Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 36 (2009) 1540–1548

Expert Systems with Applications

www.elsevier.com/locate/eswa

Adaptive fuzzy logic controller for DC–DC converters

Cetin Elmas [a], Omer Deperlioglu [b,*], Hasan Huseyin Sayan [a]

[a] *Department of Electric, Faculty of Technical Education, University of Gazi, 06500 Ankara, Turkey*
[b] *Department of Electronics and Computer, Faculty of Technical Education, University of Afyon Kocatepe, 03300 Afyon, Turkey*

Abstract

This paper introduces a complete design method to construct an adaptive fuzzy logic controller (AFLC) for DC–DC converter. In a conventional fuzzy logic controller (FLC), knowledge on the system supplied by an expert is required for developing membership functions (parameters) and control rules. The proposed AFLC, on the other hand, do not required expert for making parameters and control rules. Instead, parameters and rules are generated using a model data file, which contains summary of input–output pairs. The FLC use Mamdani type fuzzy logic controllers for the defuzzification strategy and inference operators. The proposed controller is designed and verified by digital computer simulation and then implemented for buck, boost and buck–boost converters by using an 8-bit microcontroller.

Keywords: DC–DC converters; Fuzzy logic controller; Adaptive fuzzy logic controller

1. Introduction

In recent years, there has been increasing interest in the development of efficient control strategies to improve dynamic behavior of DC–DC converters by using fuzzy logic controller (FLC), neural networks (NN), and neuro-fuzzy controller (NFC). So, Tse, and Lee (1994) have presented an FLC for DC–DC converters, and simulation results for buck, boost and buck–boost converter were presented Carrosco, Torralba, Ridao, and Franquelo (1994) have proposed an FLC that uses an optimal algorithm, and they have given experimental results. Lin (1997) has described an NN and FLC for power converters. Leyva, Martinez-Salamero, Jammes, Marpinard, and Guinjoan (1997) have used an NN to control DC–DC converter and demonstrated simulation results. So and Tse (1996) have designed an FLC and implementation has been made by using a digital signal processor (DSP) TMS320C50. Gupta, Boudreaux, Nelms, and Hung (1997) have presented a paper on an implementation of FLC for DC–DC converters using an 8-bit microcontroller. Mattavelli, Rossetta, Spiazzi, and Tenti (1997) have proposed and simulated the FLC for buck–boost and Sepic converters. In recent years, neuro-fuzzy controller or adaptive fuzzy logic controller (AFLC) have been used to control the DC–DC converters (Huh & Park, 1999; Wong, Leung, & Tam, 1995) and the performance of the controller is verified by simulation results only. Higuchi et al. (2004), Alejo, Maussion, and Faucher (2003) and Lin and Tsai (2005) have designed a different control model and implementation has been made to regulate DC–DC converter by using a digital signal processor (DSP). Rubai, Ofoli, Burge, and Garuba (2005) and Calderón, Vinagre, and Feliu (2006) have used different control technologies to control DC–DC converter and demonstrated results with using a microcontroller and extra specialized hardware.

The FLC could be useful in situations where (1) there is no acceptable mathematical model for the plant and (2) there are experienced human operators who can satisfactorily control the plant and provide qualitative control rules in terms of vague and fuzzy sentences. There are many practical situations where both (1) and (2) are true (Park, Moon, & Lee, 1995; Wang, 1993). Design of fuzzy logic

[*] Corresponding author.
E-mail addresses: celmas@gazi.edu.tr (C. Elmas), odeper@aku.edu.tr (O. Deperlioglu), hsayan@gazi.edu.tr (H.H. Sayan).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.11.029

controllers has some difficulties in the selection of optimized membership functions and fuzzy rule base, which is traditionally achieved by a tedious trial-and-error process.

This paper introduces a systematic approach to construct an AFLC for DC–DC converters. The AFLC optimizes membership functions and rule base of the FLC were obtained from training data in the pattern file. The AFLC approach is general in a sense that almost the same control rules can be applied to other applications. In this AFLC, fuzzy logic controller algorithm is a synthesis of works by Chien and Hsieh (1996) and Galichet and Foulloy (1995). The shrinking-span membership functions (SSMFs) algorithm presented by Chien and Hsieh (1996) is used to construct membership functions for FLC. By using this method, the designer of an FLC assigns only the number of elements of term set and shrinking factor for specific linguistic variable and then obtains a series of well-located membership functions. Such SSMFs make them more reasonable from the point of human experts and also provide the FLC the ability to easily adapt to different control systems by slight modification (Chien & Hsieh, 1996). The algorithm in the reference (Galichet & Foulloy, 1995) is used to deduce rule base and distribution of membership function. In the FLC, the defuzzification strategy and inference operators are Mamdani type and using center of gravity defuzzification method (Mamdani, 1977).

2. Basic design of adaptive fuzzy logic controller

The block diagram of the FLC for DC–DC converters is shown in Fig. 1. The fuzzy logic controller can be divided into four modules: fuzzification is the classification of input data into suitable linguistic values or sets; knowledge base includes rule base and data base. They contain knowledge of the control rules and linguistic labels; decision making is inferring control action from rule base; defuzzification is the conversion from the inferred fuzzy value to real crisp value, or control action.

The inputs of the FLC are the error e and difference of error de respectively and they are defined as

$$e = V_{\text{ref}} - V_{\text{o}} \tag{1}$$

$$de(k) = e(k) - e(k-1) \tag{2}$$

where V_{i} is input voltage, V_{o} is actual output voltage of DC–DC converter at the kth sampling time, V_{ref} is reference output voltage. The output of the FLC is change in duty ratio ($du(k)$). Duty ratio $d(k)$, at the kth sampling time, is defined as

$$d(k) = d(k-1) + du(k) \tag{3}$$

Then it is send through the PWM out to DC–DC converter to generate desired switching action.

