Mansoura Engineering Journal, (MEJ), Vol. 30, No. 3, September 2005. F 22

Adaptive Fuzzy Logic Controller for DC-DC Converters

عمل متحكم مهايئ باستخدام طريقة المنطق المبهم لمتغيرات التيار المستمر

Sahar Kaddah[1] and Ahmed Rubaai[2]

1: Lecturer, Electrical Power& Machines Dept., Mansoura University, Egypt
2: Professor, Electrical Engineering Department, Howard University, Washington, DC USA

الخلاصة

في الأونة الأخيرة وفي ظل التطور التكنولوجي في نظم التحكم تم عمل متحكم مهايئ باستخدام طريقة المنطق المبهم لمتغيرات التيار المستمر . يقدم هذا البحث دراسة خصائص نوعين من هذا المتحكم الذي يستخدم مميزات الخلايا العصبية بحيث يسمح بتعديل أداء النظام نتيجة حدوث أي تغير في الحمل أو في الجهد الداخل . وقد تم بناء نموذج رياضي لهذا المتحكم باستخدام برنامج MATLAB . وكذلك تم دراسة تأثير تغير الحمل والجهد الداخل علي أداء النظام .

Abstract - Two adaptive fuzzy logic controller (AFLC) topologies for the DC-DC converter are developed and presented in this paper. They essentially consist of combining fuzzy inference system and neural networks and implementing them within the framework of adaptive networks. The architecture of the AFLC along with the learning rule, which is used to give an adaptive and learning structure to a fuzzy controller, is also described. The emphasis here is on fuzzy-neural-network control philosophies in designing a novel controller for the DC-DC converter that allows the benefits of neural network structure to be realized without sacrificing the intuitive nature of fuzzy system. The AFLC topologies are built on Matlab environment and tested for both the buck and buck boost converter for load regulation and line regulation. The proposed AFLCs have satisfactory results for tracking the reference output voltage.

1. Introduction

For DC/DC Converters with constant output voltage, it is always desirable that the output voltage remains unchanged in both steady state and transient operations whenever the supply voltage and/or load current are disturbed. This condition is known as zero-voltage regulation and it means that the output voltage is independent of the supply voltage and the load current. The DC-DC converters are generally divided into two groups: hard-switching converters and soft-switching converters [1, 2]. In hard-switching converters, the power switches cut off the load current within the turn-on and turnoff *times* under the hard switching conditions. The output voltage is controlled by adjusting the on time of the power switch, which in turn adjusts the width of a voltage pulse at the output. This is known as PWM control. In soft-switching converters, resonant components are used to create oscillatory voltage or current waveforms so that the zero-voltage switching or zero-current switching conditions could be created for the power switches. For many years, control design for converters is carried out through analog circuits, which limited them to mostly PI controller structure. The PI controllers generally give overshoot in output voltage and high initial current when rise time of response is reduced. Feed forward types of controllers have also been designed by sensing the input voltage to improve line regulation in applications with a wide range of input voltages and load currents [3]. However, direct sensing of the input voltage through a feed-forward loop may induce large-signal disturbances that could upset the normal duty-cycle of the converter. Using human linguistic terms and common sense,

several fuzzy logic controllers have been developed and implemented for the DC-DC converters [4-8]. These controllers have shown promise in dealing with nonlinear systems and achieving voltage-regulation in buck converters. Fuzzy logic control uses human like linguistic terms in the form of IF-THEN rules to capture the nonlinear system dynamics. Once in place, the fuzzy rules will not be able to adapt themselves to adequately capture the dynamics and external disturbances of the converter. Although achieving many practical successes, fuzzy control has not been viewed as a rigorous science due to a lack of formal analysis and synthesis techniques. As a result of this, a lot of work has been done to develop adaptive fuzzy controllers as well as automate the modeling process as much as possible. Recently, the

Accepted July 31, 2005.

