In Proceedings of International Concerence on Computational Inteligence for Modelling, Control and Automation, Viena, 1999.

Efficient Learning of Non-Uniform B-Splines for Modelling and Control

Jianwei ZHANG, Alois KNOLL and Ingo RENNERS
Faculty of Technology, University of Bielefeld, Germany 33501 Bielefeld, Germany

Abstract. We propose an approach to designing fuzzy controllers based on the B-spline model by learning. Unlike other normalised parameterised set functions for defining fuzzy sets, B-splines do not necessarily span membership values from zero to one but possess the property of "partition of unity". B-splines can be automatically determined after each input is partitioned. Learning of a fuzzy controller based on B-splines is then equivalent to the adaptation of a B-spline interpolator. Parameters of the controller output of each rule can be rapidly adapted by gradient descent. Optimal placements of the non-uniform B-splines for specifying each input can be found by a genetic algorithm. Through comparative examples of function approximation we show that training of such a fuzzy controller generally provides results with minimal error. The approach can be extended to the problems of supervised as well as unsupervised learning.

1 Introduction

Classical fuzzy controller of the Mamdani type [5] is based on the idea of directly using symbolic rules for diverse control tasks. As application areas grow, the *systematic* design of an optimal fuzzy controller becomes more and more important. Another important type of fuzzy controllers is based on the TSK (Takagi-Sugeno-Kang) model [7]. Recently, TSK type fuzzy controllers have been used for function approximation and supervised learning [8, 1]. However, it is pointed out that the TSK model is a black-box based on multi-local-model.

We propose an approach that can build membership functions (MFs) for linguistic terms of the IF-part systematically, then adapt the control actions of the THEN-part and the shape and position of the IF-part MFs through learning. Our approach is based on the B-spline model.

B-spline models employ piecewise polynomials as MFs. The universe of discourse of each input is divided into a number of subintervals, where each subinterval is delimited by breakpoints called *knots* which determine the appearance and position of each B-spline. Figure 1 illustrates the partition of a two-dimensional B-spline model with 8 MFs on each uniformly subdivided input interval and the activated B-splines (slightly shaded) for a given input. Since learning one new part of the input space affects only a given number of controller response values (darkly shaded area of figure 1), fast on-line learning can be devised. Due to these advantages, B-spline models are proposed to be applied in control systems and will be denoted as B-spline Fuzzy Controllers [10]. By using the B-spline model the approximation ability is only limited by the number of knot-points distributed over the input intervals. Regarding that most observed data are disturbed to a certain degree, the overfitting problem may occur. Genetic optimized B-spline models are a promising approach to find sparse models, which are able to bridge the gap between high bias and high variance of a model.

2 Constructing Fuzzy Controllers with B-Splines

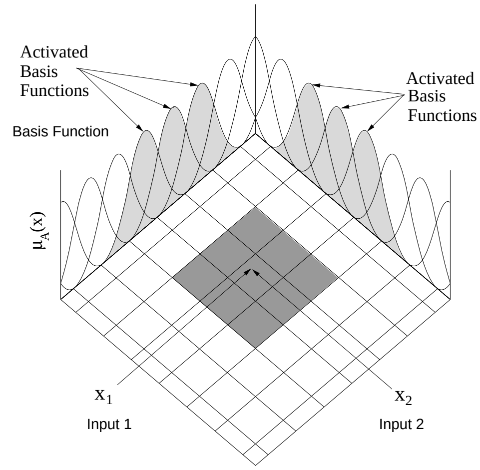
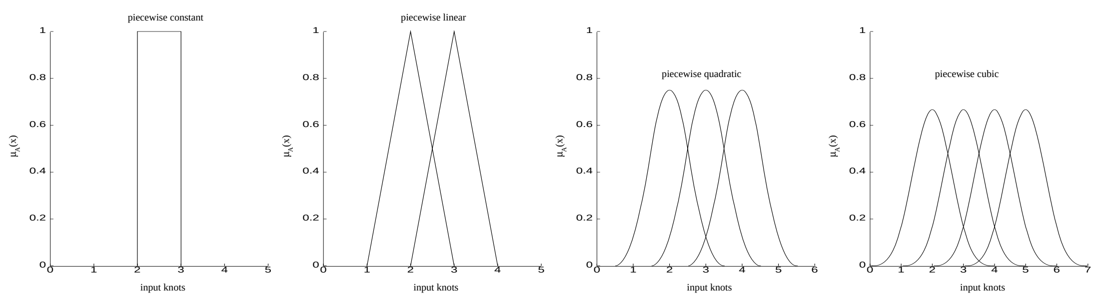


Figure 1: The B-spline model – a two-dimensional illustration.

