% Anzahl der Knoten

kn = [22];

% Eingabe der Knotenparameter
% knot(i,:) = [Masse, polares MTRM, axiales MTRM, Exzentrizitaet,Phase]
%           [kg]      [kgm2]      [kgm2]      [m]      rad

knot(2,:) = [ 3.25,    3.4E-3,    1.7E-3, 1E-5, 0 ];
knot(7,:) = [ 4.1,    355.6E-3,   177.8E-3, 1E-5, 1 ];
knot(11,:) = [ 4.1,   355.6E-3,   177.8E-3, 1E-5, 0 ];
knot(15,:) = [ 4.1,   355.6E-3,   177.8E-3, 1E-5, 2 ];
knot(20,:) = [ 3.25,    3.4E-3,    1.7E-3, 1E-5, 3 ];
knot(21,:) = [ 0.7,     1E-3,     0.5E-3, 0, 0 ];
knot(22,:) = [ 1.2,     1E-3,     0.5E-3, 0, 0 ];

% Eingabe der Parameter der FE zwischen zwei Knoten:
% fe(i,:) = [Laenge, Durchmesser]
%           [m]      [m]

fe(1,:) = [ 50E-3,    30E-3 ];
fe(2,:) = [ 44E-3,    30E-3 ];
fe(3,:) = [ 13E-3,   34.5E-3 ];
fe(4,:) = [ 100E-3,   25E-3 ];
fe(5,:) = [ 100E-3,   25E-3 ];
fe(6,:) = [ 24E-3,    36E-3 ];
fe(7,:) = [ 16E-3,    36E-3 ];
fe(8,:) = [ 100E-3,   25E-3 ];
fe(9,:) = [ 100E-3,   25E-3 ];
fe(10,:) = [ 24E-3,   36E-3 ];
fe(11,:) = [ 16E-3,   36E-3 ];
fe(12,:) = [ 100E-3,   25E-3 ];
fe(13,:) = [ 100E-3,   25E-3 ];
fe(14,:) = [ 16E-3,   36E-3 ];
```

```

fe(15,:) = [ 24E-3, 36E-3 ];
fe(16,:) = [ 100E-3, 25E-3 ];
fe(17,:) = [ 100E-3, 25E-3 ];
fe(18,:) = [ 13E-3, 34.5E-3 ];
fe(19,:) = [ 44E-3, 30E-3 ];
fe(20,:) = [ 80E-3, 30E-3 ];
fe(21,:) = [ 45E-3, 25E-3 ];

% Eingabe der Lagerparameter
% Lagerparameter im Lager i
% lag(i,1:4) = [k_y, k_z, k_b, k_g] Ersatzlagersteifigkeiten [N/m], [Nm/rad]
% lag(i,5:8) = [d_y, d_z, d_b, d_g] Ersatzlagerdaempfung [Ns/m], [Nms/rad]
% lag(i,9) = [lp] Lagerposition

lag(1,1:4) = [5E6, 5E6, 0, 0];
lag(1,5:8) = [1E2, 1E2, 0, 0];
lag(1,9) = [2];
lag(2,1:4) = [5E6, 5E6, 0, 0];
lag(2,5:8) = [1E2, 1E2, 0, 0];
lag(2,9) = [20];

% Eingabe der Kupplungsparameter
% kup(1:4) = [k_y, k_z, k_b, k_g] Kupplungssteifigkeiten [N/m], [Nm/rad]
% kup(5:8) = [d_y, d_z, d_b, d_g] Kupplungsdaempfung [Ns/m], [Nms/rad]
% kup(9) = [kp] Kupplungsposition

kup(1:4) = [1, 1, 2.5E2, 2.5E2];
kup(5:8) = [1, 1, 1, 1];
kup(9) = [22];

% Eingabe fuer die statische Kondensation
% Bem.: Knoten, auf denen Lager bzw. Massen sitzen,
% koennen nicht angegeben werden.

% Knoten, die generell nicht von Interesse sind

kond1 = [1,5,9,13,17];

% Knoten, deren Verdrehungen nicht von Interesse sind

kond2 = [3,4,6,8,10,12,14,16,18,19];

% Eingabe fuer die modale Kondensation
% Grenzfrequenz = 1E6; [1/s]

grenzfr = 1E3;

% Eingabe der Systemparameter des elektrischen Teilsystems

g0 = 0.5E-3; % m Nominaler Luftspalt
Flaeche = 100E-6; % m2 Fluszquerschnittsflaeche
Ltot = 100E-3; % m mittlere Feldlinienlaenge

```

```

mu0    = 4*pi*1E-7;      % H/m    Permeabilitaet im Vakuum mu
r      = 10;             % Ohm   Spulenwiderstand
N      = 570;           % -    Windungszahl
p      = 4;             % -    Anzahl der Polschuhpaare
alfak  = pi/4;         % rad   Teilungswinkel
Kv     = 5;             % V/V   Verstaerker
Km     = 5000;         % V/m   Meszverstaerker

% Reglerparameter

kp     = 1E3;
k1t    = 1;
k2t    = -2.9213;
k3t    = 2.8433;
k4t    = -0.9220;
ku1    = 1;

%
%
%           sysdat2
%
%
% sysdat2.m ist ein m-File, in dem die Systemdaten eines Rotors
% mit Magnetlagern festgehalten werden.
% Die Eingabe der Systemparameter fuer einen Rotor ist nach Knoten
% geordnet.

% Materialdaten der Welle

rot(1) = 7.85E3;      % kg/m3   Dichte
rot(2) = 210E9;      % N/m2   E-Modul
rot(3) = 9.81;       % m/s2   Erdbeschleunigung

% Anzahl der Knoten

kn = [22];

% Eingabe der Knotenparameter
% knot(i,:) = [Masse, polares MTRM, axiales MTRM, Exzentrizitaet,Phase]
%           [kg]           [kgm2]           [kgm2]           [m]           rad

knot(2,:) = [ 4.1, 355.6E-3, 177.8E-3, 1E-5, 1 ];
knot(7,:) = [ 3.25, 3.4E-3, 1.7E-3, 1E-5, 0 ];
knot(11,:) = [ 4.1, 355.6E-3, 177.8E-3, 1E-5, 0 ];
knot(15,:) = [ 4.1, 355.6E-3, 177.8E-3, 1E-5, 2 ];
knot(20,:) = [ 3.25, 3.4E-3, 1.7E-3, 1E-5, 3 ];
knot(21,:) = [ 0.7, 1E-3, 0.5E-3, 0, 0 ];
knot(22,:) = [ 1.2, 1E-3, 0.5E-3, 0, 0 ];

% Eingabe der Parameter der FE zwischen zwei Knoten:

```

```

% fe(i,:) = [Laenge, Durchmesser]
%           [m]           [m]

fe(1,:) = [ 50E-3, 30E-3 ];
fe(2,:) = [ 44E-3, 30E-3 ];
fe(3,:) = [ 13E-3, 34.5E-3 ];
fe(4,:) = [ 100E-3, 25E-3 ];
fe(5,:) = [ 100E-3, 25E-3 ];
fe(6,:) = [ 24E-3, 36E-3 ];
fe(7,:) = [ 16E-3, 36E-3 ];
fe(8,:) = [ 100E-3, 25E-3 ];
fe(9,:) = [ 100E-3, 25E-3 ];
fe(10,:) = [ 24E-3, 36E-3 ];
fe(11,:) = [ 16E-3, 36E-3 ];
fe(12,:) = [ 100E-3, 25E-3 ];
fe(13,:) = [ 100E-3, 25E-3 ];
fe(14,:) = [ 16E-3, 36E-3 ];
fe(15,:) = [ 24E-3, 36E-3 ];
fe(16,:) = [ 100E-3, 25E-3 ];
fe(17,:) = [ 100E-3, 25E-3 ];
fe(18,:) = [ 13E-3, 34.5E-3 ];
fe(19,:) = [ 44E-3, 30E-3 ];
fe(20,:) = [ 80E-3, 30E-3 ];
fe(21,:) = [ 45E-3, 25E-3 ];

% Eingabe der Lagerparameter
%           Lagerparameter im Lager i
% lag(i,1:4) = [k_y, k_z, k_b, k_g] Ersatzlagersteifigkeiten [N/m], [Nm/rad]
% lag(i,5:8) = [d_y, d_z, d_b, d_g] Ersatzlagerdaempfungem [Ns/m], [Nms/rad]
% lag(i,9) = [lp] Lagerposition

lag(1,1:4) = [5E6, 5E6, 0, 0];
lag(1,5:8) = [1E2, 1E2, 0, 0];
lag(1,9) = [7];
lag(2,1:4) = [5E6, 5E6, 0, 0];
lag(2,5:8) = [1E2, 1E2, 0, 0];
lag(2,9) = [20];

% Eingabe der Kupplungsparameter
% kup(1:4) = [k_y, k_z, k_b, k_g] Kupplungssteifigkeiten [N/m], [Nm/rad]
% kup(5:8) = [d_y, d_z, d_b, d_g] Kupplungsdaempfungem [Ns/m], [Nms/rad]
% kup(9) = [kp] Kupplungsposition

kup(1:4) = [1, 1, 2.5E2, 2.5E2];
kup(5:8) = [1, 1, 1, 1];
kup(9) = [22];

% Eingabe fuer die statische Kondensation
% Bem.: Knoten, auf denen Lager bzw. Massen sitzen,
% koennen nicht angegeben werden.