resurgence of interest in artificial neural networks has injected a new driving force into fuzzy literature.
One of the major features of neural networks is their learning capability. They learn from data and feedback. Once trained, neural networks will be able to deal with the nonlinear parameter of the problem at hand. A learning/training mechanism is usually used to update the nonlinear parameters of the network architecture. On the other hand, fuzzy logic or fuzzy interface systems can also deal with nonlinear systems but they do not employ any learning mechanism. Fuzzy inference systems use human like linguistic terms in the form of IF-THEN rules to capture the nonlinear dynamics of the problem at hand. To become adaptive, fuzzy interface system must be able to learn how to adjust their parameters in order to capture the dynamics of the system. To become adaptive, fuzzy interface systems can be equipped with a learning algorithm adopted from neural networks. The marriage of neural network learning techniques and fuzzy interface systems has resulted in a very powerful strategy known as adaptive-neural-network fuzzy systems. In this paper two different topologies for the adaptive fuzzy logic controller topologies are developed and implemented.

2. Development of the Adaptive Fuzzy Logic Controller

The purpose of converting the fuzzy controller to an adaptive Fuzzy logic controller (AFLC) is to tune the controller with learning mechanism (numerical data). Training the AFLC with the back propagation algorithm allows the internal representation of the input and output membership functions of the fuzzy rules to be updated to accommodate the desired numerical data. There are two different topologies developed in this research. The converter is represented by a "black box" from which we only extract the terminals corresponding to input voltage (Vi), output voltage (Vo), one inductor current (iL), and controlled switch (S). From the measurements, the controller provides a signal proportional to the converter duty cycle, which is then applied, to a standard pulse width modulation (PWM) modulator. Both fuzzy logic principles and learning functions of neural networks are employed together to construct the adaptive fuzzy-network inference system for both topologies. Initially, a basic fuzzy logic controller is set up utilizing linguistic rules and then numerical data is used for training the controller. The AFLC used in the two topologies has similar architecture in all cases so as to simplify and generalize the discussion. The number of membership functions is chosen as five so as to cover the entire input space. The triangular membership function is chosen owing to its simplicity and its symmetrical properties. The initial values of the premise parameters (the corner coordinates of the triangle) are chosen so that the membership functions are equally spaced along the operating range of each input variable.

2.1. Adaptive Fuzzy Logic Controller, Topology I

6-layer neural network architecture is proposed. Figure 1 shows the module of the neural network architecture. The inputs to this topology are the error in the output voltage and the error in the inductor current. The two input nodes in layer 1 only transmit input signals to the next layer. Each node corresponds to one input variable. For every node i in this layer, the input and the output of the network are represented, respectively.

$$net_i^1 = X_i^1,$$
$$Y_i^1 = f_i^1(net_i^1) = net_i^1$$

Where, X_i^1 represents the ith input to the node 1.
The nodes in layer 2 are term nodes that act as ق membership functions to express the input/output fuzzy linguistic variables. In this project, the triangular activation function will be used to represent the membership function. Therefore, for the j[th] node.

$$net_j^2 = -\frac{(X_i^2 - \mu_{ij})^2}{2}$$

$$Y_j^1 = f_j^1(net_j^1) = net_j^2 = \begin{cases} \dfrac{X_i - a_i}{b_i - a_i} & a_i \le x_i \le b_i \\ \dfrac{c_i - X_i}{c_i - b_i} & b_i \le x_i \le c_i \\ 0 & Otherwise \end{cases}$$

The weights between the input and the membership layer are assured to be unity.

