

KfK 4892
August 1991

Ein mit neuronaler Software gesteuertes Fahrzeug, simuliert auf der KISMET- Robotersimulationsumgebung

H.-U. Hohn, H. Knüppel, U. Kühnappel, W. Müller
Projekt Heißdampfreaktor-Sicherheitsprogramm/
Handhabungstechnik

R. Opitz, Th. Weber
IBP Pietzsch GmbH, TAI, Karlsruhe

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE

**Projekt Heißdampfreaktor-Sicherheitsprogramm /
Handhabungstechnik**

KfK 4892

**Ein mit neuronaler Software gesteuertes Fahrzeug,
simuliert auf der KISMET-Robotersimulationsumgebung**

**H.-U. Hohn, H. Knüppel, U. Kühnapfel, W. Müller,
R. Opitz*, Th. Weber***

*** IBP Pietzsch GmbH, TAI, Karlsruhe**

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript vervielfältigt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

Zusammenfassung

Ein mit neuronaler Software gesteuertes Fahrzeug, simuliert auf der KISMET-Robotersimulationsumgebung

Es wird eine Simulationsumgebung zur Untersuchung von Anwendungsmöglichkeiten neuronaler Netzwerke auf dem Gebiet der Steuerung und Regelung eines sensorbestückten autonomen Fahrzeuges beschrieben.

Die analogen Sensorsignale wurden durch den Backpropagation Algorithmus zur Steuerung der Geschwindigkeit und der Richtung verarbeitet. Bei der Beschreibung der Kinematik und der Working-Cell kam ein Roboter off-line Programmiersystem (KISMET) zum Einsatz.

Summary

A vehicle driven by neuron software and simulated on a KISMET-robot-environment

A simulation environment is described. The possibilities are examined of using neuron networks to control an autonomous vehicle equipped with sensors.

Analogous sensor signals are used to control the speed and direction. This is done through the Backpropagation algorithm. In order to describe the kinematics and the working-cell a robot off-line programming system (KISMET) is used.

Vorwort

In den vergangenen drei Jahrzehnten haben sich Digitalrechner in bemerkenswerter Weise entwickelt. Heute vollbringen schon Taschenrechner komplexe Aufgaben, für die vor einigen Jahren noch riesige Rechenmaschinen benötigt wurden. Trotzdem hat sich der grundlegende Aufbau digitaler Computer bis zur heutigen Generation von Supercomputern nicht geändert. Nach dem *Von-Neumann-Prinzip* sind Speicher und Prozessor getrennte Einheiten. Dies bedeutet ununterbrochenes Hin- und Herschieben von Daten aus dem Speicher in den Prozessor und wieder zurück.

Da sich der prinzipielle Aufbau von Rechnern nicht wesentlich geändert hat, beruht die anhaltende stürmische Entwicklung der Computertechnik auf der Tatsache, daß immer mehr Daten in immer kleineren Bausteinen gespeichert und gleichzeitig schneller vom Prozessor verarbeitet werden können. Schon zu Beginn der achtziger Jahre waren jedoch die technischen Grenzen solcher Strukturen abzusehen.

So begann man mit der Entwicklung von Architekturen, bei denen in einem Rechner mehrere Prozessoren gleichzeitig unterschiedliche Befehle ausführen, sogenannte Parallelarchitekturen. In der Bundesrepublik wurde das Forschungsprojekt *Suprenum* begonnen, aus den USA kommen die bekanntesten Vertreter von Parallelrechnern, nämlich der *Hypercube* und die *Connection Machine* mit über 65000 Prozessoren. Das Problem, wie es sich heute darstellt, ist allerdings kein Hardware-Problem mehr, sondern vielmehr besteht der eigentliche Engpaß in der Programmierung solcher Parallelrechner. Es ist zu komplex, ein Problem so zu programmieren, daß tausende von Prozessoren vernünftig ausgelastet sind. Deutlich wird dabei die Notwendigkeit, Programmsysteme zu entwickeln, die sich selbst organisieren, daß heißt, die Arbeit gleichmäßig auf alle Prozessoren verteilen und zudem wegen der Unüberschaubarkeit Fehlertoleranz aufweisen. An diesem Punkt kommt man zwangsläufig auf ein biologisches Vorbild: Das Gehirn! Das Gehirn arbeitet durch die komplexe Zusammenschaltung von über 10 Milliarden Nervenzellen (Neuronen) hochgradig parallel und fehlertolerant. Aus diesem Ansatz heraus und mit dem Ziel, parallele Vorgänge abbilden zu können, wurden sogenannte neuronale Netzwerkmodelle entwickelt.

Neuronale Netzwerkmodelle tauchen in der vorwiegend englischsprachigen Fachliteratur unter verschiedenen Oberbegriffen auf, z.B. *Konnectionismus* oder *PDP-Modelle* (Parallel Distributed Processing). Neuronale Netze werden aus der Sicht der Informatik oft als ein Teilgebiet der KI betrachtet, aber auch als Alternative zur Künstlichen Intelligenz, mit dem Ziel, die durch die *Von-Neumann-Ar-*

chitektur heutiger Computer auferlegten Grenzen zu sprengen. Mit Hilfe neuronaler Netzwerkmodelle versuchen Forscher derzeit weltweit Funktionen bzw. Eigenschaften eines Gehirns zu verstehen, zu simulieren und letztlich nachzubilden. Neuronale Netze unterscheiden sich daher von herkömmlichen Computerarchitekturen im wesentlichen durch zwei Prinzipien. Zum einen sind oder sollen neuronale Netzwerke adaptiv und trainierbar sein. Dies bedeutet, sie werden nicht programmiert, sondern mit sogenannten Trainingssets gefüttert und verbessern ihr Verhalten mit zunehmender Anzahl von Trainingsdaten (Lernen durch Beispiele). Durch die Adaptivität wird eine Anpassung an sich ändernde Umgebungsbedingungen gewährleistet, das heißt, das Netz kann mit Hilfe neuer Trainingssets dazulernen. Der andere Punkt ist eine massive Parallelität der Verarbeitung und Speicherung, woraus hohe Geschwindigkeiten und Fehlertoleranz resultieren. Die Fehlertoleranz bezieht sich nicht nur auf fehlerhafte Trainingsdaten, sondern auch auf die Architektur des Modells, was sich in einem Ausfall von Neuronen, Verbindungen und dergleichen darstellt.

Durch diese beiden Paradigmen, Adaptivität und Parallelität, versucht man nun einige Fähigkeiten des Gehirns zu ergründen und diese Erkenntnisse in moderne Systeme einzusetzen. Diese besonderen Fähigkeiten des Gehirns liegen nun nicht im schnellen Rechnen oder Hin- und Herschieben von Daten, sondern in viel komplexeren Aufgaben wie Erkennen von Bildern und Assoziieren von Mustern (Erinnerungsvermögen). Wenn man bedenkt, daß das Gehirn aus über 10 Milliarden engvermaschter Neuronen besteht und diese Signale nur etwa alle 10 msec aussenden (dies ist etwa 10^5 mal langsamer als der Zeittakt in schnellen Computern), erkennt man die Bedeutung, die einer hochgradigen Parallelität zukommt. Im Durchschnitt ist ein Neuron des Gehirns mit etwa 1000 bis 10000 anderen Neuronen verbunden.

Je nach Prinzip der Durchschaltung von Signalen und Vermaschung der Neuronen wurden nun Modelle entwickelt, die bestimmte Fähigkeiten des Gehirns nachvollziehen können. Einige Modelle sind adaptiv, daß heißt sie können jederzeit dazulernen, andere sind assoziative Speicher. Manche Modelle arbeiten mit binären (0,1) oder bipolaren (-1,1) Signalen, andere wiederum mit analogen Signalen. Alle haben sie jedoch eine Gemeinsamkeit: Jedes Neuron arbeitet unabhängig von dem anderen und produziert ein Signal, das zu einem oder mehreren anderen Neuronen weitergegeben wird. Eine Informationsverarbeitung oder Speicherung von Daten erfolgt also nicht durch einzelne, getrennte Bausteine, sondern durch das interaktive Zusammenspiel vieler Neuronen. Die Verbindung zwischen den Neuronen, die sogenannten Gewichte, bestimmen dabei das Ver-

halten des Systems. Statt Neuronen wird im Zusammenhang mit neuronalen Netzwerken auch häufig der Begriff Prozeßelement verwendet.

¹ "A neural network is a highly parallel dynamical system with the topology of a directed graph that can carry out information processing by means of its state response to either initial or continuous input." (Definition Neuronaler Netzwerkmodelle nach R.Hecht-Nielsen).

Inhaltsverzeichnis

1. Einleitung	1
2. Konfiguration	2
2.1 Hardware	2
2.2 Software	2
2.3 Konzept der Simulation	3
3. Motivation und Ausgangsbasis	5
3.1.1 Die Sensorik des Fahrzeuges (NeVIS III)	7
3.1.2 Die Aktuatorik des Fahrzeuges (NeVIS III)	10
3.1.3 Vorarbeiten und Zielsetzungen von NeVIS IV - KISMET	12
3.1.4 Kopplung Silicon Graphics - MicroVAX - PC	14
3.2 Erweiterungen in KISMET	15
3.3 Interface-Programme zur Kommunikation zwischen KISMET und der neuronalen Netz-Software	16
4. Teachphase	17
4.1 Fahrgründe	19
4.2 Training des neuronalen Netzwerkes	27
4.3 Statistik über das Lernverhalten des verwendeten Netzes	28
5. Kannphase	36
Anhang A. Beispieldatensätze	38
A.1 Beispiel eines IRDATA Programmes	38
A.2 Beschreibung der Lerndatei	40
A.3 Beispiel einer Lerndatei.	41
A.4 Beispiel einer Konvergenzdatei	45
A.5 Beispiel einer Fehlerdatei	49
A.6 Beispiel einer Netzwerkdatei	53
A.7 Literaturverzeichnis	60

Abbildungsverzeichnis

Abbildung 1.	Systemkonfiguration	4
Abbildung 2.	Erweitertes Konzept NeVIS III	6
Abbildung 3.	Anordnung und Nummerierung der Sensoren bei NeVIS III	7
Abbildung 4.	Lineare Keuleneinteilung bei NeVIS III	8
Abbildung 5.	Einzelpunktaufnahme des NeVIS III	9
Abbildung 6.	Fahrmöglichkeiten des Fahrzeuges NeVIS III	10
Abbildung 7.	Kopplung Silicon Graphics - MicroVAX - PC	15
Abbildung 8.	Fahrzeug mit Sensorik	18
Abbildung 9.	Ecke (rechts abbiegen)	20
Abbildung 10.	Ecke (links abbiegen)	21
Abbildung 11.	Wand (rechts abbiegen)	22
Abbildung 12.	Wand (links abbiegen)	23
Abbildung 13.	Gasse	24
Abbildung 14.	alleinstehender Gegenstand	25
Abbildung 15.	alleinstehender Gegenstand (nachlernen)	26
Abbildung 16.	Training des Neuronalen Netzes	27
Abbildung 17.	Verlauf des globalen Fehlers	29
Abbildung 18.	Verlauf des Beschleunigungsfehlers	30
Abbildung 19.	Verlauf des Richtungsfehlers	31
Abbildung 20.	Vergleich verschiedener Netze N1 bis N7	32
Abbildung 21.	Vergleich Netze N2, N4 und N6	33
Abbildung 22.	Vergleich Netze N2, N4 und N6	34
Abbildung 23.	Vergleich Netze N2, N4 und N6	35
Abbildung 24.	Kannphase	36

Tabellenverzeichnis

Tabelle	1. 3-Bit-Kodierungen bei NeVIS III	8
Tabelle	2. 4-Bit-Kodierung und interne Darstellung der 8 Fahrmöglichkeiten bei NeVIS III	11
Tabelle	3. ASCII-Format der Lerndatei bei NeVIS III	12
Tabelle	4. Vergleich der verschiedenen Netzkonfigurationen	28

1. Einleitung

In der Zeit von Oktober 1990 bis einschließlich Februar 1991 wurde eine Simulationsumgebung zur Untersuchung von Anwendungsmöglichkeiten neuronaler Netzwerke auf dem Gebiet der Steuerung und Regelung eines autonomen Fahrzeuges aufgebaut. Hierzu wurden mit Hilfe des im KfK entwickelten Simulationspaketes KISMET (Kinematic Simulation, Monitoring and off-line programming Environment for Tele robotics) Lerndaten erzeugt, die als Eingabe für verschiedene neuronale Netzwerkarchitekturen dienen. Ein Lerndatensatz besteht aus den Werten der auf einem Fahrzeug montierten Sensoren (Sensorik) und der vorgesehenen Fahrtrichtung und Beschleunigung (Aktuatorik).

Die erzielten Ergebnisse belegen, daß schon relativ kleine neuronale Netzwerkarchitekturen, die mit dem Backpropagation-Lernalgorithmus trainiert wurden, in der Lage sind, unerwartete Situationen mit den trainierten Lerndaten zu assoziieren.

Bei den Beschleunigungswerten des Fahrzeuges gibt es noch Probleme, da in KISMET die Geschwindigkeit in den Stützpunkten eines Fahrweges auf Null abgebremst wird (point to point (PTP) Verhalten eines Roboters). Dies bedeutet, daß das Fahrzeug vor Erreichen eines Stützpunktes abgebremst und danach wieder beschleunigt wird. Es erfolgt also kein kontinuierliches Fahren in Abhängigkeit zu den Sensorwerten. Daher sind die Beschleunigungswerte nur sehr schlecht mit den Sensorwerten zu korrelieren. Berücksichtigt man dies bei der Beurteilung der erzielten Ergebnisse, kann der vorläufige Projektabschluß als erfolgreich gewertet werden.

Folgende Einrichtungen und Personen beteiligten sich an dem Projekt:

IBP Pietzsch GmbH, TAI:

Rigobert Opitz
Thomas Weber
Achim Gayer

KfK, Institut für Reaktorentwicklung:

Helmut Knüppel
Uwe Kühnapfel

KfK, Labor für Handhabungstechnik:

Willi Müller und
Hans-Ulrich Hohn.

2. Konfiguration

2.1 Hardware

Folgende Maschinen wurden verwendet:

- eine Silicon Graphics Personal Iris zur Darstellung der Simulation (KISMET) und als Plattform für die neuronale Steuerungssoftware,
- eine MicroVax II als Plattform für die neuronale Steuerungssoftware,
- und diverse PC's als Teststationen.

2.2 Software

An Software wurden existierende Programmpakete modifiziert und weiterentwickelt:

- auf der Basis der KISMET-Simulationssoftware wurde der Aufbau eines Fahrzeugmodells, der zugehörigen Sensorik und der Umgebung (Fahrgrund) realisiert. Interfacesoftware zur online Kopplung der KISMET-Simulationsumgebung an die neuronale Steuerungseinheit wurde entwickelt.
- Ein bereits auf PC-Basis existierender Backpropagation Simulator zum Training von kleinen bis mittelgroßen neuronalen Netzwerken (maximal 50 Knoten pro Ebene) wurde auf VAX-Architekturen und Silicon Graphics Workstations portiert. Die Simulationssoftware wurde zur Überwachung der Lernphase durch diverse Kontrollstatistiken erweitert.

2.3 Konzept der Simulation

Der Simulationsaufbau bestand aus folgender Hardware:

- Eine Silicon Graphics Personal Iris mit Unix Betriebssystem zur Darstellung der Applikation als ein Modell integriert in der KISMET-Robotersimulationsumgebung.
- Eine MicroVax II mit VMS 5.3 als Plattform für die neuronale Steuerungssoftware.
- Diverse PC's mit VGA-Monitoren als Test- und Trainingsstationen.

Die Vernetzung der einzelnen Maschinen und die dabei verwendeten Protokolle sind aus Abbildung 1 auf Seite 4 -Systemkonfiguration- ersichtlich.

Das Modell eines Fahrzeuges, das die Umgebung mit simulierten Ultraschallsensoren abtastet und aktuatorische Bewegungen ausführt, ist im folgenden Kapitel schematisch dargestellt. Das Fahrzeug wird nur durch zwei Bewegungsdaten gesteuert. Zum einen eine Richtungsänderungskomponente, die angibt, in welche Richtung und mit welcher Intensität abgelenkt werden soll. Werte größer 0.5 entsprechen dabei Rechtsdrehungen, Werte kleiner 0.5 Linksdrehungen. Zum anderen eine Beschleunigungskomponente, die die auszuführende Geschwindigkeitsänderung angibt. Hier bedeuten Werte größer 0.5 eine Erhöhung, Werte kleiner 0.5 ein Vermindern der Geschwindigkeit. Diese zwei Bewegungsdaten werden durch die zwei Outputknoten des Netzwerkes repräsentiert. Die 13 Inputknoten des Netzwerkes setzen sich aus 12 Knoten für die aktuelle Sensorinformation und einem Knoten für die aktuelle Geschwindigkeit des Fahrzeuges zusammen.

Systemkonfiguration

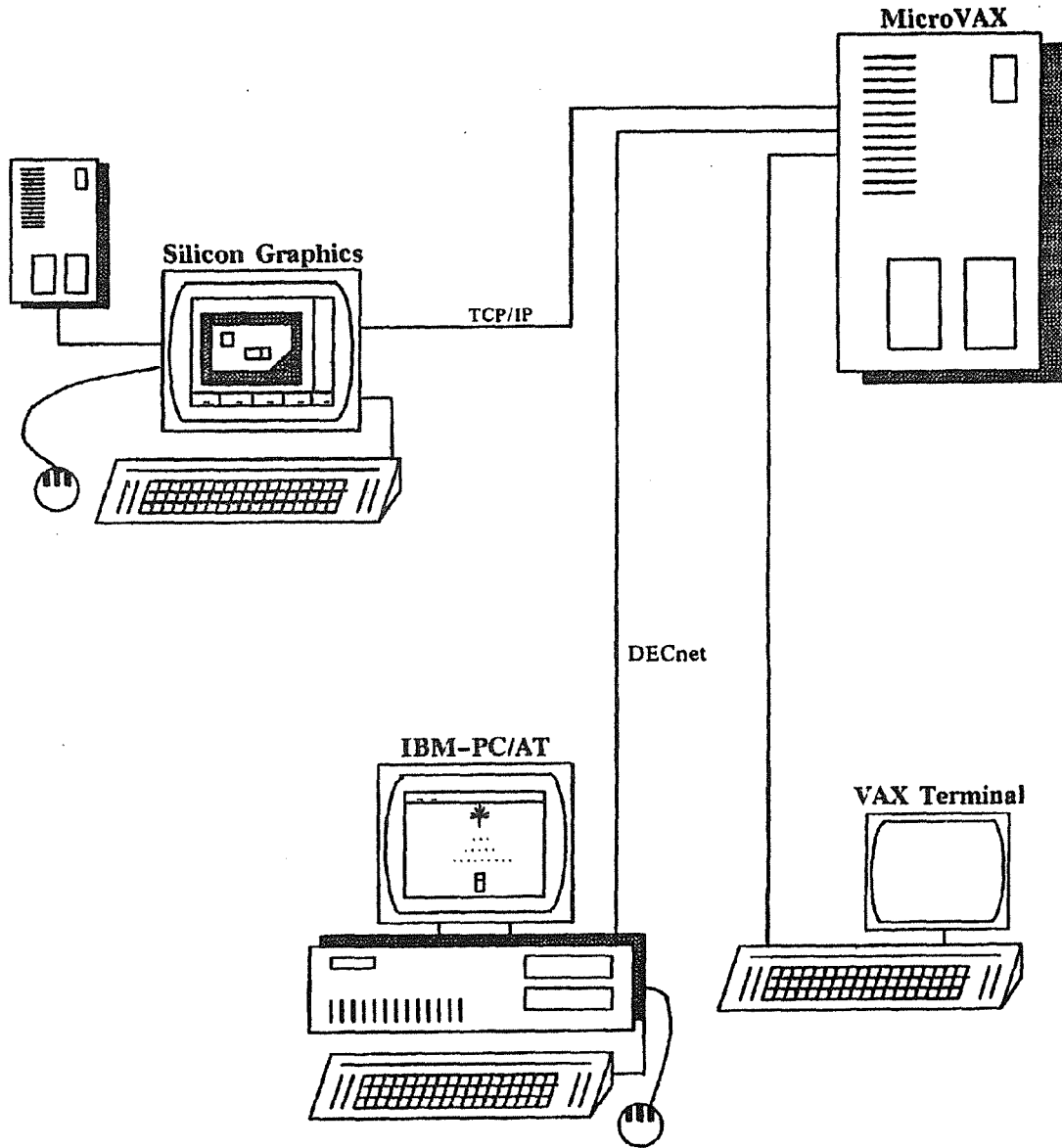


Abbildung 1. Systemkonfiguration

3. Motivation und Ausgangsbasis

Die Firma IBP Pietzsch GmbH arbeitet schon seit einigen Jahren im Rahmen von europäischen Forschungsprojekten an der Entwicklung von Softwarepaketen zur Simulation neuronal gesteuerter, autonomer Fahrzeuge in einer unbekanntem Umgebung. Hierbei ist im wesentlichen das ESPRIT Projekt ANNI (Application of Neural Networks for Industrie in Europe) zu erwähnen. In der Simulation haben die Fahrzeuge (AGV) etwa die Größe eines Kleinwagens und sind mit mehreren entfernungsdetektierenden Ultraschallsensoren ausgerüstet.

In der Abteilung TAI (Technikum für angewandte Informatik) wird die Steuerung eines autonomen Fahrzeuges mit neuronalen Backpropagation-Netzen und adaptiven Reinforcement Learning Algorithmen untersucht. In jüngster Zeit laufen nun auch Experimente mit den verschiedenen Modellen assoziativer Speicher-algorithmen.

Allen Simulationen ist gemeinsam, daß die simulierten Signale der Ultraschallsensoren so aufbereitet werden, daß man ein neuronales Netz darauf trainieren kann, auf bestimmte Merkmale der Signale in vorgegebener Weise zu reagieren. Das Fahrzeug soll dabei zunächst in der Lage sein, Hindernisse in einem Fahrgrund zu erkennen und ihnen schließlich auszuweichen. Eine weitere Zielsetzung ist, daß sich das Fahrzeug von einem vorgegebenen Ausgangspunkt zu einem ebenfalls vorgegebenen Zielpunkt bewegt (Routenplanung).

In der ersten Simulationsphase des Lernfahrzeuges wurden zunächst die Einsatzmöglichkeiten des NeuralWare-Simulators getestet, der mehrere Netzwerkmodelle zur Verfügung stellt. Man entschied sich für das Backpropagation-Modell, dessen Brauchbarkeit sich in weiteren Untersuchungen bestätigte.

Die Erzeugung einer Lerndatei erfolgte nach dem Prinzip der manuellen Simulation, d.h. eine Lerndatei entstand aus der manuellen Datenaufnahme von 10 unterschiedlichen Hindernissen. Nach der Dimensionierung des Netzes, mehreren Parametereinstellungen und zugehörigen Lernphasen, hatte das Netz alle Hindernisse korrekt gelernt. Selbst bei einem simulierten Rauschanteil in der Datenaufnahme der Sensorik von 10% arbeitete das Modell nahezu korrekt. Die Simulation wurde NeVIS I (Neural Vehicle Information System) getauft.

Da die manuelle Simulation recht mühsam und zeitaufwendig war, wurde in der zweiten Entwicklungsphase ein Graphikeditor entwickelt, mit dem eine automatische Datengenerierung möglich ist (NeVIS II und NeVIS III).

Abbildung 2 auf Seite 6 zeigt das erweiterte Konzept des Lernfahrzeuges (NeVIS III), das auf den Ergebnissen von NeVIS I aufbaut. Die Sensorik und Aktuatorik des

Fahrzeuges sowie das neuronale Netzwerk werden im folgenden näher beschrieben.

