UDC 620.92+502.174.3

Fuzzy Logic Controller Design for Photovoltaic Power Station

Imad Elzein, Yury N. Petrenko, Minsk
Belarusian National Technical University, imad.zein@liu.edu.lb; ypetrenko@bntu.by

Abstract
Solar panels have a nonlinear voltage-current characteristic, with a distinct maximum power point (MPP), which depends on the environmental factors, such as temperature and irradiation. In order to continuously harvest maximum power from the solar panels, they have to operate at their MPP despite the inevitable changes in the environment. This is why the controllers of all solar power electronic converters employ some method for maximum power point tracking (MPPT). Over the past years many MPPT techniques have been published and based on that the main paper's objective is to analyze one of the most promising MPPT control algorithms: fuzzy logic controller.

I. Introduction

According to the realization of high efficiency and low cost photovoltaic (PV) modules, interest in photovoltaic power generation system has increased over the past decade as a clean and infinite energy [1]. The PV modules have maximum operating points corresponding to the surrounding condition such as intensity of the sunlight, the temperature of the PV modules, cell area, and load. When solar energy is used as a power source, the output power has to be maximized by improving the efficiency of the power conditioning equipment used and implementing an adaptive power controller that automatically tracks the system to the point of maximum power delivered from the solar panel under all conditions.

Solar energy has offered promising results in the quest of finding the solution to the problem. The harnessing of solar energy using PV modules comes with its own problems that arise from the change in insulation conditions. These changes in insulation conditions severely affect the efficiency and output power of the PV modules. A great deal of research has been done to improve the efficiency of the PV modules. A number of methods of how to track the maximum power point of a PV module have been proposed to solve the problem of efficiency and products using these methods have been manufactured and are now commercially available for consumers [1].

Maximum Power Point Tracking (MPPT) is the newest concept which helps to extract the maximum possible power from a PV array. The MPPT methods are various in the complexity, convergence speed, popularity, cost, operating range, sensor dependence, capability of escaping from local optima and their applications [2]-[7].
One of the most significant issues in PV system and MPPT efficiency is DC-DC converter. 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 traditional PID based controllers and fuzzy logic controller (FLC), neural networks (NN), and neuro-fuzzy controller or adaptive

fuzzy logic controller (AFLC) which have been used to control buck, boost and buck–boost converter which were presented.

The authors [8-14] have designed a different control model and implementation has been made to regulate DC–DC converter by using a digital signal processor (DSP TMS320C50).

Different control technologies were used to control DC–DC converter using a microcontroller and an extra specialized hardware proposed a FLC that uses an optimal algorithm, and they have given experimental results.

Design of fuzzy logic has been applied to a broad variety of engineering problems, particularly those having nonlinear dynamics [15]-[20]. Fuzzy logic controllers have been implemented as embedded controllers for frequency controlled induction motor drives. Numerous electric motor drive problems have been solved using fuzzy principles [21]-[23]. Studies have also recommended utilizing FLC in situations where (1) there is no precise 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.

Furthermore, corresponding authors made their effort in the design of Fuzzy Logic Controllers and demonstrated 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 is a synthesis of works by [10,15,16,21,22] and introduces a systematic approach to construct FLC for DC–DC converters as a part of Maximum Power Point Tracing system of Photovoltaic station to adapt to photovoltaic modules under varying operating conditions and the nonlinear properties of DC-DC power converters.

The modified FLC (MFLC) optimizes membership functions and rule base of the FLC were obtained from training data in the pattern file.

An MFLC approach is general in the sense that it is almost the same control rules can be applied to other applications [21].

2. Fuzzy logic controller

In recent years, there has been increasing interest in the development of efficent control strategies to improve dynamic behavior of DC–DC converters by using fuzzy logic controller (FLC), neural networks (NN), and neuro-fuzzy controller or adaptive fuzzy logic controller (AFLC) - have been used to control buck, boost and buck–boost converter were presented.

The authors [10,11,17] have designed a different control model and implementation has been made to regulate DC–DC converter by using a digital signal processor (DSP TMS320C50).

Different control technologies were used to control DC–DC converter using a microcontroller and extra specialized hardware have proposed an FLC that uses an optimal algorithm, and they have given experimental results.

Design of fuzzy logic has been applied to a broad variety of engineering problems, particularly those having nonlinear dynamics. Fuzzy logic controllers have been implemented as embedded controllers for frequency controlled induction motor drives. Numerous electric motor drive problems have been solved using fuzzy principles [22,23,24]. Researches have also proposed to

Use the FLC in situations where it could be useful in (1) there is no precise 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.

The use of fuzzy logic control has become popular over the last decade because it can deal with imprecise inputs, does not need an accurate mathematical model and can handle nonlinearity. Microcontrollers have also helped in the popularization of fuzzy logic control. [5]
The implementation of fuzzy logic is used to have a faster controller response and to increase system stability once reached the MPP. The tracking of the MPP will be divided into two phases: the first phase is of tough research, with a significant step to improve the response of the MPPT controller, the second one is the fine phase where the step is very small, thus ensuring the system stability and decrease the maximum oscillations around the MPP. This feature of the fuzzy controller demonstrates its effectiveness and makes it among the best MPP tracking devices.

The fuzzy controller consists of three blocks: the Fuzzification of input variables which is performed in the first block, it allows the passage from the real domain to fuzzy domain. The second block is devoted to inference rules, while the last block is the Defuzzification for returning to the real domain. This last operation uses the center of mass to determine the value of the output [16].
The FLC block diagram to control DC-DC power converter in MPPT system of PVS is presented in Fig.1.

