Neuro-Fuzzy-Regelung mit NEFCON unter MATLAB/SIMULINK

Andreas Nürnberger, Detlef Nauck und Rudolf Kruse
Otto-von-Guericke-Universität Magdeburg
Institut für Informations- und Kommunikationssyst eme
Universitätsplatz 2, 39106 Magdeburg
E-Mail: A.Nuernberger@iik.cs.uni-magdeburg.de

Kurzfassung

Zur Erstellung und Optimierung von Fuzzy-Reglern werden häufig Verfahren eingesetzt, die aus der Kombination Neuronaler Netze mit Fuzzy-Reglern entstanden sind. Im folgenden wird die Implementierung eines hybriden Neuro-Fuzzy-Modells beschrieben, das ist in der Lage ist, die Regelbasis eines konventionellen Fuzzy-Reglers zu erlernen und zu optimieren. Durch die Implementierung unter dem Simulationssystem MATLAB/SIMULINK[1], kann das Modell sehr einfach zur Entwicklung und Optimierung von Fuzzy-Reglern für verschiedenste dynamische Systeme eingesetzt werden. Die Implementierung ist über WWW verfügbar (http://fuzzy.cs.uni-magdeburg.de/nefcon).

1 Einleitung

Um die Entwicklung von Fuzzy-Reglern, d.h. die Bestimmung der zur Problemrepräsentation verwendeten Fuzzy-Mengen und Fuzzy-Regeln (Wissensbasis), zu vereinfachen, sind sogenannte Neuronale Fuzzy-Systeme entwickelt worden. Diese kombinieren Fuzzy-Systeme und Konzepte der Neuronalen Netze miteinander, wodurch ein lernfähiges System entsteht, das sich mit Hilfe eines Lernalgorithmus seiner Aufgabe anpassen kann. Am Institut für Betriebssysteme und Rechnerverbund der Technischen Universität Braunschweig wurde in diesem Zusammenhang das NEFCON-Modell entwickelt (NEuro-Fuzzy CONtrol), das zur Realisierung Neuronaler Fuzzy-Regler geeignet ist. Im folgenden wird eine Erweiterung und die Implementierung des Modells unter MATLAB/SIMULINK beschrieben. Die Entwicklung erfolgte in Kooperation mit der Daimler-Benz Aerospace Airbus GmbH [NÜRNBERGER, 1996].

2 Das NEFCON-Modell

Das NEFCON-Modell ist ein hybrides Modell eines Neuronalen Fuzzy-Reglers. Ausgehend von der Definition eines Fuzzy-Reglers, erhält man die Netzstruktur, wenn man die Fuzzy-Mengen als Gewichte, die Meß- und Stellgrößen sowie die Regeln als Verarbeitungseinheiten interpretiert. Das Netz hat dann die Struktur eines Multilayer-Perceptrons und kann als dreischichtiges Fuzzy-Perceptron [NAUCK et. al., 1995] interpretiert werden. Ein Beispiel für eine Netzstruktur eines Fuzzy-Reglers mit 5 Regeleinheiten, 2 Meßgrößen und einer Stellgröße kann Abbildung 2.1 entnommen werden.

[1] Von The Mathworks Inc., 24 Prime Park Way, Natick, Mass.01760; http://www.mathworks.com

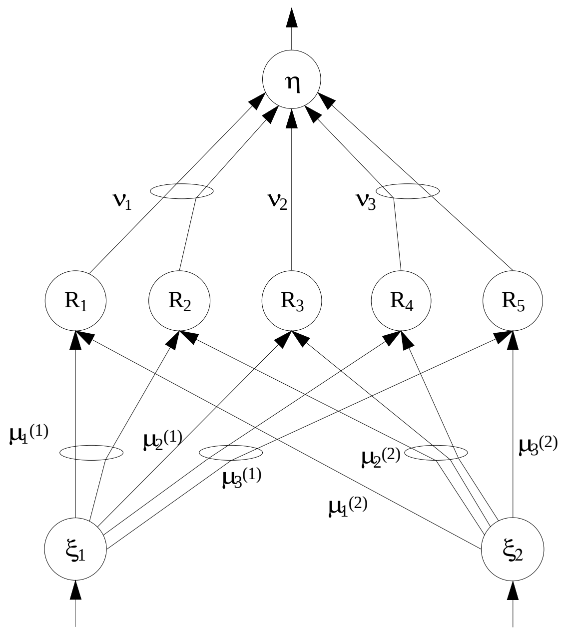

Abbildung 2.1: Ein NEFCON System mit 2 Eingangsgrößen und 5 Regeln

Die inneren Einheiten R_1, …, R_5 repräsentieren hierbei die Regeln, die Einheiten ξ_1, ξ_2 und η die Meß- bzw. Stellgrößen und $\mu_r^{(i)}$ und ν_r die Fuzzy-Mengen für die Prämissen bzw. Konklusionen. Die Verbindungen mit gemeinsamen Gewichten (im Bild durch Ellipsen symbolisiert) kennzeichnen gleiche Fuzzy-Mengen. Bei einer Veränderung dieser Gewichte müssen somit alle Verbindungen mit diesem Gewicht angepaßt werden, um sicherzustellen, daß gleiche Fuzzy-Mengen weiterhin durch gleiche Gewichte repräsentiert werden. Die durch die Netzstruktur definierte Regelbasis kann auch in Form von Kontrollregeln formuliert werden. Man erhält für die obige Netzstruktur z.B.:

$$R_1\text{: if } \xi_1 \text{ is } \mathrm{A}_1^{(1)} \text{ and } \xi_2 \text{ is } \mathrm{A}_1^{(2)} \text{ then } \eta \text{ is } \mathrm{B}_1$$

3 Der Lernalgorithmus

Das Lernverfahren des NEFCON-Modells läßt sich im wesentlichen in zwei voneinander unabhängige Phasen aufteilen. In der ersten Phase wird eine Regelbasis für das dynamische System erlernt, oder - falls a-priori Wissen vorhanden ist - manuell aufgestellt. Diese Regelbasis wird im allgemeinen das System noch nicht ausreichend genau steuern, eine Optimierung der Fuzzy-Mengen ist notwendig. Diese Anpassung erfolgt in der zweiten Phase durch den eigentlichen NEFCON-Lernalgorithmus.