The fuzzy sets defined for the input/output variables are positive big (PB), positive small (PS), zero (ZE), negative big (NB), and negative small (NS). Therefore, 10 and 25 nodes are included in layers 2 and 3, respectively to indicate the input/output linguistic variables. Each node in layer 3 is denoted by Π which multiplies the incoming signal and outputs the result of the product. Consequently, each node of this layer is a rule node that represents one fuzzy control rule. In total, there are 25 nodes in layer 3 to form a fuzzy rule base for two linguistic input variables. The links of layer 3 define the preconditions and the outcome of the rule nodes,

respectively. For each rule node, there are two fixed links from the input term nodes.
For the k^{th} rule node.

$$net_k^3 = \prod_j w_{jk}^3 X_j^3,$$
$$Y_k^3 = f_k^3(net_k^3) = net_k^3$$

Where X_j represents the j^{th} input to the node of layer 3, and w_{jk} is the link that connects the output of the j^{th} node in layer 2 with the input to the k^{th} node in layer 3. The weights between the input and the membership layer are also assumed to be unity. The links of layer 4 will be adjusted in response to varying control circumstances. The link weights, w_{kl} represent the output action of the k^{th} rule. Each node in layer 4 consists of nonlinear mapping, which are sigmoidal functions. The sigmoidal activation function imposes bounds. On the signal to enhance stability. For the l^{th} node in this layer, the input and output of the network are represented as:

$$net_l^4 = w_{kl} Y_k^3,$$
$$Y_l^4 = f_l^4(net_l^4) = \frac{2}{1+\exp(-\gamma . net_l^4)} - 1$$

The output of layer 5 is the main output and acts as a defuzzifier. The nodes δ_p and $\Delta\delta_l$ in this layer are labeled Σ and they sum all incoming signal to each branch to obtain the final inferred results for δ_p an $\Delta\delta_l$.

$$Y_m^5 = f_m^5(net_m^5) = net_m^5$$

The fifth layer is the one we training or updating its weights, w_{lm} which represents the weight connecting layer k and layer m, to satisfy the desired value.
The output of Layer 6 is the summation of δ_p and the integration of $\Delta\delta_l$ generates the change in the converter duty cycle.

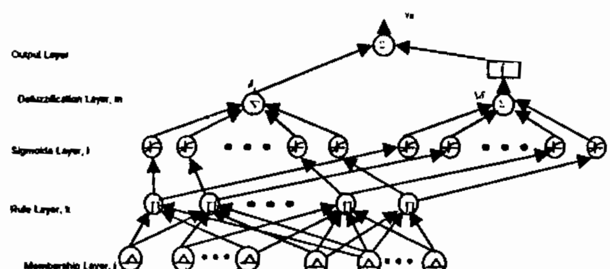

Figure 1. Architecture of the Adaptive Fuzzy Logic Controller, topology I.

2.2 Adaptive Fuzzy Network, Topology II

This topology is a 5-layer topology as shown in Figure 2. The inputs to this topology are the error in the output voltage and the rate of change in the output voltage. The first four layers are similar to the ones in the first topology. The fourth layer is the one in which the weights are to be adjusted using the ANN back-propagation based on the desired value of the output. The output of layer 5 is the output layer and acts as a defuzzifier. In this layer, all incoming signals are summed to obtain the final inferred results for the change in the duty cycle..

$$net_m^5 = \sum w_{lm}^4 Y_l^4,$$
$$Y_m^5 = f_m^5(net_m^5) = net_m^5$$

The fifth layer is the one we training or updating its weights, w_{lm} which represents the weight connecting layer k and layer m, to satisfy the desired value.

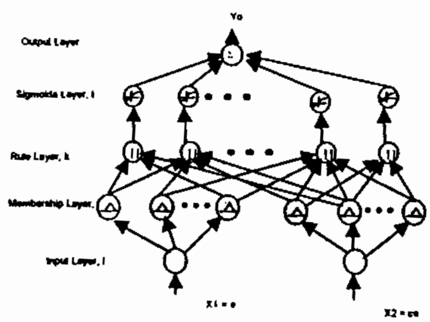

Figure 2. Architecture of the Adaptive Fuzzy Neural Network Controller, topology II

3. Learning Algorithm

The only layer that the weights are trained is the fifth layer for topology I and the fourth layer for topology II. The back-propagation network is used to train the weights of this layer. The learning algorithm used can be described in the following steps:

Step 1: Calculate the error between the desired value and the actual output, *Error(p)* = $\delta_{desired} - \delta_{output}(p)$