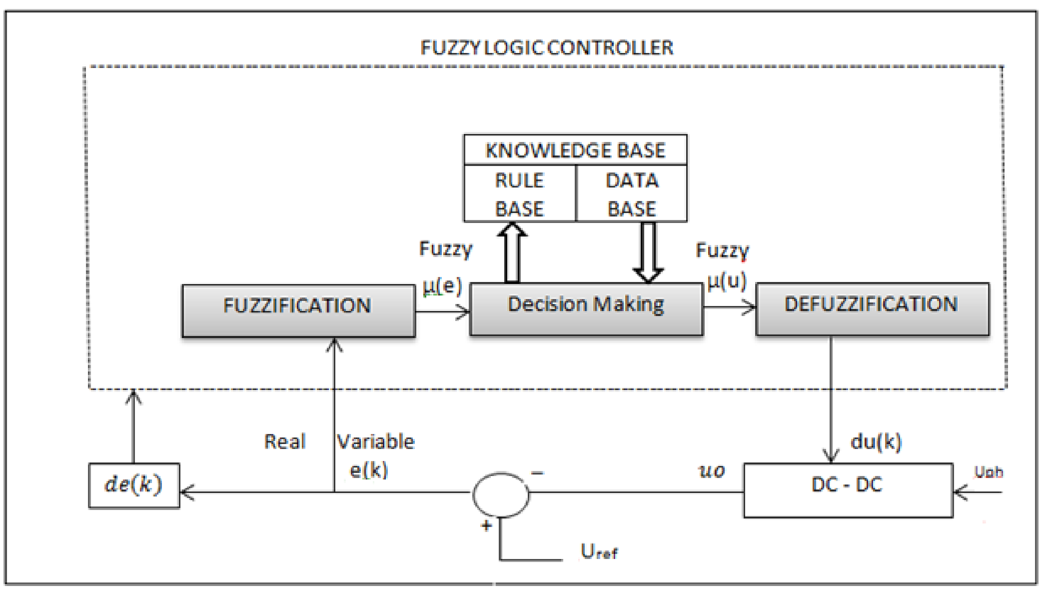


Fig.1 The basic structure of the Fuzzy Logic Controller

Corresponding authors made their effort in design of Fuzzy Logic Controllers and demonstrated some difficulties in the selection of optimized membership functions and fuzzy rule base, which is traditionally achieved by a tedious trial-and-error process.
The modified FLC (MFLC) optimizes membership functions and rule base of the FLC were obtained from training data in the pattern file.

The inputs of Fig.1 of the FLC are the error $\boldsymbol{e}$ and difference of error **de** respectively and they are defined as

$$e = U_{ref} - U_0 \qquad (4)$$

$$de(k) = e(k) - e(k-1) \qquad (5)$$

Where U_{ref} is reference output voltage, U_0 is actual output voltage of DC–DC converter at the kth sampling time.

The output of the FLC is a change in duty ratio *(du(k))*.

Duty ratio *d(k)*, at the kth sampling time, is defined as:

$$d(k) = d(k-1) + \mathrm{du(k)} \qquad (6)$$

Knowing that, the output of the controller then sends through PWM out to DC-DC converter to generate desired switching action (Fig.5).

Shrinking-span membership functions algorithm is used to construct membership functions for FLC. Then the result of (5) is send through the PWM controller to DC–DC converter to generate desired switching action. By using this method the designer of an FLC assigns only the number of elements of term set and shrinking factor.

In fact, the shrinking-span membership functions [16] (SSMF) is constructing membership functions method for FLC which, in compare to [21], generates a series of orderly arranged membership functions $A(x_i)s$ in the FLC for a linguistic variable across its universe of discourse. For example [16], widely used trapezoidal family SSMF is showing on Fig.6 for the membership number of linguistic variables m=3, shrinking factors s=0.65 and overlapping b=1. In case when shrinking factor is chosen one, the membership functions have equal span .Using various shrinking factors to the same linguistic variable, different membership function obtained to examine which is the most suitable for a specific application process.

The overlapping factor has range [0,1] and increases monotonously as $\boldsymbol{b}$ increases.

It is clear that there is no overlap between SSMF and if $\boldsymbol{b}$=1 the supports for the SSMF have proper overlapping region.

