Fuzzy System with Increased Accuracy Suitable for FPGA implementaion

Kannan Govindasamy, Sandeep Neeli, Bogdan M. Wilamowski

Department of Electrical and Computer Engineering
Auburn University, Auburn, AL 36849
govinka, neelisa, wilambm@auburn.edu

***Abstract*— Fuzzy controllers are easy to design for complex control surfaces but produce rough control surfaces which might lead to unstable operation. On the other hand neural controllers are hard and complex to train but they produce very accurate output control surfaces compared to that of fuzzy controllers. The Neuro-Fuzzy controller proposed in this paper exploits the fuzzy systems nature of utilizing expert knowledge and also produces smooth surfaces by implementing it in neural networks. Monotonic sigmoid membership function is used to make the neural implementation an easy task. LM- Levenberg-Marquardt algorithm which is used for feed forward networks is implemented to train the neurons. A defuzzification with trigonometric approximation algorithm using LUT-Lookup Table is developed to implement in low cost microcontrollers to make the control system highly cost effective. It is shown through extensive simulations that the proposed model produces accurate and smooth control surfaces.**

Keywords- Neural network, fuzzy logic, LUT.

I. INTRODUCTION

Control systems for nonlinear system [1] still remains a challenge in modern control theory, when compared to control system for linear systems. The nonlinear system becomes more difficult to understand and analyze when the system becomes a function of time. A nonlinear system is usually linearised before a controller system can be designed for the system. This is more often done by adding a reverse nonlinear function to compensate the nonlinear behavior and make the input-output relationship more linear. An adaptive nonlinear system that has nonlinear characteristics which change with time is better managed by methods of computational intelligence such as Neural Network and Fuzzy systems [1][3][4][5]. The adaptive nature of these systems enables them to model any dynamic non-linear behavior of a control system.

Fuzzy systems utilize expert knowledge and perception based information in the form of set of rules. The dynamics of a system is generally complicated, but sometimes its behavior can be defined more conveniently using linguistic terms in which case fuzzy logic is the best option to model the system. Moreover, fuzzy systems are easy to design and implement in hardware. But, the major drawback of fuzzy controllers is that the control surfaces obtained from these systems are rough, which can cause unstable operation.

Artificial neural systems, on the other hand are known by their property of performing complex nonlinear mappings. They also produce smooth control surfaces unlike the fuzzy systems. Even though neural networks are widely implemented in software many applications require the hardware implementation. But, these systems require the computation of tangent hyperbolic activation functions. This involves division algorithms which are often too complex for simple microprocessors. So, a suitable combination of the neural architecture and the fuzzy architecture is proposed in this paper. It exploits the property of producing fine control surfaces from neural architecture and utilizing expert knowledge form fuzzy logic.

The Neuro-Fuzzy controller proposed in this work uses a fuzzy logic with monotonic sigmoid membership functions that makes the neural implementation a very easy task. LM-Levenberg-Marquardt algorithm [6] which is used for feed forward networks is implemented to train the neurons. Since, the implementation of neural networks in low cost microcontrollers is a challenging task, a defuzzification with trigonometric approximation algorithm using LUT [7][8] is developed to implement neurons in low cost microcontrollers to make the control system highly cost effective.

The rest of the paper is organized as follows. Section II provides the relevant background information on fuzzy systems and neural architectures. The proposed model is explained in section III. Also, the new LUT algorithm is discussed. Section IV concludes the work.

II. BACKGROUND

In this section the relevant background information on fuzzy systems and neural networks is provided.

A. *Fuzzy systems*

Fuzzy set system theory was introduced by Zadeh [9]. The fact that a decision (output) might depend on several factors (variables) is implemented using *membership functions.*

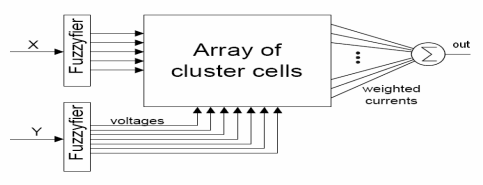


Figure 1. Block diagram of a TSK type fuzzy controller.

978-1-4244-2083-4/08/$25.00 ©2008 IEEE

A membership function specifies the effect of the particular variable on the final output. The block diagram of a typical fuzzy system as proposed by Takagi-sugeno-Kang (TSK) is shown in Fig. 1.