Step 2: Calculate the error gradient, *delto(p)*

$$delto(p) = \delta_{output}(p) * (1 - \delta_{output}(p)) * error(p)$$

Step 3: Calculate the weight correction

$$\Delta w_{lm}(m,p) = \eta * delto(p) * out(m,p)$$

Step 4: Update the weights

$$w_{lm}(m,p+1) = w_{lm}(m,p) + \Delta w_{lm}(m,p)$$

Where

η: Learning rate and p: Iteration number

Step 5: Calculate the new output voltage value and go back to step 1

4. Simulation Results

Several test cases were conducted to assess the performance of the proposed adaptive fuzzy inference control system for the two control topologies. However, for briefness, only few cases are reported for illustration purposes. After designing the best stand-alone fuzzy controllers, the effectiveness of combining both the fuzzy logic and adaptive fuzzy-neural controllers is examined. Selected of test results performed on several types of DC-DC converter are illustrated in Figs. 3-12.

4.1. Case A : Buck Converter under Load Regulat ion, Fuzzy Logic Topology II

In this case, the fuzzy controller of topology-II is considered for a buck-converter under load regulation. The load resistance is varied from 5 Ω to 3 Ω and back to 5Ω. Figure 3 shows the performance of the output voltage tracking for the fuzzy controller compared with the uncontrolled case, while Fig. 4 displays the corresponding duty cycle. Good tracking performance is achieved at all times, but still not as perfect as required as there are ripples in the output voltage and the duty cycle.

It is also shown that when the load suddenly changes from 5 Ω to 3 Ω at time 0.015 seconds due to the decrease in the load resistance, the original duty cycle decreases resulting in decreasing the converter output voltage before it stabilizes again. On the other hand, when the load resistance increases suddenly from 3 Ω to 5 Ω the duty cycle increases resulting in increasing the output voltage that oscillates a little before it stabilizes again.

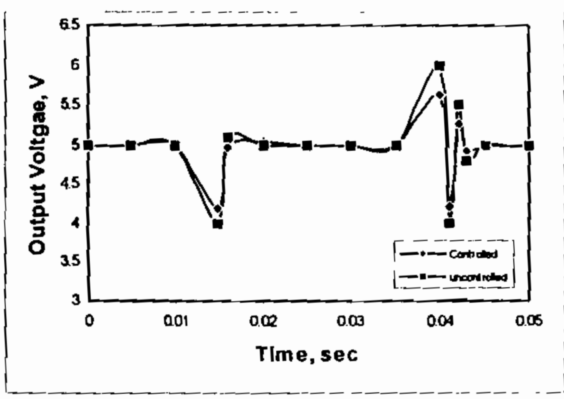

Fig. 3. Output Voltage for Buck Converter

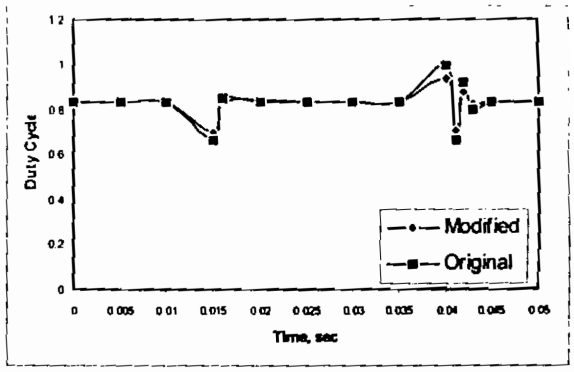

Fig. 4. Duty Cycle for Buck Converter

It is also shown that when the output voltage is lower than its reference value, the fuzzy rules always try to add positive change of the duty cycle to bring the output voltage as close as possible to its references value, t = 0.015, 0.041 and 0.043 second. On the other hand, when the output voltage is higher than its reference value, the fuzzy rules add negative change to the duty cycle to bring the output voltage back to its reference value, t= 0.016, 0.04 and 0.043 second. However, the output voltage still does not match with its reference value.