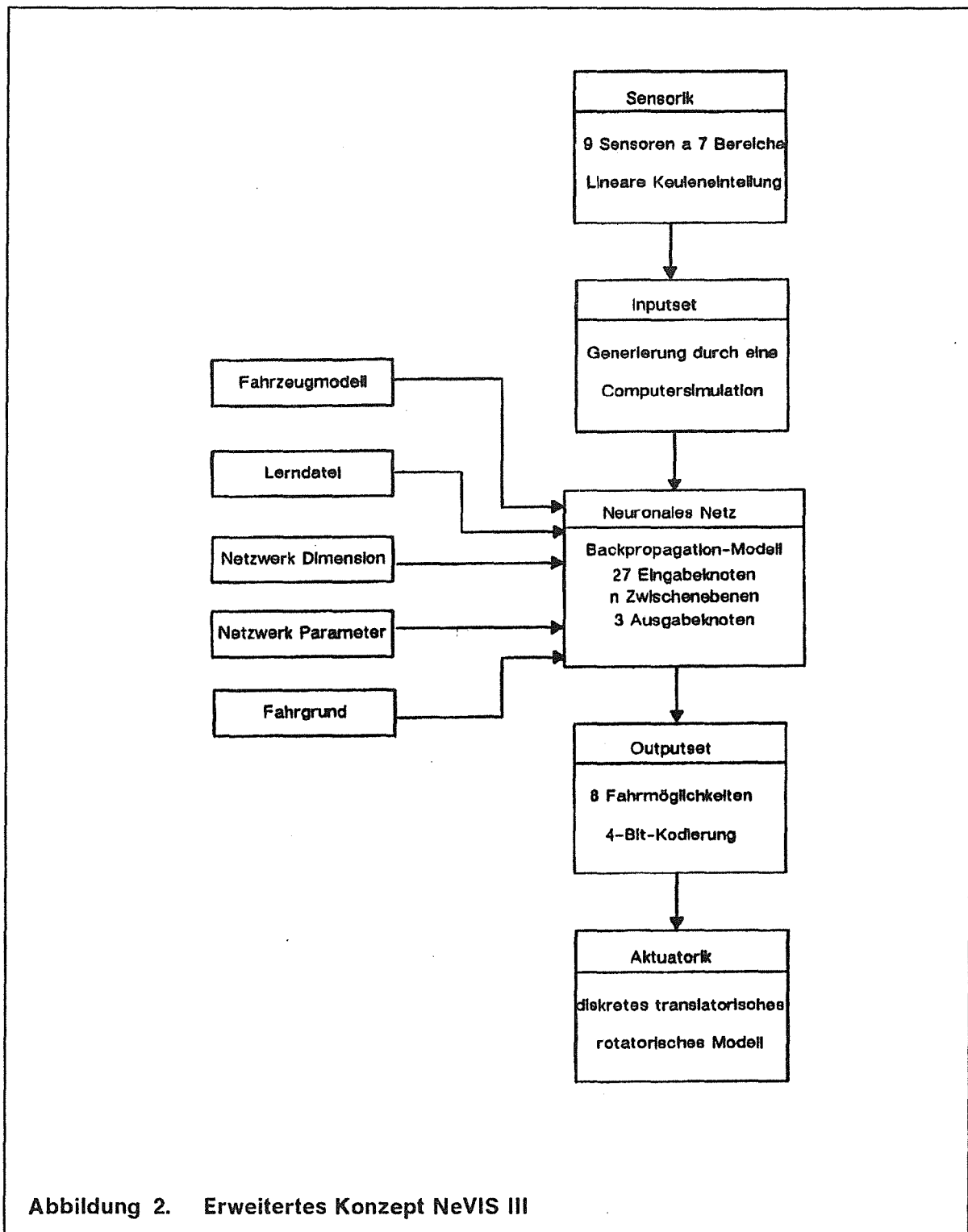


Abbildung 2. Erweitertes Konzept NeVIS III

3.1.1 Die Sensorik des Fahrzeuges (NeVIS III)

Zur Erfassung und Verarbeitung von Umweltinformationen (Objekte, Hindernisse) benötigt das Fahrzeug ein externes Sensorsystem, welches die Signale seiner physikalischen Umgebung aufnimmt, umformt und verarbeitet.

Zu diesem Zweck wurde im TAI das Sensorsystem ULTRAS entwickelt. Das System besteht aus 9 Ultraschallsensoren, die mit einem IBM-kompatiblen PC-AT gekoppelt sind, und einer Ansteuerungslogik, die die Meßwerte erfaßt und in eine digitale Darstellung transformiert. Die Sensorwerte können on-line und graphisch auf dem Bildschirm dargestellt werden. Die verwendeten Sensoren decken einen Bereich von 20 Grad ab und ermöglichen eine Abstandsmessung bis ca. 12 Meter. Für die Simulation des Fahrzeuges wurde in Anlehnung an ULTRAS die Abstandsmeßmethode durch Ultraschallsensoren übernommen. Die Anzahl und die Anordnung der Ultraschallsensoren spielte bei der Steuerung durch ein neuronales Netz eine wesentliche Rolle. Die Anzahl der Sensoren kann einerseits aufgrund der zu verarbeitenden Datenmenge und der daraus resultierenden Erhöhung der Verarbeitungszeit durch das Netz nicht zu groß gewählt werden, andererseits müssen ausreichend Sensoren vorhanden sein, um eine möglichst genaue Erfassung der Fahrumgebung zu erreichen. Bei der Simulation des Lernfahrzeuges wurden 9 Sensoren mit folgender Anordnung verwendet:

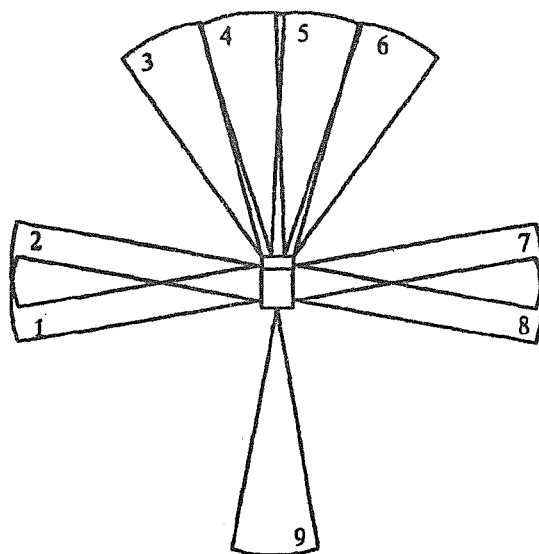
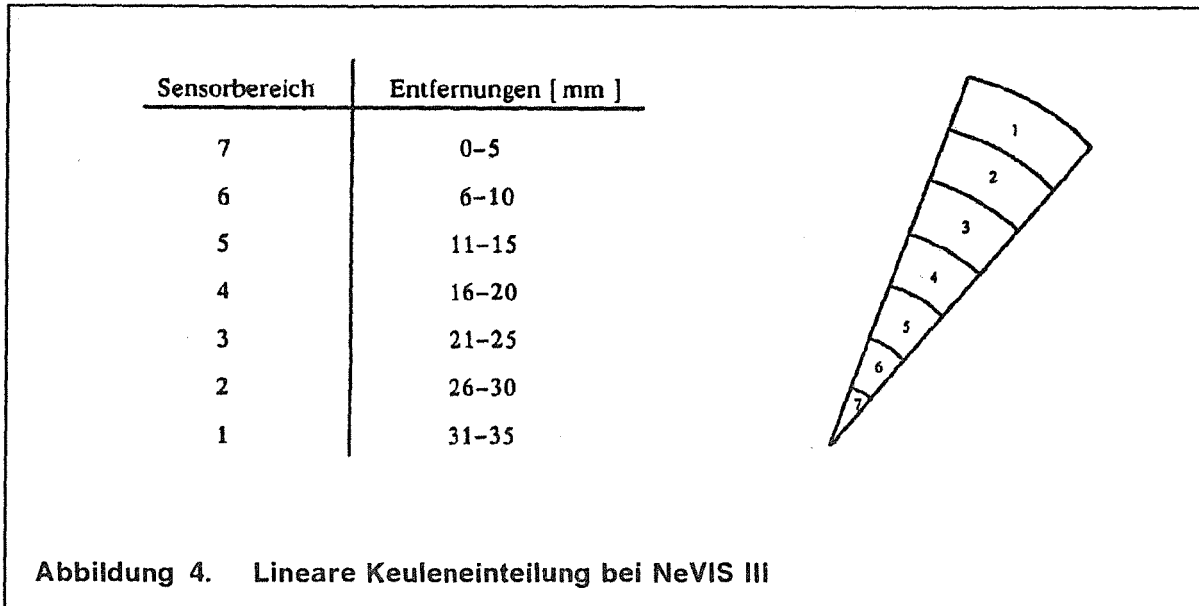


Abbildung 3. Anordnung und Nummerierung der Sensoren bei NeVIS III

Da das Fahrzeug rückwärts nur geradeaus fahren sollte, wurde an der Fahrzeugrückseite nur ein Sensor angebracht. Für die Vorwärts- und Drehbewegungen besitzt das Fahrzeug an der Vorderseite vier und seitlich jeweils zwei Sensoren. Eine Sensorkeule wurde in sieben gleichlange Bereiche unterteilt. Abbildung 4 zeigt die lineare Keuleneinteilung und die entsprechenden Entfernungen.



Zur Erzeugung eines binären Inputmusters für das neuronale Netz wurden die sieben Bereiche in eine 3-Bit-Kodierung umgesetzt:

Bereich	Code
7	111
6	110
5	101
4	100
3	011
2	010
1	001

Tabelle 1. 3-Bit-Kodierungen bei NeVIS III

Bei dieser Kodierung wird die Bitkombination 000 verwendet, wenn sich in keinem der sieben Bereiche ein Hindernis befindet.

Bei der computerunterstützten Simulation entsteht ein Inputmuster durch die Einzelpunktaufnahme in einem Fahrgrund. Das Fahrzeug wird dabei an einem beliebigen Punkt im Fahrgrund positioniert. Abbildung 5 auf Seite 9 zeigt das Fahrzeug mit den Sensorkeulen und den detektierten Hindernissen.

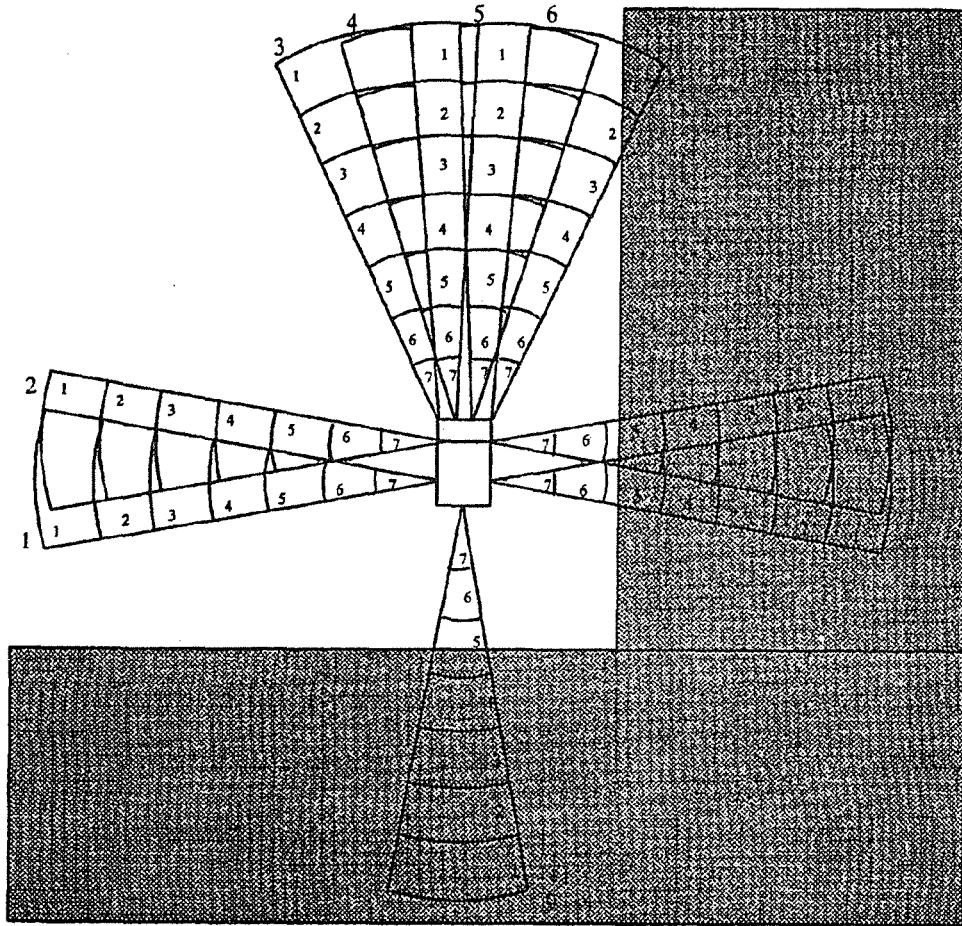


Abbildung 5. Einzelpunktaufnahme des NeVIS III

Aus der Abbildung ergibt sich für diese Fahrgrundposition folgendes Inputmuster:

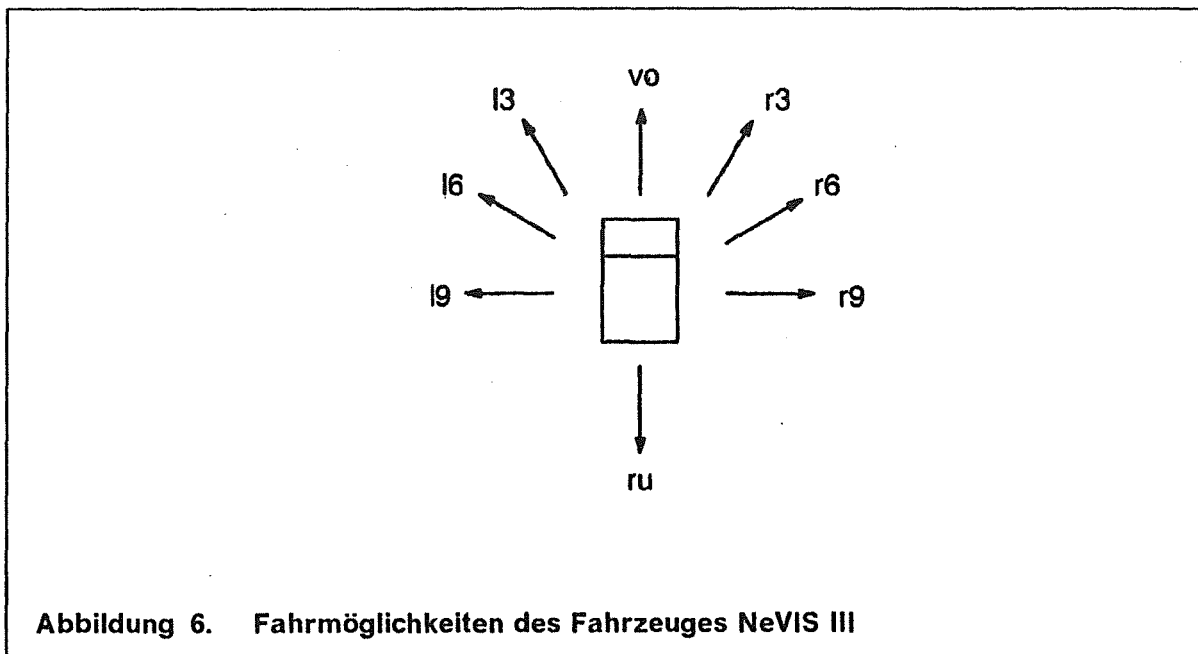
Sensor: 1 2 3 4 5 6 7 8 9

Keulenbereich: 0 0 0 0 0 2 5 5 5

Das Muster besagt also, daß sich in den Sensorbereichen der ersten 5 Sensoren kein Hindernis, im Sensor 6 im Bereich 2 und in Sensor 7, 8 und 9 im Bereich 5 ein Hindernis befindet. Auf diese Weise lassen sich viele Inputmuster eines Fahrgrundes erzeugen. Die Inputmuster werden in einem Inputset, einer sogenannten Lerndatei gespeichert, mit der das neuronale Netz dann trainiert werden kann. Eine solche Datenaufnahme ermöglicht die Generierung mehrerer Lerndateien zu unterschiedlichen Fahrgründen.

3.1.2 Die Aktuatorik des Fahrzeuges (NeVIS III)

Das Fahrzeug kann sowohl translatorische als auch rotatorische Bewegungen ausführen. Zur Ausführung von Rotationen erfolgt eine Richtungsänderung nach rechts oder links. Um diskrete Werte zu erhalten, wurden nur bestimmte Einschlagwinkel zugelassen. Das verwendete Backpropagation-Netz kann zwar mit analogen Werten arbeiten, jedoch sollte vorerst ein mit binären Ein/Ausgabewerten arbeitendes Netzwerk ausreichen. Die Simulation beschränkt sich auf 8 Fahrmöglichkeiten, woraus sich für die Aktuatorik ebenfalls eine 3-Bit-Kodierung ergibt. Die Translation besteht aus Vorwärts- und Rückwärtsbewegungen, wobei die Geschwindigkeitskomponente beim NeVIS III vorläufig nicht integriert wurde. Die Rotatorik hat sich allerdings erweitert auf die Winkel 30 Grad, 60 Grad und 90 Grad nach links bzw. nach rechts. Abbildung 6 zeigt die 8 Fahrmöglichkeiten des Fahrzeuges.



Die zuvor abgebildeten Fahrmöglichkeiten können in eine 3-Bit-Kodierung umgesetzt werden, für eine spätere Erweiterung wurde aber eine 4-Bit-Kodierung ausgewählt. Die nachfolgende Tabelle zeigt die 4-Bit-Codes der 8 Fahrmöglichkeiten und den internen Code, mit dem die Fahrbewegungen in der Lerndatei gespeichert werden.

Bit: 3 2 1 0	Fahrbewegung	Interner Code
0 1 1 1	rechts 90_(r 9)	0
0 1 1 1	rechts 60_(r 6)	1
0 1 1 1	rechts 30_(r 3)	2
0 1 1 1	vorwärts (vo)	3
0 1 1 1	links 30_(l 3)	4
0 1 1 1	links 60_(l 6)	5
0 1 1 1	links 90_(l 9)	6
0 1 1 1	rückwärts (ru)	7

Tabelle 2. 4-Bit-Kodierung und interne Darstellung der 8 Fahrmöglichkeiten bei NeVIS III

Für die NeVIS III Simulation wurde erneut das Backpropagation-Modell ausgewählt, da es sich schon in der ersten Phase für diese Aufgabe bewährt hatte. Der Aufbau des Backpropagation-Modelles für den Fahrzeugsimulator vollzieht sich Schritt für Schritt. Zuerst baut man durch Anklicken der entsprechenden Menüoption die Ebenen und gewünschten Vermaschungen zusammen und trägt zum Schluß die Lernparameter ein. Das Backpropagation-Modell wurde auf 27 Eingabeknoten (9 Sensorbereiche mit je 3 Bit), 4 Ausgabeknoten (eine Fahrmöglichkeit mit 4 Bit) und zumeist 8 Knoten in der Zwischenschicht dimensioniert. Während die Anzahl der Ein- und Ausgabeknoten vom Problem her bestimmt werden, ist die Anzahl der Zwischenknoten aufgrund heuristischer Verfahren zu bestimmen.

Hat man ein komplettes Modell aufgebaut, so werden zum Lernen der Gewichte die Inputvektoren sequenziell aus einer Trainingsdatei (Lerndatei) eingelesen. Die Trainingsdatei hat einen speziellen Aufbau:

Aus den in den vorangegangenen Kapiteln beschriebenen Kodierungen der Sensorik und Aktuatorik ergibt sich für die Lerndatei das folgende ASCII-Format:

...	...
input	4 4 0 0 0 0 0 0
desired output	3
input	4 4 0 0 0 0 0 0
desired output	3
input	4 4 1 1 1 1 0 0
desired output	3
input	4 4 2 2 2 2 0 0
desired output	3
input	4 4 3 3 3 3 0 0
desired output	3
input	4 4 4 4 4 4 0 0
desired output	3
input	3 3 0 0 0 0 0 5
desired output	3
input	3 3 0 0 0 0 0 4
desired output	3
...	...

Tabelle 3. ASCII-Format der Lerndatei bei NeVIS III

Vor Beginn der Lernphase initialisiert man das Netz mit Zufallszahlen aus einem vom Bediener eingegebenen, symmetrischen Wertebereich. Während der Lern- und Recallphase werden Gewichte und Aktivierung der Knoten entsprechend ihres Wertes in farbiger Codierung angezeigt. Die Farben können aus einer Farbpalette ausgewählt und der Problemstellung angepaßt werden. Zusätzlich werden diverse Funktionen zur graphischen Aufbereitung des Modelles in verschiedenen Größen und Formen angeboten.

3.1.3 Vorarbeiten und Zielsetzungen von NeVIS IV - KISMET

Um das vorhandene Konzept aus softwaretechnischer Sicht weiter zu verfeinern und zusätzlich neue Erkenntnisse zu gewinnen, wurde die Fahrzeugapplikation durch das Robotersimulationsprogramm KISMET modelliert und simuliert.

Dies brachte enorme Vorteile in Bezug auf analoge Sensoriksignale und Aktuatorikwerte. Die auf den vorherigen Seiten beschriebenen Diskretisierungen wurden durch eine kontinuierliche Repräsentation der Sensorsignale sowie durch eine kontinuierliche Bewegungsinformation ersetzt. Hinzu kam eine immense Ge-

schwindigkeitssteigerung durch Verwendung von Workstations und einer Micro-VAX.

Durch die Analogisierung der Ein- und Ausgabewerte für das neuronale Netzwerk sollten Erkenntnisse über die Lerngeschwindigkeit und den Konvergenzverlauf im Vergleich zu den diskreten Untersuchungen gewonnen werden. Weiterhin wurden viele Erweiterungen und Verbesserungen bezüglich Netzwerkspeicherung, Lern-datei und Konvergenzdatei vollzogen (siehe Anhang).

Es folgt eine Auflistung von Erweiterungen und Verbesserungen gegenüber dem ersten Konzept der NeVIS IV - KISMET Kopplung. Diese Verbesserungen wurden zur Optimierung des Fahrzeugverhaltens und der Lernüberwachung durchgeführt.

1.) Teachfiles:

- Erstellung von erweiterten Teachfiles in drei unterschiedlichen Fahrgründen;
- spezielle Situationen wurden an vordefinierten Einzelhindernissen "geteached";
- fehlerhaftes Fahrverhalten wurde in bestimmten Situationen beobachtet und speziell nachtrainiert (senkrechte Anfahrwinkel etc.);
- die Trainingsvektoren der Teachfiles wurden gemischt.

2.) Fahrzeug:

- die Sensorik wurde geändert, um fehlerhaftem Fahrverhalten entgegenzuwirken. Zwei rückwärtige Sensoren wurden nach vorne verlegt. Die Winkel der vorderen Sensoren wurden geändert (10/5/0/0/5/10). Die seitlichen Sensoren wurden in die Ecken gesetzt und um 1/3 gekürzt. In der nun abgewandelten Konstellation wurde ein stark verbessertes Verhalten bemerkt.
- Im nächsten Schritt wurden die vorderen Sensoren noch weiter nach außen gedreht (15/5/0/0/5/15) und zusätzlich die jeweils Äußeren um 1/3 gekürzt. Das Fahrverhalten verbesserte sich weiter.

3.) Taktung:

- Relativ bald zeigte sich, daß die Werte der Aktuatorik in den Lerndatensets eine zu schwache Ausprägung zeigten. Alle Werte befanden sich in einem kleinen Bereich um 0.5. Um dies zu ändern wurde die Taktung bei Erzeugung der Teachfiles von anfänglich 500ms auf 1000ms erhöht. Es folgten Versuchsreihen mit unterschiedlichen Taktungen. Die besten Ergebnisse wurden schließlich bei einem Wert von 700ms erzielt.

4.) neuronales Netz:

- Alle Versuchsreihen wurden mit verschiedenen Parametereinstellungen und Dimensionierungen durchgeführt. Gute Ergebnisse lieferten einschichtige Netze mit ca. 15-25 hidden nodes und einem Parameterset von 0.9/0.5. Die Lernrate war relativ hoch angesetzt, während das Momentum einen geringen Einfluß auf die Konvergenzgeschwindigkeit zeigte.

5.) Software:

- In Bezug auf die Performance wurden einige Verbesserungen erzielt. Das Backpropagation Programm wurde auf die Silicon Graphics portiert, so daß die höhere Rechengeschwindigkeit beim zeitaufwendigen Lernvorgang genutzt werden konnte.
- Die Software wurde um die Berechnung des lokalen Fehlers erweitert. Dieses Maß zeigte jeweils genau den Stand des aktuellen Lernvorganges an.
- Um einen besseren Überblick über den Lernvorgang zu bekommen, wurden alle lokalen Fehler in Richtungs- und Beschleunigungsfehler aufgeschlüsselt und in der Konvergenzdatei abgespeichert.
- In einer "Fehlerdatei" wurde gespeichert, welche Fehlerart bei welchem Lernmuster auftrat. Dadurch erkannte man sofort, welche Art von Mustern noch nicht richtig klassifiziert wurden. Die Fehlerdatei wurde nach jedem Lerndateidurchgang überschrieben.

