

INTELLIGENTE REGELUNG VON NICHTLINEAREN ELEKTROMECHANISCHEN SYSTEMEN

Dissertation

zur Erlangung des Akademischen Grades

Doktoringenieur (Dr. –Ing.)

Von M. Sc. Tsepkovskiy Yuriy
Geb. am 8 Mai 1980 in Charkiw, Ukraine

genehmigt durch die Fakultät für Elektrotechnik und Informationstechnik
der Otto-von-Guericke-Universität Magdeburg

Gutachter:
Univ.-Prof. Dr.-Ing. habil. Frank Palis
Dr. Sc.techn. Ulrich Schmucker
Prof. Dr.sc. Viktor Kalashnikov

Promotionskolloquium am 16.06.2011

Abstract

Die Entwicklung der Technik in den letzten Jahrzehnten, besonders bei Mechatroniksystemen, die Erhöhung der Forderungen nach Genauigkeit, Sicherheit, Einfachheit der Abstimmung und Verwendung stellt neue Aufgaben an die Regelungstheorie. In den letzten Jahren werden immer mehr Roboter und komplexe mechanische Systeme in Betrieb genommen. Außerdem entsteht eine große Anzahl neuer nichtlinearer komplexer Regelstrecken, in denen sich schnell verändernde und schwierig messbare Parameter und starke Wechselwirkung zwischen allen Kanälen der Regelung auftreten. Derartige Regelstrecken lassen sich nicht mehr ausreichend mit fest eingestellten Reglern beherrschen.

Adaptive Regelungen haben hier einen Vorteil gegenüber konventionellen, fest eingestellten Reglern. Während letztere ein optimales Systemverhalten nur für einen bestimmten Arbeitspunkt oder einen bestimmten Arbeitsbereich erreichen, können selbst anpassende Regler dies über den gesamten Arbeitsbereich erzielen. Die zunächst wegen der hohen Realisierungskosten nur für militärische Zwecke eingesetzte adaptive Regelung gewinnt sowohl in der Theorie als auch in Praxis immer mehr an Bedeutung. Heute können verschiedenartige adaptive Regelverfahren mittels Mikrorechnern und mathematischen Programmen wie Matlab, Mathematika und Maple erfolgreich auf vielen Anwendungsgebieten eingesetzt werden.

In dieser Arbeit werden folgende Regelstrecken aus einer Menge von Regelungssystemen entwickelt, beschrieben und erforscht:

- Der Bewegungsantrieb eines Unterwasserapparates und das Steuerungssystem für den gesamten Unterwasserapparat,
- Magnetlager mit einem und mehreren Freiheitsgraden,
- Die Hand eines mobilen anthropomorphen Roboters mit sieben Freiheitsgraden und Greifer, der fünf bis zwanzig Freiheitsgrade besitzt.

Diese Objekte sind komplizierte nichtlineare Systeme mit unbekanntem und sich schnell verändernden Parametern [1-3].

Die Bewegungssteuerung des autonomen Unterwasserapparates (UWA) dient hauptsächlich der Erhöhung der dynamischen Positionsgenauigkeit bei Bewegungen mit hohen Geschwindigkeiten auf komplizierten räumlichen Bahnen.

Der gesondert behandelte Bewegungsantrieb des UWAs ist im Wesentlichen ein nichtlineares Objekt mit unbekanntem und sich schnell ändernden Parametern [4-6]. Für die Realisierung der Regelungssysteme ähnlicher Objekte in bekannten Systemen ist die Messung der unbekanntem Größen und der Parameter erforderlich. Jedoch ist eine Identifizierung und Berücksichtigung der Parameter während der Regelung schwierig umsetzbar.

Es verändern sich im Laufe der Bewegung gleichzeitig die Winkel- und linearen Koordinaten der Raumlage und die Orientierung des UWAs. Dies führt zur Instabilität und zu wesentlichen Veränderungen der Massenträgheits- und hydrodynamischen Charakteristiken des Apparates. Das wird durch starke Wechselbeeinflussung der Freiheitsgrade untereinander und durch die Veränderung der zähflüssigen Reibung[6] hervorgerufen.

Zahlreiche Untersuchungen sind der Erhöhung der dynamischen Genauigkeit und Robustheit von Steuerungssystemen für Unterwasserapparate gewidmet. So werden in den Arbeiten [7-9] verschiedene Arten der anpassungsfähigen Regler, einschließlich der Regler mit variabler Struktur [6], erörtert. Ungeachtet der bedeutenden Erfolge auf dem Gebiet der Steuerung des UWAs, ist die Mehrheit der Regler sehr komplex und kompliziert in der Abstimmung und der Realisierung.

Das zweite Objekt der Regelung in dieser Arbeit ist ein Magnetlager. Es besteht aus geregelten Elektromagneten und einem metallischen Körper, der sich im magnetischen Feld befindet. Das Ziel der Steuerung ist die Lagestabilisierung des metallischen Körpers in einer vorgegebenen Entfernung zu den Elektromagneten. Die Stabilisierung der Lage und der Orientierung des metallischen Körpers wird umgesetzt mittels einer Regelung des magnetischen Feldes.

Die Hauptprobleme der Regelung dieses komplizierten Objekts beruhen auf dem nichtlinearen elektromagnetischen Zusammenwirken zwischen dem metallischen Körper und dem magnetischen Feld, welches von den geregelten Elektromagneten erzeugt wird. Die Anziehungskraft des Magneten ist zum Quadrat des Stromes proportional und hängt von der Spannung des magnetischen Feldes ab. Diese Kraft hängt stark von den Materialeigenschaften des Elektromagneten ab und wird durch die Magnetisierungskurve des Materials bestimmt. Diese Kurve ist nichtlinear und hängt sowohl von den Eigenschaften des Materials, als auch von der Geometrie des Magnets und der Spule ab. Die Magnetisierungskurve entsteht nur im stationären Fall und unter der Bedingung, dass sich der Strom in der Spule langsam ändert und der in Luft schwebende Körper sich in konstanter Entfernung vom Pol der Elektromagneten befindet. Bei Veränderung der Lage des metallischen Körpers ändern sich die Breite des Luftspaltes und damit die magnetische Leitfähigkeit (der magnetische Widerstand). Weiterhin gelangt die Magnetisierungskurve bei großen Strömen in einen Sättigungsbereich. Bei Veränderungen der Stromstärke spaltet sich die Magnetisierungskurve auf und es kommt zu einer Hysterese. Diese Hysterese entsteht durch unterschiedliche Abhängigkeiten der Magnetisierung und der Entmagnetisierung. Der magnetische Widerstand des Luftspaltes hängt von seiner Größe nichtlinear ab. Des Weiteren hängt der magnetische Widerstand von den magnetischen Eigenschaften der Luft, der Körper- und Polgeometrie des Magneten ab.

Eine Vergrößerung der Stromstärke führt zur Vergrößerung der Anziehungskraft und zu einer Abstandsverminderung zwischen dem Körper und dem Pol der Elektromagneten. Dies führt wieder zu einer Vergrößerung der magnetischen Durchlässigkeit des Luftspaltes und zu einer noch größeren Anziehungskraft. Das bedeutet, dass das System instabil ist, da das Objekt eine positive Rückkopplung bezüglich seiner Lage besitzt. In einem solchen System ist die Geschwindigkeit, mit der sich die Parameter ändern, sehr groß und der Magnet haftet binnen einer Zeit von ca. 0,01 Sekunden nach dem Auftreffen am Körper fest, sofern keine spezielle Steuerung eingreift.

Die Lösung der Positionierungsaufgabe des Körpers im magnetischen Feld mit Nutzung eines optimalen, robusten, anpassungsfähigen oder ähnlichen Reglers wird in den Arbeiten [10-14] erörtert. Die dort behandelten Regelungssysteme sind jedoch nur im begrenzten Umfang für Parameteränderungen der Regelstrecke und unter gewissen Bedingungen anwendbar. Solche Bedingungen umfassen beispielsweise eine Geschwindigkeitsbeschränkung der Lageveränderung. Bei der Steuerung der magnetischen Levitation werden die Bereiche mit nichtlinearen magnetischen Eigenschaften der Elektromagneten und der Luftspalte nicht berücksichtigt. Die klassischen Regler lösen die präzise Positionierungsaufgabe und Stabilisierung nur im linearen Bereich der Magnetisierungskurve. Für die Lösung der angegebenen Aufgabe werden verschiedene Ansätze verwendet, die die Aufgabe vereinfachen. Diese Ansätze können jedoch oft nicht die benötigte Qualität der Regelung bei realen Parameteränderungen gewährleisten.

Das dritte behandelte Regelungsobjekt dieser Arbeit ist die Hand eines mobilen anthropomorphen Roboters. Die Roboterhand lässt sich in zwei wichtige Abschnitte unterteilen, den Roboterarm als ein Manipulator mit sieben Freiheitsgraden und die Hand mit fünf Fingern als Greifer. Der Arm stellt eine Aufgabe der direkten und indirekten Regelung dar.

Die sieben Freiheitsgrade sind auf alle drei Gelenke (Schulter, Ellbogen und Hand) aufgeteilt. In dieser Arbeit wird angenommen, dass alle Freiheitsgrade des Roboterarms unabhängig geregelt

und mit Gleichstrommotoren und einem Getriebe ausgestattet sind. Die Regelung eines solchen Systems hat die Aufgabe Parameteränderungen wie beispielsweise eine Trägheitsmomentenänderung zu berücksichtigen [15-19]. Der Greifer der anthropomorphen Roboterhand ist mit vielen konstruktiven Problemen verbunden. In dieser Arbeit wird eine spezielle Getriebeart, der so genannte „Twisted Wire Actuator“ [20], für solche Anwendungen vorgeschlagen, das mathematische Modell entwickelt und mit Versuchstandergebnissen verglichen.

Für alle drei Regelungsobjekte ist die Entwicklung von Reglern notwendig, die folgende Bedingungen erfüllen. Die Regler müssen möglichst einfach aufgebaut sein und mit minimalen Wissen über die Regelstrecke funktionieren. Weiterhin müssen sie bei sich schnell verändernden Parametern eine präzise Lageregelung gewährleisten.

In dieser Arbeit werden allgemeine Synthesemethoden von hochgenauen Steuerungssystemen untersucht und entwickelt. Diese Methoden haben die Aufgabe eine Steuerung für nichtlineare Objekte mit unbekanntem und sich schnell ändernden Parametern zu realisieren. Dabei sind Probleme mit starken Wechselwirkungen zwischen den Regelungskanälen zu lösen.

Um die festgelegten Ziele zu erreichen, wurden gegenwärtig verwendete Systeme von anpassungsfähigen Regelungen untersucht. Dies schließt auch Untersuchungen zu Neuro-Fuzzy-Netzen ein, um deren Nachteile festzustellen und zu minimieren. Zum Testen der Berechnungs- und Entwicklungsmethoden von Neuro-Fuzzy-Reglern wurden Objekte gewählt, die unterschiedlicher Natur sind und über ähnliche Besonderheiten und mathematische Modelle verfügen. Des Weiteren wurde Fragen zur Architekturauswahl von automatisierten Regelungssystemen (oder automatisierten Steuerungssystemen) gelöst. Hierzu ist der Vergleich verschiedener Regelungsgesetze für verschiedene konkrete Objekte durchgeführt worden.

In diesen Untersuchungen wurden Methoden der klassischen und gegenwärtigen Regelungstheorie angewendet. Dazu gehören insbesondere die PID-Regelungstheorie und die Theorie der adaptiven, robusten, optimalen Steuerung. Es ist auf Verfahren der Steuerungstheorie mit Prognoseberechnung (Predictive control) und Regelungssystemen mit variabler Struktur eingegangen worden. Dafür wurden Ergebnisse der Gebiete Mechanik, Robotertechnik und Programmierung verwendet.

Die Prüfung der entstandenen theoretischen Ergebnisse wurde mittels einer numerischen Modellierung mit MATLAB durchgeführt und mit Experimentergebnissen von Laboranlagen verglichen. Dabei wurden offene Programmpakete und Ressourcen der Programmiersprache C/C++ verwendet. Für das Magnetlager und das Steuerungssystem der Hand des Roboters wurden die 3D-Visualisierung und die Animation mit der Anwendung VRML erstellt.

Letzten Endes wurden auf Grund der ausgeführten Untersuchungen in dieser Arbeit neue Synthesemethoden der Neuro-Fuzzy-Steuerungssysteme in Kombination mit verschiedenen Architekturen geschaffen. Ebenso wurden adaptive Neuro-Fuzzy-Regelungen für die Vorschubantriebe von Unterwasserapparaten, für Magnetlager mit einem und mehreren Freiheitsgraden, sowie für Positions- und Positions-Kraftregelung einer Roboterhand und einem Greifers entwickelt. Im Rahmen der Dissertationsarbeit wurde ein neuer Typ einer adaptiven Regelung, die „adaptive Regelung mit Ersatzfunktionen“, entwickelt. Diese ist bei der Regelung des Magnetlagers und des Roboterarms zur Anwendung gekommen.

Die Parameter aller drei Regelstrecken sind schwer identifizierbar und verändern sich schnell in einem sehr großen Bereich. Daher ist die Anwendung traditioneller Methoden für diese Steuerung nicht möglich oder mit großen Schwierigkeiten verknüpft.

Im Rahmen in der Dissertation ausgeführten Untersuchungen wurden Methoden entwickelt, auf deren Grundlage man hochgenaue Steuerungssysteme verschiedener nichtlinearer dynamischer Objekte unter schwierigen Bedingungen synthetisieren kann. Zu diesen Bedingungen zählen

schnelle, unvorhersehbare und teilweise sehr extreme Änderungen der Objekt- oder Umweltparameter.

Die entwickelten Steuerungssysteme der Vorschubantriebe der UWA erlauben es eine präzise Bewegung auf komplizierten Raumbahnen zu gewährleisten. Das Steuerungssystem für das Magnetlager ermöglicht es kontaktlosen Anlagen verschiedener industrieller Einsatzgebiete zu schaffen.

Im ersten Kapitel der Dissertation wird begründet, warum das Neuro-Fuzzy-Netz unter einer Vielzahl von neuronalen Netzen ausgesucht wurde. Nachfolgend werden eine Architekturanalyse des Neuro-Fuzzy-Netzes, ein Vergleich verschiedener Anpassungsmethoden sowie ihre Analyse entsprechend der zu lösenden Aufgabe durchgeführt. Hier findet man auch eine Übersicht von Methoden anpassungsfähiger Regelungsalgorithmen und ihrer Modifikation, um sie für Neuro-Fuzzy-Netze anwenden zu können. Es werden die allgemeinen Gesetzmäßigkeiten, die Vorzüge und die Nachteile jeder Methode gezeigt.

Im zweiten Kapitel werden Materialien für die Entwicklung des Steuerungssystems der Magnetlagerregelung mit einem und mehreren Freiheitsgraden angegeben und die Ergebnisse der Modellierung sowie Laborexperimente vorgestellt.

Das dritte Kapitel beinhaltet die Beschreibung des UWA-Modells und schildert Erfahrungen bei der Anwendung von Neuro-Fuzzy-Netzen als UWA-Steuerung unter verschiedenen Modifikationen des Steuerungsgesetzes. Weiterhin werden die Ergebnisse der direkt-inversen, auf Prognosedaten basierenden, UWA-Steuerung mit sich anpassenden Neuro-Fuzzy-Netzen verglichen. Am Ende werden die Ergebnisse der Steuerungssysteme mit dem Neuro-Fuzzy-Regelstreckenmodell und der Steuerung verglichen, welche auf dem Ansatz des Wurzelortverfahrens und der identifizierten Regelstrecke basiert. Dabei werden die Vorteile und Nachteile jeder Herangehensweise gezeigt.

Das vierte Kapitel ist den Modellierungsfragen der Roboterhand und der Synthese der Regelungssysteme mit Anwendung der adaptiven Regelung gewidmet. Die Roboterhand wird dabei als redundanter Manipulator mit sieben Freiheitsgraden betrachtet. Die menschenähnliche Fünf-Fingerhand ist mit Zugseilen ausgestattet. Dabei werden Aufgaben der direkten und inversen Kinematik betrachtet und gelöst. Die Roboterhand ist in lokalen Koordinaten (joint space) und in globalen kartesischen Koordinaten geregelt. Besondere Aufmerksamkeit gilt dabei der Kraft-Positionsregelung. Für den Greifer der Roboterhand wird eine „Twisted Wire Actuator“-Konstruktion dargestellt und das Regelungssystem des Greifers wird mit seinen technischen Besonderheiten betrachtet.

In der Zusammenfassung der Arbeit findet man die wesentlichen Ergebnisse der Dissertation.

Danksagung

Diese Arbeit entstand während meiner Tätigkeit als Doktorand am Fachgebiet Elektrotechnik an der Universität Magdeburg. Allen, die hierzu beigetragen haben, gilt mein persönlicher Dank.

Mein besonderer Dank gilt Herrn Univ. Prof. Dr.-Ing. habil Frank Palis, der mir großzügig bei der theoretischen und praktischen Aspekten der wissenschaftlichen Tätigkeit geholfen hat. Ich danke ihm für die damit verbundene Mühe und für die regelmäßigen, anregenden fachlichen Diskussionen und Hinweise, die für diese Arbeit richtungweisend waren.

Hier möchte ich vor allem Herrn Prof. Dr.-Ing. Vladimir Filaretov und Herrn Prof. Dr.-Ing. Wolfgang Günther danken.

Mein Dank gilt auch meinen Kollegen, M. Sc. Andrey Telesh, M.Sc. Alexander Tyshakin, Dr.-Ing. Yuriy Zavgorodniy und M.Sc. Denis Draganov, die mir viel Verständnis entgegenbrachten und es ermöglichte, neben dem Alltagsgeschäft wissenschaftlich zu arbeiten.

Schließlich möchte ich mich bei meiner Frau und meinen Eltern für Geduld und langjährige Unterstützung während der Promotionszeit bedanken.

Inhaltverzeichnis

Kapitel 1. Die Rolle der Approximation in der Steuerungstheorie	9
1.1 Unterschiedliche Approximationsverfahren für Adaptive Regelung	9
1.2 Fuzzy Approximation.	12
1.2.1 Neuro-Fuzzy-Netze	17
1.3 Beschreibung und Auswahl der Neuro-Fuzzy-Netzarchitektur	19
1.4 Gradientenbasierte Trainingsverfahren	22
1.5 Robuste Modifikationen in Anpassungsalgorithmen	29
1.6 Gradientenfreie Optimierung	31
1.6.1 Die Methode der kleinsten Quadrate	31
1.6.2 Andere alternative Trainingsverfahren	32
1.7 Beschränkte Optimierung	33
1.8 Dateianalyseverfahren als Trainingsmittel und Architekturschätzung	36
1.9 Die Linearisierungsverfahren	36
1.10 Neuro-Fuzzy-Netze im adaptiven Regelkreis	38
Kapitel 2. Die Regelung von Magnetlagersystemen	40
Einführung	40
2.1 Systembeschreibung von aktiven Magnetlagern	42
2.1.1 Das Magnetlager mit einem Freiheitsgrad und ausgeprägt inhomogenem Magnetfeld	44
2.1.2 Das Magnetlager mit dem homogenen Magnetfeld	47
2.1.3 Das Magnetlager mit fünf Freiheitsgraden	49
2.2 Regelungstechnische Ansätze für Magnetlager	53
2.2.1 PID- Regler für Magnetlager mit dem ausgeprägten inhomogenen Magnetfeld	53
2.2.2 Zustandsregelung des Magnetlagers mit dem homogenen Magnetfeld	56
2.3 Realisierung des Neuro-Fuzzy-Netzes	59
2.3.1 Herstellung eines Neuro-Fuzzy-Netzes auf einem FPGA-Chip	61
2.4 Die adaptive Regelung mittels Neuro-Fuzzy-Netze	62
2.4.1 Direkt adaptive Neuro-Fuzzy-Regelung des Magnetlagers mit einem Freiheitsgrad und dem ausgeprägten inhomogenen Magnetfeld	63
2.4.2 Direkt adaptive Neuro-Fuzzy-Regelung des Magnetlagers mit dem homogenen Magnetfeld	67
2.5 Adaptiver Neuro-Fuzzy-Sliding-Mode für Magnetlager mit einem Freiheitsgrad und dem homogenen Magnetfeld	69
2.6 Die Linearisierung durch Approximation der Nichtlinearitäten mittels Neuro-Fuzzy-Netz	72
2.7 Adaptive Linearisierung durch Ersatzfunktion	73
Zusammenfassung	78
Kapitel 3. Die Regelung des Vorschubantriebs der autonomen Unterwasserfahrzeuge	79
3.1 Autonomes Unterwasserfahrzeug(UWF)	79
3.1.2 Bereiche erfolgreicher Anwendung von UWF	81
3.1.3 Dynamik des Vorschubantriebs der Unterwasserfahrzeuge	81

3.2 Direkte inverse Steuerung mit Prognose für den Vorschubantrieb des Unterwasserfahrzeuges	82
3.2.1 Die Auswahl des effektiven Algorithmus für die Parameteranpassung in ANFIS NFN	83
3.2.2 Arbeit von NFN in Echtzeit	85
3.2.3 Simulationsergebnisse des direkt inversen Regelungssystemes mit NFN für Vorschubantrieb des UWF mit veränderlichen Parametern	88
3.3 Direkte adaptive Regelung mit Prognose für den Vorschubantrieb des UWF	90
3.3.1 Die Simulationsergebnisse	93
3.4 Indirekte adaptive Regelungen für den Vorschubantrieb des UWF mit Hilfe von Echtzeitzustandreglern	94
3.4.1 Die Simulationsergebnisse	97
Zusammenfassung	98
Kapitel 4 Die Entwicklung und Regelung einer Roboterhand	99
Einführung	99
4.1 Aufbau der Mechanik des Roboterhand	105
4.2 Twisted Wire Actuator	106
4.3 Handstruktur und Fingermechanismen	110
4.4 TWA, elektrischer Antrieb und Fingermechanismus	114
4.4.1. Einmassensystem des Motor-TWA-Fingermechanismus bei einem starr angenommenen Seil	114
4.4.2 Das Zweimassensystem Motor - TWA - Fingermechanismus	115
4.4.3 Der Motor-TWA-Fingermechanismus bei einem elastisch angenommenen Seil	116
4.5 Simulation des Fingermechanismus	116
4.5.1 PI-Regelung für den TWA - Fingermechanismus	117
4.5.2 PI-Regelung mit Kompensation der Nichtlinearität für den TWA - Fingermechanismus	119
4.6 Die Aufgabe der direkten Kinematik	120
4.7 Die Aufgabe der inversen Kinematik	121
4.8 „Computed torque control“ für den Roboterarm	122
4.8.1 Die Kompensation der nichtlinearen Dynamik des elektrischen Antriebs des Roboters durch Feedbacklinearisierung	123
4.8.2 Der äußere und innere Regelkreis	125
4.9 Die modernen Regelungssysteme und Regelungskonzepte eines mechanischen Systems	126
4.9.1 Adaptive Regelung mit Ersatzfunktionen	127
4.9.2 Positionsregelung	129
4.9.3 Positionsregelung in kartesische Koordinaten	132
4.9.4 Die Positions -Kraftregelung des Roboterarms	135
Zusammenfassung	142
Zusammenfassung	143
Ausblick	144
Literaturverzeichnis	145
Anhang A	153

Kapitel 1. Die Rolle der Approximation in der Steuerungstheorie

Bei linearen Steuerungen gibt es kaum Anlass neue Ansätze zu entwickeln, da die bestehenden Regelungsgesetze für lineare Regelstrecken alle möglichen Regelaufgaben bewältigen können. Andererseits ist in der Praxis nicht jedes Objekt ausreichend bekannt und bereitet Probleme bei der Realisierung der Steuerung. Einige Regelstrecken haben nichtlineare Abhängigkeiten zwischen ihren Parametern und machen das System nichtlinear. Manchmal lassen sich lineare Steuerungsverfahren für einen bestimmten Regelungsbereich implementieren, was für niedrige Anforderungen ausreichend ist. Insgesamt wird die Qualität des Steuerungssystems am stärksten von dem Charakter der Nichtlinearität und der Genauigkeit der Modelle beeinflusst.

Bekanntlich können ungenau bekannte Regelungsstreckenmodelle mittels einer adaptiven Funktionsapproximation beschreiben werden [1].

In der nichtlinearen Steuerungstheorie spielt die Approximation eine wichtige Rolle, weil sie es erlaubt, ungenau bekannte Parameterwerte oder Signale durch einen Näherungswert zu ersetzen, welcher ständig angepasst werden muss. Dies ermöglicht auf Approximation basierte Methoden mit einer linear adaptiven Methode und einem rückkopplungslinearisierten System zu behandeln. Tatsächlich benutzt man bei einer linear adaptiven Methode lineare lokale Schätzmodelle und bei der Approximation wird als Schätzmodell ein „Approximator“ verwendet [2]. Lineare lokale Schätzmodelle funktionieren jedoch nur hinreichend genau in einem kleinen Bereich um den Arbeitspunkt und bei sich langsam ändernden Parametern der Regelstrecke. Für Objekte mit sich schnell verändernden Parametern müssen komplexe adaptive Regler benutzt werden. Im Prinzip ermöglicht die Approximation eine Annäherung an die reale Regelstrecke und berechnet ein Regelungssignal, welches genau dem aktuellen Zustand der Regelstrecke entspricht.

1.1 Unterschiedliche Approximationsverfahren für Adaptive Regelung

Für die Approximation gibt es viele Verfahren, wie z.B. physikalisch basierte Modelle, Polynom Approximation, Spline, Neuronale Netze wie Multi layer Perseptron (MLP) oder Radial Basis Function (RBF), Fuzzy Systeme, Wavelets, etc. [1]- [3]. Jedes von ihnen hat Vor- und Nachteile.

Bei den *physikalischbasierten Modellen* ist die mathematische Abhängigkeit zwischen den Objektsignalen bekannt, aber die Modellparameter sind teilweise oder gänzlich unbekannt.

Nehmen wir als illustratives Beispiel ein einfaches Pendel. Die Dynamik kann man mit der

Gleichung $\ddot{\alpha}(t) = -\frac{g}{l} \sin(\alpha(t)) + \frac{1}{ml^2} \tau(t)$ beschreiben.

Hier sind g die Gravitation, l die Pendellänge, m die Masse τ die Kraft.

Das Anpassungsmodell lässt sich mit $\ddot{\alpha}(t) = \theta_1 \sin(\alpha(t)) + \theta_2 \tau(t)$ beschreiben, wobei θ_1, θ_2 die Anpassungsparameter sind.

Auf Grund des einfachen Regelstreckemodells kann dieses Verfahren sehr gute Ergebnisse in Kombination mit Offline- und Online-Anpassungsgesetzen erzielen. Dennoch wird eine Online-Anpassung verwendet, da der Schätzparameter ständig variiert. Haben die ermittelten Objektzustandsparameter einen komplizierten Charakter, treten bei diesem Verfahren Probleme mit der Genauigkeit der Optimierung auf. Für diesen Modelltyp werden auch Modelle in Zustandsraumdarstellung oder andere Modellformen verwendet.

Als zweite verbreitete Art von Approximationsverfahren sind die „*Model Free Approximatoren*“ zu nennen, die keinen Bezug zu den approximierten physikalischen Parametern oder Modellen haben und trotzdem als ein Teil der physikalisch basierten Modelle oder als so genannte „black box“ verwendet werden können.

Die einfachste Methode aus dieser Gruppe ist die Approximation mittels *Polynomen*.

Jedes Polynom lässt sich mit $P_n = \sum_{i=0}^N a_i x^i$ beschreiben. Sehr bekannt ist das Taylor-Reihen-

Polynom, mit welchem die Approximation der Funktion sehr gut entwickelt werden kann [4].

Es gibt eine Reihe von anderen auf Approximationsverfahren basierten Polynomen, die allgemein folgende Eigenschaften haben:

Polynome haben eine begrenzte Dimension N mit linearer Basis.

Die approximierten Abhängigkeit ist glatt (stetig).

Polynome sind einfach zu realisieren, zu berechnen und zu implementieren.

Ableitungen und Integrale von Polynomen können algebraisch gefunden werden.

Es gilt das Weierstrass-Theorem: Wenn es eine Funktion im Bereich $[a, b]$ gibt, dann existiert ein Polynom mit der endlichen Ordnung N , das eine unendlich kleine Abweichung von der tatsächlichen Funktion hat.

Der Nachteil vom Punkt 5 ist die hohe Ordnung des Polynoms, die man jedoch manchmal braucht, um eine gute Qualität der Approximation bei niedriger Parameteranzahl zu bekommen.

Mit der *Splinefunktion* kann man weiter Polynomvorteile nutzen. Der Splineapproximation liegt ein Verfahren zugrunde, das den Bereich der Approximation in kleine Teile teilt und für jeden relativ kleinen Teil ein Polynom mit niedriger Ordnung benutzt. Es ist sinnvoll die Spline-Approximation mit dem sogenannten „B-spline“ oder mit dem „natural spline“ Approximator durchzuführen. Der „B – spline“-Approximator lässt sich mit

$$f(\theta, \phi(x)) = \theta^T \cdot \phi(x) = \sum_{i=1-k}^{N-1} \theta_i \phi_i(x) \quad (1)$$

beschreiben, wobei θ der Vektor der anpassenden Parameter, $\phi(x)$ - die Spline-Funktion, N die Anzahl der Approximationsbereichsteile und k die Ordnung des Spline-Approximators sind.

Mit dem Spline-Approximator ist es möglich sich einer beliebigen Funktion im begrenzten Bereich mit einer gewünschten Genauigkeit zu nähern. Die Näherungsfehler nehmen mit zunehmender Unterteilung des Approximationsbereiches ab [4]. Es ist zu erkennen, dass die Parameter θ in der Formel (1) linearabhängig von f sind und die Splinefunktion ein nichtlinearer Teil vom Approximator ist, was die praktische Anwendung sehr vereinfacht.

Die Beschreibungsform (1) gilt für eine große Reihe von Approximationsverfahren, ungeachtet der unterschiedlichen Strukturen, Funktionierungsprinzipien und dem Ursprung.

Radialbasierte Funktionen (RBF) [5], [6] sind sehr bekannt in Neuronalen Netzen und sehr nützlich bei der Approximation. Den RBF-Approximator beschreibt man wie folgt:

$$f(x) = \sum_{i=1}^N \theta_i g(\|x - c_i\|) + \sum_{i=1}^M b_i p_i(x), \quad (2)$$

wobei RBF mit $g(x) = \exp\left(-\frac{(x-c)^2}{2\delta^2}\right)$ berechnet werden kann. Das zweite Glied in der Formel 2

ist ein Polynom mit den Koeffizienten b und der Ordnung M .

Es gibt sehr viele Funktionen, die mit dieser Approximation genähert werden können.

Bei allen dieser Verfahren ist es möglich, den RBF-basierten Approximator als

$$f(\theta, \phi(x)) = \theta^T \cdot \phi(x, c, \delta) = \sum_{i=1-k}^{N-1} \theta_i \phi_i(x, c, \delta), \quad (3)$$

zu definieren. Wie man sehen kann, ist es fast dieselbe Gleichung wie Formel (1). Die auf RBF basierte Approximation hat eigene Besonderheiten:

Der Näherungsfehler kann gleich null sein, falls das Zentrum der RBF gleich aller Trainingsdaten aus der Trainingsmenge ist.

Der Rechenaufwand wächst exponential mit zunehmender Trainingsmenge, sofern gleiche Fehlerniveauforderungen gelten.

Den Zusammenhang zwischen B-Splines und RBF kann man in [1] nachlesen.

Das Perzeptron oder *Muti layer Perseptron (MLP)* hat eine lange Geschichte [7]. Beispiele der Anwendung des MLP in Regelungssystemen sind in [1]-[3] dargestellt.

Der Ausgang des Perzeptrons wird als v_i bezeichnet, welcher mit

$$v_i = g(b_i + \sum_{j=1}^n w_{ij} x_j) \quad (4)$$

definiert wird. Das gesamte Einschicht-Perzeptron wird mit der Gleichung

$$y(x) = \sum_{i=1}^n \theta_i v_i(x, w) \quad (5)$$

beschrieben. Hier erkennt man eine Ähnlichkeit mit Gleichung (1), weil die Koeffizienten θ und der Ausgang $y(x)$ lineare von einander abhängig sind. Abbildung 1 verdeutlicht in graphischer Form die Struktur des Einschicht-Perzeptron nach Gleichung (5).

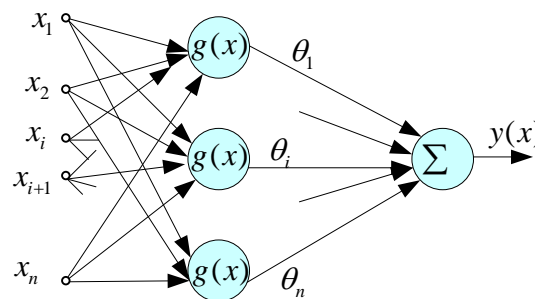


Abbildung 1. Einschicht-Perzeptron.

Die Besonderheit in diesem System ist ein Ausgang, der mit vielen Eingangssignalen x verbunden ist. Für die Funktion $g(x)$ muss mindestens die erste Ableitung existieren.

Dies ist zwingend erforderlich für das „Back Propagation“-Trainingsverfahren [8] (BP Algorithmus), weil dafür die Ableitung von $g(x)$ existieren muss. Häufige Funktionsbeispiele sind

$$g(x) = \arctan(x), g(x) = ax + b, g(x) = \text{htan}(x) \text{ oder } g(x) = \frac{1}{1 + e^{-x}}.$$

Für das MLP verwendet man eine Mehrschichtenanordnung, die durch seriell kombinierte Einschicht-Perzeptionen aufgebaut ist. Diese Netzart nennt man „feed forward“, weil die Signale nur eine Richtung – von links nach rechts (siehe Abbildung 1) – haben. In der Literatur wird das MLP-Netz oft als Netz mit überwachtem Lernen (Supervised learning) bezeichnet. In direktem Sinn bedeutet das, dass für jedes Eingangssignal ein bestimmtes Ausgangssignal existiert. Wenn man ein Anpassungsgesetz schreibt, das auf Ableitungen der Fehlerfunktion basiert, ist das Wissen über das Ausgangssignal nicht zwingend erforderlich. Wichtig ist nur eine

Fehlerfunktion, die minimiert werden soll. Dies trifft für Trainingsverfahren mit Lehrer bei der Anwendung eines MLP zu, sofern das Ausgangssignal des MLP fest eingestellt ist. Andernfalls handelt es sich um eine Trainingsmethode, die sich auf „Reinforcement“ oder Förderung stützt. Es ist immer zu beachten, dass die Minimierung des Fehlergradienten eine allgemeine Methode ist, die auch von vielen anderen Approximationsverfahren erfüllt werden kann.

In der Literatur über neuronale Netze befindet sich eine Reihe von Aussagen, die als Motivation für die Anwendung von Neuronalen Netzen in Steuerungssystemen dienen. Jedoch müssen diese stets kritisch und immer unter Berücksichtigung von Alternativen betrachtet werden.

Üblicherweise nennt man als Vorteil des MLP die Approximationsfähigkeit, andere verwendbare Methoden bleiben jedoch unbeachtet. Die Netzstruktur ist schwer zu bestimmen und wird in dem meisten Fällen durch praktische Erfahrung definiert.

Anwendungen für MLPs sind parallele Berechnungen, jedoch kann auch jeder Approximator, der sich mit Vektoralgebra beschreiben lässt, ebenfalls parallel berechnet werden.

Oftmals wird als Vorteil von MLPs genannt, dass sie eine Ähnlichkeit mit biologischen neuronalen Netzen besitzen und dadurch fähig sind komplizierte Aufgaben wie das Laufen, Sehen, die Spracherkennung und Sprachsynthese zu lösen. Leider wird dabei nicht beachtet, dass reale neuronale Netze (z.B. Gehirn) weit aus komplexer aufbaut sind und es noch nicht vollständig geklärt ist, wie das Gehirn funktioniert, Informationen bearbeitet und speichert. Ob neuronale Netze das menschliche Gehirn / Bewusstsein jemals ersetzen können, ist noch fraglich.

1.2 Fuzzy Approximation.

Weiterführende Information über die Fuzzy-Logik sind in [9]-[11] zu finden. Es gibt viele detaillierte Diskussionen über Regelungen und Steuerungen mittels Fuzzy-Logik mit adaptiven und mit festen Parametern [12]-[17].

Die Grundstruktur von Fuzzy-Logiksystemen ist in Abbildung 2 zu sehen.

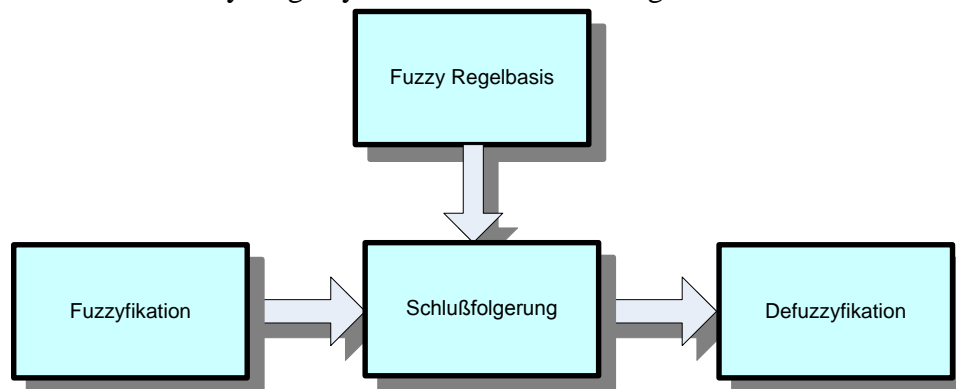


Abbildung 2. Die Grundstruktur von Fuzzy-Logiksystemen.

Unter der „Fuzzyfikation“ versteht man, dass Eingangsvariabel oder Eingangssignale bewertet werden. Dann werden auf der Basis der angegebenen Bewertungen Schlussfolgerungen berechnet. Letztendlich wird aus der aktivierten Menge von Aussagen mit der „Defuzzifikation“ das Ausgangssignal definiert.

Fuzzy-Regler bestehen, wie Abbildung 2 zeigt, im Allgemeinen aus vier Funktionsblöcken:

- Wissensbasis
- Fuzzifizierungsblock

- Inferenzmechanismus und Regelbasis
- Defuzzifizierungsblock.

Fuzzy-Regler sind wie regelbasierte Expertensysteme für Prozessregelungen geeignet. Das heißt, dass ein „Regelexperte“ mit einem Fuzzy-Regler nachgebildet werden kann, wenn das Expertenwissen in Regeln ausgedrückt werden kann.

Die sogenannte Regelbasis des Fuzzy-Reglers kann durch gezielte Befragung von Regelexperten und Systembedienern ermittelt werden. Das Wissen des Bedieners beruht auf Erfahrungen und wird deshalb auch als heuristisches Wissen bezeichnet. Kann dieses Wissen in Form von Wenn-Dann-Regeln erfaßt werden, so ist eine leichte Umsetzung in Fuzzy-Regeln möglich. Der Vorteil des sprachlichen Wissens ist die Anschaulichkeit. Außerdem werden durch die linguistische Beschreibung unscharfe, also nicht quantitativ exakte Zusammenhänge zugänglich. Aus der Wissensbasis können dann die Bezeichnungsmengen der verschiedenen linguistischen Variablen mit ihren Grundmengen, die Zugehörigkeitsfunktionen, die Regeln des Fuzzy-Reglers, die Inferenzmethode und die Defuzzifizierungsmethode extrahiert werden. Dies wird auf Abbildung 2 durch die Pfeile zwischen der Wissensbasis und den einzelnen Blöcken angedeutet.

Im Fuzzifizierungsblock werden den physikalisch scharfen Eingangswerten Zugehörigkeitswerte entsprechend ihren linguistischen Werten zugewiesen. Man sagt, eine Regel „feuert“, wenn der Zugehörigkeitswert der Regelprämisse größer als Null ist. Feuern l Regeln einer Regelbasis gleichzeitig, dann werden in einem zweiten Schritt durch den Inferenzmechanismus die Fuzzy-Mengen am Ausgang zusammengefaßt.

Die Regeln eines Fuzzy-Reglers stellen im Allgemeinen die Abhängigkeit der Ausgangsgröße von den Eingangsgrößen dar. Die Regelbasis fasst alle Regeln des Fuzzy-Reglers zusammen und kann in Tabellenform dargestellt werden. Der Inferenzmechanismus legt fest, wie Schlußfolgerungen aus den Regelprämissen entsprechend den festgelegten Regeln zu ermitteln sind. Im ersten Schritt ermittelt der Inferenzmechanismus mit dem Implikationsoperator zu dem vorliegenden Eingangswert für jede Regel die Fuzzy-Menge des Konklusionsteiles. Im zweiten Schritt werden die Fuzzy-Mengen der Regelkonklusionen durch den Inferenzmechanismus zu einer resultierenden Fuzzy-Menge, der Gesamtmenge, zusammengefasst. Dieser zweite Schritt wird als Aggregation bezeichnet.

Der Defuzzifizierungsblock wird benötigt, um aus der Fuzzy-Menge U wieder eine scharfe Ausgangsgröße zu errechnen. Die gebräuchlichsten Defuzzifizierungsmethoden sind die Defuzzifizierung nach der Schwerpunktmethode und die Defuzzifizierung nach der Methode der gewichteten Summen (Höhenmethode).

In der Fuzzy-Mengentheorie und Fuzzy-Logik werden anstelle der scharfen Mengenzugehörigkeitswerte 0 und 1 beliebige Zugehörigkeits- oder Wahrheitswerte im Intervall $[0,1]$ zugelassen. Mit den Methoden der Fuzzy-Mengentheorie können damit unscharfes („fuzzy“) Wissen, subjektive Erfahrungen und qualitative Informationen formuliert werden.

Grundlage der Regelstrategien sind Expertenwissen und heuristisches Wissen, die häufig in sprachlicher Form vorliegen und im Allgemeinen nicht numerisch exakt beschrieben, aber durch kausale Wenn-Dann-Regeln erfaßt werden können. In der Fuzzy-Regelungstechnik werden Stabilitätseigenschaften des geregelten Systems, günstiges oder optimales Einschwingverhalten oder Robustheit der Regelung untersucht [15]. In der klassischen Mengentheorie gehört ein Element entweder zu einer Menge oder nicht. Klassische Mengen werden deshalb auch scharfe Mengen, im Sinne von eindeutiger Zugehörigkeit, genannt.

Eine unscharfe Menge oder auch Fuzzy-Menge A ist eine Menge von 2-Tupeln, die die Elemente x einer Grundmenge X mit ihren Zugehörigkeitswerten $\mu_A(x)$ zu dieser Fuzzy-Menge verbindet.

Der Zugehörigkeitswert $\mu_A(x)$ liegt im Intervall $[0,1]$. Man sagt dann, ein Element x der Fuzzy-Menge A gehört umso stärker zu dieser Menge, je größer sein Zugehörigkeitswert ist. Eine Fuzzy-Menge A ist damit eindeutig durch die entsprechende Zugehörigkeitsfunktion beschrieben, die aus der gesamten Grundmenge X definiert ist:

$$A = \{(x, \mu_A(x)), x \in X\}.$$

Eine linguistische Variable ist eine Variable, deren Werte sprachliche Ausdrücke sind. Bezeichnet man die linguistische Variable mit X , dann wird die Menge der Werte, die X annehmen kann, mit LX bezeichnet und Bezeichnungsmenge („term set“) genannt. Jeder linguistische Wert LX_k wird durch eine eigene Zugehörigkeitsfunktion über Grundmenge definiert.

Durch die Zugehörigkeitsfunktion $\mu_{X_k}(x)$ wird dann einem physikalisch scharfen Wert x aus der Grundmenge X einen Wert der linguistischen Variablen LX_k mit dem entsprechenden Zugehörigkeitswert aus dem Intervall $[0,1]$ zugeordnet: $\mu_{X_k} : x \rightarrow [0,1]$.

In der Fuzzy-Logik ist eine linguistische Aussage ein Ausdruck, in dem linguistische Variablen in Beziehung zu sprachlichen Werten gesetzt werden. Solche Aussagen haben die Form linguistische Variable X - Vergleichsoperator - linguistischer Wert LX_i .

Der linguistische Wert LX_i ist ein Element der Bezeichnungsmenge LX der Variable X . Der gebräuchlichste Vergleichsoperator ist der „ist gleich“ - oder kurz „ist“ - Operator. Linguistische Aussagen können Wahrheitswerte im Intervall $[0,1]$ annehmen. Für die linguistische Aussage veranschaulicht die Zugehörigkeitsfunktion die Zuordnung der Wahrheitswerte für unterschiedliche Eingangsgrößen.

Mit dem Konzept der linguistischen Variablen und der linguistischen Aussage ist es möglich, linguistische Regeln zu definieren. Diese Regeln werden auch als Fuzzy-Regeln bezeichnet. Die kanonische Form der Fuzzy-Regel ist die Wenn-Dann-Regel:

Regelprämisse	Regelkonklusion
Wenn X ist A	Dann Y ist B

Der Wenn-Teil der Fuzzy-Regel heißt auch Regelvoraussetzung, der Dann-Teil auch Regelschlußfolgerung. X und Y sind die Namen der linguistischen Variablen, A und B sind linguistische Werte. Die Regelprämisse „Wenn X ist A “ wird ausgewertet, indem sie mit der Zugehörigkeitsfunktion $\mu_A(x)$ überprüft wird, welchen Zugehörigkeitswert die Aussage „ X ist A “ für x besitzt. Der Implikationsoperator legt fest, wie aus der Regelprämisse die Konklusion ermittelt wird. Die Fuzzy-Menge der Regelkonklusion „Dann Y ist B “ wird ermittelt, indem die Zugehörigkeitsfunktion $\mu_B(y)$ der Fuzzy-Menge B durch den Implikationsoperator entsprechend dem Prämissenwert $\mu_A(x)$ Abbildung 3 verändert wird.

Sind in der Regelprämisse mehrere Bedingungen formuliert, die durch „und“ verknüpft sind, so ergibt sich die Fuzzy-Menge der Regelprämisse als Schnittmenge der einzelnen Fuzzy-Mengen. Die Zugehörigkeitsfunktion der Fuzzy-Menge der Regelkonklusion folgt dann entsprechend: Wenn X_1 ist A_1 und X_2 ist A_2 , dann Y ist B .

Eine Regelbasis für die linguistischen Variablen X_1, \dots, X_n und Y fasst alle Fuzzy-Regeln der Form:

$$R(j): \text{ Wenn } X_1 \text{ ist } A_1 \text{ und } X_2 \text{ ist } A_2 \quad \text{ Dann } Y \text{ ist } B.$$

dieser Variablen zusammen. Im Gegensatz zu den Mengenoperationen, die auf mathematischen Definitionen beruhen, repräsentieren Fuzzy-Regeln im Allgemeinen heuristisches Wissen. Für

jede mögliche Kombination linguistischer Aussagen „ X_i ist A_{ij} “ der Regelprämisse sollte auch eine entsprechende Regel existieren. In diesem Fall ist die Fuzzy-Regelbasis vollständig. Weiterhin sollte die Regelbasis konsistent, also widerspruchsfrei, sein. In einer konsistenten Regelbasis dürfen keine Regeln mit gleicher Regelprämisse, aber unterschiedlichen Regelkonklusionen, auftreten. Seit 1972 wurde von L. Zadeh das Konzept eines Fuzzy-Systems entworfen [18]. Die ersten Fuzzy-Controller wurden von F. Mamdani und S. Assilian in [19] entwickelt. Diese Mamdani-Controller haben heute noch eine große Verbreitung in Theorie und Praxis. M. Sugeno und T. Takagi modifizierten den Mamdani-Controller, um den Berechnungsaufwand zu reduzieren. Dadurch kam es zu einigen Abweichungen vom ursprünglichen Mamdani-Controllern [20]. Diese Unterschiede betreffen nur die Ausgangsschicht oder die Defuzzifikation. Auf Abbildung 3 wird ein allgemeines Mamdani-Fuzzy-System dargestellt. Hier sieht man, dass die ersten zwei Schichten die Fuzzifikation ausführen. Die Fuzzy-Menge kann eine beliebige nichtlineare Funktion $f(x) \leq 1$ sein. Eine Regel realisiert man als Multiplikation oder Minimum der Argumente für eine „Und“-Regel und als Maximum der Argumente oder „Probor“ (probabilistic OR)-Operation für eine „Oder“-Regel. Logikoperationen wie „und“ und „oder“ lassen sich mittels so genannte t-Normen und t-Conormen realisieren.

Eine Funktion $t : [0,1]^2 \rightarrow [0,1]$, heißt t-Norm, wenn sie kommutativ und assoziativ ist, d.h.:

$$t(\alpha, \beta) = t(\beta, \alpha), \quad t(t(\alpha, \beta), \gamma) = t(\alpha, t(\beta, \gamma)), \quad \text{und} \quad t(\alpha, 1) = \alpha.$$

Eine Funktion $t : [0,1]^2 \rightarrow [0,1]$, heißt t-Conorm, wenn sie im Gegensatz zur T-Norm die Gleichung $t(\alpha, 0) = \alpha$ erfüllt.

Die dritte Schicht des Fuzzy-Netzes ist verantwortlich für Fuzzy-Regelberechnungen und zwei weitere Schichten realisieren die Defuzzifikation. Die Implikation realisiert man auch mit dem Minimum oder der Multiplikation des Regelausgänge und Fuzzy-Ausgangsfunktionen $f(x) \leq 1$ und $x \in (-\infty, +\infty)$. In der letzten Schicht wurden sehr rechnungsaufwendige Operationen versteckt. Denn die Defuzzifikation ist im Prinzip eine Integration des gemeinsamen Ausgangs der Implikationsschicht. Auch für die Defuzzifikation existiert eine Alternative. Zum Beispiel kann man die Integration des Ausgangsmengen durch „Cetroid“- , „Bisector“- , „LOM“- , „MOM“- , oder „SOM“-Berechnungen ersetzen. Der hohe Berechnungsaufwand bei Mamdani-Systemen ist fast der einzige Grund für die Verwendung des Takagi-Sugeno-Prinzips in der Fuzzy-Theorie.

Ähnlich wie die Approximationssätze bei künstlichen Neuronalen Netzen existieren auch für Fuzzy Entscheidungssysteme entsprechende Aussagen. In [19], [20] wird gezeigt, dass ein auf dem Ansatz des Mamdani-Fuzzy-Systems beruhendes System jede stetige Funktion beliebig genau approximieren kann. In [21] befindet sich eine entsprechende Behauptung für bestimmte Sugeno-Controller. Allerdings handelt es sich auch hier um eine reine Existenzaussage. Die Sätze geben keinerlei Hinweis, wie Fuzzy-Mengen und -Regeln zu wählen sind, um eine gegebene Funktion zu approximieren.

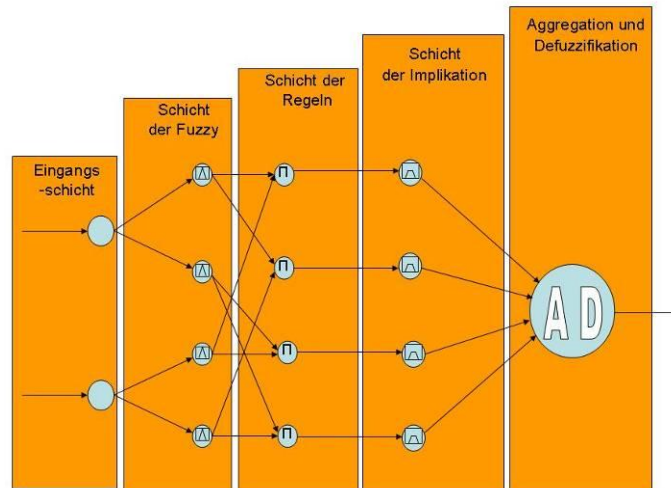


Abbildung 4. Die Architektur des Neuro-Fuzzy-Netzes nach Mamdani

Um diese Frage zu beantworten, wurde eine Reihe von Methoden entwickelt, die eine Regelbasis „automatisch“ definieren und in Anknüpfung an ein Fehlerkriterium die passenden Eingangs-Ausgangs-Funktionen bestimmen. Diese Methoden teilen sich auf in parameteranpassende und strukturanpassende Methoden und in gradientbasierte und gradientenfreie Verfahren. Die am häufigsten verwendeten Verfahren sind in Tabelle 1 dargestellt.

Tabelle 1. Einteilung der Methoden zur Verbesserung von Fuzzy-Systemen

	Gradientenabstieg-basierte Methoden	Gradientenfreie Methoden
Parameteränderung	<ul style="list-style-type: none"> -Back Propagation -Levenberg Marquardt und andere Newton und Quasi-Newton-Methoden -Nachschalten Gradienten ... 	<ul style="list-style-type: none"> -Methode der kleinsten Quadrate -Genetische Algorithmen -Vertrauensbereichverfahren -„Reinforcement Learning“ - Clusteranalyse ...
Strukturänderung	<ul style="list-style-type: none"> -Alle gradientbasierte Verfahren im Bezug zur Strukturelementen (bzw. Kopplungselementen) mit weiteren Grenzbewertung und Änderung der Struktur - Verwendung von Regelgewichte [9] 	<ul style="list-style-type: none"> -Genetische Algorithmen -„Reinforcement Learning“ -Dateianalyse, wie z.B. Clusteranalyse. -Pruning OBD, OBS [9] Kohonen Algorithmus der Selbstorganisation, etc.

Eine Strukturänderung wird nur dann durchgeführt, wenn sie tatsächlich eine Verbesserung eines Gütekriteriums bringt. Die unterschiedlichen t-Normen und t-Conormen als Hauptoperationen mit Fuzzy Mengen bilden die Gesamtmenge der Fuzzy-Systeme.

Für die Parameteranpassung werden häufig Neuro-Fuzzy-Verfahren verwendet.

1.2.1 Neuro-Fuzzy-Netze

Dieser Punkt befasst sich mit der Kopplung neuronaler Netze mit Fuzzy-Systemen. Da bei der Konstruktion eines Fuzzy-Reglers die ihn bestimmenden Parameter (Fuzzy-Mengen und linguistische Regeln) in einer heuristischen Vorgehensweise festzulegen sind, hat man erkannt, dass die Lernfähigkeit neuronaler Netze dabei eine wertvolle Hilfe sein kann. Es wurden schon viele kombinierte Ansätze entwickelt. Dabei sind die Vorteile einer Kombination von neuronalen Netzen und Fuzzy-Reglern hervorzuheben.

Neuronale Regelung und Fuzzy-Regelung weisen als Methoden der kognitiven Regelung unterschiedliche Vor- und Nachteile auf. Welche von diesen Methoden für ein gegebenes technisches System sinnvoll ist, muss in Abhängigkeit von ihren Eigenschaften, der Kenntniss und den Gegebenheiten des zugrundeliegenden Systems entschieden werden.

Ein Neuronaler Regler ist in der Lage, aus Beispieldaten zu lernen. Dabei können alle bekannten Architekturen und Lernalgorithmen für neuronale Netze eingesetzt werden. Besteht die Möglichkeit, einen Bediener zu beobachten, so lässt sich aus dem aktuellen Zustand und der Bedienerreaktion eine feste Lernaufgabe konstruieren. In diesem Fall kann als Netzstruktur z.B. ein mehrschichtiges Perzeptron mit dem Backpropagation-Verfahren gewählt werden.

Der Einsatz eines Neuronalen Reglers ist immer dann möglich, wenn kein mathematisches Prozessmodell bekannt ist oder dessen Umsetzung in einen konventionellen Regler aufgrund hoher Kosten vermieden werden soll. Weiterhin muss entweder eine feste Lernaufgabe zur Verfügung stehen, die ein Offline -Lernen ermöglicht, oder der Regler muss mit Hilfe verstärkenden Lernens die Regelaufgabe online durch Versuch und Irrtum erlernen dürfen. Diese zweite Methode kann nur dann angewandt werden, wenn eine ausreichend exakte Simulation des Systems zur Verfügung steht. Die Kenntnis linguistischer Regeln, die das zu regelnde System beschreiben, ist ebenfalls nicht notwendig. Ein Neuronaler Regler ist demnach grundsätzlich in der Lage, vollständig unbekannte Systeme zu beherrschen, soweit nur deren Ein-/Ausgabeverhalten beobachtbar ist.

Der Nachteil neuronaler Regler besteht zunächst in deren fehlender Interpretierbarkeit. Auch nach erfolgreich gelöster Lernaufgabe ist es im Allgemeinen nicht möglich, Wissen z.B. in Form von Regeln aus dem Netz zu extrahieren, um so zu einer Beschreibung des Reglerverhaltens zu gelangen. Derartige Ansätze sind auf bestimmte Netzwerkmodelle beschränkt. Es ist weiterhin nicht möglich, relevante Netzparameter anders als heuristisch zu bestimmen. Eine ungeeignete Wahl kann den Lernerfolg erheblich hinauszögern oder sogar verhindern. Obwohl neuronale Netze fehlertolerant gegenüber ihren Eingaben sind und somit auch unbekannte Systemzustände beherrschen, kann dennoch eine Änderung in wesentlichen Parametern des zu regelnden Systems ein Versagen des Neuronalen Reglers verursachen. Diese Gefahr besteht immer dann, wenn die Lernaufgabe nicht alle denkbaren Systemzustände angemessen berücksichtigt. Eine Wiederaufnahme des Lernvorgangs muss in diesen Fällen nicht zu einer schnellen Verbesserung führen. Gegebenenfalls muss der gesamte Lernvorgang wiederholt werden. Der Vorteil neuronaler Regler, dass kein a-priori Wissen vorliegen muss, kann auch als Nachteil interpretiert werden. Denn selbst wenn a-priori Wissen zur Verfügung steht, gibt es keine Möglichkeit, dieses zu verwenden. Eine Ableitung von Initialisierungsgewichten oder einer bestimmten Netzwerkarchitektur ist nicht möglich.

Vorteile der Fuzzy-Regelung:

Kein mathematisches Prozessmodell notwendig;

A -priori Wissen nutzbar;

Einfache Interpretation und Implementierung.

Nachteile:

- Regelwissen muss verfügbar sein;
- Nicht lernfähig;
- Keine formalen Methoden für „Tuning“;
- Anpassung an veränderte Parameter eventuell schwierig.

Anwendungen von Neuro-Fuzzy-Systemen haben zwei typische Ziele:

1 Lernen / Nachoptimieren eines ungenauen Fuzzy-Modells durch ein neuronales Netz Darstellung.

Man hat ein Problem, für das aufgrund geringer Problemkenntnis nur eine schlechte Fuzzy-Lösung entworfen wurde. Das erhaltene Fuzzy-System kann man in ein neuronales Netz transformieren und anhand von Beispieldaten optimieren. Das optimierte Netz kann direkt benutzt oder in ein Fuzzy-System rücktransformiert werden. Eine Rücktransformation in das Fuzzy-System ermöglicht die Einsichtnahme in das Regelwerk und in die Gestaltung der Zugehörigkeitsfunktionen. Damit ist bei einer Prozessidentifikation ein weiterer Erkenntnisgewinn über den Prozess möglich.

2 Einbringen von Fuzzy-Vorwissen in ein neuronales Netz und Strukturierung des Netzes so, dass das Fuzzy-Modell extrahiert werden kann.

Mittels eines neuronalen Netzes hat man aus einer Sammlung von Beispieldaten eine Lösung generiert oder ein unbekanntes Prozessverhalten identifiziert. Hat man eine spezielle Netzstruktur eingesetzt, so lässt sich das neuronale Netz in ein Fuzzy-System transformieren. Das Fuzzy-System – genauer sein Regelwerk – ist interpretierbar und so kann man eine verständliche Beschreibung der Lösung, z.B. des Prozessverhaltens, gewinnen. Damit können auch Aussagen über Prozesszusammenhänge abgeleitet werden, die aus den aufgenommenen Messreihen nicht sofort ableitbar waren.

Zur Verdeutlichung der Argumente für Neuro-Fuzzy sind in der Tabelle 2 Gemeinsamkeiten, Vor- und Nachteile der beiden Technologien sowie die angestrebten Eigenschaften ihrer Kombination zusammengefasst.

Tabelle 2. Eigenschaften von Systemen.

Neuronales Netz	Fuzzy-Modell
- Vorgabe der Netzstruktur	- Prozesswissen erforderlich
- Zufallsinitialisierung	- aufwendige Wissensakquisition
- Vorwissen nicht einbeziehbar	- Messdaten nicht verarbeitbar
- aufwendige Trainingsphase	- nicht unmittelbar lernfähig
- Konvergenzprobleme	- viele Zugehörigkeitsfunktionen
- Gefahr lokaler Minima	- u.U. sehr viele Regeln
- „black box“ - Lösung	- Rechenzeit
+ numerischer Approximator	+ linguistischer Approximator
+ Lernfähigkeit	+ Erfahrung wird operationalisierbar
+ objektive Optimierung	+ Entwurf durch Regelformulierung
+ rechenzeitgünstiger Betrieb	+ anschauliche Wirkungsweise
Neuro-Fuzzy-Modell	
+ Numerische Daten und unscharfes Wissen kombinierbar	
+ Interpretierbar wie Fuzzy-Modell	
+ Lernfähig wie Neuronales Netz	
+ Reduzierter Trainingsaufwand durch wissensbasierte Vorstrukturierung	
+ Datengestützte Nachoptimierung regelbasierter Beschreibung	
+ Automatische Fuzzy-Modell-Generierung	

Abschließend kann man sagen, dass ein Neuro-Fuzzy-Netz nicht anderes als ein Fuzzy-System ist, das als neuronales Netz betrachtet wurde und deswegen lässt es Lernverfahren zu, wie sie für neuronale Netze typisch sind.

1.3 Beschreibung und Auswahl der Neuro-Fuzzy-Netzarchitektur

Ein vollständig bekanntes Neuro-Fuzzy-Netz ist das so genannte **NEFCON**-Modell. Es wurde Mitte der 90er Jahre an der TU Braunschweig von D. Nauck, F. Klawonn und R. Kruse entworfen [23], [25].

Dieses Netz lässt sich als neuronales Netz betrachten. Die Regelbasis, die Parameter der Zugehörigkeitsfunktionen und die Schicht der Defuzzifikation sind mit gradientenbasierten Methoden veränderbar, um das Fehlerkriterium zu minimieren. Dazu eignen sich auch eine Reihe von speziellen Lernverfahren zur Abstimmung der Anzahl der Fuzzy-Mengen und der Regelbasis [25]. Den allgemeinen NEFCON-Aufbau zeigt Abbildung 5.

Man kann sagen, dass das NEFCON-Modell ein Art „Fuzzy-Neuron“ ist.

Die Vorteile dieses Netzes sind auch gleichfalls ein Grund für Kritik. Für den Prozess des Erlernens müssen Parameter und Architektur des Netzes bestimmt werden. Dieser Vorgang benötigt selbst mit einem geeigneten Verfahren eine gewisse Zeit, weshalb die Konkurrenzfähigkeit mit anderen Neuro-Fuzzy-Netzen wieder in Frage gestellt wird.

Die **MFOS**-Systeme (Münsteraner-Fuzzy-Optimierungs-Systeme) [33] wurden am Ende der neunziger Jahre an der Universität Münster von W.-M. Lippe, St. Niendieck und A. Tenhagen entworfen. Betrachtet man die prinzipiellen Wege, um regelbasierte Fuzzy-Systeme zu optimieren, so stimmen sie mit der Strukturänderung von Fuzzy-Systemen und deren parametrischen Anpassung überein. Die MFOS-Systeme wurden mit dem Ziel entwickelt, die von den anderen Systemen dieser Art bekannten Beschränkungen bezüglich der Art der gebrauchten Fuzzy-Controller bzw. der Optimierungsmöglichkeiten zu vermeiden und damit alle obigen Optimierungsarten anwenden zu können. Ein weiteres Ziel war es, eine höchstmögliche Flexibilität zu ermöglichen.

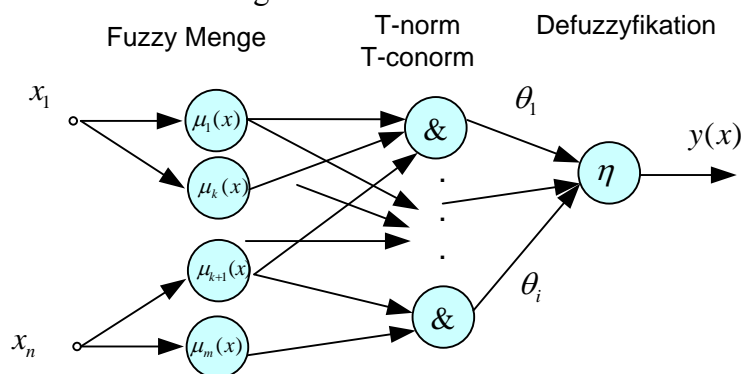


Abbildung 5. NEFCON Neuro-Fuzzy-Netz

Die einzelnen Optimierungsschnitte können interaktiv vom Benutzer gesteuert werden. So ist es z.B. möglich festzulegen, dass keine neuen Regeln oder neue Partitions Mengen erzeugt werden sollen, falls deren Anzahl z.B. durch die Hardwareforderungen eingeschränkt ist.

Das **ANFIS**-System (Adaptive-Network-based-Fuzzy-Inference-System) wurde Anfang der neunziger Jahre als ein universelles System zur Modellierung und Optimierung von Sugeno-Controllern entwickelt.

Auch bei diesem System wird ein gegebener Controller in ein funktional äquivalentes neuronales Netz überführt. Diese Transformation gestaltet sich jedoch etwas einfacher, da bei Sugeno-Controllern die Defuzzifizierung einfach zu realisieren ist [26].

Im Rahmen dieser Arbeit wird ein Neuro-Fuzzy-Netz auf der Basis vom ANFIS-Modell (engl. „Adaptive-Network-based Fuzzy Inference System“) (siehe Abbildung 5) mit „online“-Erlernen realisiert. Im Ganzen ist das ANFIS-Modell ein hybrides Neuro-Fuzzy-System, das zur Funktionsapproximation eingesetzt wird. Es repräsentiert ein Fuzzy-System vom Sugeno-Typ in einem vorwärtsbetriebebenen fünfschichtigen neuronalen Netz.

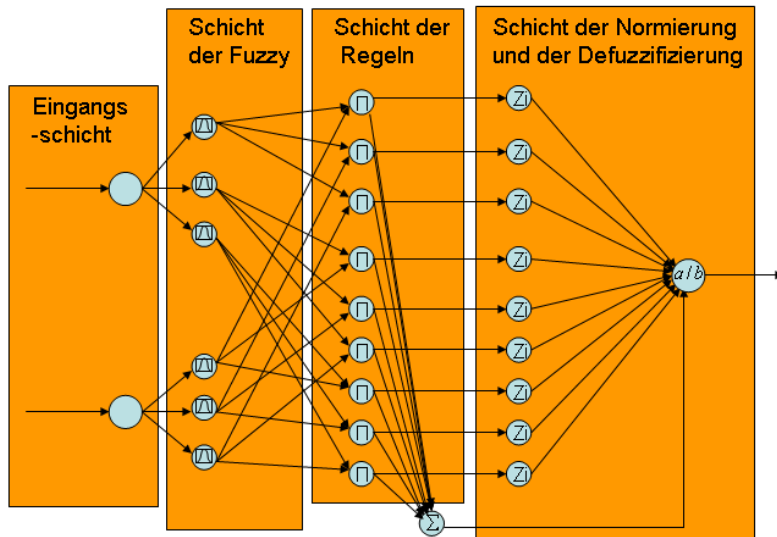


Abbildung 6. Die Struktur des ANFIS-Netzes

Im Unterschied zum NEFCON- oder MFOS-Modell existiert beim ANFIS-Netz ein bestimmter Algorithmus zur Erstellung der Regelbasis. In mathematischer Form kann man das ANFIS-Netz folgendermaßen beschreiben:

$$y(x) = \frac{\sum_{k=1}^M \left(\prod_{j=1}^N \mu_{A_i}(x_j) \right) \cdot Z_k}{\sum_{k=1}^M \left(\prod_{j=1}^N \mu_{A_i}(x_j) \right)} \quad (6)$$

wobei die Zugehörigkeitsfunktion

$$\mu_{A_i}(x_j) = \exp \left(\frac{-(x_j - c_j^{(i)})}{2\delta_j^{(i)}} \right)^{2b} \quad (7)$$

die entsprechenden Fuzzy-Mengen charakterisiert. Mit dem Koeffizienten b kann man die Form der Fuzzy-Menge von Dreiecks- bis Glockenform variieren. Die Breite der Menge wird vom Koeffizient δ und die Mengenmittelposition wird vom Koeffizient c beeinflusst. Die Defuzzifikation wurde mittels linearer Kombination

$$Z_k(x) = p_{k0} + \sum_{j=1}^N p_{kj} x_j \quad (8)$$

der Eingangssignale realisiert.

Eine Ähnlichkeit zwischen der Gleichung (6) und Gleichung (1) ist leicht festzustellen. Demnach können alle Neuro-Fuzzy-Systeme, welche sich in der Form (1) darstellen lassen, als „Universal

Approximator“ betrachtet werden. Wenn die Koeffizienten p_{kj} gleich Null gesetzt werden, bekommt man ein so genanntes Sugeno-System nullter Ordnung. Andernfalls erhält man für das o. g. ANFIS-Netz ein Sugeno-System erster Ordnung.

Das hybride Erlernen (Kombination des Fehlerrückführung und Methode des kleinsten Quadrats) ist anderen Methoden beim Entwurf der Regelbasis und der Parameteranpassung deutlich überlegen.

Das **WANG-MENDEL**-Modell ist ein Sonderfall des ANFIS-Netzes. Diese Modelle unterscheiden sich bei der Berechnung der Regelbasis. Insgesamt hat das Netz von Wang Mendel weniger Parameter und Regeln, weil man bei der Netzkonstruktion nach Erfahrungswerten und automatisierten Algorithmen vorgeht. Die Anzahl der Regeln ist gleich der maximalen Anzahl von Zugehörigkeitsfunktionen für die Eingänge des Netzes. Es ist möglich diese Netzart mittels einer Clusteranalyse automatisch zu erschaffen, weil die Architektur des Netzes in Cluster-Form realisierbar ist. Deswegen gibt es einen festen Zusammenhang zwischen deren Netzarchitektur, ihren Parametern und den Clustern, die aus den Trainingsdaten hervorgehen. In der Praxis ist es ratsam die Möglichkeit in Betracht zu ziehen, das ANFIS-Netz durch das WANG-MENDEL-Netz zu ersetzen, wenn eine große Anzahl von Trainingsdaten vorliegt.

Ein weiterer Sonderfall des ANFIS-Netzes ist das **FUZZY-HYPERBOLISCHES SYSTEM (FHM)**[27].

Bei genauer Betrachtung der ANFIS-Struktur ist es leicht das System, unter Voraussetzung der Gaußzugehörigkeitsfunktion, folgendermaßen umzuwandeln.

Die Zugehörigkeitsfunktionen werden ausgewählt als

$$\mu_P(x_j) = \exp\left(\frac{-(x_j - c_j^{(i)})}{2}\right)^{2b} \quad \text{und} \quad \mu_N(x_j) = \exp\left(\frac{-(x_j + c_j^{(i)})}{2}\right)^{2b}. \quad (9)$$

Dann wird der Ausgang des Netzes wie in Gleichung (6) mit

$$y(x) = \frac{\sum_{k=1}^M \left(\prod_{j=1}^N \mu_{A_i}(x_j) \right) \cdot Z_k}{\sum_{k=1}^M \left(\prod_{j=1}^N \mu_{A_i}(x_j) \right)} = \frac{\sum_{k=1}^M (\mu_P(x_j) - \mu_N(x_j)) \cdot Z_k}{\sum_{k=1}^M (\mu_P(x_j) - \mu_N(x_j))}, \quad (10)$$

berechnet. Auch wenn alles von der Regelbasis abhängt, können die in Form der Gleichung (12) vorliegenden Fuzzy-Systeme weiter bis zur Form:

$$y(x) = \sum_{k=1}^M \tanh(K_j x_j) \cdot Z_k, \quad (11)$$

vereinfacht werden, wobei K_j eine Matrix $[N, N]$ ist, welche dieselbe Bedeutung wie der Koeffizient c im ANFIS-Netz hat. Dieses Fuzzy-System ist ebenfalls ein universeller Approximator.

Als Vorteil der hyperbolischen Fuzzy-Systeme sind zu nennen:

Nichtlinearität des Systems.

FHM-Netze haben viel weniger Parametern als ANFIS-Netze derselben Größe und deswegen sind FHM-Netze für die Modellierung von komplexen dynamischen Systemen geeignet.

Es gibt keine Forderung für Erstellung der Regelbasis und Zugehörigkeitsfunktionen.

FHM lässt sich auch linguistisch betrachten und Expertenwissen ist leicht zum Training des FHM verwendbar

Erlaubt wie bei ANFIS- und FHM-Netzen das Erlernen als neuronales Netz.

Ein Nachteil von FHM-Netzen ist die Unempfindlichkeit des FHM, wenn $|K_{j,x_j}| \gg 0$. Deswegen kommt es zu Problemen durch schlechte Anfangsparameter beim Lernvorgang des Netzes mit gradientenbasierten Methoden.

Das **Fuzzy-ARTMAP-Netz** ist weiterhin ein bemerkenswertes System [27]. ART-Netze wurden von Gail Carpenter und Stephen Grossberg an der Boston University entwickelt. Alle ART-Netze basieren auf der gleichen Grundidee. Das erste Modell, ART-1, wurde bereits 1976 von Grossberg veröffentlicht. Die Nachfolgemodelle entstanden in der Mitte der achtzigsten Jahre.

Das ART-1-Netz kann als anfängliche Version binäre Muster in freistehender Reihenfolge lernen. Das Fuzzy-ART-Modell ist eine Erweiterung des ART-1-Netzes, so dass es analoge als auch Floating-Point-Eingangsmuster erlernen kann. ARTMAP ist eine Kombination zweier ART-Netze zu einem Gesamtnetz. Mittels dieser Kombination ist es möglich, das Gesamtnetz überwacht lernen zu lassen. Diese Eigenschaft besitzen einfache ART-Netze nicht, da bei diesen die Klassifikation der Eingabemuster selbständig, also unüberwacht, erfolgte. Ein Fuzzy-ARTMAP-Netz zeigt besondere Fähigkeiten in der Mustererkennung und Approximation. Diese lassen sich jedoch nicht in der Form von Gleichung (1) beschreiben. Fuzzy-ARTMAP-Netze benötigen einen sehr großen Berechnungsaufwand und besitzen eine hohe Komplexität. Deshalb wurde diese Approximationsart nicht weiter verfolgt.

Die anderen Typen von Neuro-Fuzzy-Netzen wie das **ARIC-**, **GARIC-** und **NNDFR-**Modell bestehen direkt aus der Kombination von neuronalen Netzen und verursachen im Vergleich mit den vorherigen Modellen einen noch höheren komplexen Berechnungsaufwand, weshalb die Anwendungen solcher Netzen fraglich ist.

Eine Reihe von Neuro-Fuzzy-Netzen wie das **FuNe-I-**, **Fuzzy-RuleNet-**, und das **FUN-Modell** haben keine besonderen Eigenheiten im Vergleich mit ANFIS-ähnlichen Netzen und wurden deswegen nicht untergesucht.

1.4 Gradientenbasierte Trainingsverfahren

Jedes der genannten Approximationsgesetze lässt sich mittels **Gradientenabstieg** im Sinn eines Gütekriteriums verbessern. Zuerst wählt man ein Verfahren für die Fehlerberechnung aus. Die Wahl fällt meist auf das s. g. MSE-, oder auf das quadratische Mittelwertsfehlerverfahren.

$$E = \frac{1}{2} \sum_{i=1}^p (y(X_i) - y_{di})^2 \quad (12)$$

Die Gleichung (14) ist ein Beispiel für eine Norm. Eine andere Art an Norm ist die Euklidian-Norm, oder die Summe von Absolutwerten von einfachen Abweichungen. Die Fehlerberechnung oder der Normentyp hat Einfluss auf die Gradientenberechnung, weil die Gleichung (14) zu differenzieren ist.