4.2. Case B : Buck-Boost Converter under Load Regulation, Fuzzy Logic Topology I

In this case, the fuzzy controller of topology-I is examined for a buck-boost converter where the load resistance is varied from 20 Ω to 150 Ω and back to 20Ω as shown in Figure 5. Figure 6 shows the corresponding inductor current due to the load variation. Figure 7 shows the corresponding duty cycle of the buck-boost converter under this condition, it also shows that when the load resistance is varied suddenly from 20 Ω to 150 Ω and back to 20Ω, the duty cycle is responding to these changes with high accuracy. Figure 8 shows that the fuzzy controller almost brings the output voltage to its reference value most of the time.

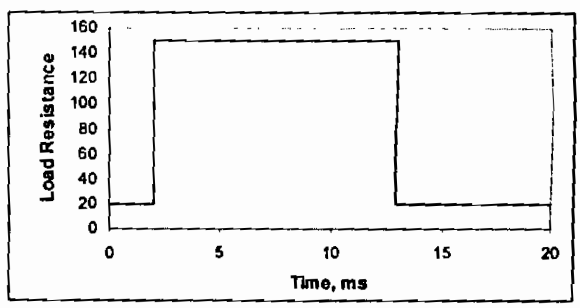

Fig. 5. Load Changes for the Buck-Boost Converter

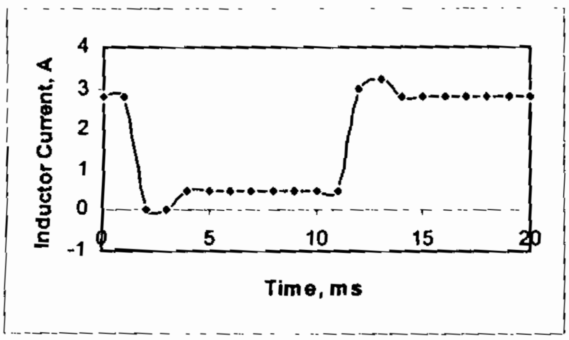

Fig. 6 Inductor Current for the Buck-Boost Converter

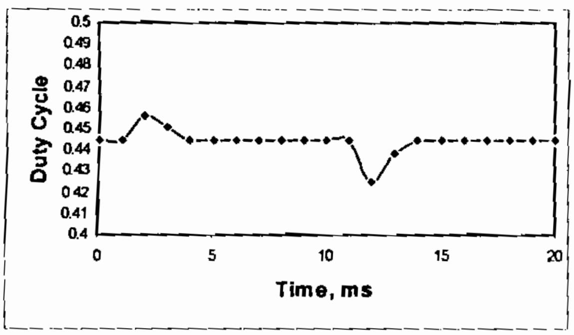

Fig. 7 Duty Cycle for the Buck-Boost Converter

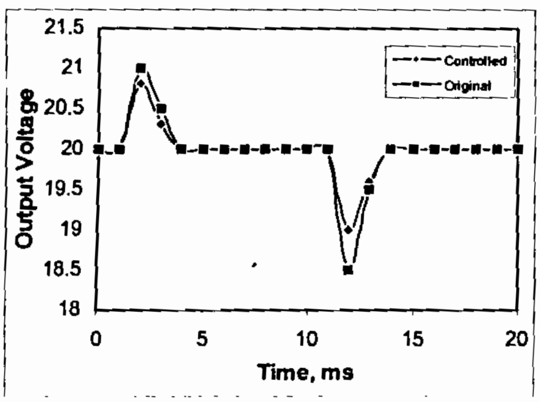

Fig. 8. Output Voltage of the Buck-Boost converter

4.3. Case C : Buck-Boost Converter under Load Regulation, Adaptive Fuzzy Logic Topology I

In the third test, the adaptive fuzzy-neural-network controller (fuzzy logic controller and its integration with the neural-network architecture) of topology-I is examined for the same system in test case B, buck-boost converter where the load resistance is varied from 20 Ω to 150 Ω and back to 20Ω. Table 1 shows the output voltage from the fuzzy and the adaptive controllers, the output of different activated fuzzy rules and the weight of different neurons at different time steps.

