Proceedings of the 2006 ANIPLA
International Congress on
Methodologies for Emerging
Technologies in Automation (ANIPLA)

Quadratic Optimal Fuzzy Control

Paulo Salgado
CETAV - Universidade de Trás-os-Montes e Alto Douro
Quinta de Prados
5000-911 Vila Real
PORTUGAL
psal@utad.pt

Getúlio Igrejas
Instituto Politécnico de Bragança
Campus de St.a Apolónia
5301-857 Bragança
PORTUGAL
igrejas@ipb.pt

Paulo Garrido
Universidade do Minho
Campus de Azurém
4800-058 Guimarães
PORTUGAL
paulo.garrido@dei.uminho.pt

***Abstract* – One presents a fuzzy logic approach for optimal control of discrete-time nonlinear dynamic systems with a quadratic criterion. The approach is based on Pontryagin's Minimum Principle. It uses back-propagation from the final co-state error and gradient descent to estimate a sequence of values for the co-state variables converging to the optimal ones. This implies that the controlled variables trajectories converge to the optimal ones. The estimator is implemented through an adaptive fuzzy inference system.**
The approach allows one to find a solution to the optimal control problem on-line by training the system, rather than by pre computing it. The use of an adaptive fuzzy inference system will allow to incorporate *a priori* knowledge about the optimal behavior of the co-state variables and to track changes in the system.

I. INTRODUCTION

In the past decade, fuzzy inference systems emerged as a most useful approach to collect human knowledge and expertise on control and to transform the collected knowledge into a basis for developing controllers [1–3]. A fuzzy logic controller is usually a fuzzy inference system establishing a static nonlinear mapping from the state variables values to the actuators values [4].
In the fuzzy logic approach to optimal control described here, the (adaptive) fuzzy inference system can be used to generate actuator values, but its primary function is to generate estimates of the co-state variables.
Co-state variables play a key role in finding the optimal control when using Pontryagin's Minimum Principle (PMP). One begins by briefly reviewing the concepts of PMP, specially focusing those related to the case herein presented, i.e. for systems modeled in discrete time.
Let one consider systems described by nonlinear difference equations of the kind:

$$x_{k+1} = f^k\left(x_k, u_k\right) \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state vector and $u_k \in \mathbb{R}^m$ the control vector of the system at time *kT*, *f* is a vector valued function, possibly non-linear and time-varying, and *T* is the sampling period. One requires that $f(x,u): \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is Lipschitz continuous and that there exists a constant $M_f > 0$ such that $\left|f(x,u)\right| \le M_f\left(|x|+|u|+1\right)$ for all $(x,u) \in \mathbb{R}^n \times \mathbb{R}$.
The control problem is to find the control sequence u_k^* that minimizes the criterion or cost function J_i:

$$J_i = \Phi\left(N, x_N\right) + \sum_{k=i}^{N-1} L^k\left(x_k, u_k\right) \tag{2}$$

In (2) [*i*, *N*] is the prescribed control time interval, $\Phi\left(N, x_N\right)$ is the cost on the final state value x_N, and $L^k\left(x_k, u_k\right)$ is the cost on both state and command at instant $k < N$.
The solution for this problem given by PMP is as follows. One defines the sequence of Hamiltonian functions H^k:

$$H^k = L^k + \lambda_{k+1}^T \cdot f^k \tag{3}$$

where $\lambda_k \in \mathbb{R}^n$ is a vector of Lagrange multipliers. Accordingly to common usage one will designate λ_k as the co-state variables. The optimal sequence u_k^* that minimizes the criterion (2) is found by solving simultaneously the following equations:

State variable equation:

$$x_{k+1} = \frac{\partial H^k}{\partial \lambda_{k+1}} = f^k\left(x_k, u_k\right) \tag{4}$$

Co-state variable equation:

$$\lambda_k = \frac{\partial H^k}{\partial x_k} = \left(\frac{\partial f^k}{\partial x_k}\right)^T \lambda_{k+1} + \frac{\partial L^k}{\partial x_k} \tag{5}$$

Stationary conditions equation:

$$0 = \frac{\partial H^k}{\partial u_k} = \left(\frac{\partial f^k}{\partial u_k}\right)^T \lambda_{k+1} + \frac{\partial L^k}{\partial u_k} \tag{6}$$

Limit conditions equations:

$$\begin{cases} \left(\dfrac{\partial L^0}{\partial x_0} + \left(\dfrac{\partial f^0}{\partial x_0}\right)^T \lambda_1\right)^T \mathrm{d}\,x_0 = 0 \\ \left(\dfrac{\partial \Phi}{\partial x_N} - \lambda_N\right)^T \mathrm{d}\,x_N = 0 \end{cases} \tag{7}$$

Finding solutions for equations (4)-(7) is not, in general, an easy task, due to the interdependence of equations (4) and (5) implying that forward and backward time sequences should be used.
However, looking at the problem on fuzzy logic grounds, the co-state variables appear to behave as the output of an expert

system that knows which sequence of values will minimize the cost function.