The analog inputs are converted to a set of *n* fuzzy values by the fuzzifier block. This process of mapping a single value into an *n* dimensional space is called fuzzification. Fuzzification is done by using membership functions. *n* membership functions are required to map to *n* dimensional space. Each analog input belongs to at least one and preferably two membership functions (overlapping). The accuracy of the system can be increased by increasing the number of membership functions. But, very dense functions can lead to frequent controller action (also known as "hunting") and sometimes may lead to instability.

The fuzzified values are processed by fuzzy logic blocks which implement MIN and MAX operations. MIN and MAX operations are analogous to AND and OR operations in Boolean logic. Boolean logic can be considered as a subset of fuzzy logic since fuzzy logic is applicable for all the values from 0 to 1, unlike Boolean logic which operates on 0 or 1 only. TSK architecture does not require a MAX operator. The MAX operator is replaced by a weighted average which is applied directly to the values of MIN operation. Fig .2a depicts a required control surface, Fig .2b,2c gives a control surface from traditional triangular and Gaussian membership function.

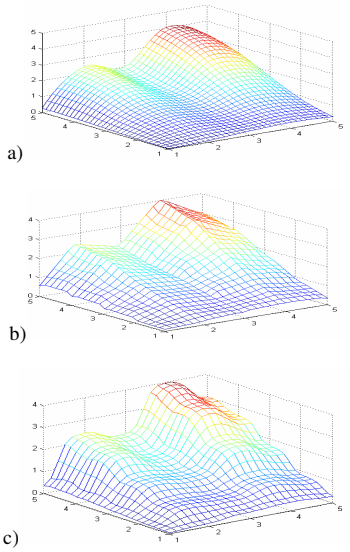


Figure 2. a) Required control sutface b)Traingular membership TSK c)guassian membership TSK

B. *Neural networks*

Neural network is composed of a large number of highly interconnected processing elements (*neurons*) working together to solve specific problems. A single neuron could be modeled by the equation (1). The output of the neuron is given by the sigmoid transfer function. Neural networks cannot be programmed to a specific task but trained to solve by using various algorithms.

$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1} \quad (1)$$

C. *Limitations of FPGA*

The TSK defuzzification architecture is not suitable for FPGA implementation because it requires normalization, which is very computational intensive. Mamdani architecture is also not suitable because its gives a very rough control surface and both the architecture require division in defuzzification which makes it a tough task to implement.

III. PROPOSED IMPLEMENTATION OF NEURO-FUZZY

A. *Neuro-Fuzzy model*

The Proposed Neuro-Fuzzy architecture consists of two blocks. The first block does the Fuzzification and is implemented using a neural network. The second block which does the processing of fuzzified values and defuzzification is done through a LUT based sinusoidal approximation algorithm, which makes a FPGA or microcontroller implementation a simple and easy task

B. *Neural Fuzzification Using Monotonic Functions*

. The control surface obtained using the triangular, trapezoidal and Gaussian membership functions are not smooth and accurate. And moreover, these membership functions are difficult to be modeled and trained in a neural network. So, a neuro-fuzzy model with monotonic membership functions is proposed and used in this work which is easier to implement in a neural network. Three monotonic membership functions are.

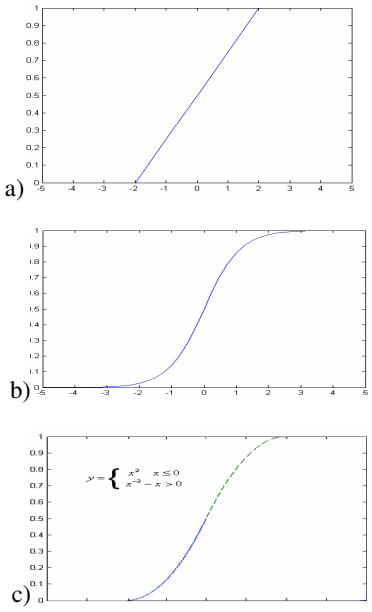


Figure 3. a) Piecewise Linear b) exponential sigmoid c)parabolic function.

used in this model. One of them is a exponential sigmoid function Fig. 3b. and is given by the formula:

$$f(net) = \frac{1}{1+\exp(-net)} \qquad (2)$$

The other two monotonic functions are Piecewise Linear Fig. 3a. and parabolic sigmoid Function Fig. 3c.

The reasons for choosing monotonic functions are:

a) A sigmoid with a exponential function is the characteristic transfer function of bipolar differential transistor pair,

b) A MOS differential transfer function has similar characteristics as first half of Fig.4b.

c) A piecewise linear function could be easily generated in a microcontroller and DSP chips.

