

Automatisches Entwicklungsverfahren für intelligente B-Spline Fuzzy-Regler

J. Zhang, I. Renners und A. Knoll
AG Technische Informatik, Technische Fakultät
Universität Bielefeld, D-33501 Bielefeld
Tel.: ++49 521 106-2951, Fax: ++49 521 106-6440
E-mail: zhang@techfak.uni-bielefeld.de

Zusammenfassung

In vielen technische Anwendungen sind adaptive Regler nützlich, die automatisch aus Trainingsdaten konstruiert werden können, und deren aus dem Lernvorgang erzeugte Struktur einer Interpretation noch zugänglich ist. B-Spline Fuzzy Regler (BFRs) haben diese Eigenschaften, sind aber nur für Probleme niedriger Dimension geeignet. Um auch Probleme höherer Dimensionalität bearbeiten zu können, eine höhere Transparenz sowie eine gute Generalisierungsleistung zu muß eine genaue Anpassung des Reglers an das Problem vorgenommen werden. In dem hier vorgestellten Ansatz wird ein genetischer Algorithmus verwendet, der die Struktur eines BFRs verändert, so daß ein Problem aus dem Bereich des visuellen Lernens gelöst werden kann.

Stichworte: Fuzzy-Regler, B-Spline, Genetische Algorithmen , Adaptiv, Generalisierung, Interpretierbarkeit.

1 Konstruktion von Fuzzy-Systemen mit B-Splines

1.1 Einführung

Die Verwendung von B-Splines als Zugehörigkeitsfunktionen in einem Fuzzy-System ist eine effektive Methode zur Funktionsapproximation und zur Konstruktion von Fuzzy-Reglern. Anstelle der allgemein üblichen Funktionen (wie Dreiecks- oder Gaußfunktion) werden hier zur Modellierung von linguistischen Termen B-Spline-Basisfunktionen verwendet. Für deren Berechnung fanden [deBoor1972] und [Cox1972] eine effiziente und numerisch stabile Rekursionsvorschrift.

1.2 B-Splines als Basisfunktionen

Sei x eine auf dem Wertebereich $[a, b]$ definierte Eingangsgröße eines Reglers. Weiterhin sei $N_{i,k}(x)$ die i^{te} normalisierte B-Spline-Basisfunktion der Ordnung k über dem Knotenintervall $[a, b]$ mit $a = \lambda_0 \leq \dots \leq \lambda_{i+k+1} = b$. Wenn $a = b$, dann ist $N_{i,k} = 0$, andernfalls werden die normalisierten B-Splines rekursiv definiert durch:

$$N_{i,k}(x) = \frac{x - \lambda_{i-k}}{\lambda_{i-1} - \lambda_{i-k}} N_{i-1,k-1}(x) + \frac{\lambda_i - x}{\lambda_i - \lambda_{i-k+1}} N_{i,k-1}(x) \quad (1)$$

$$N_{i,1}(x) = \begin{cases} 1 & \text{wenn } x \in [\lambda_i, \lambda_{i+1}) \\ 0 & \text{sonst} \end{cases}$$

mit $i = 0, 1, \dots, m - k$. Abbildung 1(a-c) zeigt B-Spline-Basisfunktionen verschiedener Ordnung, welche die Rolle von Zugehörigkeitsfunktionen übernehmen.

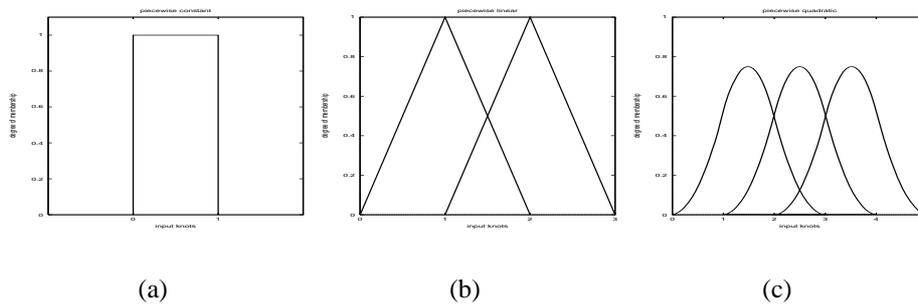


Abbildung 1: Definition von Fuzzy Mengen mittels B-Spline-Basisfunktionen verschiedener Ordnung: (a) $k=1$ (b) $k=2$ (c) $k=3$

