Self-Optimisation of a Fuzzy Controller with B-Spline Models

Jianwei Zhang and Khac Van Le
Faculty of Technology,
University of Bielefeld,
33501 Bielefeld, Germany
Phone: ++49-(0)521-106-2951
Fax: ++49-(0)521-106-2962
Email: zhang@techfak.uni-bielefeld.de

Abstract

We propose a self-optimisation approach for designing fuzzy controllers. B-spline basis functions of different orders are regarded as a class of membership functions (MFs) with some special properties. These properties lead to several interesting conclusions about fuzzy controllers if such membership functions are employed to specify the linguistic terms of the input variables. By appropriately designing the rule base, C^n-continuity of the output can be achieved (n is the order of the B-spline basis functions). This type of fuzzy controllers are applied in function approximation. Using the gradient descent technique, such a fuzzy controller can be optimised automatically.

1 Introduction

This paper will first briefly introduce the principle of constructing fuzzy controllers with B-spline models, then discuss the problem of automatical optimisation of such type of controllers.

Although fuzzy logic control (FLC) has been successfully applied to a wide range of control problems and has demonstrated some advantages, [TAS94, YLZ94], one obstacle to the wide acceptance for industrial applications is, as pointed out in [DHP95], that "it is still not clear how membership functions, defuzzification procedures, ..., contribute, either in combination or as stand-alone factors, to the performance of the FLC".

Part of these issues can be addressed by comparing B-spline models with a fuzzy logic controller. In our previous work [ZRH94] and [ZK96], we compared splines and a fuzzy controller with SISO (*single-input-single-output*) and MISO (*multi-input-single-output*) structures; periodical non-uniform B-spline basis functions are interpreted as membership functions. In this paper, we concentrate on the self-optimisation for function approximation using a fuzzy controller constructed by the B-spline models.

2 Principle of Constructing Fuzzy Controllers with B-Splines

2.1 B-Spline Basis Functions vs. Membership Functions

We consider the membership functions which are used in the context of specifying linguistic terms ("values" or "labels") of input variables of a fuzzy controller. In the following, basis functions of *Non-Uniform B-Splines* (NUBS) are summarised and compared with the membership functions. We also use *B-functions* for the NUBS basis functions.

Given a sequence of ordered parameters: $(x_0, x_1, x_2, \ldots, x_m, x_{m+1}, \ldots, x_{m+n})$, the normalised B-functions $N_{i,n}$ of order n are defined as:

$$N_{i,n}(x) = \begin{cases} \begin{cases} 1 & \text{for } x_i \le x < x_{i+1} \\ 0 & \text{otherwise} \end{cases} & \text{if } n = 1 \\ \frac{x - x_i}{x_{i+n-1} - x_i} N_{i,n-1}(x) + \frac{x_{i+n} - x}{x_{i+n} - x_{i+1}} N_{i+1,n-1}(x) & \text{if } n > 1 \end{cases} \quad \text{with } i = 0, 1, \ldots, m.$$

One important property of the B-functions is the "paritition of unity", i.e. $\sum_{i=0}^{n} N_{i,n}(x) = 1$.

The B-functions are employed to specify the MFs. Visually, the selection of n, the order of the B-functions determines the following factors of the fuzzy sets for modelling the linguistic terms, Table 1.

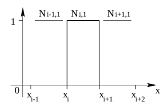
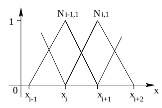
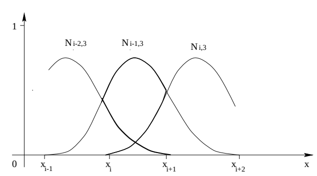
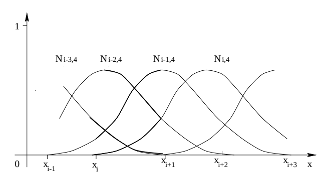
order n	1	2	3	4	...
degree	0	1	2	3	...
shape	rectangular 	triangular 	quadratic 	cubic 	... ...
width	1	2	3	4	...
overlap	1	2	3	4	...

Table 1: The visual effect of fuzzy sets depends mainly on choosing the order of B-functions

2.2 Real and Virtual Linguistic Terms

It is assumed that linguistic terms are to be used to cover $[x_0, x_m]$, the universe of an input variable x of a fuzzy controller. They are referred as *real linguistic terms*. In order to maintain the "partition of unity", some more B-functions should be added at the both ends of $[x_0, x_m]$. They are called *marginal B-functions*, defining the *virtual linguistic terms*. *Real* and *virtual* linguistic terms are denoted as A_i in Fig. 1.

