GAIN SCHEDULING BASED FUZZY CONTROLLER DESIGN

J.-S. Roger Jang and Ned Gulley
The MathWorks, Inc., 24 Prime Park Way, Natick, Mass. 01760
jang@mathworks.com, gulley@mathworks.com

Abstract— **This paper presents a gain scheduling based approach to the design of fuzzy controllers. We demonstrate that the Sugeno fuzzy controller is in fact a special case of gain scheduling where the feedback gains are adjusted ("scheduled") via membership functions in fuzzy if-then rules. When the desired gains at various operating points are available, we can apply the ANFIS learning algorithm [4] proposed earlier to construct a Sugeno fuzzy controller that can generated interpolated control actions. The validity of the proposed method is confirmed through simulation of two well-known control tasks: the cart-pole system and the ball-beam system.**

I. INTRODUCTION

The Mamdani [7] and the Sugeno [11, 10] fuzzy controllers have been used extensively in the fuzzy control community. The objective of the Mamdani fuzzy controller is to mimic a successful human operator who can express his/her expertise in terms of a set of fuzzy if-then rules. As a result, the underlying design method is often application-specific and not easily formalized. Moreover, the plant model is not required explicitly; the human operator is supposed have an implicit model in mind in order to guide the formulation of the fuzzy if-then rules. Therefore, design of a Mamdani fuzzy controller is similar to that of an expert system: it is more of a art and less of an engineering approach. This also accounts partly for the fact that fuzzy control is not quite accepted by the conventional control community.

In contrast, the Sugeno fuzzy controller is more efficient both in computation and adaptation [4]. More importantly, the Sugeno fuzzy controller can be viewed as a special case of gain scheduling where the feedback gains vary according to the membership function on the premise part of the fuzzy rule set. Based on this observation, we can easily incorporate some of the linear control design methods (such as linear quadratic optimal control and robust control) as an integrated part of our fuzzy control design procedure. If the plant under control is severely nonlinear the proposed fuzzy controller can outperform any of its constituent linear counterparts since nonlinear aspects of the plant have been taken into consideration.

This paper is organized into five sections. The next section introduces the Sugeno fuzzy controller and its learning algorithm. Section 3 explains the resemblance between the Sugeno fuzzy controller and gain scheduling, which leads to a new design method for the Sugeno fuzzy controller. We apply this method to control the cart-pole and the ball-beam systems; the simulation results are presented in Section 4. Section 5 gives a concluding remark and possible future directions.

II. SUGENO FUZZY CONTROLLER AND ITS LEARNING ALGORITHM

We will use the terms **fuzzy controller**, **fuzzy model**, and **fuzzy inference systems** interchangeably since they are all synonyms which are commonly used to refer to a fuzzy rule-based system.

A. Sugeno Fuzzy Model

The **Sugeno fuzzy model** was proposed by Takagi, Sugeno, and Kang [11, 10] in an effort to formalize a systematic approach to generating fuzzy rules from an input-output data set. A typical fuzzy rule in a Sugeno fuzzy model has the format

$$\textit{If } x \textit{ is } A \textit{ and } y \textit{ is } B \textit{ then } z = f(x,y),$$

where A and B are fuzzy sets in the antecedent; $z = f(x, y)$ is a crisp function in the consequent. Usually $f(x, y)$ is a polynomial in the input variables x and y, but it can be any other functions as long as it can appropriately describe the output of the system within the fuzzy region specified by the antecedent of the rule. When $f(x, y)$ is a first-order polynomial, we have the **first-order** Sugeno fuzzy model, which was originally proposed in [11, 10]. When f is a constant, we then have the **zero-order** Sugeno fuzzy model, which can be viewed either as a special case of the Mamdani fuzzy inference system [7] in which each rule's consequent is specified by a fuzzy singleton (or a pre-defuzzified consequent), or a special case of Tsukamoto's fuzzy model [12] in which each rule's consequent is specified by a membership function equal to a step function centered at the constant. Moreover, a zero-order Sugeno fuzzy model is functionally equivalent to a radial basis function network under certain minor constraints [5].

Consider a first-order Sugeno fuzzy inference system which contains two rules:

Rule 1: If X is A_1 and Y is B_1, then $f_1 = p_1 x + q_1 y + r_1$,
Rule 2: If X is A_2 and Y is B_2, then $f_2 = p_2 x + q_2 y + r_2$.