2.1 Definition of B-Splines

The B-spline $N_{i,k+1}$ of degree k with knots $\lambda_1, \ldots, \lambda_{i+k+1}$ is defined as (see figure 2):

$$N_{i,k+1}(x) = (\lambda_{i+k+1} - \lambda_i) \sum_{j=0}^{k+1} \frac{(\lambda_{i+j} - x)_+^k}{\prod_{\binom{l=0}{l \neq j}}^{k+1} (\lambda_{i+j} - \lambda i + l)}. \tag{1}$$



(a) Order 1 (b) Order 2 (c) Order 3 (d) Order 4

Figure 2: Univarite B-splines (UBs) of order 1-4.

The most important properties of B-splines, with respect to neurofuzzy modelling are: a) the B-spline computation is recursive; b) all B-splines are non-nagative; c) each B-spline has only local support; d) at each input point "partition of unity" is fulfilled.

2.2 Use B-Splines as Fuzzy Controller

In [10], we showed that under several slightly modified conditions, the computation of the output of such a fuzzy controller is equivalent to that of a *general B-spline hypersurface*.

The output of a Single Input Single Output (SISO) B-spline network is simply the unique representation of a B-spline $s(x),\ x \in [a, b]$:

$$y = \sum_{i=1}^{m} c_i N_{i,k}(x) \tag{2}$$

in which c_i are called *control points of* $s(x)$ (also denoted as *weights*, or *de Boor points*) and m denotes the number of basis functions. This network can also interpreted as a fuzzy system of the zero-order

TSK type. The overall output of a SISO B-spline network is:

$$y = \frac{\sum_{i=1}^{m} c_i N_{i,k}(x)}{\sum_{i=1}^{m} N_{i,k}(x)} = \sum_{i=1}^{m} c_i N_{i,k}(x) \tag{3}$$

The output of a Multiple Input Single Output (MISO) B-spline network can be computed straightforward:

$$y = \sum_{i_1=1}^{m_1} \cdots \sum_{i_n=1}^{m_n} \left(c_{i_1 i_2 \ldots i_n} \prod_{j=1}^{n} N_{i_j,k_j}^{j}(x_j) \right) \tag{4}$$

where x_j is the j^{th} input $(j = 1, \ldots, n)$; k_j is the order of the B-splines used for x_j; N_{i_j,k_j}^{j} is the i^{th} linguistic term of x_j defined by B-splines; $i_j = 1, \ldots, m_j$ represents how fine the $j-$th input is partitioned; $c_{i_1,i_2,\ldots,i_n}$ are the control points of Rule $(i_1, i_2, \ldots, i_n)$.

2.3 Generating the THEN-Part

Fuzzy singletons represented by control points can be initialised with the values acquired from expert knowledge. These parameters will be fine-tuned by a learning algorithm.

For supervised learning, we show in the following that the squared errors with respect to control points are convex functions. Therefore, rapid convergence for supervised learning is guaranteed. The control space changes locally due to the "local support" property of B-functions while the control points are modified. Based on this feature, the control points can be optimised gradually, area-by-area.

Assume that $(X^r, y^r_{desired})$ is a set of training data, where $X^r = (x^r_1, \ldots, x^r_n)$ is the r^{th} input vector with desired output $y^r_{desired}$. The output value computed by a controller output is denoted with $y^r_{computed}$. By defining the Mean-Square Error (MSE) criterion as:

$$E = \frac{1}{2} \cdot (y_{computed} - y_{desired})^2 \equiv MIN, \tag{5}$$

the derivative of each control point $c_{r_1,\ldots,r_k}$ is:

$$\Delta c_{i_1,\ldots,i_k} = -\epsilon \frac{\partial E}{\partial c_{r_1,\ldots,r_k}} = \epsilon (y^r_{computed} - y^r_{desired}) \cdot \prod_{j=1}^{n} N_{i_j,n_j}(x_j), \tag{6}$$

where ϵ denotes the *learning rate*. Since the second partial derivative of $c_{r_1,\ldots,r_k}$ is constant, the error function (5) is convex in space. If the inputs x_r are *linearly independent*, there exists, due to the convexity of the MSE performance surface, only one global minimum and no local minima. On the other hand, if the autocorrelation matrix is singular, which occurs when the inputs x_r are *linearly dependent*, there exists an infinite number of global minima in weight space.