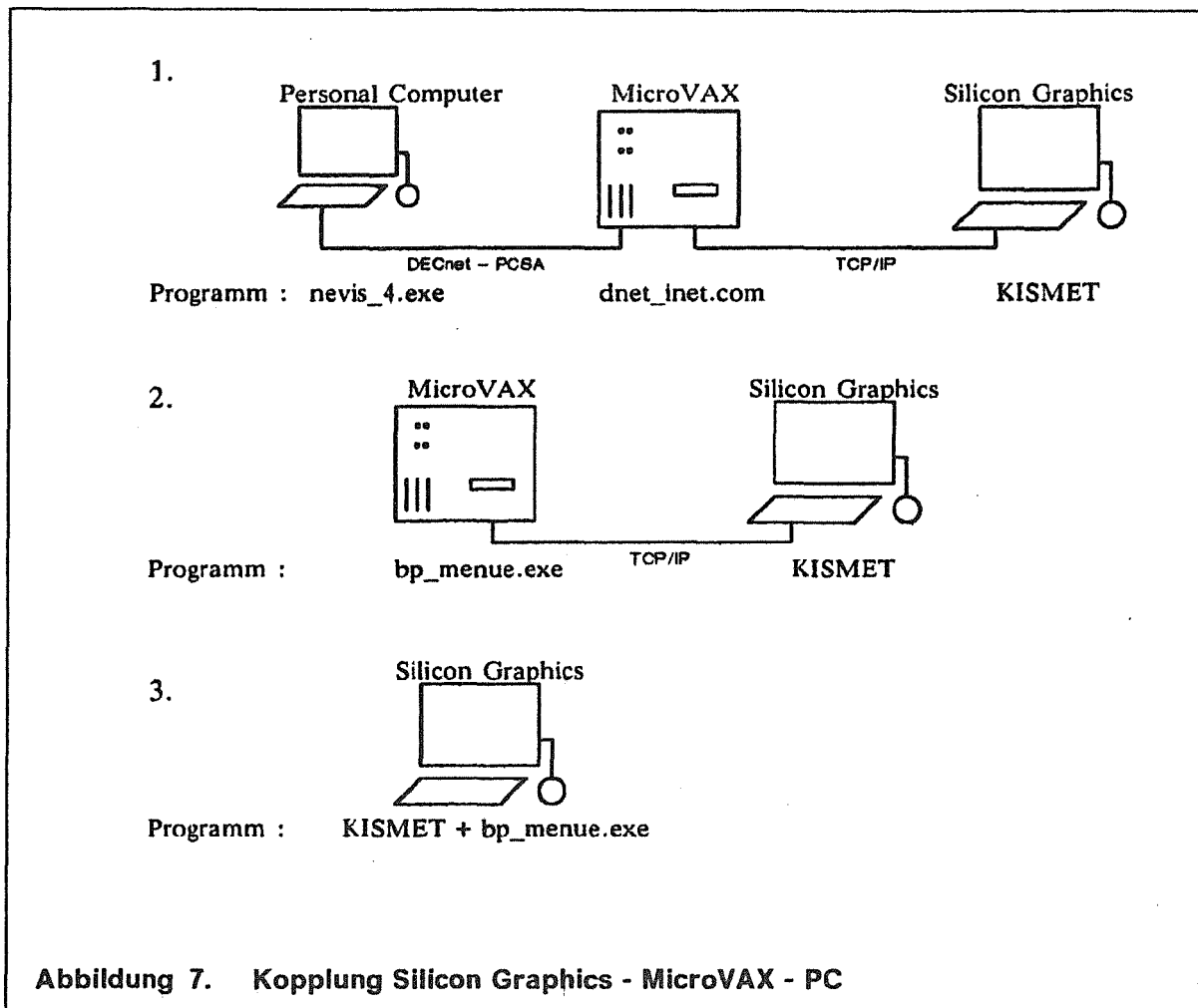
3.1.4 Kopplung Silicon Graphics - MicroVAX - PC

Es bestehen mehrere Möglichkeiten die komplette Simulation zu starten. Der Backpropagation-Simulator existiert in zwei Ausbaustufen. Die Originalversion läuft auf einem PC mit EGA-Graphik und mausorientierter Benutzeroberfläche. Diese Version wurde nun um geänderte Lern- und Recalldatensätze erweitert und angepaßt. Die PC-Version bietet die übersichtlichste und informativste Oberfläche. Damit der Backpropagation-Simulator auch auf einer MicroVAX oder SG-Workstation läuft, wurden alle graphischen Funktionen durch alphanumerische Ausgaben ersetzt. Die C-Source Programme der MicroVax- und der SG-Version unterscheiden sich daher nur um einige maschinenspezifische Aufrufe.

Die Kopplung der PC-Version mit der SG erfolgt über eine Vermittlertask, welche auf der MicroVAX gestartet wird. Die Vermittlertask bricht bei Beendigung des Programmes automatisch ab.

Bei der zweiten Version für die SG bzw. MicroVAX, ist die Vermittlertask bereits integriert. Es ist nur darauf zu achten, daß der neuronale Netzwerksimulator zuerst gestartet wird.

Diese Version kann auch als eigenständige Task im Hintergrund zum KISMET-Simulator ablaufen. Da hierbei keine Kopplungen notwendig sind, ist dies besonders nützlich für Demonstrationen.



3.2 Erweiterungen in KISMET

In KISMET wurden bisher keine fahrzeugähnlichen Roboter simuliert. Da man beabsichtigte, mit einem Fahrzeug in Teachfiles abgelegte Bahnen abzufahren, war hierzu die Entwicklung der inversen Kinematiktransformation notwendig. Auch Abstandssensoren waren nicht vorhanden und mußten mit den zugehörigen Schnittoperationen des Sensorstrahles mit beliebigen Körpern im 3D-Raum entwickelt werden. Für den Sensor wurden neue Datenstrukturen entwickelt. Ein Sensor wird analog zu einem Workframe behandelt, wobei der Meßstrahl entlang

der z-Richtung des Koordinatenkreuzes verläuft. Die Länge des Strahles kann in der zugehörigen mpplib-Datei angegeben und sowohl Position als auch Richtung über das Workframe-Menü eingestellt werden. Für die Übergabe der Sensor- und Positionsdaten an das Interface-Programm wurde die Schnittstelle neu entwickelt.

3.3 Interface-Programme zur Kommunikation zwischen KISMET und der neuronalen Netz-Software

Bei den Interfaceprogrammen muß man zwischen der Lernphase und der Kannphase der neuronalen Netz-Software unterscheiden. In der Lernphase wird eine Lerndatei erzeugt, mit der das neuronale Netz trainiert wird. Hierzu muß in KISMET der Realmode eingeschaltet werden. Die Intervallzeit wird eingestellt und mit den Sensor- und Positionsdaten übergeben. Läßt man nun das Fahrzeug im Repeatmode den Teachfile abarbeiten, so erhält man die Sensorwerte entlang der vorgegebenen Bahn. Die Werte werden mit den im Interfaceprogramm berechneten und normierten Werten der Richtungs- und Geschwindigkeitsänderung im Lernfile abgespeichert. Mit diesen Daten kann dann ein Netz trainiert werden (siehe Kapitel 4).

In der Kannphase baut das Interfaceprogramm eine TCP/IP Stream-Socket Verbindung zur neuronalen Netz Software auf. Nachdem das Fahrzeug in KISMET positioniert ist, wird das Interfaceprogramm gestartet und erhält von KISMET die Sensordaten. Diese werden an die neuronale Software weitergeleitet. Von dort werden die Aktuatorwerte und die normierte Richtungs- und Geschwindigkeitsänderung empfangen. Aus diesen Werten und dem vorgegebenen Zeitintervall wird die neue Position des Fahrzeuges berechnet und an KISMET weitergegeben. KISMET positioniert das Fahrzeug und ermittelt die neuen Sensordaten usw..

4. Teachphase

Mittels der KISMET-Simulation wurde das Fahrzeug (Abbildung 8 auf Seite 18) von Hand (supervised data generation) in einem bestehenden Fahrgrund bewegt (Situation 1-7). Einzelne Punkte dieser Bahn wurden in KISMET-Teachfiles in IR-DATA-Code abgelegt. Anschließend wurden diese Teachfiles im Repeat-Mode bei "eingeschalteter Sensorik" nachgefahren. Die dadurch entstehenden Sensorik-Aktuatorik-Daten wurden in Lerndatensätze abgelegt. Die Struktur eines Lerndatensatzes ist aus dem Anhang ersichtlich.

Einschränkungen

- Da in KISMET das Überschleifen zwischen verschiedenen Bewegungspunkten nicht realisiert ist, kann beim späteren Abrufen der IRDATA-Files keine flüssige Bewegung erzeugt werden. Das Fahrzeug fährt immer den nächsten Bezugspunkt an, stoppt und fährt dann mit der momentan aktuellen Geschwindigkeit weiter. Dieses Verhalten ist nicht erwünscht. Es verhindert, daß das Beschleunigen und Abbremsen korrekt gelernt werden kann.
- Als Folgefehler dazu zeigte sich bei trainierten Netzen oft ein Fehlverhalten in Situationen, die eigentlich korrekt gelernt wurden. Gerät das Fahrzeug in eine schwierige Situation und gibt ihm das Netz gleichzeitig eine zu hohe Geschwindigkeit vor, so kann das in vielen Fällen zum Crash führen.
- Trotz Lernbeispielen reagiert das Netz bis zum derzeitigen Entwicklungsstand oft falsch, wenn nur drei der vorderen Sensoren rechts oder links Signale erhalten. So werden gelegentlich die Enden von Mauern, Außenecken, oder alleinstehende Gegenstände touchiert.

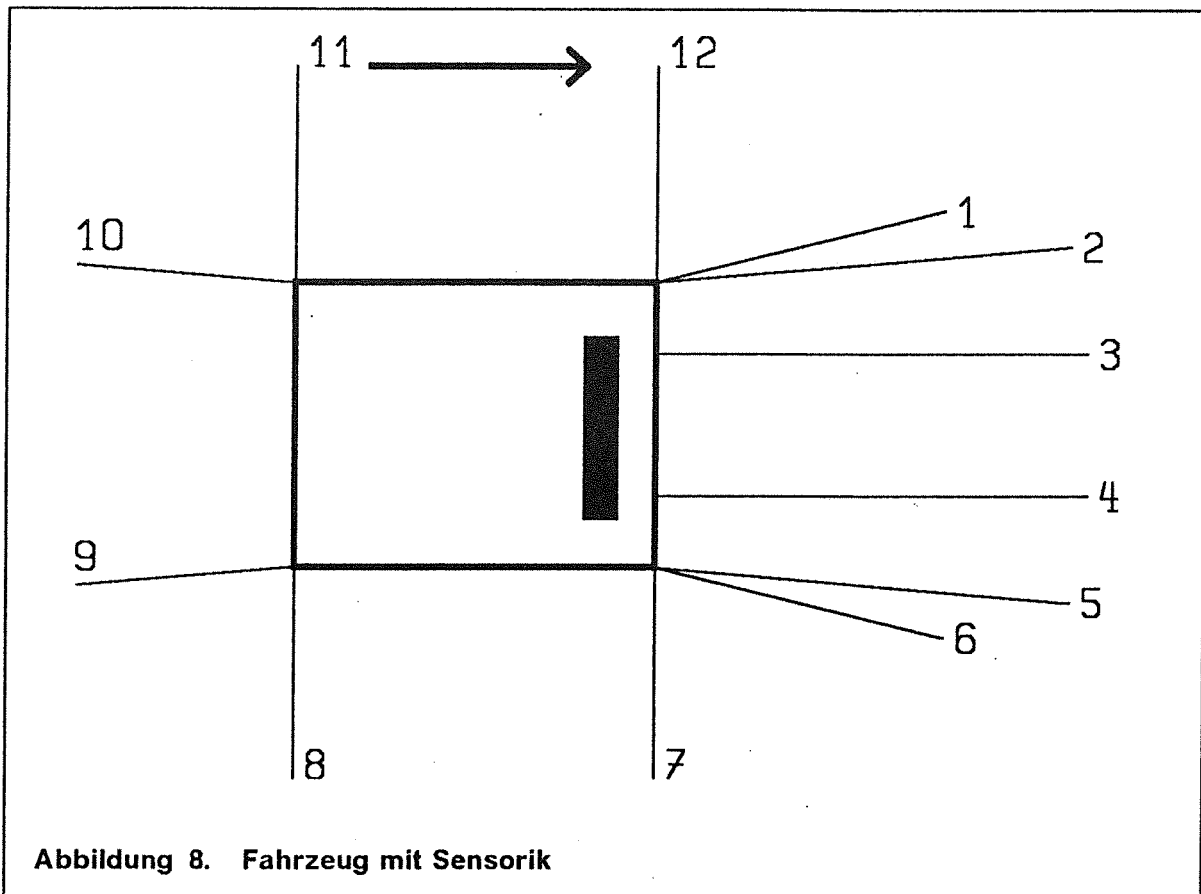


Abbildung 8. Fahrzeug mit Sensorik

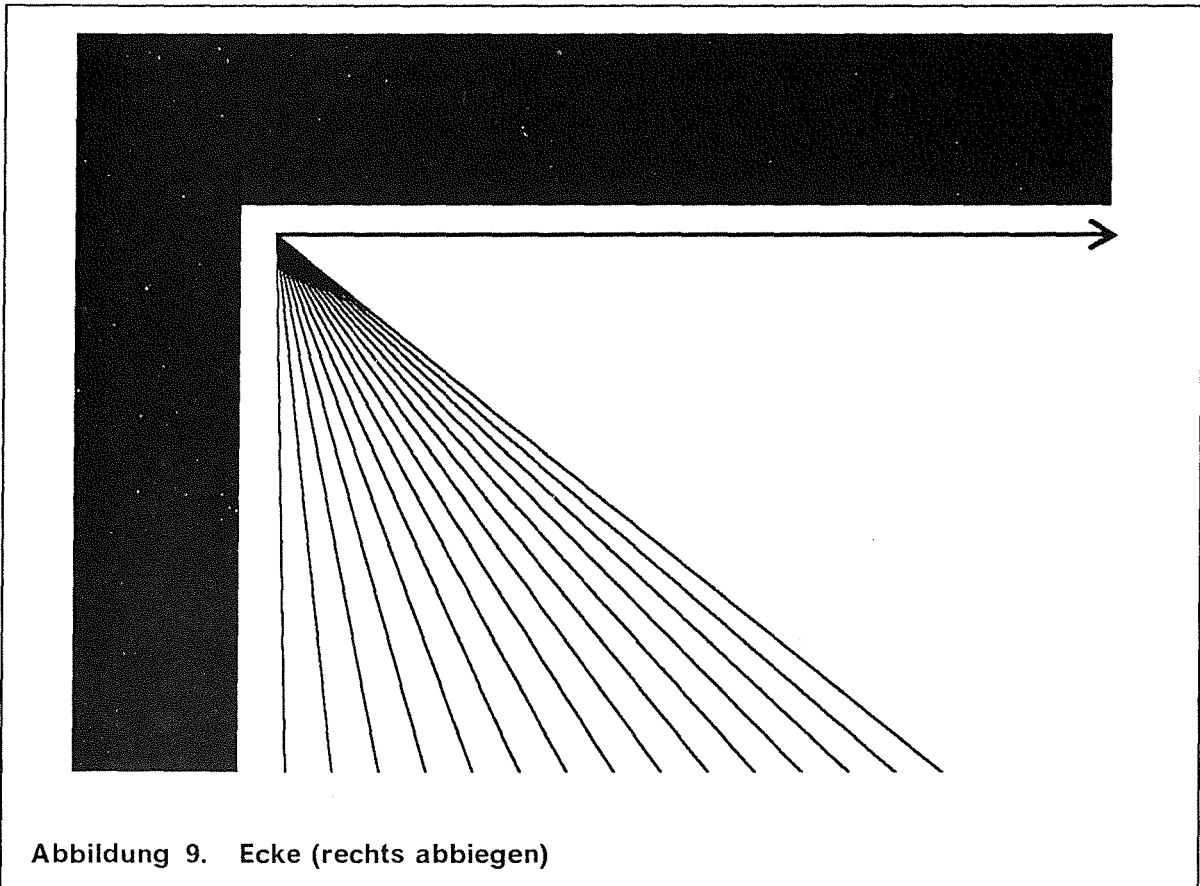
Nach einigen Tests wurde der Aufbau der Sensorik wie folgt festgelegt:

Länge der hinteren Sensoren (9, 10):	2000 mm.
Winkel der hinteren Sensoren (9, 10):	-175 Grad bzw. 175 Grad.
Länge der seitlichen Sensoren (7, 8, 11, 12):	2000 mm.
Winkel der seitlichen Sensoren (7, 8, 11, 12):	-90 Grad bzw. 90 Grad.
Länge der vorderen Sensoren (1, 6):	4000 mm.
Winkel der vorderen Sensoren (1, 6):	-15 Grad bzw. 15 Grad.
Länge der vorderen Sensoren (2, 5):	6000 mm.
Winkel der vorderen Sensoren (2, 5):	-5 Grad bzw. 5 Grad.
Länge der vorderen Sensoren (3, 4):	6000 mm.
Winkel der vorderen Sensoren (3, 4):	0 Grad.

4.1 Fahrgründe

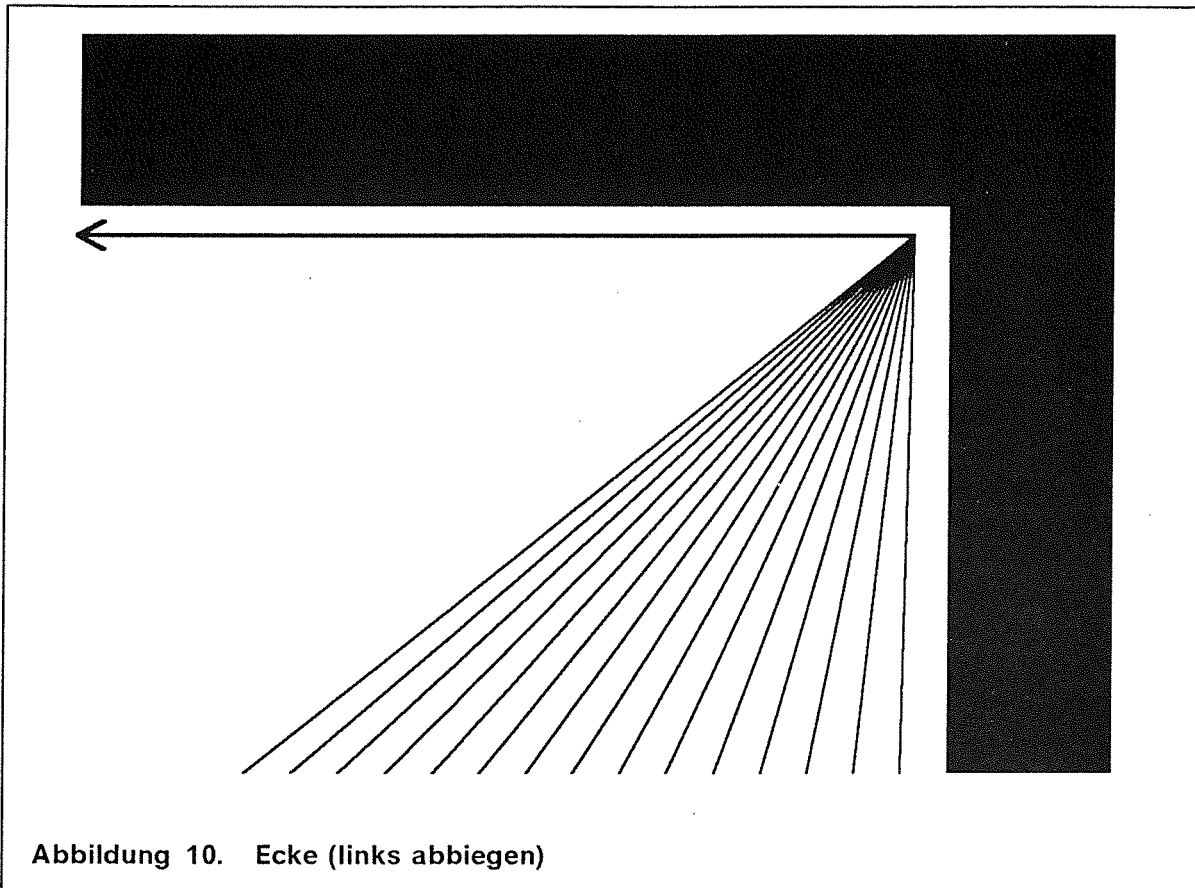
Schon nach wenigen Versuchen zeigte sich, daß eine ausreichende Generalisierung des Netzes nicht durch Trainieren eines einzelnen Fahrgrundes (proving ground) erreicht werden kann. Bei der Generierung der Lerndaten wurde schließlich davon ausgegangen, daß die "Umwelt", in der sich das Fahrzeug zu rechtfinden soll, durch einige Beispieldatensätze umfassend abbildbar ist. Im folgenden werden die einzelnen Hindernissituationen und die dazu erstellten Lernfiles aufgezeigt. Durch Aneinanderreihen sämtlicher Lernfiles wurde eine Gesamtlerndatei zum Training des Netzwerkes erzeugt.

Situation 1:



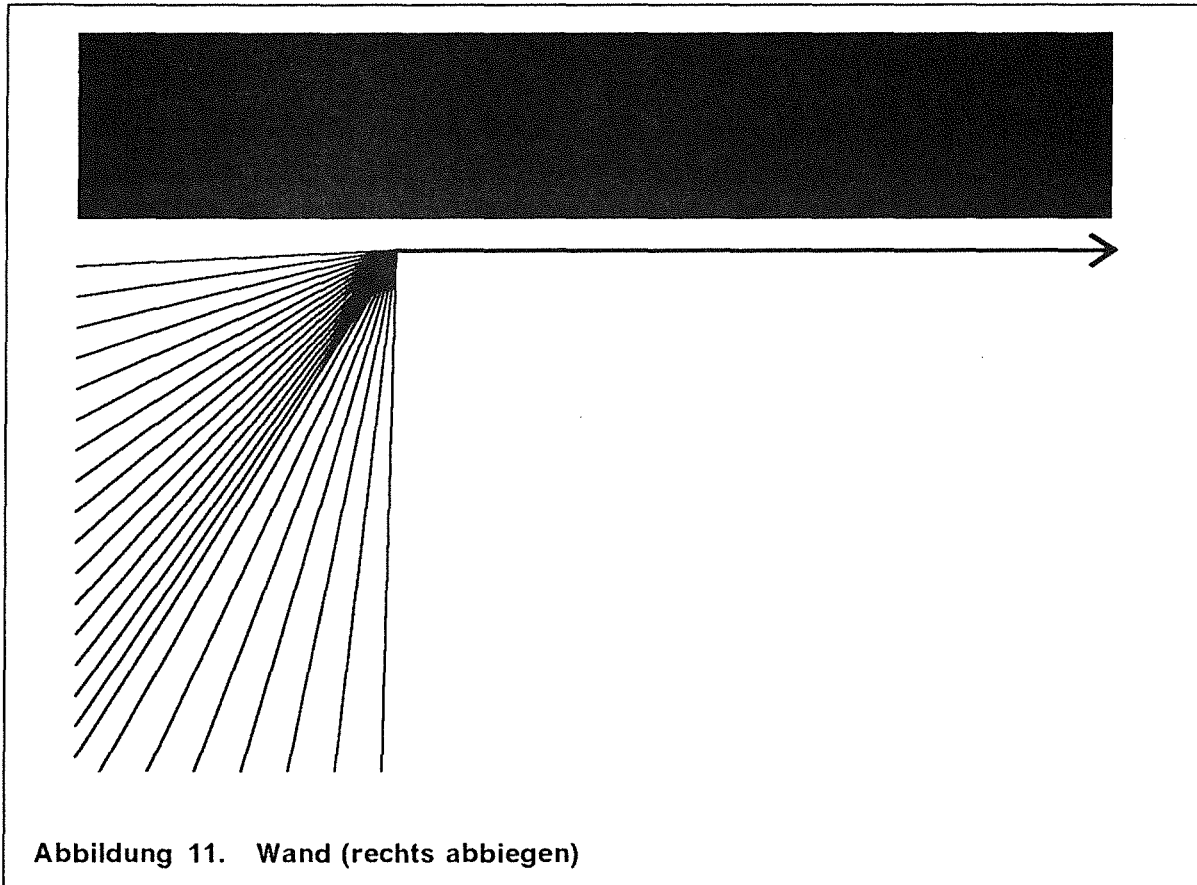
Aus verschiedenen Startpositionen wurde insgesamt 15 mal in die Ecke (Abbildung 9) gefahren und rechts abgebogen. Dabei wurde ein Winkel von 45 Grad nicht überschritten. Unabhängig von den verschiedenen Startpositionen ist nach dem Abbiegen möglichst parallel zur Wand weitergefahren worden. Beim Auftreffen der vorderen Sensoren auf die Ecke wurde die Geschwindigkeit von 100% des Maximalwertes auf 50% des Maximalwertes reduziert. Nachdem sich das Fahrzeug um eine halbe Länge der vorderen Sensoren weiter dem Hindernis genähert hatte, wurde die Geschwindigkeit erneut auf diesmal 10% der Maximalgeschwindigkeit reduziert. Mit gleichbleibender Geschwindigkeit wurde nun das Fahrzeug durch die Ecke gesteuert. Danach erfolgte eine Beschleunigung von 10% über 50% bis zu 100% der Maximalgeschwindigkeit.