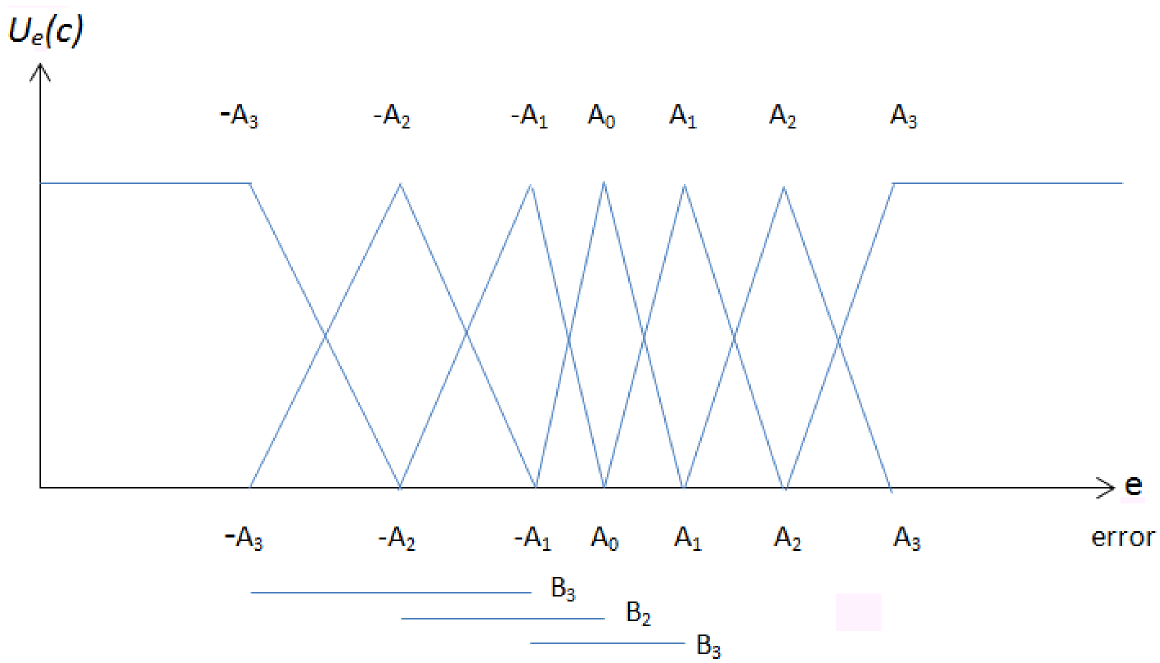


Figure 2- The family of the membership functions.

If shrinking factors is chosen one (s = 1), the membership functions have equal span. By applying various shrinking factors to the same linguistic variable, deferent membership function obtained to examine which is the most suitable for a specific application process. Let us take that *b* is the overlapping factor whose reasonable range is [0, 1]. In our case we will consider that b can take values greater than unity as long as we as experts consider that resultant membership functions are rational in applications. The overlapping region increases monotonously as *b* increases. For b = 0 it is clear that there is no overlap between the SSMFs. Diversity of *b* is shown in Fig.6 as B1, B_2, and B3. Consider B=1 the supports for the SSMFs have proper overlapping region.

For a Mamdani-type FLC [21], fuzzy rules are in the form:

R_i **: IF** *e* is $\mathbf{A_i}$ and *de* is B_i **THEN** du_k is $\mathbf{C_i}$, where $\mathbf{A_i}$ and $\mathbf{B_i}$ are fuzzy subsets in their universe of discourse and $\mathbf{C_i}$ is a fuzzy singleton.

Each universe of discourse is divided into seven (as an example; for more information see [22] 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 the way to make it easy to obtain membership functions with index representation method.

Table 1 illustrates the index representation of a simple rule mapping for m1 = m2 = 3 and the FLC has two inputs, single output

By naming the numbered symbols (0 ? Zero, 1 ? Positive Small, 2 ? Positive Medium. . ., 1 ? Negative Small, 2? Negative Medium. . .), one can recognize anti-diagonal rule base proposed by number of authors [18,20,21]. Table 2 illustrates the linguistic labels representation of the control rule table.

The inference result of each rule consists of two parts of 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 $\mu_{de}(de)$. C_i is retrieved from control rule table. As a result the inferred output of each rule using Mamdani's MIN fuzzy implication is given as

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

$$z_i = w_i \cdot C_i \quad (10)$$

where z_i denotes the fuzzy representation of change in duty ratio inferred by the i-th rule.

Table 1
Simple rule mapping with index representation

elde	-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

elde	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

The results above results were received as linguistic results therefore we must employ next a defuzification operator to obtain a crisp result. Among others we prefer center of gravity method for defuzzification *Z* is equal to *du* where *Z* or du is the result of change in duty ratio du.

3. Modernized algorithm for the fuzzy logic controller

MFLC, which is discussed in this paper, is an ordinary FLC with a modernized (adaptation) algorithm. Thus, MFLC adapts membership functions and computes the consequent parts of rules in the rule base. The inputs of MFLC are model data in the pattern file that is created from some expert knowledge data for desired output. The outputs of the controller are membership functions and the consequent parts for the controller. The MFLC updates its parameters (which are membership function's shrinking factors) S_e, S_{de} and S_u according to the pattern file, by using modified algorithm.

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

The implementation of the MFLC is made for boost, buck, and buck–boost converters as part of the MPPT system. The circuit components and parameters of these converters can be found in [21,22]. It has two inputs and one output. Number of Antecedent membership functions for inputs and output can be in range of 5 or 7, as it is shown in [22]. Thus the rule base has 49 outputs. The output of rules ***du*** is the change of duty ratio.

At the first, the pattern file is to be prepared. It contain of three vectors which are error ***e***, difference error ***de*** and change of duty ratio ***du***. Each variable vector contains a number of sample data or by another words the number of training data in the pattern file.

The MFLC algorithm described above can be implemented on a number of devices. We will consider implementation on a ST52E420 microcontroller, which is an 8-bit microcontroller and the erasable EPROM version, which has 4 Kbytes program and data EPROM. This model has been chosen 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 illustrated in Fig**. 7.**

This microcontroller has another important role in allowing describing a problem using a linguistic model instead of mathematical model. The microcontroller includes an 8-bit sampling (A/D) converter with an 8 analog channel fast 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.