This means that one may devise a control strategy based on an adaptive fuzzy inference system which generates at each time *k*, in the control time interval, an estimated value for the co-state variable at time $k+1$.

Let one define a training iteration as a sequence of control actions from $k=0$ to $k=N-1$. Then, along successive training iterations the fuzzy inference system rules may be changed in order to generate estimates converging to the true optimal values of the co-state variables. This will imply that the state variables values will also converge to the optimal ones.

Changing the rules of the fuzzy inference can be made by means of a learning algorithm which takes the final error between state and co-state variables as its input. In fact, it can be proved that, under the quadratic version of criterion (2) one can assume that if this error goes to zero then the state and co-state trajectories will converge to the optimal ones.

Subsequently this idea is explored as follows. Firstly, in section 2, a brief introduction to the fuzzy inference systems used is made. The quadratic version of the optimal control problem considered and the fuzzy logic approach to its solution are described in section 3. Section 4 presents the learning algorithm. An illustrative simulation example is presented in section 5. Finally, the main conclusions are outlined in section 6.

II. THE FUZZY INFERENCE SYSTEM

Consider the system $y = f(x_1,\cdots,x_n)$, in which $y \in V$ is the output (or consequent) variable and $x_i \in U_i$, $i=1,\cdots,n$, are the input (or antecedent) variables. In fuzzy systems modelling, this relationship is represented by a collection $\Im$ of *M* fuzzy *IF–THEN* rules:

$$R_l : IF\ x_1\ is\ A_{l1}\ and \cdots x_i\ is\ A_{ln}\ THEN\ y\ is\ B_l \qquad (8)$$

where A_{li} in U_j and B_l in V are linguistic terms characterized by fuzzy membership functions $A_{li}(x_i)$ and $B_l(y)$, respectively. The linguistic connective "and" of antecedent of rule (8) will be defined as a t-norm operation, $\star$, where an aggregated fuzzy set A_l can be viewed as the fuzzy intersection set $\mathsf{X}_{i=1}^{n} A_{li}$ with membership function $A_l(\boldsymbol{x}) = A_{l1}(x_1)\star\cdots\star A_{ln}(x_n)$. The fuzzy implication of each rule *l*, $R_l : A_l \mapsto B_l$ is a fuzzy set in product space $U \times V$ which is defined as $R_{l:A\mapsto B}(\boldsymbol{x}, y) = A_l(\boldsymbol{x}) \otimes B_l(y)$, where "$\otimes$" is an operator rule of fuzzy implication, usually min-max inference [1-2] or arithmetic inference [3].

For each rule *l*, the effective output value B_l is calculated using sup-star composition:

$$B'_l(y) = \sup_{x\in U}\left[A'(\boldsymbol{x})\star R_{l:A\mapsto B}(\boldsymbol{x}, y)\right].$$

The final fuzzy set *B*, which is determined by all the rules in the base, is obtained by the combination of the B_l and their associated membership functions.

The fuzzy inference system performs a mapping from $U \in \mathbb{R}^n$ to $V \in \mathbb{R}$ (a multi-input multi-output fuzzy system can always be seen as a parallel of several multi-input single-output fuzzy systems). It comprises four main components: fuzzifier, fuzzy rule base, fuzzy inference engine and defuzzifier. Many different choices are available within each block, and in addition, many combinations of these choices can result in a useful subclass of fuzzy systems.

The fuzzy inference system used in this study is based on a product inference engine, singleton fuzzifier, and centre-average defuzzifier. So, the fuzzy logic inference system may be represented by:

$$f(\vec{x}) = \frac{\sum_{l=1}^{M} \bar{y}_l A_l(x_i)}{\sum_{l=1}^{M} A_l(x_i)}, \qquad (9)$$

where $A_l(x) = \prod_{i=1}^{n} A_{li}(x_i)$ is the input membership function and $\bar{y}_l$ is the centre of the output membership function.

