

Fuzzy Systeme automatisch entwerfen *

Hartmut Surmann

Die Verwendung von regelbasierten Fuzzy-Systemen hat in den vergangenen Jahren im Bereich der Automatisierungstechnik interessante Erfolge verzeichnet. In vielen Haushaltsgeräten (Staubsauger, Waschmaschine, usw.), der Konsumerelektronik (Fotoapparat, Cam- u. Videorecorder, Akkuladegeräte, usw.), der Medizintechnik und vielen mehr übernehmen Komponenten, die auf Fuzzy-Regeln beruhen, die Steuerung und Regelung von Prozessen [1]. Diese Systeme bieten aufgrund der Möglichkeit, unpräzise Informationen mittels unscharfer Mengen und Regeln zu beschreiben, eine Methode zur Beschreibung komplexer, unvollständig modellierbarer Systeme mit großer Parameterstreuung. Die Regeln beschreiben dabei den Zusammenhang zwischen Ursache und Wirkung in Form von „Wenn-Dann-Regeln“. Verbunden mit einer Entwicklungsumgebung ergeben sie eine Entwurfsmethode zur Modellierung unpräziser, nichtlinearer Systeme.

Gerade in technischen Disziplinen ist es eine sehr natürliche Vorgehensweise des Menschen, von ihm beobachtete Prozesse in Form von sprachlichen Regeln zu erklären und zu beschreiben (s.u.). Sie stellt den Versuch dar, komplexe Vorgänge durch Abstraktion von weniger relevanten Aspekten und unnötigem Ballast an Details zu modellieren und in überschaubaren Kategorien Schlussfolgerungen zu treffen. Schon 1965 erkannte der Systemtheoretiker L.A. Zadeh, daß mit steigender Systemkomplexität die Fähigkeit des Menschen sinkt, präzise und signifikante Aussagen über das Verhalten des Systems zu machen. Er definierte deshalb die Theorie der unscharfen Mengen¹ [2], um unpräzise Angaben zu quantifizieren. Unscharfe Mengen bilden die Verbindung zwischen der numerischen Auswertung im Rechensystem und der linguistischen Ausdrucksweise des Menschen. Als dann 1974 Mamdani, aufbauend auf den unscharfen Mengen, eine regelbasierte Methode² [3] vorstellte, um mit Hilfe von einfachen Regeln eine Labor-Dampfmaschine zu steuern, war das die Geburtsstunde der sogenannten „unscharfen regelbasierten Systeme“³ oder auch „regelbasierten Fuzzy-Systeme (RFS)“.

Neben ihrer Komplexität sind viele Anwendungen dadurch gekennzeichnet, daß Verhaltensweisen unbekannt, und damit nicht quantifizierbar sind oder daß sie einen sehr subjektiven Charakter besitzen, d.h. Komplexe Systeme können in einem Fuzzy-System durch die gleichzeitige Auswertung aller meßbaren Informationen und einer entsprechenden Plausibilitätsprüfung regelbasiert modelliert werden, während eine subjektive Charakterisierung durch die Gewichtung der einzelnen Aussagen erfolgt.

Aufgrund ihrer regelbasierten Struktur werden Fuzzy-Systeme oft im Bereich der Expertensysteme eingeordnet, obwohl sie weder eine Tiefen- oder Breitensuchstrategie für die Regeln noch eine dort übliche Trennung von Axiomen und Ableitungsregeln verwenden. Neben diesem Mißverständnis gibt es noch eine Reihe weiterer, die zum Teil durch das Wort „fuzzy“ entstanden sind:

- Regelbasierte Fuzzy-Systeme sind keine ungenauen bzw. nicht deterministischen Verfahren. Ihre Grundlagen sind beispielsweise in [4] mathematisch exakt dargestellt.
- Eine Fuzzy-Menge ist keine Wahrscheinlichkeitsdichte. Wahrscheinlichkeitsverteilungen werden durch eine affine Abbildung die Fläche beispielsweise eines Histogramms auf die maximale Wahrscheinlichkeit 1 normiert, also $\int_{-\infty}^{\infty} h(x)dx = 1$. Für die Betrachtung als Zugehörigkeitsfunktionen wird dagegen die Höhe $h_{\max} = \sup_x h(x)$ auf 1 normiert. Trotz einer gewissen formalen Ähnlichkeit ergeben sich dadurch erhebliche semantische Unterschiede [5, Seite 137].
- Sie verwenden in regelungstechnischen Anwendungen diskrete Ein- und Ausgabewerte.
- Regelbasierte Fuzzy-Systeme erzielen formal gesehen keine Ergebnisse, die nicht auch mit anderen Verfahren erreicht werden können. Dies ist eigentlich keine große Überraschung, da jede prinzipiell berechenbare Funktion bereits durch eine Turing-Maschine berechnet werden kann. Das eigentliche Problem ist allerdings, wie beschreibt man eine Funktion möglichst schnell und gut nachvollziehbar.