1.3 Funktionsinterpolation mit B-Splines

Wir zeigen nun, daß Mehrgrößen-B-Spline-Regler sehr gute Funktionsapproximationen vornehmen können. Da bei solchen Reglern die MIMO(multiple-input–multiple-output)-Regelbasen in mehrere MISO(multiple-input–single-output)-Regelbasen zerlegt werden können, betrachten wir nur den letzteren Fall. In [ZK98a] wurde gezeigt, daß unter den folgenden Bedingungen die Ausgangsgröße y einer allgemeinen B-Spline-Hyperfläche entspricht:

- Die Zugehörigkeitsfunktionen der Eingangsgrößen sind B-Spline-Basisfunktionen;
- Die Zugehörigkeitsfunktionen der Ausgänge sind Fuzzy Singletons;
- Als Fuzzy-Verknüpfung wird das Produkt verwendet;
- Die Schwerpunktmethod dient zur Defuzzifizierung;
- An den Intervallgrenzen jeder linguistischen Variablen werden „Virtuelle linguistische Terme“ hinzugefügt;

Im allgemeinen Fall betrachten wir hier ein MISO-System mit n Eingängen x_1, x_2, \dots, x_n , einer Regelbasis mit n konjunktiven Termen in der Prämisse jeder Regel, die folgende Form annimmt:

WENN (x_1 ist N_{i_1, k_1}^1) DANN (x_2 UND N_{i_2, k_2}^2) UND ... UND (x_n ist N_{i_n, k_n}^n) DANN y ist $c_{i_1 i_2 \dots i_n}$
wobei

- x_j : der j -te Eingang ($j = 1, \dots, n$),
- k_j : die Ordnung der für x_j benutzten B-Spline Funktion,
- N_{i_j, k_j}^j : der i -te linguistische Term von x_j (definiert durch B-Spline Basis Funktionen),
- $i_j = 1, \dots, m_j$ bestimmt, wie fein der j -te Eingang partitioniert wird,
- $c_{i_1 i_2 \dots i_n}$: die Kontrollpunkte (auch Koeffizienten oder de-Boor-Punkte genannt) [BFK84] des Ausganges, wenn die Regel $Rule(i_1, i_2, \dots, i_n)$ mit Sicherheit feuert.

Dann wird der Ausgang y des MISO Fuzzy-Reglers [ZK98a]:

$$\begin{aligned}
 y &= \frac{\sum_{i_1=1}^{m_1} \cdots \sum_{i_n=1}^{m_n} (c_{i_1, \dots, i_n} \prod_{j=1}^n N_{i_j, k_j}^j(x_j))}{\sum_{i_1=1}^{m_1} \cdots \sum_{i_n=1}^{m_n} \prod_{j=1}^n N_{i_j, k_j}^j(x_j)} \\
 &= \sum_{i_1=1}^{m_1} \cdots \sum_{i_n=1}^{m_n} (c_{i_1, \dots, i_n} \prod_{j=1}^n N_{i_j, k_j}^j(x_j))
 \end{aligned} \tag{2}$$

Dies stellt ein multivariablen B-Spline Modell dar, welches eine allgemeine NUBS (nonuniform B-Spline)-Hyperfläche approximiert.

1.4 Anpassung der Kontrollpunkte

Die Anzahl der aktivierten Zugehörigkeitsfunktionen eines jeden Einganges entspricht der Ordnung der verwendeten B-Splines. Dies resultiert bei einer lokalen Änderung der Regelbasis in einer ebenfalls nur lokalen Änderung des Ausgangsverhaltens des Reglers. Damit besteht die Möglichkeit, die Regelbasis des Reglers *online* an neue Gegebenheiten anzupassen.

Die Regelbasis des B-Spline Fuzzy-Reglers kann entweder durch direktes Lösen eines überbestimmten linearen Gleichungssystems [Dierckx1993] oder aber durch Gradientenabstieg (*supervised or unsupervised online learning*) entlang des n -dimensionalen Fehlergebirges bestimmt werden.

Beim Gradientenabstieg erhalten wir das gewünschte Ergebnis, wenn wir in der üblichen Weise einen Satz von Trainingsdaten $\{(\mathbf{X}, y_d)\}$ vorgeben, wobei $\mathbf{X} = (x_1, x_2, \dots, x_n)$ jeweils ein Trainingsdatenvektor und y_d der zugehörige, gewünschte Ausgangswert für \mathbf{X} ist. Wir verwenden eine quadratische Fehlerfunktion: $E = \frac{1}{2}(y_r - y_d)^2$. In [ZK98a] wurde gezeigt, daß diese Fehlerfunktion über dem Raum c_{i_1, i_2, \dots, i_n} konvex ist und deshalb nur ein Minimum besitzt; das Verfahren also zur Konvergenz führt.