Figure 1 (a) illustrates graphically the fuzzy reasoning mechanism to derive an output f from a given input vector $[x, y]$. The **firing strengths** w_1 and w_2 are usually obtained as the product of the membership grades in the premise part, and the output f is the weighted average of each rule's output. More

0-7803-2125-1/94 $4.00 © 1994 IEEE

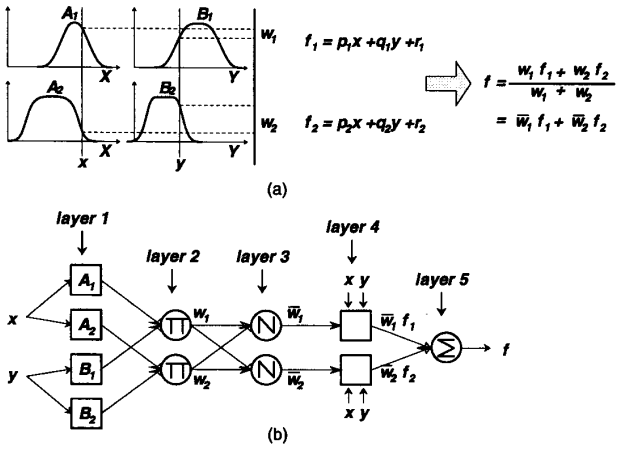

Figure 1: *(a) First-order Sugeno fuzzy model; (b) corresponding ANFIS architecture.*

explicitly, the output f can be expressed as

$$\begin{aligned} f &= \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2}, \text{ or} \\ &= \frac{\mu_{A_1}(x)\mu_{B_1}(y)(p_1 x + q_1 y + r_1) + \mu_{A_2}(x)\mu_{B_2}(y)(p_2 x + q_2 y + r_2)}{\mu_{A_1}(x)\mu_{B_1}(y) + \mu_{A_2}(x)\mu_{B_2}(y)}. \end{aligned} \tag{1}$$

The Sugeno fuzzy model is well-suited for adaptive network based learning introduced below.

B. ANFIS

To facilitate the learning (or adaptation) of the Sugeno fuzzy model, it is convenient to put the fuzzy model into the framework of adaptive networks that can compute gradient vectors systematically. The resultant network architecture, as shown in Figure 1 (b), is called **ANFIS** (Adaptive Network based Fuzzy Inference System). Nodes within a layer of ANFIS perform similar tasks that are specified by their node functions, as described below. (Note that O_i^j denotes the output of the i-th node in layer j.)

Layer 1 Each node in this layer generates a membership grades of a linguistic label. For instance, the node function of the i-th node might be

$$O_i^1 = \mu_{A_i}(x) = \frac{1}{1 + [(\frac{x - c_i}{a_i})^2]^{b_i}}, \tag{2}$$

where x is the input to node i; A_i is the linguistic label (*small*, *large*, etc.) associated with this node; and $\{a_i, b_i, c_i\}$ is the parameter set that changes the shapes of the membership function. Parameters in this layer are referred to as the **premise parameters**.

Layer 2 Each node in this layer calculates the firing strength of each rule via multiplication:

$$O_i^2 = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \ i = 1, 2. \tag{3}$$

Layer 3 The i-th node of this layer calculates the ratio of the i-th rule's firing strength to the sum of all rules' firing strengths:

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \ i = 1, 2. \tag{4}$$

Layer 4 Node i in this layer has the following node function

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i), \tag{5}$$

where $\overline{w}_i$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer will be referred to as the **consequent parameters**.

Layer 5 The single node in this layer computes the overall output as the summation of all incoming signals:

$$O_1^5 = \textit{overall output} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{6}$$

Thus we have constructed an adaptive network in Figure 1(b) which is functionally equivalent to a fuzzy inference system in Figure 1(a). The basic learning rule of ANFIS is the back-propagation gradient descent [13], which calculates error signals (the derivative of the squared error with respect to each node's output) recursively from the output layer backward to the input nodes. This learning rule is exactly the same as the back-propagation learning rule used in the common feedforward neural networks [9].