Die Grundidee besteht darin, einen Gradientenabstieg zu berechnen um, damit neue Parameter zu finden, die zu einem kleineren Fehlerniveau führen. Weiterhin sollen sich die Parameter schnell mit dem Gradientenabstieg ändern, um das Gütekriterium zu minimieren.

Die genaue Formel hängt vom jeweiligen Realisierungsfall ab. Allgemein ist für das Gradientenabstiegsverfahren Gleichung (1) in Gleichung (12) einzusetzen, um den Gradienten zu bestimmen.

$$\frac{\partial E}{\partial \theta} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial \theta} = e \phi(x) \quad (13)$$

In diesem Fall wird der Parameterschritt mit

$$\theta_{i+1} = \theta_i + \eta \frac{\partial E}{\partial \theta_i} \quad (14)$$

ermittelt. Die Variable $\eta > 0$ ist ein Schrittparameter. Die meisten Funktionen $\phi(x)$ sind von vielen inneren Parametern abhängig. Sollen diese verändert werden, muss ein Gradientenabstieg der inneren Parameter von $\phi(x) = \phi(x, c, \delta)$ Parametern berechnet werden.

$$c_{i+1} = c_i + \eta \frac{\partial E}{\partial y(x, c_i, \delta_i)} \frac{\partial y(x, c_i, \delta_i)}{\partial c_i} \quad (15)$$

Dieses Anpassungsgesetz in adaptiven Regelungssystemen bezeichnet man als einfaches Gradientenabstiegsgesetz. In neuronalen Netzen heißt dieses Gesetz „Back Propagation“ (**BP**), welches die gleiche Grundidee verfolgt.

Zuerst muss entschieden werden, wie der Fehler berechnet wird. Der Fehler ist in folgender Form definiert: $E = \frac{1}{2}(y - y_d)^2$ und stellt die Differenz zwischen dem gewünschten Signal und dem Netzsignal dar. Nimmt man als einfaches Beispiel das Netz in der Form: $y = X \cdot w$ an, dann

ergibt sich die Ableitung $\frac{\partial E}{\partial w} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial w} = (y - y_d) \cdot X$. Nun ist es möglich, die Gleichung für die Anpassungskoeffizienten aufzustellen:

$$w_{i+1} = w_i + (y - y_d) \cdot X \quad (16)$$

Die Formel, die hier dargestellt ist, ist ein sogenanntes „Online“-Anpassungsgesetz. Liegt eine große Anzahl von Daten vor, ist es möglich unterschiedliche Variante des Anpassungsgesetzes in Abhängigkeit von der Fehlermengebearbeitung zu erstellen.

Im Nachfolgenden wird die gemeinsame Datenmenge für das Trainieren als N bezeichnet.

Variante 1: Online-BP.

$$w_{i+1} = w_i + (y_i - y_{d_i}) \cdot X_i$$

Das wiederholt man für jeden Trainingsdatensatz, aber nur einmal pro Datei, solange $i \leq N$ ist. Schematisch sieht es, wie in Abbildung 7, aus.

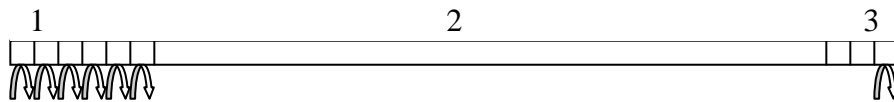


Abbildung 7. Online-Netzerlernung.

Die Pfeile stellen Änderungszyklen entlang der diskreten Zeitachse dar.

Variante 2: Batch-BP.

Zuerst nimmt man ein „Fenster“ oder Teilbereich aus der Datei, die gemeinsame Fenstergröße sei $M < N$.

Die Fensterverschiebung ist im Bereich von $l=1$ bis $l \leq M$ zulässig.

Die Anpassungsformel sieht wie folgt aus:

$$w_{i+1} = w_i + \sum_{k=i+l-1}^{M+i+l-1} (y_k - y_{d_k}) \cdot X_k \quad (17)$$

Dies ist ein so genanntes Batch-Learning-Verfahren. Hier erfolgt nur eine Parameteränderung pro Fenster, bei jedem summierenden Schritt. Ist $M=N$ kommt es zum „**Offline**“-Training.

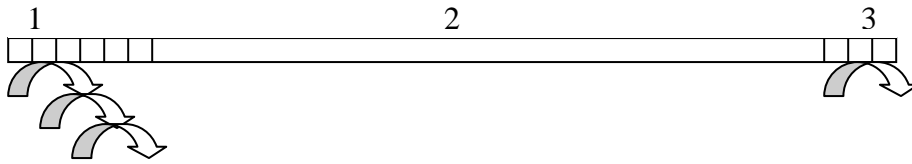


Abbildung 8. „Batch“-Netzlerlernung

Variante 3: Gemischtes Lernen

Eine *Mischung* zwischen der ersten und der zweiten Variante ist mit folgender Formel möglich:

$$w_{i+1} = w_i + \sum_{k=i+l-1}^{M+i+l-1} (y_k - y_{dk}) \cdot X_k \quad (18)$$

Für $k = i + l - 1$ und $M \leq N, l \leq M$ erfolgt eine Änderung der Parameter M-mal pro Dateifenster und das Dateifenster ist mit jedem Schritt in Richtung Trainingsdateiende zu verschieben.

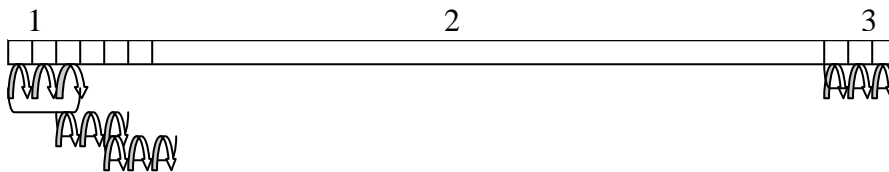


Abbildung 9. Gemischte Netzlerlernung

In der Praxis erhält man die besten Ergebnisse mit der zweiten Variante. Sie besitzt eine geringere Störempfindlichkeit, weil die Abweichungen nicht einem einzelnen Dateifehler, sondern einer Summe von Fehlern entsprechen. Bei $l > 1$ kommt es zu einer inhomogenen Dateibearbeitung, weil gewisse Dateinummern für die Änderungsberechnung häufiger verwendet werden als andere. Die größte inhomogene Dateibearbeitung tritt bei der dritten Variante auf. Den Sonderfall für $M=N$ nennt man „offline learning“.

Obwohl dieses Anpassungsgesetz viele Vorteile wie Schnelligkeit und niedrige Komplexität hat, wird es dennoch kaum verwendet, weil in der Praxis ein reines Gradientenabstieg-Verfahren nicht ausreichend ist.

Das Hauptproblem vom BP-Algorithmus sind lokale Minima. Dieses Problem wird in der Abbildung (9) verdeutlicht. Der BP-Algorithmus startet bei einem bestimmten Wert x , geht in absteigende Richtung des Gradienten weiter, um den Fehler zu minimieren. Trifft man so auf ein lokales Minimum, stoppt der Algorithmus, obwohl es deutlich bessere Lösungen geben kann. Dieses Problem versucht man durch mehrere Anläufe mit unterschiedlichen Anfangspositionen x zu vermeiden.

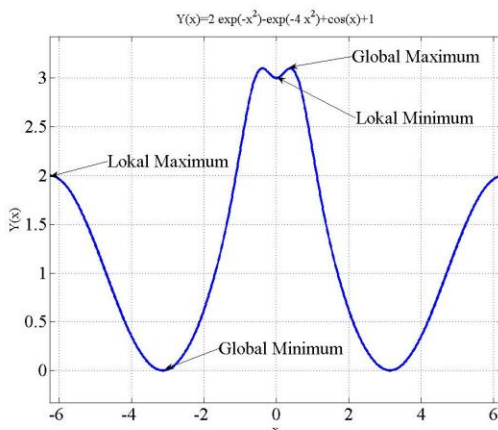


Abbildung 10. Lokalen Minima

Beim BP-Algorithmus gibt es dennoch keine Sicherheit dafür, dass es sich um ein globales Minimum handelt, da alles vom Anfangspunkt abhängt. Auch der Versuch durch eine stochastische gegebene Anzahl von Durchläufen ein Optimum der Gütefunktion zu finden, garantiert dennoch nicht, dass das berechnete Minimum ein Globalminimum ist.

Glücklicherweise besitzt die Gütefunktion viele lokale Minima, die eine praktische Aufgabe hinreichend genau lösen.

Eine Verbesserung bringt der so genannte „**Moment Back Propagation Algorithmus**“ (MBP). Die Grundidee besteht darin, ein Momentum für das Anpassungsgesetz zu finden, das ein Überspringen von relativ kleinen Lokalminima ermöglicht. Der MBP hat folgende Formel:

$$\theta_{i+1} = \theta_i + \eta \frac{\partial E}{\partial \theta_i} + \lambda \left(\frac{\partial E}{\partial \theta_i} - \frac{\partial E}{\partial \theta_{i-1}} \right). \quad (19)$$

Hier ist $\lambda > 0$ ein Momentum-Parameter.

Dieses Verfahren ist nur sinnvoll, wenn die Auslenkungen des Gütekriteriums relativ klein zur Schrittweite des Algorithmus sind. Leider resultiert hieraus oft keine wesentliche Verbesserung für die globale Konvergenz.

Eine weitere Schwäche des BP-Algorithmus ist die relativ geringe Konvergenzgeschwindigkeit. Die Ursache darin liegt im sehr einfachen Anpassungsgesetz, weil der Anpassungsschritt von der ersten Ableitung abhängt. Dies führt zu einer großen Anzahl von Iterationen, bis die Abbruchbedingungen nicht erfüllt werden. Um die Konvergenz zu beschleunigen, ist der Anpassungsschritt anders zu definieren. Eine einfache Variante stellt der **Rprop (Resilient Propagation)** Algorithmus dar. Das Prinzip ist aus der Gleichung leicht zu verstehen:

$$\theta_{i+1} = \theta_i + \eta \cdot \text{sign}\left(\frac{\partial E}{\partial \theta_i}\right), \quad (20)$$

wobei $\eta > 0$ eine ausgewählte Schrittgröße ist und die Anpassungsrichtung vom Gradienten abhängt. Mit diesem Verfahren kann man den Parameterwert schneller verbessern, aber wie die Schrittgröße auszuwählen ist, bleibt unklar.

Eine andere Modifikation des BP-Algorithmus heißt **BP-SuperSAB** [29]. Hier wird die Schrittgröße selbst angepasst. Üblicherweise verwendet man folgende Bedingung.

Wenn $E[i] < E[i-1]$, dann $\eta = \eta \cdot k$ und $k > 1$. Dies bedeutet, dass mit einer erfolgreichen Iteration der Änderungsschritt vergrößert wird. Im Gegensatz ist $\eta = \eta \cdot k$ und $k \ll 1$, wenn $E[i] > E[i-1]$. Für den Parameterschritt verwendet man die Gleichung (16). Oft wird an Stelle eines Multiplikationsgesetzes für das Wachstum der Koeffizient η ein Additionsgesetz wie $\eta = \eta + \Delta$ und $\Delta > 0$ angewendet. Dann sieht das gesamte Anpassungsgesetz für η wie folgt aus:

$$\begin{cases} \text{Falls } E[i] < E[i-1] \text{ , dann } \eta = \eta + \Delta \text{ und } \Delta > 0; \\ \text{Falls } E[i] > E[i-1] \text{ , dann } \eta = \eta - P\Delta \text{ und } \Delta < 0, P \gg 1; \end{cases}$$

Diese Grundidee kann auch mit anderen Gesetzen der Schrittänderung realisiert werden. Der **BP-SuperSAB**-Algorithmus funktioniert viel schneller als der BP-Algorithmus, jedoch bleibt dabei die Frage offen, wie k , Δ und P zu wählen sind.

Die nächste Gruppe der Methoden verwendet **konjugierte Gradienten** [30]. Das Hauptprinzip basiert auf der Verwendung des laufenden und vorhergehenden Gradienten. Sehr verbreitet unter ihnen ist der **Hestenes-Stiefel** (1952)-Verfahren, das ein Anpassungsgesetz in der Form hat:

$$\theta_{i+1} = \theta_i + \eta \left(\frac{\partial E}{\partial \theta_i} + \alpha_i \frac{\partial E}{\partial \theta_{i-1}} \right);$$

$$\alpha_i = \frac{\frac{\partial E}{\partial \theta_i} \left(\frac{\partial E}{\partial \theta_i} - \frac{\partial E}{\partial \theta_{i-1}} \right)^T}{\frac{\partial E}{\partial \theta_{i-1}} \left(\frac{\partial E}{\partial \theta_i} - \frac{\partial E}{\partial \theta_{i-1}} \right)^T}.$$
(21)

Leider besitzt dieses Verfahren nur für lineare Abhängigkeiten $E(\theta)$ eine gute Konvergenz. **Fletcher-Reeves**(1964) hat dieses Verfahren verbessert und die Formel

$$\alpha_i = \frac{\frac{\partial E}{\partial \theta_i} \left(\frac{\partial E}{\partial \theta_i} \right)^T}{\frac{\partial E}{\partial \theta_{i-1}} \left(\frac{\partial E}{\partial \theta_{i-1}} \right)^T}.$$
(22)

für α_i entwickelt. Der Nachteil dieser Methode ist eine schlechte Konvergenz wegen der kleinen numerischen Genauigkeit. Deswegen sollte der Koeffizient η in der Praxis mit speziellen eindimensionalen Suchmethoden wie dem **goldenen Schnitt**, oder dem **Halbeschnitt** ergänzt werden. Diese Lösung wurde von **Polak-Ribiere** (1969) entwickelt. Der Koeffizient α_i lässt sich mit

$$\alpha_i = \frac{\frac{\partial E}{\partial \theta_i} \left(\frac{\partial E}{\partial \theta_i} - \frac{\partial E}{\partial \theta_{i-1}} \right)^T}{\frac{\partial E}{\partial \theta_{i-1}} \left(\frac{\partial E}{\partial \theta_{i-1}} \right)^T}$$
(23)

definieren. Die Optimierungsergebnisse bekommt man fast so schnell wie mit der **Quasi-Newton-Methode**, die unter beschrieben wird.

Noch bessere Ergebnisse treten bei der BP-Methode, unter Beachtung der Empfindlichkeit des Gütekriteriums zur Anpassung des Koeffizienten θ , auf. Dies nennt man **BP-Sensitivity-Verfahren** [31]. Um diese Methode anwenden zu können, muss zunächst die Empfindlichkeit berechnet werden. Dafür wird die Ableitung des Gütekriteriums unter Verwendung der Gleichungen (1 und 15) gebildet und

$$S_\theta = \frac{\partial E}{\partial \phi(x)} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial \phi(x)} = e\theta$$
(24)

mit dem Signal des entsprechenden Koeffizienten θ multipliziert. Danach werden die Änderungen für dementsprechendes Signal berechnet.

$$\tilde{\phi}_{i+1}(x) = \tilde{\phi}_i(x) + \rho_i \frac{E}{\|S_\theta\|^2} S_\theta;$$

$$\begin{cases} \text{Falls } E[i+1] < E[i], \text{ dann } \rho_{i+1} > \rho_i; \\ \text{Falls } E[i+1] > E[i], \text{ dann } \rho_{i+1} \ll \rho_i; \end{cases}$$
(25)

Ähnlich wie in der Gleichung (16) ist hier der Schrittparameter $\rho > 0$. Das Anpassungsgesetz für θ , unter Verwendung der Gleichung (15,16), lässt sich mit

$$\theta_{i+1} = \theta_i + \eta e \phi(x) = \theta_i + \eta e \tilde{\phi}_i(x), \quad (26)$$

charakterisieren.

Die Empfindlichkeit des Gütekriterium E zum Koeffizienten θ kann auch mit anderen Anpassungsgesetzen kombiniert werden, weil diese, nach dem so genannten Jacobi-Kriterium, vom stetigen Signal $\phi(x)$ abhängt. Dieses Verfahren hat sich als schnellste Methode für Systeme erster Ordnung erwiesen.

Alle betrachteten Anpassungsgesetze nutzen nur die erste Ableitung, um die Komplexität der Fehleroberfläche zu berücksichtigen. Die nächste große Gruppe beinhaltet **Quasi-Newton-Methoden**, die auf dem **Newton-Verfahren** beruhen.

Grundsätzlich basiert die Idee auf einer Zerlegung des Gütekriteriums in eine Taylor Reihe. Die Newton Formel

$$\theta_{i+1} = \theta_i + \eta H^{-1} J^T e, \quad (27)$$

besteht aus der Jacobi-Matrix $J = \frac{\partial f(\theta)}{\partial \theta}$ und der Hessematrix $H = \frac{\partial^2 f(\theta)}{\partial \theta^2}$, die eine Matrize aus

den zweiten Ableitungen der Gütefunktion nach ihren Argumenten ist. Die Besonderheit von diesem Anpassungsgesetz ist ihre außergewöhnliche Konvergenzgeschwindigkeit. Leider ist dieses Verfahren nur für einfache Fälle anwendbar.

Das größte Problem stellt die Berechnung der Hessematrix dar, weil sie nicht immer invertierbar ist. Tatsächlich kann die Hessematrix sehr kleine Zahlen beinhalten, was bei numerischer Berechnung der inversen Hessematrix zu starken Probleme führt.

Dies verursacht Impulse in der **Quasi-Newton-Methoden**, die indirekt auf die Berechnung der Hessematrix einwirken. Als erster Lösungsansatz zur Bestimmung der Hessematrix wird das **Gauß-Newton-Verfahren** angewendet. Eine Annäherung der Hessematrix bekommt man aus der Jacobi-Matrize mit $H \approx J^T J$. Leider kommt es bei der direkten Implementierung des Gauß-Newton-Verfahrens zu denselben Problemen wie bei der Invertierung der Hessematrix. Als nächstes wird der **BFGS** (Broyden, Fletcher, Goldfarb, and Shanno) Algorithmus betrachtet.

Der Anpassungsgesetz schreibt man in diesem Fall als:

$$H_{i+1} = H_i + \frac{g_i g_i^T}{g_i^T s_i} - \frac{H_i^T s_i s_i^T H_i}{s_i^T H_i s_i} \quad (28)$$

Wobei $g_i = \left(\frac{\partial E}{\partial \theta_i} - \frac{\partial E}{\partial \theta_{i-1}} \right)^T$ und $s_i = \theta_i - \theta_{i-1}$ ist. Der Anpassungsschritt sieht wie in der Gleichung

(25) aus. Diese Methode muss mit einer eindimensionalen Suchmethode ergänzt werden. Es gibt einen mathematischen Beweis dafür, dass man dasselbe Ergebnis mit dem Polak-Ribiere-Verfahren erhält, welcher viel einfacher ist.

Seinerseits bietet das Verfahren von **Levenberg-Marquardt** eine andere Näherung zur Hessematrix. Die Hessematrix berechnet man wie im Gauß-Newton-Verfahren, das Resultat wird hingegen mit $H = (J^T J + \mu I)$ ergänzt, um die Singularität zu vermeiden. Hier ist $1 \gg \mu > 0$ ein Koeffizient, der auf die Konvergenz der gesamten Methode einwirkt. Mit $\mu = 0$ erhält man das Gauß-Newton-Verfahren und mit $\mu = 1$ das BP-Verfahren. Kommt es zu vielen Iterationen bei diesem Verfahren wird μ geregelt. Dieser Koeffizient muss sich bei einem erfolgreichen Schritt verkleinern und bei anwachsendem Fehler vergrößern.

Das Anpassungsgesetz wird mit Gleichung (25) berechnet. Die Möglichkeit, die Speicheranforderung dieses Verfahrens zu senken, ist ein weiterer Vorteil. Das ist durch die Zerlegung der Jacobi-Matrix in zwei Teile mit $H \approx J^T J = \begin{bmatrix} J_1^T & J_2^T \end{bmatrix} \cdot \begin{bmatrix} J_1 \\ J_2 \end{bmatrix} = J_1^T J_1 + J_2^T J_2$ möglich.

In der Praxis liefert das Levenberg-Marquardt-Verfahren ein besonders gutes Konvergenzverhalten und benötigt hierbei einen mittleren Berechnungsaufwand.

Zur Untersuchung der genannten Algorithmen wurde das Approximationsmodelle ANFIS ausgewählt. Es hat zwei Eingänge und einen Ausgang. Als Zugehörigkeitsfunktionen wurden drei Gaußfunktionen pro Eingang verwendet. Alle Koeffizienten des Netzes wurden in dem Lernprozess geändert. Als Testaufgabe wurden Approximationen der Sinusfunktion $f = \sin(x), x \in [-\pi, \pi]$ und die Aufgabe der Mustererkennung gewählt. Im letzten Fall bekommt das Netz „Mustersignale“, vergleicht diese mit bestimmter Mustergruppennummer und gibt am Ausgang die entsprechende Gruppennummer aus. Als Testbeispiel für Mustersignale wurden EMG Signale genommen. Bei allen Versuchen wurde dieselbe Fehlergröße

$$E = \frac{1}{N} \sum_i^N (y_i(x, \theta, \delta, c) - y_{id})^2 \text{ gleich } 10^{-4} \text{ erreicht.}$$

Die Untersuchungsergebnisse sind in Tabelle 3 aufgelistet. Aus der Simulation folgt, dass das Levenberg-Marquardt-Verfahren die besten Ergebnisse bei Approximationsaufgaben erzielt. Bei Mustererkennungen bringt das BP-Sensitivity-Verfahren die besten Ergebnisse. Dies erklärt sich daraus, dass alle Quasi-Newton-Verfahren auf der Annahme basieren, dass das Gütekriterium eine quadratische Form besitzt. Bei den meisten Approximationen ist dies der Fall, aber bei der Mustererkennung verläuft das Gütekriterium weder linear noch quadratisch. Die Empfindlichkeit erlaubt nur jene Parameter stark zu verändern, die einen größeren Einfluss auf den Fehlergradienten haben. Die anderen Methoden, wie konjugierte Gradienten oder BP-SuperSAB, besitzen eine mittlere Konvergenz und einen mittleren Berechnungsaufwand. Von Aufgabe zur Aufgabe sind stets unterschiedliche Wege zu untersuchen, um eine passende Methode zu finden. Auch die Hilfsalgorithmen wie der Probeschritt oder der eindimensionale Suchalgorithmen spielen eine Schlüsselrolle bei der Lösung der Optimierungsalgorithmen.

Ebenfalls ist zu beachten, dass für alle Methoden die Gradienten analytisch berechnet und verwendet wurden. Dies ist keine notwendige Bedingung, sodass die Gradienten auch numerisch ermittelt werden können. Leider hat das numerische Differenzieren Probleme mit der Genauigkeit, was eine schlechte Konvergenz verursachen kann.

Tabelle 3. Ergebnisse des ANFIS-Lernprozesses mit unterschiedlichen Lernverfahren bei Approximationen und Mustererkennungsaufgaben

Verfahren	Aufgabe Offline	Benötige Zeit (sek)	Vergleichkoeffizient (sek/sek)	Iterationen
BP	Sinus Approximation	62.4	54,2	674
	Muster	135.772	36,4	1466
MBP	Sinus Approximation	35.8	31,13	348
	Muster	13.801	3.7	134
Rprop	Sinus Approximation	27.69	24.29	296
	Muster	27.07	7.26	289

BP SuperSAB	Sinus Approximation	5.67	4.97	143
	Muster	9.698	2.6	244
BP Sensitivity	Sinus Approximation	2.33	2.02	67
	Muster	3.73	1	107
Hestenes-Stiefel	Sinus Approximation	9.64	8.46	56
	Muster	8	2.14	46
Fletcher-Reeves	Sinus Approximation	7.86	6.89	62
	Muster	6.62	1.77	52
Polak-Ribiere	Sinus Approximation	8.24	7.23	51
	Muster	5.13	1.38	31
BFGS	Sinus Approximation	5.22	4.58	26
	Muster	19.68	5.28	98
Levenberg-Marquardt	Sinus Approximation	1.15	1	12
	Muster	13.07	3.5	136

Robuste Modifikationen in Anpassungsalgorithmen

In der Praxis kann eine Lösung durch falsche Annahmen über die Form des Gütekriteriums und des Anpassungsgesetzes instabil werden. Um diesen Effekt zu vermeiden, nutzt man Schrittmodifikationen, die das Anpassungsgesetz stabiler machen und auch die Konvergenz verbessern.

Das PI-Anpassungsgesetz [25] ist eine sehr bekannte Lösung aus der Regelungstheorie. Es ist im Prinzip nicht anderes als ein PI-Regler für den Anpassungsregelkreis. Die Gleichung des PI-Gesetzes lautet:

$$\theta_{i+1} = \theta_i + \eta_p \frac{\partial E}{\partial \theta_i} + \theta_i$$

$$\Delta \theta_i = \theta_i + \eta_i \frac{\partial E}{\partial \theta_i}$$
(29)

Der Integrationsfaktor η_i ist durch Untersuchungen zu bestimmen.

Das Composite-Anpassungsgesetz ist eine Modifikation, die den Schrittfaktor anpasst und damit für eine schnelle Konvergenz sorgt. Das zweite Glied in der Schrittformel spielt dieselbe Rolle wie im **MBP**-Algorithmus, besitzt aber noch einen zusätzlichen Parameter w [35].

$$\theta_{i+1} = \theta_i + \eta_i (J^T e + J_f^T \varepsilon),$$
(30)

Wobei $\eta_i = J_f^T \cdot J_f^T$ der Schrittfaktor, $\varepsilon = J_f^T \cdot \theta_i$ der Fehlerparameter und $J_f^T = \frac{w}{J^T + w}$

für $w > 0$ die gefilterte Jacobi-Matrix sind.

Die **Begrenzung der Parameternorm** $\|\theta\|$ ist ein weiteres Verfahren, da dies ein begrenztes Wachstum des anpassenden Parameters θ gewährleistet. Das Anpassungsgesetz lässt sich folgendermaßen beschreiben:

$$\theta_{i+1} = \theta_i + \eta \frac{\partial E}{\partial \theta_i} + \left(1 - \frac{\|\theta_i\|}{\theta_{\max}}\right) \cdot h(\theta_i) \quad (31)$$

$$h(\theta_i) = \begin{cases} 0, & \text{falls } \|\theta_i\| \leq \theta_{\max} \\ 1, & \text{anderfalls} \end{cases}$$

Das Regelprinzip kann man mit einem Satz erklären: Wenn $\theta_{i+1} > \theta_{\max}$, dann $\theta_{i+1} \leq \theta_{\max}$ [36].

Aus der adaptiven Regelungstheorie ist auch die σ - **Modifikation** bekannt [37].

Dieses Verfahren beinhaltet wie beim LMS(Least-Mean-Squares)-Verfahren einen Vergessungsprozess und steigert letztendlich die Stabilität der gesamten Formel. Das Anpassungsgesetz beschreibt man mit:

$$\theta_{i+1} = \theta_i + \eta \frac{\partial E}{\partial \theta_i} - \sigma \theta_i, \quad (32)$$

wobei $\sigma > 0$, eine Konstante ist.

Ein weiteres Verfahren ist die **modifizierte σ -Modifikation**, die hilft die Größe $\sigma > 0$ automatisch auszuwählen.

$$\theta_{i+1} = \theta_i + \eta \frac{\partial E}{\partial \theta_i} - \sigma \eta_2 \theta_i$$

$$\sigma = \begin{cases} 0 & \|\theta_i\| < \alpha \\ \sigma_0 \left(\frac{\|\theta_i\|}{\alpha} - 1\right), & \alpha \leq \|\theta_i\| \leq 2\alpha \\ \sigma_0 & \|\theta_i\| > 2\alpha \end{cases} \quad (33)$$

Hierbei ist der Anstieg des sich anpassenden Parameters θ begrenzt.

Häufig reicht eine einfachere Variante, die man ε_1 -**Modifikation** nennt, aus.

Das Anpassungsgesetz

$$\theta_{i+1} = \theta_i + \eta \frac{\partial E}{\partial \theta_i} - \sigma \left\| \frac{\partial E}{\partial y_i} \right\| \theta_i, \quad (34)$$

$$\sigma > 0$$

erhält einen zweiten Teil des Schrittes, welcher von der Gradientennorm abhängt. Die Vergrößerung der Norm $\|\theta\|$ vermeidet ein zu schnelleres Anwachsen des Parameters θ [38].

Die **Schrittnormierung** ist eine weitere sehr effiziente Änderung, welche hilft die Stabilität des Anpassungsgesetzes zu gewährleisten. In diesem Fall hängt die Schrittgröße von der Größe der Parameternorm $\|\theta\|$ ab.

$$\theta_{i+1} = \theta_i + \frac{\eta \frac{\partial E}{\partial \theta_i}}{1 + \left\| \frac{\partial E}{\partial y_i} \right\|}. \quad (35)$$

Bei der Normierung können unterschiedliche Normen verwendet werden. In der Theorie nimmt man oft die Euklid Distanz, welche jedoch schwer zu berechnen ist. In einigen Fällen kann man die Module des Parameters θ ersetzen, so dass man $\|\theta\| = |\theta|$ erhält.

Ebenfalls ist zu entscheiden, welche Robustmodifikation benutzt werden sollten. Ausschlaggebend sind dabei die realen Anwendung und Anforderungen an die Berechnungen. Die robusten Modifikationen des Schrittes kann man auch in gradientenfreien Methoden verwenden, um Stabilität zu garantieren.

Gradientenfreie Optimierung

Gradientenabstiegsverfahren spielen für viele praktische Probleme eine besondere Rolle wegen ihrer Effektivität bei der Adaptation und Optimierung. Es gibt jedoch auch Optimierungsaufgaben, wo man wegen hoher Komplexität der analytischen Abhängigkeiten die Jakobie-Matrix nicht analytisch berechnen kann. Zuerst kann man versuchen, indirekt oder numerisch die Jacobi-Matrix zu beschtimmen. Alternativ gibt es auch Algorithmen die überhaupt kein Gradientenwissen benötigen.

1.6.1 Die Methode der kleinsten Quadrate

Sehr verbreitet und bekannt ist **die Methode der kleinsten Quadrate** oder (**least mean square LMS**) [32]. Sie ist auf der mittleren quadratischen Abweichung der Approximation von „Lerndaten“ begründet und hat viele Modifikationen.

Es gibt eine Variante der LMS, die ohne Iterationen mittels Matrizen eine exakte Lösung liefern kann. Dazu wird einem in der Form von Gleichung (1) aufgebauten Approximator die Aufgabe gestellt, die Lerndatei $Xd = [\bar{x}_1, \bar{x}_2 \dots \bar{x}_n]^T$, $Yd = [y_{d1}, y_{d2} \dots y_{dn}]^T$ wiederherzustellen. Mathematisch gesehen ergibt sich dann dieselbe wie Gleichung(1):

$$y_i(\theta, \phi(\bar{x}_i)) = \theta^T \cdot \phi(\bar{x}_i) = y_{di}. \quad (36)$$

Danach ist ein lineares Gleichungssystem in der Art $A \cdot \theta = Yd$ aufzubauen. Dies ist möglich, weil sich der Approximator linear zum Parameter θ verhält. Die Matrix A besteht aus numerischen Werten der Funktionen $\phi(\bar{x}_i)$. Die genaue Lösung schreibt man dann folgendermaßen:

$$\theta = \begin{bmatrix} \phi(\bar{x}_1) \\ \phi(\bar{x}_2) \\ \dots \\ \phi(\bar{x}_n) \end{bmatrix}^{-1} \cdot Yd. \quad (37)$$

Die Gleichung (29) kann auch für den besonderen Fall, dass die Dimension des Vektos θ nicht gleich der Dimension des Vektors Yd ist, alle Voraussetzung erfüllen. Dann sollte die

Invertierung der Matrix A mit der *pseudoinversen* Matrix $A_*^{-1} = A^T(AA^T)^{-1}$ durchgeführt werden.

Es gibt auch **iterative LMS** Verfahren die keine Invertierung brauchen, aber viele wiederholbare Berechnungsschritte erfordern. Die iterative Lösung wird für ein Gütekriterium in der

Form $J = \frac{1}{2} \int_0^t (Yd - \theta^T \cdot \phi(\bar{x}_i))^T (Yd - \theta^T \cdot \phi(\bar{x}_i)) dt$ gesucht.

Die Gleichung (38) Beschreibt das iterative Anpassungsgesetz,

$$\begin{aligned} \theta_{i+1} &= \theta_i - P_i \phi(x) e, \\ P_{i+1} &= P_i - P_i \phi(x) \phi(x)^T P_i \end{aligned} \quad (38)$$

wobei P_i eine Korrelationsmatrix und $P_0 > 0$ ist.

Bei schlechter Determiniertheit der Matrix P kann diese schnell entarten und das Anpassungsgesetz verliert seine Konvergenzeigenschaft. Eine Lösung liefert die Reinitialisierung der Matrix $P_0 > 0$ oder die Verwendung eines Vergessmechanismus, wie z.B. in der Gleichung (38)

$$\begin{aligned} \theta_{i+1} &= \theta_i - P_i \phi(x) e, \\ P_{i+1} &= P_i - P_i \phi(x) \phi(x)^T P_i + \mu P_i, \quad \mu > 0 \end{aligned} \quad (39)$$

bei der ein **exponentieller Vergessensprozeß** realisiert wurde. Der Name dieser Methode weist darauf hin, dass das Gütekriterium den Einfluss der älteren Daten vermindert:

$$J = \frac{1}{2} \int_0^t \exp\left(-\int_s^t \mu(\tau) d\tau\right) (Yd - \theta^T \cdot \phi(\bar{x}_i))^T (Yd - \theta^T \cdot \phi(\bar{x}_i)) ds.$$

Noch effektiver ist die **rekursive Methode der kleinsten Quadraten** (recursive least squares method - **RLS**) [39]:

$$\begin{aligned} \theta_{i+1} &= \theta_i - P_i \phi(x) e, \\ P_{i+1} &= \frac{1}{\lambda} (I - P_i \phi(x) (\lambda I + \phi(x)^T P_i \phi(x)^T)^{-1} \phi(x)^T P_i, \quad \lambda > 0 \end{aligned} \quad (40)$$

Wobei $1 > \lambda > 0$ ein Gewichtungsfaktor mit derselben Bedeutung wie μ in Gleichung (39) und I eine Einheitsmatrix sind. Der Nachteil ist hier die Invertierungsoperation.

Alle LMS-Algorithmen sind für lineare Systeme oder lineare Teile der nichtlinearen Systeme geeignet. Dennoch ist es möglich auch sie für die Anpassung der nichtlinearen Parameter zu verwenden. Sie zeigen aber dasselbe Ergebnis wie gradientenbasierte Verfahren. Für lineare Systeme (oder lineare Teile der nichtlinearen Systeme) zeigt diese Gruppe eine schnellere Konvergenz als Gradientenabstiegsverfahren.

1.6.2 Andere alternative Trainingsverfahren

Eine weitere große Gruppe der Methoden besteht aus Optimierungsalgorithmen, die nur den Wert des Gütekriteriums nutzen. Dazu gehören die Verfahren **Nelder & Mead** [40], **Hooke Jeeves** [41], **Simplex Methode** [42], [43] **und Zufällige Suche** [44].

Alle diese Methoden benutzen geometrische Vorstellungen der Gütekriteriumsoberfläche. Sehr theoretisch begründet ist das Simplex Verfahren. Es hat auch die höchste

Konvergenzgeschwindigkeit. Dennoch ist die zu lösende Aufgabe zu beachten, weil andere Methoden in einigen Fällen bessere Ergebnisse bringen können.

Ein gemeinsamer Nachteil dieser Verfahren ist ihre „Lokalität“, weil sie ebenso wie gradientenbasierte Algorithmen nur ein relativ kleines Gebiet der Gütekriteriumsoberfläche berücksichtigen.

Eine Lösung im globalen Sinn bekommt man unter Verwendung lokaler Verfahren mit unterschiedlichen Anfangsbedingungen. Außerdem gibt es spezielle Globalsuchalgorithmen. In der letzten Zeit entwickelt sich sehr schnell das Gebiet der **genetischen Algorithmen (GA)** [45]. GA arbeiten mit einer Menge von Punkten oder zulässigen Lösungen. Diese bezeichnet man auch als künstliche Individuen. Alle genetischen Algorithmen wurden auf Basis eines einfachen binärcodierten GA programmiert. GA arbeiten auf Basis künstlicher Individuen, die aus einem Chromosomensatz und einem Gütekriterium bestehen. In der Natur kann das Gütekriterium eine komplexe Größe sein, die die Anpassung des Individuums an die Umwelt beschreibt. Es kann aber auch eine einzelne Größe sein, je nachdem, was Gegenstand der Betrachtung ist. Im Zusammenhang mit GA geht man von der einfachen Annahme aus, dass ein Zusammenhang zwischen dem Chromosomensatz und dem Gütekriterium existiert und dass jede Änderung des Chromosomensatzes eine Änderung des Gütekriterium bewirkt. Liegen identische Chromosomensätze vor, so führen gleiche Änderungen an den Chromosomensätzen zu identischen Veränderungen im Gütekriterium.

Nur ein Nachteil, der hohe Berechnungsaufwand, beschränkt die Verwendung GA in Echtzeitsystemen. Wenn die Berechnungszeit nicht relevant ist und eine globale Lösung gefunden werden soll, ist der GA in der Praxis eines der besten Verfahren.

Reinforcement Learning (RL) [46], [47].

Dieses Verfahren ist eine neue Art von Lernalgorithmus, welches die biologische Adaption aufgreift und mathematisch anwendet. Es ist eine natürliche Tatsache, dass Lebewesen durch Interaktion mit ihrer Umgebung lernen. Ein hungriges Wesen z.B. sucht nach einer essbaren Sache. Das Ganze passiert ohne eine Form von Lehrer, nur aus der Motivation heraus etwas erreichen zu wollen. Abstrakt kann man die Reaktion der Umwelt auf die Interaktion eines Wesens oder eines Agenten mit Belohnung und Bestrafung beschreiben. Die Umwelt ist bei diesem Verfahren kodiert und der Agent befindet sich in einer Markow-Kette. Für jede Aktion bekommt der Agent eine Belohnung und bewegt sich, um noch mehr Belohnung zu erhalten. Wenn der Agent eine falsche Aktion macht, erfährt er eine Bestrafung.

Im Unterschied zum GA, kann man den RL-Algorithmus, wegen eines niedrigeren Berechnungsaufwandes, im Echtzeitbetrieb verwenden. Trotzdem hat der RL-Algorithmus eine schlechtere Konvergenzeigenschaft als gradientenbasierte Verfahren. Ihr unformaler Charakter und die Globalität der Lösungen überwiegen jedoch gegenüber allen Nachteilen.

Die klassischen Methoden bleiben nicht unbeachtet. Drei Verfahren der zufällige Suche sind die **GPS – (generalized pattern search)**, **MADS- (mesh adaptive direct search)** [48]-[50], und **Latin-Hypercube-Verfahren** [51]. Die drei Verfahren basieren auf der zufälligen Teilung des Gütekriteriums und darauf folgender Optimierung der Ergebnisse. Sie zeigen eine gute Konvergenzeigenschaft und eine hohe Verarbeitungsgeschwindigkeit für große Mengen anzupassender Parameter.

Beschränkte Optimierung

Oft verwendet man zur Identifikation ein physikalisches Modell, bei dem jeder Koeffizient eine bestimmte Bedeutung hat. Wenn Vorkenntnisse über den Prozess gegeben sind, kann man häufig

die Grenzen für Parameteränderungen festlegen. Eine Algorithmenanalyse ist eine Möglichkeit, um die Grenzen der Parameter bestimmen [52]. Die Grenzen haben die Form $\theta \in [\min \dots \max]$. In diesem Fall ist $\theta_0 \in [\min \dots \max]$ und $\theta_{i+1} \in [\min \dots \max]$ ausreichend.

Algorithmus 1: Gradientenauslöschung. Dies ist die einfachste Möglichkeit die Grenzen nicht zu überschreiten. Es wird ein Testschritt berechnet und falls der Parameter durch die Änderung den erlaubten Bereich verlässt, wird diese Änderung auf Null gesetzt.

Es wird mathematisch folgendermaßen beschrieben:

$$\begin{aligned} &\text{Weil } (\theta_{i+1} < \min) \text{ oder } (\theta_{i+1} > \max), \text{ dann} \\ &\Delta q_{i+1} = 0 \end{aligned} \tag{41}$$

Eine solche Arbeitsweise ist sehr einfach für die Realisierung. Ein Nachteil dieses Verfahrens ist, dass die Lösungspunkte oft an den Grenzen „hängen“ bleiben, was zur Erhöhung der benötigten Iterationsschritte führt.

Algorithmus 2: Gradientenspiegelung. Hier wird die Parameterkorrektur durch Gleichung (41) beschrieben. Die Bearbeitungsbedingungen für die Grenzen können so darzustellen werden:

$$\begin{aligned} &\text{Weil } (\theta_{i+1} < \min) \text{ oder } (\theta_{i+1} > \max), \text{ dann} \\ &\Delta \theta_{i+1} = -h \Delta \theta_i \end{aligned} \tag{42}$$

wobei h ein Spiegelungskoeffizient $\in [0..1]$ ist. Wenn $h=0$ ist, dann ist der Gradientenspiegelungs-Algorithmus gleich dem Gradientenauslöschungs-Algorithmus. Für den Fall, dass die Grenzbedingungen als Ungleichungen $g(q) < 0$ beschrieben werden, nimmt die Gradientenspiegelung mit Rücksicht auf die Grenzungleichung $g(q) < 0$ eine sehr komplizierte Form an.

Zu den Vorteilen des Algorithmus 2 gehören die einfache Realisierbarkeit für einfache Grenzen wie $\theta \in [\min \dots \max]$ und eine hohe Konvergenzgeschwindigkeit im Vergleich mit dem Gradientenauslöschung-Verfahren. Nachteilig ist, dass die Spiegelungsparameter h gewählt werden müssen.

Algorithmus 3: Lagrange-Multiplikatoren-Verfahren. Der Kern dieser Methode besteht in dem Ersetzen des Gütekriteriums durch die Funktion:

$$E(q, \lambda) = E(q) + \lambda \cdot g(q), \tag{43}$$

welche eine Summe des vorherigen Gütekriteriums und einer linearen Kombination von Ungleichungen der Grenzen $g(q)$ (oder Gleichungen, wenn man eine Lösung längs einer Kurve sucht) ist. Die Koeffizienten der Kombination λ werden Lagrange-Multiplikatoren genannt. Die benötigten Minimumsbedingungen der Funktion $E(q)$ mit Begrenzungen, die sich mit $g(q)$ beschreiben lassen, ist nachfolgend dargestellt:

$$\begin{cases} \frac{\partial F(q, \lambda)}{\partial q} = \frac{\partial E(q)}{\partial q} + \lambda \frac{\partial g(q)}{\partial q} = 0 \\ \frac{\partial F(q, \lambda)}{\partial \lambda} = g(q) = 0 \end{cases} \tag{44}$$

Eine optimale Lösung für das Gütekriterium (42) ist eine Lösung der Systeme (44), wobei $q = q_{opt}$, und $\lambda = \lambda_{opt}$ die optimalen Werte [52] sind.

Das Anpassungsgesetz lässt sich in folgender Form beschreiben:

$$\begin{pmatrix} q_{i+1} \\ \lambda_{i+1} \end{pmatrix} = q_i + \begin{pmatrix} H(q_i, \lambda_i), -A(q_i) \\ -A^T(q_i), \dots, \mathbf{0} \end{pmatrix}^{-1} \cdot \begin{pmatrix} \frac{\partial f(q_i)}{\partial q_i} - A(q_i)\lambda_i \\ g(q_i) \end{pmatrix} \quad (45)$$

wobei $A(q_i)$ ein Vektor der aktiven Begrenzungen (die in einem Testschritt überschritten werden können) und $H(q_i, \lambda_i)$ eine Hessematrix ist. Weil man das Levenberg-Marquardt-Verfahren verwendet, wurde die Hessematrix als $H(q_i, \lambda_i) = (J^T J + \mu I)^{-1}$ berechnet. Die Jacobi-Matrix lässt sich mit Berücksichtigung auf aktive Grenzen beschreiben:

$$J = \begin{pmatrix} \frac{\partial f(q_i)}{\partial q_i} - A(q_i)\lambda_i \\ g(q_i) \end{pmatrix}. \quad (46)$$

Der Algorithmus „Schritt für Schritt“ sieht folgend aus:

1. Initialisierung der q, λ , a und b Verfahrenskonstanten.

2. Berechnung der Straffunktion -Funktion $P = \left\| \frac{\partial f(q_i)}{\partial q_i} - A(q_i)\lambda_i \right\|_2^2 + \|g(q_i)\|_2^2$

3. Wenn $P < \varepsilon$, woher $\varepsilon \geq 0$, dann berechne $\begin{pmatrix} q_{i+1} \\ \lambda_{i+1} \end{pmatrix}$ und geh zum Schritt 5, sonst

4. berechne $\begin{pmatrix} q_{i+1} \\ \lambda_{i+1} \end{pmatrix}$ und wenn $P\left(\begin{pmatrix} q_i + a\Delta q_i \\ \lambda_i + a\Delta \lambda_i \end{pmatrix}\right) \leq (1 - ba)P\left(\begin{pmatrix} q_i \\ \lambda_i \end{pmatrix}\right)$, dann gehe zum Schritt 5,

sonst $a = a/4$ und wiederhole den Punkt 4.

5. Berechne nächste Annäherung $\begin{pmatrix} q_{i+1} \\ \lambda_{i+1} \end{pmatrix} = \begin{pmatrix} q_i \\ \lambda_i \end{pmatrix} + \begin{pmatrix} a\Delta q_i \\ a\Delta \lambda_i \end{pmatrix}$.

Die Wiederholungsanzahl beim Schritt 4 ist mit vernünftigen Limits zu begrenzen, sonst kommt es zu einem wesentlich höheren Berechnungsaufwand. Wenn Schritt 4 nicht häufig genug wiederholt wird, kann dies zu Grenzüberschreitungen führen.

Algorithmus 4: Schrankenfunktionsverfahren. Das Prinzip beruht ebenfalls auf dem Austausch des Gütekriteriums $E = f(q)$ mit:

$$E = f(q) + R \frac{1}{g(q)}, \quad (47)$$

wobei $R > 0$. Je näher R der Grenze kommt, desto mehr steigt das Gütekriterium an. Der Parameter $R > 0$ soll beim Erreichen der Grenze Schritt für Schritt verkleinert und zuletzt zu $\lim_{i \rightarrow \infty} R = 0$ werden.

Algorithmus 5: Schrankenfunktionsverfahren mit logarithmischen Funktionen. In diesem Fall wird das Gütekriterium mit:

$$E = f(q) - R \cdot \log(g(q)), \quad (48)$$

definiert. Im Unterschied zum Algorithmus 4 benutzt man hier logarithmische Funktionen als Schranke. Der Mechanismus funktioniert wie das vorgehende Verfahren, jedoch wächst die logarithmische Anhängigkeit bei Grenznäherung schneller an. Weiterhin ist zu beachten, dass $g(q) \neq 0$ ist.

Dateianalyseverfahren als Trainingsmittel und Architekturschätzung

Die Methode der Clusteranalyse verwendet man für die Trennung der Daten. Dabei ist der Bedarf für eine automatische Analyse groß. Sei es für die Entwicklung einer Elektrostimulation in der Medizin, die visuelle Qualitätskontrolle oder den Vergleich großer Bilddatenmengen. Das Problem bei der Entwicklung eines solchen Verfahrens ist, dass der Mensch das eigene Vorgehen bei der Bilderkennung nicht wiedergeben kann, da es sich außerhalb der eigenen Wahrnehmung abspielt. Umgekehrt bereitet die Erkennung von Zusammenhängen in mehrdimensionalen Datensätzen, die sich nicht graphisch veranschaulichen lassen, dem Menschen enorme Schwierigkeiten. Hier ist er auf computergestützte Datenanalysetechniken angewiesen, für die es keine besondere Rolle spielt, ob die Daten aus zwei- oder zwölfdimensionalen Vektoren bestehen [53].

In diesem Fall werden alle Daten als Klasse bzw. Cluster dargestellt. Die Form und Anzahl der Cluster hängen von dem Clustering-Verfahren und den Anforderungen des Entwicklers ab.

1.9 Die Linearisierungsverfahren

Heute existiert eine Reihe von mathematischen Verfahren, die für die Linearisierung von nichtlinearen dynamischen Systemen geeignet sind [54]-[56]. Hierzu die bekannte Methode des Ersetzens einer nichtlinearen Abhängigkeit mit einer linearen Funktion in einem relativ kleinen Bereich oder entlang einer Kurve. Es gibt auch eine Linearisierung durch harmonische Funktionen (harmonic linearization) und Linearisierung durch Invertierung der Funktion usw. Für alle Verfahren soll das gleiche Problem gelöst werden.

Es gibt eine Funktion $Y = F(X, U)$ und soll es eine Funktion $G(X, U)$ berechnen werden, so dass sich $Y = F(X, G(X, v)) = v$ ergibt.

Häufig verwendet man für komplexe nichtlineare Regelstrecken ein Linearisierungsverfahren mit Anwendung der Lie-Ableitungen [54]. Die Regelstrecke beschreibt man mit:

$$\begin{cases} \frac{\partial x}{\partial t} = f(x) + g(x)u \\ y = h(x) \end{cases}, \quad (49)$$

wobei $f(x), g(x), h(x)$ die nichtlinearen Funktionen vom Systemzustand x sind.

Nach der Linearisierung der Regelstrecke kann man die bekannten Verfahren verwenden, die für lineare Regelstrecken geeignet sind.

In unserem Fall müssen die Funktionen $f(x), g(x), h(x)$ eliminiert werden.

Zu diesem Zwecke werden die Gleichungen (49) in die Form von Gleichungen (50) umgewandelt.

$$\begin{cases} \frac{\partial x}{\partial t} = v \\ y = x \end{cases}, \quad (50)$$

wobei v ein äußeres Signal ist. Um das zu erreichen, muss das Eingangssignal u als:

$$u = \frac{1}{L_g L_f^{r-1} h(x)} (-L_f^r h(x) + v), \quad (51)$$

dargestellt werden, wobei r die Systemordnung ist. $L_f^r h(x)$ ist die Lie-Ableitung der Funktion $h(x)$ über der Funktion $f(x)$ und r ist die Ableitungsordnung, die gleich der Systemordnung ist. Als Ergebnis des Linearisierungsgesetz (51) bekommt man das System (50), welches sich mit linearen Verfahren regeln lässt. Im Prinzip ist es möglich dasselbe System (50) aus dem gegebenen nichtlinearen System (49) mittels algebraischer Umwandlung zu erhalten, dann wird das Linearisierungsgesetz mit:

$$u = h^{-1} \left(\frac{1}{g(x)} (v - f(x)) \right), \quad (52)$$

berechnet.

Es ist leicht zu beweisen, dass man das Linearisierungsverfahren (52) auf die Lagrange-Euler-Gleichungen für mechanische Systeme anwenden kann.

Die direkte Dynamik für mechanisches System beschreibt man wie folgt:

$$\ddot{q} = D^{-1}(q) \cdot (\tau - C(q, \dot{q})\dot{q} - G(q) - f(q, \dot{q})), \quad (53)$$

wobei q der Systemkoordinatenvektor, τ das äußere Moment, $D(q)$ die Inertia-Matrix, $C(q, \dot{q})$ die Matrix der Corioliskräfte, $G(q)$ die Matrix der Schwerkkräfte und $f(q, \dot{q})$ die Reibungskräfte sind.

Dann lässt sich aus der algebraischen Umwandlungen das Linearisierungsgesetz als:

$$\tau = D(q)v + C(q, \dot{q})\dot{q} + G(q) + f(q, \dot{q}), \quad (54)$$

definieren [57].

Hier ist deutlich erkennbar, dass das Gesetz (54) für dieses System eine inverse Dynamik ist.

Von Vorteil ist, dass das Linearisierungsgesetz (54) kein analytisches Wissen über das System benötigt und mit Anwendung der Newton-Euler-Gleichungen für die inverse Dynamik numerisch erfolgen kann[58]. Dieses Linearisierungsgesetz ist als „computed torque method“ bekannt[57], [58].

Wenn die äußere Regelung nur durch das v Signal in der Gleichung (54) wirkt, nennt man es auch „direct inverse control“. Diese Regelung ist nachteilig, weil der Regler auf die Störungen nicht direkt wirkt, sondern lediglich auf die Trägheits-Matrix multipliziert wird. Deswegen hat ein solches Regelungssystem eine schlechte Regelungsqualität, wenn eine Störkraft auf das System einwirkt. Zu Vermeidung dieses Problems wird ein weiteres Regelungssignal in das System eingefügt, welches zur Systemgleichungen addiert wird. Auf diese Weise ergibt sich das Linearisierungsgesetz in der Form:

$$\tau = D(q)v + C(q, \dot{q})\dot{q} + G(q) + f(q, \dot{q}) + v_2, \quad (55)$$

wobei v_2 das zusätzliche Regelungssignal ist. Es wird noch in Kapitel 4 erläutert.

Die für die Linearisierung notwendigen Koordinatenwerte \dot{q}, q nimmt man entweder aus den Messungen oder wählt bestimmte Werte. Den ersten Fall nennt man feedforward control und den zweiten Fall „computed torque method“.

Im nächsten Kapitel wird der Einsatz der o.g. Methoden tiefer erörtert und als Anwendung einer realen Regelstrecke dargestellt.

Neuro-Fuzzy-Netze im adaptiven Regelkreis

Neuro-Fuzzy-Systeme haben sich als gute und schnelle Approximationsverfahren erwiesen. Deswegen kann man sie erfolgreich in Regelkreisen mit Echtzeitadaption verwenden, da man die Approximation meistens als adaptiven Regler umsetzt. Dies ist jedoch nicht die einzige mögliche Variante. Neuro-Fuzzy-Netze liefern weiterhin beim Einsatz als Beobachter, als nichtlineares Prozessmodell oder als Predictor gute Ergebnisse. In allen Anwendungen heben sich Neuro-Fuzzy-Netze als besonders leistungsfähiges Verfahren hervor, durch die schnelle Adaptation, die Ähnlichkeit zu anderen Approximationstypen und die besondere Fähigkeiten zur linguistischen Deutung und Optimierung.

In Abbildung 11 erhält man eine Übersicht über Approximationsverfahren in Bezug auf Regelungssysteme.

Zuerst entscheidet man, für welchen Zweck die Approximation verwendet werden soll. Damit ist die genaue Aufgabe der Approximation im Regelkreis festgelegt. Die nächste Frage ist, welche Approximationsarchitektur benutzt werden soll. Diese bestimmt man auf Basis der exakten Forderungen des Systems und wählt die Optimierungsmethode aus. In der optimalen Steuerung spielt das Gütekriterium eine wichtige Rolle. Die Optimierung kann lokal, gruppennoptimal oder global für Trainingsmengen bzw. Optimierungsaufgaben sein. Dabei ist nach Online-, Batch- und Offline-Training zu unterscheiden.

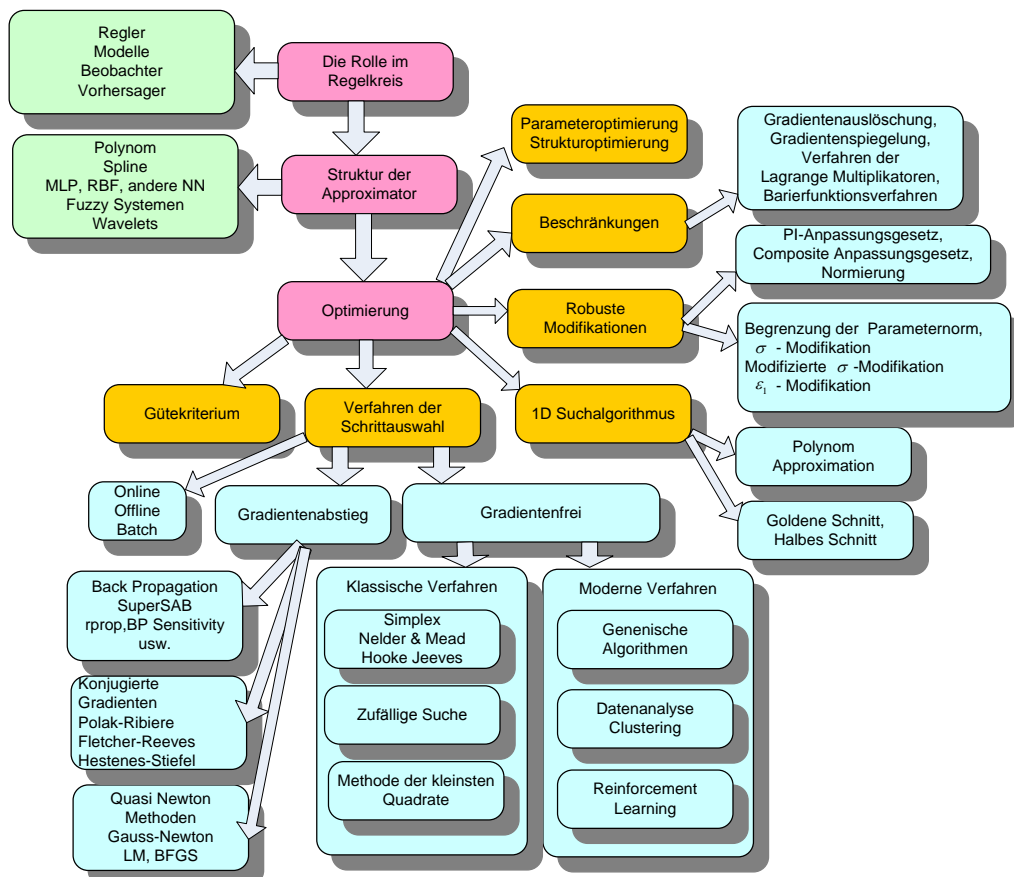


Abbildung 11. Übersicht der Approximationsverfahren in der Regelungstechnik

Anschließend ist zu entscheiden, mit welchem Verfahren die Adaptationsschritte berechnet werden. Häufig werden beschränkte Optimierungen, robuste Modifikationen, oder 1D-Suchalgorithmen verwendet.

Danach kommt man zu der praktischen Verwendung der Adaptation für die Lösung der Regelungsaufgabe im Regelkreis. Dabei müssen immer wieder der Rechenaufwand, die Anforderungen und die Möglichkeiten der ausgewählten Verfahrenkombination abgewägt werden.

Im nächsten Kapitel werden die praktischen Schwerpunkte der Anwendungen von Neuro-Fuzzy-Netzen und andere adaptiven Verfahren die für Regelung eines Magnetlagers, eines autonomen Unterwasserapparates und einer Roboterhand erörtert.

Kapitel 2. Die Regelung von Magnetlagersystemen

Einführung

Seit vielen Jahren benutzen die Menschen verschiedene Arten von Lagern. Es gibt Wälzlager und Gleitlager. Die Aufgabe von Lagern ist es auftretende Reibungskräfte bei Bewegungen zu verkleinern.

Die Nutzung von Magnetlagern in mechanischen Systemen stellt eine ganz neue Richtung des Lagerbaus dar, die mehrere Vorteile und auch eigene Nachteile aufweist. Der Hauptvorteil des Magnetlagers ist eine kontaktlose Lagerung der beweglichen Teile. Als Folge gibt es keinen Verschleiß und keine Erwärmung der Bauteile. Jedoch ist ein System mit einer Magnetlagerung im Aufbau komplexer, weil dafür ein Regelkreis mit Sensoren und mit einem Regler nötig ist, um die Funktionalität und die Bewegungsstabilität des Systems zu gewährleisten. Ein ganz wichtiger Vorteil des Magnetlagers, der alle Nachteile aufhebt, ist die Möglichkeit eine Hochgeschwindigkeitsbewegung zu realisieren, die man nur mit diesem Lagertyp erreichen kann.

Die Magnetlager spielen eine wichtige Rolle in dem modernen Lagerbau, wegen ihrer überragenden Vorteile im Vergleich mit den Wälz- oder Gleitlagern. Dazu gehören kleine Betriebskosten, eine hohe Zuverlässigkeit und niedrige Energieverluste.

In vielen Bereichen der Technik haben Magnetlager breite Anwendung gefunden. Ihr Einsatzgebiet umfasst einerseits Rotationsbewegungen, wie sie bei Zentrifugen, Frässpindeln und Pumpen und andererseits Translationsbewegungen, wie sie z.B. bei Magnetschwebbahnen auftreten. Aufgrund ihrer Eigenschaften stellen Magnetlager eine Alternative zu den bislang eingesetzten Lagerungsverfahren dar. Beispielhaft wären hier die Berührungsfreiheit der Lagerung und die damit verbundene Verschleißfreiheit sowie bei aktiven Magnetlagern die über die Reglerauslegung beeinflussbare Dämpfung und Steifigkeit zu nennen. Die Berührungsfreiheit zwischen dem gelagerten Teil und dem Lager ist dabei eine der herausragenden Eigenschaften. Die Lagerkräfte werden durch ein magnetisches Feld erzeugt, welches zwischen dem feststehenden Teil und dem zu lagernden Teil aufgebaut wird. Das ist der Grund dafür, dass sich viele Arbeitsgruppen rund um die Welt mit dem Entwurf und dem Aufbau des Magnetlagers beschäftigt haben [1]-[6].

Die Lagerung geschieht auf der Basis von Magnetkräften und ist demzufolge fast reibungsfrei. Es wird auch kein Schmiermittel benötigt. Die Steifigkeit und Dämpfung lassen sich innerhalb der physikalischen Grenzen in weiten Bereichen ändern und der Ruhepunkt des Lagers lässt sich auf elektronischem Wege verändern.

Magnetlager sind bekanntlich durch ausgeprägte Nichtlinearitäten gekennzeichnet. Ihre Regelung erfordert demzufolge eine besondere Herangehensweise unter Berücksichtigung ihrer spezifischen Eigenschaften. Robuste Regelungen erfüllen hierbei häufig nicht die geforderten Genauigkeitsanforderungen, so dass adaptive Algorithmen zum Einsatz kommen.

Das Prinzip des Magnetlagers ist einfach. Gestützt durch magnetische Kräfte kann eine Welle zwischen Magneten frei schweben. Die Lager bestehen aus Magneten. Wenn man Elektromagnete mit einem digitalen Regelsystem kombiniert, so entstehen aktive Magnetlager. Die Eigenschaften aktiver Magnetlager lassen sich abhängig von der Anwendung und den Betriebsbedingungen beliebig einstellen. Das Lager wird zu einem aktiven Bestandteil des übergeordneten Systems.

Ein aktives Magnetlager besteht aus mechanischen und elektronischen Komponenten. Mechanische Teile sind der Anker und der Stator, vergleichbar wie bei einem Elektromotor. Werden die Wicklungen, die den Eisenkern des Stators umgeben, von einem elektrischen Strom durchflossen, so entsteht ein magnetisches Feld. Das Magnetfeld durchsetzt den Eisenkern und zieht ferromagnetische Körper an, die sich in der Nähe befinden. Eine Auslenkung vom Arbeitspunkt bewirkt keine Rückstellkraft. Physikalisch wird das instabile

Verhalten durch den Zusammenhang von Magnetkraft F_m und Luftspalt beschrieben. Die axiale Kraft des Elektromagneten F_m hängt, wie Abbildung 12 zeigt, quadratisch vom Strom ab und ist umgekehrt proportional zum Quadrat der Luftspaltgröße.

Im Arbeitspunkt ist die Magnetkraft gleich der Gewichtskraft des Schwebekörpers, beide Kräfte kompensieren sich und der Körper schwebt. Eine Auslenkung von diesem Gleichgewichtspunkt hat zur Folge, dass Magnet- und Gewichtskraft nicht mehr gleich groß sind.

Vergrößert sich der Luftspalt, wird die Magnetkraft schwächer, der Schwebekörper fällt weiter ab. Erfährt der Schwebekörper dagegen eine Auslenkung in Richtung Elektromagnet, verringert sich der Luftspalt und die Magnetkraft steigt an. Somit wird der Schwebekörper weiter in Richtung des Elektromagneten gezogen. Um die Qualitäten des Magnetlagers zu nutzen, könnte ein Regler entworfen werden, der sich adaptiv an den Betriebspunkt anpasst.

Ein Regler kann diesen Effekt ausgleichen, indem er den Strom entsprechend an die neue Position anpasst, zusätzlich im Magnetlager Rückstellkräfte aufbaut und den Schwebekörper so in den Arbeitspunkt zurückführt.

Den elektronischen Teil des aktiven Magnetlagers bildet das digitale Regelungssystem.

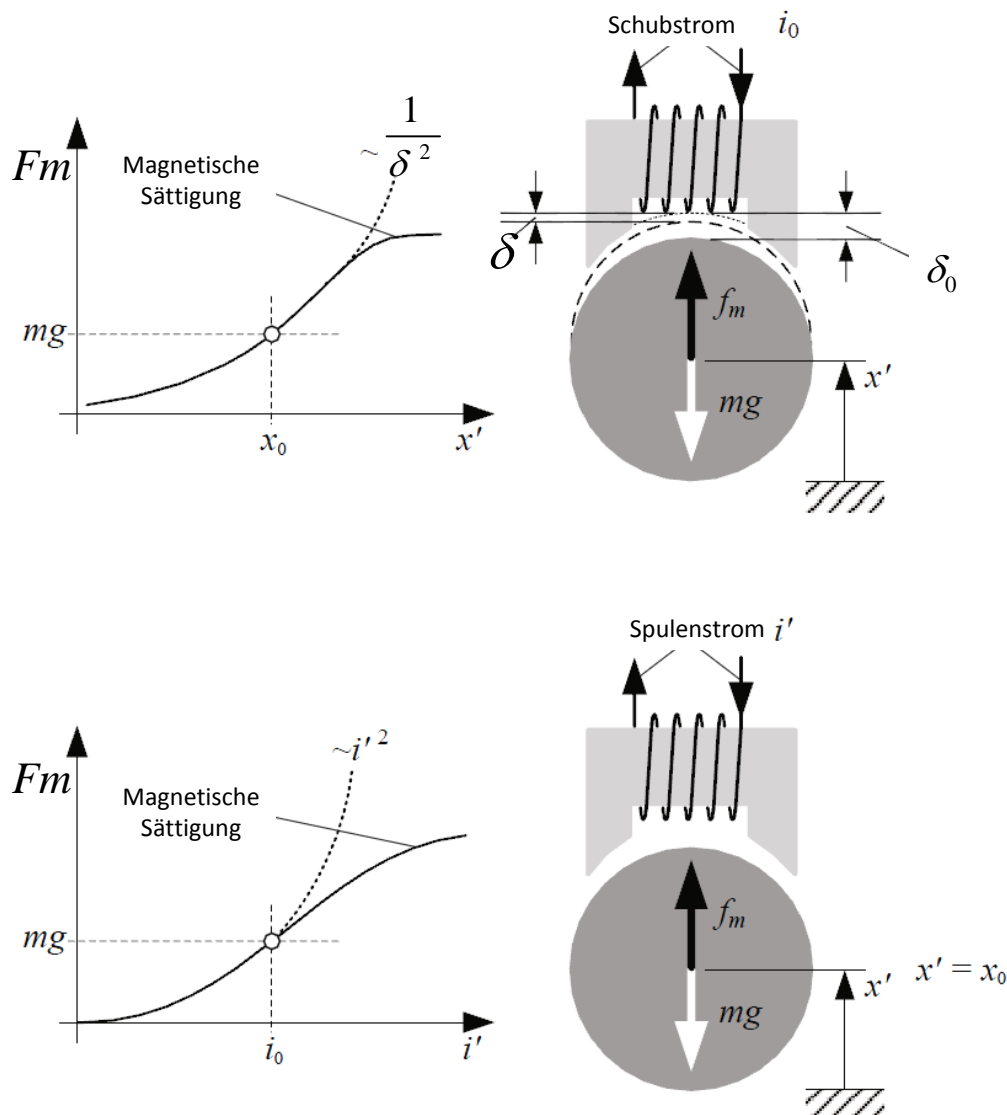


Abbildung 12. Abhängigkeit der Kraft vom Luftspalt und von Spulenstrom [7]

Die Stabilisierung von magnetisch gelagerten Körpern ist seit vielen Jahren ein bedeutendes Forschungsthema. Zur Regelung des Magnetlagers werden heute sowohl die klassische Kaskadenregelung als auch die Zustandsregelung verwendet [9]. Hierfür wurden viele verschiedene Varianten von Reglern entworfen. Sie haben verschiedene Einstellungen und Eigenschaften.

Die Anwendung des Magnetlagers in der Industrie können folgendermaßen zusammengefasst werden:

Nachteile:

- Wegen des komplexen Aufbaus ist das Magnetlager relativ teuer, deshalb ist die Herstellung noch nicht serienreif.
- Ein mit einem Magnetlager arbeitendes System hat Luftreibungs- und Wirbelstromverluste.
- Die Trägfähigkeit des Magnetlagers ist abhängig von den verwendeten ferromagnetischen Materialien, der Feldstärke, und dem Abstand zwischen den gelagerten Bauteilen.

Vorteile:

- Es besteht die Möglichkeit mit diesem Lagertyp neue Anlagen, Transportmittel und neue Werkzeugmaschinen zu entwickeln.
- Es ist eine höhere Lebensdauer des Magnetlagers wegen des fehlenden mechanischen Verschleißes zu erwarten.
- Hohe Zuverlässigkeit bei gleichzeitig niedrigem Wartungsaufwand.
- Der Energieverbrauch von magnetgelagerten Systemen ist äußerst gering.

In diesem Kapitel wird die adaptive Regelung mit klassischen Verfahren verglichen. Zu Beginn werden Systeme mit einem Freiheitsgrad dargestellt. An Hand dieser Systeme werden P-, PI-, PID- und andere Regelungskonzepte untersucht. Danach kommt der Übergang zu einer adaptiven Neuro-Fuzzy-Regelung. Anschließend werden Probleme mit Mehrfreiheitsgradsystemen analysiert. Zuerst wird mit dem Verfahren der inversen Dynamik begonnen. Danach wird die Linearisierung mittels Lie-Ableitungen betrachtet und zuletzt werden neue Approximationsverfahren zur Linearisierung des Systems behandelt.

2.1 Systembeschreibung von aktiven Magnetlagern

Das Magnetlager hat in einfachem Fall nur eine Stromspule, die Magnetkraft erzeugt. Für kompliziertere Anlagen verwendet man üblicherweise mehrere Spulen, und man kann die resultierende Magnetkraft als die Vektorsumme der einzelnen Magnetkräfte berechnen. Die Magnetenergie einer Spule:

$$W = \int_0^{\infty} L \cdot \frac{di}{dt} \cdot i \cdot dt = \frac{Li^2}{2}. \quad (1)$$

Hierfür ist es möglich die Kraft als die Ableitung der Energie nach der Positionsdivergenz $d\bar{x}$ zu bestimmen:

$$F_m = \frac{dW}{d\bar{x}}, \quad (2)$$

Die Induktivität L der Spule ist nur von der Geometrie abhängig und lässt sich durch die geometrischen Parametern beschreiben:

$$L = \frac{N^2}{R} = \mu_0 \mu_r N^2 \frac{A}{\delta}, \quad (3)$$

wobei μ_0 - die magnetische Feldkonstante , μ_r - die Permeabilität des von der Spule umschlossenen Raumes, A - ist die Fläche der Luftspalte, δ - ist die Luftspaltengröße (in Bedeutung der Positionsdifferenz), R - ist magnetische Widerstand des Magnetkreises, und N - ist die Wicklungsanzahl.

Für den Magnetkreis, welcher in der Abbildung 12 gezeigt ist, findet man die Magnetkraft aus der Gleichung (2) und (3) :

$$F_m = \frac{d}{d\delta} \left(\frac{L(Ni)^2}{2} \right) = \mu_0 \mu_r N^2 A \frac{i^2}{\delta^2} \quad (4)$$

Hieraus ist leicht zu sehen, dass der Kraftbetrag von der Strom- und Luftspaltgröße abhängt. Bei komplexen magnetischen Kreisen, muss man die Flusswege finden und entsprechende magnetische Widerstände definieren. Die Magnetlagergeometrie bringt auch eigene Besonderheiten mit sich, die auf die Induktivität $L(x)$ einwirken. Aber das Quadrat der Luftspaltgröße bleibt immer im Nenner und das Quadrat des Spulenstroms im Zähler.

Für den Stromkreis lässt sich der Marschensatz als:

$$U = Ri + \frac{d\Phi}{dt} = Ri + N \left(L \frac{di}{dt} + i \frac{dL}{dx} \frac{dx}{dt} \right) \quad (5)$$

beschreiben. In der Praxis ist es möglich den Teil $i \frac{dL}{dx} \frac{dx}{dt}$ bei „langsamen“ Bewegungen zu vernachlässigen. Die Gleichung (5) gilt nur, wenn der Magnetkreis aus Eisen und Spulen besteht. Bei Verwendung von Dauermagneten muss man auch die magnetische Spannung des Magnetes beachten.

Wie bekannt ist, besteht der gemeinsame magnetische Widerstand des magnetischen Kreises aus der Summe von Widerständen für jeden Bereich entlang eines Flussweges. Der Luftspalt hat die Permeabilität $\mu_r \approx 1$. Die Abhängigkeit zwischen der magnetischen Flussdichte B und der magnetischen Feldstärke H ist:

$$B = \mu_0 \mu_r H \quad (6)$$

Für anderen Magnetkreisbereichen, wie Eisenkern, stimmt die Gleichung (6) auch, aber $\mu_r \gg 1$. Die Permeabilität ist in diesem Fall von der Feldstärke H durch die sogenannte Hysteresekurve abhängig. In weiteren Modellen, die in dieser Arbeit näher im Betracht kommen werden, werden folgende Annahmen getroffen:

Die Bewegungen sind relativ langsam d.h. $\frac{dx}{dt} \rightarrow 0$, so dass in Gleichung (5) gilt $i \frac{dL}{dx} \frac{dx}{dt} \approx 0$.

Die Permeabilität des Eisenkerns ist eine Konstante und die Hysterese mit Sättigung und Verlusten wird vernachlässigt. Das mechanische und elektromagnetische Modell des Systems versteht man als ein Modell mit konzentrierten Parametern. Die Geometrieform wird vernachlässigt und durch den Massenmittelpunkt (Bezugspunkt) ersetzt.

Alle Abhängigkeiten sind linear. Das nichtlineare Modell des Magnetlagers wird in dem Arbeitspunktbereich linearisiert.

Unter diesen Annahmen wird eine formale Synthese der Regler wie PI, PID und Zustandsregler durchgeführt. Für andere entwickelte Regelungssysteme ist eine linearisierte und vereinfachte Modellbeschreibung nicht notwendig.

Magnetlagersysteme erlauben eine große Vielfaltigkeit der Konstruktion. Das führt zu eigenen Besonderheiten für jede Magnetkreisgeometrie. Aus diesem Grund in diesem Kapitel wurden drei unterschiedliche Magnetlagerversuchstände betrachtet.

Das Magnetlager 1 lässt sich mathematisch auf Grund seinen ausgeprägt nichthomogenes Feldlinien nicht erfassen. Es kann lediglich gefolgert werden als $F = f(i, \delta)$.

Das Magnetlager 2 hat eher homogene Feldlinie. Hier sind die Gleichungen schon anwendbar.

Für Magnetlager mit homogenen Feldbild und fünf Freiheitsgraden liegt der Unterschied in der Notwendigkeit, die Kräfte von Aktoren und die Bewegung von Schwebekörpern bezüglich des Bezugspunktkoordinatensystems oder Aktorkoordinatensystemen darzustellen.

2.1.1 Das Magnetlager mit einem Freiheitsgrad und ausgeprägt inhomogenem Magnetfeld

Den Aufbau des Betrachteten Magnetlagers zeigt Abbildung 14 und Abbildung 15.

Das Magnetlager lässt sich am besten mittels der Differentialgleichungen und Übertragungsfunktionen beschreiben [1]-[3]. Für eine Bewegung des Ankers aus der Gleichgewichtslage ist folgende Bewegungsgleichung anzusetzen:

$$m \frac{d^2 x}{dt^2} = -mg + F_m \quad (7)$$

Wobei m die Ankermasse in [kg], g die Schwerebeschleunigung in [m/s²], x die Ankerlage in [m] und F_m die Magnetkraft in [N] sind.

Die Gleichung (2) beschreibt die Magnetkraft:

$$F_m \sim c \frac{i^2}{x^2} \quad (8)$$

Wobei i der Spulenstrom in Ampere ist.