2 Anpassung von Fuzzy-Mengen

2.1 Motivation

Ein monolithischer BFR ist ein vollständig über einem n -dimensionalen Gitter definierter Regler, mit n Eingängen x_n und einem Ausgang y , und somit eine Unterklasse eines Assoziativ-Speichernetzwerkes (*Associative Memory Network*). Da die Regelbasen solcher gitterbasierter Regler exponentiell von der Dimension des Eingangsraumes abhängen, sind sie prinzipiell nur für niedrigdimensionale Probleme geeignet.

Eine Möglichkeit diesen *Fluch der Dimensionalität* zu reduzieren, ist eine (sub)optimale Anpassung der n -dimensionalen Gitterstruktur. Für BFRs geschieht eine solche Anpassung durch Veränderung der n Knotenvektoren, über denen die B-Splines definiert sind. Diese Anpassung führt nicht nur zu einer Reduzierung des Speicherbedarfs, sondern hat den noch wichtigeren Vorteil, daß die Generalisierungsleistung und die Interpretierbarkeit des Systems erhöht wird. Da durch variable Knotenpunkte die Berechnung der Regelbasis zu einem nichtlinearen Problem wird, verwenden wir (*offline*) einen genetischen Algorithmus (GA) zur Bestimmung der Knotenvektoren.

2.2 Ein Genetischer Algorithmus zur Knotenpunktoptimierung

Um eine Modellverbesserung eines BFRs durch Knotenpunktoptimierung zu realisieren, haben wir einen einfachen genetischen Algorithmus (GA) implementiert, in dem jedes *Individuum* aus n Knotenvektoren plus der zugehörigen Regelbasis besteht. Da die Struktur eines monolithischen BFRs nur

von den Knotenvektoren und der Ordnung der B-Splines abhängt, repräsentiert somit jedes Individuum einen untrainierten Regler. Die Ordnung der B-Splines, welche ebenfalls Einfluß auf die Struktur und Generalisierungsleistung des Systems hat, wird auf einen festen Wert gesetzt (in den Beispielen Ordnung 3).

Zuerst werden jedem Individuum der Population zufällige Knotenvektoren und eine mit Nullen initialisierte Regelbasis zugewiesen. Um nun die *Fitness* eines jeden Individuums zu ermitteln, lernt der Regler mit Hilfe des Gradientenabstiegsverfahren eine gewisse Anzahl von Lernschritten in die Tiefe. Danach wird der mittlere quadratische Fehler (MSE) dieses Individuums bestimmt und die Fitness gleich $\frac{1}{\text{MSE}}$ gesetzt. Die gesamte Population wird nun nach ihren Fitnesswerten sortiert. Die besten 50 Prozent (Individuen mit den höchsten Fitnesswerten) aller Individuen paaren sich und produzieren mittels *Uniform Crossover* neue Individuen, die den Platz der schlechteren Individuen einnehmen. Der Kind-Regler kopiert sich mit 50-prozentiger Wahrscheinlichkeit die Regelbasis des einen oder anderen Elternteils. Zum Schluß werden einzelne Bits der Knotenvektor-Strings der neuen Individuen mutiert, um auch in späteren Generationen eine Stagnation des Suchprozesses zu vermeiden. Der gesamte Prozeß wird solange wiederholt bis ein Abbruchkriterium erfüllt ist.

Es wurde Uniform Crossover ausgewählt, weil sich diese Rekombinationsmethode bei kleinen Populationen bewährt hat.

2.3 Die Genetische Kodierung eines BFRs

Um den oben beschriebenen Algorithmus anzuwenden, werden die Trainingsdaten jedes Einganges auf ein Intervall $[0, 1]$ normiert. Für jedes Individuum werden n Knotenvektoren mit jeweils 32 Knotenpunkten im Intervall $[0, 1]$ erzeugt, von denen $3 \leq i \leq 32$ aktiv sind. Welche Knotenpunkte aktiv sind, wird für jeden der n Knotenvektoren in einem 32 Bit langen Aktivierungsstring gespeichert, welcher ebenfalls durch Crossover und Mutation auf das Kind-Individuum vererbt wird. Knotenpunkte werden als 16 Bit Strings kodiert, so daß jeder Knotenpunkt einen Wert im Intervall $[0, 1]$ mit einer Genauigkeit von $\frac{1}{2^{16}}$ annehmen kann. Zur Verdeutlichung der Kodierung eines zweidimensionalen Reglers siehe Abbildung 2.