Situation 2:



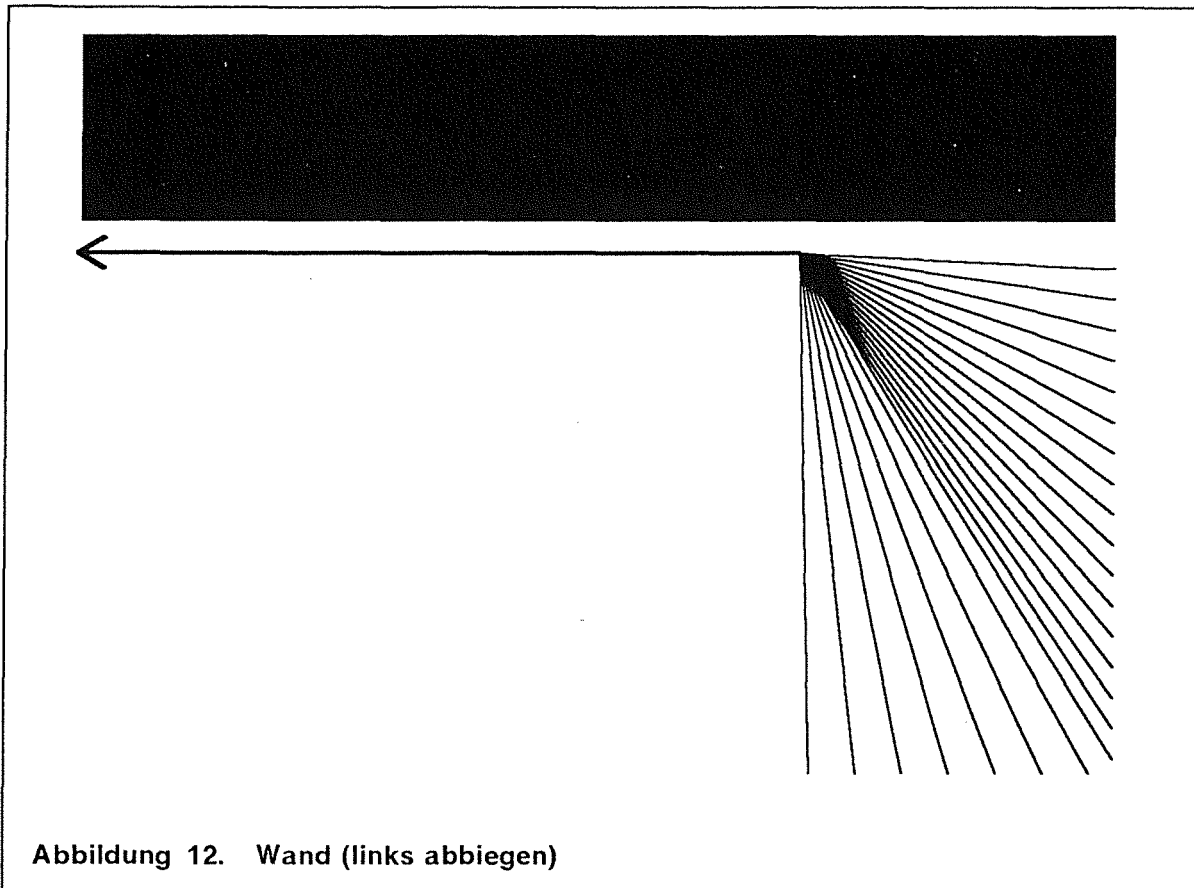
Aus verschiedenen Startpositionen wurde insgesamt 15 mal in die Ecke (Abbildung 10) gefahren und links abgebogen. Dabei wurde ein Winkel von 45 Grad nicht überschritten. Unabhängig von den verschiedenen Startpositionen ist nach dem Abbiegen möglichst parallel zur Wand weitergefahren worden. Beim Auftreffen der vorderen Sensoren auf die Ecke wurde die Geschwindigkeit von 100% des Maximalwertes auf 50% des Maximalwertes reduziert. Nachdem sich das Fahrzeug um eine halbe Länge der vorderen Sensoren weiter dem Hindernis genähert hatte, wurde die Geschwindigkeit erneut auf diesmal 10% der Maximalgeschwindigkeit reduziert. Mit dieser Geschwindigkeit wurde das Fahrzeug durch die Ecke gesteuert. Danach erfolgte eine Beschleunigung von 10% über 50% bis zu 100% der Maximalgeschwindigkeit.

Situation 3:



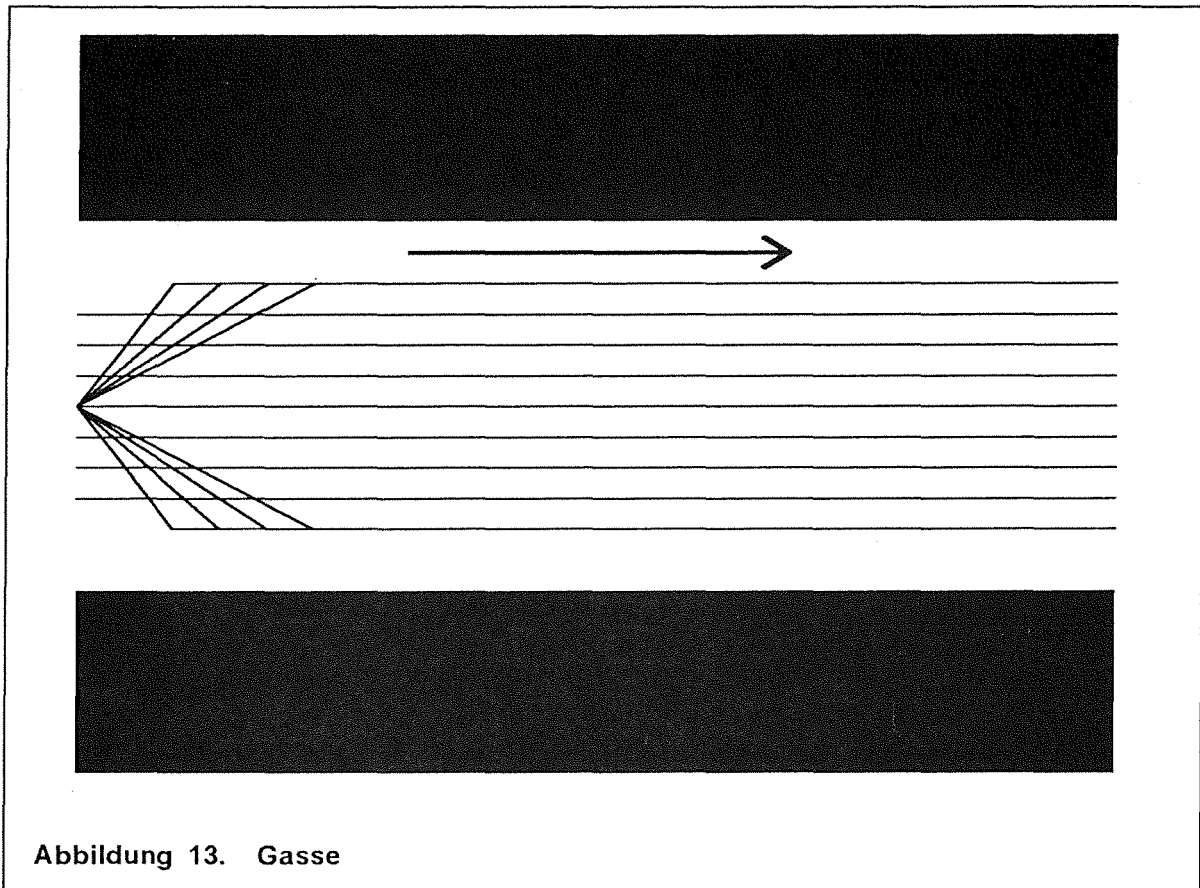
Aus verschiedenen Startpositionen wurde anfänglich insgesamt 15 mal auf die Wand (Abbildung 11) zugefahren und rechts abgebogen. Dabei wurde ein Winkel von 90 Grad nicht überschritten. Unabhängig von den verschiedenen Startpositionen ist nach dem Abbiegen möglichst parallel zur Wand weitergefahren worden. Beim Auftreffen der vorderen Sensoren auf die Wand wurde die Geschwindigkeit von 100% auf 50% des Maximalwertes reduziert. Nachdem sich das Fahrzeug um eine halbe Länge der vorderen Sensoren weiter dem Hindernis genähert hatte, wurde die Geschwindigkeit erneut auf diesmal 10% der Maximalgeschwindigkeit reduziert. Mit dieser Geschwindigkeit wurde das Fahrzeug um die Kurve gelenkt. Danach erfolgte eine Beschleunigung von 10% über 50% bis zu 100% der Maximalgeschwindigkeit. Nach der Lernphase zeigte sich, daß das Fahrzeug bei nahezu senkrechtem Auftreffen auf eine Wand diese meistens durchfuhr. Dieser Fall wurde mit weiteren 8 Beispielen nachgelernt.

Situation 4:



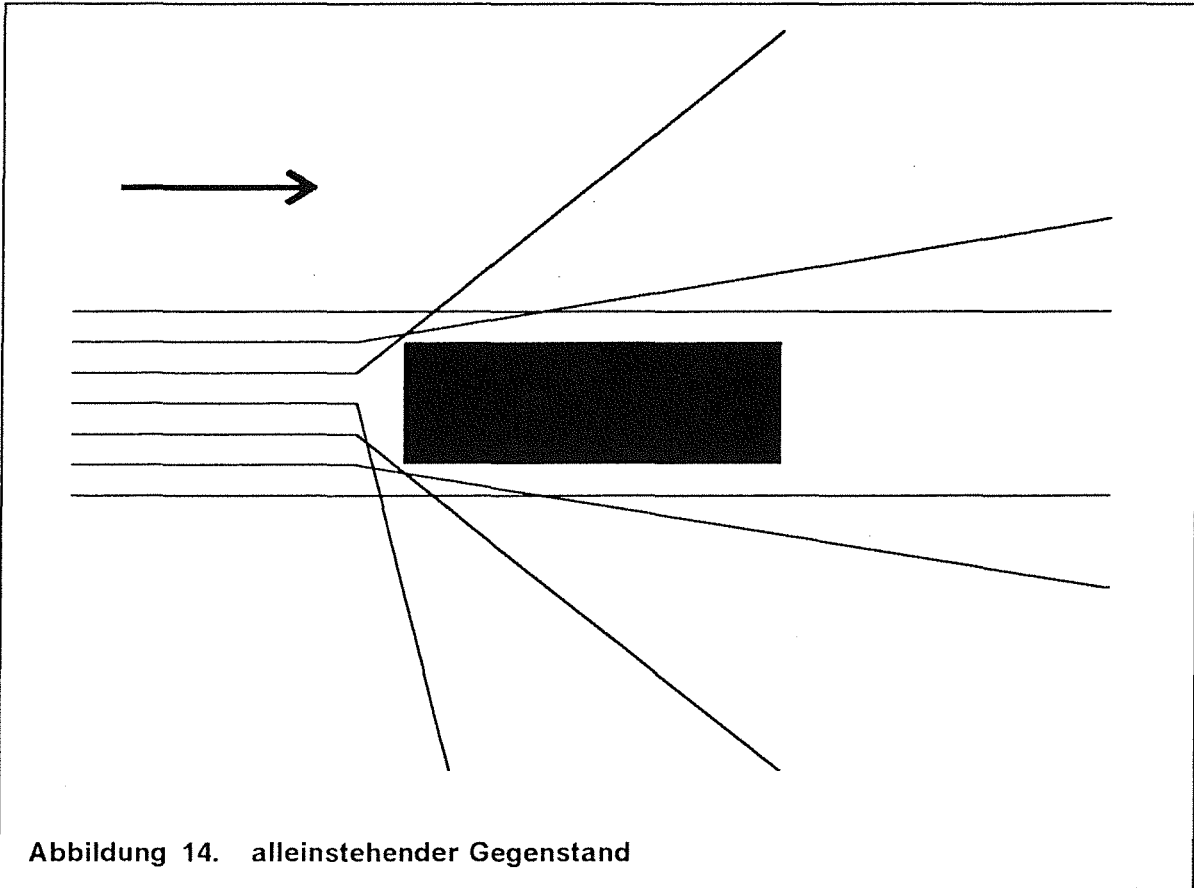
Aus verschiedenen Startpositionen wurde anfänglich insgesamt 15 mal auf die Wand (Abbildung 12) zugefahren und links abgelenkt. Dabei wurde ein Winkel von 90 Grad nicht überschritten. Unabhängig von den verschiedenen Startpositionen ist nach dem Abbiegen möglichst parallel zur Wand weitergefahren worden. Beim Auftreffen der vorderen Sensoren auf die Wand wurde die Geschwindigkeit von 100% auf 50% des Maximalwertes reduziert. Nachdem sich das Fahrzeug um eine halbe Länge der vorderen Sensoren weiter dem Hindernis genähert hatte, wurde die Geschwindigkeit erneut auf diesmal 10% der Maximalgeschwindigkeit reduziert. Mit dieser Geschwindigkeit wurde das Fahrzeug um die Kurve gelenkt. Danach erfolgte eine Beschleunigung von 10% über 50% bis zu 100% der Maximalgeschwindigkeit. Nach der Lernphase zeigte sich, daß das Fahrzeug bei nahezu senkrechtem Auftreffen auf eine Wand diese meistens durchfuhr. Dieser Fall wurde mit weiteren 7 Beispielen nachgelernt.

Situation 5:



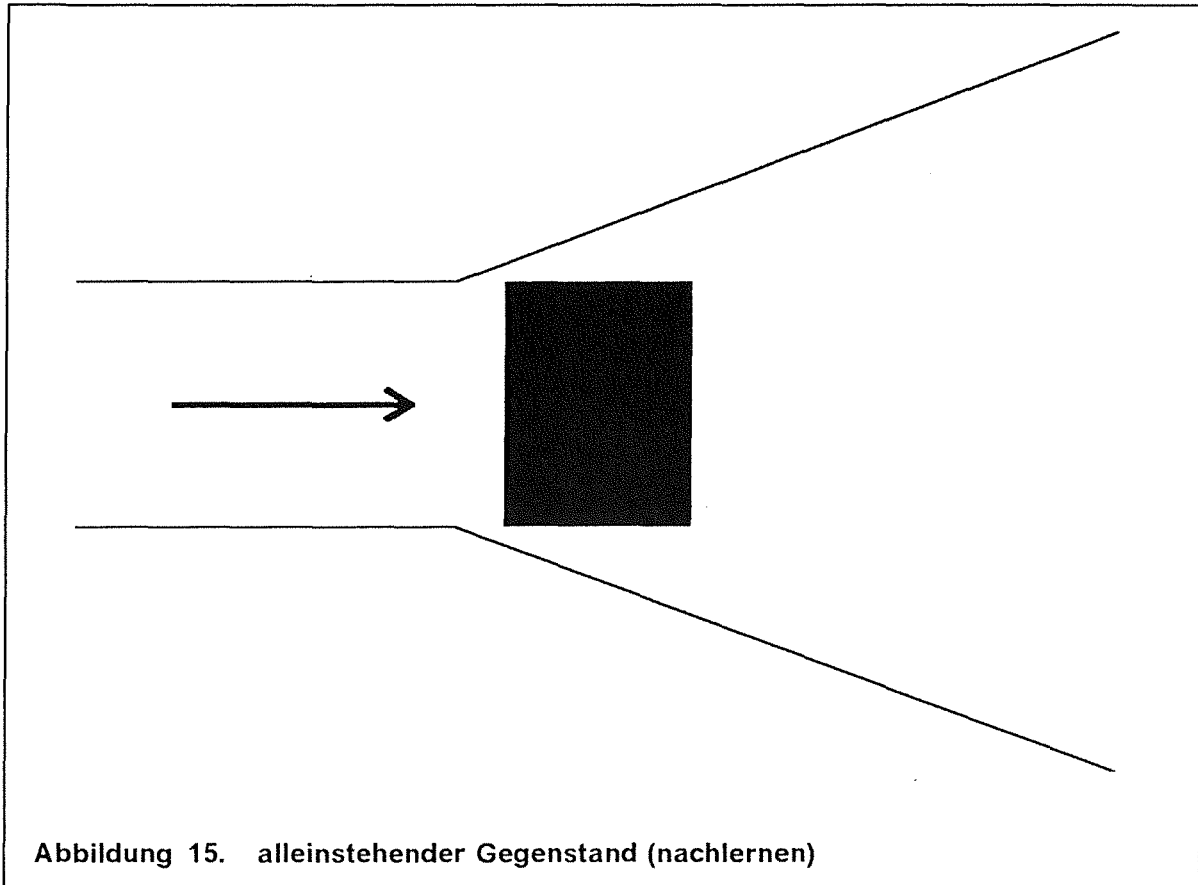
Aus verschiedenen Startpositionen wurde insgesamt 13 mal durch eine Gasse (Abbildung 13) hindurchgefahren. Hierbei wurde die Gasse zunächst aus 5 verschiedenen Startpositionen parallel zu den Wänden durchquert. Danach wurde insgesamt 4 mal mit unterschiedlich großen Winkeln auf die linke Wand zugefahren. Das Abbremsen staffelte sich von 100% über 50% bis zu 10% des Geschwindigkeits-Maximalwertes. Nach dem Abbiegen wurde das Fahrzeug parallel zu den Wänden orientiert. Die Beschleunigung staffelte sich von 10% über 50% bis zu 100% des Geschwindigkeits-Maximalwertes. Weitere 4 mal wurde nun das Fahrzeug auf die rechte Wand zubewegt. Es wurde wie im oben beschriebenen Fall verfahren.

Situation 6:



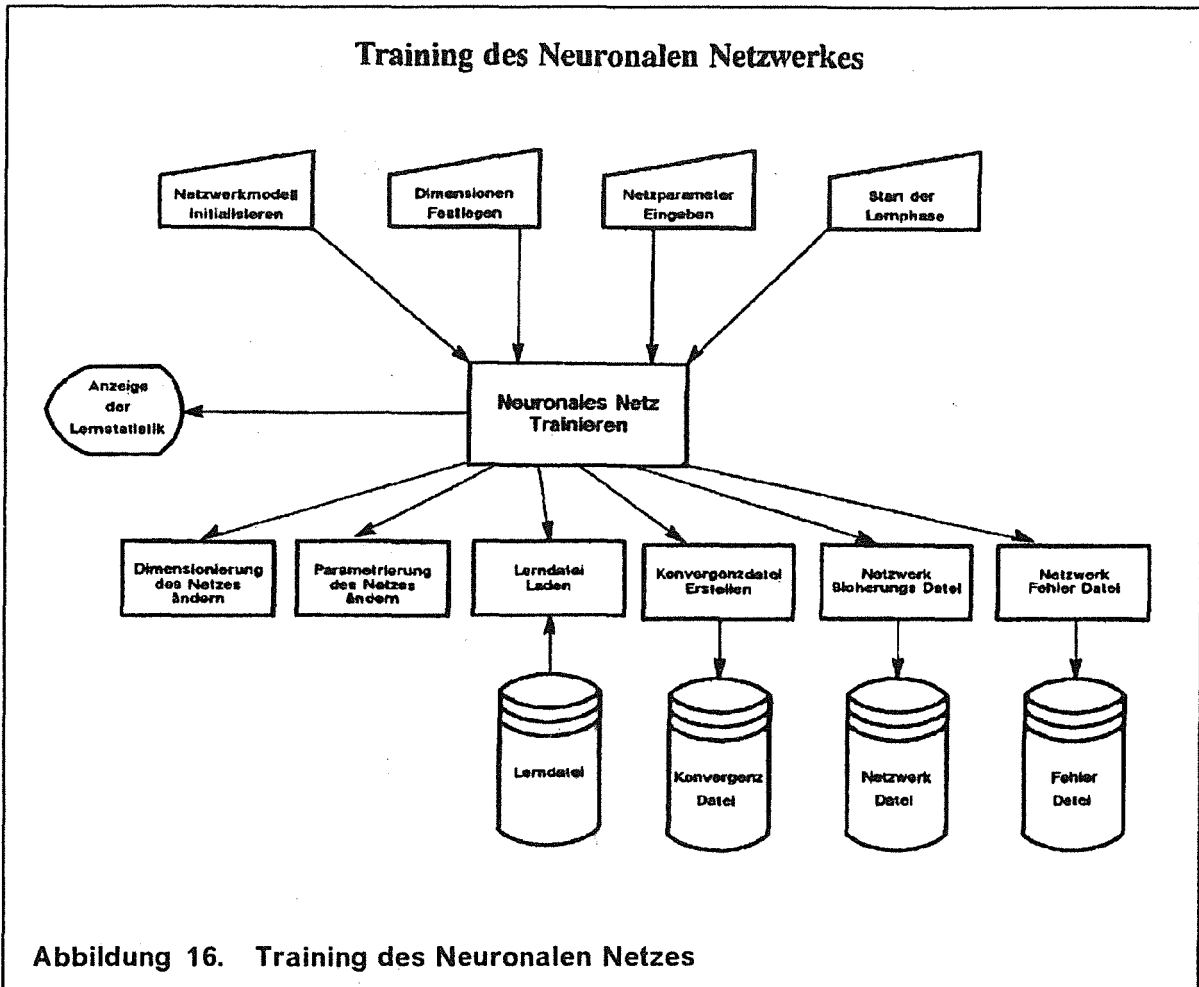
Aus verschiedenen Startpositionen wurde insgesamt 15 mal auf ein alleinstehendes Hindernis (Abbildung 14) zugefahren. Beim Auftreffen der vorderen Sensoren auf das Hindernis wurde die Geschwindigkeit von 100% auf 50% des Maximalwertes reduziert. Nachdem sich das Fahrzeug um eine halbe Länge der vorderen Sensoren weiter dem Hindernis genähert hatte, wurde die Geschwindigkeit erneut auf diesmal 10% der Maximalgeschwindigkeit reduziert. Mit dieser Geschwindigkeit wurde das Fahrzeug je nach Position 8 mal nach rechts und 7 mal nach links an dem Hindernis vorbeigelenkt. Es wurde keine parallele Orientierung zum Hindernis nach dem Abbiegen vorgenommen. Danach erfolgte eine Beschleunigung von 10% über 50% bis zu 100% der Maximalgeschwindigkeit. Nach der Lernphase zeigte sich, daß das Fahrzeug, sofern nur 3 der vorderen Sensoren auf ein Hindernis trafen, meistens nicht reagierte und das Hindernis durchquerte. Dieser Fall wurde, wie in Abbildung 8 beschrieben, nachgelernt.

Situation 7:



Aus verschiedenen Startpositionen wurde insgesamt 2 mal auf ein alleinstehendes Hindernis (Abbildung 15) zugefahren. In einem Fall trafen die rechten 3 der vorderen Sensoren auf das Hindernis, und das Fahrzeug wurde links an ihm vorbeigelenkt. Im anderen Fall trafen die linken 3 der vorderen Sensoren auf das Hindernis, und das Fahrzeug wurde rechts an ihm vorbeigelenkt. Im einzelnen wurde wie in Abbildung 7 beschrieben verfahren. Die zwei verschiedenen Beispiele wurden zunächst jeweils 4 mal in den später erstellten großen Lernfile gemischt. Nach einigen Tests mit diesen Zusatzdaten war erkennbar, daß die nachgelernte Situation nun zwar durch das resultierende Netz bewältigt werden konnte, daß allerdings andere zuvor gelernte Situationen wieder verlernt wurden. Die Zusatzdaten wurden daraufhin gestrichen.

4.2 Training des neuronalen Netzwerkes



Das neuronale Netzwerk besteht aus zwei Outputknoten und 13 Inputknoten. Dabei repräsentieren 12 Inputknoten die aktuelle Sensorinformation und der 13. Inputknoten die aktuelle Geschwindigkeit. Der Aufbau einer Lerndatei und eines Netzwerkes sind im Anhang ersichtlich. In Abbildung 16 ist der Lernvorgang der neuronalen Netze mitsamt unterschiedlichen Einstellparametern skizziert. Während des Trainings werden sogenannte Konvergenzdateien und Fehlerdateien aufgebaut (siehe Anhang). Sie enthalten Daten wie globale Fehler, Richtungsfehler, Beschleunigungsfehler und beinhalten die exakte Position der noch nicht gelernten Daten in der Lerndatei. Das neuronale Netzwerk kann während des Lernvorganges nach jedem Lerndateidurchgang zwischengespeichert werden. Bezüglich der Netzwerkdimension und den Parametereinstellungen wurden mehrere Tests durchgeführt, die anschließend aufgezeigt werden. Sie erheben keinerlei Anspruch auf Vollständigkeit.