Die Konstante c errechnet sich aus dem Gleichgewichtszustand des Systems im Arbeitspunkt i_0, x_0 (vgl. Abbildung 12):

$$c \frac{i_0^2}{x_0^2} = mg \quad (9)$$

Hier sind i_0, x_0 die Werte der Variablen im Arbeitspunkt.

Mit der Taylor-Reihenentwicklung kann man unter Vernachlässigung aller höheren Reihenglieder die im Arbeitspunkt linearisierte Kraftgleichung bestimmen:

$$F_m = -\frac{2mg}{x_0} x + 2 \frac{mg}{i_0} i \quad (10)$$

Die für kleine Auslenkungen um den Arbeitspunkt gültige und linearisierte Bewegungsgleichung ergibt sich aus den Gleichungen (7) und (10) zu:

$$\frac{d^2 x}{dt^2} + \frac{2g}{x_0} x = \frac{2g}{i_0} i \quad (11)$$

Aus der Laplacetransformation resultiert die Übertragungsfunktion des linearisierten Modells:

$$G_s(s) = \frac{X(s)}{I(s)} = \frac{2g/i_0}{s^2 + 2gx_0} \quad (12)$$

Diese Gleichung entspricht einem ungedämpften Schwinger mit einem konjugiert komplexen Polpaar:

$$s_\infty^2 = -\frac{2g}{x_0} \quad (13)$$

Geht man von der eben beschriebenen Regelstrecke aus, so stellt der Elektromagnet mit vorgeschaltetem Wechselrichter das Stellglied dar. Der Stromregelkreis des Wechselrichters ist hier nicht betrachtet. Dieses muss zur Kompensation relativ kleiner Auslenkungen eine schnelle Reaktionsgeschwindigkeit aufweisen. Die Induktivität L des Stellgliedes wirkt jedoch schnellen Stromänderungen entgegen. Somit kann die Wahl eines falschen

Arbeitspunktes oder ein zu schwerer Anker zu Instabilität führen. Der Magnet wird durch folgende Differentialgleichung beschrieben, wobei R der ohmsche Widerstand der Spule ist:

$$u(t) = L \frac{di}{dt} + R \cdot i \quad (14)$$

Über die Laplacetransformation ergibt sich:

$$\frac{I(s)}{U(s)} = \frac{1/L}{s + R/L} \quad (15)$$

Dieses Glied muss mit allgemeinen Parametern angesetzt werden, da die Spule des Magnetlagers mit dem Wechselrichter nähernd als:

$$G_{sg} = \frac{I(s)}{U(s)} = \frac{K_v}{s + 1/T_v} \quad (16)$$

dargestellt werden kann.

Der Wegsensor mit Oszillator fungiert als linearer Weg-Spannungs-Umformer mit konstanter Verstärkung K_w .

Die Parameter des Versuchsstandes sind in Tabelle 4 zusammengefasst. Hierbei wurden die Vereinfachungskoeffizienten K_{ks} und K_{ki} verwendet. Alle Werte lassen sich nur in einem statischen Arbeitspunkt zu ermitteln.

Der Versuchsstand hat den Magnetpol, der zum beweglichen Teil des Magnetlagers ausgerichtet wird, und hat eine konische Form für die Konzentrierung des Magnetfeldes. Das CAD-Modell der Regelstrecke (links) und ihr Foto (rechts) wurden in Abbildung 15 dargestellt.

Tabelle 4. Die Parameter des Magnetlagers

$K_v = 63.333$	$K_w = 4000, [\text{V/mm}]$	$x_0 = 0.001, [\text{m}]$
$T_v = 0.0015, [\text{s}]$	$g = 9.81, [\text{m/s}^2]$	$i_0 = 0.293, [\text{A}]$
$m = 0.844, [\text{kg}]$	$K_{ks} = 2 \frac{gm}{x_0}, [\frac{\text{A}}{\text{V} \cdot \text{s}}]$	$K_{ki} = 2 \frac{gm}{i_0}, [\frac{\text{m} \cdot \text{kg} \cdot \text{A}}{\text{s}^2}]$

Die tatsächliche Magnetfeldlinie sieht man in Abbildung 13. Die mit Hilfe von COMSOL durchgeführte FEM-Simulation bestätigt die erwartete hohe Inhomogenität des Magnetfeldes und zeigt die Zerlegung des Feldes Inhomogenität entlang des Luftspalts.

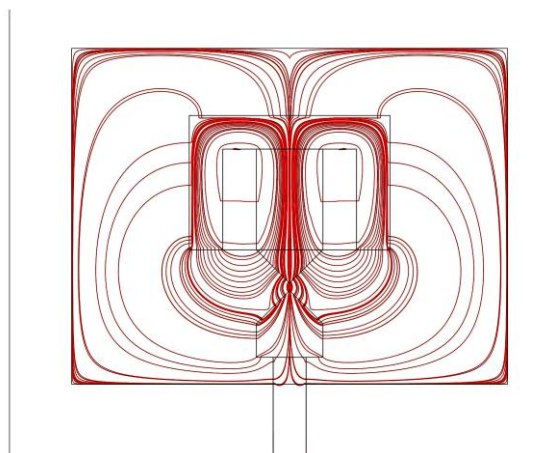
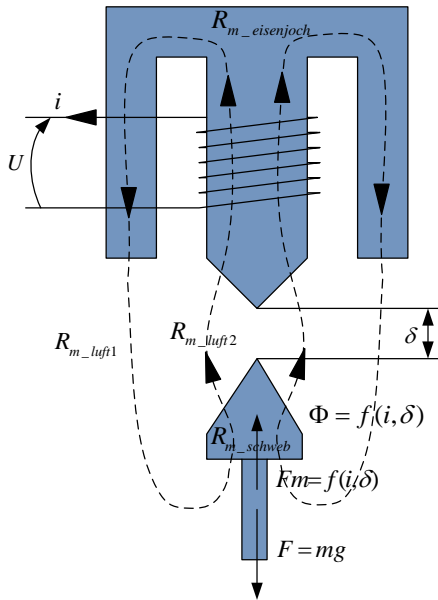


Abbildung 13 Die Feldlinien des Magnetlagers mit dem ausgeprägten inhomogenen Feld.

Bezüglich der Gleichung (4) und (8) berechnet man die Magnetkraft durch die Ableitung der Induktivität. Der Magnetkreis ist in Abbildung 14 dargestellt. Hierbei sind die magnetischen Widerstände des Magnetkreises:

$$R_{m_luft} = R_{m_luft1} + R_{m_luft2} \approx \frac{2\delta}{A\mu_0} \quad (17)$$

mit A – die Mittelquerschnittfläche des Magnetkreises.



$$R_{eisen} = R_{m_eisenjoch} + R_{schweb} \approx \frac{l_{Fe}}{A\mu_r\mu_0} \quad (18)$$

Die Induktivität berechnet man als:

$$L = \frac{N^2}{R_{\Sigma}} \approx \frac{\mu_0 AN^2}{\left(\frac{l_{Fe}}{\mu_r} - 2\delta\right)} \quad (19)$$

wobei $R_{\Sigma} = R_{luft} + R_{eisen}$.

Die magnetische Kraft lässt sich nach Gleichung (4) berechnen.

$$F_m = \frac{d}{d\delta} \left(\frac{L(Ni)^2}{2} \right) \Rightarrow$$

$$F_m \approx \mu_0 \mu_r N^2 A \frac{2i^2}{\left(\frac{l_{Fe}}{\mu_r} - 2\delta\right)^2} \quad (20)$$

Abbildung 14 Der Magnetkreis des Magnetlagers mit dem ausgeprägten inhomogenen Feld

In Abbildung 15 gelten folgende Bezeichnungen:

- a) Magnetspule der Regelstrecke, die mittels Stellglied gesteuert wurde b) Luftspalt c) Wegsensor d) Anker (der bewegliche Teil des Magnetlagers).

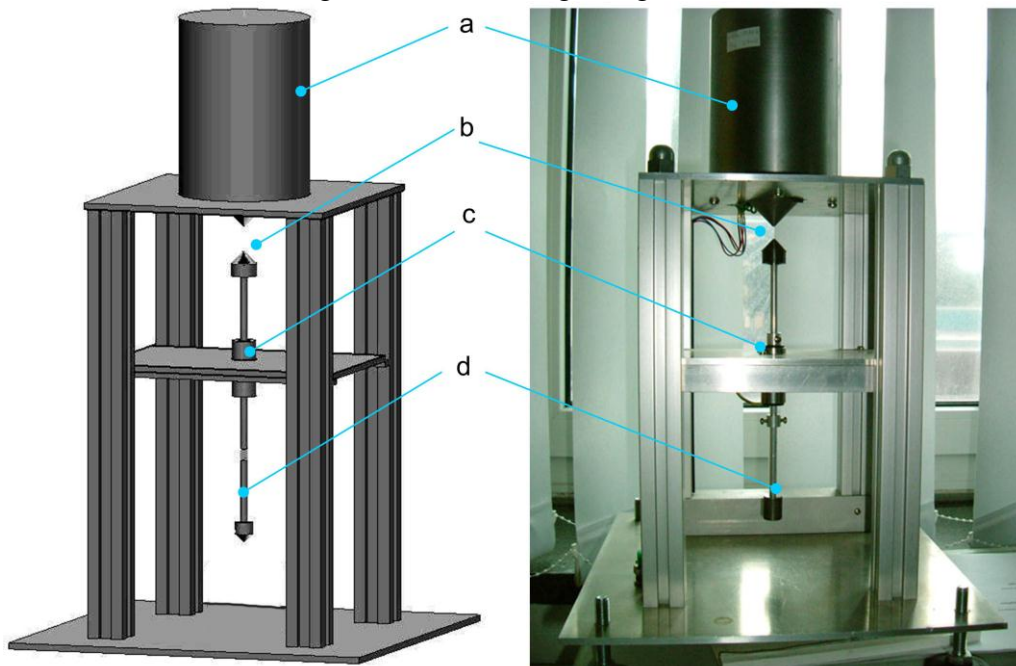


Abbildung 15. Das Magnetlager mit dem ausgeprägten inhomogenen Feld

Mit Hilfe der Gleichungen (7- 16) wurde der Signalflussplan der Regelstrecke abgeleitet, vgl. Abbildung 16.

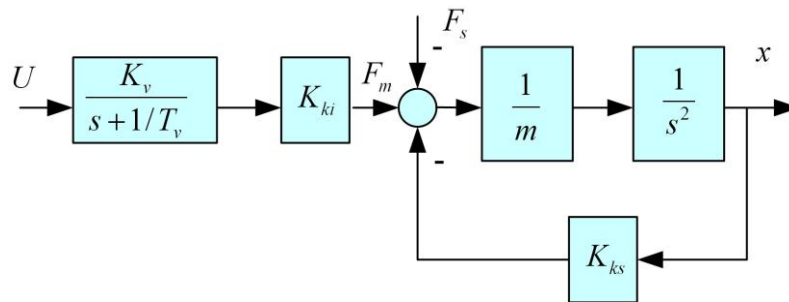


Abbildung 16. Der Signalflussplan des Magnetlagers

Die Regelstrecke besteht aus zwei Spulen und zwei spannungsgesteuerten Stromquellen, wobei je eine Spule und Stromquelle eine Einheit bilden.

Die erste Einheit setzt sich aus der Hilfsspule und der unipolaren Konstantstromquelle zusammen und wird zur Kompensation des Ankergewichtes im Arbeitspunkt benötigt. Somit muss der Regler keine Verschiebung für den Arbeitspunkt erzeugen. Die Hilfsspule besteht, wie die Regelspule, aus ca. 2600 Windungen Kupferlackdraht mit 0,63 mm Durchmesser. Beide sitzen auf dem gleichen Weicheisenjoch, wodurch eine Überlagerung von zwei Magnetkräften möglich wird. Mit einem Potentiometer stellt man das Eingangssignal von 0 V bis 10 V ein. Die Stromquelle ist so dimensioniert, dass eine Spannung von ± 10 V am Eingang einen Strom von ± 1 A am Ausgang hervorruft.

Auf den Anker wirkt die Kraft des Magnetlagers und bewirkt eine Verschiebung des Ferritkernes im Differentialtransformator. An einer zweiseitigen Aluminiumachse, in die der Ferritkern eingeschraubt ist, sitzt am oberen Ende der Weicheisenkegel mit einem axial magnetisierten Dauermagnet. Dieser bildet den Gegenpol zum Magnetfeld der Hilfsspule. Am unteren Ende der Achse ist ein Stahlzylinder angeschraubt, der für die nötige Schwerpunktverschiebung sorgt. Die Masse des Zylinders beträgt 29 g. Der Weicheisenkegel wiegt inklusive Magnet, Aluminiumachse und Ferritkern 55,4 g.

Der Wegsensor ist ein Messwertaufnehmer nach dem Differentialtransformatorprinzip. Er zeichnet sich durch gute Linearität aus und sein Messbereich beträgt $\pm 2,5$ mm. Dieser wird auf ± 10 V abgebildet.

2.1.2 Das Magnetlager mit dem homogenen Magnetfeld

Bei dem homogenen Magnetfeld liefern die Formeln (3), (4), und (8)-(13) schon fast genaue Werte. Das heißt, dass FEM Simulation schon nicht erforderlich für die Ermittlung von Parameterwerten des Magnetlagers ist.

Das Magnetlager hat in dem Arbeitspunkt (i_0, x_0) die in Tabelle 5 abgebildeten Parameter.

Tabelle 5. Die Magnetlagerparameter

$i_0 = 0.8[A]$	$x_0 = 0.75[mm]$	die Ankermasse $m = 0.58[kg]$	die Spulenwindungszahl $w = 250$
Spulenwiderstand $R = 1.77[\Omega]$	$L = 8.26[mH]$	$K_{ki} = 27.3668 \left[\frac{N}{A} \right]$	$K_{ks} = 29191 \left[\frac{N}{m} \right]$

Die bisherigen Betrachtungen gingen davon aus, dass der Wegsensor ohne Verzögerung arbeitet. Unter praktischen Bedingungen ist dies häufig nicht der Fall. Abbildung 17 zeigt die Versuchsaufbau, bei dem die Wegmessung mit einer Verzögerung von $T_{fi}=0,01\text{s}$ behaltet ist. Eine solche Zeitkonstante ist sehr groß und kann nicht vernachlässigt werden.



Abbildung 17. Der Versuchsaufbau des Magnetlagers mit dem homogenen Magnetfeld

Das Magnetlager lässt sich in Cauchy Form $\begin{cases} \dot{X} = AX + BU \\ Y = CX + DU \end{cases}$ beschreiben. Der Zustandsvektor bildet sich aus dem Sensorwert x_{fi} , Ankerposition x , und Ankergeschwindigkeit \dot{x} , daraus folgt $X = \{x_{fi}, x, \dot{x}\}$. Für die Eingangsmatrix U gilt $U = \{i\}$, wobei i - der Strom des Magnetlagers ist.

Die Systemmatrix lautet dann im Zustandsraum ohne Berücksichtigung des Stromstellgliedes:

$$A = \begin{pmatrix} -\frac{1}{T_{fi}} & \frac{1}{T_{fi}} & 0 \\ 0 & 0 & 1 \\ 0 & \frac{K_{ks}}{m} & 0 \end{pmatrix} \quad (21)$$

Die Eingangs- und die Ausgangsmatrizen besitzen die Strukturen:

$$B = \begin{pmatrix} 0 \\ 0 \\ \frac{K_{ki}}{m} \end{pmatrix}, \quad C = (1 \ 0 \ 0) \quad (22)$$

Die Durchgangsmatrix D enthält keinen Anteil des Stellgrößenvektors in dem Ausgangsvektor, demnach ist $D = 0$.

Wie man sieht, ist der Anker des Magnetlagers eine Platte in der Mitte des Versuchsaufbaus. Diese bewegt sich entlang der Führung, welche aus zwei Metallstangen besteht. Unter dem Anker des Magnetlagers befindet sich der Wegsensor. Im oberen Teil des Versuchsaufbaus wurde die Magnetspule eingebaut.

Die Gleichungen (7- 18) bilden den Signalflussplan der Regelstrecke, vgl. Abbildung 18

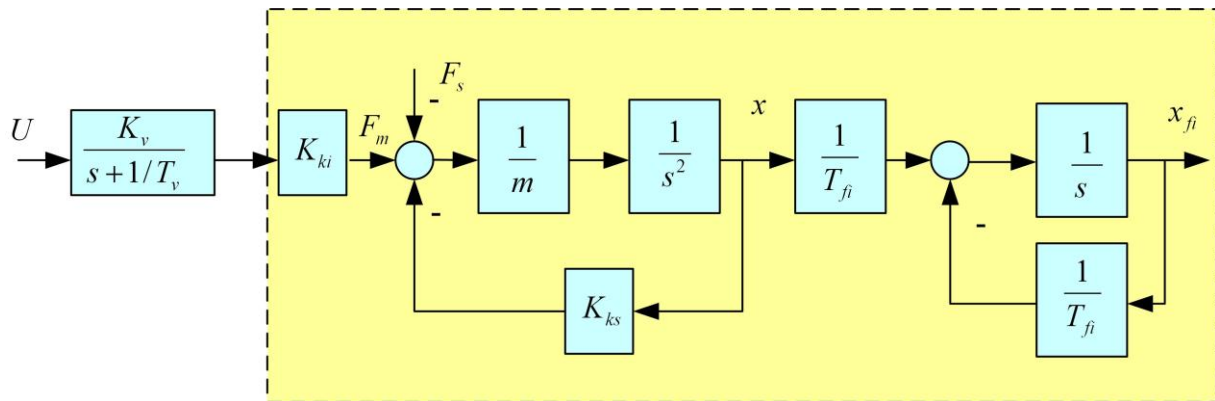


Abbildung 18. Das Magnetlager mit dem homogenen Magnetfeld

Die Gleichungen (17-18) beschreiben den gelb hinterlegten Systemteil. Die Darstellung des Stromstellgliedes als ein PT₁-Glied ist eine Vereinfachung, jedoch konnte kein messbarer Unterschied im Vergleich zu dem betrachteten System festgestellt werden.

2.1.3 Das Magnetlager mit fünf Freiheitsgraden

Die vorherigen Magnetlagersysteme waren relativ einfach, weil sie nur einen Freiheitsgrad besitzen. Für praktische Zwecke sind häufig Magnetlager mit mehreren Freiheitsgraden notwendig. Für diese Untersuchungen wurde ein Versuchsstand mit fünf Freiheitsgraden verwendet, welcher in Abbildung 19 dargestellt ist.

Die Regelung der Magnetführung, die aus den Modellgleichungen abgeleitet ist, muss in *Echtzeit* auf einem Prozessrechner durchführbar sein. Die Aufgabe des Modells besteht weiterhin darin, die Eigenschaften der Magnetführung zusammen mit unterschiedlichen Werkstücken, die von ihr getragen werden, mit hoher Genauigkeit wiederzugeben. Dies gewährleistet eine hohe Regelungsqualität.

Als mechanisches Modell für die Magnetführung ist ein Mehrkörpersystem (MKS) besonders gut geeignet. Das MKS ermöglicht die Beschreibung komplizierter Geometrien und liefert ein mathematisches Modell mit nur wenigen gewöhnlichen Differentialgleichungen. Das MKS der Magnetführung besteht aus zwei Starrkörpern, dem Führungsbett und dem Führungsschlitten, vgl. Abbildung 19.

Der Schwebezustand wird durch Aktuatorkräfte aufrechterhalten, die zwischen Bett und Schlitten wirken. Das Führungsbett kann für die Aktuatoren als feste Einspannung betrachtet werden, da es mit dem Maschinenfundament verbunden ist und dadurch eine wesentlich größere Masse im Vergleich zur Schlittenmasse besitzt.

Die gewählte Modellbeschreibung besteht aus den Blöcken der Aktoren, den mechanischen Teilsystemen und der Kopplung zwischen Aktoren und dem mechanischen System. Im Folgenden werden zunächst die Bewegungsgleichungen für das mechanische Teilsystem sowie die Gleichungen für die Aktoren beschrieben. Zusammen mit der Kopplung zwischen den beiden Teilsystemen wird anschließend das Modell des Gesamtsystems entwickelt [8].

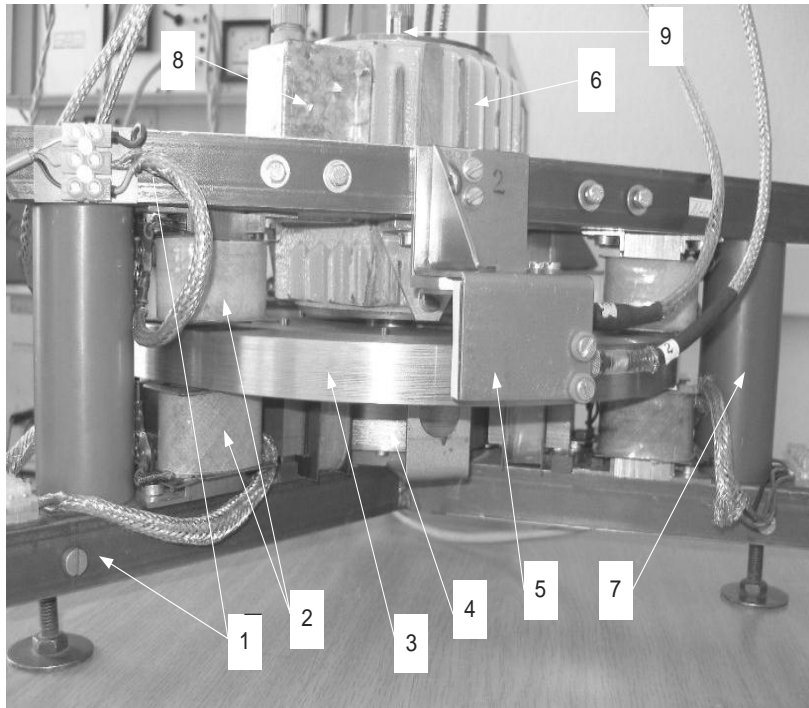


Abbildung 19. Die magnetische Anlage mit fünf Freiheitsgraden

Hier haben die Bezeichnungen folgende Bedeutung:

- | | |
|---|--|
| 1. Oberer und unterer Gerüstteil (Dreieck und Stern); | 5. Sensorhalterung mit Sensor; |
| 2. Trag- und Halte-Aktuator; | 6. Asynchronmotor (Ständer); |
| 3. Gelagerter Teil (Scheibe); | 7. Tragsäule; |
| 4. Zentrierblechpaket des gelagerten Teils; | 8. Klemmkasten des Motors; |
| | 9. Kurzschlußläufer des gelagerten Teiles. |
- Das Magnetlager besitzt einen Arbeitspunkt (i_0, s_0) mit den Parametern:

$$i_0 = 0.8[A]$$

$$x_0 = 0.75[mm];$$

der Spulenwiderstand $R = 2,3[\Omega]$;

die Induktivität $L = 50[mH]$;

die Spulenwindungszahl $\omega = 250$;

die Ankermasse $m = 7,8675[kg]$;

der Kraftstromkoeffizient $k_i = 27,3 \left[\frac{N}{A} \right]$;

der Kraftwegkoeffizient $k_s = 29191 \left[\frac{N}{m} \right]$.

Der Vektor der verallgemeinerten Kräfte ${}_B f$ des mechanischen Systems wird durch die Kopplungsgleichungen aus dem Vektor der zusammengefassten Aktuatorkräfte ${}_A F$ der n Aktuatoren gebildet. Der Weg ${}_B x$ sowie die Geschwindigkeit ${}_B \dot{x}$ der f Freiheitsgrade des mechanischen Systems wirken auf die Aktuatorwege ${}_A x$ sowie die Aktuatorgeschwindigkeiten ${}_A \dot{x}$ zurück. Die Bezeichnung „A“ deutet auf die Seite der Aktoren und „B“ auf die Seite des gewählten Bezugspunktes hin, welcher gleich dem Schwerpunkt des Ankers vom Magnetlager ist.

Die Kopplung zwischen beiden Teilsystemen wird durch die Jacobimatrizen ${}^{AB} J$ sowie ${}^{AB} J^T$ beschrieben. Diese Jacobimatrizen stellen den Zusammenhang der Aktuatorkräfte zum Bezugspunkt des Magnetlagers und den Rückkopplungskräften zum Koordinatensystem der Aktuatoren her.

Die n Aktuatoren der Magnetführung sind an das mechanische System gekoppelt, welches f Freiheitsgrade (in unserem Fall $f = 5$) besitzt. Der Vektor der verallgemeinerten Kräfte ${}_B F$ im Bezugspunkt des mechanischen Systems wird aus dem Vektor der Aktuatorkräfte ${}_A F$ gebildet. Der Weg ${}_B x$ und die Geschwindigkeit ${}_B \dot{x}$ des Bezugspunktes bewirken eine Wegauslenkung ${}_A x$ sowie eine Geschwindigkeit ${}_A \dot{x}$ bei den Aktuatoren. Die Kopplung lässt sich basierend auf dem Prinzip der virtuellen Arbeit durch die Jacobimatrizen beschreiben: Grundlage ist die $(n \times f)$ Jacobimatrix

$${}^{AB} J = \frac{d {}_A x}{d {}_B x^T}, \quad (23)$$

welche die Ableitung der Aktuator-Koordinaten ${}_A x$ nach den Bezugspunkt-Koordinaten ${}_B x$ ist. Durch sie lassen sich die Beziehungen zwischen dem Weg relativ zum Aktuator ${}_A x$ und dem Weg relativ zum Bezugspunkt ${}_B x$ angeben mit:

$${}_A x = {}^{AB} J {}_B x \quad (24)$$

Für die Beziehungen zwischen den Geschwindigkeitsvektoren der Aktuatoren und denen des Bezugspunktes gilt die analoge Beziehung:

$${}_A \dot{x} = {}^{AB} J {}_B \dot{x} \quad (25)$$

Der Zusammenhang zwischen dem Kraftvektor ${}_B F$ im Bezugspunkt und dem Vektor ${}_A F$ der zusammengefassten Aktuatorkräfte lässt sich durch die $(f \times n)$ Jacobimatrix

$${}^{BA} J = {}^{AB} J^T \quad (26)$$

beschreiben; sie ist die Transponierte der Jacobimatrix ${}^{AB} J$, da auch hier das Prinzip der virtuellen Arbeit greift. Im nächsten Schritt wird diese Matrix in der Schreibweise der Transponierten ${}^{AB} J^T$ verwendet. Die Kraftgleichung lautet somit:

$${}_B F = {}^{AB} J^T {}_A F \quad (27)$$

Die Gleichungen für die Magnetführung mit Jacobimatrizen lauten:

$${}_A \ddot{i} = \left(\frac{{}_A u}{L} - \frac{{}_A \dot{i}}{T_A} \right) \frac{1}{S}, \quad (28)$$

$${}_B \ddot{x} = {}^{AB} J^T \cdot {}_B M \cdot ({}_A \ddot{i} \cdot K_{ki} + {}_B \dot{x} \cdot {}^{AB} J \cdot K_{ks}). \quad (29)$$

Die Koeffizienten für die Massenmatrix ${}_B M^{-1}$ lauten für die *unbeladene* Magnetführung:

$$M = \begin{pmatrix} m & 0 & 0 & 0 & mr_{BCz} \\ 0 & m & 0 & -mr_{BCz} & 0 \\ 0 & 0 & m & mr_{BCz} & -r_{BCx} \\ 0 & -mr_{BCz} & mr_{BCy} & J_{xx} & J_{xy} \\ mr_{BCz} & 0 & mr_{BCx} & J_{xy} & J_{yy} \end{pmatrix} \quad (30)$$

wobei die Trägheitsmomente $J_{xx} = J_{yy} = 0.03, [kg \times m^2]$ sowie $J_{xy} = 0.01, [kg \times m^2]$ betragen und die Abstände $r_{BCx} = r_{BCy} = r_{BCz} = 0.3, [m]$ sind.

Die Jacobimatrix ${}^{AB}J = \frac{d {}_A x}{d {}_B x^T}$, die die Kopplung zwischen den Wegkoordinaten des Bezugspunktes ${}_B x$ und den Koordinaten der zusammengefassten Aktuatorwege ${}_A x$ beschreibt, lautet:

$${}^{AB}J = \begin{pmatrix} 0 & 0 & 1 & 0 & r \\ 0 & 0 & 1 & d \cos\left(\frac{\pi}{6}\right) & d \sin\left(\frac{\pi}{6}\right) \\ 0 & 0 & 1 & -d \cos\left(\frac{\pi}{6}\right) & d \sin\left(\frac{\pi}{6}\right) \\ 1 & 0 & 0 & 0 & -h \\ 0 & 1 & 0 & h & 0 \end{pmatrix} \quad (31)$$

wobei der Durchmesser des Ankers $d = 0.305[m]$ und die Hälfte des Abstandes zwischen den Vertikalkräften $h = 0.02[m]$ beträgt.

In Abbildung 20 ist der Signalflussplan des Magnetlagers mit fünf Freiheitsgraden dargestellt. Der Unterschied zu eindimensionalen Magnetlagern ist der Bezug der Messungen und Kräfte der Aktoren zum Bezugspunkt.

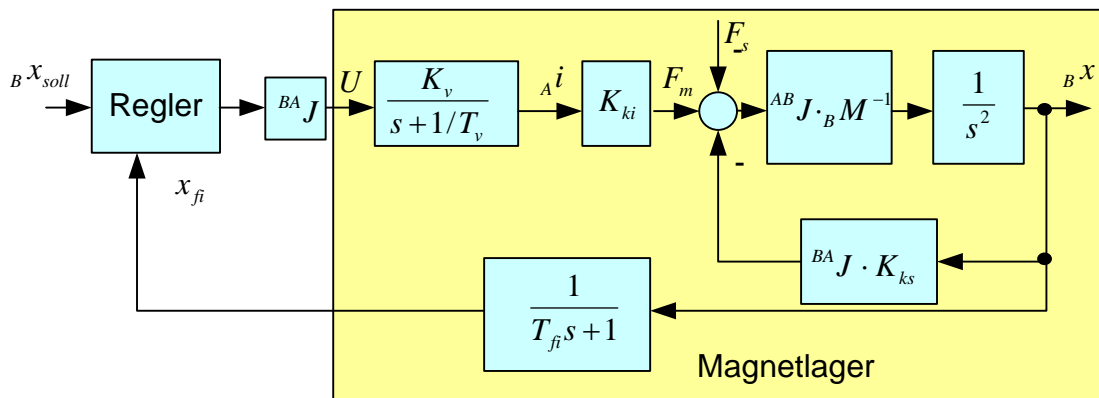


Abbildung 20. Der Signalflussplan des Magnetlagers mit fünf Freiheitsgraden

2.2 Regelungstechnische Ansätze für Magnetlager

2.2.1 PID- Regler für Magnetlager mit dem ausgeprägten inhomogenen Magnetfeld

Bei dem inhomogenen magnetischen Feld des Magnetlagers führt die Anwendung der klassischen linearen Methoden wegen der fehlenden Systemskenntnis und Nichtlinearität zu keinem befriedigenden Ergebnis. Aus der Streckenstruktur folgt, dass aufgrund der Zeitkonstantenanzahl von Stellglied und Regelstrecke ein PID-Regler eine hohe stationäre Genauigkeit gewährleisten kann. Da der D-Anteil eines solchen Reglers in der Praxis durch die verwendete Hardware einem DT-Glied entspricht, wird demnach ein PID-Regler eingesetzt. Die einzelnen Reglerübertragungsfunktionen lauten:

$$G_p(s) = K_p ; G_i(s) = \frac{K_i}{s} ; G_d(s) = K_d \frac{s}{1+T_1s} \quad (32)$$

Das entsprechende Summenergebnis zeigt die nächste Gleichung:

$$G_r(s) = K_p + \frac{K_i}{s} + K_d \frac{s}{1+T_1s} \quad (33)$$

Die vier Reglerparameter wurden vorerst experimentell auf das System abgestimmt. Für die genauere Regleranpassung wurde die Matlab/Simulink-Toolbox „NCD“ (Nonlinear-Control-Design-Blockset) verwendet. Damit lassen sich Grenzbedingungen und Grenzwerte der Sprungantwort festlegen. Durch einen Optimierungsprozess werden die jeweiligen Parameter des Modells in mehreren Schritten so geändert, dass diese Bedingungen erfüllt werden. Wie Abbildung 21 zeigt, schließt man den NCD-Output-Block am Ausgang des Simulink-Modells an, weil man hier die o.g. Bedingungen erzwingen möchte. In diesem Modell wurden die bereits erläuterten Grenzen des Stellgliedes in Form von Sättigungsblöcken berücksichtigt.

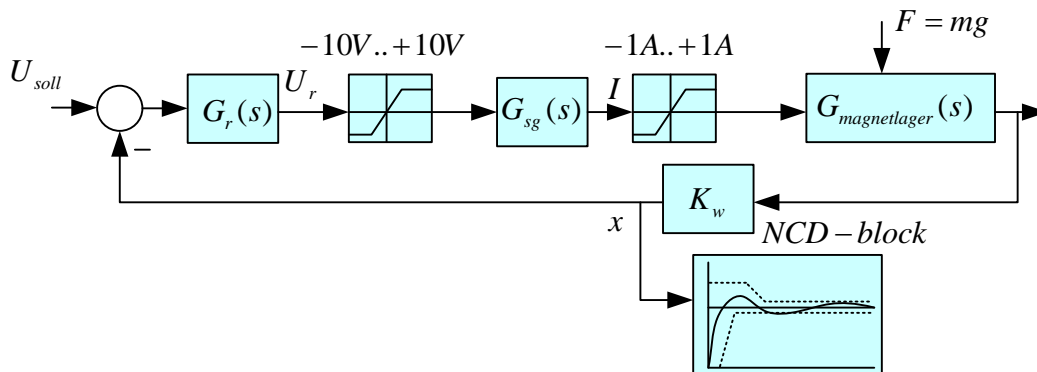


Abbildung 21. Signalflussplan des Systems mit NCD-Output-Block

Hierzu muss ein funktionierender analoger Regler ausgewählt werden.

Die optimierten Reglerparameter lauten:

$$K_p = 1,0154 ; K_i = 100 ; K_d = 0,0131 ; T_1 = 0,0001496 \text{ s.}$$

Im Folgenden soll besonders die Regelgüte dieses klassischen analog geregelten Systems für spätere Vergleichsuntersuchungen ausgewertet werden.

Die Abbildung 21 zeigt das Übergangsverhalten mit der Anregelzeit t_{an} und der Überschwingweite $h_{\ddot{u}}$. Man sieht, dass die NCD-Optimierung sehr gute Ergebnisse liefert. Abschließend wird auf das Störverhalten des Systems eingegangen.

In Abbildung 16 sieht man die Regelstrecke, die in Einzelkomponenten aufgeteilt ist. Dadurch lässt sich eine Kraft als Störgröße F_s einspeisen.

Da die Störgröße vor den großen Zeitkonstanten der Regelstrecke auftritt, kann man diese als Störung am Anfang der Strecke betrachten und damit die Realität gut abbilden. Der Kraft-Strom-Koeffizient K_{ki} in [N/A] und der Kraft-Weg-Koeffizient K_{ks} in [N/m] entsprechen jeweils den Steigungen der nichtlinearen Kennlinien im Arbeitspunkt.

Der Kraft-Weg-Koeffizient hat die physikalische Dimension einer Steifigkeit. Aufgrund des negativen Vorzeichens wird dieser Koeffizient auch als „negative“ Steifigkeit bezeichnet:

$$K_{ki} = 2 \frac{gm}{i_o} = 5.65 [N / A], K_{ks} = 2 \frac{gm}{x_o} = 1660 [N / m] \quad (34)$$

Die Abbildung 22 zeigt, wie das Magnetlager auf eine Kraftstörgröße von einem Newton reagiert.

Für die Umsetzung der Regelung an der realen Strecke findet die Soft- und Hardwarelösung Matlab/Simulink und dSPACE Anwendung. Die kleinstmögliche Abtastzeit des Systems beträgt 0,1ms. Der Ausgang des Reglers wird hierzu über einen Digital/Analog-Wandler mit dem Stellglied des Magnetlagers verbunden. Das Signal des Wegsensors wird wiederum über einen Analog/Digital-Wandler an das dSPACE-System zurückgeführt.

Bei Anwendung der durch die Simulation errechneten Reglerparameter erreicht man am Versuchsstand keinen stabilen Zustand. Ausschlaggebende Gründe sind unter anderem die bekannten Nachteile zeitinvarianter Regler wie ihr nicht optimales Verhalten bei zeitabhängigen Regelstreckenparametern. Außerdem können die Verwendung eines statischen, mathematischen Prozessmodells und die damit verbundene Linearisierung um einen festen Arbeitspunkt zu Divergenzen zwischen den Simulations- und Messergebnissen führen. Anlagenspezifische Fehlerquellen könnten die digitalen Wandler hervorrufen, die simulationstechnisch nicht ausreichend Berücksichtigung fanden. Des Weiteren ist bei dem hier verwendeten dSPACE-System das Eingangssignal des Stellgliedes auf einen Arbeitsbereich von ± 1 V begrenzt. Das eigentliche Signal wird rechnerisch auf diesen Bereich abgebildet.

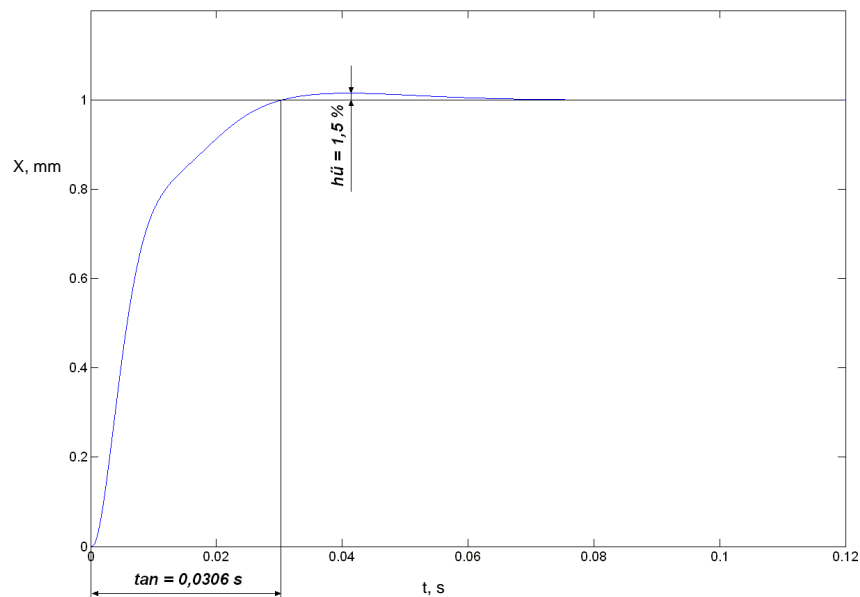


Abbildung 22. Einheits sprungantwort des geschlossenen Regelkreises

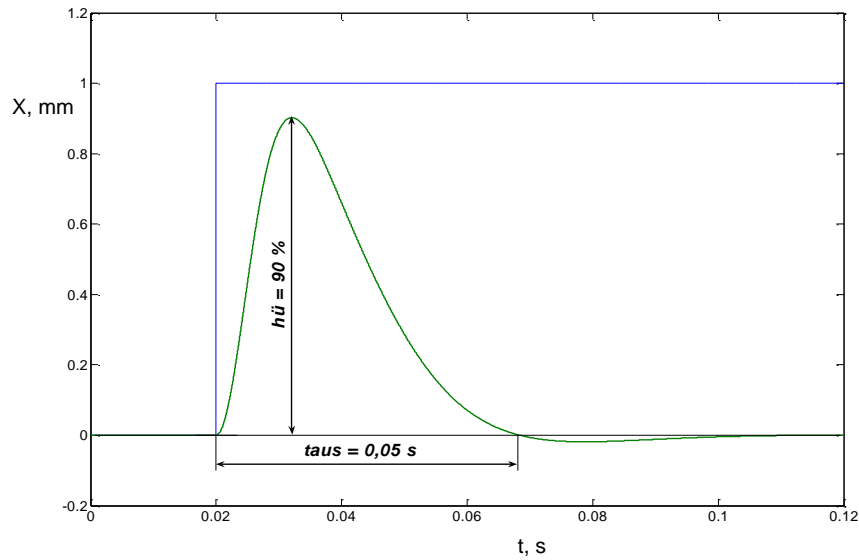


Abbildung 23. Störgrößen sprungantwort der Lageregelung

Die für das Störverhalten angepassten Reglerparameter lauten:

$$K_p = 0,75 ; K_i = 5 ; K_d = 0,03 ; T_i = 0,002 .$$

Um die Qualität dieser Regelung einzuschätzen, wurde die Sprungantwort mit der verglichen, die unter Verwendung des voreingestellten, integrierten analogen Reglers der Anlage erreicht wird. Dabei ergaben sich bei der Überschwingweite und bei der Ausregelzeit nur geringe Abweichungen.

Abbildung 24 stellt die Sprungantwort der Regelung mit dem durch dSPACE digitalisierten Regler dar. Um von Beginn an eine selbstständige Stabilisierung des Magnetlagers zu gewährleisten wurde ein Vorfilter zur Sollwertdämpfung eingesetzt. Die Übertragungsfunktion dieses Tiefpasses bei einer Abtastzeit von 0,1ms ergibt die Gleichung:

$$G_v = \frac{1}{3.5T_A s + 1} \quad (35)$$

Trotz der großen Überschwingweite arbeitet der online eingestellte PIDT₁-Regler über den gesamten Arbeitsbereich stabil. Die Stabilität ist das wichtigste Gütekriterium bei einer Regelung.

Durch die offensichtlich bestehenden Unterschiede zwischen mathematischen Streckenmodell und realer Versuchsstrecke bietet sich im Folgenden die Verwendung adaptiver Regelungsstrukturen an.

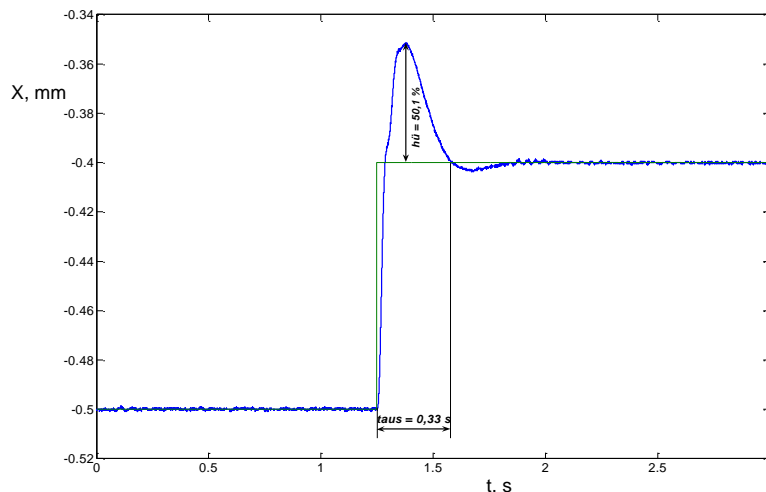


Abbildung 24. Messung der Sprungantwort vom Magnetlager

2.2.2 Zustandsregelung des Magnetlagers mit dem homogenen Magnetfeld

Am zweiten Versuchstand des Magnetlagers für die Regelung des Gesamtsystems wurden ein PI-Lageregler und ein Beobachter verwendet, denn nicht alle Zustände wie die Geschwindigkeit und der Strom der Strecke können gemessen werden.

Da der PI-Regler einen I-Anteil besitzt, vergrößert sich die Anzahl der Zustandsvariablen und ändern sich die Matrizen, die die Strecke im Zustandsraum beschreiben.

Der Signalflussplan des Gesamtsystems mit PI-Lageregler und Beobachter ist in der Abbildung 25 dargestellt. Die Regelstrecke wird als zeitinvariantes, kontinuierliches System beschrieben, auf das die Störkraft bzw. Schwerkraft $F = mg$ wirkt. Der Weg x_{sen} wird von einem Sensor gemessen und dient der Stellgrößenberechnung des Stroms i_{soll} .

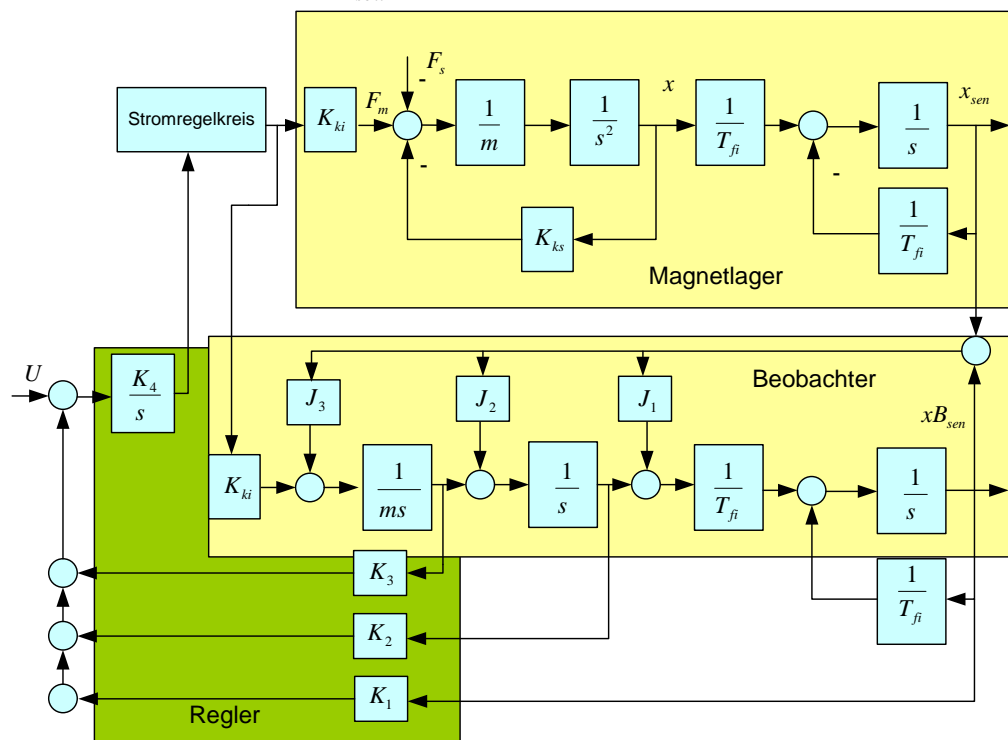


Abbildung 25. Der Signalflussplan des Magnetlagers

Der in Abbildung 25 dargestellte Stromregelkreis hat eine Struktur wie in Abbildung 26.

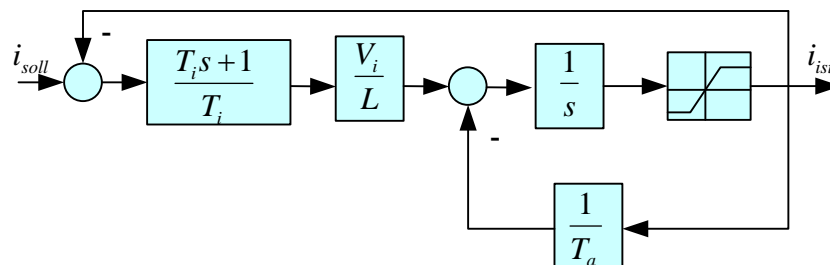


Abbildung 26. Der Signalflussplan der Stromregelkreise

Im oberen Teil von Abbildung 25 ist die Magnetlagerstrecke abgebildet. Diese besteht aus einem mechanischen Teil mit den Zuständen Weg x und Geschwindigkeit $\frac{dx}{dt}$ sowie dem Aktuaterteil mit

dem Strom als Zustand. Auf das mechanische Teilsystem wirkt die Störkraft $F = mg$. Nach dem Lagesensor ergibt sich der Integratorzustand, welcher sich mit $x_I = \int (x_{sen} - w)$ berechnen lässt. Mathematisch gesehen bildet der Beobachter die Struktur der Regelstrecke ohne den Stromregelkreis nach, weil dieser hier vernachlässigt wird. Im mittleren Teil des Blockschaltbildes ist der vollständige Beobachter dargestellt, dessen Struktur mit dem Modell der Strecke identisch ist. Er wird mit Hilfe des Rückführvektors $J = (J_1 \ J_2 \ J_3)^T$ stabilisiert, so dass der Beobachterfehler $e = x_{sen} - x_{B_{sen}}$ abklingt und die Ausgangsgröße des Beobachters $x_{B_{sen}}$ gegen die Ausgangsgröße x_{sen} der Magnetlagerstrecke konvergiert. Damit das Streckenmodell, auf dem der Beobachter aufbaut, das Verhalten der Magnetlagerstrecke gut wiedergeben kann, müssen die Parameter des Beobachters k_i, k_s und m mit denen der Magnetlagerstrecke übereinstimmen. Die Stellgröße i wird mit Hilfe der Reglerverstärkungen k_1 und k_2 aus den berechneten und gemessenen Zuständen x_I und x_{sen} sowie aus dem beobachteten Zuständen x_B und $\frac{dx_B}{dt}$ durch die Verstärkungskoeffizienten k_3 und k_4 gebildet.

Unter der Voraussetzung, dass die Koeffizienten der Beobachtermatrizen und die Koeffizienten der Regelstrecke übereinstimmen, kann das Separationsprinzip angewendet werden. Dieses Prinzip ermöglicht eine getrennte Berechnung der Rückführung des Beobachters und des Reglers.

Für die Berechnung der Regler- und Beobachterrückführung wird das Verfahren der Polzuweisung verwendet. Der Entwurfparameter besteht in der Wahl der Polstellen. Für die Praxis hat es sich als günstig herausgestellt, eine Polverteilung zu wählen, die keine Überschwingungen in der Sprungantwort des Systems erzeugt. Die Regler- und Beobachtereinstellungen, die eine hohe Steifigkeit gewährleisten, werden mit Polstellen für den Regler $p_{1,2,3,4} = -160$ und den Beobachter $p_{1,2,3} = -250$ erzielt, bei der die Wurzelverteilung nach Newtonbinom einstellt.

Die Ergebnisse der praktischen Untersuchungen bei den obengenannten Einstellungen nach der Newton-Verteilung kann man in Abbildung 27 und Abbildung 28 sehen.

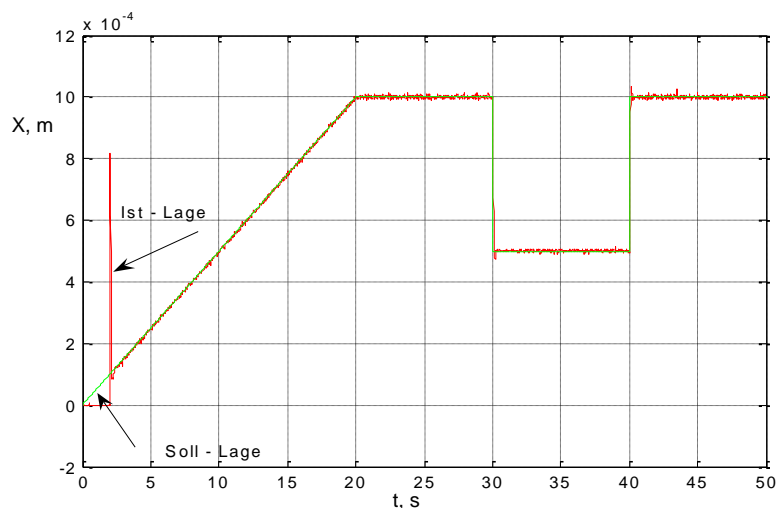


Abbildung 27. Soll- und Ist-Lage bei der Einstellung nach der Newtonverteilung (praktische Untersuchungen)

Die Simulations- und Untersuchungsergebnisse sind fast identisch. Auf den Abbildungen sind kleine zeitliche Verzögerungen bei der Ist-Lage am Anfang des Versuches zu sehen. Das wird durch die große Zeitkonstante des Lagesensors verursacht.

In den Weg- und Stromdiagrammen kann man einen großen Rauschpegel erkennen. Dieser Rauschpegel entsteht durch die Verwendung der Pulsweitenmodulatoren (PWM) für die Leistungstransistoren im Stromregelkreis, die jeweils mit einer Taktfrequenz von 50 kHz arbeiten. Dieses Rauschen kann mit den zwei Integratoren im mechanischen Teil des Magnetlagers nicht ganz geglättet werden. Deshalb führt dies zu einem Rauschen in der Sensoranzeige.

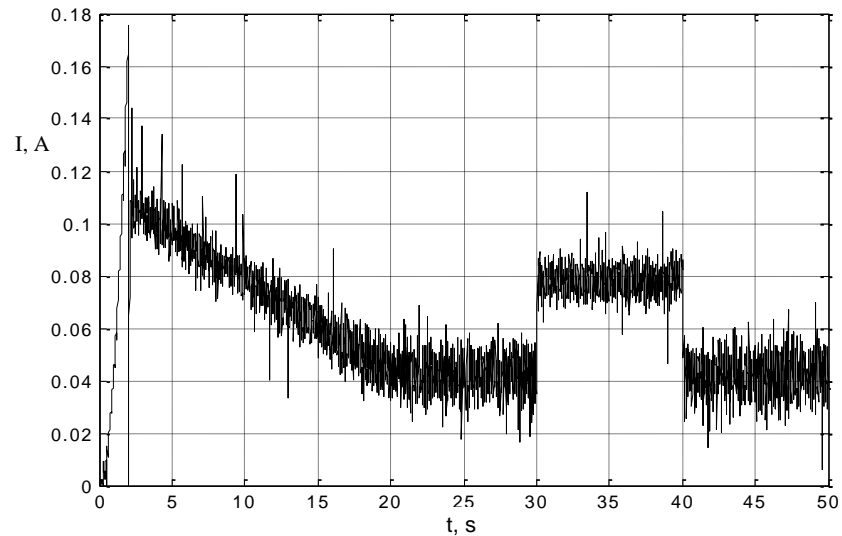


Abbildung 28. Stromsollwert bei der Einstellung nach der Newtonverteilung (praktische Untersuchungen)

Die Regler- und Beobachtereinstellung, die auch eine hohe Steifigkeit gewährleisten, werden für den Regler mit den Polstellen $p_{1,2} = -r_R \cdot \frac{\sqrt{3}}{2} + \frac{1}{2} \cdot r_R$, $p_{3,4} = -r_R \cdot \frac{\sqrt{3}}{2} - \frac{1}{2} \cdot r_R$ erreicht, wobei $r_R = 160$ der Abstand der Reglerpole zum Ursprung ist und für den Beobachter wird dies mit den Polstellen $p_{1,2,3} = -250$ erzielt, wobei sich eine Butterworth-Verteilung einstellt.

Die Ergebnisse der praktischen Untersuchungen bei den obengenannten Einstellungen nach der Butterworth-Verteilung kann man in Abbildung 29 und Abbildung 30 sehen.

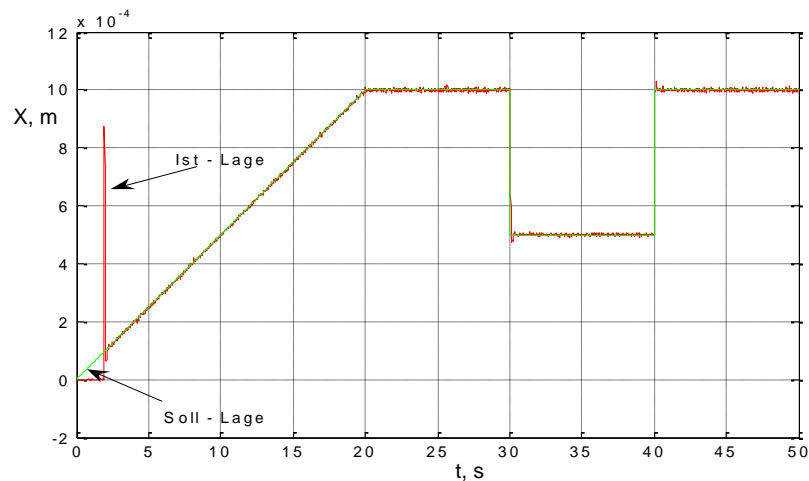


Abbildung 29. Soll- und Ist-Lage bei der Einstellung nach der Butterworth-Verteilung (praktische Untersuchungen)

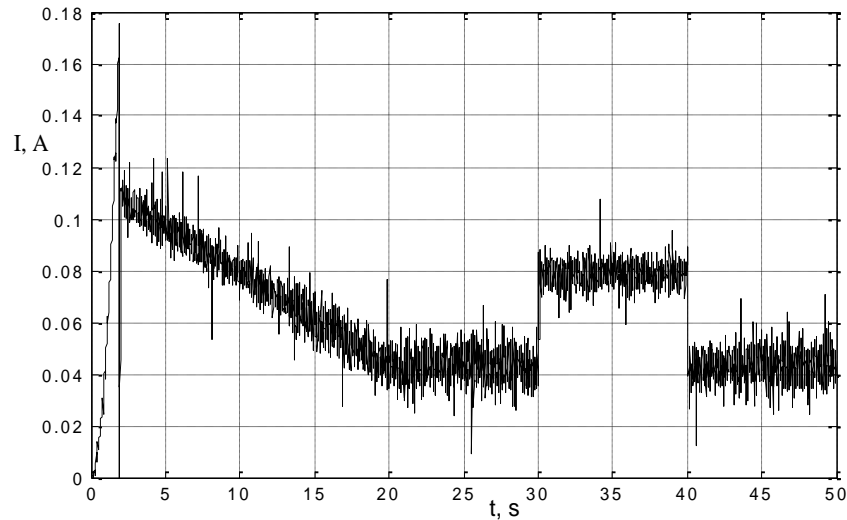


Abbildung 30. Stromsollwert bei der Einstellung nach der Butterworth-Verteilung (praktische Untersuchungen)

Aus dem Vergleich der Regelungssysteme unter Verwendung der charakteristischen Systempolynome der Newton- und Butterworth-Wurzelverteilung kommt man zum Schluss, dass die Auswahl eines bestimmten Polynoms lediglich einen Einfluss auf einen relativ kleinen Zeitabschnitt hat, der mit der Anregelzeit vergleichbar ist. Bei einer größeren Zeitdauer sind die Übergangprozesse fast gleich.

2.3 Realisierung des Neuro-Fuzzy-Netzes

Befasst man sich mit der Realisierung bestimmter Approximationsstrukturen, stellt sich die Frage wie diese umsetzbar sind. Für einfache Fälle stehen unterschiedliche Werkzeuge, wie z.B. die Fuzzy-Toolbox in MATLAB/Simulink zur Verfügung. Manchmal mangelt es diesen Werkzeugen jedoch an Funktionalität. Weiterhin ist vorher zu überlegen, ob der gewünschte Algorithmus für die gegebene Hardware verwendet werden kann. Ebenso ist es möglich eine „Open-Source-Software“ zu gebrauchen, welche jedoch weitere Probleme mit sich bringen kann.

Im Ergebnis solcher Überlegungen scheint es zweckmäßig eine eigene Software und ein eigenes adaptives Neuro-Fuzzy-Netz zu entwickeln.

Anforderungen eines solchen Systems sind:

- Flexibilität der Struktur des Neuro-Fuzzy-Netzes.
- Erweiterungsfähige Architektur für Berechnungen.
- Realisierung von Optimierungsalgorithmen für die Netzparameter.

Aus diesen Forderungen heraus ist es leichter, alles in Matrizenform zu programmieren. Die Anwendung von Matrizen minimiert die Zyklenanzahl, vereinfacht den Quellcode und ermöglicht eine kurze und klare Beschreibung. Zum Verdeutlichen wird nachfolgend das ANFIS-Neuro-Fuzzy-Netz behandelt, siehe auch Kapitel 1.

Die Eingangssignale des Netzes können mit dem Vektor $X = [x_1, x_2, \dots, x_N]$ beschrieben werden. Das bedeutet, dass ein Netz N Eingänge besitzt. Dann erhält man nach Gleichung (8) aus Kapitel 1 den Ausgangsvektor entsprechend der Eingangsschicht des NFN. Dafür wird zunächst festgelegt, wie viele

Zugehörigkeitsfunktionen jeder Eingang besitzt. Hat der Eingang einen Vektor $m = [m_1, m_2, \dots, m_N]$, dann ist die Gesamtanzahl der Zugehörigkeitsfunktionen gleich $M = \sum_{i=1}^N m_i$.

Um die Schicht der Zugehörigkeitsfunktionen mit den Eingängen zu verbinden, ist eine Verbindungsmatrix M_{con0} zu entwerfen. Die Verbindungsmatrix M_{con0} besteht aus Nullen und Einsen. Die Eingänge werden dann für eine zweite Schicht gleich $X_{2[1..M]} = X_{[1..N]} \cdot M_{con0[N..M]}$ sein. Gelangt das Eingangssignal zur nächsten Schicht, dann steht in der Matrix an entsprechender Stelle eine Eins, andernfalls eine Null. Der Aufbau der Matrix M_{con0} sieht wie folgt aus:

$$M_{con0} = \begin{pmatrix} \underbrace{1, 1, 1, 0, \dots, 0}_{m_1} & \dots & 0 \\ 0, 0, 0, \underbrace{1, 1, 0, \dots, 0}_{m_2} & \dots & 0 \\ \dots & \dots & \dots \\ 0, 0, 0, 0, \dots, \underbrace{1, 1}_{m_n} \end{pmatrix}. \quad (36)$$

Wenn man als Zugehörigkeitsfunktion eine Gaußfunktion verwendet, dann werden die Ausgänge der Schicht folgendermaßen berechnet:

$$\mu(x_j) = \exp\left(\frac{-(x_j - c_j)^2}{2\delta_j}\right) \quad \text{oder} \quad \text{in}$$

Matrizenform $\mu(X_2) = \exp\left(\frac{-(X_2 - C)}{2\Omega}\right)$. Hierbei ist die Division elementweise durchzuführen. Die Dimension der Matrix ist $D_{\mu(X_2)} = [1..M]$.

Die ANFIS-Struktur bedingt eine Knotenanzahl von $D_R = \prod_{i=1}^N m_i$ in der Regelschicht. Jeder Knoten ist für eine Kombination der Zugehörigkeitsfunktionen verantwortlich. Jeder Eingang bietet einen Platz, welcher von jeder Zugehörigkeitsfunktion eingenommen werden kann. Dies stellt eine Kombination von m Einheiten dar, die n Plätzen belegen können. Die Gesamtanzahl der Kombinationen beträgt $D_R = m^n$, weil für jeden Eingang dieselbe Anzahl von Zugehörigkeitsfunktionen angeordnet werden können. Die Verbindungsmatrix zwischen der Schicht der Zugehörigkeitsfunktionen und der Regelschicht M_{con1} hat die Dimension $[D_{\mu(X_2)}, D_R]$. Um die Ausgänge der Regelschicht zu erhalten, sind die Ausgänge der vorherigen Schicht mit der Verbindungsmatrix zu multiplizieren.

$$\phi(X)_{[1..D_R]} = \mu(X_2)_{[1..M]} \cdot M_{con1[M..D_R]}. \quad (37)$$

Die benötigte Normierungsschicht hat keinen Einfluss auf die Dimension der Vektoren:

$$\bar{\phi}(X)_{[1..D_R]} = \frac{\phi(X)_{[1..D_R]}}{\frac{1}{D_R} \sum_{i=1}^{D_R} \phi_i(x)}. \quad (38)$$

Jetzt ist die Anzahl der Netzausgänge und den Typ des ANFIS-Netzes (Sugeno 0. Ordnung oder 1. Ordnung) festzulegen. Die Ausgangsschicht besteht aus einer Matrix der Koeffizienten $\theta_{[D_R, g]}$, wobei die Anzahl der Netzausgänge gleich g ist.

Die Matrixgleichung des ANFIS-Netzes lässt sich letztendlich wie folgt beschreiben, vgl. Gleichung (8) in Kapitel 1:

$$y(X)_{[1..g]} = \left(\frac{\exp\left(\frac{-(X_{[1..N]} \cdot M_{con0[N,M]} - C_{[1..M]})}{2\Omega_{[1..M]}}\right)_{[1..M]} \cdot M_{con1[M,D_R]} }{\frac{1}{D_R} \sum \exp\left(\frac{-(X_{[1..N]} \cdot M_{con0[N,M]} - C_{[1..M]})}{2\Omega_{[1..M]}}\right)_{[1..M]} \cdot M_{con1[M,D_R]} }_{[1..D_R]} \right) \cdot \theta_{[D_R,g]} \quad (39)$$

Wie man sehen kann, ist die letzte Beschreibung kompakt, universell und kann leicht für alle Optimierungsalgorithmen verwendet werden. Die Verbindungsmatrizen erlauben auch eine Strukturanpassung, wenn man die Koeffizienten in den Verbindungsmatrizen als Variablen nimmt.

Wurde die Normierungsoperation ausgelassen, erhält man die NEFCON-Architektur. Ist M_{con1} gleich der Einheitsmatrix, so bekommt man die Wang-Mendel-Architektur. Die Matrixform ermöglicht es eine beliebige Architektur ohne Verwendung eines Zyklus zu beschreiben. Weiterhin ist es leichter, ein Neuro-Fuzzy-Netz durch die Anwendung von Matrizen in C/C++ zu programmieren.

2.3.1 Herstellung eines Neuro-Fuzzy-Netzes auf einem FPGA-Chip

Die moderne Entwicklung der Digitaltechnik wird durch die breite Anwendung der sogenannten System-On-Chip-Hardware gekennzeichnet. Als Bauelementenbasis für die Realisierung solcher Systeme werden immer öfter „Field Programmable Gate Array’s“ (FPGA) benutzt. Sie haben eine hohe Geschwindigkeit und große Anzahl der Logik- und speziellen Elemente und gleichzeitig einen relativ niedrigen Preis. Zu den wichtigen Eigenschaften von FPGAs gehört die parallele Datenverarbeitung. Diese Besonderheiten erlauben es gleichzeitig Regelungen nicht nur für eine, sondern auch für mehrere Regelstrecken zu realisieren.

Ein FPGA ist ein integrierter Schaltkreis, der programmierbare Logikbestandteile, die "Logikblöcke" und programmierbare Verbindungen enthält. Logikblöcke können programmiert werden, um die Funktion von elementaren Logikelementen wie UND und XOR oder komplizierte Combination-Funktionen wie die eines Decoders oder einfache mathematische Funktionen durchzuführen. Den größten Teil des FPGAs belegen die Logikblöcke und Speicherelemente.

Die Entwicklungsphasen einer FPGA-Anwendung sind in der Abbildung 31 dargestellt. Zuerst ist der Algorithmus in Matlab mittels „fixed point“-Berechnungen zu simulieren. Dann erfolgt die Formulierung des Algorithmus in Verilog oder VHDL. Hierfür kann man den Quellcode per Hand schreiben oder auf automatisierte Tools wie „Simulink HDL coder“ oder „Xilinx System Generator“ zurückgreifen. In beiden Fällen bekommt man den Quellcode aus einem vorformatierten Simulink-Modell.

Wenn das Programm realisierbar ist, kann es zusammen mit der Regelstrecke in MATLAB simuliert werden, um mögliche Probleme zu verhindern. Zuletzt wird der Quellcode mittels Xilinx ISE kompiliert und auf den Chip geschrieben.

Durch die Standardisierung der Architektur einerseits und die Trennung von applikationsorientierter Beschreibung sowie Hersteller-spezifischer Synthese andererseits, stehen inzwischen fertige, komplett nutzbare Elemente als Open-Source zur Verfügung. Solche Bausteine sind z.B. AD/DA-Wandler und RAM-Controller.

Der hier verwendete ANFIS-Regler wurde in Verilog programmiert. Vor der Programmierung ist es notwendig die Wirkung der Bitbreite des Zählers auszuwählen und zu bewerten. Die Bitbreite beeinflusst vor allem die Genauigkeit der Berechnungen im Modell. Deswegen wurde festgestellt, dass 18-Bits ausreichend für die zulässige Genauigkeit der Regelung sind. Es sind die ANFIS-Modelle für einen Geschwindigkeitsregler mit einer Bitbreite von 12.12 Bits (insgesamt 24 Bits:12 Bits geben die

Ganzzahl und 12 Bits die Nachkommastellen an), 18.18 Bits und 12.6 Bits zu vergleichen. Die Ergebnisse sind in der Tabelle 6 dargestellt.

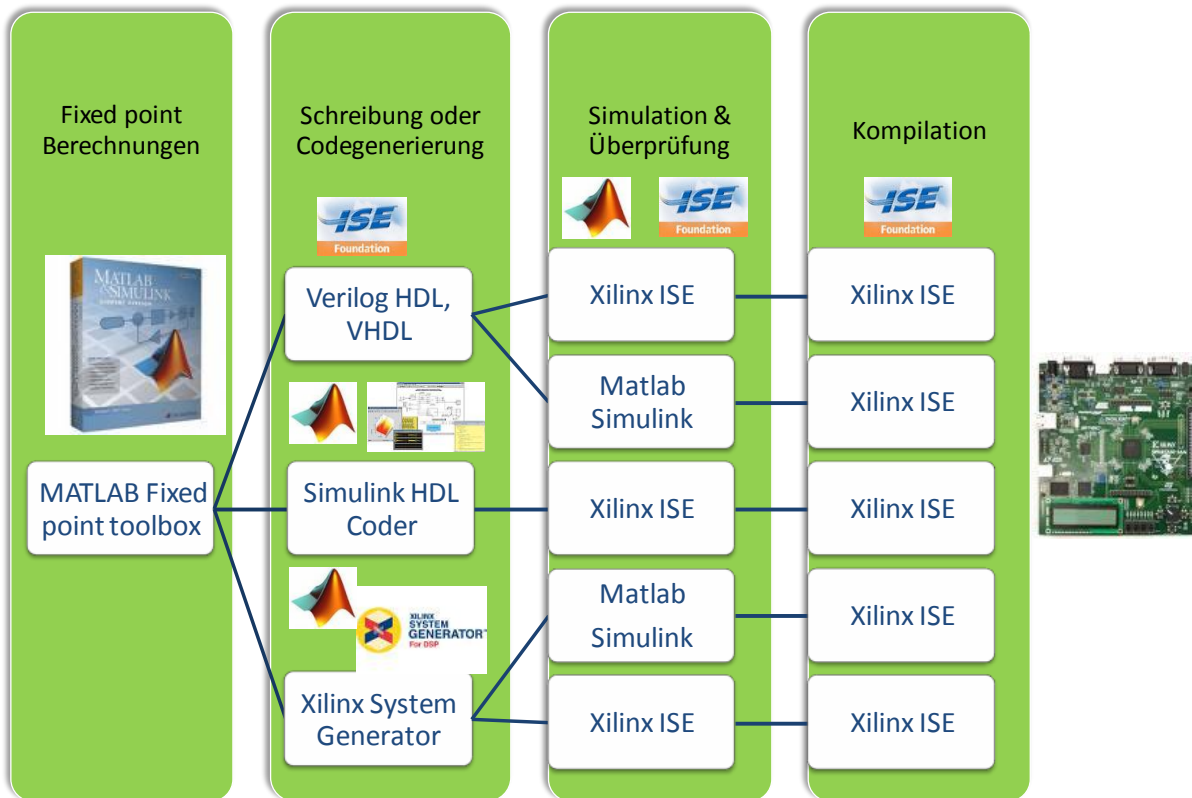


Abbildung 31. Die Entwicklungsphasen beim Entwurf eines Programms für ein FPGA

Tabelle 6: Die Auswirkung der Zahlgenauigkeit bei der Simulation des ANFIS bezogen auf den quadratischen Mittelwertfehler im Vergleich zu der Gleitkommaberechnung

Bitbreite	32 Bits (18.18)	24 Bits (12.12)	18 Bits (12.6)	18 Bits (12.5)	18 Bits (12.4)
Fehler, %	0,0005	0,05	1	3	5

Bei der Verwendung einer Bitbreite von 18 Bits für die Ganzzahl und mehr als 6 Bits für die Nachkommastellen kommt es zu einer Bereichsüberschreitung. Aus Tabelle 6 ist zu sehen, dass die optimale Konfiguration eine Fixed-Point-Darstellung mit der Aufteilung 12.6 ergibt. Eine Bitbreite von 18 Bits ist für die Regelung genau genug, da die Genauigkeit im Bereich von 2-3% liegt.

Tests haben gezeigt, dass ein FPGA „Spartan III“ mehrere ANFIS-Regler gleichzeitig berechnen kann. Dies belegt, dass Neuro-Fuzzy-Regler für Mehrachsenregelung geeignet sind.

2.4 Die adaptive Regelung mittels Neuro-Fuzzy-Netze

Eine adaptive Regelung ist meist ein spezieller Typ einer Regelung mit nichtlinearer Rückführung [11], [12]. Die Strukturen adaptiver Regler können grob in drei Klassen eingeteilt werden. Zum einen gibt es ein Regelungskonzept mit gesteuerter Adaption. Dieses Verfahren kann dann eingesetzt werden, wenn das Verhalten des Regelungssystems für unterschiedliche Parameteränderungen der Regelstrecke und Störungen bekannt ist. Hierbei gibt es keine Rückkopplung zur Korrektur der Parametereinstellungen, wodurch es sich mehr um eine Steuerung handelt. Da die Reglerparameter jedoch in Abhängigkeit von

Störungen nach einer festen Zuordnungsvorschrift angepasst werden, spricht man trotzdem von einem adaptiven Regelungssystem.

Des Weiteren redet man von Verfahren der geregelten Adaption ohne Vergleichsmodell. Hier handelt es sich um selbststellende Regler, die die schwankenden Parameter der Regelstrecke aus der Messung ihrer Ein- und Ausgangsgröße ermitteln. Dabei sollte ein explizites Modell der Regelstrecke vorhanden sein. Meist werden die Identifikation der Streckenparameter und die Anpassung des Reglers voneinander getrennt vorgenommen. Der Nachteil dieses Verfahrens ist das benötigte exakte Vorwissen über die Regelstrecke. Nur damit lässt sich eine hohe Regelgüte erreichen.

Zuletzt soll eine Struktur vorgestellt werden, die die geregelte Adaption mit Hilfe eines parallelen Referenzmodells ausführt. Hier wird ein mathematisches Modell für das gewünschte Verhalten des geschlossenen Regelkreises vorgegeben. Aus der Abweichung des Referenzmodellausganges vom Regelstreckenausgang wird die Änderung der Reglerparameter so berechnet, dass der entsprechende Fehler minimal wird. Den Einstellmechanismus entsprechend festzulegen, dass ein stabiles System entsteht, welches den Fehler gegen Null führt, ist nicht trivial. Dennoch bietet sich dieses Verfahren besonders bei nichtlinearen Regelstrecken an, da meist nur ein lineares und damit ungenaues Modell der Strecke vorliegt. Für einen entsprechenden Neuro-Fuzzy-Regler in dieser Struktur spielt es auch keine Rolle, wann oder wie Störgrößen oder spezifische Streckenparameter variieren [13].

Beim Einsatz dieser zuletzt beschriebenen Klasse sind verschiedene Abstufungen möglich.

Die Adaption kann wahlweise in jedem oder in jedem n-ten Abtastschritt online durchgeführt werden. Die Adaptionsschleife wird zum Einstellen („tunen“) des Reglers eingesetzt.

Die Adaption endet, sobald das Regelverhalten des Gesamtsystems die Anforderungen erfüllt. Dann wird die Adaptionsschleife vom Regler abgetrennt und der Regler mit konstanten Parametern weiterbetrieben. Gegebenenfalls kann bei Verschlechterung des Regelverhaltens der Adaptionsmechanismus wieder zugeschaltet und der Regler nachgestellt werden. Hierzu ist eine übergeordnete Überwachungsstruktur erforderlich, die das Regelverhalten beurteilt und die Adaption zu- beziehungsweise abschaltet. Die Adaptionsschleife wird benutzt, um für verschiedene Arbeitspunkte optimale Reglereinstellungen zu ermitteln.

An dieser Stelle muss bedacht werden, dass wegen der Komplexität eines adaptiven Reglers immer erst geprüft werden sollte, ob das regelungstechnische Problem nicht auch mit einem zeitinvarianten Regler gelöst werden kann.

2.4.1 Direkt adaptive Neuro-Fuzzy-Regelung des Magnetlagers mit einem Freiheitsgrad und dem ausgeprägten inhomogenen Magnetfeld

Abbildung 32 zeigt die Struktur der geregelten Adaption mit parallelem Vergleichsmodell. Dieses stellt dynamisch betrachtet einen Tiefpass dar, wodurch der Verlauf des Fehlers stetig differenzierbar bleibt und damit eine höhere Regelgüte erreicht werden kann.