All input linguistic variables are assumed to have the same number of linguistic values, that is $m_1 = m_2 = \cdots = m_I = m$. With the simple rule mapping $m_y = \sum_{i=1}^{I} m_i = I_m$ holds and there for the number of linguistic values for output is $M_y = 2m_y + 1 = 2I_m + 1$ where I_m is the number of input and $I_{my} = \{-m_y, \Lambda, -1, 0, 1, \Lambda, m\}$ is index set with M_y terms for linguistic variable x. Conceptually, the SSMFs, which is constructing membership functions method of the FLC, are a series of orderly arranged membership functions for a linguistic variable across its universe of discourse. By using a series of $A(x_i)$s, propose the following representation for the trapezoidal family. For instance, typical trapezoidal family SSMFs is shown Fig. 2 for the number of linguistic variables $m = 3$, shrinking factors $s = 0.7$ and overlapping factor $\beta = 1.$, SSMFs $A^{*}_{(i,L_i)}(x_i)$ for linguistic variable x_i:

$$A^{\beta}_{(i,L_i)} = A\left(\begin{matrix} x_i; & \frac{(1+\beta)A^{*}_{(i,L_{i-1})} + (1-\beta)A^{*}_{(i,L_i)}}{2}, & A_{(i,L_i)}, \\ \frac{(1-\beta)A^{*}_{(i,L_i)} + (1+\beta)A^{*}_{(i,L_{i+1})}}{2} & & \end{matrix} \right) \quad \text{for } L_i \varepsilon I_m, \tag{4}$$

where $A^{*}\ (i, l_i)$ is the principal value of $A^{*}_{(i,L_i)}(x_i)$ defined by

$$A^{*}_{(i,l_i)} = \frac{L_i}{m_i} s_i^{m_i - |L_i|} \quad \text{for } L_i \varepsilon I_m \tag{5}$$

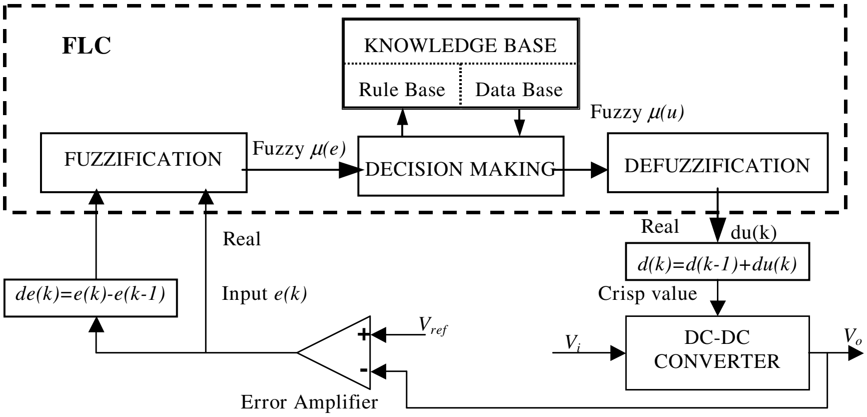


Fig. 1. Block diagram of the FLC for DC–DC converters.

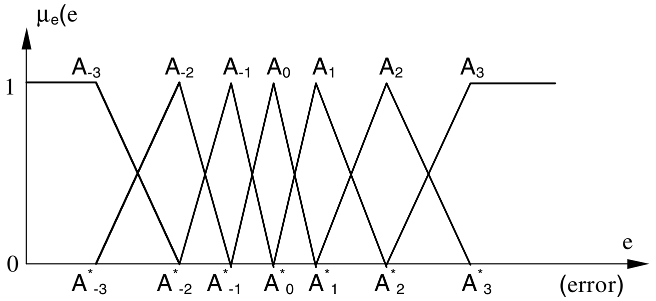


Fig. 2. Membership functions for typical trapezoidal family for $m = 3$, $s = 0.7$, $\beta = 1$.

where $s_i \in [0,\ 1]$ is the shrinking factor for linguistic variable x_i. L_i is the index set for the term set of linguistic variable of input.

In the following three equivalencies, let $A^*_{(i,-m_i-1)} = A^*_{(i,-m_i)}$, and $A^*_{(i,m_i+1)} = A^*_{(i,m_i)}$, $A_{(i,-m_i)}(A^*_{(i,-m_i)}) = 1$, $A_{(i,m_i)}(A^*_{(i,m_i)}) = 1$ and $\beta = 1$. The membership functions of input linguistic variable x_i is as

$$\begin{aligned} A_{(i,L_i)}(x_i) &= A^{\beta}_{(i,L_i)}(x_i)\Big|_{\beta=1} \\ &= \begin{cases} \frac{x_i - A^*_{(i,L_i-1)}}{A^*_{(i,L_i)} - A^*_{(i,L_i-1)}} & \text{for } A^*_{(i,L_i-1)} \leqslant x_i \leqslant A^*_{(i,L_i)} \\ \frac{A^*_{(i,lL_i+1)} - x_i}{A^*_{(i,L_i+1)} - A^*_{(i,L_i)}} & \text{for } A^*_{(i,L_i)} \\ \leqslant x_i \leqslant A^*_{(i,L_i+1)} 0 & \text{otherwise} \end{cases} \\ &= \wedge\left(x_i; A^*_{(i,L_i-1)}, A^*_{(i,L_i)}, A^*_{(i,L_i+1)}\right) \end{aligned} \tag{6}$$

Eq. (6) is illustrated in Fig. 3.