From the ANFIS architecture in Figure 1, it is observed that given the values of premise parameters, the overall output f can be expressed as a linear combinations of the consequent parameters:

$$\begin{aligned} f &= \overline{w}_1 f_1 + \overline{w}_2 f_2 \\ &= (\overline{w}_1 x) p_1 + (\overline{w}_1 y) q_1 + (\overline{w}_1) r_1 \\ &\quad + (\overline{w}_2 x) p_2 + (\overline{w}_2 y) q_2 + (\overline{w}_2) r_2. \end{aligned} \tag{7}$$

Based on this observation, we have proposed a hybrid learning algorithm [3, 4] which combines the gradient descent and the least-squares method for effective search of a set of optimal parameters. Both on-line and off-line learning paradigms were developed and reported in [4].

III. FROM GAIN SCHEDULING TO FUZZY CONTROL

The overall output f in equation (7) can be rewritten as

$$\begin{aligned} f &= \frac{w_1}{w_1 + w_2}(p_1 x + q_1 y + r_1) + \frac{w_2}{w_1 + w_2}(p_2 x + q_2 y + r_2) \\ &= (\frac{w_1 p_1 + w_2 p_2}{w_1 + w_2}) x + (\frac{w_1 q_1 + w_2 q_2}{w_1 + w_2}) y + (\frac{w_1 r_1 + w_2 r_2}{w_1 + w_2}). \end{aligned}$$

If we assume both r_1 and r_2 are zero, then the above equations actually represent a state feedback controller with gain scheduling [1, 8], where the dependency of the gain vector $\mathbf{k}$ on the operating point $[x, y]$ is expressed explicitly as

$$\begin{aligned} \mathbf{k} &= \left[\frac{w_1 p_1 + w_2 p_2}{w_1 + w_2}, \frac{w_1 q_1 + w_2 q_2}{w_1 + w_2}\right] \\ &= \left[\frac{\mu_{A_1}(x)\mu_{B_1}(y)p_1 + \mu_{A_2}(x)\mu_{B_2}(y)p_2}{\mu_{A_1}(x)\mu_{B_1}(y) + \mu_{A_2}(x)\mu_{B_2}(y)}, \right. \\ &\quad \left. \frac{\mu_{A_1}(x)\mu_{B_1}(y)q_1 + \mu_{A_2}(x)\mu_{B_2}(y)q_2}{\mu_{A_1}(x)\mu_{B_1}(y) + \mu_{A_2}(x)\mu_{B_2}(y)}\right]. \end{aligned}$$

This allows us to employ the common technique of linearizing a plant at certain operating points and then finding the corresponding gains via linear control methods such as robust

and/or optimal design. These gains are then blended via membership functions to form a complete first-order Sugeno fuzzy controller (without constant terms).

Figure 2 demonstrates a simple example, where the state space of a second-order system is divided into nine regions, each of which corresponds to a fuzzy if-then rule. Basically, there are two ways of using gain scheduling in fuzzy control, as discussed below.

- If the number of operating points is small (Figure 2 (a)), then these operating points are usually located where the membership functions achieve unity and the number of operating points are equal to the number of rules. Usually the gains at each of these points are taken as the coefficients of the output equation of the corresponding fuzzy rule. In other words, the fuzzy if-then rules are constructed directly from each of the operating point and corresponding gains, such that the controller can generate an exact desired control action at each point and an interpolated control action between points.

  We would like the fuzzy controller to reproduce the exact gains at each of the operating points.
- If the number of operating points are large (Figure 2 (b)), then these points (and corresponding control actions) are taken as the training data for identifying a fuzzy logic controller to approximate the input-output mapping.

Simply speaking, the first situation is an **interpolation problem** while the second one is an **approximation problem**. This is an effective method of designing a fuzzy controller, though we need to identify a model for the plant first. If the overall model of the plant is not available, a shortcut is to perturb the plant at these operating points and numerically derive the linearized models.

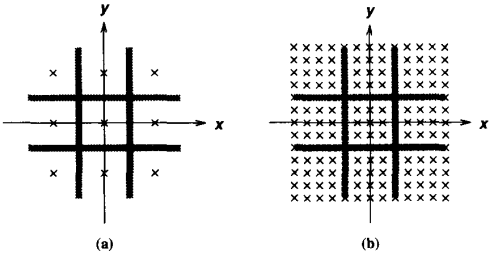

Figure 2: *Applying gain scheduling to find a fuzzy controller: (a) interpolation if the number of operating points is small; (b) approximation if the number of operating points is large .*

In summary, the proposed gain scheduling based design approach can be described as follows.