Beide Phasen erfordern die Definition eines Systemfehlers E zur Bestimmung bzw. Anpassung der Kontrollregeln (siehe hierzu Abschnitt *3.3*). Neben dem Fehler E muß weiterhin bekannt sein, ob der optimale Stellwert η_{opt} für einen gegebenen Zustand positiv oder negativ ist. Zusammen ergibt sich somit der erweiterte Fehler $E^* \in [-1,1]$ eines NEFCON-Systems:

$$E^*(x_1, ..., x_n) = \mathrm{sgn}(\eta_{opt}) \cdot E(x_1, ..., x_n),$$

wobei $(x_1, ..., x_n)$ die aktuellen Meßwerte des zu regelnden Systems bzw. die Eingabe in das NEFCON-System sind.

3.1 Optimieren einer Regelbasis

Der NEFCON-Lernalgorithmus und die vorgestellten Variationen basieren auf dem Prinzip des Backpropagation-Algorithmus für das mehrschichtige Perzeption. Der Fehler wird, beginnend bei der Ausgabeeinheit, rückwärts durch das Netzwerk propagiert und lokal zur Adaption der Fuzzy-Mengen genutzt.

Das Lernverfahren NEFCON I

Beim Lernverfahren NEFCON I werden die Fuzzy-Mengen einer Regel in Abhängigkeit von ihrem Beitrag zu einer Regelaktion und dem daraus resultierenden Fehler entweder ‘belohnt’ oder ‘bestraft’. Es wird somit das Prinzip des verstärkenden Lernens (reinforcement learning) verwendet (siehe auch [BARTO et. al., 1983; BARTO, 1992; WHITE und SOFGE, 1992]). Eine ‘Belohnung’ oder ‘Bestrafung’ erfolgt durch Verschieben oder Verkleinern/Vergrößern des Trägers der Fuzzy-Mengen. Das Verfahren wird in [NAUCK et. al., 1996] für Regeln mit monotonen Zugehörigkeitsfunktionen in den Konklusionen beschrieben (siehe hierzu auch [LEE, 1990; TSUKAMOTO, 1979]).

In der vorliegenden Implementierung wurden die monotonen Fuzzy-Mengen der Konklusionen durch symmetrische Fuzzy-Mengen ersetzt. Das Belohnen/Bestrafen erfolgt jetzt durch entsprechende Verschiebung der Konklusion. Das Lernverfahren ist somit für einen Mamdani-Regler (siehe [MAMDANI und ASSILIAN, 1975; KRUSE et. al., 1995]) einsetzbar. Die Prämissen werden analog dem Verfahren in [NAUCK et. al., 1996] verändert, wobei auch hier symmetrische Fuzzy-Mengen zugelassen werden[1]. Eine formale Beschreibung des Verfahrens kann [NÜRNBERGER, 1996] entnommen werden.

Das Lernverfahren NEFCON II

Beim Lernverfahren NEFCON II wird die Fehleränderung bei zwei aufeinanderfolgenden Zuständen betrachtet (*Fehlertendenz*), um so eine effektivere Optimierung zu erreichen. Die Optimierung erfolgt sowohl in den Prämissen, als auch in den Konklusionen durch Verschiebung. Die Fehlertendenz $\tau \in \{-1, 0, 1\}$ wird mittels des erweiterten Fehlers E^* zu einem Zeitpunkt t und $E^{*\prime}$ zum Zeitpunkt t+1 definiert:

$$\tau = \begin{cases} 1 & falls \quad (|E^{*\prime}| \geq |E^*|) \wedge (E^{*\prime} \cdot E^*) \geq 0) \\ 0 & falls \quad (|E^{*\prime}| < |E^*|) \wedge (E^{*\prime} \cdot E^*) \geq 0) \\ -1 & sonst \end{cases}$$

Für τ=0 strebt der Fehler dem optimalen Zustand entgegen, es wird keine Änderung an den Fuzzy-Mengen vorgenommen. Für τ=1 wächst der Fehler, und sein Vorzeichen ändert sich nicht. Der Stellwert wird durch Verschieben der Regelkonklusionen vergrößert. Die Regelprämissen der Regeln mit Stellwert steigerndem Beitrag werden belohnt, die mit senkendem Beitrag bestraft. Für τ=-1 hat das System übersteuert. Der Stellwert wird gesenkt und die Regelprämissen entsprechend belohnt bzw. bestraft. Eine formale Definition des Verfahrens kann [NAUCK et. al., 1995] entnommen werden.

In der vorliegenden Implementierung wurde analog dem Lernverfahren NEFCON I eine Erweiterung auf trapez- und gaußförmige Fuzzy-Mengen durchgeführt [NÜRNBERGER, 1996].

---

[1] Implementiert wurden dreiecks-, trapez- und gaußförmige Fuzzy-Mengen. Eine Erweiterung auf weitere (symmetrische) Fuzzy-Mengen ist leicht möglich.

Gegenüberstellung der Lernverfahren

Beide Verfahren unterstützen jetzt einen 'echten' Mamdani-Regler, wodurch die optimierten Regelbasen in Standard Fuzzy-Reglern eingesetzt werden können.

Die beim Lernverfahren NEFCON II verwendete Fehlertendenz betrachtet lediglich zwei aufeinanderfolgende Zustände. Dies kann bei Lernvorgängen mit dynamischen Systemen, wie auch bei dem Verfahren NEFCON I, zur Verhinderung einer Konvergenz beitragen, da die Trägheit des Systems nicht beachtet wird. Eine Regelungsaktion kann hier erst zeitverzögert eine Verbesserung des Systemzustandes auslösen (siehe auch [LEONHARD, 1992]). Eine Verfeinerung der Fehlerbeschreibung kann eine Verbesserung, aber keine Beseitigung des Problems bringen. Zu Problemen kognitiver Regelung siehe auch [WERBOS, 1992].