% Knoten, die generell nicht von Interesse sind

```

```

kond1 = [1,5,9,13,17];

% Knoten, deren Verdrehungen nicht von Interesse sind

kond2 = [3,4,6,8,10,12,14,16,18,19];

%   Eingabe fuer die modale Kondensation
% Grenzfrequenz = 1E6;   [1/s]

grenzfr = 1E3;

%   Eingabe der Systemparameter des elektrischen Teilsystems

g0      = 0.5E-3;          % m      Nominaler Luftspalt
Flaeche = 100E-6;        % m2     Fluszquerschnittsflaeche
Ltot    = 100E-3;        % m      mittlere Feldlinienlaenge
mu0     = 4*pi*1E-7;     % H/m   Permeabilitaet im Vakuum mu
r       = 10;            % Ohm    Spulenwiderstand
N       = 570;           % -     Windungszahl
p       = 4;             % -     Anzahl der Polschuhpaare
alfak   = pi/4;          % rad   Teilungswinkel
Kv      = 5;             % V/V   Verstaerker
Km      = 5000;          % V/m   Meszverstaerker

% Reglerparameter

kp      = 1E3;
k1t     = 1;
k2t     = -2.9213;
k3t     = 2.8433;
k4t     = -0.9220;
ku1     = 1;

%                               pw_sim2
%
% Dieses Programm simuliert einen Rotor auf Magnetlagern
%
clc
disp('');
disp('');
disp(' Dieses Programm simuliert einen Rotor auf Magnetlagern');
disp('');
name = pw_inp(' Welches Modell soll geladen werden ', 'sysdat0');
eval(name);
pw_trans;

global kn mkn ex phi lagn invMstern MKstern MGstern MDstern R Fstat dim;
global g0 Flaeche Ltot mu0 r N p alfak betak;
global Kv Km u0 K1 K2 K3 K4;
clc;

```

```

disp('');
disp(' Generiere Systemmatrizen');

[M,G,D,Der,K,Ker] = pw_matg(kn,knot,fe,lag,kup,rot);

wahl = pw_inp(' Modale (m) oder statische (s) Kondensation ', 'm');
if wahl == 's'
    [Mstern,Gstern,Dstern,Kstern,R] = pw_skond(M,G,D,K,Ker,kond1,kond2);
end
if wahl == 'm'
    [Mstern,Gstern,Dstern,Kstern,R] = pw_mkond(M,G,D,K,Ker,grenzfr);
end

disp('');
disp(' Berechne globale Matrizen');
invMstern = inv(Mstern);
MKstern = -invMstern*Kstern;
MGstern = -invMstern*Gstern;
MDstern = -invMstern*Dstern;

[dim,nix] = size(Mstern);

% Startvektor
% Magnetisierungszustand im Arbeitspunkt

B0 = 0.5;
Phi0 = B0*Flaeche;
A1 = 22.7971;    % A/m
A2 = 1.18639;    % T-1
mu = (1/A1*A2)*(cos(A2*B0))^2;
H0 = B0/mu;
Rloc = (Ltot/mu + 2*g0/mu0)/Flaeche;

% Statische Durchbiegung

y0 = zeros(2*dim+4*p,1);
Fstat = pw_Fstat(kn,knot,fe,rot);
y0(1:dim) = inv(R'*Ker*R)*R'*Fstat;
for i = 1:p
    y0(2*dim+i) = Phi0;
    y0(2*dim+2*p+i) = Phi0;
    y0(2*dim+p+i) = H0;
    y0(2*dim+3*p+i) = H0;
end

% Berechnung der Verteilfunktionsmatrix
% fuer die Aktuatoren

K0 = zeros(p,2);
for i = 1:p
    K0(i,1) = sin(betak(i));
    K0(i,2) = cos(betak(i));
end

```

```

K1 = kp*k1t*K0;
K2 = kp*k2t*K0;
K3 = kp*k3t*K0;
K4 = kp*k4t*K0;
u0 = r*Rloc*Phi0/N;
u0 = u0*ones(2*p,1);

clc;
disp('');
disp(' Berechnungen fuer die Simulation abgeschlossen');
disp('');
t0d = 0;
t0 = pw_inp(' Startzeit t0 ',0);
tend = pw_inp(' Abbruchzeit tend ',10);
dt = pw_inp(' Abtastzeit dt (mus)',10);
int = pw_inp(' Rechenintervalle per Abtastzeit',1);
dt = dt*1E-6/int;

disp('');
disp(' Speichere berechnete Daten unter simx2.mat/sime2.mat/simu2.mat');
disp('');
disp(' Start mit Tastendruck' );
pause

delete simu2.mat;
delete simx2.mat;
delete sime2.mat;
pw_ne('pw_sfunk2',t0,tend,dt,int,y0);
clear;
%Ende

```

```

function pw_exrk2(FunFcn, t0, tfinal, h, iter, y0)
%
%           pw_exrk2(FunFcn, t0, tfinal, dt, y0)
%
%   Explizites Runge-Kutta-Verfahren
%   pw_exrk2 integriert ein Differentialgleichungssystem
%   der Gestalt
%
%           yp = f(t,y)
%
%   fuer das Programm pw_sim2 wobei die Funktion f durch
%   das Modul FunFcn erzeugt wird.
%   Der Zustandsvektor zerfaellt in x und xp, deren Dimension
%   dim als globale Variable definiert sein musz, um in den
%   hier verwendeten Modulen einen Wert zu haben und in den
%   Vektor der elektrischen Groeszen xe.
%   Die in den Integrationsschritten errechneten Zustands -
%   vektoren werden in simx2.mat respektive in sime2.mat gesichert.
%   Die Stellgroesze in simu2.mat

```

```

% Die Fehlberg Koeffizienten:
alpha = [1/4 3/8 12/13 1 1/2]';
beta = [ [ 1 0 0 0 0 0 ]/4
         [ 3 9 0 0 0 0 ]/32
         [ 1932 -7200 7296 0 0 0 ]/2197
         [ 8341 -32832 29440 -845 0 0 ]/4104
         [-6080 41040 -28352 9295 -5643 0 ]/20520 ]';
gamma = [ [902880 0 3953664 3855735 -1371249 277020]/7618050
          [-2090 0 22528 21970 -15048 -27360]/752400 ]';
pow = 1/5;
tol = 1.E-6;

% Initialisierung
step = 0;
clc
string = sprintf(' Schritt %g, t = %g',step,t0);
disp(string);

t = t0;
y = y0(:);
f = y*zeros(1,6);
xm = real(R*y0(1:dim));
x0t = -Km*xm;
x1t = x0t;
x2t = x1t;
x3t = x2t;
x4t = x3t;
u1t = zeros(2*p,1);
u0t = pw_reg(u1t,x1t,x2t,x3t,x4t);
u = Kv*u0t + u0;

up = [t; u0t];
xp = [t; xm];
ye = [t; y0(2*dim+1:2*dim+4*p)];
rvprintf('simu2.mat',up);
rvprintf('simx2.mat',xp);
rvprintf('sime2.mat',ye);

tau = tol * max(norm(y, 'inf'), 1);

% Hauptschleife
while (t < tfinal)

    clc
    string = sprintf(' Schritt %g, t = %g',step,t);
    disp(string);

    % Berechnen der Steigungen
    f(:,1) = feval(FunFcn,t,y,u);
    for j = 1:5
        f(:,j+1) = feval(FunFcn, t+alpha(j)*h, y+h*f*beta(:,j), u);
    end
end

```

```

% Fehlerabschaetzung
delta = norm(h*f*gamma(:,2),'inf');
tau = tol*max(norm(y,'inf'),1.0);

% Erneuerung der Integrationsvariablen
if delta <= tau
    t = t + h;
    y = y + h*f*gamma(:,1);

    if rem(step,iter) == 0
        u1t = u0t;
        xm = real(R*y(1:dim));
        x4t = x3t;
        x3t = x2t;
        x2t = x1t;
        x1t = x0t;
        x0t = -Km*xm;
        u0t = pw_reg(u1t,x1t,x2t,x3t,x4t);
        u = Kv*u0t + u0;
        if rem(step,50*iter) == 0
            up = [t; u0t];
            xp = [t; xm];
            ye = [t; y(2*dim+1:2*dim+4*p)];
            rvprintf('simu2.mat',up);
            rvprintf('simx2.mat',xp);
            rvprintf('sime2.mat',ye);
        end
    end
    step = step + 1;
% disp('rkK');
% keyboard
else
    t = tfinal + h;
    disp('');
    disp('Berechnungen abgebrochen,');
    disp('Singularitaet wahrscheinlich');
    disp('');
end

end;

function pw_ne(FunFcn, t0, tfinal, h, iter, y0)
%
%           pw_ne(FunFcn, t0, tfinal, dt, y0)
%
%   Explizites Newton-Euler-Verfahren
%   pw_ne integriert ein Differentialgleichungssystem
%   der Gestalt
%           yp = f(t,y)
%
%   fuer das Programm pw_sim2 wobei die Funktion f durch

```

```

% das Modul FunFcn erzeugt wird.
% Der Zustandsvektor zerfaellt in x und xp, deren Dimension
% dim als globale Variable definiert sein musz, um in den
% hier verwendeten Modulen einen Wert zu haben und in den
% Vektor der elektrischen Groeszen xe.
% Die in den Integrationsschritten errechneten Zustands -
% vektoren werden in simx2.mat respektive in sime2.mat gesichert.
% Die Stellgroesze in simu2.mat

```

```

% Initialisierung
step = 0;
clc
string = sprintf(' Schritt %g, t = %g',step,t0);
disp(string);

```

