

Universität der Bundeswehr München
Fakultät für Luft- und Raumfahrttechnik
Institut für Systemdynamik und Flugmechanik

Identifizierung nichtlinearer aerodynamischer Derivative mit einem Modularen Neuronalen Netzwerk

Dipl.-Ing. Jost Seifert

Dissertation

Vollständiger Abdruck der von der Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktoringenieurs
(Dr.-Ing.)

genehmigten Dissertation

Vorsitzender: Prof. Dr.-Ing. Ferdinand Svaricek

1. Berichterstatter: Prof. Dr.-Ing. Reiner Onken

2. Berichterstatter: Prof. Dr.-Ing. Dr.-Ing.habil. Otto Wagner

Die Dissertation wurde am 05.05.2003 bei der Universität der Bundeswehr München D-85577 Neubiberg, Werner-Heisenberg-Weg 39 eingereicht, und durch die Fakultät für Luft- und Raumfahrttechnik angenommen.

Tag der Promotion: 06.11.2003

Diese Arbeit entstand bei der EADS Deutschland GmbH
Geschäftsbereich Militärflugzeuge
Flugsimulation

Identifizierung nichtlinearer aerodynamischer Derivative mit einem Modularen Neuronalen Netzwerk

Übersicht: Die Modellierung der Aerodynamik eines Flugzeuges führt auf ein nichtlineares System. Neben der Beschreibung der physikalischen Zusammenhänge durch eine Modellstruktur stellt die Bestimmung der aerodynamischen Parameter die größte Ungewissheit dar. In mehreren Schritten werden diese Parameter durch Windkanalmessungen und numerische Verfahren ermittelt. Die letzten Ungenauigkeiten können mit Hilfe von Testflügen eines Prototypen und einem Verfahren zur Parameteridentifikation (PID) ausgeräumt werden.

Diese PID- Aufgabe wird hier durch ein Modulares Neuronales Netzwerk gelöst. Für jeden aerodynamischen Parameter wird ein Modul eingesetzt. Durch den Einsatz von Eingabeverbindungen ist es nun möglich, nichtlineare Derivative zu identifizieren.

Ein Neuronales Netz bietet einige Vorteile gegenüber vergleichbaren Ansätzen. So kann es zum Beispiel in einen Echtzeit- Flugsimulator integriert werden. Durch einen objektiven Vergleich von Flugtestdaten und Simulationsergebnissen werden die Auswirkungen der aerodynamischen Optimierung aus flugmechanischer Sicht beurteilt.

Identification of nonlinear aerodynamic derivatives with a Modular Neural Network

Abstract: Modelling an aircraft's aerodynamics leads to a nonlinear system. The depiction of physical relations is a complex inaccuracy task, however the determination of aerodynamic parameters is even more complex. It takes several steps to obtain these parameters from wind tunnel tests and numeric methods. Remaining discrepancies can be reduced by flight test and parameter identification methods (PID).

In this paper, this PID task is solved with a Modular Neural Network. Each aerodynamic parameter is represented by one module. Using so called input-weights allows the identification of nonlinear derivatives.

A Neural Network offers some advantages towards comparable methods. For example, it is possible to integrate an adaptive model into a realtime flight simulator. The effects of optimized aerodynamic parameters on flight mechanics are assessed by objective comparison of flight test data and simulation results.

Vorwort

Als ich neulich im Cockpit eines Segelflugzeugs saß und den Blick nach draußen richtete, bewunderte ich die realistische dreidimensionale Darstellung der Landschaft. Die fein detaillierten Objekte bewegten sich in Abhängigkeit meiner Flugbewegungen und wurden teilweise von anderen verdeckt. Das Licht warf Schatten, Wolken veränderten ihre Formen. Auf den Straßen konnte ich animierte Fahrzeuge erkennen; "Nicht schlecht gemacht", dachte ich. Am Steuerknüppel spürte ich eine leichte Kraft in meiner Hand und in einer Kurve wurde ich leicht in den Sitz gepreßt. Der Kompaß drehte sich entsprechend und zeigte mir die Himmelsrichtung an. Die anderen Instrumente halfen mir, die optimale Fluggeschwindigkeit einzuhalten. Neben den Windgeräuschen war auch noch das Piepsen des Variometers zu vernehmen. Alles war sehr realistisch. – Natürlich, denn unwillkürlich hatte ich die Realität mit den Gütekriterien moderner Trainingssimulatoren bewertet.

Dieses Erlebnis zeigt zum Einen, wie weit die Technik heute ist, und daß dadurch die Grenzen zwischen Realität und Illusion verschwimmen. Aber es zeigt auch, wie breit das Spektrum der Disziplinen ist. Nur wenn in allen nötigen Bereichen der Ingenieurwissenschaften (Audio-, Video - Technik, Ergonomie, Haptik, Mechanik) eine originalgetreue Wiedergabe des zu simulierenden Gerätes geschaffen wird, dann kann man von einer High Fidelity Simulation sprechen.

In den letzten sechs Jahren meiner Tätigkeit in der Flugsimulation habe ich immer wieder erfahren, wie schon Kleinigkeiten diese Illusion störten. Falsche Farben oder Symbologie in den Flugführungsdisplays, falsche Ergonomie bei Knüppel oder Pedalen reichten schon dafür aus. Nicht tolerierbar sind jedoch ruckelnde Außensichtanlagen oder ein falsches Flugverhalten.

Wenn auch oft Kritik am Simulator berechtigt ist, so argumentiert man im Falle eines scheinbar unechten Flugverhaltens besonders schwer. Denn hier wird zunächst subjektiv beurteilt. Ein objektiver Vergleich zwischen Flugzeug und Simulation schafft dann erst Klarheit.

Dies war für mich die Motivation, tiefer in die Technik der Parameteridentifikation und der Möglichkeiten der objektiven Nachweisführung einzusteigen. Ich entwickelte im Rahmen eines Technologieprojektes der Firma EADS Deutschland ein neues Verfahren zur Identifikation aerodynamischer Parameter, und verifizierte meine Ergebnisse mit der Technik der Parallelsimulation. Die Ergebnisse meiner Arbeit werden nun in dieser Dissertation vorgestellt.

Mit Rücksicht auf die Firmeninteressen der EADS können vertrauliche Informationen nicht veröffentlicht werden. Bei Ergebnissen aus dem Flugversuch oder der Flugsimulation werden die dargestellten Daten unkenntlich gemacht. Dies schadet weder dem Verständnis der Arbeit, noch der Glaubwürdigkeit. Bei der Beschriftung der Graphiken wurden die Zahlenwerte auf der Ordinatenachse weggelassen. Die dargestellten Graphiken sind also qualitativ zu bewerten.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Otto Wagner für seine anregenden Ideen und seinem Blick für das Wesentliche. Herrn Prof. Dr.-Ing. Reiner Onken danke ich ebenfalls für wertvolle Hinweise, die meine Arbeit zu einer runden Sache werden ließ. Dipl.- Informatiker Udo v. Garrel danke ich für seine Tipps zu RBF Netzen. Insbesondere möchte ich Sven Kuhlmann danken, der mich zu Beginn meiner Arbeit davor bewahrte, bei der Softwareentwicklung auf das falsche Pferd zu setzen. Meinen Vorgesetzten Dipl.-Ing. Ralph Michael Fleischer und Dipl.-Ing. Claus Frey danke ich dafür, daß sie sich für die Finanzierung und Durchführung meines Technologieprojektes eingesetzt haben. Großer Dank gebührt auch Dr. Peter Hunkel, der mich mit seinem Interesse an meiner Arbeit stets motiviert hat. Dipl.-Ing. Erfried Muthsam danke ich für die Bereitstellung von Flugmeßdaten. Für seine kritischen Fragen und Einblicke in die Technik der Parameteridentifikation danke ich Dr. Hans- Christoph Oelker. Außerdem möchte ich Dr.-Ing. Frank Thielecke und den Teilnehmern seines Arbeitskreises Softcomputing für die interessante und anregende Diskussion danken. Ich möchte es auch nicht versäumen, allen Kollegen und Freunden zu danken, die mir, wahrscheinlich ohne es zu wissen, durch Ihre Fragen geholfen haben, diese Materie verständlicher zu formulieren.

Besonders herzlich danke ich meiner Freundin Kerstin für ihr interessiertes Begleiten meiner Arbeit und ihre Geduld beim Entstehen der Dissertation.

Jost Seifert
Schliersee, im Frühling 2003

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	VI
Symbolverzeichnis	VII
1 Einführung	1
1.1 Realität und Simulation	1
1.2 Problemstellung	3
1.3 Künstliches Neuronales Netz	5
1.4 Ziel und Aufbau der Arbeit	6
1.5 Begriffsdefinitionen	11
2 Methoden der Parameteridentifikation	13
2.1 Anforderungen an eine Methode zur Parameteridentifikation	13
2.2 Klassische Parameteridentifikation	14
2.2.1 Übersicht	14
2.2.2 Ausgangsfehlerverfahren	16
2.2.3 Gleichungsfehlerverfahren	17
2.3 Alternative Methoden der Parameteridentifikation	17
2.3.1 Vorwärtsgerichtetes mehrschichtiges Netz (<i>MLP</i>)	18
2.3.2 Rekurrentes Neuronales Netz (<i>RNN</i>)	20
2.3.3 Neuro-Fuzzy System	20
2.3.4 Dimensionshomogenes Neuronales Netz	22
2.3.5 Neuronales Netz mit Radialsymmetrischer Basisfunktion (<i>RBF</i>)	23
2.3.6 Polynomansatz	24
2.3.7 Lokalmodell-Netz (<i>LMN</i>)	24
2.3.8 Modulares Neuronales Netz (<i>MNN</i>)	26
2.3.9 Genetische Algorithmen (<i>GA</i>)	26
2.4 Die Anwendung eines MNN mit Eingabeverbindungen	28
3 Flugversuche zur Parameteridentifikation	30
3.1 Spezialmanöver	30
3.2 Sensoren	32

4	Modellierung mit einem Modularen Neuronalem Netz	33
4.1	Die Schritte vom linearisierten Modell zum High Fidelity Modell	33
4.2	Neuronales Netz	34
4.2.1	Komponenten	34
4.2.1.1	Neuronen	34
4.2.1.2	Verbindungen	35
4.2.1.3	Schichten	35
4.2.1.4	Gruppen	36
4.2.2	Neuronales Netz strukturieren	37
4.2.2.1	Lineare und Quadratische Einflußparameter	37
4.2.2.2	Nichtlineare Einflußparameter	38
4.2.2.3	Vorgabe durch die physikalischen Zusammenhänge	39
4.2.2.4	Modellkorrektur	41
5	Parameteridentifikation mit einem Modularen Neuronalem Netz	42
5.1	Berechnung der aerodyn. Kraft- und Momentenbeiwerte aus Flugversuchsdaten	42
5.1.1	Koordinatensysteme	42
5.1.2	Auftriebsbeiwert C_A	43
5.1.3	Nickmomentenbeiwert C_m	46
5.1.3.1	Nichtrekursives differenzierendes Tiefpaßfilter	50
5.1.4	Zusammenhang zwischen C_A und C_m	51
5.2	Lerndatensätze	52
5.2.1	Lerndatensatz aus einem Referenzmodell	52
5.2.2	Lerndatensatz aus Flugversuchsdaten	52
5.2.3	Differenz- Lerndatensatz	53
5.3	Initialisierung	54
5.4	Optimierungsalgorithmus Backpropagation	55
5.4.1	Charakteristik	55
5.4.2	Algorithmus	56
5.4.3	Erforderliche Modifikation	58
5.4.4	Beispiel	59
5.4.5	Rechenzeit	61
5.4.6	Analysemöglichkeit des Modularen Neuronalem Netzes	62
5.5	Validierung des Verfahrens	62
5.6	Informationsgehalt des Lerndatensatzes	68
5.7	Einschränkungen	69
5.8	Hinweise aus der Praxis	70
5.8.1	Unpassender Lerndatensatz	70
5.8.2	Zu große Lernrate	70
5.8.3	Falscher Wertebereich der Gruppen	70
5.8.4	Numerische Instabilität	71
5.8.5	Dynamische Lernrate η_{dyn}	71
5.8.6	Einzelne Gruppe initialisieren	72

6	Analyse der aerodynamischen Derivative aus Flugversuchsdaten	73
6.1	Das aerodynamische Modell	73
6.2	Die Struktur des Neuronalen Netzes	74
6.3	Ergebnisse der Optimierung unter Verwendung eines Referenzmodells	76
6.4	Ergebnisse der Modelloptimierung ohne Referenzdaten	83
7	Objektive Beurteilung der Flugsimulation	87
7.1	Simulationstechnik	87
7.1.1	Trimmen	88
7.1.2	Manöver	89
7.1.3	Versuchsdurchführung	89
7.2	Charakteristische flugmechanische Bewegungsgrößen	89
7.3	Fehlerquellen in der Flugsimulation	90
7.3.1	Beispiel für eine Fehldiagnose	91
7.3.2	Bessere Ergebnisse durch Nachführen des Rollwinkels mit Hilfe eines Proportionalreglers	92
7.4	Analyse der Simulationsergebnisse	93
7.4.1	Toleranzen und ihre Anwendbarkeit	93
7.4.2	Analyse des Trimmzustands allgemein	93
7.4.3	Analyse von Manövern allgemein	94
7.4.4	Beispiele	94
7.4.4.1	Manöver 1: Wind Up Turn	95
7.4.4.2	Manöver 2: Landeklappen ausfahren	99
7.5	Flugsimulation mit einem Neuronalen Netz	103
7.5.1	Softwaremodelle	103
7.5.2	Anforderungen an das Neuronale Modell	104
7.5.3	Online Adaption	104
8	Zusammenfassung und Ausblick	106
	Literaturverzeichnis	109
	Abkürzungsverzeichnis	114
	Index	115

Abbildungsverzeichnis

1.1	PC-Simulator	1
1.2	Virtual Reality	2
1.3	Beispielhafte Ergebnisse unterschiedlicher Identifikationsverfahren	6
1.4	Begriffsdefinition der Modelle	12
2.1	Entwicklungsstufen der Verfahren zur Parameteridentifikation beim DLR	15
2.2	Ausgangsfehlerverfahren	17
2.3	Unterschiedliche Ergebnisse der Funktionsapproximation	19
2.4	Rekurrentes Neuronales Netzwerk mit verdeckten Neuronen	20
2.5	Neuro-Fuzzy Netzwerk	21
2.6	Strukturierung des Dimensionshomogenen Netzwerks nach Analyse der physikalischen Zusammenhänge	22
2.7	Neuronales Netzwerk mit einer Schicht von radialsymmetrischen Basisfunktionen	23
2.8	Prinzip eines Lokalmodell-Netzwerks	25
2.9	Blockdiagramm eines Modularen Netzwerks	26
2.10	Schema eines Iterationsschritts des GA zur Bildung von zwei neuen Individuen einer Kinderpopulation.	27
3.1	Zeitverläufe eines Pitch Doublets	31
3.2	Stationärer Kurvenflug	32
4.1	Graph der Funktion Tangens Hyperbolicus	34
4.2	Graph der Gauß Funktion	35
4.3	Analogie zwischen einer Gruppe und einem einzelnen Neuron	36
4.4	Netzwerkstruktur entsprechend dem obigen Modell	37
4.5	Gute Approximation mit Derivativen der 1. und 2. Ordnung	38
4.6	Schlechte Approximation mit Derivativen der 1. und 2. Ordnung	38
4.7	Netzwerkstruktur der C_A Gruppe	39
4.8	Ausgabe der C_A Gruppe	39
4.9	Hybrides Modell	40
5.1	Kräfte und Momente in der Längsbewegung	44
5.2	Hebelarme der Triebwerkskräfte	48

5.3	Drehrate und differenzierte Drehbeschleunigung während eines Fullstick Manövers	51
5.4	Initialisierung der Gewichte wird durch isoliertes Training der Gruppen ermöglicht	54
5.5	Verknüpfung zwischen Neuronen zweier Schichten	56
5.6	Das Neuron j in der verdeckten Schicht	57
5.7	Das Modell $C_A = C_{A\alpha} \cdot \alpha + C_{A\eta} \cdot \eta$	59
5.8	Anfangszustand	60
5.9	Nach dem Training des ersten Patterns	60
5.10	Endergebnis	61
5.11	Mit simuliertem Lerndatensatz optimiertes MNN	64
5.12	Bestmögliches Ergebnis der Optimierung	66
5.13	Prozentuale Anteile am Gesamtauftrieb	67
5.14	Vergleich vor und nach der Optimierung	67
5.15	Anstellwinkel - Machzahl Bereich	69
5.16	Dynamische Lernrate abhängig vom Netzwerkfehler (Summed Square Error, SSE)	71
6.1	Modellstruktur für C_A	74
6.2	Modellstruktur für C_m	75
6.3	Basismodelle für C_A und C_m , initialisiert mit einem Referenzmodell, dargestellt für einen Klappenwinkel $\delta_f = 5^\circ$	76
6.4	Zeitverlauf des Auftriebsbeiwerts vor der Optimierung	77
6.5	Optimierter Auftriebsbeiwert	78
6.6	Korrelation zwischen Spoilerausschlag und Schiebewinkel	78
6.7	Nickmomentenbeiwert vor der Optimierung	79
6.8	Optimierter Nickmomentenbeiwert	79
6.9	Typische Lernkurve und Fehler der Testmenge	80
6.10	Optimierte Modelle nach 20 Epochen	81
6.11	Optimierte Modelle nach 20 Epochen, $Ma = 0.5$	82
6.12	Mit Zufallswerten initialisiertes Neuronales Netz im Vergleich zum Basismodell	83
6.13	Generiertes Modell (nach 5000 Epochen)	84
6.14	Generiertes Modell, $Ma = 0.5$	85
7.1	Nachsimulieren aufgezeichneter Manöver	88
7.2	Beeinflussung der Simulationsergebnisse durch die Kopplung von Längs- und Seitenbewegung	91
7.3	Besseres Simulationsergebnis durch Nachführung des Rollwinkels	92
7.4	WUT mit Referenzdatensatz	96
7.5	WUT mit optimiertem Modell	97
7.6	WUT mit neu generiertem Modell	98
7.7	Landeklappen ausfahren mit Referenzmodell	100
7.8	Landeklappen ausfahren mit optimiertem Modell	101
7.9	Landeklappen ausfahren mit neu generiertem Modell	102

Tabellenverzeichnis

5.1	Lerndatensatz	59
5.2	Korrelationsmatrix	69
7.1	Toleranzen für dynamische Manöver (<i>short period</i>) gemäß JAR	93

Symbolverzeichnis

Flugmechanische Formelzeichen wurden, soweit nicht anders angegeben, der DIN 9300 [1] entnommen.

Skalare Größen in griechischen Buchstaben

α	Anstellwinkel
β	Schiebewinkel
δ_f	Ausschlag der Hinterkantenklappe
δ_s	Ausschlag des Spoilers
η	Höhenruderausschlag
η_{dyn}	Dynamische Lernrate
η_{min}	Kleinste Lernrate
ϵ	Fehler
κ	Einbauwinkel der Schubdüse um die z_f -Achse
ω_f	Vektor der Winkelgeschwindigkeiten
Φ	Hängewinkel
ρ	Luftdichte
σ	Einbauwinkel der Schubdüse um die y_f -Achse

Skalare Größen in lateinischen Buchstaben

C_A	Auftriebsbeiwert
C_W	Widerstandsbeiwert
C_m	Nickmomentenbeiwert
C_{A0}	Auftriebsbeiwert, für $q, \dot{\alpha}, \eta$ und $\delta_s = 0$
C_{m0}	Nickmomentenbeiwert, für $q, \dot{\alpha}, \eta$ und $\delta_s = 0$
E	Netzwerkfehler
F_x	Triebwerksschub in Richtung der Flugzeuglängsachse
F_y	Triebwerksschub in Richtung der Flugzeugquerachse
F_z	Triebwerksschub in Richtung der Flugzeughochachse
I_x	Trägheitsmoment um die Flugzeuglängsachse
I_y	Trägheitsmoment um die Flugzeugquerachse
I_z	Trägheitsmoment um die Flugzeughochachse

I_{xy}	Deviationsmoment
I_{xz}	Deviationsmoment
I_{yz}	Deviationsmoment
Ma	Machzahl
M_{arp}	Nickmoment bzgl. Aerodynamischen Angriffspunkt
M_{cg}	Nickmoment bzgl. Schwerpunkt
M_x	Rollmoment des Triebwerkschubs
M_y	Nickmoment des Triebwerkschubs
M_z	Giermoment des Triebwerkschubs
S	Referenzflügelfläche
T_{tot}	Totaltemperatur
V	Fluggeschwindigkeit gegenüber Luft
l	Referenzflügeltiefe
\bar{q}	Staudruck
a_x	Beschleunigung in Richtung der Flugzeuglängsachse
a_y	Beschleunigung in Richtung der Flugzeugquerachse
a_z	Beschleunigung in Richtung der Flugzeughochachse
dx	Abstand zwischen Schwerpunkt und Aerodynamischen Angriffspunkt in Richtung der Flugzeuglängsachse
dy	Abstand zwischen Schwerpunkt und Aerodynamischen Angriffspunkt in Richtung der Flugzeugquerachse
dz	Abstand zwischen Schwerpunkt und Aerodynamischen Angriffspunkt in Richtung der Flugzeughochachse
m	Flugzeugmasse
\dot{m}	Massenstrom
net	Eingangsgröße für ein Neuron
o_j	Ausgangsgröße für ein Neuron der Schicht j
p	Rollrate
p_{stat}	Statischer Luftdruck
q	Nickrate
r	Gierrate, bzw. Korrelationskoeffizient
t_j	Sollwert für die Ausgabe eines Neuron der Schicht j
\dot{p}	Rollbeschleunigung
\dot{q}	Nickbeschleunigung
\dot{r}	Gierbeschleunigung
x_{CG}	Schwerpunktlage in Richtung der Flugzeuglängsachse
y_{CG}	Schwerpunktlage in Richtung der Flugzeugquerachse
z_{CG}	Schwerpunktlage in Richtung der Flugzeughochachse
x_N	Lage des Neutralpunktes in Richtung der Flugzeuglängsachse

Aerodynamische Derivative

$C_{A\eta}$	Auftrieb infolge Höhenruderausschlag
$C_{A\delta_s}$	Auftrieb infolge Spoilerausschlag
C_{Aq}	Auftrieb infolge Nickdrehrate
$C_{m\eta}$	Nickmoment infolge Höhenruderausschlag
C_{mq}	Nickdämpfung

Vektoren und Matrizen

\mathbf{B}	Drehimpuls
\mathbf{B}_T	Triebwerksdrall
\mathbf{I}_f	Trägheitstensor
\mathbf{F}	Kraftvektor
\mathbf{F}_a	Kraftvektor im aerodynamischen Koordinatensystem
\mathbf{F}_g	Gewichtvektor im geodätischen Koordinatensystem
\mathbf{F}^F	Schubvektor
\mathbf{M}	Äußere Momente
\mathbf{M}_A	Aerodynamisches Moment
\mathbf{M}_F	Triebwerksmoment durch Schub
\mathbf{M}_E	Triebwerksmoment durch Einlauf
\mathbf{M}_{fa}	Transformationsmatrix vom aerodynamischen ins flugzeugfeste Koordinatensystem
\mathbf{X}	Zustandsvektor
$\dot{\mathbf{X}}$	Derivativvektor
\mathbf{u}	Steuervektor
$\boldsymbol{\omega}_f$	Vektor der Winkelgeschwindigkeiten
\mathbf{y}	gemessener Ausgangsvektor
\mathbf{y}^*	Modell Ausgangsvektor
\mathbf{r}_{EL}	Ortsvektor vom Schwerpunkt zum Angriffspunkt des linken Einlaufimpulses
\mathbf{r}_{ER}	Ortsvektor vom Schwerpunkt zum Angriffspunkt des rechten Einlaufimpulses

1 Einführung

1.1 Realität und Simulation

Die Entwicklung der Digitalrechner wurde in den letzten Jahren enorm vorangetrieben. Die Rechenleistung verdoppelte sich im Schnitt alle zwei bis drei Jahre. Mitgewachsen sind die Möglichkeiten, die moderne Software bieten kann. Bei den besonders rechenintensiven Prozessen, wie sie zur Darstellung einer realistischen Landschaft benötigt werden, kann man den Fortschritt in den letzten Jahren sehr gut verfolgen. Auf Billig- PCs, wie sie heutzutage schon zum Haushalt gehören, wird eine Performance im Multimedibereich geboten, die vor zehn Jahren nur kostspielige Großrechner zustande brachten.



Abbildung 1.1: PC-Simulator ¹

In Abbildung 1.1 ist die Darstellung der Außensicht und des Cockpits mit einem Low Cost Simulator zu sehen. Solche Simulatoren werden zum Beispiel als Verfahrenstrainer in der Pilotenausbildung eingesetzt.

¹Quelle: CAT III Systems, Virtual Wings Professional

Die rasante Entwicklung der Hardware hat dazu geführt, daß heute auf dem Markt sehr realistische Flugsimulatoren mit hochauflösendem Sichtsystemen und der Möglichkeit einer Rundum-Sicht angeboten werden.

Eine Rundum-Sicht wird aber zum Beispiel auch mit Hilfe der Virtual Reality Technik geboten. Dazu trägt eine Person einen speziellen Helm, dessen Position und Lage von mehreren Sensoren ermittelt wird, so daß die Bewegung des Kopfes für die Berechnung des neuen Blickwinkels herangezogen wird. Die in Echtzeit berechnete Außensicht wird über zwei, am Helm montierte, Monitore präsentiert.



Abbildung 1.2: Virtual Reality ²

Zusätzlich können hydraulisch bewegte Plattformen dem Piloten das Gefühl des echten Fluges in der Luft verleihen. Sound-Generatoren bilden mit digitalisierten Tönen die Außen- und Innengeräusche des Flugzeuges nach, so daß die Illusion perfekt ist.

Die hohen Kosten für eine hochwertige Simulationsanlage werden von den Hardware-Komponenten verursacht, die oft als Spezialanfertigungen produziert werden müssen. Die Investition in einen Simulator stellt aber nur einen Bruchteil der gesamten Entwicklungskosten für ein modernes Flugzeug dar. Für die Auslegung der heute üblichen *fly-by-wire* Flugsteuerungs- und Regelsysteme ist eine qualitativ hochwertige Simulationsanlage nicht mehr wegzudenken.

Die Qualität der Simulationsanlage kann aber nur so hoch sein, wie die Qualität der Simulationsmodelle, die als Software auf dem Simulationsrechner implementiert sind. Die Leistungsfähigkeit heutiger Computer ist so groß, daß für die Berechnung komplexer Modelle (z.B. Modelle von Triebwerk, Aerodynamik oder Flugregler) etwa eine Millisekunde benötigt wird.

²Quelle: EADS Deutschland, Simulation

Die echtzeitfähigen Modelle können aber andererseits nur so gut sein, wie die zugrundeliegenden Daten (siehe dazu [2]). Die Anforderungen der Anwender an die Simulationsmodelle sind mit den Möglichkeiten der Hardware gestiegen. Sofern die Entwicklung eines Flugzeugs und ein Großteil des Flugtrainings mit Hilfe eines Flugsimulators durchgeführt wird, gilt "gut ist nicht mehr gut genug".

Natürlich muß man je nach Anwendungsfall differenzieren. Flugmechanische Analysen lassen sich zum Beispiel auf einfachen Simulationsumgebungen durchführen. Hier steht die Qualität der Software im Vordergrund, weniger die Hardware. Die Beurteilung der Simulationsergebnisse ist hier auch nicht subjektiv, sondern basiert auf Analysen der Systemantwort im Zeit- und Frequenzbereich.

Für die Zulassung von Trainingssimulatoren gelten ähnliche Anforderungen an die Softwaremodelle, wie sie auch für die Entwicklungssimulatoren gelten. Maßgeblich sind hier die Toleranzen [3], die bei einem *end-to-end-test* erfüllt werden müssen. Bei diesem Test werden Flugmanöver, die während des Flugversuchs aufgezeichnet wurden, am Flugsimulator wiederholt. Die Zeitverläufe einiger charakteristischer Bewegungsgrößen werden dann aus beiden Versuchen verglichen. Liegen die Simulationsergebnisse innerhalb der Toleranz, dann erhält der Simulator seine Zulassung.

Es mag sein, daß einem der Aufwand sehr hoch erscheint, der für die Zulassung eines Trainingssimulators getrieben wird. Dieser ist jedoch gerechtfertigt, wenn man sich überlegt, wie viele Notfallsituationen im Simulator trainiert werden, die im echten Flugzeug zu riskant wären. Dadurch lernen die Piloten das richtige Verhalten, um aus diesen Situationen sicher herauszufinden. Würden diese Anforderungen an die Genauigkeit der Simulation nicht gelten, bestünde die Gefahr, daß das Training umsonst wäre bzw. im Extremfall dem Piloten schadet (*negative training*).

1.2 Problemstellung

Die Rechenverfahren und die Modellierung der Umströmung eines Körpers sind (zur Zeit) noch zu rechenintensiv und ungenau, als daß repräsentative Ergebnisse erwartet werden könnten. Stattdessen werden maßstabsgerechte Modelle eines Körpers (z.B. Flugzeug) in Windkanälen untersucht. Da die hier gewonnenen Ergebnisse nicht ohne weiteres auf das Original skaliert werden können, werden üblicherweise Parameteridentifikationsverfahren eingesetzt, um die charakteristischen Größen mit Hilfe von Meßdaten (z.B. Flugversuch) abzugleichen.

Dies ist natürlich erst möglich, nachdem ein erster Prototyp fertiggestellt und getestet wurde.

Die Parameteridentifikation von aerodynamischen Derivativen ist auch heute noch ein komplexes und vielseitiges Forschungsgebiet. Es erfordert die Zusammenführung einer Reihe unterschiedlicher Disziplinen, wie beispielsweise die Erfassung von Meßdaten durch verschiedenartige Sensoren, die Interpretation der Ergebnisse durch einen

Ingenieur, aber auch die Versuchsplanung und Versuchsdurchführung. Für die Qualität der Identifizierung ist eine sorgfältige Arbeitsweise in allen Disziplinen erforderlich.

Heute ist die Parameteridentifikation in der Flugmechanik integraler Bestandteil jeder Flugzeugentwicklung. Die zentrale Aufgabe in der Flugmechanik ist die Vorhersage und Bewertung der Flugleistung und der Flugdynamik. Als Grundlage zur Beschreibung des Fluges dienen die gekoppelten nichtlinearen Bewegungsgleichungen. Die Gültigkeit und Anwendbarkeit der mathematischen Modelle hängt in hohem Maße von der korrekten Modellierung der äußeren Kräfte und Momente ab, die auf das Flugzeug wirken. Sie können klassifiziert werden als: 1. *Aerodynamische*, 2. *Inertiale*, 3. *Gravitations* und 4. *Antriebs* Kräfte und Momente.

Die Berechnung der aerodynamischen Kräfte stellt jedoch immer noch die größte Unsicherheit dar. Das Problem sind die vielen physikalischen und flugmechanischen Effekte, die während eines Flugmanövers zusammenspielen. Eine genaue Zuordnung von Ursache und Wirkung ist bei der Parameteridentifikation die hohe Kunst der Ingenieure. So werden üblicherweise nur die Parameter identifiziert, die bei diesem Typ Manöver auch angesprochen werden. Wenn zum Beispiel eine Steuerfläche im Vergleich zu den anderen nur wenig ausgeschlagen wurde, dann werden diese Steuerflächenparameter nicht in die Optimierung mit aufgenommen. Man erhält zwar meistens eine sinnvolle Lösung, aber es gibt keine Garantie dafür, daß diese die einzig Richtige ist.

“Es läßt sich schwer sagen, was Wahrheit ist, aber manchmal ist es leicht, etwas Falsches zu erkennen.”³

Zur Bewältigung dieser Identifikationsaufgabe stehen unterschiedliche Lösungsmethoden zur Verfügung. Etabliert haben sich das Ausgangsfehlerverfahren und das Gleichungsfehlerverfahren. Sie werden verwendet, wenn für das dynamische System ein linearisiertes Modell und Meßdaten vorliegen bzw. wenn das physikalische Verhalten durch Gleichungen beschrieben werden kann. Die Optimierungsverfahren sind aber nicht sehr robust gegenüber Meßfehlern. Es ist dadurch viel Aufwand für Datenaufbereitung und Ingenieurwissen zur Optimierung der freien Parameter notwendig.

Alternativ zu diesen Methoden können künstliche Neuronale Netze für Optimierungsaufgaben eingesetzt werden. Sie besitzen die Fähigkeit, anhand von Meßdaten, das Ein- und Ausgabeverhalten eines Systems nachzubilden. Sie bieten dabei Robustheit gegenüber verrauschten Meßdaten und die Fähigkeit zum Generalisieren.

Ob Neuronale Netze zur Parameteridentifikation von aerodynamischen Beiwerten herangezogen werden können und wie praxistauglich die gewonnenen Ergebnisse sind, wurde mit der vorliegenden Arbeit untersucht. Nach einem kurzen einleitenden Abschnitt über künstliche Neuronale Netze, wird darauf näher eingegangen.

³Albert Einstein

1.3 Künstliches Neuronales Netz

Die Bezeichnung “Künstliches Neuronales Netz” wurde aus der Neurobiologie, aufgrund der groben Analogie zum Gehirn von Säugetieren, abgeleitet. Es steht für ein informationsverarbeitendes System, das aus einer großen Anzahl einfacher Einheiten (*Neuronen*) besteht, die sich Informationen in Form der Aktivierung der Neuronen über gewichtete Verbindungen zusenden. Auf neurobiologische Grundlagen und die Analogie zwischen einem künstlichen Neuronalen Netz und den Strukturen im Gehirn wird in dieser Arbeit nicht weiter eingegangen. Hierzu findet man in der Fachliteratur zahlreiche Quellen; zum Beispiel [4, 5].

Es gibt ein weitgefächertes Anwendungsgebiet für Künstliche Neuronale Netze. Sie werden zum Beispiel sehr erfolgreich zur Spracherkennung eingesetzt, außerdem finden sie Verwendung bei der Erstellung von Prognosen (z.B. Aktienkurse). In der Regelungstechnik werden adaptive Regler auf Basis Neuronaler Netze eingesetzt. Auch in der Luftfahrttechnik finden Neuronale Netze Anwendung, zum Beispiel bei der Erkennung von Flugzuständen [6].

Theoretisch gibt es sehr viele Arten, wie ein Neuronales Netz lernen kann. In diesem Fall wird ein überwachtes Lernverfahren für nichtrekurrente Netze eingesetzt. Es handelt sich um das weit verbreitete Gradientenabstiegsverfahren *Backpropagation*, das für die in dieser Arbeit verwendete Netzwerkstruktur modifiziert wurde. Es zeichnet sich durch große Fehlertoleranz bei verrauschten Meßdaten aus. Der Lernvorgang entspricht einer Optimierung freier Parameter mit Hilfe eines Gütekriteriums.

Für das hier verwendete Verfahren muß, auf Grundlage der physikalischen Zusammenhänge eines Systems, ein Modell in Form einer Netzwerkstruktur (*Topologie*) definiert werden. Die Netzwerkstruktur legt fest, welche Neuronen miteinander verknüpft werden.

Für die Neuronen werden nichtlineare Funktionen als Aktivierungsfunktion gewählt. Das bedeutet, ein Neuron hat nicht nur die Zustände aktiv (*gefeuert*) oder inaktiv, sondern liefert, abhängig von seinen Eingangswerten, auch Ausgangswerte in einem schmalen Wertebereich. Dadurch kann das Verhalten eines Systems nichtlinear approximiert werden.

Die Netzwerkstruktur legt auch fest, welches Neuron eine charakteristische Größe für die Beschreibung des Systems (z.B. *aerodynamisches Derivativ*) darstellt. Während der Trainingsphase des Neuronalen Netzes können die Ausgaben dieser Neuronen überwacht werden, so daß eine Plausibilität der Ergebnisse abgeschätzt werden kann.

Die Netzwerkstruktur wird während der Optimierung der Verbindungen (*Gewichte*) und der Schwellenwerte (*bias*) meistens nicht mehr verändert. Sollte sich herausstellen, daß die Struktur nicht für die Beschreibung des Systems geeignet ist, dann muß die Optimierung mit einer geänderten Struktur wiederholt werden.

1.4 Ziel und Aufbau der Arbeit

In der folgenden kurzen Diskussion soll dargestellt werden, wieso der favorisierte Ansatz mit einem Modularen Neuronalen Netz sich für die Identifikation nichtlinearer aerodynamischer Derivative sehr gut eignet. Es werden Verweise auf den Abschnitt 2.3 gegeben, indem die alternativen Methoden noch etwas genauer erläutert werden.

Wie schon Zadeh [7] Anfang der 1960'er Jahre definierte, handelt es sich bei der Systemidentifizierung um die Modellierung eines Systems unter Beobachtung der Ein- und Ausgabe, wobei Modell und System in einem Test das gleiche Verhalten zeigen sollen.

In den letzten Jahrzehnten wurde die Technik der Systemidentifizierung, und insbesondere die Identifikation von aerodynamischen Parametern, auch mit Hilfe der Digitalrechner weit vorangetrieben [8]. Über die Jahre hinweg haben sich so Methoden etabliert, die größtenteils vertrauenswürdige Parameter identifizieren können.

Die Güte der Parameteridentifikationsergebnisse hängt aber stark von der Qualifikation des Anwenders ab. Ein flugmechanisches und aerodynamisches Verständnis ist erforderlich, um einerseits vernünftige Ergebnisse zu erzielen, und um andererseits diese Ergebnisse interpretieren zu können.

Zur Identifikation von aerodynamischen Derivativen werden heute üblicherweise einzelne Flugmanöver benötigt. Ein Optimierungsverfahren versucht dann durch Variation der aerodynamischen Parameter ein Modell zu erstellen, das möglichst gut dieses Manöver nachbilden kann. Diese Methode nennt man auch Einzelpunkt- Identifikation.

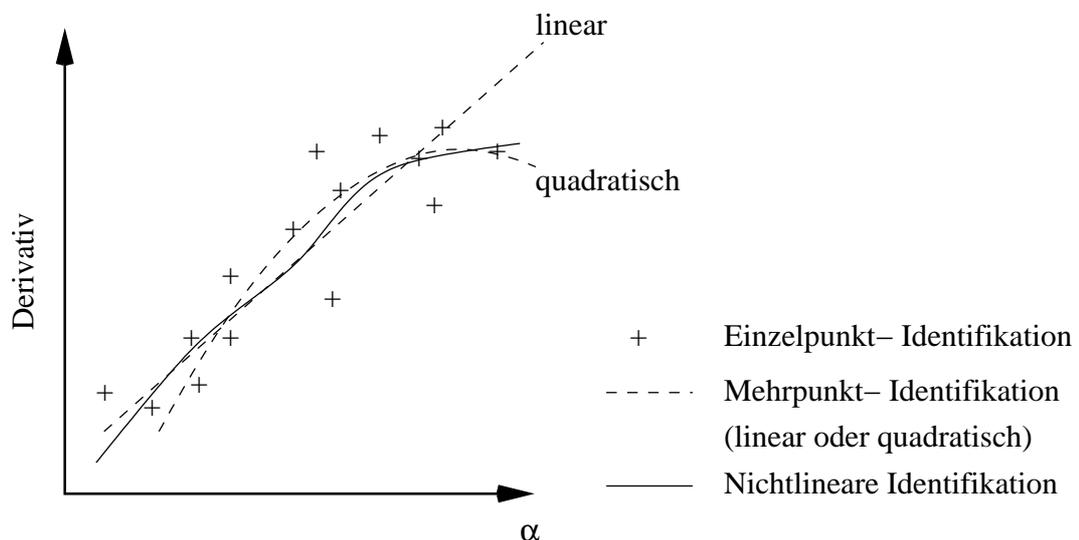


Abbildung 1.3: Beispielhafte Ergebnisse unterschiedlicher Identifikationsverfahren

Nach der Bearbeitung mehrerer Flugmanöver, unterliegen die Ergebnisse üblicherweise einer großen Streuung (siehe Abb. 1.3). Hier wird auch deutlich, warum diese

Ergebnisse fachmännisch interpretiert werden müssen. Die einzelnen Punkte kann man nicht einfach miteinander verbinden, das wäre aerodynamischer Unsinn. Der Fachmann muß sich überlegen, ob durch diese Punkte eine Gerade oder eine Kurve gelegt werden kann, und somit die Einzelergebnisse gut genug approximiert werden. Bei der Mehrpunkt- Identifikation können zusätzliche Parameter in das Modell eingebracht werden, die dann durch eine Optimierung (z.B. Maximum Likelihood Verfahren) die Kurvenverläufe höherer Ordnung identifizieren können [9]. Ein Nachteil dieser beiden Verfahren ist, daß die Identifikationsergebnisse eines Flugmanövers dem Trimmzustand des Flugzeugs vor dem Manöver zugeordnet werden. Gerade bei Spezialmanövern in der Längsbewegung werden aber Anstellwinkel erfolgen, die mehrere Grad Abweichung vom Trimmwert erreichen. Die Ergebnisse werden dadurch ungenau. Dieses Problem vermeidet man durch nichtlineare Identifikation, wie es zum Beispiel mit Modularen Neuronalen Netzen möglich ist.

Dies ist etwa der Stand der Technik bei der Identifikation von aerodynamischen Derivativen mit Hilfe der klassischen Methoden. Wie man sehen kann ist hierbei sehr viel Ingenieurwissen und Handarbeit bei der Datenverarbeitung nötig.

Moderne Verfahren versprechen bei diesen genannten Problemen Abhilfe. In Abschnitt 2.3 werden solche Methoden im Hinblick auf die Identifikation von nichtlinearen aerodynamischen Derivativen aufgezeigt.

Neuronale Netze, und insbesondere Modulare Neuronale Netze, sind diesbezüglich sehr vielversprechend, vor allem in Kombination mit den sogenannten Eingabeverbindungen.

Warum ausgerechnet dieser Netzwerktyp bei den doch sehr vielfältigen Arten von Neuronalen Netzen gewählt wurde, wird im folgenden diskutiert.

Wie kann ein Neuronales Netz nun zur Identifikation von aerodynamischen Derivativen herangezogen werden, und welcher Netzwerktyp und welches Lernverfahren sind dafür am Besten geeignet?

Zunächst einmal liegt es nahe, ein überwachtes Lernverfahren für das Neuronale Netz anzuwenden, da für diese Aufgabe Meßwerte zur Verfügung stehen. Wie bereits erwähnt, wird ein sehr robustes Gradientenabstiegsverfahren gewählt.

Man unterscheidet weiterhin zwischen statischen und dynamischen Neuronalen Netzen mit Rückkopplung (*Rekurrente Neuronale Netze*, siehe Abschnitt 2.3.2). Die Parameteridentifizierung mit dynamischen Neuronalen Netzen wurde bereits untersucht und ähnelt stark dem Ausgangsfehlerverfahren [10]. Die statischen Netzwerktypen mit vorwärtsgerichteten Neuronenverbindungen sind sehr vielversprechend und universell einsetzbar (siehe Abschnitt 2.3.1). Deswegen werden sie hier für die Parameteridentifikation favorisiert.

Bei diesen Typen unterscheidet man, ob die Netzwerkstruktur fest oder adaptiv

gestaltet wird. Eine adaptive Netzwerkstruktur bietet den Vorteil, daß während des Lernvorgangs nicht nur die Gewichte zwischen den Neuronen optimiert werden, sondern auch Verbindungen vollständig entfernt werden können. Beim sogenannten *Pruning* werden diese Neuronenverbindungen, oder auch Neuronen, im Laufe des Lernvorgangs entfernt, die am Gesamtergebnis nur wenig beteiligt sind. Dies erkennt man daran, daß ein Wichtungsfaktor nahe dem Wert Null ist, bzw. ein Neuron immer eine sehr kleine Aktivierung hat. Mit dieser Methode wird eine unnötige Komplexität des Netzes vermieden.

Bei Lokalmodell-Netzen (siehe Abschnitt 2.3.7) kristallisiert sich auch während des Lernvorgangs eine Struktur heraus, die Aufschluß über die Abhängigkeit von Modellparametern gibt. Allerdings wird zunächst von linearen Abhängigkeiten der Eingangsgrößen ausgegangen, und je nach Komplexität des Systems werden die Nichtlinearitäten eingefügt.

Im Falle der Modellierung der Aerodynamik eines Flugzeuges sind jedoch die physikalischen Zusammenhänge bekannt. Wünschenswert ist es, alle Vorkenntnisse über so ein System mit in den Optimierungsprozess einzubringen. Dies ist bei Modularen Neuronalen Netzen möglich, durch Strukturbildung und Initialisierung der Module mit einem Referenzmodell. Jedes Modul repräsentiert hier einen aerodynamischen Effekt. Die Module können unterschiedlich aufgebaut sein (hybrides Netz), zum Beispiel als trainierbare Neuronengruppe oder auch als Datentabelle, aus der die Parameter einfach interpoliert werden.

Die Modularisierung von Neuronalen Netzen hat in der Vergangenheit immer mehr Beliebtheit gefunden, da hier die Möglichkeit der sinnvollen Strukturierung gegeben ist. Bei den bekannten *Gated Modular Neural Networks* (siehe Abschnitt 2.3.8) sorgt eine Instanz dafür, welches Modul passend zum Eingabevektor angesprochen wird.