Der Nachteil dieses Verfahrens ist, dass der Fehler am Streckenausgang berechnet beziehungsweise ausgewertet wird. Da aber das Magnetlager ohne Stellglied bereits zwei Integratoren besitzt, wäre es sehr aufwendig und komplex den Fehler am Ausgang des Neuro-Fuzzy-Netz-Reglers (NFN-Regler) zu berechnen. Deswegen würde ein solches System auch eher zur Instabilität neigen. Es muss demnach versucht werden, den Nachteil der späteren Fehlerrückführung mit einem schnellen und geeigneten Adaptionalgorithmus möglichst klein zu halten.

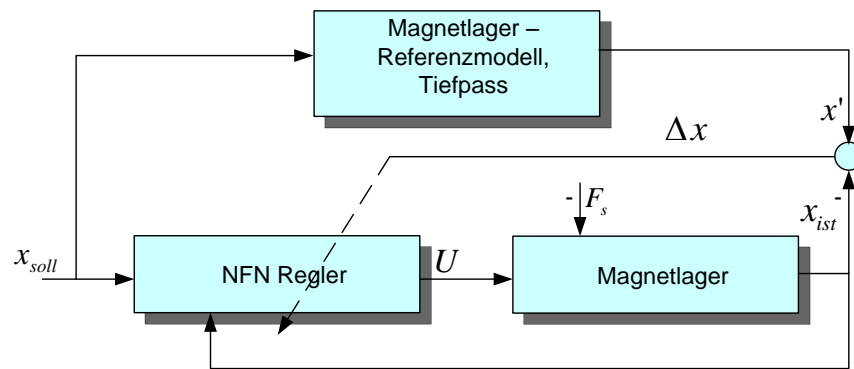


Abbildung 32. Direkte adaptive Regulationsstruktur mit Referenzmodell

Das im Folgenden verwendete und bewährte Referenzmodell der direkten adaptiven Regulationsstruktur mit $T_r=0,005$ s lautet:

$$G_{ref}(s) = \frac{1}{T_r s + 1} \quad (40)$$

Der Neuro-Fuzzy-Regler kann unterschiedliche Reglertypen erweitern wie z.B. PD-, PID- oder Zustandsregler. Die direkt adaptive Regelung funktioniert nur als ein Mittel, das wegen der Wirkung seiner Anpassungsverfahren auf die Reglerparameter die Regelgüte verbessern kann.

Zu Beginn muss das Regelungssystem im Betrieb genommen werden. Der einfachere Weg ist die Berechnung eines zeitinvarianten Reglers durch Anwendung von klassischen Verfahren. Als Beispiel dient ein PID-Regler. Wie schon Abbildung 24 zeigt, hat das System eine hohe Regelungsqualität, welche jedoch nur in dem Arbeitsbereich gewährleistet werden kann.

Ein anderer Weg ist es die Reglerstruktur auszuwählen und die Anfangsparameter des Reglers per Zufall zu bestimmen. Durch Anpassung der Reglerparameter gelangt der Regler zu einem Zustand, der ein Minimum des Gütekriteriums darstellt. Diese Methode benötigt jedoch eine gewisse Einstellzeit. Ein Magnetlager ist ein instabiles System und verlässt schnell den regelungsfähigen Bereich. Deswegen sind zufällige Anfangsparameter für einen NFN-Regler nicht zulässig und müssen vor dem Systemstart definiert werden, um einen stabilen Arbeitspunkt zu erreichen.

Aus diesem Grund wurde zuerst ein PID-Regler entworfen, der die Stabilität der Regelstrecke gewährleistet. Um das NFN zu verwenden, sind dessen Anfangsparameter zu bestimmen und der PID-Regler ist durch das NFN zu ersetzen. Dabei soll der Regelkreis keine Unterschiede zu einem Regelkreis mit klassischem PID-Regler ausweisen. Das erreicht man mittels Offline-Training des NFN. Das Trainingsschema ist in Abbildung 33 dargestellt.

Wie schon in Kapitel 1 beschrieben wurde, liegt der Unterschied zwischen den Online- und Offline-Trainingsverfahren in den Besonderheiten der Lernzyklusorganisation. Trotzdem sieht das Lernschema gleich aus. Für das Trainieren der NFN wurde ein stufenartiges Eingangssignal verwendet. Diese Signalform weist mehr Frequenzkomponenten nach der Fourier-Transformation als andere Signale auf.

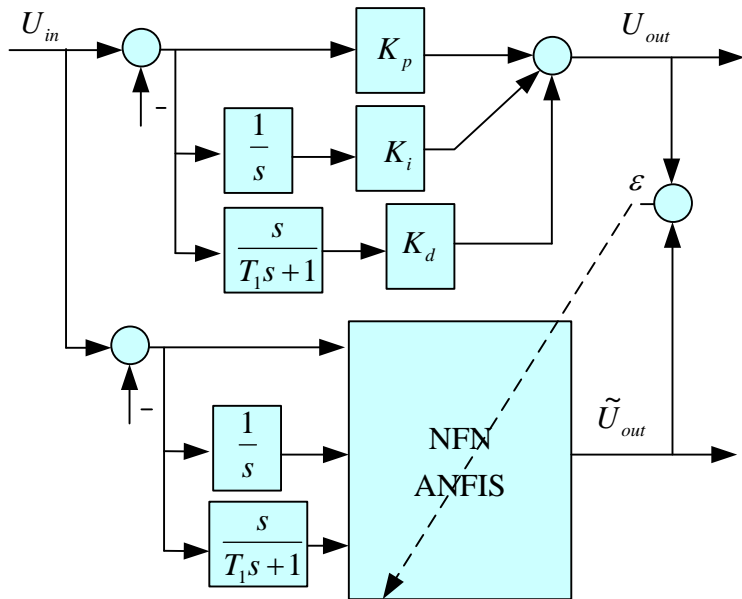


Abbildung 33 . Aufgetrennter PID-Regler für Lerndaten des Netzes

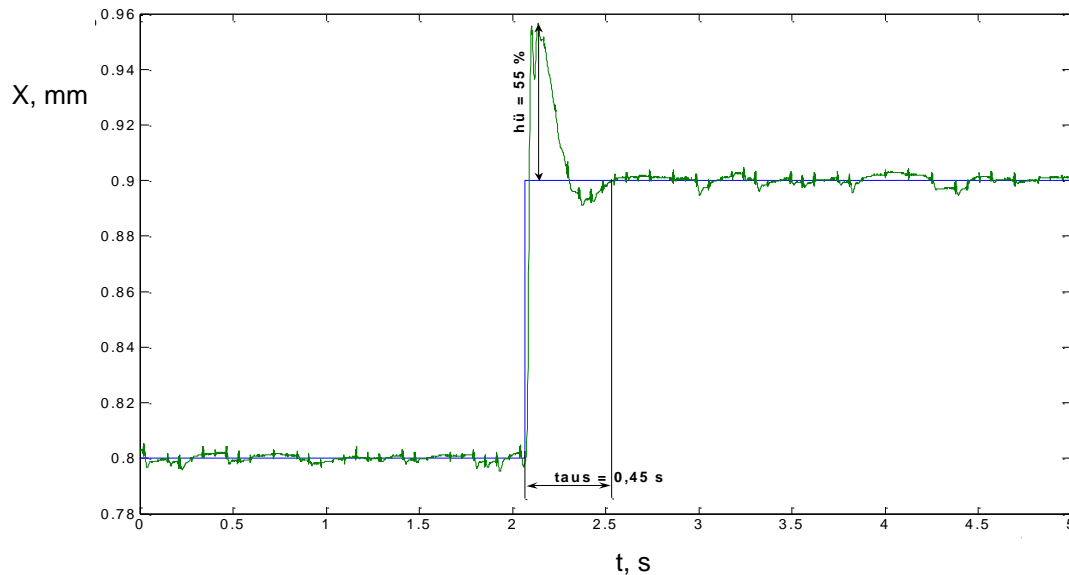


Abbildung 34. Messung der Sprungantwort mit statischem PID-ANFIS-Regler für Magnetlager mit einem Freiheitsgrad und dem ausgeprägten inhomogenen Magnetfeld

In Abbildung 34 sieht man die Regelungsqualität des PID-Reglers, der durch das NFN ersetzt wird. Dieser Versuch wurde ohne Anwendung des Anpassungsgesetzes durchgeführt. Zum Vergleich ist in Abbildung 20 der Verlauf desselben Regelkreises nach Einschalten des Anpassungsalgorithmus dargestellt. Im Kapitel 1 wurde eine Übersicht der Anpassungsgesetze erstellt. Hier kam eine Adaptation mittels „online back propagation“ mit PI-Modifikation zur Anwendung, siehe Gleichung (27).

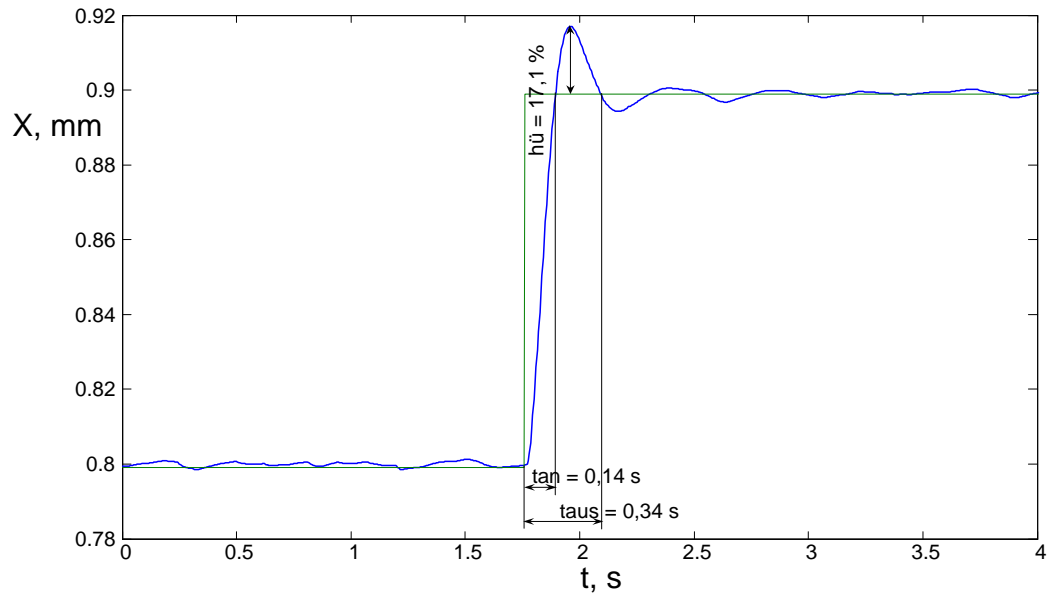


Abbildung 35. Messung der Sprungantwort des adaptiven ANFIS-Controllers

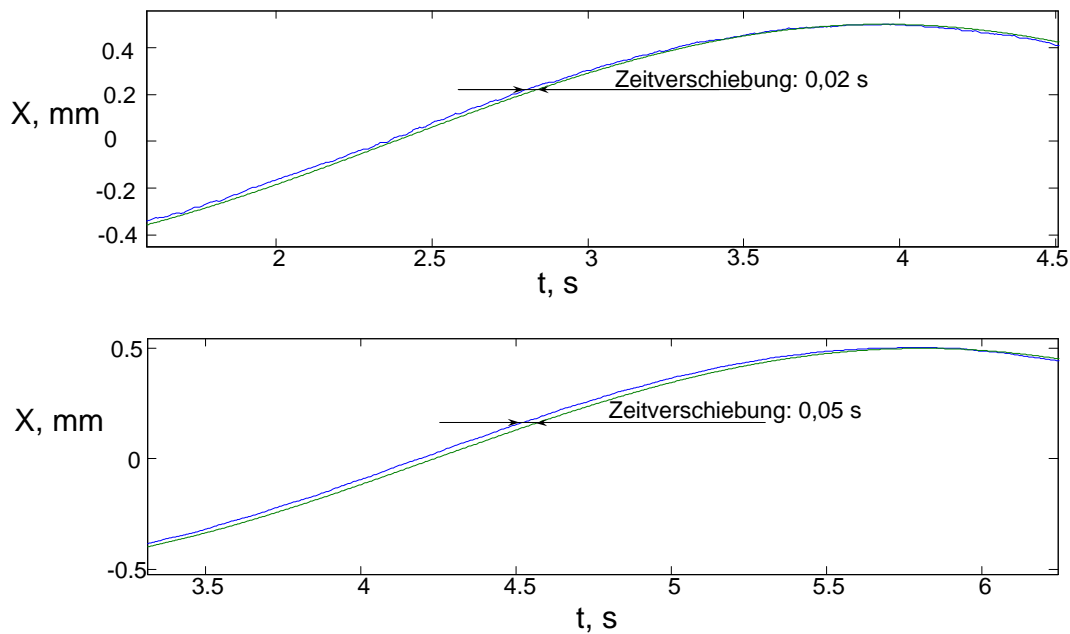


Abbildung 36. Messungen der Zeitverschiebung bei Sinusanregung; adaptiver Regler (oben), PIDT₁-Regler (unten)

Abbildung 36 untermauert nochmals die höhere Regelgüte des adaptiven Reglers. Mit Hilfe des NFN-Reglers folgt das Magnetlager einem Sinussollwertverlauf deutlich genauer. Diese Messungen entstanden ohne Nutzung eines Referenzmodells beim adaptiven Regler und ohne Verwendung eines Vorfilters beim klassischen Regler. In der Praxis sind die zu erwartenden Sollwertverläufe häufig durch vorherige Untersuchungen bekannt.

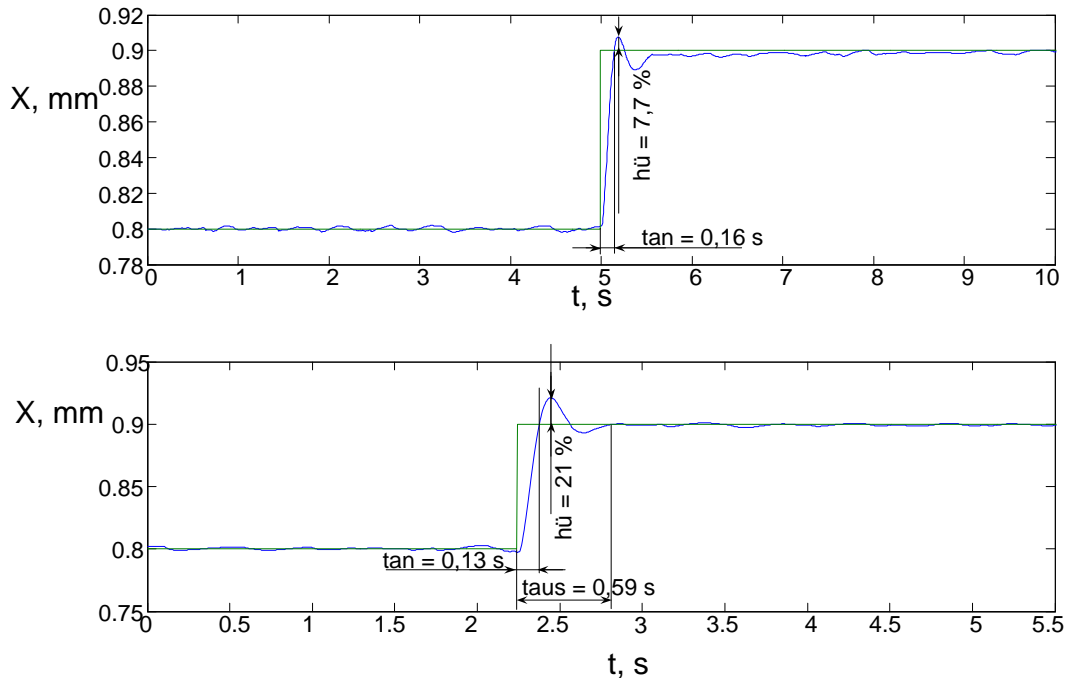


Abbildung 37. Messungen von Sprungantworten der adaptiven Regler;
Referenzmodell mit Zeitkonstante: $T_r = 0,05$ s (oben), und $T_r = 0,1$ s (unten)

Die abschließende Messung, in Abbildung 37 dargestellt, macht deutlich wie die Sprungantwort der adaptiven Regelungsstruktur vom Referenzmodell abhängt. Man erkennt die typischen Zusammenhänge zwischen der Überschwingweite, Anregelzeit und der Ausregelzeit. Bei der Messung mit dem Referenzmodell mit $T_r = 0,1$ s ist zwar die Ausregelzeit größer als beim klassischen Regler, jedoch lässt sich ein deutlich ruhigerer Stationärwert im Vergleich zur Regelung mit $T_r=0,05$ s verzeichnen. Dies liegt daran, dass die 18 dynamischen Gewichte bei einem steileren Fehlerverlauf bessere Optimalwerte finden. Die Überschwingweite beträgt hierbei 21 %.

Aus den letzten Abbildungen ist der Vorteil der Anpassungsregelungssysteme im Vergleich mit klassischen Verfahren ohne Anpassung deutlich erkennbar. Natürlich kann man in diesen Regelungssystemen nicht nur das NFN, sondern auch andere Approximationsmodelle verwenden.

2.4.2 Direkt adaptive Neuro-Fuzzy-Regelung des Magnetlagers mit dem homogenen Magnetfeld

Das im Folgenden verwendete und bewährte Referenzmodell der direkten adaptiven Regelungsstruktur lautet: $G_{ref}(s) = 1$.

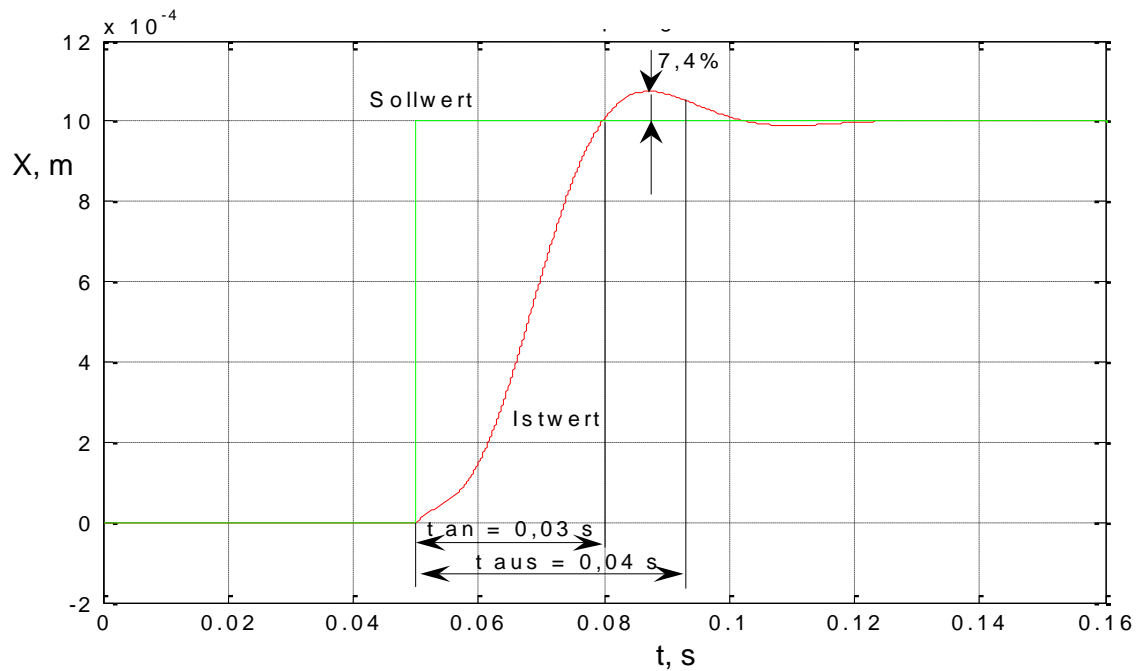


Abbildung 38. Messungssprungantwort des Systems mit PI-Lageregler bei der Einstellung nach der Butterworth-Verteilung

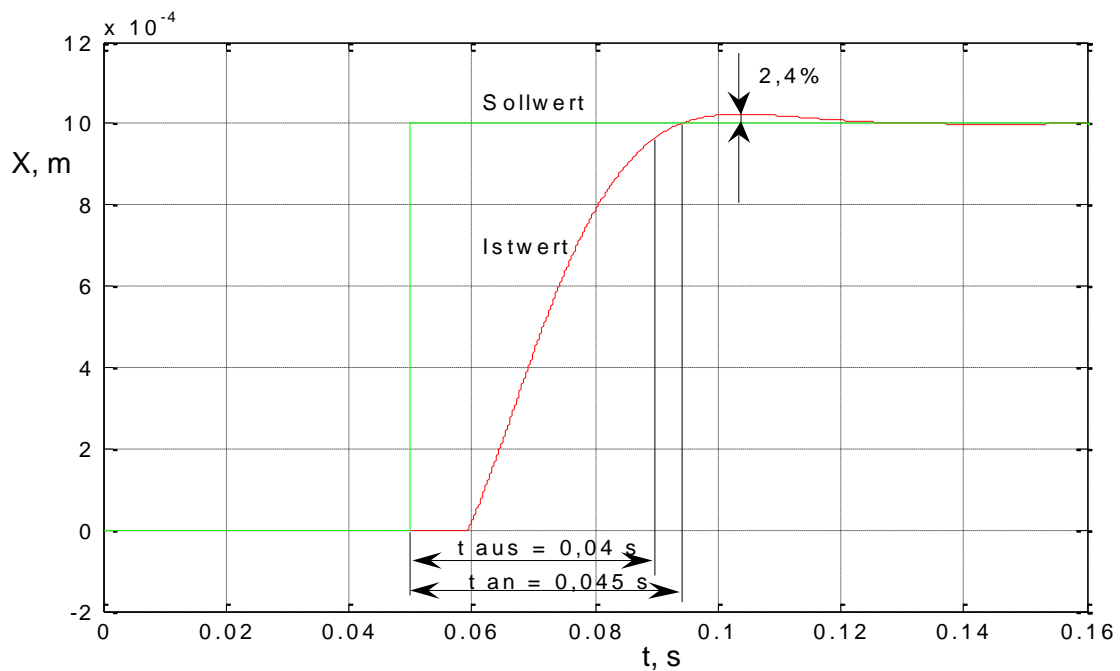


Abbildung 39. Messungssprungantwort des Systems mit adaptivem Lageregler (Lernrate $\eta = 1$)

Wie aus Abbildung 38 und Abbildung 39 ersichtlich ist, kann die Überschwingweite unter Verwendung des adaptiven NF-Reglers bei gleicher Abtastzeit von $7,4\%$ auf $2,4\%$ reduziert werden, die Ausregelzeit bleibt unverändert.

Abbildung 38 und Abbildung 39 bestätigen nochmals die höhere Regelgüte des adaptiven Reglers.

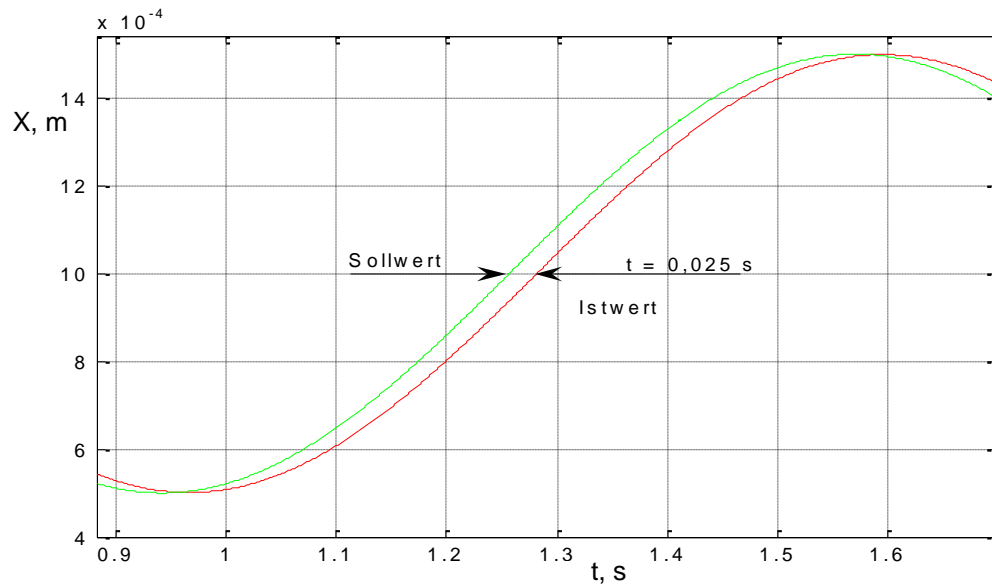


Abbildung 40. Messung der Zeitverschiebung bei Sinusanregung (PI-Regler bei der Einstellung nach der Newton-Verteilung)

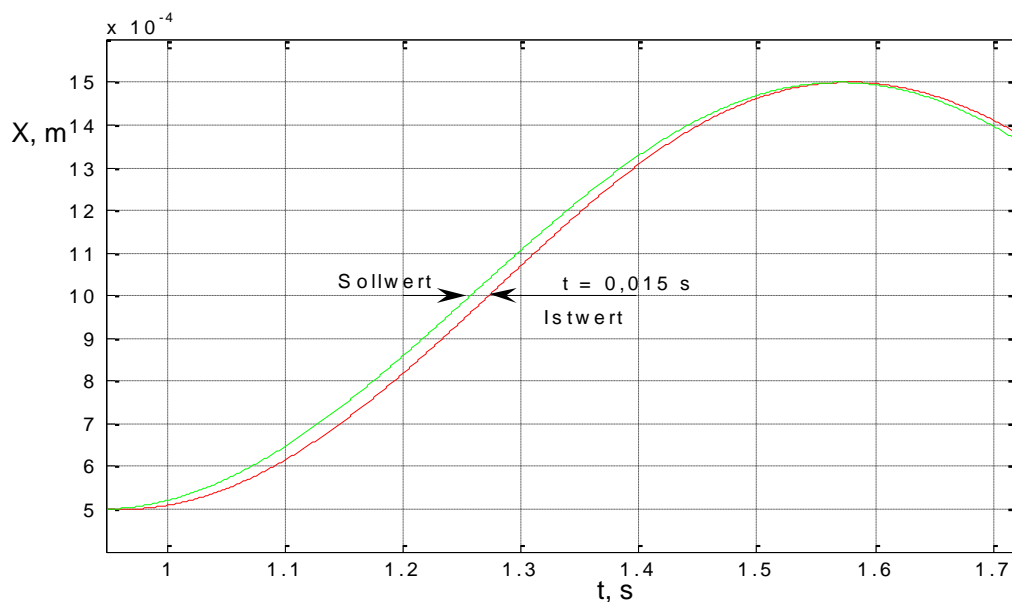


Abbildung 41. Messung der Zeitverschiebung bei Sinusanregung (Adaptiver Lagregler, $\eta = 2$)

Mit Hilfe des NFN-Reglers folgt das Magnetlager einem Sinussollwertverlauf deutlich genauer. Diese Ergebnisse entstanden unter Nutzung eines Referenzmodells beim adaptiven Regler und bei Verwendung eines Zustandsreglers mit PI-Anteil.

2.5 Adaptiver Neuro-Fuzzy-Sliding-Mode für Magnetlager mit einem Freiheitsgrad und dem homogenen Magnetfeld

Eine breite Art des Regelungssystems stellen die so genannten Systeme mit veränderlicher Struktur dar. Das bekannteste und weit verbreitetste davon ist der „Zweipunktregler“ (auf Englisch „sliding mode

controller“). Zum Vorteil dieser Reglerart gehört die hohe Robustheit und Unempfindlichkeit bezüglich parametrischen Störungen in der Regelstrecke. Das ist in der Besonderheiten der Systembewegung entlang der „Schaltlinie“ [14]-[16] begründet. Solche Systeme zeigen gute Leistung und Effizienz auch bei der Regelung des Magnetlagers. Eine Reduzierung des Hochfrequenzschaltverhaltens und Anpassung der Schaltlinie zur Erhöhung der Regelgüte wurden mittels Anwendung der NFN erreicht. Als Regelstrecke wurde das Magnetlager mit einem Freiheitsgrad ausgewählt. Es lässt sich folgendermaßen beschreiben:

$$\dot{x}^{(n)}(t) = f(x(t)) + u(t) + d(t), \quad (41)$$

woher $f(x(t))$ - das nichtlineare Teil des Systems, $u(t)$ - Eingangssignal, $d(t)$ - Störgröße.

Den Regelungsfehler kann man mit $e = x_d - x$ darstellen. Für den Gleitbetrieb ist folgende Bedingung zu erfüllen [1,10]:

$$\dot{q} \text{sign}(q) < -\eta, \quad (42)$$

wobei $\eta > 0$ sein muss und $q = f(e)$ eine Fehlerfunktion ist, die die Schaltlinie definiert.

Dies kann man mit der folgenden Bedingung für die Schaltlinie realisieren:

$$S = q(e) = (\dot{e} + \lambda e)^{n-1} \cong 0, \quad (43)$$

wobei λ der Neigungskoeffizient der Schaltlinie und n die Systemordnung ist.

Für diese Regelstrecke kann man $S = 0\ddot{e} + 2\lambda\dot{e} + \lambda^2 e$ verwenden. Das Regelungsgesetz eines klassischen Sliding-Mode-Control berechnet man wie folgt [17]:

$$U = Um \cdot \text{sign}(S) = \begin{cases} +Um, & \text{falls } S > \Delta S^+ \\ -Um, & \text{falls } S < \Delta S^- \end{cases}. \quad (44)$$

Die Variable Um erhält man mit $Um = K_1\ddot{e} + K_2\dot{e} + K_3$, wobei K_1, K_2, K_3 die Koeffizienten sind, welche dieselbe Bedeutung wie die entsprechenden Faktoren in Gleichung (43) haben. Die Faktoren K_1, K_2, K_3 kann man aus der mehrmaligen Modellerprobung finden. Aus dem Regelungsgesetz (44) ist das Schaltelement leicht zu erkennen, welches zwei Reglerstrukturen nacheinander schaltet. Dies führt zum Hochfrequenzschaltverhalten im Regelungssignal, was zu verhindern ist.

Die Anwendungen von NFN zur besseren Umsetzung von Regelungsgesetzen können in drei Gruppen gegliedert werden. Im ersten Fall funktioniert das NFN als reines Expertensystem und im Verlauf des Prozesses werden die Koeffizienten des Zweipunktreglers angepasst. Im zweiten Fall geht es um Approximation der Schaltlinie gemeinsam mit dem Schaltungselement. Im dritten Fall wird das Regelungsgesetz als eine Summe von einem NFN-Signal und einem anderen Signal aufgebaut.

In diesem Abschnitt wird die dritte Systemgruppe untersucht. Das Regelungsgesetz lässt sich beschreiben mit [17]:

$$U_R = u_{fuzzy} + u_s \quad (45)$$

Der Ausgang des NFN ist (vgl. Gleichung 6, Kapitel 1):

$$u_{fuzzy} = \theta^T \cdot \psi(x) \quad (46)$$

Hier ist θ^T der Ausgangvektor der Koeffizienten, $\psi(x)$ ein Regressionsvektor, $x = [2\lambda\dot{e}, \lambda^2 e]$ sind Eingangssignalen des NFN.

Der zweite Teil der Gleichung (45) wurde wie folgt berechnet:

$$u_s = -\frac{S}{\rho^2}, \quad (47)$$

wobei $\rho > 0$ ein Koeffizient ist, der von der Signalamplitude beeinflusst wird.

Das Anpassungsgesetz erfolgt nach dem „back propagation“ gradientenbasierten Algorithmus:

$$\dot{\theta}^T = -\eta \cdot S \cdot \psi(x) \quad (48)$$

Der Vergleich der Regelungsgesetze (44) und (45) zeigt, dass in der letzten Variante nahezu kein Hochfrequenzschaltverhalten generiert wurde.

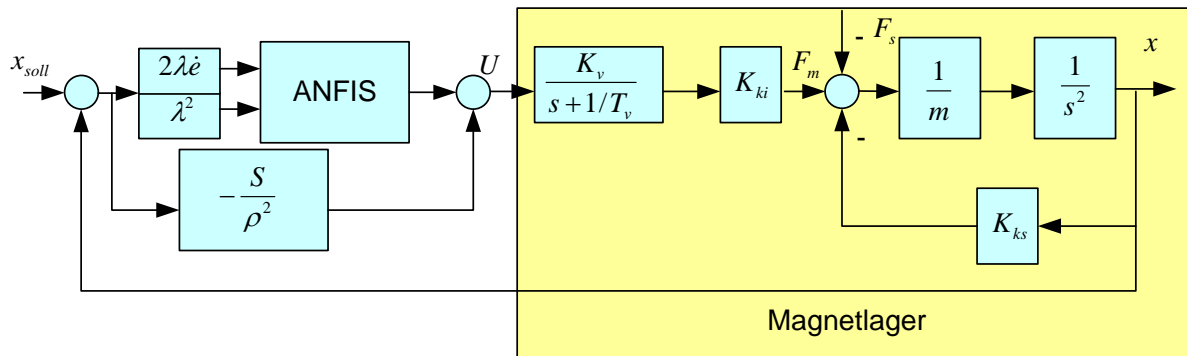


Abbildung 42. Das Signalflussbild des adaptiven NFN sliding mode Reglers für Magnetlager mit einem Freiheitsgrad

Auf Abbildung 43 wird der Übergangsvorgang an dem Reglerausgang dargestellt. Hier ist die Verringerung des Signaleinschwingvorganges im stationären Vorgang deutlich sichtbar.

Das Regelungssystem, das in Abbildung 42 gezeigt wird, hat einen sehr kleinen statischen Fehler. Das gewährleistet eine hohe Regelungsqualität bei beliebigen Änderungen des Stromfaktors $K_{ki} = (0.3..4)K_{ki}$, der Anker Masse $m = (0.3..2)m$ und des Positionsfaktors $K_{ks} = (0.3..2)K_{ks}$. Die Simulationsergebnisse zeigen ein höheres Robustheitsniveau als bei der Anwendung eines direkt adaptiven NFN-Reglers.

Die Bereiche, bei denen der Regler die genannten Eigenschaften aufweist, hängen von den Koeffizienten der Schaltlinie und dem Faktor ρ ab.

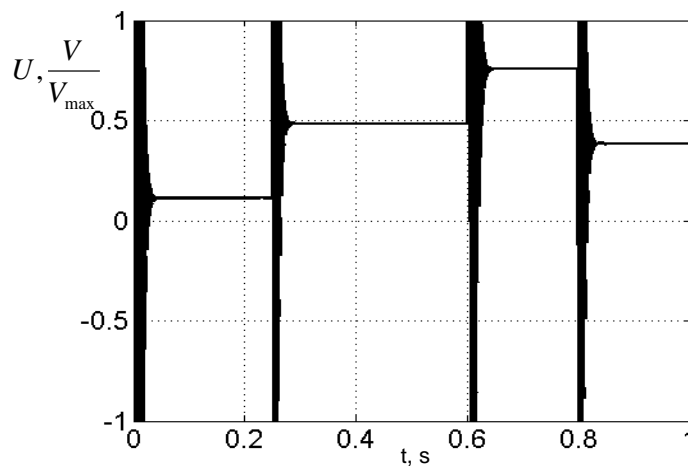


Abbildung 43. Die Spannung an dem Reglerausgang

Die gewünschte Regelgüte am Ausgang der Regelstrecke bei einem gegebenen Positionsverlaufes des Magnetlagers sieht man in Abbildung 44.

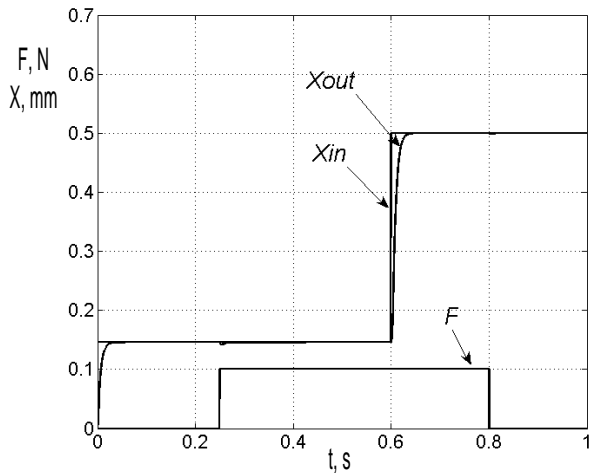


Abbildung 44. Die Sprungantwort des Magnetlagers

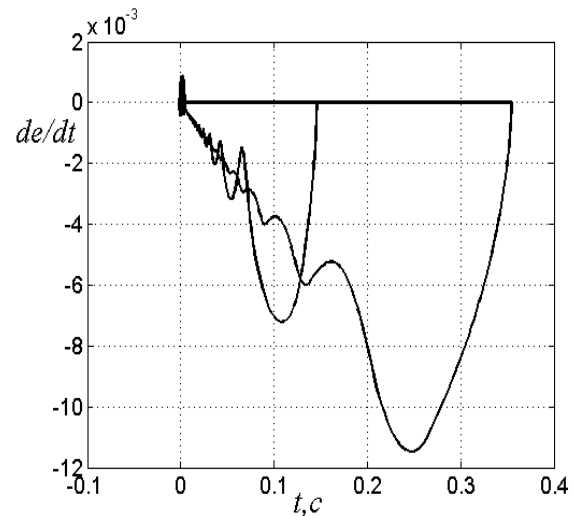


Abbildung 45. Das Phasenbild des Magnetlagers

Das vorgestellte Regelungssystem zeigt eine hohe Effizienz und Robustheit. Dafür ist vor allem der Sliding-Mode-Regler verantwortlich. Einen weiteren Vorteil des Systems stellt die Adaptivität des NFN dar. Das Schaltverhalten des Stellgliedes hängt vom Faktor ρ ab und kann nahezu unterdrückt werden.

2.6 Die Linearisierung durch Approximation der Nichtlinearitäten mittels Neuro-Fuzzy-Netz

Das schwierigste Problem des Magnetlagers bleibt die Nichtlinearität, die zur Instabilität der Regelstrecke führen kann. Der Gleitbetrieb lässt eine Handhabung des Systems in einem engen Bereich zu. Ohne Stabilitätsuntersuchung des Regelungsprozesses kann die Stabilität jedoch verloren gehen, sobald der Arbeitspunkt diese enge Region verlässt.

Die Linearisierung der Regelstrecke kann im gesamten Arbeitsbereich der Systemkoordinaten die Stabilität und eine hohe Regelgüte gewährleisten und ermöglicht eine Kombination mit anderen Regelungskonzepten. Um dieses Ziel zu erreichen, müssen zunächst die Nichtlinearitäten der Regelstrecke am Eingang und die Daten am Ausgang erfasst und gespeichert werden. Weiterhin ist die gemessene Nichtlinearität dank der Approximationsfähigkeit eines NFN oder, im allgemeinen Fall, durch einen beliebigen universalen Approximator, zu invertieren [18]. Danach ist es möglich die inverse Abhängigkeit zum Ausgleichen der Nichtlinearität zu verwenden.

Das gesamte Linearisierungsschema wird auf Abbildung 46 dargestellt.

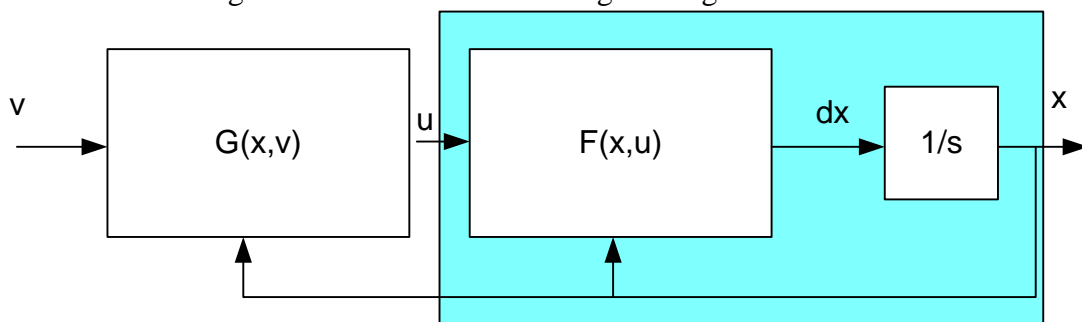


Abbildung 46. Das gesamte Linearisierungsschema

Die Nichtlinearität der Regelstrecke stellt die Funktion $F(x,u)$ dar. Die Linearisierungsfunktion $G(x,v)$ ist eine Funktion, welche die Bedingung $F(x,G(x,v))=1$ erfüllt.

Dazu werden die Daten $[u \in (U_{\min}..U_{\max}), x, dx]$ in einem Massiv gespeichert, wobei das Eingangssignal u alle erlaubten Werte annehmen kann. Die Spalten im Datenmassiv $[u \in (U_{\min}..U_{\max}), x, dx]$ werden bezeichnet mit U, X, DX .

Dann ergibt sich die Annäherungsfunktion $dx = F_{\text{exp}}(u, x) = \{U, X\} \rightarrow DX$. Daraus folgt die inverse Funktion mit $u = G(x, v) = \{X, DX\} \rightarrow U$.

Zum Schluss wird die inverse Funktion in die direkte Funktion eingefügt und man erhält:

$$dx = G(x, v)\{X, DX\} \rightarrow F(u, x)\{U, X\} \rightarrow v. \quad (49)$$

Diese Gleichung beweist, dass numerische Eingangs-/Ausgangs-Messdaten bei Invertierungen von Funktionen erfolgreich verwendet und für Linearisierungen benutzt werden können..

Für dieses Verfahren sprechen folgende Vorteile:

Genauere Invertierung von stetigen Funktionen.

Einfache Implementierung mit beliebigen Approximationsverfahren (NFN, NN, Spline, etc.).

Möglichkeit der Approximationsverbesserung im Betrieb mittels Anpassung.

Als Nachteile zählt man folgende:

Probleme mit der Datenaufnahme für den gesamten Arbeitsbereich der Signale.

Schnell ansteigende Komplexität der Approximatoren bei MIMO-Systemen.

Dieses Verfahren ist demnach für die Anwendung mit relativ einfachen Regelstrecken zu empfehlen, bei denen die benötigten Messdaten erfassbar sind.

2.7 Adaptive Linearisierung durch Ersatzfunktion

Wie im ersten Kapitel gezeigt wurde, existiert heutzutage eine Vielzahl von Regelungsverfahren, die man bei nichtlinearen Systemen sehr effizient einsetzen kann. Es gibt Methoden, die im Grundprinzip die Nichtlinearität der Regelstrecke vernachlässigen [11], [12], [19]. Einige Regelungsalgorithmen bemühen sich die Nichtlinearität und den linearen Integralanteil der Regelstrecke gleichzeitig zu kompensieren. Diese heißen „direct inverse control“ [20]. Eine solche Methode ist nicht genau auf Grund des Messrauschens, Ungenauigkeiten des Regelstreckenmodells und dem daraus resultierenden ungenauen inversen Modell der Regelstrecke. Alle Versuche diese Methode zu verbessern, waren bis jetzt erfolglos.

Andere Methoden, wie die „computed torque method“ oder die so genannte „kombinierte Regelung“ versuchen die Nichtlinearität aus der Regelstrecke abzuteilen und zu kompensieren mittels des Modellwissens [11], [19], [20]. Der verbleibende Integralanteil der Regelstrecke lässt sich gut und problemlos behandeln. Wenn das Modell der Realität entspricht, erreicht man die gewünschte Qualität der Regelungssysteme, andernfalls führt es zum Verlust der Stabilität oder zumindest zur Verschlechterung der Regelungsqualität.

Als Lösung wurden von vielen Autoren adaptive Verfahren vorgeschlagen. Meistens verwenden diese Autoren für Entwicklungen, Beweise und Begründungen des Anpassungsalgorithmus und des Gütekriteriums die Lyapunov-Funktion. Ein Nachteil dieses Verfahrens liegt in der Verwendung des Gradientenabstiegsverfahrens. Besonders wichtig sind die „Lokalen Minimum Probleme“, die dazugehörige Konvergenzgeschwindigkeit und die Stabilität des Anpassungsgesetzes. Die Ursache von solchen Schwierigkeiten sind oft nicht die Optimierungsalgorithmen selbst, sondern eine nicht korrekte Aufgabestellung.

Das Grundproblem solcher Systeme wird separat in einer Systembetrachtung behandelt.

Betrachtet man das System in einer bestimmten Form, muss man die Lösung für diese ausgewählte Systembetrachtung suchen. Es ist eine einfache Systembeschreibung zu finden, die sich mit dem gewünschten Reglungsalgorithmus beschreiben lässt.

In der Literatur [21] begegnet man oft Systemen in der Cauchy-Form:

$$\dot{X} = A(X)X + B(X)U. \quad (50)$$

Um die Lie-Ableitungen anwenden zu können, müssen $A(X), B(X)$ differenzierbar sein. Sollten diese Differenzierungen nur numerisch lösbar sein, kann die Lie-Ableitungen mit dem dazugehörigen Linearisierungsalgorithmus [21] nicht verwendet werden. Außerdem muss dieses Regelungsgesetz dividierbar sein (siehe Kapitel 1 Gleichung (52)) und es muss für MIMO-Systeme eine Invertierungsoperation existieren.

Wenn man „computed torque control“ verwendet, wird nicht unbedingt ein *analytisches* Modell der Regelstrecke benötigt, weil der Algorithmus in der Praxis nur numerische Werte verlangt und sich das Regelungsgesetz (bzw. Linearisierungsgesetz) in der Form $U = B(x)^{-1}(v - A(x)X)$ beschreiben lässt. Hier ist $v = \dot{x}$ ein äußeres Regelungssignal für den linearen Teil der Regelstrecke. Trotzdem muss auch hier die Matrix $B(X)$ invertiert werden. Die Matrizen $A(X), B(X)$ werden in Echtzeit berechnet.

Die Systemdarstellung in der Lagrange-Euler-Form $F = M(x)\ddot{x} + C(x)\dot{x} + G(x)$ liefert nach einem Umformungsschritt das bereits bekannte Linearisierungsgesetz (Computed Torque Control): $\ddot{x} = M(x)^{-1} \cdot (v - C(x)\dot{x} - G(x))$.

Wie man sieht, wurde das komplexe nichtlineare System auch bei diesem Linearisierungsverfahren konnte bis zu $\ddot{x} = v$ umgeformt. Die genannten Verfahren funktionieren ausgezeichnet unter der Bedingung eines sehr genauen Systemmodells, welches nahezu der Realität entspricht. Leider führen Reibungskräfte, Kontaktkräfte, nichtgenaue Messungen und Daten zu Abweichungen zwischen der Realität und dem mathematischen Modell. Weitere Schwierigkeiten resultieren aus der ausgewählten Systemdarstellung.

Die Gleichung (50) kann in folgende Form umgewandelt werden:

$$\dot{x} = K(x)U \quad (51)$$

Dazu muss $K(x) = A(x)XU^{-1} + B(x)$ gesetzt werden. Eine solche Systemdarstellung birgt wesentliche Einschränkung auf Grund der Invertierung U^{-1} in sich, was bedeutet, dass U invertierbar sein muss. Dennoch lässt sich das Linearisierungsgesetz mit:

$$U = v / K(x) \quad (52)$$

beschreiben.

Hier ist leicht zu erkennen, dass *das Regelungsgesetz von der Systemdarstellung abhängig ist*. Das bedeutet, dass eine solche Systemdarstellung ausgewählt werden sollte, die automatisch auftretende numerische Probleme löst.

Für diesen Zweck wird die folgende Systemdarstellung gewählt:

$$\dot{x} = K(x) + U, \quad (53)$$

wobei unter Berücksichtigung von Gleichung (50) sich $K(x) = U(B(x) - I) + A(x)X$ ergibt. Dann lautet das Linearisierungsgesetz:

$$U = v - K(x) \quad (54)$$

Vorteilhaft ist dabei die einfache Form der Gleichung (54).

Eine Systemdarstellung mit einem *invertierbaren Funktionspaar*, wie es die Gleichungen (51) sowie (52) oder die Gleichungen (53) sowie (54) wiedergeben, und entsprechender Linearisierung wird *Linearisierung durch Ersatzfunktion* genannt.

Vorteile des Gesetzes (54) bestehen in einer einfachen Matrixinvertierung und in der universellen Einsetzbarkeit, da dieses Prinzip auf unterschiedlichste Systeme erweiterbar ist. Dasselbe gilt

beispielsweise für Lagrange-Euler-Gleichungen der Mechanik, denn es gilt:
 $K(x) = F(M^{-1}(x) - I) - C(x)\dot{x} - G(x)$.

Eine weitere Verbesserung ist möglich, indem der Koeffizient K nicht berechnet, sondern durch Messungen am Realsystem bestimmt wird. Die diskrete Systembeschreibung der Gleichung (53) lautet (vergl. mit digitale Cauchy-Form):

$$x[n+1] = K(x)[n] + U[n], \tag{55}$$

woraus sich ergibt:

$$K(x)[n] = x[n+1] - U[n]. \tag{56}$$

Wenn man $K(x)[n] \approx K(x)[n+1]$ oder $K(x)[n-1] \approx K(x)[n]$ annimmt, folgt das Linearisierungsgesetz:

$$U[n+1] = v[n+1] - K(x)[n] = v[n+1] - (x[n-1] - U[n]) \tag{57}$$

Jetzt ist erkennbar, dass die Linearisierung mittels Gleichung(57) *kein* Systemwissen vorausgesetzt. Sie beinhaltet nur Zustands- und Eingangssignalmessungsergebnisse und die Systemordnung muss bekannt sein. Der lineare Teil des Systems ist nach dem nichtlinearen Teil des Systems angeordnet.

Das Regelungssystem muss noch mit einem Regler für den linearen Teil der Regelstrecke ergänzt werden. Für ein System in der Form $\dot{x} = F(x,u)$ sieht das Schema der Linearisierung wie in Abbildung 47 aus. Hier v ist das Signal des Positionsreglers.

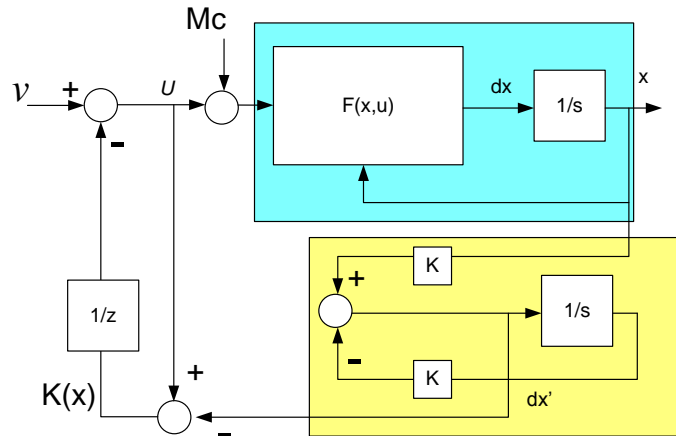


Abbildung 47. Das Schema der Linearisierung durch Ersatzfunktion (57)

In realen Systemen ist das Element $1/z$, entsprechend Abbildung 47, durch ein Tiefpassfilter beispielsweise der Form $W_{kf}(s) = \frac{1}{T_{kf}s + 1}$ auszutauschen, um eine sichere Konvergenz des

Linearisierungsverfahrens zu gewährleisten. Dies erlaubt große Unterschied zwischen dem Anfangszustand u und Beobachterzustand dx zu dämpfen, wobei T_{kf} die Konvergenzzeit des Anpassungsalgorithmus ist.

Eine solche Dämpfung funktioniert für relativ langsame Regelstrecken, die keine sprunghaften Änderungen aufweisen, wie zum Beispiel die Roboterhand (Kapitel 4) oder ein elektrischer Antrieb. Für Magnetlager funktioniert dieses Verfahren ebenfalls, solange es sich um Magnetlager mit nur einem Freiheitsgrad handelt.

Für Regelstrecken wie Magnetlager mit fünf Freiheitsgraden, die sehr instabil sind, kann ein solches Dämpfungsverfahren nicht angewendet werden. Deswegen wurden adaptive NFN-Filter für das Signal $K(x)$ verwendet. Das Schema des Regelungssystems wird in Abbildung 48 dargestellt.

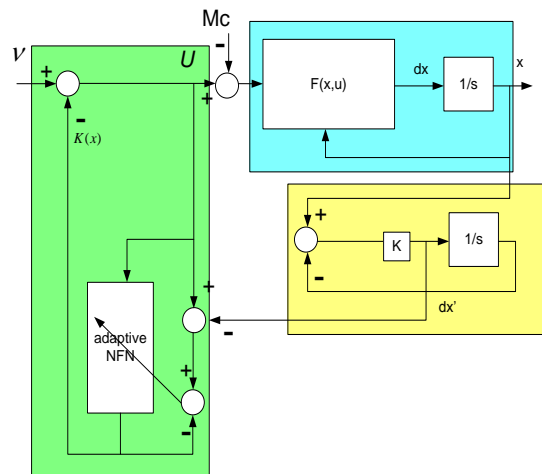


Abbildung 48. Regelungssystem mit Linearisierung und NFN als adaptiven Filter für $K(x)$

Hier sieht man, dass das Eingangssignal des NFN das Regulationssignal U ist, welches gleich dem Trägheitsmoment des Magnetlagers ist. Deswegen müssen die Zugehörigkeitsfunktionen des NFN alle möglichen Werte des Signals U einschließen. Die Aufgabe des Linearisierungsverfahrens ist es eine Robustheit und Unempfindlichkeit gegenüber der Nichtlinearität der Regelstrecke zu erreichen. Abbildung 49 zeigt das Fehlerverhältnis zwischen der gewünschten und realen Beschleunigung des Systems. Die Linearisierung selbst ist dabei nur ein Teil des Regelungssystems, weil die *gewünschte* Beschleunigung noch berechtigt werden muss. Dafür wird ein PD-Regler mit $K_p=1000$ und $K_d=50$ verwendet.

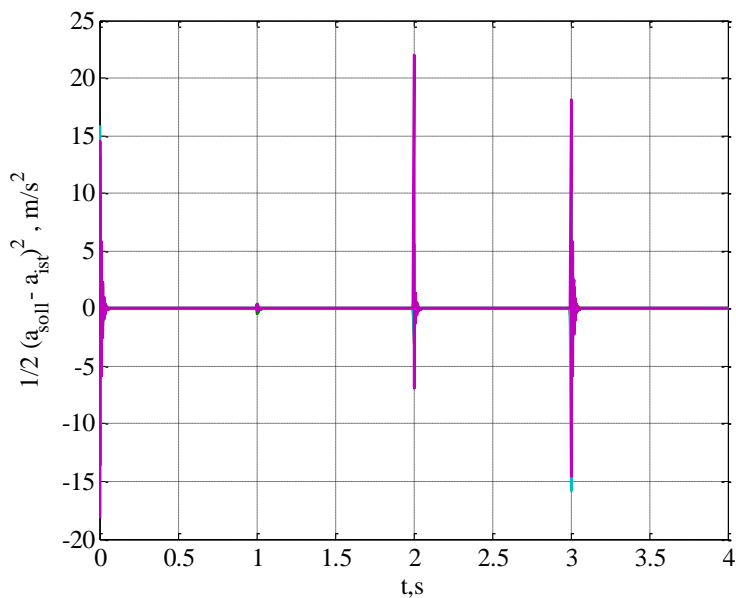


Abbildung 49. Fehler zwischen gewünschter und realer Beschleunigung des Magnetlagers mit fünf Freiheitsgraden in $[m/s^2]$

Für eine gute Anfangskonvergenz müssen die Startwerte für das Signal $K(x)$ entsprechend Abbildung 48 vor dem Systemstart definiert werden. Ansonsten muss der Regler beim Systemstart die Anfangsfehler ausgleichen, was zu hohen Überschwingungen im Stromregelkreis führt.

Auf Abbildung 50 ist der Übergangprozess des Magnetlagers mit fünf Freiheitsgraden dargestellt. Hier sieht man die Sprungantwort des Systems mit $X_{soll}=1\text{mm}$ und die Störungsreaktion des

Regelungssystem bei $t = 1$ s. Die Überschwingweite beträgt dabei 2,5%, die Anregelzeit ist gleich 0,12 s und Ausregelzeit beläuft sich auf 0,2 s. Ab dem Zeitpunkt $t = 2$ s wird das Sollsignal für die Winkel α und ϕ geliefert. Ab der Zeit $t = 3$ s ist der Lagesollwert $X_{soll} = 0$ mm.

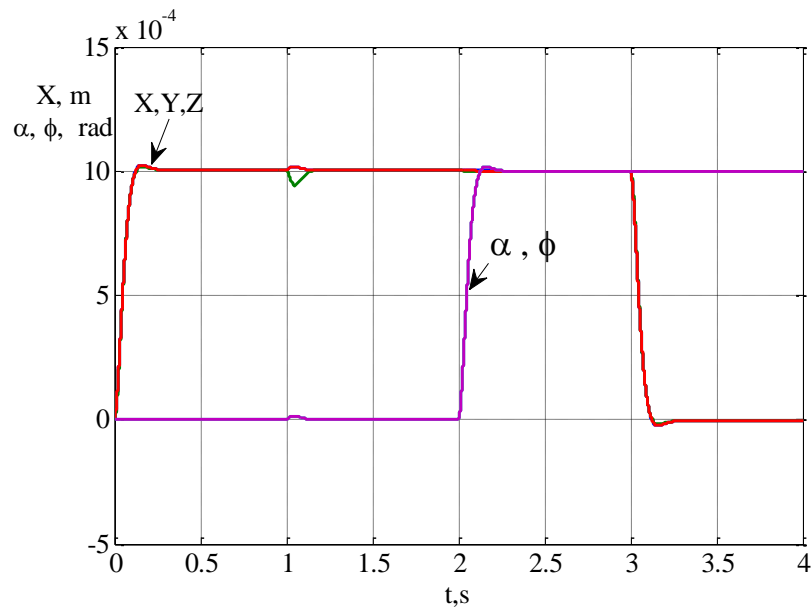


Abbildung 50. Sprungantwort des Magnetlagers mit fünf Freiheitsgraden mit NFN Linearisierung und PD Regler

Auf Abbildung 50 ist deutlich zu sehen, dass das Systemverhalten bei allen Freiheitsgraden gleich ist. Das wurde mit Hilfe des Linearisierungsalgorithmus erreicht. Außerdem bleibt das betrachtete System fast vollständig unempfindlich gegenüber Änderung der Systemparameter, wie Änderungen des Kraftstromkoeffizienten $k_i = (0.01..100)k_i$, der Ankermasse $m = (0.01..100)m$ und des Kraftwegkoeffizienten $k_s = (0.01..100)k_s$.

Es ist dafür jedoch auf eine ausreichende Leistung des Stellgliedes zu achten.

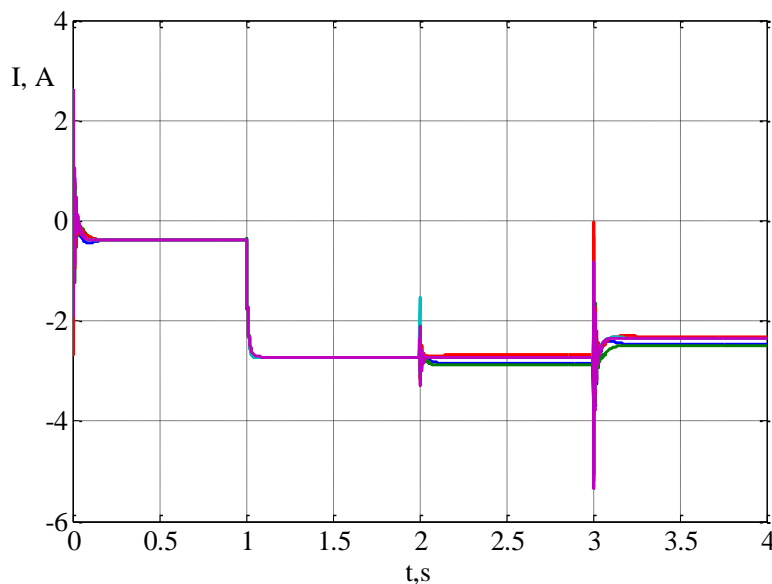


Abbildung 51. Der Stromübergangsprozess des Magnetlagers mit fünf Freiheitsgraden bei NFN-Linearisierung

Die Abbildung 51 zeigt den Stromverlauf für die nominalen Regelstreckeparameter. Besonderheit dieses Regelungssystems ist die gedämpfte Schwingung bei stufenartigen Änderungen an den Systemeingängen. Die maximalen Stromüberschwingungen können mit dem gedämpften Sollsignal gemindert werden.

Zusammenfassung

In diesem Kapitel wurden unterschiedliche Regelungssystemen für Magnetlager mit unterschiedlich ausgeprägter Nichtlinearität und Anzahl den Freiheitsgraden entworfen. Für Magnetlager mit einem Freiheitsgrad (Versuchstände 1 und 2) wurden die experimentalen Ergebnisse dargestellt. An diesen Regelstrecken wurden adaptive Regelungssysteme mit NFN verwendet und ihre Vorteile im Vergleich zu PI-, PID- und Zustandsreglern mit vielen Experimenten nachgewiesen. An der Simulation wurde ebenso der adaptive Neuro-Fuzzy-Sliding-Mode-Regler getestet und dessen Vor- und Nachteile analysiert.

Im zweiten Teil des Kapitels wurde eine Einführung in die nichtlineare Regelung am Beispiel der Approximationsverfahren gegeben. Für die Magnetlager mit fünf Freiheitsgraden wurden klassische Linearisierungsverfahren betrachtet. Dazu gehört die Linearisierung mittels Lie-Ableitungen, die unbedingt analytische Gleichungen des Systems voraussetzt. Als zweites sehr gängiges Konzept ist der so genannte „computed torque control“ vorgestellt worden, welcher zumindest ein numerisches Modell des Systems verlangt und bei ungenauer Modellbeschreibung die Regelungsqualität nicht gewährleistet.

Die Vor- und Nachteile von klassischen Linearisierungsverfahren wurden gründlich analysiert. Zum Schluss wurde auf ihrer Basis ein ganz neues Konzept der Linearisierung vorgeschlagen und mit Hilfe der Simulation getestet. Dieses Konzept wurde „Linearisierung durch Ersatzfunktion“ genannt. Im Gegensatz zu allen zuvor genannten Regelungssystemen ist hier kein genaues Wissen über die Systemparameter nötig und der Regler zeigt außerordentliche Stabilität bei Veränderungen der Systemparameter.

Die Weiterentwicklung dieses Verfahren wird in Kapitel 4 dargestellt.

Kapitel 3. Die Regelung des Vorschubantriebs der autonomen Unterwasserfahrzeug

3.1 Autonomes Unterwasserfahrzeug(UWF)

In Bezug auf die aktive Erschließung von Meer und besonders der Küstengebiete wurde der Entwicklung von Unterwassertechnik große Aufmerksamkeit gewidmet. Hier müssen unterschiedliche technische Operationen in der Wasserumgebung durchgeführt werden. Dies ist besonders in großen Tiefen wichtig, die auch für gut ausgerüstete Taucher schwer erreichbar sind.

Zu diesen technischen Mitteln gehören autonome Unterwasser-Fahrzeuge (UWF), die für Seenotrettungsarbeiten, Sucharbeiten eingesetzt werden. UWF benutzt man üblicherweise für die Untersuchung an versunkenen Schiffen oder anderen Objekten. Weiterhin helfen UWF bei der Inspektion von Unterwasserversorgungslinien und Unterwasseranlagen, bei Probeentnahme von Gestein, Wasser, Pflanzen, biologischen Organismen usw.

Probleme der Entwicklung und Perspektiven von UWF sind in vielen Literaturquellen [1]-[8] und mehrere anderen Quellen behandelt. Gegenwärtig sind in dem Unterwassertechnikbereich viele Probleme bezüglich des Aufbaus und der Produktion von UWF gelöst. Trotzdem gibt es viele Fragen, welche noch zu lösen sind.

Die Erweiterung des Anwendungsbereiches von UWF führt zur Notwendigkeit der Ausführung von komplexen und genauen Bewegungen in einem breiten Geschwindigkeitsbereich[1]. UWF müssen sicherstellen, dass sie sich schnell und genau an den Einsatzpunkt bewegen, ohne in Berührung mit Hindernissen oder anderen Objekten zu kommen. Bei Bewegungen können sich die Parameter der Umgebung und die Trägheit des UWFs schnell ändern. Dies führt zu einer unbestimmten hydrodynamischen Charakteristik des UWFs und zur starken gegenseitigen Beeinflussung der Freiheitsgrade des UWF.

Auf die Regelungsqualität wirken sich die Besonderheiten des Vorschubantriebes wesentlich aus, welche mittels nichtlinearer differentialer Gleichungen mit variablen Parametern beschrieben werden. Heutzutage existieren eine Vielzahl an Verfahren und Methoden für die Projektierung des UWF-Regelungssystems. Besonders attraktiv und erfolversprechend sind Methoden, welche eine Robustheit aufweisen. Solche Regelungen sind unempfindlich bezüglich Änderungen der Regelstreckenparameter oder Umgebung und können ebenso für Systeme verwendet werden, deren Parameter nicht genau bekannt sind.

Systeme, welchen robuste Regelungen zugrunde liegen, weisen aber auch erhebliche Nachteile auf. Sie ermöglichen nicht die benötigte Regelungsqualität in allen Arbeitssituationen des UWF oder werden sehr kompliziert und kostspielig. Das kann deren praktische Anwendung verhindern.

Bis heute wurden nur solche UWF angewendet, die für relativ einfache Aufgaben geeignet sind und fast keine intelligenten Regelungssysteme besitzen. Die Erschließung neuer Anwendungsgebiete und die Erweiterung der Funktionalität von UWF werden in naher Zukunft neue hochgenaue und robuste Regelungssysteme fordern, die ein neues Qualitätsniveau ermöglichen. Deswegen sind die UWF ein gutes Testfeld für neue hochwertige Regelungsalgorithmen, die später erläutert werden. Das Lösung der o.g. Aufgaben bildet die Basis dieses Kapitels.

Unbemannte UWF erlauben die Ausführung von unterschiedlichsten Unterwasseroperationen und Untersuchungen, ohne unmittelbare Gegenwart eines Menschen unter Wasser. Dies führt zur Minderung des Risikos für das Personal. UWF sind wegen ihrer relativ kleinen Größe günstig in der Anschaffung und im Unterhalt und brauchen keine speziellen Versorgungsschiffe. Weiterhin hängt ihr Einsatz nur wenig von Wetterbedingungen ab.

Infolge der rasanten Entwicklung der Informationssystemen, der Geräte- und Rechnerausrüstung, Verbindungsmitteln und Informationsverarbeitungsmethoden vollzieht sich ein Wandel der unbemannten UWF zu effektiven ferngesteuerten, autonomen Unterwasserrobotersysteme. Mit dieser Entwicklung erreichen Untersuchungen und unterschiedliche Unterwasserarbeiten zur See ein neues Qualitätsniveau, wobei die Realisierung der Idee von „Global Monitoring“ und realer Überwachung des Meeres begünstigt wird.

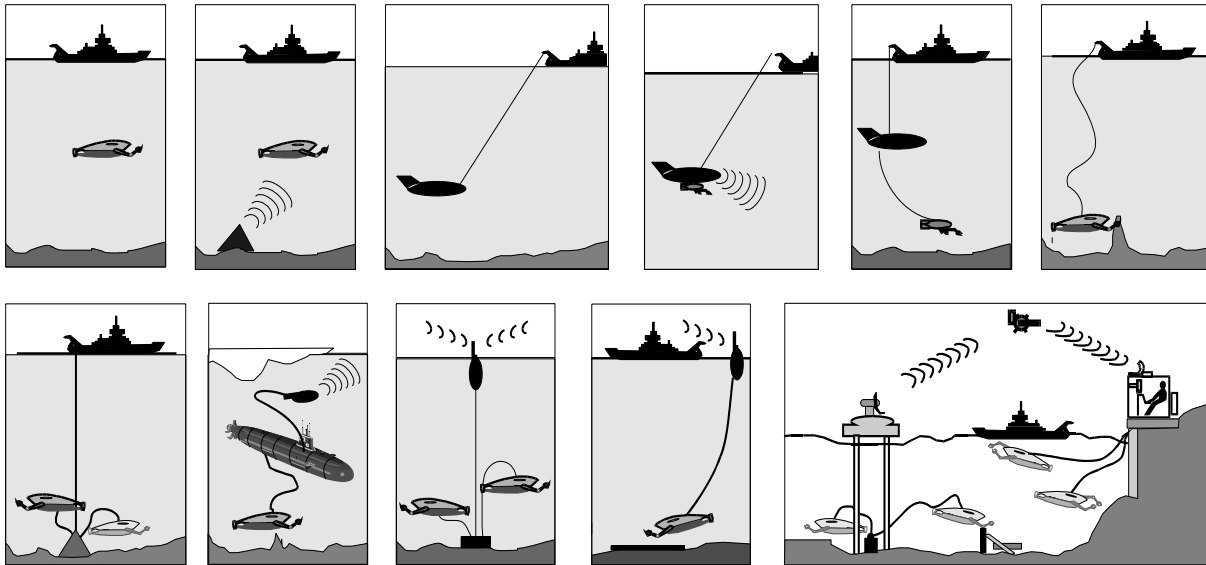


Abbildung 52. Varianten des Aufbaus von unbemannten Unterwassergeräten und Anlagen [1]

Da die UWF eine Schwebefähigkeit haben, steht ihr Gewicht im direkten Zusammenhang mit dem Außenmaß. Daraus folgt, dass die technischen Möglichkeiten des Apparates mit dem Gewicht des Apparates verbunden sind.

Es gibt auch eine Abhängigkeit zwischen dem Gewicht des Apparats und seiner Tauchtiefe.

Nach dem Gewicht werden UWF in folgende Gruppen eingeteilt [1]:

- Mikro	Weniger als	1kg,
- Mini	1 kg -	30 kg,
- Kleine	30 kg -	100 kg,
- Mittlere	100 kg -	1000 kg,
- Große	Mehr als -	1000 kg.

Die Gruppe „Mikro“ besteht aus einzelnen exotischen Exemplaren, wie z.B. aus Apparaten für Bohrlöcher. Die Gruppe „Mini“ ist die am häufigsten vorkommende Gruppe. Sie vereint sehr leichte Apparate, die für eine Versorgungsperson geeignet sind. Solche UWF benutzt man meistens für Fernsehinspektion (eye-ball) von Hafenanlagen, Unterwasserteilen von Schiffen, für die Überwachung der Unterwasserwelt und dergleichen. Üblich sind hierbei Apparate für kleine Tiefen (bis 100 m) und geringes Gewicht (bis 30 kg). Sie haben einen relativ kleinen Kaufpreis (bis \$ 10.000) und werden in der Literatur als „Low-Cost ROV“ (remotely operated vehicle) bezeichnet. Niedrige Preise ermöglichen es auch Privatpersonen UWF zu kaufen. Solche Apparate heißen Personal- UWF oder UWF für Freizeit und Unterhaltung. UWF können heutzutage einen guten Geschäftserfolg aufweisen.

3.1.2 Bereiche erfolgreicher Anwendung von UWF

Moderne UWF finden mehr und mehr Anwendung bei allen technischen Unterwasserarbeiten und Untersuchungen. Nach Meinung von USA-Spezialisten kann man ab dem Jahr 2010 die Aufnahme von UWF als Teil der Kriegsmarine erwarten, wo sie sich als sehr effektive und lohnenswerte Komponente von Kriegssystemen zeigen können. Insofern verursachen sie im Vergleich mit anderen modernen Waffen relativ geringe Kosten und können als Erzeugnis einer Serienproduktion überall Anwendung finden.

Es ist leicht ausgesprochene Anwendungsbereiche der UWF [2] zu nennen:

- Aufspüren und Rettungsarbeiten;
- Erkundung und Beobachtung auf offener See und ozeanischer Bereiche;
- Streifendienst;
- Suche, Entdeckung und Überwachung von U-Booten;
- Verwendung als Teil der Waffensysteme eines U-Boots;
- Als U-Boot-Abwehrrtrieb;
- Vernichtung unterschiedlicher Objekte und Ziele;
- Minenaufstellung;
- Minenentschärfung;
- Elektronische und hydroakustische Kampfführung.

Fernsehgesteuerte UWF haben den Vorteil, dass ein Spezialist aus einem speziellen Wissenschaftsbereich die Arbeit vornehmen kann, da ein Taucher üblicherweise kein Spezialist für eine komplizierte Arbeit und die inspierte Technik ist. Außerdem schließt die Anwendung von UWF ein Risiko für die Taucharbeit aus, besonders bei komplizierten und schweren Tauchbedingungen. Der Erfolg des Tauchvorgangs wird jedoch durch Unwetter, unklares Wasser, chemische und andere Verschmutzungsarten und durch unzugängliche Gebiete erschwert.

Sehr viel versprechend ist die Anwendung von UWF auch für historische und archäologische Arbeiten.

3.1.3 Dynamik des Vorschubantriebs der Unterwasserfahrzeuge

Die Regelstrecke in Abbildung 52 stellt ein Objekt dar, das sich mit einer hohen Geschwindigkeit auf einer komplizierten Bahn bewegen soll. Die Genauigkeit der UWF-Positionierung hängt von der Regelungsqualität des Vorschubantriebes ab.

Zahlreiche Untersuchungen von dynamischen Eigenschaften eines UWF-Vorschubantriebes [1], [6], [8] zeigen, dass es sich hierbei um ein kompliziertes nichtlineares Differenzialgleichungssystem handelt. Leider haben diese Gleichungen auch unterschiedliche, schwer identifizierbare Parameter. Außerdem ändern sich diese Parameter ständig und hängen von den Arbeitsbedingungen des UWF ab. Deswegen erreicht man bei der Anwendung von unterschiedlichen traditionellen Regelungssystemen nicht die benötigte Genauigkeit der Regelung für UWF.

Es gibt sehr viele Untersuchungen, die sich der mathematischen Beschreibung des Vorschubantriebs des UWF [1]-[8] widmen. In dieser Arbeit wurde ein Modell des Vorschubantriebs benutzt, welches die wesentlichen Effekte des Zusammenwirkens zwischen der Schraube und dem Wasser erfasst. Deswegen ist dieses Modell sehr genau und erfasst alle Arbeitssituationen vollständig [1].

Das Modell beschreibt man [1], [8] mit dem folgenden Gleichungssystem:

$$\begin{aligned}
J_d \dot{\omega}_d + \frac{K_m K_w}{R_d} \omega_d + M_B &= \frac{K_m K_u}{R_d} u; \\
\tau_d &= F_\tau s_\tau |\omega_d| - C_f \frac{S_d \rho_g |\nu_g| \nu_g}{2(1 + C_\omega \omega_d^2)}; \\
M_B &= F_m (s_\tau + H_{gd} C_r \omega_d) |\omega_d| - C_m \frac{S_d \rho_g |\nu_g| \nu_g}{2(1 + C_\omega \omega_d^2)}; \\
p_\tau &= H_{gd} \omega_d - \frac{\nu_g}{2} + \frac{F_\tau \omega_d}{4 \rho_g S_d}; \\
s_\tau &= p - \text{sign}(\omega_d) \sqrt{p_\tau - q_\tau}; \\
q_\tau &= H_{gd} \omega_d (H_{gd} \omega_d - \nu_p); \\
H_{gd} &= H + \delta_H \text{sign}(\omega_d); \\
\nu_p &= \begin{cases} \nu_g, & \text{wenn } \text{sign}(\omega_d) = \text{sign}(\nu_g) \\ 0, & \text{wenn } \text{sign}(\omega_d) \neq \text{sign}(\nu_g). \end{cases}
\end{aligned} \tag{1}$$

Wobei τ_d die Schubleistung des Vorschubantrieb; ω_d die Geschwindigkeit der Schraube von Vorschubantriebs; H_{gd} der hydrodynamische Schritt der Schraube; H der geometrische Schritt des Vorschubantrieb; F_τ, F_m die homogenen Koeffizienten von der Stütze und dem Moment; C_m, C_f die Koeffizienten der Auftriebskraft und des Moments der Schraube bei $\omega_d = 0$; C_ω der Koeffizient der Minderung der Auftriebskraft und des Moments bei $\omega_d \neq 0$; C_r der Koeffizient des Kraftverlustes infolge des Schraubenprofiles; ρ_g die Wasserdichte; S_d die projizierte Fläche (senkrecht zu Schraubenachse) der Schraubenflügel; J_d das Trägheitsmoment der rotierten Teile des Vorschubantriebs, unter Vorbehalt des hinzugefügten Trägheitsmoments des Wassers; R_d der aktive Widerstand des Gleichstrommotors; K_m, K_ω die Koeffizienten des Trägheitsmoments des Gleichstrommotors und sein Geschwindigkeitskoeffizient; K_u der Verstärkungskoeffizient des Stromrichters; M_B das Moment des Motors; ν_g die Geschwindigkeit des Wassers in Verhältnis zu UWF entlang der Vorschubantriebachse; p_τ, q_τ, ν_p Hilfsvariablen.