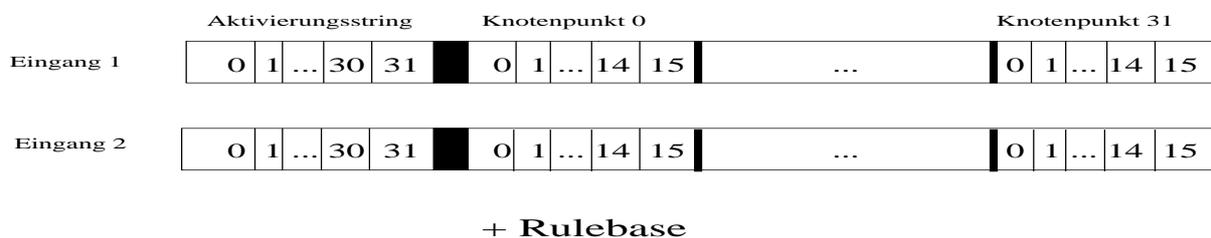


Abbildung 2: Die Kodierung eines zweidimensionalen Reglers

Mit Hilfe der Aktivierungsstrings eines jeden Knotenvektors kann die Anzahl der Basisfunktionen des n -ten Einganges beeinflusst werden. Wenn ein Individuum mit weniger Basisfunktionen (also weniger aktiven Knotenpunkten) einen MSE der gleichen Größenordnung wie ein Individuum mit mehr aktiven Knotenpunkten erzeugt, ist die Wahrscheinlichkeit zur Reproduktion des kleineren Reglers größer. Dies wird durch einen Strafterm erreicht, der den Fitnesswert von Individuen mit vielen Knotenpunkten senkt.

2.4 Ergebnisse von Funktionsapproximation

Es wurden die Funktionen f_1, \dots, f_6 sowie g_1, g_2, g_3 [MK96] mit Hilfe eines BFRs angenähert. Die durchschnittlichen quadratischen Restfehler der Regler mit äquidistant verteilten B-Spline-Basisfunktionen (E-BSBF), der Regler mit GAs modifizierten B-Spline-Basisfunktionen (GA-BFBS) so-

wie die besten Modelle aus [MK96] sind in Tabelle 1 gegenübergestellt. Zur besseren Veranschaulichung sind zwei dieser Funktionen sowie die zugehörigen genetisch optimierten Regler in Abbildung 3 zu sehen.

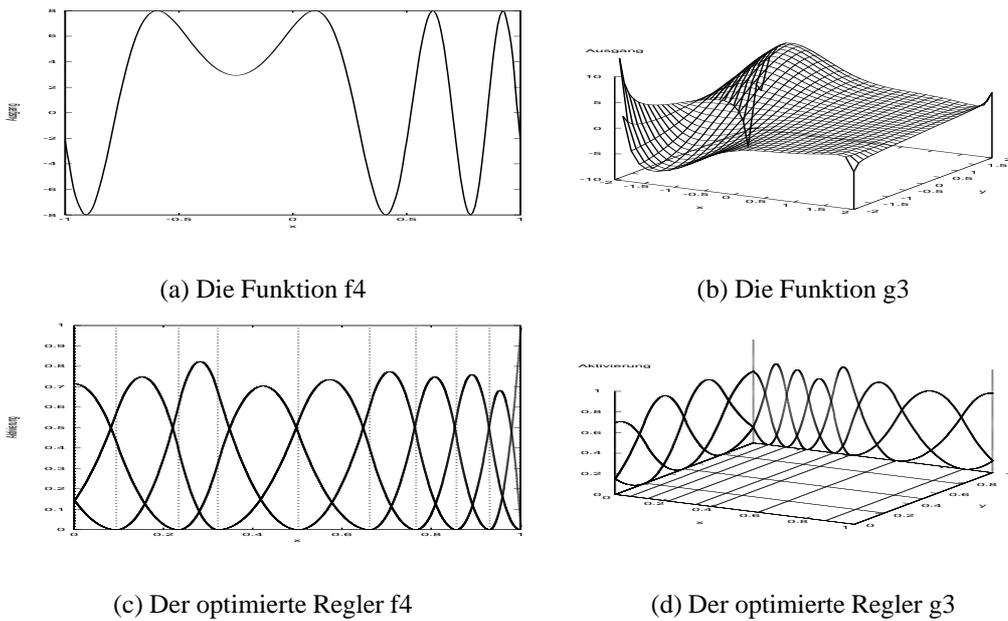


Abbildung 3: Die eindimensionale Funktion f4 (a) und die zweidimensionale Funktion g3 (b) sowie die für das Problem optimierten Regler(c) und (d)