1. Define domain of interest in the state space and set operating points within the domain of interest.
2. For each of the operating points, find the linearized model of the plant and then identify the feedback gains and corresponding control action via linear control techniques, such as linear quadratic optimal control or robust control.
3. Use ANFIS to identify a set of parameters that can best reproduce the control action at each operating point.

This design method is used in the following section to find fuzzy controllers for the cart-pole and the ball-beam systems.

IV. APPLICATION EXAMPLES

A. The Cart-Pole System

Figure 3 shows the schematic diagram of the cart-pole system (also known as the inverted pendulum), where a rigid pole is hinged to a cart through a free joint with only one degree of freedom, and the cart slides on a smooth surface to its right or left depending on the force (shown as an arrow in Figure 3) exerted on it. The control goal is to move the cart to a target position (shown as a triangle in Figure 3) while keeping the pole balanced.

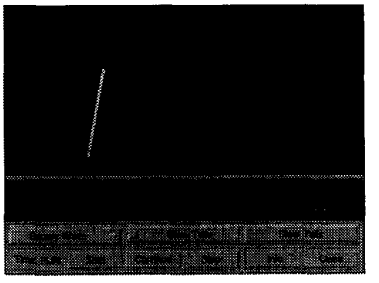

Figure 3: *The cart-pole system with the target position shown on the right hand side.*

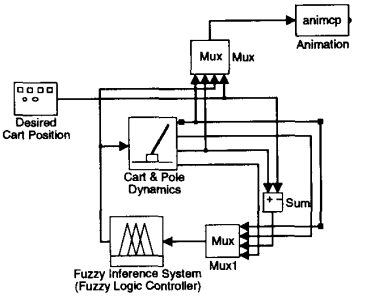

Figure 4: *SIMULINK block diagram for simulation and animation of the cart-pole system.*

The inverted pendulum system is characterized by four state variables: θ (angle of the pole with respect to the vertical axis), $\dot{\theta}$ (angular velocity of the pole), z (position of the cart on the track) and $\dot{z}$ (velocity of the cart). These four state variables are related via the following dynamical equations [2, 6]:

$$\begin{cases} \ddot{\theta} &= \dfrac{g\sin\theta+\cos\theta\left(\frac{-u-ml\dot{\theta}^2\sin\theta}{M+m}\right)}{l\left(\frac{4}{3}-\frac{m\cos^2\theta}{M+m}\right)}, \\ \ddot{z} &= \dfrac{u+ml(\dot{\theta}^2\sin\theta-\ddot{\theta}\cos\theta)}{M+m}, \end{cases} \tag{8}$$

where g (acceleration due to gravity) is 9.8 $meter/sec^2$, M (mass of cart) is 1.0 kg, m (mass of pole) is 0.1 kg, l (half length of pole) is 0.5 m, and u is the applied force in Newtons.

The domain of interest is selected as a hypercube $[-0.3, 0.3] \times [-1, 1] \times [-3, 3] \times [-3, 3]$ in the four-dimensional state space $\mathbf{x} = (\theta, \dot{\theta}, z, \dot{z})$. By choosing 5 equidistant points within the range of each of the four state variable, we can obtain 625 $(= 5^4)$ grid points, which are used as the representative operating points for this problem.

For each operating point at $\mathbf{x} = \mathbf{x_0}$, we can derive a linearized model using the MATLAB command *linmod*. If we adopt the design method for linear quadratic optimal control, the MATLAB command *lqr* can be used to find the optimal gain vector $\mathbf{k}$ such that the feedback law $\mathbf{u} = -\mathbf{kx}$ minimizes the quadratic cost function

$$J = \int_{t=0}^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \tag{9}$$

subject to the constraint of the linearized model

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}.$$

obtained at the operating point $\mathbf{x} = \mathbf{x_0}$. Here we set both $\mathbf{Q}$ and $\mathbf{R}$ equal to identity matrices in our simulation.