There are three main reasons for using this fuzzy system as a basic building block for adaptive fuzzy controllers or identification systems:

–It has been showed that fuzzy logic systems given by (9) are universal function approximators [4-9];

–These fuzzy logic systems are constructed from fuzzy *IF-THEN* rules using specific fuzzy inference, fuzzification, and defuzzification strategies, which allow the incorporation of information from human experts into controllers;

–The functional form of (9) can be represented as a three-layer feedforward network. Therefore, it is possible to apply the back-propagation learning methods or other neuro-fuzzy techniques, for adjusting the parameters of the membership functions of the rules [4-5], yielding an adaptive fuzzy system.

An adaptive fuzzy system is defined as a fuzzy logic system equipped with a learning algorithm. The fuzzy system is constructed from a set of fuzzy IF-THEN rules and the learning algorithm adjusts the parameters of the fuzzy system based on the training information.

III. THE OPTIMAL CONTROL ALGORITHM

In this section, a method for the optimal control of systems described by (1) under a quadratic version of criterion (2) is presented. The method is based on the ideas and framework sketched above.

Consider the rewriting of the nonlinear discrete dynamic system in (1) for the case of a scalar u_k as:

$$x_{k+1} = g(x_k) + h(x_k)u_k \qquad (10)$$

where $g : \mathbb{R}^n \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}^n$ are continuous over $\mathbb{R}^n$. Assume that $g + hu_k$ is Lipschitz continuous on a set $U \in \mathbb{R}^n$ containing the origin, and that system (10) is

stabilizable in the sense that there exists a continuous control on U that asymptotically stabilizes the system. It is desired to find a sequence of u_k, which minimizes the cost function:

$$J_0 = \frac{1}{2}(x_N - r_N)^T R_N (x_N - r_N) + \frac{1}{2}T^2 \sum_{k=o}^{N-1} u_k{}^T Q_k u_k \qquad (11)$$

where r_N is the desired final state, R_N and Q_k are matrices that allow to weight attainment of the desired final state versus control effort.
Now, (4)-(7) can be rewritten:

$$\lambda_k = \left(\frac{\partial f^k}{\partial x_k}\right)^T \lambda_{k+1}, \qquad (12)$$

$$u_k = -Q_k^{-1}\left(\frac{\partial f^k}{\partial u_k}\right)^T \lambda_{k+1}, \qquad (13)$$

$$\lambda_N = (x_N - r_N) R_N . \qquad (14)$$

Given (10), (13) allows one to write the control variable at time k as:

$$u_k = -Q_k^{-1} h(x_k) \lambda_{k+1} . \qquad (15)$$

This equation may be taken as an optimal feedback control law, if the optimal value of the co-state variable, λ^*_{k+1} is known at time k+1.
Now the approach proposed in this paper may be made explicit. One takes λ_{k+1} as the output of a fuzzy inference system $\boldsymbol{\Lambda}$ that at instant k generates an estimate of λ^*_{k+1}, having as inputs the observed state x_k and the time to go $N-k$:

$$\lambda_{k+1} = \hat{\lambda}^*_{k+1} = \boldsymbol{\Lambda}(x_k, N-k) . \qquad (16)$$

This gives the feedback control law

$$u_k = -Q_k^{-1} h(x_k) \boldsymbol{\Lambda}(x_k, N-k) \qquad (17)$$

which by incorporation of the h function into the fuzzy inference system can be streamlined to:

$$u_k = -Q_k^{-1} \boldsymbol{\Lambda}_h(x_k, N-k) \qquad (18)$$

If, by adaptation or learning of the fuzzy inference system, along successive runs or training iterations of the system from $k=0$ to N, the estimates are made to converge to the optimal ones, then any of the control laws (17) or (18) becomes optimal.
That this indeed can be done is the subject of the next section.