Bekanntlich können sich die Parameter J_d und F_m des Vorschubantriebs, wegen des Einflusses vom hinzugefügten Trägheitsmoment des Wassers und der viskosen Reibung bei der Regelung der UWF wesentlich ändern.

Im Weiteren wird bei der Anwendung der vorgeschlagenen Algorithmen eine Analyse der Regelungsgenauigkeit durchgeführt. Das Regelungssystem ist in Abbildung 52 dargestellt. Hierfür wurde das mathematische Modell des Vorschubantriebs der UWF mit Gleichung (1) beschrieben.

3.2 Direkte inverse Steuerung mit Prognose für den Vorschubantrieb des Unterwasserfahrzeuges

Heutzutage gibt es eine hinreichende Anzahl von Methoden, die einen Entwurf hochwertiger Regelungssysteme zulassen. Leider sind solche Systeme üblicherweise sehr kompliziert und benötigen Messungen oder Identifikationen von unbekanntem Werten, Parametern oder Signalen.

Deswegen werden neue Methode und Verfahren auf der Basis von neuen theoretischen Ideen wie z.B. neuronalen Netzen (NN) und Fuzzy-Logik (FL) [9] weiterentwickelt. Regelungssysteme, die mit NN

und FL aufgebaut sind, haben wesentliche Vorteile im Vergleich zu traditionellen Regelungssystemen. Zum Beispiel setzen diese keine Kenntnis der mathematischen Beschreibungen der Regelstrecke voraus und können auch bei ungenau bekannten Regelstreckenparametern arbeiten. Solche Systeme weisen auch eine hohe Robustheit und Stabilität bei unterschiedliche Störungen auf. Die vorgeschlagenen Algorithmen lassen sich mit Standard-Mikrorechnertechnik, auch unter Verwendung von parallelen Rechnungssystemen wie FPGA, realisieren.

Es gibt einerseits System mit festen Parametern, die vor dem Regelprozess bestimmt werden müssen und andererseits gibt es Systeme, wo alle oder ein Teil der Parameter in Echtzeit bestimmt werden [10]. Die erste Variante ist für Systeme geeignet, in denen sich die inneren Parameter im Betriebszustand nicht stark ändern. Die zweite Variante verwendet man für System mit veränderlichen Parametern und stark nichtlinearen Abhängigkeiten. Dabei ist normalerweise die Anwendung von sehr leistungsfähigen Mikrorechnern nötig, weil Änderungen der Controllerparameter mit unterschiedlichen Optimierungsverfahren einen relativ hohen Berechnungsaufwand erfordern.

In den letzten Jahren sind Publikationen erschienen, in denen man die Besonderheiten der Anwendung von neuronalen Netzen und Fuzzy-Logik in Regelungssystemen beschreibt [4]-[8]. Dabei wurden die Objektparameter im Betriebszustand als konstant angenommen und ein Optimierungsprozess für ein NN ist nur in einigen Fällen erfolgreich realisiert worden.

Zwischen den unterschiedlichen NN lösen die so genannten Neuro-Fuzzy-Netze (NFN) sehr hohes Interesse aus. Ihr Algorithmus basiert auf der Fuzzy-Logik, die Struktur wird aber in Netzbegriffen, wie Knoten und Verbindungen, ausgedrückt. Im Unterschied zur klassischen Fuzzy-Logik bedarf es hierbei keiner Definition der Systemparameter, weil dafür automatische Optimierungsverfahren wie beispielsweise das Gradientenabstiegsverfahren existieren. Dieses Verfahren hilft die Probleme bei der Verwendung von Fuzzy-Logik-Algorithmen in adaptiven Regelungssystemen zu lösen.

Bei der Parameteranpassung von NFN ist trotzdem eine hohe Rechnungsleistungen nötig, weil der Berechnungsaufwand mit der Parameteranzahl und Trainingsdatenmenge proportional ansteigt.

Echtzeitsysteme mit sich anpassenden Parametern erhalten stets neue Trainingsdaten, die sich ständig verändern, obwohl die Kapazität im Vergleich mit Offline-Trainingsalgorithmen vielfach kleiner ist.

Es sind sehr viele NFN bekannt, welche sich durch ihre Aufbauarchitektur unterscheiden. Aus dieser Menge wurde das ANFIS (adaptive neuro fuzzy inference system) [9] ausgewählt. Dieses System ist auch in MATLAB als „fuzzy logic toolbox“ verfügbar. Eine erweiterte Beschreibung des ANFIS befindet sich im Kapitel 1.

Weil NFN relativ leicht einstellbar und robust sind - dies ist besonders wichtig für Systeme mit Störungen - sollten deren Eigenschaften eine effektive Regelung der nichtlinearen Systeme gewährleisten. Hier wurde ein NFN-Regler für das Vorschubantrieb des UWF verwendet.

3.2.1 Die Auswahl des effektiven Algorithmus für die Parameteranpassung in ANFIS NFN

Bei der Verwendung von NFN in adaptiven Regelkreisen ist, für die Regelung des nichtlinearen Systems mit veränderlichen Parametern, das schnellste und berechnungsgünstigste Optimierungsverfahren auszuwählen.

Hierfür eignen sich besonders gut gradientenbasierte Algorithmen, die traditionell in der Literatur als „Fehlerrückführung“ oder „backpropagation“ bezeichnet wird [9]-[15]. Der Anpassungsprozess des Verfahrens basiert auf einer iterativen Konvergenz der Fehlergradienten, welche minimiert wird.

Um die Effektivität der unterschiedlichen Trainingsmethoden vergleichen zu können, wurde als Test und Simulation eine ANFIS-Anpassung durchgeführt. Bei dem Training wurde das gleiche mittelquadratische Fehlerniveau erreicht, welches gleich 0.0001 ist. Alle Berechnungen wurden in MATLAB durchgeführt. Die Ergebnisse sind in Abbildung 53 dargestellt.

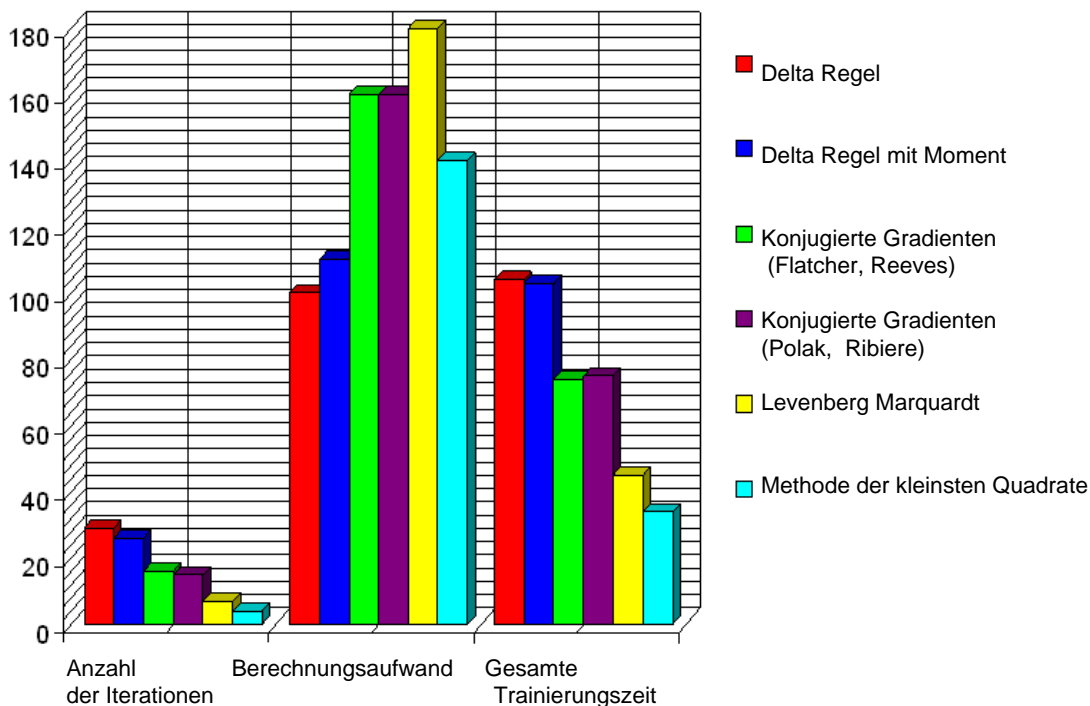


Abbildung 53. Vergleich der unterschiedlichen Trainingsverfahren der ANFIS

Für die Parameterberechnung von NFN benutzt man meistens die erste Ableitung der Fehlerfunktion nach dem Netzparameter. In diesem Fall kommt es zu einer langsamen Konvergenz, was mehr Iterationsschritte nötig macht, um das gewünschte Fehlerniveau zu erreichen. Eine Reduzierung der benötigten Iterationsschritte erreicht man durch Verwendung z.B. des Newton-Verfahrens. Dafür muss man die zweite Ableitung des Gütekriteriums nach dem Netzparameter berechnen. Um den Berechnungsaufwand zu minimieren, wurden Quasi – Newton Verfahren verwendet.

Bei der Anpassung des NFN im Offline-Betrieb benutzt man traditionell die Delta-Regel zusammen mit der Methode der kleinsten Quadrate. Weil Angaben über andere Optimierungsverfahren für NFN in den Publikationen fehlen, wurden hier Analysen durchgeführt, welche die Perspektiven und Möglichkeiten von unterschiedlichen Optimierungsverfahren für die Parameteranpassung in NFN aufzeigen.

Bei diesen Untersuchungen wurden einige Verfahren, welche wenige Iterationen brauchen, gefunden. Aus den Ergebnissen, die auf Abbildung 53 dargestellt sind, kann man schlussfolgern, dass die Kombination des Levenberg Marquardt Verfahren (für die Schicht der Zugehörigkeitsfunktionen) mit der Methode der kleinsten Quadrate (für die Ausgangsschicht) die günstigste Variante ist.

Im Weiteren wird das Training des ANFIS-NFN mit dem o.g. Verfahrenkombination durchgeführt. Jetzt ist ein Regelungsalgorithmus zu entwickeln, der komplizierte nichtlineare Regelstrecken in Echtzeit regeln kann. Dieser Algorithmus wird im nächsten Kapitel beschrieben.

3.2.2 Arbeit von NFN in Echtzeit

In traditionellen Systemen, in denen NN oder NFN für die Regelung der Regelstrecke mit konstanten Parametern verwendet werden, werden das Training und die Parameterabstimmung des NN nur einmal vor Beginn des Arbeitsbetriebs durchgeführt. Damit soll eine Trainingsdatei geschaffen werden, die aus einem Datenmassiv besteht, welches eine Signalreihenfolge darstellt. Diese Signale werden auf den Regelstreckeeingang gegeben und die Ausgangssignale (Reaktion der Regelstrecke) gespeichert. Dabei besteht die Eingangssignalreihenfolge aus stochastischen Werten vom gesamten Frequenz- und Amplitudenspektrum der Regelstreckeeingänge. Die Anzahl der Trainingsdaten und die Abtastzeit hängen von den Regelstreckeigenschaften ab. Auf der Basis der daraus resultierenden Datei wurde das inverse NFN-Modell der Regelstrecke bestimmt. Nach der Einfügung dieses inversen NFN-Modells in die Regelkreis wurde die bekannte „direkt inverse control“ Architektur realisiert[4], [11], [15]. Diese vermag dem geschlossenen Regelkreis die gewünschten Eigenschaften zu geben. Das o.g. Verfahren funktionierte nur dann gut, wenn die Regelstreckeparameter im Arbeitsbereich auch dieselben wie bei den vorhergehenden Tests sind. Andernfalls kann das trainierte NFN auf Grund einer bestimmten dynamischen Regelstreckeneigenschaft die gewünschte Regelungsqualität nicht gewährleisten, obwohl das NFN ein robustes Verhalten im Regelkreis aufweist.

Ist der Signalumfang der Regelstrecke relativ groß, ist, um ein gutes Trainingsergebnis zu erhalten, eine große Trainingsmenge für das NFN zu verwenden. Deswegen wächst die Trainingszeit des NFN schnell an. Wenn man weniger Trainingsdaten benutzt als nötig, führt dies ebenfalls zur Verschlechterung der Regelungsqualität.

Darum sollte man für NFN relativ kleine Datenmengen benutzen, welche sich im Arbeitsbereich verändern und in Echtzeit angepasst werden.

In Abbildung 54 ist das verwendete Strukturschema dargestellt. Hier wird für das Regelungssystem ein NFN verwendet, welches in Echtzeit angepasst wird. Das NFN funktioniert als inverses Modell der Regelstrecke. Wenn eine Regelstrecke eine Übertragungsfunktion gleich $W(p)$ besitzt, ist die

„Übertragungsfunktion“ des NFN $\approx \frac{1}{W_{NFN}(p)}$ und im geschlossenen Regelkreis gilt

auch $W_{sys} = \frac{1}{W_{NFN}(p)} W(p) \approx 1$. Obwohl diese Bedingung nicht vollständig erfüllt wird, kommt es bei

der Anpassung in jeder Abtastzeit zu einer dynamischen Stabilität.

Die Bedeutungen der in Abbildung 54 verwendeten Bezeichnungen sind: U_{soll} - gewünschtes Ausgangssignal; τ_d - Schubleistung des Vorschubantriebs; U_R - Regelungssignal; \tilde{U}_R - Ausgangssignal des NFN beim Anpassungsprozess; Δ - ein Block der Verzögerung; ε - ein Gütekriterium für das NFN; Regler – ein PID-Regler. Der Umschaltungsblock schaltet das inverse Modell im Regelkreis ein, wenn der erste Anpassungszyklus vorbei ist.

Zu Beginn der Arbeit, sind die Regelstreckeparameter unbekannt. Das NFN ist noch nicht angepasst und kann nicht die gewünschte Regelungsqualität gewährleisten. Das System soll eine Anfangszeit für das NFN liefern. Ein PID-Regler übernimmt solange die Regelung, bis die Parameter der Regelstrecke durch das NFN erstmalig geschätzt worden sind. Diese Schätzdaten werden in einer Datei gesammelt, um das NFN zu trainieren. Danach wird das NFN erstmalig angepasst, indem ein lokales inverses Modell der Regelstrecke entworfen wird. Jetzt schaltet der Umschaltungsblock dieses inverse Modell in den Regelkreis ein und deaktiviert gleichzeitig den PID-Regler.

Um eine Echtzeitanpassung vornehmen zu können, verwendet man gleichzeitig zwei NFN. Das erste Netz wurde vollständig als inverses Modell trainiert und die inneren Parameter des ersten NFN wurden in das zweite NFN übernommen, welches im geschlossenen Regelkreis integriert ist. Nach dem

nächsten Anpassungszyklus wird die Übernahme der Parameter erneut wiederholt. Die ermöglicht eine Netzanpassung, ohne den Regelkreis unterbrechen zu müssen.

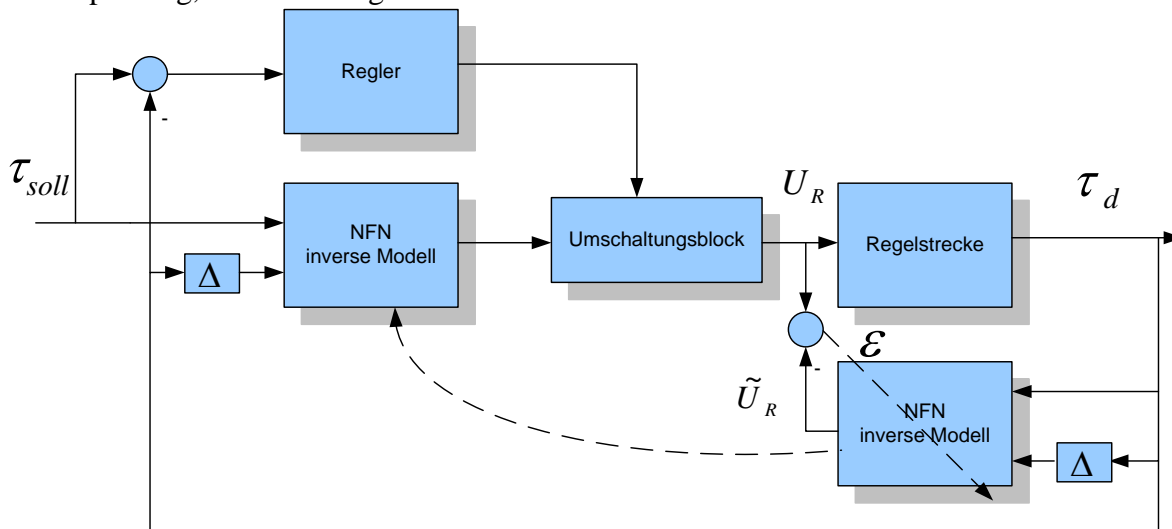


Abbildung 54. Strukturschema der Regelungssysteme bei direkter, inverser Regelung

Wie es zuvor geschildert wurde, muss man für die NFN-Anpassung eine Trainingsdatei vorbereiten, die Werte für U_R enthält, welche in einem vorher abgestimmten Bereich liegen. Für stationäre Systeme, in denen NFN offline trainiert werden, lässt sich dies leicht lösen, weil man die Trainingsmenge frei auswählen kann. Bei Echtzeitanwendungen kommt es zu einigen Problemen, weil U_R ein bestimmtes Regelungssignal ist, das eine entsprechende Regelgüte liefern soll. Deswegen treten bei der Arbeit in diesem System solche Situationen auf, in denen das NFN Eingangssignale erhält, die nicht zum abgestimmten Signalbereich für die NFN-Anpassung gehören. In diesem Fall kann das NFN nicht die gewünschte Regelungsqualität gewährleisten.

Um dieses Problem zu beheben, wird nachfolgend ein neues Verfahren vorgestellt. Die Trainingsdatei ist mit Hilfe von Prognosedaten der Regelstrecke zu erweitern. Dann besteht das erste Massiv $H \in \mathbb{R}^{m \times s}$ aus Messdaten. Das zweite Massiv $P \in \mathbb{R}^{p \times s}$ besteht aus Prognosedaten und basiert auf dem ersten Massiv. Es sagt die Regelstreckenbewegung in einem kleinen Zeitabschnitt vorher.

Die Daten, die zu einem bestimmten Zeitabschnitt gehören, schreibt man in eine Zeile des Datenmassivs. Wenn das ganze Datenmassiv H voll ist, werden alle Zeilen beim nächsten Zeitabschnitt eine Zeile nach unten verschoben. Die letzte Zeile fällt aus der Trainingsmenge heraus und die aktuellen Daten werden in die erste Zeile geschrieben.

Um Rechenleistung zu sparen wird die Anpassung des NFN nicht in jedem Zeitabschnitt durchgeführt, sondern erst nach einem Zeitabschnitt für $p > l$. Die Zahl p hängt von der Leistung des Rechners und von den dynamischen Eigenschaften der Regelstrecke ab. Auf diese Weise wird das, für eine Anpassung verwendete Massiv H_k (k – natürliche Zahl), durch ein neues Massiv H_{k+1} ersetzt. Das Massiv H_{k+1} wird beim nächsten Anpassungszyklus verwendet. Abhängig von den Werten m und p können einige Daten in den Zeilen der Massive H_k und H_{k+1} gleich sein.

Vor Beginn eines neuen Anpassungszyklus, ist jedes Massiv H_k ($k = 1, 2, 3, \dots$) mit dem Massiv P_k zu ergänzen, welches aus Prognosedaten besteht. Eine Vorhersage wird für p Schritte in die Zukunft gemacht.

In Abbildung 55 sieht man die Zeitreihenfolge der Spalten von den Massiven H_k und P_k ($k = 1, 2, 3, \dots$). Das Strukturschema wurde in Abbildung 54 dargestellt. Beim Aufbau der Massivspalte H_k^1 wurden die Daten benutzen, die als „volle Linie“ in Abbildung 55 abgebildet ist. Dazugehörige Prognosedaten wurden mit einer Strichlinie gekennzeichnet.

Wenn das System mit der Arbeit ($t_0 = 0$ auf der Abbildung 55) beginnt, fängt ebenso der PID-Regler an zu arbeiten. Die Regelstrecke bekommt das Signal U_R und die Daten werden mit der Abtastzeit T_0 im Massiv H_1 Zeile für Zeile gespeichert. Wenn die Zeit $t_1 = m \cdot T_0$ erreicht ist, steht das Massiv H_1 zur Verwendung bereit. Gleichzeitig wird ein Prognosemassiv (Extrapolation) aus dem erhaltenen Massiv H_1 für jede Spalte, mit Methode der kleinsten Quadrate, berechnet. Für die Extrapolation wurde das polynomische Modell für die Datenabhängigkeit verwendet.

Dieser Prozess wurde in der Zeit $t_1^e = t_1 + t_e$ beendet, wobei t_e die Extrapolationszeit und die Zeit zum Schreiben des Massives P_1 ist. Nach dem Aufbau der Massives P_1 und H_1 ist das System für den ersten Anpassungszyklus bereit. Das Training des NFN wird in der Zeit $t_1^a = t_1^1 + t_a$ angefangen, wobei t_a die für die Anpassung des NFN benötigte Zeit ist. Das Gütekriterium des NFN wird dabei kleiner als ein zugelassener Wert $\varepsilon < \varepsilon_z$. Hier ist ε_z ein kleiner Wert, der für das Stoppkriterium benötigt wird. Nach der Anpassung des inversen NFN-Modells wurde dieses, mit Hilfe eines Umschaltblocks, in den Regelkreis (siehe Abbildung 54) eingeschaltet und der PID-Regler bis zum Ende der Arbeitszeit ausgeschaltet.

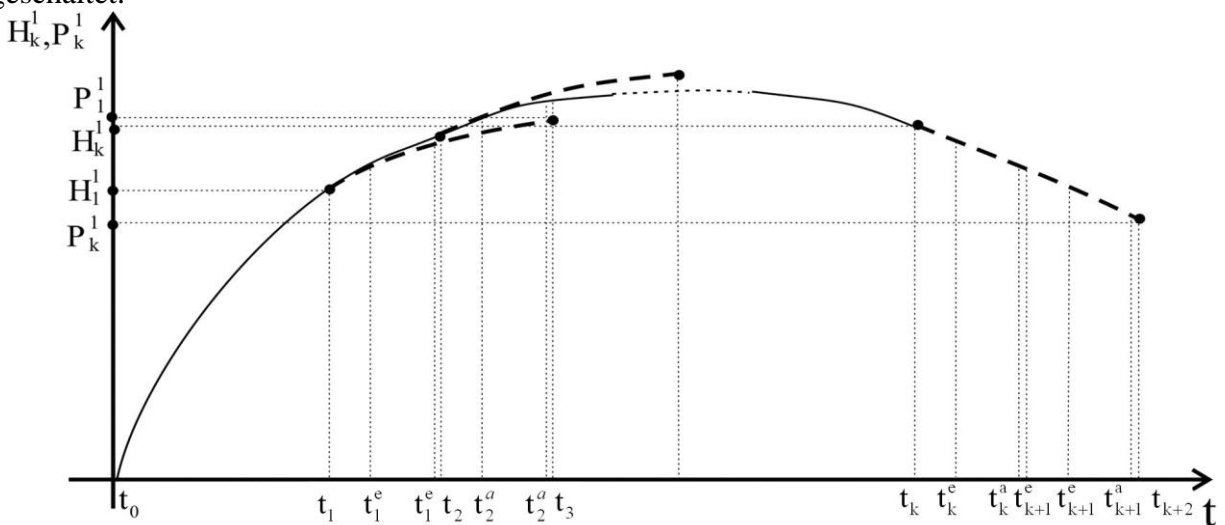


Abbildung 55. Die Zeitreihenfolge bei Dateisammlung in Massiven H_k und P_k

Es ist noch einmal festzuhalten, dass das Massiv H nach jeder Zeit $T_0 \cdot h \geq t_1^a - t_1$ (h ist die kleinste Zahl, für die die Ungleichung erfüllt ist) neue Daten zu jedem Schritt T_0 erhält, welche in der ersten Zeile des Massivs gespeichert wird. Alle Daten in dem Massiv H werden um eine Zeile nach unten verschoben, um die erste Zeile für neue Daten freizumachen. In der Zeit $t_2 = t_1 + T_0 \cdot h \geq t_1^a$, mit $t_2 - t_1^a < T_0$ (siehe Abbildung 55) erfolgt die Speicherung des Massives H_2 und danach setzt der Extrapolationsalgorithmus ein. Die erhaltenen Prognosedaten wurden in dem Massiv P_2 gespeichert. Zum Zeitpunkt $t_2^e = t_2 + t_e$ ist das Auffüllen von P_2 abgeschlossen und das System beginnt den nächsten Anpassungszyklus. Ab dem Zeitpunkt $t_3 = t_2 + T_0 \cdot h \geq t_2^a$ tritt die Speicherung des erneuerten H_3 Massives ein und der ganze Prozess wiederholt sich.

Untersuchungen haben ergeben, dass die Extrapolationszeit und die NFN-Anpassungszeit ($t_e + t_a$) für unterschiedliche Zeitabschnitte fast gleich sind, weil der Wert h konstant bleibt. Wenn das NFN in der Zeit t_k in dem System arbeitet, welches mit den Massiven H_{k-1} und P_{k-1} trainiert wurde, kann ein neuer Zyklus nur bei einem hinreichend großen Prognosehorizont problemlos anfangen werden. Andererseits soll der Prognosehorizont relativ zuverlässig sein.

Bei diesem Arbeitsalgorithmus gewährleistet die Konvergenz des Gütekriteriums ε , siehe Abbildung 54, die Regelungsqualität.

Wie nachfolgend gezeigt wird, stellt dieses Verfahren die gewünschte Regelgüte sicher, obwohl eine Abweichung zwischen dem realen Systemverhalten und dessen Prognose besteht. Dies kann man damit erklären, dass die Prognoseabweichungen bei einer kleinen Abtastzeit T_0 in einem relativ kleinen

Prognosehorizont für zukünftige p Schritte relativ klein sind. Außerdem ist zu beachten, dass die Parameter m , h , T_0 von den dynamischen Systemeigenschaften, den Besonderheiten der Parameteränderungen, der benötigten Genauigkeit bei der NFN-Anpassung und der Rechnerleistung abhängen.

3.2.3 Simulationsergebnisse des direkt inversen Regelungssystems mit NFN für Vorschubantrieb des UWF mit veränderlichen Parametern

In der Arbeit wurde der Levenberg-Marquardt-Algorithmus für die Echtzeitanpassung der Zugehörigkeitsfunktionen verwendet. Dafür wurde die Methode der kleinsten Quadrate bei der Anpassung der Ausgangsschicht für das NFN benutzt. Bei der Simulation in MATLAB wurde eine Abtastzeit von 0.1 ms gewählt.

Weil die Regelstrecke nur einen Eingang U_R und einen Ausgang τ_d besitzt, haben die Massive H_k und P_k lediglich zwei Spalten. Die Zeileanzahl m im Massiv H_k wurde als 32 angenommen. Wie es die Simulation gezeigt hat, war für eine hochgenaue Arbeit des Systems eine kleine Optimierungszeit des NFN ausreichend.

Bei der Simulation wurde ein Vorschubantrieb, siehe Gleichung (1), mit den folgenden Parametern modelliert:

$K_u = 20$, $F_\tau = 4 \text{ H}\cdot\text{s}^2/\text{m}$, $H = 0.12 \text{ m}$, $\delta_H = 0.002 \text{ m}$, $K_m = 0.5 \text{ H}\cdot\text{m}/\text{A}$, $K_\omega = 0.5 \text{ H}\cdot\text{m}/\text{B}$, $C_r = 0.12$, $S_d = 0.01\text{m}^2$, $C_f = 0.002$, $C_\omega = 0.01$, $C_m = 0.001$, $R_d = 2 \text{ Ohm}$. Die Parameter F_m und J_d

Folgen dem Änderungsgesetz: $F_m = F_{mn} + k_1\omega^2$, $J_d = J_{dn} + k_2\omega^2$, wobei $F_{mn} = 0.075 \text{ H}\cdot\text{c}^2$ – der nominale Momentenkoeffizient; $J_{dn} = 0.015 \text{ kg}\cdot\text{m}^2$ – das nominale Trägheitsmoment des Vorschubantriebs; $k_1 = 10^{-5} \text{ H}\cdot\text{s}^4$, $k_2 = 3\cdot 10^{-5} \text{ s}^2\cdot\text{kg}\cdot\text{m}^2$ – die Konstanten des Änderungsgesetzes sind.

Das inverse NFN-Regelstreckemodell mit Optimierung arbeitet in Echtzeit.

In Abbildung 56 sind die Ergebnisse der NFN-Simulation dargestellt, welche als ein inverses Modell der Regelstrecke in Echtzeit trainiert wurden. Einen solchen Trainingstyp des NFN kann man als „Batch training“ bezeichnen, wie man es im Kapitel 1 sieht. Die Kurve 1 stellt das Signal $U_R = \sin(6t)$ dar und die Kurve 2 zeigt \tilde{U}_R – den Ausgang des NFN. Aus dieser Abbildung erkennt man, dass eine Fehlerakkumulation zur „Zykluszeit“ des NFN beim Abschluss eines Anpassungszyklus ($t = 0.4\text{s}$, $t = 0.45\text{s}$, $t = 0.5\text{s}$), wegen den Parameteränderungen in der Regelstrecke und den Prognoseabweichungen auftritt. Dieses Fehlerwachstum wurde mit einem neuen Anpassungszyklus des NFN behoben, wenn die alten NFN-Parameter durch die neuen Parameter ersetzt werden. Die Abtastzeit bei der Simulation beträgt $T_0 = 0.001\text{s}$ und die Zeit des Anpassungszyklus ist $t_{k+1} - t_k = 0.05\text{s}$.

Abbildung 56 zeigt den Vergleich zwischen dem inversen NFN Modell der Regelstrecke und dem realen Systemverhalten.

Das Fehlerniveau hängt von der Abtastzeit ab. So fällt der Fehler zwischen U_R, \tilde{U}_R bei $T_0 = 10^{-4} \text{ s}$ mehr als in 5 Mal so hoch aus, als bei $T_0 = 10^{-3} \text{ s}$. Damit liefert das System bei $T_0 = 10^{-3} \text{ s}$ eine annehmbare Regelungsqualität.

Durch die in Abbildung 56 erkennbaren Optimierungsfähigkeiten und die Prognosequalität des NFN kommt man zu dem Schluss, dass die Anwendung eines NFN als inverses Modell der Regelstrecke zu einer guten Regelungsqualität des geschlossenen Regelkreises führen kann. Die Untersuchungsergebnisse des Regelungssystems des Vorschubantriebs des UWF, siehe Gleichung (1), welche auf der Anwendung der NFN als inverses Regelstreckemodell basiert, sind in Abbildung 57 dargestellt. Die Kurve 1 stellt das Signal $U_{soll} = \sin(6t)$ dar und die Kurve 2 zeigt τ_d – den Ausgang der Regelstrecke. Beide Kurven sind normiert. Für U_{soll} ist der Normierungskoeffizient $1/24 [1/\text{V}]$ und

der entsprechende Koeffizient der Schubleistung τ_d des UWF ist gleich $1/91$ [1/H]. In Abbildung 58 ist das Systemverhalten im geschlossenen Regelkreis bei zusätzlicher Glättung der Trainingsdaten

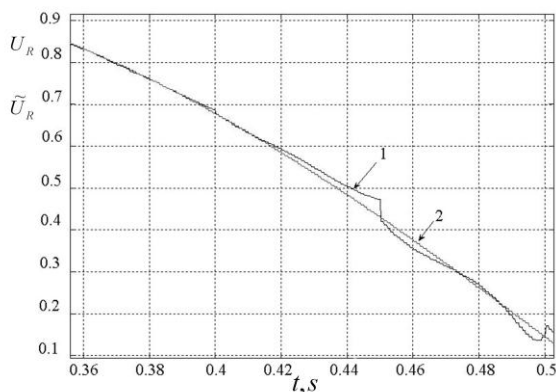


Abbildung 56. Vergleich des NFN als inverses Modell der Regelstrecke mit dem realen Systemverhalten.

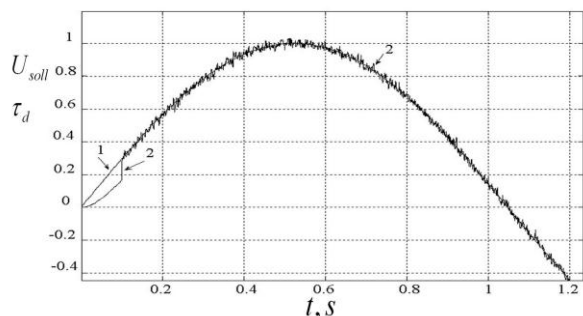


Abbildung 57. Übergangverhalten des UWF mit direkter inverser Regelung.

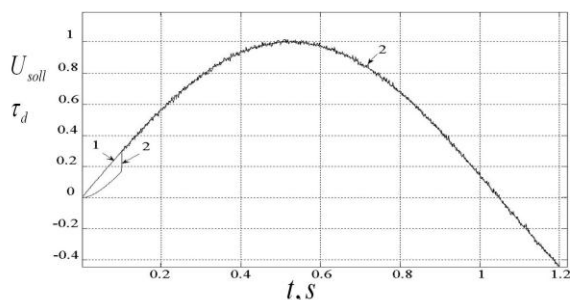


Abbildung 58. Übergangverhalten des UWF mit direkter inverser Regelung bei Trainingsdatenglättung.

für die nächsten Anpassungszyklen der NFN.

Abschließend kann man sagen, dass sich ein ANFIS-NFN als inverses Modell der Regelstrecke im geschlossenen Regelkreis wie vorgesehen verhält. Eine Besonderheit des vorgeschlagenen Regelungskonzepts ist, dass der Regler ohne Kenntnis über die Regelstrecke entworfen werden kann. Das inverse NFN-Modell wurde in Echtzeit mit den neuen Arbeitsdaten angepasst. Dies ermöglicht eine Regelung mit variablen Parametern und nichtlinearen Abhängigkeiten.

dargestellt. Für die Glättung der Trainingsdaten in dem Massiv H_k ist das arithmetische Mittel jeder Spalte zu berechnen und danach wird jede Spalte durch ihr arithmetisches Mittel dividiert.

Durch einen Vergleich der Abbildungen 57,58 ist festzustellen, dass die Regelungsabweichung zwischen τ_{soll} und τ_d den Faktor 2,5 hat. Dies bestätigt, dass eine Minderung des Hochfrequenzanteils in den Signalen τ_d und U_R die Regelgüte des geschlossenen Regelkreises verbessert.

Aus Abbildung 56 geht hervor, dass anstatt des inversen NFN-Modells ein PID-Regler die Regelung bis zum Zeitpunkt $t \approx 0.1s$ übernommen hat. Dieser sorgt für die Anfangsdaten, die für die Anpassung des NFN nötig sind. Die Regelstrecke bleibt in dieser Zeit stabil, erreicht jedoch keine gute Regelungsqualität in diesem Zeitabschnitt. Ab dem Zeitpunkt $t \approx 0.1s$ wurde das inverse NFN-Modell in den Regelkreis eingeschaltet. Als Abtastzeit wurde $T_0 = 0,1$ ms gewählt. Wie man weiter sieht, hat die direkte inverse Regelung bei der Anwendung in NFN für geringe Abweichungen gesorgt. Trotzdem ist nicht zu vergessen, dass das NFN-Modell als inverses Modell der Regelstrecke trainiert wurde. Deswegen bekommt das NFN automatisch Differenziereigenschaften und beim Ausgang des NFN können Hochfrequenzsignale erscheinen. Die dargestellten Eigenschaften führen zur Verschlechterung der Regelgüte und erhöhen beim Ausgangssignal den Hochfrequenzanteil. Das Regelungssignal des NFN ist dem Regelungssignal eines Zweipunktreglers ähnlich. Der Hochfrequenzanteil im Ausgangssignal der Regelstrecke verschlechtert die Trainingsdaten

Bei Untersuchungen an diesem Regelungssystem wurde ein höchst effizientes Trainingsverfahren entwickelt, welches eine sehr schnelle Konvergenz des NFN-Anpassungsprozesses gewährleistet und gleichzeitig eine minimale Berechnungsleistung des Rechners benötigt.

Das entwickelte Verfahren ist für das Training des inversen Modells der Regelstrecke geeignet. Es besitzt die Besonderheit, dass reale und extrapolierte Messdaten kombiniert werden können und sich so die Anzahl der Basisdaten erhöht. Diese spielt eine Schlüsselrolle bei der Versorgung des adäquaten inversen Modells der Regelstrecke für einen lokalen Zeitbereich.

Mathematische Simulationen und Untersuchungen an diesen Systemen weisen eine hohe Effizienz bei technischer Anwendung in Regelungssystemen für Regelstrecken mit veränderlichen Parametern und Nichtlinearitäten auf.

3.3 Direkte adaptive Regelung mit Prognose für den Vorschubantrieb des UWF

Wie man im vorigen Kapitel sieht, hat die direkt inverse Regelung wesentliche Nachteile. Der bedeutendste Nachteil sind die großen Hochfrequenzstörungen des Ausgangs, was in den Abbildungen 57,58 dargestellt wurde. Es gibt jedoch andere Verfahren, die diesen Nachteil nicht besitzen.

Der Hauptgrund der Schwankungen ist die Verwendung eines dynamischen inversen Modells der Regelstrecke. Da die Schaltung Integratoren enthält, benötigt das inverse Modell unbedingt Differenzierungselemente. Dies führt zur Verschlechterung der Regelgüte[16]-[19].

Um die genannten Probleme zu lösen, wird ein neues Verfahren vorgeschlagen, das kein inverses Modell der Regelstrecke verwendet, aber trotzdem auf Verhaltenprognosen der Regelstrecke in relativ kleinen Zeitabschnitten basiert.

Das Verfahren besteht aus den folgenden Schritten:

1. Es ist ein lineares Modell der Regelstrecke im Form eines IIR (infinity impulse response) Filter in Echtzeit anzupassen, was mit der Methode der kleinsten Quadrate gemacht wird.
2. Zu jeder Abtastzeit verwendet man eine „virtuelle Zeit“ und berechnet den IIR-Filter als Regelstreckenersatz mit NFN als Regler im geschlossenen Regelkreis, um die Koeffizienten der NFN so anzupassen, dass ein Minimum des Regelgütekriteriums gefunden wird.
3. Das angepasste NFN wird als Regler im realen Regelkreis verwendet.

Das Strukturschema des vorgeschlagenen Regelungssystems ist in Abbildung 59 dargestellt.

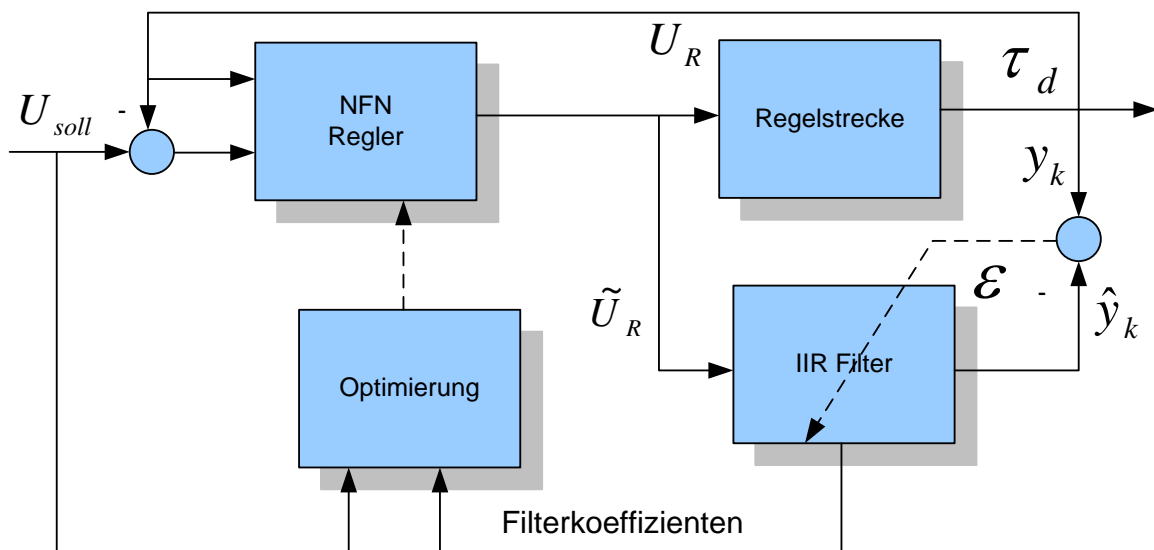


Abbildung 59 . Das Strukturschema der Regelungssysteme bei adaptiver Prognoseregelung

Im Unterschied zum vorheriger System haben NFN nur 2 Eingänge: einen Eingang für den Sollwert und einen zweiten Eingang für den Regelungsfehler. Die Eingangsanordnung hängt von der Regelstrecke ab und kann von Regelstrecke zu Regelstrecke variieren.

Das NFN, welches in diesem Verfahren getestet wurde, wird als ANFIS (Adaptive Neuro Fuzzy Inference System) [9]- [11] klassifiziert. Die mathematische Beschreibung wurde im Kapitel 1 ausführlich betrachtet. Die Abhängigkeit des Ausgangsignals des ANFIS-NFN wurde mit den Gleichungen aus Kapitel 1 (6-8) berechnet.

Der lineare IIR-Filter lässt sich in digitaler Form folgendermaßen darstellen[20]-[22]:

$$\hat{y}(k+1) = \sum_{i=0}^M \theta_{1i}(k) \cdot X_i(k) + \sum_{j=0}^L \theta_{2j}(k) \cdot U_j(k), \quad (2)$$

wobei $\hat{y}(k+1)$ das Filterausgangssignal, $\mathbf{X}(k)$ ein Vektor der inneren Filterzustände, $\mathbf{U}(k)$ ein rekursiver Eingangsvektor und θ ein Vektor des Filterkoeffizienten sind.

In Gleichung (2) hängt der erste Summand von den inneren Filterzuständen ab und die Koeffizienten θ_{1i} sind Rückkopplungskoeffizienten des Filters. Deswegen sind diese die Nennerkoeffizienten der Filterübertragungsfunktion. Die Koeffizienten θ_{2j} sind beim zweiten Summand die Koeffizienten im Zähler der Filterübertragungsfunktion. Die Filterordnung soll nicht geringer als die Ordnung der Regelstrecke sein.

Ein relativ genaues Modell der nichtlinearen Regelstrecke mit Hilfe von IIR-Filtern kann man nur für einen beschränkten Zeitabschnitt realisieren, weil IIR-Filter eine lineare Systemdarstellung sind. Bei einem Online-Anpassungsalgorithmus ändern sich die Koeffizienten des IIR-Filters nach der Methode der kleinsten Quadrate und das Verhalten des IIR-Filters nähert sich dem Verhalten der nichtlinearen Regelstrecke an. Es ist ebenfalls möglich ein gradientenbasiertes Optimierungsverfahren für die IIR-Filterkoeffizientenanpassung zu verwenden [20]. Die gradientenbasierte Optimierung ist grundsätzlich nicht anders als die Verfahren, welche in Kapitel 1 betrachtet wurden.

Zu Beginn funktioniert das NFN als ein P-Regler und das Regelungssignal U_r bringt die Regelstrecke in Bewegung. Die Anfangskoeffizienten des Filters sind zufällig initialisiert worden. Sobald der Arbeitsbetrieb beginnt, wird der IIR-Filter angepasst. Der Anpassungsprozess wird zu jeder Abtastzeit durchgeführt und der Übergangsprozess sorgt dafür, dass der IIR-Filter sich mit jedem Schritt, mit jeder neuer Abtastzeit dem Regelstreckeverhalten annähert.

Bei der gradientenbasierter Anpassung der Filterkoeffizienten wurde die folgende Formel verwendet:

$$w_{k+1} = w_k - \eta \frac{\partial J}{\partial w_k}, \quad (3)$$

wobei J ein Gütekriterium ist.

Für adaptive IIR-Filter ist das Gütekriterium nach der Gleichung $J_{af} = (\hat{y}_k - y_k)^2$ berechnet worden, wobei \hat{y}_k, y_k die Signale an den Ausgängen der Regelstrecke und des adaptiven Filters sind. Das Anpassungsgesetz Gleichung (2) führt zur Minimierung des Gütekriteriums.

Gleichzeitig beginnt der zweite Teil des Regelungsalgorithmus, welcher die Anpassung der NFN-Reglerkoeffizienten vornimmt. Das Schema der NFN-Anpassung ist in Abbildung 60 dargestellt.

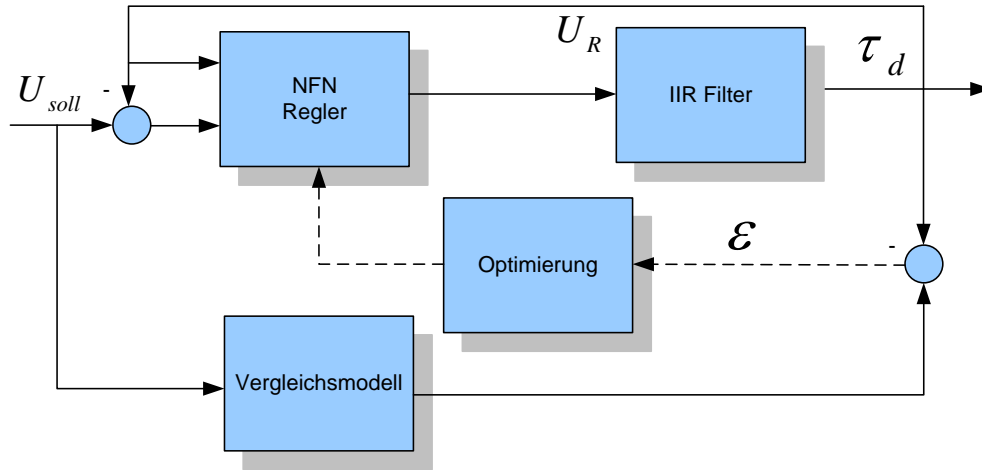


Abbildung 60. Das Schema der NFN Anpassung

Bei dem NFN-Training wurde $U_{soll} = const$ angenommen. Die Anpassung dauert einige Zyklen und während der NFN-Anpassung bleiben die NFN-Koeffizienten für den realen Regelkreis konstant. Sie werden nur am Ende des Anpassungsprozesses verändert. Dieser Prozess dauert für reale Systeme eine Abtastzeit, trotzdem läuft die Anpassung der NFN in einer virtuellen Zeit. Der Regelkreis wird im Laufe der Anpassung des NFN mit dem IIR-Filter betrieben.

Die Besonderheit in diesem Anpassungsprozess ist das Gütekriterium:

$$J_{nfl} = \frac{1}{2} (r_{k+1} - \hat{y}_{k+1})^2 Q + \frac{1}{2} u_k^2 R, \quad (4)$$

wobei \hat{y}_{k+1} das IIR-Filterausgangssignal; R ein Strafkoeffizient für das Regelungssignal u_k , Q ein Strafkoeffizient für Abweichungen der Regelstrecke von einem Vergleichmodell, k die Nummer der Zeitdiskrete sind. Die Koeffizientenanpassung wird bis zum Zeitpunkt $k \geq f$ vom Gütekriterium Gleichung (4) übernommen, wobei f die gesamte Anzahl der IIR-Filterparametern ist. Nach f Schritte kann man die Methode der kleinsten Quadrate verwenden.

Die Zustände X des IIR-Filters werden in dem Massiv A zusammen mit dem Filtereingangssignal u gesammelt. Jede Zeile in der Matrix A kann als $a_i = [X_{k-i}^T, u_{k-i}]$, $i = \overline{0, \dots, f-1}$ dargestellt werden. Der Ausgang der Regelstrecke wurde im Vektor $Y = [y_k, y_{k-1}, \dots, y_{k-(f-1)}]$ gespeichert. Das Ziel der Filteranpassung $\hat{Y} \cong Y$ kann die folgende Matrixberechnung gewährleisten. Der Filter in Matrixform [22] lässt sich mit

$$\hat{y} = c_k \theta \quad (5)$$

beschreiben, wobei $c_k = [X_k^T, U_k^T]$ und X_k^T Filterzustandsvektoren und U_k^T der Eingangrekursivvektor sind.

Der IIR-Filter hat eine bestimmte Anzahl von Parametern, welche man in einem Schritt bestimmen kann, da es sich um ein lineares System handelt.

Zunächst werden f Schritte der Zeitreihe betrachtet. Die Gleichung (5) in Matrixform ist:

$$\hat{Y} = C\theta, \quad (6)$$

Wobei die Matrix C aus den Zeilen c_k besteht.

Jetzt kann man $\hat{Y} \cong Y$ annehmen, was zu der Matrixgleichung führt:

$$Y = C\theta \quad (7)$$

Die Lösung der Gleichung (7) kann somit wie folgt dargestellt werden:

$$\theta = C^{-1}Y. \quad (8)$$

Nach dem Einsetzen der Filterkoeffizienten beginnt die Anpassung des NFN-Reglers. Die NFN-Optimierung läuft im geschlossenen System, siehe Abbildung 60. Der größte Unterschied im Vergleich mit bekannten Systemen besteht in der Einführung einer virtuellen Zeit, welche in dem Rechner das zukünftige Regelstreckenverhalten für eine bestimmte Zeitdauer liefert. Dies erlaubt eine Art Offline-Optimierung (batch training) auf eine Reihe zeitdiskreter Daten anzuwenden. Die Dauer der zukünftigen Zeit (in den Zeitabschnitten) heißt Prognosehorizont. Dieser hängt von den Besonderheiten der Regelstrecke und der Rechnerleistungen ab. Die Anpassung der NFN wurde mit Gradientenabstiegsverfahren durchgeführt. Dabei berechnet man das Gütekriterium mit:

$$J_{nf2} = \frac{1}{2} \sum_{i=k}^{k+H_p} (r_i - \hat{y}_i)^2 Q_i + \frac{1}{2} \sum_{j=k-1}^{H_c} \Delta u_j^2 R_j, \quad (9)$$

wobei H_p der Prognosehorizont, H_c der Regelungshorizont (die Menge des diskreten Schritte des Regelungsprozesses), $\Delta u = u_j - u_{j-1}$, $Q \in \mathbb{R}^{1 \times H_p}$ ein Vektor der Strafkoeffizienten für die Abweichungen des Regelstreckenausgangs vom Vergleichsprozess, $R \in \mathbb{R}^{1 \times H_c}$ ein Vektor der Strafkoeffizienten für das Regelungssignal sind.

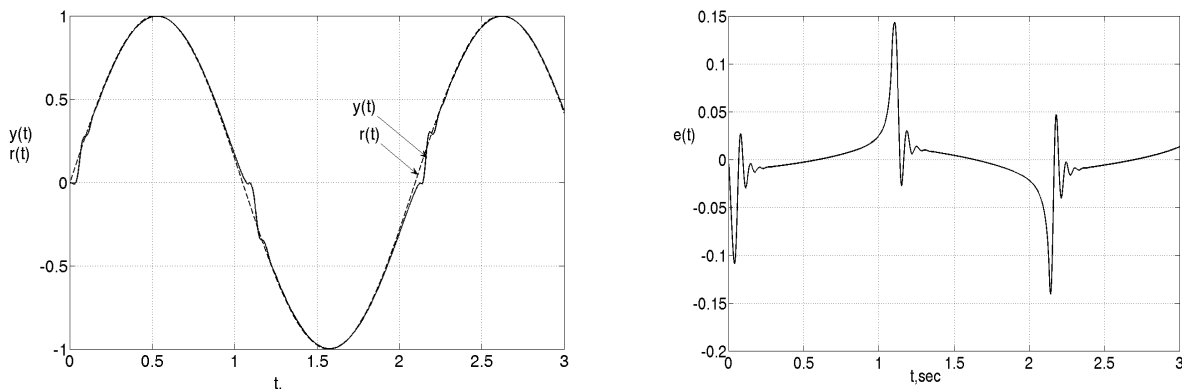
Die Gleichung (9) erfasst sowohl ältere Daten als auch die vorhersagten Daten. Deswegen liefert dieses Gütekriterium aktuellere Daten im Vergleich mit Verfahren ohne Prognose.

Als Vorteil im vorgeschlagenen Algorithmus kann man die Gewährleistung der Regelgüte bei wesentlichen, dynamischen Parameteränderungen wegen der Anpassung des NFN-Reglers in Echtzeit nennen. Die Prognose verbessert erheblich die Regelgüte und Anpassungsfähigkeit ohne Anwachsen der Lernkoeffizienten. Dies ist wichtig, weil ein Anwachsen der Lernkoeffizienten zur Instabilität des Regelkreises führen kann. Das Gütekriterium Gleichung (9) ist zur Berücksichtigung der Höhe des Regelungssignals fähig.

Als Nachteil von diesem Algorithmus sind der Berechnungsaufwand, die Berücksichtigung des Prognosehorizontes, die Filterordnung, und benötigte Koeffizienten wie den Lernkoeffizienten, von welchen die Qualität des Systems abhängt, zu nennen.

3.3.1 Die Simulationsergebnisse

Im dem System, das in Abbildung 61 dargestellt ist, wurden folgende Parameterwerte bei der Simulation ausgewählt: Prognosehorizont $H_p=5$, Regelungshorizont $H_c=1$.



a) b)
Abbildung 61. Simulationsergebnisse im System mit direkter adaptiven neuro-fuzzy Regelung und Prognose

Der Übergangsprozess ist in Abbildung 61 gezeigt.

In Abbildung 61a wurden das Sollsignal $r(t)=\sin(6t)$ und das Ausgangssignal der Regelstrecke (die Trägheitskraft) gezeigt. Auf Abbildung 61b sieht man den Fehlerübergangprozess $e(t)=r(t)-y(t)$. Durch die vorgestellten Abbildungen kommt man zum Schluss, dass die Regelungsfehler immer weniger als 3% vom Sollwert betragen. Der Fehler wächst nur beim Annähern des Nullbereichs schnell an, weil die Parameter der Umgebung sich stufenartig ändern beim Rückwärts-Betrieb des Vorschubantriebs von UWF. Trotzdem endet die Fehlerunterdrückung nach 0,2 s bei der Regleranpassung.

3.4 Indirekte adaptive Regelungen für den Vorschubantrieb des UWF mit Hilfe von Echtzeitzustandreglern

Es gibt eine Reihe von Reglerentwurfsverfahren mittels Rückkopplung des Zustandsvektors. Wenn man das Systemmodell kennt, ist es leicht die gewünschte Regelungsqualität mit Hilfe der Angleichung des charakteristischen Systempolynoms auf ein gewünschtes Systempolynom zu erhalten. Wenn man dasselbe Verfahren der Polstellen für eine Echtzeitanpassung verwendet, erlangt man ein noch leistungsstärkeres adaptives Regelungssystem [3, 4, 9]. Dieses Verfahren ist für lineare Systeme geeignet, die veränderliche Parameter haben.

Diese Idee funktioniert nur gut, wenn das Systemmodell bekannt und linear ist und sich die Parameter des Systems auf eine bestimmte Weise ändern. Wenn das Systemmodell nicht bekannt ist, kann die gewünschte Regelungsqualität nicht gewährleistet werden.

Um diese Probleme zu lösen, muss ein lokales Echtzeitmodell der Regelstrecke abgeschätzt werden. Wenn dieses Modell das Systemverhalten gut beschreibt, kann man das Verfahren mit Polstellen in Echtzeit verwenden und die gewünschte Regelgüte erreichen.

Bei dieser Methode treten für komplizierte nichtlineare Systeme Probleme mit der Regelstreckenidentifikation auf. Mit einem iterativen Optimierungsverfahren ist es möglich die Identifikation zu verbessern. Dafür braucht man jedoch die Daten vom Übergangprozess der Regelstrecke. Um die Datenmenge zu erhöhen und dadurch die Identifikation der Regelstrecke zu verbessern, wurde folgender Algorithmus entworfen und getestet.

Zuerst bedarf es eines lokalen Regelstreckemodells, welches eine sichere Prognose ermöglicht. Die Prognose ist sehr wichtig, weil das Regelstreckemodell nur „historische“ Daten liefert und den Regler verspätet nachregelt. In diesem Sinn bietet der Prognosemechanismus die Möglichkeit, vorher Regelungshindernisse zu erkennen und deswegen eine präzise Regelung auszuführen.

Der adaptive IIR-Filter ermöglicht ein lineares adaptives Regelungsgesetz zu realisieren. Trotzdem hat dieser Filter selbst Identifikationsprobleme. Da der IIR-Filter Probleme bei der Identifikation hat, wurde ein lokales adaptives ANFIS-NFN-Modell der Regelstrecke verwendet. Die Simulation des NFN-Modell der Regelstrecke in einer virtuellen Zeit gibt die notwendigen Daten über ein mögliches zukünftiges Regelstreckeverhalten an. Genau das braucht man, um ein adäquates lokales lineares Modell der Regelstrecke zu finden.

Der zweite Schritt der Regelungsalgorithmus besteht aus der Platzierung der Polstellen des IIR-Filters, welcher als lokale, lineare Kopie der Regelstrecke betrachten werden kann.

In dem dritten Schritt schließt der Regelkreis die ermittelten Werte ein. Das in dem zweiten Schritt gefundene Reglersignal wirkt auf die Regelstrecke und der ganze Prozess wiederholt sich.

Das Strukturschema wurde auf der Abbildung 62 dargestellt. Hier ist der Regler ein Zustandsregler oder ein Wurzelortsregler, das NFN-Modell ist ein ANFIS-NFN, das als Modell der Regelstrecke funktioniert. Der IIR-Filter ist ein adaptiver Filter.

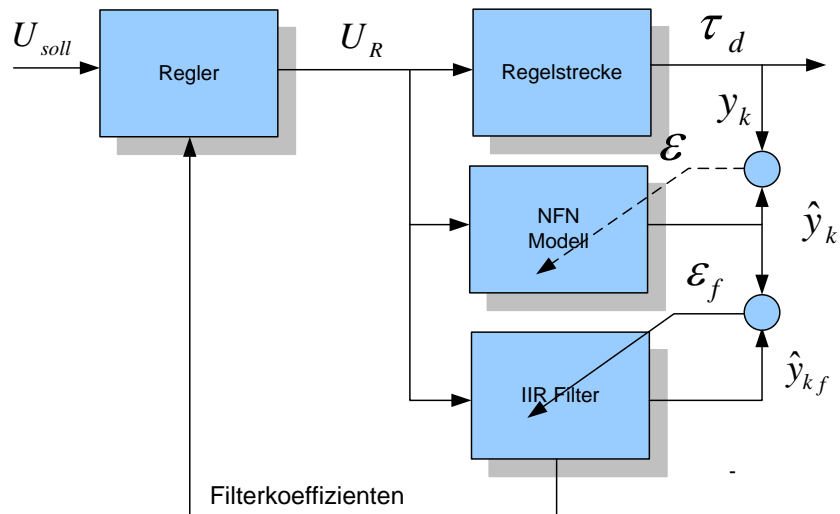


Abbildung 62. Das Strukturschema des Regelungssystem mit NFN Modell der Regelstrecke und der Echtzeitzustandregler

Das NFN benötigt dynamische Eigenschaften des Systems, um ein Modell der dynamischen Regelstrecke bilden zu können. Deswegen ist eine eigene dynamische Rückkopplungen in das NFN realisiert worden. Das Schema des dynamischen NFN ist auf Abbildung 63 zu sehen.

Die Anzahl der verspäteten Signale in dem NFN-Modell wurde gleich der Höhe der Regelstreckenordnung gewählt.

Zum Unterschied zu den vorhergehenden Systemen wurde das NFN nicht als Regler verwendet, sondern als ein nichtlineares lokales Modell der Regelstrecke. Es ermöglicht mehrdeutige Prognose des Regelstreckeverhaltes zu errechnen.

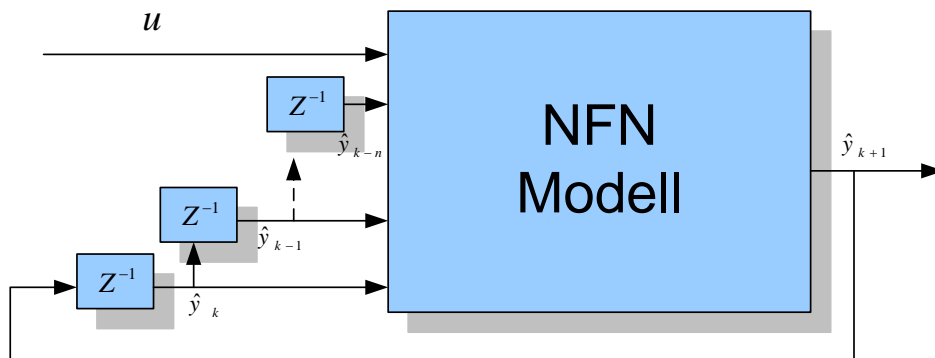


Abbildung 63. Das dynamische NFN-Modell der Regelstrecke

Bei der Anpassung des NFN wurde folgendes Gütekriterium benutzt:

$$J_{nfn} = \sum_{j=1}^F (\mathbf{Y}_i - \hat{\mathbf{Y}}_i)^2, \quad (10)$$

mit $\hat{\mathbf{Y}} \in R^{F \times 1}$ dem Vektor des NFN-Modellausgangs.

Die Minimierung des Gütekriteriums Gleichung (10) wurde mit einem hybriden Optimierungsverfahren des ANFIS erreicht.

In jeder Abtastzeit des Systems wurde eine Prognose mittels NFN-Modell folgender maßen erstellt. Zu den Eingängen des NFN kommen ein Regelungssignal u und N verspätete Signale $\hat{\mathbf{Y}}$. Der

Eingangsvektor des NFN ist somit $[u_k^1, \hat{y}_k^1, \hat{y}_{k-1}^2, \dots, \hat{y}_{k-N}^N]$, wobei der obere Index eine Nummer im Eingangsvektor ausdrückt und der untere Index die Nummer der Zeitdiskreten zeigt. Der Ausgang des NFN wurde mit \hat{y}_{k+1} bezeichnet, weil dieser einen nächsten Wert des Modells darstellt. Danach verwendet man den neuen Eingangsvektor $[u_k^1, \hat{y}_{k+1}, \hat{y}_k^1, \dots, \hat{y}_{k-N-1}^{N-1}]$, um den nächsten Prognosewert \hat{y}_{k+2} zu berechnen. Dieser Zyklus wiederholt sich bis zum Erreichen des Prognosehorizonts, des H_c Schritts. Die Prognosedaten werden in dem Vektor $P = [\hat{y}_{k+H_c}^1, \hat{y}_{k+H_c-1}^2, \dots, \hat{y}_{k+1}^{H_c}]$ gespeichert.

Für einen Zustandsvektorregler wird ein lineares System benötigt. Abgesehen von dieser Tatsache ist das NFN-Modell selbst kein lineares Modell. Für die Berechnung der Polstellen der Regelstrecke in Echtzeit wurde ein IIR-Filter verwendet.

Als Gütekriterium für die Anpassung des IIR-Filters wurde der Quadratfehler in der Form von Gleichung (10) genutzt. Als Optimierungsverfahren wurde die Methode der kleinsten Quadrate implementiert. Im Unterschied zur Anpassung des NFN-Modells, wurde der Vektor Y mit dem Vektor $\hat{Y}_d = [p_1, \dots, p_{H_c}, \hat{y}_1, \dots, \hat{y}_n]$ ersetzt. Zum Eingang des IIR-Filters wurden die Daten $U_f = [u_1^1, u_2^1, \dots, u_{H_c}^1, u_{H_c+1}^1, u_{H_c+2}^2, \dots, u_{H_c+F}^F]$ in einer virtuellen Zeit geschickt. Hierbei bedeutet der obere Index die Ordnungsnummer in einer Zeitreihe des Vektors U .

Nach der Anpassung kann man den IIR-Filter als lokales, lineares Modell der Regelstrecke für weitere Berechnungen benutzen. Der IIR-Filter lässt sich in kanonischer Regelungsform darstellen [20]:

$$\begin{cases} zX = A_f X + B_f U \\ \hat{Y}_f = C_f X \end{cases}, \quad (11)$$

wobei $A_f = \begin{bmatrix} -\theta_{11} & -\theta_{12} & \dots & -\theta_{1M} \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{bmatrix} \in R^{M \times M}, B_f = \begin{bmatrix} 1 \\ 0 \\ \dots \\ 0 \end{bmatrix} \in R^M$, und

$$C_f = [\theta_{21}, \dots, \theta_{2L}, 0, \dots, 0] \in R^M.$$

Wenn man das lineare Modell der Regelstrecke Gleichung (11) verwendet, kann das System in dem geschlossenen Regelkreis wie folgt betrachtet werden:

$$zX = (A_f - B_f K_d)X + B_f U, \quad (12)$$

wobei $K_d = [K_{d1}, K_{d2}, \dots, K_{dM}]$ ein Rückkopplungsvektor ist.

Dann nimmt das charakteristische Systempolynom für das System Gleichung (12) die Form

$$\det | zI - A_f + B_f K_d |, \quad (13)$$

an, wobei I die Einheitsmatrix ist.

Die Hauptaufgabe bei der Reglerberechnung ist es den Rückkopplungskoeffizienten K_d so auszuwählen, dass das Systempolynom gleich einem Vergleichspolynom wird. Dieses Vergleichspolynom M soll die gewünschte Regelungsqualität ermöglichen:

$$M = z^n + a_1 z^{n-1} + \dots + a_{n-1} z + a_n. \quad (14)$$

Die genannte Lösung ist leicht zu finden, wenn man die Koeffizienten K_d so wählt, dass die Koeffizienten der Systempolynome der Gleichungen (13,14), bei gleicher Ordnung z gleich zu einander werden. Die Rückkopplungskoeffizienten berechnet man wie in [22] mit:

$$K_{di} = a_i - \theta_{li}. \quad (15)$$

Auf diese Weise wird das Regelungssignal nach Ermittlung der Rückkopplungskoeffizienten mit Hilfe von Gleichung:

$$u(k) = r(k) - \sum_{i=1}^M K_{di} x_i(k), \quad (16)$$

berechnet.

Der Vorteil dieses Algorithmus ist ein kleinerer Berechnungsaufwand, weil die Reglerberechnung selbst nicht viele Ressourcen braucht. Dieses Verfahren funktioniert gut für die Regelung von unbekanntem nichtlinearen Systemen mit relativ langsam und glatt veränderlichen Parametern. Bei schnellen Systemänderungen können die Systemstellen den Stabilitätsbereich verlassen und das System so instabil werden.

3.4.1 Die Simulationsergebnisse

Den Übergangsprozess sieht man in Abbildung 64.

Auf der Abbildung 64a werden das Sollsignal $r(t) = \sin(6t)$ und das Ausgangssignal der Regelstrecke (die Trägheitskraft) gezeigt. Abbildung 64b stellt den Fehlerübergangsprozess $e(t) = r(t) - y(t)$ dar. Aus den vorgestellten Abbildungen ist zu schlussfolgern, dass die Regelungsfehler immer geringer als 0,5% vom Sollwert bleiben. Der Fehler wächst erst beim Annähern des Nullbereichs schnell an, weil sich die Parameter der Umgebung stufenartig beim Revers-Betrieb des Vorschubantriebes des UWF ändern. Trotzdem endet die Fehlerunterdrückung der Regleranpassung nach 0.5 s.

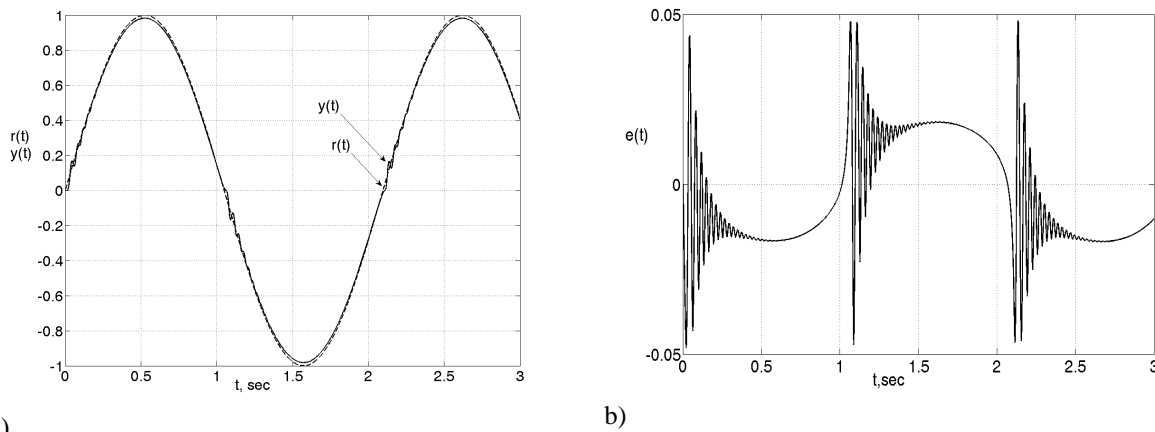


Abbildung 64. Simulationsergebnisse in dem System mit indirekten adaptiven Regelungen für Vorschubantrieb des UWF mit Hilfe von einem Echtzeitzustandregler.

Wie die Untersuchungen gezeigt haben, gewährleisten die beiden der letzten Regelungsverfahren die gewünschte Regelgüte. Die direkte adaptive Regelung mit NFN-Regler und Prognose mit IIR-Filter zeigt eine viel höhere Anpassungsgeschwindigkeit bei Änderung der Regelstreckenparameter. Das Verfahren mit NFN-Prognose der Regelstrecke und einem Echtzeitzustandsregler hat weniger Regelungsfehler, aber eine größere Regelungszeit.

Zusammenfassung

In diesem Kapitel wurden drei unterschiedliche Regelungsverfahren für eine nichtlineare Regelstrecke wie Vorschubantrieb des UWF entworfen und untersucht.

Das erste auf inversen nichtlinearen Modellen der Regelstrecke basierte System hat wesentliche Vorteile im Vergleich mit klassischen Regelungssystemen wie der PID-Regelung, aber wegen seines inversen dynamischen Verhalten besitzt es Stabilitätsprobleme und eine schwache Rauschstabilität.

Das zweite auf der NFN-Regelung mit Prognose mittels IIR-Filter basierte System ermöglicht ein stabileres und genaueres Regelungsverhalten, ist aber für Echtzeitsysteme schwer zu realisieren auf Grund des erheblichen Berechnungsaufwands.

Der dritte Systemtyp hat im Vergleich mit den vorhergehenden Systemen eine gute Regelungsqualität und eine hohe Genauigkeit. Außerdem braucht dieser Systemtyp weniger Berechnungen und ist relativ leicht implementierbar.

Alle drei Systeme haben bekannte Regelungskonzepte verwendet, wurden aber modifiziert, um die typischen Schwächen dieser Methoden zu vermeiden.

Kapitel 4 Die Entwicklung und Regelung einer Roboterhand

Einführung

Die Robotertechnik hat schon eine lange Geschichte. Den ersten Schritt hat vermutlich Leonardo Da Vinci im Jahre 1497 mit dem Modell eines mechanischen humanoiden (menschenähnlichen) Roboters gemacht[1]. In Abbildung 65 kann man eine Rekonstruktion von diesem Mechanismus sehen.

Seitdem hat sich die Robotertechnik in mehrere Richtungen entwickelt. Entsprechend der Aufgabestellung teilt man sie heute beispielweise in Industrieroboter, Spielroboter oder humanoide Roboter ein [2].



Abbildung 65. Robot von Leonardo da Vinci [1]



Abbildung 66. KUKA Industrial Robot KR10 SCARA [3]

Zuerst muss man den Begriff „Industrieroboter“ erklären. Man kann ihn definieren als einen universell einsetzbaren Bewegungsautomat mit mehreren Achsen, deren Bewegungen programmiert werden können und der der Handhabung, Bearbeitung von Werkstücken und Fertigung dient. Im Allgemeinen besteht ein Roboter aus Manipulator, Controllern, Antrieben, Greifern und Sensoren. Unter dem Begriff Manipulator versteht man ein Gerät, welches eine physikalische Zusammenwirkung mit der Umgebung ermöglicht.

Der erste programmierbare Industrieroboter “Unimation robot“ wurde im Jahr 1954 von Georg Devol erfunden. Georg Devol hat in den USA ein Patent für einen programmierbaren Manipulator angemeldet.

Im Jahr 1979 wurde das SCARA-Prinzip (Selective Compliance Assembly Robot Arm) an der Yamanashi Universität im Labor von Professor Hiroshi Makino entwickelt. SCARA-Roboter verfügen in der Regel über vier Achsen (drei parallele Rotationsachsen und eine Linearachse) und vier Freiheitsgrade. Auf Grund ihrer schnellen Bewegung nutzt man SCARA-Roboter speziell für sogenannte „Pick-and-Place“ Anwendungen, sofern es sich nicht um planparallele Flächen handelt, weil ihm für komplexe Aufgaben die nötigen Freiheitsgrade fehlen. Für solche Anwendungen sind nur Fünf- oder Sechs-Achs-Roboter geeignet. Seitdem sind Industrieroboter in einer speziellen Gruppe aufgeführt, die der Warenproduktion dient.

Im Gegensatz dazu überdecken die humanoiden Roboter ein breiten Themenfeld für zahlreiche Forschungen und Entwicklungsprojekte, weil man immer wieder neue Probleme und Herausforderungen zu lösen hat. Dazu gehört unter anderem das Laufen zweibeinigen Maschinen, der Aufbau und die Regelung für humanoiden Hände zur Erfüllung von Serviceoperationen, die Interaktion mit der Umgebung, die Bilderkennung, Spracherkennung, Emotionsanzeige und Emotionserkennung. Der weltbekannteste humanoide Roboter ist in Abbildung 67 dargestellt. Ein besonderes Themenfeld ist die Hand

des humanoiden Roboters.

Anthropomorphe Roboterhände finden als eines der Hauptthemen der Robotertechnik Anwendung in vielen Industriebranchen. Besonders aktuell ist ihre Anwendung in für Menschen schädlicher bzw. gefährlicher Umgebung (beispielsweise bei Strahlung im Weltraum) oder an unzugänglichen oder schwer zugänglichen Orten. Weiterhin hat die Robotertechnik viele Perspektiven in der Medizin z.B. bei der Entwicklung intelligenter Prothesen oder bei der ferngesteuerten Operationsdurchführung.

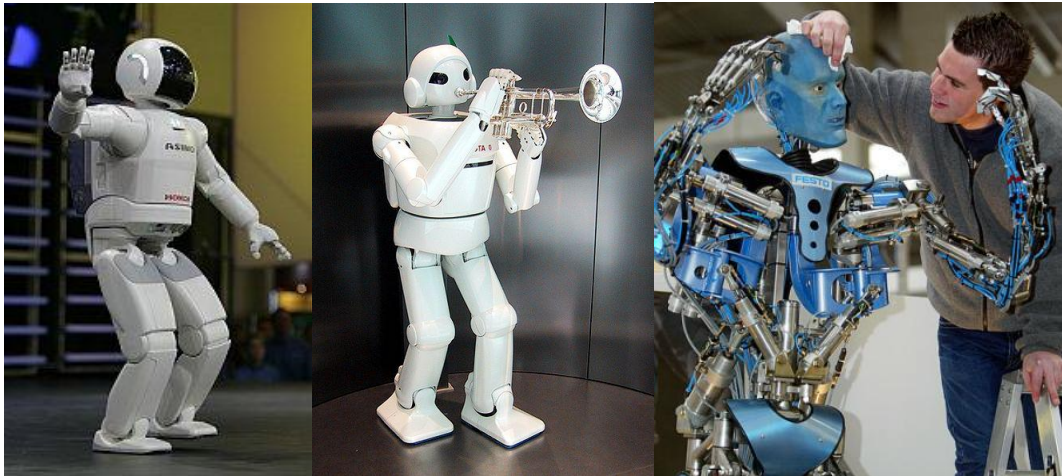


Abbildung 67. Roboter ASIMO[4], humanoider Roboter von Toyota[5], Tron X von Festo[6]

Zur Betreibung der Roboterhände können verschiedene Arten von Aktoren verwendet werden. Hier kann man elektromechanische, hydraulische, pneumatische Antriebe sowie Cable-Drive-Mechanismen nennen. Im Folgenden werden alle diese Mechanismen ausführlicher betrachtet.

