Multi Dimensional Second Order Defuzzification Algorithm (M-SODA)

Nirav Dharia[1], Jagadeesh Gownipalli[1], Bogdan M.Wilamowski[1] and Okyay Kaynak[2]
[1]University of Idaho, College of Engineering, Boise, U.S.A
[2] Bogazici University, Electrical and Electronic Engineering Department, Istanbul, Turkey

ABSTRACT

Multi Dimensional Second Order Defuzzification Algorithm (SODA) minimizes the hardware requirement and ensures easier implementation of a fuzzy controller as compared to the LUT and the triangular membership approach when extended to multiple dimensions. In this paper we have put forward the extension of the SODA to multiple dimensions. Fuzzy controllers can handle multiple inputs and outputs however size of the fuzzy table grows exponentially with each input added. In this paper we put forward the extension of SODA into 4 Dimension using three six bits input, (8 x 8 x 8) and (4 x 4 x 4) fuzzy look up tables (LUT) were used each having 512 and 64 values respectively. The system uses three inputs, the upper three bits of each input are used for addressing the exact position of the nearest data point in the look up table, and the rest of the input is the information about the membership function. The advantage of this method over other defuzzification methods like the LUT and the triangular membership function approach is its extensibility to multiple dimensions with the efficient usage of a relatively smaller look up table, which minimizes the hardware requirement and ensures easier implementation. In this paper we also demonstrate the extension of the method to more than 4 dimensions.

1 INTRODUCTION

Fuzzy Logic, a direct extension of fuzzy system that Lotfi Zadeh introduced in 1965 [1] resembles human reasoning in its use of approximate information and uncertainty to generate decisions. Fuzzy logic has been applied in a wide variety of disciplines such as system identification, classification, control and decision support [2]. Fuzzy logic mathematically represents uncertainty and vagueness and provides formalized tools for dealing with imprecision in many problems. Stability analysis and systematic design are the most important issues for fuzzy control systems [4][5][12][13]. Fuzzy logic helps in using information, which is not precise or is incomplete. Knowledge can be expressed more naturally using Fuzzy logic and thus problems requiring decision-making can be greatly simplified. Fuzzy logic can be considered to be the superset of Boolean logic. Fuzzy logic is a very useful tool to handle approximate information in a very systematic way and makes it an attractive option when it comes to nonlinear systems control.

The system can be classified using the Fuzzy If-Then rule and this classification method differentiates Fuzzy Systems from other approximation methods [1]. Fuzzy systems can be implemented by converting human knowledge to Fuzzy Rules [2][3]. In recent years, digital designers have been very interested in fuzzy systems because of its ability to be applicable in non-linear controllers [3]. Using the established methods [1][7][9] for defuzzification, the control surfaces obtained are rough and there extension to multiple dimensions results in increasing the size of the hardware and thus increasing the computational complexities to overcome this difficulty we proposed the use of second order defuzzification in our previous paper [2]. Here we have proposed the extension of this method to multiple dimensions. We have demonstrated the extension of this method to 4^{th} dimension and also proposed the extension of this method to higher dimensions. Using this approach five-dimensional implementation on a FPGA [6] can be achieved (3x5=15 bit LUT address), as the size of the LUT is much smaller as compared to the ones used in other approaches.

We considered a three-dimension surface: s = cos (x) * sin (y) * cos (z) to compare the results obtained by LUT, Triangular and M-SODA. The required control surface is shown in the Fig. 1, the plot of a 4 dimensional figure is obtained by keeping one of the dimensions constant. Here *z* is kept constant and the plot is obtained to compare the results.

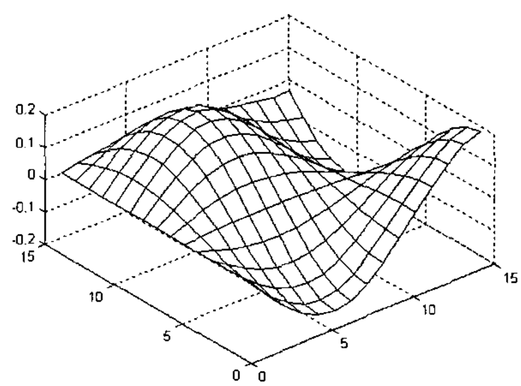

Fig. 1 Required control surface of function s=cos (x)*sin (y)*cos (z)

0-7803-7474-6/02/$17.00 ©2002 IEEE

Classical Look Up Table approach

Using the classical LUT approach, using a 4x4x4 and 8x8x8 LUT the results obtained are shown in the Fig. 2 and Fig. 3, which shows that the surface obtained are very rough and are unacceptable in most applications. This plot was obtained by keeping one of the dimensions constant. To obtain a control surface as smooth as the one obtained by SODA in Fig. 19 the size of the LUT will increase considerably.

