Particle Swarm Optimized T-S Fuzzy Logic Controller for Maximum Power Point Tracking in a Photovoltaic System

Lawrence K. Letting *, Josiah L. Munda *
* Department of Electrical Engineering
Tshwane University of Technology
Pretoria, South Africa
E-mail: LettingLK@tut.ac.za, MundaJL@tut.ac.za

Alex Hamam
F'SATI
Tshwane University of Technology
Pretoria, South Africa
Email: HamamA@tut.ac.za

***Abstract*—A particle swarm optimized Takagi-Sugeno (T-S) fuzzy logic controller for maximum power point tracking in a photovoltaic (PV) system is presented. The method proposed automates the tuning of fuzzy logic controller (FLC) rules and membership functions as opposed to the trial-and-error approach. Expert knowledge used for tuning the FLC is extracted from an improved PV module model under varying solar radiation, temperature, and load conditions. The proposed optimized FLC provides fast and accurate tracking of the maximum power point (MPP) under varying operating conditions. The formulation, implementation, and simulation results are presented. Results obtained has shown that the optimized FLC gives a better performance compared with the conventional FLC tuned using trial and error.**

I. Introduction

Photovoltaic (PV) power generation is a reliable and economical source of electricity in rural areas, especially in developing countries where the population has low incomes and the grid power supply is not fully extended due to viability and financial constraints. It is crucial to operate the PV energy conversion systems near the maximum power point (MPP) to increase the efficiency of the PV system. PV module current and power varies non-linearly with the terminal voltage, solar radiation, and temperature. Many MPP tracking strategies have been proposed such as perturb and observe and incremental conductance [1]–[3]. Recently artificial intelligence based methods using genetic algorithms, neural networks, and fuzzy logic have been introduced [4]–[6] in order to improve on the tracking efficiency. Fuzzy logic is appropriate for non-linear control because it does not use complex mathematical equations. The behaviour of a FLC depends on shape of membership functions and the rule base. However, there is no formal method to determine accurately the fuzzy parameters to yield optimum operating point and a good control system depends on the experience of the designer. This paper proposes an automated method for choosing the FLC parameters using particle swarm optimization.

II. Maximum power point tracking

A. Principle of operation

The power produced from a PV module depends on the operating voltage of the load, the solar radiation level, and cell temperature. This is illustrated in Fig.1 and Fig.2. If a variable load resistance R, is connected across the module's terminals, the operating point is determined by the intersection of module I-V curve and the load I-V characteristic as shown in Fig.3. The I-V characteristic consists of two regions: *Zone I* is the current source region, and *Zone II* is the voltage source region. In Zone I, the internal impedance of the module is high, while in Zone II the internal impedance is low. The maximum power point P_{mp} , is located at the knee of the power curve. The power delivered to the load is maximum when the source internal impedance matches the load impedance. The load characteristic is a straight line with a slope of $I/V = 1/R$. If R is small, the module operates in the region AB only and behaves like a constant current source at a value close to module short circuit current, I_{sc}. If R is large, the module operates in the region CD behaving like a constant voltage source, at a value almost equal to module open-circuit voltage, V_{oc}. Maximum power point tracking is based on load line adjustment under varying atmospheric and load conditions by searching for an optimal equivalent output resistance R_{opt} . A dc-dc converter is used to perform load-line adjustment by varying the converter duty cycle using a controller.

B. MPPT control system

The model of a maximum power point tracker (MPPT) was modelled in Matlab/Simulink. The PV module was modeled using model equations given in [7] and implemented as a Simulink S-function whose inputs are the ambient solar radiation G_a, ambient temperature T_a, and the load resistance R. The model outputs are: module operating voltage V_m, output current I_m, voltage at maximum power V_{mp}, and the maximum power P_{mp}. BP solar SX75TU [8] PV module was used to validate the PV module model and for testing the MPPT algorithm. A buck-boost converter was chosen for the MPPT because it is able to perform maximum power tracking in both