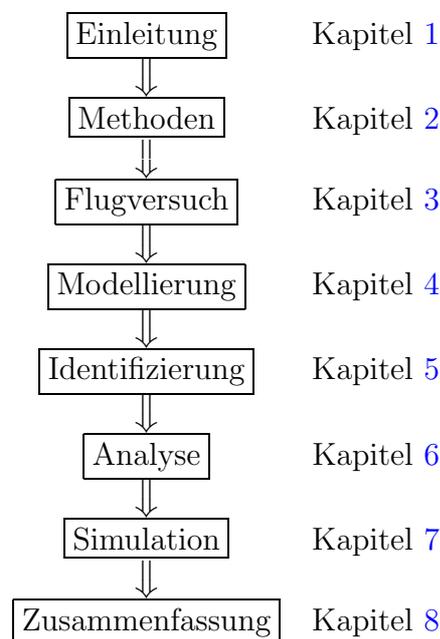
Diese Art und Weise der Schaltung zwischen den Modulen ist aber für ein aerodynamisches Modell weniger geeignet. Hierfür scheinen die sogenannten Eingabeverbindungen passender zu sein. Mit dieser speziellen Art von Neuronenverbindungen wird die Ausgabe eines Moduls mit einem zugehörigen Eingabewert multipliziert. Die Struktur des Modularen Neuronalen Netzes kann somit analog zum aerodynamischen Modell aus Derivativen gestaltet werden. Die einzelnen Module werden also nicht an- oder ausgeschaltet, sondern die unterschiedlichen Anteile pro Modul werden durch Eingangsgrößen gewichtet.

Das grundlegende Ziel dieser Arbeit ist die theoretische und numerische Untersuchung der Parameteridentifizierung eines Systems mit Hilfe eines modularisierten künstlichen Neuronalen Netzes. Als exemplarisches Untersuchungsgebiet dient hier die Modellierung der Aerodynamik eines

Kampfflugzeugs.

Durch detaillierte Hinweise auf die einschlägige Literatur zur Parameteridentifizierung ist die Möglichkeit zur weiteren Vertiefung der Kenntnisse gegeben. Im Hinblick auf die Entwicklung alternativer Methoden zur Systemidentifizierung, beispielsweise mit Hilfe eines Neuronalen Netzes, gestützt durch ein Expertensystem, möge diese Arbeit zum einen den Einstieg in die Konzepte und Methoden der Bestimmung von Systemparametern erleichtern, zum anderen aber auch die Grundlage für Weiterentwicklungen legen.

Um die beschriebene Aufgabenstellung entlang eines “Roten Fadens” behandeln zu können, wurde für diese Arbeit folgende Struktur gewählt.



Nach der Einleitung werden in Kapitel 2 zunächst die klassischen Methoden der Parameteridentifikation vorgestellt. Hierbei handelt es sich um die sogenannten Ausgangsfehlerverfahren und Gleichungsfehlerverfahren. Diese Verfahren haben sich weltweit etabliert und wurden für die Identifizierung unterschiedlicher Systeme eingesetzt. Anschließend werden in Abschnitt 2.3 alternative Methoden zur Parameteridentifizierung beschrieben, die für die Anwendung bei der Identifizierung der aerodynamischen Parameter eines Flugzeugs in Frage kommen.

Für die unterschiedlichen Ansätze werden Literaturhinweise angegeben, die einen tiefergehenden Einstieg ermöglichen sollen.

Die vorgestellten Methoden unterscheiden sich unter anderem in der Möglichkeit, eine Modellstruktur vorzugeben. Sofern das System physikalisch beschrieben werden kann, ist es immer von Vorteil ein entsprechendes Modell zu bilden.

In dem hier vorgestellten Verfahren wird anhand der bekannten aerodynamischen Zusammenhänge dem Neuronalen Netz eine Struktur mitgegeben, die es ermöglicht, die nichtlinearen aerodynamischen Beiwerte und Derivative zu identifizieren. Am Ende des Kapitels 2 wird gezeigt, in welchen Schritten bei diesem Verfahren vorgegangen wird.

Verfährt man nach diesen Schritten, dann muß man sich zunächst Gedanken zum Flugversuch und der Meßdatenerfassung machen. In Kapitel 3 wird deswegen auf spezielle Flugmanöver zur Datenerfassung eingegangen. Diese Manöver werden mit extra dafür ausgerüsteten Flugzeugen durchgeführt. Sie haben Sensoren an unterschiedlichen Positionen, um alle wichtigen Daten zu erfassen, die dann per Telemetrie oder an Bord aufgezeichnet werden können.

In Kapitel 4 wird gezeigt, wie nach dem Baukastenprinzip eine sinnvolle Struktur in das Modulare Neuronale Netz eingebracht wird. So können mit Hilfe von Gruppen und sogenannten Eingabeverbindungen Nichtlinearitäten und Linearitäten gleichermaßen berücksichtigt werden. Es ist sogar möglich, die Gruppen mit A-Priori-Informationen zu initialisieren.

Die Modellierung eines Systems ist die Grundlage für eine anschließende Identifizierung. In Kapitel 5 wird das für die Identifizierung verwendete Gradientenabstiegsverfahren *Backpropagation* vorgestellt und die Anpassung an die Besonderheiten des hier verwendeten Neuronalen Netzwerktyps mathematisch formuliert.

Die identifizierten Parameter müssen auf Plausibilität überprüft werden. Dies setzt voraus, daß das Modell und seine Parameter analysiert werden können. Wie dies bei diesem Neuronalen Netz möglich ist, wird in Kapitel 6 vorgeführt.

Ein schlechtes Ergebnis für einen identifizierten Parameter wird durch eine anschließende Analyse der Sensitivität des Simulationsmodells relativiert. Mit Hilfe eines Flugsimulators zeigt sich, wie stark ein Parameter Einfluß auf die Flugbewegung hat. Gleichzeitig muß man sich aber auch überlegen, ob dieser Parameter identifiziert werden muß bzw. kann. Hierzu sollte vor der Identifizierung der Meßdatensatz auf seinen Informationsgehalt überprüft werden. Die nötige Information kann natürlich nur in den Meßdaten enthalten sein, wenn dem Versuch eine entsprechende Planung vorausging.

In Kapitel 7 wird anhand von Beispielen gezeigt, wie durch das beschriebene Verfahren die Qualität der Simulationsergebnisse verbessert werden konnte. Als Beispiele werden zwei unterschiedliche Manöver gewählt, die das Potential des Verfahrens aufzeigen. In diesem Kapitel wird auch auf die Verwendung des Neuronalen Netzes in der Flugsimulation eingegangen. Hier gelten Anforderungen wie zum Beispiel Echtzeitfähigkeit und Implementationsmöglichkeit des Source- Codes.

Mit einer Zusammenfassung der wichtigsten Ergebnisse und einem Ausblick auf zukünftige Untersuchungen schließt die Arbeit mit Kapitel 8 ab.

1.5 Begriffsdefinitionen

Im Hinblick auf ein besseres Verständnis, sollen zu Beginn dieser Dissertation öfters verwendete Begriffe, wie *Eingabeverbindungen* und *Referenzmodell* etc. erklärt werden.

Eingabeverbindungen:

Unter Eingabeverbindungen verstehen wir eine Verbindung zwischen zwei Neuronen, wobei das Gewicht der Verbindung identisch mit einem Eingabewert des zu modellierenden Systems (hier: Aerodynamik) ist. Es können nun, ganz konventionell, Eingabeneuronen mit Eingabewerten versehen werden, und zusätzlich auch Eingabeverbindungen.

Aerodynamisches Modell:

Das Aerodynamische Modell beschreibt die Kräfte und Momente, die an einem Flugzeug aufgrund der Umströmung der Luft angreifen. Das Modell setzt sich aus aerodynamischen Parametern und einer Vorschrift zusammen. Die aerodynamischen Parameter können in Tabellen abgelegt sein, oder analytisch berechnet werden. Die Vorschrift, oder Modellstruktur, bestimmt die Art und Weise, wie diese Parameter zu einem Gesamtbeiwert zusammengefaßt werden müssen. Im Laufe dieser Arbeit werden unterschiedliche aerodynamische Modelle definiert.

Referenzmodell:

Von einem Referenzmodell wird gesprochen, wenn bereits aerodynamische Parameter und eine Modellstruktur irgendeiner Art vorliegen. Die Parameter befinden sich üblicherweise in Tabellen oder Diagrammen, und eine Vorschrift definiert die Anwendungsweise dieser Parameter.

Basismodell:

Mit Hilfe eines Referenzmodells kann ein Neuronales Netz so trainiert (initialisiert) werden, das eine Basis für eine weitere Optimierung mit Flugmeßdaten zur Verfügung steht. Mit Hilfe eines Basismodells verläuft das Gradiententraining schneller und sicherer.

Optimiertes Modell:

Ein Basismodell, das mit Flugmeßdaten trainiert wurde, nennen wir Optimiertes Modell.

Generiertes Modell:

Ein Neuronales Netz, das ausschließlich mit Flugmeßdaten (ohne Referenzdaten) trainiert wurde, nennen wir Generiertes Modell. Dieses Modell wird ohne Vorkenntnisse neu erstellt.

Differenzmodell:

Ein optimiertes oder generiertes Modell enthält Gesamtbeiwerte (z.B. C_{A0}). In manchen Fällen mag es sinnvoll sein, ein MNN zu erstellen, das die Differenz zwischen einem Referenzmodell und den Sollwerten aus den Flugmeßdaten modelliert (z.B. ΔC_{A0}). Solche Modelle nennen wir Differenzmodelle.

In Abbildung 1.4 werden die Zusammenhänge zwischen diesen unterschiedlichen Modellen aufgezeigt.

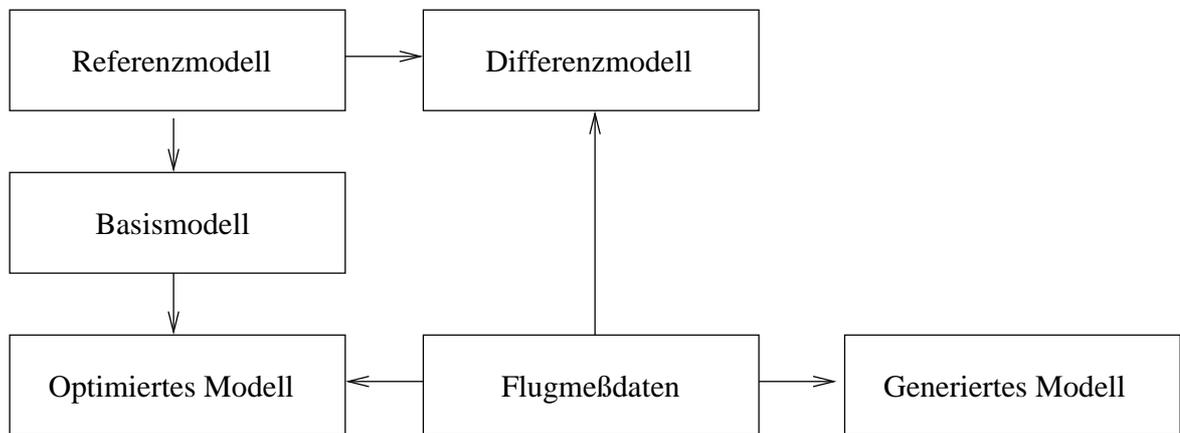


Abbildung 1.4: Begriffsdefinition der Modelle

2 Methoden der Parameteridentifikation

In diesem Kapitel sollen alle Methoden, die sich für eine Identifikation von aerodynamischen Parametern eignen, kurz vorgestellt werden. Ausgehend von den klassischen statistischen Verfahren wird zu den alternativen Ansätzen übergegangen.

2.1 Anforderungen an eine Methode zur Parameteridentifikation

Eine Methode zur Parameteridentifikation soll möglichst genau die Parameter schätzen, mit denen das System gut genug beschrieben werden kann. Im speziellen Fall der Identifikation eines Aerodynamischen Modells wird aus dieser trivialen Forderung folgende Spezifikation abgeleitet:

1. Es sollen die statischen und dynamischen Derivative und Nullbeiwerte bestimmt werden.
2. Nichtlineare und lineare Abhängigkeiten sollen in gleicher Weise erkannt werden.
3. Die Datenverarbeitung soll weitgehend automatisiert sein.
4. Ausgehend von einem Referenzmodell soll eine Optimierung stattfinden.
5. Ein neues Modell soll auf Basis der Meßdaten generiert werden können, falls kein Referenzmodell existiert.
6. Die Methode soll weitgehend universal sein. Das bedeutet, daß auch Systeme unbekannter Art und ohne Vorwissen identifiziert werden können. Eine Modell-erstellung oder Änderung sollte auch während der Optimierung erfolgen.
7. Die Ergebnisse der Optimierung sollen analysierbar sein, um die Plausibilität der Ergebnisse abschätzen zu können.
8. Das optimierte Modell soll ein echtzeitfähiges Simulationsmodell sein (*recall*).
9. Die Methode soll fähig sein, *Online* und *Offline* zu funktionieren; Online bedeutet schon während des Tests (z.B. Flugversuch), Offline bedeutet erst nach dem Test, wenn alle Meßdaten zur Verfügung stehen.

10. Die Methode soll robust sein gegenüber verrauschten Meßdaten, und gut generalisieren können.

2.2 Klassische Parameteridentifikation

2.2.1 Übersicht

Alle großen Flugzeughersteller der Welt blicken auf eine langjährige Erfahrung im Umgang mit klassischen Verfahren zur Parameteridentifikation zurück. Die erfolgreichsten und am weitesten verbreiteten Verfahren sind das Gleichungsfehler- und das Ausgangsfehlerverfahren. Die meisten firmeneigenen Tools basieren auf diesen Verfahren und unterscheiden sich jedoch in der Datenaufbereitung und in den Analysemöglichkeiten.

In Europa sind es wohl die Firmen BAES, Alenia, CASA und DASA (heute EADS), die eine Expertise bei der Parameteridentifikation von flugmechanisch stabilen und instabilen Flugzeugkonfigurationen vorweisen können.

Desweiteren ist das DLR zu nennen, das langjährige Erfahrung mit klassischen Methoden der Parameteridentifikation aufweist, und darüber hinaus alternative Verfahren erforscht.

Am DLR in Braunschweig werden mehrere Verfahren zur Parameteridentifikation verwendet und weiterentwickelt. Eine Übersicht der Entwicklungen bis zum aktuellen Produkt *ESTIMA* ist in Abbildung [2.1](#) dargestellt.

Eine gute Übersicht und einen genauen Einblick in diese Verfahren findet man in [\[8, 11\]](#).

Welche Methoden sich bei den bekanntesten europäischen Flugzeugherstellern etabliert haben, soll hier kurz vorgestellt werden.

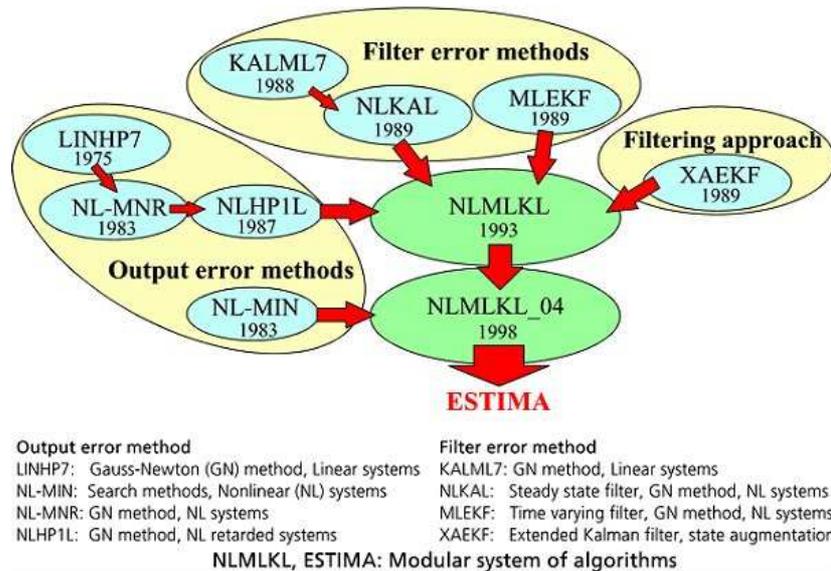


Abbildung 2.1: Entwicklungsstufen der Verfahren zur Parameteridentifikation beim DLR ⁴

Alenia: Hier wird eine Software, basierend auf der klassischen Ausgangsfehlermethode, eingesetzt, die in Verbindung mit dem Maximum Likelihood Verfahren auch für die Identifikation der Parameter anderer Systeme Verwendung findet.

Das Problem bei der Identifikation der aerodynamischen Parameter von flugmechanisch instabilen Flugzeugkonfigurationen ist, daß die Integration der flugmechanischen Bewegungsgleichungen generell divergiert. Deswegen wird hier als Lösung das vollständige nichtlineare Flugzeugmodell mit der Software des Flugsteuersystems (FCS) integriert. Der Vorteil dabei ist: Das vollständige nichtlineare Aerodynamikmodell (aeroelastische Anteile, Terme zweiter Ordnung) wird verwendet, und eine direkte Verifizierung mit dem nichtlinearen Simulationsmodell ist möglich. Die Nachteile bei dieser Methode sind: Ungenauigkeiten im FCS- Modell können die Abschätzung der Parameter beeinflussen, und starke Nichtlinearitäten in der Aerodynamik beeinflussen die Konvergenz des Verfahrens.

BAES: Bei BAES wird eine Software namens *Athena* eingesetzt. Sie basiert auf einem Gleichungsfehlerverfahren und verwendet eine Regressionsanalyse der Hauptkomponenten. Dieses Verfahren wird beschrieben in [12] und wurde erfolgreich eingesetzt (z.B bei den Programmen Harrier und Tucano).

Die herausragende Eigenschaft dieses Tools ist, daß die Modellstruktur iterativ angepaßt werden kann. Außerdem kann mit Hilfe einer Simulation die Flugzeugantwort und die Antwort des optimierten Modells verglichen werden.

EADS Deutschland: Das heute verwendete Parameteridentifikationsverfahren bei der EADS Deutschland basiert auf den Forschungsarbeiten in den siebziger Jahren. Es

⁴Quelle: DLR Institut für Flugsystemtechnik

handelt sich um ein Ausgangsfehlerverfahren mit Maximum Likelihood Optimierung, basierend auf einem linearisierten Modell. Durch entkoppelte Bewegungsgleichungen eignet sich dieses Verfahren auch für hochinstabile Flugzeugkonfigurationen [13, 14, 15]. Der Trick dabei ist, daß für die Integration der Bewegungsgleichungen gemessene Zustandsgrößen verwendet werden.

Angewendet wurde das Verfahren erfolgreich bei Tornado, F-104 CCV (Control Configured Vehicle), EAP (Experimental Aircraft Programme) und auch bei Hubschraubern. Im Laufe der Zeit kamen weitere Tools hinzu, um den Arbeitsprozeß zu beschleunigen, zum Beispiel ein Enttrimmprogramm.

EADS Spanien: Es werden zwei Verfahren eingesetzt. Zum einen das Ausgangsfehlerverfahren, ebenfalls in Verbindung mit entkoppelten Bewegungsgleichungen, und zum anderen ein Gleichungsfehlerverfahren mit Regressionsanalyse, wobei während der Optimierung auch noch enttrimmt wird.

Mit Hilfe der Massen- und Triebwerksmodelle werden die Koeffizientenfehler über der Zeit berechnet. Diese Fehler werden dann mit dem Enttrimmprogramm korrigiert. Das Enttrimmprogramm subtrahiert mit Hilfe der Flugmeßdaten die Inkremente infolge Ruderausläge vom Gesamtbeiwert. Unter Anwendung der Bewegungsgleichungen wird zusätzlich eine flugmechanische Kompatibilitätsprüfung durchgeführt, die dann Aufschluß gibt, ob die Ergebnisse verwertbar sind oder nicht.

2.2.2 Ausgangsfehlerverfahren

Werden die Ausgänge eines Systems mit denen eines Modells verglichen, wobei für Modell und System eine identische Struktur vorausgesetzt wird, spricht man von einem Ausgangsfehlerverfahren (siehe Abbildung 2.2).

Die klassische Methode basiert auf einem linearisierten Modell im Zustandsraum. Der Nachteil dieses Verfahrens ist, daß physikalische Effekte, die nicht in diesem Modell enthalten sind, auch nicht identifiziert werden können.

Vorteile des Ausgangsfehlerverfahrens:

1. Das Modell liefert gerechnete Zeitverläufe aller Zustands- und Bewegungsgrößen.
2. Systematische Meßfehler können identifiziert werden.
3. Alle gemessenen Ausgangsgrößen können in die Auswertung einbezogen werden, auch redundante Messungen.

Zur Optimierung der aerodynamischen Parameter wird zum Beispiel das Maximum-Likelihood-Verfahren eingesetzt. Zu Beginn der Rechnung müssen entweder geschätzte Werte für die Parameter eingesetzt werden oder es wird dafür eine Regressionsanalyse angewendet.

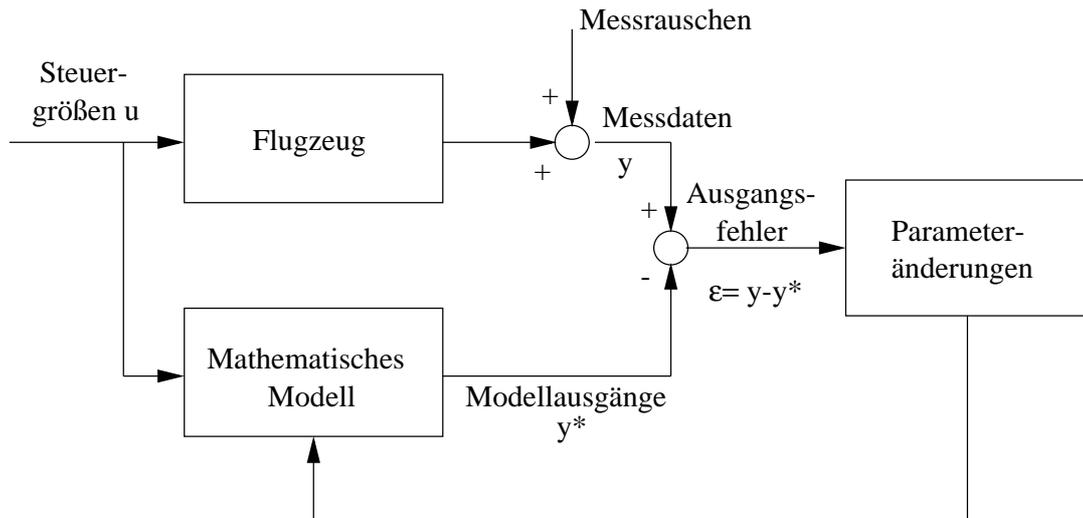


Abbildung 2.2: Ausgangsfehlerverfahren

2.2.3 Gleichungsfehlerverfahren

Bei dem Gleichungsfehlerverfahren werden anstelle der Ausgänge die Fehler eines Satzes von Gleichungen betrachtet, die eine äquivalente Beschreibung bestimmter Eigenschaften des Systems darstellen; zum Beispiel die Fehler der die Beschleunigungen beschreibenden Bewegungsgleichungen der Flugmechanik. Anhand der Gleichungsfehler kann beurteilt werden, ob oder wie gut die Gleichungen durch Einsetzen der gemessenen Werte erfüllt werden können.

Vorteile des Gleichungsfehlerverfahrens:

1. Keine Integration erforderlich, dadurch geringerer Rechenaufwand.
2. Keine Anfangswertbestimmung erforderlich.
3. Identifizierung kann unter Umständen für jede Gleichung getrennt erfolgen.

Das Gleichungsfehlerverfahren ist ähnlich zum Ansatz mit einem Neuronalem Netz (Abschnitt 2.3.1), da auch hier die aerodynamischen Beiwerte betrachtet werden, und nicht die Flugzustände (*states*) aus einer Simulation.

2.3 Alternative Methoden der Parameteridentifikation

In den 90'er Jahren entwickelten künstliche Neuronale Netze eine Reife, die eine industrielle Anwendung erlaubte. In den unterschiedlichsten Bereichen werden seither viele Arten von Neuronalen Netzen sinnvoll eingesetzt. Zum Beispiel bei der Bild- und Spracherkennung, aber auch zur Funktionsapproximation.

Auf dem Gebiet der Luftfahrttechnik wurde eine Anwendbarkeit zur Identifizierung von aerodynamischen Beiwerten untersucht. Mitte der neunziger Jahre kam man zu der Erkenntnis, daß statische vorwärtsgerichtete mehrschichtige Netze (Multi Layer Perceptron, *MLP*) grundsätzlich geeignet sind, ein Modell anhand von Meßdaten zu erstellen. Jedoch scheiterte dieser Ansatz an den mangelhaften Analysemöglichkeiten [16]; der Begriff *Black Box* prägte sich in die Köpfe der Wissenschaftler und Ingenieure. Konsequenterweise wurden weitergehende Forschungen angestellt, um die Vorteile der Neuronalen Netze bezüglich Generalisierung und Robustheit mit der Fähigkeit zur Analyse zu vereinen.

Ein weiteres Ziel war und ist es, Expertenwissen mit in den Optimierungsprozeß einzubringen. Die klassischen Lernalgorithmen wie *Backpropagation* orientieren sich nur am Gradienten einer Fehlerfläche. Das Ergebnis der Optimierung ließ sich aber nicht beeinflussen. Die einzige Möglichkeit bestand darin, die anfängliche Initialisierung des Neuronalen Netzes zu ändern, und auf ein besseres Ergebnis zu hoffen.

2.3.1 Vorwärtsgerichtetes mehrschichtiges Netz (MLP)

Vorwärtsgerichtete Neuronale Netze wurden analog zu der netzartigen Verknüpfung der Nervenzellen im Gehirn konstruiert. Mehrere Neuronen werden im künstlichen Neuronalen Netz in Schichten angeordnet. Üblicherweise werden alle Neuronen einer Schicht mit den Neuronen der Nachbarschicht verknüpft. Jedes Neuron erhält ein Signal von dem Vorgängerneuron und verarbeitet dies mit einer sogenannten Aktivierungsfunktion.

Das Ergebnis wird über eine gewichtete Verbindung an die nachfolgenden Neuronen weitergegeben. Die Anpassung des Neuronalen Netzes an Meßdaten erfolgt üblicherweise mit einem Gradientenabstiegsverfahren (z.B. Backpropagation). Abhängig von dem Ausgabefehler werden für jede Verbindung individuelle Gewichtsänderungen berechnet, so daß iterativ der Ausgabefehler minimiert wird.

Die zwei unterschiedlichen Trainingsmethoden *Online - Offline* sind bekannt. Während beim Online- Training die Netzwerkparameter nach jedem Trainingsmuster verändert werden, so werden sie beim Offline- Training erst nach der Berechnung aller Muster verändert (auch bekannt als *batch mode*). Das Offline- Training ist dafür bekannt, stabiler zu sein, und eine bessere Konvergenz zu zeigen als das Online- Training.

Die Vorteile dieser Art von Neuronalem Netz und dem Trainingsverfahren sind:

1. Robustheit gegenüber verrauschten und fehlerbehafteten Meßdaten
2. Generalisierung und Erkennen von Trends
3. Eine Modellstruktur und Anfangswerte für die zu optimierenden Parameter sind nicht nötig.
4. *MLP* und Backpropagation eignen sich zur automatisierten Datenverarbeitung

Ebenso existieren aber eine Reihe von Nachteilen:

1. Langsame Konvergenz aufgrund des Gradientenabstiegsverfahrens
2. Die Wahrscheinlichkeit ein lokales Minimum zu erreichen, steigt mit der Anzahl der Neuronen.
3. A-Priori-Wissen über eine Modellstruktur oder Anfangsparameter kann nicht mit eingebracht werden.
4. Mangelhafte Analysemöglichkeit
5. Die Ausgabe ist begrenzt auf einen Wertebereich $]-1.0; 1.0[$ (abh. von der Aktivierungsfunktion bzw. Ausgabefunktion).

Generell sind aber diese vorwärtsgerichteten Neuronalen Netze geeignet, mehrdimensionale Funktionen zu approximieren. Abbildung 2.3 zeigt unterschiedliche Ergebnisse der Approximation (Sollwerte sind mit '+' gekennzeichnet).

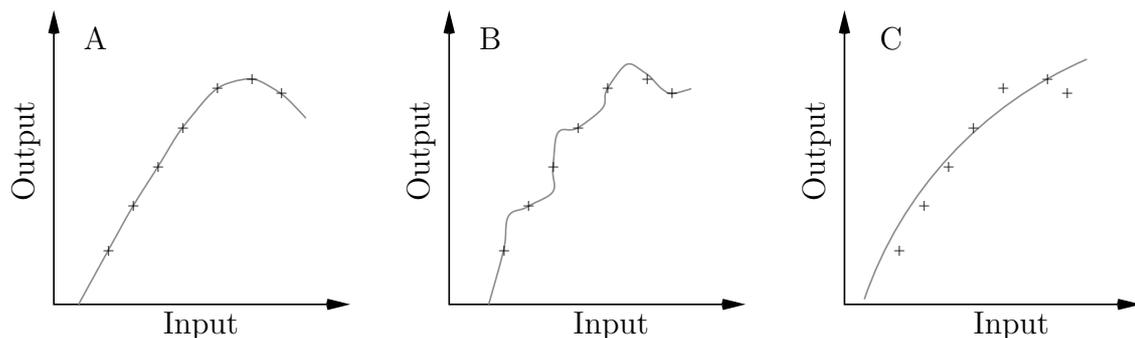


Abbildung 2.3: Unterschiedliche Ergebnisse der Funktionsapproximation

A: Optimale Approximation. Das Neuronale Netz besitzt genauso viele Freiheitsgrade wie das Problem.

B: Zu Viele Freiheitsgrade durch zuviele verdeckte Neuronen (*overfitting*).

C: Gute Verallgemeinerung, aber ungenaue Approximation durch zuwenig Freiheitsgrade (*underfitting*).

Das Ergebnis ist abhängig von der Anzahl der Neuronen und dem Lerndatensatz. Die Bestimmung der optimalen Anzahl der Neuronen und der verdeckten Schichten beruht auf Erfahrungswerten. Die Anwendbarkeit bei der Erstellung eines Aerodynamischen Modells wurde in der Vergangenheit überprüft [16].

2.3.2 Rekurrentes Neuronales Netz (RNN)

Vorwärtsgerichtete Netze werden, wie eben beschrieben, unidirektional berechnet und sind somit statisch. Rekurrente Netze sind im Gegensatz dazu dynamisch, da sie eine Rückführung der eigenen Ausgabe beinhalten. Deshalb leiden Rekurrente Netze oft unter Instabilität und langer Trainings- und Berechnungsdauer.

Dieser Netzwerktyp ist aber geeignet zur Darstellung eines Systems (z.B. Flugzeug) im Zustandsraum und kann somit zur Parameteridentifikation herangezogen werden. Am Institut für Flugmechanik des DLR wurde eine Parameteridentifikation mit Rekurrentem Neuronalem Netz verglichen mit klassischen Ansätzen wie Ausgangsfehlermethode und Gleichungsfehlermethode [10]. Die Ergebnisse sind durchaus vergleichbar im Hinblick auf Konvergenz und Schätzwerte.

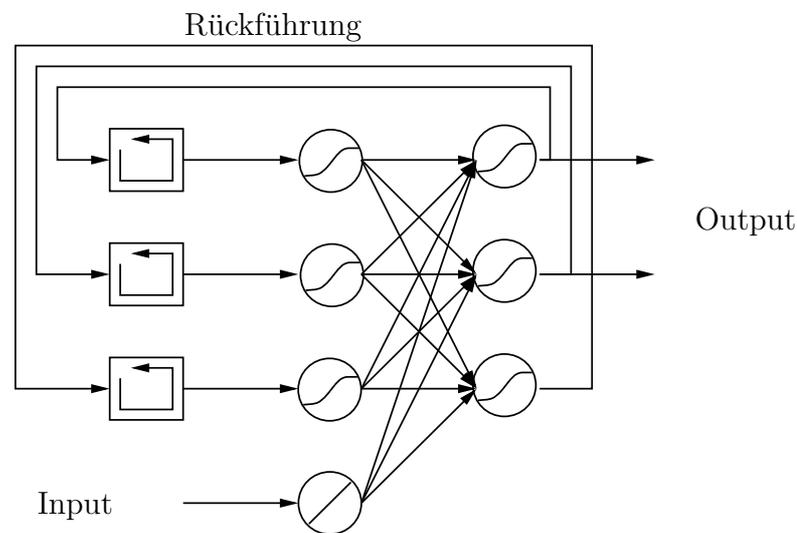
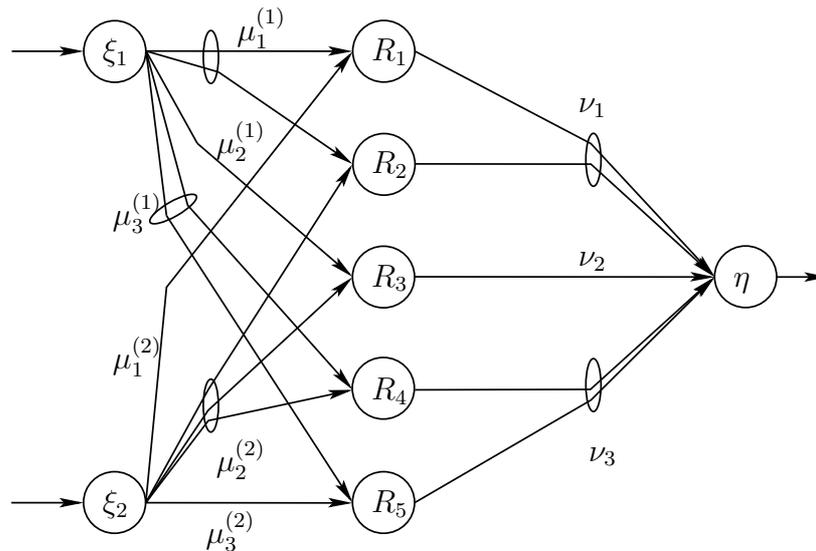


Abbildung 2.4: Rekurrentes Neuronales Netzwerk mit verdeckten Neuronen

2.3.3 Neuro-Fuzzy System

Bestimmte Fuzzy Systeme sind wie bestimmte Neuronale Netze auch universelle Funktionsapproximatoren [17]. Um für ein gegebenes Problem ein geeignetes Fuzzy System zu finden, müssen die Zugehörigkeitsfunktionen μ (Parameter) und eine Regelbasis R (Struktur) spezifiziert werden. Dies kann entweder durch Vorwissen, einen Lernalgorithmus oder durch beides geschehen. Verbindet man ein lernfähiges Neuronales Netz mit einem Fuzzy-System, erhält man sogenannte Neuro-Fuzzy Systeme.

Moderne Neuro-Fuzzy Systeme werden meist als mehrschichtiges, vorwärtsgerichtetes Neuronales Netz dargestellt, wobei sich jedoch die Verbindungsgewichte ν und die in den Neuronen enthaltenen Funktionen von herkömmlichen Neuronalen Netzen unterscheiden.

Abbildung 2.5: Neuro-Fuzzy Netzwerk ⁵

Ein solches Neuro-Fuzzy-System zeichnet sich durch die folgenden Eigenschaften aus:

1. Es besteht aus einem Fuzzy System, das mit Hilfe eines aus dem Bereich der Neuronalen Netze stammenden Lernalgorithmus trainiert wird. Das Lernverfahren operiert auf der Basis lokaler Information und führt auch nur lokale Änderungen im Fuzzy System durch.
2. Es läßt sich als dreischichtiges, vorwärtsgerichtetes Neuronales Netz darstellen, wobei die erste Schicht die Eingaben, die mittlere Schicht die Fuzzy- Regeln und die dritte Schicht die Ausgaben repräsentieren. Die Fuzzy Mengen sind als Verbindungsgewichte kodiert.
3. Es ist jederzeit als ein System aus Fuzzy- Regeln interpretierbar.
4. Das Lernverfahren berücksichtigt die semantischen Eigenschaften des zugrundeliegenden Fuzzy Systems, was zu einer Einschränkung zulässiger Parametermodifikationen führt.

Im Gegensatz zu Neuronalen Netzen, kann bei Neuro-Fuzzy-Systemen A-priori-Wissen genutzt werden. Das bedeutet, wenn für bestimmte Bereiche der zu approximierenden Funktion bereits geeignete Regeln bekannt sind, so kann das Neuro-Fuzzy-System damit initialisiert werden. Die noch fehlenden Regeln müssen durch ein Lernverfahren gefunden werden.

Neuro-Fuzzy-Systeme wurden vor allem in der Regelungstechnik erfolgreich eingesetzt [18], aber auch zur Funktionenapproximation [19].

⁵Quelle: Institut für Betriebssysteme und Rechnerverbund, TU Braunschweig

2.3.4 Dimensionshomogenes Neuronales Netz

Dimensionshomogene Neuronale Netze besitzen einige wichtige Vorteile gegenüber Neuronalen Netzen mit inhomogenen Dimensionen. Mit Hilfe der Dimensionsanalyse lassen sich dimensionsbehaftete Gleichungen vereinfachen, indem die gleiche physikalische Beziehung durch eine kleinere Anzahl von dimensionslosen Größen beschrieben wird.

In Abbildung 2.6 ist ein dimensionshomogenes Neuronales Netzwerk für ein einfaches physikalisches System (Biegestab) angegeben, welches durch folgenden Zusammenhang beschrieben wird.

$$u = \frac{1}{3} \cdot \frac{Pl^3}{EI}$$

Die Auslenkung u des freien Endes ist abhängig von der Last P , der Länge des Stabes l , dem Elastizitätsmodul E und dem Trägheitsmoment I .

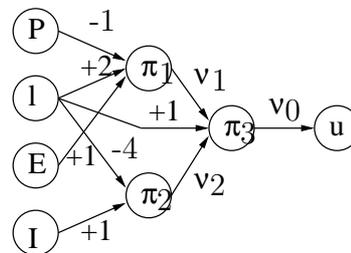


Abbildung 2.6: Strukturierung des Dimensionshomogenen Netzwerks nach Analyse der physikalischen Zusammenhänge ⁶

Bei diesem Netzwerk gibt es lediglich drei freie Parameter $\nu_{0,1,2}$, die durch ein Training mittels Meßdaten identifiziert werden müssen. Ordnet man diesen Parametern folgende Werte zu, $\nu_1 = -1$, $\nu_2 = -1$ und $\nu_0 = \frac{1}{3}$, dann erhält man ein gültiges Modell für den Biegestab.

$$u = l \cdot \nu_0 \cdot \pi_1^{\nu_1} \cdot \pi_2^{\nu_2} = l \cdot \nu_0 \cdot \left(\frac{El^2}{P}\right)^{\nu_1} \left(\frac{l}{I}\right)^{\nu_2}$$

$$u = \frac{1}{3} l \pi_1^{-1} \pi_2^{-1}$$

Durch die Anwendung der Dimensionsanalyse kann also die Größe der Neuronalen Netze reduziert werden. Das begünstigt das Lernergebnis. Der wichtigste Punkt ist jedoch, daß das Neuronale Netz nicht nur die Meßdaten approximiert, sondern daß auch alle anderen Muster, die den gemessenen Mustern physikalisch ähnlich sind, approximiert werden. Somit generalisieren diese Netze in Bereichen des Parameterraums, wo keine Meßdaten verfügbar sind, wesentlich zuverlässiger.

⁶Quelle: Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart

Diese Methode findet zum Beispiel Anwendung in der Regelungstechnik (Adaptiver Regler [20]), aber auch bei der Systemidentifizierung [21].

2.3.5 Neuronales Netz mit Radialsymmetrischer Basisfunktion (RBF)

Neuronale vorwärtsgerichtete Netze, die aus nur einer Schicht von RBF Neuronen und einem Summierungsneuron als Ausgabeneuron bestehen, eignen sich sehr gut zur Funktionenapproximation.

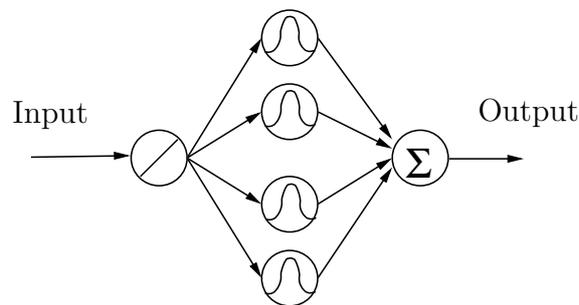


Abbildung 2.7: Neuronales Netzwerk mit einer Schicht von radialsymmetrischen Basisfunktionen

Die große Stärke dieses Netzwerktyps liegt in der Möglichkeit der Initialisierung mit A-Priori-Wissen. Sofern die Gaußfunktion als Aktivierungsfunktion in den RBF Neuronen verwendet wird und pro Meßwert auch ein Neuron zur Verfügung steht, können durch Matrixinvertierung die Gewichte so berechnet werden, daß die Meßwerte exakt wiedergegeben werden. Dieses Netz kann sogar nach der Initialisierung weiter mit Backpropagation trainiert werden. Dabei werden dann typischerweise nur die Neuronen aktiv, die dem Eingabemuster am nächsten liegen. Die Gewichtsänderungen beim Training hängen dann auch nur von diesen Neuronen ab. Die anderen Neuronen und Gewichte bleiben unverändert.

Diese Netze haben aber gewisse Nachteile, wenn aerodynamische Parameter identifiziert werden sollen. Der aerodynamische Datensatz besteht aus einer Vielzahl von Stützstellen, so daß mehrere hundert Neuronen verwendet werden müßten, um den Vorteil der Initialisierung zu nutzen. Die Tendenz zu einem oszillierenden Funktionsverlauf steigt dadurch. Für die Verarbeitung fehlerbehafteter Meßdaten ist aber eine gewisse Generalisierungsfähigkeit erforderlich. Aus diesen Gründen werden die RBF Netze häufiger zur Approximation und nicht zur Interpolation einer Funktion herangezogen.

Anwendung findet dieser Netzwerktyp zum Beispiel bei der Modellierung eines Fahrzeugführers (siehe [22]). Hier kommt es darauf an, während der Fahrt (Online) das Verhalten eines Fahrers zu modellieren. Das lokale Training verhindert die Zerstörung

der Netzwerkbereiche, die durch das aktuelle Eingabemuster nicht aktiviert werden. Einen Versuch zur Abschätzung von Aerodynamischen Beiwerten findet man zum Beispiel in [23].

2.3.6 Polynomansatz

Bei der Verwendung von Polynomen zur Modellierung von Meßdaten können sich Probleme in zwei Bereichen ergeben. Zum einen birgt die Verwendung von Termen höherer Ordnung die Gefahr, daß die Extrapolation in Bereiche, die nicht durch Versuchsdaten abgedeckt sind, fragwürdig wird. Zum anderen erfordern lokale Nichtlinearitäten oft zusätzliche Polynomkoeffizienten, die sich dann aber global auf das gesamte Modell auswirken. Polynome mit hoher Ordnung tendieren zur Oszillation.

Der direkte Polynomansatz zur Interpolation einer eindimensionalen Funktion lautet:

$$g(x) = \sum_{j=0}^n a_j x^j$$

Nach dem Satz von Faber sollte der Grad n des Polynoms nicht zu groß sein ($n \leq 4$), sonst erhält man ungewollte Schwingungen im Funktionsverlauf. Zur Bestimmung der unbekannt Parameter a_j muß das lineare Gleichungssystem aus der Bedingung $g(x) := y(x_i)$ gelöst werden. Da üblicherweise aufgrund der vielen Meßwerte $y(x_i)$ das Gleichungssystem überbestimmt ist, wird eine Näherung statt der Berechnung durchgeführt.

Zur Approximation eines hochgradig nichtlinearen aerodynamischen Systems scheint dieser Ansatz nicht geeignet zu sein.

2.3.7 Lokalmode-Netz (LMN)

Ausgehend von der problematischen Anwendung eines Polynomansatzes wurden am DLR die Lokalmode-Netze entwickelt [24]. Diese bestehen aus einer gewichteten Überlagerung lokal approximierter Ansatzfunktionen, die ein stetig differenzierbares Gesamtmodell ergeben. Bei der Modellerstellung wird der Eingangsraum sukzessive in Teilräume zerlegt, für die dann Teilmodelle bestimmt werden. Die Teilung erfolgt adaptiv, indem jeweils der Teilraum mit dem größten Restfehler geteilt und damit das Modell in diesem Bereich verfeinert wird. Eine Modellverfeinerung tritt damit nur in Bereichen auf, in denen es die zu approximierenden Daten erfordern, ohne daß die anderen Teilmodelle negativ beeinflußt werden. Die Optimierung der veränderlichen Parameter erfolgt zum Beispiel durch das Maximum Likelihood Verfahren.

Durch Beobachtung der Modellverfeinerung im mehrdimensionalen Raum können physikalische Zusammenhänge interpretiert werden. So kann man zum Beispiel erkennen, ob ein Einflußparameter linear oder nichtlinear in das Modell eingeht. Andererseits kann aber auch physikalisches Wissen durch die Erstellung von Teilmodellen

miteingebracht werden.

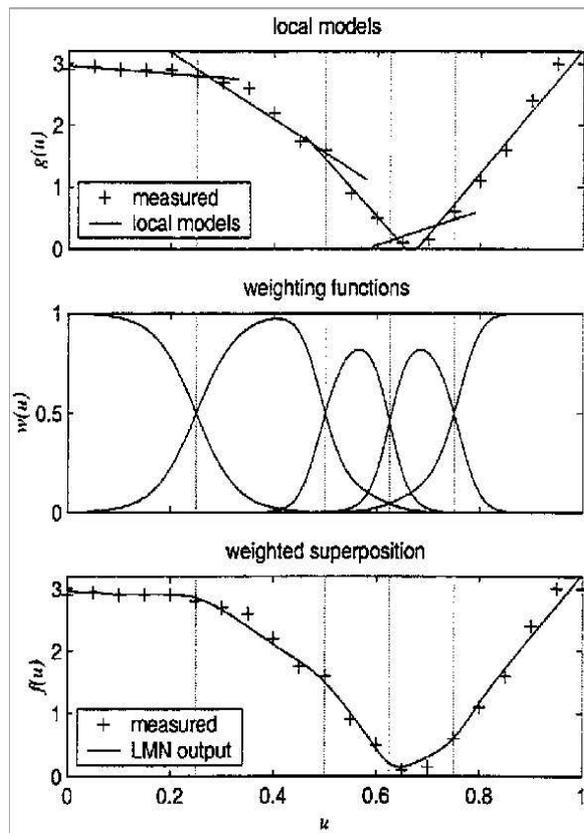


Abbildung 2.8: Prinzip eines Lokalmittel-Netzwerks⁷

Die Analyse hochdimensionaler Lokalmittel-Netze gestaltet sich aber schwierig. Jenseits der dritten oder vierten Dimension ist eine grafische Darstellung nicht mehr möglich.