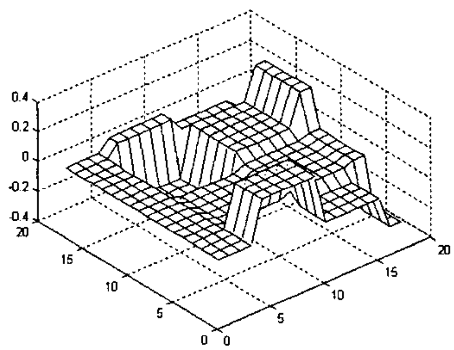

Fig. 2 Look up table results (4x4x4) LUT

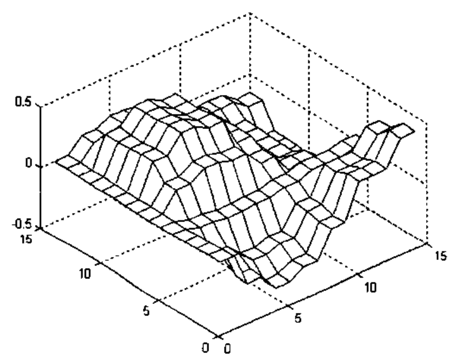

Fig. 3 Look up table results (8x8x8) LUT

In the next section we discuss the Triangular membership function where the error is reduced considerably as compared to the Classical LUT approach.

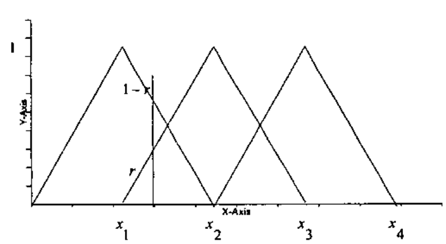

Fig. 4 Triangular membership function

Triangular Membership Function Approach

We have used the Tagagi Sugeno method [10][11] as shown in Fig. 4 having 8 membership function as we have lower 3 bits providing information about the membership functions. The results obtained using the Triangular Membership Function approach with 4x4x4 and 8x8x8 LUT [8] are shown in the Fig. 5 and Fig. 6. The results obtained using this method were satisfactory but at the same time the error generated using this method is unacceptable for most applications. The error plots are shown in Fig. 7 to Fig. 8.