Ein Beispiel für eine Hand **mit elektromechanischen Aktoren** ist die DLR-Hand, die im Gebiet der Robotertechnik sehr bekannt ist (siehe Abbildung 68).

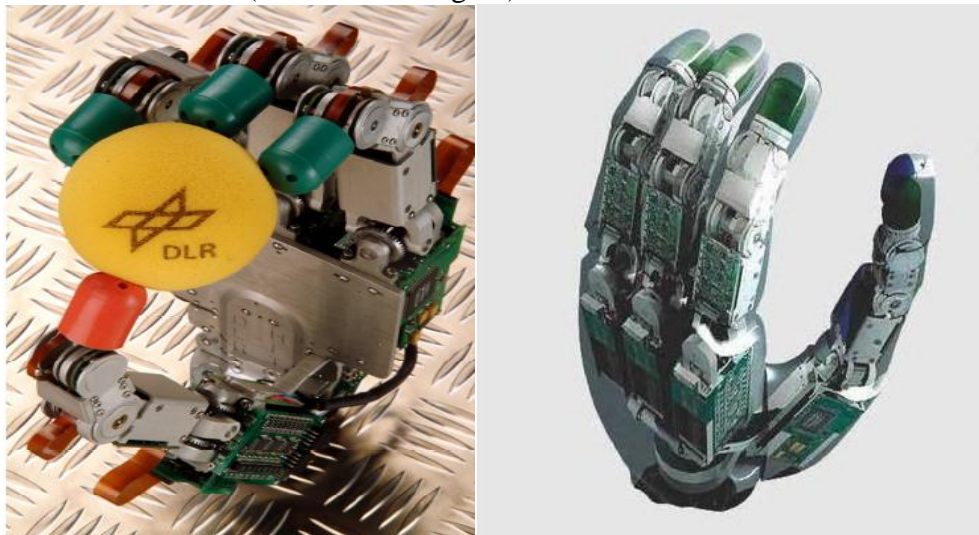


Abbildung 68. Die DLR-HIT-Hand[7]

Es handelt sich um ein gemeinsames Projekt des deutschen Zentrums für Luft- und Raumfahrt (DLR) und des Harbin Institute of Technology (HIT). Die 4-Finger-Hand besitzt an jedem Finger 4 Gelenke mit 3 aktiven Freiheitsgraden und einem zusätzlichen Freiheitsgrad an der Daumenbasis. Jedes Gelenk ist mit einem kontaktlosen magnetischen Winkelsensor und einem DMS basierten Drehmomentsensor ausgestattet (insgesamt 13 Motoren). Dabei hat die Hand 65 Analog-Sensoren und 4-Fingertip-Sensoren (optional). Das Gewicht der Hand beträgt circa 2,2 kg.

Neben all den Vorteilen haben Hände dieser Art auch Nachteile. Die Vielzahl von Motoren macht die Hand sehr schwer. Dazu kommt eine geringe Baugröße der Motoren und damit hohe Anschaffungskosten. Die zahlreichen Motoren und Sensoren verursachen eine hohe Anzahl von Drähten und bringen neue Probleme mit sich, wie das Problem der vorteilhaften Anordnung zur bequemen Konstruktion und Installation. Weiterhin verursachen die Motoren Lärm, was bei Prothesen unerwünscht ist. Deshalb ist die Vereinfachung der Konstruktion der Hand von großer Bedeutung. Als

Lösung kann die Versetzung von Motoren aus den Fingern z.B. in die Handfläche oder in den Unterarm dienen. Auf solche Weise kann man die genannten Nachteile bei Verwendung von hydraulischen oder pneumatischen Aktoren umgehen.

Roboterhände mit hydraulischen Aktoren haben im Vergleich zum vorgehenden Aufbaukonzept eine relativ einfache Konstruktion und Steuerung, was sie preisgünstiger macht. Nach diesem Prinzip wurde z. B. die „New Ultralight Anthropomorphic Hand“ im Jahre 2000 am Forschungszentrum Karlsruhe gefertigt[8].

Die Hand besitzt fünf Finger, 11 Freiheitsgrade und ist mit 18 Fluidaktoren ausgestattet, die hydraulisch betrieben werden können. Die Wahl der Fluidversorgung hängt von den Einsatzbedingungen der Hand ab. Die flexiblen Fluidaktoren ermöglichen neben den Bewegungen der Fingergelenke eine Reduktion des Gewichts und machen den Einsatz der hydraulisch betriebenen Hand als Prothese möglich. Die Hand ist in Abbildung 69 dargestellt.



Abbildung 69. Die Ultralight Anthropomorphic Hand[8]

Die mechanische Konstruktion der Roboterhand besteht aus einem Aluminiumskelett mit je einem flexiblen Fluidaktor pro Gelenk. Jeder Finger besitzt zwei Gelenke und zwei unabhängige Rotations-Freiheitsgrade. Der Daumen enthält ein zusätzliches drittes Gelenk zur Realisierung der Oppositionsstellung, so dass sich insgesamt elf Freiheitsgrade ergeben. Das Funktionsprinzip des flexiblen Fluidaktors beruht auf der Expansion eines flexiblen Behälters unter Druck. Wird der Behälter gefüllt, dehnt er sich aus, beim Entleeren zieht er sich wieder zusammen. In Verbindung mit einem Gelenk lassen sich so Drehbewegungen realisieren [9]. Trotzdem eröffnet die Verwendung des Hydrauliksystems mehrere Fragen über die Dichtheit, Sicherheit und Zuverlässigkeit des Systems.

Bei Händen unter **Verwendung von Drive Cable Mechanismen** liegen die Motoren nicht in den Gelenken und die Bewegung wird durch Seile und Fäden erzeugt, welche den Mechanismus mit den Aktoren verbinden.

Hier kann man als Beispiel die Hände RTR1 und RTR2 nennen (Abbildung 70) [10]. Die RTR1-Hand hat drei Finger, sechs unabhängige Freiheitsgrade und wiegt 250

Gramm, während die RTR2-Hand ein zusätzliches Gelenk im Daumen enthält, insgesamt neun Freiheitsgrade hat und 350 Gramm wiegt. Dabei sind die beiden Hände mit Sensoren ausgestattet, die die aktuellen Zustände bestimmen. Die RTR1-Hand ist mit sechs Motoren und die RTR2-Hand ist mit zwei Motoren ausgestattet. Die Gelenke werden mittels Zugkraft des Seils in Bewegung gebracht.

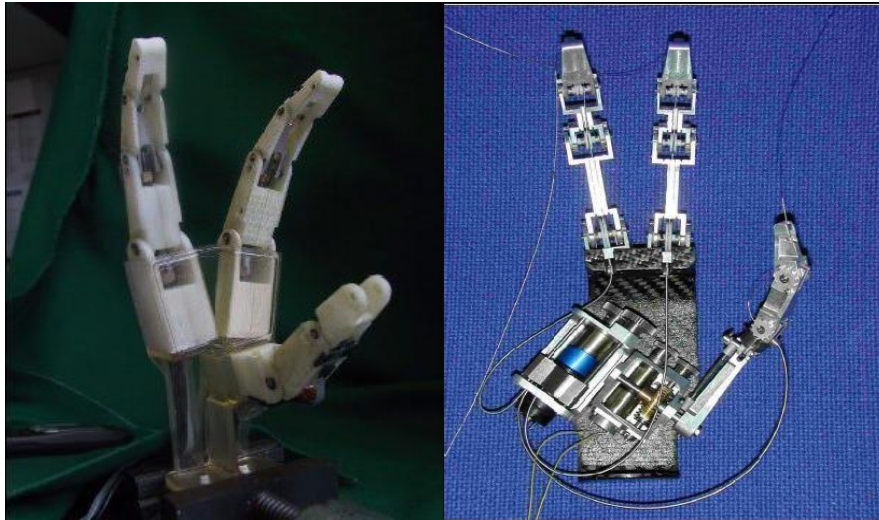


Abbildung 70. RTR1- Hand und RTR2-Hand

Die genaue Kraftverteilung hängt von der mechanischen Konstruktion ab. Die DC-Motoren sind in die Handoberfläche integriert. Noch ein Vertreter dieses Typs ist die Shadow-Dextrous-Hand (Abbildung 71). Sie hat fünf Finger und insgesamt 25 Freiheitsgrade: Fünf im Daumen, drei in jedem weiteren Finger, zwei in der Handfläche und zwei im Handgelenk.

Sie wird über 36 dafür entwickelte „Luftmuskeln“ gesteuert, die zusammen den Unterarm bilden und mit Hilfe von Bowdenzügen mit den Gelenken verbunden sind. Die Muskeln dehnen sich, wenn sie mit Druckluft gefüllt werden und sind vor allem leichter und billiger als Druckluftzylinder. Durch diesen pneumatischen Antrieb lassen sich sehr weiche und nachgiebige Greifbewegungen erzeugen. Der Unterarm bildet mit der Hand eine feste Einheit.

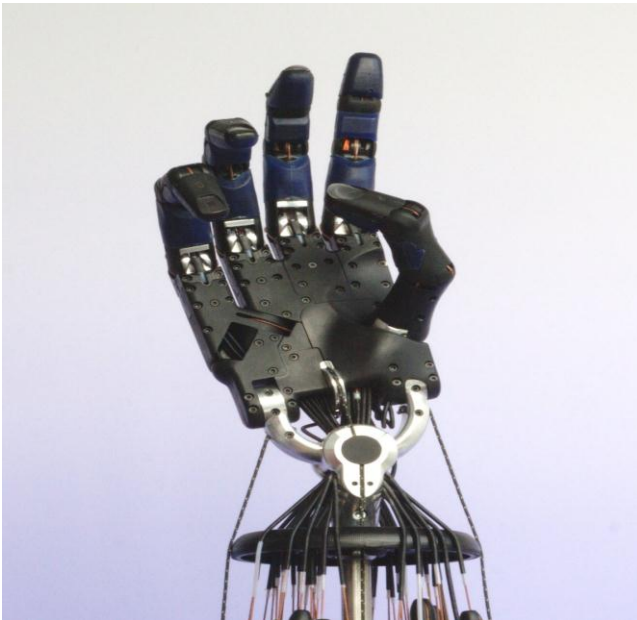


Abbildung 71. Die Shadow-Dextrous-Hand [11]

Die Multi-DOF-Powered-Prosthetic-Hand [13]. Die Multi-DOF-Powered-Prosthetic-Hand besteht aus fünf Fingern mit je drei Gelenken und dem Handgelenk. Die Streckung geschieht passiv mit Hilfe eines adaptiven Jointmechanismus. Jeder Finger hat zwei Freiheitsgrade mit einer aktiven Bewegungssteuerung. Nur der Daumen ist direkt mit dem Servomotor verbunden. Das Handgelenk ist mit zwei Motoren für die Pronation/Supination und Extension/Flexion ausgestattet. Insgesamt besitzt die Multi-DOF-Powered-Prosthetic-Hand zwölf Freiheitsgrade und ein Gesamtgewicht von 1,207 kg. Bei der UB-Hand[12] (Abbildung 72) werden die Finger wie bei der Multi-DOF-Powered-Prosthetic-

Der große Vorteil des Antriebs der Shadow-Hand ist die geringere Steifigkeit des Systems[12]. Die Hand ermöglicht eine flexible Reaktion auf Einflüsse von außen, da sie durch den pneumatischen Betrieb nicht in einer Position fest fixiert ist. Deshalb ist eine sicherere Anwendung im direkten Umgang mit Menschen möglich. Die Bewegungssteuerung stellt hier allerdings eine sehr große Herausforderung dar. Für eine komplexe Bewegung müssen viele der 36 pneumatische Antriebe zusammenspielen, was eine genaue Bewegungsanalyse erfordert. Zudem besteht das Problem, dass der Zusammenhang zwischen Kraft, Druck und Verdichtung nicht linear ist. Dazu ist die Konstruktion sehr kompliziert und mit ihren 3,5 kg ist die Shadow-Hand sehr schwer.

Eine weitere Hand, die hier zu nennen wäre, ist

Hand mit einem Drahtseilmechanismus angetrieben, bei dem jedes einzelne Gelenk mit einem eigenen Drahtseil ausgerüstet ist. Die Konstruktion basiert auf Fingergliedern (Phalangen), welche durch elastische Verbindungsgelenke miteinander verbunden sind. Jedes Gelenk kann getrennt oder gemeinsam mit anderen Gelenken gesteuert werden und wird durch bürstenlose DC-Motoren in Bewegung gesetzt. Jeder Finger hat vier Freiheitsgrade. Die Hand hat redundante Freiheitsgrade, wodurch gleichzeitig Schwierigkeiten bei der Steuerung entstehen.

Eine weitere Roboterhand ist Ben's mechanische Hand[15] (Abbildung 73). Ihre Konstruktion ist sehr einfach, da die einzelnen Gelenke über Bowdenzüge durch 14-Modellbau-Servomotoren bewegt werden. Ihr Gewicht beträgt circa 2 kg.

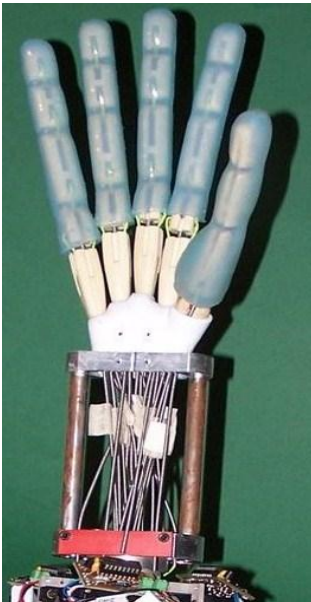


Abbildung 72. UB-Hand 3[12]

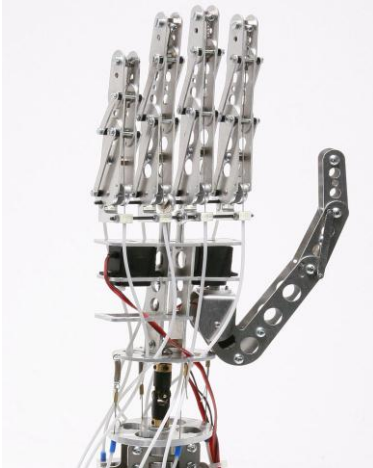


Abbildung 73. Ben's mechanische Hand [15]

Einen weiteren Vertreter der Drive-Cable-Hände entwickelte Carl Pisaturo[16].

Die von Carl Pisaturo projektierte Hand ist halb so groß wie eine menschliche Hand, hat vier Arbeitsfinger, einen unbeweglichen Daumen und wird mit zwei Drahtseilen angetrieben. Eine solche Steuerung erlaubt zwar keine Kontrolle über jeden Finger, ermöglicht aber viele unterschiedliche Gesten und Greifbewegungen. Die Phalangen sind miteinander durch Wellen verbunden und jedes Gelenk enthält kleine Derlin-Scheiben, die an den Wellen liegen. Durch diese Scheiben geht ein Drahtseil, das jede Scheibe in Bewegung bringt. In dieser Hand wurde eine Drahtlaufbahn verwendet, die sich in der Phalanx kreuzt.

Noch eine besondere Charakteristik für mechanische Hände ist die Redundanz, die man entweder als Erhöhung der Freiheitsgrade des Mechanismus oder als Erhöhung der Aktorenanzahl und somit der Freiheitsgradanzahl versteht. Hier ist letzteres von Bedeutung: Die Anzahl der Aktoren ist größer als die Anzahl der Freiheitsgrade.

Bei dem Entwurf soll man nach minimaler Anzahl der Aktoren trachten, weil es automatisch geringere Produktionskosten bedeutet. Andererseits können die Konstruktionsbesonderheiten eigene Anforderungen stellen, die keine geringere Anzahl von Aktoren zulassen. Wie diese Problematik von anderen Autoren gelöst wurde, sieht man in der Tabelle 7.

Alle betrachteten Hände sind in der Tabelle 8 zusammengefasst. Aus dieser Tabelle ist leicht zu entnehmen, dass die Mehrheit der Hände Cable-Drive-Hände sind. Das kann man dadurch erklären, dass ein solches Prinzip eine einfachere Konstruktion einschließt, weil z.B. die Motor- und Sensorszahl im Vergleich zur DLR-Hand geringer ist. Damit werden auch die Kosten bedeutend niedriger.

Tabelle 7. Die Charakteristiken einiger Hände [29]

Name	Finger	DOF	Aktor	Redundant
SARAH	3	10	mechanisch	Ja
AMADEUS	3	keine	mechanisch	Ja

		Daten		
Karlsruhe RC	5	keine Daten	mechanisch	Ja
Bologna University	3	10	mechanisch	Nein
LMS	4	16	mechanisch	Nein
NASA Robonaut	4	12	mechanisch	Nein
Okada	3	11	mechanisch	Nein
RTR Hand 2	3	9	mechanisch	Ja
Salisbury	3	9	mechanisch	Nein
Soft Gripper 1	2	10	mechanisch	Ja
Soft Gripper 3	3	10	mechanisch	Ja
SPRING Hand	3	8	mechanisch	Ja
RTR Hand 1	3	6	hydraulisch	Nein
Omni-Hand	3	n. a.	fluid	Nein
Utah/MIT	4	16	pneumatisch	Nein
Shadow C3 Hand	5	24	cable drive	Nein
ASI Gripper	3	3	cable drive	Nein
BarrettHand BH8	3	7	cable drive	Ja
Belgrade/USC	5	6	cable drive	Ja
DLR Hand 1	4	12	cable drive	Nein
DLR Hand 2	4	13	cable drive	Nein
Goldfinger	4	12	cable drive	Nein
HIT/DLR Hand	4	13	cable drive	Nein
Karlsruhe Hand 2	4	12	cable drive	Nein
MARS	3	12	cable drive	Ja

Tabelle 8. Übersicht der Hände nach Aufbauart und Entwickler

Hände	Aktor	Entwickler
Die DLR-HIT-Hand	13 Motoren	das deutsche Zentrum für Luft und Raumfahrt (DLR) und Harbin Institute of Technology (HIT)
Die Ultralight Anthropomorphic Hand	18 Fluidaktoren	Forschungszentrum Karlsruhe
RTR1 und RTR2 Hand	Cable Drive	ARTS Lab in Zusammenarbeit INAIL
die Shadow Dextrous Hand		Shadow Robot Company
Multi-DOF Powered Prosthetic Hand		Precision Engineering Laboratory, University of Tokyo mit Laboratory of Bioinstrumentation and Biomechatronics, University of Chiba
UB Hand 3		University of Bologna
Ben's mechanische Hand		privates Modell von Ben Schaefer
Hand von Carl Pisaturo		Carl Pisaturo mit MIT Media Lab

Ziel dieses Kapitels ist der Entwurf einer humanoiden Roboterhand, die ähnliche Bewegungsmöglichkeiten wie eine menschliche Hand hat.

Diese komplizierte Aufgabe wurde in folgende Teilaufgaben unterteilt:

Entwurf eines angenäherten CAD-Modells der Hand. Dieses erlaubt die Geometrieform und Größe der Hand zu bestimmen, die VRML-Modelle zu entwickeln und 3D-Animationen mit Simulation der Kinematik mit MATLAB durchzuführen.

Entwicklung eines preisgünstigen, leichten, kompakten, zuverlässigen Getriebes.

Untersuchung von Lösungsmöglichkeiten für die inverse Kinematik mit Hilfe von Optimierungen mit Nebenbedingungen in Echtzeit.

Entwicklung eines Simulationsprototyps der Hand zum Testen der Steuerungs- und Regelungs-Algorithmen für den Positions- und Kraftregelkreis.

4.1 Aufbau der Mechanik des Roboterhand

Für die Vereinfachung der Aufgabe werden die Hand und der Arm vom Schultergelenk bis Handgelenk, den man als redundanten Manipulator mit sieben Freiheitsgraden darstellen kann, getrennt betrachtet.

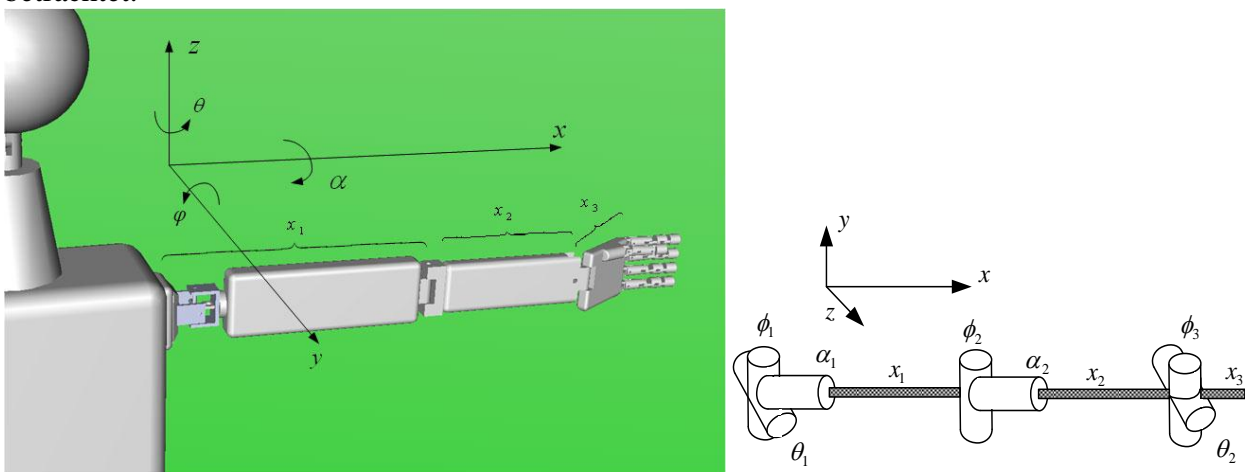
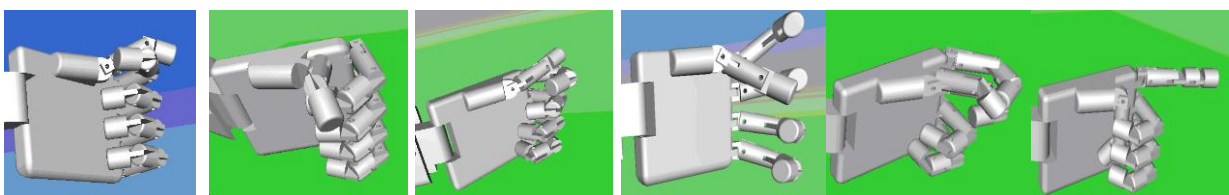


Abbildung 74. Die anthropomorphe künstliche Hand

Der Arm besteht aus der Schulter, dem Ellbogen und der Hand, die 5 Finger hat. Jeder Finger hat 4 Freiheitsgrade. Der Arm (Handgelenk ohne Finger) hat insgesamt 7 Freiheitsgrade und stellt deswegen einen redundanten Manipulator dar.

Einerseits ergibt das Flexibilität der Konstruktion und andererseits Freiheit im Bewegungsvorgang. Das CAD-Modell der Roboterhand ist in der Abbildung 74 dargestellt.

Die Schultern haben drei Freiheitsgrade. Die Achsen von diesen Freiheitsgraden kreuzen sich in einem Punkt der Schulter. Der Ellbogen und die Hand haben zwei Freiheitsgrade, wie es in Abbildung 74 gezeigt ist. In Abbildung 75 sieht man realisierte Griffe, die den am häufigsten verwendeten Handgriffen entsprechen.



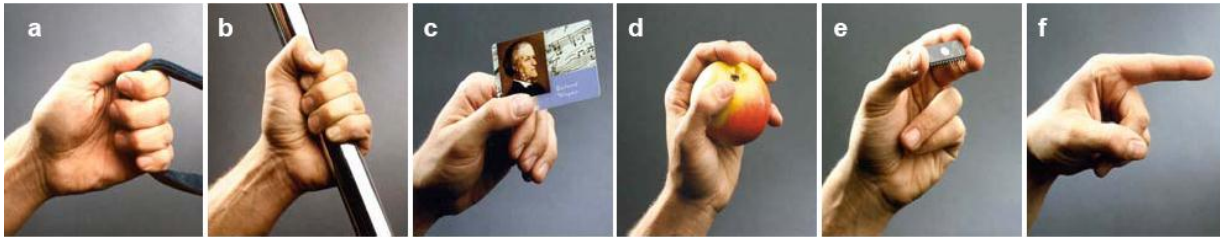


Abbildung 75. Hakengriff (a), Zylindergriff (b), Lateral Griff (c), Sphärischer Griff (d), Dreipunktgriff (e), Indexstellung (f)

Wie aus der Abbildung 75 hervor geht, entspricht jedem Objekt eine passende Greifbewegung. Um eine erfolgreiche Greifbewegung zu verwirklichen, braucht der Roboter eine visuelle Rückkopplung.

4.2 Twisted Wire Actuator

Es gibt grundsätzlich nur lineare und rotierende Aktoren. Die linearen Aktoren haben den wesentlichen Vorteil, dass sie sofort eine lineare Bewegung erzeugen. Diese Bewegung und entsprechende Kraft verwendet man direkt in Cable-Drive-Mechanismen. Im Fall der Hydraulik oder der Pneumatik braucht man automatisch eine Pumpe oder einen Druckballon. Elektrische lineare Motoren oder künstliche Polymeraktoren[17] haben entweder einen hohen Preis oder große Außenmaße.

Rotierende Aktoren sind auf Grund ihres Preises und ihrer geringeren Außenmaße anderen Antrieben im Vorteil, jedoch benötigen sie Getriebe, welche die Rotationsbewegungen in eine Linearbewegung umwandeln. Solche Getriebe können Kugelgewindetriebe oder eine Seil-Trommel sein. Getriebe haben jedoch Nachteile wie hohe Kosten, Getriebespiel, große Außenmaße und einen geringen Wirkungsgrad. Deshalb wird ein Wirkprinzip vorgeschlagen, das auf der Verkürzung eines Seils durch Verdrillen beruht. Das Prinzip ist in Abbildung 76 gezeigt.

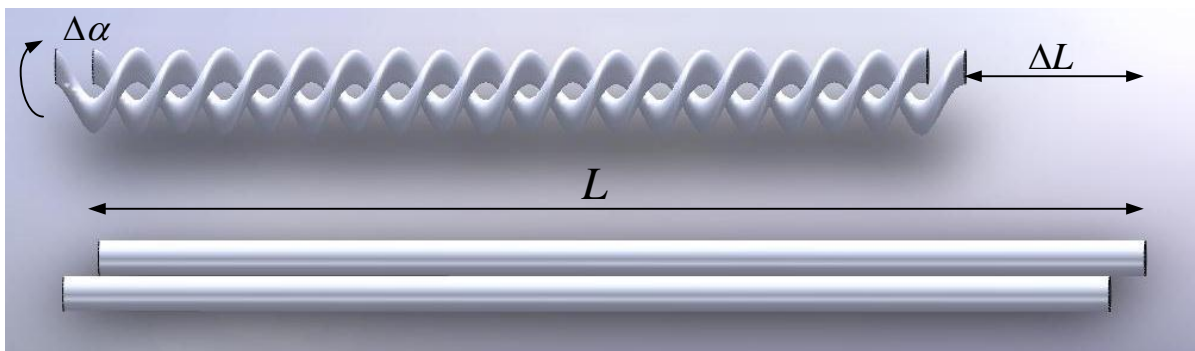


Abbildung 76. Grundprinzip der Verkürzung eines Fadens durch Verdrillen (TWA)

In Englisch wurde dies Twisted Wire Actuator (TWA) genannt [18], [19]. Ein TWA enthält keine der oben genannten Nachteile und hat eine relativ einfache und kompakte Konstruktion. Nach diesem Prinzip des Verdrillens elastischer Materialien ohne plastische Verformung wird eine Rotation in eine Längsbewegung umgewandelt. Diese Längsbewegung wird zur Hand mit Hilfe von Metallseilen übertragen, die mit dem Fingermechanismus verbunden sind. Durch das Ziehen des Seils erfolgt die Beugung oder die Streckung des Fingers.

Ein TWA weist neben den Vorteilen auch einige Nachteile auf [20].

Vorteile von TWAs:

- Der Übersetzungsfaktor ist sehr groß.
- Es ist ein sehr einfacher, kostengünstiger Mechanismus, der aus beliebigen biegsamen Werkstoffen hergestellt werden kann.
- Ein TWA erlaubt die Fertigung einfacher Mechanismen für die Handdrehung.

Nachteile von TWAs:

- Die geringe Zuverlässigkeit. Es gibt Probleme mit der Befestigung des sich verkürzenden Seils. Als Folge können die Grenzbelastungen zu einer Störung führen.
- Die Nichtlinearität des Übertragungsfaktors.
- Mangel an einer vollständigen mathematischen Beschreibung des Systemverhaltens wegen des Entstehens einer Elastizität des Seils nach dem Verdrillen.

Im nächsten Abschnitt wird die Funktionsweise von TWAs im Zusammenhang mit dem Motor detaillierter betrachtet.

Wie oben schon erwähnt, funktioniert ein TWA nach dem Prinzip der Verkürzung eines Seils durch Verdrillen. Zur Vereinfachung wird angenommen, dass das Seil von dem TWA ein plastisch undehnbare Federstab ist. Aus der Theorie der Federstäbe existiert der folgende Zusammenhang zwischen dem Torsionswinkel α und einem durch diesen Winkel erzeugten Drehmoment M_k (Abbildung 77 a))[21]:

$$\alpha = \frac{2L}{G\pi r^4} M_k, \quad (1)$$

wobei G der Drehungsmodul (hängt von dem Material des Stabs ab), $L = const$ die Länge der Staberzeugende (Zylindererzeugende) im Anfangszustand (bei $M_k = 0$) und r der Radius der Fläche im Stabquerschnitt ist.

Falls noch eine Längskraft außer dem Außendrehmoment M_k entlang der Stabachse wirkt, wird der Winkel α durch die Gleichung (1) bestimmt [21]:

$$\alpha = \frac{2L}{G\pi r^4} (M_k - M_t) \quad (2)$$

wobei M_t das die Stabtorsion verhindernde und durch die Kraft F bedingte Moment ist. Nach der Stabtorsion um den Winkel α unter der Wirkung des Momentes $M_k - M_t$ wandelt sich die unveränderliche Stablänge L in eine Spirale um (Abbildung 77 a)) mit der Länge $S(\alpha) < L$, die zur Längsachse dieses Stabes parallel ist und die Oberfläche des verdrilltes Stabes erzeugt.

Die Größe M_t kann aus einem Dreieck ABC (Abbildung 77 b)), welches die Zerlegung eines Objektes mit zylindrischer Staboberfläche in einer Ebene darstellt, berechnet werden.

In Abbildung 77 b) ist F_t eine Kraft, die die Stabtorsion durch das Moment M_k verhindert und β ein Winkel zwischen der Erzeugenden des verdrillten Stabs $AC = S$ und der Anfangserzeugende dieses Stabes $BC = L$.

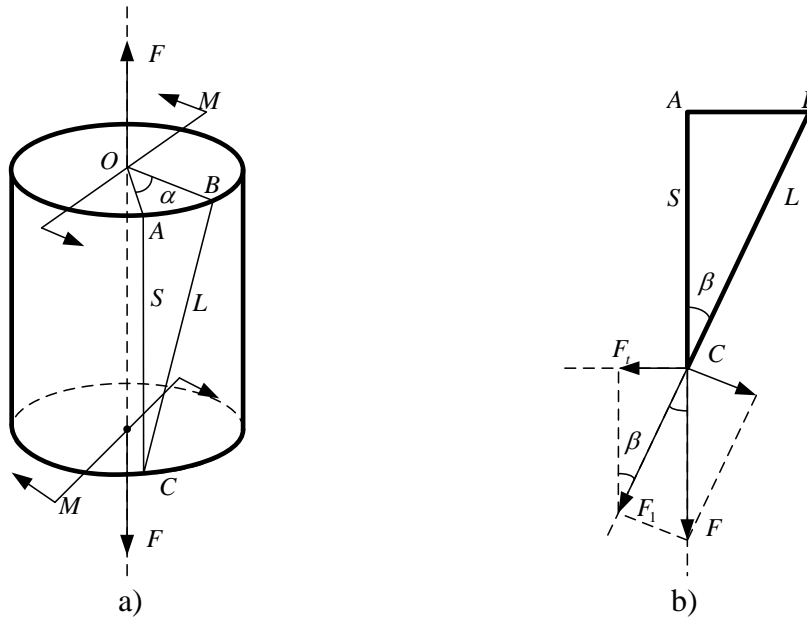


Abbildung 77. a) Die Torsionsbesonderheiten der elastischen Stange mit einem kreisförmigen Querschnitt b) Die Zerlegung eines Teils mit zylindrischer Stangenoberfläche in einer Ebene

Aus der Abbildung 77 b) ist zu sehen, dass:

$$F_t = F_1 \sin \beta = F \cos \beta \sin \beta = \frac{1}{2} F \sin 2\beta, \quad (3)$$

wobei das Moment M_t aus der Gleichung(1) und (2) bestimmt wird:

$$M_t = \frac{1}{2} r F \sin 2\beta. \quad (4)$$

Die Länge des Bogens AB (Abbildung 77 a)) ist gleich $r\alpha$. Dann ist aus dem Dreieck ABC (Abbildung 77 b)) leicht die Längeverkürzung des um den Winkel α verdrehten elastischen Stabes ΔL zu erkennen (bei einem Anfangstorsionswinkel $\alpha_0 = 0$):

$$\Delta L(M_k, F) = AB - BC = L - S = L - \sqrt{L^2 - r^2 \alpha^2 (M_k, F)} \quad (5)$$

Es ist offensichtlich, dass Gleichung (5) nur bei $r\alpha < L$ gültig ist.

Mit der Gleichung (5) kann der umgekehrte Zusammenhang $\alpha(t)$ bei bekannter Längeverkürzung ΔL gefunden werden:

$$\alpha(t) = \frac{\sqrt{\Delta L(t)(2L - \Delta L(t))}}{r}. \quad (6)$$

Unter Berücksichtigung, dass $\beta = \arcsin \frac{r\alpha}{L} = \arccos \frac{S}{L}$ ist, resultiert aus Gleichung (3):

$$M_t = \frac{r^2 S}{L^2} F \alpha. \quad (7)$$

Der verdrehte Stab kann aus einem beliebigen plastisch undehnbar Material z.B. aus der Spleißung von Kevlar- oder Basaltfäden hergestellt werden. Dabei muss man berücksichtigen, dass die Gleichungen (1)-(5) die Bewegung der Fäden durch Torsion zwar nur annähernd beschreiben, der Charakter dieser Zusammenhänge jedoch erhalten bleibt.

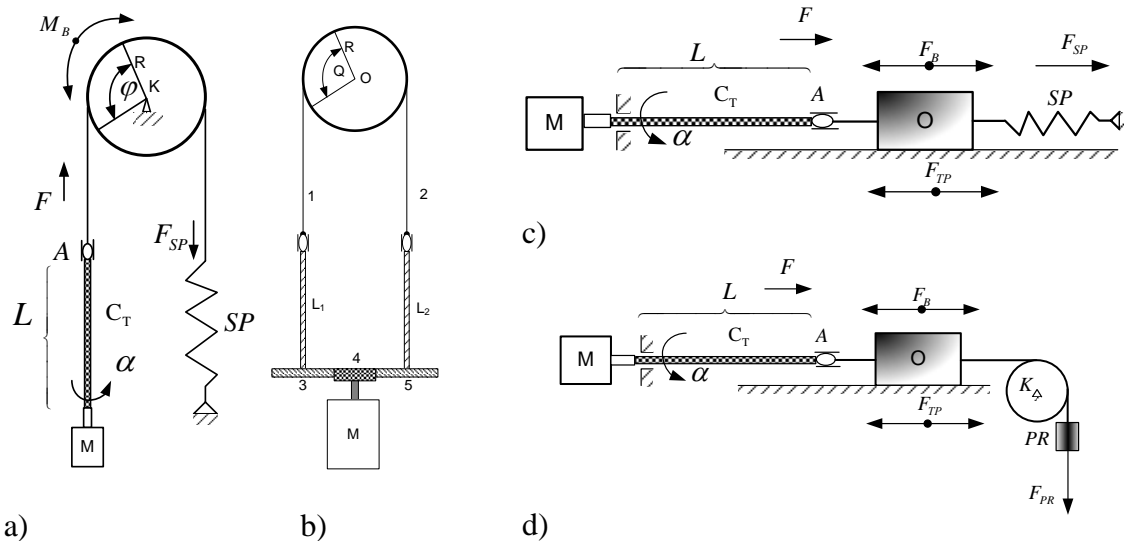


Abbildung 78. TWA Mechanismen

- a) Drehungsmechanismus mit der Feder;
 b) Ein Rotationsmechanismus mit zwei gedrehten Seilen.
 c) Ein linearer Mechanismus mit Rückführfeder
 d) Ein linearer Mechanismus mit Gegengewicht

In Abbildung 78a) ist das kinematische Schema des Mechanismus für die Drehung einer Scheibe mit dem Radius R dargestellt.

Dabei ist φ der Drehwinkel der Scheibe und $M_B = F_{BR}$ das angreifende äußere Drehmoment.

Aus Gleichung (5) folgt:

$$\varphi = \frac{\Delta L}{R} = \frac{L - \sqrt{L^2 - \alpha^2 \cdot r^2}}{R} \quad (8)$$

Setzen wir (6) und (5) in (1) ein mit Berücksichtigung, dass $S = L - \Delta L$, dann bekommt man:

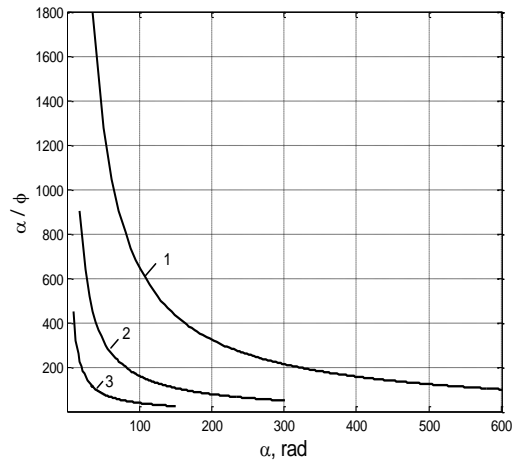
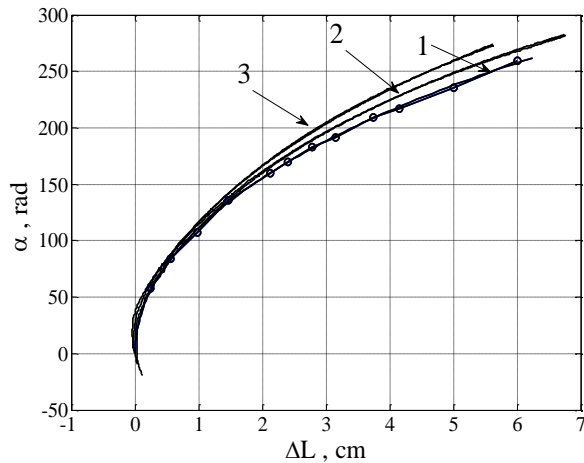
$$M_k = \frac{r \sqrt{R\varphi(2L - R\varphi)}}{L} \left(\frac{G\pi r^2}{2} + \frac{(L - R\varphi)}{L} F \right) \quad (9)$$

Der Übersetzungsfaktor dieses Mechanismus ist:

$$i = \frac{\alpha}{\varphi} \quad (10)$$

Er hängt nichtlinear von α ab.

Wenn man annimmt, dass sich das verdrehte Seil auch strecken kann, erlaubt dies auch einen Mechanismus ohne Rückführfeder zu bauen, wie es in der Abbildung 78 b) dargestellt ist. Um diesen Streckungseffekt zu untersuchen wurde ein linearer Mechanismus mit einem Gegengewicht aufgebaut (Abbildung 78 d)). Für die Untersuchungen wurde ein Dynema-Seil (Litze) ausgewählt, welches folgende Parameter besitzt: $L = 0.3$ m, $R = 0.26$ mm, $G = 2.9 \cdot 10^9$ N/m².



a)

b)

Abbildung 79. a) Experimentelle Abhängigkeit des $\alpha(\Delta L)$ bei unterschiedlichen Gegengewichte,

b) Das Übersetzungszahl des TWA, nach Gleichung (10), in Abhängigkeit von Seilradius

Die Kurve 1 in der Abbildung 79 a) wurde mit einer Gegengewichtsmasse von 50 g aufgenommen. Hierbei kann man die Dehnung des Seils vernachlässigen. Die theoretische Kurve ist fast identisch mit der Kurve 1. Die Kurve 2 wurde mit einer Gegengewichtsmasse von 1 kg aufgenommen. In der Abbildung 79 a) sieht man eine Abweichung von den theoretischen Berechnungen. Die Kurve 3 entspricht dem Gegengewicht von 3 kg und die Abweichung ist noch größer. Die Hin- und Rückbewegung des Mechanismus folgen derselben Kurve und es wurde keine Hysterese beobachtet. Die Elastizität des Seils erlaubt Mechanismen wie in Abbildung 78 b) dargestellt zu bauen. Auf Abbildung 79 b) sieht man die Abhängigkeit des Übertragungsmechanismus von dem Seilradius. Die Kurven 1,2 und 3 werden für unterschiedlichen Radien berechnet: $r_1 = 1.75 \cdot 10^{-4}$ m, $r_2 = 3.5 \cdot 10^{-4}$ m $r_3 = 7 \cdot 10^{-4}$ m. Hieraus kann gefolgert werden, dass bei wachsendem Radius der Übersetzungsfaktor sinkt.

4.3 Handstruktur und Fingermechanismen

Für eine Roboterhand existieren viele Varianten, die sich nach mechanischer Komplexität und Flexibilität unterscheiden. In Bewegung kann entweder der ganze Fingermechanismus oder jede Fingerphalanx unabhängig voneinander gebracht werden. Darüber hinaus gibt es viele andere Varianten [22].

In jedem Fall kann ein TWA nur ziehen, deswegen ist für die Rückbewegung ein weiteres aktives System mit zusätzlichem Motor oder einer Feder nötig. Die einfachste Konstruktion ist die Streckung der Fingermechanik mittels eines Seils und Beugung mit Torsionsfedern.

Den Aufbau sieht man in Abbildung 80.

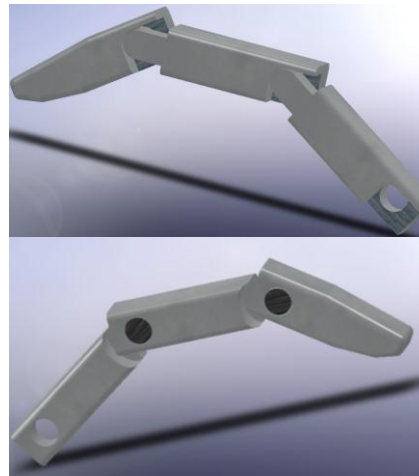
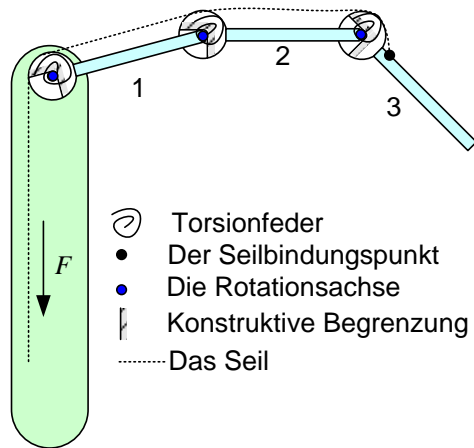


Abbildung 80. Der Fingermechanismus mit Seilstrecker und Federbeuger (links); Das CAD Fingermodell mit $\pi/2$ Winkel in den Freiheitsgraden (rechts)

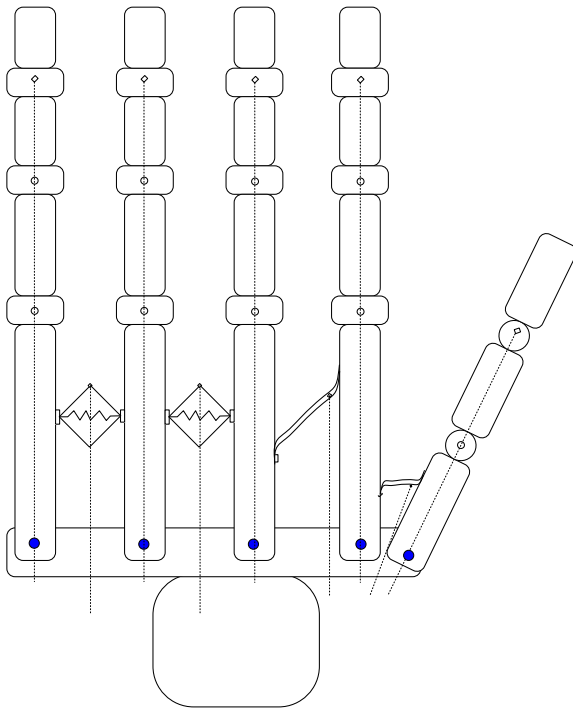


Abbildung 81. Handkonstruktion mit Seilbeuger und speziellen Fingerspreizmechanismus (links); CAD-Modell von vereinfachter Handkonstruktion(rechts)

Eine Beugung des Fingermechanismus mittels eines Seils ist ebenfalls möglich. Dies führt jedoch zur Erhöhung der mechanischen Komplexität der Konstruktion, weil dafür eine komplexe Seilführung notwendig ist, wie es Abbildung 82 zeigt.

Das Beugungsmodell des Fingermechanismus ist equivalent zum Streckermodell aufgebaut mit dem Unterschied, dass das Beugungsseil mechanisch von der Leitöffnung getrennt ist. Das ist leicht realisierbar, indem das Seil in einen Leitkanal eingeführt oder als „Bowdenzug“ angefertigt wird [23].

Als preisgünstigste und einfachste Variante in Bezug auf die Modellbildung und Konstruktion gilt das mathematische Modell für einen Fingermechanismus mit Seilsteckern und Torsionsfedern für die Beugung. Das komplette Handmodell zeigt Abbildung 81. Hier ist die vereinfachte Konstruktion dargestellt, wobei spezielle Fingerspreizmechanismen realisiert wurden. Für diese Mechanismen kann

man z.B. Federklammern oder ein Parallelogramm aus Stangen benutzen. Dafür braucht man jedoch eine Rücklauffeder.

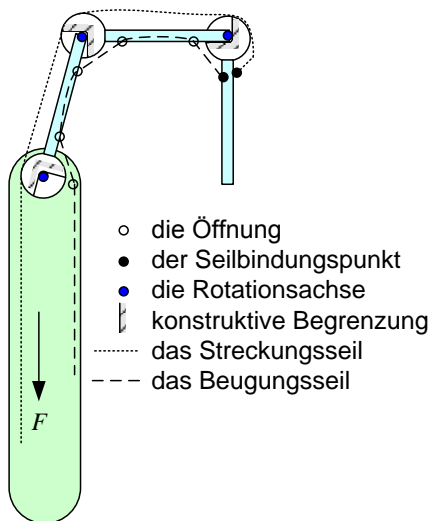


Abbildung 82. Die Beugung und Streckung des Fingermechanismus mit den Seilen.

In der Abbildung 78 ist zu sehen, dass nach der Drehung des Motors um den Winkel α eine Längenverkürzung des Seils um ΔL erfolgt, was eine Kraft F verursacht. Mit dieser Kraft F wird das Seil gezogen, das das Scharnier in eine Rotation um den Winkel φ_Σ versetzt.

In dieser Arbeit wird der Fingermechanismus in *SimMechanics* von *MATLAB* modelliert. Der Finger besteht aus drei Gelenken mit den Längen l_1, l_2, l_3 . Jedes Gelenk hat seine eigene Masse und ein eigenes Trägheitsmoment (Tabelle 13, Anhang A). Bei der Trägheitsmomentberechnung wird angenommen, dass jedes Gelenk eine sich um die Achse z drehende Stange ist. Deshalb ist das Trägheitsmoment hinsichtlich der z -Achse gleich Null. Die Rotation der Gelenke ist auf 0 bis $\pi/2$ begrenzt. Diese Begrenzung wird durch einen mechanischen Anschlag realisiert. Beim Erreichen des Anschlagpunktes bildet sich eine Kraft F (Abbildung 80), die dem Drehmoment entgegenwirkt, welches am Scharnier anliegt.

Im Modell ist der Kontakt als Punktkontakt (Gleichung(11))

dargestellt.

Ohne Wirkung der Kraft bleibt der Finger gebogen, was eine normale Lage für ihn ist. Im Modell ist dies durch die Anfangsbedingungen für jedes Gelenk berücksichtigt(Tabelle 13, Anhang A). Bei Eintreten der Kraftwirkung kommt es zur Streckung des Fingers. Nach der Streckung durch die Kraft wird die Beugung durch die Torsionsfeder ausgeübt. Jedes Gelenk ist mit einer Torsionsfeder ausgestattet, welche ein Moment gegen die wirkende Kraft F bildet. Außer den Momenten von dem Punktkontakt und der Torsionsfeder wirkt gegen die Kraft F das Moment, das durch die Reibungskraft bei der Rotation gebildet wird.

Das vom TWA ausgehende Seil ist oben am Scharnier des drittes Gelenks (Bindungspunkt siehe Abbildung 80)befestigt und liegt dicht an den Scharnieren des zweiten und ersten Gelenks.

Beim Ziehen des Seils mit der Kraft F entsteht am Gelenk ein Drehmoment, das dem Drehmoment der Torsionsfeder, dem Drehmoment durch die Reibungskraft und dem im Kontakt entstehenden Drehmomente entgegenwirkt. Das resultierende Drehmoment wird nach der Gleichung (11) berechnet.

Das dritte Gelenk rotiert bis zu der Lage, in der der Anschlag erreicht ist. Nach dem Erreichen des Anschlagpunktes bleibt das Gelenk unbeweglich und die Kraft F bildet über das Scharnier des zweiten Gelenks ein Drehmoment. Diesem Drehmoment wirkt neben dem Moment der eigenen Torsionsfeder noch das in der Torsionsfeder des dritten Gelenks entstandene Moment entgegen.

Anschließend rotieren die Gelenke 2 und 3 bis sie ebenfalls in den Anschlagpunkt für das zweite Gelenk gelangen. Danach geht die Kraft F auf das Scharnier des ersten Gelenks über und bildet das Drehmoment, welches gegen die Torsionsfeder aller drei Gelenke wirkt. Danach rotieren alle drei Gelenke bis sie in den Anschlagpunkt des ersten Gelenks gelangen. Dies ist die gestreckte Lage für den Fingermechanismus.

Auf Grund des oben Genannten ergibt sich die Gleichung für das Moment:

$$M = F \cdot R - M_s - M_f - M_H \quad (11)$$

wobei F die durch die Längeverkürzung des Seils von dem TWA entstehende Kraft, R der Radius des Scharniers, M_s das durch die Torsionsfeder gebildete Drehmoment, M_f das durch die Reibungskraft gebildete Drehmoment, M_H das im Anschlagpunkt entstehende Drehmoment ist.

Das resultierende Drehmoment M setzt das Gelenk in Bewegung. Der Drehwinkel φ und die Drehgeschwindigkeit ω des Gelenkes sind die Eingangsgrößen zur Bestimmung der drei durch die Reibungskraft, Punktkontakt und Sprungfeder gebildeten Momente.

Den Punktkontakt kann man als Block „*Rotational Hard Stop*“ darstellen, der ein Block von MATLAB/Simscape ist. Er begrenzt die Bewegung zwischen der oberen und unteren Grenze. Der *Rotational Hard Stop* wird nach der Gleichung (12) beschrieben:

$$M_H = \begin{cases} k_p \cdot \delta + D_p \cdot (\omega_R - \omega_C) & \text{für } \delta \geq g_p \\ 0 & \text{für } g_n < \delta < g_g \\ k_n \cdot \delta + D_n \cdot (\omega_R - \omega_C) & \text{für } \delta \leq g_n \end{cases} \quad (12)$$

wobei $\delta = \varphi_R - \varphi_C$, $\omega_R = \frac{d\varphi_R}{dt}$, $\omega_C = \frac{d\varphi_C}{dt}$, M_H das zusammenwirkende Moment zwischen dem

Gelenk und der Kontaktoberfläche, δ die relative Versetzung zwischen dem Gelenk und der Kontaktoberfläche, g_p der Spalt zwischen dem Gelenk und der Kontaktoberfläche in positiver

Richtung, g_n der Spalt zwischen dem Gelenk und der Kontaktoberfläche in negativer Richtung ist,

ω_R, ω_C den absoluten Drehgeschwindigkeiten der Kontaktkörpern R und C entsprechen, φ_R, φ_C den absoluten Drehwinkelversetzungen der Kontaktkörpern R und C entsprechen, k_p die

Kontaktsteifigkeit der oberen Grenze, k_n die Kontaktsteifigkeit der unteren Grenze, D_p die

Kontaktdämpfung der oberen Grenze, D_n die Kontaktdämpfung der unteren Grenze und t die Zeit ist.

Die gewählten Parameter für jedes Gelenk wurden in Tabelle 13 Anhang A eingetragen.

Das Torsionsfedermoment wird nach der Gleichung (13) gebildet:

$$M_S = -k \cdot (\varphi - \varphi_0) - b \cdot \omega, \quad (13)$$

wobei M_S das Drehmoment, das durch die Torsionsfeder gebildet wird, k die Torsionskonstante in $N \cdot rad$, φ der Drehwinkel des Gelenks, φ_0 der Anfangstorsionswinkel, b die Dämpfungs konstante und ω die Drehgeschwindigkeit sind.

Im Modell wurde die Reibung mit Hilfe von dem Block *Rotational Friction* berücksichtigt. Dieser Block modelliert die Reibung im Kontakt zwischen den drehenden Körpern. Das Moment, das die Reibungskraft bildet, ist eine Funktion der Geschwindigkeit und ist die Summe der drei Komponenten: statische (Stribeck), kinetische (Coulomb) und viskose Komponente (Gleichung (14)):

wenn $|\omega| \geq \omega_{th}$, dann ist $M_f = (T_C + (T_{brk} - T_C) \cdot \exp(-c_v \cdot |\omega|)) \cdot \text{sign}(\omega) + f_\omega$

$$\text{wenn } |\omega| < \omega_{th}, \text{ dann ist } M_f = \omega \frac{f_{\omega th} + T_C + (T_{brk} - T_C) \cdot \exp(-c_v \cdot \omega_{th})}{\omega_{th}} \quad (14)$$

wobei M_f das Drehmoment, das die Reibungskraft bildet, T_C die kinetische (Coulomb) Reibung, T_{brk} die statische (Stribeck) Reibung, c_v der Aproximationkoeffizient in $[s/rad]$, ω die Drehgeschwindigkeit, f_ω der Koeffizient der viskosen Reibung in $[m \cdot s \cdot N/rad]$ und ω_{th} der Grenzwert der Drehgeschwindigkeit ist.

Das Ersatzschema für den Fingermechanismus ist in Abbildung 83 dargestellt.

Im betrachteten Mechanismus kann das Seil durchhängen, dann erzeugt der TWA kein Moment, nur wenn das Seil gespannt ist, erzeugt der TWA ein Moment $M = F \cdot R$. Jedes Gelenk bewegt sich nur im Rahmen eines Zwischenraums, der von der Konstruktion abhängig ist. Wenn das Gelenk maximal ausgelenkt ist, wird aus der Kontaktkraft ein Moment M_{Hi} erzeugt und die Bewegung wird zum nächsten Gelenk übertragen.

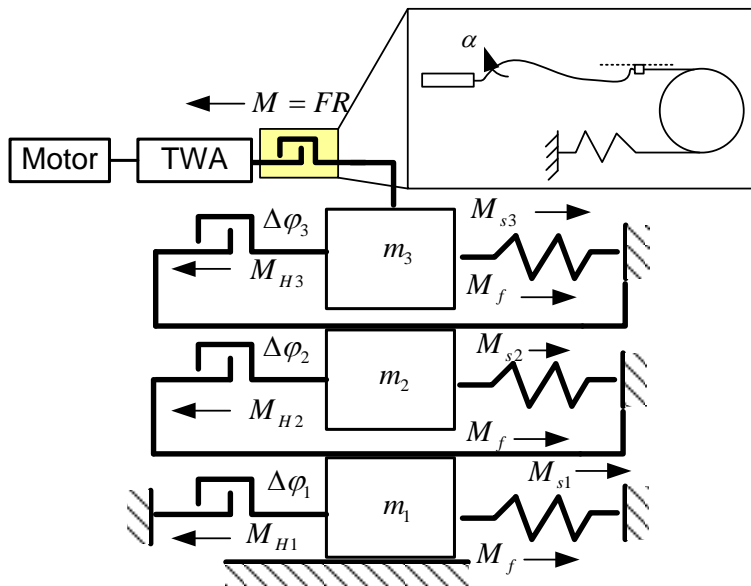


Abbildung 83. Das Ersatzschema für einen Fingermechanismus mit Seilbeuger

4.4 TWA, elektrischer Antrieb und Fingermechanismus

Die Einheit Motor-TWA-Fingermechanismus stellt ein Zweimassensystem dar. Es gibt verschiedene Möglichkeiten dies zu modellieren. Der Unterschied liegt bei den vereinfachenden Annahmen, welche bei der Systemmodellierung getroffen werden. Als elektrischer Motor wurde ein Gleichstrommotor eingesetzt. Die Motordaten sind im Anhang A dargestellt.

4.4.1. Einmassensystem des Motor-TWA-Fingermechanismus bei einem starr angenommenen Seil

In der ersten Variante der Modellierung werden alle Elemente der kinematischen Kette zu einem Trägheitsmoment zusammengefasst. Das ist möglich, sofern das Seil vom TWA stets in einem gespannten Zustand ist. Dabei kann man diese auf das Motorträgheitsmoment:

$$J_{\Sigma} = J_M + \frac{J_2}{i^2}, \quad (15)$$

oder das Trägheitsmoment des Arbeitsmechanismus:

$$J_{\Sigma} = J_M \cdot i^2 + J_2, \quad (16)$$

beziehen. In Gleichungen (15)-(16) ist J_{Σ} das gesamte Trägheitsmoment des Einmassensystems in $\text{kg} \cdot \text{m}^2$, J_M das Motorträgheitsmoment in $\text{kg} \cdot \text{m}^2$, J_2 das Trägheitsmoment des Arbeitsmechanismus in $\text{kg} \cdot \text{m}^2$ und i die Übersetzungszahl zwischen dem Motor und dem Arbeitsmechanismus. Dafür wird für den TWA eine starre Verbindung angenommen.

Beim Reduzieren des Motorträgheitsmoments auf das Trägheitsmoment des Fingers sieht das Blockschaltbild des Motor - TWA - Fingermechanismus wie in Abbildung 84 gezeigt aus.

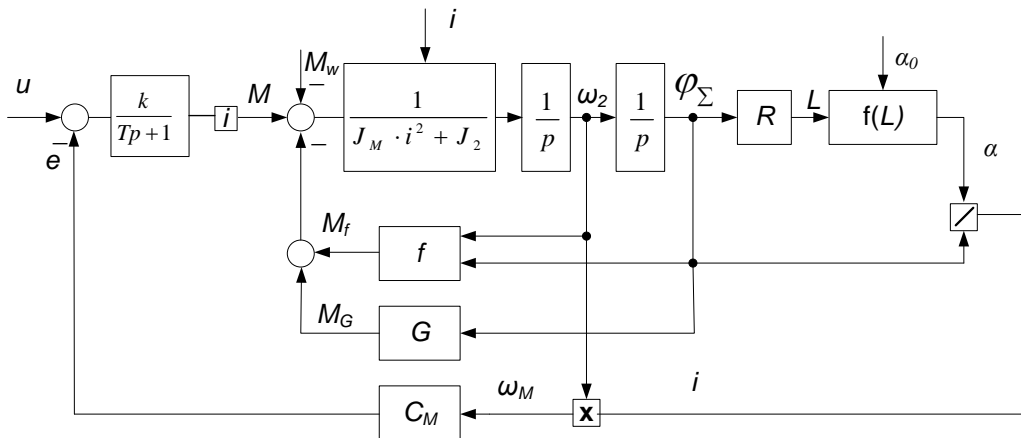


Abbildung 84. Das Blockschaltbild des Motor - TWA - Fingermechanismus bei starr angenommenem Seil

Hier ist ω_2 die Geschwindigkeit des Fingers in rad/s und φ_Σ der Fingerdrehwinkel in rad. Durch die Multiplikation des Fingerdrehwinkels mit dem Radius des Scharniers der Fingerphalanx bekommt man die Verkürzung des Seils L in m vom TWA, aus welcher der Drehwinkel des Motors als Funktion von L berechnet wird. Die Gleichung $\alpha = f(L)$ wird nach der Gleichung (6) bestimmt.

Aus dem Verhältnis des Motorwinkels α zu dem Fingerdrehwinkel φ ergibt sich die Übersetzungszahl i zwischen dem Motor und dem Finger. Außer dem Widerstandsmoment wirkt das durch die Reibungskraft im Finger M_f und die Gravitation M_G gebildete Moment auf die Welle des Motors.

Bei der Modellierung eines solchen Systems können einige Schwierigkeiten auftreten, weil es schwierig ist repräsentative Parameter der drei Gelenke für die Reduktion des Motorträgheitsmoments zu bestimmen.

4.4.2 Das Zweimassensystem Motor - TWA - Fingermechanismus

Die zweite Variante der Modellierung des Systems Motor-TWA-Finger (Abbildung 85) ist unter der Annahme möglich, dass das gemeinsame System als Zweimassensystem dargestellt wird und dass die Übergabe des Motormomentes stets bei einem gespannten Seil erfolgt.

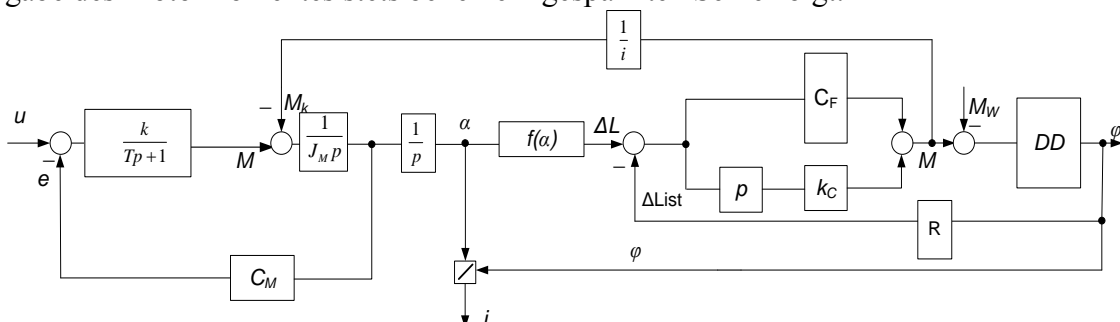


Abbildung 85. Signalflussplan des Motor-TWA-Fingermechanismus beim Zweimassensystem

Die Besonderheit in diesem System ist, dass man für den Koeffizienten $C_F \approx 10^5 \cdot 10^6$ N/m annimmt, um einen starren Kontakt zu simulieren. Hier treten die Kräfte aus dem mechanischen Kontakt und der Seilspannkraft in Wechselwirkung.

4.4.3 Der Motor-TWA-Fingermechanismus bei einem elastisch angenommenen Seil

Nach zahlreichen Experimenten wurde festgestellt, dass sich bei Verdrillung der Steifkoeffizient C_F des Seils verringert. Deswegen kann man annehmen, dass das Seil des TWA elastisch ist. Bei einer solchen Annahme lässt sich das System mit dem Modell eines elastisch verkoppelten Zwei-Massensystems beschreiben.

Unter Vernachlässigung der Verdrillkräfte des Seils und bei Annahme, dass das Seil vom TWA elastisch ist, ergibt sich ein Signalflussplan des Systems Motor-TWA-Finger wie in Abbildung 85 dargestellt. Im linken Teil des Signalflussplans ist der Motorregelkreis mit dem Drehwinkel α als Ausgangsgröße dargestellt. Der Motordrehwinkel ist die Eingangsgröße für die Berechnung der Seilverkürzung nach dem Verdrehen ΔL . Diese Verkürzung wird als Funktion von α nach der Gleichung (5) berechnet. Weiter wird die Elastizität des Seils durch den Koeffizienten der Steifigkeit $C_F = f(\alpha)$ und der Dämpfung $k_C = f(\alpha)$ berücksichtigt. Diese Koeffizienten müssen normalerweise durch Versuche mit dem Seil bestimmt werden. In der Arbeit wurden diese Koeffizienten während der Modellierung ausgewählt ($C_F = const; k_C = const$), wobei $C_F < 10^3$ N/m ist. „DD“ ist eine Bezeichnung für die direkte Dynamik des Fingermechanismus, aus der sich der gesamte Fingerwinkel φ_Σ ergibt. ΔL_{ist} (den benötigten Wert der Seilverkürzung) bekommt man durch Multiplikation des gesamten Fingerwinkels φ_Σ mit dem Radius des Gelenkscharniers R , der für alle Gelenke gleich angenommen wurde.

Aus der Differenz $\Delta L - \Delta L_{ist}$ und bei Berücksichtigung der Elastizität des Seils bekommt man das Moment M :

$$M = (\Delta L - \Delta L_{ist}) \cdot C_F + \frac{d}{dt}(\Delta L - \Delta L_{ist}) \cdot K_C, \quad (17)$$

für die direkte Dynamik des Fingers. Das Lastmoment für den Motor ist gleich $M \cdot i$.

Die direkte Dynamik des Fingermechanismus stellt ein zusammenhängendes Modell dreier einzelner Gelenke dar.

4.5 Simulation des Fingermechanismus

Die Modellbildung des Fingermechanismus als Einmassensystem ist problematisch wegen der variablen Übersetzungszahl, welche vom Verdrillungswinkel des Seils abhängig ist. Das heißt, dass in dem Modell die Trägheitsmomente ständig verändert werden müssen.

Deswegen wurde das Modell des Fingermechanismus als Zweimassensystem dargestellt. Beim starr angenommenen Seil ($C_F \approx 10^5 \dots 10^6$ N/m) kann man die Ausdehnung vom Seil vernachlässigen. Die Parameter vom Gleichstrommotor, welcher für die Simulation verwendet wurde, sind im Anhang A dargestellt.

In Abbildung 86 b) sieht man, dass sich der Fingermechanismus bei voller Spannung nach 0,4 s öffnet. Bei kleinen Verdrillungswinkeln ($\alpha < 100$) ist die Übersetzungszahl sehr hoch, deswegen hat der Motor fast keine Last beim Anlauf. Danach wächst der Strom schnell und kontinuierlich an und verändert sich sprunghaft, wenn sich das nächste Gelenk in Bewegung setzt. Dies geschieht wegen des Kontaktverlaufs zwischen den Gelenken des Fingermechanismus. Zuerst bewegt sich das dritte Gelenk bis zur maximalen Auslenkung, dann übergibt es die Zugkraft des Seils zum nächsten Gelenk, welches sich bis zum eigenen Lagegrenzwert auslenkt. Wenn das letzte Gelenk maximal ausgelenkt ist, wirkt die Zugkraft als Widerstand gegen die Begrenzungsanschlüsse aller Gelenke. Für den Gleichstrommotor bedeutet dies einen Kurzschluss, weil die Geschwindigkeit des Motors bei der vollen Spannung gleich Null ist (Abbildung 86 a)).

Die Rückbewegung des Fingermechanismus (Beugung des Fingers) erfolgt mit Hilfe der Torsionsfedern, weil der TWA nur eine Zugkraft erzeugt. Deswegen ist die Rückbewegung in die Ausgangslage von den Torsionsfederparametern abhängig (Abbildung 86 c) d)). Die Beugung des Fingermechanismus ist wegen der hohen Übersetzungszahl des TWAs nur durch Rücklauf des Motors möglich.

Die Abweichungen zwischen den Lagekurven (in Abbildung 86 c) ca. 1s) treten nur beim Rücklauf des Fingermechanismus auf, wenn der TWA zu schnell entlastet wird.

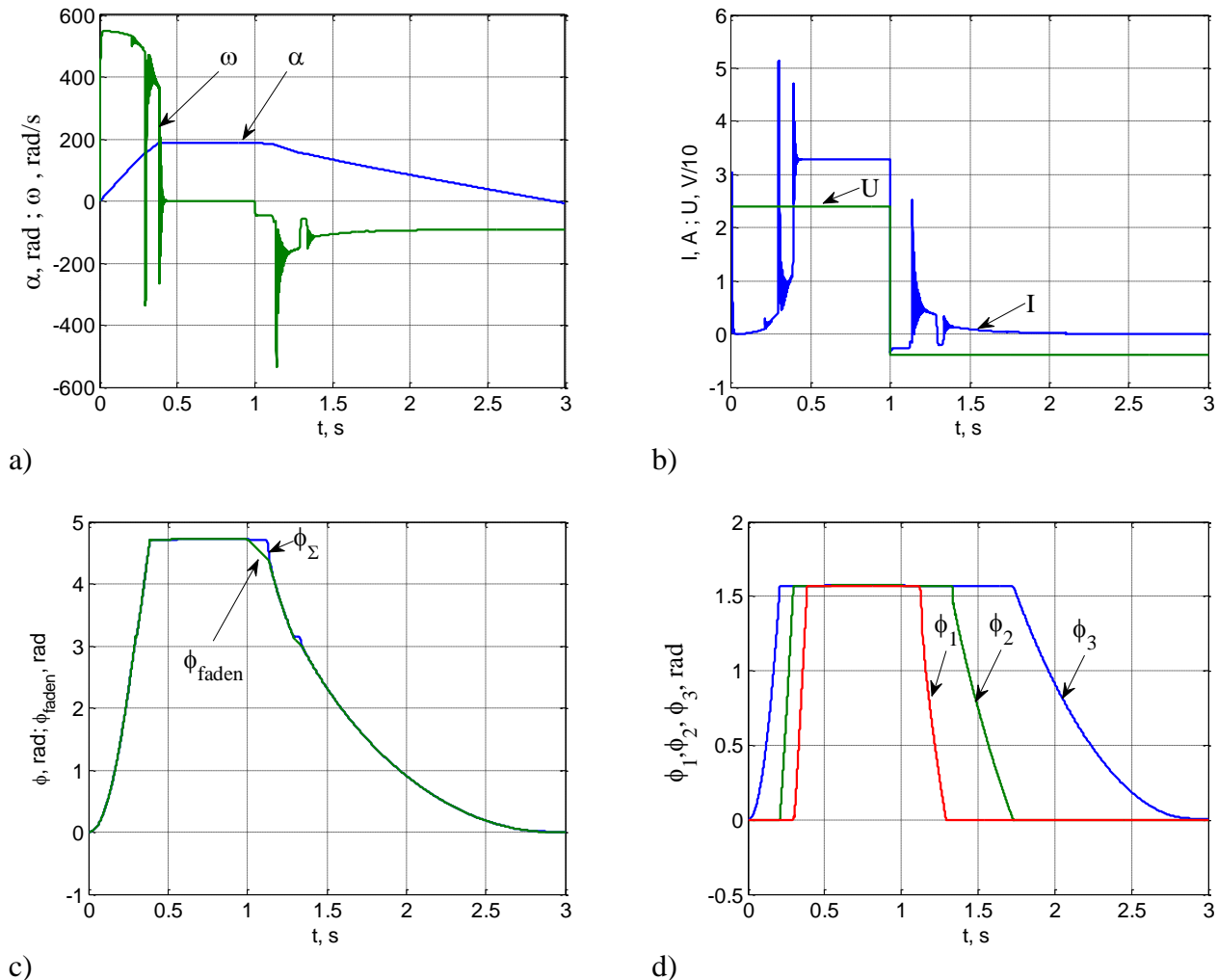


Abbildung 86. Die Simulationsergebnisse für den Leerlauf des Fingermechanismus

- a) Die Geschwindigkeit des Motors und die Position des Rotors
- b) Der Verlauf des Stroms und der Spannung im Gleichstrommotor
- c) Die Position des TWAs und der gemeinsame Winkel des Fingermechanismus
- d) Die Position der Gelenke vom Fingermechanismus

4.5.1 PI-Regelung für den TWA - Fingermechanismus

Als Regler für den Fingermechanismus wurde PI-Regler ausgewählt. Die Reglerparameter $K_p=10$, $K_i=4,4$ wurden für eine akzeptable Regelungsqualität durch Simulation bestimmt. Das Reglersignal wird wie folgt berechnet:

$$U_R = (\varphi_{soll} - \varphi_\Sigma) \left(K_p + K_i \frac{1}{s} \right). \quad (18)$$

Für den ausgewählten Motor ist der Eingangsspannungsbereich $U_R \leq \pm 24$ V.

Beim Anlauf des Motors wurde eine maximale Spannung erreicht (siehe Abbildung 87 a) und b)). Wenn sich ein Gelenk in Bewegung setzt, entsteht ein Widerstandssprung, was auf Abbildung 87 b) und d) deutlich zu sehen ist.

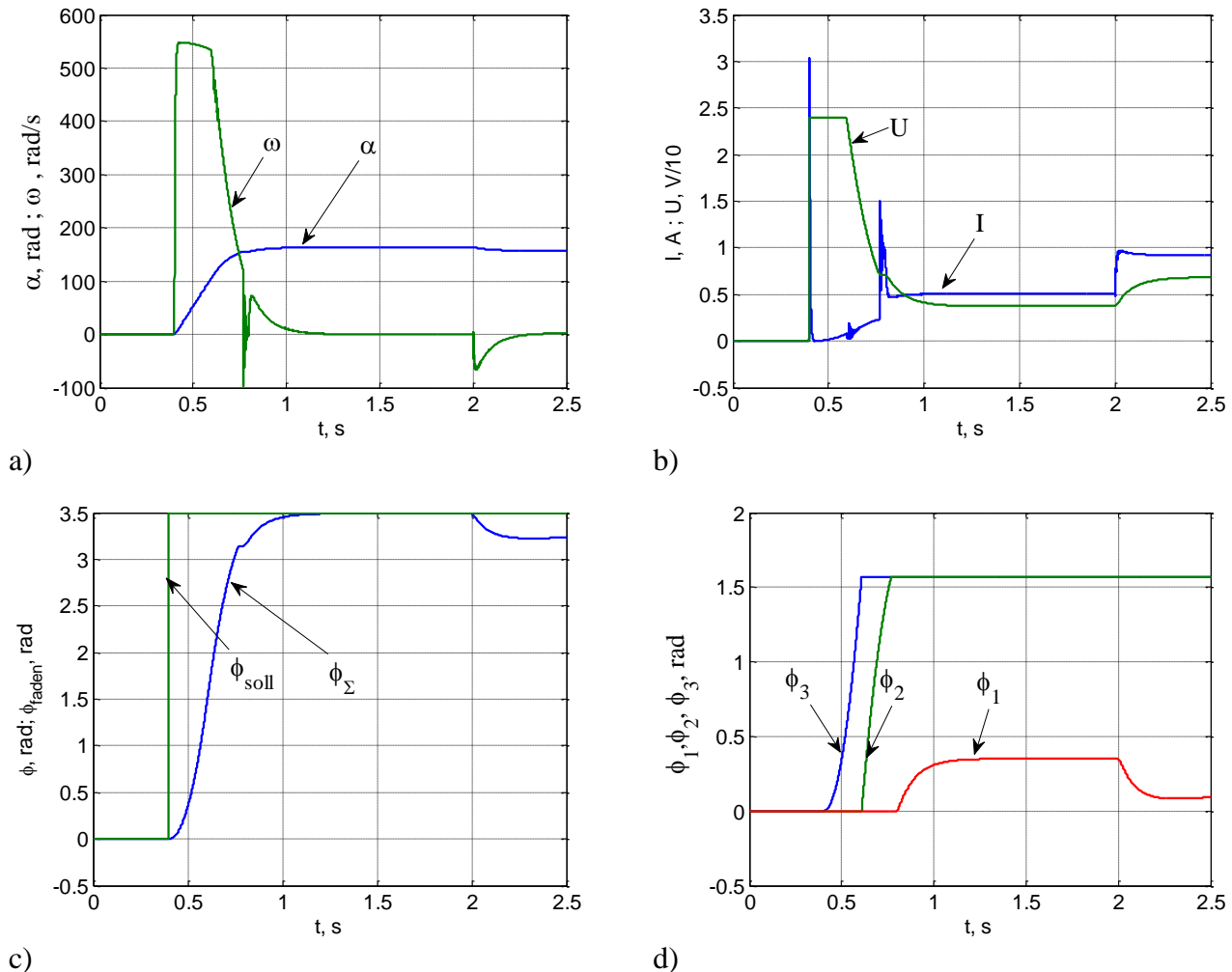


Abbildung 87. Die Simulationsergebnisse für die Sprungantwort des Fingermechanismus bei der PI-Regelung

- a) Die Geschwindigkeit des Motors und die Position des Rotors
- b) Der Verlauf des Stroms und der Spannung im Gleichstrommotor
- c) Die Position des TWAs und der gemeinsame Winkel des Fingermechanismus
- d) Die Position der Gelenke vom Fingermechanismus

In der Zeit von 2 s an wurde ein äußeres Störmoment von $M_c = 1$ Nm angelegt. Dieses hat eine Stromerhöhung von 0,5 A auf 0,9 A verursacht. Die Positionsabweichung auf Grund des Störmoments ist gleich 0,25 rad.