6 FUZZY COMPUTATION (DP)

The ST52T410/ST52x420 Decision Processor

(DP) main features are: UNLIMITED number of Rules and Fuzzy Blocks. The limits on the number of Fuzzy Rules and Fuzzy program blocks are only related to the Program/Data Memory size.

■ Up to 8 Inputs with 8-bit resolution;

■ 1 Kbyte of Program/Data Memory available to store more than 300 to Membership Functions (Mbfs) for each Input;

■ Up to 128 Outputs with 8-bit resolution;

■ Possibility of processing fuzzy rules with an UNLIMITED number of antecedents;

■ UNLIMITED number of Rules and Fuzzy Blocks.
The limits on the number of Fuzzy Rules and Fuzzy program blocks are only related to the Program/Data Memory size.

Fuzzy Inference. The block diagram shown in Figure 3 describes the different steps performed during a Fuzzy algorithm. The ST52T410/ST52x420 Core allows for the implementation of a Mamdani type fuzzy inference with crisp consequents. Inputs for fuzzy inference are stored in 8 dedicated Fuzzy input registers. The LDFR instruction is used to set the Input Fuzzy registers with values stored in the Register File. The result of a Fuzzy inference is stored directly in a location of the Register File.

In Fuzzyfication phase the intersection (alpha weight) between the input values and the related Mbfs (Fig.4) is performed. Eight Fuzzy Input registers are available for Fuzzy inferences.

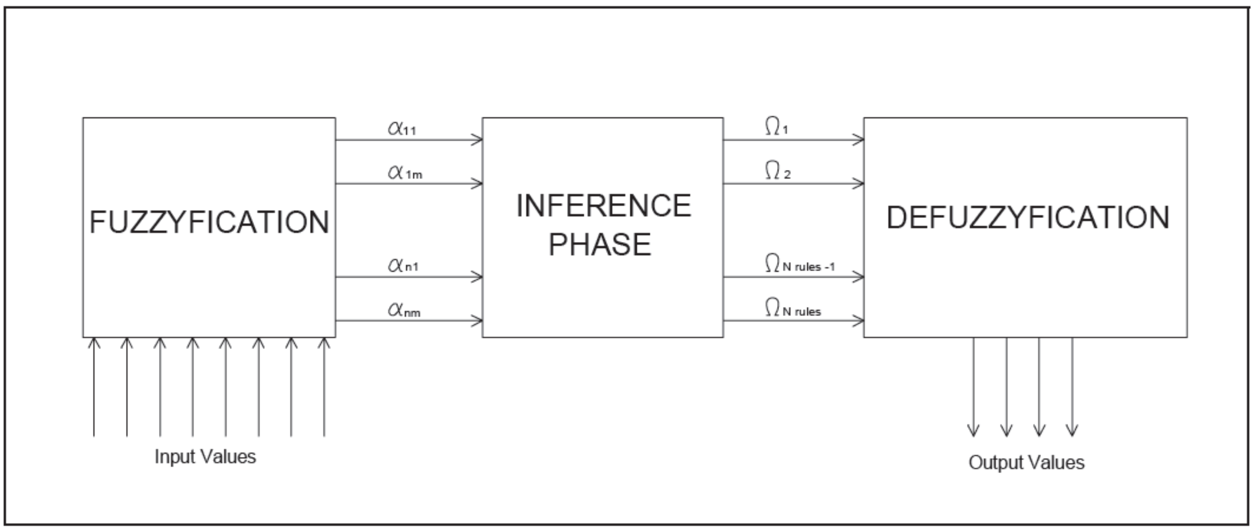


Figure 3- Fuzzy Inference

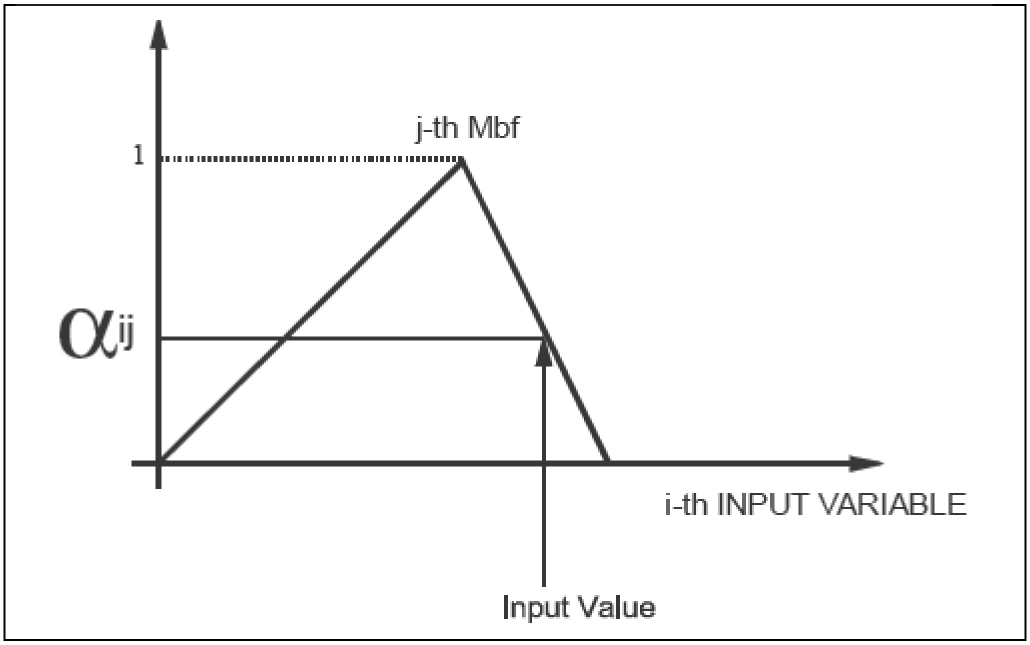


Figure 4- Alpha Weight Calculation

After loading the input values by using the LDFR assembler instruction, the user can start the fuzzy inference by using the FUZZY assembler instruction. During fuzzyfication input data is transformed in the activation level (alpha weight) of the Mbf's.