Hence a single neuron could be could be easily modeled in hardware using these monotonic functions. Now a membership functions can be constructed by mirror imaging the sigmoid monotonic functions as shown in Fig. 4b. The membership function is got by $F1-F2$, where F1 and F2 are two sigmoid monotonic functions.

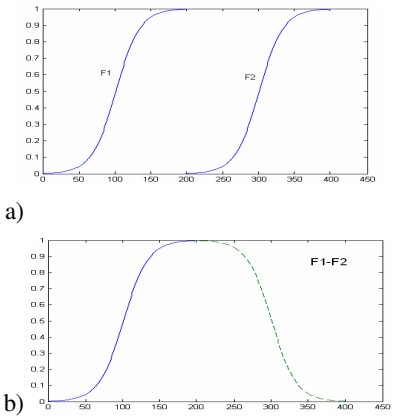


Figure 4. A membership function Implemented using exponential functions

The membership functions are constructed from these basic functions. This membership function can be implemented using two neurons. One of the neurons output gives the first half of the membership function and the negated output of the second neuron implements the second half of the function. Extending this idea a set of neurons is trained to give the required number of sigmoid membership functions. The output of a trained neural network for a sample application is shown in Fig. 5.

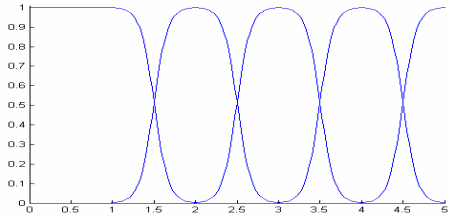


Figure 5. Exponential Sigmoid membership functions

All the neurons use unipolar activation function and if designed properly any fuzzy system can be easily modeled. It can be observed that there was no training process required for this architecture. If training is allowed the network architecture is significantly simplified. LM (Levenberg-Marquardt) [6] is used in this work to train the network. This algorithm is implemented in MATLAB. Cross layer weights have been introduced to make the feed forward network [6] more accurate. Thus the above idea shows, how a fuzzifier block can be implemented through neural network by consciously choosing the membership functions.

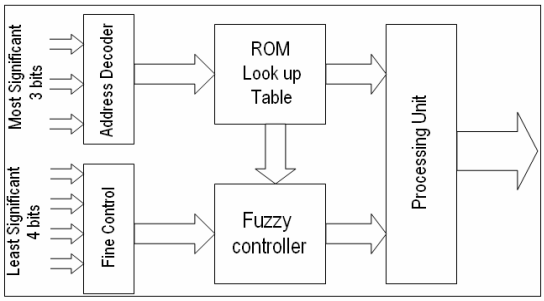


Figure 6. Defuzzification Block.

C. Defuzzification with Trignometric approximation algorithm

Using the established methods for defuzzification results in rougher control surface also hardware implementation of defuzzification by a neural network is very expensive and requires more processing capability. Micro controller like Motorola HC 11 which is 8-bit processor is incapable of doing such processing. These limitations warrant the use of a LUT based solution for high precision and cost effective control system.

A LUT based model which is suitable for FPGA implementation is shown in Fig .6. The inputs to the system are the digital values of the fuzzified values from the first neural fuzzification block. The digital values are obtained using a Analog to digital (A/D) converters in the FPGA. In this approach the model consist of a ROM and address of which is determined by the 3 most significant bits of the fuzzified values. Weighted average is done using the values stored in the LUT. This method results in higher precision and smoother control surface. However the major disadvantage is if the number of input is increased the size of LUT increases exponentially. To handle this problem the size of LUT is kept constant and Least significant bits are used finer calculation, but essentially the size of LUT is kept constant. In this model we use a (4×4) LUT is for the simulation.

The advantage of this method is the efficient usage a very small lookup table (LUT), which minimizes the hardware requirement hence easier implementation and also the error produced by this approach is very minimal resulting in a very smooth control surface.

The values from the LUT are defuzzified by a second order sinusoidal approximation algorithm which is better than the SODA algorithm (Second order defuzzification algorithm) [12][13]. The algorithm is explained by Fig. 7. The single space approximation is a linear approximation; the double space approximation needs the value of 'y' from the LUT, but gives higher resolution output. The single space Sin approximation is the one which the paper deals with. The value of 'y' is found by fixing a Sin curve.