After obtaining 625 training data pairs, the next step is to find a fuzzy controller that can hopefully reproduce these data. This is done by using ANFIS with 16 rules (104 parameters), where each input is assigned two membership functions. For simplicity, we perform only a single epoch of ANFIS learning, yielding a root mean squared error of 0.030. Further reduction of approximation error is very likely if learning epochs are increased.

Figure 3 is the SIMULINK block diagram of our simulation and animation environment for the cart-pole system. (The *animcp* block is for animation using handle graphics.) Figure 5 (a) plots the tracking performance when the desired cart position (shown as a dotted line) varies as a sinusoidal wave (when $t \leq 33$), a square wave (when $33 < t \leq 66$), and a tooth wave (when $t > 66$). The diagram shows that the actual cart position (shown as a solid line) can follow the desired signal satisfactorily while keeping the pole balanced. Figure 5 (b) is the pole angle and angular velocity; (c) is the control force on the cart.

B. The Ball-Beam System

Figure 6 shows the schematic diagram of the ball-beam system, where a ball is rolling frictionlessly on a beam and a motor is used to tilt the beam in order to send the ball to a desired location (shown as a small hollow triangle in Figure 6). The ball-beam system is also characterized by four state variables: z (position of the ball), $\dot{z}$ (velocity of the ball), θ (angle of the beam with respect to the horizontal axis), and $\dot{\theta}$ (angular velocity of the beam). These four state variables are related via the following second-order differential equations:

$$\begin{cases} \ddot{z} &= 0.7143(z\dot{\theta}^2 - 9.81 sin\theta), \\ \ddot{\theta} &= u, \end{cases} \tag{10}$$

where u is the torque applied at the beam. (Note that this system is feedback linearizable, but we do not exploit this property in the following discussion.)

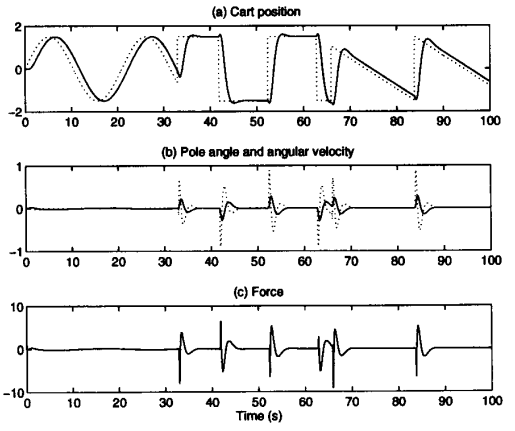

Figure 5: *Simulation results for the cart-pole system: (a) desired (dotted line) and actual (solid line) cart positions; (b) pole angle (solid line) and angular velocity (dotted line); (c) control force.*

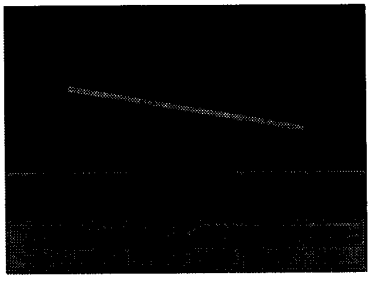

Figure 6: *The ball-beam system.*

The domain of interest is selected as a hypercube $[-1.5, 1.5] \times [-1.5, 1.5] \times [-0.2, 0.2] \times [-0.4, 0.4]$ in the four-dimensional state space $\mathbf{x} = (z, \dot{z}, \theta, \dot{\theta})$. Again, we choose 625 grid points within the domain of interest as the representative operating points; a similar approach which minimizes the same objective function (equation (9)) subjected to linearized constraints is used to find the gains and the control action at each point. After a single epoch of batch learning, a 16 rules ANFIS is able to reproduce the training data with a root mean squared error of 0.029.

Figure 6 is the SIMULINK block diagram of our simulation and animation environment for the ball-beam system. Figure 8 (a) shows the tracking test where the desired signal (shown as a dotted line) is the same as the one used for the cart-pole system. Our fuzzy controller can drive the ball to follow the desired signal satisfactorily. Figure 8 (b) is the corresponding beam angle and angular velocity; (c) is the exerted torque on the beam.