Die Anwendung der Lokalmittel-Netze ist zur Zeit beschränkt auf die Approximation mit statischen Funktionen. An der Erweiterung auf dynamische Modelle wird momentan geforscht.

Zusammenfassend kann man sagen, daß Lokalmittel-Netze geeignet sind, Einflußparameter und lokale Nichtlinearitäten zu identifizieren, und sich besonders dann eignen, wenn kein Wissen über eine Modellstruktur vorhanden ist.

Die Methode der Lokalmittel-Netze wurde bei der Erstellung eines aerodynamischen Modells für den Do-328 Entwicklungssimulator erfolgreich angewendet [25].

⁷Quelle: DLR Institut für Flugsystemtechnik

2.3.8 Modulares Neuronales Netz (MNN)

Das Miteinbringen von physikalischem Wissen resultiert bei Neuronalen Netzen in einer strukturierten und weniger komplexen Topologie. Die Anzahl der gewichteten Verbindungen lassen sich reduzieren, und somit auch der Freiheitsgrad des gesamten Netzes. Dies hat positive Auswirkungen auf die Trainingsgeschwindigkeit und die Gefahr in lokalen Minima zu enden. Es wird daher empfohlen, Neuronale Netze in mehrere kleine Sub-Netzwerke aufzuspalten, die für unterschiedliche Effekte zuständig sind. Dies ermöglicht das isolierte Bearbeiten einzelner Sub-Netzwerke und verhindert das Beeinflussen anderer, schon trainierter Sub-Netzwerke. Auch hybride Netzwerke können so realisiert werden.

Das Modularisieren eines großen Neuronalen Netzes wurde vor allem bei den Gated Modular Neural Networks (*GMNN*) realisiert [26]. Eine beispielhafte Anwendung zur Systemidentifizierung findet man in [27].

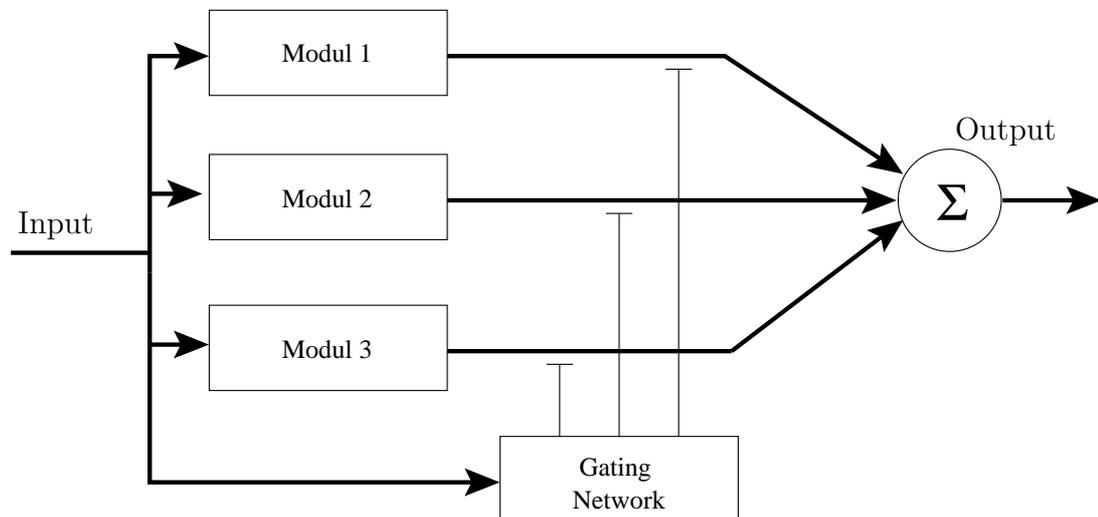


Abbildung 2.9: Blockdiagramm eines Modulare Netzwerks ⁸

Ein Modulares Neuronales Netz ist auch die Grundlage für das in dieser Arbeit präsentierte Parameteridentifikationsverfahren. Im Gegensatz zu einem *Gating Network*, das ein Modul an- oder ausschaltet, werden Eingabeverbindungen verwendet. Der durch ein Modul repräsentierte Effekt wird dadurch proportional zu einem Faktor berücksichtigt. Die Identifikation von nichtlinearen Derivativen wird somit möglich.

2.3.9 Genetische Algorithmen (GA)

Genetische Algorithmen zählen zu den stochastischen Suchverfahren. Diese versuchen ausgehend von einer oder mehreren zulässigen Lösungen (Punkten im Suchraum) sich Schritt für Schritt der optimalen Lösung zu nähern. Genetische Algorithmen

⁸Quelle: Simon Haykin, Neural Networks

verarbeiten eine Menge solcher Punkte / zulässige Lösungen. Diese bezeichnet man auch als künstliche Individuen.

Genetische Algorithmen arbeiten auf Basis dieser künstlichen Individuen, die (in Anlehnung an die Evolution) aus einem Chromosomensatz und einem Fitneßwert bestehen. In der Natur kann der Fitneßwert eine komplexe Größe sein, die wirklich die Anpassung des Individuums an die Umwelt beschreibt. Er kann aber auch eine einzelne Größe sein (wie Länge, Augenfarbe, ...). Bei Genetischen Algorithmen geht man von der einfachen Annahme aus, daß ein Zusammenhang zwischen dem Chromosomensatz und dem Fitneßwert existiert und daß jede Änderung des Chromosomensatzes zu einer Änderung des Fitneßwertes führt. Liegen identische Chromosomensätze vor, so führen gleiche Änderungen an den Chromosomensätzen zu identischen Fitneßveränderungen. Die Elemente eines Genetischen Algorithmus sind neben den Individuen auch Population, Selektion, Kreuzung und Mutation.

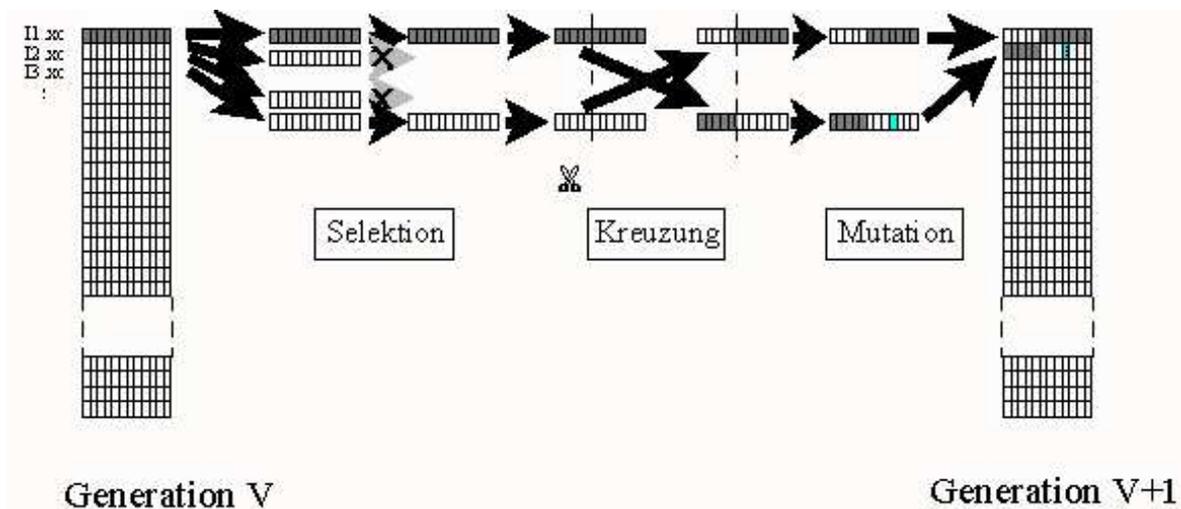


Abbildung 2.10: Schema eines Iterationsschritts des GA zur Bildung von zwei neuen Individuen einer Kinderpopulation.⁹

In [28] werden mehrere Lernverfahren für Neuronale Netze verglichen. Neben dem reinen Genetischen Algorithmus, wird auch eine Kombination von Genetischen Algorithmen und Extended Kalman Filter *EKF* untersucht. Diese Algorithmen wurden zur Abschätzung des aerodynamischen Auftriebsbeiwertes verwendet. Es zeigte sich, daß für eine erste globale Suche der Genetische Algorithmus sehr gut konvergiert. Ab einer gewissen Fehlerschranke wird aber dann besser mit *EKF* weiteroptimiert.

⁹Quelle: Institut für Technische Chemie der Universität Leipzig

2.4 Die Anwendung eines MNN mit Eingabeverbindungen

Es wurden nun unterschiedliche Methoden genannt, die dazu geeignet sind, Systemparameter zu identifizieren. Im Abschnitt 2.3.8 wurde schon ein Modulares Neuronales Netz mit Eingabeverbindungen erwähnt. In diesem Abschnitt sollen nun die Schritte aufgezeigt werden, in welchen eine optimale Identifikation von nichtlinearen aerodynamischen Derivativen mit diesem Netzwerktyp durchgeführt wird.

Bevor ein Flugversuch zur Sammlung von Meßdaten durchgeführt wird, muß die Flugzeugkonfiguration feststehen und die zu erfliegenden Punkte der Flugenvelope bestimmt werden. Alle benötigten Flugmanöver werden in einer *flight card* in einer sinnvollen Reihe angeordnet, so daß sie der Pilot nacheinander durchführen kann.

Sofern die Flugmeßdaten nicht durch Fehler in der Datenerfassung oder Speicherung verfälscht wurden und genügend Information über die gewünschten Flugzustände enthalten sind, kann eine Parameteridentifikation durchgeführt werden. Dazu muß zunächst eine Modellstruktur erstellt werden. Für die klassischen Verfahren verwendet man meistens ein Modell für die Längs- und / oder Seitenbewegung.

Je nach Manövertyp werden dann die Parameter identifiziert, die durch die Ruderauslässe und die Bewegung des Flugzeugs angesprochen werden. Bei einem Pitch-Doubling Manöver sind dies zum Beispiel C_{A0} , $C_{A\alpha}$, $C_{A\eta}$, C_{m0} , $C_{m\alpha}$, $C_{m\eta}$ und eventuell C_{mq} .

Ein Modulares Neuronales Netz bietet bei der Modellierung der Aerodynamik alle Freiheiten. Im Prinzip können alle Parameter von einer beliebigen Meßgröße abhängig gemacht werden. Zum Beispiel könnte man einen Parameter für den Auftrieb in das Modell aufnehmen, der von der Flughöhe nichtlinear abhängig ist. Dieser Parameter ist bei klassischen Methoden nicht so einfach zu modellieren, da typischerweise immer nur ein einziges Manöver betrachtet wird. Zur Identifikation einer Höhenabhängigkeit sind aber viele Meßdaten aus allen Höhen erforderlich.

Und hier liegt auch der große Unterschied zwischen den angesprochenen Verfahren. Die klassischen Methoden betrachten nur ein einziges Manöver; ein MNN betrachtet bei der Identifikation von nichtlinearen Derivativen alle verfügbaren Meßdaten zugleich.

Diese Meßdaten müssen zu einem Lerndatensatz verarbeitet werden. Mit Hilfe einer *flightpath reconstruction* werden die gemessenen Bewegungsgrößen zunächst auf Plausibilität geprüft. Dann berechnet man mit Hilfe der flugmechanischen Bewegungsgleichungen die Gesamtbeiwerte (hier: C_A und C_m).

Mit dem Lerndatensatz, bestehend aus Ein- und Ausgabewerten, wird das MNN mit einem Gradientenabstiegsverfahren trainiert. Schon während des Trainings können die aerodynamischen Derivative (durch Module repräsentiert) analysiert werden, und zum Beispiel mit Referenzdaten verglichen werden.

Nach einer gewissen Trainingsdauer werden die Netzwerkgewichte der Module eingefroren. Das MNN kann dann in einen Flugsimulator integriert werden, mit dem die Flugmanöver nachsimuliert werden können. Ein *proof of match* zeigt, ob die Bewegungsgrößen innerhalb festgelegter Toleranzen liegen.

Zusammengefaßt ergeben sich also folgende Schritte:

1. Planung der Flugmanöver
2. Durchführung des Flugversuchs
3. Modellbildung mit einem Modularen Neuronalen Netz
4. Meßdatenverarbeitung zu einem Lerndatensatz
5. Training des Modularen Neuronalen Netzes
6. Analyse der aerodynamischen Derivative
7. Flugsimulation mit dem optimierten Modularen Neuronalen Netz

3 Flugversuche zur Parameteridentifikation

Flugversuche müssen effizient und sicher durchgeführt werden. Die Wiederholung eines Flugversuchs wäre immer teuer und ist von vielen Umständen (z.B. Wetter) abhängig. Ein Flugmessprogramm stellt eine Aneinanderreihung fest definierter Manöver dar, die mit dem Piloten im Briefing besprochen werden.

3.1 Spezialmanöver

Flugmechanische Manöver zur Parameteridentifikation sind zum Beispiel vom Typ *3211* oder *1123*. Hier wird der Steuerknüppel oder die Steuerpedale mit unterschiedlichen Zeitspannen von einem Anschlag zum anderen maximal ausgeschlagen. Das Flugzeug wird dabei zur Schwingung um eine Achse angeregt.

Desweiteren gibt es noch das *bank-to-bank* Manöver. Hier wird mit konstantem Anstellwinkel zwischen zwei Schräglagen (z.B. $\Phi = \pm 60^\circ$) gewechselt. Dieses Manöver dient vor allem der Identifikation von aerodynamischen Parametern der Seitenbewegung.

Ein sehr häufig verwendetes Manöver für alle drei Flugzeugachsen ist das *doublet*. In Abbildung 3.1 ist zum Beispiel ein pitch-doublet zu sehen.

Der zugehörige Steuerknüppelausschlag (Pitch Stick) ist in der unteren Graphik zu sehen. Die Antwort des Flugzeugs ist durch die Zeitverläufe von Anstellwinkelrate $\dot{\alpha}$ und Nickrate q in der oberen Graphik dargestellt.

Wie man sehen kann, korrelieren die Verläufe der beiden Bewegungsgrößen definitionsgemäß sehr stark. Bei der Identifikation von aerodynamischen Parametern wie zum Beispiel $C_{m\dot{\alpha}}$ und C_{mq} wird bei diesem Manöver eine sinnvolle Trennung schwer möglich sein.

MANOEUVRE: Pitch Doublet
 PRINTDATE: Wed Dec 26 11:31:32 2001

XP8 V2.9
 page: 1 of 1

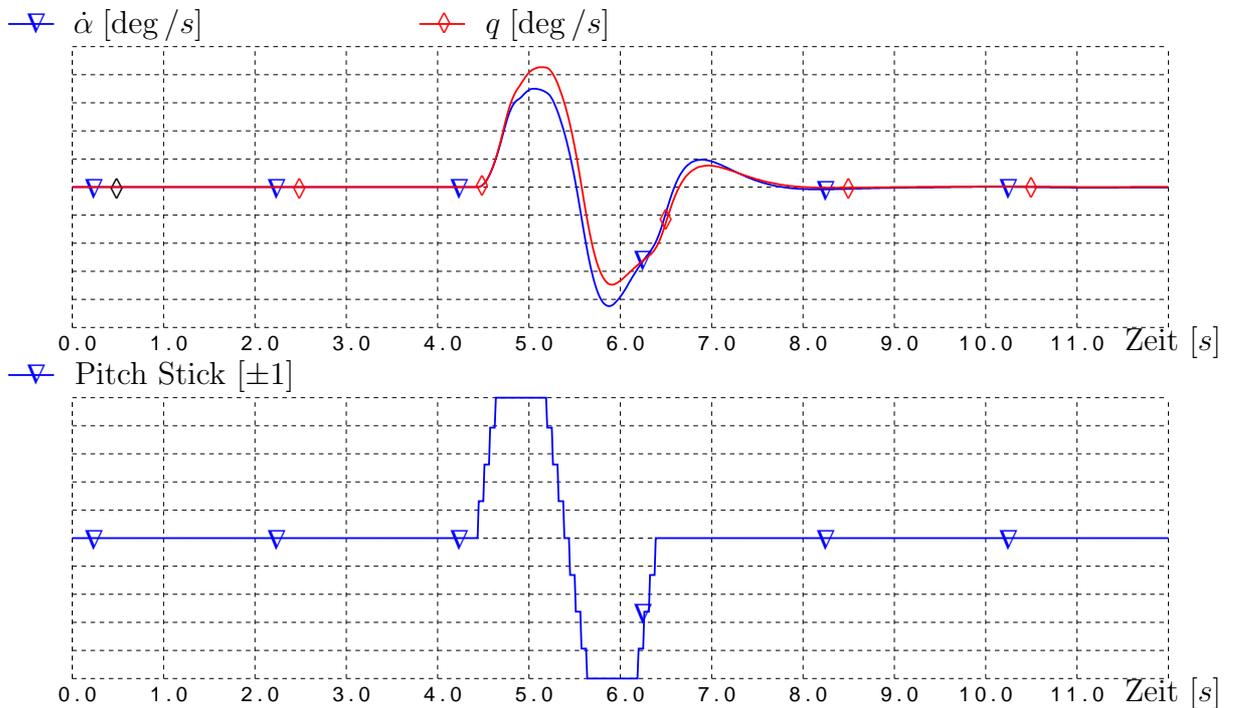


Abbildung 3.1: Zeitverläufe eines Pitch Doublets

Ein anderes Manöver ist dafür besser geeignet (s. Abb. 3.2). Im stationären Kurvenflug korrelieren Anstellwinkelrate und Nickrate nur gering, da der Anstellwinkel sich nicht ändert. Die von der Nickrate abhängigen Derivative, wie zum Beispiel C_{mq} , lassen sich mit solchen Meßdaten besser bestimmen.

Der Pilot wird bei der Durchführung von Spezialmanövern durch die sogenannte *FBI-Box* unterstützt. Diese Box generiert ein exaktes Steuersignal, wie es zum Beispiel beim *doublet* benötigt wird, und speist es in den Flugregler ein. Der Pilot muß nur durch Knopfdruck das Manöver starten. Komplizierte Sonderfälle, wie die Ansteuerung einzelner Steuerflächen (z.B. nur linkes Höhenruder) lassen sich durch entsprechende Softwareänderung realisieren. Der Vorteil dieser Technik liegt dabei in der eindeutigen Zuordnung von Effekt und Ursache.

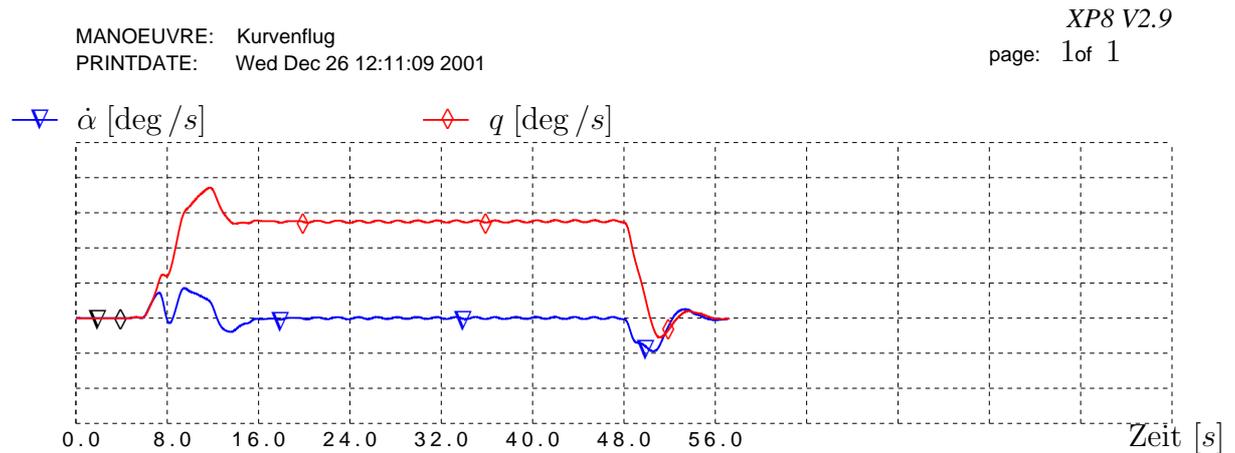


Abbildung 3.2: Stationärer Kurvenflug

3.2 Sensoren

Das Testflugzeug ist mit vielen Sensoren ausgerüstet, deren Meßwerte mit einer festen Taktrate von etwa 32 Hz abgespeichert werden. Dazu gehören die Längsbeschleunigungen a_X, a_Y, a_Z und die Drehraten p, q, r , die mit einer Trägheitsplattform (IMU) gemessen werden. Es wird hier angenommen, daß sich die IMU im Schwerpunkt des Flugzeugs befindet.

Mit einem an der Flugzeugnase befestigten *Noseboom* kann man die Werte der Luftumströmung (α, β, V) messen. Die Ruderausschläge werden mit Potentiometern, und die Luftdrücke (statisch, dynamisch) mit einer Staudrucksonde gemessen. Die Flugzeugmasse m ermittelt man aus der Leermasse und dem Füllstand der Treibstofftanks. Der Flugzeugschwerpunkt x_{CG}, y_{CG}, z_{CG} läßt sich mit der Kenntnis über die Tankpositionen und deren Inhalten bestimmen. Mit dem Satz von Steiner können die Trägheitsmomente und Deviationsmomente berechnet werden.

Alle Sensoren müssen ab und zu kalibriert werden, um konstant ausreichend korrekte Werte zu liefern. Trotzdem kann es vorkommen, daß Meßwerte offensichtlich verfälscht sind, zum Beispiel kinematisch nicht kompatibel sind. Dann können die Sensordaten mit einer *Flightpath Reconstruction* [29] auf Plausibilität überprüft werden.

Aus all diesen Meßwerten sollten die aerodynamischen Kraft- und Momentenbeiwerte der sechs Freiheitsgrade berechnet werden können. Diese Arbeit beschränkt sich auf die zwei Beiwerte C_A und C_m . Sie werden mit Hilfe des Impuls- und Drallsatzes berechnet (siehe Abschnitt 5.1).

4 Modellierung mit einem Modularen Neuronalem Netz

4.1 Die Schritte vom linearisierten Modell zum High Fidelity Modell

Während der Entwicklung eines neuen Flugzeugs durchläuft ein aerodynamisches Modell mehrere Qualitätsstufen:

1. Trimpunkte
2. Nichtlineare Abhängigkeiten (α, Ma), Koppelterme
3. Globales Modell (Mehrpunkt- Identifikation)
4. Inkremente hinzufügen (Überziehen, Bodeneffekt, Fahrwerk)
5. Rudermomente
6. Integriertes Modell

Die ersten drei Punkte werden bei der Parameteridentifikation mit einem Modularen Neuronalem Netz zu einem Schritt zusammengefaßt. Denn es können Trimpunkte und nichtlineare Abhängigkeiten gleichzeitig identifiziert werden. Der hierfür benötigte Lerndatensatz wird aus der gesamten erfliegenen Envelope erstellt. Somit liegt als Ergebnis schon ein globales Modell vor.

Der so erstellte aerodynamische Datensatz wird dann noch mit Inkrementen (zum Beispiel für Bodeneffekte) versehen, um das Flugverhalten in allen Bereichen des Fluges gut zu simulieren. Diese Inkremente werden meistens nicht durch Parameteridentifikation ermittelt, sondern teilweise von vergleichbaren Flugzeugkonfigurationen übernommen oder auch im Windkanal gemessen.

Die aerodynamischen Rudermomente werden für die Auslegung der Flugzeugstruktur und der Hydraulik benötigt. Die Identifizierung der Rudermomentbeiwerte mit einem Neuronalem Netz wird hier nicht untersucht, ist aber mit einem entsprechenden Modell denkbar.

4.2 Neuronales Netz

Im Abschnitt 2.3 wurden verschiedene Arten von Neuronalen Netzen vorgestellt, die zur Parameteridentifikation herangezogen werden können. Sie erfüllen die gestellten Anforderungen überwiegend, boten aber Nachteile bzgl. Analysemöglichkeit und Einbringen von A-Priori-Wissen, die mit der Idee der Modularität beseitigt werden können.

Es wird nun gezeigt, wie ein Modulares Neuronales Netz aufgebaut werden kann, und wie hiermit nichtlineare Derivative identifiziert werden können. Die Vorteile der Modularität sollen in diesem Abschnitt zum Ausdruck kommen.

Zunächst werden die unterschiedlichen Komponenten des Modularen Neuronales Netzwerks beschrieben.

4.2.1 Komponenten

Durch eine Strukturierung des Neuronales Netzwerks, ähnlich der modularisierten GM-NN (siehe Abschnitt 2.3.8), werden die oben erwähnten Vorteile erreicht. Die Struktur entsteht durch Verwendung von Neuronengruppen die über spezielle Verbindungen verknüpft werden. Die Komponenten dieses Netzwerktyps sind also neben den Neuronen, Schichten und Verbindungen auch die Gruppen. Der Netzwerkeingang besteht nicht nur aus Eingabeneuronen, sondern zusätzlich aus Eingabeverbindungen. Der Ausgang wird durch ein Neuron mit linearer Aktivierungsfunktion gebildet. In diesem Neuron werden lediglich alle Ausgänge der vorangestellten Gruppen summiert.

4.2.1.1 Neuronen

Neuronen werden im Zusammenhang mit künstlichen Neuronales Netzen auch Zellen, nodes oder units genannt. Sie besitzen eine Propagierungsfunktion, eine Aktivierungsfunktion und eine Ausgabefunktion. Für die Ausgabefunktion wird meistens die Identität gewählt. Als Aktivierungsfunktion wird bei den hier untersuchten Netzen eine sigmoide Funktion (Abb. 4.1) oder eine radialsymmetrische Funktion gewählt (Abb. 4.2). Jede gewählte Aktivierungsfunktion muß stetig und differenzierbar im Definitionsbereich sein.

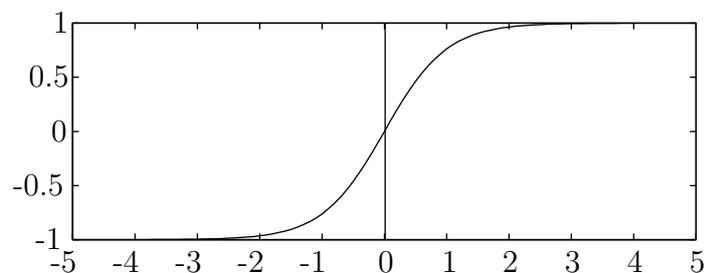


Abbildung 4.1: Graph der Funktion Tangens Hyperbolicus

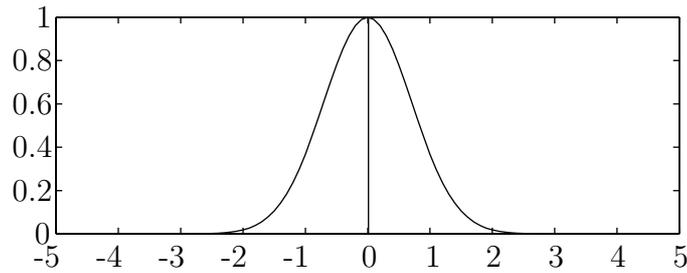


Abbildung 4.2: Graph der Gauß Funktion

Die Zellen der Eingangsschicht werden Eingabeneuronen genannt. Die Zellen der Ausgangsschicht heißen Ausgabeneuronen. Sie haben als Aktivierungsfunktion die Identität. Neuronen in den verdeckten Schichten besitzen eine nichtlineare Aktivierungsfunktion.

4.2.1.2 Verbindungen

Neuronen sind mit gewichteten Verbindungen verknüpft. Das Gewicht der Verbindung von Neuron i nach Neuron j wird hier mit w_{ij} bezeichnet. Bei vorwärtsgerichteten Neuronalen Netzen findet man nur Verbindungen in einer Richtung, von den Eingabeneuronen in Richtung der Ausgabeneuronen. Hier wird unterschieden zwischen:

1. Vollständige Verbindung
2. Vollständige Verbindung mit shortcut connections
3. Nichttrainierbare Verbindung
4. Eingabeverbindung

Bei einer vollständigen Verbindung ist jedes Neuron einer Schicht mit jedem Neuron der Nachbarschicht verbunden. Zusätzliche Verbindungen zu weiteren nicht direkt benachbarten Schichten nennt man *shortcut connections*. Für den hier verwendeten Netzwerktyp werden außerdem nichttrainierbare Verbindungen verwendet. Sie haben das Gewicht 1.0 und werden während einer Optimierung nicht verändert. Sie dienen lediglich der unveränderten Weitergabe des Aktivierungszustandes eines Neurons. Die Eingabeverbindungen ermöglichen das Identifizieren dynamischer Parameter. Sie stellen die gewichtete Verbindung zwischen einer Gruppe und dem Ausgabeneuron dar. Der Wert für ihr Gewicht wird dem Eingabevektor des gesamten Netzes entnommen.

4.2.1.3 Schichten

In der Literatur findet man unterschiedliche Definitionen von Schichten. Entweder bildet man Schichten aus Neuronen oder Schichten aus Verbindungen. Hier werden Neuronschichten verwendet. Es gibt eine Eingangsschicht bestehend aus Eingabeneuronen

und eine Ausgangsschicht aus Ausgabeneuronen. Dazwischen befindet sich eine oder mehrere verdeckte Schichten. Statt einer verdeckten Schicht aus Neuronen ist auch eine Schicht aus Gruppen möglich.

4.2.1.4 Gruppen

Eine Gruppe stellt ein kleines Neuronales Netz dar mit definierten Eingabe- und Ausgabeneuronen. Sie setzt sich zusammen aus mehreren Schichten, die wiederum mehrere Neuronen enthalten können. Für die Wahl der Neuronenanzahl in den verdeckten Schichten kann man die Faustformel $2n + 1$ anwenden. Dabei entspricht n der Anzahl der Eingabeneuronen.

Das Verwenden einer zweiten verdeckten Schicht bietet sich bei stark "zerklüfteten" Funktionsverläufen an. Die Anzahl der dort verwendeten Neuronen basiert auf Erfahrungswerten. Grundsätzlich wird durch Hinzufügen eines verdeckten Neurons, mit der *tanh* Funktion als Aktivierungsfunktion, ein weiterer Wendepunkt der Funktion hinzugefügt.

Während ein einzelnes Neuron entweder eine lineare oder eine einfache nichtlineare Aktivierungsfunktion besitzt, so kann mit einer Gruppe eine Funktion mit vielen Wendepunkten approximiert werden. Mit Hilfe einer Gruppe kann ein Derivat identifiziert werden, daß von einer oder mehreren Eingangsgrößen nichtlinear abhängig ist. Es kann aber auch ein konstantes Derivat identifiziert werden. Dann bietet es sich an, statt der Gruppe nur ein Neuron mit linearer Aktivierungsfunktion zu verwenden. (siehe Abb 4.3).

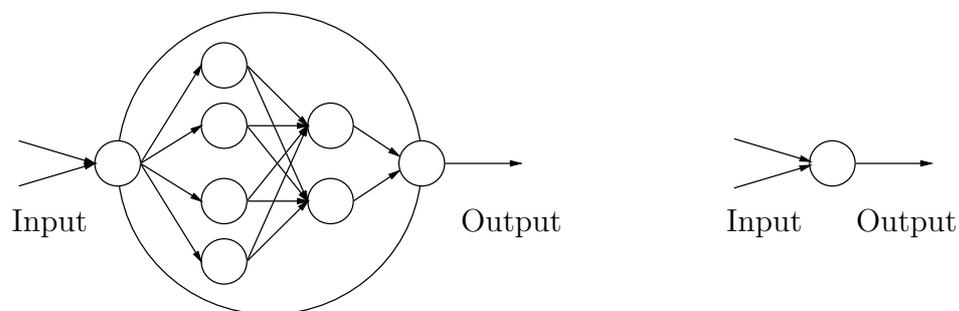


Abbildung 4.3: Analogie zwischen einer Gruppe und einem einzelnen Neuron

Der Wertebereich einer Gruppe kann angepaßt werden, indem das Ausgabeneuron dieser Gruppe mit einem Schwellenwert (*bias*) und einem Faktor beaufschlagt wird.

$$f(net) = a \cdot \tanh(net) + b$$

Mit dem Faktor a wird der Wertebereich der Aktivierungsfunktion *tanh* skaliert und mit dem Wert für b wird eine konstante Verschiebung erreicht.

4.2.2 Neuronales Netz strukturieren

Der große Vorteil des hier verwendeten modularen Netzwerktyps ist, wie bereits erwähnt, die Möglichkeit der Strukturierung. Diese Strukturierung erfolgt entsprechend der physikalischen Zusammenhänge des Systems, das beschrieben werden soll. In der Aerodynamik ist bekannt, welche Auswirkungen die Einflußgrößen auf die Beiwerte haben. Daher können die wahren Effekte und Zusammenhänge sehr gut modelliert werden.

Es wird nun gezeigt, wie Gruppen nach dem Baukastenprinzip hinzugefügt werden und an die relevanten Eingabeparameter angeschlossen werden können. Lineare und nichtlineare Abhängigkeiten, können je nach Verwendung eines Eingabeneurons oder einer Eingabeverbinding vorgegeben werden.

Durch die Verwendung von Gruppen kann ein hochdimensionales Neuronales Netz in mehrere Netze mit geringerer Dimensionalität geteilt werden. Dies wirkt sich positiv auf die Trainingsdauer und das Ergebnis aus.

4.2.2.1 Lineare und Quadratische Einflußparameter

Bei der Modellierung der Aerodynamik eines Flugzeugs werden üblicherweise Derivative eingesetzt. Diese Derivative sind für einen Parameter linearisiert worden. Zum Beispiel werden häufig Steuerflächen- Derivative eingesetzt. Man geht dann davon aus, daß sich die aerodynamischen Kräfte und Momente linear mit dem Steuerflächenausschlag ändern.

Es ist auch möglich Derivative 2. Ordnung zur besseren Funktionsapproximation einzusetzen. So kann man zum Beispiel mit dem folgenden Modell eine nichtlineare Funktion, wie in Abbildung 4.5 dargestellt, hinreichend genau approximieren. Der lineare Einflußparameter ist in diesem Fall allein der Anstellwinkel α .

$$C_A(\alpha) = C_{A0} + C_{A\alpha} \cdot \alpha + C_{A\alpha^2} \cdot \alpha^2$$

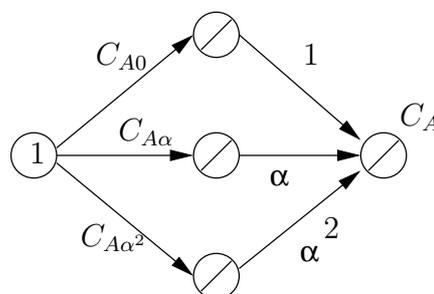


Abbildung 4.4: Netzwerkstruktur entsprechend dem obigen Modell

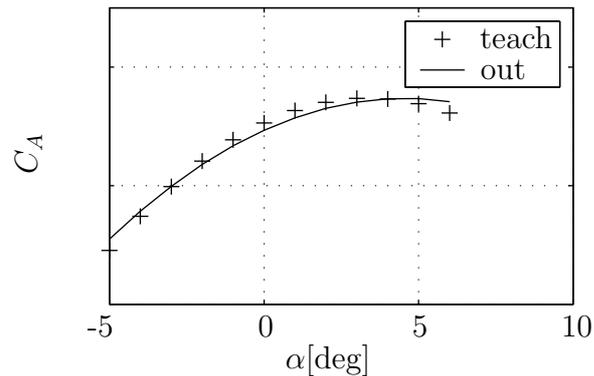


Abbildung 4.5: Gute Approximation mit Derivativen der 1. und 2. Ordnung

Mit diesem Modell stößt man allerdings in der Aerodynamik oft an seine Grenzen. Ein typischer Verlauf für ein von α stark nichtlinear abhängiges Derivativ ist in Abbildung 4.6 zu sehen. Für diesen Funktionsverlauf muß dann ein anderer Ansatz verwendet werden.

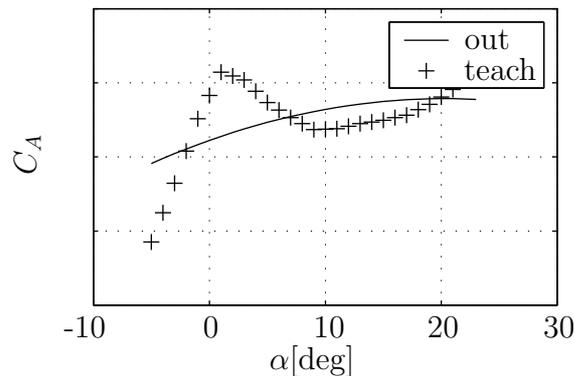
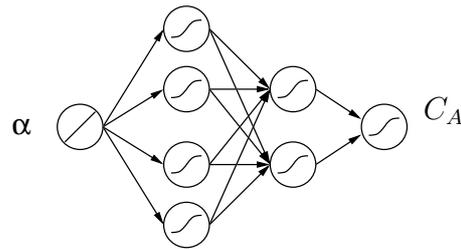
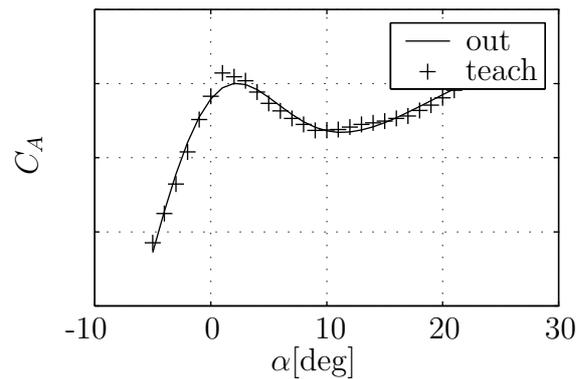


Abbildung 4.6: Schlechte Approximation mit Derivativen der 1. und 2. Ordnung

4.2.2.2 Nichtlineare Einflußparameter

Starke Nichtlinearitäten in den Funktionsverläufen werden vorzugsweise mit kleinen Gruppen von Neuronen approximiert. In Abbildung 4.7 ist beispielsweise so eine Gruppe vom Typ *MLP* zu sehen. Damit ist es möglich, den Funktionsverlauf besser wiederzugeben (vgl. Abb. 4.6 und Abb. 4.8). Für die Wahl der Anzahl der Neuronen gilt: je mehr Neuronen, umso besser die Anpassung bei den Stützstellen; je weniger Neuronen, umso besser die Generalisierung.

Abbildung 4.7: Netzwerkstruktur der C_A GruppeAbbildung 4.8: Ausgabe der C_A Gruppe

4.2.2.3 Vorgabe durch die physikalischen Zusammenhänge

Generell kann mit Hilfe der Gruppen und der Eingabeverbindungen folgende allgemeine Funktion approximiert werden:

$$y = \sum_{i=1}^n a_i \cdot f_i(\mathbf{x})$$

Für jede Funktion f_i muß eine Gruppe bereitgestellt werden, die abhängig von dem Eingabevektor \mathbf{x} ist. Sie wird dann mit dem Faktor a_i multipliziert. Als anschauliches Beispiel dient das folgende Modell mit Berücksichtigung der Einflußparameter α , β , h , $\dot{\alpha}$, q , η und dem Normierungsfaktor $\frac{l}{2V}$. Üblicherweise muß auch noch der Einfluß der Machzahl berücksichtigt werden, in diesem Beispiel, das lediglich die Möglichkeiten eines hybriden Netzes aufzeigen soll, wird sie vernachlässigt.

f_1	$= C_{A0},$	a_1	$= 1$
f_2	$= C_{A\alpha},$	a_2	$= \alpha$
f_3	$= C_{A\alpha^2},$	a_3	$= \alpha^2$
f_4	$= C_{A\eta}(\alpha, \beta),$	a_4	$= \eta$
f_5	$= C_{A\dot{\alpha}}(h),$	a_5	$= \frac{\dot{\alpha}l}{2V}$
f_6	$= C_{Aq}(h),$	a_6	$= \frac{ql}{2V}$

Ausgeschrieben erhält man:

$$C_A = C_{A0} + C_{A\alpha} \cdot \alpha + C_{A\alpha^2} \cdot \alpha^2 + C_{A\eta}(\alpha, \beta) \cdot \eta + C_{A\dot{\alpha}}(h) \cdot \frac{\dot{\alpha}l}{2V} + C_{Aq}(h) \cdot \frac{ql}{2V} \quad (4.1)$$

Durch die modulare Netzwerkstruktur ist es auch möglich, hybride Modelle zu erstellen. Es kann vorkommen, daß existierende aerodynamische Datensätze zwar eingebunden aber nicht optimiert werden sollen oder können. An die Stelle einer Gruppe tritt dann ein Funktionsaufruf zu einem Unterprogramm, welches den Parameter $C_{A\dot{\alpha}}$ aus einer Tabelle ausliest. Da es sich bei dieser Routine nicht um ein Neuronales Netz handelt, wird beim Training des hybriden Netzes der rückwärtspropagierte Fehler an dieser Stelle nicht weiter verfolgt. Die Optimierung der Netzwerkgewichte findet in den anderen drei Gruppen statt. Dieses Unterprogramm muß nur softwaretechnisch mit dem Neuronales Netz vereint werden.

Ein Beispiel für ein hybrides Modell nach (4.1) ist in Abbildung 4.9 zu sehen.

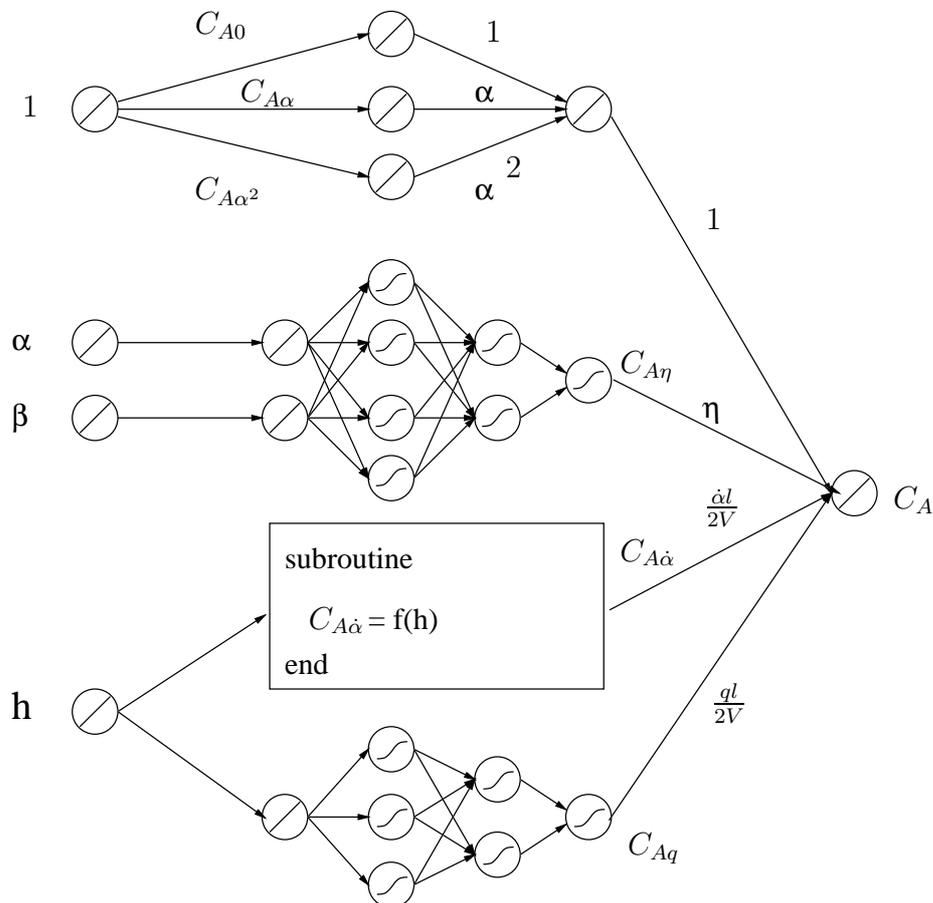


Abbildung 4.9: Hybrides Modell

Bei diesem Modell wird der Nullauftrieb (C_{A0}) und der anstellwinkelabhängige Auftriebsanstieg ($C_{A\alpha}$) mit konstanten Derivativen der 1. und 2. Ordnung modelliert. Der

Zusatzauftrieb durch Ruderausschlag ($C_{A\eta}$) wird mit Hilfe einer Gruppe realisiert, die nichtlinear abhängig ist von α und β . Das Derivat $C_{A\eta}$ wird schließlich mit η multipliziert.

Für das Derivat $C_{A\dot{\alpha}}$ wird bei diesem hybriden Modell eine Software Routine hinzugenommen, die zum Beispiel in einer Hochsprache programmiert sein kann. Als freier Parameter wird die Flughöhe h übergeben.

Das Derivat C_{Aq} wird nichtlinear abhängig von der Flughöhe modelliert. Ein MLP mit einem Eingabeneuron und zwei verdeckten Schichten wird hierfür verwendet.

4.2.2.4 Modellkorrektur

Der Bedarf nach einer Modellkorrektur wird deutlich, wenn der Netzwerkfehler während des Trainings zu groß bleibt. Dann kann das System durch das verwendete Modell nur schlecht wiedergegeben werden. Eine Parameterschätzung kann ja nur dann vernünftige Werte liefern, wenn das Modell die Möglichkeit dazu bietet.

Nach Abbruch des Trainings muß die Netzwerkstruktur geändert werden. Durch zusätzliche Verknüpfungen von Eingabeneuronen mit Gruppen können weitere mögliche physikalische Zusammenhänge berücksichtigt werden. Danach sollte das gesamte Neuronale Netz wieder auf seinen Anfangszustand initialisiert werden. Das anschließende Training kann mit dem selben Lerndatensatz wie vorher durchgeführt werden.