As for the output linguistic variable y, similar representations can be obtained:

$$B_l(y) = \wedge(y; B^*_{i-1}, B^*_i, B^*_{i+1}) \tag{7}$$

Here, B^*_i is principal value defined by

$$B^*_i = \frac{L}{m_y} s_u^{m_y-|L|}, \quad L \in I_{m_i} \equiv \{0, \pm1, \ldots, \pm I_m\} \tag{8}$$

where $s_u \in [0,\ 1]$ is the shrinking factor for output linguistic variable y_i. L is the index set for the term set of linguistic variable of output.

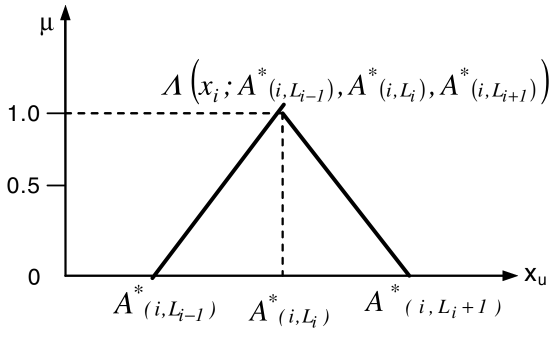

μ
1.0
0.5
0
$\Lambda\left(x_i; A^*(i,L_{i-1}), A^*(i,L_i), A^*(i,L_{i+1})\right)$
x_u
$A^*(i,L_{i-1})$
$A^*(i,L_i)$
$A^*(i,L_i+1)$

Fig. 3. Parameters of membership functions of typical trapezoidal family.

If shrinking factors is chosen one ($s = 1$), the membership functions have equal span. By applying various shrinking factors to the same linguistic variable, different membership function obtained to examine which is the most suitable for a specific application process. β is the overlapping factor whose reasonable range is [0, 1]. β can take values greater than unity as long as the resultant membership functions are rational in applications. The overlapping region increases monotonously as β increases. For $\beta = 0$ it is clear that there is no overlap between the SSMFs and if $\beta = 1$ the supports for the SSMFs have proper overlapping region.

For a Mamdani-type FLC, fuzzy rules are in the form

R_i : **IF** e is A_i and de is B_i **THEN** du_k is C_i

where A_i and B_i are fuzzy subsets in their universe of discourse and C_i is a fuzzy singleton. Each universe of discourse is divided into seven fuzzy subsets: PB (Positive Big), PM (Positive Medium), PS (Positive Small), ZE (Zero), NS (Negative Small), NM (Negative Medium) and NB (Negative Big).

The rule base of the FLC is created for obtained membership functions easily with index representation method. Table 1 illustrates the index representation of a simple rule mapping for $m_1 = m_2 = 3$ and the FLC has two input, single output.

By naming the numbered symbols (0 → Zero, 1 → Positive Small, 2 → Positive Medium..., −1 → Negative Small, −2 → Negative Medium ...), we recognize the classical antidiagonal rule base proposed by Mac Vicar Whelan (1977). This assumption has also been used by most other authors. Table 2 illustrates the linguistic labels representation of the control rule table.

The inference result of each rule consists of two parts, the weighting factor, w_i, of the individual rule, and degree of change in duty ratio C_i, according to the rule. The weighting factor w_i is obtained by means of *Mamdani's MIN fuzzy implication* of membership degrees $\mu_e(e)$ and μ_{de} $(de) \cdot C_i$ is retrieved from control rule table. The inferred output of each rule using Mamdani's MIN fuzzy implication is written as

$$w_i = \min\{\mu_e(e), \mu_{de}(de)\} \tag{9}$$

$$z_i = w_i \cdot C_i \tag{10}$$

where z_i denotes the fuzzy representation of change in duty ratio inferred by the ith rule.

Table 1
Simple rule mapping with index representation

e/*de*	−3	−2	−1	0	1	2	3
3	0	1	2	3	4	5	6
2	−1	0	1	2	3	4	5
1	−2	−1	0	1	2	3	4
0	−3	−2	−1	0	1	2	3
−1	−4	−3	−2	−1	0	1	2
−2	−5	−4	−3	−2	−1	0	1
−3	−6	−5	−4	−3	−2	−1	0

Table 2
The linguistic labels representation of rule base

e/*de*	NB	NM	NS	ZE	PS	PM	PB
PB	ZE	PS	PM	PB	PB	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PS	NM	NS	ZE	PS	PM	PB	PB
ZE	NB	NM	NS	ZE	PS	PM	PB
NS	NB	NB	NM	NS	ZE	PS	PM
NM	NB	NB	NB	NM	NS	ZE	PS
NB	NB	NB	NB	NB	NM	NS	ZE

Since the inferred output is a linguistic result, a defuzzification operation is performed next to obtain a crisp result. Here center of gravity method is preferred for defuzzification operator.

$$z = du = \frac{\sum_{i=1}^{4} z_i}{\sum_{i=1}^{4} w_i} = \frac{\sum_{i=1}^{4} w_i \cdot C_i}{\sum_{i=1}^{4} w_i} \tag{11}$$

where z or du is the result of change in duty ratio.

3. Adaptation algorithm for the fuzzy logic controller

AFLC is a FLC with an adaptation algorithm. Thus, AFLC adapts membership functions and computes the consequent parts of rules in the rule base. The inputs of AFLC are model data in the pattern file that is created from some data for desired output. The outputs of AFLC are membership functions and the consequent parts for FLC.