```

t = t0;
y = y0(:);
xm = real(R*y0(1:dim));
x0t = -Km*xm;
x1t = x0t;
x2t = x1t;
x3t = x2t;
x4t = x3t;
u1t = zeros(2*p,1);
u0t = pw_reg(u1t,x1t,x2t,x3t,x4t);
u = Kv*u0t + u0;

```

```

up = [t; u0t];
xp = [t; xm];
ye = [t; y0(2*dim+1:2*dim+4*p)];
rvprintf('simu2.mat',up);
rvprintf('simx2.mat',xp);
rvprintf('sime2.mat',ye);

```

```

% Hauptschleife
while (t < tfinal)

    clc
    string = sprintf(' Schritt %g, t = %g',step,t);
    disp(string);

    % Berechnen der Steigungen
    f = feval(FunFcn,t,y,u);

    t = t + h;
    y = y + h*f;

    if rem(step,iter) == 0
        u1t = u0t;
        xm = real(R*y(1:dim));
        x4t = x3t;
    end
end

```

```

        x3t = x2t;
        x2t = x1t;
        x1t = x0t;
        x0t = -Km*xm;
        u0t = pw_reg(u1t,x1t,x2t,x3t,x4t);
        u   = Kv*u0t + u0;
        if rem(step,250) == 0
            up = [t; u0t];
            xp = [t; xm];
            ye = [t; y(2*dim+1:2*dim+4*p)];
            rvprintf('simu2.mat',up);
            rvprintf('simx2.mat',xp);
            rvprintf('sime2.mat',ye);
        end
    end
    step = step + 1;
end;

```

```

function fun = pw_sfun2(t,y,ereg)
%
%           fun = pw_sfun2(t,y,ereg)
%
%   Diese Funktion generiert die fuer die Simulation
%   notwendige Funktion fun
%
oml = pw_omega(t);
Fdyn = pw_fdyn(oml,t);
Flag = pw_flag(y);
F = Fstat + Fdyn + Flag;
Nlin = zeros(2*dim+4*p,1);
Nlin(dim+1:2*dim) = invMstern*R'*F;
Nlin(2*dim+1:2*dim+4*p) = pw_mag(y,ereg);
L = zeros(2*dim+4*p,2*dim+4*p);
L(1:dim,dim+1:2*dim) = eye(dim);
L(dim+1:2*dim,1:dim) = MKstern;
L(dim+1:2*dim,dim+1:2*dim) = oml*MGstern + MDstern;

fun = L*y + Nlin;

```

```

function fdyn = pw_fdyn(omegal,t1)
%
%           fdyn = pw_fdyn(omegal,t1)
%
%   pw_fdyn erzeugt den dynamischen Anteil des Stoervektors
%   in Abhaengigkeit des Massenvektors m, des Unwuchtvektors e, der
%   Kreisfrequenz omega und der Zeit t.
%   mn, ex und phi muessen global definiert sein.
%
fdyn = zeros(4*kn,1);

```

```

for i = 1:kn
    fdyn(4*i - 3) = mkn(i)*ex(i)*omegal^2*sin(omegal*tl-phi(i));
    fdyn(4*i - 2) = mkn(i)*ex(i)*omegal^2*cos(omegal*tl-phi(i));
end

```

```

function Flagl = pw_flag(y)
%
%           Flag = pw_flag(y)
%
% pw_flag erzeugt die Lagerkraefte in Abhaengigkeit
% der magnetischen Fluesse
%
ym = y(2*dim+1:2*dim+4*p);
Flagl = zeros(4*kn,1);

for i = 1:p
    Flagl(4*lagn(1)-2) = Flagl(4*lagn(1)-2) + ..
        ym(i)^2*cos(alfak)*cos(betak(i))/(Flaeche*mu0);
    Flagl(4*lagn(1)-3) = Flagl(4*lagn(1)-3) + ..
        ym(i)^2*cos(alfak)*sin(betak(i))/(Flaeche*mu0);
    Flagl(4*lagn(2)-2) = Flagl(4*lagn(2)-2) + ..
        ym(2*p+i)^2*cos(alfak)*cos(betak(i))/(Flaeche*mu0);
    Flagl(4*lagn(2)-3) = Flagl(4*lagn(2)-3) + ..
        ym(2*p+i)^2*cos(alfak)*sin(betak(i))/(Flaeche*mu0);
end

```

```

function maggy = pw_mag(y,ereglerl)
%
%           maggy = pw_mag(y,ereglerl)
%
% pw_mag beschreibt die Zustandsgleichungen
% elektrischen Groeszen. Elektrische und Magnetische
% Konstanten muessen global definiert sein.
%
x = real(R*y(1:dim));
ym = y(2*dim+1:2*dim+4*p);

maggy = zeros(4*p,1);
for i = 1:p
    maggy(i) = ereglerl(i)/N - (ym(i+p)*Ltot + 2*ym(i)*(g0 - ..
        x(4*lagn(1)-2)*cos(alfak)*cos(betak(i)) - ..
        x(4*lagn(1)-3)*cos(alfak)*sin(betak(i)))/(mu0*Flaeche))*r/N^2;
    maggy(2*p+i) = ereglerl(p+i)/N - (ym(3*p+i)*Ltot + 2*ym(2*p+i)*(g0 - ..
        x(4*lagn(2)-2)*cos(alfak)*cos(betak(i)) - ..
        x(4*lagn(2)-3)*cos(alfak)*sin(betak(i)))/(mu0*Flaeche))*r/N^2;
end

for i = 1:p
    Bp = maggy(i)/Flaeche;

```

```

B = ym(i)/Flaeche;
H = ym(p+i);
Bstrich = pw_Bstr3(H,B,Bp);
maggy(p+i) = maggy(i)/(Flaeche*Bstrich);
Bp = maggy(2*p+i)/Flaeche;
B = ym(2*p+i)/Flaeche;
H = ym(3*p+i);
Bstrich = pw_Bstr3(H,B,Bp);
maggy(3*p+i) = maggy(2*p+i)/(Flaeche*Bstrich);
end

```

```

function bstr1 = pw_Bstr1(Hl,Bl,Bpl);
%
%           bstr1 = pw_Bstr1(Hl,Bl,Bpl)
%
% pw_Bstr1 erzeugt die differentielle Permeabilitaet fuer den
% Magnetisierungszustand eines Lagers nach einem bilinearen Modell
% mit mue fuer B < Bsatt und mu0 fuer B > Bsatt.
%
Bsatt = 1.3;    % Tesla
mur = 3E4;
mu0 = 4*pi*1E-7;

```

```

function bstr1 = pw_Bstr2(Hl,Bl,Bpl);
%
%           bstr1 = pw_Bstr2(Hl,Bl,Bpl)
%
% pw_Bstr2 erzeugt die differentielle Permeabilitaet fuer den
% Magnetisierungszustand eines Lagers
%
A1 = 22.7971;    % A/m
A2 = 1.18639;    % T-1
A3 = -11.817;    % -
A4 = 0.44265;    % -
Bkr = 1.3;       % T
Bcl = 1.3;       % T
mucl = 3E-5;     % Vs/Am
alf = 10;        % T-1

```

```

if abs(Bl) <= Bkr
    f = A1*tan(A2*Bl);
    fstr = A1*A2/(cos(A2*Bl))^2;
end
if Bl > Bkr
    f = A1*tan(A2*Bkr) + (Bl-Bkr)/mucl;
    fstr = 1/mucl;
end

```

```

if Bl < -Bkr

```

```

f = -A1*tan(A2*Bkr) + (B1+Bkr)/mucl;
fstr = 1/mucl;
end

if abs(B1) < Bcl
    g = fstr*(1-A3*exp(-A4*abs(B1)/(Bcl-abs(B1))));
end
if abs(B1) >= Bcl
    g = fstr;
end

bstr1 = 1/(alf*sign(Bpl)*(f-H1)+g);

function bstr1 = pw_Bstr3(H1,B1,Bpl);
%
%           bstr1 = pw_Bstr3(H1,B1,Bpl)
%
% pw_Bstr2 erzeugt die differentielle Permeabilitaet fuer den
% Magnetisierungszustand eines Lagers
%
A1 = 22.7971;    % A/m
A2 = 1.18639;   % T-1
A3 = -11.817;   % T-1
A4 = 0.44265;   % T-1
Bkr = 1.3;
Bcl = 1.3;      % T
mucl = 3E-5;    % H/m
alf = 10;       % T-1

if abs(B1) <= Bkr
    fstr = A1*A2/(cos(A2*B1))^2;
end
if B1 > Bkr
    fstr = 1/mucl;
end

if B1 < -Bkr
    fstr = 1/mucl;
end

bstr1 = 1/fstr;

function om = pw_omega(t1)
%
%           om = pw_omega(t1)
%
% omega erzeugt die Kreisfrequenz als Function der Zeit t
%
```

```

teck = .2;
omax = 100;

function uregl = pw_reg(u1,x0z,x1z,x2z,x3z);
%
%           ureg = pw_reg(u1,x0z,x1z,x2z,x3z)
%
% pw_reg erzeugt den Steuervektor der Eingangsspannungen
% fuer die Magnetlager
%

uregl(1:p,1) = u1(1:p) + ..
    K1*[x0z(4*lagn(1)-3); x0z(4*lagn(1)-2)] + ..
    K2*[x1z(4*lagn(1)-3); x1z(4*lagn(1)-2)] + ..
    K3*[x2z(4*lagn(1)-3); x2z(4*lagn(1)-2)] + ..
    K4*[x3z(4*lagn(1)-3); x3z(4*lagn(1)-2)];

uregl(p+1:2*p,1) = u1(p+1:2*p) + ..
    K1*[x0z(4*lagn(2)-3); x0z(4*lagn(2)-2)] + ..
    K2*[x1z(4*lagn(2)-3); x1z(4*lagn(2)-2)] + ..
    K3*[x2z(4*lagn(2)-3); x2z(4*lagn(2)-2)] + ..
    K4*[x3z(4*lagn(2)-3); x3z(4*lagn(2)-2)];