Eine Kombination der Verfahren (d.h. Verschieben und Verkleinern/Vergrößern des Trägers der Fuzzy-Mengen) verbessert die Lösungsgüte meist nicht weiter [NAUCK et. al., 1995]. Eine Kombinationsmöglichkeit wurde aus diesem Grund nicht implementiert.

3.2 Erlernen einer Regelbasis

Falls für das zu steuernde System noch keine ausreichende Regelbasis existiert, bzw. aufgestellt werden kann, muß sie durch ein geeignetes Regellernverfahren erzeugt werden. Die im folgenden vorgestellten Algorithmen zum Erlernen einer Regelbasis kommen, anders als die im allgemeinen für Neuronale Fuzzy-Modelle verwendeten Clustering-Verfahren (siehe hierzu z.B. [BERENJI und KHEDKAR, 1993]), ohne eine vorgegebene feste Lernaufgabe aus.

Erlernen einer Regelbasis nach NEFCON I

Das Verfahren zum Erlernen einer Regelbasis nach NEFCON I ist ein Eliminationsverfahren, das mit einer vollständigen überbestimmten Regelbasis beginnt. Diese wird mittels der initialen Partitionierung der Variablen für die Meß- und Stellgrößen gebildet. Der Lernvorgang teilt sich im wesentlichen in zwei Phasen auf. In der ersten Phase werden die Regeleinheiten entfernt, deren Ausgaben ein unterschiedliches Vorzeichen zum optimalen Stellwert aufweisen. In der zweiten Phase werden aus den verbliebenen Mengen der Regeln mit gleichen Prämissen jeweils die Regeln ausgewählt, die in mehreren Regelungszyklen den geringsten Fehler akkumuliert haben. Die restlichen Regeln und Regeln, die sehr selten oder überhaupt nicht aktiv waren, werden eliminiert.

Die vorliegenden Implementierung basiert auf dem in [NAUCK et. al., 1996] beschriebenen Verfahren. Die Entfernung der Regeleinheiten, deren Ausgabe ein unterschiedliches Vorzeichen zum optimalen Stellwert aufweist, erfolgt hier jedoch erst dann, wenn der optimale Stellwert außerhalb eines Schwellwert-Intervalls um 0 liegt. Somit werden Regeln, die eine stationäre Regelabweichung ausgleichen können oder ein Überschwingen verhindern, nicht gelöscht. (Diese Regeln müssen für $\eta_{opt}=0$ einen Stellwert ungleich Null liefern.) Eine ausführliche Beschreibung des überarbeiteten Lernverfahrens kann [NÜRNBERGER, 1996] entnommen werden.

Das 'bottom-up'-Lernverfahren

Neben dem Lernverfahren NEFCON II zur Optimierung von Fuzzy-Mengen wurde in [NAUCK et. al., 1995] ein Verfahren zum Erlernen der Fuzzy-Regeln, das mit einer leeren Regelbasis beginnt, vorgestellt.

Ausgehend von einer vorgegebenen Partitionierung der linguistischen Variablen, werden die Verbindungen zwischen den Schichten - die Regeln - erstellt. Das Lernverfahren besteht dabei aus zwei Phasen. In der ersten Phase werden die Meßwerte mit Hilfe der vorhanden Partitionierung klassifiziert, d.h. für jeden Meßwert wird die Fuzzy-Menge mit dem höchsten Zugehörigkeitsgrad ausgewählt. Der zugehörige Stellwert wird mittels einer Heuristik direkt aus dem Fehler bestimmt (lineare Abbildung des erweiterten Fehlers E^* auf das Intervall des Stellwerts) und anschließend analog den Meßwerten klassifiziert. Die so gefundene Regel wird in die Regelbasis aufgenommen. Dabei geht man davon aus, daß Meßwerttupel mit ähnlichen Fehlerwerten auch ähnliche Stellwerte erfordern. In der zweiten Phase wird versucht, die Regelkonklusionen zu optimieren (durch neue Zuordnung der Fuzzy-Menge der Konklusion, bei starker Stellwertabweichung vom aktuell ermittelten Fehler E^*) und die Regelbasis zu minimieren (selten benutzte Regeln werden gelöscht). Eine formale Beschreibung des Verfahrens kann [NAUCK et. al., 1995] entnommen werden.

Gegenüberstellung der Regellernverfahren

Das Lernverfahren NEFCON I ist für Systeme mit vielen Meßwerten oder vielen Fuzzy-Mengen zur Beschreibung der Meß- und Stellgrößen sehr speicherplatz- und rechenzeitaufwendig. Es liefert aber, wenn während der Lernphasen eine gute Abdeckung des Zustandsraums sichergestellt wird eine gute Regelbasis.

Das 'bottom-up'-Lernverfahren ermöglicht es sehr gut, Vorwissen in die Regelbasis einfließen zu lassen. Fehlende Regeln werden durch das Verfahren hinzugefügt. Das 'bottom-up'-Lernverfahren ist jedoch nicht für Systeme geeignet, die bei verschiedenen optimalen Zuständen (Fehlerwert $E \approx 0$) unterschiedliche Stellwerte erhalten müssen. Diese Zustände erhalten aufgrund der Konstruktion des Verfahrens[1] identische Konklusionen, die in Phase 2 nur bei ausreichend groß gewählter Lernrate korrigiert werden können. Dies führt zu einem Alternieren während der Optimierung der Fuzzy-Mengen (die Konklusionen 'wandern' zwischen den jeweils optimalen Stellgrößen). Dies ist z.B. bei regelungstechnischen Systemen der Fall, die eine Integralanteil zur Regelung benötigen (I-Regler) um eine stationäre Regelabweichung auszugleichen. Weiterhin muß bei Systemen, die zur Einregelung ständig eine Stellgröße $o_\eta \neq 0$ benötigen, in der Lernphase sichergestellt werden, daß der gesamte Meßwertbereich abgedeckt wird. Für Systeme, die keinen Integralanteil zur Regelung benötigen und auch bei undefinierten Zuständen (Stellgröße $o_\eta = 0$) ihren Zustand nicht stark verschlechtern, ist dieses Verfahren jedoch aufgrund seiner Schnelligkeit sehr gut geeignet (z.B. Inverses Pendel). Bei der Partitionierung der Meß- und Stellwertintervalle muß eine Überdeckung sichergestellt werden, um dem Verfahren eine Klassifikation der Meßwerte zu ermöglichen.