The AFLC can update its parameters which are membership function's shrinking factors s_e, s_{de} and s_u, according to the pattern file, by using adaptation algorithm. The update rule for each parameter in AFLC could derive as following. Assuming that the given training data sets has P entries, an error measure for the pth ($1 \leqslant p \leqslant P$) entry of the training data could defined as:

$$E_p = \frac{1}{2}(d_k - y_k) \tag{12}$$

where d_k is the kth component of the pth desired output vector and y_k is the kth component of the actual output vector. Obviously, when E_p is equal to zero or target error, the network is able to reproduce exactly the desired output vector in the pth training data pair. Thus the task here is to minimize an overall error measure, which is defined as:

$$E = \sum_{p=1}^{P} E_p \tag{13}$$

Finally, application of this adaptation algorithm can be accepted as adaptation of parameters as well as the training data in the pattern file.

4. Computer simulation of the AFLC

The implementation of the FLC is made for buck, boost and buck–boost converters. The circuit components and some properties of these converters is shown in Fig. 4.

In the simulation, the AFLC structure is as described in Section 3. It has two inputs and one output. Antecedent membership functions for inputs and output have seven linguistic variables as described Section 2. Thus the rule

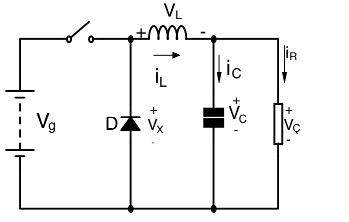
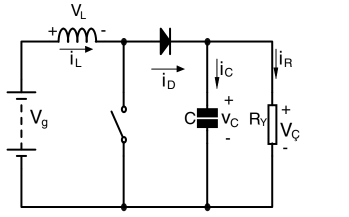
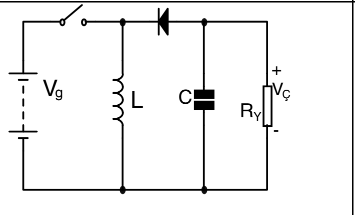
TYPE	BASIC CIRCUIT SCHEMA	VALUES
BUCK	 V_L + - i_L i_C i_R V_g D + V_X - + V_C - + V_Ç -	Input voltage V_{IN}= 15 V. Output voltage V_o=5 V. Max. Current I_R=6 A. Inductor L= 1 mH. Capacitor C=100 μF. Equivalent load Resistance R_L= 4 Ω. Switching frequency f_s= 20 kHz.
BOOST	 V_L + - i_L i_D i_C i_R V_g C + V_C - R_Y + V_Ç -	Input voltage V_{IN}= 5 V. Output voltage V_o=10 V. Max. Current I_R=6 A. Inductor L= 1 mH. Capacitor C=200 μF. Equivalent load Resistance RL= 4 Ω. Switching frequency f_s= 20 kHz.
BUCK-BOOST		Input voltage V_{IN}= 15 V. Output voltage V_o=5 V. Max. Current I_R=6 A. Inductor L= 1 mH. Capacitor C=200 μF. Equivalent load Resistance R_L= 4 Ω. Switching frequency f_s= 20 kHz.

Fig. 4. Some properties of buck, boost and buck–boost converters.

base has 49 rules. The output of rules *du* is change of duty ratio.

At the beginning, the pattern file is prepared. It contains three vectors which is error *e*, difference error *de* and change of duty ratio *du*. Each variable vector contains 600 sample data or another words the number of training data in the pattern file is 600 ($P = 600$).

Then the target error in total measure ε is chosen 1.0×10^{-4} and the program is run. Sixty one epochs have been performed and the normalized error measure E is computed as 9.281×10^{-5} and membership function's shrinking factors was found as $s_e = 1$, $s_{de} = 1$ and $s_u = 0.3$. At this condition, obtaining membership functions for both input error *e*, difference error *de* and output *du* are shown in Fig. 5a and b, respectively.

The consequent part of the rules is output, which is change of duty ratio *du*. After adaptation phase, the obtaining rule base of new FLC is shown in Table 3.

At the end of the AFLC simulation, the FLC is constructed obtaining new parameters from the adaptation algorithm and then the same FLC was run for buck, boost and buck–boost converters, separately.

The first simulation is for the buck converter and the obtained results are shown in Fig. 6. In this figure, Buck converter's duty ratio, inductor current, capacitor voltage and output voltage are shown for start up transient and load transient, respectively. For buck converter, the start up transient takes approximately 0.9 ms. After 3 ms, the equivalent load resistance R_L is reduced from 4 Ω to 2 Ω. The load transient takes approximately 1 ms.

The second simulation, the FLC was run for boost converter. Duty ratio, inductor current, capacitor voltage and output voltage of boost converter are shown in Fig. 7. As shown in Fig. 7d, the start up transient of output voltage takes approximately 1.3 ms, for boost converter. After 3 ms, the equivalent load resistance R_L is reduced from 4 Ω to 2 Ω. The load transient of output voltage is made immediately.

In the last example, the FLC was run for buck–boost converter. Buck–boost converter's duty ratio, inductor current, capacitor voltage and output voltage are shown in Fig. 8 for start up transient and load transient, respectively. For buck converter, the start up transient takes approximately 1.2 ms. After 3 ms, the equivalent load resistance R_L is reduced from 4 Ω to 2 Ω. The load transient of output voltage is update immediately.

The results indicate that this method provides an easy and systematic way in designing the AFLC. The generated membership functions and rule base are general and could be used for any DC–DC converter without any modification.

5. Implementation of the AFLC with microcontroller

The AFLC described in the previous sections is implemented on a ST52E420 microcontroller. It is 8-bit microcontroller and the erasable EPROM version, which has 4 Kbytes program and data EPROM. It is able to perform, in an efficient way, both Boolean and fuzzy algorithms, in order to reach the best performances that the two methodologies allow. The schematic diagram of the controller circuit is given in Fig. 9.