Ein Ziel bei der Realisierung von Anwendungen ist es deshalb, mit einfachen Mitteln und Werkzeugen möglichst komplexe Systeme nicht allein unter wissenschaftlichen, sondern auch unter marktwirtschaftlichen Randbedingungen zu entwerfen.

Dr.-Ing. Hartmut Surmann hat in Dortmund Informatik und Elektrotechnik studiert und dort im Fachbereich Elektrotechnik promoviert. Seit 1994 ist er als Postdoc bei der GMD in Sankt Augustin mit der Entwicklung von fuzzy-gesteuerten autonomen mobilen Robotern beschäftigt.



*Dieser Artikel ist ein Auszug aus [7]

¹engl. *fuzzy set theory*

²im Original *fuzzy algorithm*

³Im Engl. werden unterschiedliche Bezeichnungen, wie *fuzzy controller*, *fuzzy logic controller* oder *fuzzy rule-based system* verwendet.

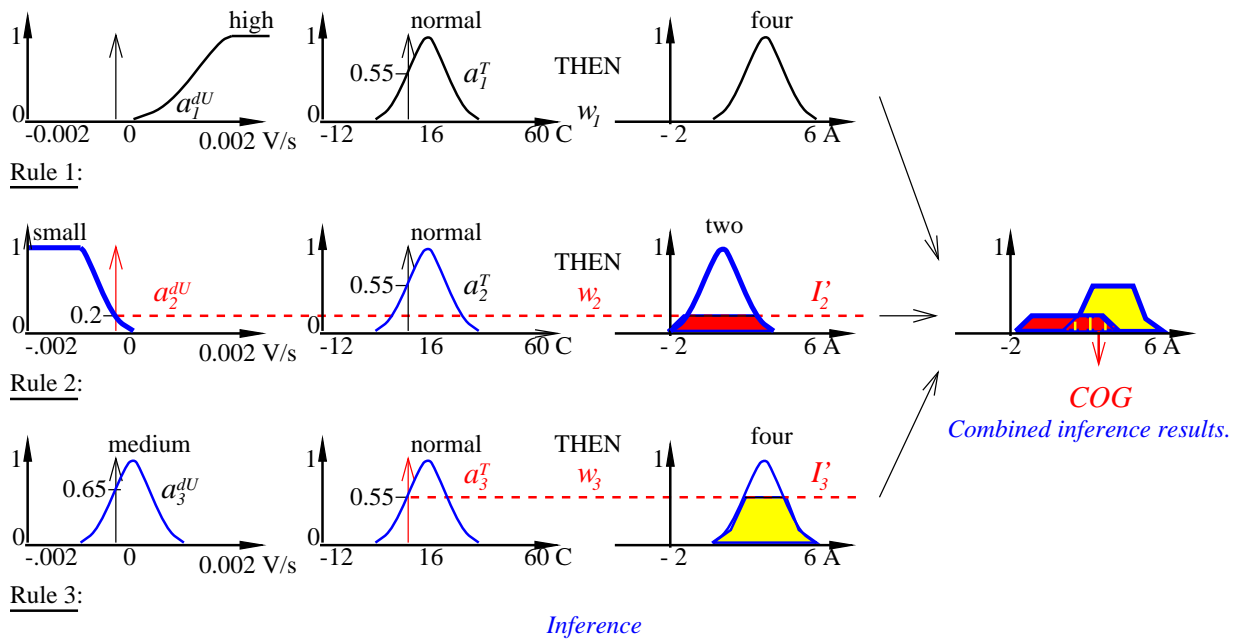


Bild 1: Algorithmus eines RFS mit zwei Eingangs- sowie einer Ausgangsgröße und drei Regeln für ein Akkuladegerät

1 Der Algorithmus

Der Algorithmus eines RFS basiert auf einer Verallgemeinerung des Modus Ponens [6]:

Prämisse:	A is true
Implikation:	If A then B
Konklusion:	B is true

Fuzzy-Mengen ersetzen dabei die binären Prämissen A und B. Fuzzy-Mengen sind charakterisierbar über Funktionen $\mu_i : U \rightarrow [0, 1]$ über einem Universum U mit $x \mapsto \mu_i(x)$, $i = \{(x, \mu_i(x)) | x \in U, \mu_i(x)\}$, den sog. Zugehörigkeitsfunktionen. In den Anwendungen erhalten die Zugehörigkeitsfunktionen einen sprachlichen Bezeichner (meist Adjektive oder Adverbien), wodurch sich eine gewisse Nähe zur menschlichen Ausdrucksweise ergibt. Zadeh [2] hat für Fuzzy-Mengen drei wichtige Operationen eingeführt:

$$\begin{aligned}
 &\text{Intersection } C = A \cap B, \\
 &\mu_C(x) = \min(\mu_A(x), \mu_B(x)) \\
 &\text{Union } C = A \cup B, \\
 &\mu_C(x) = \max(\mu_A(x), \mu_B(x)) \\
 &\text{Komplement } \bar{A}, \\
 &\mu_{\bar{A}}(x) = 1 - \mu_A(x)
 \end{aligned}$$

Diese fundamentalen Operatoren bilden über der Menge $[0, 1]$ eine sog. Fuzzy-Algebra.

Das Beispiel in Bild 1 zeigt den Basisalgorithmus für die folgenden drei Fuzzy-Regeln eines Akkuladegerätes [7]:

- R1: IF dU is high and T is normal THEN I is four
R2: IF dU is small and T is normal THEN I is two
R3: IF dU is medium and T is normal THEN I is four

Die Regeln haben $n = 2$ Eingänge (Spannungableitung und Temperatur) und $m = 1$ Ausgang (Strom). Die Eingangssignale werden in den Teilprämissen gleichzeitig ausgewertet. $\alpha_1^{dU} = 0.0$ bedeutet, daß diese Regel keinen Beitrag zur Ausgangsgröße liefert. $\alpha_1^{dU} = 0.2$ und $\alpha_1^T = 0.55$ in Regel 2 ergeben einen Wahrheitswert von $\omega_2 = \min(\alpha_1^{dU}, \alpha_1^T) = 0.2$. Die Konklusion einer Regel ergibt sich aus:

$$I'_i = \{\min(\omega_i, I_i(x)) \mid x \in I_i\}, \text{ für } i = 1 \dots 3 \quad (1)$$

und wird mit den jeweils anderen aktivierten ($\omega_i > 0$) Regeln vereinigt:

$$I' = \bigcup I'_i, \text{ für } i = 1 \dots 3^4. \quad (2)$$

In den meisten Anwendungen wird aus diesen Fuzzy-Mengen dann wieder ein binärer Ausgangswert beispielsweise durch die Schwerpunktmethode ermittelt:

$$COG_I = \frac{\int x \times I'(x) dx}{\int I'(x) dx} \quad (3)$$

1.1 Was leistet dieser Algorithmus

Formaler ausgedrückt bildet das 6-Tupel $(\mu_{ab}, R, T, I, T - CO, DE)$ eine Familie von Fuzzy-Systemen mit den Zugehörigkeitsfunktionen μ , der Fuzzy-Regelbasis R , einem Konjunktionsoperator T , der die $T - Norm$ ⁵ Eigenschaft erfüllt, einer Implikation I mit der Eigenschaft $I(a, 0) = 0$ if $a \neq 0$ (z.B. R-Implikation oder T-Norm), einer T-Co-Norm T-CO und einer Defuzzifizierungsmethode DE (z.B. Schwerpunkt oder Maximum). Derartige Systeme beschreiben eine partielle, n-dimensionale, nicht-lineare und dynamikfreie

⁴compositional rule of inference [3]

⁵Eine kommutative und assoziative, zweistellige Abbildung $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$, mit $\forall a, b, c, d \in A : 1. T(0, 0) = 0, T(a, 1) = T(1, a) = a$ und $2. T(a, b) \leq T(c, d)$ wenn $a \leq c$ und $b \leq d$.

Funktion $f : U \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$. Dabei entstehen komplexere Funktionen $m > 1$ durch das kartesische Produkt von Funktionen $f : U \rightarrow \mathbb{R}$.

Satz 1 Sei $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ eine kontinuierliche, reelle Funktion auf einer kompakten Menge U . Falls I eine T -Norm ist, so existiert für alle $\epsilon > 0$ ein RFS = $(\mu_{ab}, R, T, I, T - CO, DE)$, so daß

$$\sup\{|f(\vec{x}) - RFS(\vec{x})| \mid \forall \vec{x} \in U\} \leq \epsilon.$$

Beweis siehe [8].