%
%           pw_plot2
%
% pw_plot2 zeigt die in pw_sim2 berechneten, und somit in
% simx2 gesicherten Simulationsergebnisse.
%
clear;

Flaeche = 1E-4;
weiter = 'j';
while weiter == 'j'
    clc;
    disp('');
    disp(' Graphische Darstellung der in pw_sim2 berechneten Daten');
    disp('');
    disp(' Welche Zustandsgroeszen sollen gezeichnet werden?');
    zust = input(' Lage/Feldgroeszen/Stellgroeszen [1/2/3] ');

    if zust == 1
        load simx2.mat;
        noch = 'j';
        while noch == 'j'
            knin = input(' Knotennr. = ');
            t = simx2(:,1);
            y = simx2(:,knin*4-2);
            plot(t,y);

```

```

grid
title('Zeitverlauf y');
xlabel('Zeit t [s]');
ylabel('Verschiebung y [m]');
pause
z = simx(:,knin*4-1);
plot(t,z);
grid
title('Zeitverlauf z');
xlabel('Zeit t [s]');
ylabel('Verschiebung z [m]');
pause
plot(z,y);
grid
title('Orbit im Knoten');
xlabel('Verschiebung z [m]');
ylabel('Verschiebung y [m]');
pause
disp('');
noch = input(' Eine weitere Ortskoordinate ? [j] ', 's');
if isempty(noch)
    noch = 'j';
end
end
end

if zust == 2
load sime2.mat;
noch = 'j';
while noch == 'j'
    knin = input(' Elektromagnetische Zustandsgroesze ? : ');
    H = sime2(:,knin+5);
    B = sime2(:,knin+1)/Flaechе;
plot(H,B);
grid
title(' Magnetische Groeszen');
xlabel('Feldstaerke [A/m]');
ylabel('Induktion [T]');
pause
disp('');
noch = input(' Eine weitere Elektromagnetische Zustandsgroesze ? [j] ', 's');
if isempty(noch)
    noch = 'j';
end
end
end

if zust == 3
load simu2.mat;
noch = 'j';
while noch == 'j'
    knin = input(' Elektrische Zustandsgroesze ? : ');
    t = simu2(:,1);

```

```

y = simu2(:,knin+1);
plot(t,y);
grid
xlabel('Zeit t [s]');
ylabel('u [V]');
title('Stellgroesze - u0');
pause
disp('');
noch = input(' Eine weitere Elektrische Zustandsgroesze ? [j] ', 's');
if isempty(noich)
    noch = 'j';
end
end
end

weiter = input(' Eine weitere Zustandsgroesze zeichnen ? [j] ', 's');
if isempty(weiter)
    weiter = 'j';
end
end
clear;
clc

```

A.2 Hilfsprogramme

```
%                               magtest
%
clc
magdat;

B = 0:.01:1.5;
Phi = A*B;
n = max(size(B));

for i = 1:n
    if abs(B(i)) <= Bkr
        f = A1*tan(A2*B(i));
        mu = (1/A1*A2)*(cos(A2*B(i)))^2;
    end
    if B(i) > Bkr
        f = A1*tan(A2*Bkr) + (B(i)-Bkr)/mucl;
        mu = mucl;
    end
    H(i) = f;
    M(i) = mu;
    Rloc(i) = (Ltot/mu + 2*g0/mu0)/A;
    KL(i) = 8*(Phi(i)*cos(alfa))^2/((A*mu0)^2*Rloc(i));
    KS(i) = 4*Phi(i)*cos(alfa)*N/(A*mu0*Rloc(i));
end

subplot(111);
plot(H,B)
grid
title('Ahysteretische Kurve');
xlabel('Feldstaerke [ A ]');
ylabel('Induktion [ T ]');
pause
clg

plot(B,M)
grid
title('Differentielle Permeabilitaet');
xlabel('Flusz Phi [ Vs ]');
ylabel(' [ Vs/Am ]');
pause
clg

plot(Phi,Rloc)
grid
title('Magnetischer Widerstand');
xlabel('Flusz Phi [ Vs ]');
ylabel('R_loc [ A/Vs ]');
keyboard
clg
```

```

plot(Phi,KL)
grid
title('Lagekoeffizient');
xlabel(' Flusz Phi [ Vs ]');
ylabel(' K_L [ N/m ]');
keyboard
clg

plot(Phi,KS)
grid
title('Stromkoeffizient');
xlabel(' Flusz Phi [ Vs ]');
ylabel(' K_S [ N/A ]');
pause;

function fun = hystest(t,y)
%
% Integrationsfunktion fuer das Hysteresemodell
%

magdat;
% Rampe mit Sinus

if t <= t1
    e = a*t/t1;           % V
end
if (t > t1)
    e = a + b*sin(o*(t-t1)); % V
end

Bp = (N*e - r*L*y(1))/(A*N^2);
bstr = pw_Bstr2(y(1),y(2),Bp);

fun(1,1) = Bp/bstr;
fun(2,1) = Bp;

%
%          magdat
% Datenfile fuer den elektromagnetischen Teil
A1 = 22.7971;    % A/m
A2 = 1.18639;   % T-1
A3 = -11.817;   % T-1
A4 = 0.44265;   % T-1
Bkr = 1.3;      % T
Bcl = 1.3;      % T
mu0 = 4*pi*1E-7; % Vs/Am
mucl = 3E-5;    % Vs/Am
B = 0.5;        % T
N = 570;
Kv = 5;         % V/V

```

```

Km = 5000;          % V/m
r = 10;            % Ohm
A = 1E-4;          % m2
Ltot = 0.1;        % m
g0 = 5E-4;         % m
alfa = 23*pi/180;
T = 1E-4;          % s
Kst = 5E8;
Cst = 1E2;
m = 12;
Phi = A*B;

%                dregler
%
% Dieses Programm generiert die Pole und Nullstellen
% eines Magnetlagers im z-Bereich und zeigt die WOK
% der Strecke mit und ohne Regler
%
clc
magdat;

if abs(B) <= Bkr
    f = A1*tan(A2*B);
    mu = (1/A1*A2)*(cos(A2*B))^2;
end
if B > Bkr
    f = A1*tan(A2*Bkr) + (B-Bkr)/mucl;
    mu = mucl;
end

Rloc = (Ltot/mu + 2*g0/mu0)/A;
Kl = 8*(Phi*cos(alfa))^2/((A*mu0)^2*Rloc);
Ks = 4*Phi*cos(alfa)*N/(A*mu0*Rloc);
u0 = r*Rloc*Phi/N;
ol = sqrt(Kl/m);

Kmk = Rloc/N^2;
omk = Rloc*r/N^2;

K = Kmk*Kv*Ks*Km/m;

on = sqrt(Kst/m);
zeta = 0.5;
od = sqrt(1-zeta^2)*on;
absz = exp(-zeta*on*T);
argz = T*od;
dom(1) = absz*(cos(argz) + sqrt(-1)*sin(argz));
dom(2) = absz*(cos(argz) - sqrt(-1)*sin(argz));

a = exp(-omk*T);
b = exp(ol*T);

```

```

c = exp(-ol*T);

z1 = [1 -1];
za = [1 -a];
zb = [1 -b];
zc = [1 -c];
zn = [1 1];
KD = -K*(1-a)*(1-b)*(1-c)/(8*omk*ol^2);

num = conv(zn,zn);
num = conv(num,zn);
den = conv(zc,zb);
den = conv(den,za);
dek = [1 0 0 0];
den = conv(den,dek);

disp('Strecke')
disp('  Streckenvertaerkung:');
disp(KD);
disp('  Zaehlerpolynom :');
disp(num);
disp('  Nennerpolynom :');
disp(den);
disp('  Wurzeln Zaehler :');
disp(roots(num));
disp('  Wurzeln Nenner:');
disp(roots(den));
disp('');
disp('Fortsetzung mit Taste');
pause

nul = roots(num);
t2 = zeros(max(size(nul)));
pol = [0;a;b;c];
t1 = [0; 0; 0; 0];

k = [0:1E-4:1E-3];
R = rlocus(num,den,k);

axis('square');
axis([-1.5,1.5,-1.5,1.5]);
plot(R, '.');
hold on
phi = [0:.1:2*pi+.2];
kreis = cos(phi) + sqrt(-1)*sin(phi);
plot(kreis, '.');
plot(pol,t1,'x');
plot(nul,t2,'o');
plot(dom,'x');
title('Wurzelortskurve ohne Regler');
ylabel('Imaginaerteil');
xlabel('Realteil');

```

```

hold off
pause

axis([.9,1.1,-.1,.1]);
plot(R, '.');
hold on
phi = [-.1:.005:.1];
kreis = cos(phi) + sqrt(-1)*sin(phi);
plot(kreis, '.');
plot(pol,t1,'x');
plot(nul,t2,'o');
plot(dom,'x');
title('Detail WOK ohne Regler');
ylabel('Imaginaerteil');
xlabel('Realteil');
hold off
pause

regpz = [1 0 0];
regpi = [1 -1];
regn1 = [1 -b];
regn2 = [1 -a];
regn3 = [1 -0.95];
regp = conv(regpi,regpz);
regn = conv(regn1,regn2);
regn = conv(regn,regn3);
denr = conv(den,regp);
numr = conv(num,regn);