Consider the equation

$$y(x) = ax + b + c\sin(\pi\Delta x / x) \tag{3}$$

Considering 4 points x_1, x_2, x_3 and x_4 separated by distance x. Suppose x is a point between x_2 and x_3, at a distance Δx from x_2.The corresponding Y values are y_1, y_2, y_3 and y_4 which are got respectively from the LUT

At 'x_2' and 'x_3' the value of sin is zero

$$y(x_2) = ax_2 + b \tag{4}$$

$$y(x_3) = ax_3 + b \tag{5}$$

Solving equation (4) and equation (3) gives 'a' and 'b'

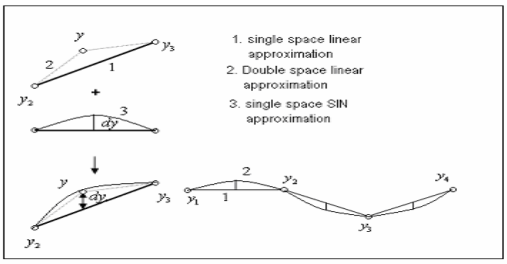


Figure 7. Defuuzification with trignometric approximation

The first derivative is

$$\partial y / \partial x = a + c\cos(\pi\Delta x / x) \tag{6}$$

At 'x_2' and 'x_3' the value of cos is +1 and -1 respectively Therefore,

$$y'(x_2) = a - c \tag{7}$$

$$y'(x_3) = a + c \tag{8}$$

Equation (8) – equation (7) gives:

$$y'(x_3) - y'(x_2) = 2c \tag{9}$$

The slopes are determined by

Slope of y at x_2 $$y_{31} = \frac{y_3 - y_1}{2x} \tag{10}$$

Slope of y at x_3 $$y_{42} = \frac{y_4 - y_2}{2x} \tag{11}$$

Slope of y at x $$y_{32} = \frac{y_3 - y_2}{x} \tag{12}$$

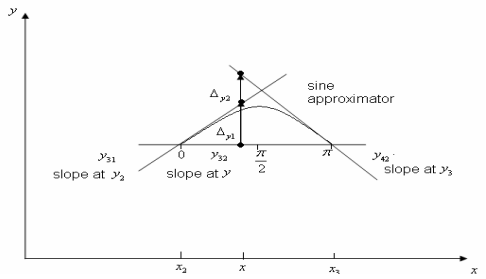


Figure 8. Illustration of approximation algorithm.

Substituting equation (10) and equation (11) in equation (9)

$$c = 0.25\left(y_4 + y_1 - y_2 - y_3\right) \tag{13}$$

The second derivative

$$\Delta_{y1} = \frac{y_{31} - y_{32}}{2} \tag{14}$$

$$\Delta_{y2} = \frac{y_{42} - y_{32}}{2} \tag{15}$$

Substituting equation(13), equation (14), equation (15) in equation(3)

$$y(x) = y_2 + (y_3 - y_2)\Delta x + \sin(\pi\Delta x / x)\Delta_{y1} + \sin(\pi\Delta x / x)\Delta_{y2} \tag{16}$$

The first two terms of equation (16) gives the linear approximated value. These sin values are also retrieved from the LUT.

1) Application of algorithm to 1-dimension problem

To compare the performance of the new algorithm with SODA, consider the equation:

$$y = x\sin(x) \tag{17}$$

Linear approximation is applied and the result is shown in Fig. 9. Results obtained by new algorithm are depicted in Fig. 11.and Fig .12. Error pattern observed by the new algorithm is shown in Fig. 13. The total error is less than the total error of SODA by 25%.