---

[1] Annahme einer direkten Abhängigkeit zwischen Stellwert und Fehler ('ähnliche Stellwerte für ähnliche Fehler')

3.3 Bestimmung des Fehlers

Die Beschreibung des Fehlers bzw. der Güte des aktuellen Systemzustands ist meist recht aufwendig, da der optimale Stellwert in den meisten Fällen unbekannt ist. Eine gute Fehlerbeschreibung verbessert die Güte der Lösung jedoch zum Teil erheblich. Für die Bestimmung des Fehlers $E(x_1, ..., x_n)$, wobei $(x_1, ..., x_n)$ die aktuellen Meßwerte des zu regelnden Systems sind, werden aus diesem Grund einige Möglichkeiten aufgezeigt.

Fehlerbeschreibung durch linguistische Regeln

Betrachtet man die Güte eines Systems als eine unscharfe Größe, so kann man sie auch als Fuzzy-Güte und den Fehler somit als Fuzzy-Fehler bezeichnen (siehe hierzu auch [NAUCK et. al., 1996]).

Eine für allgemeine Regelkreise verwendbare linguistische Fehlerbeschreibung erhält man, wenn man zur Bestimmung des Fuzzy-Fehlers die Regelabweichung (dx) und die Steigung bzw. Änderung der Regelabweichung (dx') benutzt. Für dx, dx' und den erweiterten Fehler error wurden folgende Fuzzy-Mengen und Regeln aufgestellt:

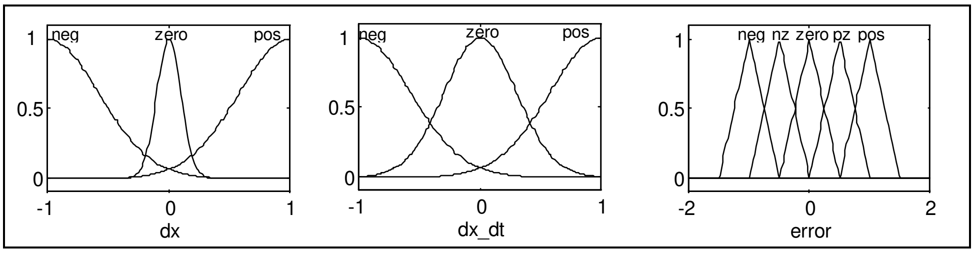

Abbildung 3.2: Partitionierung der Variablen zur Bestimmung des Fuzzy-Fehlers

```
1. If (dx is neg) and (dx_dt is pos) then (error is zero)
2. If (dx is pos) and (dx_dt is neg) then (error is zero)
3. If (dx is pos) and (dx_dt is pos) then (error is pos)
4. If (dx is neg) and (dx_dt is neg) then (error is neg)
5. If (dx is pos) and (dx_dt is zero) then (error is pos)
6. If (dx is neg) and (dx_dt is zero) then (error is neg)
7. If (dx is zero) and (dx_dt is neg) then (error is nz)
8. If (dx is zero) and (dx_dt is pos) then (error is pz)
9. If (dx is zero) and (dx_dt is zero) then (error is zero)
```

Abbildung 3.3: Regeln zur Bestimmung des Fuzzy-Fehlers

Durch die Regeln 1 und 2 werden hierbei die kompensatorischen Zustände beschrieben. Die Regeln 3-6 beschreiben ‘schlechte’ Zustände und definieren die Stärke des Fehlers und das Vorzeichen des optimalen Stellwerts. Die Regeln 7 und 8 sollen ein Überschwingen dämpfen, indem sie starke Steigungen für $dx \approx 0$ bestrafen. Die Regel 9 beschreibt schließlich den Optimalzustand.

Diese Fehlerbeschreibung kann, durch Abbildung von dx und dx_dt auf das Intervall [-1,1], auf beliebige Systeme angewandt werden. Anpassungen sind in einigen Fällen in der Breite der Fuzzy-Menge *zero* und für das Vorzeichen des optimalen Stellwerts erforderlich.

Fehlerbeschreibung durch ‘Fuzzy-Intervalle’

Das Verfahren basiert auf der Beschreibung des Fehlers mittels des Sollwerts umgebender scharfer Intervallgrenzen (siehe *Abbildung 3.4*).