The microcontroller allows describing a problem using a linguistic model instead of mathematical model. The microcontroller includes an 8-bit sampling Analog to Digital (A/D) converter with a 8 analog channel fast

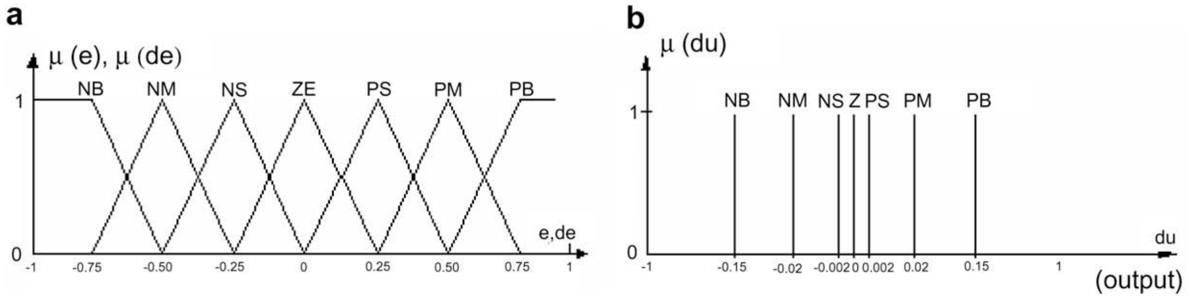


Fig. 5. Membership functions: (a) for error and difference error, (b) for output *du*.

Table 3
Rules table of FLC

e\de	NB	NM	NS	ZE	PS	PM	PB
NB	−0.015	−0.015	−0.015	−0.015	−0.02	−0.002	0
NM	−0.015	−0.015	−0.015	−0.02	−0.002	0	0.002
NS	−0.015	−0.015	−0.02	−0.002	0	0.002	0.02
ZE	−0.015	−0.02	−0.002	0	0.002	0.02	0.15
PS	−0.02	−0.002	0	0.002	0.02	0.15	0.15
PM	−0.002	0	0.002	0.02	0.15	0.15	0.15
PB	0	0.002	0.02	0.15	0.111	0.15	0.15

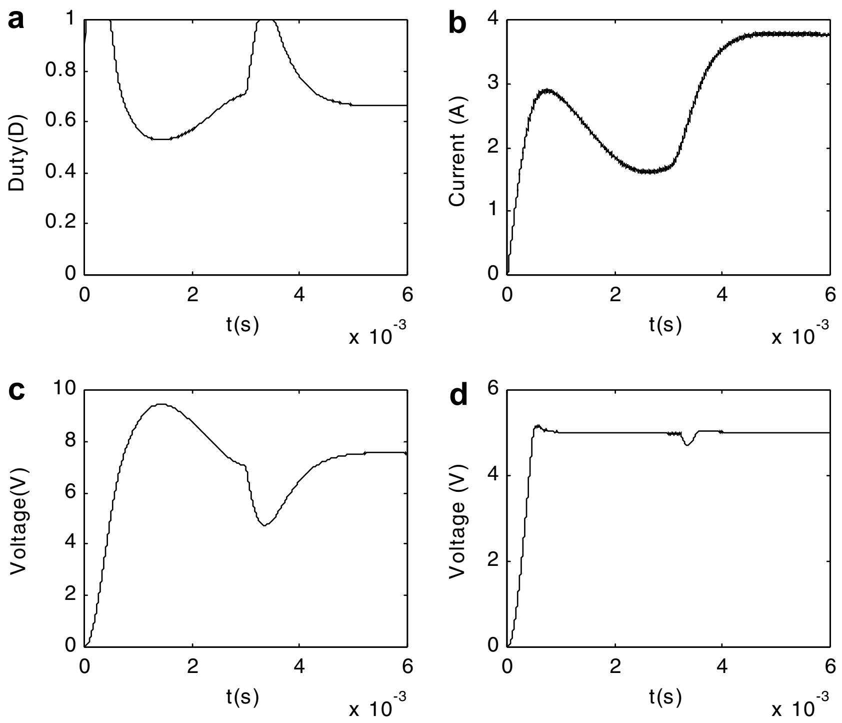


Fig. 6. For buck converter, (a) duty ratio, (b) inductor current, (c) capacitor voltage, (d) output voltage.

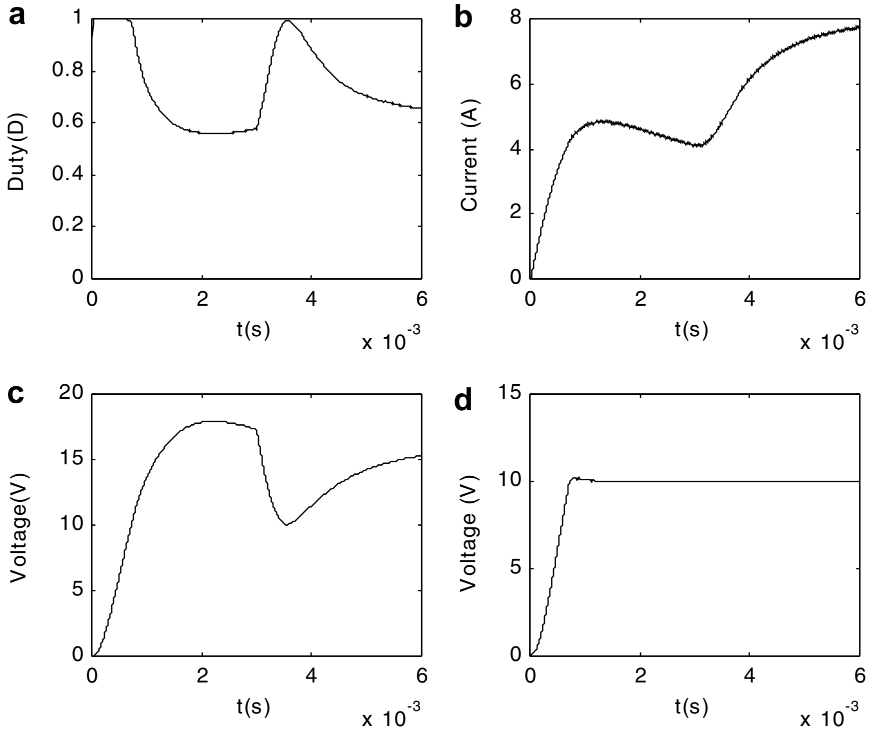


Fig. 7. For boost converter, (a) duty ratio, (b) inductor current, (c) capacitor voltage, (d) output voltage.