Die PI-Regelung ohne Berücksichtigung der Nichtlinearitäten des TWAs kann als Kompromisslösung gelten. Die Regelungsparameter wie die Anregelzeit, die Überschwingweite und die Positionsabweichung beim Störmoment sind von der Position des Fingermechanismus abhängig.

4.5.2 PI-Regelung mit Kompensation der Nichtlinearität für den TWA - Fingermechanismus

Die nichtlineare Regelung des Fingermechanismus gleicht die Nichtlinearität des TWAs aus und verbessert die Regelgüte. Die Kompensation der Nichtlinearität des TWA erlaubt unabhängig von der Position des Fingermechanismus die gleiche Regelgüte.

Als Regler für den Fingermechanismus wurde ein PI-Regler ausgewählt. Die Reglerparameter $K_p=2$, $K_i=0,15$ würden bei einer akzeptablen Regelungsqualität durch Simulation bestimmt.

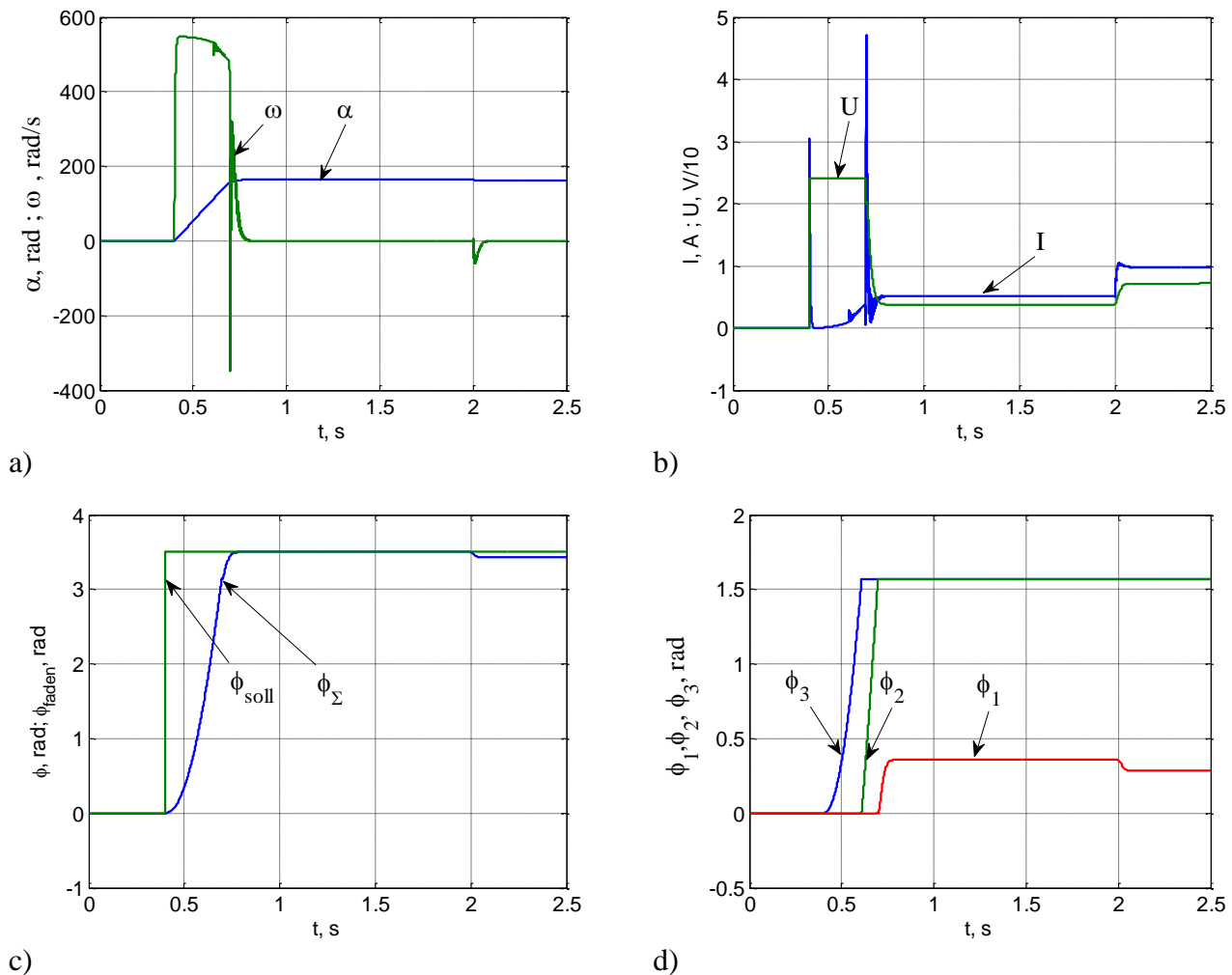


Abbildung 88. Die Simulationsergebnisse für die Sprungantwort des Fingermechanismus bei der nichtlinearen PI-Regelung

- Die Geschwindigkeit des Motors und die Position des Rotors
- Der Verlauf des Stroms und der Spannung in dem Gleichstrommotor
- Die Position des TWAs und der gemeinsame Winkel des Fingermechanismus
- Die Position der Gelenke von Fingermechanismus

Zuerst muss die gemessene Lage des Fingermechanismus mittels Gleichung (6) auf die Motorseite reduziert werden. Auf dieselbe Weise berechnet man mittels Gleichung (6) den Verdrillungswinkel α_{soll} . Dadurch ist es möglich das Regelungsgesetz wie folgt zu beschreiben:

$$U_R = \left(\frac{\sqrt{\varphi_{soll} R(2L - \varphi_{soll} R)}}{r} - \frac{\sqrt{\varphi_\Sigma R(2L - \varphi_\Sigma R)}}{r} \right) \left(K_p + K_i \frac{1}{s} \right). \quad (19)$$

Der Anlauf startet bei $t = 0,4$ s und der Fingermechanismus kommt zur gewünschten Position in 0,5 s, siehe Abbildung 88 a) bis d). Die Spannung kommt beim Anlauf in den Sättigungsbereich, so erreicht der Motor seine maximale Beschleunigung und der Haltestrom beträgt 0,5 A, wie bei der einfachen PI-Regelung.

Nach 2 s wurde ein äußeres Störmoment von $M_c = 1$ Nm angelegt. Dieses verursacht, wie bereit bei der linearen Regelung, eine Stromerhöhung von 0,5 A auf 1 A. Die Positionsabweichung auf Grund des Störmoments verändert sich auf 0,1 rad. Man kann zusammenfassend sagen, dass die Kompensation der Nichtlinearitäten von TWAs die Regelungsqualität deutlich verbessern kann.

Ein TWA-Antrieb zeigt eine gute Leistungseffizienz und ist bei der Konstruktion von „Cable Drive“ Mechanismen sowie bei der Handkonstruktion sehr empfehlenswert.

4.6 Die Aufgabe der direkten Kinematik

Für Serviceoperationen des Roboterarmes muss man die Position des Greifers in den kartesischen Koordinaten bestimmen. Dieser Schritt heißt „Aufgabe der direkten Kinematik“ (ADK). Direkte kinematische Transformationen kann man durch homogene Transformationen lösen. Dabei kann man 3 unterschiedliche Varianten auswählen:

- Koordinatentransformation mittels homogener-Matrizen.
- Koordinatentransformation, wobei die Rotation mittels Rotationsmatrizen [3x3] berechnet wird und die Koordinatenübertragung mittels Vektoren erfolgt.
- Koordinatentransformation, wobei die Rotation mittels Quaternionen berechnet wird und die Koordinatenübertragung mittels Vektoren erfolgt.

Nach der Analyse aller dieser Verfahren wurden homogene Matrizen als mathematisches Fundament zur Lösung ausgewählt. Sie beschreiben den mathematischen Hintergrund genau und haben denselben oder geringeren Berechnungsaufwand im Vergleich zu anderen Varianten.

Die Matrix, die die Translationsbewegung berücksichtigt, heißt homogene Translationsmatrix (4x4) und

sieht wie folgt aus: $T = \begin{bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ f_{1 \times 3} & 1 \times 1 \end{bmatrix}$, wobei $R_{3 \times 3}$ die Rotations-, $p_{3 \times 1}$ die Verschiebungs-, $f_{1 \times 3}$ die

Perspektiveumwandlungsmatrix und 1×1 die Maßstabsänderung ist. In der Kinematik nimmt man an, dass $f_{1 \times 3} = [0,0,0]$ und der Maßstab gleich eins ist.

Die direkte Kinematik in homogenen Matrizen wird wie folgt beschrieben:

$$T_{end} = T_R(\alpha_1) \cdot T_R(\varphi_1) \cdot T_R(\theta_1) \cdot T_T(x_1) \cdot T_R(\alpha_2) \cdot T_R(\varphi_2) \cdot T_T(x_2) \cdot T_R(\varphi_3) \cdot T_R(\theta_3) \cdot T_T(x_3) \quad (20)$$

In der Gleichung (20) ist T_R die Rotationsmatrix über die entsprechende Achse und T_T die Translationsmatrix entlang der entsprechenden Achse.

Wenn die Auslenkungen der Gelenke bekannt sind, findet man mittels Gleichung (20) auch die kartesischen Koordinaten des Greifers.

4.7 Die Aufgabe der inversen Kinematik

Im Gegensatz zur direkten Kinematik, beantwortet die inverse Kinematik die Frage, welche Gelenkwinkel benötigt werden, um einen bestimmten Punkt in kartesischen Koordinaten zu erreichen.

Die Antwort auf diese Frage findet man mit Hilfe von iterativen Optimierungsmethoden, d. h. die Lösungen der inversen Kinematik kann man durch die iterative Berechnung der direkten Kinematik ersetzen.

Dabei wird als Optimierungskriterium der Mittelwert des quadratischen Fehlers verwendet. Die Aufgabe besteht darin, einen Wert des Vektors des Gelenkwinkel θ zu finden, welcher das lokale

Minimum der Funktion des Fehlers $E = \sum_{i=1}^6 ((p_x, p_y, p_z, \alpha, \beta, \gamma) - f_i(\theta_n))^2$ oder der Differenz der

Rotationsmatrizen angeben:

$$\begin{aligned}
 E = & (R_{G11} - R_{A11})^2 + (R_{G12} - R_{A12})^2 + (R_{G13} - R_{A13})^2 + \\
 & + (R_{G21} - R_{A21})^2 + (R_{G22} - R_{A22})^2 + (R_{G23} - R_{A23})^2 + \\
 & (R_{G31} - R_{A31})^2 + (R_{G32} - R_{A32})^2 + (R_{G33} - R_{A33})^2 + \\
 & + (p_{Gx} - p_{Ax})^2 + (p_{Gy} - p_{Ay})^2 + (p_{Gz} - p_{Az})^2
 \end{aligned} \tag{21}$$

ergibt, wobei R_A die aktuelle Matrix, R_G die gewünschte Matrix, p_A und p_G die Vektoren der aktuellen und gewünschten Position darstellen.

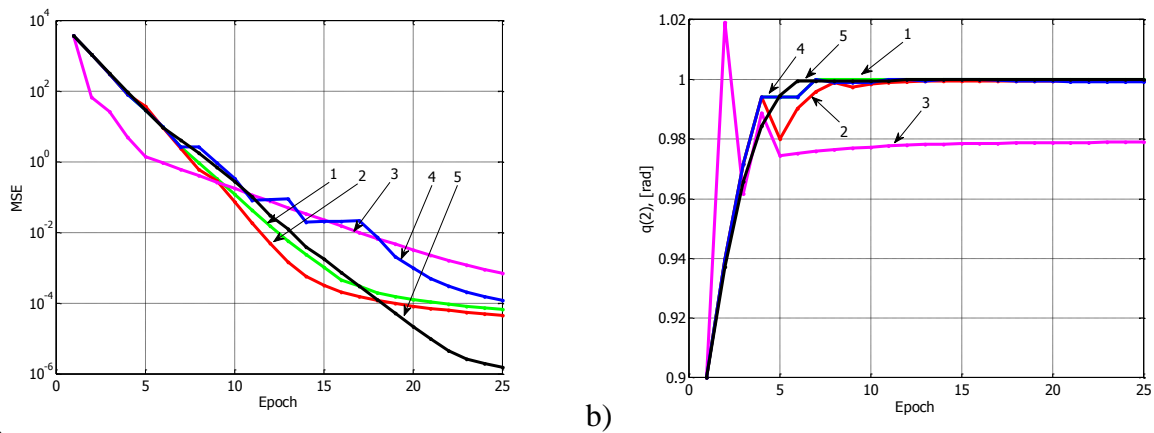
Als Optimierungsverfahren wurde der Levenberg-Marquardt Algorithmus implementiert (Kapitel 1). Einfache iterative Lösungen betrachten keine Nebenbedingungen wie z.B. die Drehwinkelbegrenzungen für jeden Freiheitsgrad. Eine Drehwinkelbegrenzung ist wichtig, weil Konstruktionsbesonderheiten nicht alle möglichen Gelenkparameter zulassen. Übliche Winkelbegrenzungen sind in der Tabelle 9 aufgeführt.

Tabelle 9. Winkelbegrenzungen für die Freiheitsgrade des humanoiden Roboterarmes [25]

[grad]	α_1	φ_1	θ_2	α_2	φ_2	φ_3	θ_3
Max	+80	+80	160	+80	150	+30	+60
Min	-70	-90	-40	-80	-10	-30	-60

Für die Lösung der inversen Kinematik einschließlich der Optimierung mit Nebenbedingungen wurden die Methoden Gradientenauslöschung (1), Gradientenspiegelung (2), Lagrange Multiplikatoren(3), Methode der Schrankenfunktion $R \cdot 1/g(q)$ (4) und die Methode der Schrankenfunktion $-R \cdot \log(g(q))$ (5) realisiert. Hier ist der Vektor der Freiheitsgradwinkel gleich $q = [\alpha_1, \varphi_1, \theta_1, \alpha_2, \varphi_2, \varphi_3, \theta_3]$. Die Ergebnisse der Methoden ist in Abbildung 89 zu sehen. Der Nummerierung entspricht Tabelle 10.

Die genaue Beschreibung von diesem Verfahren ist in Kapitel 1, Gleichungen (40-47) zu finden. In Abbildung 89 a) sieht man den Zeitverlauf des Optimierungsprozesses mittels Levenberg-Marquardt Algorithmus. Ein akzeptables Niveau des Fehlers wurde nach zehn Zyklen erreicht. In Abbildung 89 b) sieht man, wie die Grenzbedingung für eine Koordinate φ_1 bearbeitet wurde. Aus dem Zeitverlauf des Optimierungsprozesses ist zu schlussfolgern, dass das Lagrange-Verfahren Lösungen einschließt, die den zulässigen Bereich verlassen können.



a) Abbildung 89. a) Fehlerminimierung b) Bearbeitung der Koordinatebegrenzung auf 1 rad für die Koordinate φ_1

Alle hier dargestellten Methoden haben dasselbe Fehlerniveau erreicht. Die oben genannten Algorithmen wurden für die Lösung der inversen Kinematik für einen Roboterarm verwendet. Die Gradientenauslöschung, Gradientenspiegelung und das Barrierfunktionsverfahren haben fast dieselben Ergebnisse gebracht. Als bestes Verfahren ist das Barrierfunktionsverfahren mit logarithmischen Funktionen und als langsamstes Verfahren ist das Lagrange-Multiplikatoren-Verfahren wegen seines hohen Berechnungsaufwands hervorgegangen. Die Ergebnisse wurden in Tabelle 10 dargestellt.

Tabelle 10. Der Vergleich des Levenberg-Marquardt-Optimierungsverfahrens bei Begrenzungen der Adaptation auf 50 Zyklen

Beschränkungsverfahren	Mittelquadratische Fehler	Rechenzeit in s	Relative Zeit, s/s
1 Auslöschung	1.83e-005	0.016681	1
2 Spiegelung	1.4004e-005	0.016416	0.9841
3 Lagrange	6.2976e-005	0.045967	2.7557
4 Barrier $R \cdot 1/g(q)$	1.644e-004	0.020461	1.2266
5Barrier $-R \cdot \log(g(q))$	2.1e-0.07	0.026842	1.6091

Wie aus Tabelle 10 deutlich erkennbar ist, ist eine Lösung der inversen Kinematik mit einer geringen Zyklenanzahl notwendig, wenn es sich um eine Echtzeitanwendung handelt.

Wenn man ein einfaches Gradientenabstiegsverfahren verwenden möchte (Kapitel 1, Gleichung (16)), wählt man für Echtzeitsysteme zur Lösung der inversen Kinematik das Jacobi-Verfahren [27]. Es ist nicht so präzise, dafür jedoch sehr schnell. Alle nötigen Beschränkungen und Nebenbedingungen können in diesem Verfahren berücksichtigt werden.

4.8 „Computed torque control“ für den Roboterarm

Wie schon in Kapitel 1 beschrieben wurde, ist das „computed torque control“ Verfahren für ein dynamisches System wie der Roboterarm implementierbar, wenn es sich mit den Lagrange-Euler Gleichungen (Kapitel 1 Gleichungen(54-55)) beschreiben lässt.

Dieses Verfahren liefert die Linearisierung der Regelstrecke und erlaubt für nichtlineare Systeme einen linearen Regler zu verwenden. Der Einfluss von Motoren und Getrieben auf den Regelungsprozess wird im kommenden Abschnitt berücksichtigt und minimiert.

4.8.1 Die Kompensation der nichtlinearen Dynamik des elektrischen Antriebs des Roboters durch Feedbacklinearisierung

Für die Steuerung des Systems unter Berücksichtigung der Dynamik des Gleichstrommotors wird der elektrische Teil des Motors als offene Integrationskette dargestellt. Die Gleichung des elektrischen Teils des Motors ist:

$$M(t) = \frac{C_m}{T_e p + 1} (U_r(t) - C_w \omega(t)) \quad (22)$$

Wobei $M(t)$ das Moment des Motors, $C_m = 1/R$ - die Momentkonstante, R - Ankerwiderstand, $U_r(t)$ die Spannung des Motors, $C_w = C\Phi$ die Geschwindigkeitskonstante, C - Maschinenkonstante, Φ - Motorfluss, $\omega(t)$ die Geschwindigkeit des Motors, $T_e = L/R$ die elektrische Zeitkonstante, L - Ankerkreisinduktivität und p der Laplace Operator ist.

Für die Unterdrückung der elektromotorischen Kraft des Motors wird ein neues Regelungsgesetz verwendet:

$$U_r(t) = (U_r^*(t) + C_w \omega(t)) \frac{1}{C_m}, \quad (23)$$

wobei $U_r^*(t)$ die Kompensationsgröße ist. Weiterhin kann man das Moment des Gleichstrommotors als Signal der Rückkopplung verwenden. Dazu kompensiert man die Rückkopplung des elektromagnetischen Gliedes. Dann kann man das Gesetz der Steuerung durch die folgende Formel beschreiben:

$$U_r(t) = (U_r^{**}(t) + C_w \omega(t) + M(t) \cdot 1/T_e) \frac{1}{C_m}. \quad (24)$$

Nun wird das Regelungsgesetz (24) in die Gleichung des Systems (22) eingesetzt. Anstelle des elektromagnetischen Gliedes bleibt nur der Integrator:

$$U_r^{**}(t) = M(t) p \quad (25)$$

Nach der Linearisierung wird der Roboter in ein System dritter Ordnung überführt.

Es bilden sich zwei Varianten der Steuerung:

1. Man lässt das elektromagnetische Glied ohne Veränderung und kompensiert nur die elektromotorische Kraft des Motors und den Verstärkungskoeffizient des Stromgliedes. Dann vernachlässigt man die Dynamik des elektrischen Teils des Motors und gewinnt in der Einfachheit des Steuerungsgesetzes für den mechanischen Teil des Systems. Das Blockdiagramm des Systems ist in Abbildung 90 gezeigt.

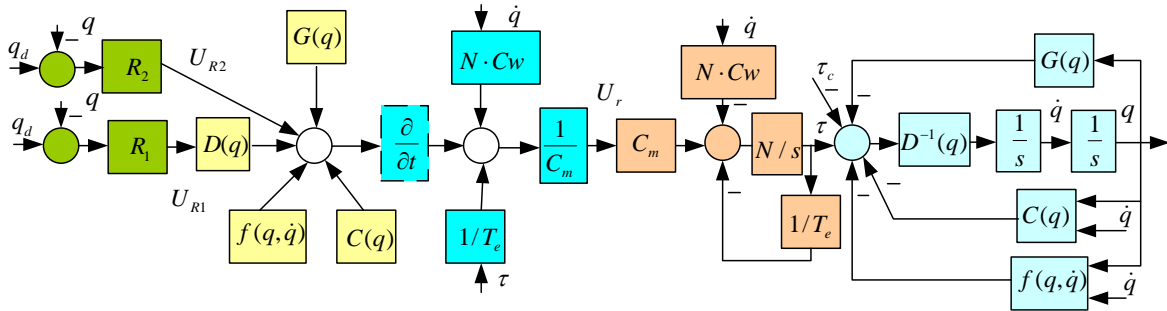


Abbildung 90. Verallgemeinerter Steuerschaltkreis mit Kompensation des Antriebs und der Nichtlinearitäten des Roboterarms

Wie es aus Abbildung 90 für den ersten Fall vom elektrischen Teil des Gleichstrommotors folgt, bleibt nur das Glied $\frac{1}{T_e p + 1}$, was auf seiner Kompensation verzichtet, wird die Übertragungsfunktion des

$$\text{Systems } W_o(s) \approx \frac{1}{T_e p + 1} \frac{1}{s^2}.$$

2. Man kompensiert das elektromagnetische Glied, welches zum Integrator wird, indem man das Regelungssignal (Abbildung 90) „differenziert“, weil das Gesetz der Linearisierung für den mechanischen Teil des Systems durch das Moment (siehe die Gleichung (25)) und nicht durch die Spannung des Motors U_r bestimmt wird. Eine reine Differenzierung ohne Stabilitätsverlust sehr problematisch. Man kann jedoch die Rückkopplung τ/T_e kompensieren und das Integrator des Gliedes $\frac{1}{T_e p + 1}$ mit anderer Rückkopplung τ/T_k ersetzen, wobei $T_k \ll T_e$.

Die Berechnung der Trägheitstensoren der Roboterarmglieder, unter Berücksichtigung der Übertragungszahl des Getriebes und des Trägheitsmomentes des Motors, wird wie folgt vorgenommen. Es ist bekannt, dass man bei der Berechnung der Trägheitsmatrix $D(q)$ einen Trägheitstensor verwendet, der die Verteilung der Masse des sich bewegenden Gliedes bezüglich der entsprechenden Freiheitsgraden beschreibt. Gewöhnlich schreibt man Trägheitstensoren in der Art:

$$J_i = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} \quad (26)$$

Wobei i die Nummer des Gliedes ist. Wenn man beispielsweise annimmt, dass sich das Glied nur um die Achse z dreht, muss man den Koeffizient J_{zz} unter Berücksichtigung der Motorträgheit und der Übertragungszahl des Getriebes korrigieren. Es gibt zwei Möglichkeiten die Kräfte und Geschwindigkeiten zu messen: entweder in Roboterwinkeln oder in Motorwinkeln. Von dieser Wahl hängt auch die Darstellung der Trägheitsmomente ab. Wenn die Bewegungen in Roboterwinkeln dargestellt werden, bekommt man die Gleichung:

$$J_{zz}^* = J_{zz} + J_{motor} N^2 \quad (27)$$

Wobei N die Übertragungszahl des Getriebes ist.

Wie es aus der Gleichung (27) ersichtlich ist, beeinflusst die Übertragungszahl des Getriebes die Wechselwirkung zwischen den verschiedenen Freiheitsgraden des Roboterarms stark.

Wenn die Übertragungszahl $N \gg 1$ ist, wird die entsprechende Komponente der Trägheitsmatrix J_i (Gleichung (26)) unter Berücksichtigung der Gleichung (27) und der entsprechenden

Komponente $J_{zz}^* \gg J_{zz}$ berechnet. Dies hat zur Folge, dass die Wechselwirkungen zwischen den Freiheitsgraden des mechanischen Systems vernachlässigt werden können. In Abbildung 90 ist das Moment $\tau = M(t) \cdot N$.

4.8.2 Der äußere und innere Regelkreis

Es existieren zwei Hauptverfahren für den Einsatz des Reglers. Für den ersten Fall wird der Regler vor der Kompensation der Nichtlinearitäten in den Steuerschaltkreis eingesetzt. In der Regel ist die Kompensation auf den Messungen von \dot{q} gegründet. Dieser Fall wird „computed torque control“ genannt [26]-[28]. Für den zweiten Fall wird der Regler in den Kreis der Steuerung nach der Kompensation der Nichtlinearitäten eingesetzt, in welcher die Vorgabewerte verwendet werden, die durch Integration der gewünschten Beschleunigung des Systems \ddot{q} berechnet wurden. Das ist für „Feedforward control“ traditionell [28].

Jedoch kommt es für den ersten Fall zu Problemen bei Störeinträgen, weil das Signal des Reglers vor der Kompensation der Nichtlinearitäten wirkt. Das zweite Verfahren kommt besser mit Störungen wie dem äußeren Widerstandsmoment zurecht, da dieses auf den Regler mit einwirkt.

Der verallgemeinernde Steuerschaltkreis des Systems unter Berücksichtigung aller kompensierenden Signale ist in Abbildung 90 dargestellt. Aus der Zeichnung ist erkennbar, dass sich das Signal der Steuerung aus den Signalen der Kompensation der elektromotorischen Kraft $\dot{q}C_w$, der Kompensation des Koeffizienten der Verstärkung des elektrischen Teiles des Motors C_m , der Kompensation der Gravitationskräfte, der Corioliskräfte, der Kräfte der Reibung, und der Matrix des Trägheit $D^{-1}(q)$ zusammensetzt.

Der Regler funktioniert entsprechend der Abweichung zwischen der realen Lage q und dem Sollwert q_d . In Bezug auf Störungen wie dem Widerstandsmoment τ_c arbeitet der Regler R_1 indirekt, da bei diesem System die Beschleunigung nur mit Nutzung von U_{R1} bekommt:

$$\ddot{q} = D^{-1}(q) \cdot \left(U_{R1} \cdot \tilde{D}(q) - \tau_c - C(q, \dot{q}) - G(q) - f(q, \dot{q}) + \tilde{C}(q, \dot{q}) + \tilde{G}(q) + \tilde{f}(q, \dot{q}) \right), \quad (28)$$

Wobei $\tilde{C}(q, \dot{q}) + \tilde{G}(q) + \tilde{f}(q, \dot{q})$ die modellberechneten Kräfte, $\tilde{D}(q)$ die modellberechnete Matrix der Trägheit, $-\tau_c - C(q, \dot{q}) - G(q) - f(q, \dot{q})$ die realen Kräfte und $D(q)$ die reale Matrix der Trägheit sind.

Man muss beachten, dass die realen Werte der Systemparameter nur annähernd bekannt sind.

Wie es aus Gleichung **Error! Reference source not found.** erkennbar ist, soll das Signal des Reglers U_{R1} die Störung τ_c neutralisieren, jedoch nimmt dieses Einfluss auf die Trägheitsmatrix $D(q)$, weil $D^{-1}(q)\tilde{D}(q) \approx 1$ ist. Der Regler R_2 , der die Kompensation der Störung bewirkt, hat keinen Einfluss auf die Trägheitsmatrix. Das heißt, dass der Regler R_2 im Vergleich zum Regler R_1 im *inneren* Regelkreis arbeitet. Auf diese Weise bekommt man die gemeinsame Systemgleichung:

$$\ddot{q} = D^{-1}(q) \cdot \left(U_{R1} \cdot \tilde{D}(q) + U_{R2} - \tau_c - C(q, \dot{q}) - G(q) - f(q, \dot{q}) + \tilde{C}(q, \dot{q}) + \tilde{G}(q) + \tilde{f}(q, \dot{q}) \right). \quad (29)$$

Die Regler R_1 und R_2 können zusammen oder getrennt funktionieren.

Als ein Fazit ist festzustellen, dass die Realisierung von „computed torque control“ mit Hilfe der inversen Dynamik die Neutralisation der Nichtlinearitäten des Roboterarms ermöglicht.

4.9 Die modernen Regelungssysteme und Regelungskonzepte eines mechanischen Systems

Jedes Jahr erscheinen mechanischen Systeme höherer Komplexität[24]-[29] und immer mehr innovative Mechanismen und Materialien finden Anwendung im Aufbau von mechanischen Systemen. Dies bringt jedoch neue Problemstellungen und Herausforderungen für die Regelung solcher Systeme mit sich. Wenn über Probleme der Mechanik gesprochen wird, sind meist unbekannte Parameter bzw. Störgrößen in der Regelstrecke gemeint. Heutzutage ist es relativ leicht möglich ein Modell für die Regelstrecke zu entwerfen, jedoch ist es schwierig Modellparameter zu definieren, die das Realverhalten genau widerspiegeln.

Ein genaues Modell erlaubt die Implementierung beispielsweise „computed torque control“ Methode, um die gewünschte Regelgüte zu erreichen. Das ist jedoch nicht möglich, wenn die Modellparameter eine relativ große Abweichung von den realen Werten aufweisen. Solche Abweichungen führen zunächst zur Verschlechterung der Regelungsqualität und im Extremfall zum Verlust der Stabilität.

Systeme, welche keine genauen Parameterwerte oder Modellgleichungen benötigen lassen sich folgendermaßen einteilen:

Adaptive Regelungssysteme mit Identifikation der Regelstrecke (indirect adaptive control). Diese Systeme haben Probleme mit nichtlinearen Systemen und haben einen hohen Berechnungsaufwand durch die Echtzeitberechnung der Regelstrecke und des Reglers.

Robuste Regelungssysteme verwenden ein Modell der Regelstrecke mit typischen Parameterwerten. Dabei wird die Empfindlichkeit des Reglers minimiert, indem man die Empfindlichkeit der Parameteränderungen abschätzt.

Adaptive Regelungssysteme, welche nach den Methoden „Singular Perturbation Approach“ oder „Passivity Based Approach“ aufgebaut sind, verwenden spezielle adaptive Signale, um den Regelkreis zu stabilisieren und die Regelqualität zu verbessern. Diese Systeme wurden als Verbesserung für die „computed torque control“ Methode verwendet, welche die Stabilität bei Parameteränderungen gewährleistet.

Adaptive Regelungssysteme kommen ohne Modellwissen aus und sind damit die besten Systeme im Sinne der Unabhängigkeit von Modellparametern und Systemgleichungen. Für solche Systeme verwendet man adaptive Zweipunkt-Regler und adaptive Linearisierungsverfahren[31].

Im Rahmen dieser Dissertationsarbeit wurde die „adaptive neuro-fuzzy sliding mode“ Methode untersucht[32].

Die Ergebnisse zeigen, dass man die beste Regelungsqualität bei einer relativ einfachen Reglerstruktur mit Verwendung einer adaptiven Linearisierung erreicht. Es gibt zahlreiche Publikationen über adaptive Linearisierung der Regelstrecke mittels neuronaler Netze oder neuro-fuzzy Netze (NFN) und anderen universellen Approximationsverfahren[33]-[37] (siehe auch Kapitel 1). In allen Fällen wurden die nichtlinearen Differentialgleichungen in Form der Gleichung (49) Kapitel 1 oder in Cauchy Form verwendet. Bei mehrdimensionalen Systemen haben solche Verfahren numerische Probleme wegen der Berechnungskomplexität und der benötigten Konvergenzfähigkeit. Es gibt folgende Annahmen und daraus folgende Schwierigkeiten.

Die Struktur der Regelstrecke ist bekannt und es ist lediglich nötig die Parameter zu identifizieren. Dies funktioniert gut bei einer relativ einfachen Regelstreckestruktur und geringer Anzahl von Parametern.

Die Struktur der Regelstrecke ist unbekannt, kann jedoch in Cauchy-Form oder in einer anderen allgemeinen Systemgleichung (z. B. Lagrange-Euler Gleichungen) dargestellt werden. Manche Autoren folgen dieser Logik und versuchen alle Elemente in diesen Gleichungen zu identifizieren, aber bei höherer Systemordnung sind solche Verfahren sehr komplex und berechnungsaufwendig.

Die Struktur der Regelstrecke ist bei gewöhnlichen Systemparametern bekannt. Bei Betrieb soll das Modell in Echtzeit angepasst werden. Dieser Logik folgt z.B. die „Passivity Based Approach“

Methode. Diese Methode bietet gute Stabilitätseigenschaften und besitzt eine exakte mathematische Beschreibung.

Im Kapitel 2, Gleichungen (51)-(57) wurde ein neues Regelungssystem vorgeschlagen. Dabei verwendet man eine Linearisierung der Regelstrecke. Die Besonderheit dieses Systems ist die Verminderung von unbekanntem Signalwerten, die man die für Rückkopplung der Regelstrecke benötigt. Dieses oben genannte Verfahren ist auch für den Roboterarm oder andere nichtlineare Regelstrecke verwendbar.

4.9.1 Adaptive Regelung mit Ersatzfunktionen

Die *Hauptidee* dieses Verfahren ist die Auswahl des Systemmodells in solcher Form, dass sich „die Linearisierung“ des Systemmodells leicht ausführen lässt.

Es ist ebenfalls möglich *die Linearisierung der Regelstrecke als Optimierungsproblem darzustellen*. In Verbindung mit der Gleichung $\dot{x} = F(x, u)$ ergibt dies die Beziehung:

$$\dot{x} = F(x, u) = v \tag{30}$$

zum Zweck der Linearisierung. Hier und im Weiteren ist die Variable x eine unabhängige Größe und u ist die manipulierbare Variable. Für die Optimierung ist die Formulierung der Aufgabe als $J(x, u) \rightarrow \min$ typisch, wobei J – ist ein Gütekriterium. In diesem Sinn ist die Linearisierung (Gleichung (30)) als Optimierungsproblem z.B. mit folgender Formel realisiert:

$$(F(x, u) - v) \rightarrow \min \tag{31}$$

Bei der Verwendung von klassischen Gradientenverfahren müssen die Gradienten $\frac{\partial(F(x, u) - v)}{\partial u}$ bekannt sein. Sind die Systemgleichungen nicht bekannt, ist die Anwendung dieses Verfahrens nicht möglich.

In dieser Arbeit wird vorgeschlagen eine Ersatzfunktion von $F(x, u)$ auszuwählen, die das reale System annähernd genau beschreibt. Ist eine Annäherung zur realen Funktion $F(x, u)$ gefunden, ermöglicht dies die Bestimmung eines Signals u , welches eine Berechnung (30) gewährleistet.

Zur Auswahl solcher Ersatzfunktionen stehen beliebige invertierbare Funktionen, die zumindest einen unbekanntem Parameter enthalten. In der ersten Zeile der Tabelle 11 verwendet die Funktion nur Summierungen. Deswegen gibt es keine Ausnahmereiche wie bei den Funktionen ab der dritten Zeile. Die Auswahl der Ersatzfunktion beeinflusst ebenfalls die Konvergenz der Optimierung. So liefert die Funktion $y = kx$ das Optimum für die Rosenbrock-, „Bananenfunktion“ zehn bis hundert Mal schneller als die Ersatzfunktion $y = k + x$. Die Ersatzfunktion $y = k(x + b)$ mit $b = \text{const}$ bietet den Vorteil der Berechnungsbedingung $x_i - b \neq 0$. Potentielle Ersatzfunktionen wie $y = k^x$ zeigen noch höhere Konvergenz, besitzen jedoch schlechtere Stabilitätseigenschaften als $y = kx$. Allgemein lässt dies vermuten, dass Ersatzfunktionen instabiler werden, je schneller sie konvergieren, sofern keine speziellen Stabilitätsverfahren verwendet werden. Solche spezielle Verfahren wurden im Kapitel 1 als robuste Modifikationen von Gradientenalgorithmien beschrieben.

Tabelle 11. Die Ersatzfunktionen mit einem unbekanntem Parameter k

Ersatzfunktion	Umrechnung	eingesetzt	Ergebnis	Bedingung
$y = k + u$	$k_i = y_i - u_i$	$y_{new} = k + (-k_i + v)$	$y = v$	-
$y = k + \sin(u)$	$k_i = y_i - \sin(u_i)$	$y_{new} = k + \sin(\arcsin(k_i + v))$	$y = v$	-
$y = ku$	$k_i = y_i / u_i$	$y_{new} = k(k_i^{-1}v)$	$y = v$	$u_i \neq 0$

$y = u/k$	$k_i = u_i / y_i$	$y_{new} = k_i^{-1}(k \cdot v)$	$y = v$	$y_i \neq 0$
$y = ku + b$	$k_i = (y_i - b) / u_i$	$y_{new} = k(k_i^{-1}v) + b$	$y = v$	$u_i \neq 0$ $b = const$
$y = k(u + b)$	$k_i = y_i / (u_i - b)$	$y_{new} = k(k_i^{-1}v - b)$	$y = v$	$x_i - b \neq 0$ $b = const$
$y = k^u$	$k_i = \sqrt[u_i]{y_i}$	$y_{new} = k^{(\log(v)/\log(k_i))}$	$y = v$	$v > 0$ $k > 0$
$y = u^k$	$k_i = \frac{\log(y_i)}{\log(u_i)}$	$y_{new} = (\sqrt[k_i]{v})^k$	$y = v$	$u > 0$ $y > 0$

Wie leicht zu erkennen ist, kann der unbekannte Parameter der Ersatzfunktion mit der Methode der kleinsten Quadrate berechnet werden. Wenn man für N-dimensionale Probleme (d.h. $F(x, u) \rightarrow \min$

und $u \in R^N$) die Ersatzfunktion $y = \sum_{i=1}^N k_i \cdot u_i + b$ nimmt, erhält man das so genannte Simplex

Verfahren. Es ist auch möglich andere lineare und nichtlineare Ersatzfunktionen mit **mehreren** unbekanntem Parametern zu verwenden. Die Simplex Methode ist eine lineare Näherung zur Funktion $F(x, u)$. Für die quadratische Näherung kann man z.B. ein Paraboloid in Form

$y = \sum_{i=1}^N k_1^2 \cdot u_i + k_{2i} \cdot u_i + k_{3i}$ auswählen. Eine detaillierte Analyse dieses Themas soll jedoch nicht

Gegenstand der vorliegenden Arbeit sein.

Die *Optimierung mittels einer Ersatzfunktion* kann auf folgende Weise zusammengefasst werden:

1. Für die Optimierungsfunktion $y = F(x, u)$ wird eine Ersatzfunktion $y = f(k, u)$ angenommen.
2. Nach den Messungen und Berechnungen muss die Bedingung $F(x, u) \cong f(k, u)$ im Lokalbereich gültig sein. Hier berechnet man die unbekanntem Parametern k_i der Ersatzfunktion $y = f(k, u)$.
3. Für die Konvergenz des Gütekriteriums $F(x, u) \rightarrow v$ muss der nächste Optimierungsschritt u_{i+1} aus der Lösung von der Gleichung $f(k_i, u_{i+1}) = v$ angenommen und in die optimierte Funktion $y_{i+1} = F(x, u_{i+1})$ eingesetzt werden (Tabelle 11, Spalte 3).
4. Die Stabilität des Optimierungsverfahrens ist durch die Monotonie $y_i - v \geq y_{i+1} - v$ gewährleistet, welche mittels einer Normierung der Schrittgröße Δu_i erreicht wird.

Die Punkten 2-4 wiederholt man bis zum Erreichen der Haltbedingung $\|y_i - v\| \leq \varepsilon$, wobei $\varepsilon > 0$ eine Konstante ist. Der Unterschied zwischen der vorgeschlagenen Methode und anderen zahlreichen Adaptationsverfahren [38]-[48] ist die Teilung der Regelstrecke in den Integrationsteil und den nichtlinearen Teil. Dies ist z.B. für Lagrange-Euler-Gleichungen möglich.

Der zweite Unterschied ist die Verwendung des gradientenfreien Optimierungsalgorithmus auf Basis von Ersatzfunktionen. Ergänzend ist zu erwähnen, dass es im Optimierungsverfahren mittels Ersatzfunktionen auch möglich ist, Grenzbedingungen wie beim gradientenbasierten Verfahren zu berücksichtigen (siehe auch Kapitel 1). Die Simulationsergebnisse werden in dem nächsten Abschnitt gezeigt.

4.9.2 Positionsregelung

Um die Ergebnisse von unterschiedlichen Regelungssystemen für gleiche Regelstrecke miteinander vergleichen zu können, wurde ein PD-Regler eingesetzt. Das Regelungsgesetz lässt sich mit

$$U_R = Kp \cdot (q_{soll} - q_{ist}) + Kd(\dot{q}_{soll} - \dot{q}_{ist}) \quad (32)$$

beschreiben.

Für den Antrieb des Roboterarms wurden Gleichstrommotore ausgewählt. Die Parameter der Motoren sieht man im Anhang A. Aus der gewünschter Regelgüte ergeben sich die Parameter vom PD-Regler zu $Kd=10$ und $Kp=250$. Der Übersetzungsfaktor N vom Getriebe wird als 60 angenommen. Jeder Achse wird unabhängig von einem eigenen Motor angetrieben.

In der Abbildung 91 ist der Positionsverlauf (links) der Roboterachsen und der Spannungsverlauf (rechts) dargestellt.

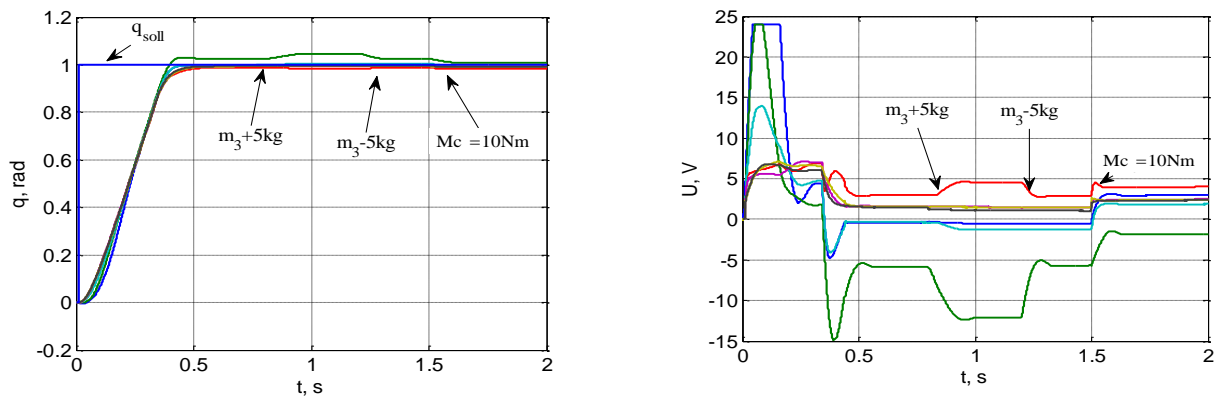


Abbildung 91. Reine PD-Regelung ohne Kompensation, $N=60$, $Kd=10$, $Kp=250$

Das Sollsignal wird stufenartig angelegt. Um eine Überspannung zu vermeiden, wurde der Anstieg vom Sollsignal auf 3 rad/s begrenzt. Wie man aus der Abbildung 91 (links) erkennt, besitzt der PD-Regler einen statischen Fehler, der schon ohne zusätzliche Störungen auftritt. Bei 0,8 s wurde die Masse vom letzten Gelenk auf 5 kg stufenartig erhöht. Dies kann bei Serviceoperationen auftreten, wenn der Roboterarm ein 5 kg Gewicht aufnehmen und die Position halten soll. Zum Zeitpunkt $t = 1,2$ s wurde die Last entfernt. Bei $t = 1,5$ s wurde ein Störmoment für alle Freiheitsgrade in Höhe von $M_c=10$ Nm hinzugefügt. Die größte Abweichung hat der Schulterfreiheitsgrad, welcher das gesamte Gewicht aufnimmt. Das ist derselbe Freiheitsgrad θ_1 entsprechend der Abbildung 74 (links). Die Maximalabweichung bei einer zusätzlichen Masse beträgt $\Delta q=0,05$ rad. Der zweite schwer belastete Freiheitsgrad ist θ_3 , entsprechend Abbildung 74 (links). Die zwei höchsten Spannungen in Abbildung 91 treten bei θ_1 und θ_3 auf und sind gleich groß.

Wie bekannt ist, kann man die Wechselwirkung zwischen den Freiheitsgraden und der Nichtlinearität der Regelstrecke mit Hilfe des „Computed Torque Control“ Verfahren vermeiden. In Abbildung 92 (links) ist der Zeitverlauf der Freiheitsgrade des Roboterarms dargestellt. Das Regelungssystem entspricht der Abbildung 90 und ermöglicht das gewünschte Regelungsverhältnis. Trägt der Roboter eine zusätzliche Masse in dem Arm, unterscheidet sich das verwendete Dynamikmodell der

Regelstrecke vom Realsystem, was eine Positionsabweichung $\Delta q=0.06$ im Zeitraum von 0,8 bis 1,2 s zur Folge hat.

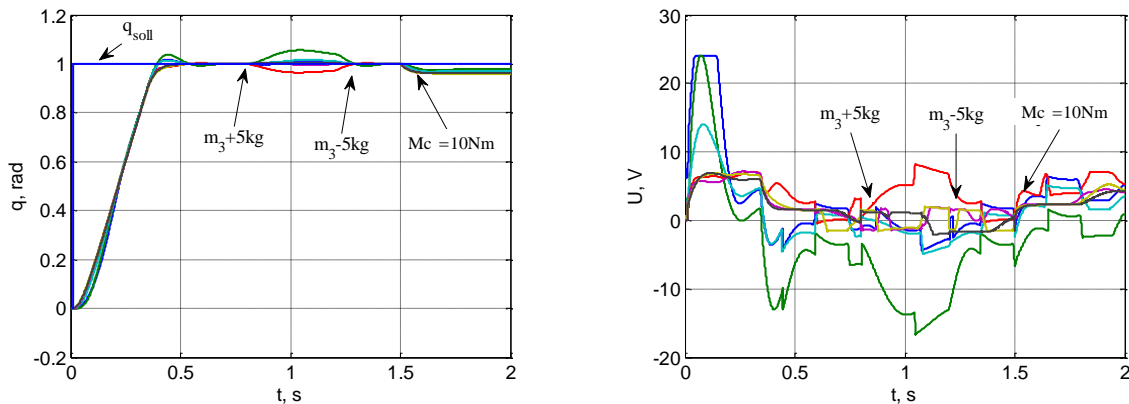


Abbildung 92. „Computed Torque Control“ mit PD-Regelung, $N=60$, $K_d=10$ $K_p=250$, $m_{3inv}=m_3$

In Abbildung 93 sieht man den Zeitverlauf der Freiheitsgrade (links) des Roboterarmes und die entsprechende Motorspannungen (rechts). Im Unterschied zum vorangegangenen Versuch wurde die Masse des letzten Armglieds um 5 kg erhöht. In Abbildung 93 (links) sieht man im Zeitraum von 0,8 s bis 1,2 s, dass die Positionsabweichung minimiert wurde. Diese Positionsabweichung treten jedoch erneut auf, wenn sich die Massen im Systemmodell und im Realsystem wieder unterscheiden. Die Maximalabweichung beträgt $\Delta q=0,02$ rad.

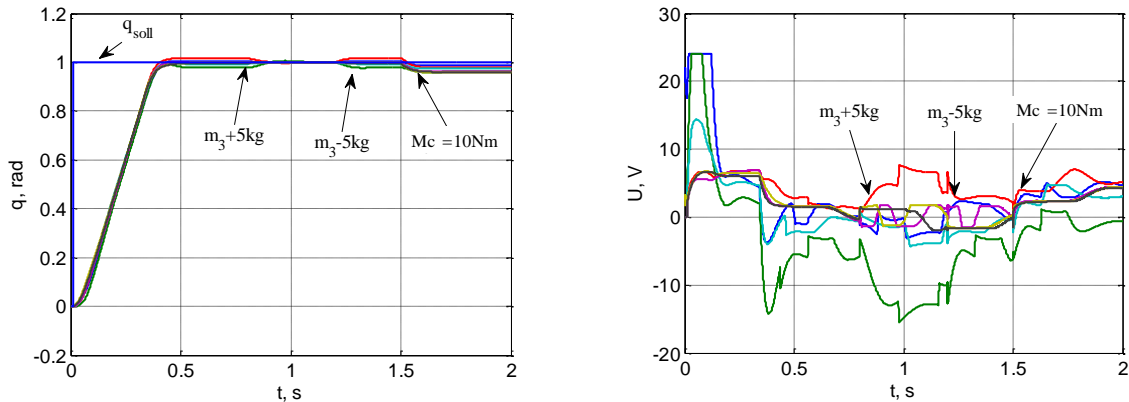


Abbildung 93. „Computed Torque Control“ mit PD-Regelung, $N=60$, $K_d=10$ $K_p=250$, $m_{3inv}=m_3+5\text{ kg}$

Aus der Abbildung 93 geht hervor, dass das „Computed Torque Control“ Verfahren sehr genaue Systemparameter in Bezug auf das Realsystem benötigt, andernfalls verschlechtert sich die Regelungsqualität und es treten große Regelungsfehler auf.

Die Lösungen dieses Problems wurden in Paragraph § 4.5 behandelt.

Durch die Verwendung des Linearisierungsverfahrens mittels Ersatzfunktion $y = k + u$ bekommt man ein Regelungssystem wie in Abbildung 94 dargestellt. Ein ähnliches Verfahren wurde in Kapitel 2 für die Regelung von Magnetlager mit 5 Freiheitsgraden implementiert.

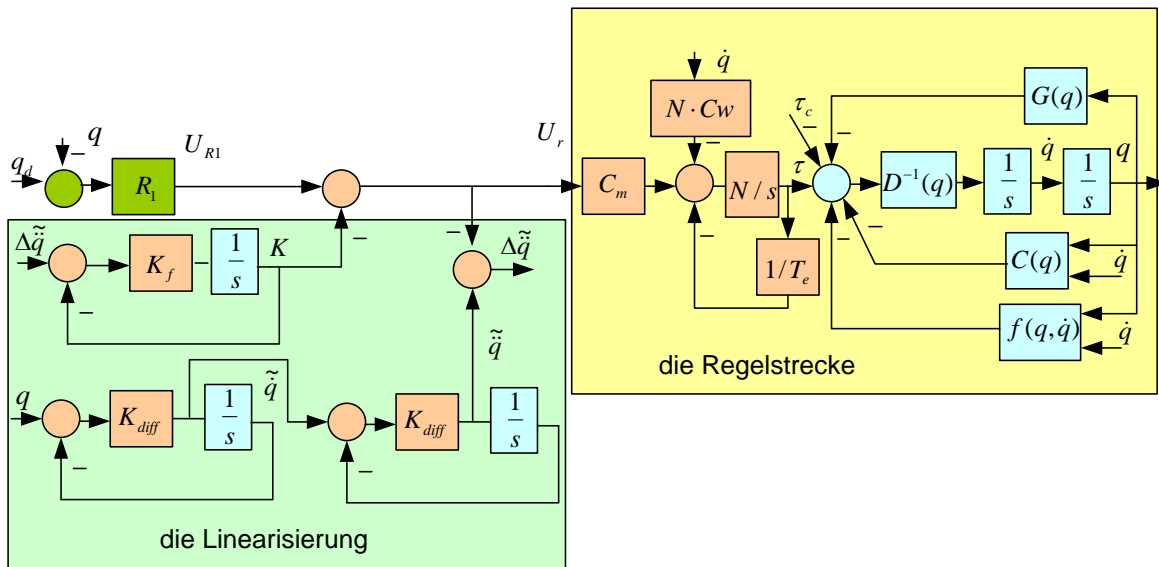


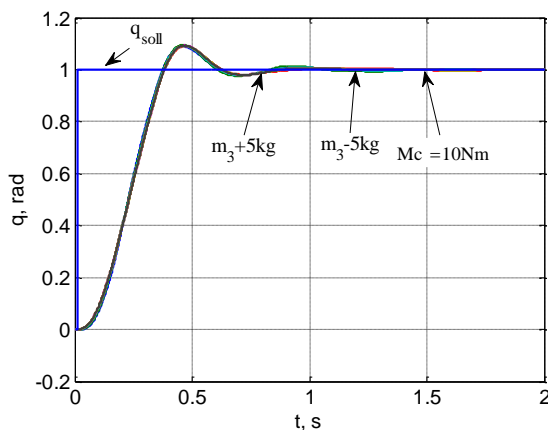
Abbildung 94. Positionsregelung des Roboterarms mit Linearisierung mittels Ersatzfunktion $y = k + u$

Das Regelstreckenmodell in Abbildung 94 besteht aus einem mechanischen Teil und einem elektrischen Teil mit den Gleichstrommotorantrieben. Der Linearisierungskreis ist ebenfalls in Abbildung 94 dargestellt.

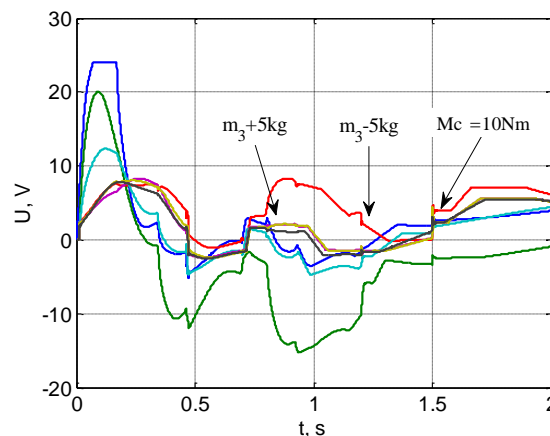
Hier wurde, wie in Kapitel 2, die reine Linearisierung mittels Ersatzfunktion durch einen Beschleunigungsbeobachter ergänzt. Die Konvergenz des Beschleunigungsbeobachters wird mit dem Parameter K_{diff} beeinflusst. Durch diesen Beobachter wurde ein geschätztes Signal der Beschleunigung $\tilde{\ddot{q}}$ berechnet. Die Differenz $\Delta\tilde{\dot{q}}$ ist gleich dem Parameter k aus der Ersatzfunktion $y = k + u$. Um einen Anfangssprung zu dämpfen wurde ein Filter mit dem Faktor K_f in den Kreis aufgenommen. Wird das Signal K aus dem Signal U_R des Linearisierungskreises subtrahiert, wird das Signal U_R annähernd gleich $\tilde{\dot{q}}$. Bei der Simulation sind die Parameter der Linearisierung $K_{diff} = 1000$ und $K_f = 100$.

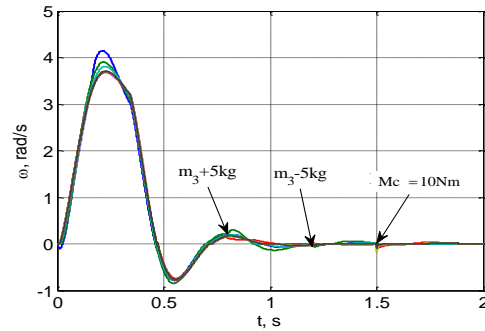
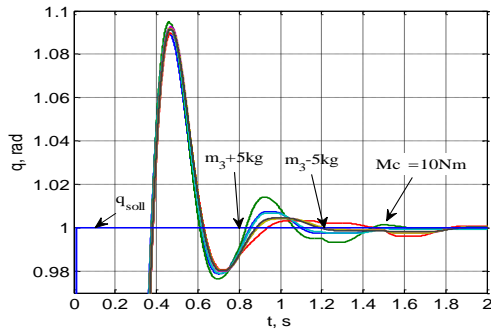
In Abbildung 95 a) und c) sieht man den Zeitverlauf der Freiheitsgrade und in Abbildung 95 b) ist der Spannungsverlauf der Gleichstrommotoren dargestellt. Die Untersuchungen zeigen, dass das vorgeschlagene Regelungssystem keine Schwankungen hat und keine Überlastungen verursacht. In Abbildung 95 d) sieht man den Zeitverlauf der Geschwindigkeiten.

a)



b)





c)

d)

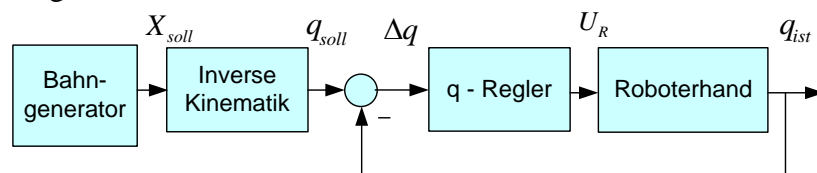
Abbildung 95. Linearisierung mit Hilfe von Ersatzfunktion $y = k + u$ mit PD-Regelung, $N=60$, $K_d=10$, $K_p=250$

Im Ergebnis der Untersuchungen kann man zusammenfassend sagen, dass die Linearisierung mittels Ersatzfunktion $y = k + u$ dasselbe Verhalten mit und ohne Kompensation der Motordynamik aufweist. Für diese Regelungssysteme benötigt man kein genaueres Systemwissen über die Massenverteilung oder die Geometrie. Bei Verwendung eines PD-Reglers entstehen statische Fehler, welche als Folge des Linearisierungskreises und dem Ausgleich von U_R und \ddot{q} nivelliert werden.

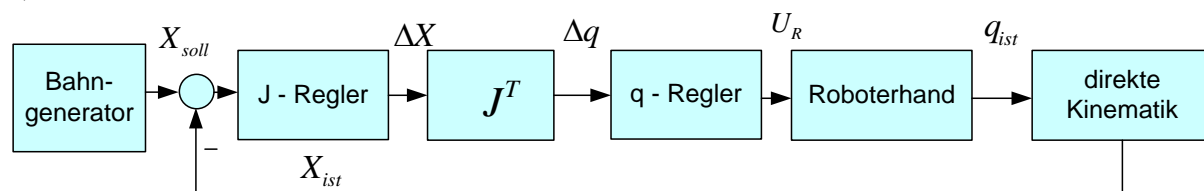
4.9.3 Positionsregelung in kartesische Koordinaten

Für die Positionsregelung in kartesische Koordinaten wird die Bewegung des letzten Armpunktes (siehe auch Abbildung 74 (links)) in globalen Koordinaten mit Hilfe der direkten Kinematik berechnet. Die Aufgaben der direkten und inversen Kinematik sind in § 4.2 beschrieben.

Wurde eine inverse Kinematik mit einem geschlossenen inneren Optimierungskreis verwendet, in diesem Fall ist eine Lösung so zu wählen, dass die Abweichung zwischen der berechneten und der gewünschten Bahnkurve möglichst klein werden. Diese Regelungsstruktur ist in Abbildung 96 a) dargestellt.



a)



b)

Abbildung 96 . a) Positionsregelung in kartesischen Koordinaten beim Levenberg-Marquardt-Verfahren in inverser Kinematik

b) Positionsregelung in kartesischen Koordinaten bei der jacobi-basierten inversen Kinematik

Wie hier gezeigt ist, treten Positionsfehler nur auf, wenn die Regelung den lokalen q Koordinaten des Roboterarms nachfolgt. Solche Regelungssysteme haben den Vorteil, dass die Regelungsqualität nur vom Regler in q -Koordinaten abhängig ist. Nachteilig ist der relativ hohe Berechnungsaufwand, den die inverse Kinematik verursacht.

Der Zeitverlauf der Armbewegung in kartesischen Koordinaten nach dem Schema aus Abbildung 96 a) ist in Abbildung 98 dargestellt. Aus Gründen der Übersichtlichkeit ist dabei die Orientierung des Greifers nicht in den Abbildungen erfasst. Die Regelung in kartesischen Koordinaten erlaubt neben der Positionsregelung aber ebenfalls die Orientierungsregelung des Greifers.

Für dieses Regelungssystem gelten folgenden Besonderheiten:

Das System hat dynamische Fehler in den globalen kartesischen Koordinaten, weil die Positionierungsfehler nur in lokalen Koordinaten verfolgt werden (Abbildung 98 b).

Die Abweichung zwischen der gewünschten kartesischen Bahnkurve und berechneten Bahnkurve kann auf ein beliebiges Niveau reduziert werden.

In Abbildung 98 b) sieht man die Positionsabweichung durch ein Störmoment von $M_c = 10 Nm$, welches zum Zeitpunkt $t = 2.8 s$ angelegt wurde. Die Spannung und die dynamischen Fehler in lokalen Koordinaten sind in Abbildung 98 c) und d) dargestellt.

Bei der Simulation wurden die Bahnkurven „offline“ für jeden Positionspunkt in zehn Optimierungszyklen berechnet. Die mittlere Laufbahnabweichung zu den gewünschten Werten beträgt weniger als $10^{-4} m$.

Die inverse Kinematik, unter Verwendung des Levenberg-Marquardt-Algorithmus, ist auch in Echtzeit möglich (siehe auch §4.2), wenn der Algorithmus die aktuellen Positionsdaten des Roboterarms geliefert bekommt. Die Positionsregelung mit der jacobi-basierten inversen Kinematik ist dem Gradientenabstiegsverfahren gleich, wobei der J-Regler als Proportionalregler dieselbe Bedeutung wie der Geschwindigkeitskoeffizient η in der Gleichung $q_{n+1} = q_n - \eta J_{ac} \Delta X$ hat. Bei der Simulation ist der Koeffizient des J- Reglers $K_j = 50$.

Die Positionsregelung mit jacobi-basierter inverser Kinematik, die in der Abbildung 96 b) dargestellt ist, hat folgenden Besonderheiten:

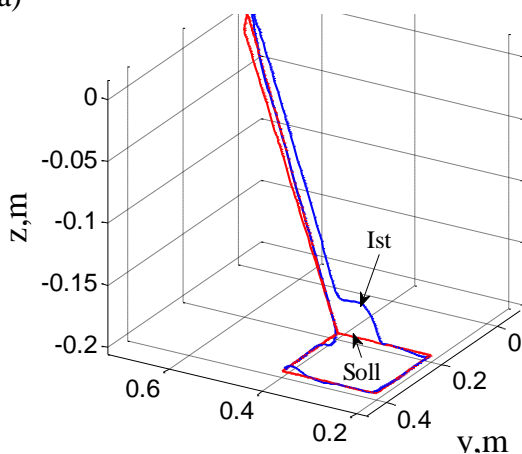
Die jacobi-basierte inverse Kinematik braucht einen relativ geringen Berechnungsaufwand, um die benötigten q -Bahnkurven zu ermitteln.

Die Bahnkurven bekommt man nur in relativen Koordinaten Δq .

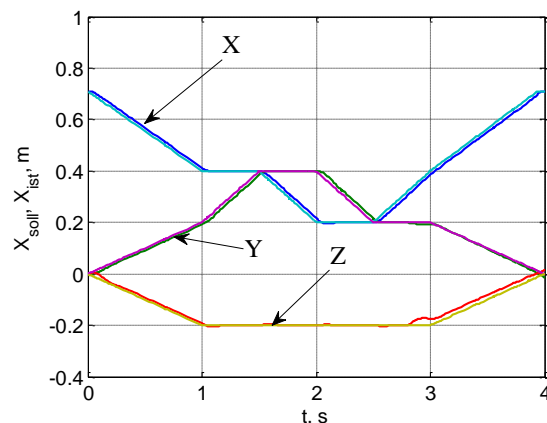
Es ist notwendig einen zusätzlichen proportionalen J-Regler hinzuzufügen, um die Dynamik vom Positionsregelkreis konstant zu halten, bei Verwendung eines q-Reglers.

Es ist möglich den q-Regler an die Stelle des J-Reglers zu verschieben, jedoch müssen dann die Reglerkoeffizienten neu abgestimmt werden.

a)



b)



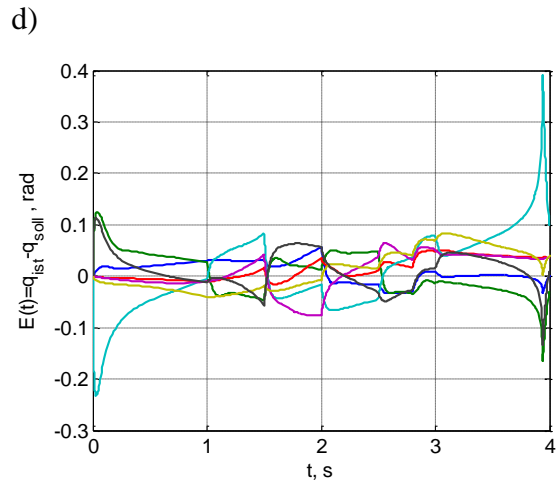
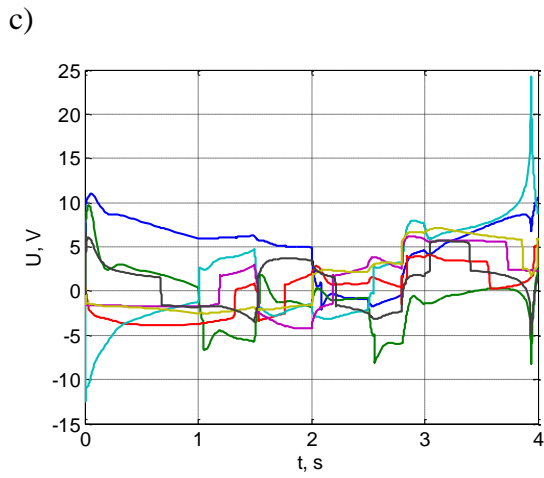


Abbildung 97. PD-Positionsregelung in kartesischen Koordinaten mit Levenberg-Marquardt-Verfahren in inverser Kinematik

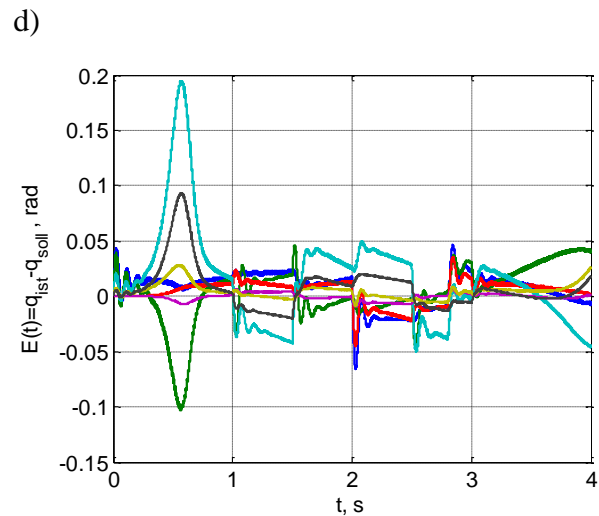
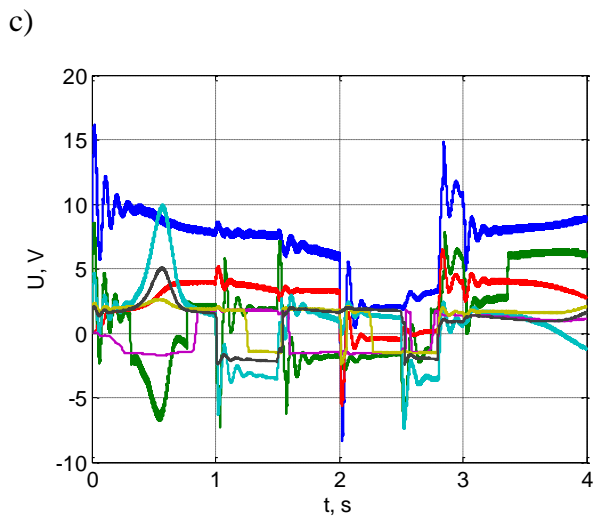
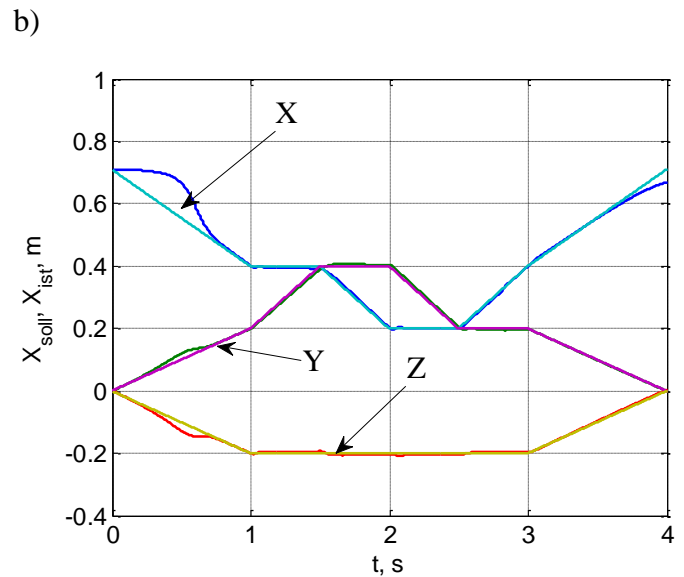
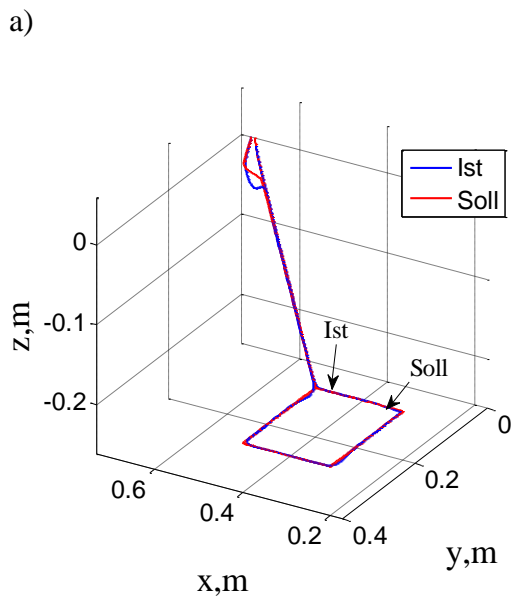


Abbildung 98. Jacobi-basierte inverse Kinematik und PD-Regelung in kartesischen Koordinaten

In Abbildung 98 a) sieht man die Anfangsabweichung der Armposition. Diese ist durch die jacobibasierte inverse Kinematik verursacht, weil der Anfangsfehler sehr groß ist und dessen Abklingzeit 0,6 s beträgt.

Die dynamische Positionsabweichung verschwindet (Abbildung 98 a) b)), weil der Regelkreis in kartesische Koordinaten rechnet. Der Zeitverlauf der Motorspannungen und der dynamische Fehlern ist in Abbildung 98c) und d) dargestellt.

4.9.4 Die Positions -Kraftregelung des Roboterarms

Als Kraftregelung wird die Regelung der Kraft bezeichnet, mit welcher der Roboterarm auf die Objekte einwirken kann. Die Kraftregelung ist immer mit der Positionsregelung gekoppelt, weil die Bewegung im Kontakt mit gegenwirkender Umgebung eine Kontaktkraft erzeugt.

Zur Kraftaufnahme existieren Kraftsensoren, die Kräfte und Drehmomente in allen drei Raumrichtungen messen können, die Kräfte können jedoch anhand der Motorströme geschätzt werden.

Im Umfeld der Industrie- und Servicerobotik ist die Regelung der Kontaktkraft zwischen einem Roboter und seiner Umgebung eine wichtige Aufgabe. Ein prinzipieller Unterscheid besteht zwischen Anwendungen mit ausgeprägtem Kontakt und Anwendungen mit potentiellm Kontakt. Der Kontakt des Roboters mit der Umwelt oder dem Werkstück ist zentraler Bestandteil der Aufgabe bei Anwendungen mit ausgeprägtem Kontakt und muss explizit geregelt werden. Dazu gehören Aufgaben der mechanischen Verformung und Oberflächenbearbeitung[24]-[29]. Die Positionierung des Roboters bei Aufgaben mit potentiellm Kontakt ist sehr wesentlich. Durch eine dynamische Umwelt oder ein ungenaues Umweltmodell können größere Kontaktkräfte zwischen Roboter und Umwelt entstehen. In diesem Fall soll das Robotersystem der Umwelt nachgeben und große Kontaktkräfte vermeiden.

Die Hauptanwendungen der Kraftregelung sind Fertigungsarbeiten, wie zum Beispiel Polieren, Schleifen sowie kraftgesteuerte Prozesse, wie das kontrollierte Fügen, Schweißen usw. Die Abtastung unbekannter Oberflächen ist ein häufiger Einsatz von Kraftregelung.

Die Regelungskonzepte unterscheiden sich nach verschiedenen Kriterien die in der Tabelle 12 dargestellt sind.

Tabelle 12 Die Positions- Kraftregelungssysteme

Kraftmessung	Direkte Kraftmessung		Indirekte Kraftmessung	
Geregelte Variabel	Kraft	Impedanz	Nachgiebigkeit	Position
Kontaktparametern	Bekannt		Unbekannt	Identifiziert
Der Regler	Adaptiv	Robust	Mit Linearisierung	Ständige Parametern
Regelkreis	Hybrid	Parallel	Separate Kraftgeregelt Freiheitsgrade	
Kontaktmodell nach Geometrie	Punktkontakt	Flächenkontakt		Mehrpunktkontakt
Kontaktmodell nach Dynamik	Einfache Feder		Ein paar serienweise gekoppelten Federn	
	Die Feder mit variablen Steifigkeit und Dämpfung			
Wirkungsart	Kontaktkraft		Kontaktmoment	

Ein häufiger Anwendungsfall für direkte Kraftmessung und bekannte, definierte Umgebungsparametern sind Industrieroboter.

In der Medizintechnik, Spielrobotertechnik, intelligenten Robotertechnik und bei kooperierenden Robotern finden sich weitere Anwendungen der Kraftregelung mit potentiellm Kontakt. Ist eine genaue Kraftregelung häufig nicht nötig, dann muss die Mechanik adaptiv auf die unterschiedlichen

Umgebungsparameter reagieren können. In solchen Systemen Verfahren mit Nachgiebigkeitsregelung und anderen Regelungsalgorithmen mit indirekten Kraftermittlungen anwendbar. Für die Simulation der Kraftregelung wurde eine parallele Kraftregelung entworfen. Der Kraftregelkreis und innere Positionsregelkreis sind in Kaskadenstruktur angeordnet. In Abbildung 99 ist das Schema der parallele Positions- Kraftregelung dargestellt. Eine Besonderheit besteht in der Anwendung der inversen Kinematik mit dem Levenberg-Marquardt Verfahren. Der Optimierungsalgorithmus findet die genauen Werte für q_{soll} .