Ko = (5*abs(dom(1))*abs(dom(1)-c)*abs(dom(1)-1))/(3*abs(dom(1)+1)*abs(dom(1)-0.95));
Kp = Ko/KD;
clc
disp('');
disp('Regler')
disp('  Zaehlerpolynom :');
disp(regn);
disp('  Nennerpolynom :');
disp(regp);
disp('Verstaerkung');
disp(Kp);
disp('');
disp('Fortsetzung mit Taste');
pause

k = [0:1E-3:5E-2];
R = rlocus(numr,denr,k);

clg
axis([-1.5,1.5,-1.5,1.5]);
plot(R, '.');
hold on

```

```

phi = [0:.1:2*pi+.2];
kreis = cos(phi) + sqrt(-1)*sin(phi);
plot(kreis, '.');
plot(pol,t1,'x');
plot(nul,t2,'o');
plot(dom,'x');
title('Wurzelortskurve mit Regler');
ylabel('Imaginaerteil');
xlabel('Realteil');
hold off
pause

axis([.9,1.1,-.1,.1]);
plot(R, '.');
hold on
phi = [-.1:.005:.1];
kreis = cos(phi) + sqrt(-1)*sin(phi);
plot(kreis, '.');
plot(pol,t1,'x');
plot(nul,t2,'o');
plot(dom,'x');
title('Detail WOK mit Regler');
ylabel('Imaginaerteil');
xlabel('Realteil');
hold off
pause
clc

%           rottest
%
%   rottest dient der Analyse eines Rotors
%   auf Magnetlagers.
%
pw_titel;
nun = pw_inp('      Was nun ?, sprach Zeus ',0);
while (nun == 1) | (nun == 2) | (nun == 3)
    if nun == 1 statrot; end
    if nun == 2 dynrot; end
    if nun == 3 kondtest; end
    pw_titel;
    nun = pw_inp('      Was nun ?, sprach Zeus ',0);
end
clc

%
%           pw_titel
clc
disp('');
disp('           ROTTEST');

```

```

disp('      Analyse eines Rotors auf Magnetlagern');
disp('');
disp('');
disp('      Es werden fuer die Analyse angeboten :');
disp('');
disp('      0 Aussteigen');
disp('      1 Statische Analyse ');
disp('      2 Dynamische Analyse');
disp('      3 Kondensation');
disp('');

%
%          statrot
%
% statrot dient der statischen Analyse eines Rotors
% auf Magnetlagern.
%
%
% Laden der Systemdaten

clc
name = pw_inp(' Welche Systemdaten sollen geladen werden ', 'sysdat0');
eval(name);
pw_trans;

% Generierung und Kondensation der Systemmatrizen

clc
disp('generiere Systemmatrizen');

[M,G,D,Der,K,Ker] = pw_matg(kn,knot,fe,lag,kup,rot);

% Berechnung der statischen Durchbiegung des natuerlichen
% und des kondensierten Systems.
% Diese muessen per definitionem identisch sein.

clc
disp(' Berechne statische Durchbiegung');

Fstat = pw_Fstat(kn,knot,fe,rot);
w = inv(Ker)*Fstat;

% Graphische Ausgabe der errechneten Werte unter Beruecksichtigung
% der Ansatzfunktionen und Berechnung der Randfaserspannungen

[y,z,be,ga] = pw_yzbg(w);
[t,v,w] = pw_form(1,y,z,be,ga);
[t,sigmay,sigmaz] = pw_stres(rot(2),d,l,y,z,be,ga);
clg
axis([0,max(t),min(v),max(v)]);
subplot(111);
plot(t,v,t,w);

```

```

grid;
title('Statische Durchbiegungen');
xlabel('Rotorlaenge [m]');
ylabel('Durchbiegung [m]');
keyboard
axis([0,max(t),min(sigmay),max(sigmay)]);
subplot(111);
plot(t,sigmay,t,sigmaz);
grid;
title('Randfaserspannung');
xlabel('Rotorlaenge [m]');
ylabel('sigma_x [N/m2]');
keyboard
clear ws
ws(:,1) = t';
ws(:,2) = v';
ws(:,3) = w';
ws(:,4) = sigmay';
ws(:,5) = sigmaz';

clc
disp('');
disp('Speichere Daten unter stat.mat ');
save stat.mat ws /ascii

%
%           dynrot
%
% dynrot dient der dynamischen Analyse eines Rotors
% auf Magnetlagern.
%

% Laden der Systemdaten

clc
name = pw_inp(' Welche Systemdaten sollen geladen werden ', 'sysdat0');
eval(name);
pw_trans;
omega = pw_inp(' Kreisfrequenz des Rotors ', 100);

% Generierung und Kondensation der Systemmatrizen

clc
disp('generiere Systemmatrizen');

[M,G,D,Der,K,Ker] = pw_matg(kn,knot,fe,lag,kup,rot);
GD = G*omega+Der;

% Transformation auf Zustandsform,
% um die Eigenwerte, respektive die Eigenformen zu bestimmen
% sowohl fuer das natuerliche als auch fuer das kondensierte System

```

```

clc
disp('Berechne Zustandsmatrizen');

[A,nix] = pw_zust(M,GD,Ker);

clc
disp('Berechne Eigenwerte');

[ev ew] = eig(A);

% Eigenwerte des Modells.

[ew index] = sort(diag(ew));      % Umsortieren der Eigenwerte
evh = ev; clear ev;
for i = 1:8*kn                    % Umsortieren der Eigenvektoren
    ev(:,i) = evh(:,index(i));
end
ews(:,1) = real(ew);
ews(:,2) = imag(ew);

clc
disp('');
disp('Speichere Eigenwerte unter eigw.mat');
save eigw.mat ews /ascii

% Es werden nur die ersten 12 Eigenwerte gezeigt

for i = 1:3
    e(1:8) = ew(i*8-7:i*8);
    e1= e(1:2);
    e2 = e(3:4);
    e3 = e(5:6);
    e4 = e(7:8);

    clg;
    subplot(221);
    plot(real(e1),imag(e1),'x');
    xlabel('Realteil');
    ylabel('Imaginaerteil');
    grid
    subplot(222);
    plot(real(e2),imag(e2),'x');
    xlabel('Realteil');
    ylabel('Imaginaerteil');
    grid
    subplot(223);
    plot(real(e3),imag(e3),'x');
    xlabel('Realteil');
    ylabel('Imaginaerteil');
    grid
    subplot(224);
    plot(real(e4),imag(e4),'x');

```

```

xlabel('Realteil');
ylabel('Imaginaerteil');
grid
subplot(111)
titel = sprintf('Die %g-ten 8 Eigenwerte ',i);
title(titel);
pause
end

clc
ne = 4*kn;
k = 0;
weiter = 'j';
while (weiter == 'j') & (k <= ne)
k = k+1;
str = sprintf(' Berechne %g-te Eigenform',k);
disp(str);
om = abs(imag(ew(2*k)));
de = real(ew(2*k));
evp = ev(1:ne,2*k);

% Berechnung der realen Durchbiegungen

arc1 = [0 pi/2 pi 3*pi/2];
arc2 = arc1*de/om;
for j = 1:4
xp = 2*exp(arc2(j))*(real(evp)*cos(arc1(j)) - imag(evp)*sin(arc1(j)));
[y,z,b,g] = pw_yzbg(xp);
[t,vh,wh] = pw_form(1,y,z,b,g);
v(j,:) = vh;
w(j,:) = wh;
end

clg;
norm = max(max(v));
v = v/norm;
axis([0,max(t),-1,1]);
subplot(111);
plot(t,v);
grid;
xlabel('Rotorlaenge [m]');
ylabel('Normierte Auslenkung [m]');
str = sprintf('Eigenfrequenz = %g [rad/s], Daempfung = %g', ...
imag(ew(2*k)),-real(ew(2*k))/imag(ew(2*k)));
title(str);
keyboard
clc
weiter = pw_inp(' Weitere Eigenform berechnen', 'j');
end

```

```

%           kondtest
%
% kondtest dient dem Vergleich der Eigenwerte des kondensierten
% Modells mit denen des urspruenglichen.
%

% Laden der Systemdaten

clc
name = pw_inp(' Welche Systemdaten sollen geladen werden ', 'sysdat0');
eval(name);
pw_trans;

% Generierung und Kondensation der Systemmatrizen

clc
disp('generiere Systemmatrizen');

[M,G,D,Der,K,Ker] = pw_matg(kn,knot,fe,lag,kup,rot);

wahl = pw_inp(' Modale (m) oder statische (s) Kondensation ', 'm');
if wahl == 's'
    [Mst,Gst,Dst,Kst,R] = pw_skond(M,G,Der,Ker,Ker,kond1,kond2);
end
if wahl == 'm'
    [Mst,Gst,Dst,Kst,R] = pw_mkond(M,G,Der,Ker,Ker,grenzfr);
end

% Berechnung der statischen Durchbiegung des natuerlichen
% und des kondensierten Systems.
% Diese muessen per definitionem identisch sein.

clc
disp(' Berechne statische Durchbiegung');

Fstat = pw_Fstat(kn,knot,fe,rot);
w = inv(Ker)*Fstat;
wst = R*inv(Kst)*R'*Fstat;

% Graphische Ausgabe der errechneten Werte unter Beruecksichtigung
% der Ansatzfunktionen