multiplexer and 2.5 reconfigurable digital ports in order to transfer data from/to the on-chip Register Files. A three independent PWM/Timers are included allows managing directly power devices and high frequency PWM controls. Working clock frequency is 20 MHz to drive on-chip clock oscillator. The switching frequency is selected 19.6 kHz. The analog input of AIN1 is connected the voltage reference is a 5 V, through the 4.7 kΩ potentiometer for tuning

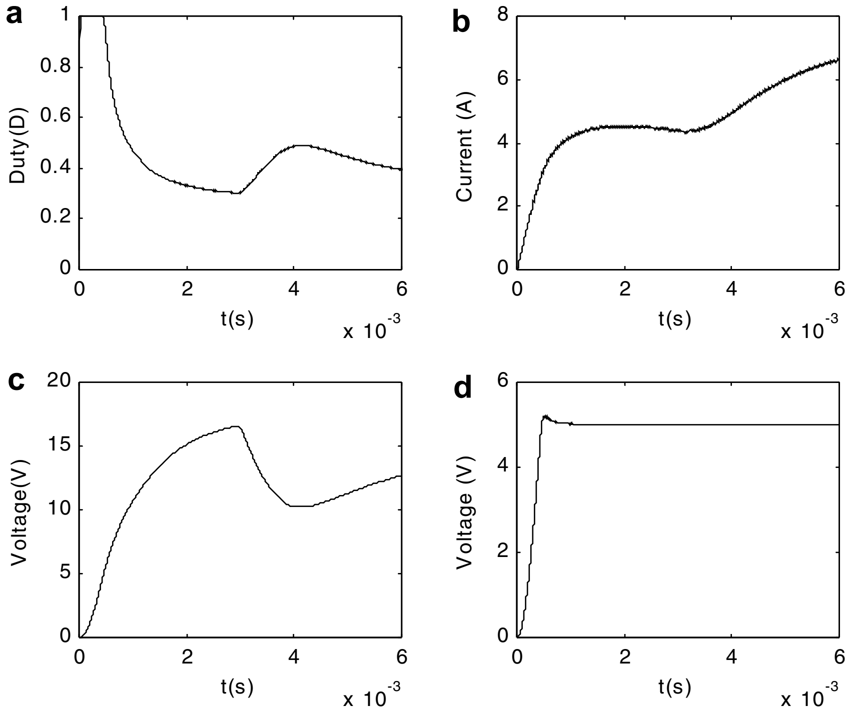


Fig. 8. For buck–boost converter, (a) duty ratio, (b) inductor current, (c) capacitor voltage, (d) output voltage.

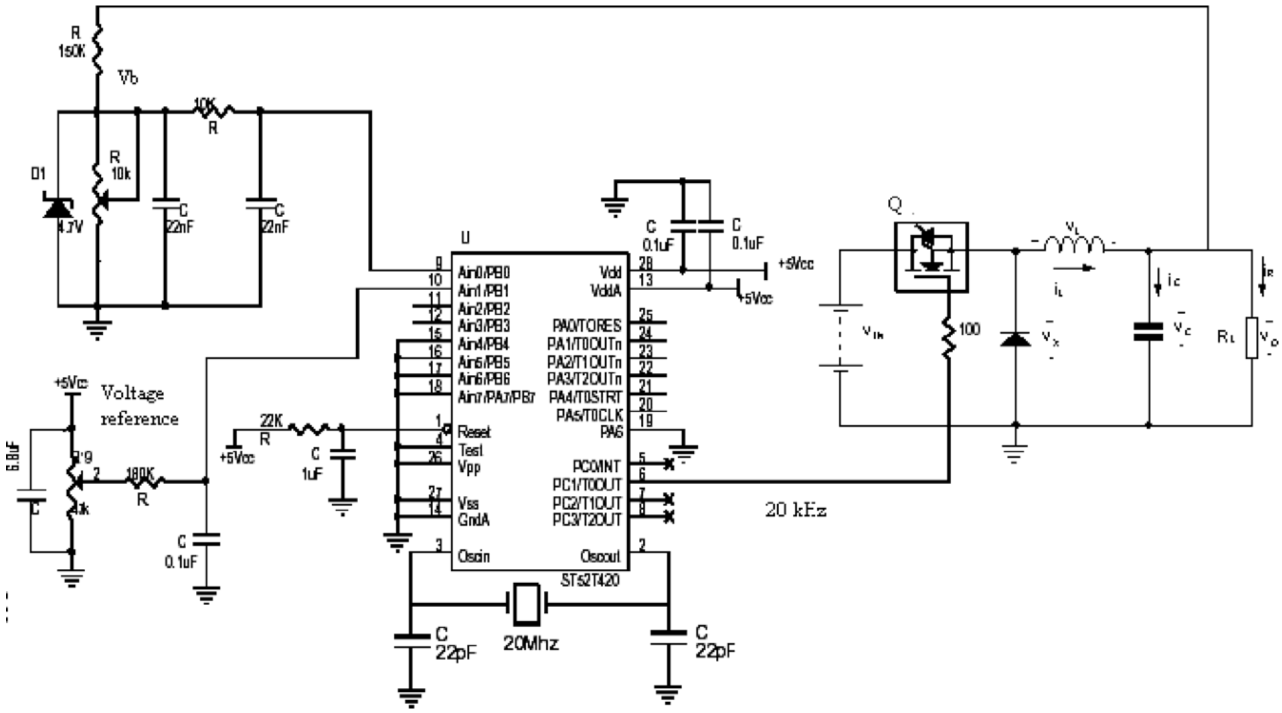


Fig. 9. The schematic diagram of the controller circuit.