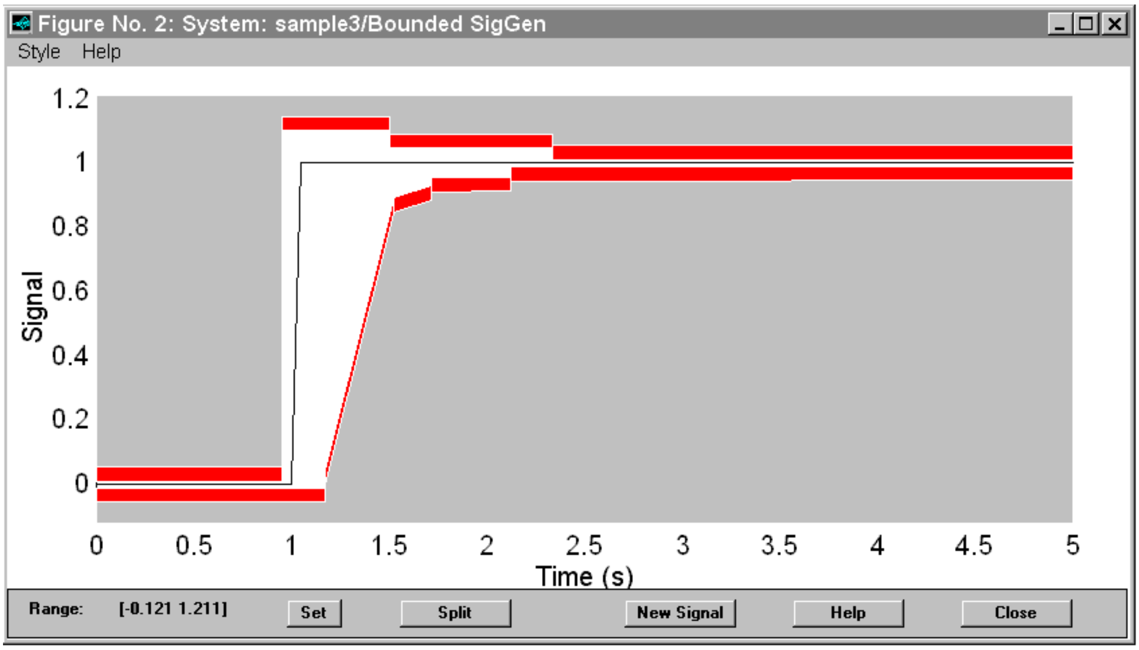

Abbildung 3.4: Fehlerbeschreibung durch ’Fuzzy-Intervalle’ (Sprungfunktion)

In der Regellernphase wird der Fehler analog der Fehlerbeschreibung durch linguistische Regeln bestimmt, da bei der Aufstellung der Regelbasis der exakte Signalverlauf betrachtet werden muß. In der Optimierungsphase wird ein Signalverlauf innerhalb der Intervallgrenzen als Optimalzustand angesehen ($E^*=0$). Überschreiten die Meßwerte die Intervallgrenzen, wird die Intervallgrenze als Signal bzw. Sollwert interpretiert und der Fehler aus der Abweichung zu diesem Wert bestimmt.

Durch dieses Verfahren werden in der Optimierungsphase gezielt nur die Regeln verändert, die das gewünschte Verhalten zu diesem Zeitpunkt nicht ausreichend unterstützen. Eine Vermeidung unnötiger Veränderungen bzw. Optimierung des Signalverlaufs innerhalb des Toleranzbereichs, verhindert ebenfalls eine unbeabsichtigte Verschlechterung in den kritischen Bereichen. ‘Bessere’ Lösungen werden bei diesem Verfahren ausgeschlossen. Dies wird implizit durch die Intervallgrenzen beabsichtigt. Eine Alternative besteht darin, innerhalb der Intervallgrenzen die Lernrate zum Sollwert hin zu verkleinern, um so auch innerhalb der Intervallgrenzen eine Verbesserung zu erzielen.

Durch dieses Verfahren ist es möglich, dem Benutzer ein einfaches und intuitiv benutzbares Werkzeug zur Fehlerbeschreibung zur Verfügung zu stellen.

4 NEFCON unter MATLAB

Das NEFCON-Modell wurde als interaktives Werkzeug konzipiert und in die vorhandene Umgebung von SIMULINK und der Fuzzy Logic Toolbox[1] integriert. Es ist somit während der Laufzeit jederzeit möglich, mit den Tools der Toolbox

---

[1] Eine Bibliothek zur Erstellung von Fuzzy-Reglern von ‘The Mathworks’

und des MATLAB/SIMULINK Systems, aktuelle Daten aus dem NEFCON-Modell auszulesen bzw. zu bearbeiten.

Die prinzipielle Einbindung des NEFCON-Modells kann der folgenden Strukturdarstellung eines einfachen Regelkreises entnommen werden.

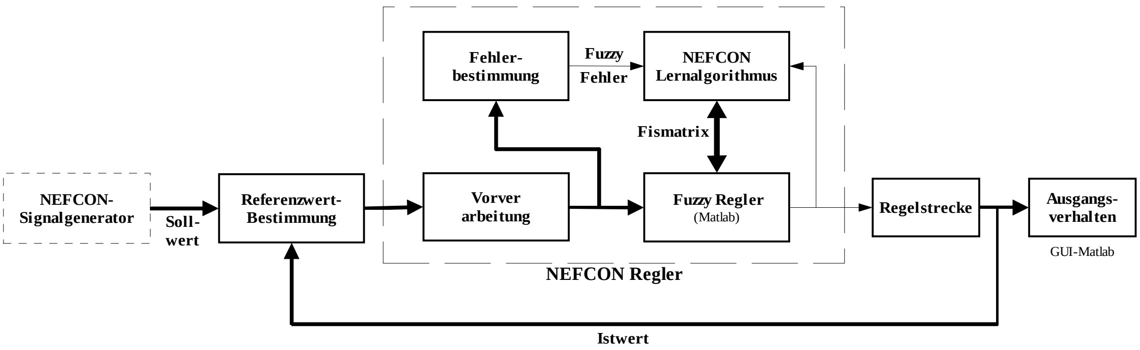

Abbildung 4.1: Einbindung des NEFCON-Reglers in MATLAB/SIMULINK

Bei einer konkreten Implementierung (hier am Beispiel einer Füllstandsregelung für einen Wassertank, die auf einem Beispiel der Fuzzy Logic Toolbox basiert) erhält man folgendes Simulationsmodell:

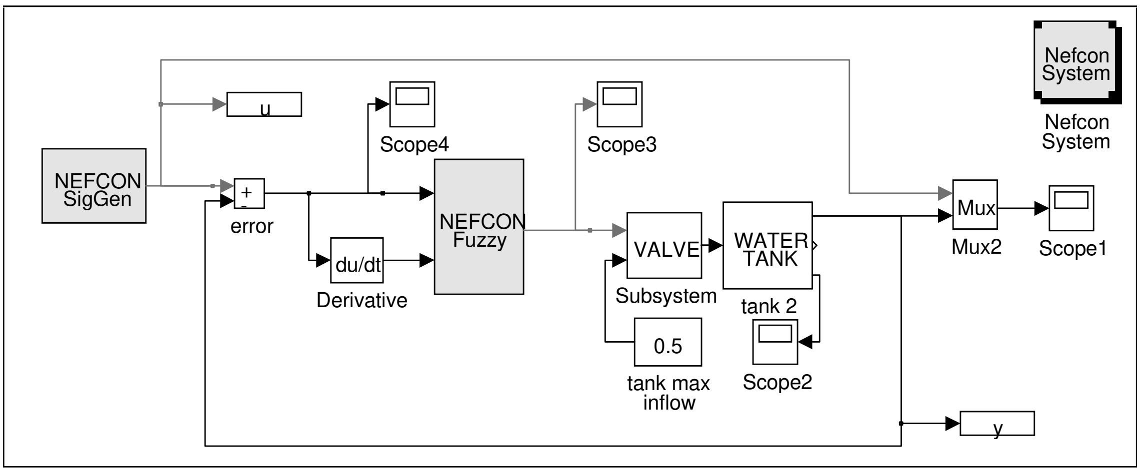

Abbildung 4.2: MATLAB/SIMULINK Simulation einer Füllstandsregelung

Hierbei wurde zur Sollwerterzeugung der NEFCON-Signalgenerator benutzt, der verschiedene Möglichkeiten der Signaldefinition zur Verfügung stellt. Der eigentliche Fuzzy-Regler und das Optimierungsverfahren werden über den Block ‘Nefcon Fuzzy’ ausgeführt.

Ein Beispiel für die integrierte Simulationsumgebung kann *Abbildung 4.3* entnommen werden. Hier wurde das NEFCON Lernverfahren in einem Regelkreis mit PT_2-Glied eingesetzt. Zur Darstellung ausgewählter Signale (Soll-/Istwert, Regelabweichung, Fehler und Stellwert) wurden SIMULINK Scope Blöcke benutzt. Das Beispiel wurde unter Microsoft Windows NT 4.0 erstellt.

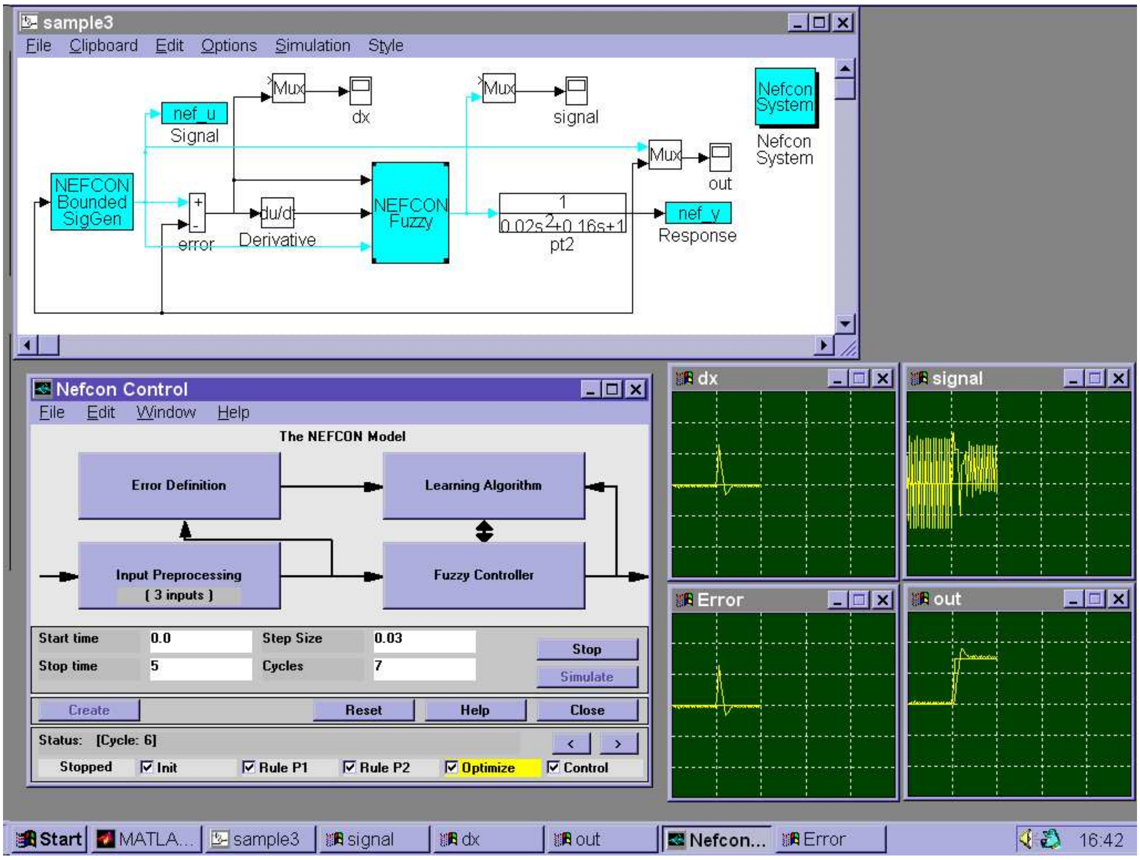

Abbildung 4.3: Beispiel einer Simulationsumgebung

Die Verwaltung der Parameter und die Steuerung des Lernverfahrens erfolgt über verschiedene GUI-Tools. Das Tool 'Nefcon Control' übernimmt dabei die gesamte Ablaufsteuerung. Es können hiermit gezielt Lernphasen wiederholt oder übersprungen werden. Ebenso können Ergebnisse früherer Lernschritte - oder Daten bestehender Fuzzy-Regler - übernommen und weiter optimiert werden. Die Parameter der Lernvorgänge und die Simulationsergebnisse werden automatisch gespeichert und können für spätere Analysen verwendet werden. Eine Änderung der Parameter und eine gezielte Auswahl der Lernphasen ist jederzeit während der Simulation möglich.

5 Anwendungsbeispiele

5.1 Füllstandsregelung für einen Wassertank

Aufgabe des Fuzzy-Reglers ist es hierbei, den Wasserstand eines Tanks an ein vorgegebenes Niveau anzupassen. Der gewünschte Füllstand wird über eine Rechteckfunktion vorgegeben. Der Stellwert des Reglers ist ein Maß für die Geschwindigkeit, mit der ein Ventil den Zufluß verringert bzw. erhöht. Die Regelung benötigt somit keinen Integralanteil, da das Ventil für einen Stellwert 0 in einem stabilen Zustand bleibt (Ventilzustand wird nicht verändert). Das SIMULINK-Modell kann *Abbildung 4.2* entnommen werden.