Funktion	Regeln	verwendete Zugehörigkeitsfunktion			
		E-BSBF	GA-BSBF	Besten aus [MK96]	Schlechtesten aus [MK96]
f_1	12	0.02	0.007	0.08	0.7
f_2	12	0.0005	0.00003	0.02	0.3
f_3	12	0.0008	0.00006	0.002	0.03
f_4	12	4.9	0.03	0.1	10
f_5	12	0.04	0.0002	0.01	1
f_6	12	0.6	0.02	0.1	0.4

g_1	64	670.9	17.61 (60 Regeln)	9	26
g_2	64	14.6	5.21 (63 Regeln)	7	19
g_3	64	0.36	0.0165 (60 Regeln)	1.2	6

Tabelle 1: MSE Vergleich der jeweils besten Resultate aus [MK96] mit den Ergebnissen unter Verwendung von äquidistant verteilten B-Spline Basis Funktionen (E-BSBF) und B-Spline-Basisfunktionen die mittels des GAs modifiziert wurden(GA-BFBS).

3 Anwendungen

Basierend auf dem adaptiven B-Spline Modell wurden überwachtes und unüberwachtes Lernen in den folgenden intelligenten Regelungen von mobilen Robotern und Roboterarmen realisiert:

3.1 On-Line Lernen

Schnelles Reinforcement-Lernen von sensorbasierten Operationen. Ein mobiler Roboter mit einfachen Abstandssensoren kann Verhalten wie Kollisionsvermeidung, Konturverfolgung, Zielfahrt usw. über nur wenige Lernschritte beherrschen [ZS98]. Ein Zweiarmsystem kann an Hand von Kraft-Moment-Sensoren kooperative Bewegungsregelung schnell lernen [ZF98].

3.2 Visuelles Lernen

Visuelles Lernen von Roboterpositionen, das auf effizienter Dimensionsreduktion und adaptiver Interpolation beruht. Durch Einsatz eines omnidirektionalen Sichtsystems kann ein mobiler Roboter seinen Standort über die Erkennung natürlicher Szenen erfassen [ZSK98]. Ein Roboterarm mit einer Handkamera kann die Grauwertbilder als Eingang benutzen und die optimale Greifposition lernen [ZK98b].

Literatur

- [Cox1972] M.G. Cox The numerical evaluation of B-splines. In *Journal of the Institute of Mathematics and its Applications*, 10, 134-149, 1972
- [deBoor1972] C. deBoor On calculating with B-splines. In *Journal of Approximation Theory*, 6, 50-62, 1972
- [Dierckx1993] P. Dierckx Curve and Surface Fitting with Splines. *Oxford Science Publications*, 1993
- [BFK84] W. Böhm, G. Farin, and J. Kahmann. A survey of curve and surface methods in cagd. *Computer Aided Geometric Design*, 1:1-60, 1984.
- [MK96] S. Mitaim and B. Kosko. What is the best shape of a fuzzy set in function approximation. In *IEEE International Conference on Fuzzy Systems*, 1996.
- [ZF98] J. Zhang and M. Ferch. Rapid on-line learning of compliant motion for two-arm coordination. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1998.
- [ZK98a] J. Zhang and A. Knoll. Constructing fuzzy controllers with B-spline models - principles and applications. *International Journal of Intelligent Systems*, 13(2/3):257-285, Feb./Mar. 1998.
- [ZK98b] J. Zhang and A. Knoll. A neuro-fuzzy control model for grasp learning. In *Proceedings of the Int. Symposium on Image, Speech, Signal Processing and Robotics, Hong Kong*, 1998.
- [ZS98] J. Zhang and V. Schwert. Rapid learning of sensor-based behaviours of mobile robots based on B-spline fuzzy controllers. In *Proceedings of the IEEE International Conference on Fuzzy Systems, Anchorage*, 1998.
- [ZSK98] J. Zhang, V. Schwert, and A. Knoll. Comprehensive fuzzy control of systems with complex sensor patterns. In *Proceedings of the FLINS Workshop on Intelligent Systems and Soft Computing for Nuclear Science and Industry, Antwerp*, 1998.