The rules from 1 to 25 and the weights from 1 to 25 are related to the fuzzy-p controller. While the rules from 26-50 and the weights from 26-50 are related to the fuzzy-I controller.

As it is shown from Table 1, when the fuzzy output is greater than its reference value, the neural network has to increase different neuron weights from their original values "1.0" for the neurons that has negative output and decrease the weights for the neurons that has positive output. So, the summation will be negative, decreasing the duty cycle and so output voltage will decrease to be closer to the reference voltage. Such as the cases at t = 2 and 3 ms. However, when the fuzzy output is less than the reference voltage, the neural networks has to decrease the weights for the neurons that has the negative output and increase the weights for the neurons that has positive output. So, the summation will be positive that means the neural network will increase the duty cycle and increasing the output voltage as well. Such as the case at t = 12 and 13 ms. Figure 9 shows a sample of how the adaptive fuzzy controller approaches its reference value at t= 2 ms.

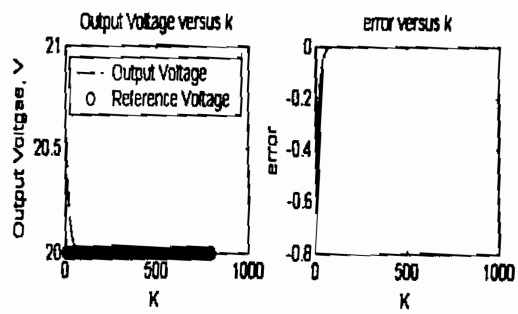

Figure 9. Applying the Adaptive Fuzzy Controller for Case A at time, 2 ms.

It is shown from Figure 9 that the initial output voltage value of "20.8 V", which is the output of the fuzzy logic controller. By starting training the ANN by changing the weights of different neurons of the last layer, the output voltage is getting closer and closer to its reference value(20V) until it approaches the reference at almost 100 training iterations. At the same time, the error starts from a value of "-0.8" that is the difference between the reference and the output voltage and reduced gradually until it approaches the zero at almost 100 training iterations.

Fig. 10 shows the performance of fuzzy and adaptive fuzzy controllers. Also, It can be seen from Fig. 10 that the adaptive fuzzy controller reduces the ripple in the output voltage. In addition, it almost brings the output voltage to its reference value most of the time. The maximum tolerance that the adaptive fuzzy generates occurs at "0.016

second" with a value of "1.304 %" while at the same point the fuzzy logic has a tolerance of "6.6006 %".

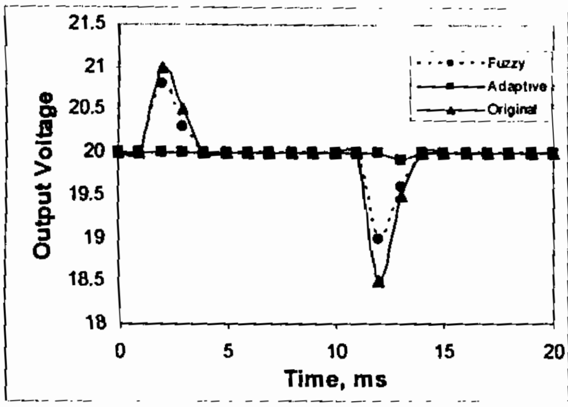

Fig. 10. Output Voltage for buck-boost converter, Original, Fuzzy and Adaptive Fuzzy

4.4. Case D: Line Regulation of Buck Converter; Input has step changes from 15 V to 20 V and back to 15 V

Figure 11 shows the input voltage for the buck converter subject to the step change of the line voltage. While, Figure 12 shows the corresponding change on the output voltage.