2.4 Adaptive Modelling of the IF-Part

Based on the granularity of the input space and the distribution of extrema in the control space (if known), the fuzzy sets can be initialised using the recursive computation of B-splines. These fuzzy sets based on non-uniform B-splines can be further adapted during the generation of the whole system by using Genetic Algorithms (GAs).

By freeing the knot-positions, the task of finding control points and accurate knot-vectors to fit training data becomes a non-linear minimization problem. To solve this problem we follow a strategy of problem splitting. We first consider the underlying model $\delta(\lambda)$ of the controller and then compute the control points to minimize $\delta(c)$. Instead of using constrained least-square methods (constrained because of avoiding to "ride" on the gradient edge of coincident knots [2]), we try to estimate the knot-positions by using GAs, because GAs are both theoretically and empirically proven to provide the means for efficient search, even in complex spaces [3]. Therefore each individual, in the example each B-spline controller with its special knot-point distribution, represents one point in search space.

2.4.1 The Genetic Algorithm

We applied the basic GA introduced by Holland [4] with some modifications as follows:

- We used *gray coding* instead of standard binary code while representing coded chromosomes, a common modification.

- To bypass the undesirable effect of the increasing probability along the descendent chromosome-string to receive a changed allele (bit) (thus the conjointly heredity of genes decreases when the distance of their position increases) while using n-point crossover, we used *uniform crossover*. This kind of crossover has no positional and a high distributional bias, so that a high blending rate between participant chromosomes is granted. This leads to an algorithm producing permanently solutions which explore new locations by bridging even great distances of the search space.

- Instead of using fitness-proportional selection it is advantageous to use *tournament selection*. This selection schema draws ξ individuals ($2 \leq \xi \leq \mu$) with a probability $\frac{1}{\mu}$ from the current population and copies the individual with the best fitness into the mating pool. Besides saving computational power as a result of no need to sort the population (as in ranking based selection schemes), it is easier to bias the *takeover time*.

2.4.2 Chromosome Encoding for the Knot Placement Problem

To minimize $\delta(\lambda)$ each individual consists of n knot-vectors, where n is the problem dimension. Each encoded knot-vector consists of 32 knot-points and a so-called *activation string* of 32 bit length. Which knot-points are in use to define the current model is encoded through the activation string. Activated knots are represented by 1 and inactivated knots are represented by 0.

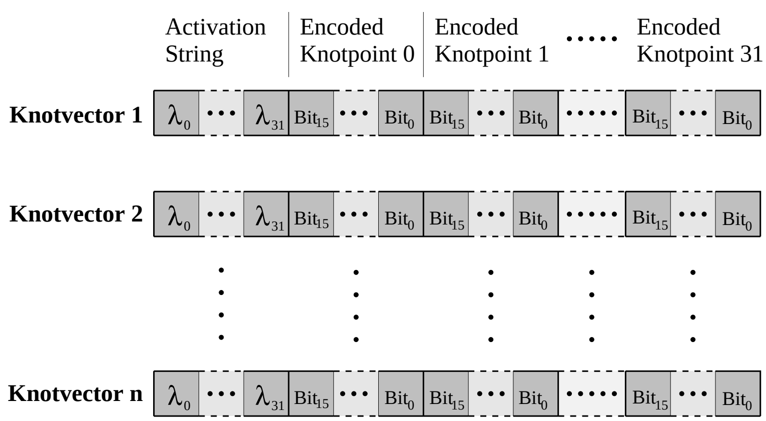


Figure 3: Encoded B-spline model.

Every knot-point is encoded by 16 bit (see figure 3) and therefore each knot-point can be placed on its respective input interval $[a, b]$ with an accuracy of $\frac{1}{2^{16}} \times (b - a)$. The fitness values for each individual are simply computed by determining the control points of the controller. Using these control points the mean square error is evaluated and the fitness for one individual is set equal to $\frac{1}{MSE}$.