5 Parameteridentifikation mit einem Modularen Neuronalem Netz

In dem folgenden Kapitel wird gezeigt, wie mit einem Modularen Neuronalem Netz die aerodynamischen Derivative eines Flugzeugs nichtlinear identifiziert werden können. Zunächst berechnet man mit Hilfe des Drallsatzes, Impulssatzes und den flugmechanischen Bewegungsgleichungen einen Lerndatensatz für das Neuronale Netz aus Flugtestdaten. Mit diesem Lerndatensatz wird die Optimierung der Verbindungsgewichte durchgeführt.

5.1 Berechnung der aerodyn. Kraft- und Momentenbeiwerte aus Flugversuchsdaten

5.1.1 Koordinatensysteme

Zur Aufstellung der Gleichungssysteme werden das flugzeugfeste, das aerodynamische und das geodätische Koordinatensystem verwendet. Der Ursprung liegt, außer beim geodätischen Achsensystem, im Flugzeugschwerpunkt. Zur Kennzeichnung des Achsensystems erhalten Vektoren und ihre Komponenten einen entsprechenden Index (f, a, g).

Als flugzeugfestes Koordinatensystem wird ein Achsensystem entsprechend [1] gewählt, dessen x,y- Ebene parallel zur Grundrißebene des Flugzeugs ist. Die x- Achse liegt in seiner Längssymmetrieebene und zeigt in Richtung der Flugzeugnase. Die Vektoren in diesem Koordinatensystem werden mit dem Index f versehen. Die Vektoren des aerodynamischen Achsensystems werden mit dem Index a versehen.

Im flugzeugfesten Achsensystem ist der Schubvektor wie folgt definiert:

$$\mathbf{F}_f^F = \begin{pmatrix} F_x^F \cdot \cos\kappa \cdot \cos\sigma \\ F_y^F \cdot \sin\kappa \\ F_z^F \cdot (-\cos\kappa) \cdot \sin\sigma \end{pmatrix} \quad (5.1)$$

Im aerodynamischen Koordinatensystem können die aerodynamischen Kräfte einfach angeschrieben werden:

$$\mathbf{F}_a^A = \begin{pmatrix} -W \\ Q \\ -A \end{pmatrix}$$

Ebenso einfach lässt sich der Kraftvektor des Einlaufimpulses im aerodynamischen System anschreiben:

$$\mathbf{F}_a^E = \begin{pmatrix} -\dot{m} \cdot V \\ 0 \\ 0 \end{pmatrix} \quad (5.2)$$

Im geodätischen Koordinatensystem erhält man für den Vektor der Gewichtskraft:

$$\mathbf{F}_g^G = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix}$$

Um vom aerodynamischen ins flugzeugfeste System wechseln zu können, wird folgende Transformationsmatrix \mathbf{M}_{fa} benötigt:

$$\mathbf{M}_{fa} = \begin{pmatrix} \cos\alpha \cdot \cos\beta & -\cos\alpha \cdot \sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha \cdot \cos\beta & -\sin\alpha \cdot \sin\beta & \cos\alpha \end{pmatrix} \quad (5.3)$$

Für die inverse Transformationsmatrix gilt: $\mathbf{M}_{af} = \mathbf{M}_{fa}^{-1}$

Für die Transformation vom geodätischen System ins flugzeugfeste System benötigt man die Matrix \mathbf{M}_{fg} .

$$\begin{aligned} \mathbf{M}_{fg} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{pmatrix} \cdot \begin{pmatrix} \cos\Theta & 0 & -\sin\Theta \\ 0 & 1 & 0 \\ \sin\Theta & 0 & \cos\Theta \end{pmatrix} \cdot \begin{pmatrix} \cos\Psi & \sin\Psi & 0 \\ -\sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.4) \\ &= \begin{pmatrix} \cos\Theta \cdot \cos\Psi & \cos\Theta \cdot \sin\Psi & -\sin\Theta \\ -\cos\Phi \cdot \sin\Psi + \sin\Phi \cdot \sin\Theta \cdot \cos\Psi & \cos\Phi \cdot \cos\Psi + \sin\Phi \cdot \sin\Theta \cdot \sin\Psi & \sin\Phi \cdot \cos\Theta \\ \sin\Phi \cdot \sin\Psi + \cos\Phi \cdot \sin\Theta \cdot \cos\Psi & -\sin\Phi \cdot \cos\Psi + \cos\Phi \cdot \sin\Theta \cdot \sin\Psi & \cos\Phi \cdot \cos\Theta \end{pmatrix} \end{aligned}$$

5.1.2 Auftriebsbeiwert C_A

Mit Hilfe des Impulssatzes werden die aerodynamischen Kräfte im flugzeugfesten Koordinatensystem berechnet. Für einen starren Körper der Masse m mit der Schwerpunktschwindigkeit \mathbf{V} im körperfesten System lautet er in allgemeiner Form:

$$\frac{d}{dt}(m \cdot \mathbf{V}) \approx m \cdot \frac{d\mathbf{V}}{dt} = \sum \mathbf{F}$$

Die Massenänderung wird im Schub berücksichtigt. Die rechte Seite der Gleichung steht dabei für die äußeren am System angreifenden Kräfte. Für ein Flugzeug mit Triebwerken erhält man im flugzeugfesten System:

$$m \cdot \frac{d\mathbf{V}}{dt} = \mathbf{F}_f^A + \mathbf{F}_f^F + \mathbf{F}_f^E + \mathbf{F}_f^G \quad (5.5)$$

In dieser Gleichung steht \mathbf{F}_f^A für die aerodynamischen Kräfte, \mathbf{F}_f^F für die Triebwerkschubkräfte, \mathbf{F}_f^E für die Kräfte des Einlaufimpulses und \mathbf{F}_f^G für die am Flugzeug angreifende Gewichtskraft.

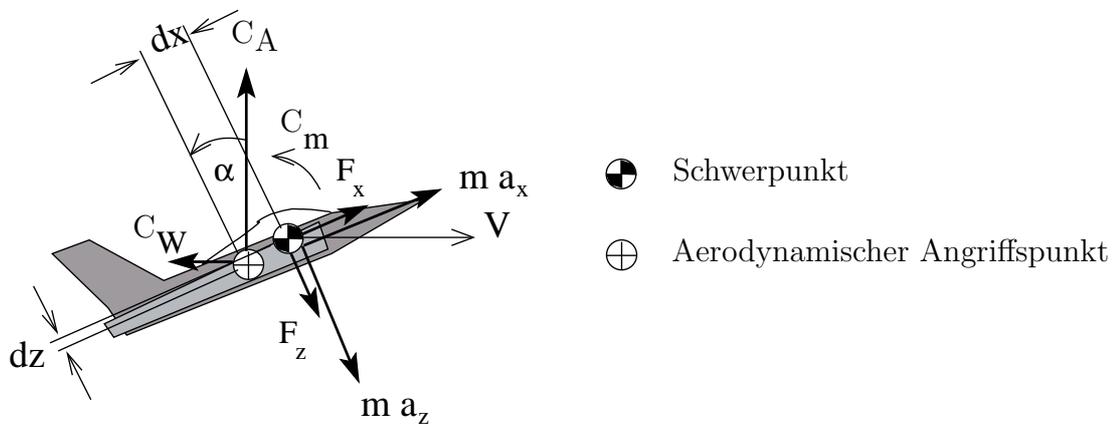


Abbildung 5.1: Kräfte und Momente in der Längsbewegung ¹⁰

Für die Absolutgeschwindigkeit des Flugzeugs im Inertialsystem gilt folgender Zusammenhang:

$$\mathbf{V} = \frac{d\mathbf{r}}{dt} + \boldsymbol{\Omega} \times \mathbf{r}$$

Mit dem Ortsvektor $\mathbf{r} = (x, y, z)^T$ und der Absolutwinkelgeschwindigkeit des Flugzeugs

$$\boldsymbol{\Omega} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

erhält man:

$$\mathbf{V} = \begin{pmatrix} \dot{x} + q \cdot z - r \cdot y \\ \dot{y} + r \cdot x - p \cdot z \\ \dot{z} + p \cdot y - q \cdot x \end{pmatrix} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (5.6)$$

¹⁰Die Gravitationskräfte sind in den Beschleunigungen a_x und a_z enthalten.

5.1 Berechnung der aerodyn. Kraft- und Momentenbeiwerte aus Flugversuchsdaten 45

Die Absolutbeschleunigung $d\mathbf{V}/dt$ ergibt sich wie folgt:

$$\frac{d\mathbf{V}}{dt} = \frac{d'\mathbf{V}}{dt} + \boldsymbol{\Omega} \times \mathbf{V} = \begin{pmatrix} \dot{u} + q \cdot w - r \cdot v \\ \dot{v} + r \cdot u - p \cdot w \\ \dot{w} + p \cdot v - q \cdot u \end{pmatrix} \quad (5.7)$$

Für den Vektor der aerodynamischen Kräfte erhält man angeschrieben im flugzeugfesten System:

$$\begin{aligned} \mathbf{F}_f^A &= \mathbf{M}_{fa} \cdot \begin{pmatrix} -W \\ Q \\ -A \end{pmatrix} = \begin{pmatrix} X^A \\ Y^A \\ Z^A \end{pmatrix} \\ &= \begin{pmatrix} -W \cdot \cos\alpha \cdot \cos\beta - Q \cdot \cos\alpha \cdot \sin\beta + A \cdot \sin\alpha \\ -W \cdot \sin\beta + Q \cdot \cos\beta \\ -W \cdot \sin\alpha \cdot \cos\beta - Q \cdot \sin\alpha \cdot \sin\beta - A \cdot \cos\alpha \end{pmatrix} \end{aligned} \quad (5.8)$$

Für den Vektor der Gewichtskraft im flugzeugfesten System gilt:

$$\mathbf{F}_f^G = \mathbf{M}_{fg} \cdot \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = mg \cdot \begin{pmatrix} -\sin\Theta \\ \sin\Phi \cdot \cos\Theta \\ \cos\Phi \cdot \cos\Theta \end{pmatrix} \quad (5.9)$$

Mit den Gleichungen (5.1), (5.7) und (5.8) erhält man im flugzeugfesten System aus (5.5) folgende Kraftgleichung:

$$\begin{aligned} m \cdot \begin{pmatrix} \dot{u} + q \cdot w - r \cdot v \\ \dot{v} + r \cdot u - p \cdot w \\ \dot{w} + p \cdot v - q \cdot u \end{pmatrix} &= \begin{pmatrix} X^A \\ Y^A \\ Z^A \end{pmatrix} + \begin{pmatrix} F_x^F \cdot \cos\kappa \cdot \cos\sigma \\ F_y^F \cdot \sin\kappa \\ F_z^F \cdot (-\cos\kappa) \cdot \sin\sigma \end{pmatrix} + \begin{pmatrix} F_x^E \\ F_y^E \\ F_z^E \end{pmatrix} \\ &+ mg \cdot \begin{pmatrix} -\sin\Theta \\ \sin\Phi \cdot \cos\Theta \\ \cos\Phi \cdot \cos\Theta \end{pmatrix} \end{aligned} \quad (5.10)$$

Die Berechnung des Normalkraftbeiwertes wird durch Umstellen der Gleichung (5.10) ermöglicht:

$$C_Z = \frac{Z^A}{q \cdot S} = \frac{m(\dot{w} + p \cdot v - q \cdot u) + F_z^F \cdot \cos\kappa \cdot \sin\sigma - F_z^E - mg(\cos\Phi \cdot \cos\Theta)}{q \cdot S}$$

Den Normalkraftbeiwert muß man mit der Transformationsmatrix \mathbf{M}_{af} multiplizieren, um schließlich den Auftriebsbeiwert C_A zu erhalten.

$$C_A = C_Z \cdot \mathbf{M}_{af} = \frac{-C_W \cdot \sin\alpha \cdot \cos\beta - C_Y \cdot \sin\alpha \cdot \sin\beta - C_Z}{\cos\alpha}$$

5.1.3 Nickmomentenbeiwert C_m

Die folgende Herleitung ist an die Ausführungen von H.U. Schäufele angelehnt [14]. Allerdings wurden dort Annahmen bezüglich der Symmetrie des Flugzeugs gemacht. Dies ist hier nicht der Fall, deshalb wird mit dem vollständigen Trägheitstensor gerechnet. Der Vektor der Winkelgeschwindigkeiten in den Komponenten des flugzeugfesten Koordinatensystems lautet:

$$\boldsymbol{\Omega}_f = \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

Der allgemeine Trägheitstensor lautet:

$$\mathbf{I}_f = \begin{pmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{pmatrix}$$

Die zeitliche Änderung des Drehimpulses ist gleich der Summe der äußeren Momente.

$$\frac{d\mathbf{B}}{dt} = \mathbf{M}$$

Da der Trägheitstensor \mathbf{I}_f im flugzeugfesten System konstant ist, ergibt sich die Ableitung des Dralls nach der Zeit zu:

$$\frac{d\mathbf{B}_f}{dt} = \frac{d'\mathbf{B}_f}{dt} + \boldsymbol{\Omega}_f \times \mathbf{B}_f = \mathbf{M}_f$$

Nach der Drallbeschleunigung aufgelöst erhält man:

$$\dot{\mathbf{B}}_f = -\boldsymbol{\Omega}_f \times \mathbf{B}_f + \mathbf{M}_f \quad (5.11)$$

Der Drehimpuls eines Flächenflugzeugs ohne Berücksichtigung des triebwerksbedingten Drehimpulses in den Komponenten des flugzeugfesten Koordinatensystems lautet wie folgt:

$$\mathbf{B}_{f0} = \mathbf{I}_f \cdot \boldsymbol{\Omega}_f = \begin{pmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} I_x p - I_{xy} q - I_{xz} r \\ -I_{xy} p + I_y q - I_{yz} r \\ -I_{xz} p - I_{yz} q + I_z r \end{pmatrix}$$

Für den Gesamtdrehimpuls des Flugzeugs ergibt sich:

$$\mathbf{B}_f = \mathbf{B}_{f0} + \mathbf{B}_f^T$$

Dabei ist \mathbf{B}_f^T der Triebwerksdrall bezüglich des flugzeugfesten Koordinatensystems. Dieser wird allerdings mit der folgenden Begründung vernachlässigt:

Die verwendeten Triebwerke sind Zwei-Wellen-Triebwerke. Es wird angenommen, daß ein resultierender Drehimpuls aufgrund des entgegengesetzten Drehsinns der

5.1 Berechnung der aerodyn. Kraft- und Momentenbeiwerte aus Flugversuchsdaten 47

Wellen nahezu Null wird. Außerdem wird bei den zu untersuchenden Manövern davon ausgegangen, daß der Pilot keine abrupten Änderungen der Schubhebelstellung durchführen wird, was möglicherweise Einfluß auf die Drehimpulsbilanz haben könnte.

Daher folgt für den Gesamtdrehimpuls: $\mathbf{B}_f = \mathbf{B}_{f0}$.

Für die Ableitung von \mathbf{B}_f nach der Zeit ergibt sich gemäß Gleichung (5.11):

$$\begin{aligned}\dot{\mathbf{B}}_f &= - \begin{pmatrix} p \\ q \\ r \end{pmatrix} \times \begin{pmatrix} I_x p - I_{xy} q - I_{xz} r \\ -I_{xy} p + I_y q - I_{yz} r \\ -I_{xz} p - I_{yz} q + I_z r \end{pmatrix} + \mathbf{M}_f \\ &= \begin{pmatrix} I_{xz} p q + I_{yz} q^2 - I_z q r - I_{xy} p r + I_y q r - I_{yz} r^2 \\ -I_x p r + I_{xy} q r + I_{xz} r^2 - I_{xz} p^2 - I_{yz} p q + I_z p r \\ I_{xy} p^2 - I_y p q + I_{yz} p r + I_x p q - I_{xy} q^2 - I_{xz} q r \end{pmatrix} + \mathbf{M}_f\end{aligned}$$

Für $\dot{\mathbf{B}}_f$ gilt außerdem:

$$\dot{\mathbf{B}}_f = \frac{d\mathbf{B}_f}{dt} = \begin{pmatrix} I_x \dot{p} - I_{xy} \dot{q} - I_{xz} \dot{r} \\ -I_{xy} \dot{p} + I_y \dot{q} - I_{yz} \dot{r} \\ -I_{xz} \dot{p} - I_{yz} \dot{q} + I_z \dot{r} \end{pmatrix}$$

Somit erhält man folgende Beziehung:

$$\begin{pmatrix} I_x \dot{p} - I_{xy} \dot{q} - I_{xz} \dot{r} \\ -I_{xy} \dot{p} + I_y \dot{q} - I_{yz} \dot{r} \\ -I_{xz} \dot{p} - I_{yz} \dot{q} + I_z \dot{r} \end{pmatrix} = \begin{pmatrix} I_{xz} p q + I_{yz} q^2 - I_z q r - I_{xy} p r + I_y q r - I_{yz} r^2 \\ -I_x p r + I_{xy} q r + I_{xz} r^2 - I_{xz} p^2 - I_{yz} p q + I_z p r \\ I_{xy} p^2 - I_y p q + I_{yz} p r + I_x p q - I_{xy} q^2 - I_{xz} q r \end{pmatrix} + \mathbf{M}_f \quad (5.12)$$

\mathbf{M}_f stellt die Summe aller äußeren Momente dar. Es setzt sich aus folgenden Teilmomenten zusammen:

$$\mathbf{M}_f = \mathbf{M}^A + \mathbf{M}^F + \mathbf{M}^E \quad (5.13)$$

Das aerodynamische Moment lautet:

$$\mathbf{M}^A = \begin{pmatrix} L \\ M \\ N \end{pmatrix} \quad (5.14)$$

Das Moment durch den Schub eines zweistrahligen Triebwerks lautet:

$$\mathbf{M}^F = \begin{pmatrix} \Delta F_z \cdot dy \\ F_x \cdot dz - F_z \cdot dx \\ \Delta F_x \cdot (-dy) \end{pmatrix} \quad (5.15)$$

wobei ΔF_z die Differenz der beiden Triebwerkskräfte in z -Richtung bedeutet, und ΔF_x entsprechend in x -Richtung.

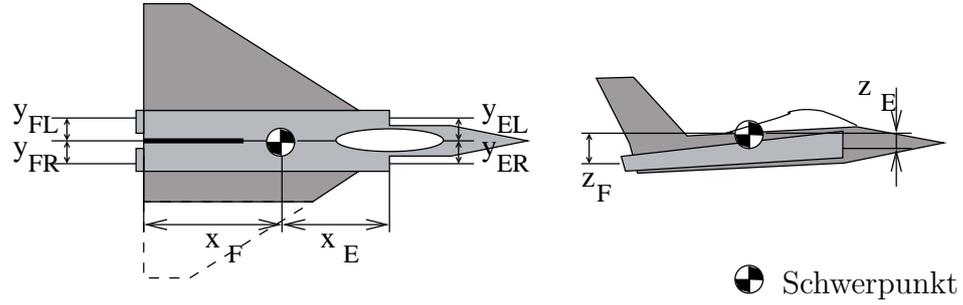


Abbildung 5.2: Hebelarme der Triebwerkskräfte

Das Moment durch den Einlaufimpuls eines zweistrahliges Triebwerks wird nun hergeleitet.

Der Angriffspunkt des Einlaufimpulses für den linken und rechten Einlauf ist durch folgende Ortsvektoren bestimmt:

$$\mathbf{r}_{ER} = \begin{pmatrix} x_E \\ y_{ER} \\ z_E \end{pmatrix}, \quad \mathbf{r}_{EL} = \begin{pmatrix} x_E \\ y_{EL} \\ z_E \end{pmatrix}$$

Das resultierende Moment für den rechten Einlauf ist

$$\mathbf{M}_{ER} = \mathbf{r}_{ER} \times \left\{ \mathbf{M}_{fa} \begin{pmatrix} -V \\ 0 \\ 0 \end{pmatrix} \right\} \cdot \dot{m}_{LR}$$

Dabei ist \mathbf{M}_{fa} die Transformationsmatrix vom aerodynamischen Koordinatensystem in das flugzeugfeste Koordinatensystem (Gl. 5.3). Somit folgt für das Moment:

$$\begin{aligned} \mathbf{M}_{ER} &= \begin{pmatrix} x_E \\ y_{ER} \\ z_E \end{pmatrix} \times \begin{pmatrix} -V \cdot \cos\alpha \cos\beta \\ -V \cdot \sin\beta \\ -V \cdot \sin\alpha \cos\beta \end{pmatrix} \cdot \dot{m}_{LR} \\ &= -V \cdot \dot{m}_{LR} \begin{pmatrix} -z_E \cdot \sin\beta + y_{ER} \cdot \sin\alpha \cos\beta \\ z_E \cdot \cos\alpha \cos\beta - x_E \cdot \sin\alpha \cos\beta \\ x_E \cdot \sin\beta - y_{ER} \cdot \cos\alpha \cos\beta \end{pmatrix} \end{aligned}$$

Analog folgt für den linken Einlauf

$$\mathbf{M}_{EL} = -V \cdot \dot{m}_{LL} \begin{pmatrix} -z_E \cdot \sin\beta + y_{EL} \cdot \sin\alpha \cos\beta \\ z_E \cdot \cos\alpha \cos\beta - x_E \cdot \sin\alpha \cos\beta \\ x_E \cdot \sin\beta - y_{EL} \cdot \cos\alpha \cos\beta \end{pmatrix}$$

Damit wird das Gesamtmoment unter Berücksichtigung von $y_{EL} = -y_{ER}$, $\dot{m}_L = \dot{m}_{LR} + \dot{m}_{LL}$ und $\Delta\dot{m}_L = \dot{m}_{LR} - \dot{m}_{LL}$ zu

$$\mathbf{M}_E = \mathbf{M}_{ER} + \mathbf{M}_{EL} \tag{5.16}$$

$$= -V \cdot \dot{m}_L \begin{pmatrix} -z_E \cdot \sin\beta + (\Delta\dot{m}_L/\dot{m}_L) \cdot y_{ER} \cdot \sin\alpha \cos\beta \\ z_E \cdot \cos\alpha \cos\beta - x_E \cdot \sin\alpha \cos\beta \\ x_E \cdot \sin\beta - (\Delta\dot{m}_L/\dot{m}_L) \cdot y_{ER} \cdot \cos\alpha \cos\beta \end{pmatrix} \tag{5.17}$$

Nach dem Einsetzen von (5.14), (5.15) und (5.17) in Gleichung (5.13) und (5.12) erhält man:

$$\begin{pmatrix} I_x \dot{p} - I_{xy} \dot{q} - I_{xz} \dot{r} \\ -I_{xy} \dot{p} + I_y \dot{q} - I_{yz} \dot{r} \\ -I_{xz} \dot{p} - I_{yz} \dot{q} + I_z \dot{r} \end{pmatrix} = \begin{pmatrix} I_{xz} pq + I_{yz} q^2 - I_z qr - I_{xy} pr + I_y qr - I_{yz} r^2 \\ -I_x pr + I_{xy} qr + I_{xz} r^2 - I_{xz} p^2 - I_{yz} pq + I_z pr \\ I_{xy} p^2 - I_y pq + I_{yz} pr + I_x pq - I_{xy} q^2 - I_{xz} qr \end{pmatrix}$$

$$+ \begin{pmatrix} L \\ M \\ N \end{pmatrix} + \begin{pmatrix} \Delta F_z \cdot dy \\ F_x \cdot dz - F_z \cdot dx \\ \Delta F_x \cdot (-dy) \end{pmatrix}$$

$$+ -V \cdot \dot{m}_L \begin{pmatrix} -z_E \cdot \sin\beta + (\Delta\dot{m}_L/\dot{m}_L) \cdot y_{ER} \cdot \sin\alpha \cos\beta \\ z_E \cdot \cos\alpha \cos\beta - x_E \cdot \sin\alpha \cos\beta \\ x_E \cdot \sin\beta - (\Delta\dot{m}_L/\dot{m}_L) \cdot y_{ER} \cdot \cos\alpha \cos\beta \end{pmatrix}$$

Werden jetzt nur die Anteile die für die Momentenbilanz um die Querachse relevant sind berücksichtigt, folgt daraus die nach der Nickbeschleunigung aufgelöste vollständige nichtlineare Nickmomentengleichung:

$$\dot{q} = \frac{I_{xy}}{I_y} \dot{p} + \frac{I_{yz}}{I_y} \dot{r} + \frac{I_z - I_x}{I_y} pr + \frac{I_{xy}}{I_y} qr + \frac{I_{xz}}{I_y} r^2 - \frac{I_{xz}}{I_y} p^2 - \frac{I_{yz}}{I_y} pq + \frac{M}{I_y}$$

$$+ \frac{z_F \cdot F_x - x_F \cdot F_z}{I_y} - \frac{V \cdot \dot{m}_L}{I_y} (z_E \cdot \cos\alpha \cos\beta - x_E \cdot \sin\alpha \cos\beta)$$

Oder aufgelöst nach dem aerodynamischen Nickmoment:

$$\begin{aligned}
 M &= I_y \dot{q} - I_{xy} \dot{p} - I_{yz} \dot{r} - (I_z - I_x) pr - I_{xy} qr - I_{xz} r^2 + I_{xz} p^2 + I_{yz} pq \\
 &- z_F \cdot F_x + x_F \cdot F_z + V \cdot \dot{m}_L (z_E \cdot \cos \alpha \cos \beta - x_E \cdot \sin \alpha \cos \beta)
 \end{aligned} \tag{5.18}$$

Aus dem aerodynamischen Moment kann nun mit folgender Beziehung der Nickmomentenbeiwert C_m bezüglich des Flugzeugschwerpunktes berechnet werden.

$$C_m = \frac{M}{\bar{q} \cdot S \cdot l} \tag{5.19}$$

5.1.3.1 Nichtrekursives differenzierendes Tiefpaßfilter

In Gleichung (5.18) müssen nicht nur die Drehraten p, q, r sondern auch die Drehbeschleunigungen $\dot{p}, \dot{q}, \dot{r}$ eingesetzt werden. Die Sensorik im Testflugzeug lieferte keine Drehbeschleunigungen, sondern nur die Drehraten bei einer Aufzeichnungsrate von 32 Hz. Aus diesem verrauschten Meßsignal muß durch Differenzieren die Drehbeschleunigung errechnet werden, um dann aus ihr den Nickmomentenbeiwert C_m berechnen zu können.

Für diese Aufgabe wurde ein digitales differenzierendes Filter verwendet [30, 31]. Dieses Filter besitzt, gegenüber analogen Filtern, den Vorzug, daß es ohne Phasenverschiebung geglättete Werte liefert. Gleichzeitig differenziert der Algorithmus noch die Eingangswerte.

Es ist sehr wichtig, die Eckfrequenz und die Anzahl der Filterkoeffizienten richtig zu wählen. Das Ergebnis hängt stark von diesen Einstellungen ab. Für das verwendete Tiefpaßfilter wurden 27 Filterkoeffizienten verwendet, und als Eckfrequenz 0.5 Hz eingestellt, bei einer Abtastrate von 32 Hz. Das Ergebnis ist in Abbildung 5.3 zu sehen.

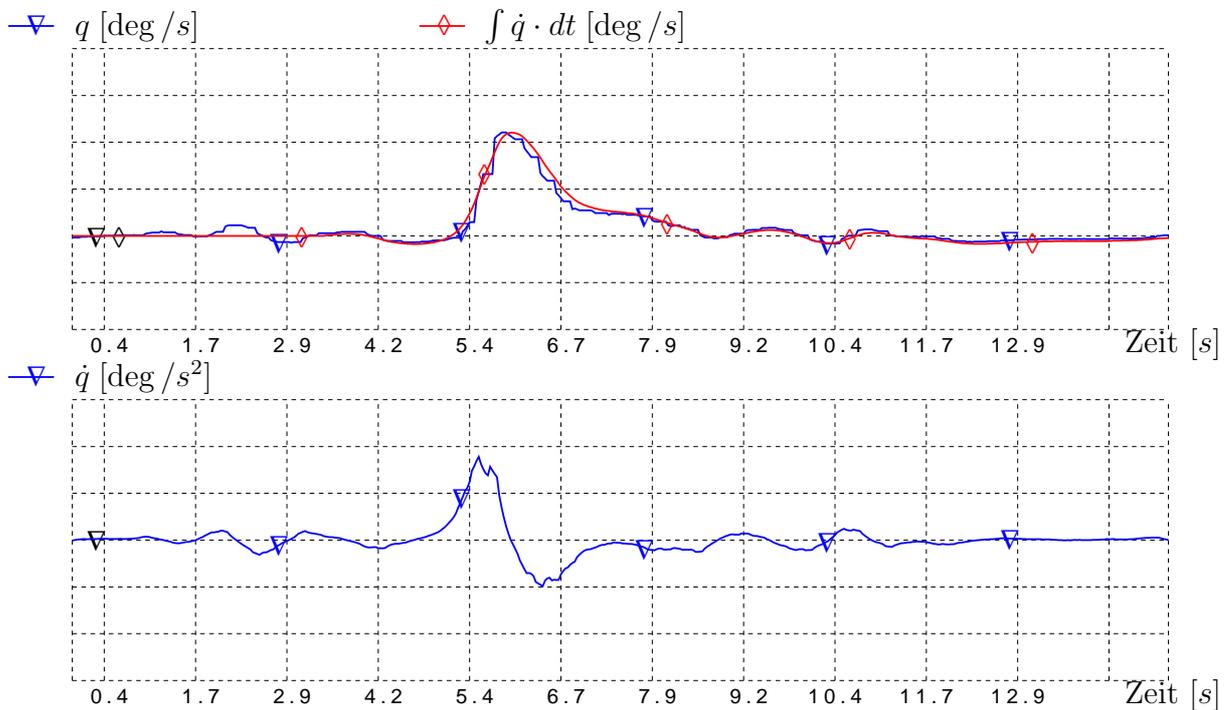


Abbildung 5.3: Drehrate und differenzierte Drehbeschleunigung während eines Fullstick Manövers

Zum Vergleich wurde neben dem q -Signal auch die integrierte Drehbeschleunigung $\int \dot{q} dt$ dargestellt, die mit Hilfe des digitalen Filters gewonnen wurde. Die Integration der Drehbeschleunigung wurde nach 4 Sekunden begonnen. Man sieht, daß diese beiden Zeitverläufe sehr gut übereinanderliegen, und somit die differenzierten Werte akzeptabel sind.

Aus den Werten von $\dot{p}, \dot{q}, \dot{r}$ kann dann, wie oben beschrieben (Abschnitt 5.1.3), der Gesamtbeiwert C_m berechnet werden.

5.1.4 Zusammenhang zwischen C_A und C_m

Theoretisch hängen die Drehmomente von den angreifenden Kräften ab; wie zum Beispiel das Nickmoment von der Auftriebskraft. Beide Beiwerte sind somit über den Hebelarm $dx = x_N - x_S$ gekoppelt (siehe Abb. 5.1). Somit ist eine kombinierte Identifikation von z.B. $C_{m\eta}$ und $C_{A\eta}$ denkbar.

$$C_{m\eta} \approx -C_{A\eta} \cdot \cos\alpha \cdot dx$$

Bei Militärflugzeugen gilt dieser Zusammenhang nur eingeschränkt. Die Umströmung des Flügels hat aufgrund des kurzen Leitwerkhebelarms einen geänderten Anstellwinkel

am Höhenleitwerk zur Folge. Deswegen gilt obige Beziehung nur näherungsweise.

5.2 Lerndatensätze

Lerndatensätze sind nötig, damit ein Neuronales Netz lernen kann. Aus den im Flugversuch gewonnenen Meßdaten können Lerndatensätze erstellt werden, oder auch aus einem aerodynamischen Referenzmodell, sofern es zur Verfügung steht.

Der Datensatz enthält eine Menge von Lernmustern (*pattern*). Pro Lernmuster stehen die Werte für alle Eingänge und Ausgänge zur Verfügung. Ein Beispiel für einen einfachen Lerndatensatz mit drei Lernmustern ist in Tabelle 5.1 angegeben. Optional können auch Gradienten für alle Eingänge enthalten sein, falls diese für das Training oder zur Validierung des Modells benötigt werden.

Häufig werden aus den Meßdaten zwei getrennte Datensätze gewonnen. Aus einer Hälfte wird ein Lerndatensatz erstellt, und aus der anderen Hälfte ein Testdatensatz. Während des Trainings dient der Lerndatensatz der Optimierung der Verbindungsgewichte. Zwischendurch kann mit dem Testdatensatz der Fehler des Neuronalen Netzes beurteilt werden. Ein ansteigender Fehler wird häufig als Abbruchkriterium beim Lernen eines Neuronalen Netzes verwendet. Die Fehlerverläufe der Lernmenge und der Testmenge sind zum Beispiel in Abbildung 6.9 zu sehen.

5.2.1 Lerndatensatz aus einem Referenzmodell

Das Vorhandensein eines Referenzmodells vereinfacht und beschleunigt die Optimierungsphase des Neuronalen Netzes, da die Gewichte schon näherungsweise initialisiert werden können. Es muß nur ein Weg gefunden werden, wie aus dem Referenzmodell die relevanten Daten für den Lerndatensatz extrahiert werden können. Wenn die Daten des Referenzmodells in Form von Tabellen vorliegen, kann der Lerndatensatz durch Umformatieren erstellt werden. Falls die Netzwerkstruktur anders gestaltet wurde als die Struktur des Referenzmodells, müssen eventuell mehrere Parameter zusammengefaßt werden.

Idealerweise erstellt man für jede Gruppe des Neuronalen Netzes einen eigenen Lerndatensatz. Mit diesem wird die Gruppe sozusagen vortrainiert und in den Anfangszustand für die weitere Optimierung mit Meßdaten gebracht.

5.2.2 Lerndatensatz aus Flugversuchsdaten

Bei der klassischen Parameteridentifikation werden aus den Flugmeßdaten einzelne Abschnitte von der Länge eines Manövers ausgeschnitten. Die identifizierten Parameter

können dann auch nur dem Trimmzustand vor dem Manöver zugeordnet werden. Bei der Parameteridentifikation mit einem Neuronales Netz geht man einen anderen Weg, da nun ein nichtlineares Modell erstellt wird. Eine Identifikation nur eines Manövers ergäbe eine spezielle Anpassung der Gewichte für das eine Manöver und gleichzeitig eine Verschlechterung für ein anderes Manöver. Deswegen wird aus allen verfügbaren Manövern ein großer Lerndatensatz erstellt.

Da dieser große Lerndatensatz schon mal mehr als 100.000 Lernmuster enthalten kann, bietet es sich an, die Daten zu komprimieren. Dazu wird ein Raster über den Eingaberaum gelegt und Lernmuster, die zu den gleichen Stützstellen gehören, werden gemittelt. Mehrere Tests zeigten, daß das Ergebnis der Optimierung in beiden Fällen (komprimiert, unkomprimiert) gleich gut war. Die Lerndauer war aber mit dem komprimierten Lerndatensatz erheblich kürzer.

Ein weiterer Vorteil des komprimierten Lerndatensatzes liegt auch in der gleichmäßigen Verteilung der Lernmuster. Die Mehrzahl der Meßdaten während eines Testflugs können Geradeausflügen zugeordnet werden. Die Minderheit der Daten entspricht dynamischen Manövern, in denen die ganze Systeminformation steckt.

5.2.3 Differenz- Lerndatensatz

In einigen Fällen ist es sinnvoll ein Differenzmodell zu erstellen, und damit Differenzwerte zu identifizieren. Zum Beispiel dann, wenn ein Großteil der aerodynamischen Derivative nicht identifiziert werden kann, weil zu wenig Information in den Meßdaten enthalten ist. Oder, weil die Gesamtwerte nicht interessieren, sondern die Korrekturen eines Referenzmodells ermittelt werden sollen. Für diesen Zweck erstellt man einen Differenz- Lerndatensatz:

$$\Delta C_A = C_{A_{FT}} - C_{A_{Ref}}$$

Index *FT*: Berechnet aus Meßdaten

Index *Ref*: Aus einem aerodynamischen Referenzmodell entnommen

Es ist auch möglich einen Lerndatensatz aus Flugversuchsdaten für eine Untermenge von Derivativen zu erstellen. Zum Beispiel für $C_{A\eta}$:

$$C_{A\eta} = C_{A_{FT}} - \left(C_{A_{Ref}} - C_{A\eta_{Ref}} \right)$$

Implizit begeht man hier den Fehler, etwaige Defizite der anderen Parameter mit in $C_{A\eta}$ zu stecken. Man muß hier genau überlegen, ob dieser Fehler vernachlässigbar ist.

5.3 Initialisierung

Die Verbindungsgewichte sind die freien Parameter eines Neuronales Netzes. Sofern diesen Gewichten keine physikalische Bedeutung zugeordnet werden kann, müssen sie mit Zufallswerten zum Beispiel im Bereich $[-1;1]$ initialisiert werden. Würden die Gewichte eines mehrschichtigen Netzes mit dem Werte Null initialisiert, dann würde der Backpropagation Algorithmus beim Training als Gewichtsänderung immer den Wert Null für alle Verbindungen liefern. Eine Fehlerminimierung wird dadurch blockiert.

Neuronale Netze vom Typ *MLP* werden also zu Beginn mit Zufallswerten für die Gewichte initialisiert. Auch die oben angesprochenen Gruppen können als *MLP* realisiert werden, und müssen auf die erwähnte Art initialisiert werden.

Die einzelnen Gruppen können dann mit Hilfe eines Lerndatensatzes, den man aus einem Referenzmodell erstellen kann, weitertrainiert werden. Wie dieser Lerndatensatz erstellt wird, wurde in Abschnitt 5.2.1 gezeigt. Das Ergebnis sieht dann wie in Abbildung 5.4 aus.

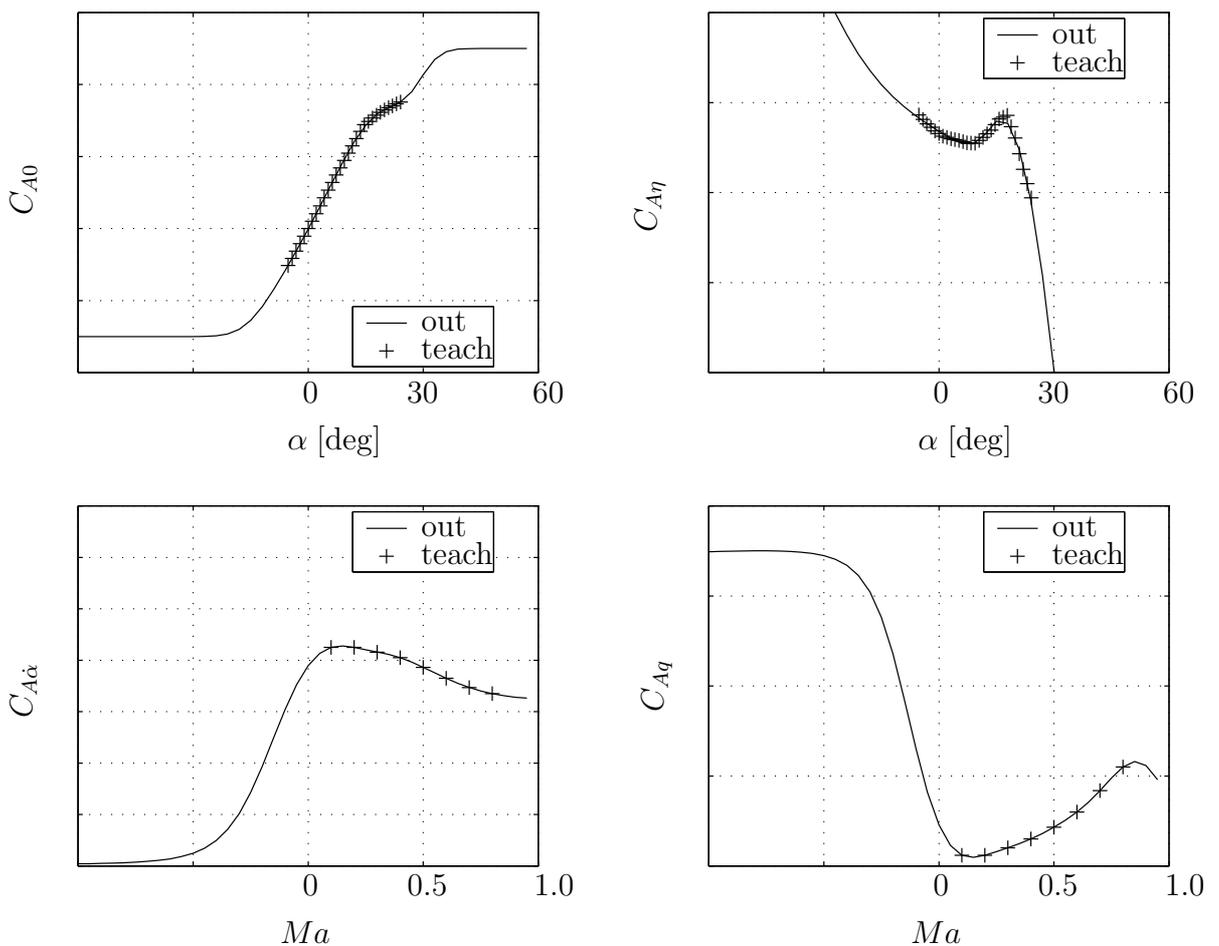


Abbildung 5.4: Initialisierung der Gewichte wird durch isoliertes Training der Gruppen ermöglicht

Für vier unterschiedliche Gruppen wurden Lerndatensätze mit einer Eingangsgröße (α bzw. Ma) erstellt. Die Gruppen wurden dann isoliert trainiert, bis der Fehler eine vorher definierte Schranke unterschritt.

Es gibt allerdings einen Netzwerktyp, der sich auch sehr gut für die Gruppen eignet und einen großen Vorteil gegenüber dem *MLP* besitzt. Neuronale Netze mit einer Schicht von Neuronen mit der Radialbasis Funktion *RBF* können mit direkter Gewichtsrechnung initialisiert und später mit Backpropagation weitertrainiert werden. Eine detaillierte Erläuterung zur Initialisierung von *RBF* Netzen findet man in [4, 5].

Nach der Initialisierung stellt das Neuronale Netz ein Basismodell dar. Davon ausgehend, wird dann das gesamte Netz mit einem Lerndatensatz aus Meßdaten weitertrainiert.

5.4 Optimierungsalgorithmus Backpropagation

Bekanntermaßen konvergiert der Backpropagation- Algorithmus sehr langsam. Denn die Gewichtsänderungen, und somit auch die Lerngeschwindigkeit, sind proportional zum Netzwerkfehler.

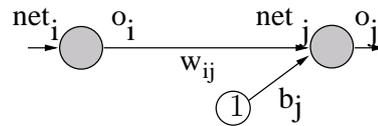
Für den Fall, daß die Hyperfläche des gewichteabhängigen Fehlers arg zerklüftet ist, eignet sich ein Gradientenverfahren weniger als zum Beispiel globale Suchalgorithmen [28].

5.4.1 Charakteristik

Ein neuronales Netz stellt eine Funktion f mehrerer Parameter (Gewichte und Schwellenwerte) dar. Zur Bestimmung der Parameter muß die Umkehrfunktion berechnet werden. Dies ist aber bei mehrschichtigen Neuronalen Netzen analytisch nicht möglich. Deswegen bedient man sich iterativer Optimierungsalgorithmen. Der Backpropagation Algorithmus eignet sich für mehrstufige Netze mit nichtlinearen Aktivierungsfunktionen. Das Problem bei mehrstufigen Netzen besteht darin, daß keine Lerneingabe für die Zellen verdeckter Schichten vorhanden ist.

Das Verhalten des Backpropagation Algorithmus ändert sich mit der Lernrate. Bei zu großer Lernrate springen die Gewichte zu stark hin und her, eine Konvergenz wird nicht erreicht. Eine zu kleine Lernrate bedeutet aber lange Trainingszeiten.

5.4.2 Algorithmus



Verdecktes Neuron Ausgabe Neuron

Abbildung 5.5: Verknüpfung zwischen Neuronen zweier Schichten

Der Fehler, den ein Neuronales Netz liefert, ist an seinem Ausgang meßbar. Es wird jedoch nicht einfach der Abstand zwischen Sollwert und Istwert berechnet, sondern eine Fehlerfunktion definiert. Als solche wird häufig der quadratische Abstand zwischen erwarteter Ausgabe t_j und realer Ausgabe o_j verwendet. Die Summe der Fehler an allen Ausgabeneuronen ist definiert zu:

$$E = \frac{1}{2} \sum_j (t_j - o_j)^2 \quad (5.20)$$

Zur Minimierung des Fehlers untersucht man dessen Abhängigkeit von den zu modifizierenden Verbindungsgewichten. Diese werden dann um einen Bruchteil des negativen Gradienten der Fehlerfunktion E verändert. Der Faktor η in (5.21) wird als Lernfaktor oder Schrittweite bezeichnet.

$$\Delta w_{ij} = -\eta \cdot \frac{\delta E}{\delta w_{ij}} \quad (5.21)$$

Die Fehlerfunktion E kann nicht unmittelbar nach w_{ij} abgeleitet werden. Hier erweist sich die Anwendung der Kettenregel für die Berechnung des Gradienten als notwendig.

$$\frac{\delta E}{\delta w_{ij}} = \frac{\delta E}{\delta net_j} \cdot \frac{\delta net_j}{\delta w_{ij}} \quad (5.22)$$

Die Propagierungsfunktion gibt an, wie die Nettoeingabe net_j einer Zelle der Schicht j berechnet wird. Sie ist die Summe aus den Produkten der Gewichte w_{ij} und der Ausgabe o_i der Zellen der vorderen Schicht i , und eventuell auch einem Schwellenwert (*bias*) b_j (siehe auch Abb. 5.5).

$$net_j = \sum_i o_i w_{ij} + b_j \quad (5.23)$$

Für eine Zelle mit nichtlinearer Aktivierungsfunktion f gilt:

$$o_j = f(net_j) \quad (5.24)$$

Aus Gleichung (5.23) ergibt sich für den zweiten Faktor aus (5.22)

$$\frac{\delta net_j}{\delta w_{ij}} = \frac{\delta}{\delta w_{ij}} \sum_i o_i w_{ij} + b_j = o_i$$

Der erste Faktor aus (5.22) wird als Fehlersignal definiert und mit $-\epsilon$ substituiert. Zur Berechnung dieses Faktors wird erneut die Kettenregel angewendet.

$$\epsilon_j = -\frac{\delta E}{\delta net_j} = -\frac{\delta E}{\delta o_j} \cdot \frac{\delta o_j}{\delta net_j} \quad (5.25)$$

Für den zweiten Faktor aus (5.25) folgt aus Gleichung (5.24)

$$\frac{\delta o_j}{\delta net_j} = \frac{\delta}{\delta net_j} f(net_j) = f'(net_j)$$

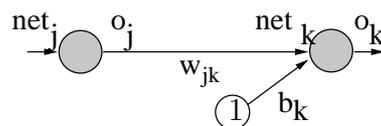
Das ist also die erste Ableitung der Aktivierungsfunktion f von Neuron j . Zu bestimmen ist jetzt noch der erste Faktor aus Gleichung (5.25). Für diesen Faktor lassen sich zwei Fälle unterscheiden:

1. j ist Index einer Ausgabezelle
2. j ist Index einer Zelle der verdeckten Schichten

Im ersten Fall ergibt sich nach Ableitung der Fehlerfunktion E aus (5.20)

$$-\frac{\delta E}{\delta o_j} = -\frac{\delta}{\delta o_j} \frac{1}{2} \sum_j (t_j - o_j)^2 = (t_j - o_j)$$

Im zweiten Fall kann die partielle Ableitung nur indirekt berechnet werden.