Damit sind derartige Fuzzy-Systeme universelle Funktionsapproximatoren.

2 Automatischer Entwurf

Von den 6 möglichen Freiheitsgraden sind die Zugehörigkeitsfunktionen und die Fuzzy-Regeln wesentlich für den automatischen Entwurf. Aufgabe eines automatischen Systems ist nun:

- ein RFS komplett zu entwerfen, d.h. Daten in Form von Faustregeln zusammenzufassen oder
- ein bestehendes System anzupassen oder zu erweitern, d.h. die Justierung von Parametern (z.B. Zugehörigkeitsfunktionen) bis hin zu Spezialisierung allgemeinen Wissens an konkrete Umgebungsbedingungen. Dabei kann zusätzlich Hintergrundwissen in Form von Fuzzy-Regeln eingebracht werden.

Dazu sei RD die Menge der Referenzvektoren $\vec{r}_i \in \mathbb{R}^n$, $i = 1 \dots p = |RD|$, die als Gesamtheit das gewünschte Verhalten des Systems charakterisieren. Entsprechende Vektoren erhält man beispielsweise aus den Sensorwerten bei der Beobachtung eines Experten. Außerdem wird im Referenzdatensatz durch die Referenzvektoren die Anzahl der Eingangs- und der Ausgangsgrößen vorgegeben. Falls die Vektoren in Klassen unterteilt sind, ist damit auch die Anzahl der Klassen angegeben. Der Datensatz sollte nicht nur positive, sondern auch negative Beispiele enthalten.

Insgesamt läßt sich das Entwurfsproblem somit als ein Optimierungsproblem ansehen. Bei der Auswahl eines Entwurfsverfahrens beispielsweise mittels statistischer Verfahren, neuronaler Netze oder simulated annealing / genetischer Algorithmen ist dabei nur die Komplexität des zu entwerfenden Problems entscheidend. Kleinere Systeme lassen sich am schnellsten mit gradienten Abstiegsverfahren lösen. Komplexere Fuzzy-Systeme benötigen dagegen Strategien, die auch lokale Minima wieder verlassen können, denn insgesamt gehört das Entwurfsproblem zur Klasse der NP-vollständigen Probleme.

Um nun eine RFS zu entwerfen muß zuerst ein initial System generiert werden. Dafür wird eine Strukturanalyse der Daten, beispielsweise mittels statistischer Berechnungen oder mittels eines Selbstorganisationsvorgangs, durchgeführt [7]. Dadurch erhält man einen Schätzer für die Zugehörigkeitsfunktionen. Die Zugehörigkeitsfunktionen teilen dabei die

Achsen des n -dimensionalen Raumes in kleinere Abschnitte auf, ähnlich der Vorstellung von kleinen Würfeln in einem großen Würfel. Die Überlappung der Zugehörigkeitsfunktionen ermöglichen in den Fuzzy-Regeln einen sanften Übergang von einem Teilwürfel in den nächsten, wenn sich eine Eingangsgröße kontinuierlich verändert. Zur Generierung der Zugehörigkeitsfunktionen wird in erster Näherung angenommen, daß die Werte im Referenzdatensatz normalverteilt sind. Die Zugehörigkeitsfunktionen werden dann aus einer $N(0,1)$ Verteilung generiert indem die maximale Höhe auf 1 normiert und die Funktion bei $2.9 * \sigma$ beschränkt wird:

$$\mu(x; m, \sigma^2) := \begin{cases} \sigma\sqrt{2\pi} \varphi(x; m, \sigma^2), & , x \leq |2.9\sigma| \\ 0 & , \text{sonst} \end{cases} \quad (4)$$

Dreieckförmige Funktionen sind demnach eine Approximation dieser Zugehörigkeitsfunktionen durch jeweils 2 Geraden. Zur Generierung der Regelbasis wird eine Regel für einen Referenzvektor generiert durch die konjunktive Verknüpfung der Zugehörigkeitsfunktionen jeder Eingangsvariablen, in welchem der Referenzvektor liegt. Gleich Regeln werden dabei aus der Regelbasis entfernt. Damit werden die linguistischen Variablen extrahiert und die Regelbasis vollständig initialisiert.

Für größere RFS muß die Regelbasis dabei in ihrer Komplexität beschränkt werden. Die dabei entstehende unvollständige Regelbasis überdeckt zwar den gesamten Referenzdatensatz, nicht aber alle möglichen Kombinationen von Ein- und Ausgabewerten.