The Inference Phase manages the alpha weights obtained during the fuzzyfication phase to compute the truth value (ω) for each rule. This is a calculation of the maximum (for the OR operator) and/or minimum (for the AND operator) performed on alpha values according to the logical connectives of Fuzzy Rules (Fig.5). Several conditions may be linked together by linguistic connectives AND/OR, NOT operators and brackets.

The truth value ω and the related output singleton are used by the Defuzzyfication phase, in order to complete the inference calculation.

Defuzzyfication. In this phase the output crisp values are determined by implementing the consequent part of the rules. Each consequent Singleton Xi is multiplied by its weight values ωi, calculated by the Decision processor, in order to compute the upper part of the Defuzzyfication formula.

Each output value is obtained from the consequent crisp values (Xi) by carrying out the following Defuzzyfication formula

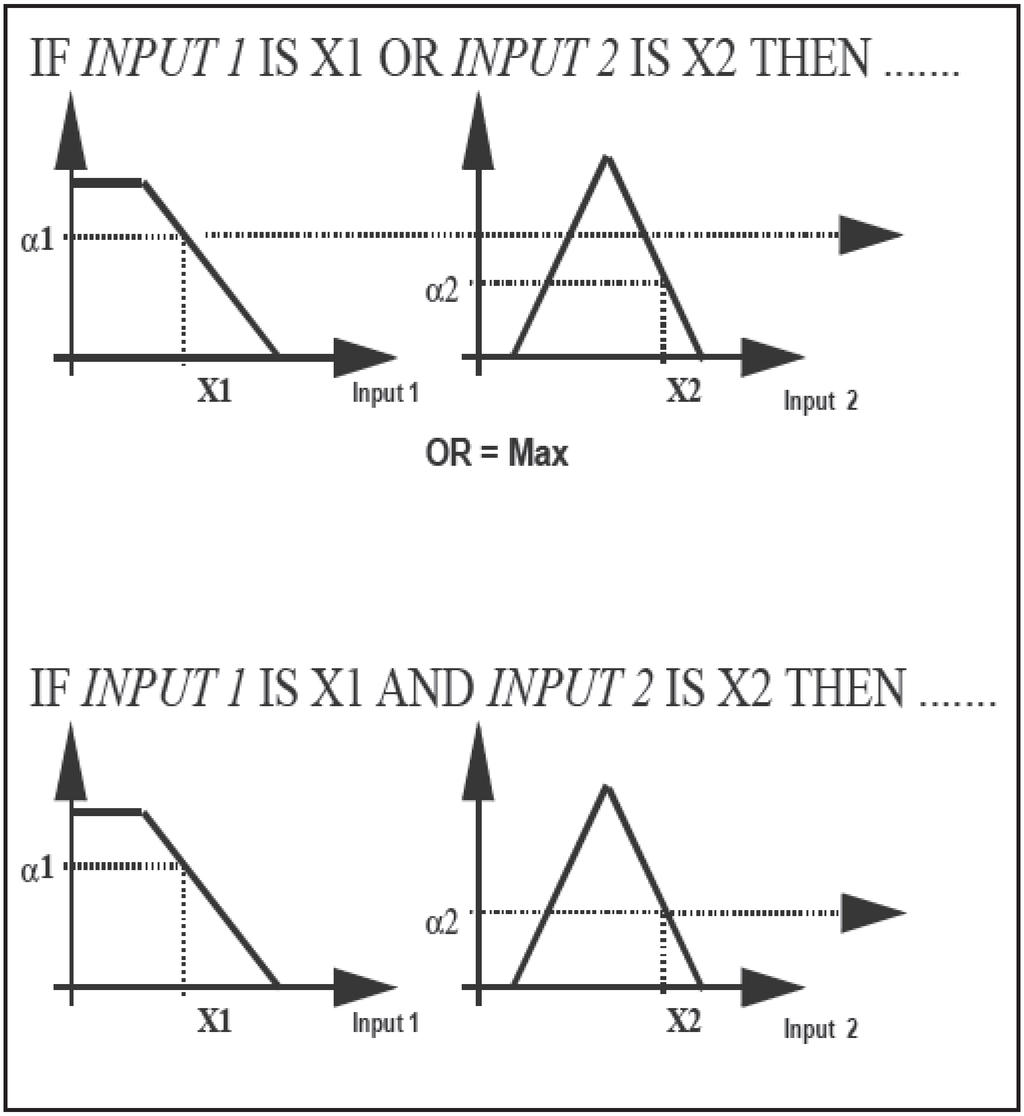


Figure 5- Fuzzyfication.