4.3 Statistik über das Lernverhalten des verwendeten Netzes

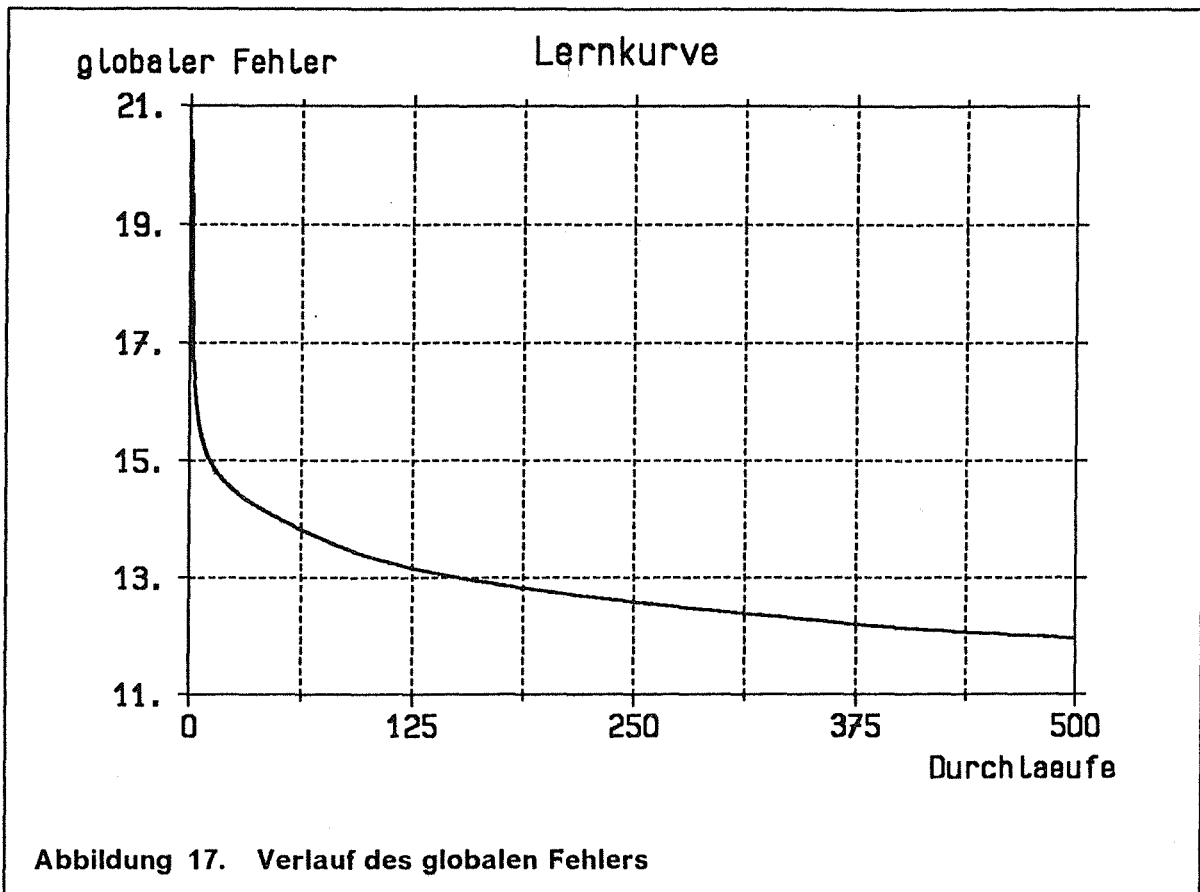
Das Training der neuronalen Netze wurde mit unterschiedlichen Dimensionen und Parametereinstellungen durchgeführt. Nach einiger Zeit und vielen Tests legten wir uns auf eine "Standard"-Konfiguration fest, die die besten Ergebnisse und schnellste Konvergenz lieferte (vergleiche Abbildung 20 auf Seite 32).

	N1	N2	N3	N4	N5	N6	N7
Eingabeknoten:	13	13	13	13	13	13	13
1. hidden layer:	19	21	23	25	27	24	26
2. hidden layer:	0	0	0	0	0	0	0
3. hidden layer:	0	0	0	0	0	0	0
Ausgabeknoten:	2	2	2	2	2	2	2
Lernrate:	0.9	0.9	0.9	0.9	0.9	0.9	0.9
Momentum:	0.6	0.6	0.6	0.6	0.6	0.6	0.6

Tabelle 4. Vergleich der verschiedener Netzkonfigurationen

Die Lernrate wurde hin und wieder auf 0.99 gesetzt. Die Auswahl der Dimensionen und Parameter ist sehr heuristisch und schwer zu begründen. Eine umfassende Analyse und Statistik über Parameter/Dimensionen wäre hier wünschenswert.

Auf den folgenden Seiten folgen Statistiken von Parametern, die den Lernzustand des neuronalen Netzes in Abhängigkeit zu der Anzahl von Lernzyklen wiedergeben. Die Funktionen werden anhand der Daten der jeweiligen Konvergenzdateien gezeichnet und sollen hier beispielhaft erläutert werden.



Der globale Fehler des neuronalen Netzes ist der wichtigste Wert, um den momentanen Lernzustand zu beurteilen. Er berechnet sich nach D.E.Rummelhard zu:

$$E = \frac{1}{2} \sum_m \sum_n \text{lok}_{\text{err}}^2$$

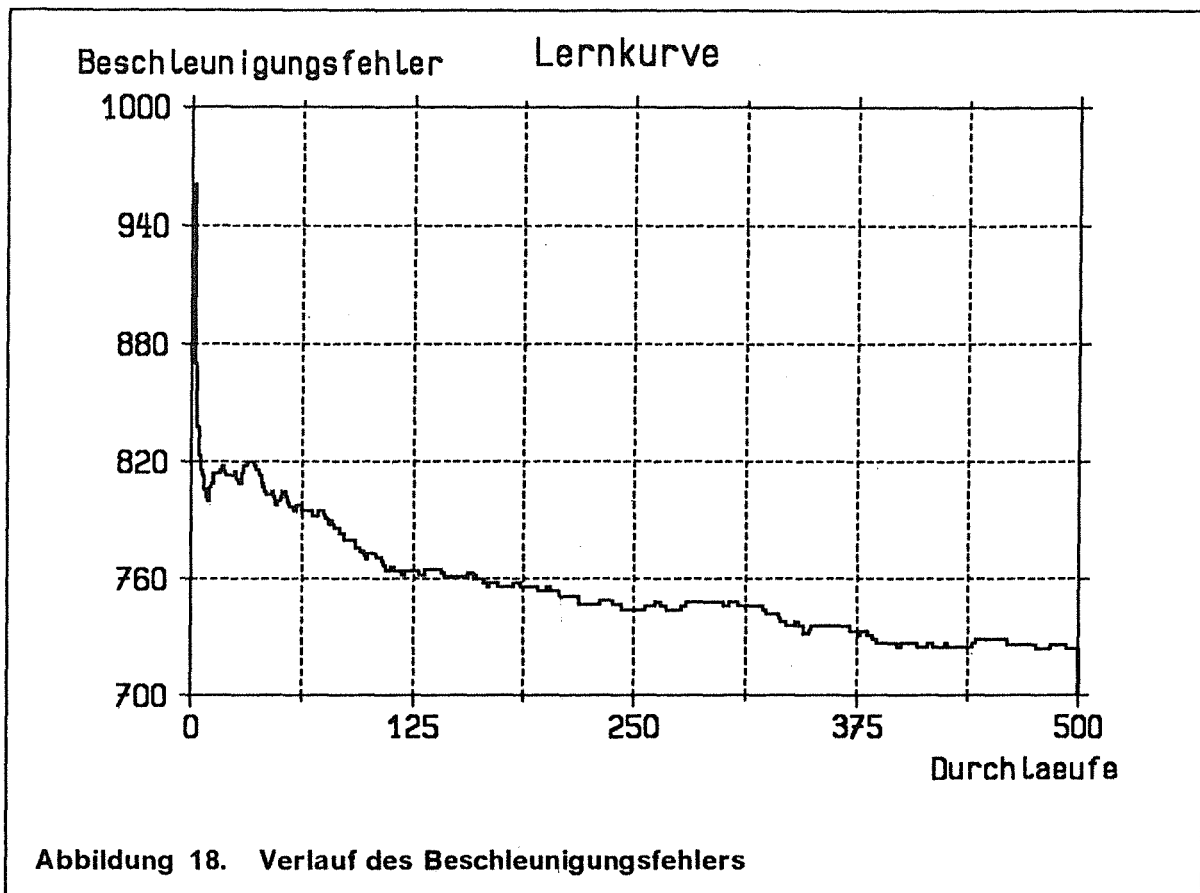
wobei:

lok_{err} : lokaler Fehler (Differenz aus Soll- und Ist-Wert)

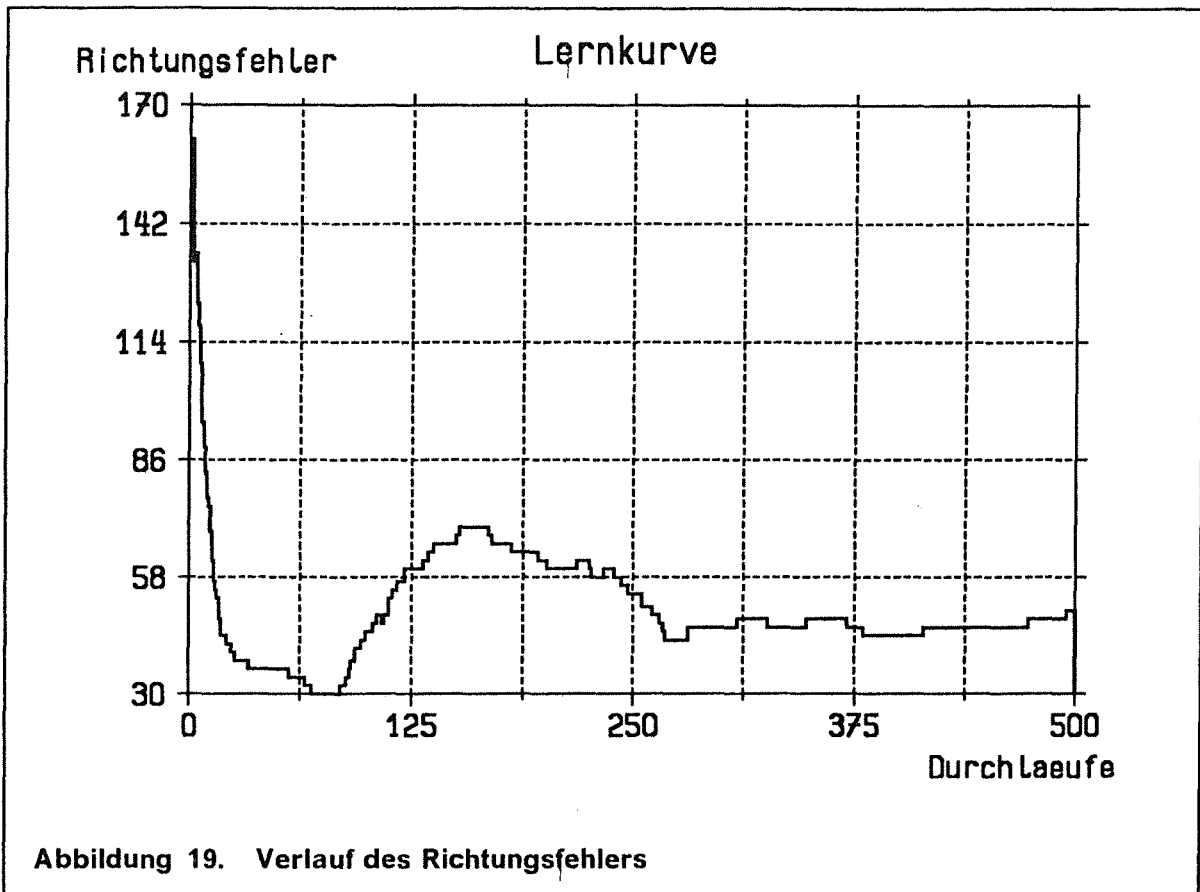
n: Anzahl der Outputknoten

m: Anzahl der Lernmuster

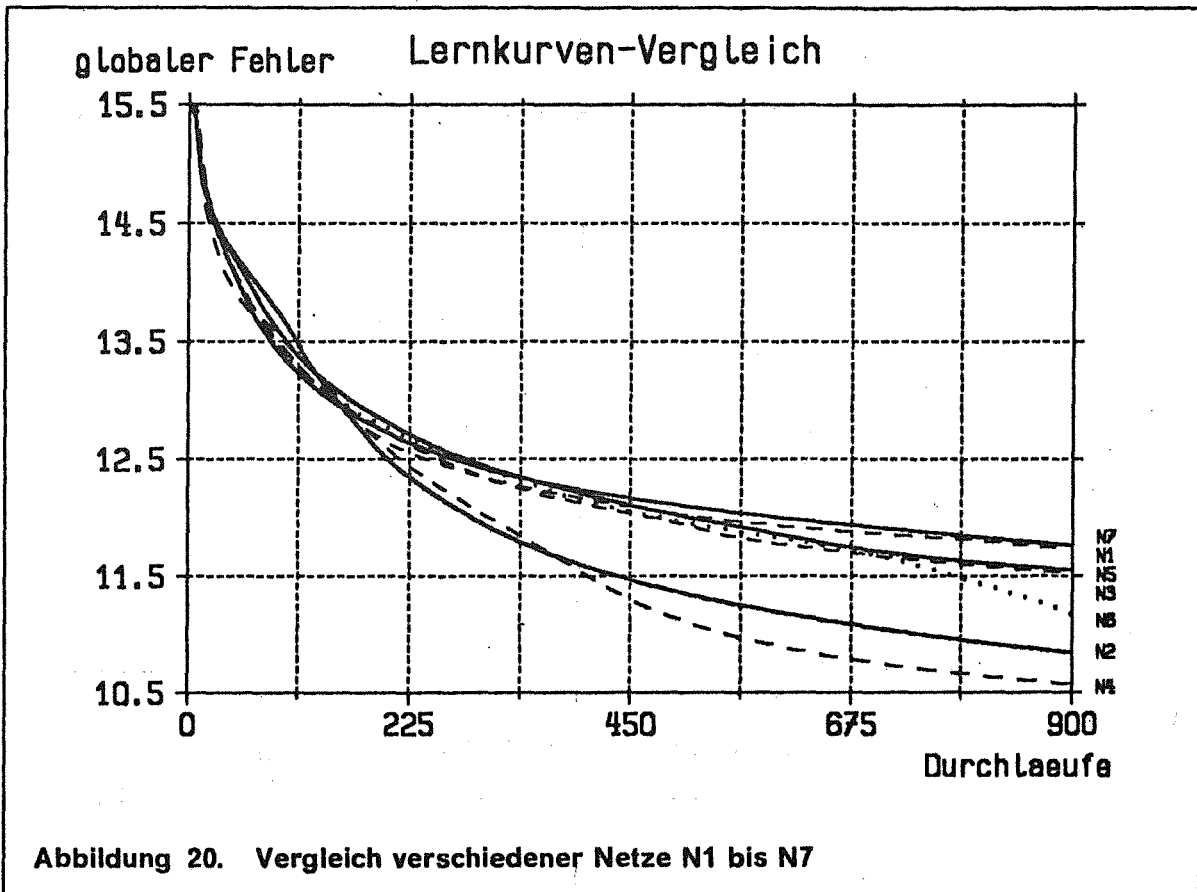
Wie man erkennt, ist das Netz nach ca.100 bis 150 Durchläufen in seiner Konvergenz schon sehr weit fortgeschritten. Obwohl das Netz durchaus noch in der Lage ist, weiter zu konvergieren, wird die Konvergenzgeschwindigkeit jedoch gering. Es gilt, ein "Übertrainieren" des Netzes zu verhindern. Es zeigte sich zum Beispiel des öfteren eine relativ gute Performance nach ca. 10-20 Mio. Lernschritten, wobei die Performance nach 80 Mio. Lernschritten deutlich nachließ.



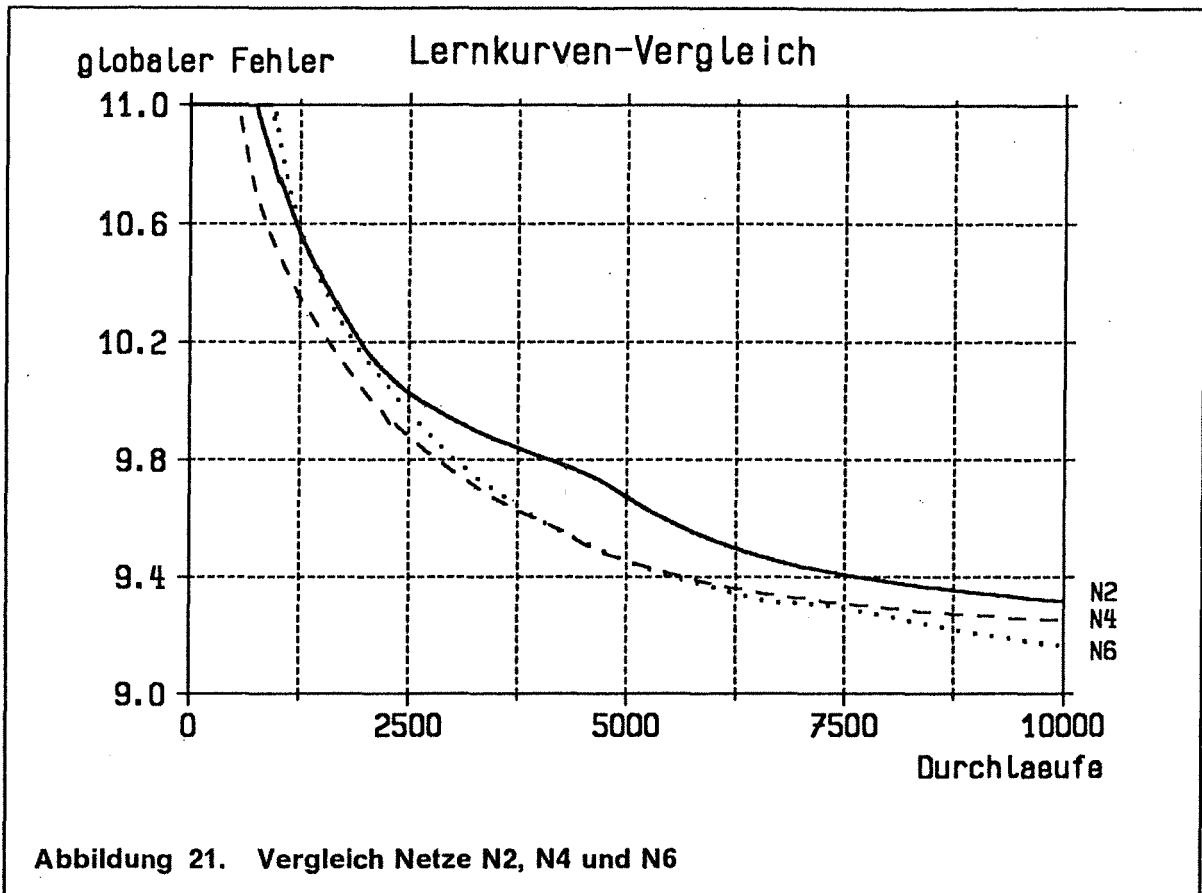
In Abbildung 18 wird der Verlauf des Beschleunigungsfehlers dargestellt. Dieser gibt an, in wie vielen Fällen eine falsche Zuordnung bezüglich des Beschleunigungsoutputknotens des Netzes gemacht wurde. Bei dem hier verwendeten Beispielen gibt er wieder, wie oft das Netz bei 2399 verwendeten Lerndatensätzen falsch reagiert hat. Gleiches gilt übertragen für das Maß des Richtungsfehlers, der auf der folgenden Seite beschrieben wird.



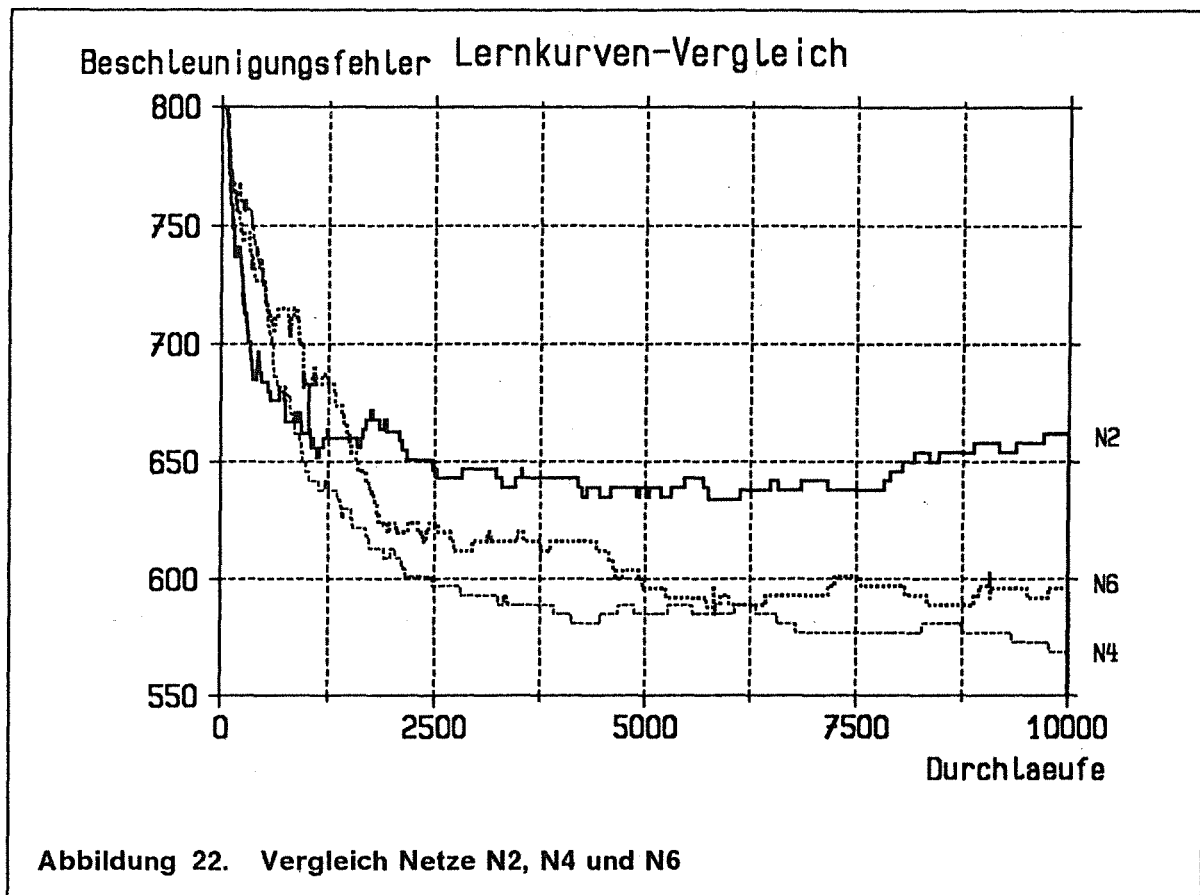
In Abbildung 19 wird der Verlauf des Richtungsfehlers dargestellt. Er gibt an, in wie vielen Fällen eine falsche Zuordnung bezüglich des Richtungsoutputknotens des Netzes gemacht wurde. Deutlich zu erkennen ist, daß das Netz sehr rasch ein relatives Minimum bezüglich des Richtungsfehlers erreicht hat. Im weiteren Verlauf verschlechterte sich dieser Wert jedoch wieder, während der Beschleunigungsfehler langsam geringer wurde.



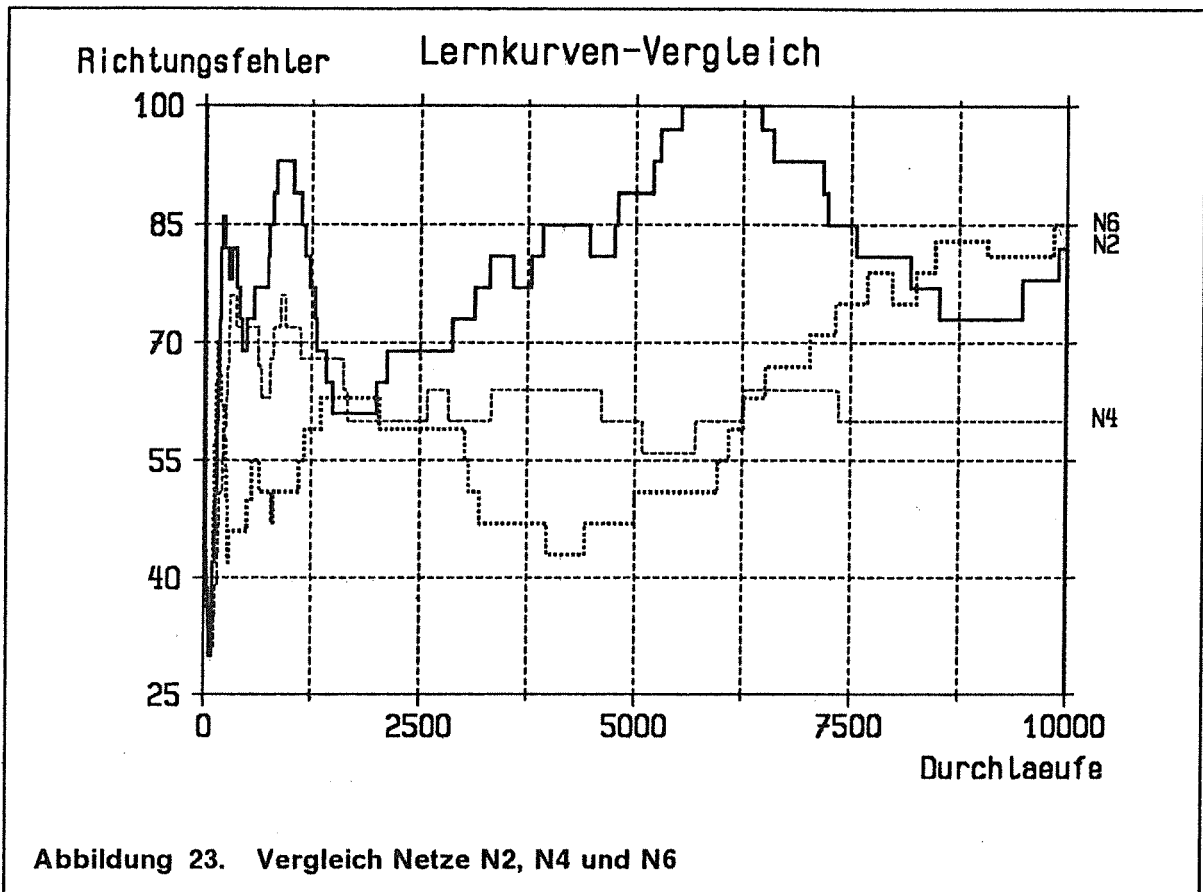
Um sich für ein bestimmtes Netz entscheiden zu können, wurden verschiedene Netzwerkarchitekturen (Tabelle 4 auf Seite 28) eine begrenzte Zeit trainiert (Abbildung 20). Nach 900 Lerndateidurchläufen zeigte sich eine bessere Konvergenz bei den Netzen N2, N4, und N6. Diese wurden daraufhin weitertrainiert.