Im zweiten Schritt wird dann, ebenfalls auf der Basis der Referenzdaten und einer zusätzlichen Bewertungsfunktion Q , das erzeugte RFS optimiert, und zwar mit Hilfe eines genetischen Algorithmus. Dieser verbessert die Schätzer für m und σ der jeweiligen Zugehörigkeitsfunktion. Von besonderem Interesse ist hier die Bewertungsfunktion Q . Für die Anpassung des Ein- / Ausgangsverhaltens eignen sich i.A. metrische Distanzen als grundlegende Bewertungsfunktion q . Analysiert man die internen Zustände eines automatisch entworfenen Fuzzy-Systems, so zeigt sich, daß analog zu Entwürfen mit neuronalen Netzen sehr viele Regeln einen Beitrag zur Ausgangsgröße beitragen. Dieses entspricht allerdings nicht dem Entwurf und den Vorstellungen eines menschlichen Experten, der die Fuzzy-Regeln mehr nach einem lokalen Kriterium entwirft. Weiterhin verschlechtern viele aktivierte Regeln nicht nur die Wartbarkeit und Robustheit, sondern auch das Laufzeitverhalten [9, 7]. Damit können Beschränkungen in der Hardware bezüglich der maximalen Anzahl von aktivierten Regeln, wie sie in ASICs vorgegeben sind [10], nicht berücksichtigt werden.

2.1 Unschärfemaß für Fuzzy-Regeln

Wie schon angedeutet, ist eine wesentliche Idee bei der Beschreibung eines Funktionals durch ein RFS die Berücksichtigung von Nachbarschaftsbeziehungen im Zustandsraum. Die Beschreibung dieser Nachbarschaftsbeziehungen durch lokale Fuzzy-Regeln sollte auch beim Entwurf durch automatische Algorithmen berücksichtigt werden, damit die Vorteile eines Fuzzy-Systems, wie die Robustheit, Verstehbarkeit und Wartbarkeit, erhalten bleiben.

Eine Verbesserung des Entwurfs bezüglich dieser Eigenschaft kann durch die Spezifikation entsprechender Gütefunktionen erreicht werden. Dazu wird die *Unschärfe* oder *Entropie* eines Fuzzy-Systems, also der *Grad der Unordnung* eines Regelsatzes, im informationstechnischen Sinne nach Shannon durch die mittlere Anzahl der aktivierten Regeln definiert:

$$R_\phi = \frac{\sum_{i=1}^p R_{act,i}}{p} \quad (5)$$

mit $R_{act,i}$ = Anzahl der aktivierten Regeln bei Eingabe des i -ten Referenzvektors.

Ein RFS, in dem für einen Referenzvektor $\vec{r}_i \in \mathbb{R}^n$, $i = 1 \dots p$ viele Regeln aktiviert werden, enthält wenig Information, d.h. das Fuzzy-System ist in einem sehr ungeordneten Zustand und für einen Menschen relativ unverständlich. Die meiste Information enthält in diesem Sinne ein binäres System, in dem jeweils immer genau eine einzige Regel aktiv ist. Mehrere Regeln werden durch die Überlappung der Zugehörigkeitsfunktionen aktiviert. Ein RFS mit hoher Entropie enthält damit Zugehörigkeitsfunktionen, die sich stark überlappen. Um nun die Entropie eines Fuzzy-Systems und damit den Grad der Überlappung der Zugehörigkeitsfunktionen zu reduzieren, wird eine maximale Anzahl von aktivierten Regeln $R_{max} \in \mathbb{N}$ (maximale Entropie) vorgegeben und die Gütefunktion Q modifiziert durch:

$$Q = \begin{cases} \frac{q}{\left(\frac{R_{act}}{R_{max}} - 1\right)a + \left(\frac{R_\phi}{R_{max}} - 1\right)b + 1} & , \text{ für } 0 < R_{max} < R_{act} \\ \frac{\frac{R_\phi}{R_{max}} q}{\left(\frac{R_\phi}{R_{max}} - 1\right)b + 1} & , \text{ für } 0 < R_{max} \end{cases} \quad (6)$$

mit $a, b \in \mathbb{R}$.