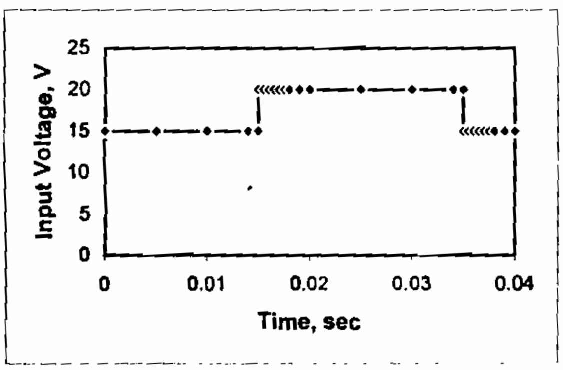

Figure 11. Input Voltage of the Buck Converter for Case D

Figures 13 and 14 show a comparison between using the fuzzy logic and the adaptive fuzzy controllers for this case in terms of the duty cycle and the output voltage. It is shown from Figures 13 and 14 that the adaptive fuzzy controller reduces the ripple in the output voltage as it almost brings the output voltage to its reference value most of the time.
The maximum tolerance that the adaptive fuzzy generates occurs at "0.016 second" with a value of "0.254 %" while at the same point the fuzzy logic has a tolerance of "4%". It is also shown that when the input voltage is increased suddenly from 15 V to 20 V, the duty cycle has to compensate by a reduction from 0.3333 to 0.25 to keep the output voltage constant.

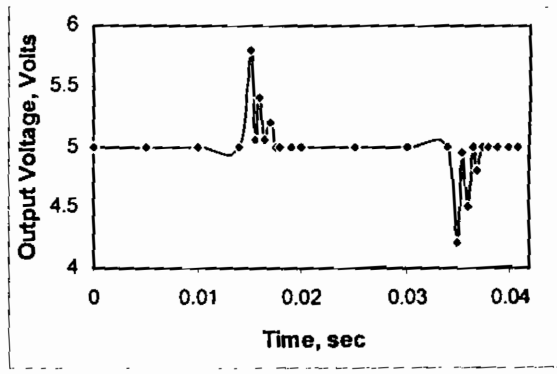

Figure 12. Output Voltage of the Buck Converter for Case D

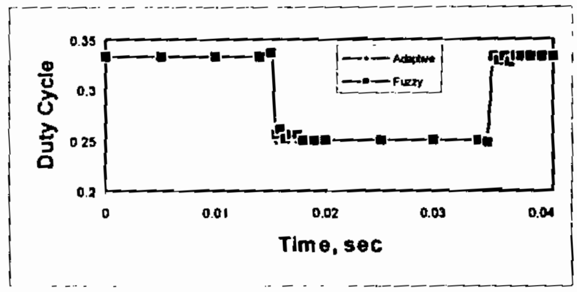

Figure 13. Duty Cycle of the Buck Converter for Case D Using Fuzzy and Adaptive Controller

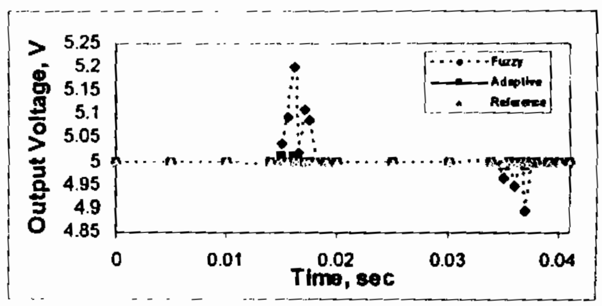

Figure 14. Output Voltage of the Buck Converter for Case D Using Fuzzy and Adaptive Controllers

5. Conclusions

Two adaptive fuzzy logic controllers (AFLC) of the DC-DC switch mode power converter have been presented. The proposed two topologies have the advantage that they cannot only take linguistic information from human experts but also adapt themselves using numerical data to achieve better performance. The main feature of the proposed AFLC is an on-line learning control architecture, which deals with a noisy operating environment, and captures the unknown nonlinear dynamics of the DC-DC converter. The two controller topologies considered have been shown to yield satisfactory performance.

6. References

[1] J. Foutz, "Switching-Mode Power Supply Design," Internet Site: http://www.smpstech.com/tutorial/tO2top.htm, August 1999.