Das Netz N6 zeigt bei 10000 Lerndateidurchläufen ein besseres Konvergenzverhalten als N2 und N4, obwohl diese N6 bei 900 Durchläufen noch überlegen waren. Man sieht allerdings, daß das anfänglich schnelle Absinken des globalen Fehlers im Laufe der Zeit immer langsamer wird.



Der Beschleunigungsfehler zeigt ein anderes Verhalten. Sein Graph ist nicht wie der des globalen Fehlers mehr oder weniger monoton, sondern Auf- und Abwärtsschwankungen unterworfen. Aus Anschauungsgründen ist er hier als Histogramm dargestellt. Man erkennt leicht, daß längeres Lernen nicht unbedingt einen Abfall des Beschleunigungsfehlers zur Folge hat.



Der Vergleich der Lernkurven in Bezug auf den Richtungsfehler spiegelt am deutlichsten den Effekt des Übertrainierens von Netzen wieder. Während der ersten 500 Lerndateidurchläufen entstehen Peeks nach oben und unten. Wie bei Abbildung 19 auf Seite 31 beschrieben, wird zunächst das Lenkverhalten erlernt. Zeitverschoben dazu setzt das Lernen der Beschleunigung ein (vergleiche Abbildung 22 auf Seite 34). Das Netz verliert jetzt wieder sein anfängliches "Richtungswissen". Es ist nicht sinnvoll, nur bis zu dem ersten frühen relativen Tiefpunkt des Richtungsfehlers zu lernen. Ein solches Netz zeigt kein ausreichendes Fahrverhalten. Hier wird der große Einfluß einer korrekten Beschleunigung sichtbar.

Der nächste relative Tiefpunkt des Richtungsfehlers tritt bei N6 nach 4040 Lerndateidurchläufen bzw. 9691960 Lernschritten auf. Trainiert man das Netz bis zu diesem Punkt, zeigt es das beste Verhalten. Weitertrainieren führt zusehends zur Verschlechterung des Fahrverhaltens.

5. Kannphase

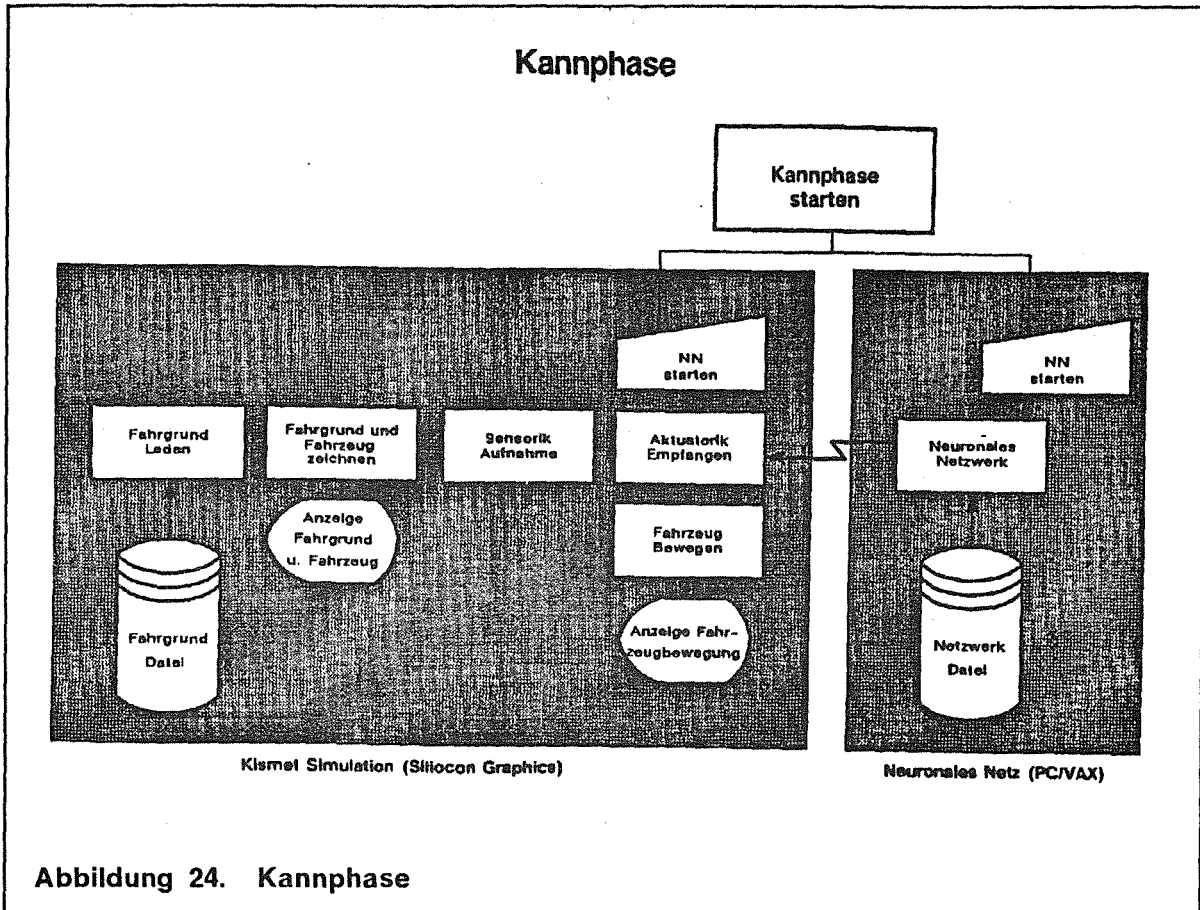


Abbildung 24. Kannphase

Zum Start der kompletten Simulation sind zwei Programmpakete notwendig. Zuerst wird ein trainiertes neuronales Netz in den Simulator geladen und die Option "Recall Remote" ausgewählt. Es werden nun sogenannte logische Sockets aufgebaut und das Neuronale Netz wartet auf Requests von außen.

In der KISMET Simulationsumgebung wird nun das Fahrzeug in einem frei wählbaren Fahrgrund beliebig postiert und ausgerichtet. Anschließend wird die Verbindung zum neuronalen Netz aufgebaut.

Im folgenden sieht man ein autonomes Fahrzeug, das durch ein neuronales Netz gesteuert wird. Die Qualität des Fahrverhaltens ist hierbei natürlich abhängig von der Güte des neuronalen Netzes. Die Qualität des neuronalen Netzes wiederum ist abhängig von der Trainingszeit, der Lerndatei und diversen Netzwerkparametern. Da für dieses Projekt nur 3-4 Monate eingeplant waren, konnten keine optimalen Ergebnisse erzielt werden. Dies begründet sich durch mehrere Faktoren, wie unzureichende Beschleunigungswerte der Lerndatei und eine heuristische, zeitaufwendige Suche nach Netzwerkarchitekturen und Parametern.

Angesichts dieser Faktoren sind die erzielten Fahrverhalten beachtlich und machen die vielfältigen und umfassenden Einsatzmöglichkeiten neuronaler Netze auf dem Gebiet Neurocontrol offensichtlich. Der enorme Grad an Assoziationsmöglichkeiten verbunden mit sehr kurzen Reaktionszeiten ist eine gute Basis für weiterführende Untersuchungen und Projekte.

Anhang A. Beispieldatensätze

A.1 Beispiel eines IRDATA Programmes

```
00001,22100,2,35,0,'G_02_A01.tcp';
00002,22200,1,0;
00003,21400,64,4,0,132,1,-10510.238,
132,1,-568.611,
132,1,94.000;
00004,5000,80,64,4,0,0,0;
00005,2003,64,1,1.000;
00006,21400,64,4,0,132,1,-10757.183,
132,1,2962.763,
132,1,94.000;
00007,5000,80,64,4,0,0,0;
00008,2003,64,1,0.500;
00009,21400,64,4,0,132,1,-10974.129,
132,1,6065.187,
132,1,94.000;
00010,5000,80,64,4,0,0,0;
00011,2003,64,1,0.100;
00012,21400,64,4,0,132,1,-11025.750,
132,1,6803.380,
132,1,94.000;
00013,5000,80,64,4,0,0,0;
00014,21400,64,4,0,132,1,-11006.909,
132,1,7343.055,
132,1,88.000;
00015,5000,80,64,4,0,0,0;
00016,21400,64,4,0,132,1,-10991.901,
132,1,7772.795,
132,1,73.500;
00017,5000,80,64,4,0,0,0;
00018,21400,64,4,0,132,1,-10858.639,
132,1,8171.091,
132,1,51.000;
00019,5000,80,64,4,0,0,0;
00020,21400,64,4,0,132,1,-10504.041,
132,1,8448.138,
132,1,38.000;
00021,5000,80,64,4,0,0,0;
00022,21400,64,4,0,132,1,-9826.906,
132,1,8661.638,
132,1,17.500;
00023,5000,80,64,4,0,0,0;
00024,21400,64,4,0,132,1,-8963.392,
132,1,8767.669,
132,1,7.000;
00025,5000,80,64,4,0,0,0;
00026,21400,64,4,0,132,1,-8043.951,
132,1,8735.563,
132,1,-2.000;
00027,5000,80,64,4,0,0,0;
00028,2003,64,1,0.500;
00029,21400,64,4,0,132,1,-6015.186,
132,1,8664.717,
132,1,-2.000;
```

00030,5000,80,64,4,0,0,0;
00031,2003,64,1,1.000;
00032,21400,64,4,0,132,1,1500.240,
132,1,8402.279,
132,1,-2.000;
00033,5000,80,64,4,0,0,0;
00034,22210;
00035,22150;

Im nächsten Schritt werden die IRDATA-Programme in KISMET aufgerufen. Die so entstehenden Sensorwerte werden auf Datei geschrieben.

A.2 Beschreibung der Lerndatei

- Zeile 1: Name der Lerndatei.
- Zeile 2: verwendeter Fahrgrund.
- Zeile 3: Name des Teachfiles.
- Zeile 4: Erstellungsdatum.
- Zeile 5: Leerzeile.
- Zeile 6: Fahrzeugname.
- Zeile 7: Anzahl der Sensoren.
- Zeile 8: Weg-Einheit.
- Zeile 9: Zeit-Einheit.
- Zeile 10: Winkel-Einheit.
- Zeile 11: Taktung in 1/1000 sec.
- Zeile 12: X-Position.
- Zeile 13: Y-Position.
- Zeile 14: α -Position.
- Zeile 15-26: Sensor-Werte 1-12. --> Input
- Zeile 27: Aktuatorik (Geschwindigkeit). --> Input
- Zeile 28: Aktuatorik (Geschwindigkeits-Änderung). --> desired Output
- Zeile 29: Aktuatorik (Winkel-Änderung). --> desired Output
- Zeile 30-n: Wiederholung der Zeilen 12-29.

A.3 Beispiel einer Lerndatei.

G_02_A01.NNI
m_grund\$2.BIN
G_02_A01.tcp
Tue Jan 29 12:14:44 1991

ares1
12
1
1
1
700
-10757.182617
999.999878
-2962.762695
0.484167
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
0.179267
0.158017
0.367001
0.361287
0.500000
-10857.704102
999.999817
-4400.262695
0.374583
0.520217
0.499467
0.489083
0.499000
0.508267
1.000000
1.000000
1.000000
1.000000
0.143500
0.122250
0.411717
0.522358
0.500000
-10971.235352
999.999817
-6023.809570
0.374583
0.520217
0.499467
0.489083
0.499000
0.508267
1.000000

1.000000
1.000000
1.000000
0.143500
0.122250
0.465003
0.526643
0.500000
-11013.966797
999.999817
-7140.879883
0.332000
0.321667
0.306850
0.306200
0.320500
0.330283
1.000000
1.000000
1.000000
1.000000
0.126617
0.123733
0.096487
0.540984
0.520846
-11006.909180
999.999817
-7343.055176
0.289950
0.282900
0.270500
0.275667
0.292150
0.303217
1.000000
1.000000
1.000000
1.000000
0.122433
0.130550
0.057800
0.480656
0.512487
-10995.123047
999.999695
-7680.555176
0.209100
0.211267
0.209250
0.244567
0.272317
0.293733
1.000000
1.000000
1.000000
0.308933
0.102367
0.167333
0.096487
0.519344

0.563265
-10991.901367
999.999695
-7772.794922
0.190067
0.193900
0.194467
0.238400
0.269333
0.293900
1.000000
1.000000
1.000000
0.251667
0.098300
0.179567
0.026370
0.464941
0.517290
-10878.979492
999.999695
-8110.294922
0.130683
0.142100
0.160333
0.266383
0.324417
0.387983
1.000000
1.000000
1.000000
0.130917
0.106533
0.210367
0.101683
0.537656
0.605920
-10858.638672
999.999695
-8171.090820
0.122850
0.135367
0.157383
0.277500
0.342650
0.419333
1.000000
1.000000
1.000000
0.123433
0.112183
0.178333
0.018317
0.458317
0.519080
-10521.138672
999.999695
-8434.779297
0.089817
0.105183
0.144767

0.334633
0.440367
0.613817
1.000000
1.000000
1.000000
0.166750
0.229067
0.091033
0.122370
 0.552027
 0.568740
-10166.541016
 999.999695
 -8554.551758
0.116433
0.163467
0.297750
0.768133
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
0.269533
0.155567
0.065600
0.101108
 0.547454
 0.556765
-9826.997070
 999.999695
 -8661.608398
0.116450
0.163483
0.297800
0.768250
1.000000
1.000000
1.000000
1.000000
1.000000
0.269533
0.155550
0.065600
0.101720
 0.500306
 0.557109

A.4 Beispiel einer Konvergenzdatei

Global-Err.: 20.409 Cycle: 1 F.Beschl.: 961 F.Richtung: 162
Global-Err.: 16.797 Cycle: 2 F.Beschl.: 870 F.Richtung: 133
Global-Err.: 16.172 Cycle: 3 F.Beschl.: 838 F.Richtung: 135
Global-Err.: 15.855 Cycle: 4 F.Beschl.: 823 F.Richtung: 123
Global-Err.: 15.645 Cycle: 5 F.Beschl.: 816 F.Richtung: 118
Global-Err.: 15.489 Cycle: 6 F.Beschl.: 813 F.Richtung: 109
Global-Err.: 15.364 Cycle: 7 F.Beschl.: 806 F.Richtung: 95
Global-Err.: 15.260 Cycle: 8 F.Beschl.: 802 F.Richtung: 89
Global-Err.: 15.170 Cycle: 9 F.Beschl.: 800 F.Richtung: 83
Global-Err.: 15.093 Cycle: 10 F.Beschl.: 807 F.Richtung: 77
Global-Err.: 15.025 Cycle: 11 F.Beschl.: 809 F.Richtung: 75
Global-Err.: 14.964 Cycle: 12 F.Beschl.: 814 F.Richtung: 69
Global-Err.: 14.910 Cycle: 13 F.Beschl.: 815 F.Richtung: 62
Global-Err.: 14.861 Cycle: 14 F.Beschl.: 815 F.Richtung: 57
Global-Err.: 14.816 Cycle: 15 F.Beschl.: 815 F.Richtung: 55
Global-Err.: 14.774 Cycle: 16 F.Beschl.: 816 F.Richtung: 53
Global-Err.: 14.736 Cycle: 17 F.Beschl.: 818 F.Richtung: 48
Global-Err.: 14.700 Cycle: 18 F.Beschl.: 815 F.Richtung: 44
Global-Err.: 14.667 Cycle: 19 F.Beschl.: 813 F.Richtung: 43
Global-Err.: 14.635 Cycle: 20 F.Beschl.: 812 F.Richtung: 43
Global-Err.: 14.606 Cycle: 21 F.Beschl.: 813 F.Richtung: 42
Global-Err.: 14.577 Cycle: 22 F.Beschl.: 813 F.Richtung: 41
Global-Err.: 14.550 Cycle: 23 F.Beschl.: 814 F.Richtung: 41
Global-Err.: 14.524 Cycle: 24 F.Beschl.: 815 F.Richtung: 40
Global-Err.: 14.500 Cycle: 25 F.Beschl.: 811 F.Richtung: 40
Global-Err.: 14.475 Cycle: 26 F.Beschl.: 809 F.Richtung: 38
Global-Err.: 14.452 Cycle: 27 F.Beschl.: 810 F.Richtung: 37
Global-Err.: 14.430 Cycle: 28 F.Beschl.: 814 F.Richtung: 37
Global-Err.: 14.408 Cycle: 29 F.Beschl.: 818 F.Richtung: 37
Global-Err.: 14.386 Cycle: 30 F.Beschl.: 818 F.Richtung: 38
Global-Err.: 14.365 Cycle: 31 F.Beschl.: 820 F.Richtung: 38
Global-Err.: 14.345 Cycle: 32 F.Beschl.: 820 F.Richtung: 38
Global-Err.: 14.325 Cycle: 33 F.Beschl.: 820 F.Richtung: 37
Global-Err.: 14.305 Cycle: 34 F.Beschl.: 819 F.Richtung: 36
Global-Err.: 14.286 Cycle: 35 F.Beschl.: 818 F.Richtung: 35
Global-Err.: 14.267 Cycle: 36 F.Beschl.: 816 F.Richtung: 35
Global-Err.: 14.248 Cycle: 37 F.Beschl.: 816 F.Richtung: 35
Global-Err.: 14.230 Cycle: 38 F.Beschl.: 813 F.Richtung: 35
Global-Err.: 14.212 Cycle: 39 F.Beschl.: 809 F.Richtung: 35
Global-Err.: 14.194 Cycle: 40 F.Beschl.: 807 F.Richtung: 35
Global-Err.: 14.176 Cycle: 41 F.Beschl.: 805 F.Richtung: 35
Global-Err.: 14.159 Cycle: 42 F.Beschl.: 803 F.Richtung: 35
Global-Err.: 14.141 Cycle: 43 F.Beschl.: 802 F.Richtung: 35
Global-Err.: 14.124 Cycle: 44 F.Beschl.: 802 F.Richtung: 35
Global-Err.: 14.107 Cycle: 45 F.Beschl.: 805 F.Richtung: 35
Global-Err.: 14.090 Cycle: 46 F.Beschl.: 802 F.Richtung: 35
Global-Err.: 14.074 Cycle: 47 F.Beschl.: 798 F.Richtung: 35
Global-Err.: 14.058 Cycle: 48 F.Beschl.: 800 F.Richtung: 35
Global-Err.: 14.041 Cycle: 49 F.Beschl.: 801 F.Richtung: 35
Global-Err.: 14.025 Cycle: 50 F.Beschl.: 802 F.Richtung: 35
Global-Err.: 14.009 Cycle: 51 F.Beschl.: 805 F.Richtung: 35
Global-Err.: 13.994 Cycle: 52 F.Beschl.: 804 F.Richtung: 36
Global-Err.: 13.978 Cycle: 53 F.Beschl.: 802 F.Richtung: 35
Global-Err.: 13.962 Cycle: 54 F.Beschl.: 799 F.Richtung: 35
Global-Err.: 13.947 Cycle: 55 F.Beschl.: 797 F.Richtung: 35
Global-Err.: 13.932 Cycle: 56 F.Beschl.: 796 F.Richtung: 34
Global-Err.: 13.917 Cycle: 57 F.Beschl.: 795 F.Richtung: 34
Global-Err.: 13.902 Cycle: 58 F.Beschl.: 796 F.Richtung: 35

Global-Err.: 13.887 Cycle: 59 F.Beschl.: 798 F.Richtung: 35
Global-Err.: 13.872 Cycle: 60 F.Beschl.: 798 F.Richtung: 34
Global-Err.: 13.857 Cycle: 61 F.Beschl.: 797 F.Richtung: 34
Global-Err.: 13.843 Cycle: 62 F.Beschl.: 795 F.Richtung: 34
Global-Err.: 13.828 Cycle: 63 F.Beschl.: 795 F.Richtung: 33
Global-Err.: 13.814 Cycle: 64 F.Beschl.: 796 F.Richtung: 33
Global-Err.: 13.799 Cycle: 65 F.Beschl.: 795 F.Richtung: 32
Global-Err.: 13.785 Cycle: 66 F.Beschl.: 796 F.Richtung: 32
Global-Err.: 13.771 Cycle: 67 F.Beschl.: 795 F.Richtung: 31
Global-Err.: 13.757 Cycle: 68 F.Beschl.: 792 F.Richtung: 31
Global-Err.: 13.743 Cycle: 69 F.Beschl.: 793 F.Richtung: 30
Global-Err.: 13.729 Cycle: 70 F.Beschl.: 793 F.Richtung: 29
Global-Err.: 13.715 Cycle: 71 F.Beschl.: 795 F.Richtung: 29
Global-Err.: 13.701 Cycle: 72 F.Beschl.: 795 F.Richtung: 29
Global-Err.: 13.688 Cycle: 73 F.Beschl.: 794 F.Richtung: 29
Global-Err.: 13.674 Cycle: 74 F.Beschl.: 793 F.Richtung: 29
Global-Err.: 13.661 Cycle: 75 F.Beschl.: 791 F.Richtung: 30
Global-Err.: 13.648 Cycle: 76 F.Beschl.: 790 F.Richtung: 30
Global-Err.: 13.635 Cycle: 77 F.Beschl.: 788 F.Richtung: 30
Global-Err.: 13.622 Cycle: 78 F.Beschl.: 790 F.Richtung: 31
Global-Err.: 13.609 Cycle: 79 F.Beschl.: 788 F.Richtung: 31
Global-Err.: 13.597 Cycle: 80 F.Beschl.: 786 F.Richtung: 31
Global-Err.: 13.584 Cycle: 81 F.Beschl.: 786 F.Richtung: 31
Global-Err.: 13.572 Cycle: 82 F.Beschl.: 786 F.Richtung: 31
Global-Err.: 13.560 Cycle: 83 F.Beschl.: 783 F.Richtung: 31
Global-Err.: 13.548 Cycle: 84 F.Beschl.: 782 F.Richtung: 31
Global-Err.: 13.536 Cycle: 85 F.Beschl.: 782 F.Richtung: 32
Global-Err.: 13.524 Cycle: 86 F.Beschl.: 780 F.Richtung: 31
Global-Err.: 13.513 Cycle: 87 F.Beschl.: 779 F.Richtung: 32
Global-Err.: 13.501 Cycle: 88 F.Beschl.: 780 F.Richtung: 34
Global-Err.: 13.490 Cycle: 89 F.Beschl.: 779 F.Richtung: 34
Global-Err.: 13.479 Cycle: 90 F.Beschl.: 779 F.Richtung: 36
Global-Err.: 13.468 Cycle: 91 F.Beschl.: 779 F.Richtung: 38
Global-Err.: 13.457 Cycle: 92 F.Beschl.: 776 F.Richtung: 37
Global-Err.: 13.447 Cycle: 93 F.Beschl.: 776 F.Richtung: 41
Global-Err.: 13.436 Cycle: 94 F.Beschl.: 776 F.Richtung: 42
Global-Err.: 13.426 Cycle: 95 F.Beschl.: 774 F.Richtung: 42
Global-Err.: 13.415 Cycle: 96 F.Beschl.: 773 F.Richtung: 43
Global-Err.: 13.405 Cycle: 97 F.Beschl.: 772 F.Richtung: 44
Global-Err.: 13.395 Cycle: 98 F.Beschl.: 770 F.Richtung: 44
Global-Err.: 13.385 Cycle: 99 F.Beschl.: 773 F.Richtung: 45
Global-Err.: 13.376 Cycle: 100 F.Beschl.: 773 F.Richtung: 46
Global-Err.: 13.366 Cycle: 101 F.Beschl.: 774 F.Richtung: 46
Global-Err.: 13.357 Cycle: 102 F.Beschl.: 774 F.Richtung: 46
Global-Err.: 13.347 Cycle: 103 F.Beschl.: 772 F.Richtung: 47
Global-Err.: 13.338 Cycle: 104 F.Beschl.: 771 F.Richtung: 48
Global-Err.: 13.329 Cycle: 105 F.Beschl.: 772 F.Richtung: 49
Global-Err.: 13.320 Cycle: 106 F.Beschl.: 771 F.Richtung: 48
Global-Err.: 13.311 Cycle: 107 F.Beschl.: 769 F.Richtung: 48
Global-Err.: 13.303 Cycle: 108 F.Beschl.: 767 F.Richtung: 47
Global-Err.: 13.294 Cycle: 109 F.Beschl.: 764 F.Richtung: 49
Global-Err.: 13.285 Cycle: 110 F.Beschl.: 765 F.Richtung: 49
Global-Err.: 13.277 Cycle: 111 F.Beschl.: 765 F.Richtung: 50
Global-Err.: 13.269 Cycle: 112 F.Beschl.: 766 F.Richtung: 53
Global-Err.: 13.261 Cycle: 113 F.Beschl.: 767 F.Richtung: 54
Global-Err.: 13.253 Cycle: 114 F.Beschl.: 764 F.Richtung: 55
Global-Err.: 13.245 Cycle: 115 F.Beschl.: 764 F.Richtung: 55
Global-Err.: 13.237 Cycle: 116 F.Beschl.: 765 F.Richtung: 56
Global-Err.: 13.229 Cycle: 117 F.Beschl.: 764 F.Richtung: 57
Global-Err.: 13.221 Cycle: 118 F.Beschl.: 762 F.Richtung: 58
Global-Err.: 13.214 Cycle: 119 F.Beschl.: 762 F.Richtung: 58