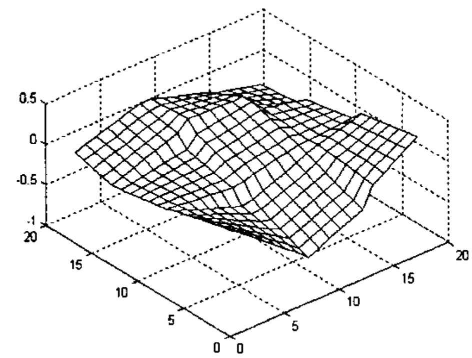

Fig. 5 Triangular approximation (4x4x4) LUT

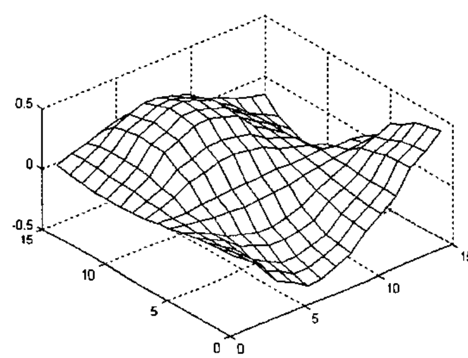

Fig. 6 Triangular approximation (8x8x8) LUT

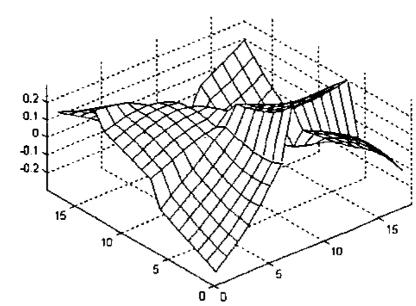

Fig. 7 Error surface for the Triangular membership function approach (4x4x4) LUT

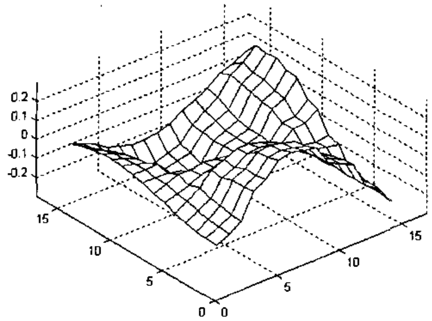

Fig. 8 Error surface for the Triangular membership function approach (8x8x8) LUT

With increasing the size of the LUT the error reduces as can be observed from Fig. 7 and Fig.8 but at the same time the hardware implementation and computational complexities increases. Using the second order defuzzification algorithm proposed in this paper we get better results for a 3-bit input with a substantial reduction in errors [2].

2 SECOND ORDER DEFUZZIFICATION ALGORITHM (SODA)

To explain this concept we shall first list the equations derived for a 1-D case and then consider the 2-dimension approach.

Algorithm in 1-D

Consider a Second order equation of the form

$$y(x) = x(ax + b) + c \qquad (1)$$

Where the values of a, b and c are as follows:

$$c = y_2 \qquad (2)$$

$$b = y_3 - y_2 \qquad (3)$$

$$a = 0.25(y_1 - y_2 - y_3 + y_4) \qquad (4)$$

The detailed derivation of the above equations can be obtained in our previous paper [2]. The equations are derived referring to the Fig. 9.

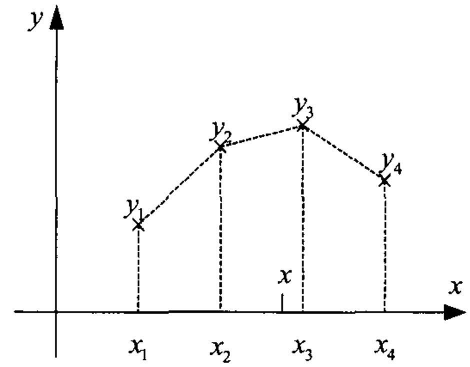

Fig 9 Second Order Defuzzification Algorithm in 1-D

Application of SODA to Single dimension

We considered a simple $y = \sin(x)$ function and applied the Classical LUT [8], Linear approximation and SODA to obtained results depicted in Fig. 10-12. The dotted line shows the plot of the original function.

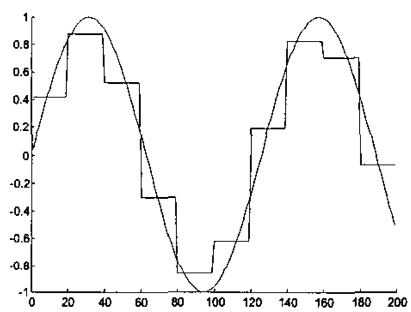

Fig. 10. Data stored in LUT

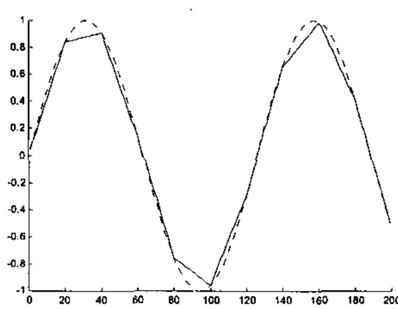

Fig. 11 Linear approximation

The above Figs. 10 and 11 shows that the surface obtained by using the LUT and Linear approximation are not smooth. Now applying the concept of the SODA explained in our previous paper [2] we have plot the curve shown in Fig. 12. As we can see the obtained surface is very smooth and almost similar to the required surface. The error plot shown in Fig. 13 proves the error obtained by this method is very small.