Verdecktes Neuron Ausgabe Neuron

Abbildung 5.6: Das Neuron j in der verdeckten Schicht

$$-\frac{\delta E}{\delta o_j} = -\sum_k \frac{\delta E}{\delta net_k} \cdot \frac{\delta net_k}{\delta o_j} = \sum_k (\epsilon_k \cdot \frac{\delta}{\delta o_j} \sum_j o_j w_{jk} + b_k) = \sum_k \epsilon_k w_{jk}$$

Dies bedeutet, daß man den Fehler der Zelle j in diesem Fall aus den gewichteten Fehlern ϵ_k aller Nachfolgezellen k und der Verbindungsgewichte von Zelle j zu

den Zellen k berechnen kann. Der Name Backpropagation steht für diese Art der Berechnung der Aktivierungsfehler von verdeckten Zellen. Somit ergibt sich insgesamt als Fehlersignal ϵ_j :

Für den ersten Fall (j ist Ausgabezelle)

$$\epsilon_j = f'(net_j) \cdot (t_j - o_j) \quad (5.26)$$

Für den zweiten Fall (j ist verdeckte Zelle)

$$\epsilon_j = f'(net_j) \cdot \sum_k \epsilon_k w_{jk} \quad (5.27)$$

Der Gradient der Fehlerfläche berechnet sich somit nach (5.22) folgendermaßen:

$$\frac{\delta E}{\delta w_{ij}} = -\epsilon_j \cdot o_i$$

Zusammengefaßt ergibt sich aus (5.21) folgende Backpropagation Regel zur Gewichtsänderung:

$$\boxed{\Delta w_{ij} = \eta \cdot o_i \cdot \epsilon_j} \quad (5.28)$$

Für die Biasänderung vereinfacht sich diese Gleichung, da $o_i = 1.0$.

$$\boxed{\Delta b_j = \eta \cdot \epsilon_j} \quad (5.29)$$

5.4.3 Erforderliche Modifikation

Das Verfahren zum Training eines Neuronalen Netzes mit dem Backpropagation Algorithmus wurde etwas abgeändert, so daß die neue Struktur (siehe Abb. 4.9) berücksichtigt wird. Die Vorgehensweise läuft nun folgendermaßen ab:

1. Die Werte für die Eingänge des Netzwerks werden aus einem Lerndatensatz übernommen.
2. Die Aktivierung der Eingabeneuronen und die untrainierbaren Eingabeverbindungen werden mit diesen Werten versehen.
3. Das Neuronale Netz wird beginnend von der Eingangsschicht bis zur Ausgangsschicht berechnet. Dazu wird die Aktivierung jedes Neurons abhängig von den Vorgängerneuronen und den Verbindungen berechnet.
4. Die Aktivierung des Ausgabeneurons wird mit dem Sollwert aus dem Lerndatensatz verglichen. Aus der Differenz wird der Netzwerkfehler berechnet (siehe Gleichung 5.20).

5. Aus dem Netzwerkfehler wird schichtweise von hinten nach vorne der Fehler an jedem Neuron berechnet (siehe Gleichung 5.27).
6. Abhängig von dem Fehler eines Neurons und dessen Aktivierung wird eine Gewichtsänderung der Verbindungen zu den benachbarten Neuronen berechnet (siehe Gleichung 5.28). Ausgenommen sind hierbei die untrainierbaren Eingabeverbindungen und die untrainierbaren Verbindungen ($w = 1.0$).
7. Zusätzlich zu den Gewichtsänderungen werden noch die Biasänderungen gemäß Gleichung 5.29 berechnet. Ausgenommen sind alle Neuronen der Ein- und Ausgangsschicht mit linearer Aktivierungsfunktion, da sie den Eingangswert unverändert wieder ausgeben sollen.
8. Abschließend werden die Gewichtsänderungen zu den jeweiligen Gewichten, und die Biasänderungen zu den jeweiligen Biaswerten addiert.

5.4.4 Beispiel

Um die Funktionsweise des Backpropagation Algorithmus mit Eingabeverbindungen anschaulich zu gestalten, folgt ein kleines idealisiertes Beispiel. Es werden zwei Neuronen mit linearer Aktivierungsfunktion für die Derivative $C_{A\alpha}$ und $C_{A\eta}$ verwendet. Diese werden mit den Eingabeverbindungen α und η multipliziert, und im Ausgabeneuron C_A summiert (siehe Abb. 5.7).

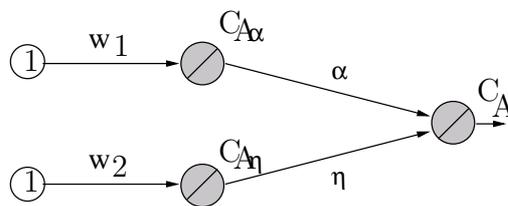


Abbildung 5.7: Das Modell $C_A = C_{A\alpha} \cdot \alpha + C_{A\eta} \cdot \eta$

Die Gewichte w_1 und w_2 müssen durch die Optimierung auf die Zielwerte $w_1 = 2.0$ und $w_2 = 1.5$ gebracht werden. Dazu dient der Lerndatensatz, bestehend aus drei Patterns (siehe Tabelle 5.1). Die Werte von t_{C_A} sind die Sollwerte für die Ausgabe des Netzes.

Pattern	α	η	t_{C_A}
1.	0.0	-1.0	-1.5
2.	1.0	0.0	2.0
3.	1.0	-1.0	0.5

Tabelle 5.1: Lerndatensatz

Zu Beginn der Optimierung seien die Gewichte mit den Werten $w_1 = 3.0$ und $w_2 = 2.0$ initialisiert. Das ergibt für die Aktivierung der Neuronen der verdeckten Schicht: $C_{A\alpha} = 3.0$ und $C_{A\eta} = 2.0$.

$$C_A = 3.0 \cdot 0.0 + 2.0 \cdot (-1.0) = -2.0$$

Dieser Anfangszustand ist in Abbildung 5.8 zu sehen.

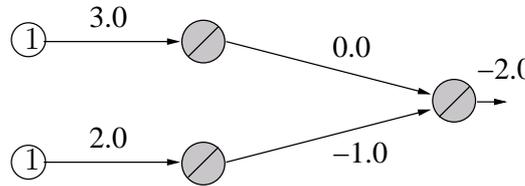


Abbildung 5.8: Anfangszustand

Das Ergebnis für C_A lautet -2.0 . Aus der Tabelle 5.1 entnimmt man für den Sollwert $t_{C_A} = -1.5$. Für den Fehler am Ausgabeneuron ϵ_{C_A} erhält man mit Gleichung (5.26)

$$\epsilon_{C_A} = f'_{C_A} \cdot (t_{C_A} - C_A) = 1.0 \cdot (-1.5 - (-2.0)) = 0.5$$

Die erste Ableitung f' der linearen Aktivierungsfunktion f hat für alle drei Neuronen den Wert 1.0. Die Fehler an den verdeckten Neuronen berechnet man mit der Gleichung (5.27).

$$\begin{aligned} \epsilon_{C_{A\alpha}} &= f'_{C_{A\alpha}} \cdot \epsilon_{C_A} \cdot \alpha = 1.0 \cdot 0.5 \cdot 0.0 = 0.0 \\ \epsilon_{C_{A\eta}} &= f'_{C_{A\eta}} \cdot \epsilon_{C_A} \cdot \eta = 1.0 \cdot 0.5 \cdot (-1.0) = -0.5 \end{aligned} \quad (5.30)$$

Besonders hervorheben muß man hier, daß der Fehler $\epsilon_{C_{A\alpha}}$ Null ist. Denn dieses Derivativ hat auch wegen $\alpha = 0.0$ keinen Anteil am Gesamtwert C_A gehabt. Mit der Gleichung (5.28) erhält man für die Gewichtsänderungen (Lernrate $\eta = 1.0$) :

$$\begin{aligned} \Delta w_1 &= o_1 \cdot \epsilon_{C_{A\alpha}} = 1.0 \cdot 0.0 = 0.0 \\ \Delta w_2 &= o_2 \cdot \epsilon_{C_{A\eta}} = 1.0 \cdot (-0.5) = -0.5 \end{aligned} \quad (5.31)$$

Das Gewicht w_1 wird also nicht verändert, weil durch die Eingabeverbindung ($\alpha = 0.0$) der Fehler mit Null gewichtet wurde. Die Zwischenwerte betragen jetzt für $w_1 = 3.0$ und $w_2 = 1.5$. Mit dem zweiten Pattern wird nun auch noch $C_{A\alpha}$ optimiert (Abb. 5.9).

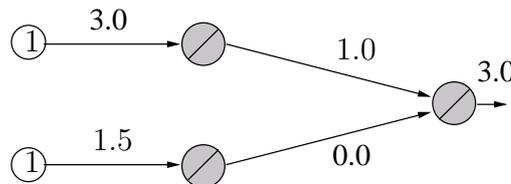


Abbildung 5.9: Nach dem Training des ersten Patterns

$$C_A = 3.0 \cdot 1.0 + 1.5 \cdot 0.0 = 3.0$$

Mit Gleichung (5.26) und (5.27) erhält man wieder für die Aktivierungsfehler der Neuronen:

$$\epsilon_{C_A} = 1.0 \cdot (2.0 - 3.0) = -1.0$$

$$\epsilon_{C_{A\alpha}} = 1.0 \cdot (-1.0) \cdot 1.0 = -1.0$$

$$\epsilon_{C_{A\eta}} = 1.0 \cdot (-1.0) \cdot 0.0 = 0.0$$

(5.32)

Mit Gleichung (5.28) lauten die Gewichtsänderungen:

$$\Delta w_1 = o_1 \cdot \epsilon_{C_{A\alpha}} = 1.0 \cdot (-1.0) = -1.0$$

$$\Delta w_2 = o_2 \cdot \epsilon_{C_{A\eta}} = 1.0 \cdot 0.0 = 0.0$$

(5.33)

Die Werte für die Derivative lauten nun: $C_{A\alpha} = 2.0$, $C_{A\eta} = 1.5$. Mit dem dritten Pattern lassen sich diese Werte noch einmal überprüfen.

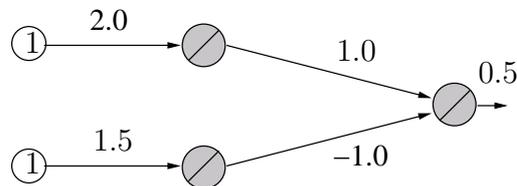


Abbildung 5.10: Endergebnis

Dieses idealisierte Beispiel zeigt die Funktionsweise der Optimierung. In der Praxis hingegen werden nichtlineare Aktivierungsfunktionen, wesentlich mehr Neuronen und verrauschte Meßdaten verwendet. Auch unter diesen Bedingungen arbeitet das Optimierungsverfahren stabil. Die Resultate sind in Kapitel 6 zusammengefaßt.

5.4.5 Rechenzeit

Gradientenabstiegsverfahren haben eine Geschwindigkeit, die proportional zum gemessenen Fehler ist. Deswegen werden anfangs bei großen Fehlern auch schnell erkennbare Verbesserungen erzielt. Danach flacht die Lernkurve ab (siehe Abb. 6.9). Bei geringem Fortschrittsgrad oder bei einer vorher definierten Schranke wird das Verfahren abgebrochen.

In den Ingenieurwissenschaften ist meistens eine 90%- Lösung ausreichend. Aus diesem Grund, und weil eine geringe Rechengeschwindigkeit kein wirkliches K.O. Kriterium ist, wurden alle hier gezeigten Ergebnisse mit dem Backpropagation Algorithmus im *batch* Modus ermittelt. Allerdings wurde eine dynamische Lernrate verwendet (siehe Kap. 5.8.5).

5.4.6 Analysemöglichkeit des Modularen Neuronalen Netzes

Die Gruppen im Neuronalem Netz sind in diesem Fall kleinere Neuronale Netze mit Blackbox- Charakter. Durch die niedrige Dimension des Eingabevektors (üblicherweise bis zu drei Parameter: α, Ma, h) ist eine graphische Darstellung noch möglich. Dazu wird eine Gruppe aus dem gesamten Netz isoliert und berechnet. Die Eingabewerte werden in einem Raster perturbiert und das Netz bis zu seinem Ausgang berechnet. Die Ein- und Ausgabewerte werden in einer Tabelle abgelegt. So ist es auch möglich, die Funktion in 2D- oder 3D- Plots zu visualisieren. Auf diese Weise wurden alle Graphiken (wie z.B. Abb. 5.11) erstellt.

Zu sehen sind die Funktionsverläufe von vier Gruppen; jeweils Anfangszustand (init), das Ergebnis der Optimierung (out) und die Sollwerte (teach).

Nach [5] ist es auch möglich, die partiellen Ableitungen der Funktion nach einem Eingabeneuron zu bilden. So kann mit der gleichen Methode auch der Funktionengraph f' graphisch dargestellt werden, um die Funktion f auf Stetigkeit und Monotonie zu überprüfen.

5.5 Validierung des Verfahrens

Die Leistungsfähigkeit eines Modularen Neuronalem Netzes soll hier demonstriert werden, das wie in Abschnitt 4.2.2 strukturiert und mit einem Optimierungsverfahren gemäß Abschnitt 5.4 trainiert wird.

Zu diesem Zweck schafft man sich eine ideale Testumgebung. Das bedeutet, daß alle möglichen Fehlerquellen ausgeschlossen werden, die den Test behindern würden. Dazu gehören Sensorfehler, aber auch Verluste bei der Datenaufzeichnung. Der Vorteil einer künstlichen Umgebung liegt im Besonderen darin, daß man das Ziel der Optimierung kennt. Man weiß, welche Zahlenwerte die zu schätzenden Parameter annehmen müssen.

Dieser Test soll auf der einen Seite vereinfachte Randbedingungen haben, auf der anderen Seite aussagekräftig für reale Bedingungen sein. Es können bei diesem Test Vereinfachungen gemacht werden, soweit sie zulässig sind. So werden bei der Aufzeichnung eines simulierten Testfluges nicht nur die Bewegungsgrößen gespeichert, die sonst im echten Flugzeug gemessen werden, sondern zum Beispiel auch die Gesamtbeiwerte

aus dem aerodynamischen Modell. Dieses Vorgehen erübrigt die Berechnung dieser Parameter aus den Drehraten und Längsbeschleunigungen, wie es sonst erforderlich wäre.

Mit diesem Test sollen die Leistungsfähigkeit und die Grenzen der in Abschnitt 2.4 beschriebenen Methode unter Laborbedingungen aufgezeigt werden. Ein besseres Ergebnis, als hier dargestellt, wird man in der Praxis nicht erhalten.

$$C_A = C_{A0}(\alpha) + C_{A\eta}(\alpha) \cdot \eta + C_{A\dot{\alpha}}(Ma) \cdot \frac{\dot{\alpha}l}{2V} + C_{Aq}(Ma) \cdot \frac{ql}{2V} \quad (5.34)$$

Das aerodynamische Modell nach Gleichung (5.34) wurde aus folgenden Gründen gewählt:

1. Alle vier Parameter (C_{A0} , $C_{A\eta}$, $C_{A\dot{\alpha}}$, C_{Aq}) sind von α oder Ma nichtlinear abhängig. Die Fähigkeit zur Identifikation dieser Nichtlinearitäten wird hiermit geprüft.
2. Alle aerodynamischen Effekte haben unterschiedlich großen Einfluß auf den Gesamtbeiwert C_A . Hierdurch wird geprüft, wie gut während der Optimierung zwischen kleinen Effekten im Vergleich zu einflußreichen Effekten unterschieden wird.
3. Die Bewegungsgrößen $\dot{\alpha}$ und q korrelieren definitionsgemäß sehr stark. Zwischen $C_{A\dot{\alpha}}$ und C_{Aq} kann das Optimierungsverfahren nur schwer differenzieren.

Für das Modell wurden zwei unterschiedliche aerodynamische Parametersätze generiert, die jeweils einer echten Flugzeugkonfiguration ähnlich sind. Jeder Parametersatz enthält Tabellen für die vier Parameter C_{A0} , $C_{A\eta}$, $C_{A\dot{\alpha}}$ und C_{Aq} .

Der erste Parametersatz stellt den Simulatorendatensatz dar, der noch Defizite in der Repräsentation eines echten Flugzeugs haben soll. Er stellt den Ausgangszustand vor der Optimierung dar (Referenzmodell).

Das Neuronale Netz wurde mit diesen Daten initialisiert (siehe hierzu auch Abschnitt 5.3). In Abbildung 5.11 sind die nichtlinearen Verläufe dieser Parameter mit "init" bezeichnet (Basismodell).

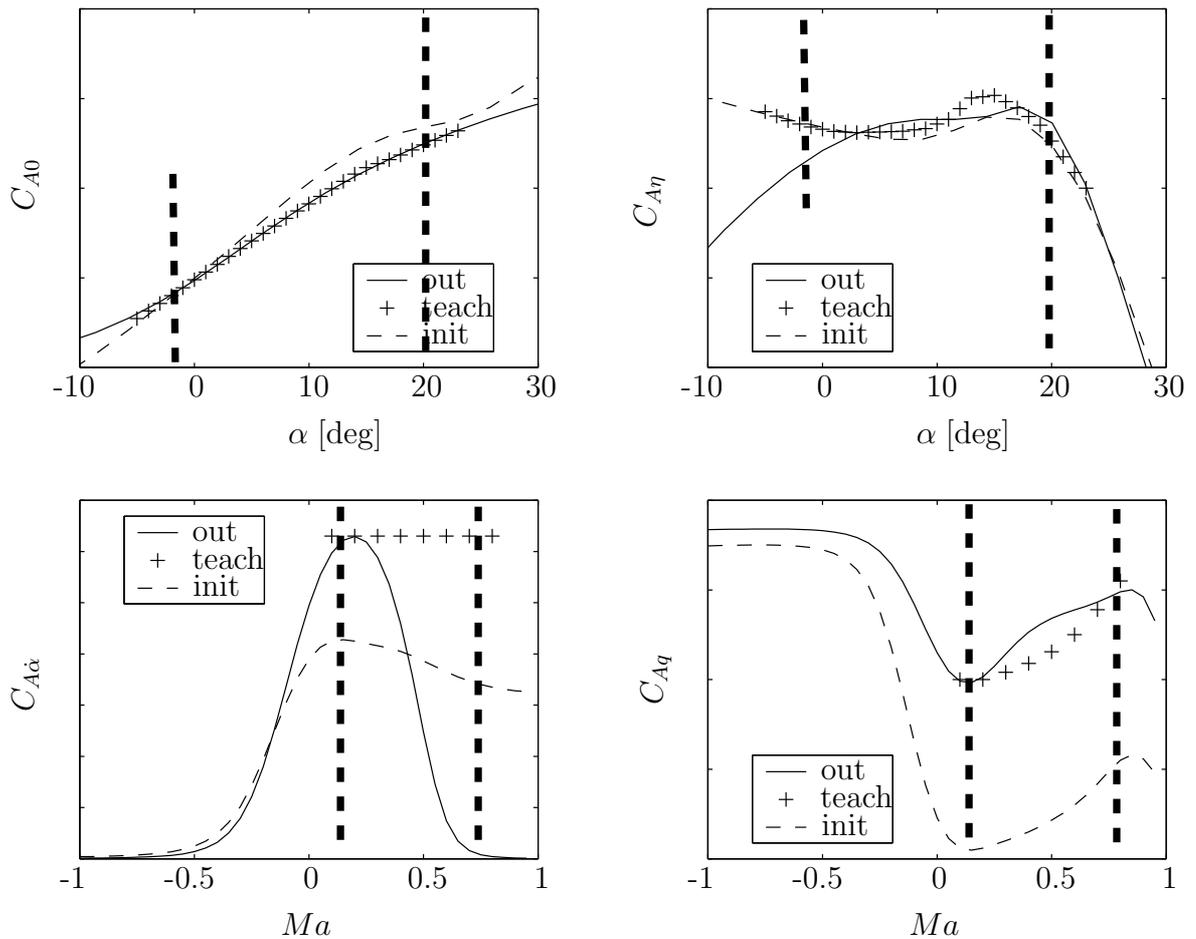


Abbildung 5.11: Mit simuliertem Lerndatensatz optimiertes MNN

Der zweite Parametersatz wurde erstellt um eine Zielvorgabe für die Parameteridentifikation zu besitzen. Dieser wurde in einen Flugsimulator integriert. Mit Hilfe dieses Flugsimulators wurden Flugversuche durchgeführt (simuliertes Idealsystem). Durch das Fliegen mehrerer dynamischer Manöver wurde, wie bei einem echten Testflug, eine Datenbasis für die Optimierung aufgezeichnet. Entsprechend dem Modell (Gleichung (5.34)) wurden die Eingabegrößen α , Ma , η , $\dot{\alpha}$, q und V aufgezeichnet. Die zugehörigen C_A -Werte wurden ebenfalls gespeichert. Die Referenzflügelgröße l ist bekannt. Aus diesen Meßdaten wird dann später ein Lerndatensatz erstellt, mit dem das bereits initialisierte Modulare Neuronale Netz optimiert werden soll.

Unabhängig von der Flugsimulation und der Optimierung, wurden aus diesem zweiten Parametersatz die einzelnen Anteile am Gesamtauftriebsbeiwert extrahiert. Sie sind zum Vergleich in Abbildung 5.11 mit “teach” bezeichnet. Natürlich werden diese Daten während des Lernvorgangs nicht verwendet! Der Lerndatensatz enthält nur die Werte des Gesamtbeiwertes C_A .

Für die eigentliche Testprozedur stehen nun zwei aerodynamische Datensätze zur

Verfügung (“init” und “teach”) sowie ein auf Flugversuchen basierender Lerndatensatz.

Das Modulare Neuronale Netz wurde nun mit dem Lerndatensatz im *batch mode*¹ trainiert. Der anfängliche mittlere Fehler (mean square error, *mse*) betrug $mse = 0.01$. Nach mehreren Tagen Rechendauer und etwa 2 Millionen Epochen wurde ein Ergebnis erzielt, wie es in Abbildung 5.11 mit “out” dargestellt ist. Der Fehler betrug dann nur noch $mse = 0.0002$.

Der Fortschritt der Fehlerminimierung war anfangs sehr groß und wurde, für das Gradientenverfahren typisch, zunehmend kleiner. Die größte Verbesserung wurde beim Verlauf von C_{A0} erzielt. Schon nach wenigen Epochen entsprach die Netzwerkausgabe den Sollwerten (“teach”).

Der Verlauf von $C_{A\eta}$ pendelte anfangs um die Sollwerte. Nach etwa 100.000 Epochen konvergierte die Ausgabe.

Die von der Machzahl abhängigen dynamischen Derivative $C_{A\dot{\alpha}}$ und C_{Aq} benötigten wesentlich mehr Rechenzeit um die dargestellten Verläufe zu bieten. Der geringe Anteil der $\dot{\alpha}$ - und q - Effekte ist dafür ausschlaggebend.

Es hatte auch den Anschein, daß durch die Korrelation von $\dot{\alpha}$ und q die Zuordnung des Netzwerkfehlers zu einem der beiden Derivative schwieriger war. Das Training wurde abgebrochen, als die Ausgaben dieser beiden Gruppen sich nur noch geringfügig änderten.

Bei der Betrachtung der Funktionsverläufe muß man bedenken, daß der Eingaberaum der Trainingsdaten nur einen Anstellwinkelbereich von $\alpha = [-2; 20]$ und einen Machzahlbereich zwischen 0.2 und 0.8 aufspannt. Diese Grenzen sind deswegen zusätzlich in den Graphiken mit senkrechten Balken markiert. Außerhalb dieses Bereichs, sind die optimierten Parameter ungültig.

Es ist möglich, daß die Netzwerkgewichte das globale Minimum finden, falls noch weiter trainiert wird. Aber dies würde unendlich lange dauern.

Nebenbei ist in Abbildung 5.12 das theoretisch bestmögliche Ergebnis einer Optimierung mit einem speziellen Lerndatensatz zu sehen. Dieser Lerndatensatz zeichnet sich dadurch aus, daß die Eingangsgrößen nicht miteinander korrelieren. Er ist dadurch zwar absolut praxisfremd, denn die Bewegungsgrößen α und η korrelieren im Fluge immer stark. Aber man sieht durch den Vergleich von Abbildung 5.11 und Abbildung 5.12 auch, welche Probleme durch diese Korrelation entstehen.

¹batch mode: Die Gewichtsänderungen werden erst nach Präsentation aller Lernmuster durchgeführt

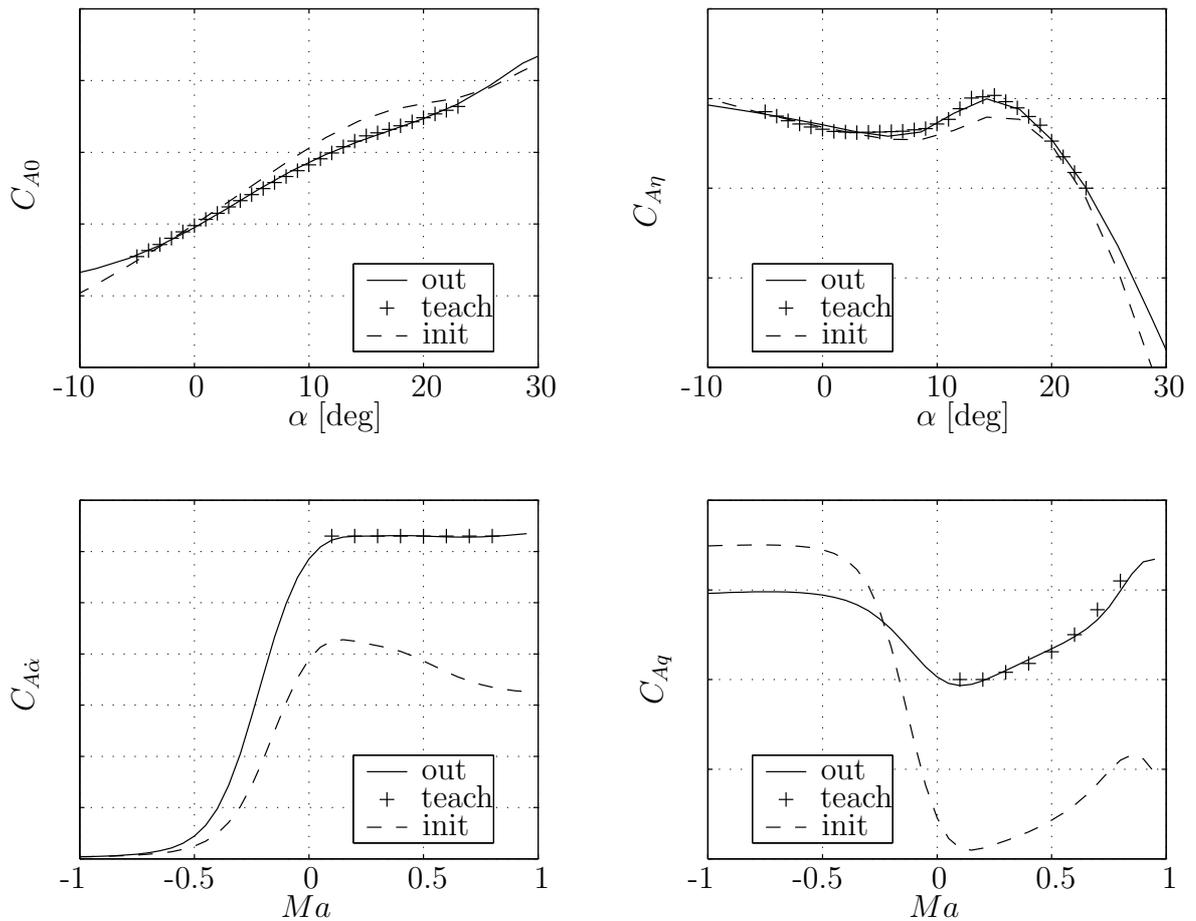


Abbildung 5.12: Bestmögliches Ergebnis der Optimierung

Bezüglich der langen Lerndauer muß man auch bedenken, welchen Nutzen ein langes Training bringt. Die C_A -Anteile, die einen großen Beitrag zum Gesamtbeiwert liefern, werden beim Training bevorzugt optimiert. So ist dann auch schon nach wenigen Epochen ein akzeptables Ergebnis zu erwarten. In Abbildung 5.13 sieht man, daß die beiden dynamischen Anteile zusammen nur etwa 1 % zum Gesamtauftrieb beitragen. Die Qualität der identifizierten Parameter wurde bis jetzt aus aerodynamischer Sicht beurteilt. Mit Hilfe des Flugsimulators können die Identifikationsergebnisse aus flugmechanischer Sicht betrachtet werden.

In den Flugsimulator wurden drei unterschiedliche aerodynamische Datensätze integriert:

1. Fehlerbehafteter Anfangsdatsatz (Basismodell, "init")
2. Mit Backpropagation optimierte Parameter (Optimiertes Modell, "out")
3. Datensatz für Vergleichszwecke ("teach")

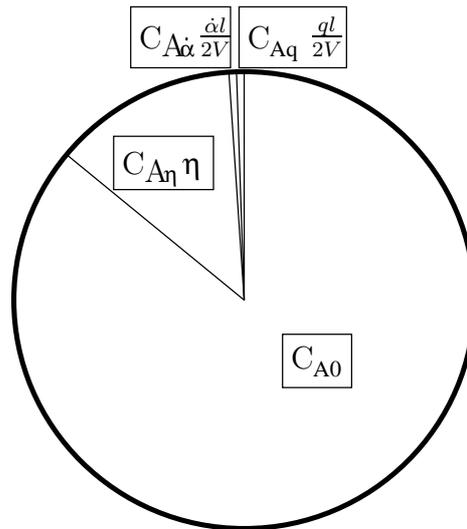


Abbildung 5.13: Prozentuale Anteile am Gesamtauftrieb

Mit jedem Datensatz wurde ein pitch doublet Manöver durchgeführt (siehe auch Abbildung 3.1). Das Ergebnis dieser Simulationen ist in Abbildung 5.14 zu sehen.

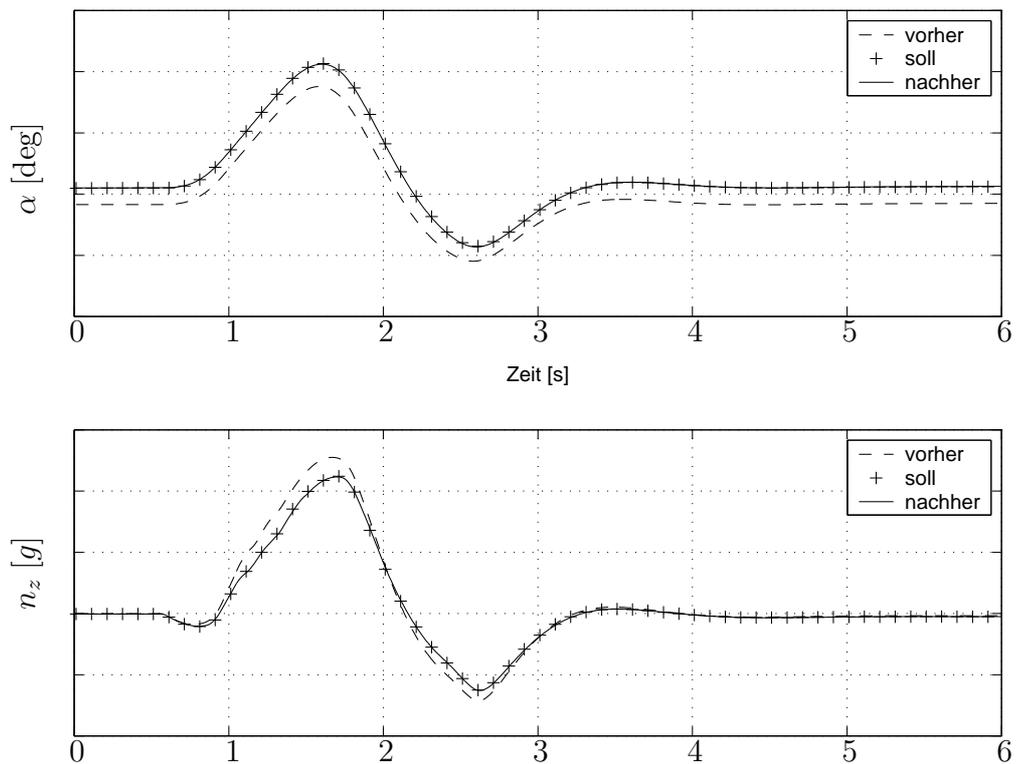


Abbildung 5.14: Vergleich vor und nach der Optimierung

Die Simulation mit Datensatz 1 (Basismodell) ist mit “vorher” gekennzeichnet. Dieses Ergebnis liefert der Simulator vor der Optimierung der aerodynamischen Parameter. Eine Wiederholung des gleichen Manövers mit dem Datensatz 2 (Optimiertes Modell) zeigt eine deutliche Veränderung in den Zeitverläufen (“nachher”). Die Maximalwerte der Bewegungsgrößen α und n_z stimmen nun sehr gut mit den als “soll” gekennzeichneten Werten überein. Diese Sollwerte wurden durch Simulation mit Datensatz 3 erzeugt.

Der Flugsimulator mit dem optimierten aerodynamischen Modell verhält sich jetzt aus flugmechanischer Sicht wie das Original.

5.6 Informationsgehalt des Lerndatensatzes

Die Güte der identifizierten Parameter hängt direkt vom Informationsgehalt des Lerndatensatzes ab. Allzu häufig wird bei der Identifizierung von hochgradig nichtlinearen Systemen übersehen, daß die berechneten Ergebnisse aufgrund der statistischen Problemnatur immer mit Unsicherheiten verbunden sind. Je größer die Störungen der experimentellen Beobachtung sind, desto größer kann die Abweichung der Schätzung von den wahren Parametern sein. Es reicht somit keinesfalls aus, nur das reine Schätzproblem zu lösen. Vielmehr ist die Genauigkeit und Zuverlässigkeit des identifizierten Modells zu bestimmen und zu entscheiden, ob es seinem Zweck entsprechend eingesetzt werden kann.

Verwendet man zum Beispiel bei den klassischen PID- Methoden das Gauß-Newton Verfahren, so können mit Hilfe der Iterationsmatrix Aussagen über die Genauigkeit und die Korrelation der Parameterschätzung abgeleitet werden [24].

Einen groben Überblick über den Informationsgehalt des Lerndatensatzes kann man sich mit XY- Plots verschaffen. Nachdem unser einfaches Modell von vorhin (5.34) von den zwei Eingangsgrößen α und Ma nichtlinear abhängig ist, gibt eine Darstellung wie Abbildung 5.15 erste Erkenntnisse über die Anzahl der gemessenen Zustandsgrößen. Die Farbcodierung bezieht sich auf eine normierte Menge von Meßwerten; schwarze Stellen bedeuten am meisten, helle Kästchen sehr wenige bzw. keine Meßwerte. Wie es für einen Flug typisch ist, erhält man bei hohen Geschwindigkeiten viele Meßwerte mit kleinen Anstellwinkeln, und bei niedrigen Geschwindigkeiten entsprechend viele Meßwerte mit hohen Anstellwinkeln. Insgesamt liegen in diesem Beispiel im $\alpha - Ma$ Bereich genügend Meßdaten vor.

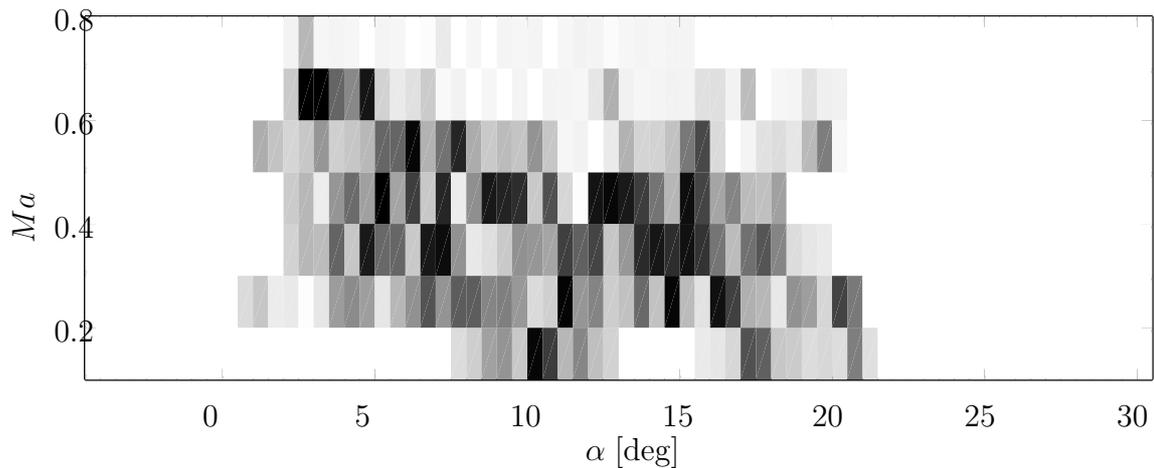


Abbildung 5.15: Anstellwinkel - Machzahl Bereich

Mit Hilfe der Korrelationsanalyse kann die Stärke eines linearen Zusammenhangs zwischen zwei Variablen beschrieben werden.

	α	Ma	η	$\dot{\alpha}$	q
α	1.00	-0.42	-0.24	-0.10	0.19
Ma	-0.42	1.00	0.19	-0.08	0.20
η	-0.24	0.19	1.00	-0.17	-0.07
$\dot{\alpha}$	-0.10	-0.08	-0.17	1.00	0.76
q	0.19	0.20	-0.07	0.76	1.00

Tabelle 5.2: Korrelationsmatrix

Die negative Korrelation zwischen Anstellwinkel und Machzahl ist für Flugzeuge typisch und bedarf keiner weiteren Erläuterung. Die positive Korrelation zwischen Anstellwinkelrate $\dot{\alpha}$ und Nickrate q ist definitionsgemäß groß. Eine Differenzierung der Effekte mit Hilfe der Parameteridentifikation ist, wie wir eben gesehen haben, schwierig.

Eine besondere Rolle spielen die linearen Eingabeverbindungen bei Modulen neuronalen Netzwerken. Wenn diese Parameter betragsmäßig kleine Werte annehmen oder wenn die Abweichungen untereinander gering sind, vergrößert sich die Lerndauer. Betragsmäßig kleine Werte der Eingabeverbindungen verringern den rückwärtspropagierten Fehler. Der Lernfortschritt wird dadurch proportional kleiner. Eine große Korrelation zwischen Eingabeverbindungen (hier zum Beispiel $\dot{\alpha}$ und q) erlaubt keine eindeutige Zuweisung des Netzwerkfehlers zu einer Gruppe.

5.7 Einschränkungen

Als Aktivierungsfunktion der Neuronen wurde die sigmoide Funktion \tanh favorisiert. Dadurch kann eine Neuronengruppe auch eine Funktion mit mehreren Wendepunkten,

entsprechend der Anzahl an Neuronen, approximieren. Große Gradientenänderungen können nur näherungsweise realisiert werden.

5.8 Hinweise aus der Praxis

Dieser Abschnitt soll auf Probleme aufmerksam machen, die die Anwendung des hier verwendeten Optimierungsverfahrens mit sich bringen kann. Eventuell lassen sich dadurch Fehler vermeiden. Vielleicht helfen auch die folgenden Hinweise, die auf vielen Tests beruhen, bessere Ergebnisse zu erzielen.

5.8.1 Unpassender Lerndatensatz

Der Lerndatensatz für das Modulare Neuronale Netz muß unbedingt Werte für alle Eingabeneuronen und Eingabeverbindungen enthalten. Die Modellstruktur und der Lerndatensatz müssen zusammenpassen.

Wenn mehrere Trainingsdatensätze zur Optimierung verwendet werden, muß sichergestellt sein, daß sie auch Meßdaten von dem gleich konfigurierten System enthalten.

Natürlich muß im Datensatz auch genügend Information enthalten sein, damit das charakteristische Ein- Ausgabeverhalten auch modelliert werden kann.

5.8.2 Zu große Lernrate

Bei zu hoher Lernrate besteht die Gefahr, daß sehr große Sprünge bei den Gewichtsänderungen gemacht werden. Ein Gradientenabstieg ist somit unmöglich. Der Netzwerkfehler kann dadurch stark ansteigen und für die Gewichte erhält man dann riesige Zahlenwerte. Üblicherweise bewegen sie sich, bei Neuronen mit der Aktivierungsfunktion *tanh*, in einem Bereich von etwa ± 10 .

Der Lernalgorithmus konvergiert nicht bei zu großer Lernrate. Dies macht sich aber schon zu Beginn der Trainingsphase durch eine stark ansteigende Lernkurve bemerkbar. Leider kann man keine Faustregel für die Größenordnung der Lernrate geben. Sie hängt stark von der Art der Lerndaten und der Initialisierung der Verbindungsgewichte ab.

5.8.3 Falscher Wertebereich der Gruppen

Wie in Abschnitt 4.2.1.4 erwähnt, kann für jede Gruppe ein eigener Wertebereich definiert werden. Falls dies nicht getan wird, besitzt eine Gruppe den selben Wertebereich wie das Ausgabeneuron dieser Gruppe. Ist zum Beispiel die Obergrenze zu niedrig angesetzt, dann versucht der Optimierungsalgorithmus durch Vergrößern der Verbindungsgewichte das Maximale aus den Neuronen herauszuholen. Das Ausgabeneuron

liefert dann im Extremfall für unterschiedliche Eingabewerte nur noch konstante Ausgabewerte.

Dieses Problem bemerkt man schon frühzeitig durch Beobachtung der Funktionsverläufe aller Gruppen, während der Optimierung. Das Training muß dann mit einem geänderten Wertebereich wiederholt werden. Als Grundregel sollte man den Wertebereich lieber zu groß als zu klein wählen.

5.8.4 Numerische Instabilität

Hier noch ein Hinweis zur Softwareerstellung. Bei der Verwendung von Fließkommazahlen mit einfacher Genauigkeit (32 bit) kann es zu numerischer Instabilität kommen. Vor allem nach langer Trainingszeit und geringem Netzwerkfehler werden die Zahlenwerte für die Gewichtsänderungen sehr klein. Diese können eventuell nicht mehr numerisch dargestellt werden. Es besteht dann die Gefahr, daß die Gewichte anfangen zu springen. Die Sprünge können sich auf alle Gewichte des Neuronalen Netzes ausweiten, so daß sich letztendlich das ganze Netz aufschaukelt. Abhilfe schafft eine Umstellung auf doppelt genaue Fließkommadarstellung mit 64 bit.

5.8.5 Dynamische Lernrate η_{dyn}

Oben wurde erläutert, welche negativen Auswirkungen eine zu große Lernrate haben kann. Deswegen sollte lieber ein kleiner Zahlenwert gewählt werden. Der daraus resultierende Nachteil ist aber dann eine zu lange Trainingsdauer.

Bei den Versuchen wurde erfolgreich eine dynamische Lernrate verwendet. Der Fehler von einer Epoche zur nächsten Epoche war entscheidend dafür, ob die Lernrate verkleinert oder vergrößert wurde. Falls der Fehler anstieg, wurden die letzten Gewichtsänderungen rückgängig gemacht und mit einer kleinen Lernrate weitertrainiert.

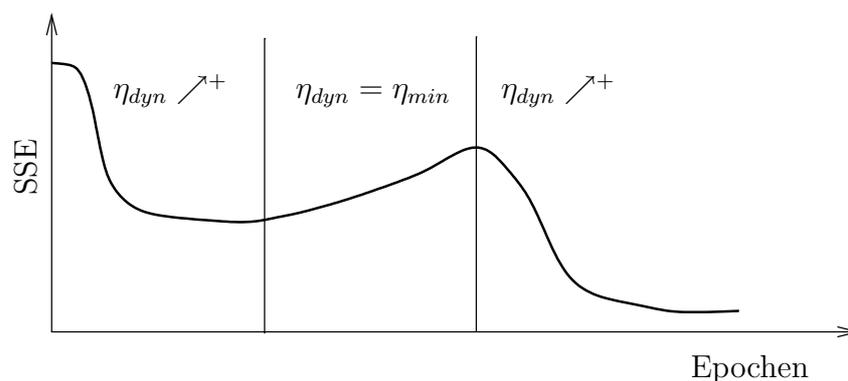


Abbildung 5.16: Dynamische Lernrate abhängig vom Netzwerkfehler (Summed Square Error, SSE)

5.8.6 Einzelne Gruppe initialisieren

Jede Gruppe in einem Modularen Neuronalem Netz stellt für sich ein kleines Netz dar. Während des Trainings kann es vorkommen, daß eine Gruppe sich in einem lokalen Minimum befindet. Stellt man dies durch Beobachten des Funktionsverlaufes fest, dann kann man auch nur diese einzelne Gruppe neu mit Zufallswerten initialisieren. Das Training wird daraufhin wieder fortgesetzt.