Der Lernvorgang wurde mit dem 'bottom-up'-Regellernverfahren und dem Optimierungsverfahren NEFCON I durchgeführt. Die initiale Partitionierung wurde mit drei Fuzzy-Mengen in den Prämissen und fünf in den Konklusionen aufgestellt:

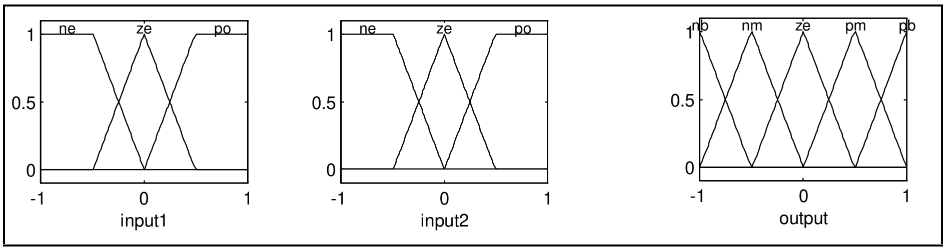

Abbildung 5.1: Partitionierung der Meß- und Stellwerte für eine Füllstandsregelung

Der Fehler wird über die in *3.3* vorgestellte linguistische Fehlerbeschreibung ermittelt. In der ersten Regellernphase wird durch den NEFCON Signalgenerator Gaußsches Rauschen erzeugt, um eine gute Überdeckung des Zustandsraums zu erreichen. Bereits nach der zweiten Regellernphase wurde eine sehr gute Lösung erreicht, die auch durch den Optimierungsalgorithmus nicht mehr verbessert werden konnte.

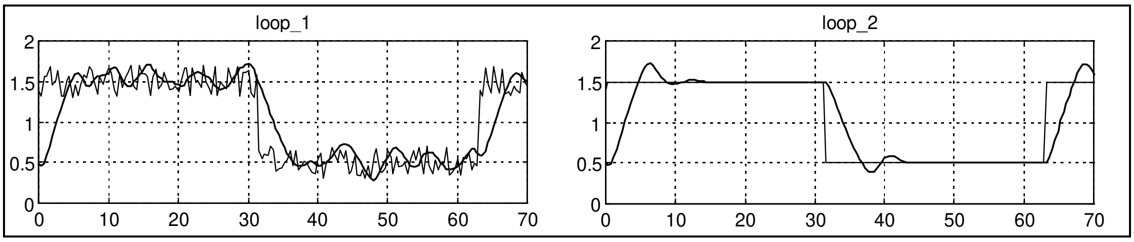

Abbildung 5.2: Ergebnis des Regellernverfahrens für eine Füllstandsregelung

Das Verfahren hat eine den Zustandsraum überdeckende Regelbasis erlernt:

```
1. If (input1 is po) and (input2 is ze) then (output is pb)
2. If (input1 is po) and (input2 is po) then (output is pb)
3. If (input1 is po) and (input2 is ne) then (output is pm)
4. If (input1 is ze) and (input2 is ne) then (output is ze)
5. If (input1 is ze) and (input2 is ze) then (output is ze)
6. If (input1 is ne) and (input2 is ne) then (output is nb)
7. If (input1 is ne) and (input2 is po) then (output is nm)
8. If (input1 is ze) and (input2 is po) then (output is pm)
9. If (input1 is ne) and (input2 is ze) then (output is nm)
```

5.2 Regelung eines Systems der Ordnung 2 (PT_2)

Weiterhin wurde das Lernverfahren an einer Regelstrecke getestet, die zum Ausgleich einer stationären Regelabweichung einen Integralanteil benötigt. Die Regelstrecke enthält hierbei als dynamisches System ein Übertragungselement 2. Ordnung (PT_2-Glied). PT_2-Glieder werden in der konventionellen Regelungstechnik mit einem PI- oder einem PID-Regler geregelt. Zu dieser Problematik siehe auch [LEONHARD, 1992] und [KNAPPE, 1994].

An diesem Beispiel soll ebenfalls die praktische Einsetzbarkeit auf Probleme der klassischen Regelungstechnik dargestellt werden. Das Lernverfahren ist hier in der Lage, mit einer sehr einfachen Fehlerbeschreibung und einer sehr groben Partitionierung der Meß- und Stellwertintervalle eine gute Regelbasis zu erstellen. Der Aufbau des Modells kann *Abbildung 4.3* entnommen werden.

Zur Erzeugung eines Integralanteils wird hier, neben der Regelabweichung und deren Ableitung, der Sollwert an das NEFCON-Lernsystem übergeben. Als Signal wird eine Sprungfunktion erzeugt. Die Partitionierung der Meß- und Stellwertintervalle erfolgt analog dem Beispiel der Füllstandsregelung aus Abschnitt 5.1. Als Regellernverfahren und zur Optimierung wurde NEFCON I verwendet. Der Fehler wurde durch Fuzzy-Intervalle beschrieben (siehe *Abbildung 3.4*).