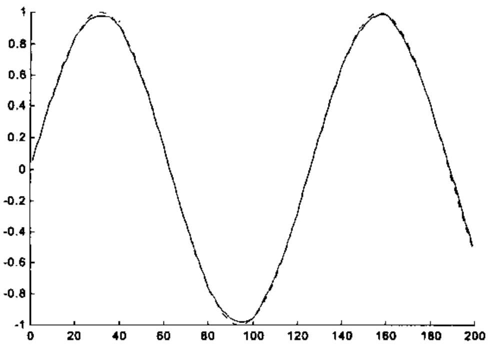

Fig. 12 Second Order Defuzzification for 1-D

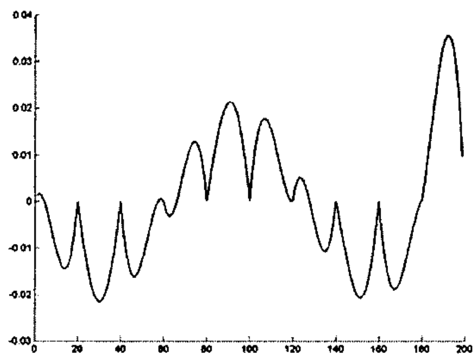

Fig. 13 Error using Second Order method in 1-D

Extension of SODA to 2-Dimension

Special approach has to be taken to extend the concept into multi-dimension. Here we are considering only 2-dimension and extension of the method to higher dimensions is discussed later. The fuzzification interface takes two 6 bits input from an analog to digital converter of which the higher 3-bits are the address bits and the lower bits provide information about the membership index of the input fuzzy groups. The membership function used here is the symmetric triangular membership function as shown in the Fig. 4. The points marked in the Fig. 14, represent the data points in the LUT [8] and the point marked 'X' is the input.

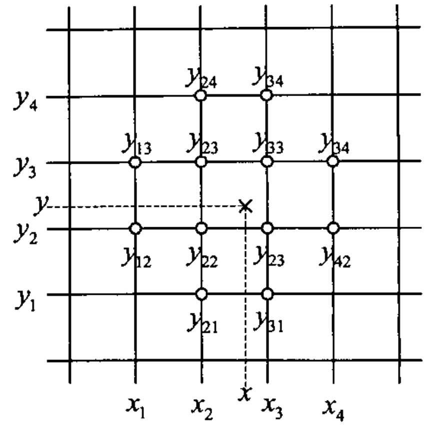

Fig. 14 Twelve points used for calculation surrounding the input point

Now in this algorithm the approximation is done both in the X and Y direction, using the 8 adjacent points in the X and Y directions shown in the Fig. 14. Following the algorithm and using equations (17) and (18) four points in each direction are approximated from the 8 adjacent points as shown the dotted lines in Fig. 15.

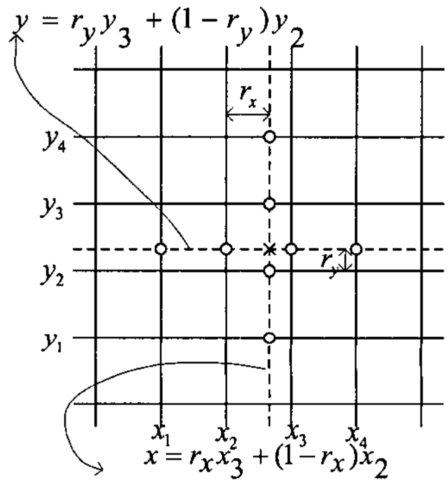

Fig. 15 Reduction of twelve points from the Fig. 17 to 8 points for calculating first and second order derivative.

Now these 4 points are approximately in line with the required input point, in both the directions. Thus the second order defuzzification algorithm applied to single dimension is applied in both X and Y direction separately following the same procedure explained in the previous section and 2 points are obtained. Taking the average of these two points we obtain the final result and the entire procedure is followed for all the input data elements. Several tests were carried out for testing the SODA and the results are listed in the following section.

Extension to Three Dimensions

The main purpose of this paper was to put forward the concept of extension to multiple dimensions. Here we put forward the extension of the algorithm to 3 dimensions, however this is done by finding the required points in two dimension say x and y, keeping the value in z constant. Thus the points are calculated varying two dimensions at a time.

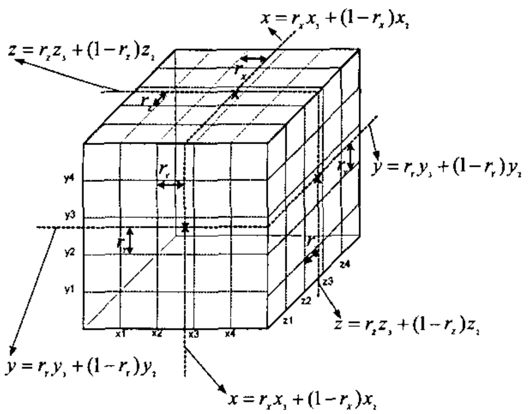

Fig. 16 Three-dimensional representation.