3 Numerical Results

For comparison (see table 1) the above described GA was applied on one and two-dimensional functions used in [6]. Parameter settings were chosen as crossover probability= 0.75, mutation probability= 0.0005, $\mu = 40$ and $\xi = 3$. A maximum generation index of 200 was used as stop criterion. The MSE of each adapted B-spline model represents the average MSE of 3 runs. These functions with the optimized B-splines are shown in figure 4 and 5.

4 Applications

Besides function approximation, this model has been applied to supervised and unsupervised learning in the following intelligent control and robot systems:

4.1 On-Line Learning

Rapid on-line learning of sensor-based operations is implemented. A mobile robot with distance sensors can realise the intelligent behaviours like collision-avoidance, contour tracking, goal tracking, etc., through only a few learning steps [12]. A two arm robot system can learn on-line the cooperative motion by evaluating force/torque sensors [9].

Function	Rules	Used Membership Function			
		Uniform B-splines	Adapted B-splines	Best of [6]	Worst of [6]
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	0.000048	0.002	0.03
f_4	12	4.9	0.04	0.1	10
f_5	12	0.04	0.0002	0.01	1
f_6	12	0.6	0.012	0.1	0.4
g_1	64	766	26.36 (60 rules)	9	26
g_2	64	10.91	2.8 (60 rules)	7	19
g_3	64	5.78	0.01 (60 rules)	1.2	6

Table 1: MSE results from [6] in comparison to results of a uniformly distributed B-spline controller and results of a genetic modified B-spline controller.

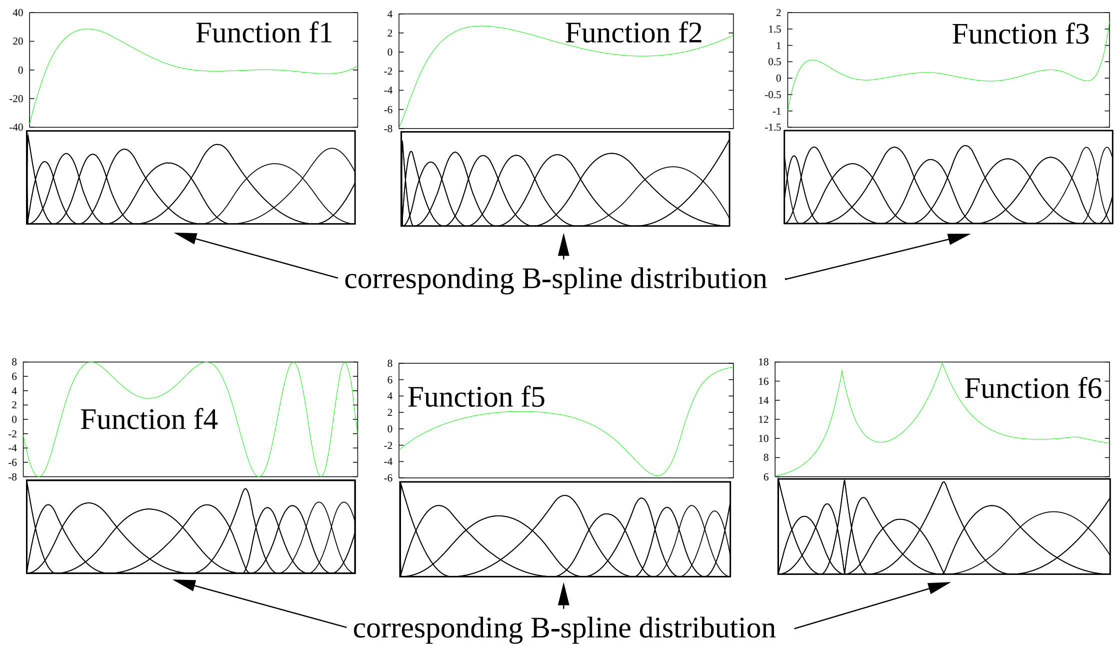


Figure 4: Optimized B-splines for one-dimensional test functions.

4.2 Visual Learning

We implemented visual learning of robot positions based on PCA (principal component analysis) dimension reduction and the B-spline model. With an omnidirectional vision system, a mobile robot can locate itself by using just the natural landmarks [13]. A robot arm with hand-eye can use the raw image data as input and learn the optimal grasping motion [11].

References

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

[2] Jupp, D.L.B. The 'Lethargy' theorem - a property of approximation by γ-polynomials. In *Journal of Approximation Theory, 14, p.204-217.*

[3] Goldberg, D.E. Genetic Algorithms in Search Optimization, and Machine Learning. *Reading, MA: Addison-Wesley*, 1989.