[y,z,be,ga] = pw_yzbg(w);
[yst,zst,best,gast] = pw_yzbg(wst);
[t,v,w] = pw_form(l,y,z,be,ga);
[t,vst,wst] = pw_form(l,yst,zst,best,gast);
clg

axis([0,max(t),min(v),max(v)]);
subplot(111);
plot(t,v,t,w);
grid;

```

```

title(' Statische Durchbiegungen ');
xlabel('Rotorlaenge [m]');
ylabel('Durchbiegung [m]');
keyboard

subplot(111);
if wahl == 'm'
    str = 'Differenz fuer modale Kondensation';
end
axis([0,max(t),min(real(v-vst)),max(real(v-vst))]);
plot(t,real(v-vst),t,real(w-wst));
grid;
title(str);
xlabel('Rotorlaenge [m]');
ylabel('Durchbiegung [m]');
keyboard

% Transformation auf Zustandsform,
% um die Eigenwerte, respektive die Eigenformen zu bestimmen
% sowohl fuer das natuerliche als auch fuer das kondensierte System

clc
disp('Berechne Zustandsmatrizen');

[A,nix] = pw_zust(M,Der,Ker);
[Ast,nix] = pw_zust(Mst,Dst,Kst);

clc
disp('Berechne Eigenwerte');

ew = eig(A);
ewst = eig(Ast);

% Vergleich der Eigenwerte des natuerlichen mit denen
% des kondensiertesn Modells.

nest = max(size(ewst));
ew = sort(ew);          % Umsortieren der Eigenwerte
ewh = sort(ewst);
ewst = zeros(8*kn,1);
ewst(1:nest) = ewh;

% Es werden nur die ersten 12 Eigenwerte gezeigt

for i = 1:3
    e(1:8) = ew(i*8-7:i*8);
    est(1:8) = ewst(i*8-7:i*8);
    e1= e(1:2);
    est1 = est(1:2);
    e2 = e(3:4);
    est2 = est(3:4);
    e3 = e(5:6);

```

```

est3 = est(5:6);
e4 = e(7:8);
est4 = est(7:8);

clg;
subplot(221);
plot(real(e1),imag(e1),'+',real(est1),imag(est1),'x');
xlabel('Realteil');
ylabel('Imaginaerteil');
grid
subplot(222);
plot(real(e2),imag(e2),'+',real(est2),imag(est2),'x');
xlabel('Realteil');
ylabel('Imaginaerteil');
grid
subplot(223);
plot(real(e3),imag(e3),'+',real(est3),imag(est3),'x');
xlabel('Realteil');
ylabel('Imaginaerteil');
grid
subplot(224);
plot(real(e4),imag(e4),'+',real(est4),imag(est4),'x');
xlabel('Realteil');
ylabel('Imaginaerteil');
grid
subplot(111)
titel = sprintf('Vergleich der %g-ten 8 Eigenwerte real(+), kond(x)',i);
title(titel);
pause;
end
clc

```

```

function [Ml, Gl, Dl, Del, Kl, Kel] = pw_matg(knl,knotl,fel,lagl,kupl,rotl)
%
% [Ml, Gl, Dl, Del, Kl, Kel] = pw_matg(knl,knotl,fel,lagl,kupl,rotl)
%
% generiert die fuer die Biegeschwingungen relevanten Systemmatrizen
% eines Rotors aus den Eingangsmatrizen.
%
% knotl Knotendaten
% fel Daten der FE
% lagl Lagerdaten
% kupl Kupplungsdaten
% rotl Materialdaten
%
% knl Knotenanzahl
%
% Mr Massenmatrix der Scheiben
% Mw Massenmatrix der Wellenabschnitten
% Ml Massenmatrix des Rotors

```

```

% Klag  Steifigkeitsmatrix der Lager
% Kkup  Steifigkeitsmatrix der Kupplung
% Kw    Steifigkeitsmatrix der Wellenabschnitte
% Kl    Steifigkeitsmatrix des Rotors ohne Lager
% Kel   Ersatzsteifigkeitsmatrix des Rotors

% Dlag  Daempfungsmatrix der Lager
% Dkup  Daempfungsmatrix der Kupplung
% Dl    Daempfungsmatrix des Rotors ohne Lager
% Del   Ersatzdaempfungsmatrix des Rotors

% Gl    Gyroskopiematrix

[nl nix] = size(lagl);
% Berechnung der Matrizen

for i = 1:kn1
    Mr(4*i - 3,4*i - 3) = knotl(i,1);
    Mr(4*i - 2,4*i - 2) = knotl(i,1);
    Mr(4*i - 1,4*i - 1) = knotl(i,3);
    Mr(4*i,4*i)         = knotl(i,3);
end

% Initialisierung der Matrizen

Kkup = zeros(4*kn1,4*kn1);      % Steifigkeitsmatrizen
Klag = zeros(4*kn1,4*kn1);
Dlag = zeros(4*kn1,4*kn1);      % Daempfungsmatrizen
Dkup = zeros(4*kn1,4*kn1);
Gl   = zeros(4*kn1,4*kn1);      % Gyroskopiematrix
for i = 1:4
    for j = 1:nl
        Klag(4*lagl(j,9)-4+i,4*lagl(j,9)-4+i) = lagl(j,i);
        Dlag(4*lagl(j,9)-4+i,4*lagl(j,9)-4+i) = lagl(j,i+4);
    end
    Kkup(4*kupl(9)-4+i,4*kupl(9)-4+i) = kupl(i);
    Dkup(4*kupl(9)-4+i,4*kupl(9)-4+i) = kupl(i+4);
end

for i = 1:kn1
    Gl(4*i,4*i - 1) = -knotl(i,2);
    Gl(4*i - 1,4*i) = knotl(i,2);
end

Mw = zeros(4*kn1,4*kn1);
Kw = zeros(4*kn1,4*kn1);
k = 1;
rho = rotl(1);
E = rotl(2);
for k = 1:kn1-1
    dlok = fel(k,2);

```

```

llok = fel(k,1);
Mw(4*k-3:4*k+4,4*k-3:4*k+4) = Mw(4*k-3:4*k+4,4*k-3:4*k+4)..
    + pw_fema(dlok,llok,rho);
Kw(4*k-3:4*k+4,4*k-3:4*k+4) = Kw(4*k-3:4*k+4,4*k-3:4*k+4)..
    + pw_fest(dlok,llok,E);
end

Ml = Mw + Mr;
Kl = Kkup + Kw;
Kel = Klag + Kl;
Del = Dlag + Dkup;
Dl = Dkup;

%Ende

%
%   pw_trans erzeugt aus den Knotendaten die Vektordaten
%

[knd nix] = size(knot);
if knd < kn
    knot(kn,:) = zeros(1,5);
end

mkn = knot(:,1)';    % Vektor der Massen
Jp  = knot(:,2)';    % Vektor der polaren MTRM
Ja  = knot(:,3)';    % Vektor der axialen MTRM
ex  = knot(:,4)';    % Vektor der Schwerpunktsexzentrizitaeten
phi = knot(:,5)';    % Vektor der Phasenwinkel der Exzentrizitaeten
l   = fe(:,1)';      % Vektor der Laengen der FE
d   = fe(:,2)';      % Vektor der Durchmesser der FE
lagn = lag(:,9);     % Vektor, der die Lager adressiert

db = 2*pi/p;
for i = 1:p
    betak(i) = (i-1)*db; % Vektor der Teilungswinkel am Magnetlager
end

function mwelle = pw_fema(dl,ll,rhol)
%
%           mwelle = pw_fema(dl,ll,rhol)
%
%   pw_fema erzeugt die 8*8 Massen-Matrix mwelle eines
%   finiten Wellenelements des Beroulli-Euler Typs bei gegebener
%   Dichte rho, Laenge l und Durchmesser d.
%

m = rhol*ll*d1*d1*pi/4;    % Masse des Wellenabschnittes

```

```
% Berechnung der Massenmatrix eines FE
```

```
H(1,1) = m*156;  
H(2,2) = m*156;  
H(5,5) = m*156;  
H(6,6) = m*156;  
H(3,3) = m*11*11*4;  
H(4,4) = m*11*11*4;  
H(7,7) = m*11*11*4;  
H(8,8) = m*11*11*4;  
H(2,3) = -m*11*22;  
H(1,4) = m*11*22;  
H(6,7) = m*11*22;  
H(5,8) = -m*11*22;  
H(1,5) = m*54;  
H(2,6) = m*54;  
H(3,7) = -m*11*11*3;  
H(4,8) = -m*11*11*3;  
H(4,5) = m*11*13;  
H(3,6) = -m*11*13;  
H(2,7) = m*11*13;  
H(1,8) = -m*11*13;
```

```
mwelle = (H + H' - diag(diag(H)))/420;
```

```
function kwelle = pw_fest(dl,ll,EI)
```

```
%
```

```
%           kwelle = pw_fest(dl,ll,EI)
```

```
%
```

```
% pw_fest erzeugt die 8*8 Steifigkeitsmatrix kwelle
```

```
% eines Wellenabschnittes des Bernoulli-Euler Typs bei gegebenem
```

```
% E-Modul E, der Länge l und des Durchmessers d.
```

```
%
```

```
EI = EI*(dl^4)*pi/64; % Steifigkeit des Wellenabschnittes
```

```
% Berechnung der Steifigkeitsmatrix eines FE
```

```
H(1,1) = 12*EI/11^3;  
H(2,2) = 12*EI/11^3;  
H(5,5) = 12*EI/11^3;  
H(6,6) = 12*EI/11^3;  
H(1,5) = -12*EI/11^3;  
H(2,6) = -12*EI/11^3;  
H(3,3) = 4*EI/11;  
H(4,4) = 4*EI/11;  
H(7,7) = 4*EI/11;  
H(8,8) = 4*EI/11;  
H(2,3) = -6*EI/11^2;  
H(1,4) = 6*EI/11^2;  
H(6,7) = 6*EI/11^2;
```