The above figure shows the three-dimensional representation.

$$x = r_x x_3 + (1 - r_x) x_2 \qquad (17)$$

$$y = r_y y_3 + (1 - r_y) y_2 \qquad (18)$$

$$z = r_z y_3 + (1 - r_z) z_2 \qquad (19)$$

Using the Equations (17) (18) and (19) the required point is obtained using 36 points, 12 in each dimension. Now using these 12 points in each dimension 8 more points are calculated as discussed in the 2-Dimension approach. These 8 points are used to calculate one point marked 'x' using SODA in each dimension and thus we end up with 6 points which are averaged out to find out the required point. This concept can be applied to higher dimension keeping two dimensions as variable and keeping the remaining constant, though the accuracy of the algorithm might decrease with increase in dimensions.

Experimental Results

The sample surface is s = sin (x) * sin (y) * cos (z) as discussed in the Introduction and the original surface is shown in Fig. 1. We first plot the data points that are stored in the 4x4 and 8x8x8 LUT shown in figures 17 and 18 respectively.

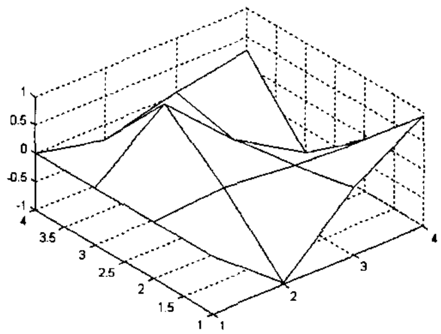

Fig. 17 Stored data in the 4x4x4 look up table (2 bits per input)

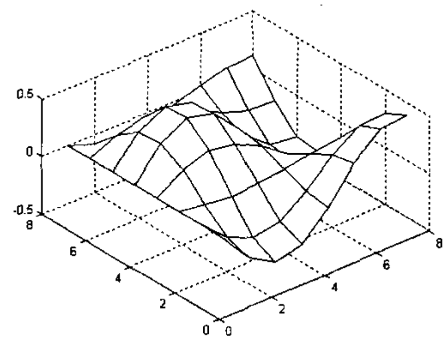

Fig. 18 Stored data in the 8x8x8 look up table (3 bits per input)

Results obtained by using the proposed algorithm are shown in the Figs. 19 and 20 using the 4x4 and 8x8 LUT [9] respectively. The results were obtained by carrying out MATLAB simulations.

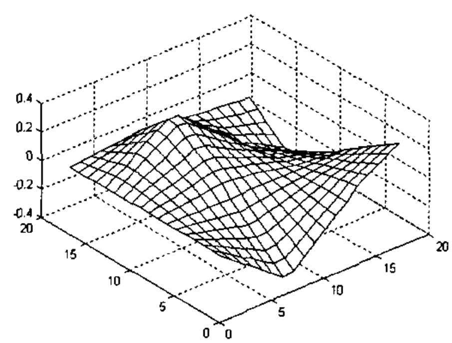

Fig. 19 Second Order Defuzzification (4x4x4) (2 bits per input)

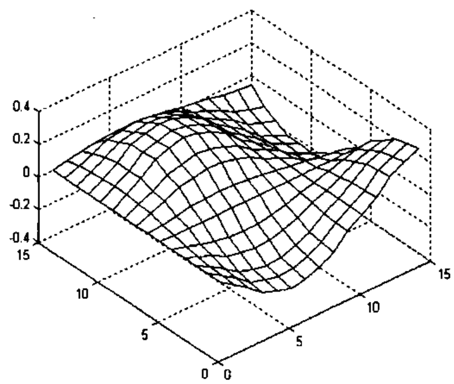

Fig. 20 Second Order Defuzzification (8x8x8)

As we can see from the above figures the surface obtained is very smooth.

3 CONCLUSION

Most of the Fuzzy Controllers are used where the control surfaces is not smooth and thus cannot be used for application where one requires precision control. By using the proposed SODA method we observe that control surfaces as shown in Figs. 19 and 20 are very smooth and relatively closer to the original stored surface in Fig. 1 as compared to the control surfaces obtained by using LUT approach Figs. 2 and 3, Triangular Membership function approach Figs. 5 and 6. This can also be proved by comparing the error plots obtained for the Triangular Membership function approach Figs. 7 to 8 and SODA Figs. 21 to 22.