- In case of order 2, no marginal B-function is needed, Fig. 1(a).
- In case of order 3, two marginal B-functions are needed, one for the left end and another for the right end, Fig. 1(b).
- In case of order 4, if the two B-functions $N_{-2,4}$ and $N_{m-2,4}$ are regarded as the real linguistic terms, two marginal B-functions are needed, one for the left end and the other one for the right end, Fig. 1(c).
- If higher order n is used, more marginal B-functions are needed.

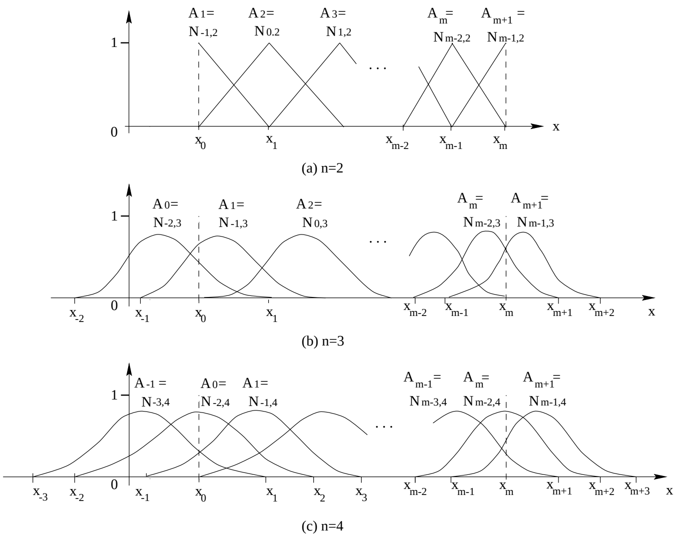


Figure 1: Real and virtual linguistic terms

In the following section, it can be seen that additional rules should be generated for dealing with virtual linguistic terms. Therefore, linguistic terms should be selected appropriately in order to use the least number of marginal B-functions.

2.3 Core and Marginal Rules

We define the *core rules* as linguistic rules which use *real linguistic terms*. If *virtual linguistic terms* appear in the premise, in order to maintain the output continuity at both ends of the universe of x, additional rules are needed to deal with the cases. Since these rules use the *virtual linguistic terms* which are defined by MFs neighbouring the ends of the universe of each variable, they are called *marginal rules*. The output value of each *marginal rule* is just selected as the output value of the "nearest" *core rule*, i.e. the rule using the direct adjacent linguistic terms in its premise (Fig. 2).

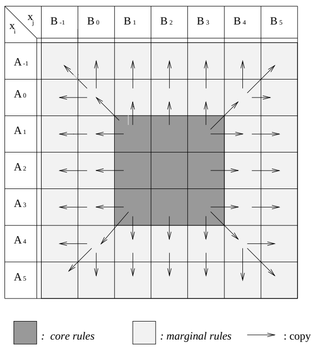


Figure 2: Core and marginal rules

2.4 The General MISO Controllers

Since MIMO rule base is normally divided into several MISO rule bases, we need only to consider the MISO case. Generally, rules with q conjunctive terms in the premise are given in the following form:

$\{Rule(i_1, i_2, \ldots, i_q)$: IF ($x_1$ is $N_{i_1,n_1}(x_1)$) and (x_2 is $N_{i_2,n_2}(x_2)$) and ... and (x_q is $N_{i_q,n_q}(x_q)$) THEN y is $y_{i_1 i_2 \ldots i_q}\}$

Under the following conditions:

- B-functions as MFs for inputs and fuzzy-singletons as MFs for outputs,
- "product" as fuzzy-conjunctions, and
- "center average" defuzzification method,

the output y of a MISO fuzzy controller is:

$$y = \sum_{i_1=-n_1+1}^{m_1+n_1-1} \cdots \sum_{i_q=-n_q+1}^{m_q+n_q-1} y_{i_1 i_2 \ldots i_q} \prod_{j=1}^{q} \cdot N_{i_j,n_j}(x_j) \tag{1}$$

This is called a *general NUBS hypersurface*, which possesses the following features:

- If the B-functions of order $n_1, n_2, \ldots, n_q$ are employed to specify the linguistic terms of the input variables $x_1, x_2, \ldots, x_q$, it can be guaranteed that the output variable y is $(n_j - 2)$ times continuously differentiable with respect to the input variable $x_j, j = 1, \ldots, q$.
- If the input space is partitioned enough fine, the interpolation with B-spline hypersurface can reach a given precision.