```

H(5,8) = -6*EI/11^2;
H(4,5) = -6*EI/11^2;
H(3,6) = 6*EI/11^2;
H(2,7) = -6*EI/11^2;
H(1,8) = 6*EI/11^2;
H(3,7) = 2*EI/11;
H(4,8) = 2*EI/11;

kwelle = H + H' - diag(diag(H));

function Fst = pw_Fstat(knl,knotl,fel,rotl)
%
%       Fst = pw_Fstat(knl,knotl,fel,rotl)
%
%   berechnet den statischen Stoervektor
%
Fst = zeros(4*knl,1);

[knd nix] = size(knotl);
if knd < knl
    knotl(knl,:) = zeros(1,5);
end
mk = knotl(:,1);

for i = 1:knl-1
    mfe(i) = rotl(1)*fel(i,1)*(fel(i,2)^2)*pi/4;
end

m(1) = mfe(1)/2;
for i = 2:knl-1
    m(i) = (mfe(i-1)+mfe(i))/2;
end
m(knl) = mfe(knl-1)/2;

for i = 1:knl
    Fst(4*i-3,1) = -(mk(i) + m(i))*rotl(3);
end

```

```

function [Asternl, invMsternl] = pw_zust(Msternl,GDsternl,Ksternl)
%
%       [Astern, invMstern] = pw_zust(M,GD,K)
%
%   erstellt die Zustandsmatrix eines dynamischen Systems der Form
%       zd = A z + Fzust
%   aus dem Differentialgleichungssystem
%       M xdd + GD xd + K x = F
%   wobei M,GD,K,A die Systemmatrizen beschreiben und z,zd den
%   Zustandsvektor, respektive dessen erste Ableitung und

```

```

% F den Stoervektor.
%
[m,n] = size(Msternl);
if (m == n)
    invMsternl = inv(Msternl);
    Asternl = zeros(2*m,2*m);
    Asternl(m+1:2*m,1:m) = -invMsternl*Ksternl;
    Asternl(1:m,m+1:2*m) = eye(m);
    Asternl(m+1:2*m,m+1:2*m) = -invMsternl*GDsternl;
% disp(' Speichere berechnete Matrizen unter zust.mat');
% save zust;
end

```

```

function [Msternl, Gsternl, Dsternl, Ksternl, Rl] = pw_mkond(Ml,Gl,Dl,Kl,Kel,grenzfr)
%
% [Msternl, Gsternl, Dsternl, Ksternl, Rl] = pw_mkond(Ml,Gl,Dl,Kl,Kel,grenzfr)
%
% Modale Kondensation
% pw_mkond kondensiert ein Modell mit n Freiheitsgraden auf ein Modell mit
% pk Freiheitsgraden, wobei m die aus der Grenzfrequenz bestimmt wird,
% und die Ersatzsteifigkeitsmatrix zur Kondensation heran-
% gezogen wird.
%

```

```

[n nix] = size(Ml);
[ev ew] = eig(inv(Ml)*Kel);

[ew index] = sort(diag(ew)); % Umsortieren der Eigenwerte
evh = ev; clear ev;
for i = 1:n % Umsortieren der Eigenvektoren
    ev(:,i) = evh(:,index(i));
end

```

```

pk = 1;
while (abs(ew(pk)) < grenzfr^2) & pk<n
    pk = pk+1;
end
pk = pk-1;

```

```

Rl = ev(:,1:pk);

```

```

Rtl = Rl';
Msternl = Rtl*Ml*Rl; %
Gsternl = Rtl*Gl*Rl; % kondensierte Ausgangsmatrizen
Dsternl = Rtl*Dl*Rl; %
Ksternl = Rtl*Kl*Rl; %

```

```

function [Msternl, Gsternl, Dsternl, Ksternl, Rl] = pw_skond(Ml,Gl,Dl,Kl,Kel,kond1,kond2)
%

```

```

% [Mstern1, Gstern1, Dstern1, Kstern1, R1] = pw_skond(M1,G1,D1,K1,Kel,kond1,kond2)
%
%           Statische Kondensation
% pw_skond kondensiert ein Modell mit n Freiheitsgraden auf ein Modell mit
% n-m Freiheitsgraden, wobei m die Anzahl der uninteressanten DOF
% darstellt und die Ersatzsteifigkeitsmatrix zur Kondensation heran-
% gezogen wird.
%
[n nix] = size(M1);
kn = floor(n/4);

kn1 = zeros(1,kn);           % Vektor, der die uninteressanten Knoten markiert
kn2 = zeros(1,kn);           % Vektor, der die uninteressanten Knotenverdrehungen
                             % markiert

if isempty(kond1)
    n1 = 0;
else
    n1 = max(size(kond1));
    for i = 1:n1
        kn1(kond1(i)) = 1;           % Der uninteressanten Knotennummer
    end                               % wird logisch 1 zugeordnet
end

if isempty(kond2)
    n2 = 0;
else
    n2 = max(size(kond2));
    for i = 1:n2
        kn2(kond2(i)) = 1;           % Dem Knoten mit den uninteressanten
    end                               % Verdrehungen ebenfalls logisch 1
end

kn3 = ones(1,kn) - (kn1 + kn2);     % Vektor, der die interessanten Knoten markiert

nn = 4*n1 + 2*n2;                 % Anzahl der 'masters'
nv = n - nn;                       % Anzahl der 'slaves'
T = zeros(n,n);

% Generierung der Transformationsmatrix

% Verschiebung der 4 Knoten-Freiheitsgrade
w1 = 0;
w2 = nv;
for i = 1:kn
    if (kn3(i) == 1)
        for j = 1:4
            T(w1+j,4*i-4+j) = 1;
        end
        w1 = w1 + 4;
    end
    if (kn1(i) ~= 0)

```

```

    for j = 1:4
        T(w2+j,4*i-4+j) = 1;
    end
    w2 = w2 + 4;
end
end

% Verschiebung der uninteressanten Verdrehungen
for i = 1:kn
    if (kn2(i) ~= 0)
        for j = 1:2
            T(w1+j,4*i-4+j) = 1;
        end
        w1 = w1 + 2;
        for j = 1:2
            T(w2+j,4*i-2+j) = 1;
        end
        w2 = w2 + 2;
    end
end

if ((n1 > 0) | (n2 > 0))
    Kstrichl = T*Kel*T'; % K' im Skriptum
    Knn = Kstrichl(nv+1:n,nv+1:n);
    Knv = Kstrichl(nv+1:n,1:nv);
    Rl = zeros(n,nv); % Reduktionsmatrix
    H = zeros(n,nv);
    H(1:nv,1:nv) = eye(nv);
    H(nv+1:n,1:nv) = -inv(Knn)*Knv;
    Rl = T'*H;
    Rtl = Rl';
    Msternl = Rtl*Ml*Rl; %
    Gsternl = Rtl*Gl*Rl; % kondensierte Ausgangsmatrizen
    Dsternl = Rtl*Dl*Rl; %
    Ksternl = Rtl*Kl*Rl; %
else
    Rtl = eye(Ml);
    Msternl = Ml;
    Gsternl = Gl;
    Dsternl = Dl;
    Ksternl = Kl;
end
%disp(' Speichere berechnete Matrizen unter kon.mat');
%save kond2;

%
%          pw_ew
%
% pw_ew dient der Eigenwertanalyse in Abhaengigkeit der
% Lagerparameter

```

```

clc
disp('    Eigenwertanalyse');
disp('');

name = pw_inp(' Welche Systemdaten sollen geladen werden ', 'sysdat0');
eval(name);
pw_trans;

omega = pw_inp(' Drehfrequenz des Rotors ', 100);
wahl = input(' Welcher Lagerparameter soll variiert werden ? (d/k) ', 's');

clc;
imax = 3;
[nix kn] = size(mkn);

if wahl == 'd'
delete ewd.mat;
dg = lag(1,5);
for i = -imax:24*imax
disp(i);
dd = dg*(1+0.9*(i/imax)^3);
lag(1,5) = dd;
lag(1,6) = dd;
lag(2,5) = dd;
lag(2,6) = dd;
[M,G,D,Der,K,Ker] = pw_matg(kn,knot,fe,lag,kup,rot);
GD = G*omega+Der;
[A,nix] = pw_zust(M,GD,Ker);
evec = eig(A);
kvprintf('ewd.mat',evec(2:2:12));
end
end

if wahl == 'k'
delete ewk.mat;
eval(name);
pw_trans;
kg = lag(1,1);
for i = -imax:9*imax
disp(i);
kd = kg*(1+0.9*(i/imax));
lag(1,1) = kd;
lag(2,1) = kd;
lag(1,2) = kd;
lag(2,2) = kd;
[M,G,D,Der,K,Ker] = pw_matg(kn,knot,fe,lag,kup,rot);
GD = G*omega+Der;
[A,nix] = pw_zust(M,GD,Ker);
evec = eig(A);
kvprintf('ewk.mat',evec(2:2:12));
end
end

```

```

%
% pw_ewo dient der Berechnung der Eigenwerte und der Unwuchtloesung
% bei Variation der Drehzahl Omega
%