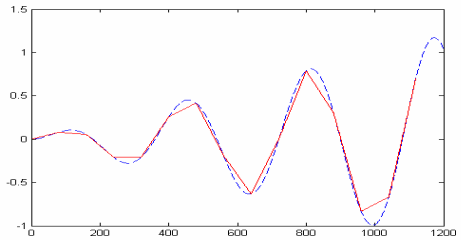


Figure 9. Double space linear approximation

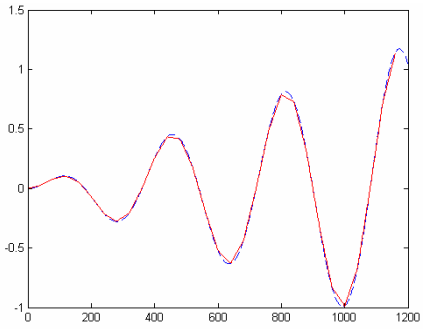


Figure 10. Single space linear approximation

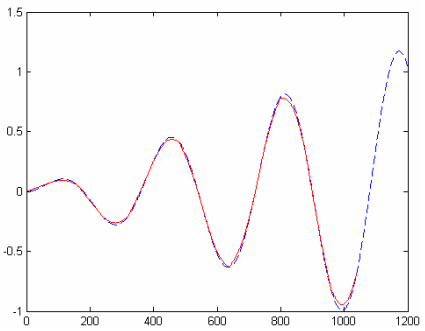


Figure 11.) Double space sin approximation

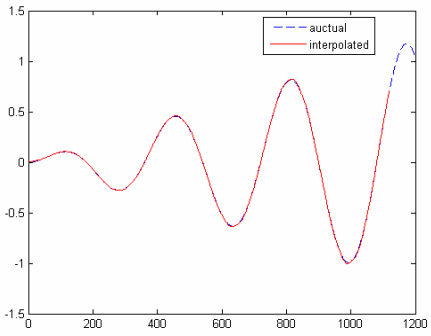


Figure 12. Single space Sin approximation

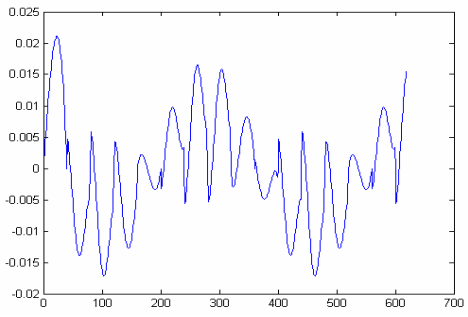
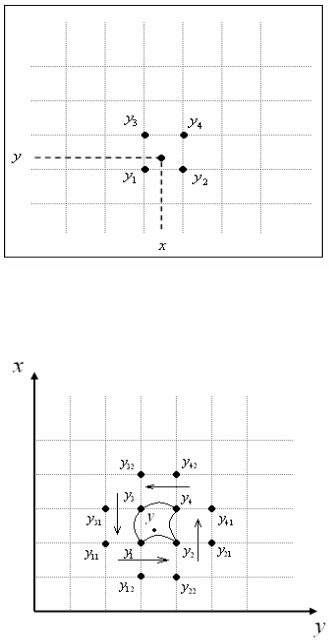


Figure 13. Error using LUT algorithm in 1-D.

The difference between the Single space approximationa and the double space approximation is that the double space algoritm uses a (4×4) LUT while the single space approximation uses a (8×8) LUT.

2) Extension of algorithm to 2-dimension and more

The same approach is extended for the 2-dimensional problem. 4 values form the LUT are required to find the linear approximation shown in Fig .14a.



a)

b)

Figure 14. Application of algoritm to 2-dimensional problem

The algorithm is extended to 2-dimension by solving in both x-axis and y-axis. The Fig .14b depicts the approximation is done along the four direction and the average is found. 12 adjacent point are required to find the value of ' y '. In the same manner the algorithm could also be extended to more than 2-dimension problems.

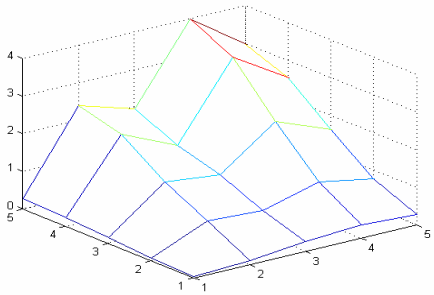


Figure 15. A plot of the Stored data in LUT.