Existieren Hardwarebeschränkungen, so „bestraft“ der lokal wirkende Faktor $a > 0$ ein Fuzzy-System, welches zu viele Regeln für einen einzelnen Eingabevektor aktiviert. Der Faktor $b > 0$ hingegen wirkt globaler und bestraft ein System, das im Mittel mehr Regeln aktiviert, also ungeordneter ist als ein anderes. Da aus einem Entropiemaß ein Energiemaß und umgekehrt aus einem Energiemaß ein Entropiemaß hergeleitet werden kann [4], ist auch eine Interpretation als Energie möglich.

Bild 2 zeigt ein Beispiel, wie sich die Entropiereduktion beim automatischen Entwurf auf die Anzahl der aktivierten Regeln auswirkt. Gleichzeitig werden auch zu viel erzeugte Zugehörigkeitsfunktionen und damit Fuzzy-Regeln eliminiert [7].

3 Ausblick

Derzeit wird eine Erweiterung eines RFS für ein hochkomplexes System bestehend aus einem autonomen mobilen Roboter (MORIA [11]) getestet. Die Steuerung eines solchen Roboters umfaßt derzeit über 300 Regeln mit 30 Eingängen, 15 Ausgängen und ca. 15 internen Variablen.

Literatur

[1] H.-J. Zimmermann /C. v. Altrock (Hrsg.). *Fuzzy Logic*, Band 2, Anwendungen. Oldenbourg-Verlag, München, Wi-

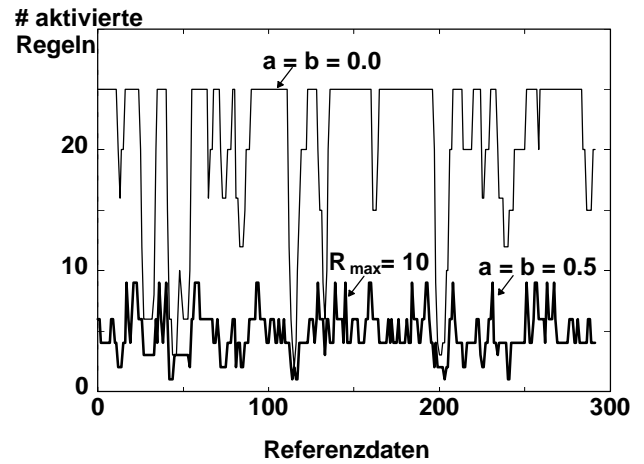


Bild 2: Aktivierte Regeln zweier optimierter Fuzzy-Systeme mit und ohne Entropiereduktion

en, Oldenbourg, 1994. 2. Fuzzy-Praxistagung, Augsburg, 16./17. März 1993.

[2] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8:338 – 353, 1965.

[3] E. H. Mamdani. Application of fuzzy algorithms for control of a simple dynamic plant. *Proc. IEE*, 121, Nr. 12:1585 – 1588, 1974.

[4] Hans Bandemer and Siegfried Gottwald. *Einführung in Fuzzy-Methoden*. Verlag Harri Deutsch, Thun, Frankfurt am Main, 1990.

[5] Didier Dubois and Henri Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, Inc., Boston, 1980.

[6] H. Watanabe, W. Dettloff, and K. Yount. A VLSI Fuzzy-Logic-Controller with Reconfigurable, Cascadable Architecture. In *IEEE Journal of Solid-State Circuits*, Band 25, Nr. 2, pages 376 – 382, 12/1990.

[7] Hartmut Surmann. *Automatischer Entwurf von Fuzzy Systemen*. Dissertation, Universität Dortmund, Fakultät für Elektrotechnik, VDI-Verlag Düsseldorf, ISBN 3-18-345208-1, 1995.

[8] J.L. Castro. *Fuzzy logic controllers are universal approximators*. University of Grenada, Department of Computer Science and Artificial Intelligence, Technical Report #DECSAI-93101, 6/1993.

[9] Hartmut Surmann and Ansgar P. Ungering. Fuzzy-rule-based systems on general purpose processors. *IEEE MICRO, Special issue on fuzzy systems*, pages 40–48, August 1995.

[10] Ansgar Ungering and Karl Goser. Architecture of a PDM VLSI Fuzzy Logic Controller with Pipeline and Optimized Chip-Area. In *Second IEEE International Conference on Fuzzy Systems, San Francisco*, pages 447 – 452, 28.3. - 1.4.1993.

[11] Hartmut Surmann, Jörg Huser, and Liliane Peters. A fuzzy system for indoor mobile robot navigation. In *Fourth IEEE International Conference on Fuzzy Systems, Yokohama, Japan*, pages 83–88, 20–24 March 1995. Distinguished with *Robot Intelligence Award*.