6 Analyse der aerodynamischen Derivative aus Flugversuchsdaten

Die nun folgenden Ergebnisse basieren auf einer Parameteridentifikation mit zwei Neuronalen Netzen, deren Struktur in den Abbildungen 6.1 und 6.2 dargestellt sind. Für beide Anwendungsfälle (Optimierung mit/ohne Verwendung eines Referenzmodells) folgen dann jeweils unterschiedliche graphische Darstellungen.

1. In Form von Zeitverläufen werden zunächst die Gesamtbeiwerte (Soll- / Istwert) vor und nach dem Training des Neuronalen Netzes verglichen. Die Sollwerte für C_A und C_m wurden hierfür aus den Flugmeßdaten während eines Wind Up Turn (WUT) Manövers mit Hilfe des Impulsatzes berechnet (detaillierte Vorgehensweise, siehe Abschnitt 5.1.2 und 5.1.3).
2. In dreidimensionalen Graphiken werden die einzelnen aerodynamischen Komponenten des Modells dargestellt. Der Klappenausschlag δ_f wurde hierfür auf einen Ausschlag von 5° festgehalten.
3. Zusätzlich wurden für die einzelnen aerodynamischen Parameter zweidimensionale Graphiken erstellt. Denn teilweise sind die geringfügigen Änderungen aufgrund der Optimierung des Modells aus den 3D- Graphiken kaum zu entnehmen. Dafür wurde zusätzlich die Machzahl Ma auf 0.5 fixiert. Etwa bei dieser Fluggeschwindigkeit wurde das Manöver im Flugversuch durchgeführt.

Auch wenn für beide Optimierungen (mit und ohne Referenzmodell) des aerodynamischen Modells derselbe Lerndatensatz verwendet wurde, sieht man doch deutlich, daß das Ergebnis unterschiedlich ausfällt. Der Grund dafür ist wohl ein unterschiedliches lokales Minimum in der Fehlerfunktion.

6.1 Das aerodynamische Modell

Das aerodynamische Modell sollte so gestaltet sein, daß alle wichtigen Einflußparameter berücksichtigt werden und dadurch die gestellten Anforderungen erfüllt werden können. Für den Einsatz in der Flugmechanik sind die Flugzustandsgrößen und Ruderausschläge des Flugzeugs die wesentlichen Parameter.

Bei der folgenden Modellierung wird von einer symmetrischen Umströmung ausgegangen. Es wird also nicht unterschieden zwischen den Anstellwinkeln auf der

rechten und linken Seite eines Flugzeugs. Außerdem wurden aerodynamische Effekte vernachlässigt, die bei klassischen Konfigurationen nur geringfügige Auswirkungen auf das Flugverhalten haben (z.B. C_{Aq}).

Für den Gesamtauftriebsbeiwert C_A gilt nun:

$$C_A = C_{A0}(\alpha, Ma, \delta_f) + C_{A\eta}(\alpha, Ma, \delta_f) \cdot \eta + C_{A\delta_s}(\alpha) \cdot \delta_s$$

Für den Nickmomentenbeiwert C_m :

$$C_m = C_{m0}(\alpha, Ma, \delta_f) + C_{m\eta}(\alpha, Ma, \delta_f) \cdot \eta + C_{mq}(\alpha, Ma, \delta_f) \cdot \frac{q\ell}{2V}$$

Hinweis: C_{A0} und C_{m0} sind die Beiwerte für den Fall, daß die Größen q, η und δ_s den Wert Null annehmen. Die Gradienten (z.B. $C_{m\alpha}$) tauchen hier nicht explizit auf.

6.2 Die Struktur des Neuronalen Netzes

Das Neuronale Netz wurde entsprechend dem aerodynamischen Modell (siehe oben) strukturiert.

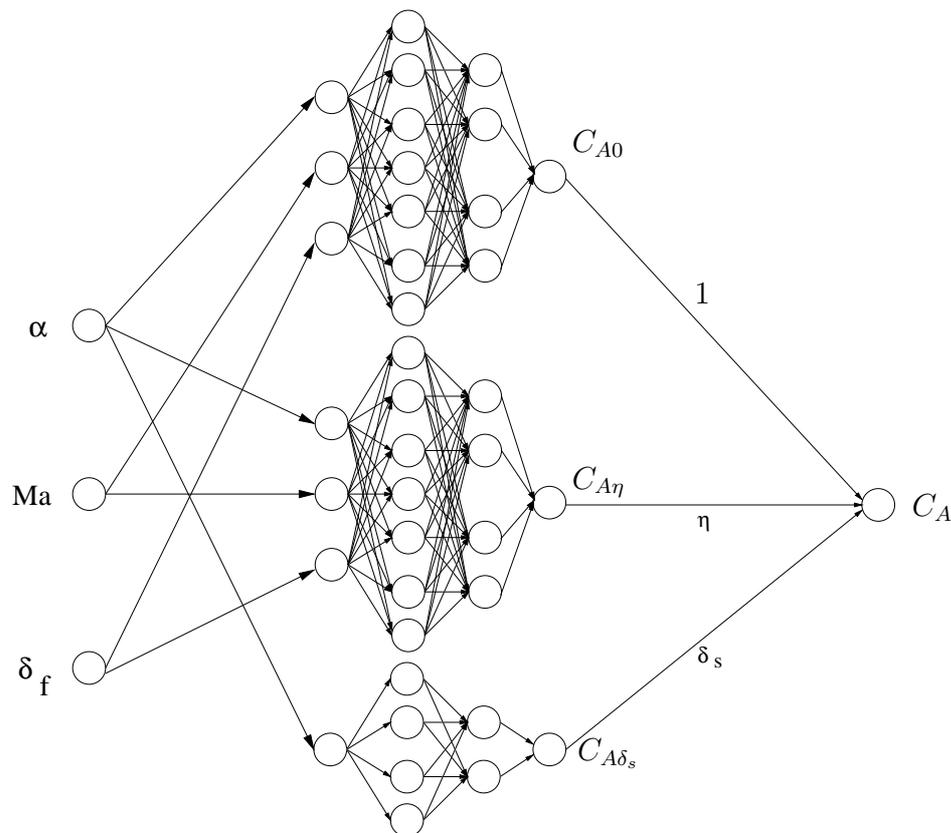


Abbildung 6.1: Modellstruktur für C_A

Als Eingabeneuronen sieht man auf der linken Seite α , Ma , δ_f , und als Eingabeverbindungen η und δ_s . In der mittleren Schicht befinden sich drei Gruppen für C_{A0} , $C_{A\eta}$ und $C_{A\delta_s}$.

Als Aktivierungsfunktion wurde bei allen Neuronen der drei Gruppen die sigmoide Funktion \tanh verwendet. Die Eingangs- und Ausgangsneuronen besitzen hingegen die Identität als Aktivierungsfunktion. Folglich werden im Neuron C_A lediglich die Eingänge summiert.

Der Wertebereich der \tanh Funktion mußte für jede Gruppe mit einem Faktor und einem Bias- Wert passend zum Referenzmodell (Abb. 6.3) erweitert werden.

Die Struktur des Neuronalen Netzes für den Nickmomentenbeiwert C_m ist ebenfalls dem aerodynamischen Modell entsprechend gestaltet. Sie ist der Struktur für C_A sehr ähnlich.

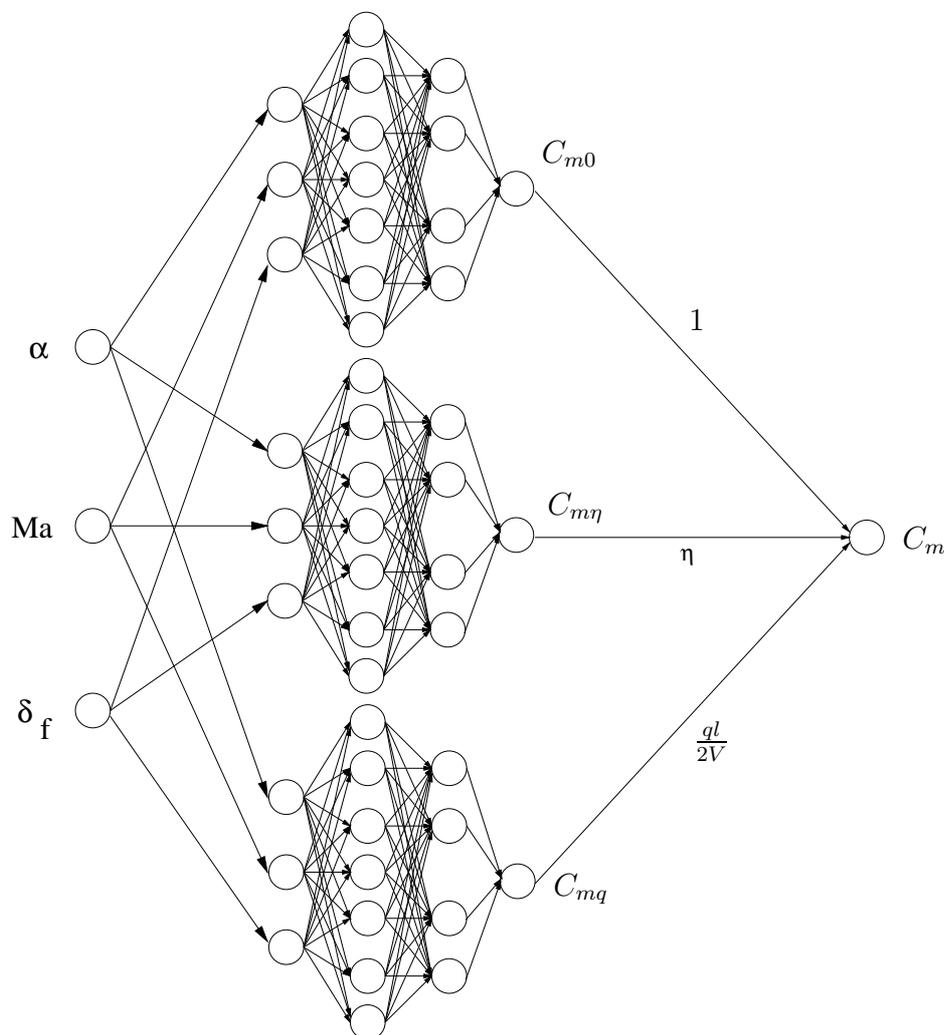


Abbildung 6.2: Modellstruktur für C_m

Alle drei Gruppen sind hier mit den selben drei Eingabeneuronen verbunden, die eine

nichtlineare Abhängigkeit ermöglichen. Die Eingangsgrößen η und $\frac{gl}{2V}$ sind als Eingabeverbindungen mit den Gruppen $C_{m\eta}$ und C_{mq} verknüpft. Von diesen Variablen erwartet man eine lineare Abhängigkeit.

6.3 Ergebnisse der Optimierung unter Verwendung eines Referenzmodells

Aus einem vorhandenen aerodynamischen Referenzmodell, welches sehr komplex aufgebaut ist¹, wurden einfache Lerndatensätze für die sechs Gruppen C_{A0} , $C_{A\eta}$, $C_{A\delta_s}$, C_{m0} , $C_{m\eta}$ und C_{mq} erstellt; siehe dazu auch Abschnitt 5.2.1.

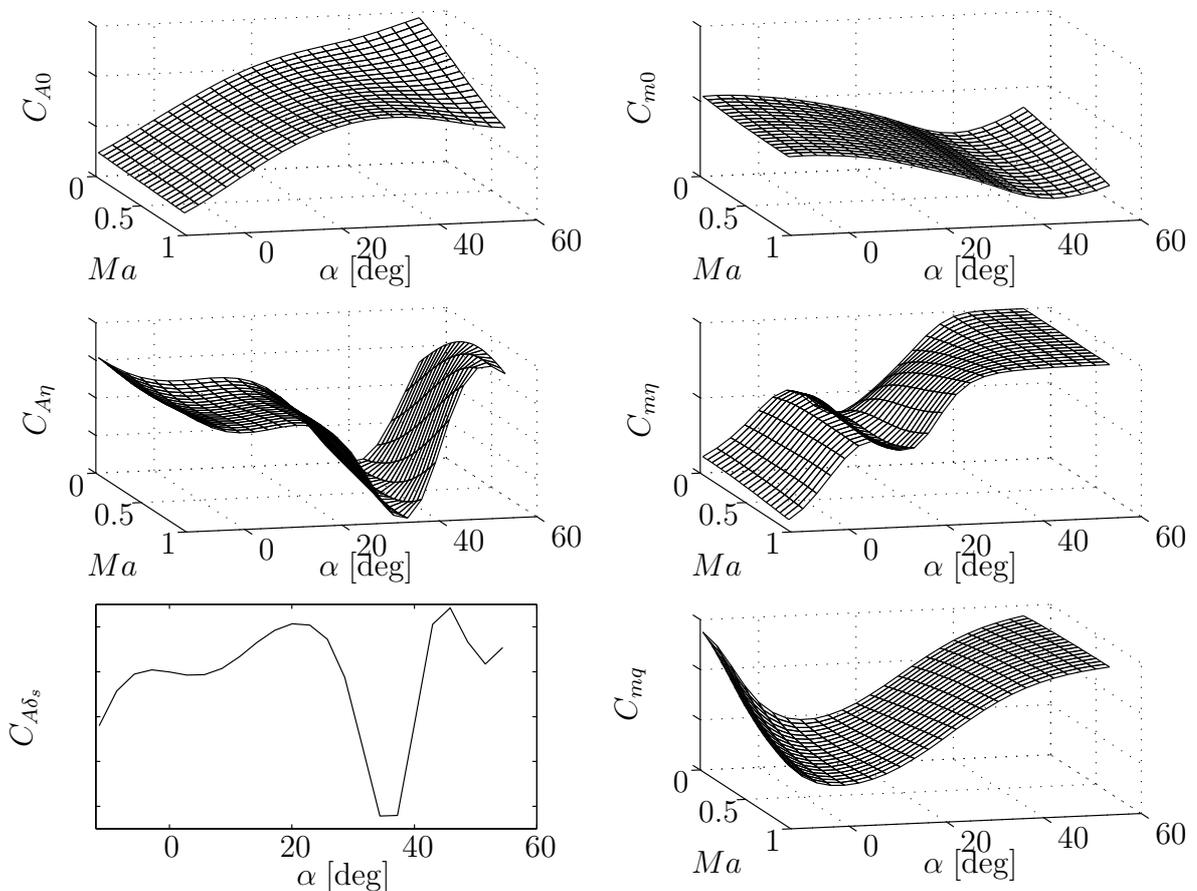


Abbildung 6.3: Basismodelle für C_A und C_m , initialisiert mit einem Referenzmodell, dargestellt für einen Klappenwinkel $\delta_f = 5^\circ$

Anhand der Modellstruktur erkennt man, daß alle drei Gruppen, jeweils für C_A und C_m von drei Variablen nichtlinear abhängig sind. Diese Lerndatensätze sind also

¹Anmerkung: Das Referenzmodell enthält vertrauliche Daten und kann deswegen hier nicht genau beschrieben werden.

derart aufgebaut, daß den perturbierten Eingangsgrößen α , Ma und δ_f die aus dem Referenzmodell errechneten Werte für diese Anteile ($C_{A0\dots}$) zugeordnet wurden. Für α wurde eine konstante Schrittweite von 1° , für Ma 0.1 und für δ_f vier Klappenstellungen gewählt. So ein Datensatz enthält dann etwa 1000 Wertepaare.

Die Neuronalen Netze für C_A und C_m wurden durch isoliertes Training der jeweils drei Gruppen initialisiert. Die Eingangsgröße für $C_{A\delta_s}$ war hierfür δ_f , für alle anderen Gruppen waren es α , Ma und δ_f und für alle Gruppen natürlich die Sollwerte der jeweiligen Derivative. Das Ergebnis ist in Abbildung 6.3 zu sehen.

Leider lassen sich keine aussagekräftigen Graphiken mit drei freien Parametern erstellen. Deswegen wurde hier ein freier Parameter festgehalten ($\delta_f = 5^\circ$). Die Funktionsverläufe sind über den zulässigen Bereich von $\alpha = [5; 20]$ hinaus dargestellt.

Es ist eine aufschlußreiche und einfache Methode zur objektiven Beurteilung des aerodynamischen Modells, wenn die Gesamtbeiwerte verglichen werden (*static checks*). Einerseits erhält man aus dem Basismodell die Gesamtbeiwerte C_A und C_m , in den folgenden Abbildungen mit "Ausgabewerten" bezeichnet. Andererseits errechnet man sie aus den im Fluge gemessenen Beschleunigungen, wie in Abschnitt 5.1 beschrieben. In den Abbildungen werden sie "Sollwerte" genannt.

Ein Vergleich der Beiwerte C_A (in Form eines Zeitverlaufs) für ein WUT- Manöver ist in Abbildung 6.4 zu sehen.

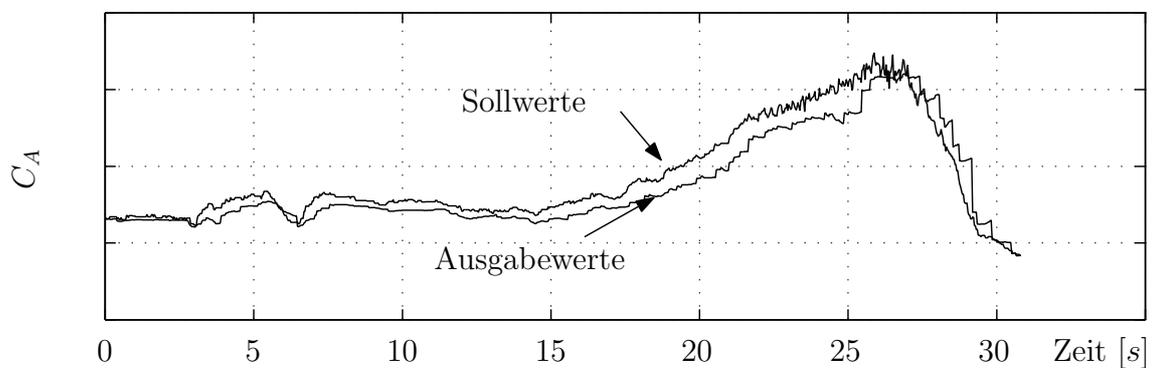


Abbildung 6.4: Zeitverlauf des Auftriebsbeiwerts vor der Optimierung

Ohne weiter ins Detail gehen zu müssen, erkennt man schon mit Hilfe dieser einfachen Darstellung, daß das Basismodell im Trimmzustand einen zu geringen Wert liefert. Ab Beginn des Manövers (nach ca. 15 Sekunden) wird die Differenz zwischen Soll- und Istwert noch größer.

Nach einem Training von 20 Epochen mit einem Lerndatensatz bestehend aus Flugmeßdaten eines einstündigen Fluges (enthaltene Manöver: WUT, Fullstick, Looping,

Rolle) ist ein verbessertes aerodynamisches Modell verfügbar. Der Vergleich von oben wurde nun wiederholt und in Abbildung 6.5 dargestellt.

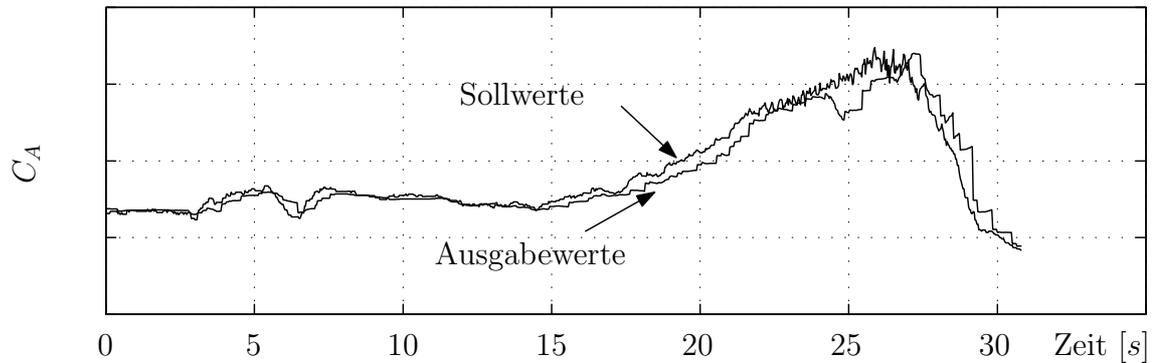


Abbildung 6.5: Optimierter Auftriebsbeiwert

Die C_A -Werte sind für den Trimmzustand besser als vorher. Im mittleren Anstellwinkelbereich erkennt man ebenfalls eine Verbesserung. Aber bei etwa 25 Sekunden bleibt eine große Abweichung, die ihre Ursache in einem rapiden Schiebewinkelanstieg auf etwa 10° haben kann. Dieser große Schiebewinkel stellt sich beim Ausleiten des Manövers ein, da das Flugsteuersystem nicht über eine Regelung des Schiebewinkels verfügt.

Die gewählte Modellstruktur bietet, genauso wie das Referenzmodell, keinen Einfluß des Schiebewinkels auf den Auftriebsbeiwert. Diese Schwäche könnte durch das Einbringen eines Eingabeneurons für β in die Modellstruktur und einer Verbindung zur C_{A0} Gruppe gelöst werden.

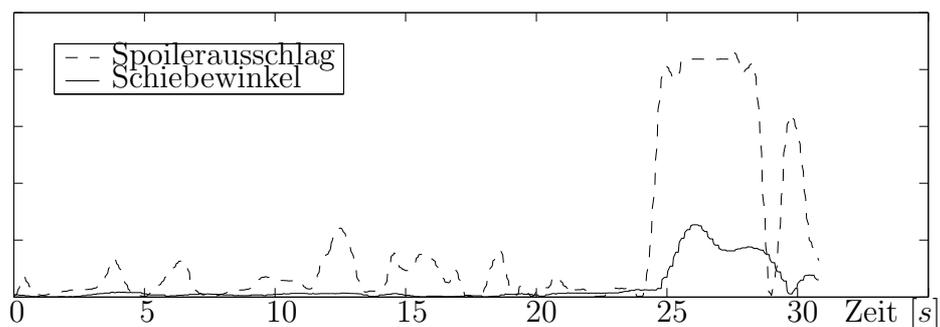


Abbildung 6.6: Korrelation zwischen Spoilerausschlag und Schiebewinkel

In den Meßdaten steckt allerdings nicht ausreichend Information, um einen Schiebewinkeleinfluß identifizieren zu können. Der Schiebewinkel korreliert zu stark mit dem Spoilerausschlag ($r = 0.63$). In Abbildung 6.6 sind die beiden Winkel für das WUT Manöver zu sehen.

Betrachten wir nun den Gesamtnickmomentenbeiwert C_m in Abbildung 6.7. Die Ausgabewerte des Basismodells liegen im Trimmzustand zuweit im negativen Bereich. Die Differenz wird ab Beginn des Manövers (nach 15 Sekunden) noch größer. Nahe dem maximalen Anstellwinkel sind sich die Werte aber sehr ähnlich.

Der Sprung des Sollwertes bei etwa 25 Sekunden stört nicht weiter die Optimierung. Er wurde aus Versuchszwecken bei der Datenaufbereitung nicht entfernt.

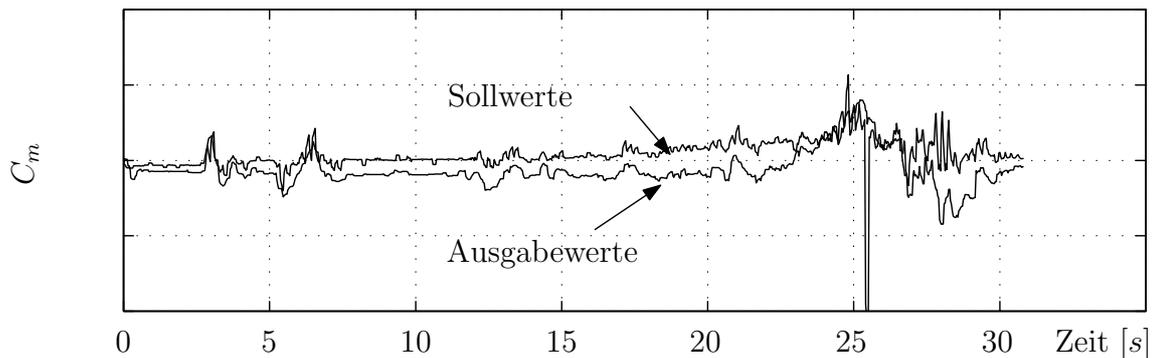


Abbildung 6.7: Nickmomentenbeiwert vor der Optimierung

Nach dem Training des Neuronalen Netzes erhält man eine recht ordentliche Übereinstimmung zwischen Sollwerten und Ausgabewerten; sowohl im Trimmzustand als auch während des Manövers. Siehe hierzu Abbildung 6.8.

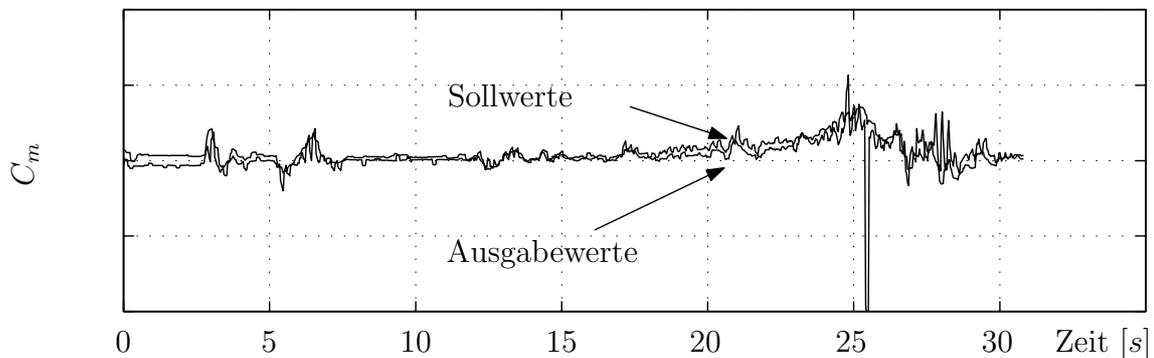


Abbildung 6.8: Optimierter Nickmomentenbeiwert

In Abbildung 6.9 ist eine typische Lernkurve für das Optimieren eines vortrainierten Modularen Neuronalen Netzes zu sehen. Sie unterscheidet sich insofern von den üblichen Fehlerverläufen, daß nach einer ersten großen Fehlerreduktion keine wesentlichen Änderungen zu erkennen sind. Hier ist zwar nur ein Bereich von 40 Epochen dargestellt, aber auch nach mehreren tausend Zyklen kann mit keiner großen Verbesserung

mehr gerechnet werden. Wie bei klassischen MLP- Netzen besteht aber die Gefahr der Überanpassung (*overfitting*). Deswegen wurde das Training schon nach 20 Epochen abgebrochen.

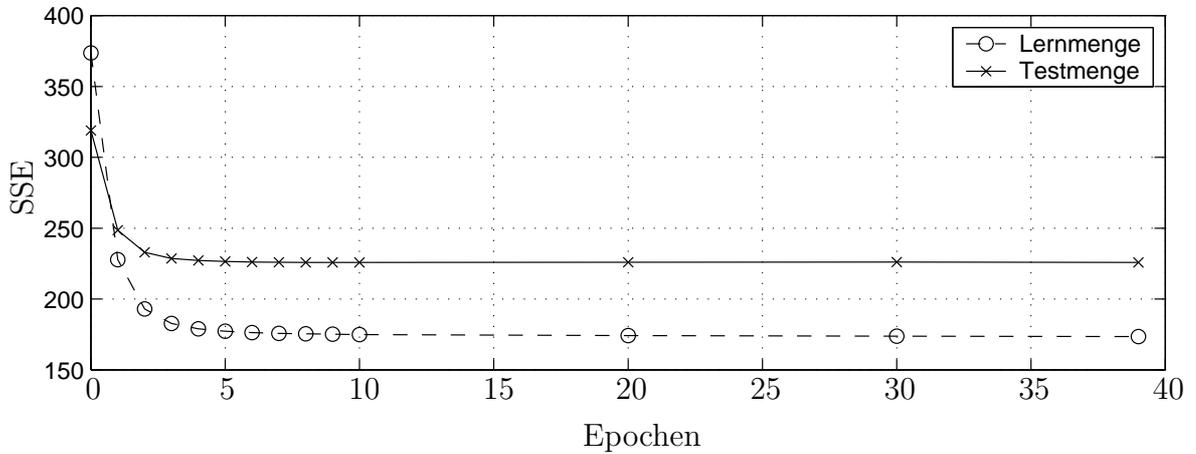


Abbildung 6.9: Typische Lernkurve und Fehler der Testmenge

Es folgt nun eine detaillierte Betrachtung der aerodynamischen Anteile an den Gesamtbeiwerten C_A und C_m .

In Abbildung 6.10 sieht man die Funktionsverläufe der zwei optimierten Modelle. Zum Vergleich ist zusätzlich das Basismodell (Punkte) eingeblendet.

Auch wenn ein Anstellwinkelbereich von $[-10;60]$ dargestellt ist, es interessiert nur der Bereich $\alpha = [5; 20]$.

In den 3D- Graphiken sind allerdings die Veränderungen kaum zu erkennen. Deswegen wird in Abbildung 6.11 der freie Parameter Ma auf 0.5 festgehalten. Die Funktionsverläufe sind demnach nur noch vom Anstellwinkel abhängig.

Die Veränderungen sind nun deutlicher wahrzunehmen. Durch die Optimierung wird der Verlauf von C_{A0} zu größeren Werten hin verschoben. Gleichzeitig erhält man für $C_{A\eta}$ kleinere Zahlenwerte.

Bei der $C_{A\delta_s}$ Gruppe wurde der Funktionsverlauf nicht nur einfach parallel verschoben. Die Spoilerwirksamkeit wurde nur bei höheren Anstellwinkeln reduziert.

Der Nickmomentenbeiwert C_m konnte, allein durch Verschieben des C_{m0} Verlaufs zu weniger negativen Werten, optimiert werden. Die Derivative $C_{m\eta}$ und C_{mq} blieben weitgehend unverändert. Die Verläufe liegen in der Graphik übereinander.

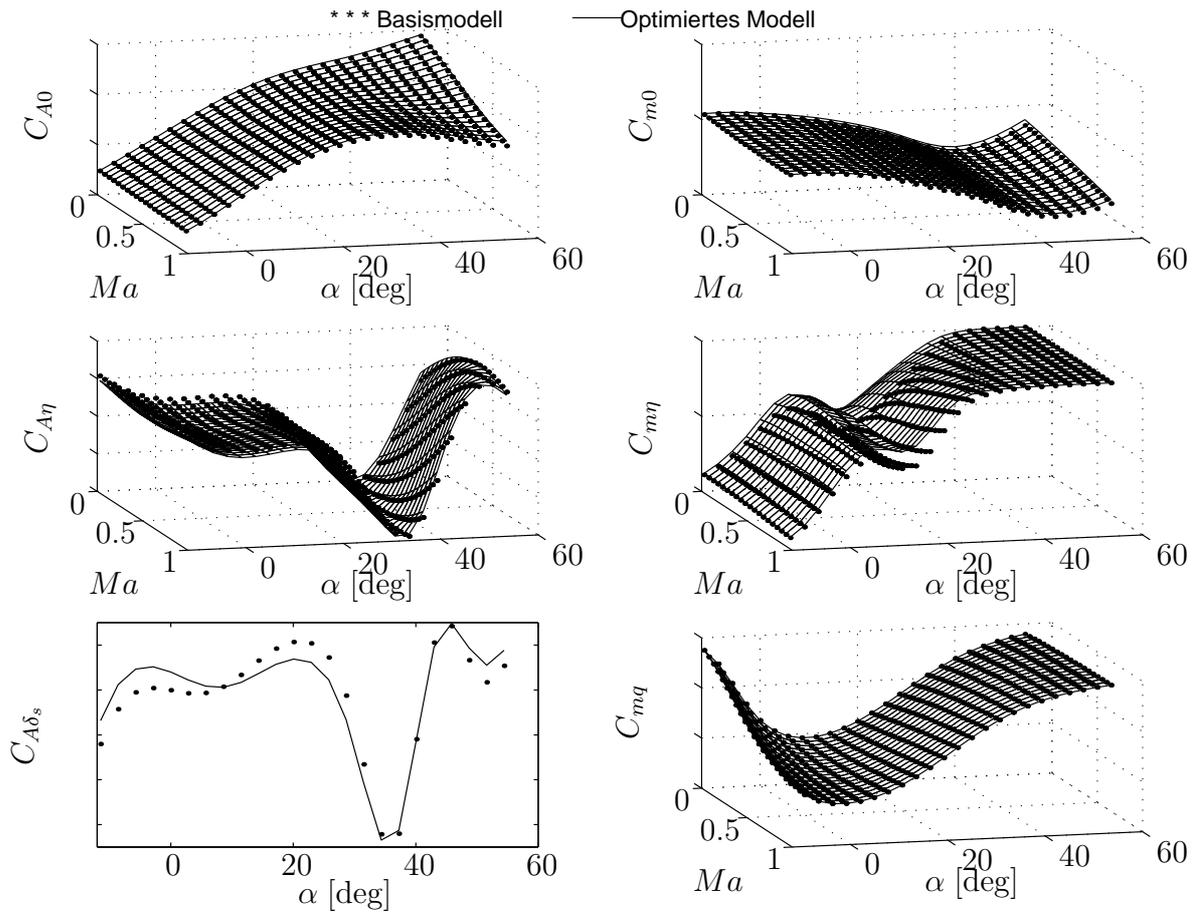
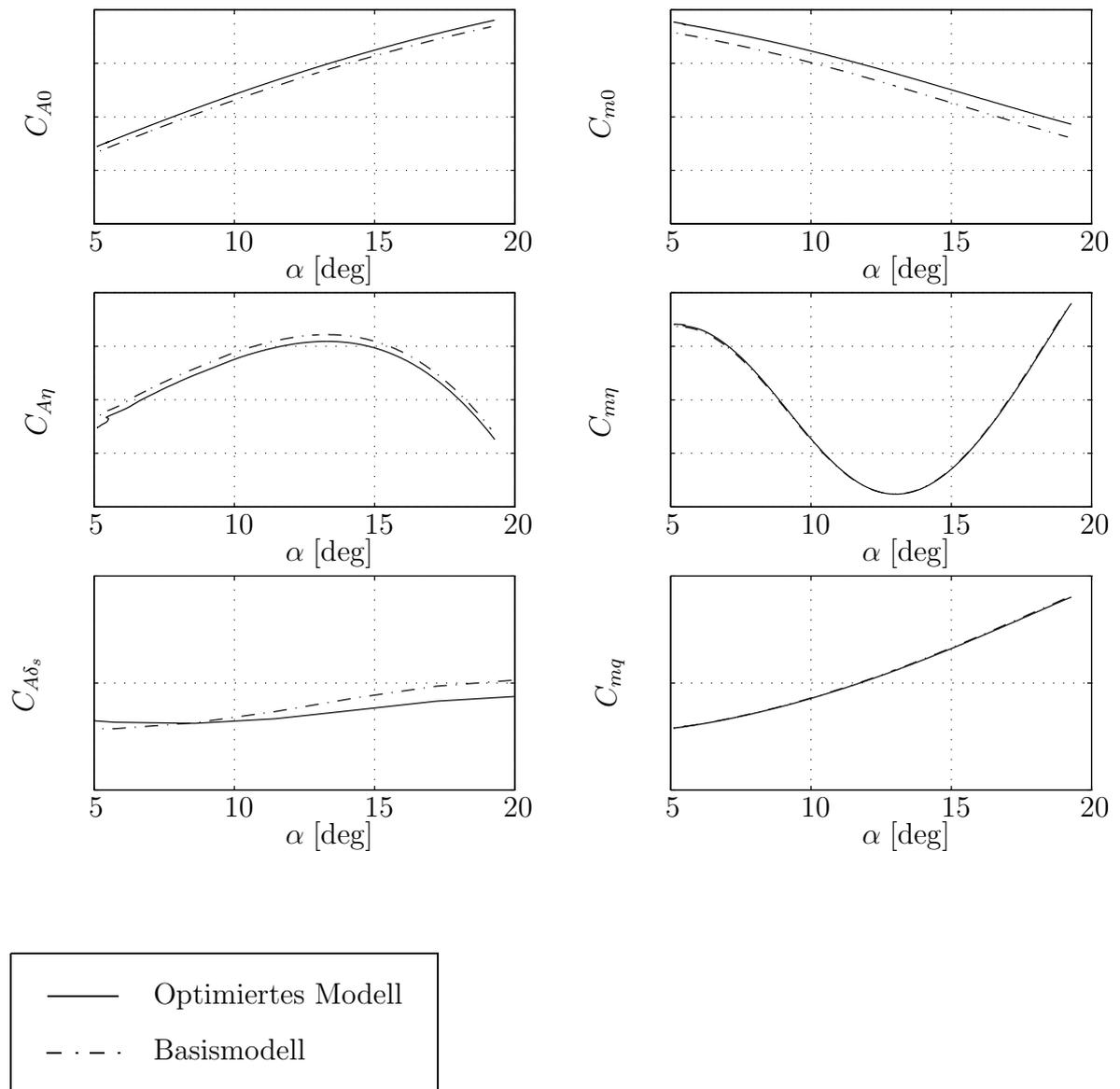


Abbildung 6.10: Optimierte Modelle nach 20 Epochen

Abbildung 6.11: Optimierte Modelle nach 20 Epochen, $Ma = 0.5$

6.4 Ergebnisse der Modelloptimierung ohne Referenzdaten

Sofern für ein System noch kein Modell existiert, muß es erst physikalisch beschrieben werden. Bei Modulare neuronalen Netzen geschieht das durch eine geeignete Wahl der Netzwerkstruktur. Um die Ergebnisse der Optimierung, basierend auf einem Basismodell (Abschnitt 6.3), mit denen ohne Basismodell verglichen zu können, wurde hier dieselbe Netzwerkstruktur und derselbe Lerndatensatz verwendet.

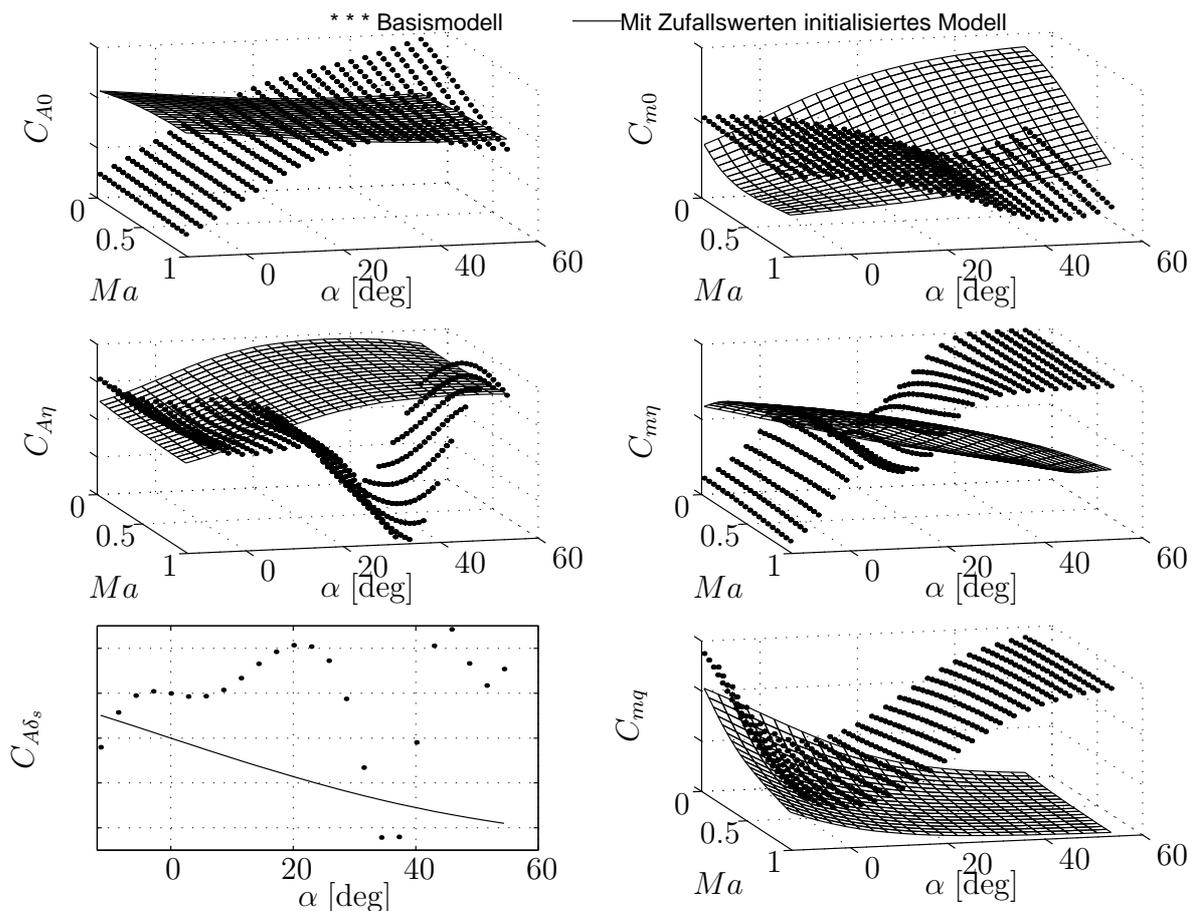


Abbildung 6.12: Mit Zufallswerten initialisiertes Neuronales Netz im Vergleich zum Basismodell

Der Unterschied der hier genutzten Methode bzgl. Optimierung (mit/ohne Basismodell) liegt in den unterschiedlich initialisierten Netzwerkgewichten. Für die Optimierung des Basismodells wurden die Netzwerk-Gruppen vortrainiert. Dies ist natürlich ohne Vorhandensein eines Referenzmodells nicht möglich.

In Abbildung 6.12 sind die Verläufe des mit Zufallswerten initialisierten Neuronalen

Netzes dargestellt. Die Funktionsverläufe der Gruppen stellen, im Vergleich zum Basismodell (Punkte), keine aerodynamisch und somit auch flugmechanisch sinnvolle Lösung dar.

Zu Beginn des Trainings ist der Netzwerkfehler noch sehr groß, deswegen ändern sich die Funktionsverläufe der sechs Gruppen zunächst sehr deutlich. Trotzdem ist noch ein langes Training (ca. 5000 Epochen) erforderlich.

Die Wahrscheinlichkeit, ein lokales Minimum zu treffen, ist bei der Optimierung ohne Basismodell wesentlich größer, als wenn Referenzdaten zur Initialisierung verwendet werden können.

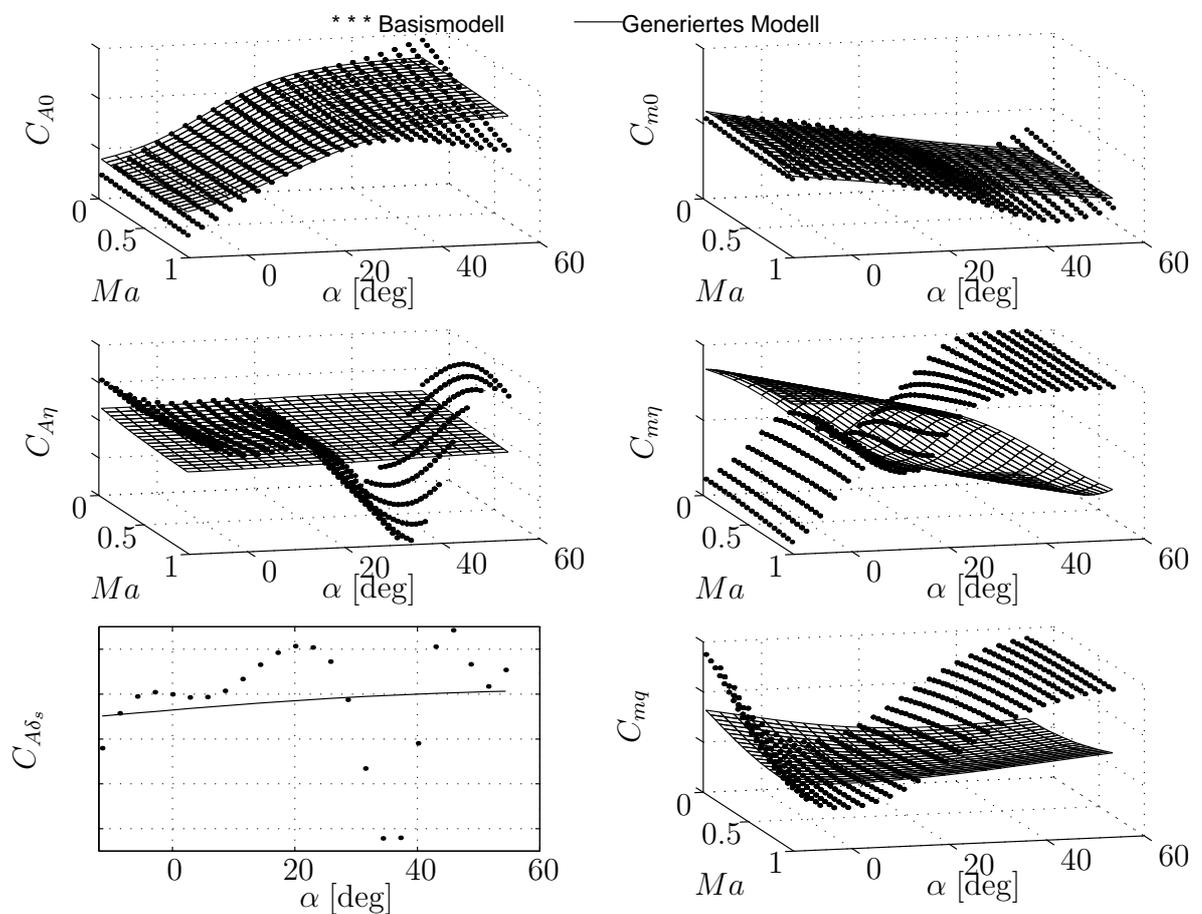


Abbildung 6.13: Generiertes Modell (nach 5000 Epochen)

In Abbildung 6.13 ist das Ergebnis der Optimierung ohne Basismodell wieder durch die Funktionsverläufe der sechs Gruppen graphisch dargestellt. Es ist zu erkennen, daß die Beiwerte C_{A0} und C_{m0} sehr gut approximiert wurden.