The size of the LUT is reduced considerably as compared to the Classical LUT and triangular membership function for obtaining such smooth surface. Using this method there is a lot reduction in memory requirement, which reduces computational complexities and makes hardware implementation easier for higher dimensions.

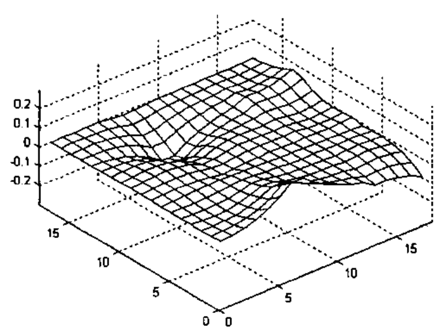

Fig. 21 Error surface for the proposed SODA method (4x4x4) LUT (2 bits per input)

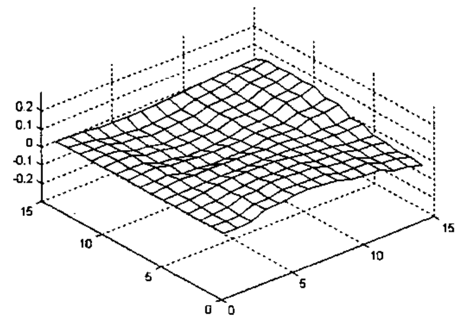

Fig. 22 Error surface for the proposed SODA method (8x8x8) LUT (3 bits per input)

4 REFERENCES

[1] L. A. Zadeh, Fuzzy sets. Information and Control, New York, Academic Press vol 8, pp. 338-353, 1965.

[2] Nirav Dharia, Jagadeesh Gownipalli, Okyay Kaynak and Bogdan M.Wilamowski "Fuzzy Controller With Second Order Defuzzification Algorithm" 2002 IEEE World Congress On Computational Intelligence, Honolulu, Hawaii, May 12-17, 2002.

[3] Mike McKenna and B. M Wilamowski, " Implementing a Fuzzy System on a Field Programmable Gate Array ", International Joint Conference on Neural Networks (IJCNN'01), pp. 189-194, Washington DC, July 15-19, 2001.

[4] Passino, K. M., S. Yurkovich, Fuzzy Control, Addison-Wesley, 1998.

[5] Wang, L., Adaptive Fuzzy Systems and Control, Design and Stability Analysis, PTR Prentice Hall, 1994.

[6] Patyra, M.J.; Grantner, J.L.; Koster, K. "Digital fuzzy logic controller: design and implementation" IEEE Transactions on Fuzzy Systems, Volume: 4 Issue: 4 Page(s): 439 -459, Nov. 1996

[7] Hao Ying "General SISO Takagi–Sugeno Fuzzy Systems with Linear Rule Consequent Are Universal Approximators", IEEE Transaction on Fuzzy System, vol. 6, no. 4, Page(s): 582 -587 NOVEMBER

[8] D. Tikk and P.Baranyi,"Comprehensive Analysis of a New Fuzzy Rule Interpolation Method IEEE Transaction on Fuzzy System, vol. 8, no. 3, Page(s): 281 -296,JUNE 2000 281

[9] Tan H., R. Sandige, and B. Wilamowski, "Hardware implementation of PLD-based fuzzy logic controllers using a look-up table technique" presented at ANNIE'94 - Artificial Neural Networks in Engineering, St. Louis, Missouri, USA, November 13-16, 1994; also in Intelligent Engineering Systems Through Artificial Neural Networks vol. 4, pp. 89-94, ed. C. H. Dagli, B. R. Fernandez, J. Gosh, R.T. S. Kumara, ASME PRESS, New York 1994

[10] Hornik, K., "Multilayer Feedforward Networks are Universal Approximators," Neural Networks, v.2, pp 359-366, 1989.

[11] T. Takagi and M. Sugeno, Derivation of Fuzzy Control Rules from Human Operator's Control Action. Proc. of the IFAC Symp. on Fuzzy Inf Knowledge Representation and Decision Analysis, pp. 55-60, July 1989.

[12] Wilamowski, B. M., "Neural Networks and Fuzzy Systems" chapters 124.1 to 124.8 in The Electronic Handbook. CRC Press 1996, pp. 1893-1914.

[13] Output feedback robust H/sup /spl infin// control of uncertain fuzzy Dynamic systems with time varying delay Kap Rai Lee; Jong Hae Kim; Eun Tae Jeung; Hong Bae Park Page(s): 657 –664.