[2] Y. Liu and P. C. Sen, "A Novel Method to Achieve Zero- Voltage Regulation in Buck Converter," IEEE Trans. Power Electronics, Vol. 10, No. 3, pp. 171-180, May 1995.

[3] B. Arbetter and D. Maksimovic, "Feed-forward Pulse Width Modulators for Switching Power Converters," IEEE Trans. Power Electronics, Vol. 12, No. 2, March 1997

[4] P. Maftavelli, L. Rossetto, et al., "General-purpose Fuzzy Controller for DC-DC Converters," IEEE Trans. Power Electronics, Vol. 12, NO 1, January 1997.

[5] T Gupta and R. R. Boudreaux, "Implementation of a Fuzzy Controller for DC-DC Converters Using an Inexpensive 8-b Micro-controller," IEEE Trans. Industrial Electronics, Vol. 44, No. 5, October 1997.

[6] W. So, C. K. Tse, and Y. Lee, "Development of a Fuzzy Logic Controller for DC/DC Converters: Design, Computer Simulation, and Experimental Evaluation," IEEE Trans. Power Electronics, Vol. 11, No. 1, January 1996.

[7] J. Arias, A. Arias, S. Gomariz, and F. Gumjoan, "Generating Design Rules For Buck Converter-Based Fuzzy Controllers," Proceedings of the IEEE International Symposium on Circuits and Systems, Vol. 1, May 12-15, pp. 585-588, 1996.

[8] B. R. Lin, "Fuzzy PWM DC-DC Converter Control," Intelligent Engineering Systems through Artificial Neural Networks, ASME, Vol. 5, No. 12-15, pp. 587-592, 1995.

[9] J. Principe, N. R. Euliano, W. C. Lefebvre, Neural and adaptive Systems, New York: John Wiley & Sons, 2000.

[10] J. S. R. Jang, C. Sun, and E. Mizutani, Neuro-Fuzzy and Soft Computing, New Jersey: Prentice Hall, 1997.

[11] D. S. Yeung, E. C. C. Antony, and Y. T. Cheng, "Fuzzy Production Rule Refinement Using Multilayer Perceptrons," Proceedings of the 3rd IEEE International Conference on Fuzzy Systems, Orlando, Florida, Vol. 1 pp. 211-217, June 26-29, 1994.

[12] W. Li, "Optimization of Fuzzy Controller Using Neural Networks," Proceedings of the 3rd IEEE International Conference on Fuzzy Systems, Vol. 1, pp. 223-227, June 26- 29, Orlando, Florida, , 1994.

Table 1: The output Voltage, Output of Different Fuzzy Rules and Weights of Different Neurons for Case C of Topology I

Time, ms	Output Voltage, Volts		Different fuzzy rules O/P (pu)	Different neuron Weights
	Fuzzy	Adaptive		
0 -1	20	20	NA	
2	20.8	20.	R_1 = -0.1342 R_2 = -0.0337 R_{26} = 0.0265 R_{27} = 0.0951	w_1 = 1.4947 w_2 = 1.1066 w_{26} = 0.9231 w_{27} = 0.7342
3	20.3	20	R_1 = -0.1286 R_2 = -0.0314 R_{26} = 0.0254 R_{27} = 0.0951	w_1 = 1.1644 w_2 = 1.0379 w_{26} = 0.9703 w_{27} = 0.8932
4-11	20	20	NA	
12	19	20	R_6 = -0.1027 R_7 = 0.0199 R_{31} = 0.0203 R_{32} = 0.0199	w_6 = 0.0009 w_7 = 3.792 w_{31} = 3.9026 w_{32} = 3.792
13	19.6	19.9092	R_{11} = -0.0367 R_{12} = -0.0108 R_{36} = 0.0072 R_{37} = -0.0108	W_{11} = 0.0043 W_{12} = 0.0375 w_{36} = 9.0046 w_{37} = 0.0375
14-20	20	20	NA	