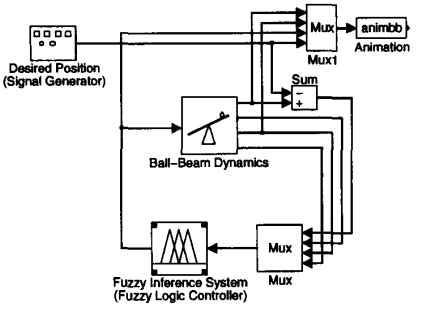

Figure 7: *SIMULINK block diagram for simulation and animation of the ball-beam system.*

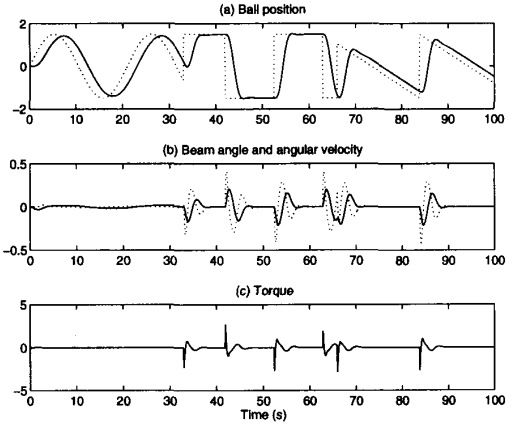

Figure 8: *Simulation results for the ball-beam system: (a) desired (dotted line) and actual (solid line) ball positions; (b) beam angle (solid line) and angular velocity (dotted line); (c) control torque.*

V.. CONCLUSION AND FUTURE DIRECTIONS

Based on the resemblance between gain scheduling and the Sugeno fuzzy controller, we have proposed a design method for fuzzy controllers of Sugeno's type that can take advantage of a number of linear control techniques. The feasibility of this method is confirmed via simulations of the cart-pole and the ball-beam systems.

Unlike design methods for the Mamdani fuzzy controller, our method does need an exact model of the plant. If the model is nonlinear, then the proposed method constructs a fuzzy controller that performs the desired gain scheduling. On the other hand, if the model is linear, then the constructed fuzzy controller reduces to a simple linear controller. Conceptually speaking, becasue of the extra consideration of nonlinearity at different operating points, the proposed fuzzy controller can outperform the corresponding linear controller in minimizing the same quadratic objective function if the plant under control is severely nonlinear. However, this is achieved at the cost of extra computational complexity both in design and application of the fuzzy controller.

Besides gain scheduling, we believe that various nonlinear control design techniques can be embedded into the design process of the Sugeno fuzzy controller. For example, to design a fuzzy controller that can deal with plant uncertainty or time-varying characteristics, we can employ adaptive control, sliding mode, or H-∞ techniques. This kind of integration will further blur the boundary between fuzzy (Sugeno's type) and conventional control. On the other hand, the Mamdani fuzzy inference system will still be better suited for fuzzy decision making or fuzzy expert systems which do not require a plant model explicitly.

VI. REFERENCES

[1] K. J. Astrŏm and B. Wittenmark. *Adaptive Control.* Addison-Wesley Publishing Company, 1989.

[2] R. H. Cannon. *Dynamics of Physical Systems.* McGraw-Hill, New York, 1967.

[3] J.-S. Roger Jang. Fuzzy modeling using generalized neural networks and Kalman filter algorithm. In *Proc. of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, pages 762–767, July 1991.

[4] J.-S. Roger Jang. ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Trans. on Systems, Man, and Cybernetics*, 23(03):665–685, May 1993.

[5] J.-S. Roger Jang and C.-T. Sun. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Trans. on Neural Networks*, 4(1):156–159, January 1993.

[6] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems.* Wiley-Interscience, 1972.

[7] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13, 1975.

[8] W. J. Rugh. Analytical framework for gain scheduling. *IEEE Control Systems Magazine*, 11(1):79–84, 1991.

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, volum 1*, chapter 8, pages 318–362. The MIT Press, 1986.

[10] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28:15–33, 1988.

[11] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on Systems, Man, and Cybernetics*, 15:116–132, 1985.

[12] Y. Tsukamoto. An approach to fuzzy reasoning method. In Madan M. Gupta, Rammohan K. Ragade, and Ronald R. Yager, editors, *Advances in Fuzzy Set Theory and Applications*, pages 137–149. North-Holland, Amsterdam, 1979.

[13] P. Werbos. *Beyond regression: New tools for prediction and analysis in the behavioral sciences.* PhD thesis, Harvard University, 1974.