$$Y_i=\frac{\sum_i^N X_{ij}\,\omega_{ij}}{\sum_j^N \omega_{ij}}$$

$$Y_i=\frac{\sum_i^N X_{ij}\,\omega_{ij}}{\sum_j^N \omega_{ij}}$$

where:
i = identifies the current output variable,
N = number of the active rules on the current output,
ω_{ij} = weight of the j-th singleton,
X_{ij} = abscissa of the j-th singleton.
The Decision Processor outputs are stored in the RAM location i-th specified in the assembler instruction OUT.
Input Membership Function. The Decision Processor allows the management of

triangular Mbfs. In order to define an Mbf, three different parameters must be stored on the Program/Data Memory (see Figure 6):

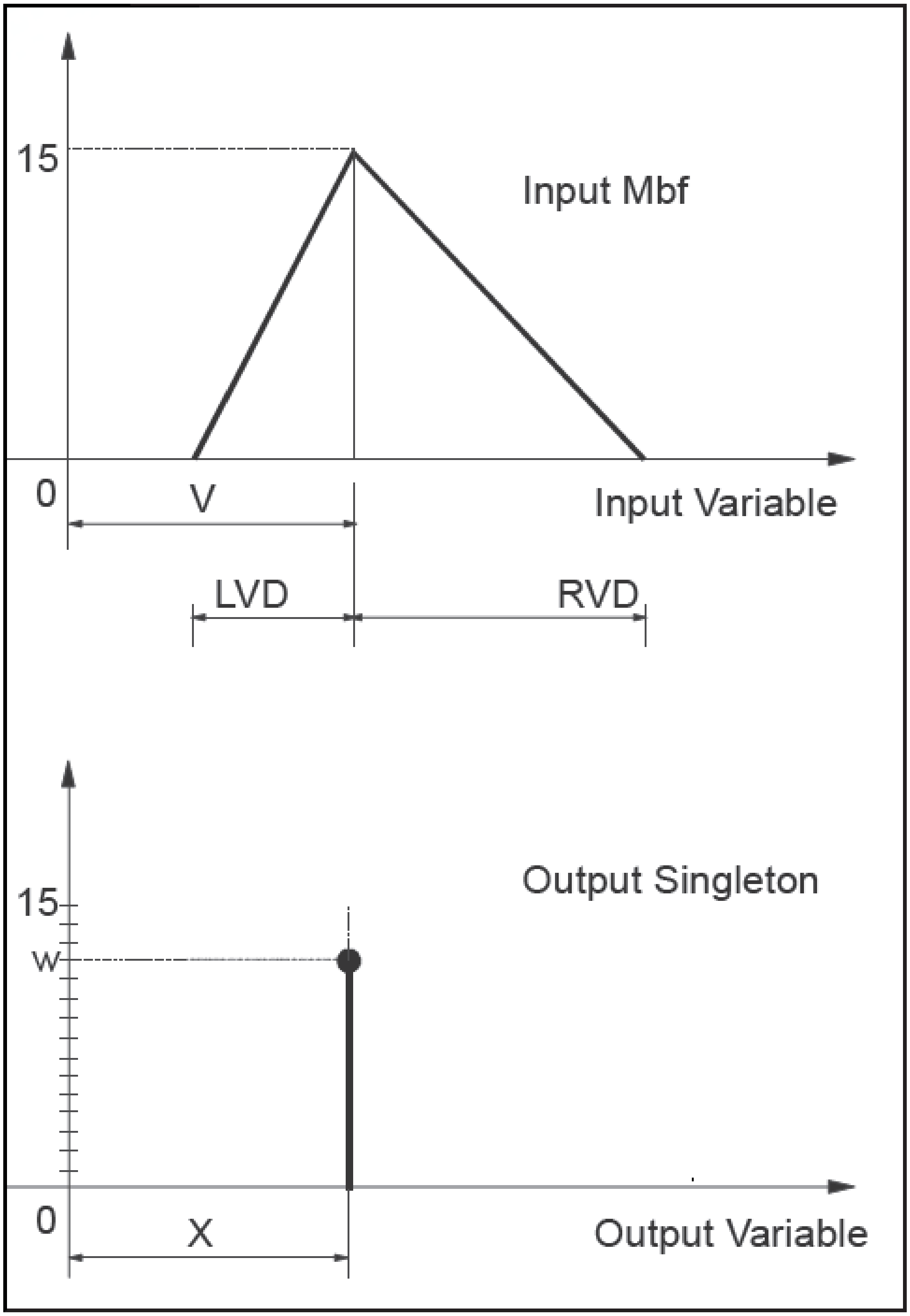


Figure 6- Mbfs Parameters.

■ the vertex of the Mbf: V;
■ the length of the left semi-base: LVD;

■ the length of the right semi-base: RVD;
In order to reduce the size of the memory area and the computational effort the vertical range of the vertex is fixed between 0 and 15 (4 bits).
By using the previous memorization method different kinds of triangular Membership Functions may be stored. Figure 7 shows some examples of valid Mbfs that can be defined in ST52T410/ST52x420.