% Laden der Systemdaten

clc
name = pw_inp(' Welche Systemdaten sollen geladen werden ','sysdat0');
eval(name);
pw_trans;
anz = kn;
delete ewo.mat;
delete evo.mat;

omega1 = pw_inp(' Untere Grenze der Kreisfrequenz des Rotors ',10);
omega2 = pw_inp(' Obere Grenze der Kreisfrequenz des Rotors ',1000);
iter = pw_inp(' Anzahl der Intervalle ',10);

% Generierung und Kondensation der Systemmatrizen

clc
disp('generiere Systemmatrizen');

[M,G,D,Der,K,Ker] = pw_matg(kn,knot,fe,lag,kup,rot);

% Transformation auf Zustandsform,
% um die Eigenwerte, respektive die Eigenformen zu bestimmen
% sowohl fuer das natuerliche als auch fuer das kondensierte System

dom = (omega2-omega1)/(iter-1);

for j = 1:iter
    clc
    disp(j);
    omega = omega1 + (j-1)*dom;
    GD = G*omega+Der;
    [A,nix] = pw_zust(M,GD,Ker);
    ew = eig(A);
    ew = sort(ew);

    % Bestimmung der zu den Eigenformen gehoerenden Amplituden
    f1 = zeros(4*kn,1);
    f2 = zeros(4*kn,1);
    for i = 1:kn
        Amp = mkn(i)*ex(i)*omega^2;
        f1(4*i-3,1) = Amp*sin(phi(i));
        f2(4*i-3,1) = Amp*cos(phi(i));
        f1(4*i-2,1) = Amp*cos(phi(i));
        f2(4*i-2,1) = -Amp*sin(phi(i));
    end
    dim = 4*kn;

```

```

A(1:dim,1:dim) = Ker-M*omega^2;
A(dim+1:2*dim,1:dim) = -GD*omega;
A(1:dim,dim+1:2*dim) = GD*omega;
A(dim+1:2*dim,dim+1:2*dim) = Ker-M*omega^2;
x = inv(A)*[f1;f2];
x1 = x(1:dim);
[y,z,be,ga] = pw_yzbg(x1);
vo = sqrt(y.*y + z.*z);
wo = [omega; imag(ew(2:2:24))];
rvprintf('ewo.mat',wo);
rvprintf('evo.mat',vo);

end

%
%           pw_ewpl
%
% pw_ewpl zeichnet die berechneten Eigenwerte, die in ew.mat
% gesichert worden sind.

clc
name = pw_inp('Laden von Datei (ewd/ewk/ewo) ? ','ewo');

if name == 'ewo'
    lade = 'load ewo.mat';
    eval(lade);
    lade = 'load evo.mat';
    eval(lade);
    o = ewo(:,1);
    axis([0,1500,0,1500]);
    plot(o,ewo(:,1:11));
    grid;
    title('Die ersten zehn Biegeeigenfrequenzen');
    xlabel('Drehfrequenz [rad/s]');
    ylabel('Eigenfrequenzen [rad/s]');
    keyboard
    mesh(evo');
    %title('Unwuchtloesungen ueber Drehzahl');
    keyboard
    pr = [2 7 11 15 20 22];
    for i = 1:6
        evon(:,i) = evo(:,pr(i));
    end
    axis([0,1500,0,5E-4]);
    plot(o,evon);
    grid
    title('Stationaere Unwuchtschwingungen');
    xlabel('Drehfrequenz [rad/s]');
    ylabel('Auslenkungen in den Knoten [m]');
    pause

```

```

else

lade = ['load ',name, '.mat'];
eval(lade);
kopie = ['ew = ',name, ';'];
eval(kopie);
[m n] = size(ew);
for i = 1:floor(n/2)
    ewp = ew(:,2*i-1) +sqrt(-1)*ew(:,2*i);
    titel = sprintf(' Variation des %g-ten Eigenwertes',i);
    plot(ewp);
    title(titel);
    xlabel('Realteil');
    ylabel('Imaginaerteil');
end

end
clc
disp(' That's it');

```

```

function [yl, zl, bl, gl] = pw_yzbg(vl)
%
%      [yl, zl, bl, gl] = pw_yzbg(vl)
%  liest die Werte y, z, beta und gamma aus dem Zustandsvektor
%

```

```

[anz nix] = size(vl);
anz = floor(anz/4);
for i = 1:anz
    yl(i) = vl(4*i-3);
    zl(i) = vl(4*i-2);
    bl(i) = vl(4*i-1);
    gl(i) = vl(4*i);
end

```

```

function [tl, vl, wl] = pw_form(l1,yl,zl,bel,gal)
%
%      [tl, vl, wl] = pw_form(l1,yl,zl,bel,gal)
%
%  pw_form.m erzeugt die Biegelinien vl,bzw wl eines Rotors aus
%  den Durchbiegungen yl,bzw zl und Verdrehungen gal,bzw bel an
%  den Stuetzpunkten mittels Ansatzfunktionen ( Hermite Polynome ).
%

```

```

int = max(size(l1));
fein = floor(120/int);          % abgestimmt auf Bildschirmausgabe

```

```

ld = 0;
tl(1) = 0;

```

```

v1(1) = yl(1);
w1(1) = zl(1);

% Berechnung der Biegelinien und des Abszissenvektors t1
% h1(xi),h2(xi),h3(xi),h4(xi) : Hermite-Polynome

for i = 1:int
  for j = 1:fein
    xi = j/fein;
    h1 = 1-3*xi^2+2*xi^3;
    h2 = xi*(1-xi)^2*ll(i);
    h3 = 3*xi^2-2*xi^3;
    h4 = -(xi^2)*(1-xi)*ll(i);
    t1((i-1)*fein+j+1) = ld + j*ll(i)/fein;
    v1((i-1)*fein+j+1) = h1*yl(i) + h2*gal(i) + h3*yl(i+1) + h4*gal(i+1);
    w1((i-1)*fein+j+1) = h1*zl(i) - h2*bel(i) + h3*zl(i+1) - h4*bel(i+1);
  end
  ld = ld + ll(i);
end

% Ende

```

```

function [t1, sigmayl, sigmazl] = pw_stres(E1,d1,ll,yl,zl,bel,gal)
%
%       [t1, sigmal] = pw_stres(E1,d1,ll,yl,zl,bel,gal)
%
% pw_stres.m erzeugt die die Randfaserspannung sigmal eines Rotors
% aus den Durchbiegungen yl,bzw zl und Verdrehungen gal,bzw bel an
% den Stuetzpunkten mittels Ansatzfunktionen ( Hermite Polynome ).
%

```

```

int = max(size(ll));
fein = floor(120/int);          % abgestimmt auf Bildschirmausgabe

ld = 0;
t1(1) = 0;
v2str = (-6*yl(1) - 4*ll(1)*gal(1) + 6*yl(2) - 2*ll(1)*gal(2))/ll(1)^2;
w2str = (-6*zl(1) + 4*ll(1)*bel(1) + 6*zl(2) + 2*ll(1)*bel(2))/ll(1)^2;
sigmayl(1) = E1*d1(1)*v2str/2;
sigmazl(1) = E1*d1(1)*w2str/2;

% Berechnung der Biegelinien und des Abszissenvektors t1
% h1(xi),h2(xi),h3(xi),h4(xi) : Zweite Ableitungen der
% Hermite-Polynome

```

```

for i = 1:int
  for j = 1:fein
    xi = j/fein;
    h1 = (-6 + 12*xi)/ll(i)^2;
    h2 = (-4 + 6*xi)/ll(i);

```

```

h3 = (6 - 12*xi)/ll(i)^2;
h4 = (-2 + 6*xi)/ll(i);
t1((i-1)*fein+j+1) = ld + j*ll(i)/fein;
v2str = h1*yl(i) + h2*gal(i) + h3*yl(i+1) + h4*gal(i+1);
w2str = h1*zl(i) - h2*bel(i) + h3*zl(i+1) - h4*bel(i+1);
sigmayl((i-1)*fein+j+1) = E1*d1(i)*v2str/2;
sigmazl((i-1)*fein+j+1) = E1*d1(i)*w2str/2;
end
ld = ld + ll(i);
end

% Ende

```

```

function inp = pw_inp(string,default);
%
%       pw_inp(string,default)
%
%       ist ein Inputfile, welches in den Namen inp auf die Abfrage
%       string mit dem Defaultwert default einliest
%

```

```

if isstr(default)
    str = [string,' [' ,default,'] : '];
else
    dstr = sprintf(' [%g] : ',default);
    str = [string,dstr];
end
h = input(str,'s');
if isempty(h)
    inp = default;
else
    if isstr(default)
        inp = h;
    else
        inp = eval(h);
    end
end
end

```

```

function kvprintf(name,vec)
%
%       kvprintf fuegt einen den komplexen Vektor vec dem
%       file name als weitere Zeile hinzu, und zwar ab-
%       wechselnd Real und Imaginaerteil
%
[n1 n2] = size(vec);
if n1 == 1
    n = n2;
else
    n = n1;
end

```

```

end;
str = ' ';
for i = 1:n
    str = [str sprintf(' %e %e',real(vec(i)),imag(vec(i)))];
end
str = [str sprintf(' \n')];
fprintf(name,str);

```

```

function rvprintf(name,vec)
%
%     rvprintf fuegt einen den reellen Vektor vec dem
%     file name als weitere Zeile hinzu
%
[n1 n2] = size(vec);
if n1 == 1
    n = n2;
else
    n = n1;
end;
str = ' ';
for i = 1:n
    str = [str sprintf(' %e',vec(i))];
end
str = [str sprintf(' \n')];
fprintf(name,str);

```

æ