[4] Holland, J.H. Adaption in Neural and Artificial Systems. *Ann Arbor, MI: University of Michigan Press*,1975.

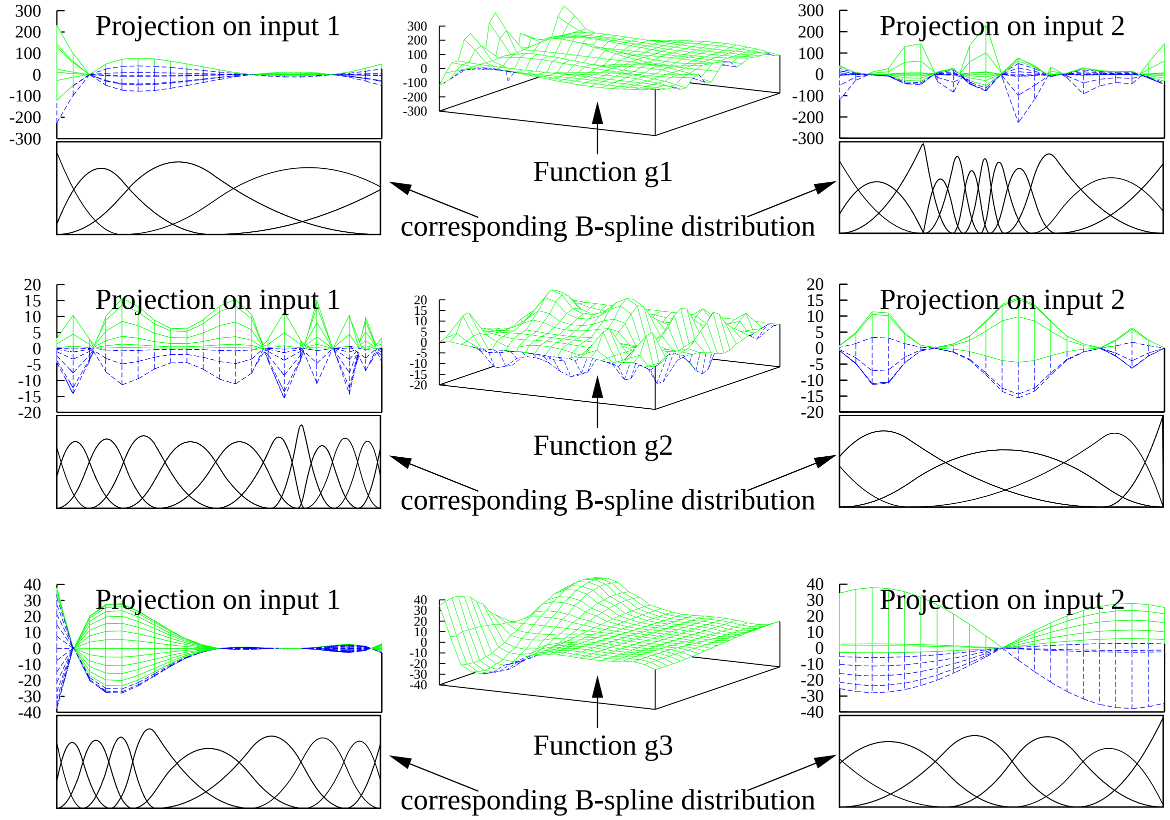


Figure 5: Optimized B-splines for two-dimensional test functions.

[5] E. H. Mamdani. Twenty years of fuzzy control: Experiences gained and lessons learned. *IEEE International Conference on Fuzzy Systems*, pages 339–344, 1993.

[6] Mitaim, S., AND Kosko, B. What is the best shape of a fuzzy set in function approximation. In *IEEE International Conference on Fuzzy Systems*, 1996.

[7] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modelling and control. *IEEE Transactions on System, Man and Cybernetics*, SMC-15(1):116–132, 1985.

[8] L. Wang. *Adaptive Fuzzy Systems and Control*. Prentice Hall, 1994.

[9] 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*, pages 1853–1858, 1998.

[10] 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.

[11] J. Zhang and A. Knoll. A neuro-fuzzy control model for grasp learning. In *Proceedings of the Int. Symposium on Image, Speech, Signal Proc essing and Robotics, Hong Kong*, volume I, pages 307–312, 1998.

[12] 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.

[13] 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.