Der Ablauf der Optimierung kann der folgenden Abbildung entnommen werden:

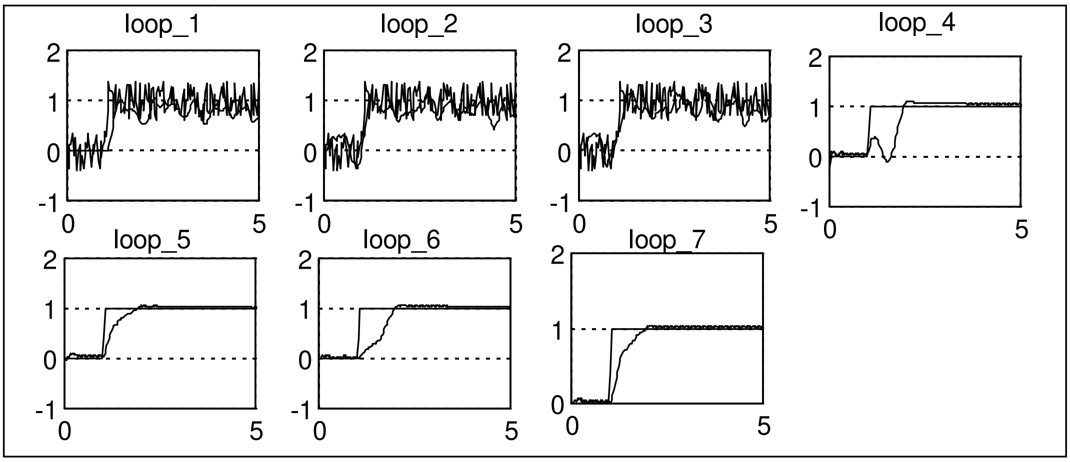

Abbildung 5.3: Ergebnisse der Lernphasen (PT_2-Regelung)

6 Zusammenfassung

Durch die Implementierung der modifizierten NEFCON Algorithmen unter MATLAB/SIMULINK ist es sehr einfach möglich, das Lernverfahren für beliebige, komplexe dynamische Systeme einzusetzen. Dazu ist meist keine Programmierung, sondern lediglich eine Erstellung des SIMULINK-Modells notwendig. Auch Änderung an der Systemkonfiguration sind schnell und effektiv realisierbar.

Mit dem vorgestellten Lernverfahren können, bei entsprechender Wahl der Meß- und Stellwertintervalle, sehr gute Ergebnisse erzielt werden. Die Güte der Lösung ist jedoch sehr stark von der Fehlerbeschreibung abhängig. Um hier den Benutzer weitgehend zu unterstützen, wurden verschiedene Methoden der Fehlerbeschreibung vorgestellt, und einige 'Standard-Fehlerbeschreibungen' implementiert. Mit der Fehlerbeschreibung durch 'Fuzzy-Intervalle' hat man die Möglichkeit, den zeitlichen Ablauf (der Sollwertfunktion) in die Fehlerbeschreibung einzubringen. Somit kann die Optimierung gezielt an kritischen Stellen verstärkt und ein Alternieren unterdrückt werden.

Das gesamte Entwicklungssystem sollte als interaktives Werkzeug zu Generierung eines Fuzzy-Reglers verstanden werden, da die implementierten Lernverfahren nicht in jedem Fall eine gute Lösung liefern (können). Der Benutzer sollte in solchen Fällen die Möglichkeit heranziehen, Vorwissen einfließen zu lassen. Auch die manuelle Nachbearbeitung der erlernten Wissensbasis, insbesondere der Regelbasis, sollte als Optimierungsmöglichkeit in Betracht gezogen werden. Ebenso kann das System zur Optimierung bestehender Regler eingesetzt werden.

Literatur

[BARTO et. al., 1983] Barto, A.G., Sutton R. S., Anderson, C. W. (1983): Neuronlike adaptive elements that can solve difficult learning control problems, IEEE Transactions on Systems, Man and Cybernetics, 13:834-846

[BARTO, 1992] Barto, A.G. (1992): Reinforcement Learning and Adaptive Critic Methods, In [WHITE und SOFGE, 1992], Seiten 469-491

[BERENJI und KHEDKAR, 1993] Berenji, H.R.; Khedkar, P. (1993): Clustering in Product Space for Fuzzy Inference, In Proc. IEEE Int. Conf. on Fuzzy Systems 1993, S. 1402-1407, San Francisco

[KNAPPE, 1994] Knappe, Heiko (1994): Comparison of Conventional and Fuzzy-Control of Non-Linear Systems, in [KRUSE et. al. 1994] S. 75-87

[KRUSE et. al. 1994] Kruse, Rudolf; Gebhardt, Jörg; Palm, Rainer (Eds.) (1994): Fuzzy Systems in Computer Science, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, Wiesbaden

[KRUSE et. al., 1995] Kruse, Rudolf; Gebhardt, Jörg; Klawonn, Frank (1995): Fuzzy-Systeme, 2. Auflage, B. G. Teubner, Stuttgart

[LEE, 1990] Lee, C.C. (1990): Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II, IEEE Transactions on Systems, Man and Cybernetics, 20: 419-435

[LEONHARD, 1992] Leonhard, Werner (1992): Einführung in die Regelungstechnik, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, Wiesbaden

[MAMDANI und ASSILIAN, 1975] Mamdani, E. H.; Assilian S. (1973): An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, International Journal of Man-Machine Studies, 7:1-13

[NAUCK et. al., 1996] Nauck, Detlef; Klawonn, Frank; Kruse, Rudolf (1996): Neuronale Netze und Fuzzy-Systeme, 2. Auflage, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, Wiesbaden

[NAUCK et. al., 1995] Nauck, Detlef; Kruse, Rudolf; Stellmach, Roland (1995): New Learning Algorithms for the Neuro-Fuzzy Environment NEFCON-I, In Proceedings of Neuro-Fuzzy-Systeme'95, 357-364, Darmstadt

[NÜRNBERGER, 1996] Nürnberger, Andreas (1996): Entwurf und Implementierung des Neuro-Fuzzy-Modells NEFCON zur Realisierung Neuronaler Fuzzy-Regler unter MATLAB/SIMULINK, Diplomarbeit, Technische Universität Braunschweig, www: http://fuzzy.cs.uni-magdeburg.de/nefcon.

[TSUKAMOTO, 1979] Tsukamoto, Y. (1979): An Approach to Fuzzy Reasoning Method, In M. Gupta, R. Ragade und R. Yager, Hrsg.: Advances in Fuzzy Set Theory, North-Holland, Amsterdam

[WERBOS, 1992] Werbos, P. J. (1992): Neurocontrol: Where it is Going and Why it is Crucial, Artificial Neural Networks, 2:61-68

[WHITE und SOFGE, 1992] White, D. A., Sofge, D. A., Hrsg. (1992): Handbook of Intelligent Control. Neural, Fuzzy and Adaptive Approaches, Van Nostrand Reinhold, New York