Für die restlichen Derivative kann man sagen, daß die Größenordnung der Funktionswerte stimmt, aber die nichtlinearen Abhängigkeiten nur schlecht identifiziert wurden.

Bei der Beurteilung sollte man wieder nur den Bereich $\alpha = [5, 20]$ betrachten. Die Hauptanteile, und vor allem deren Gradienten, wurden gut identifiziert. Die Auswirkungen der schlecht identifizierten dynamischen Derivative auf das Flugverhalten in der Simulation sind nicht sehr groß, wie in Abschnitt 7.4.4 (Abbildung 7.6 und 7.9) gezeigt wird.

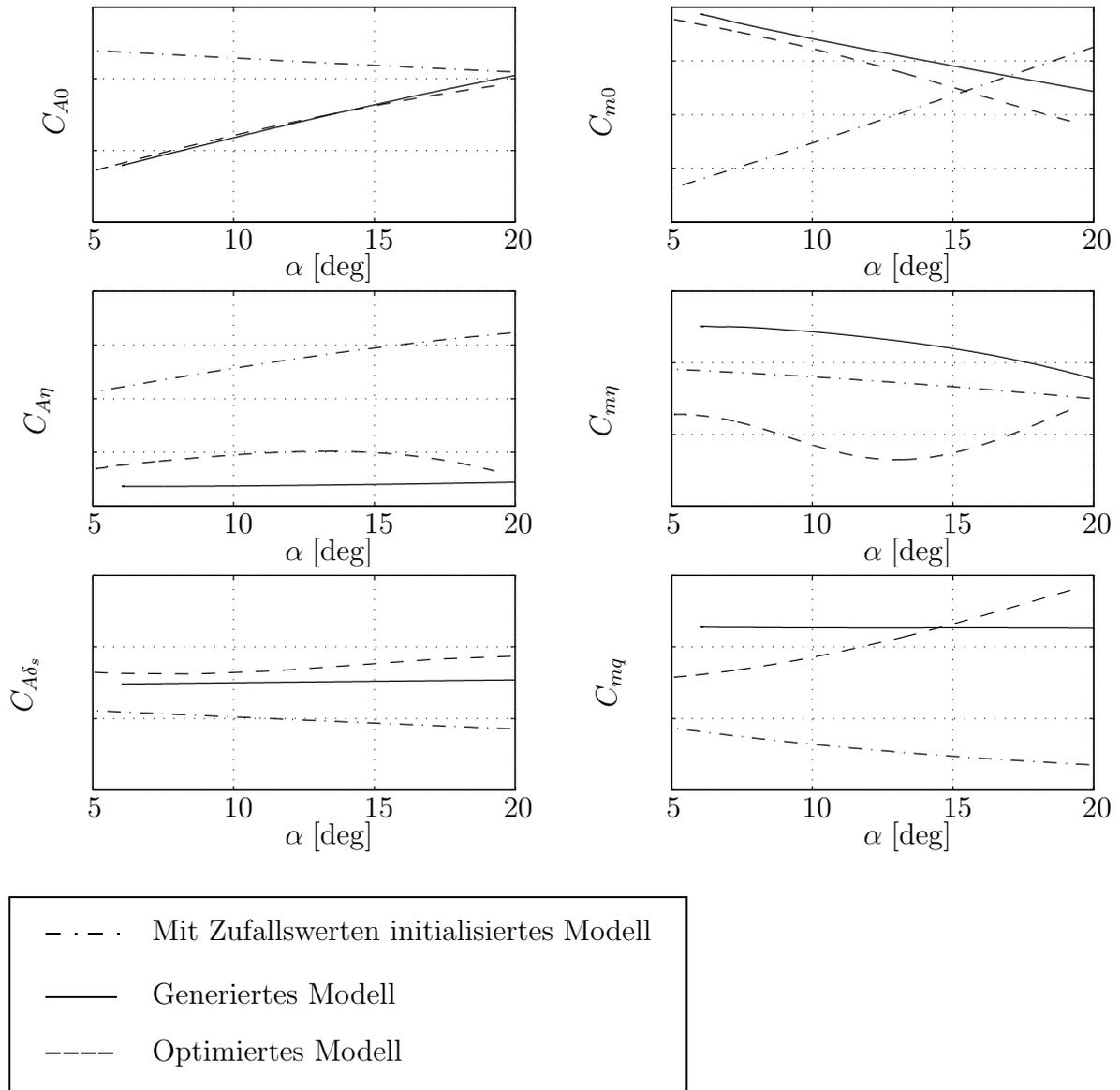


Abbildung 6.14: Generiertes Modell, $Ma = 0.5$

In Abbildung 6.14 sind die Funktionsverläufe wieder zweidimensional und im relevanten Anstellwinkelbereich gezeichnet. Neben dem Anfangszustand (initialisiertes Modell) und dem Ergebnis der Optimierung (Generiertes Modell) ist hier zusätzlich

das Optimierte Modell aus Abbildung 6.11 eingeblendet.

Der Hauptanteil C_{A0} bietet, wie schon erwähnt, eine sehr gute Übereinstimmung mit dem optimierten Modell! Der Verlauf von C_{m0} hat eine geringfügige Abweichung der Absolutwerte, und einen etwas positiveren Gradienten. Das wirkt sich in der Simulation auf die Flugstabilität aus.

Die Derivative $C_{A\eta}$ und $C_{A\delta_s}$ sind dem optimierten Verlauf ähnlich; vor allem in Bezug auf den Gradienten, weniger bzgl. der Absolutwerte.

Das Derivat $C_{m\eta}$ ist fehlerhaft. Von der initialisierten Position wurde es durch das Training noch weiter von den optimierten Werten aus Abschnitt 6.3 entfernt. Eine gegenseitige Beeinflussung durch C_{m0} ist wahrscheinlich der Grund dafür.

Die Funktionswerte der Gruppe C_{mq} stimmen in etwa mit den optimierten Werten überein. Der Gradient wurde zwar verbessert, ist aber noch zu flach.

7 Objektive Beurteilung der Flugsimulation

Die Analyse der identifizierten aerodynamischen Parameter, wie sie im vorangegangenen Kapitel beschrieben wurde, ist wichtig, um deren Gültigkeit aus aerodynamischer Sicht bewerten zu können. In diesem Kapitel wird mit Hilfe einer Flugsimulation mit sechs Freiheitsgraden gezeigt, welche Auswirkungen die identifizierten Parameter auf das Flugverhalten haben. Die Sensitivität der Simulation auf die einzelnen aerodynamischen Parameter spielt dabei die Schlüsselrolle.

Durch einen objektiven Vergleich von Flugversuchsdaten und Flugsimulationsdaten in Form von Zeitverläufen, muß die Qualität der Simulation in einem *end-to-end Test* nachgewiesen werden.

7.1 Simulationstechnik

In der numerischen Simulation eines Flugzeuges mit sechs Freiheitsgraden werden die nichtlinearen Bewegungsgleichungen durch Integration gelöst. Der Integrationsschritt für eine echtzeitfähige Flugsimulation liegt heute bei wenigen Millisekunden. Als Integrationsverfahren wird zum Beispiel das implizite Adams Bashford- Verfahren, oder auch Newton- Verfahren angewendet. In einem Simulationsschritt wird so aus einem Derivativvektor \dot{X} ein Zustandsvektor X berechnet. In diesem Zustandsvektor sind zum Beispiel die Fluggeschwindigkeit und die Lagewinkel im Raum enthalten.

Für das Nachsimulieren aufgezeichneter Manöver, z.B. aus dem Flugversuch, wird eine besondere Technik verwendet. Diese betrifft zum einen das Trimmen des Simulators vor dem Manöver, als auch die Manöverdurchführung.

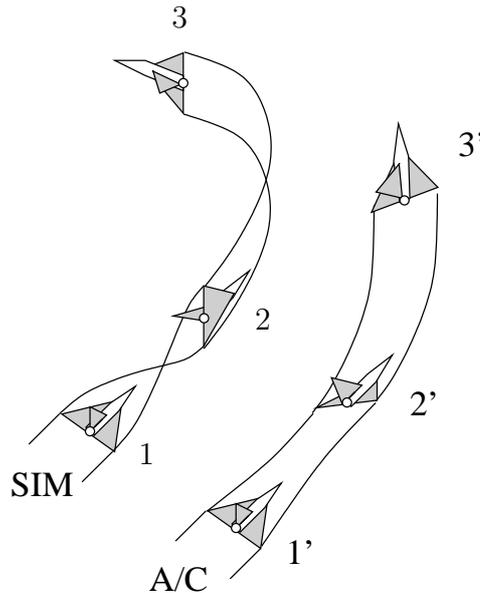


Abbildung 7.1: Nachsimulieren aufgezeichneter Manöver

7.1.1 Trimmen

Üblicherweise trimmt der Testpilot das Flugzeug stationär aus, bevor ein Manöver zur Datenaufzeichnung geflogen wird. Der Sinn dafür liegt darin, vordefinierte Anfangsbedingungen zu setzen. In der Simulation werden die Zustandsgrößen entweder einfach auf diese Meßwerte gesetzt oder man trimmt den Simulator mit Hilfe der Trimmknöpfe solange, bis sich ein ähnlicher Anfangszustand einstellt. Für Flugzeuge mit einem elektronischen Flugsteuersystem gilt noch eine Besonderheit. Die internen Reglergrößen, wie zum Beispiel Integratoren, müssen berücksichtigt werden. Da diese Werte im Flugversuch nicht alle aufgezeichnet werden, ist es auch nicht möglich, sie in der Simulation mit Meßwerten zu überschreiben. Folglich wird auch der Flugregler getrimmt. Dieses Trimmverfahren ist in [32] beschrieben.

Während des Trimmvorgangs wird nicht nur die Bewegung des Flugzeugs angepaßt, sondern auch Simulationsmodelle abgeglichen. Im Atmosphärenmodell wird aus den Meßwerten T_{tot} und p_{stat} ein QNH in Meereshöhe und ein Wert ΔT ausgerechnet. Damit erhält man eine bessere Übereinstimmung für den dynamischen Druck, der wesentlich ist für die Berechnung von aerodynamischen Kräften und Momenten. Außerdem wird in der Trimmphase das Gewicht des Flugzeugs auf die Meßwerte gesetzt. Die Schwerpunktslage, die Trägheits- und Deviationsmomente werden dann aus dem Massenmodell entsprechend angepaßt.

7.1.2 Manöver

Ein Manöver wird nachsimuliert, indem entweder die aufgezeichneten Ruderausschläge oder die Pilotenkommandos in die Simulation eingespeist werden. Welche Eingabe man wählt, hängt davon ab, wie groß die Eigenstabilität der Flugzeugkonfiguration ist. Handelt es sich um eine Konfiguration mit niedriger Eigenstabilität, dann wird üblicherweise im Flugzeug ein Flugregler zur Erhöhung der Stabilität eingesetzt. In diesem Fall ist es ratsam, die Pilotenkommandos (Steuerknüppel, Pedale) einzuspeisen. Andernfalls erhält man schon nach wenigen Sekunden divergierende Verläufe von Meßdaten und Simulationswerten.

Generell wird in der Flugmechanik zwischen der Längs- und der Seitenbewegung unterschieden. Eine isolierte Betrachtung einer der beiden Bewegungsformen ist auch hier ratsam. Denn die meisten Flugtest- Manöver lassen sich einer der Kategorien zuordnen. Bei der Simulation von Manövern werden zur Trennung der Kopplungseffekte die sechs Freiheitsgrade reduziert. Siehe dazu Abschnitt 7.3.

7.1.3 Versuchsdurchführung

Bei allen Manövern, die in Abschnitt 7.4.4 dargestellt werden, war der Ablauf folgendermaßen: zunächst wurden die Anfangsbedingungen aus den Meßwerten übernommen. Die Simulation wird mit der Fluggeschwindigkeit, Flughöhe, Flugzeugmasse und Fluglage des echten Flugzeugs initialisiert. Die sekundären Steuerflächen (Hinterkantenklappen, etc.) werden in Position gebracht.

Die ersten zwölf Sekunden wird dann der Simulator, wie in Abschnitt 7.1.1 beschrieben, getrimmt. Danach wird das Manöver durchgeführt.

Es wurden hier die Ausschläge des Steuerknüppels, der Pedale und der Schubhebel in die Simulation eingespeist. Dadurch ist es möglich, ein Manöver von etwa zwanzig Sekunden durchzuführen, ohne größere Abweichungen zu bekommen.

Bei dem Flugregler handelt es sich um ein *rate-command* System, welches abhängig vom Steuerknüppelausschlag Flugzeugdrehraten in der Längs- und Querachse kommandiert. Die im Fluge gemessenen Drehraten werden in das System zurückgeführt, um die Sollwerte regeln zu können.

7.2 Charakteristische flugmechanische Bewegungsgrößen

In einem Testflugzeug gibt es Tausende von Meßwertaufnehmern, deren Daten aufgezeichnet werden können. Es sind aber nur ein paar wenige Größen, die charakteristisch

für die Bewegung eines Flugzeugs sind. Dies sind zum einen die Parameter der Zustandsvektoren X der Längs- und Seitenbewegung.

$$\mathbf{X}_L = \begin{pmatrix} V \\ \alpha \\ q \\ \theta \end{pmatrix}, \quad \mathbf{X}_S = \begin{pmatrix} \beta \\ p \\ r \\ \phi \end{pmatrix}$$

Auch die Ausschläge der Steuerflächen sind wichtig. Sie sind im Steuervektor $u = (\eta, \dots)^T$ zusammengefaßt.

Bei sehr dynamischen Manövern ist auch das Lastvielfache n_z wichtig, da diese Bewegungsgröße einen bestimmten Maximalwert, abhängig von der Zulassung, nicht überschreiten darf.

7.3 Fehlerquellen in der Flugsimulation

Voraussetzung für eine seriöse Durchführung der Flugsimulation ist neben qualifizierter Software eine gute Dokumentation. Ohne die Informationen über Konfiguration, Flugzustand, evtl. Klappen- und Schalterstellungen und Einheiten, sind keine fehlerfreien Ergebnisse mit der Simulation zu erzielen.

Wenn durch große Sorgfalt diese Fehlerquelle ausgeschlossen werden kann, sollte man immer noch bei der Betrachtung von Simulationsergebnissen vorsichtig mit Schlußfolgerungen sein. Die Ursachen für etwaige Abweichungen zwischen Flugversuch und Flugsimulation können vielfältig sein. Oft ist es nützlich, den Blick nicht nur auf ein Manöver, sondern auf mehrere zu richten.

Bei der Analyse von Flugtest- und Simulationsdaten spielt die Kopplung der Längs- und Seitenbewegung eine große Rolle. Zum Beispiel wirkt sich eine falsche Rollage auf das normale Lastvielfache n_z aus, da der Gravitationsvektor immer zum Erdmittelpunkt zeigt.

Ein anderes Beispiel ist die anstellwinkelabhängige Aerodynamik. Die meisten Beiwerte, egal ob Längs- oder Seitenbewegung, verhalten sich über den Anstellwinkelbereich nichtlinear. Leidet die Simulation unter fehlerhaften Auftriebsbeiwerten, dann werden zwangsweise für die Seitenbewegung falsche Werte aus den aerodynamischen Tabellen ausgelesen. Die Beurteilung, zum Beispiel einer Rollbewegung, ist dadurch nicht möglich.

Auch den Schiebewinkel darf man nicht übersehen, denn durch ihn wird die Umströmung des Flugzeugs beeinflusst. In den Abbildungen 6.4 und 6.5 konnte man diese Auswirkungen auf den Auftriebsbeiwert sehen.

7.3.1 Beispiel für eine Fehldiagnose

Der eben erwähnte Einfluß des Gravitationsvektors spielt bei dem folgenden Beispiel eine große Rolle. Als Manövertyp wurde ein Wind Up Turn gewählt. Zweimal wurde dieses zweiachsige Flugmanöver mit identischen Piloteneingaben simuliert. Im zweiten Fall (“Flugsimulation”) wurde jedoch der Rolldämpfungsbeiwert C_{Lp} mit dem Faktor 0.5 versehen, um die Auswirkung einer Ungenauigkeit im Simulationsdatensatz zu demonstrieren.

Zum Zeitpunkt $t = 1$ befinden sich beide Flugzeuge noch in Horizontallage (siehe auch Abbildung 7.1). Eine Sekunde später erkennt man bereits eine große Abweichung im Rollwinkel ϕ . Ab diesem Zeitpunkt rotieren beide Flugzeuge schnell um die Querachse. Nach drei Sekunden weichen beide n_z - Verläufe voneinander ab. Der Grund dafür ist der resultierende Beschleunigungsvektor in Richtung der flugzeugfesten z - Achse. Aufgrund des größeren Rollwinkels zeigt die zweite Simulation (“Flugsimulation”) einen zu kleinen Wert für n_z an.

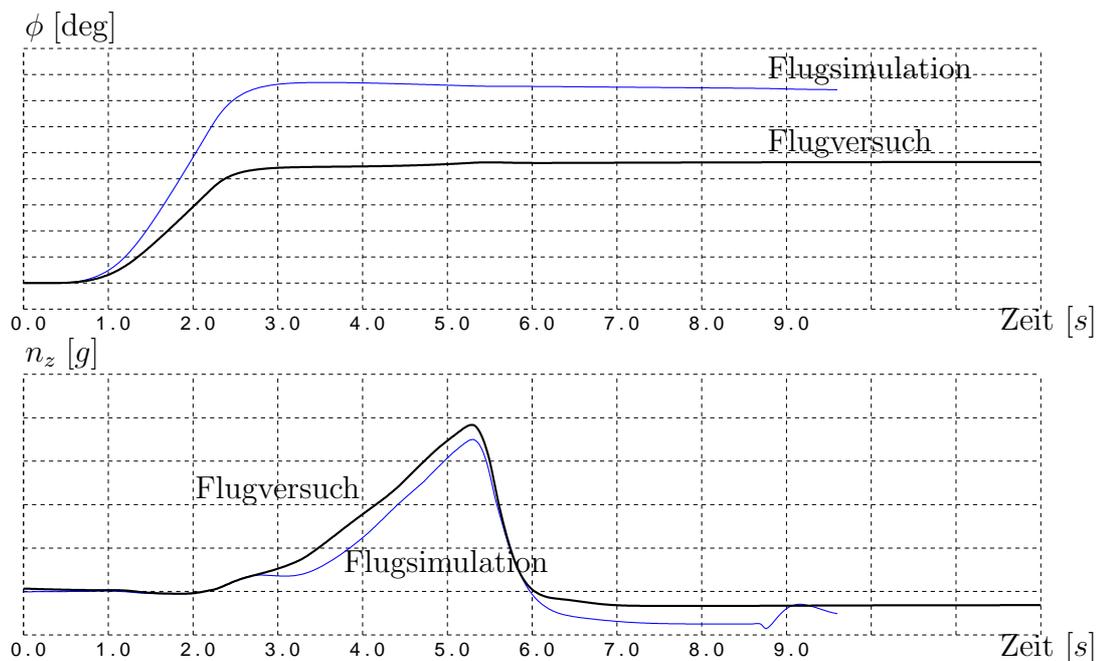


Abbildung 7.2: Beeinflussung der Simulationsergebnisse durch die Kopplung von Längs- und Seitenbewegung

Die Differenz im Lastvielfachen folgt der Gleichung für stationären Kurvenflug:

$$\Delta n_z = \frac{\cos \Phi_{AC} - \cos \Phi_{SIM}}{g}$$

Das gezeigte Problem läßt sich auf zwei Arten lösen.

1. Die Simulationswerte werden teilweise mit Flugmeßdaten überschrieben. Für unser Beispiel bedeutet das: Der Rollwinkel aus der zweiten Simulation wird mit den Meßwerten der ersten Simulation überschrieben. Das Pilotenrollkommando wird ignoriert. Dieses Vorgehen bedingt allerdings einen Eingriff in den Kern der Simulationssoftware.
2. Die zweite Lösung ist eleganter. Ohne das Simulationsprogramm ändern zu müssen, kann der Rollwinkel mit einem Proportionalregler um die Sollwerte der ersten Simulation geregelt werden.

Mit Hilfe einer dieser Lösungen kann die Längsbewegung isoliert betrachtet werden. Für Lösung 2 werden nun die Ergebnisse gezeigt.

7.3.2 Bessere Ergebnisse durch Nachführen des Rollwinkels mit Hilfe eines Proportionalreglers

Für die isolierte Betrachtung der Längsbewegung eines Flugzeugs wurden soeben zwei Lösungsmöglichkeiten vorgeschlagen. Für die favorisierte zweite Variante ist das Simulationsergebnis in Abbildung 7.3 zu sehen. Die Zeitverläufe für n_z liegen nun übereinander (vgl. Abb. 7.2).

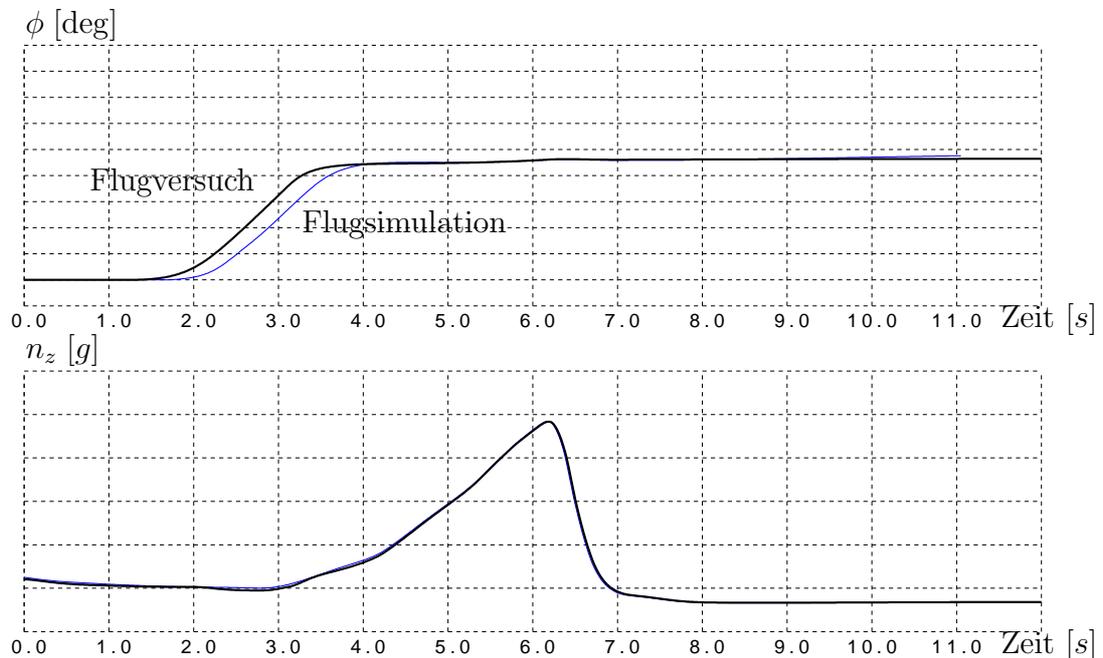


Abbildung 7.3: Besseres Simulationsergebnis durch Nachführung des Rollwinkels

Für diese Simulation wurde ein Pilotenmodell eingesetzt (siehe dazu [33, 34]), daß die Rollachse mit Hilfe der gemessenen Rollwinkel geregelt hat. Die Nickachse des Flugzeugs wurde über die gespeicherten Piloteneingaben gesteuert. Die Phasenverschiebung des Rollwinkels ϕ kann toleriert werden.

7.4 Analyse der Simulationsergebnisse

7.4.1 Toleranzen und ihre Anwendbarkeit

Die Zulassungsvorschriften für zivile Trainingssimulatoren findet man in den Joint Aviation Requirements [3]. Generell wird dort verlangt, daß die Performance und die Handhabung eines Simulators anhand eines Vergleichs von Simulatortestdaten und Flugtestdaten objektiv nachgewiesen wird. Diese Vergleiche müssen mit dem vollständigen Simulationsmodell durchgeführt werden (*end-to-end test*). Es ist nicht gestattet, nur die einzelnen Subsysteme zu testen.

Für die Vergleiche werden Manövertyp und Toleranzen angegeben. Jedoch genügt es nicht, daß die Simulatortests innerhalb der Toleranzen liegen. Darüber hinaus muß der Simulator die Flugleistung und die Steuerbarkeit des echten Flugzeuges wiedergeben. Üblicherweise werden die Vergleiche in Form von übereinandergelegten Zeitverläufen durchgeführt. Dort wo die relevanten Bewegungsgrößen außerhalb der Toleranz liegen, ist eine fundierte Analyse und Beurteilung durch einen Ingenieur erforderlich, um trotzdem eine Zulassung erhalten zu können (*engineering judgement*). Denn man sollte im Kopf behalten, daß im gesamten Datenverarbeitungsprozeß (Datenaufzeichnung im Flugversuch, Simulation, Auswertung) Ungenauigkeiten enthalten sind.

Außerdem müssen die Simulationsmodelle in erster Linie echtzeitfähig sein, und können dadurch nicht die Qualität bieten, wie es numerisch möglich wäre.

Nickwinkel θ	± 1.5 deg
Nickrate q	± 2.0 deg/sec
Lastvielfaches n_z	± 0.1 g

Tabelle 7.1: Toleranzen für dynamische Manöver (*short period*) gemäß JAR

7.4.2 Analyse des Trimmzustands allgemein

Vor der Durchführung eines Flugmanövers trimmt der Pilot das Flugzeug normalerweise in einen stationären Zustand aus. Für den stationären Trimmzustand gelten besondere Bedingungen. Die Drehbeschleunigungen und die Längsbeschleunigungen sind dann normalerweise Null (abgesehen von der auf das Flugzeug wirkenden Erdbeschleunigung).

Im Trimmzustand werden einige Derivative des aerodynamischen Modells nicht angesprochen. Dies sind die dynamischen Derivative, und auch viele Derivative der Seiten-

bewegung. Denn die Steuerflächen von Querruder, Seitenruder und Spoiler etc. sind normalerweise nicht ausgeschlagen.

Bei der Analyse von Schwachstellen im aerodynamischen Modell kann man sich dadurch auf ein paar wenige Derivative beschränken.

7.4.3 Analyse von Manövern allgemein

Die Analyse von Manövern kann nur sorgfältig durchgeführt werden, wenn zuvor der Trimmzustand mit den Meßwerten aus dem Flugversuch übereinstimmt. Denn hier wirken neben den statischen zusätzlich viele dynamische Derivative. Das Augenmerk trifft dann auch diese dynamischen Anteile an den Gesamtbeiwerten, falls Abweichungen während des Manövers festgestellt werden.

Eine Besonderheit muß bei Flugzeugen mit Flugreglern beachtet werden. Meistens werden die gemessenen Drehraten in den Flugregler zurückgeführt. Die Ausschläge der primären Steuerflächen hängen also nicht nur von dem Pilotenkommando ab, sondern auch von der Bewegung und der Reaktion des Flugzeugs.

Bei den Vergleichen von Meßwerten aus dem Flugversuch und Simulationsergebnissen wird man bei solchen Flugzeugen immer eine sehr gute Übereinstimmung der Drehraten beobachten. Mängel im aerodynamischen Modell zeigen sich aber dann in unterschiedlichen Ruderausschlägen.

7.4.4 Beispiele

Anhand zweier Beispiele wird nun demonstriert, welche Auswirkungen die Optimierung des aerodynamischen Modells auf das Flugverhalten des Simulationsmodells hat. Die folgenden Manöver wurden mit der in Abschnitt 7.1 erläuterten Technik nachsimuliert:

1. Wind Up Turn, zur Demonstration der Verbesserung der Modellparameter
2. Landeklappen ausfahren, zur Demonstration der Verbesserung der Modellstruktur

In den Graphiken werden immer die Flugmeßdaten und die Ergebnisse der Flugsimulation übereinandergelegt. Dabei bedeutet:

- ▽ — Flugsimulation
- ◇ — Flugversuch (FTI)

7.4.4.1 Manöver 1: Wind Up Turn

Das Manöver wurde in einer Höhe von 20.000 ft bei einer Fluggeschwindigkeit von 200 kts durchgeführt. Die Landeklappen befanden sich in Manöverstellung. Zu Beginn des Manövers rollte der Pilot das Flugzeug in eine Schräglage von 60° , um dann kontinuierlich den Anstellwinkel auf etwa 20° zu erhöhen. Das ganze Manöver dauerte 16 Sekunden.

In den Graphiken sind jeweils von oben nach unten zu sehen:

1. Anstellwinkel α
2. Normales Lastvielfaches n_z
3. Höhenruderausschlag η
4. Nickrate q

Hinweis: Das normale Lastvielfache n_z ist bei diesem Manöver größer als 1.0, da das Manöver bei etwa 60° Schräglage geflogen wurde.

In Abbildung 7.4 werden zunächst die Simulationsergebnisse, unter Verwendung des aerodynamischen Basismodells, mit den Flugmeßdaten verglichen.

Der Simulator wurde gut ausgetrimmt. Der Anstellwinkel hat im Trimmzustand nur geringe Abweichungen (etwa $0.5-1.0^\circ$). Während des Manövers wächst die Differenz vor allem im hohen Anstellwinkelbereich auf etwa 2° . Diese Abweichung liegt außerhalb der festgelegten Toleranz von 1.5° .

Das normale Lastvielfache zeigt eine gute Übereinstimmung. Der Höhenruderausschlag weicht im Trimmzustand etwa 1° ab. Die Differenz wird während des Manövers kleiner. In der Simulation ist das Ruder etwas weiter nach oben ausgeschlagen. Das deutet darauf hin, daß das aerodynamische Nullmoment zu negativ ist.

Die Nickraten von Simulation und Flugversuch stimmen gut überein. Dies ist, wie schon erwähnt, bei Flugzeugen mit Fly-by-wire Steuerung und rückgeführter Drehrate fast immer zu beobachten.

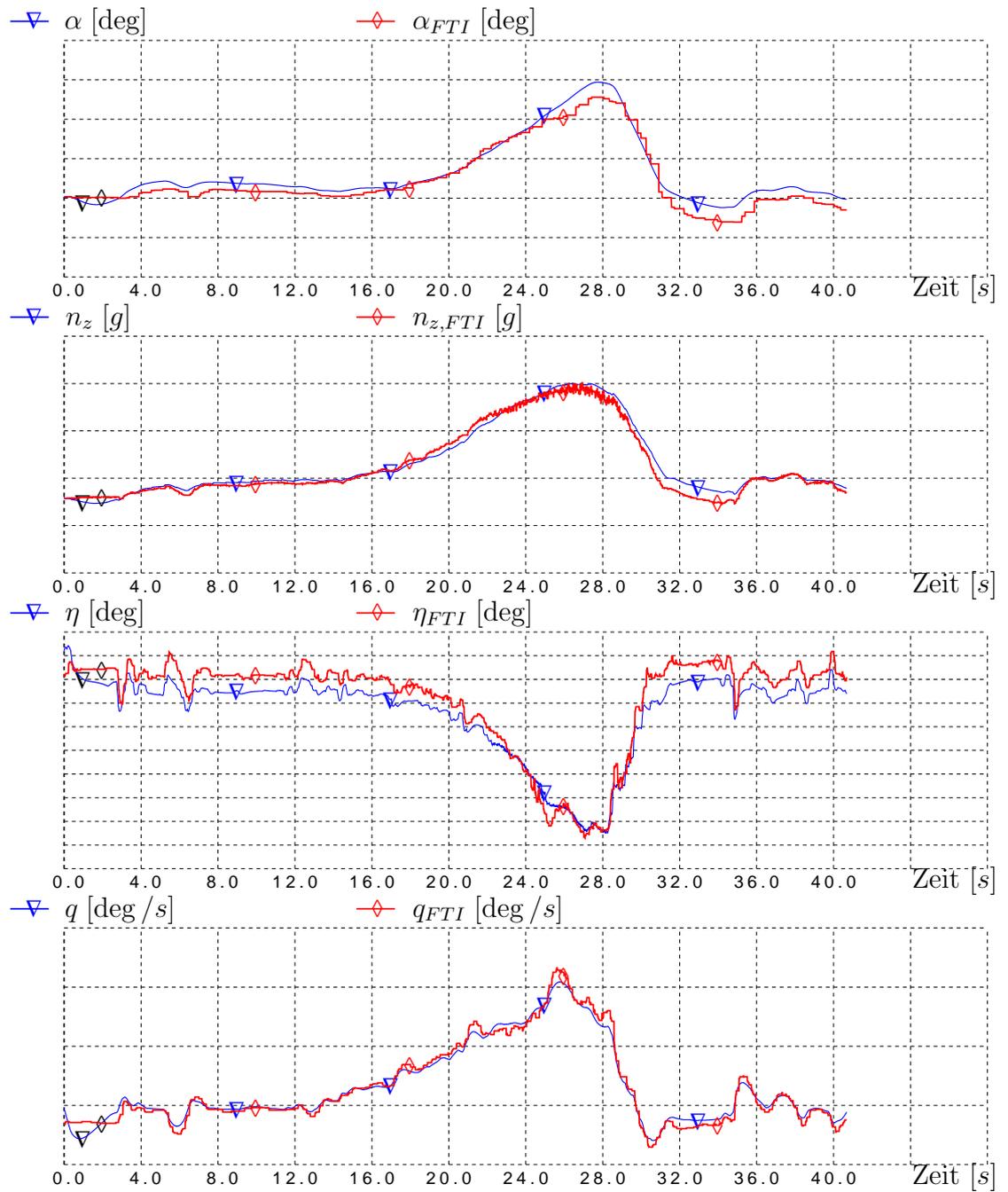


Abbildung 7.4: WUT mit Referenzdatensatz

Das Ziel ist nun die Verbesserung von Anstellwinkel (durch Optimierung von C_A) und Höhenruderausschlag (durch Optimierung von C_m)

Nach einem Training von 20 Epochen lag das Ergebnis vor (siehe Abbildung 6.10). Mit diesem optimierten aerodynamischen Modell konnte eine verbesserte Simulation erzielt werden. Das Ergebnis ist in Abbildung 7.5 zu sehen.

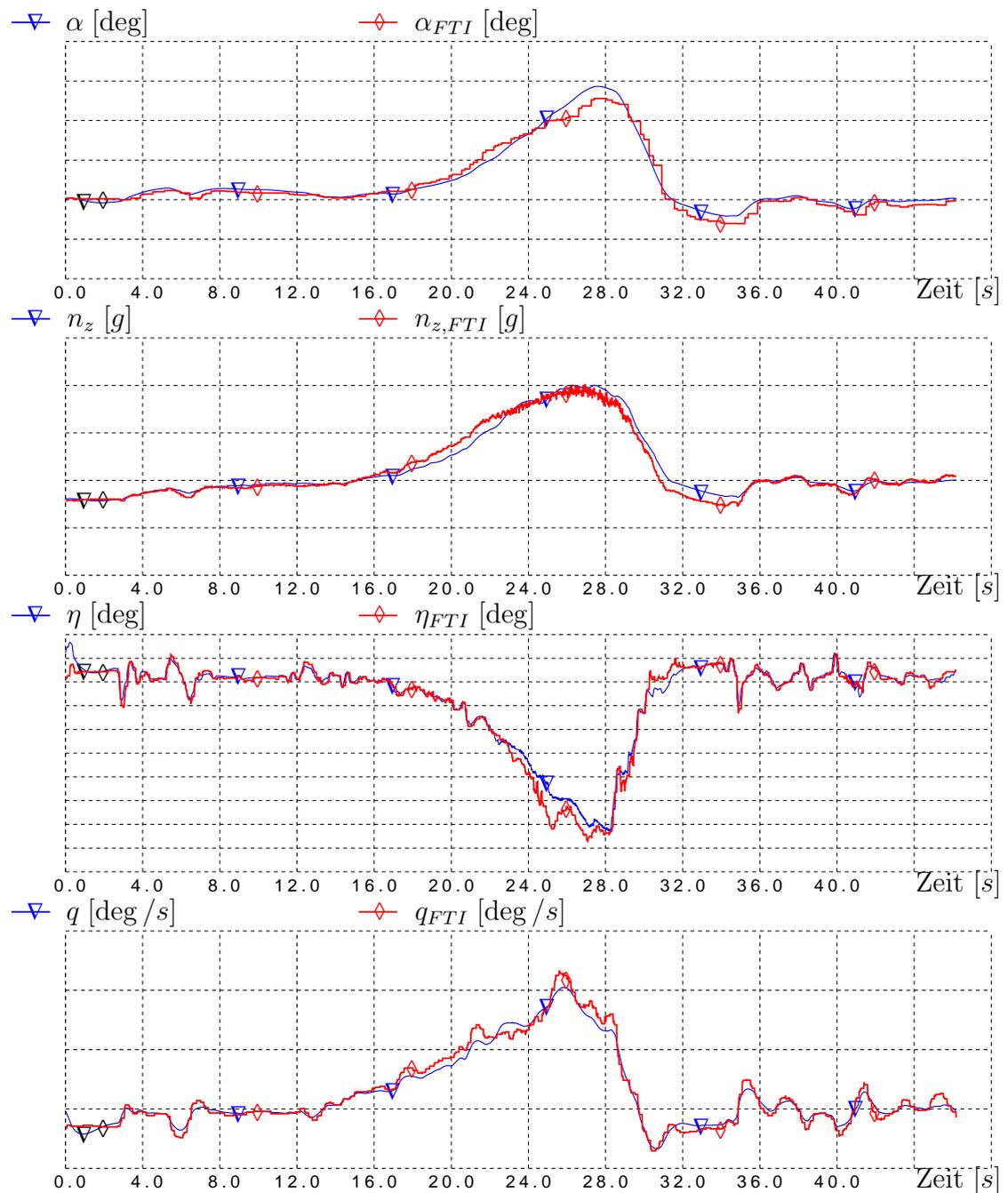


Abbildung 7.5: WUT mit optimiertem Modell

Der Anstellwinkel verbesserte sich im gesamten geflogenen Bereich. Im Trimmzustand ist fast kein Unterschied zu den Flugversuchsdaten auszumachen. Auch der Maximalwert der Simulation weicht jetzt nur noch 1.5° ab. Die Verbesserung hier ist zwar gering, aber die Differenz liegt nun innerhalb der Toleranz.

Der Verlauf des normalen Lastvielfachen blieb weitgehend unverändert. Für den

Höhenruderausschlag erhält man bessere Trimmwerte und eine geringfügige Verschlechterung während des Manövers (etwa 0.2°).

Die Nickraten von Simulation und Flugversuch stimmen, wie vorher schon, gut überein.

Das Ziel der Optimierung wurde erreicht: Die Verläufe von Anstellwinkel und Höhenruderausschlag wurden verbessert und alle Werte befinden sich nun in Toleranz.

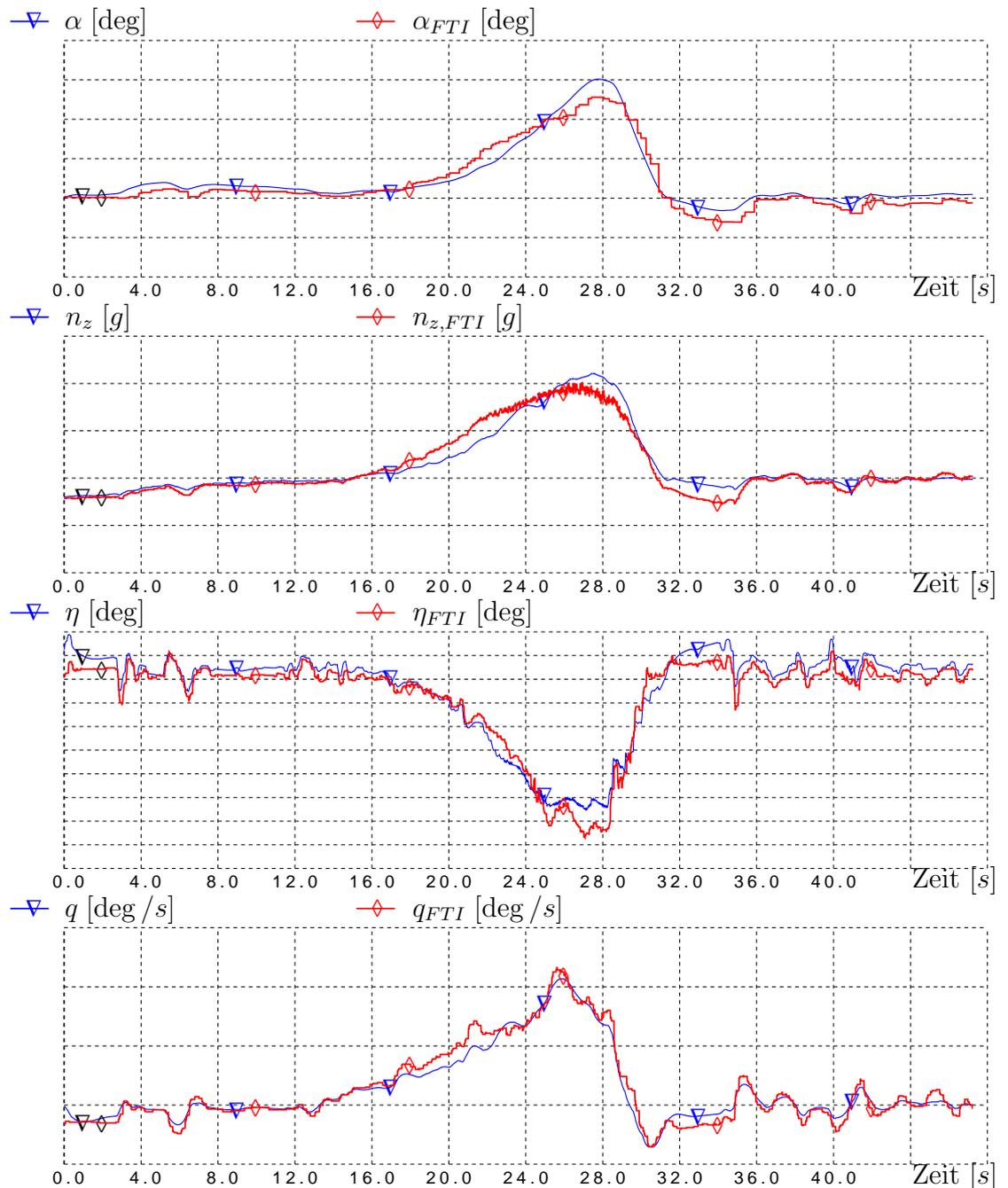


Abbildung 7.6: WUT mit neu generiertem Modell

In Abbildung 7.6 ist das Ergebnis der Simulation mit einem neu generierten aerodynamischen Modell (siehe auch Abbildung: 6.13). Die Übereinstimmung von Simulations- und Flugmeßdaten ist beachtlich. Dies ist auf die gute Identifizierung der Hauptanteile C_{A0} und C_{m0} zurückzuführen.

Der Höhenruderausschlag stimmt eigentlich, im Vergleich zur Simulation mit optimiertem Modell, zu gut mit den Meßdaten überein. Denn der Verlauf von $C_{m\eta}$ weicht von dem optimierten Verlauf weit ab. Eventuell sorgt noch ein Anteil von C_{m0} dafür, daß eine gute Anpassung bei speziell diesem Manövertyp existiert.

Für den Verlauf des normalen Lastvielfachen erkennt man allerdings große Abweichungen mit wechselndem Vorzeichen.

Die Nickrate q des Simulators paßt wieder gut zu den Flugmeßdaten (Grund: Flugregler mit q - Rückführung).

Bis auf die Abweichung des maximalen α - Wertes, liegen die Zeitverläufe innerhalb der Toleranzen.

7.4.4.2 Manöver 2: Landeklappen ausfahren

Dieses Manöver wurde in einer Höhe von 20.000 ft bei einer Fluggeschwindigkeit von 300 kts durchgeführt. Die Landeklappen waren anfangs eingefahren und wurden dann in Manöverstellung gebracht. Das Ausfahren der Landeklappen dauerte etwa zwei Sekunden. Ansonsten befand sich das Flugzeug im getrimmten Geradeausflug.

In den Graphiken sind jeweils von oben nach unten zu sehen:

1. Anstellwinkel α
2. Normales Lastvielfaches n_z
3. Höhenruderausschlag η
4. Landeklappenausschlag δ_f

In Abbildung 7.7 werden zunächst die Simulationsergebnisse, unter Verwendung des aerodynamischen Referenzmodells, mit den Flugmeßdaten verglichen.

Der Simulator wurde gut ausgetrimmt. Der Anstellwinkel hat im Trimmzustand nur geringe Abweichungen (etwa $+0.5^\circ$). Nach dem Ausfahren der Landeklappen erkennt man eine Abweichung von etwa -0.5° . Diese Vorzeichenumkehr ist auf Mängel in den aerodynamischen Datensätzen (C_A) für beide Klappenstellungen zurückzuführen.

Der Verlauf des normalen Lastvielfachen stimmt anfangs gut mit den Flugmeßdaten überein. Dann sieht man aber einen heftigen Sprung in den Simulationswerten. Der Grund dafür liegt im aerodynamischen Referenzmodell. Es gibt dort unterschiedliche Datensätze für die jeweiligen Klappenstellungen. Zwischen diesen Datensätzen wird einfach umgeschaltet. Dadurch ergeben sich zwangsweise Sprünge bei Änderung

der Landeklappenstellung.