IV. THE LEARNING ALGORITHM

To solve the equations resulting from framing a discrete optimal control problem under PMP, one usually applies an off-line optimization method. Here, one proposes a learning algorithm based on an approximate gradient descent method that, during the training iterations, progressively refines the accuracy of the co-state fuzzy estimator. This strategy reduces the necessary computing time and memory, avoiding calculating the exact adjoint and the directional derivatives of the cost functional. The first algorithm of this method, in theoretical form, is given below. The implementation of the algorithm will be described in future work.
Without loss of generality, let one consider that in (11) $r_N = 0$ and $R_N = 1$. From (14) it follows that for optimal trajectories one must necessarily have $x_N = \lambda_N$ so $x^*_N = \lambda^*_N$.
Let $E = x_N - \lambda_N$ be the error or difference between the end value of the state and the co-state variable trajectories, as exemplified in Fig. 1. As pointed above, for optimal trajectories $x^*(k)$ and $\lambda^*(k)$, it is a necessary condition that $x^*_N = \lambda^*_N$ or $E = 0$. It is possible to prove that this is also a sufficient condition. If $E \to 0$, then $x_k \to x^*_k$ and $\lambda_k \to \lambda^*_k$, i.e. the trajectories of the state and co-state variables converge to the optimal ones. Fig. 1 graphically depicts the idea for one state and co-state variable.
It follows that to attain optimal state trajectories, it is necessary that the error E converges to zero. This objective is achieved by adjusting the λ_k co-state variables or parameters in order to minimize:

$$E^2 = (\lambda_N - x_N)^T (\lambda_N - x_N) . \qquad (19)$$

The gradient descent algorithm was employed to determine the adjustments to the co-state values:

$$\lambda_k^{q+1} = \lambda_k^q - 2\alpha E^q \frac{\partial E^q}{\partial \lambda_k^q} \qquad (20)$$

where, $q = 0,1,2,\ldots$ is the training iteration number and α is a scalar step-size variable.
For all q one has that:

$$\frac{\partial E}{\partial \lambda_k} = \frac{\partial \lambda_N}{\partial \lambda_k} - \frac{\partial x_N}{\partial \lambda_k} \qquad (21)$$

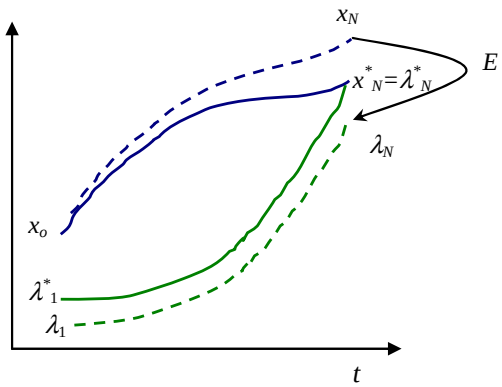

Fig. 1. Trajectories of optimal (-) and non-optimal (--) state (blue line) and co-state variable (green line)

The summands at the right side of (21) have as expressions:

$$\frac{\partial \lambda_N}{\partial \lambda_k} = \left(\prod_{i=k}^{N-1} \frac{\partial f^i}{\partial x_i} \right)^{-1} \tag{22}$$

$$\frac{\partial x_N}{\partial \lambda_k} = \frac{\partial x_N}{\partial \lambda_N} \frac{\partial \lambda_N}{\partial \lambda_k} + \frac{\partial x_N}{\partial x_{N-1}} \frac{\partial x_{N-1}}{\partial \lambda_k} \tag{23}$$

The expression (23) can be solved iteratively as:

$$\frac{\partial x_j}{\partial \lambda_k} = \frac{\partial x_j}{\partial \lambda_j} \frac{\partial \lambda_j}{\partial \lambda_k} + \frac{\partial x_j}{\partial x_{j-1}} \frac{\partial x_{j-1}}{\partial \lambda_k}, \quad \text{for } j > k$$

V. EXPERIMENTAL RESULTS

For illustrative purposes, the method described was implemented to regulate the plant:

$$x_{k+1} = x_k + T \cdot f(x_k, u_k) \tag{24}$$

with:

$$f(x_k, u_k) = 10 \frac{1 - e^{-x_k}}{1 + e^{-x_k}} + u_k \tag{25}$$

In this case, $r_N = 0$. The plant (25) is unstable if no control action exists:

$$\begin{aligned} u_k = 0 \wedge x_k > 0 &\rightarrow x_{k+1} - x_k = 10 \frac{1 - e^{-x_k}}{1 + e^{-x_k}} > 0 \\ u_k = 0 \wedge x_k < 0 &\rightarrow x_{k+1} - x_k = 10 \frac{1 - e^{-x_k}}{1 + e^{-x_k}} < 0 \end{aligned} \tag{26}$$

The model has been identified with a fuzzy identification method, resulting in a fuzzy model. A sampling time of $T = 0.1$s was used.

Having divided the input space in an adequate number of membership functions, the partial derivatives of the system can be calculated by (5) and (6).

For each trajectory generated the algorithm above was used to adjust the values of the co-state variable. At the end of this process, the results were stored in the fuzzy inference system. Fig. 2 shows the values of the co-state variable generated by the fuzzy inference system as a function of the inputs state and time remaining to end.