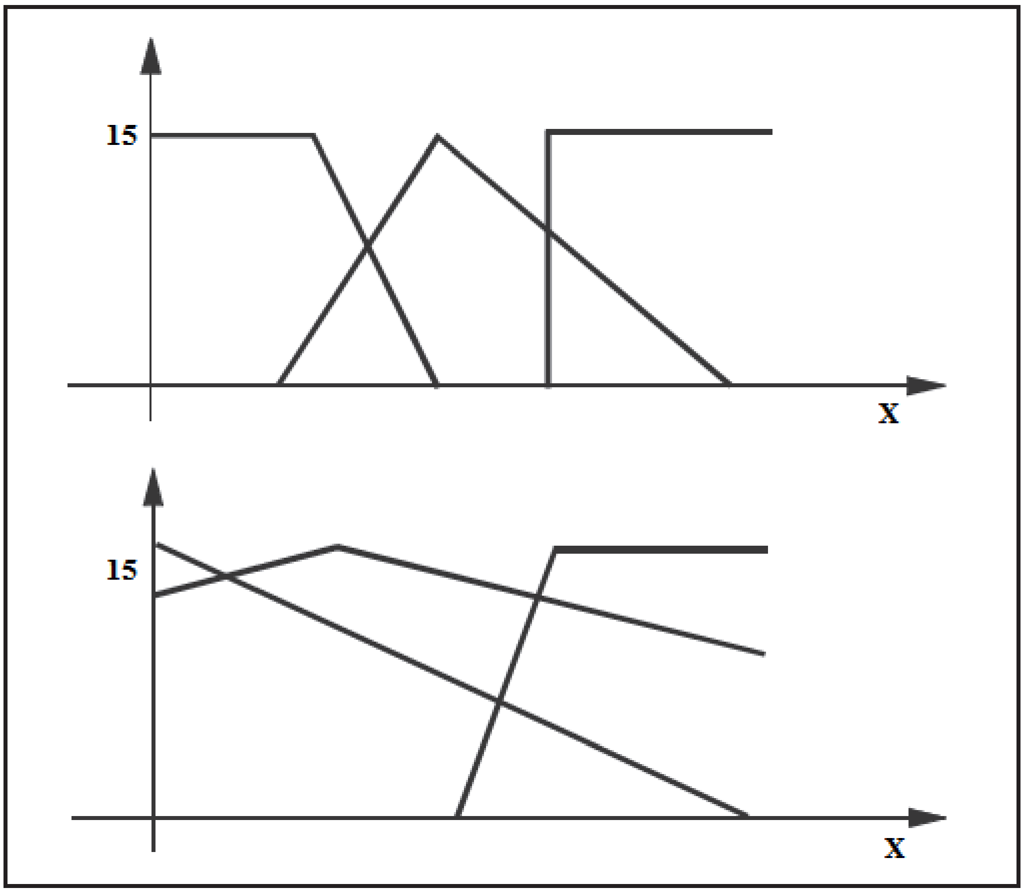
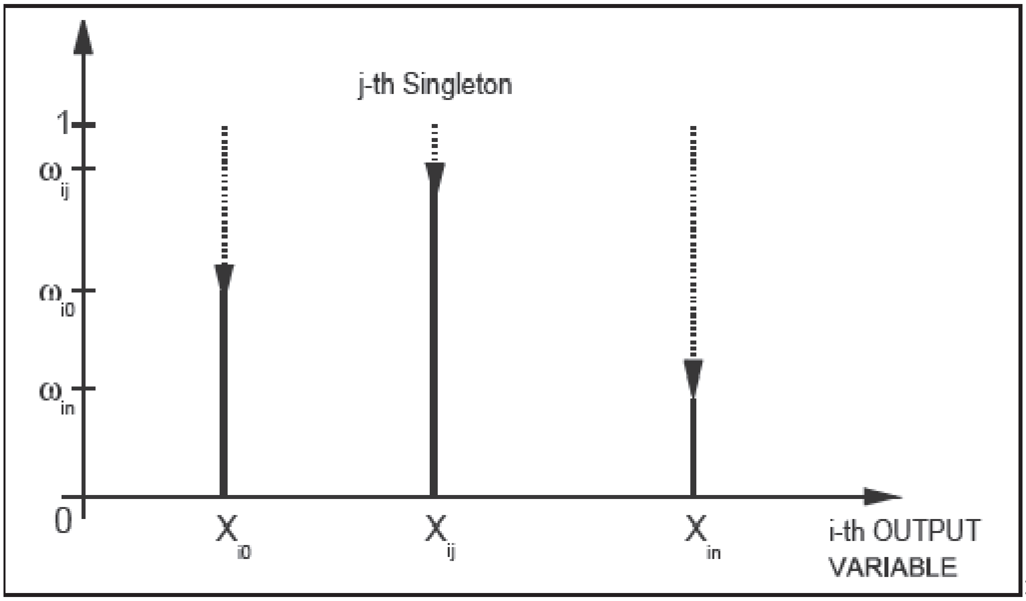


Figure 7- Some examples of valid Mbfs, that can be defined in ST52T410/ST52x420.

Each MBF is then defined storing 3 bytes in the first Kbyte of the Program/Data Memory.
The Mbf is stored by using the following instruction:
MBF n_mbf lvd v rvd
where:
N_mbf is a tag number that identifies the Mbf lvd;
Lvd, v and rvd are the parameters that describe the Mbf"s shape as describe above.
Output Singleton
The Decision Processor uses a particular kind of membership function called Singleton for its output variables. A Singleton doesn't have a shape, like a traditional Mbf, and is characterized by

a single point identified by the couple (X, w), where w is calculated by the Inference Unit as described earlier. Often, a Singleton is simply identified with its Crisp Value X, Fig.8.



:

Figure 8- Output Membership Functions.