Für den Höhenruderausschlag erkennt man einen konstanten offset von etwa 0.5° .

Das Ziel ist nun die Verbesserung von Anstellwinkel (durch Optimierung von C_A), Höhenruderausschlag (durch Optimierung von C_m) und das Vermeiden von Sprüngen im n_z .

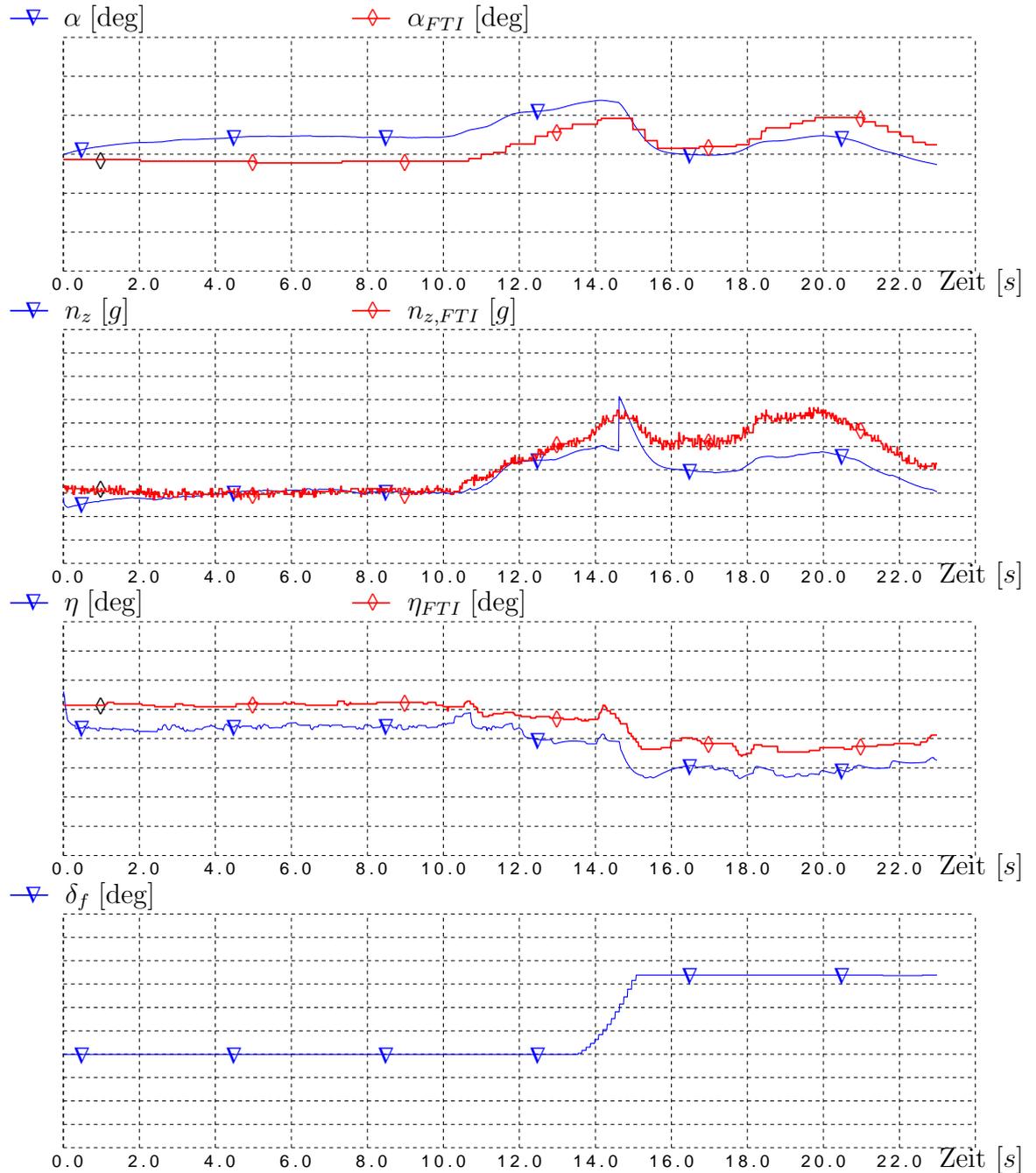


Abbildung 7.7: Landeklappen ausfahren mit Referenzmodell

Mit dem optimierten aerodynamischen Modell (Abbildung 6.10) konnte eine verbesserte Simulation erzielt werden. Das Ergebnis ist in Abbildung 7.8 zu sehen.

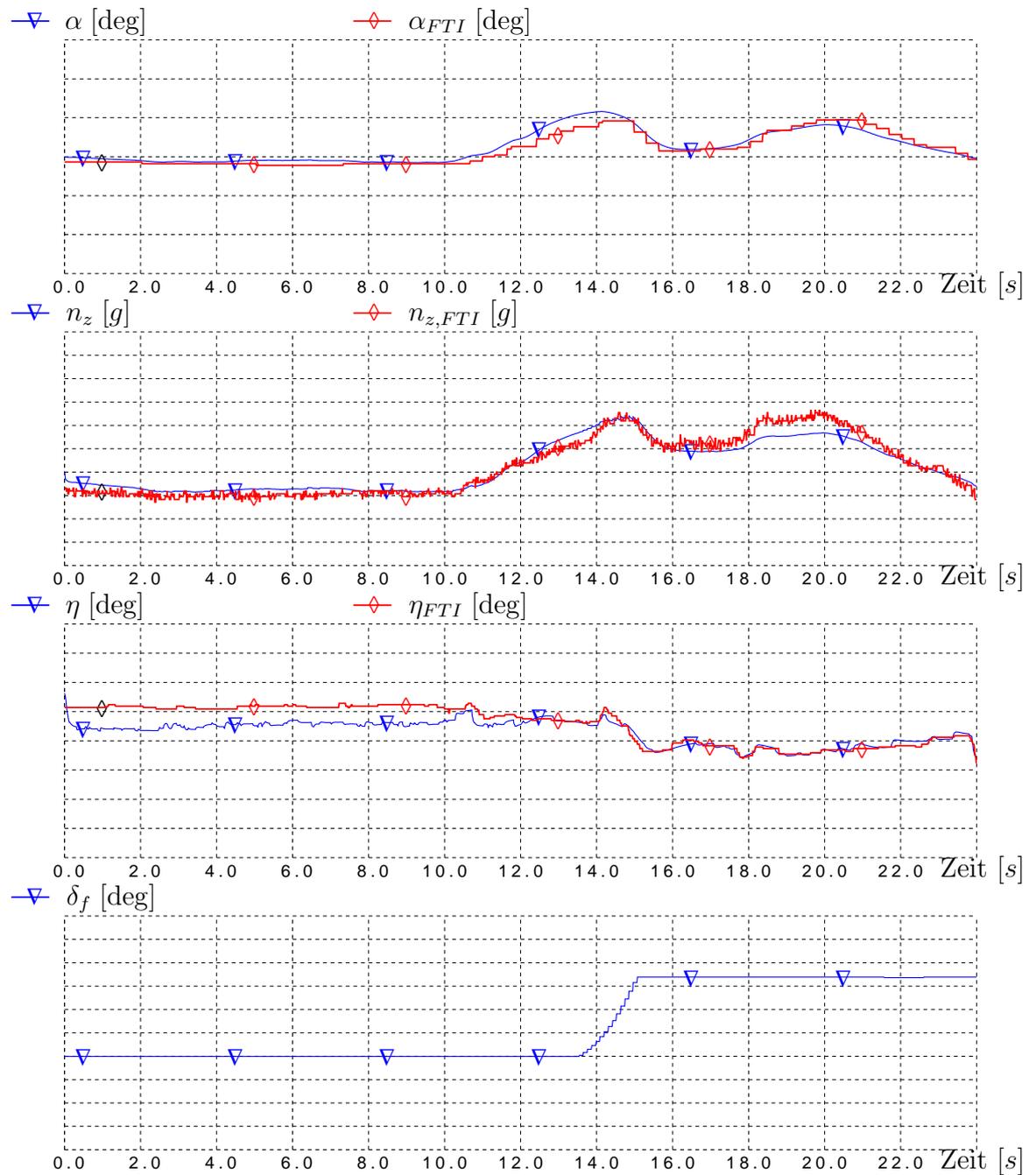


Abbildung 7.8: Landeklappen ausfahren mit optimiertem Modell

Der Anstellwinkel und der Höhenruderausschlag verbesserten sich ebenso wie im ersten Beispiel (Abschnitt 7.4.4.1). Es fällt auf, daß im ersten Teil des Manövers (Landeklappen eingefahren) der Ausschlag des Höhenruders in der Simulation nur wenig

verbessert wurde, jedoch bei ausgefahrenen Landeklappen sehr gut übereinstimmt. Das liegt wahrscheinlich daran, daß im Flugversuch nur wenige Manöver mit eingefahrenen Landeklappen durchgeführt wurden. Deswegen enthält der Lerndatensatz für die Optimierung des Neuronalen Netzes zu wenig Meßwerte für diesen Flugzustand.

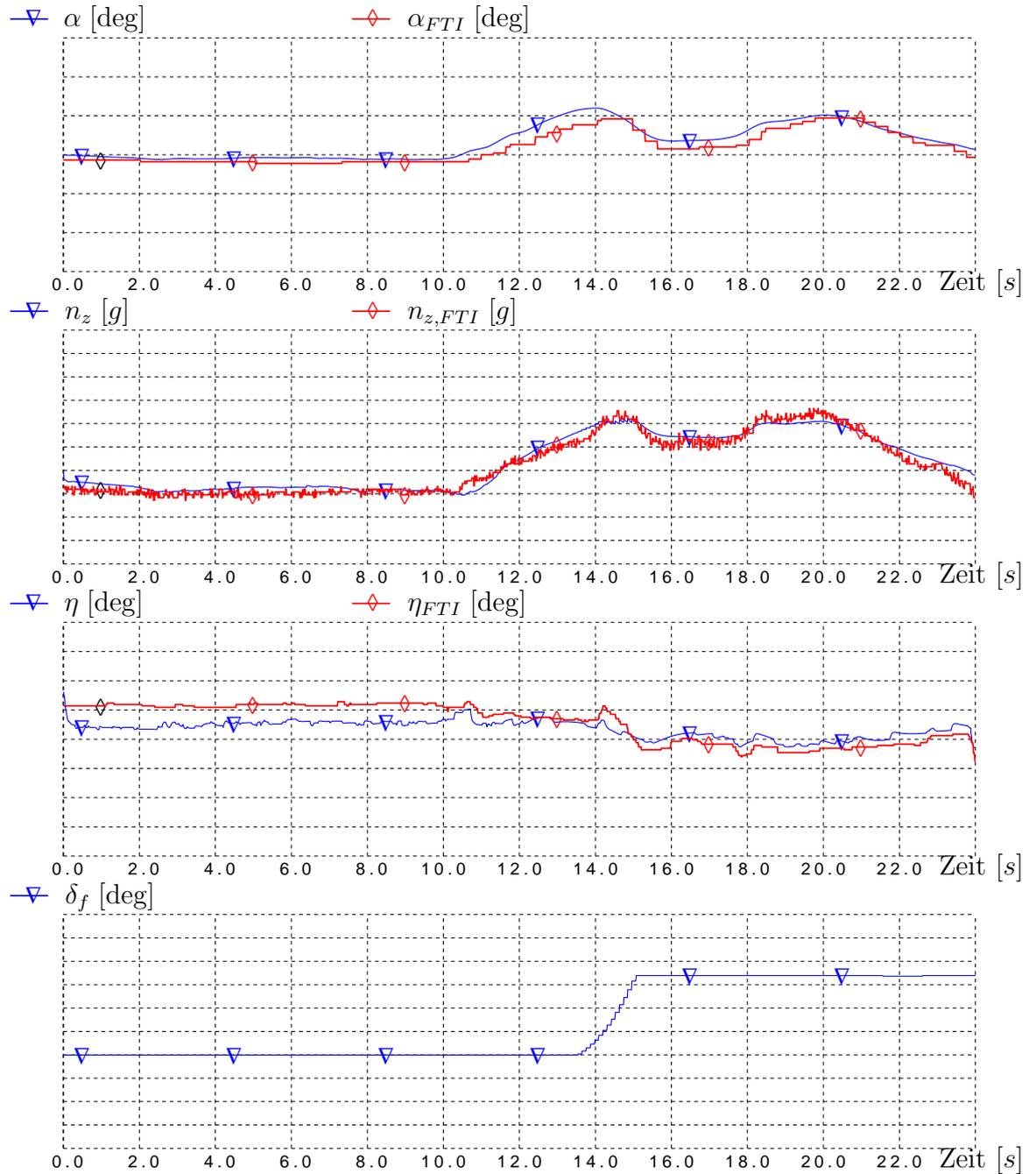


Abbildung 7.9: Landeklappen ausfahren mit neu generiertem Modell

Das Hauptziel der modifizierten Modellstruktur und der Optimierung der aerodynamischen Parameter war das Vermeiden des Sprunges im n_z . Dies gelang durch

Hinzunahme des nichtlinearen Einflußparameters in Form eines Eingabeneurons. Das hat auch positive Auswirkungen auf die weitere Flugbahn.

Zuletzt noch die Simulation des Manövers mit dem neu generierten Modell. In Abbildung 7.9 ist wieder die erstaunlich gute Übereinstimmung von Simulations- und Flugmeßdaten zu sehen.

Der Verlauf des Anstellwinkels ist so gut wie bei der Simulation mit dem optimierten Modell. Das normale Lastvielfache springt natürlich nicht, dank der Modellstruktur. Der Verlauf ist etwa so gut wie in Abbildung 7.8.

Der Höhenruderausschlag ist etwas besser als bei Verwendung des Referenzmodells, aber schlechter als es das Optimierte Modell bieten kann.

7.5 Flugsimulation mit einem Neuronalem Netz

In diesem Abschnitt wird auf die besonderen Anforderungen in der Echtzeitsimulation und die Möglichkeit der Online Adaption eingegangen. Es wird gezeigt, wie ein aerodynamisches Modell in Form des vorgestellten Neuronalem Netzes in die Simulations-Software integriert werden kann, und was bei den Schnittstellen beachtet werden muß.

7.5.1 Softwaremodelle

Ein vollständiger Entwicklungssimulator, in dem ein Neuronales Netz als adaptives Modell integriert ist, stellt heute noch nicht den Stand der Technik dar. Es bietet sich aber an, einen Flugregler oder ein Aerodynamisches Modell als adaptives Modell zu gestalten und zu integrieren.

Die aerodynamischen Beiwerte werden bei klassischen Modellen in Tabellen abgelegt. Während das Simulationsprogramm läuft, werden diese Werte aus den Tabellen einfach ausgelesen bzw. zwischen den Stützstellen linear interpoliert. Aus den einzelnen Anteilen werden dann die Gesamtbeiwerte summiert.

In einem Neuronalem Netz sind keine vergleichbaren Tabellen abgelegt. Vielmehr werden die Beiwerte durch die Berechnung einer nichtlinearen Funktion zur Verfügung gestellt.

Aus den Beiwerten ermittelt man schließlich die aerodynamischen Kräfte und Momente, die wiederum im Simulationskern weiterverarbeitet werden.

Für die Integration des Neuronalem Netzes in die Simulationsumgebung sollte der Sourcecode zur Verfügung stehen. Nachdem keine kommerzielle Software die spezielle Netzwerkstruktur (Eingabeverbindungen) unterstützt, mußte das Modell

neu programmiert werden. Es bietet sich an, eine objektorientierte Hochsprache, wie z.B. C++, zu verwenden. Folgende Objekte können dann als Klasse definiert werden: Neuron, Schicht, Gruppe, Verbindung.

7.5.2 Anforderungen an das Neuronale Modell

Das Interface des Neuronalen Netzes ist einfach aufgebaut. Als Eingang werden alle Eingabeneuronen und Eingabeverbindungen definiert. Alle Ausgabeneuronen bestimmen zusammen den Ausgang. Für ein adaptives Modell kommt als Eingang noch ein Sollwert pro Ausgang hinzu.

Wenn die Verbindungsgewichte des Neuronalen Netzes während einer Simulation optimiert werden sollen, dann ist noch die Integration eines Lernalgorithmus erforderlich (siehe auch Abschnitt 7.5.3).

Wie bei anderen Modellen auch, sollte eine Bereichsüberschreitung der Eingabewerte vermieden werden. Bei den Eingabeneuronen stellt dies zwar kein großes Problem dar, weil die nichtlinearen Aktivierungsfunktionen nur einen kleinen Wertebereich haben. Die Eingabeverbindungen hingegen multiplizieren die Eingabewerte zu den Gruppenausgängen. Dadurch können physikalisch unsinnige Werte berechnet werden. Der Wertebereich der Eingabeverbindungen sollte also dem System (hier: Flugzeug) angepaßt sein.

Für ein Neuronales Netz mit stetig differenzierbaren Aktivierungsfunktionen können sehr einfach die partiellen Ableitungen nach den Eingabewerten berechnet werden [5]. Somit kann das Modell an dieser Stelle linearisiert werden. Dadurch können Stabilitätsbetrachtungen durchgeführt werden. Der Gradient $\delta C_{m0}/\delta\alpha$ ist zum Beispiel das Maß für die statische Längsstabilität eines Flugzeugs.

Aufgrund der einfachen Mathematik und der kompakten Größe eines Modularen Neuronalen Netzes ist es kein Problem, eine Echtzeit- Flugsimulation mit so einem Netzwerktyp auszurüsten. Sofern die CPU- Rechenleistung ausreichend ist, gibt es auch keine Beschränkung bzgl. der Taktrate nach oben hin. Auf heutigen Rechnern ist eine Berechnungsdauer von 0.1 Millisekunden für die vorgestellten Modelle erreichbar.

7.5.3 Online Adaption

Zu Beginn dieser Arbeit wurden Anforderungen an eine Methode zur Parameteridentifikation gestellt. Darunter befand sich auch die Forderung nach einer Online und Offline Tauglichkeit. Alle im Kapitel 6 vorgestellten Ergebnisse wurden mit einer

Offline Optimierung erzielt.

Die Online Fähigkeit der hier vorgestellten Methode wurde in dieser Arbeit nicht unter Beweis gestellt. Es gibt jedoch zwei unterschiedliche Lösungsansätze, die sehr erfolgversprechend sind:

1. Durch Organisation des Lerndatensatzes, ähnlich der Komprimierungsmethode aus Abschnitt 5.2.2, erhält man eine gleichmäßige Verteilung der Lernmuster. Dabei geht keine Information über das System verloren. Zu Beginn des Trainings enthält der Lerndatensatz entweder keine Wertepaare oder noch welche aus vorhergehenden Versuchen.
2. Mit Hilfe der lokal trainierenden RBF- Netze wird verhindert, daß das Lernen neuer Muster die gelernten alten Assoziationen zerstört. Dieses Problem wird auch als das sogenannte *Stabilitäts- Plastizitäts- Dilemma* bezeichnet. Details zu dieser Methode findet man zum Beispiel in [22].

Die Online Fähigkeit ist auch ein interessanter Aspekt für eine Simulation, die parallel zum Flugversuch abläuft. Dadurch können Simulationsmodelle schon während des Versuchs verbessert werden und nicht wie üblich erst hinterher. Es ist dann ratsam die Ausgabe des Neuronalen Netzes zu beobachten, um Veränderungen sofort erkennen zu können.

8 Zusammenfassung und Ausblick

Die Möglichkeit, mit Hilfe eines Modularen Neuronalen Netzwerks Simulationsmodelle zu generieren oder vorhandene Modelle zu optimieren, wurde anhand eines generischen Versuchs und in der Praxis am Beispiel eines Kampfflugzeugs aufgezeigt. Die Ergebnisse der Parameteridentifikation wurden analysiert und durch eine numerische Simulation untermauert. Dabei wurde das Neuronale Netz in die Simulationsumgebung des EADS Entwicklungssimulators integriert und dessen Eignung als echtzeitfähiges Simulationsmodell demonstriert.

Die Netzwerkstruktur des hier verwendeten Modularen Neuronalen Netzwerks stellt ein Novum dar. Die Verknüpfung der Module (Neuronengruppen) mit dem Ausgabe-neuron über sogenannte Eingabeverbindungen ermöglicht erst die Identifizierung von nichtlinearen Derivativen.

Als Optimierungsverfahren wurde der Backpropagation- Algorithmus gewählt, der für die Verwendung von Eingabeverbindungen modifiziert werden mußte. Ein einfaches Beispiel zeigte die Funktionsweise der iterativen Gewichthsoptimierung durch Bestimmung des Fehlergradienten.

Rückblickend auf die gestellten Anforderungen an eine Methode zur Parameteridentifikation, erweist sich das vorgestellte Verfahren als praxisnahe und flexibel. Wie es sich gegenüber den alternativen Methoden behaupten kann, muß ein direkter Vergleich der Optimierungsergebnisse bei der selben Aufgabenstellung zeigen.

Es wurde gezeigt, daß sowohl statische Nullbeiwerte und dynamische Derivative (z.B. Nickdämpfung C_{mq}) gleichermaßen identifiziert werden können. Die Derivative können sowohl linear gewählt werden als auch von mehreren Eingangsgrößen nichtlinear abhängig gemacht werden. Darüber muß der Ingenieur bei der Modellierung des Neuronalen Netzes entscheiden.

Der Automatismus bei der Datenverarbeitung ist vergleichsweise groß, da die Meßdaten nicht manuell bearbeitet werden müssen. Zum Beispiel entfällt aufgrund der nichtlinearen Identifikation das Extrahieren einzelner Flugmanöver aus den Flugmeßdaten. Die Optimierungsergebnisse müssen anschließend auch nicht weiterverarbeitet werden, da das Modulare Neuronale Netz ein echtzeitfähiges Simulationsmodell darstellt.

Zudem ist es möglich die Module zu analysieren und zum Beispiel die Funktionsgra-

phen der Parameter zu zeichnen. Im direkten Vergleich mit dem Ausgangszustand (Referenzmodell) können so die Unterschiede aufgrund der Optimierung der Netzwerkgewichte schnell erfasst werden.

Die Resultate der Parameteridentifikation boten, im Falle der Optimierung eines Basismodells, eine akzeptable Qualität. Bei der Generierung eines neuen Modells übertrafen sie die Erwartungen. Die Gefahr ist hier doch sehr groß, daß der Backpropagation-Algorithmus in einem lokalen Minimum endet, obwohl der Netzwerkfehler noch sehr groß ist. Die Zeitverläufe der relevanten Simulationsgrößen stimmen aber mit den Vorgabewerten so gut überein, daß das gesteckte Ziel (*Pilot Familiarisation Simulation*) mehr als nur erreicht wurde.

Die gezeigten Identifikationsergebnisse wurden alle offline im *Postprocessing* erstellt. Die Möglichkeit der online Identifikation (hier: während des Fluges) wurde aufgezeigt. Die Robustheit der Methode resultiert aus einer geringen Schrittweite des iterativen Gradientenverfahrens. Dies ist bei der Online Identifikation auch nötig, da die Übertragung der Meßwerte über eine Funkstrecke fehlerträchtig ist.

Den gestellten Anforderungen wird diese Methode also weitestgehend gerecht. Die Nachteile des Verfahrens wurden aber auch deutlich gemacht. Teilweise ist sehr viel Rechenzeit nötig, um brauchbare Ergebnisse zu erzielen. Das liegt in der Natur der Gradientenabstiegsverfahren, die bei abnehmendem Fehler immer langsamer konvergieren. Die Universalität des Verfahrens wird durch die Modularisierung des Neuronalen Netzes, im Vergleich zu einem MLP, gemindert. Im Falle der Identifizierung von aerodynamischen Parametern ist das Einbringen von A-Priori-Wissen aber wichtiger, als die universelle Anwendbarkeit für unbekannte Systeme.

Ein weiterer Nachteil von Neuronalen Netzen ist die große Tendenz beim Training der Gewichte in ein lokales Minimum zu fallen. Bei den bekannten MLP Netzen ist das Erreichen des globalen Minimums fast ausgeschlossen. Mit der Technik der Modularität ist die Wahrscheinlichkeit, dieses Ziel zu finden, viel größer.

Die Reproduzierbarkeit der Ergebnisse wurde durch viele Versuche bestätigt. Sie hängt, wie bei anderen Verfahren, nur von den Anfangswerten der zu optimierenden Parameter und den zugrundeliegenden Meßdaten ab. Insofern unterscheiden sich die Trainingsverfahren der Neuronalen Netze nicht von anderen iterativen Optimierungsmethoden.

Es wäre noch interessant zu untersuchen, wie reproduzierbar die präsentierten Ergebnisse mit denen der vorgestellten alternativen Methoden sind.

Der wirtschaftliche Vorteil bestimmt den Einsatz von neuartigen Tools. Aufgrund des hohen Automatismus und dem geringen Aufwand beim Pre- und Postprocessing kann sich das vorgestellte Verfahren gegenüber den etablierten Verfahren durchsetzen. Die Praxistauglichkeit muß in punkto Rechenzeit noch erhöht werden. Dies könnte durch

Einsatz modifizierter Backpropagationverfahren (z.B. Nutzen der zweiten Ableitung) erzielt werden.

Mehr Universalität könnte dem Verfahren auch ein breiteres Einsatzspektrum eröffnen. Es ist denkbar, andere Techniken zur Strukturfindung hinzuzunehmen. Eine vielversprechende Idee ist auch das Optimieren der partiellen Ableitungen der nichtlinearen Derivative. Dadurch werden nicht nur Funktionswerte, sondern auch die Steigungen optimiert.

Der Anwendungsbereich könnte sich auf andere Ingenieurdisziplinen erstrecken. Zum Beispiel die Modellierung der Strömungserfassung mit Hilfe eines Airdatensystems. Die Aufgabe ist hier, mit Hilfe mehrerer Drucksensoren die Anströmwinkel einer Flugzeugnase zu bestimmen. Es ist denkbar, daß die Kalibrierung des Systems mit Hilfe von Modularen Neuronalen Netzen erfolgen kann.

Ebenso interessant wäre ein Flugregler auf Basis eines Modularen Neuronalen Netzes, der im Fluge seine, vom Anstellwinkel oder Machzahl nichtlinear abhängigen, Verstärkungsfaktoren anpasst. Die Stabilität des geregelten Systems muß natürlich nachgewiesen werden.

Erfreulich ist, daß nun nach Abschluß der Grundlagenforschung, der Einsatz des Verfahrens im Eurofighter Entwicklungsprozeß gewünscht wird. Bedingung dafür ist, daß bereits identifizierte Parameter mit der Methode der Modularen Neuronalen Netze kostengünstiger reproduzierbar sind. Ausgehend von den bisherigen Erfolgen, kann man diese Aufgabe optimistisch angehen.

Literaturverzeichnis

- [1] DIN9300: *Teil 1 Flugmechanik*, Normenstelle Luftfahrt, 72022 Leinfelden, Oktober 1990 [5.1.1](#)
- [2] W.G. SCHWEIKHARD, D.J. SCHUELER: *Why Simulators don't fly like the Airplane - Data*, Kohlman System Research, Lawrence, Kansas [1.1](#)
- [3] WESTWARD DIGITAL LIMITED: *Joint Aviation Requirements, JAR-STD 1A Aeroplane Flight Simulators*, United Kingdom, 1997 [1.1](#), [7.4.1](#)
- [4] A. ZELL: *Simulation Neuronaler Netze*, Addison-Wesley, ISBN 3-89319-554-8, 1994 [1.3](#), [5.3](#)
- [5] R. ROJAS: *Theorie der neuronalen Netze*, Springer-Lehrbuch Berlin, ISBN 3540563539, 1996 (4.Auflage) [1.3](#), [5.3](#), [5.4.6](#), [7.5.2](#)
- [6] T. GROHS, O. WAGNER: *Erkennung von Hubschrauber- Flugzuständen mit Hilfe eines Neuronalen Netzes*, Diplomarbeit am Lehrstuhl für Flugmechanik und Flugregelung, Technische Universität München, 1997 [1.3](#)
- [7] L. A. ZADEH: *From Circuit Theory to System Theory*, Proceedings of the IRE, Vol. 50, Mai 1962, pp. 856-865 [1.4](#)
- [8] P.G. HAMEL, R.V. JATEGAONKAR: *Evolution of Flight Vehicle System Identification*, Journal of Aircraft Vol.33, No. 1, January-February 1996 [1.4](#), [2.2.1](#)
- [9] R. JATEGAONKAR, W. MÖNNICH, D. FISCHENBERG, B. KRAG: *Identification of C-160 Simulator Data Base from Flight Data*, Proceedings of the 10th IFAC Symposium on System Identification, pp. 3.67-3.74, 1994 [1.4](#)
- [10] J.R. RAOL, R.V. JATEGAONKAR: *Aircraft Parameter Estimation Using Recurrent Neural Networks - A Critical Appraisal*, AIAA Atmospheric Flight Mechanics Conference, 1995 [1.4](#), [2.3.2](#)
- [11] R. BAVA, G.T. HOARE, G. GARCIA-MESURO, H.-C. OELKER: *Recent Experiences on Aerodynamic Parameter Identification for EUROFIGHTER at Alenia, British Aerospace, CASA and Daimler-Benz Aerospace*, NATO-RTO Systems Concepts and Integration Panel, Symposium on System Identification for Integrated Aircraft Development and Flight Testing, 5-7 May 1998, Madrid Spain [2.2.1](#)

- [12] L.C. ANDERSON: *Robust Parameter Identification for Non-linear Systems using a Principal Components Regression Algorithm*, AIAA paper 85-1766 [2.2.1](#)
- [13] H.U. SCHÄUFELE: *Parameteridentifikation mit einem modifizierten Maximum-Likelihood Verfahren*, Fortschritt-Ber. VDI- Zeitschrift, Reihe 8, Nr. 40, 1981 [2.2.1](#)
- [14] H.U. SCHÄUFELE: *Bewegungsgleichungen eines Flugzeugs und Anwendung von Parameteridentifizierungsverfahren*, Institut für Flugmechanik, Technische Universität München, 1974 [2.2.1](#), [5.1.3](#)
- [15] H. PREISLER, H. SCHÄUFELE: *Equation Decoupling - A New Approach to the Aerodynamic Identification of Unstable Aircraft*, AIAA 5th Flight Test Conference, Ontario, CA 1990, paper 90-1276 [2.2.1](#)
- [16] R.A. HESS: *On the use of Back Propagation with Feed-Forward Neural Networks for the Aerodynamic Estimation Problem*, AIAA Atmospheric Flight Mechanics Conference, 1993 [2.3](#), [2.3.1](#)
- [17] B. KOSKO: *Fuzzy systems as universal approximators*, Proceedings IEEE International Conference on Fuzzy Systems, 1992, San Diego [2.3.3](#)
- [18] C. VON ALTROCK: *Fuzzy Logic and NeuroFuzzy Applications Explained*, Prentice Hall, ISBN 0-13-368465-2, 1995 [2.3.3](#)
- [19] D. NAUCK, R. KRUSE: *A Neuro-Fuzzy Approach to Obtain Interpretable Fuzzy Systems for Function Approximation*, Proceedings IEEE International Conference on Fuzzy Systems 1998, Anchorage, AK, pp. 1106-1111 [2.3.3](#)
- [20] S. BRÜCKNER, S. RUDOLPH: *Neural Control and System Identification Using a Similarity Approach*, Proceedings SPIE 7th International Symposium on Smart Structures and Materials, 2000, Newport Beach, California [2.3.4](#)
- [21] S. BRÜCKNER, S. RUDOLPH: *Dimensionally Homogeneous Neural Networks for System Identification*, Proceedings International Workshop on Similarity Methods, 1998, University of Stuttgart, Germany [2.3.4](#)
- [22] U.V. GARREL, H.J. OTTO, R. ONKEN: *Adaptive Modellierung des fertigungs- und regelbasierten Fahrzeugführungsverhaltens*, Universität der Bundeswehr München [2.3.5](#), [2](#)
- [23] H.M. YOUSSEF, J. JUANG: *Estimation of Aerodynamic Coefficients Using Neural Networks*, AIAA, 1993 [2.3.5](#)
- [24] S. WEISS, F. THIELECKE, H. HARDERS: *Ein neuer Ansatz zur Modellierung von Luftdatensystemen*, Proceedings Deutscher Luft- und Raumfahrtkongress, DGLR Jahrestagung 1999, Berlin [2.3.7](#), [5.6](#)

- [25] R.V. JATEGAONKAR, W. MÖNNICH: *Identification of Do-328 Aerodynamic Database for a Level D Flight Simulator*, AIAA Modeling and Simulation Technologies Conference, New Orleans, 1997 [2.3.7](#)
- [26] M.I. JORDAN: *Modular and hierarchical learning systems*, aus 'The Handbook of Brain Theory and Neural Networks', Cambridge, MIT Press, 1995 [2.3.8](#)
- [27] J. HAECKER, S. RUDOLPH: *On neural network topology design for nonlinear control*, Proceedings SPIE Aerosense 2001 Conference On Application and Science of Computational Intelligence IV, Orlando, Florida, 2001 [2.3.8](#)
- [28] K.A. RICHARDSON, R.F. STENGEL: *A Comparison of Neural Network Training Algorithms*, AIAA Atmospheric Flight Mechanics Conference, New Orleans, Aug. 1997 [2.3.9](#), [5.4](#)
- [29] D. A. KESKAR, V. KLEIN: *Determination of Instrumentation Errors from Measures Data Using Maximum Likelihood Method*, AIAA paper 80-1602, 1980 [3.2](#)
- [30] R.W. HAMMING: *Digitale Filter*, Wiley/VCH, ISBN 3527264639, 1987 [5.1.3.1](#)
- [31] O. BRIEGER: *Entwicklung eines Programms zur Berechnung und Enttrimmung des Nickmomentenbeiwertes anhand von Flugversuchsdaten*, Lehrstuhl für Flugmechanik und Flugregelung, Technische Universität München, 1999 [5.1.3.1](#)
- [32] G. SCHULZ, T. MEYER: *Trimmverfahren zum Abgleich eines Simulationssystems mit einem geregelten Referenzsystem*, Patentschrift DE 19850612 C1, Deutsches Patent- und Markenamt, 2000 [7.1.1](#)
- [33] J. SEIFERT, P. HUNKEL: *Modell eines idealen Piloten*, DGLR Jahrbuch 2001, DGLR-JT2001-140, Deutscher Luft- und Raumfahrtkongress 2001, Hamburg [7.3.2](#)
- [34] M. EBERLE: *Autopilotensystem*, Diplomarbeit, Uni Stuttgart [7.3.2](#)
- [35] K.W. ILIFF: *Aircraft Parameter Estimation*, AIAA paper 87-0623, 25th Aerospace Sciences Meeting, Reno, Nevada, 1987
- [36] S. WEISS, F. THIELECKE: *Aerodynamic Model Identification using Local Model Networks*, AIAA Atmospheric Flight Mechanics Conference, 2000, Denver, Colorado
- [37] C.M. BISHOP, G. HINTON: *Neural Networks for Pattern Recognition*, Clarendon Press, ISBN 0198538642, 1995
- [38] M. V. COOK: *Flight Dynamics Principles*, Wiley/NY, ISBN 0-470-23590-X, 1997

- [39] F. THIELECKE: *Parameteridentifizierung von Simulationsmodellen für das viskoplastische Verhalten von Metallen - Theorie, Numerik, Anwendung*, Dissertation, Mechanik Zentrum der Technischen Universität Braunschweig, ISBN 3-920395-33-6, 1998
- [40] W. J. GERBER, A. J. GONZALEZ: *Realtime Synchronisation and Modification of a behavioral Vehicle Model for Distributed Simulation*
- [41] S. R. CHU, R. SHOURESHI, M. TENORIO: *Neural Networks for System Identification*, IEEE Control Systems Magazine, Vol. 10, Number 3, April 1990
- [42] E.A. MORELLI, V. KLEIN: *Accuracy of Aerodynamic Model Parameters Estimated from Flight Test Data*, Journal of Guidance, Control, and Dynamics, Vol. 20, No. 1, January-February 1997
- [43] K.W. ILIFF, K.C. WANG: *X-29A Lateral-Directional Stability and Control Derivatives Extracted From High-Angle-of-Attack Flight Data*, NASA Technical Paper 3664, December 1996
- [44] S. RUDOLPH: *Entwurf, Anwendung und Interpretation Neuronaler Netze im Ingenieurwesen*, VDI- Fortschrittsberichte Reihe 20, Nummer 173, Seiten 124-130, VDI Verlag Düsseldorf, 1995
- [45] G. SCHNEIDER, D. KORTE, S. RUDOPH: *Neural Network Correspondencies of Engineering Principles*, Proceedings SPIE Conference On Applications and Science of Computational Intelligence III, Orlando, Florida, April 24-28th, 2000
- [46] S. RUDOLPH, B.-H. KRÖPLIN: *On Modelling Assumptions and Artificial Neural Networks*, Workshop Proceedings "Selbstorganisation von adaptivem Verhalten (SOAVE'97)", Universität von Ilmenau, 23.-24. September 1997, VDI Fortschrittsberichte, Reihe 8, Nr. 663, Seite 222-231
- [47] S. RUDOLPH: *On Topology and Generalisation in Feed-Forward Neural Networks*, Neuronale Netze in Ingenieur Anwendungen, Internationaler Workshop, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart, 15.-16. Februar 1996
- [48] S. HAYKIN: *Neural Networks, A Comprehensive Foundation*, Prentice Hall, 2nd edition, ISBN 0-13-273350-1, 1999
- [49] T.S. HUSSAIN: *Modularity within Neural Networks*, Dissertation 1995, Queen's University Kingston, Ontario, Canada
- [50] K.A. RICHARDSON, R.F. STENGEL: *Effects of localized Inputs on Neural Network Training*, AIAA Atmospheric Flight Mechanics Conference, Boston, Aug. 1998

-
- [51] H. SCHLICHTING, E. TRUCKENBRODT: *Aerodynamik des Flugzeugs*, Band 1, Springer Verlag, ISBN 3540673741, 3. Auflage 2001
- [52] H. SCHLICHTING, E. TRUCKENBRODT: *Aerodynamik des Flugzeugs*, Band 2, Springer Verlag, ISBN 354067375X, 3. Auflage 2000
- [53] X. HAFFER, G. SACHS: *Flugmechanik*, Springer Verlag, ISBN 3540569618, 3. Auflage 1993
- [54] D. BOEKMAN: *Structure Optimisation of Modular Neural Network*, Thesis, Leiden University, Netherlands, 1995
- [55] E. RONCO, P. GAWTHROP: *Modular Neural Networks: a state of the art*, Technical Report CSC-95026, University of Glasgow, May 1995
- [56] A. NÜRNBERGER, A. RADEZKY, R. KRUSE: *Using recurrent neuro-fuzzy techniques for the identification and simulation of dynamic systems*, Journal Neurocomputing, pp. 123-147, 2001

Abkürzungsverzeichnis

GA	Genetische Algorithmen
GMNN	Gated Modular Neural Network
LMN	Lokalmodell-Netz
MLP	Multi Layer Perceptron
MNN	Modulares Neuronales Netz
MSE	Mean Square Error
PID	Parameter Identification
QNH	Luftdruck in Meereshöhe
RBF	Radiale Basis Funktion
RNN	Rekurrentes Neuronales Netzwerk
SSE	Summed Square Error
WUT	Wind-Up-Turn

Index

A

A-Priori-Wissen, 10, 19, 23
 Abhängigkeit
 linear, 13, 37, 76
 Modellparameter, 8
 nichtlinear, 13, 33, 37, 76, 85
 Achsensystem, 42
 Aktivierungsfunktion, 18, 23, 34–36, 69, 70, 75
 Analyse
 flugmechanische, 3, 87
 Sensitivität, 10
 Anfangswertbestimmung, 17
 Anforderungen
 Echtzeitsimulation, 3, 103
 Parameteridentifikation, 13, 34, 104, 106
 Anstellwinkel, 30, 37, 51, 68, 69, 74
 Anwendung, 5, 9
 Approximation, 19, 38
 Athena, 15
 Aufgabenstellung, 9, 106
 Auftriebsbeiwert, 27, 43, 45, 77, 78, 90
 Ausgabe-
 funktion, 34
 neuron, 23, 35, 36, 56, 59, 60, 70, 104, 106
 Ausgangsfehlerverfahren, 4, 7, 14, 16, 17
 Ausgangsschicht, 35, 36, 58, 59
 Automatismus, 107

B

Backpropagation, 5, 10, 18, 23, 54, 55, 58, 59, 62, 106

Basismodell, 11, 55, 77, 79, 80, 83, 84, 95, 107
 Baukastenprinzip, 10, 37
 Bewegungsgleichung, 4, 15–17, 42, 87
 Bewegungsgröße, 3, 30, 63, 68, 89, 90, 93
 Black Box, 18, 62

D

Definitionsbereich, 34
 Derivativ, 3, 5, 6, 8, 13, 31, 34, 36–38, 40, 42, 59, 65, 86, 93, 94, 106, 108
 Deviationsmoment, 32, 88
 Dimensionshomogenes Neuronales
 Netz, 22
 Doublet, 30, 31
 Drallsatz, 42
 Drehbeschleunigung, 50, 51, 93
 Drehimpuls, 46
 Drehrate, 32, 50, 51, 63, 94, 95
 Dynamische Lernrate, 62, 71

E

Echtzeit-
 fähigkeit, 10, 13, 87, 93, 106
 simulation, 103, 104
 Eckfrequenz, 50
 Eigenstabilität, 89
 Einfluß
 der Gravitation, 91
 parameter, 24, 25, 37, 38, 73, 103
 Eingabe-

- neuron, 34–37, 41, 58, 62, 70, 75, 78, 103, 104
- raum, 53, 65
- vektor, 8, 35, 39, 62
- verbindung, 8, 10, 34, 35, 37, 39, 58–60, 69, 70, 75, 76, 103, 104, 106
- Eingangsschicht, 35, 58
- Einlaufimpuls, 43, 44, 48
- end-to-end-test, 3, 87, 93
- engineering judgement, 93
- Enttrimmprogramm, 16
- Envelope, 33
- Epoche, 65, 66, 71, 77, 79–82, 84, 96
- ESTIMA, 14
- Extended Kalman Filter, 27
- F**
- Faustformel, 36
- Fehler-
 - signal, 57, 58
 - toleranz, 5
- Filterkoeffizienten, 50
- Flightpath Reconstruction, 32
- Flugmanöver
 - 1123, 30
 - 3211, 30
 - spezial, 30, 31
 - Wind Up Turn, 73, 91, 94, 95
- Flugmeßdaten, 16, 52, 73, 77, 92, 94, 95, 99, 103, 106
- G**
- Gated Modular Neural Network, 8, 26
- Gaußfunktion, 23, 35
- Generalisieren, 4, 14, 22
- Gleichungsfehlerverfahren, 4, 15–17
- Gradientenabstiegsverfahren, 5, 7, 10, 18, 19, 62, 107
- Gruppe, 10, 34–41, 52, 54, 55, 62, 65, 69–72, 75–78, 80, 84, 86, 104
- I**
- Impulssatz, 42, 43, 73
- Informationsgehalt, 10, 68
- Initialisierung, 8, 18, 23, 54, 55, 70, 84

K

Komprimierung, 53, 105
Korrelation, 65, 68, 69
 Analyse, 69
 Matrix, 69

L

Lern-
 datensatz, 19, 33, 42, 52, 54, 55, 58,
 59, 64, 68, 76, 77, 83, 105
 kurve, 62, 70, 79, 80
 muster, 52, 53, 105
 rate, 55, 60, 70, 71
Lokalmodell-Netz, 8, 24

M

Minimum, 65, 72, 73, 84, 107
MLP, 18, 38, 54, 55, 80, 107
Modellierung, 3, 4, 8, 10, 23, 24, 37, 74,
 108
Modellkorrektur, 41
Modellstruktur, 9, 15, 18, 19, 25, 70, 74–
 76, 78, 94, 102, 103
Modul, 8
Modulares Neuronales Netzwerk, 26

N

Nachführen, 92
Netzwerk-
 fehler, 41, 55, 59, 65, 69–71, 84
 gewichte, 65, 83
 struktur, 5, 7, 37, 39–41, 52, 83, 103,
 106
Neuro-Fuzzy-System, 20, 21
Neuronenverbindung
 nichttrainierbare, 35, 58, 59
Nickmomenten-
 beiwert, 46, 50, 74, 75, 79, 80
 gleichung, 49

O

Objektorientierte Hochsprache, 104
Optimierungsergebnis, 106
overfitting, 19, 80

P

Pitch Doublet, 31
Polynomansatz, 24
Propagierungsfunktion, 34, 56

R

Rückkopplung, 7
Radialbasisfunktion, 23, 34, 55
RBF, 23
Referenzmodell, 8, 11, 13, 52, 54, 73, 75–
 78, 83, 99, 100, 103
Rekurrentes Neuronales Netz, 7, 20
Reproduzierbarkeit, 107
Robustheit, 4, 18

S

Schicht, 18, 19, 21, 23, 34–36, 55–57, 60,
 75, 104
Schrittweite, 56, 77
Schwellenwert, 5, 36, 55, 56
Sensitivität, 87
Sensor, 2, 4, 32, 50, 62, 108
SSE, 71
Stabilitäts- Plastizitäts- Dilemma, 105
static checks, 77

T

Tangens Hyperbolicus, 34
Testdatensatz, 52
Toleranz, 3, 93, 95, 97–99
Topologie, 5, 26
Trimm-
 verfahren, 88
 zustand, 53, 77–79, 93–95, 97, 99

V

Verbindungsgewicht, 20, 21, 35, 52, 54,
 56, 58, 70, 104
Virtual Reality, 2
Vorwärtsgerichtetes Neuronales Netz,
 18, 20, 21

W

Wertebereich, 5, 19, 36, 70, 71, 75, 104