This function may be linguistically interpreted as follows:

–When the initial state equals the final state, the co-state value is zero everywhere.

–The system symmetry corresponds to one co-state symmetry.

–Because the system is unstable, less energy consumption is obtained if the control system drives the state variable to the equilibrium point ($x = 0$) as soon as possible – and with more strength the shorter is the remaining time.

These conclusions could be achieved before the training process, through analysis of the system model. Incorporating them in the fuzzy inference system would result in a faster convergence process.

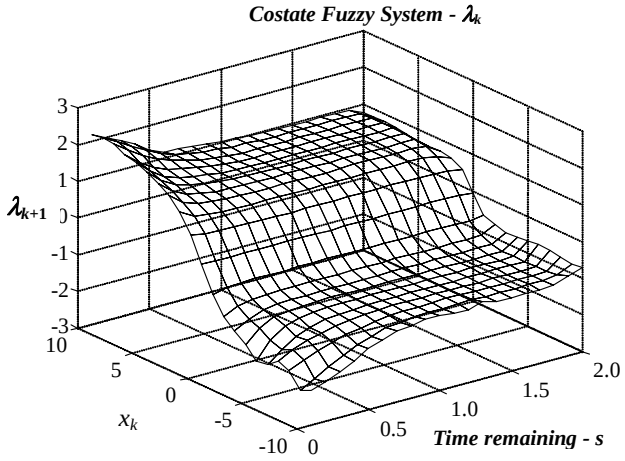

Fig. 2. Surface map of co-state with respect to state and time remaining

Fig. 3 shows test responses, comparing the described method (curve of × points) to an off-line optimization, obtained by use of the Quasi-Newton method with a mixed quadratic and cubic line search procedure (curve of + points).

In a system where noise is present, it is expected that a closed-loop strategy works better than an open-loop obtained by off-line optimization. Fig. 4 shows one test using off-line optimization control (+) and the co-state calculated by the fuzzy inference system (×), with Gaussian noise added to the state space variable. It is possible to observe that the proposed control strategy is more robust.

VI. CONCLUSION

In this paper the implementation of non-linear quadratic optimal control is described using a fuzzy logic methodology based on Pontryagin's Minimum Principle. A learning algorithm, interacting with the controlled system, calculates the values of co-state variables along a sequence converging to the optimal ones. The values found are saved in a fuzzy inference system. The methodology proposed allows for attaining calculation of optimal control actions on-line and in closed-loop. This fact should make possible to design feedback strategies more robust than standard off-line open loop optimal strategies, with respect to inaccuracies of the process model and unpredictable disturbances. Moreover, it should allow for tracking process changes.

VII. ACKNOWLEDGMENTS

This work was supported by the Portuguese "Fundação para a Ciência e a Tecnologia (FCT)" under grant POSI/SRI/41975/2001.