If the order of the B-functions and the number of linguistic terms used in the premise are chosen, the output of the fuzzy controller can be flexibly adapted to anticipated values by adjusting the positions of the fuzzy-singletons (control vertices) of the *core rules*.

3 Application in Function Approximation

If the pure B-spline interpolation is employed for function approximation, a straightforward computation model can be applied which has to mainly solve tridiagonal matrices. Such a procedure is generously used in CAD/CAM areas. For realising adaptive control system, we propose a self-optimisation approach to find the appropriate control vertices iteratively.

3.1 Self-Optimisation of Control Vertices

The optimisation procedure is based on the gradient descent approach (See [Jan93] and [WWLT95]). An error function is defined as:

$$E = \frac{1}{2}(Expected\_Value - Controller\_Output)^2$$

In each optimisation step, the control vertices are modified with:

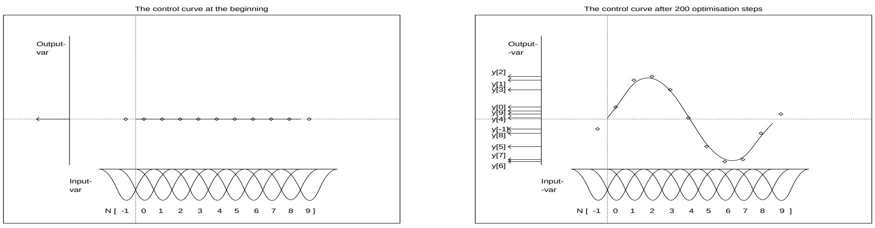
$$\delta_{y_i} = (Expected\_Value - Controller\_Output) \cdot N_{i,n}$$

3.2 Examples

3.2.1 A One-Input-One-Output Controller

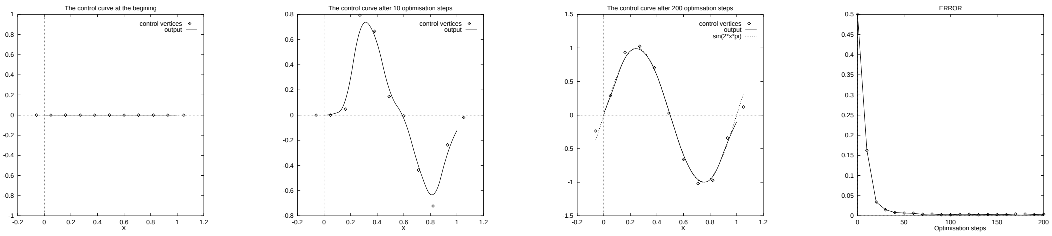
A function $y = sin(2\pi x)$ is assumed to be approximated with a fuzzy controller. Fig. 3 depicts the mapping of the input x to the output y, where x is covered with B-functions of order 3 and fuzzy-singletons are defined on y. The initial positions of the fuzzy-singletons are arbitrarily chosen, e.g. as zero, see Fig. 3(a). The output curve and the fuzzy-singletons after the self-optimisation process are illustrated in Fig. 3(b).

Fig. 4(a)–(c) show several intermediate steps during the optimisation. The approximation error is shown in Fig. 4(d).



(a) before optimisation (b) after optimisation

Figure 3: Mapping the input to the output with B-functions as MFs

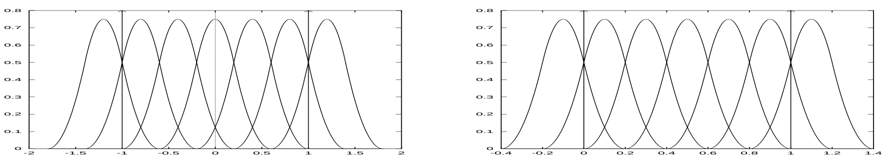


(a) step 0 (b) step 10 (c) step 200 (d) approx. error

Figure 4: Self-optimisation of positions of the fuzzy-singletons defined on the output

3.2.2 A 2D Example

A 2D example is implemented to approximate the function $z = sin(2\pi x) \cdot cos(\pi y)$, with $-1 \leq x \leq 1$ and $0 < y < 1$. Fig. 5 (a) and (b) show the MFs defining the real and virtual linguistic terms of x and y. The control surface in several intermediate steps of the optimisation can be seen in Fig. 6. Fig. 7 shows the final results after the optimisation process terminates: Fig. 7(a) depicts the automatically generated control vertices, Fig. 7(b) depicts the convergence of the approximation error.