Fuzzy Rules
Rules can have the following structures:
If A op B op C...........then Z
If (A op B) op (C op D op E...) ....then Z
where op is one of the possible linguistic operators (AND/OR).
In the first case the rule operators are managed sequentially; in the second one, the priority of the operator is fixed by the brackets.
Each rule is codified by using an instruction set, the inference time for a rule with 4 antecedents and 1 consequent is about 3 microseconds at 20 MHz.
The Assembler Instruction Set used to manage the Fuzzy operations is reported in the table ST52T410/ST52x420.

.

Conclusion

In this paper, a modernized fuzzy logic controller was introduced for DC–DC converter output voltage regulation in MPPT system in PhV station and have implemented on an 8-bit microcontroller. The MFLC 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 MFLC algorithm without any modifications to microcontroller program. This shows that the proposed algorithm is general and can be applied to any DC–DC converter topologies practically.

References

1. E. Koutroulis, F.Blaabjerg, “ A New Technique for Tracking Global Maximum Power Point of PV Arrays Operating Under Partial-Sheding Conditions” IEEE JORNAL OF PHOTOVOLTAICS,vol. 2, no.2, April 2012
2. Трещ, А.М. Система управления фотоэлектрической установкой при автономном использовании / Ю.Н. Петренко, А.М. Трещ // Наука и техника. Международной научно-технической журнал. Минск 2013 №1. С .53-56
3. M. Azab, "A New Maximum Power Point Tracking for Photovoltaic Systems," in WASET.ORG, vol. 34, 2008, pp. 571-574.
4. D. P. Hohm and M. E. Ropp,"Comparative Study of Maximum Power Point Tracking Algorithms", Prog. Photovolt: Res. Appl. 2003; 11:47–62 (DOI:1002/pp.459)
5. T. Esram, P.L. Chapman, "Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques," IEEE Transactions on Energy Conversion, vol. 22, no. 2, pp. 439-449, June 2007.
6. Messaouda AZZOUZI,"Comparaison between MPPT P&O and MPPT Fuzzy Controls in Optimizing the Photovoltaic Generator", International Journal of Advanced Computer Science and Applications, Vol. 3, No. 12, 2012
7. Hairul Nissah Zainudin, "Comparison Study of Maximum Power Point Tracker Techniques for PV Systems",14th International Middle East Power Systems Conference (MEPCON’10), Cairo University, Egypt, December 19-21, 2010
8. D. P. Hohm and M. E. Ropp,"Comparative Study of Maximum Power Point Tracking Algorithms", Prog. Photovolt: Res. Appl. 2003; 11:47–62 (DOI:1002/pp.459)
9. Alejo, D., Maussion, P., & Faucher, J. (2003). Multiple model control of a buck DC/DC converter. Mathematics and Computers in Simulation, 63, pp.249–260.
10. 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), pp. 661–668.
11. 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, pp.1839–1844).
12. Caldero ´n, A. J., Vinagre, B. M., & Feliu, V. (2006). Fractional order control strategies for power electronic buck converters. Signal Processing, 86(10), pp. 2803–2819.
13. Mac Vicar Whelan, P. J. (1977). Fuzzy set for man machine interaction”. International Journal of Man–Machine Studies, 8, pp. 687–697.
14. Bor-Ren Lin, "Analysis of Fuzzy Control Method Applied to DC-DC Converter Control", IEEE Prowe. APK'93, pp.22-28,1993.

15. Rubai, A., Ofoli, A. R., Burge, L., & Garuba, M. (2005). Hardware Implementation of an adaptive network-based fuzzy controller for DC–DC converters. IEEE Transactions on Industry Applications, 41(6),pp. 1557–1565.
16. Elmas,C., Deperlioglu,O., Sayan, H.H. Adaptive fuzzy logic controller for DC-DC converters. Expert Systems with Applications, 36 (2009), pp. 1540-1548
17. 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.
18. Chien, C. L., & Hsieh, C. T. (1996). User-friendly design method for fuzzy logic controller. IEE Pro-Control Theory Application, 143(4), 358–366.
19. Galichet, S., & Foulloy, L. (1995). IEEE fuzzy logic controllers: synthesis and equivalencies. Transactions on Fuzzy Systems, 3(2), pp. 140–148.
20. Huh, S., & Park, G. (1999). An adaptive fuzzy logic controller for power converters. In FUZZY-IEEE '99. (Vol. 1, pp. 434–439).
21. Mattavelli, P., Rossetta, L., Spiazzi, G., & Tenti, P. General-purpose fuzzy logic controller for DC–DC converters. IEEE Transactions on Power Electronics, 12(1), 1997, pp.79–85.
22. Petrenko, Y.N. Fuzzy logic and genetic algorithm technique for non-liner system of overhead crane / Y.N. Petrenko, S.E. Alavi Computational Technologies in Electrical and Electronics Engineering (SIBIRCON), 2010 IEEE Region 8 International Conference , 11-15 July 2010. pp. 848 - 851.
23. Алави, С.Э. Fuzzy logic controller for non-linear system design / С.Э. Алави, Ю.Н. Петренко // Информационные технологии в промышленности: сборник тезисов пяти Международной научно-технической конференции ITI*2008, г. Минск, 22-24 октябрь 2008.– С. 214-215.
24. Alavi, S.E. System control, based on the artificial intelligence techniques / S.E. Alavi, Y.N. Petrenko // Информационные технологии в промышленности: сборник тезисов шестой Международной научно-технической конференции ITI*2010, г. Минск, 28-29 октябрь 2010.– С. 170-171.