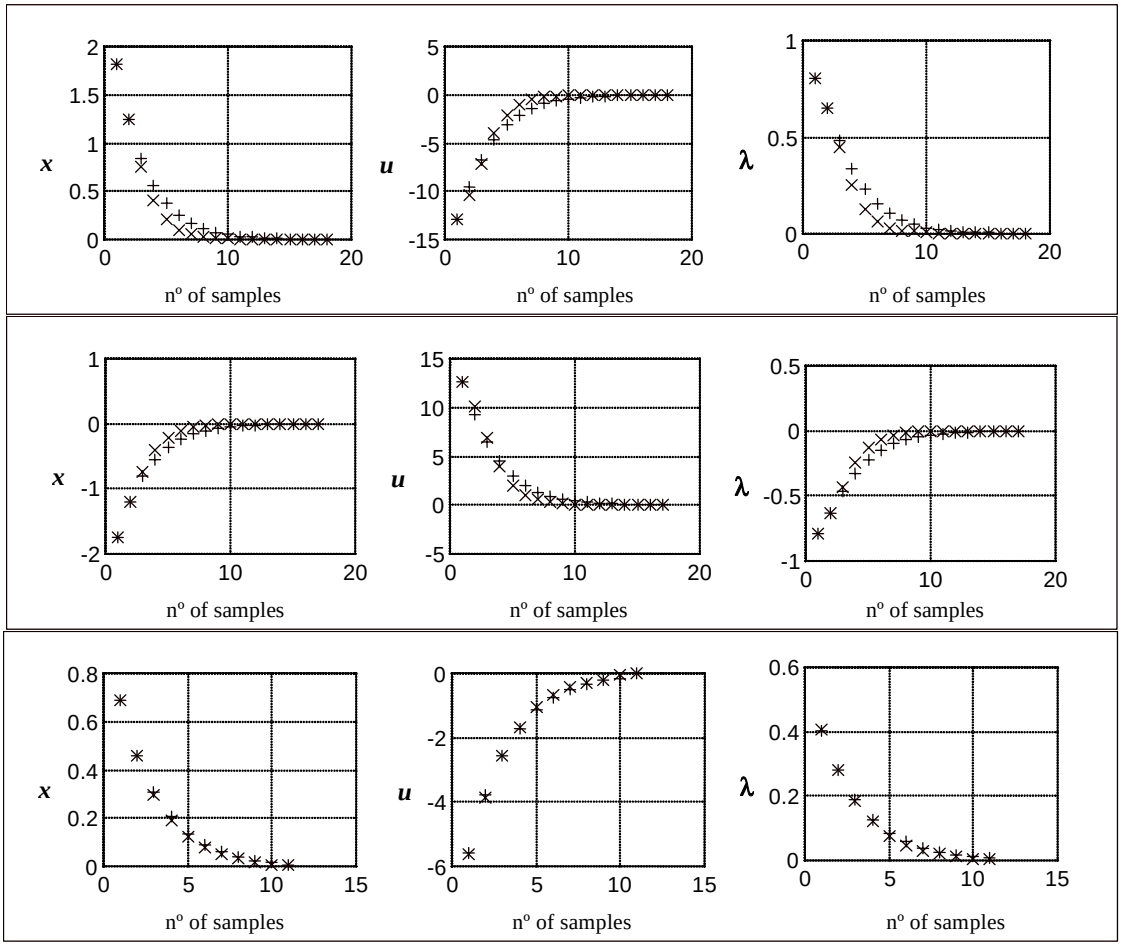

Fig. 3. Optimal trajectories: optimal co-state fuzzy control (×) and off-line numerical optimization (+)

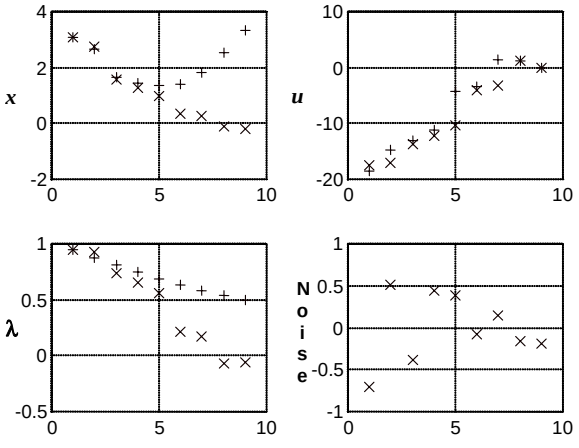

Fig. 4. Discrete trajectory with white noise added to the state space variable: optimal fuzzy control (×) and off-line optimization (+)

VIII. REFERENCES

[1] C. Chuen-Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part I", *IEEE Trans. On Syst., Man and Cybernetics*, 20, 1990a, pp. 404-418.

[2] C. Chuen-Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II", *IEEE Trans. on Syst., Man, and Cybernetic*, 20, 1990b, pp. 419-435.

[3] Li-Xin Wang, *A course in fuzzy systems and control*, NJ, Prentice-Hall PTR, 1997.

[4] B. Kosko, "Fuzzy system as universal approximators," in *Proceedings of the IEEE Int. Conf. Fuzzy Syst.*, San Diego, CA, Mar. 1992, pp. 1153–1162.

[5] H. Ying, "Sufficient conditions on general fuzzy systems as function approximators," *Automatica*, vol. 30, 1994, pp. 521–525.

[6] H. T. Nguyen, V. Kreinovich, and O. Sirisaengtaksin, "Fuzzy control as a universal control tool," *Fuzzy Sets Syst.*, vol. 80, 1996, pp. 71–86.

[7] J. A. Dickerson and B. Kosko, "Fuzzy function approximation with ellipsoidal rules," *IEEE Trans. Syst., Man, Cybern.*, vol. 26, 1996, pp. 542–560.

[8] J. L. Castro and M. Delgado, "Fuzzy systems with defuzzification are universal approximators," *IEEE Trans. Syst., Man, Cybern.*, vol. 26, 1996, pp. 149–152.

[9] L. X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-square learning," *IEEE Trans. Neural Networks*, vol. 3, 1992, pp. 807–814.