Global-Err.: 13.206 Cycle: 120 F.Beschl.: 764 F.Richtung: 58
Global-Err.: 13.199 Cycle: 121 F.Beschl.: 763 F.Richtung: 60
Global-Err.: 13.192 Cycle: 122 F.Beschl.: 764 F.Richtung: 60
Global-Err.: 13.184 Cycle: 123 F.Beschl.: 764 F.Richtung: 59
Global-Err.: 13.177 Cycle: 124 F.Beschl.: 764 F.Richtung: 60
Global-Err.: 13.170 Cycle: 125 F.Beschl.: 763 F.Richtung: 60
Global-Err.: 13.163 Cycle: 126 F.Beschl.: 764 F.Richtung: 60
Global-Err.: 13.156 Cycle: 127 F.Beschl.: 763 F.Richtung: 60
Global-Err.: 13.150 Cycle: 128 F.Beschl.: 762 F.Richtung: 60
Global-Err.: 13.143 Cycle: 129 F.Beschl.: 762 F.Richtung: 61
Global-Err.: 13.136 Cycle: 130 F.Beschl.: 763 F.Richtung: 61
Global-Err.: 13.129 Cycle: 131 F.Beschl.: 765 F.Richtung: 62
Global-Err.: 13.123 Cycle: 132 F.Beschl.: 765 F.Richtung: 63
Global-Err.: 13.116 Cycle: 133 F.Beschl.: 765 F.Richtung: 63
Global-Err.: 13.110 Cycle: 134 F.Beschl.: 764 F.Richtung: 64
Global-Err.: 13.104 Cycle: 135 F.Beschl.: 765 F.Richtung: 65
Global-Err.: 13.097 Cycle: 136 F.Beschl.: 766 F.Richtung: 65
Global-Err.: 13.091 Cycle: 137 F.Beschl.: 765 F.Richtung: 66
Global-Err.: 13.085 Cycle: 138 F.Beschl.: 765 F.Richtung: 66
Global-Err.: 13.079 Cycle: 139 F.Beschl.: 764 F.Richtung: 66
Global-Err.: 13.073 Cycle: 140 F.Beschl.: 763 F.Richtung: 66
Global-Err.: 13.067 Cycle: 141 F.Beschl.: 762 F.Richtung: 66
Global-Err.: 13.061 Cycle: 142 F.Beschl.: 761 F.Richtung: 66
Global-Err.: 13.055 Cycle: 143 F.Beschl.: 761 F.Richtung: 67
Global-Err.: 13.049 Cycle: 144 F.Beschl.: 761 F.Richtung: 67
Global-Err.: 13.043 Cycle: 145 F.Beschl.: 761 F.Richtung: 67
Global-Err.: 13.037 Cycle: 146 F.Beschl.: 761 F.Richtung: 67
Global-Err.: 13.031 Cycle: 147 F.Beschl.: 762 F.Richtung: 67
Global-Err.: 13.026 Cycle: 148 F.Beschl.: 761 F.Richtung: 67
Global-Err.: 13.020 Cycle: 149 F.Beschl.: 762 F.Richtung: 67
Global-Err.: 13.014 Cycle: 150 F.Beschl.: 762 F.Richtung: 68
Global-Err.: 13.009 Cycle: 151 F.Beschl.: 762 F.Richtung: 68
Global-Err.: 13.003 Cycle: 152 F.Beschl.: 762 F.Richtung: 70
Global-Err.: 12.998 Cycle: 153 F.Beschl.: 762 F.Richtung: 69
Global-Err.: 12.992 Cycle: 154 F.Beschl.: 762 F.Richtung: 69
Global-Err.: 12.987 Cycle: 155 F.Beschl.: 762 F.Richtung: 69
Global-Err.: 12.982 Cycle: 156 F.Beschl.: 763 F.Richtung: 69
Global-Err.: 12.976 Cycle: 157 F.Beschl.: 763 F.Richtung: 69
Global-Err.: 12.971 Cycle: 158 F.Beschl.: 762 F.Richtung: 69
Global-Err.: 12.966 Cycle: 159 F.Beschl.: 760 F.Richtung: 69
Global-Err.: 12.961 Cycle: 160 F.Beschl.: 762 F.Richtung: 69
Global-Err.: 12.956 Cycle: 161 F.Beschl.: 760 F.Richtung: 70
Global-Err.: 12.950 Cycle: 162 F.Beschl.: 760 F.Richtung: 70
Global-Err.: 12.945 Cycle: 163 F.Beschl.: 760 F.Richtung: 70
Global-Err.: 12.940 Cycle: 164 F.Beschl.: 759 F.Richtung: 69
Global-Err.: 12.935 Cycle: 165 F.Beschl.: 758 F.Richtung: 69
Global-Err.: 12.930 Cycle: 166 F.Beschl.: 757 F.Richtung: 69
Global-Err.: 12.925 Cycle: 167 F.Beschl.: 756 F.Richtung: 69
Global-Err.: 12.920 Cycle: 168 F.Beschl.: 758 F.Richtung: 68
Global-Err.: 12.916 Cycle: 169 F.Beschl.: 758 F.Richtung: 67
Global-Err.: 12.911 Cycle: 170 F.Beschl.: 758 F.Richtung: 66
Global-Err.: 12.906 Cycle: 171 F.Beschl.: 757 F.Richtung: 66
Global-Err.: 12.901 Cycle: 172 F.Beschl.: 757 F.Richtung: 66
Global-Err.: 12.896 Cycle: 173 F.Beschl.: 756 F.Richtung: 66
Global-Err.: 12.891 Cycle: 174 F.Beschl.: 757 F.Richtung: 66
Global-Err.: 12.887 Cycle: 175 F.Beschl.: 756 F.Richtung: 67
Global-Err.: 12.882 Cycle: 176 F.Beschl.: 755 F.Richtung: 67
Global-Err.: 12.877 Cycle: 177 F.Beschl.: 756 F.Richtung: 67
Global-Err.: 12.873 Cycle: 178 F.Beschl.: 757 F.Richtung: 67
Global-Err.: 12.868 Cycle: 179 F.Beschl.: 757 F.Richtung: 66
Global-Err.: 12.864 Cycle: 180 F.Beschl.: 757 F.Richtung: 65

Global-Err.: 12.859 Cycle: 181 F.Beschl.: 756 F.Richtung: 64
Global-Err.: 12.854 Cycle: 182 F.Beschl.: 758 F.Richtung: 63
Global-Err.: 12.850 Cycle: 183 F.Beschl.: 758 F.Richtung: 63
Global-Err.: 12.845 Cycle: 184 F.Beschl.: 757 F.Richtung: 63
Global-Err.: 12.841 Cycle: 185 F.Beschl.: 757 F.Richtung: 63
Global-Err.: 12.836 Cycle: 186 F.Beschl.: 756 F.Richtung: 63
Global-Err.: 12.832 Cycle: 187 F.Beschl.: 756 F.Richtung: 63
Global-Err.: 12.828 Cycle: 188 F.Beschl.: 756 F.Richtung: 63
Global-Err.: 12.823 Cycle: 189 F.Beschl.: 757 F.Richtung: 63
Global-Err.: 12.819 Cycle: 190 F.Beschl.: 755 F.Richtung: 63
Global-Err.: 12.815 Cycle: 191 F.Beschl.: 755 F.Richtung: 63
Global-Err.: 12.810 Cycle: 192 F.Beschl.: 756 F.Richtung: 63
Global-Err.: 12.806 Cycle: 193 F.Beschl.: 756 F.Richtung: 63
Global-Err.: 12.802 Cycle: 194 F.Beschl.: 755 F.Richtung: 63
Global-Err.: 12.797 Cycle: 195 F.Beschl.: 755 F.Richtung: 63
Global-Err.: 12.793 Cycle: 196 F.Beschl.: 754 F.Richtung: 62
Global-Err.: 12.789 Cycle: 197 F.Beschl.: 754 F.Richtung: 62
Global-Err.: 12.785 Cycle: 198 F.Beschl.: 754 F.Richtung: 62
Global-Err.: 12.781 Cycle: 199 F.Beschl.: 755 F.Richtung: 62
Global-Err.: 12.776 Cycle: 200 F.Beschl.: 755 F.Richtung: 61
Global-Err.: 12.772 Cycle: 201 F.Beschl.: 756 F.Richtung: 60
Global-Err.: 12.768 Cycle: 202 F.Beschl.: 756 F.Richtung: 60
Global-Err.: 12.764 Cycle: 203 F.Beschl.: 754 F.Richtung: 60
Global-Err.: 12.760 Cycle: 204 F.Beschl.: 753 F.Richtung: 60
Global-Err.: 12.756 Cycle: 205 F.Beschl.: 753 F.Richtung: 60
Global-Err.: 12.752 Cycle: 206 F.Beschl.: 753 F.Richtung: 60
Global-Err.: 12.748 Cycle: 207 F.Beschl.: 753 F.Richtung: 60
Global-Err.: 12.744 Cycle: 208 F.Beschl.: 751 F.Richtung: 60
Global-Err.: 12.740 Cycle: 209 F.Beschl.: 750 F.Richtung: 60
Global-Err.: 12.736 Cycle: 210 F.Beschl.: 751 F.Richtung: 60
Global-Err.: 12.732 Cycle: 211 F.Beschl.: 751 F.Richtung: 59
Global-Err.: 12.728 Cycle: 212 F.Beschl.: 750 F.Richtung: 59
Global-Err.: 12.724 Cycle: 213 F.Beschl.: 750 F.Richtung: 59
Global-Err.: 12.720 Cycle: 214 F.Beschl.: 751 F.Richtung: 59
Global-Err.: 12.716 Cycle: 215 F.Beschl.: 750 F.Richtung: 60
Global-Err.: 12.712 Cycle: 216 F.Beschl.: 750 F.Richtung: 61
Global-Err.: 12.709 Cycle: 217 F.Beschl.: 750 F.Richtung: 61
Global-Err.: 12.705 Cycle: 218 F.Beschl.: 750 F.Richtung: 62
Global-Err.: 12.701 Cycle: 219 F.Beschl.: 747 F.Richtung: 62
Global-Err.: 12.697 Cycle: 220 F.Beschl.: 747 F.Richtung: 62
Global-Err.: 12.693 Cycle: 221 F.Beschl.: 747 F.Richtung: 61
Global-Err.: 12.690 Cycle: 222 F.Beschl.: 747 F.Richtung: 61
Global-Err.: 12.686 Cycle: 223 F.Beschl.: 747 F.Richtung: 61
Global-Err.: 12.682 Cycle: 224 F.Beschl.: 747 F.Richtung: 61
Global-Err.: 12.678 Cycle: 225 F.Beschl.: 747 F.Richtung: 60
Global-Err.: 12.675 Cycle: 226 F.Beschl.: 747 F.Richtung: 58
Global-Err.: 12.671 Cycle: 227 F.Beschl.: 747 F.Richtung: 59
Global-Err.: 12.667 Cycle: 228 F.Beschl.: 747 F.Richtung: 59
Global-Err.: 12.664 Cycle: 229 F.Beschl.: 747 F.Richtung: 59
Global-Err.: 12.660 Cycle: 230 F.Beschl.: 748 F.Richtung: 59
Global-Err.: 12.656 Cycle: 231 F.Beschl.: 749 F.Richtung: 59
Global-Err.: 12.653 Cycle: 232 F.Beschl.: 749 F.Richtung: 59
Global-Err.: 12.649 Cycle: 233 F.Beschl.: 750 F.Richtung: 60
Global-Err.: 12.646 Cycle: 234 F.Beschl.: 750 F.Richtung: 59
Global-Err.: 12.642 Cycle: 235 F.Beschl.: 750 F.Richtung: 59
Global-Err.: 12.639 Cycle: 236 F.Beschl.: 748 F.Richtung: 59
Global-Err.: 12.635 Cycle: 237 F.Beschl.: 748 F.Richtung: 59
Global-Err.: 12.631 Cycle: 238 F.Beschl.: 747 F.Richtung: 59
Global-Err.: 12.628 Cycle: 239 F.Beschl.: 748 F.Richtung: 58
Global-Err.: 12.624 Cycle: 240 F.Beschl.: 748 F.Richtung: 58

A.5 Beispiel einer Fehlerdatei

2 ->Beschleunigung
5 ->Beschleunigung
6 ->Beschleunigung
8 ->Beschleunigung
11 ->Richtung
19 ->Beschleunigung
20 ->Beschleunigung
22 ->Beschleunigung
23 ->Beschleunigung
29 ->Beschleunigung
43 ->Beschleunigung
47 ->Beschleunigung
48 ->Beschleunigung
58 ->Beschleunigung
59 ->Beschleunigung
60 ->Beschleunigung
62 ->Beschleunigung
63 ->Beschleunigung
69 ->Beschleunigung
70 ->Beschleunigung
72 ->Beschleunigung
74 ->Beschleunigung
76 ->Beschleunigung
77 ->Beschleunigung
79 ->Beschleunigung
81 ->Beschleunigung
82 ->Beschleunigung
83 ->Richtung
84 ->Richtung
85 ->Richtung
89 ->Beschleunigung
93 ->Beschleunigung
99 ->Beschleunigung
108 ->Beschleunigung
110 ->Beschleunigung
113 ->Beschleunigung
117 ->Beschleunigung
123 ->Beschleunigung
129 ->Beschleunigung
133 ->Beschleunigung
138 ->Beschleunigung
140 ->Beschleunigung
143 ->Beschleunigung
147 ->Beschleunigung
150 ->Beschleunigung
153 ->Beschleunigung
155 ->Beschleunigung
161 ->Beschleunigung
162 ->Beschleunigung
172 ->Beschleunigung
176 ->Beschleunigung
177 ->Beschleunigung
186 ->Beschleunigung
189 ->Beschleunigung
200 ->Beschleunigung
202 ->Beschleunigung
204 ->Beschleunigung
205 ->Beschleunigung

206 ->Beschleunigung
208 ->Beschleunigung
209 ->Beschleunigung
210 ->Beschleunigung
216 ->Beschleunigung
220 ->Beschleunigung
221 ->Beschleunigung
223 ->Beschleunigung
224 ->Beschleunigung
226 ->Beschleunigung
227 ->Beschleunigung
230 ->Beschleunigung
231 ->Beschleunigung
240 ->Beschleunigung
243 ->Beschleunigung
246 ->Beschleunigung
258 ->Beschleunigung
260 ->Beschleunigung
261 ->Beschleunigung
262 ->Beschleunigung
268 ->Beschleunigung
270 ->Beschleunigung
275 ->Beschleunigung
277 ->Beschleunigung
278 ->Beschleunigung
280 ->Beschleunigung
281 ->Beschleunigung
282 ->Beschleunigung
284 ->Beschleunigung
286 ->Beschleunigung
295 ->Beschleunigung
297 ->Beschleunigung
300 ->Beschleunigung
310 ->Beschleunigung
312 ->Beschleunigung
314 ->Beschleunigung
315 ->Beschleunigung
318 ->Beschleunigung
319 ->Beschleunigung
325 ->Beschleunigung
327 ->Beschleunigung
331 ->Beschleunigung
334 ->Richtung
337 ->Beschleunigung
339 ->Beschleunigung
341 ->Beschleunigung
343 ->Beschleunigung
344 ->Beschleunigung
351 ->Beschleunigung
356 ->Beschleunigung
359 ->Beschleunigung
360 ->Beschleunigung
361 ->Beschleunigung
362 ->Beschleunigung
368 ->Beschleunigung
378 ->Beschleunigung
379 ->Beschleunigung
381 ->Beschleunigung
383 ->Beschleunigung
385 ->Beschleunigung
387 ->Beschleunigung

388 ->Beschleunigung
393 ->Beschleunigung
395 ->Beschleunigung
397 ->Beschleunigung
403 ->Beschleunigung
413 ->Beschleunigung
415 ->Beschleunigung
423 ->Beschleunigung
425 ->Beschleunigung
431 ->Beschleunigung
437 ->Beschleunigung
445 ->Beschleunigung
447 ->Beschleunigung
448 ->Beschleunigung
451 ->Beschleunigung
454 ->Beschleunigung
461 ->Beschleunigung
464 ->Richtung
466 ->Beschleunigung
467 ->Beschleunigung
468 ->Beschleunigung
474 ->Beschleunigung
475 ->Beschleunigung
481 ->Beschleunigung
483 ->Beschleunigung
490 ->Richtung
498 ->Beschleunigung
500 ->Beschleunigung
501 ->Beschleunigung
513 ->Beschleunigung
517 ->Beschleunigung
518 ->Beschleunigung
521 ->Beschleunigung
522 ->Beschleunigung
532 ->Beschleunigung
534 ->Beschleunigung
536 ->Beschleunigung
537 ->Beschleunigung
539 ->Beschleunigung
540 ->Beschleunigung
549 ->Beschleunigung
552 ->Beschleunigung
555 ->Beschleunigung
557 ->Beschleunigung
563 ->Beschleunigung
565 ->Beschleunigung
567 ->Beschleunigung
568 ->Beschleunigung
578 ->Beschleunigung
584 ->Beschleunigung
586 ->Beschleunigung
588 ->Beschleunigung
590 ->Beschleunigung
598 ->Beschleunigung
600 ->Beschleunigung
603 ->Beschleunigung
604 ->Beschleunigung
606 ->Beschleunigung
607 ->Beschleunigung
615 ->Richtung
616 ->Beschleunigung

621 ->Beschleunigung
624 ->Beschleunigung
625 ->Beschleunigung
631 ->Beschleunigung
633 ->Beschleunigung
634 ->Beschleunigung
637 ->Beschleunigung
648 ->Beschleunigung
653 ->Beschleunigung
654 ->Beschleunigung
659 ->Beschleunigung
662 ->Beschleunigung
667 ->Beschleunigung
668 ->Beschleunigung
669 ->Beschleunigung
670 ->Beschleunigung
672 ->Beschleunigung
673 ->Beschleunigung
675 ->Beschleunigung
676 ->Beschleunigung
682 ->Beschleunigung
688 ->Beschleunigung
690 ->Beschleunigung
691 ->Beschleunigung
693 ->Beschleunigung
702 ->Beschleunigung
705 ->Beschleunigung
708 ->Beschleunigung
718 ->Beschleunigung
719 ->Beschleunigung
720 ->Beschleunigung
721 ->Beschleunigung
723 ->Beschleunigung
724 ->Beschleunigung
732 ->Beschleunigung
737 ->Beschleunigung
739 ->Beschleunigung
744 ->Richtung
752 ->Beschleunigung
754 ->Beschleunigung
758 ->Beschleunigung
760 ->Beschleunigung
761 ->Beschleunigung
767 ->Beschleunigung
774 ->Beschleunigung
781 ->Beschleunigung
784 ->Beschleunigung
786 ->Beschleunigung
789 ->Beschleunigung
791 ->Beschleunigung
794 ->Richtung
799 ->Beschleunigung
803 ->Beschleunigung
805 ->Beschleunigung
806 ->Beschleunigung
812 ->Beschleunigung
816 ->Beschleunigung
817 ->Beschleunigung
819 ->Beschleunigung
820 ->Beschleunigung

A.6 Beispiel einer Netzwerkdatei

```
kfk_n_70.nnd ->Netzwerk Name
9687162      ->Lernschritte
4038         ->Durchlaeufer durch Lerndatei
658          ->Fehler im letzten Durchlauf
9.582950     ->Globaler Fehler
1            ->Anzahl Zwischenebenen
13           ->Anzahl Knoten Eingabeebene
24           ->Anzahl Knoten Zwischenebene[0]
2            ->Anzahl Knoten Ausgabebene
0.900000     ->Lernrate
0.600000     ->Momentum
16.565834   ->Gewicht [0][ 0][ 0] (Eingangsmatrix)
-5.324837   ->Gewicht [0][ 0][ 1] (Eingangsmatrix)
-1.756895   ->Gewicht [0][ 0][ 2] (Eingangsmatrix)
0.254240    ->Gewicht [0][ 0][ 3] (Eingangsmatrix)
-2.442225   ->Gewicht [0][ 0][ 4] (Eingangsmatrix)
0.310319    ->Gewicht [0][ 0][ 5] (Eingangsmatrix)
-1.805565   ->Gewicht [0][ 0][ 6] (Eingangsmatrix)
2.629442    ->Gewicht [0][ 0][ 7] (Eingangsmatrix)
5.503578    ->Gewicht [0][ 0][ 8] (Eingangsmatrix)
7.488010    ->Gewicht [0][ 0][ 9] (Eingangsmatrix)
3.606339    ->Gewicht [0][ 0][10] (Eingangsmatrix)
1.865231    ->Gewicht [0][ 0][11] (Eingangsmatrix)
11.964598   ->Gewicht [0][ 0][12] (Eingangsmatrix)
-2.573597   ->Gewicht [0][ 0][13] (Eingangsmatrix)
-0.827674   ->Gewicht [0][ 0][14] (Eingangsmatrix)
-0.544002   ->Gewicht [0][ 0][15] (Eingangsmatrix)
0.515900    ->Gewicht [0][ 0][16] (Eingangsmatrix)
4.198334    ->Gewicht [0][ 0][17] (Eingangsmatrix)
0.326035    ->Gewicht [0][ 0][18] (Eingangsmatrix)
-3.936324   ->Gewicht [0][ 0][19] (Eingangsmatrix)
17.166435   ->Gewicht [0][ 0][20] (Eingangsmatrix)
1.261845    ->Gewicht [0][ 0][21] (Eingangsmatrix)
2.438602    ->Gewicht [0][ 0][22] (Eingangsmatrix)
-8.391750   ->Gewicht [0][ 0][23] (Eingangsmatrix)
4.904677    ->Gewicht [0][ 1][ 0] (Eingangsmatrix)
11.669352   ->Gewicht [0][ 1][ 1] (Eingangsmatrix)
2.351252    ->Gewicht [0][ 1][ 2] (Eingangsmatrix)
-0.801679   ->Gewicht [0][ 1][ 3] (Eingangsmatrix)
-6.394163   ->Gewicht [0][ 1][ 4] (Eingangsmatrix)
-1.184166   ->Gewicht [0][ 1][ 5] (Eingangsmatrix)
-3.717960   ->Gewicht [0][ 1][ 6] (Eingangsmatrix)
5.419369    ->Gewicht [0][ 1][ 7] (Eingangsmatrix)
2.080274    ->Gewicht [0][ 1][ 8] (Eingangsmatrix)
-1.866578   ->Gewicht [0][ 1][ 9] (Eingangsmatrix)
-4.393514   ->Gewicht [0][ 1][10] (Eingangsmatrix)
-2.091750   ->Gewicht [0][ 1][11] (Eingangsmatrix)
-14.443846  ->Gewicht [0][ 1][12] (Eingangsmatrix)
-2.192763   ->Gewicht [0][ 1][13] (Eingangsmatrix)
-2.023238   ->Gewicht [0][ 1][14] (Eingangsmatrix)
2.460438    ->Gewicht [0][ 1][15] (Eingangsmatrix)
-0.722865   ->Gewicht [0][ 1][16] (Eingangsmatrix)
-0.812266   ->Gewicht [0][ 1][17] (Eingangsmatrix)
-2.410997   ->Gewicht [0][ 1][18] (Eingangsmatrix)
1.235273    ->Gewicht [0][ 1][19] (Eingangsmatrix)
4.296744    ->Gewicht [0][ 1][20] (Eingangsmatrix)
5.808286    ->Gewicht [0][ 1][21] (Eingangsmatrix)
-1.725964   ->Gewicht [0][ 1][22] (Eingangsmatrix)
```