reference voltage. The other analog input of AIN0 is connected output of DC–DC converter for tuning level of DC–DC converter output. Here the controller is used for buck, boost and buck–boost converters, but it can be implemented to other types of switch-mode DC–DC converters without making any change.

Experimental results are as follow and the circuit parameters are exactly the same as those used in the simulations. These experiments are made for every three converters in the simulations, separately.

The output voltage start up transient and load transient of buck converter are shown in Fig. 10a and b, respectively.

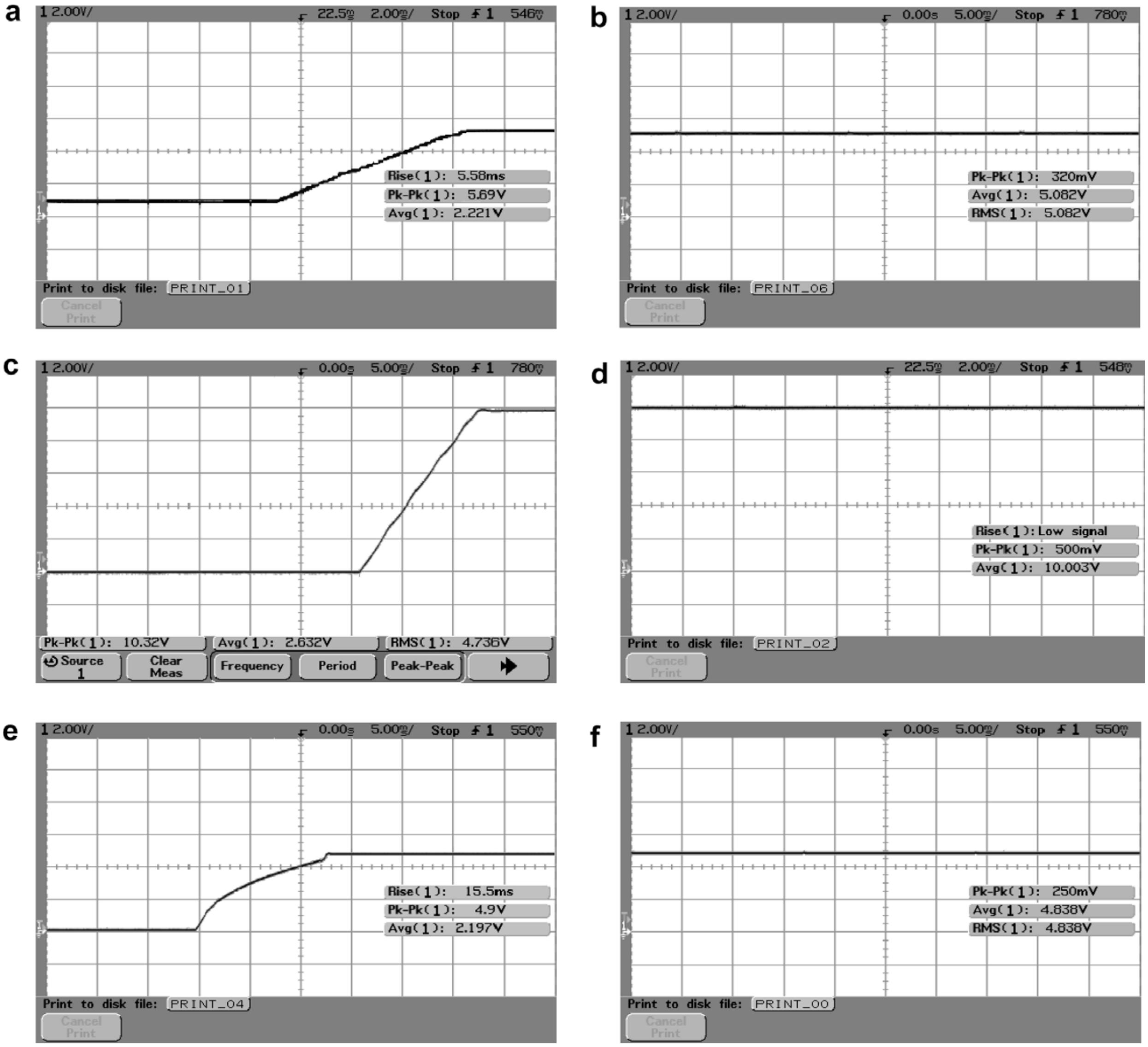


Fig. 10. For output voltage of buck converter (a) start up transient (b) load transient, for output voltage of boost converter, (c) start up transient (d) load transient, for output voltage of buck–boost converter, (e) start up transient (f) load transient.

The start up transient takes approximately 10 ms. At the beginning load was 4 Ω. It is changed to 2 Ω and output voltage has almost same value about (5.082 V). The load transient takes approximately 1 ms. But output voltage ripple increased from 250 mV to 320 mV.

For boost converter, the output voltage start up transient and load transient are shown in Fig. 10c and d, respectively. The start up transient takes approximately 13 ms. The load transient takes approximately 1 ms with 500 mV output voltage ripple.