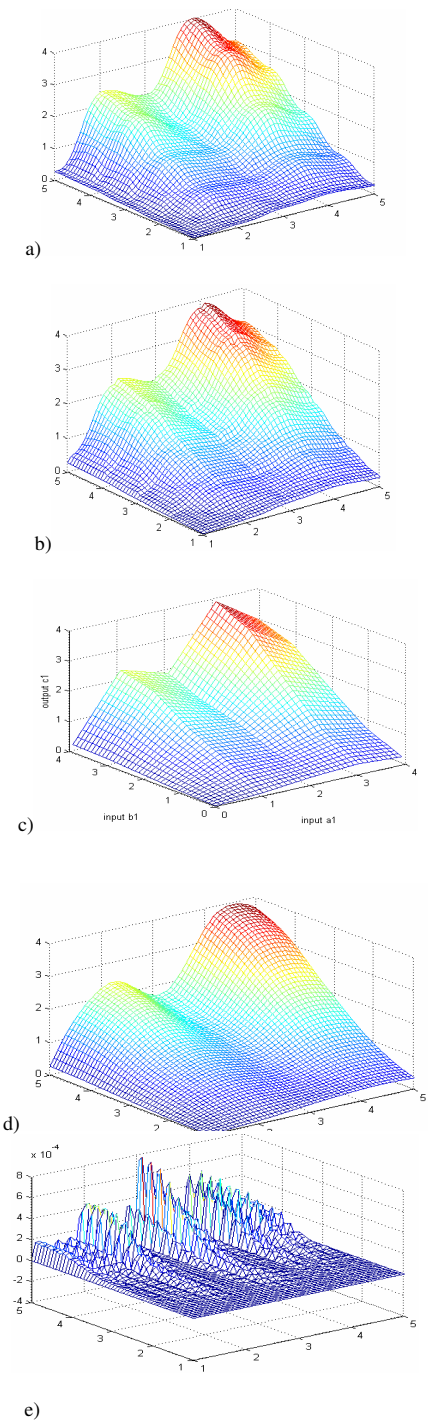


Figure 16. control surfaces obtained with different fuzzification methods: a) parabolic with traditonal TSK b) exponential with traditional TK c)Single space linear approximation d) defuuzification with trignometric approximation in 2-D e)error plot of defuuzzification with trignometric approximation.

IV. Simulation Results

In this section the performance of the proposed membership functions, are compared with traditional triangular and Gaussian membership functions. Fig. 15 shows the given data points from which the control surface has to be constructed. Fig. 16 shows the control surfaces constructed using various methods.. The TSK with and the parabolic and exponential monotonic functions Fig 16a 16b have smoother control surface when compared to triangular. Fig. 16d shows the control surface constructed using trigonometric approximation algorithm. This algorithm produces a surface which is very close to the expected one and demonstrated to be better than all the other membership functions.

V. Conclusion

In this paper a simple yet accurate implementation of a neuro-fuzzy system was proposed. Sigmoid membership functions were used in fuzzy logic so that they could be implemented using neural networks. A LUT (Lookup Table) algorithm was introduced to implement in the FPGA. This LUT algorithm in itself can be used to model a non-linear control surface with sufficient accuracy. It was demonstrated using MATLAB simulations that the proposed neuro-fuzzy model produces accurate and smooth control surfaces.

References

[1] Shouling He; Relf, K.; Unbehauen, R, neural approach for control of nonlinear systems with feedback linearization Neural Networks, IEEE Transactions on On page(s): 1409-1421, Volume: 9, Issue: 6, Nov 1998.

[2] S. Deutsch and A. Deutsch. Understunding the NerwusSjmem: An Engineering Perspective, IEEE Press, Piscataway, NJ, 1993.

[3] M. L. Padgett, P. J. Werbos, and T. Kohonen, Strategies ond Tactics for the Application of Neural h'ehwrk.Y in Irxfirstrial Electronics, CRC Handbook for Industrial Electronics, J. D. Irwin Ed., CRC Press and IEEE Press, pp. 835-857, 1997.

[4] Zurada, J., Intrudiiction to Artifkid Naira1 Systems, West Publishing Company, 1992.

[5] Wilamowski, B. M., R. C. Jaeger, and M. 0. Kaynak, "NewFuuy Architecture for CMOS Implementation" IEEE Trun,sucrion on 1,xfuWial Electrunia, vol. 46, No. 6, pp. 1132-1136, Dec. 1999.

[6] Bogdan M. Wilamowski,Yixin Chen,"Efficient Algorithm for Training Neural Networks with one Hidden Layer", in IJCNN. Proc. 1999 International Joint Conference Vol. 3, page(s): 1725-1728.

[7] Tan H., R. sandige and B. Wilamowski, "Hardware implementation of PLD based fuzzy logic controllers using Look-up table technique", proc ANNIE '94, Nov -1994, vol. 4, pp.89-94.

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

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

[11] 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.

[12] Dharia, N. Gownipalli, J. Kaynak, O. Wilamowski, B.M., Fuzzy controller with second order defuzzification algorithm., in IJCNN '02. Proc. 2002 International Joint Conference Vol. 3, page(s): 2327-2332.

[13] Dharia,N.Gownipalli,J.Wilamowski, B.M. Kaynak, O.Multi dimensional second order defuzzification algorithm (M-SODA), IECON '02, Industrial Electronics Society, Nov. 2002,Vol. 4, page(s): 3215- 3220.