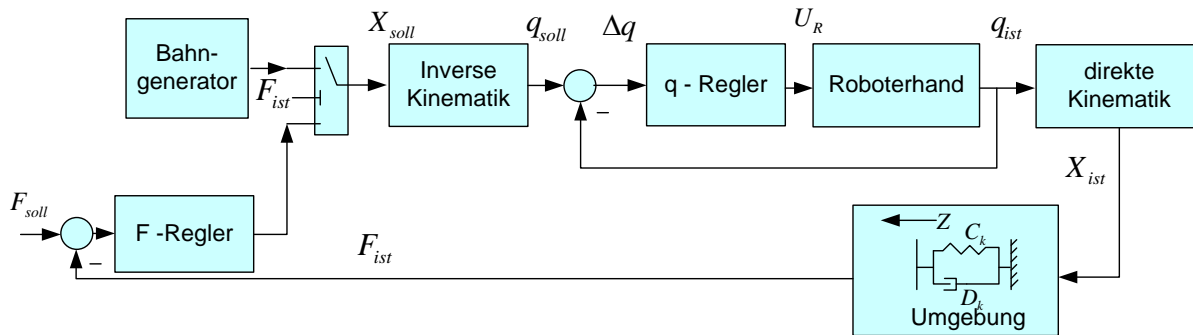


Abbildung 99. Das Schema von Positions - Kraftregelung des Roboterarms mit Levenberg-Marquardt Verfahren in inverser Kinematik

Die Regelungsqualität ist nur vom q-Regler abhängig, wenn es um Positionsregelung geht. Der Kraftregelkreis ist nur dann eingeschaltet, wenn sich der Roboterarm in Kontakt befindet. Unter dem q-Regler versteht man hier einen PD Regelkreis mit „Computed Torque Control“ (Abbildung 90).

Die Umgebung hat eine Ebene, die mit $0x + 0y + z + 0.3 = 0$ beschrieben wird.

Die Kontaktebene teilt den Raum in zwei Teile. Es wurde angenommen, dass bei $z > -0.3$ ein freier Raum ist. Aus der Ebenengleichung folgt, dass eine Kontaktkraft nur dann erzeugt wird, wenn $z < -0.3$ ist. Die Kontaktkraft kann nur im Arbeitspunkt des Greifers wirken (zuzüglich der Gleichung (33)). Andere mögliche Kontaktpunkte, die bei schlechter Bewegungsplanung entstehen können, wurden vernachlässigt. Bei der Simulation wurde der Roboterarm mit idealem Kraftsensor ausgerüstet. Die Y und X Achse werden mit Positionsregelkreis geregelt. Wenn die Kontaktkraft >0 ist, wurde die Z-Achse der Kraftregler eingesetzt. Andernfalls funktioniert der Positionsregler, der eine XYZ Bahnkurve verfolgt.

Das Kontaktmodell ist ein einfacher Punktkontakt. Die Kontaktkraft berechnet sich nach der Gleichung:

$$F = \begin{cases} C_k \Delta Z + D_k \Delta \dot{Z}, & \text{if } Z < -0.3 \\ \text{else } 0 \end{cases} \quad (33)$$

Wobei ΔZ - ist die Kontaktebenevertiefung, $\Delta \dot{Z}$ - entsprechende Geschwindigkeit der Vertiefung. Die Standardwerte für Kontaktparameter sind Steifigkeit $C_k=1000$ N/m und die Dämpfung $D_k=10$ N·s/m.

Wenn der Roboterarm sich im Kontakt befindet und sich entlang der Kontaktebene bewegt, wird auch eine Reibungskraft erzeugt. Der Reibkraftvektor liegt in der Kontaktebene und zeigt gegen die Bewegungsrichtung des Arbeitspunkts des Greifer. Die Reibkraft berechnet sich aus $F_{\text{reibung}} = V_{\text{greifer}} C_{\text{reibung}}$, wobei V_{Greifer} die Geschwindigkeit des Greifers in kartesischen Koordinaten, und $C_{\text{reibung}} = 0.1$ N·s/m der Reibungskoeffizient ist.

In Abbildung 100 a) ist der Zeitverlauf der Kraft dargestellt. Die Sollkraft ist gleich 30 N. Zwischen 0 bis 3.2 s bewegt sich der Greifer des Roboterarms entlang einer Bahnkurve um eine bestimmte Position in der Kontaktebene zu erreichen. Am Anfang sieht man eine kleine Positionsabweichung

100 c), die von den kinematische Begrenzungen und der Unvollkommenheit des q-Reglers Verursacht wird.

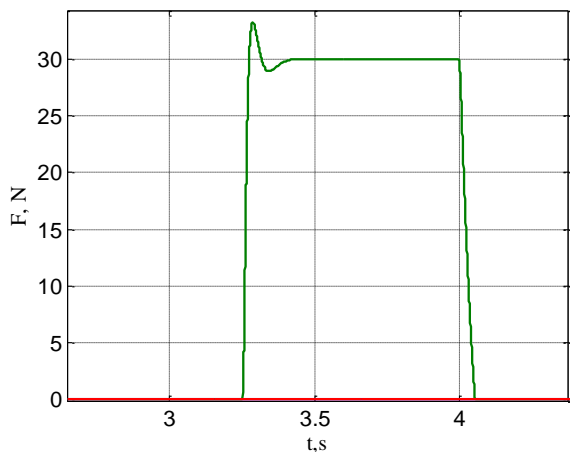
Zwischen 3.2 bis 4 s befindet sich der Arbeitspunkt des Greifers in Kontakt (Abbildung 100 c)) und die Position in der XY Ebene bleibt konstant. Der Positionsregelkreis hat einen statischen Fehler in der XY Ebene wegen der PD Regelung, die eine Positionsabweichung bedingt, wenn Störungen auftreten. Um den statischen Fehler zu vermeiden wird eine PID Regelung benötigt.

In Abbildung 100 b) sieht man ein sprungartiges Aussteigen der Lastwirkung, bei Erreichen der Kontaktebene. Der zweite Sprung nach dem Verlassen der Kontaktebene wird von der Positionsabweichung in der Z Achse zwischen Z_{soll} vom Kraftregelkreis und Z_{soll} vom Positionsregelkreis verursacht.

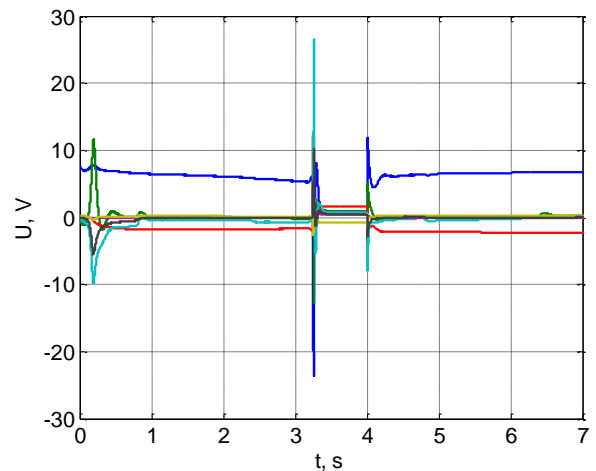
Als Kraftregler wurde ein PI Regler verwendet. Die Koeffizienten sind $K_i=10^{-2}$, $K_p=3 \cdot 10^{-3}$. Das Überschwingen ist gleich 10%, die Regelzeit ist 0.2 s.

Die Untersuchungen zeigen eine akzeptable Regelqualität in dem Kraftregelkreis (Abbildung 99). Die Regelgüte wird jedoch von der Lösung der inversen Kinematik beeinflusst, wie aus dem Vergleich des Systems mit Positionsregelung zu sehen ist (vgl. Abbildung 96).

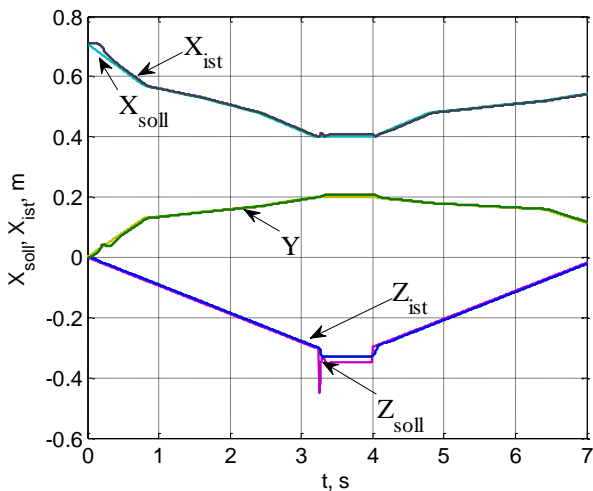
a)



b)



c)



d)

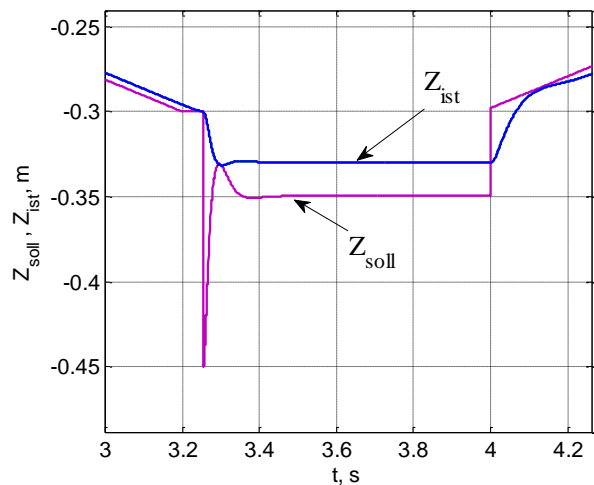


Abbildung 100. Kraftregelung beim Levenberg-Marquardt-Verfahren mit inverser Kinematik und PD-Regelung in kartesischen Koordinaten

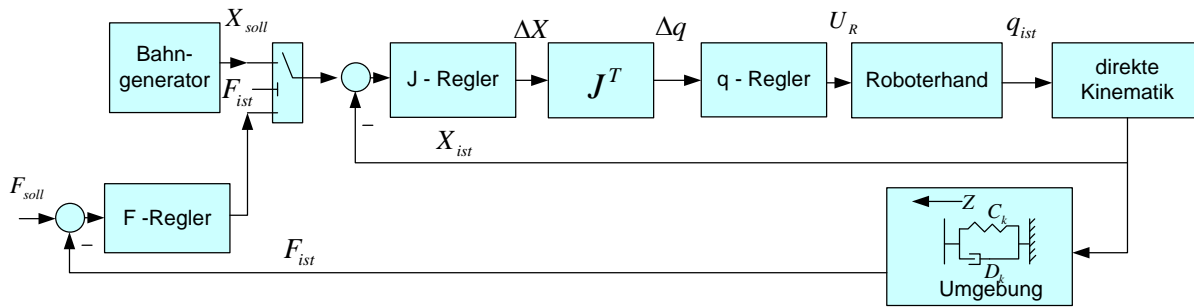
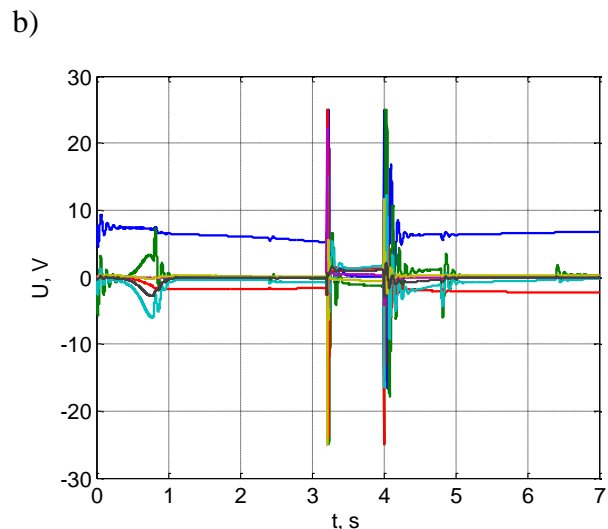
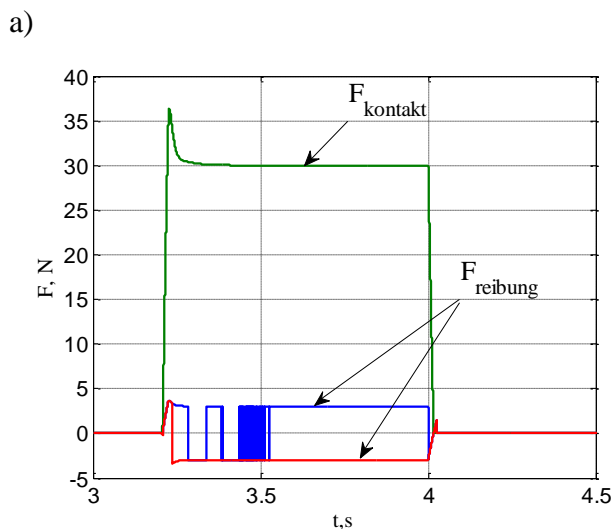


Abbildung 101. Das Schema der Positions-Kraftregelung des Roboterarms

Wenn der Positionsregelkreis in kartesische Koordinaten realisiert wird, erhält das Regelungssystem die in Abbildung 101 dargestellte Struktur. Hier ist die Jacobi-Matrix anstelle der inversen Kinematik verwendet worden. Die Verwendung des Levenberg-Marquardt-Verfahren ist bei dieser Struktur auch möglich, sofern für die Anfangsposition die aktuelle Armposition verwendet wird.

Die Jacobi-inverse Kinematik ist ein Einschrittalgorithmus, wobei sich der Optimierungsschritt mit $\Delta q_i = J^T \Delta X$ berechnet. Das Levenberg-Marquardt-Verfahren hat die Schrittgleichung: $\Delta q_i = (J^T J + \mu I)^{-1} J^T \Delta X$, welche mehrfach angewendet werden muss (ca. fünf- bis zehnmal), um eine gute Annäherung in einer Abtastzeit zu bekommen. Die zyklisch wiederholte Jacobi-inverse Kinematik ist dieselbe Optimierungsmethode wie das Gradientenabstiegsverfahren, welches nur die erste Ableitung des Gütekriteriums verwendet.

In Abbildung 102 sind die Prozesse in dem Regelungssystem mit der Jacobi-inversen Kinematik (Abbildung 101) für eine lineare Bewegung entlang einer Kontaktebene dargestellt. Abbildung 102 a) zeigt die Zeitverläufe der Kontaktkraft und der Reibungskräfte. In Abbildung 102 b) sieht man die Motorspannungen. Ein sprunghafter Zeitverlauf der Spannungen ist nur beim Herstellen des mechanischen Kontakts und bei der Kontakttrennung zu sehen. In Abbildung 102 c) sind die Bahnkurven in kartesischen Koordinaten dargestellt. Beim Systemstart sieht man eine Abweichung auf der X-Achse, welche von einer entarteten Jacobi-Matrix verursacht wird. Solch ein Problem besteht beim Levenberg-Marquardt-Verfahren nicht. Dennoch vermindert die Rückkopplung in kartesischen Koordinaten dynamische Positionsabweichungen, die bei der Bewegung entlang der Bahnkurve auftreten.



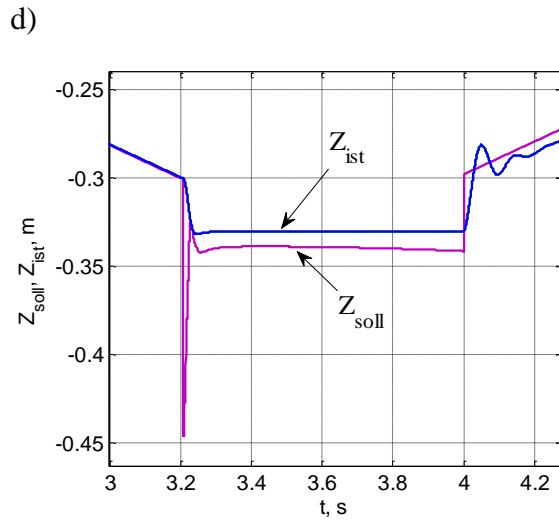
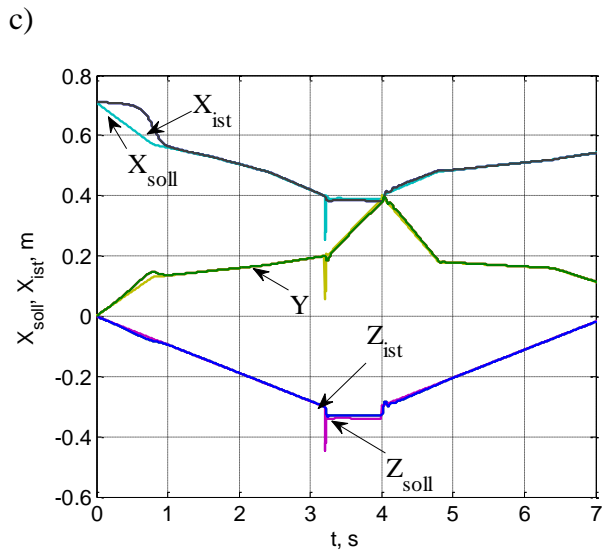


Abbildung 102. Kraftregelung bei jacobi-inverser Kinematik und PD-Regelung in kartesische Koordinaten; Lineare Bewegung bei der Kraftregelung entlang der Y-Achse

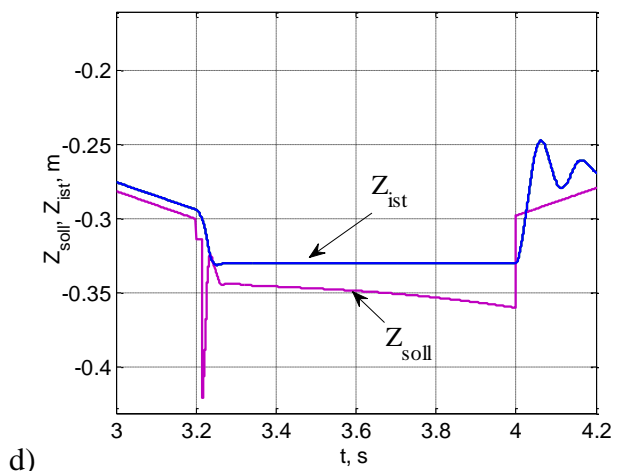
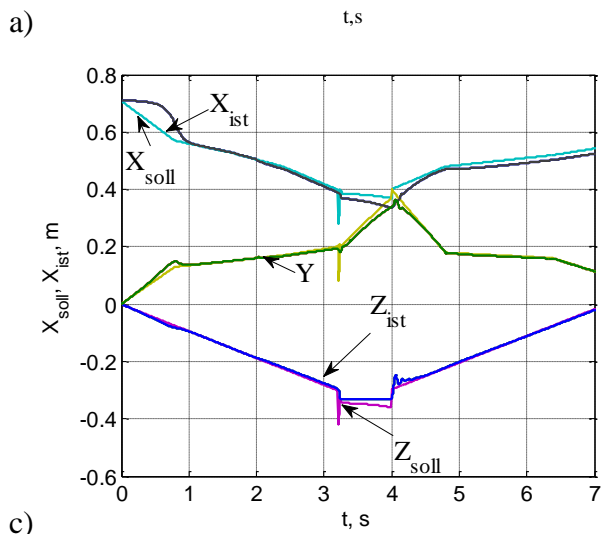
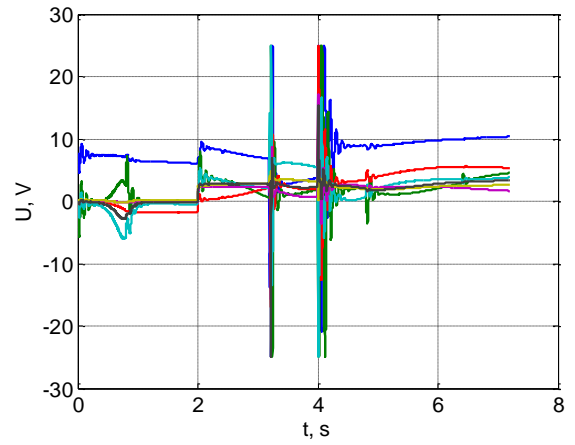
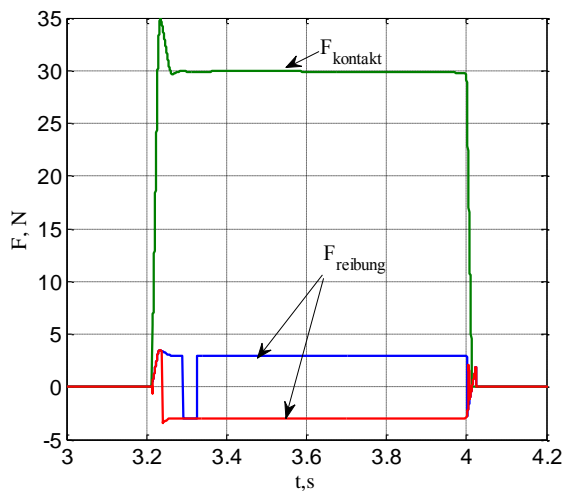


Abbildung 103. Kraftregelung bei jacobi-inverser Kinematik und PD-Regelung in kartesische Koordinaten; Lineare Bewegung bei der Kraftregelung entlang der Y-Achse; Hinzufügen eines Störmoments $M_c = 10 \text{ Nm}$ bei $t = 2 \text{ s}$

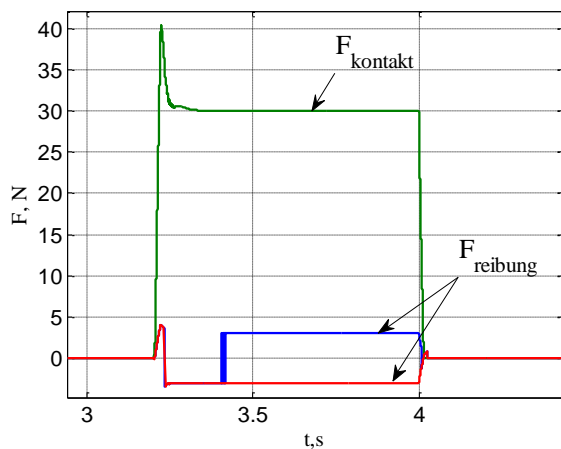
Der Unterschied von Abbildung 103 zur Abbildung 102 ist das Störmoment $M_c = 10 \text{ Nm}$, welches zum Zeitpunkt $t = 2 \text{ s}$ angelegt wurde und die Bahnabweichung des Roboterarms verursacht. Während die Störung auf den Kraftregelkreis keine große Auswirkung hat, bewirkt diese beim Ausgang der Kraftregelung im Positionsregelkreis besonders auf der X-Achse eine große Abweichung.

In Abbildung 104 sieht man den Prozessverlauf des positions-kraftgeregelten Systems mit jacobibasierter inversen Kinematik unter Einwirkung des gleichen Störmoments. Hierbei ist im inneren Positionsregelkreis die Linearisierung mittels Ersatzfunktion $y = k + u$ verwendet worden. Im Vergleich zum „Computed Torque Control“-Verfahren der vorangegangenen Simulationen sieht man kleinere Positionsabweichungen. Der Kraftregelkreis hat jedoch eine akzeptierbare Qualität. Die Überschwingweite ist doppelt so hoch und erreicht 33 % des Eingangssignals. Da die Überschwingweite von der Bahn des Roboterarms abhängt, kann diese durch „vorsichtigere“ Kontaktherstellung verringert werden.

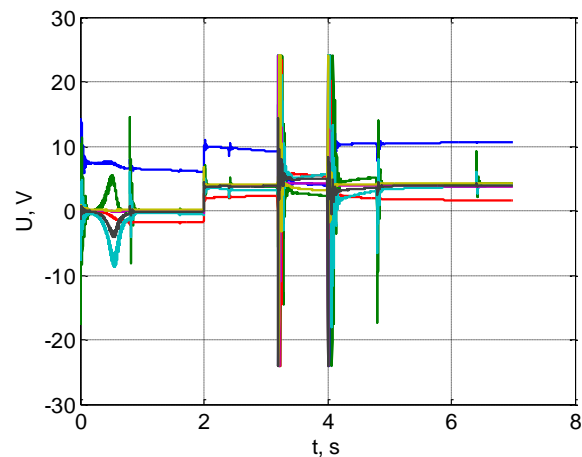
Die dynamischen Positionsabweichungen bei Verwendung der Linearisierung mittels Ersatzfunktionen zeigen in alle Bereiche eine verbesserte Regelungsqualität.

Betrachtet man die Einfachheit und Robustheit der Linearisierung mittels Ersatzfunktionen im Vergleich mit anderen Verfahren, ist dieses Verfahren noch überzeugender.

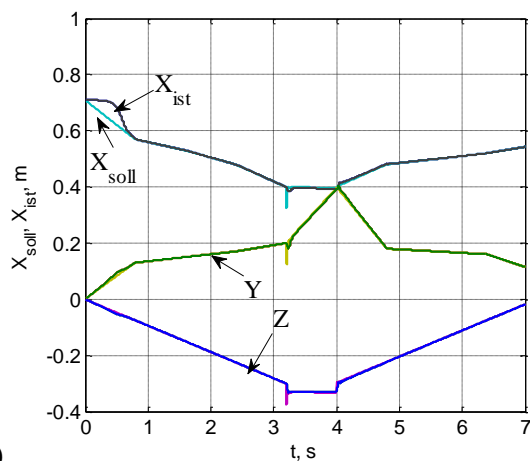
a)



b)



c)



d)

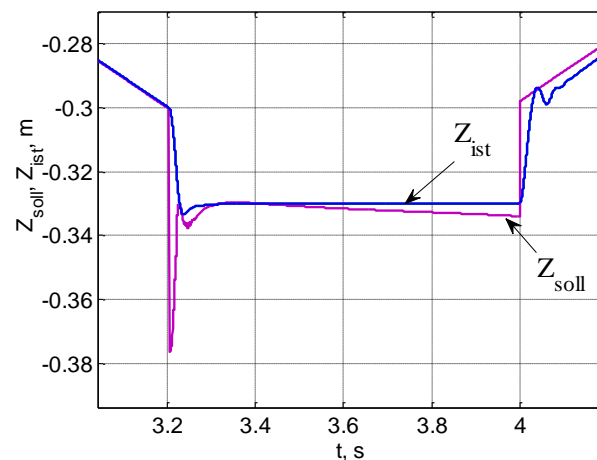


Abbildung 104. Kraftregelung bei jacobii-inverser Kinematik und PD-Regelung in kartesische Koordinaten; Lineare Bewegung bei der Kraftregelung entlang der Y-Achse; Linearisierung mittels Ersatzfunktion $y = k + u$; Hinzufügen eines Störmoments $M_c = 10 \text{ Nm}$ bei $t = 2 \text{ s}$

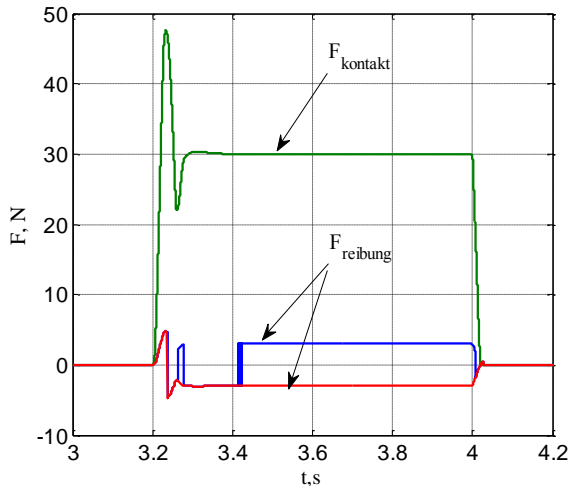
In Abbildung 105 sieht man den Vergleich der Linearisierung mittels Ersatzfunktion $y = k + u$ und dem „Computed Torque Control“-Verfahren im inneren Positionsregelkreis. Der Dämpfungskoeffizient wurde zu $D_k=3 \text{ N}\cdot\text{c}/\text{m}$ abgeändert.

Ersatzfunktion $y = k + u$

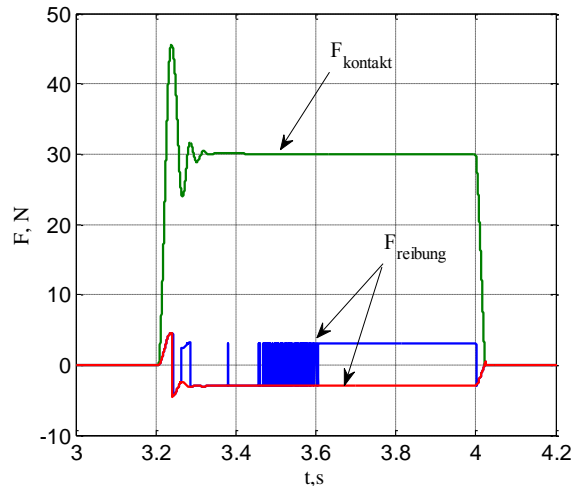
$D_k=3 \text{ N}\cdot\text{c}/\text{m}$, $C_k=1000 \text{ N}/\text{m}$

„Computed Torque Control“

$D_k=3 \text{ N}\cdot\text{c}/\text{m}$ $C_k=1000 \text{ N}/\text{m}$



$C_k=10000 \text{ N}/\text{m}$, $D_k=10 \text{ N}\cdot\text{c}/\text{m}$



$C_k=10000 \text{ N}/\text{m}$, $D_k=10 \text{ N}\cdot\text{c}/\text{m}$

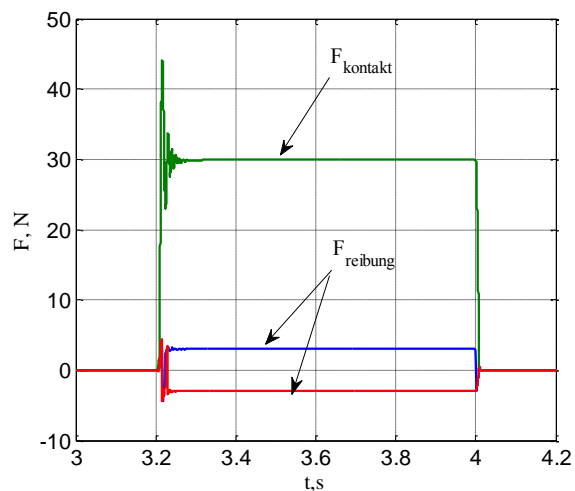
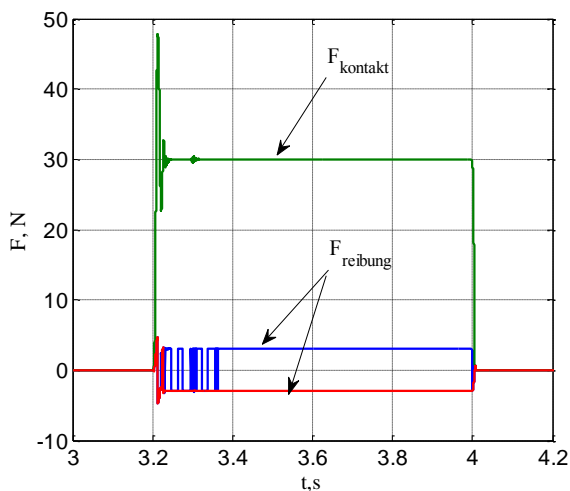


Abbildung 105. Vergleich der Kraftverhältnisse bei der Linearisierung mittels Ersatzfunktion $y = k + x$ und „Computed Torque Method“

Beide Verfahren liefern auch bei unterschiedlichen Kontaktparametern fast dieselben Ergebnisse. Dies zeigt erneut, dass das Ersatzfunktionsverfahren auch im inneren Positionsregelkreis verwendet werden kann. Die Qualität der Kraftregelung ist von der inversen Kinematik und dem Kraftregler abhängig, welche in beiden Simulationen gleich waren.

Kraftregelungssysteme unter Verwendung von Ersatzfunktionen im Kraftregelkreis bieten ausreichend Stoff für weitere Forschungsprojekte, was jedoch im Rahmen dieser Arbeit nicht weiter verfolgt wird. Hier wurde lediglich das einfachste Beispiel des Positions-Kraftregelungssystems mit einem einfachen Kontaktmodell dargestellt, welches ausreichend die Besonderheiten und Lösungsfähigkeit des vorgeschlagenen Linearisierungsverfahrens demonstriert.

Zusammenfassung

In diesem Kapitel werden unterschiedliche Handbauarten analysiert und verglichen. Als aussichtsreiche Variante wurde der „Cable Drive“-Mechanikaufbau ausgewählt. Dafür ist ein neues Getriebeprinzip, das so genannte „Twisted Wire Actuator“-Prinzip auf dem Versuchstand und in der Simulation detailliert untersucht und für den Einsatz in Roboterhänden vorgeschlagen worden.

Für die Fünffingerhand wurden unterschiedliche Greifbewegungen entworfen und modelliert. Diese Bewegungen hängen vom zu fassenden Objekten ab. Solche Verfahren verlangen einen komplexen Einsatz von Bildverarbeitungsalgorithmen.

Außerdem wurde die direkte und inverse Kinematik für Roboterhände untersucht. Für die Aufgabe der inversen Kinematik wurden iterative Optimierungsalgorithmen verwendet, sowie Gradientenabstiegsverfahren und die Levenberg-Marquardt-Methode, welche man auch für Echtzeitsysteme einsetzen kann. Ein spezieller Abschnitt ist der Konvexen Optimierung (mit Nebenbedingungen) der inversen Kinematikaufgabe gewidmet. Für den Arm als redundanten mechanischen Manipulator mit 7 Freiheitsgraden wurde ein neues Regelungskonzept mit Hilfe von Ersatzfunktionen vorgeschlagen. Dieses Verfahren ist mit „Computed Torque Control“ verglichen worden. Die Vorteile des neuen Verfahrens wurden anhand von Simulationen bestätigt. Die Regelung erfolgte in Antriebskoordinaten und in globalen kartesischen Koordinaten. Für die Roboterarm mit 7 Freiheitsgraden wurde eine Positions-Kraftregelung entwickelt. Unter Betrachtung verschiedener Regelungsverfahren und Kontaktbewegungen.

Zusammenfassung

Intelligente Regelungssysteme kommen immer häufiger für Lösungen von Steuerungsaufgaben in unterschiedlichen Gebieten der Technik zur Anwendung. Technische Progressse wie beispielsweise Robotersysteme, Magnetlevitation oder autonome Unterwasserfahrzeuge ziehen ständig neue Regelstrecken nach sich. Die mathematischen Modelle von solchen Systemen haben schwer ermittelbare Parameter. Dies führt zur Entwicklung von Regelungssystemen, die sich nicht auf ein Modellwissen stützen, sondern auf Messungen und Berechnungen.

Die effektivste Verfahrensgruppe von Regelungsalgorithmen ist die nichtlineare Regelung, die sich in viele Untergruppen einteilen lässt. Jeder nichtlinearer Regler hat eine vorgegebene Struktur. Die Struktur muss entweder so komplex sein wie die Regelstrecke selbst oder einfach, aber flexibel, um die Anpassungsfähigkeit zu erhöhen und gleichzeitig den Berechnungsaufwand zu mindern.

Solche komplexen Strukturen verwendet man bei Computed Torque Control Systemen, bei der Linearisierung mittels Lie-Ableitungen oder bei Regelungen mit einem Zustandsregler zusammen mit einem Beobachter. Im Falle einer flexiblen und einfachen Struktur hängt das Ergebnis größtenteils nicht von der Struktur selbst ab, sondern von dem Anpassungsgesetz. Am Beispiel von Magnetlagern mit unterschiedlichen nichtlinearitäten sowie einem und fünf Freiheitsgraden wurden mehrere Regelungssysteme entworfen und implementiert. Weiterhin wurden PI-, PID- und Zustandsregler mit adaptiven Neuro-Fuzzy-Reglern verglichen. Der adaptive Neuro-Fuzzy-Sliding-Mode-Regler bietet die Möglichkeit, das klassische Konzept eines Zweipunktreglers mit einem modernen Verfahren zu kombinieren und von den Vorteilen beider Methoden zu profitieren. Für den Vorschubsantrieb eines Unterwasserfahrzeuges wurde ein NFN-Regler mit Prognose entworfen. Dieser wurde nach dem „direct inverse control“ Konzept aufgebaut. Die NFN-Regelung mit Prognose ermöglicht, durch ein auf IIR-Filtern basierendes System, ein stabileres und genaueres Regelungsverhalten, obwohl dieses wegen des erheblichen Berechnungsaufwands für Echtzeitsysteme schwer zu realisieren ist. Das letzte System in Kapitel 3 beruht auf der Positionierung von Polstellen und liefert die besten Ergebnisse bei Echtzeitanwendungen.

Die Entwicklung einer Roboterarm umfasst den Entwurf der mechanischen Struktur sowie die Konstruktion und Entwicklung eines modernen, leichten, kompakten Getriebes und des Kraft-Position-Regelungssystems. Die Roboterhand besteht aus dem Roboterarm und der Roboterhand. In dem Abschnitt von der Schulter bis zum Ellbogengelenk lässt sich der Roboterarm als ein Manipulator mit sieben Freiheitsgraden darstellen. Dieser stellt eine Reihe von Standardaufgaben bestehend aus direkter und inverser Kinematik, der Modellbildung des elektromechanischen Systems, der Regelung im kartesischen Koordinatensystem und der Positionierung der Hand für Kontaktaufgaben dar.

Die Mechanik der Hand stellt hohe Anforderungen an die Gesamtmasse der Hand und deren Abmessungen. Die Verwendung eines klassischen Zahngetriebes macht die Hand entweder zu schwer oder zu teuer. Dank neuer im Rahmen dieser Arbeit einwickelten Getriebe mit verdrehten Seilen (Twisted Wire Aktuator) kann die Handkonstruktion sehr kompakte Abmaße und ein geringes Gewicht annehmen. Diese Getriebe erlauben es die elektrischen Antriebe für die Hand in das Ellbogengelenk auszulagern und senken dadurch erheblich die Handmasse. Die Seile und Laufrollen besitzen ein relativ kleines Gewicht, kleine Maße und einen niedrigen Preis. Auf diese Weise wurden hohe Anforderungen erfüllt. Für die Roboterhand wurden weiterhin Greifbewegungen entwickelt, die die Serviceoperationen des Systems ergänzen.

Die im Rahmen der Dissertationsarbeit einwickelte adaptive Regelung mit Ersatzfunktionen zeigt eine hohe Effizienz und ist für unterschiedliche nichtlineare Regelstrecken wie das Magnetlager und die Roboterhand anwendbar.

Ausblick

In der vorliegenden Dissertationsarbeit wurden Probleme zur Regelung von nichtlinearen Regelstrecken behandelt. Die adaptive Regelung mit Ersatzfunktionen hat sich als sehr effiziente und aussichtsreiche Methode herausgestellt, wenngleich diese nur mit einigen Ersatzfunktionen getestet und untersucht wurde. Das Verfahren ist auf mehrdimensionale Systeme erweiterbar, wobei gegenseitige Einwirkungen berechnet werden müssen. Die Auswahl der Ersatzfunktion beeinflusst die Eigenschaften des Verfahrens wie die Konvergenz, Robustheit und den benötigten Berechnungsaufwand. Die resultierenden Ergebnisse, die Systemarchitektur und das Verfahren können auch für andere Regelstrecken verwendet werden. Der Kern der Regelung mit Ersatzfunktionen ist ein Optimierungsalgorithmus, der ebenso auf anderen Gebieten verwendet werden kann, bei denen eine schnelle Optimierung benötigt wird.

Die Entwicklung der Roboterhand ist eine komplexe Aufgabe. In der vorliegenden Arbeit wird die Roboterhand als Teil eines Systems betrachtet. Die Roboterhand ist eine Komponente von einem mobilen Roboter und ist in die gesamte Konstruktion und Hardware zu integrieren. Dies setzt weitere Forschungsarbeiten Roboterentwicklung und -herstellung voraus.

Für mobile Systeme ist das Eigengewicht stets eine wichtige Größe. Je weniger Gewicht, desto geringer ist der Energiebedarf. Die Kosten der Herstellung sind die zweitwichtigste Größe. Kleinere Motoren und Getriebe stellen bei gleicher Leistung einen hohen Kostenfaktor dar. Das entwickelte Getriebe mit verdrehten Seilen eröffnet in diesem Sinne neue Perspektiven. In dieser Dissertationsarbeit wurden die theoretischen Grundlagen und mathematische Beschreibungen von solchen Systemen im Ganzen betrachtet. In der Praxis bleiben jedoch noch viele unerforschte Fragen in Bezug auf die Zuverlässigkeit und Herstellung solcher Getriebe offen. Für die Untersuchungen in dieser Arbeit wurde das Seil aus „Dynema“ verwendet. Die Anwendung von unterschiedlichen elastischen Materialien und geometrischen Formen des verdrehten Materials wie Bänder, Seile und Fäden bieten weiteren Forschungsstoff. Das sich verkürzende Material des Mechanismus führt zur Entwicklung von unterschiedlichen Getrieben der linearen oder rotierenden Bewegung. In der Dissertationsarbeit wurden aus der Vielzahl von Mechanismen einige solcher Getriebe untersucht.

Die Bewegungsplanung der Roboterhand kann mit Hilfe von Bildaufnahmen erfolgen. Dies zieht zahlreiche Aufgaben der Bildverarbeitung und der Mustererkennung nach sich. Die gleichzeitige Verwendung von zwei Kameras erlaubt es eine so genannte Tiefenkarte zu berechnen, die den Abstand zwischen der Kameraebene und den beobachteten Objekten darstellt. Die Roboterhand hat kooperativ mit anderen Teilen des Roboters sowie mit der anderen Hand zu arbeiten. Deswegen erfordert die Entwicklung der Roboterhand neuste Ergebnisse und Erkenntnisse auf den Gebieten der Mechanik, Elektrotechnik, Robotertechnik, künstlichen Intelligenz, Bildverarbeitung und der Regelungstechnik.

Literaturverzeichnis

Lirteratur für Abstract

- [1] Antonelli G.: Underwater Robots, Motion and Force Control of Vehicle-Manipulator Systems, Springer-Verlag Berlin Heidelberg 2006, 283p.
- [2] Bauman's Moscow State Technical University Research Institute of Special Machinery Department of Special Machinery and ROVs Übersicht: Unterwasserfahrzeuge: <http://aqua.sm.bmstu.ru/> .
- [3] Antonelli G., Chiaverini S., Sarkar N., West M.: *Adaptive Control of an Autonomous Underwater Vehicle: Experimental Results on ODIN*, IEEE transactions on control systems technology, vol. 9, no. 5, September 2001.
- [4] Volker T.: Missionsbezogener modellgestützter Entwurf mobiler automatischer Systeme am Beispiel eines autonomen Unterwasserfahrzeugs, Dissertation Fakultät für Informatik und Automatisierung der Technische Universität Ilmenau, 2005.
- [5] Ageyev M. D.: *The Simplified Procedure of Calculation of Propellers for AUV*. Underwater Robots and Their System - Vladivostok: Dal'nauka, 1999, pp.: 33 - 49. (RUS)
- [6] Filaretov V. F., Alekseev Ju. K., Lebedev A. V. Sistemy upravlenija podvodnymi robotami. – M.: “Kruglyj god”, 2000. - 286 s.(RUS)
- [7] Feijun Song; Smith, S.M.: *Design of sliding mode fuzzy controllers for an autonomous underwater vehicle without system model*. OCEANS 2000 MTS/IEEE Conference and Exhibition Volume 2, Issue, 2000 Page (s):835 - 840 vol.2
- [8] Kim, T.W.; Yuh, J. A novel neuro-fuzzy controller for autonomous underwater vehicles Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on Volume 3, Issue, 2001 Page (s): 2350 - 2355 vol.3
- [9] Hyun-Sik Kim and Yong-Ku Shin: *Design of Adaptive Fuzzy Sliding Mode Controller based on Fuzzy Basis Function Expansion for UFV Depth Control*. International Journal of Control, Automation, and Systems, vol. 3, no. 2, pp. 217-224, June 2005.
- [10] Al-Muthairi N. DAS FLUOR., Zribi M.: *Sliding mode control of a magnetic levitation system*. Mathematical Problems in Engineering Volume 2004 (2004), Issue 2, pp. 93-107.
- [11] Masoud Mohammadian: *Computational Intelligence for Modelling, Control* UND Automation, IOS Press, US (Oktober 1999), 474 p.
- [12] Christofides P. D., Nael El-Farra: Control of Nonlinear and Hybrid Process Systems. Designs for Uncertainty, Constraints and Time-Delays. Springer, Berlin; (September 2005), 455p.
- [13] Takagi T., Maugin G. A.: Proceedings of the Eleventh International Symposium on Applied Electromagnetics and Mechanics: 19, IOS Press, US (Juni 2004) 701p.
- [14] T. Meressi and M.C. Kao: *Modelling and Control of a Magnetic Levitation System*. Control and Applications, CA 2006.
- [15] R.J. Schilling: *Fundamentals of robotics, Analysis and control*. Prentice Hall, 1996, 464 p.
- [16] Chiacchio P. Chiaverini S.: *Complex Robotic Systems*. Springer Verlag Berlin 1998, 184 p.
- [17] Frank L.Lewis Darren M.Dawson Chaouki T.Abdallah: *Robot Manipulator Control Theory and Practice*, 2004 by Marcel Dekker, 607p.
- [18] R.V. Patel DAS FLUOR. Shadpey: *Control of Redundant Robot Manipulators, Theory and Experiments*. Springer-Verlag Berlin Heidelberg 2005, 205 p.

- [19] Kozłowski K.: *Robot Motion and Control, Recent Developments*. Springer-Verlag London Limited 2006, 397 p.
- [20] Inventor/Applicant: SHOHAM, Moshe [IL/IL]: *Twisting Wire Actuator*, INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY, International Publication Number WO 2005/041211 A2.

Lirteratur für Kapitel 1

- [1] Jeffrey T. Spooner, Manfredi Maggiore, Raúl Ordóñez, and Kevin M. Passino. *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques*, John Wiley and Sons, NY, 2002.
- [2] Jay A. Farrell, Marios M. Polycarpou Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches, Wiley Interscience, 2006, 440 p.
- [3] Castillo, Oscar, Melin, Patricia. *Soft Computing for Control of Non-Linear Dynamical Systems Series: Studies in Fuzziness and Soft Computing*, Vol. 63 2001, XIV, 224 p.
- [4] Günther Nürnberger: *"Approximation by Spline Functions"*, Springer Verlag, 1989.
- [5] Jagannathan Sarangapani. *Neural network control of nonlinear discrete-time systems*. CRC Press, 2006 602 p.
- [6] M. Powell. *Radial basis functions for multivariable interpolation: A review*. In J. Mason and M. Cox, editors, *Algorithms for Approximation of Functions and Data*, pages 143-167. Oxford University, Oxford, UK, 1987.
- [7] Rosenblatt, Frank: *The perception: a probabilistic model for information storage and organization in the brain*. *Psychological Reviews* 65 (1958) 386-408 p.
- [8] Burkhard Lenze. *Einführung in die Mathematik neuronaler Netze*. Logos-Verlag, Berlin, 2003.
- [9] Ingrid Gerdes, Frank Klawonn, Rudolf Kruse. *Neuro-fuzzy-systeme: Von den Grundlagen künstlicher neuronaler Netze zur Kopplung mit Fuzzy-systemen*. Vieweg Teubner Verlag, 2003 434 p.
- [10] Leszek Rutkowski. *Flexible neuro-fuzzy systems: structures, learning, and performance evaluation*. Springer, 2004. 279 p.
- [11] Marco Russo, L. C. Jain. *Fuzzy learning and applications*. CRC Press, 2001. 391 p.
- [12] Abraham Kandel, Gideon Langholz. *Fuzzy control systems*. CRC Press, 1994 624 p.
- [13] Hung T. Nguyen, Michio Sugeno. *Fuzzy systems: modeling and control*. Springer, 1998 519 p.
- [14] Rolf Isermann. *Mechatronische Systeme: Grundlagen*. Springer, 2007. 623 p.
- [15] Kai Michels, Frank Klawonn, Rudolf Kruse, Andreas Nürnberger. *Fuzzy-regelung: Grundlagen, Entwurf, Analyse*. Springer, 2002. 420 p.
- [16] János Abonyi. *Fuzzy model identification for control*. Springer, 2003. 273 p.
- [17] Frank L. Lewis, J. Campos, R. Selmic. *Neuro-fuzzy control of industrial systems with actuator nonlinearities*. SIAM, 2002. 244 p.
- [18] L. Zadeh. *A fuzzy-set-theoretic interpretation of linguistic hedges* *Cybernetics and systems* [0196-9722] yr: 1972 vol: 2 iss: 3 pages: 4.
- [19] E. H. Mamdani and S. Assilian, *An experiment in linguistic synthesis with a fuzzy logic controller*, *Int. J. Man-Machine Studies* 7(1) (1975), pp 1-13.
- [20] Sugeno, M. *An introductory survey of fuzzy control*. *Information Sciences*, Volume 36, 1985, pp. 59–83.
- [21] Wang L.-X. *Fuzzy systems are universal Approximators*. *Fuzzy Systems IEEE International Conference on Volume, Issue, 8-12 Mar 1992 Page(s):1163 – 1170*.

- [22] James J. Buckley, Yoichi Hayashi. *Fuzzy input-output controllers are universal Approximators*. Fuzzy Sets and Systems archive. Volume 58, Issue 3. Elsevier North-Holland, Inc. (September 1993), 273 - 278 pp.
- [23] Detlef Nauck, F. Klawonn, Rudolf Kruse. *Foundations of neuro-fuzzy systems*. John Wiley, 1997. 305 p.
- [24] Abdelhafid Zeghib, Frank Palis, Fathi Ben-Ouezdou, *EMG-based Finger Movement Classification Using Transparent Fuzzy System*. EUSFLAT - LFA 2005 , Soft Computing for Signal Processing pp. 816-821.
- [25] Nauck, D., Klawonn, F., Kruse, R.: *Neuronale Netze und Fuzzy-Systeme*. Friedr. Vieweg and Sohn Verlag, Braunschweig-Wiesbaden, 1994.
- [26] Jyh Shing Roger Jang. *ANFIS : Adaptive Network-Based Fuzzy Inference System*. IEEE transactions on systems, man, and cybernetics, vol. 23, no. 3, 1993 665-683pp.
- [27] Huaguang Zhang, Derong Liu. *Fuzzy modeling and fuzzy control*. Springer, 2006. 416 p.
- [28] L. C. Jain, Beatrice Lazzerini, Halici Ugur. *Innovations in ART neural networks*. Springer, 2000, 258p.
- [29] Tom Tollenaere. *SuperSAB: fast adaptive back propagation with good scaling properties*. Neural Networks archive, Volume 3 , Issue 5 (1990) pp.561 – 573.
- [30] Radoslaw Pytlak. *Conjugate Gradient Algorithms in Nonconvex Optimization*. Springer, 2008, 477 p.
- [31] Enrique Castillo. *A Very Fast Learning Method for Neural Networks Based on Sensitivity Analysis*. Journal of Machine Learning Research 7 (2006) pp. 1159–1182.
- [32] Simon S. Haykin, Bernard Widrow. *Least-mean-square adaptive filters*. Wiley-IEEE, 2003. 494 p.
- [33] Niendiek, Steffen. *Eine universelle Repräsentation von Fuzzy-Controllern durch Neuronale Netze*. Diplomarbeit am Institut für Mathematik und Informatik der Westfälischen Wilhelms-Universität Münster, 1998.
- [34] Y. D. Landau. *Adaptive control: the Model Reference Approach*. Marcel Dekker, New York, 1979
- [35] R. Kelly, R. Ortega. *Adaptive motion control design of robot manipulators: an input-output approach*. Int. J. Control, vol. 50, no. 6, pp. 2563-2581, 1989.
- [36] G. Kreisselmeier, K.S. Narendra. *Stable model reference adaptive control in the presence of bounded disturbances*. IEEE Trans. Automatic control, vol. 27. no. 6, pp. 1169-1175, 1982.
- [37] Tsakalis, K.S. *The σ -modification in the adaptive control of linear time-varying plants*. Decision and Control, 1992., Proceedings of the 31st IEEE Conference on Volume , Issue , 1992 Page(s):694 - 698 vol.1
- [38] H.A. Hassan; M.P.R.V. Rao. *Novel e-Modification Robust Adaptive Control Scheme Using Non-Quadratic Lyapunov Functions for Higher Order Systems*. Computer as a Tool, 2005. EUROCON 2005. The International Conference on Vol. 1, Issue, 2005 Page(s):290 – 293.
- [39] J. T. Spooner, M. Maggiore, R. Ordóñez, K. M. Passino. *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques*. John Wiley and Sons, NY, 2002
- [40] Nelder, A.; Mead, R. *A simplex method for function minimization*. The Computer Journal 7, 308–313 ,1965.
- [41] R. Hooke and T. A. Jeeves. *Direct search solution of numerical and statistical problems*. J.Assn. Comp. Mack, 8, 212-229, 1961
- [42] Hamdy A. Taha. *Operations Research: An Introduction*. 8th ed., Prentice Hall, 2007
- [43] J. A. Nelder and R. Mead, *A simplex method for function minimisation*. The Comp Journal, 7, 308-313, 1965
- [44] Zhigljavsky, A. A. *Theory of Global Random Search*. Dordrecht, Netherlands: Kluwer, 1991

- [45] Karsten Weicker. *Evolutionäre Algorithmen*. Vieweg+Teubner Verlag, 2007, 313 S.
- [46] Anne Wolter. Reinforcement Learning in der Roboter-navigation: Grundlagen, Methoden, Realisierung und Experimente. VDM Verlag, 2008. 144 S.
- [47] Richard S. Sutton, Andrew G. Barto. *Reinforcement learning: an introduction*. MIT Press, 1998. 322 p.
- [48] C. Audet. *Convergence results for pattern search algorithms are tight*. Optimization and Engineering, 5(2):101–122, 2004
- [49] C. Audet and J. E. Dennis, Jr. *Analysis of generalized pattern searches*. SIAM Journal on Optimization, 13(3):889–903, 2003.
- [50] Lewis, R.M., Torczon, V. *Pattern search algorithms for linearly constrained minimization*. SIAM Journal on Optimization 10(3), 917–941 (2000)
- [51] Iman, R.L.; Davenport, J.M. ; Zeigler, D.K. (1980). Latin hypercube sampling (program user's guide)
- [52] Stephen P. Boyd, Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. 716 p.
- [53] Christian Borgelt, Frank Klawonn . *Neuro-fuzzy-systeme: Von den Grundlagen künstlicher neuronaler Netze zur Kopplung mit Fuzzy-systemen*. Vieweg+Teubner Verlag, 2003. 434 S.
- [54] Farshad Khorrami, Prashanth Krishnamurthy, Hemant Melkote. *Modeling and adaptive nonlinear control of electric motors*. Springer, 2003. 523 p.
- [55] G. Conte, C. H. Moog, A. M. Perdon. *Algebraic methods for nonlinear control systems*. Springer, 2007. 178 p.
- [56] Lesław Socha. *Linearization methods for stochastic dynamic systems*. Springer, 2008. 383p
- [57] Frank L. Lewis, D. M. Dawson, Chaouki T. Abdallah. *Robot manipulator control: theory and practice*. CRC Press, 2004. 614 p.
- [58] Kelly R., Santibanez V., Loria A., “*Control of Robot Manipulators in Joint Space*”, Springer-Verlag London Limited, 2005

Lirteratur für Kapitel 2

- [1] Akira Chiba, Tadashi Fukao, Osamu Ichikawa, “*Magnetic bearings and bearingless drives*”, Cima Verlag, 400 S.
- [2] John R. Brauer, “*Magnetic Actuators and Sensors*”, Wiley-IEEE Press ,2006, 328 S.
- [3] Gerhard Schweitzer, Eric H. Maslen, “*Magnetic Bearings: Theory, Design, and Application to Rotating Machinery*”, 2009, 535 S.
- [4] Richard M. Crowder, “*Electric drives and electromechanical systems*”, published by Butterworth-Heinemann-Elsevier, 2006, 292 pp.
- [5] Joseba Quevedo Casín, Vincenç Puig Cayuela, “*Identification and control: the gap between theory and practice*”, Springer Verlag, 2007, 330 pp.
- [6] Jean Lévine, “*Analysis and Control of Nonlinear Systems: A Flatness-based Approach*”, Springer Verlag, 2009, 319 pp.
- [7] Gerhard Schweitzer, Eric H. Maslen, “*Magnetic Bearings Theory, Design, and Application to Rotating Machinery*”, Springer-Verlag Berlin Heidelberg 2009, 358 S.
- [8] Petzold O., “*Hybridmagnete für einen magnetisch gelagerten Rundtisch*“, TECHNISCHE MECHANIK, Band 26, Heft 2, (2006), 85– 91 S.
- [9] Föllinger, O., “*Regelungstechnik*“, 8. Aufl. Hüthig Verlag, Heidelberg. 1994.
- [10] Stefan Palis, Modellbildung, Entwurf und experimentelle Erprobung einer nichtlinearen Regelung für den Magnetlagerversuchsstand MA400. Diplomarbeit. Universität Otto von Guericke. 2006.

- [11] Farrell J., Polycarpou M., „*Adaptive Approximation Based Control*“, Wiley-Interscience, 2006.
- [12] Zoran Vukic, Ljubomir Kuljaca, Dali Donlagic, Sejid Tesnjak, “*Nonlinear control Systems*”, by Marcel Dekker, Inc, 2003.
- [13] Christian Borgelt, “*Neuro-Fuzzy-Systeme: von den Grundlagen kuenstlicher Neuronaler Netze zur Kopplung mit Fuzzy-Systemen*“, Wiesbaden: Vieweg, 2003.
- [14] X. Yu, Z. Man, B. Wu, “*Design of fuzzy sliding mode control systems*”, Fuzzy Sets and Systems 95 (3) (1998)295–306.
- [15] Diantong Liu , Jianqiang Yi, Dongbin Zhao , Wei Wang, “*Adaptive sliding mode fuzzy control for a two-dimensional overhead crane*”, Mechatronics 15 (2005) 505–522 pp.
- [16] Q. P. Ha, Q. H. Nguyen, D. C. Rye, and H. F. “*Durrant-Whyte. Fuzzy Sliding-Mode Controllers with Applications*”, IEEE, Transactions on industrial electronics, Vol 48, NO 1, 2001, 38-46pp.
- [17] Najib Essounbouli and Abdelaziz Hamzaoui ,“*Direct and Indirect Robust Adaptive Fuzzy Controllers for a Class of Nonlinear Systems*”, International Journal of Control, Automation, and Systems, vol. 4, no. 2, pp. 146-154, April 2006
- [18] Schallschmidt, T. Draganov, D. Palis, F. „*Adaptive fuzzy control of magnetically suspended rotary table*“, Power Electronics and Motion Control Conference, 2008. EPE-PEMC 2008. 13th ,1610 – 1613pp.
- [19] Kelly R., Santibanez V., Loria A., “*Control of Robot Manipulators in Joint Space*”, Springer-Verlag London Limited, 2005.
- [20] António E. Ruano, Institution of Electrical Engineers, “*Intelligent control systems using computational intelligence techniques*” Institution of Engineering and Technology; illustrated edition edition 2005, 667p.
- [21] Adamy, Jürgen, „*Nichtlineare Regelungen*“, Springer Verlag 2009, 382 S.

Lirteratur für Kapitel 3

- [1] Filaretov V. F. Alekseev Y.K, Lebedev A.V.:*Control System of Underwater robots*. Moskau, Krugliy god, 2001, 288pp.(RUS)
- [2] Gwyn Griffiths: Technology and applications of autonomous underwater vehicles. Taylor & Francis Group2003, 369 pp.
- [3] Geoff Roberts und Robert Sutton: *Advances in Unmanned Marine Vehicles*. Institution of Engineering and Technology, 2005, 441pp
- [4] Filimonov N. A.: About the applicability of neural circuits control based on direct inverse learning models. Mechatronic, Automation&Control. № 10, 2004. – p. 54 – 60.(RUS)
- [5] Fjellstad O. E., Fossen T. I., Egeland O.: *Adaptive Control of ROVs with Actuator Dynamics and Saturation*. Proc. of the 2nd International Offshore and Polar Engineering Conference, San Francisco, USA. 1992, pp. 513 – 519.
- [6] Whitcom L. L., Yoerger D. R.: *Preliminary Experiments in the Model-Based Dynamic Control of Marine Thrusters*. Proc. of the IEEE International Conference on Robotics and Automation. 1996, pp. 467 – 472.
- [7] Fossen T.I., Blanke M.: Nonlinear Output Feedback Control of Underwater Vehicle Propellers Using Feedback Form Estimated Axial Flow Velocity. IEEE Journal of Oceanic Engineering. V. 25, № 2, 2000, pp. 241-255.

- [8] Ageev M.D.: A simplified method of calculation of propeller for the AUV, Underwater robots and their systems. Pod Society. Ed. MD Ageev. - Vladivostok: Dalnauka, 1995. 33 – 49pp.
- [9] Rutkovskaja D., Plynskiy M, Rudkovskiy L.: *Neural network, genetic algorithms and fuzzy systems*. Moskau, Gorjachaja linia 2004. - 452 pp. (RUS)
- [10] Egupov N.D.: *Methods of robust, Neuro - Fuzzy and Adaptive Control*. Moskau, Bauman Moscow Higher Technical School, 2002. – 744 pp.(RUS)
- [11] Borgelt Ch.: *Neuro-Fuzzy-Systeme: von den Grundlagen kuenstlicher Neuronaler Netze zur Kopplung mit Fuzzy-Systemen*. Wiesbaden: Vieweg, 2003, 434 S.
- [12] Nelles O.: *Nonlinear system identification with local linear neuro-fuzzy models*. Aachen: Shaker, 1999, 179 p.
- [13] Timothy J. Ross: *Fuzzy logic with engineering applications*. Wiley, 2004, 628 p.
- [14] Liu G. P.: *Nonlinear identification and control: a neural network approach*. London. Springer, 2001, 210 p.
- [15] Vas P.: *Artificial-intelligence-based electrical machines and drives: application of fuzzy, neural, fuzzy-neural and genetic-algorithm-based techniques*. Oxford Univ. Press, 1999, 625 p.
- [16] Jeffrey T. Spooner: *Stable adaptive control and estimation for nonlinear systems : neural and fuzzy approximator techniques*. New York: Wiley-Interscience, 2002, 545 p.
- [17] Denai M. A., Attia S. A.: *Fuzzy and neural control of an induction motor*. Appl. Math. Comput. Sci. 2002. , Vol. 12, No 2, pp. 221-233.
- [18] Kim H., Fok S., Fregene K., Lee D.-H., Oh T.-S., Wang D.: *Neural Network-Based System Identification and Controller Synthesis for an Industrial Sewing Machine*. Int. Journal of Control, Automation, and Systems Vol. 2, No. 1, 2004, pp. 83-91.
- [19] Kumar G., Harley R. G.: *Two Separate Continually Online-Trained Neurocontrollers for Excitation and Turbine Control of a Turbogenerator*. IEEE Trans. in industry applications, Vol. 38, No.3, 2002, pp. 887-893.
- [20] Haykin, S.: *Adaptive Filter Theory*. Third Edition. - Prentice-Hall, Inc., 1996.
- [21] Rossiter, J.A.: *Model Based predictive control: a practical approach*. - CRC press LLC, 2003.
- [22] Goodwin Graham C., Graebe Stefan F., Saldago Mario E.: *Control System Design*. - Prince Hall, 2001.

Lirteratur für Kapitel 4

- [1] M. Taddei: *Leonardo da Vinci's robots*. Leonardo3, Milano, 2007.
- [2] E.J.Kreuzer, H.G.Meißner, J.-B. Lugtenburg und A.Truckenbrodt: *Industrieroboter*. Springer-Verlag Berlin Heidelberg, 1994.
- [3] <http://www.kuka-robotics.com>
- [4] <http://world.honda.com/ASIMO/>
- [5] <http://www.toyota.co.jp/en/special/robot/>
- [6] <http://www.festo.com/>
- [7] <http://www.dlr.de/>
- [8] S. Schulz, C. Pylatiuk and G. Bretthauer: *A New Ultralight Anthropomorphic Hand*. Proceedings of the 2001 IEEE International Conference on Robotics & Automation Seoul, Corea, May 2001.
- [9] S. Beck, A. Lehmann, T. Lotz, J. Martin, R. Keppler, R. Mikut: *Modellgestützte adaptive Regelungskonzepte für eine fluidisch betriebene Roboterhand*. Proc., GMA-Kongress 2003, VDI-Berichte 1756, S. 65-72; 2003.
- [10] M. Zecca, G. Cappiello, F. Sebastiani, S. Roccella, F. Vecchi, M. C. Carrozza, P. Dario: *Experimental analysis of the proprioceptive and exteroceptive sensors of an underactuated*

- prosthetic hand*. International Journal of Human-friendly Welfare Robotic Systems, Vol 4, No. 4, 2003.
- [11] <http://www.shadowrobot.com>
- [12] Roboterhände von Andreas Vogel: <http://tams-www.informatik.uni-hamburg.de/lehre/2005ss/proseminar/robotik/pdf/2005-06-13-Text.pdf>
- [13] Wenwei Yu, Daisuke Nishikawa, Yasuhiro Ishikawa, Hiroshi Yokoi, and Yukinori Kakazu, “Multi-functional electrical prosthetic hand —Development of tendon-driven mechanism and controller”, Journal of Robotics & Mechatronics, Vol. 14, No.6, pp. 324-331, (2002)
- [14] F. Lotti, P. Tiezzi and G. Vassura: *Development of UB Hand 3: Early Results*, Robotics and Automation, ICRA 2005. Proceedings of the 2005 IEEE International Conference.
- [15] <http://www.hs-robots.com/fotogalerie.htm>
- [16] http://carlpisaturo.com/Slave_Hand.html
- [17] Kwang J. Kim and Satoshi Tadokoro: *Electroactive Polymers for Robotic Applications, Artificial Muscles and Sensors*. Springer-Verlag London Limited 2007. 287 p.
- [18] M. Suzuki: *Complex and Flexible Robot Motions by Strand-Muscle Actuators*. Climbing & Walking Robots, Towards New Applications, 2007.
- [19] Inventor/Applicant: SHOHAM, Moshe [IL/IL]; Hoshaya, M.P. Hamovil, 17915 Hoshaya (IL): *Twisting Wire Actuator*, INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY, International Publication Number WO 2005/041211 A2, .
- [20] V. Filaretov, Y. Zavgorodniy, Y. Tsepkovskiy: Mechanisms for linear and rotation movements based on nonstretchable line”, Problems of mechanical engineering and reliability of machines №11, 2009, Russia. (RUS)
- [21] Timoshenko S.: *Strength of Materials*. D. Van Nostrand company, Inc. New York. 1940. - 450 p.
- [22] L.W. Tsai: *Robot Analysis: the mechanics of serial and parallel manipulators*. John Wiley& Sons, Inc. New York, 1999, 515p.
- [23] Max Bohner, Richard Fischer, Rolf Gscheidle: *Fachkunde Kraftfahrzeugtechnik*. 27.Auflage, Verlag Europa-Lehrmittel, Haan-Gruiten, 2001.
- [24] G.S. Monkman, S.Hesse, R.Steinmann, H.Schunk: *Robot Grippers*, WILEY-VCH Verlag GmbH & Co KGaA, Weinheim, 2007.
- [25] G. Schreiber, Dissertation: Steuerung für redundante Robotersysteme: Benutzer - und aufgabenorientierte Verwendung der Redundanz. DLR Institut für Robotik und Mechatronik, 2004, 161 S.
- [26] Lewis F., Dawson D., Abdallah C.: *Robot Manipulator Control: Theory and Practice*. Marcel Dekker New York, 2004.
- [27] Kelly R., Santibanez V., Loria A.: *Control of Robot Manipulators in Joint Space*. Springer-Verlag London Limited, 2005.
- [28] Chiacchio P., Chiaverini S.: *Complex Robotic Systems* , Springer-Verlag London Limited, 1998.
- [29] L.Birglen, T. Laliberté, C. Gosselin: *Underactuated Robotic Hands*. Springer-Verlag Berlin Heidelberg, 2008, 242 p.
- [30] <http://www.touchbionics.com>
- [31] H. Unbehauen: *Regelungstechnik III* , Braunschweig, 1988, 456 S.
- [32] Y. Tsepkovskiy, L. Antonov, F. Palis: *Dynamic Compensation of Nonlinear Dynamics Research of Mechanic Anthropomorphic Robotic Hand*. Proceedings of the 10th WSEAS International Conference on AUTOMATION & INFORMATION (ICAI'09), 70-77 pp.

- [33] Guo Yi-shen, Chen Li: *Adaptive neural network control for coordinated motion of a dual-arm space robot system with uncertain parameters*. Applied Mathematics and Mechanics. Springer Verlag Netherlands Heft. Volume 29, Number 9 / September 2008 S: 1131-1140.
- [34] Chang-Woo Park, Young-Ouk Kim, and Chong Ho Yi: *Robust and Adaptive Fuzzy Feedback Linearization Regulator Design*. Advanced Intelligent Computing Theories and Applications with Aspects of Artificial Intelligence. Verlag Springer Berlin - Heidelberg 2008, S: 476-483.
- [35] F. Payam: An Adaptive Input-Output Feedback Linearization Controller for Doubly-Fed Induction Machine Drives. Serbian journal of electrical engineering Vol. 5, No. 1, May 2008, 139-154 pp.
- [36] Hua Deng, Han-Xiong Li, Senior Member, IEEE, and Yi-Hu Wu.: *Feedback-Linearization-Based Neural Adaptive Control for Unknown Nonaffine Nonlinear Discrete-Time Systems*. IEEE transactions on neural networks, vol. 19, no. 9, September 2008 1615 – 1625pp.
- [37] Mohamed Azlan Hussain, Pei Yee Ho, and J. C. Allwright: *Adaptive Linearizing Control with Neural-Network-Based Hybrid Models*, Ind. Eng. Chem. Res., 2001, 5604-5620 pp.
- [38] Zoran Vukic, Ljubomir Kuljaca, Dali Donlagic, Sejid Tesnjak: *Nonlinear control Systems*. By Marcel Dekker, Inc, 2003.
- [39] Tarn T.-J., Chen S.-B., Zhou C.: *Robotic Welding, Intelligence and Automation*. Springer-Verlag Berlin Heidelberg, 2004
- [40] Farrell J., Polycarpou M.: *Adaptive Approximation Based Control*. Wiley-Interscience, 2006.
- [41] Bompos N., Artemiadis P., Oikonomopoulos A., Kyriakopoulos K.: *Modelling, Full Identification and Control of the Mitsubishi PA-10 Robot Arm*. IEEE International Conference of Advanced Intelligent Mechatronics, 4-7 September, 2007.
- [42] Ott C.: *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*. Springer-Verlag Berlin Heidelberg, 2008.
- [43] Chiacchio P., Chiaverini S.: *Complex Robotic Systems*. Springer-Verlag London Limited, 1998.
- [44] Ghosh B., Xi N., Tarn T.-J.: *Control in Robotics and Automation: Sensor-Based Integration*. Academic Press, 1999.
- [45] Essounbouli N., Hamzaoui A.: *Direct and Indirect Robust Adaptive Fuzzy Controllers for a Class of Nonlinear Systems*. International Journal of Control, Automation and Systems, Vol. 4, № 2, April 2006, pp.146-154.
- [46] Wong C.-C., Huang B.-C., Lai H.-R.. *Genetic-based Sliding Mode Fuzzy Controller Design*. Tamkang Journal of Science and Engineering, Vol. 4, № 3, 2001, pp.165-172.
- [47] Kozłowski K.: *Robot Motion and Control*. Springer-Verlag London Limited, 2006.
- [48] Khare R., Sapra A.: *Robotics. Appin Knowledge Solutions*. Infinity Science Press, 2007.

Anhang A

Die Parametern des DC Motors

- P_{max} = 20 W – die maximale Leistung;
 U_n = 1,21 V – die Nennspannung;
 I_n = 1,21 A – der Nennstrom;
 ω_{max} = 9550 rad/s – die maximale Drehgeschwindigkeit ohne Last;
 I_{max} = 10,4 A - der maximale Strom;
 R = 2.32 Ω – der Motorwiderstand;
 L = $24 \cdot 10^{-3}$ H – die Induktivität;
 C_ω = $\frac{407}{2 \cdot \pi / 60} 10^{-3}$ V · s / rad - die Geschwindigkeitskonstante;
 C_m = $23,4 \cdot 10^{-3}$ N · m / A - die Momentkonstante;
 J_m = $10,3 \cdot 10^{-7}$ kg · m² - das Trägheitsmoment des Rotors;
Gear = 250 - der Übertragungsfaktor des Getriebes;

Tabelle 13 Fingerparameter

Parametern	Gelenk 1	Gelenk 2	Gelenk 3
der Scharnierradius, m	0.005	0.005	0.005
Länge, m	0.03	0.03	0.03
Masse, kg	0.02	0.02	0.02
die Trägheitsmomentmatrix, $kg \cdot m^2$	$\begin{bmatrix} \frac{1}{12}ml^2 & 0 & 0 \\ 0 & \frac{1}{12}ml^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{12}ml^2 & 0 & 0 \\ 0 & \frac{1}{12}ml^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{12}ml^2 & 0 & 0 \\ 0 & \frac{1}{12}ml^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Anfangswinkel, rad	0	$-\pi/2$	$-\pi/2$
Hard Stop Parameter			
Obere Grenze, rad	$\pi/2$	$\pi/2$	$\pi/2$
Untere Grenze, rad	0	0	0
Kontaktsteifigkeit auf der oberen Grenze, Nm/rad	1e+04	1e+04	1e+04
Kontaktsteifigkeit auf der unteren Grenze, Nm/rad	1e+04	1e+04	1e+04
Kontaktdämpfung auf der oberen Grenze, $N \cdot m/(rad/s)$	0.7	0.7	0.7
Kontaktdämpfung auf der unteren Grenze, $N \cdot m/(rad/s)$	0.7	0.7	0.7

Tabelle 14 Bewegungsparametern der Fingermechanik

Springparameter			
Offset, rad	0	$\pi/2$	$\pi/2$
Steifigkeit	5	5	5
Dämpfung	0	0	0
Reibungsparameter			
statische Reibung, $N \cdot m$	0.01	0.01	0.01
kinetische Reibung, $N \cdot m$	0.01	0.01	0.01
Koeffizient viskoser Reibung, $m \cdot s \cdot N/rad$	0.01	0.01	0.01
Drehgeschwindigkeits threshold, rad/s	0.01	0.01	0.01
Aproximationkoeffizient, s/rad	0.01	0.01	0.01

Tabelle 15 Parameter der Bestandteile einer siebengliedrigen Roboterhand

	Masse	Ausmaße	Trägheitsmoment	Rotationsachse
Kd Schulter	$m_1 = 0,01kg$	$r_1 = 0,01$ m	$\begin{bmatrix} \frac{2}{5}m_1r_1^2, 0,0 \\ 0, \frac{2}{5}m_1r_1^2, 0 \\ 0,0, \frac{2}{5}m_1r_1^2 \end{bmatrix}$	z
Schulterdreher	$m_2 = 0,01kg$	$r_2 = 0,01$ m	$\begin{bmatrix} \frac{2}{5}m_2r_2^2, 0,0 \\ 0, \frac{2}{5}m_2r_2^2, 0 \\ 0,0, \frac{2}{5}m_2r_2^2 \end{bmatrix}$	y
Schulter	$m_3 = 0,5kg$	$r_3 = 0,03$ m, $l_3 = 0,3$ m	$\begin{bmatrix} \frac{1}{2}m_3r_3^2, 0,0 \\ 0, \frac{1}{12}m_3 \cdot (3r_3^2 + l_3^2), 0 \\ 0,0, \frac{1}{12}m_3 \cdot (3r_3^2 + l_3^2) \end{bmatrix}$	x
Ellbogendreher	$m_4 = 0,01kg$	$r_4 = 0,01$ m	$\begin{bmatrix} \frac{2}{5}m_4r_4^2, 0,0 \\ 0, \frac{2}{5}m_4r_4^2, 0 \\ 0,0, \frac{2}{5}m_4r_4^2 \end{bmatrix}$	y
Ellbogen	$m_5 = 0,5kg$	$l_5 = 0,23$ m	$\begin{bmatrix} 0,0,0 \\ 0, \frac{1}{3}m_5l_5^2, 0 \\ 0,0, \frac{1}{3}m_5l_5^2 \end{bmatrix}$	x
Handdreher	$m_6 = 0,01kg$	$r_6 = 0,01$ m	$\begin{bmatrix} \frac{2}{5}m_6r_6^2, 0,0 \\ 0, \frac{2}{5}m_6r_6^2, 0 \\ 0,0, \frac{2}{5}m_6r_6^2 \end{bmatrix}$	z
Hand	$m_7 = 0,3kg$	$l_7 = 0,1$ m	$\begin{bmatrix} \frac{1}{3}m_7l_7^2, 0,0 \\ 0,0,0 \\ 0,0, \frac{1}{3}m_7l_7^2 \end{bmatrix}$	y