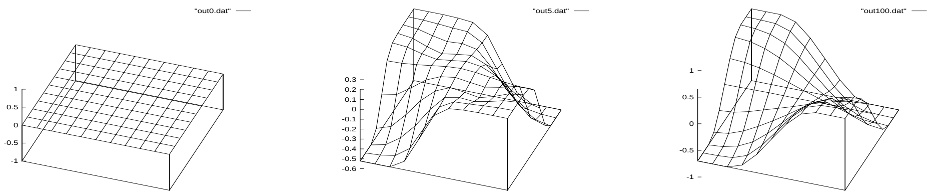
(a) variable x (b) variable y

Figure 5: B-functions as MFs for the input variables

4 Conclusions

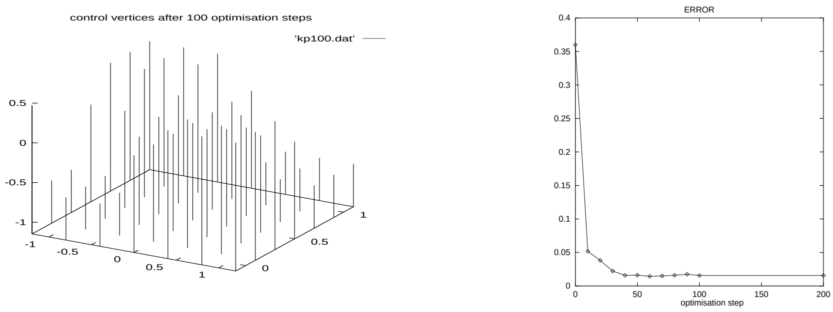
We propose an approach for constructing a fuzzy controller to approximate a sequence of points of a known function. The advantages with B-Spline fuzzy controllers can be summarised as followings: a). transparency of the interpolation process using fuzzy-controllers; b). smoothness of the output and c). no information loss after the defuzzification.

Although the proposed self-optimisation approach is a kind of supervised learning methods, we can easily extend the concept to the usage for unsupervised learning based on this controller structure. If no training data available, an evaluation function can be designed to represent the goal of the controller, the control



(a) step 0 (b) step 5 (c) step 100

Figure 6: The control surfaces during the optimisation



(a) control vertices (b) error

Figure 7: After 100 optimisation steps

vertices can be then adapted to realise the optimal value. We are applying unsupervised learning with this kind of fuzzy controller in the navigation problem for mobile robots, some preliminary results have shown the feasibility of the idea. In our point of view, fuzzy controllers with B-spline models provide a suitable structure for realising adaptive systems as well as smooth control tasks.

References

[DHP95] D. Driankov, H. Hellendoorn, and R. Palm. *Some Research Directions in Fuzzy Control*, chapter 11, pages 281–312. in “Theoretical Aspects of Fuzzy Control”, edited by H. T. Nyuen, M. Sugeno, R. Tong, and R. R. Yager, John Wiley & Sons, 1995.

[Jan93] J.-S. R. Jang. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on System, Man and Cybernetics*, 23(3):665–685, 1993.

[TAS94] T. Terano, K. Asai, and M. Sugeno. *Applied Fuzzy Systems.* AP Professional, 1994.

[WWLT95] C.-H. Wang, W.-Y. Wang, T.-T. Lee, and P.-S. Tseng. Fuzzy B-spline membership function (BMF) and its applications in fuzzy-neural control. *IEEE Transactions on System, Man and Cybernetics*, 25(5):841–851, 1995.

[YLZ94] J. Yen, R. Langari, and L. A. Zadeh. *Industrial Applications of Fuzzy Logic and Intelligent Systems.* IEEE Press, 1994.

[ZK96] J. Zhang and A. Knoll. Constructing fuzzy controllers with B-spline models. In *Pcoceedings of 1996 IEEE Int. Conf. on Fuzzy Systems*, 1996.

[ZRH94] J. Zhang, J. Raczkowsky, and A. Herp. Emulation of spline curves and its application in robot motion control. *Proceedings of IEEE International Conference on Fuzzy Systems, Orlando*, pages 831–836, 1994.