-1.613349 ->Gewicht [0][1][23] (Eingangsmatrix)
10.682789 ->Gewicht [0][2][0] (Eingangsmatrix)
4.400877 ->Gewicht [0][2][1] (Eingangsmatrix)
0.082349 ->Gewicht [0][2][2] (Eingangsmatrix)
-1.238097 ->Gewicht [0][2][3] (Eingangsmatrix)
2.910676 ->Gewicht [0][2][4] (Eingangsmatrix)
-4.663059 ->Gewicht [0][2][5] (Eingangsmatrix)
3.187781 ->Gewicht [0][2][6] (Eingangsmatrix)
3.774009 ->Gewicht [0][2][7] (Eingangsmatrix)
-1.544590 ->Gewicht [0][2][8] (Eingangsmatrix)
-3.023214 ->Gewicht [0][2][9] (Eingangsmatrix)
-4.807230 ->Gewicht [0][2][10] (Eingangsmatrix)
0.506814 ->Gewicht [0][2][11] (Eingangsmatrix)
1.519198 ->Gewicht [0][2][12] (Eingangsmatrix)
0.440285 ->Gewicht [0][2][13] (Eingangsmatrix)
3.575520 ->Gewicht [0][2][14] (Eingangsmatrix)
-0.065912 ->Gewicht [0][2][15] (Eingangsmatrix)
2.160000 ->Gewicht [0][2][16] (Eingangsmatrix)
3.447945 ->Gewicht [0][2][17] (Eingangsmatrix)
2.044131 ->Gewicht [0][2][18] (Eingangsmatrix)
-4.493224 ->Gewicht [0][2][19] (Eingangsmatrix)
-0.557790 ->Gewicht [0][2][20] (Eingangsmatrix)
1.698615 ->Gewicht [0][2][21] (Eingangsmatrix)
-1.096956 ->Gewicht [0][2][22] (Eingangsmatrix)
2.529954 ->Gewicht [0][2][23] (Eingangsmatrix)
12.584558 ->Gewicht [0][3][0] (Eingangsmatrix)
-1.337516 ->Gewicht [0][3][1] (Eingangsmatrix)
1.070201 ->Gewicht [0][3][2] (Eingangsmatrix)
-0.220601 ->Gewicht [0][3][3] (Eingangsmatrix)
1.685132 ->Gewicht [0][3][4] (Eingangsmatrix)
-4.304408 ->Gewicht [0][3][5] (Eingangsmatrix)
1.640245 ->Gewicht [0][3][6] (Eingangsmatrix)
-1.866638 ->Gewicht [0][3][7] (Eingangsmatrix)
-5.155675 ->Gewicht [0][3][8] (Eingangsmatrix)
6.829137 ->Gewicht [0][3][9] (Eingangsmatrix)
-0.960420 ->Gewicht [0][3][10] (Eingangsmatrix)
-1.649903 ->Gewicht [0][3][11] (Eingangsmatrix)
-4.322678 ->Gewicht [0][3][12] (Eingangsmatrix)
-2.874646 ->Gewicht [0][3][13] (Eingangsmatrix)
4.508694 ->Gewicht [0][3][14] (Eingangsmatrix)
-1.224559 ->Gewicht [0][3][15] (Eingangsmatrix)
-9.170248 ->Gewicht [0][3][16] (Eingangsmatrix)
3.323212 ->Gewicht [0][3][17] (Eingangsmatrix)
-0.873221 ->Gewicht [0][3][18] (Eingangsmatrix)
-5.018178 ->Gewicht [0][3][19] (Eingangsmatrix)
6.541056 ->Gewicht [0][3][20] (Eingangsmatrix)
-7.503230 ->Gewicht [0][3][21] (Eingangsmatrix)
-5.125054 ->Gewicht [0][3][22] (Eingangsmatrix)
-1.154517 ->Gewicht [0][3][23] (Eingangsmatrix)
-3.809010 ->Gewicht [0][4][0] (Eingangsmatrix)
-0.045551 ->Gewicht [0][4][1] (Eingangsmatrix)
4.557229 ->Gewicht [0][4][2] (Eingangsmatrix)
7.510928 ->Gewicht [0][4][3] (Eingangsmatrix)
1.304737 ->Gewicht [0][4][4] (Eingangsmatrix)
-1.985391 ->Gewicht [0][4][5] (Eingangsmatrix)
-1.502891 ->Gewicht [0][4][6] (Eingangsmatrix)
-2.558965 ->Gewicht [0][4][7] (Eingangsmatrix)
-4.254581 ->Gewicht [0][4][8] (Eingangsmatrix)
-0.160860 ->Gewicht [0][4][9] (Eingangsmatrix)
-4.744864 ->Gewicht [0][4][10] (Eingangsmatrix)
2.935480 ->Gewicht [0][4][11] (Eingangsmatrix)

-1.264368 ->Gewicht [0][4][12] (Eingangsmatrix)
-0.582798 ->Gewicht [0][4][13] (Eingangsmatrix)
-0.537755 ->Gewicht [0][4][14] (Eingangsmatrix)
1.318221 ->Gewicht [0][4][15] (Eingangsmatrix)
-1.875869 ->Gewicht [0][4][16] (Eingangsmatrix)
-3.691969 ->Gewicht [0][4][17] (Eingangsmatrix)
-4.287775 ->Gewicht [0][4][18] (Eingangsmatrix)
-0.607503 ->Gewicht [0][4][19] (Eingangsmatrix)
4.008976 ->Gewicht [0][4][20] (Eingangsmatrix)
0.723672 ->Gewicht [0][4][21] (Eingangsmatrix)
-2.209572 ->Gewicht [0][4][22] (Eingangsmatrix)
2.468240 ->Gewicht [0][4][23] (Eingangsmatrix)
13.546119 ->Gewicht [0][5][0] (Eingangsmatrix)
-13.254043 ->Gewicht [0][5][1] (Eingangsmatrix)
9.933015 ->Gewicht [0][5][2] (Eingangsmatrix)
-1.307340 ->Gewicht [0][5][3] (Eingangsmatrix)
11.212589 ->Gewicht [0][5][4] (Eingangsmatrix)
-2.219288 ->Gewicht [0][5][5] (Eingangsmatrix)
1.657956 ->Gewicht [0][5][6] (Eingangsmatrix)
2.388556 ->Gewicht [0][5][7] (Eingangsmatrix)
-5.387863 ->Gewicht [0][5][8] (Eingangsmatrix)
3.597967 ->Gewicht [0][5][9] (Eingangsmatrix)
2.483467 ->Gewicht [0][5][10] (Eingangsmatrix)
-0.428317 ->Gewicht [0][5][11] (Eingangsmatrix)
2.405954 ->Gewicht [0][5][12] (Eingangsmatrix)
1.199430 ->Gewicht [0][5][13] (Eingangsmatrix)
0.356861 ->Gewicht [0][5][14] (Eingangsmatrix)
-2.686191 ->Gewicht [0][5][15] (Eingangsmatrix)
-0.675770 ->Gewicht [0][5][16] (Eingangsmatrix)
-4.203692 ->Gewicht [0][5][17] (Eingangsmatrix)
1.636974 ->Gewicht [0][5][18] (Eingangsmatrix)
1.620267 ->Gewicht [0][5][19] (Eingangsmatrix)
-20.061966 ->Gewicht [0][5][20] (Eingangsmatrix)
11.325229 ->Gewicht [0][5][21] (Eingangsmatrix)
-7.483869 ->Gewicht [0][5][22] (Eingangsmatrix)
1.494649 ->Gewicht [0][5][23] (Eingangsmatrix)
6.564773 ->Gewicht [0][6][0] (Eingangsmatrix)
2.723815 ->Gewicht [0][6][1] (Eingangsmatrix)
-10.945279 ->Gewicht [0][6][2] (Eingangsmatrix)
-3.639830 ->Gewicht [0][6][3] (Eingangsmatrix)
-6.810479 ->Gewicht [0][6][4] (Eingangsmatrix)
0.339261 ->Gewicht [0][6][5] (Eingangsmatrix)
-2.455022 ->Gewicht [0][6][6] (Eingangsmatrix)
-3.518387 ->Gewicht [0][6][7] (Eingangsmatrix)
3.182555 ->Gewicht [0][6][8] (Eingangsmatrix)
2.989203 ->Gewicht [0][6][9] (Eingangsmatrix)
-3.398922 ->Gewicht [0][6][10] (Eingangsmatrix)
0.143110 ->Gewicht [0][6][11] (Eingangsmatrix)
1.558525 ->Gewicht [0][6][12] (Eingangsmatrix)
2.294815 ->Gewicht [0][6][13] (Eingangsmatrix)
-2.753867 ->Gewicht [0][6][14] (Eingangsmatrix)
-0.159066 ->Gewicht [0][6][15] (Eingangsmatrix)
3.040231 ->Gewicht [0][6][16] (Eingangsmatrix)
3.572966 ->Gewicht [0][6][17] (Eingangsmatrix)
-0.755952 ->Gewicht [0][6][18] (Eingangsmatrix)
1.488163 ->Gewicht [0][6][19] (Eingangsmatrix)
7.132093 ->Gewicht [0][6][20] (Eingangsmatrix)
-11.146221 ->Gewicht [0][6][21] (Eingangsmatrix)
-0.209490 ->Gewicht [0][6][22] (Eingangsmatrix)
-2.106546 ->Gewicht [0][6][23] (Eingangsmatrix)
-1.789796 ->Gewicht [0][7][0] (Eingangsmatrix)

5.201964 ->Gewicht [0][7][1] (Eingangsmatrix)
-15.625748 ->Gewicht [0][7][2] (Eingangsmatrix)
-3.499256 ->Gewicht [0][7][3] (Eingangsmatrix)
-7.630852 ->Gewicht [0][7][4] (Eingangsmatrix)
-0.062573 ->Gewicht [0][7][5] (Eingangsmatrix)
-2.125496 ->Gewicht [0][7][6] (Eingangsmatrix)
-0.779142 ->Gewicht [0][7][7] (Eingangsmatrix)
0.509695 ->Gewicht [0][7][8] (Eingangsmatrix)
3.436378 ->Gewicht [0][7][9] (Eingangsmatrix)
-0.237468 ->Gewicht [0][7][10] (Eingangsmatrix)
0.171680 ->Gewicht [0][7][11] (Eingangsmatrix)
5.405082 ->Gewicht [0][7][12] (Eingangsmatrix)
0.985321 ->Gewicht [0][7][13] (Eingangsmatrix)
-3.504826 ->Gewicht [0][7][14] (Eingangsmatrix)
1.242141 ->Gewicht [0][7][15] (Eingangsmatrix)
1.306920 ->Gewicht [0][7][16] (Eingangsmatrix)
2.961253 ->Gewicht [0][7][17] (Eingangsmatrix)
1.229076 ->Gewicht [0][7][18] (Eingangsmatrix)
0.905842 ->Gewicht [0][7][19] (Eingangsmatrix)
0.627606 ->Gewicht [0][7][20] (Eingangsmatrix)
-22.357256 ->Gewicht [0][7][21] (Eingangsmatrix)
-1.096193 ->Gewicht [0][7][22] (Eingangsmatrix)
4.713191 ->Gewicht [0][7][23] (Eingangsmatrix)
-23.636497 ->Gewicht [0][8][0] (Eingangsmatrix)
-13.750572 ->Gewicht [0][8][1] (Eingangsmatrix)
6.639669 ->Gewicht [0][8][2] (Eingangsmatrix)
-0.111342 ->Gewicht [0][8][3] (Eingangsmatrix)
-3.510155 ->Gewicht [0][8][4] (Eingangsmatrix)
-1.026550 ->Gewicht [0][8][5] (Eingangsmatrix)
1.634076 ->Gewicht [0][8][6] (Eingangsmatrix)
-3.536360 ->Gewicht [0][8][7] (Eingangsmatrix)
-5.359181 ->Gewicht [0][8][8] (Eingangsmatrix)
-14.822902 ->Gewicht [0][8][9] (Eingangsmatrix)
0.349942 ->Gewicht [0][8][10] (Eingangsmatrix)
-2.081456 ->Gewicht [0][8][11] (Eingangsmatrix)
-9.906072 ->Gewicht [0][8][12] (Eingangsmatrix)
-2.733570 ->Gewicht [0][8][13] (Eingangsmatrix)
3.771189 ->Gewicht [0][8][14] (Eingangsmatrix)
1.205914 ->Gewicht [0][8][15] (Eingangsmatrix)
-0.704575 ->Gewicht [0][8][16] (Eingangsmatrix)
-5.448370 ->Gewicht [0][8][17] (Eingangsmatrix)
1.182762 ->Gewicht [0][8][18] (Eingangsmatrix)
-1.696616 ->Gewicht [0][8][19] (Eingangsmatrix)
-6.338223 ->Gewicht [0][8][20] (Eingangsmatrix)
-8.781989 ->Gewicht [0][8][21] (Eingangsmatrix)
-10.776859 ->Gewicht [0][8][22] (Eingangsmatrix)
0.686226 ->Gewicht [0][8][23] (Eingangsmatrix)
-25.979189 ->Gewicht [0][9][0] (Eingangsmatrix)
-5.138209 ->Gewicht [0][9][1] (Eingangsmatrix)
5.125982 ->Gewicht [0][9][2] (Eingangsmatrix)
0.910162 ->Gewicht [0][9][3] (Eingangsmatrix)
1.655106 ->Gewicht [0][9][4] (Eingangsmatrix)
-0.315450 ->Gewicht [0][9][5] (Eingangsmatrix)
-0.031761 ->Gewicht [0][9][6] (Eingangsmatrix)
-5.314590 ->Gewicht [0][9][7] (Eingangsmatrix)
-0.089360 ->Gewicht [0][9][8] (Eingangsmatrix)
-8.334297 ->Gewicht [0][9][9] (Eingangsmatrix)
-4.196630 ->Gewicht [0][9][10] (Eingangsmatrix)
-1.992922 ->Gewicht [0][9][11] (Eingangsmatrix)
-7.428616 ->Gewicht [0][9][12] (Eingangsmatrix)
-0.297519 ->Gewicht [0][9][13] (Eingangsmatrix)

-9.673105 ->Gewicht [0][9][14] (Eingangsmatrix)
-7.303991 ->Gewicht [0][9][15] (Eingangsmatrix)
-1.682430 ->Gewicht [0][9][16] (Eingangsmatrix)
-5.022735 ->Gewicht [0][9][17] (Eingangsmatrix)
1.090209 ->Gewicht [0][9][18] (Eingangsmatrix)
-0.369200 ->Gewicht [0][9][19] (Eingangsmatrix)
-4.888967 ->Gewicht [0][9][20] (Eingangsmatrix)
-6.787325 ->Gewicht [0][9][21] (Eingangsmatrix)
-9.407678 ->Gewicht [0][9][22] (Eingangsmatrix)
-3.922818 ->Gewicht [0][9][23] (Eingangsmatrix)
-12.907464 ->Gewicht [0][10][0] (Eingangsmatrix)
-0.493437 ->Gewicht [0][10][1] (Eingangsmatrix)
-8.796753 ->Gewicht [0][10][2] (Eingangsmatrix)
-3.852510 ->Gewicht [0][10][3] (Eingangsmatrix)
-2.629842 ->Gewicht [0][10][4] (Eingangsmatrix)
1.600676 ->Gewicht [0][10][5] (Eingangsmatrix)
-3.291164 ->Gewicht [0][10][6] (Eingangsmatrix)
1.564077 ->Gewicht [0][10][7] (Eingangsmatrix)
-0.329657 ->Gewicht [0][10][8] (Eingangsmatrix)
-23.410769 ->Gewicht [0][10][9] (Eingangsmatrix)
0.570135 ->Gewicht [0][10][10] (Eingangsmatrix)
0.241041 ->Gewicht [0][10][11] (Eingangsmatrix)
7.879543 ->Gewicht [0][10][12] (Eingangsmatrix)
0.048849 ->Gewicht [0][10][13] (Eingangsmatrix)
-2.964844 ->Gewicht [0][10][14] (Eingangsmatrix)
2.317350 ->Gewicht [0][10][15] (Eingangsmatrix)
3.638913 ->Gewicht [0][10][16] (Eingangsmatrix)
7.284052 ->Gewicht [0][10][17] (Eingangsmatrix)
1.724375 ->Gewicht [0][10][18] (Eingangsmatrix)
0.244758 ->Gewicht [0][10][19] (Eingangsmatrix)
0.871858 ->Gewicht [0][10][20] (Eingangsmatrix)
-0.067530 ->Gewicht [0][10][21] (Eingangsmatrix)
3.385618 ->Gewicht [0][10][22] (Eingangsmatrix)
0.420576 ->Gewicht [0][10][23] (Eingangsmatrix)
6.510088 ->Gewicht [0][11][0] (Eingangsmatrix)
4.692452 ->Gewicht [0][11][1] (Eingangsmatrix)
-9.001779 ->Gewicht [0][11][2] (Eingangsmatrix)
-3.940931 ->Gewicht [0][11][3] (Eingangsmatrix)
-4.356757 ->Gewicht [0][11][4] (Eingangsmatrix)
1.273675 ->Gewicht [0][11][5] (Eingangsmatrix)
1.285162 ->Gewicht [0][11][6] (Eingangsmatrix)
-0.681519 ->Gewicht [0][11][7] (Eingangsmatrix)
-0.911892 ->Gewicht [0][11][8] (Eingangsmatrix)
-5.672645 ->Gewicht [0][11][9] (Eingangsmatrix)
7.518724 ->Gewicht [0][11][10] (Eingangsmatrix)
-1.174918 ->Gewicht [0][11][11] (Eingangsmatrix)
-6.076286 ->Gewicht [0][11][12] (Eingangsmatrix)
-1.747004 ->Gewicht [0][11][13] (Eingangsmatrix)
-4.613016 ->Gewicht [0][11][14] (Eingangsmatrix)
1.036958 ->Gewicht [0][11][15] (Eingangsmatrix)
-1.973949 ->Gewicht [0][11][16] (Eingangsmatrix)
-6.928124 ->Gewicht [0][11][17] (Eingangsmatrix)
-1.128777 ->Gewicht [0][11][18] (Eingangsmatrix)
0.032683 ->Gewicht [0][11][19] (Eingangsmatrix)
-15.665625 ->Gewicht [0][11][20] (Eingangsmatrix)
0.178573 ->Gewicht [0][11][21] (Eingangsmatrix)
-10.058521 ->Gewicht [0][11][22] (Eingangsmatrix)
-2.068517 ->Gewicht [0][11][23] (Eingangsmatrix)
-3.473562 ->Gewicht [0][12][0] (Eingangsmatrix)
14.881746 ->Gewicht [0][12][1] (Eingangsmatrix)
-43.999157 ->Gewicht [0][12][2] (Eingangsmatrix)

0.932441 ->Gewicht [0][12][3] (Eingangsmatrix)
1.186331 ->Gewicht [0][12][4] (Eingangsmatrix)
5.157071 ->Gewicht [0][12][5] (Eingangsmatrix)
1.207917 ->Gewicht [0][12][6] (Eingangsmatrix)
-12.473046 ->Gewicht [0][12][7] (Eingangsmatrix)
5.160373 ->Gewicht [0][12][8] (Eingangsmatrix)
31.020823 ->Gewicht [0][12][9] (Eingangsmatrix)
-7.642282 ->Gewicht [0][12][10] (Eingangsmatrix)
-0.246128 ->Gewicht [0][12][11] (Eingangsmatrix)
20.977287 ->Gewicht [0][12][12] (Eingangsmatrix)
-12.849347 ->Gewicht [0][12][13] (Eingangsmatrix)
-2.456974 ->Gewicht [0][12][14] (Eingangsmatrix)
-0.231538 ->Gewicht [0][12][15] (Eingangsmatrix)
-8.895576 ->Gewicht [0][12][16] (Eingangsmatrix)
-1.748066 ->Gewicht [0][12][17] (Eingangsmatrix)
-39.071228 ->Gewicht [0][12][18] (Eingangsmatrix)
1.148637 ->Gewicht [0][12][19] (Eingangsmatrix)
-5.309079 ->Gewicht [0][12][20] (Eingangsmatrix)
32.711712 ->Gewicht [0][12][21] (Eingangsmatrix)
56.188721 ->Gewicht [0][12][22] (Eingangsmatrix)
-15.344237 ->Gewicht [0][12][23] (Eingangsmatrix)
-0.551167 ->Gewicht [1][0][0] (Ausgangsmatrix)
-0.030157 ->Gewicht [1][0][1] (Ausgangsmatrix)
0.770617 ->Gewicht [1][1][0] (Ausgangsmatrix)
0.045490 ->Gewicht [1][1][1] (Ausgangsmatrix)
-1.102542 ->Gewicht [1][2][0] (Ausgangsmatrix)
0.021606 ->Gewicht [1][2][1] (Ausgangsmatrix)
-1.885314 ->Gewicht [1][3][0] (Ausgangsmatrix)
0.009557 ->Gewicht [1][3][1] (Ausgangsmatrix)
1.735528 ->Gewicht [1][4][0] (Ausgangsmatrix)
0.036327 ->Gewicht [1][4][1] (Ausgangsmatrix)
-0.895136 ->Gewicht [1][5][0] (Ausgangsmatrix)
-0.787398 ->Gewicht [1][5][1] (Ausgangsmatrix)
-0.955918 ->Gewicht [1][6][0] (Ausgangsmatrix)
-0.072024 ->Gewicht [1][6][1] (Ausgangsmatrix)
0.717696 ->Gewicht [1][7][0] (Ausgangsmatrix)
-0.180366 ->Gewicht [1][7][1] (Ausgangsmatrix)
-1.474463 ->Gewicht [1][8][0] (Ausgangsmatrix)
-0.040291 ->Gewicht [1][8][1] (Ausgangsmatrix)
1.151305 ->Gewicht [1][9][0] (Ausgangsmatrix)
-0.017085 ->Gewicht [1][9][1] (Ausgangsmatrix)
1.005993 ->Gewicht [1][10][0] (Ausgangsmatrix)
-0.022680 ->Gewicht [1][10][1] (Ausgangsmatrix)
-1.650213 ->Gewicht [1][11][0] (Ausgangsmatrix)
0.937746 ->Gewicht [1][11][1] (Ausgangsmatrix)
0.847388 ->Gewicht [1][12][0] (Ausgangsmatrix)
-0.003965 ->Gewicht [1][12][1] (Ausgangsmatrix)
-0.238134 ->Gewicht [1][13][0] (Ausgangsmatrix)
-1.741606 ->Gewicht [1][13][1] (Ausgangsmatrix)
2.286117 ->Gewicht [1][14][0] (Ausgangsmatrix)
0.003906 ->Gewicht [1][14][1] (Ausgangsmatrix)
-0.810104 ->Gewicht [1][15][0] (Ausgangsmatrix)
-0.179392 ->Gewicht [1][15][1] (Ausgangsmatrix)
1.175635 ->Gewicht [1][16][0] (Ausgangsmatrix)
-0.096861 ->Gewicht [1][16][1] (Ausgangsmatrix)
-0.563960 ->Gewicht [1][17][0] (Ausgangsmatrix)
-0.046649 ->Gewicht [1][17][1] (Ausgangsmatrix)
-0.936291 ->Gewicht [1][18][0] (Ausgangsmatrix)
0.153696 ->Gewicht [1][18][1] (Ausgangsmatrix)
-0.094611 ->Gewicht [1][19][0] (Ausgangsmatrix)
1.845917 ->Gewicht [1][19][1] (Ausgangsmatrix)

0.408699 ->Gewicht [1][20][0] (Ausgangsmatrix)
0.023311 ->Gewicht [1][20][1] (Ausgangsmatrix)
0.740642 ->Gewicht [1][21][0] (Ausgangsmatrix)
0.007772 ->Gewicht [1][21][1] (Ausgangsmatrix)
-1.511164 ->Gewicht [1][22][0] (Ausgangsmatrix)
-0.024992 ->Gewicht [1][22][1] (Ausgangsmatrix)
0.836840 ->Gewicht [1][23][0] (Ausgangsmatrix)
0.041362 ->Gewicht [1][23][1] (Ausgangsmatrix)

A.7 Literaturverzeichnis

- /1/ Arbib, M.: "Brains, Machines and Mathematics", 2nd edition, McGraw-Hill Press, NY, 1987
- /2/ Rumelhart, D.E.; McClelland, J.L.: "Parallel Distributed Processing", vol.1, MIT Press, Cambridge Mass, 1986
- /3/ Kohonen, T.: "Self-Organization and Assoziative Memory", 2nd edition, Springer Verlag, 1987
- /4/ Grossberg, S.: "Neural Networks and Natural Intelligence", MIT Press, Bradford Books, Cambridge Mass, 1988
- /5/ Eckmiller, R.; v.d.Malsburg (Eds.): "Neural Computers", Springer Verlag, 1987
- /6/ Lippmann, R.P.: "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, April 1987, S.4-22
- /7/ Kemke, C.: "Der Neuere Konnektionismus", Informatik-Spektrum, November 1988
- /8/ Palm, G.: "Assoziatives Gedächtnis und Gehirntheorie", Spektrum der Wissenschaft, Juni 1988
- /9/ Benner, J.; Blume, C.; Leinemann, K.; Schlechtendahl, E.G.; Wadle, M.: "Advanced Carrier Systems and Telerobotics", Proc. Symposium on Robot Control (SYROCO), Karlsruhe, 5.-7. Oktober 1988, VD/VDE-Gesellschaft Meß- und Automatisierungstechnik, S.82.1-82.6
- /10/ Smidt, D.; Blume, C.; Wadle, M.: "A Multi-link Multi-purpose Advanced Manipulator with a large Handling Hemisphere for Out-door Applications.", Proc. 5.Int. Symp. on Robotics in Construction, Tokyo, 1988, vol.2, S.625-634
- /11/ Kühnappel, U.; Leinemann, K.; Schlechtendahl, E.G.: "Graphics support for JET boom control.", Proc. ANS 2nd International Topical Meeting on remote Systems and Robotics in Hostile Environments, Pasco, Wa. La. Grange Park, Ill.: ANS, 1988, S.28-34
- /12/ Leinemann, K.; Kühnappel, U.; Ludwig, A.: "Remote Handling Control with Graphical Man-Machine Interface for NET and JET.", Proc. IAEA Technical Committee Meeting on Robotics and Remote Maintenance Concepts for Fusion Machines, Karlsruhe, 22.-24.2.1988, IAEA-TECDOC-495, ISSN 1011-4289, 1989, S.215-226
- /13/ Hirzinger, G.; Dietrich, J.; Heindl, J.: "Zum Stand der Robotikarbeiten der DLR." DLR-Nachrichten (Heft 58), November 1989
- /14/ Benner, J.; Leinemann, K.: "Architecture of a telemanipulation system with combined sensory and operator control", in: W. Guttropf (Ed.), Proc. RoVi-SeC-7, 2.-4.2.1988, Zürich, S. 259-271