The output regulation and the load regulation of buck–boost converter are shown in Fig. 10e and f, respectively. The start up transient takes approximately 13 ms. The load transient of output voltage is formed immediately. Output voltage ripple is about 250 mV at 4.838 V output voltages.

Experimental results for buck, boost and buck–boost converters demonstrated that responses can be obtained with the AFLC. The simulation and the implementation of the AFLC for buck, boost and buck–boost converters results demonstrated that the converters are stable and can be regulated with a good performance under different input disturbance and load variation. The results also indicate that the AFLC is general and can be applied to any DC–DC converter topologies. Thus, the same microcontroller software can be used to control any switch-mode converters without any modifications.

6. Conclusion

In this paper, an adaptive fuzzy logic controller has described for DC–DC converter output voltage regulation and have implemented on a 8-bit microcontroller. The AFLC is able to regulate the output voltage of buck, boost and buck–boost converters to desired value despite change in load. Since these converters, buck, boost and buck–boost, are controlled using the same AFLC algorithm without any modifications to microcontroller program. This shows the proposed algorithm is general and can be applied to any DC–DC converter topologies practically.

Experiments results for buck, boost and buck–boost converters have demonstrated effectiveness of described AFLC and showed satisfactory results without preconstructed rules of any expert. The results indicate that the AFLC is general and can be applied to any DC–DC converter topologies.

References

Alejo, D., Maussion, P., & Faucher, J. (2003). Multiple model control of a buck DC/DC converter. *Mathematics and Computers in Simulation, 63*, 249–260.

Calderón, A. J., Vinagre, B. M., & Feliu, V. (2006). Fractional order control strategies for power electronic buck converters. *Signal Processing, 86*(10), 2803–2819.

Carrosco, C. M., Torralba, A., Ridao, F. P., & Franquelo, L. G. (1994). A fuzzy logic controller for power converters using a cell state algorithm. In *IEEE international conference on industrial electronics, control and instrumentation, IECON'94* (pp. 1325–1330), Bologna.

Chien, C. L., & Hsieh, C. T. (1996). User-friendly design method for fuzzy logic controller. *IEE Pro-Control Theory Application, 143*(4), 358–366.

Galichet, S., & Foulloy, L. (1995). IEEE fuzzy logic controllers: synthesis and equivalencies. *Transactions on Fuzzy Systems, 3*(2), 140–148.

Gupta, T., Boudreaux, R. R., Nelms, R. M., & Hung, J. Y. (1997). Implementation of a fuzzy logic controller for DC–DC converters using an inexpensive 8-b microcontroller. *IEEE Transaction on Industrial Electronics, 44*(5), 661–668.

Higuchi, K., Nakano, K., Kajikawa, T., Takegami, E., Tomioka, S., & Watanabe, K. (2004). Robust control of DC–DC converter by high-order approximate 2-degree-of-freedom digital controller, IECON 2004. In *30th annual conference of IEEE, 2–6 November* (Vol. 2, pp. 1839–1844).

Huh, S., & Park, G. (1999). An adaptive fuzzy logic controller for power converters. In *FUZZY-IEEE '99*. (Vol. 1, pp. 434–439).

Leyva, R., Martinez-Salamero, L., Jammes, B., Marpinard, J. C., & Guinjoan, F. (1997). Identification and control of power converters by means of neural networks. *IEEE Transactions on Circuits and Systems-1: Fundamental Theory and Applications, 44*(8), 735–742.

Lin, B. R. (1997). Analysis of neural and fuzzy-power electronic control. *IEE Proceedings – Science, Measurement and Technology, 144*(1), 25–33.

Lin, S. C., & Tsai, C. C. (2005). Adaptive backstepping control with integral action for PWM buck DC–DC converters. *Journal of the Chinese Institute of Engineers, 28*(6), 977–984.

Mac Vicar Whelan, P. J. (1977). Fuzzy set for man machine interaction". *International Journal of Man–Machine Studies, 8*, 687–697.

Mamdani, E. H. (1977). Application of fuzzy logic approximate reasoning using linguistic synthesis. *IEEE Transactions on Computer, C-26*(12), 1182–1191.

Mattavelli, P., Rossetta, L., Spiazzi, G., & Tenti, P. (1997). General-purpose fuzzy logic controller for DC–DC converters. *IEEE Transactions on Power Electronics, 12*(1), 79–85.

Park, Y. M., Moon, U. C., & Lee, K. Y. (1995). A self-organizing a fuzzy logic controller for dynamic systems using a fuzzy auto-regressive average model. *IEEE Transactions on Fuzzy Systems, 3*(1), 75–82.

Rubai, A., Ofoli, A. R., Burge, L., & Garuba, M. (2005). Hardware İmplementation of an adaptive network-based fuzzy controller for DC–DC converters. *IEEE Transactions on Industry Applications, 41*(6), 1557–1565.

So, W. C., & Tse, C. K. (1996). Development of a fuzzy logic controller for DC/DC converters: Design, computer simulation, and experimental evaluation. *IEEE Transactions on Power Electronics, 11*(1), 24–31.

So, W. C., Tse, C. K., & Lee, Y. S. (1994). A fuzzy logic controller for DC–DC converter. In *IEEE power electronics specialists conference records* (pp. 315–320).

Wang, L. X. (1993). Stable adaptive fuzzy logic controller of nonlinear systems. *IEEE Transactions on Fuzzy Systems, 1*(2), 146–155.

Wong, L. K., Leung, F. H. F., & Tam, P. K. S. (1995). A neuro-fuzzy logic controller applying to a Cuk converter. In *IEEE IECON 21st international conference* (Vol. 1, pp. 446–450).