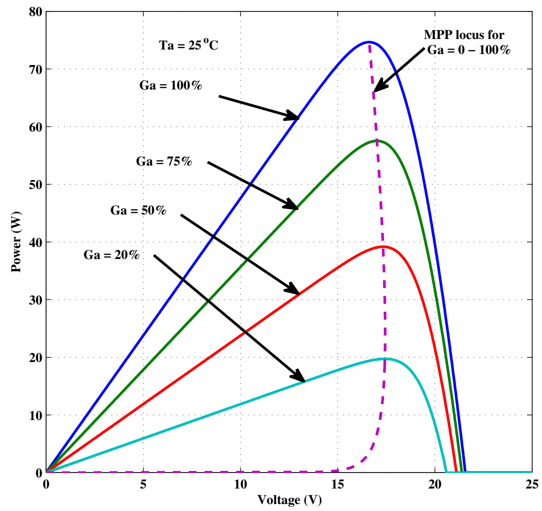

Fig. 1. Effects of ambient solar radiation for constant temperature

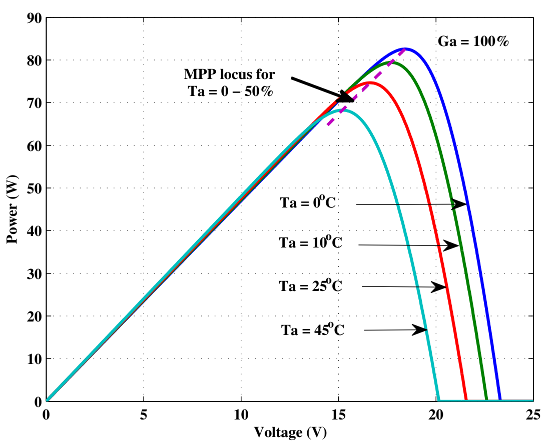

Fig. 2. Effects of ambient temperature for constant solar radiation

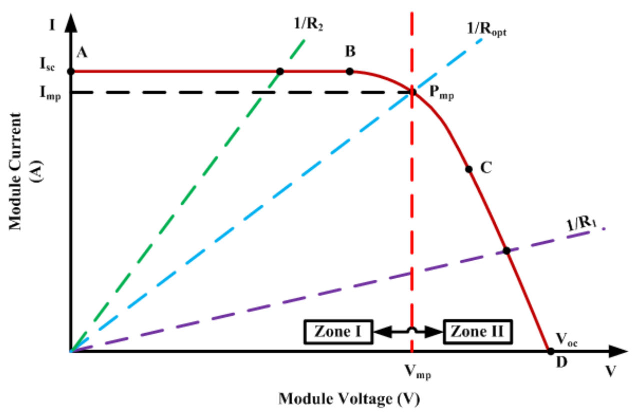

Fig. 3. Tracking the maximum power point by varying load resistance

zones I and II of Fig. 3. An averaged state space model [9] of the converter was developed for simulation. The simulation

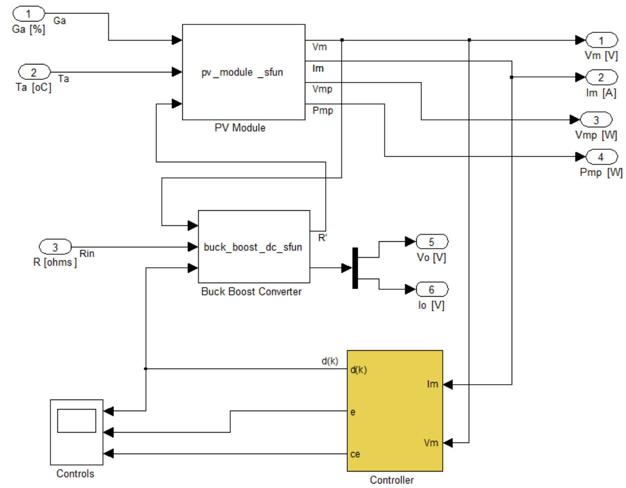

Fig. 4. MPPT Simulink model

model is shown in Fig. 4. A solar radiation level of $1000W/m^2$ is taken as $G_a = 100\%$.

III. Fuzzy Logic Controller Structure

The general structure of a FLC is presented in Fig. 5. It comprises of three principal components: fuzzification, inferencing, and defuzzification. Fuzzification is the conversion of input data into suitable linguistic values such as small, large, etc using a membership function. Rule inferencing is the simulation of human decision making process in order to infer the fuzzy control action from the knowledge of the control rules and the linguistic variable definitions. Defuzzification is the conversion of an inferred fuzzy controller output into a non-fuzzy control action.

There are two input variables, error $E(k)$, and change of error $CE(k)$ at the k_{th} sampling instant defined as:

$$E(k) = \frac{P_m(k) - P_m(k-1)}{V_m(k) - V_m(k-1)} \tag{1}$$

$$CE(k) = E(k) - E(k-1) \tag{2}$$

where $P_m(k)$ is the instantaneous power of the PV module. $E(k)$ is the gradient of the P-V curve of Fig. 1. There are three possible values for $E(k)$: $E(k) > 0$ system is moving towards the MPP ; $E(k) = 0$ system operating at the MPP; $E(k) < 0$ system is moving away from the MPP.

The fuzzy control algorithm was developed on the Matlab fuzzy logic toolbox using Takagi-Sugeno inference system. The Sugeno method is more compact and has a computationally efficient representation than a Mamdani system [10]. The input membership functions for $E(k)$ and $CE(k)$ are as shown in Fig. 6, while the output is the change in duty cycle μD. The inputs are fuzzified using five membership functions: *Negative Big* (NB), *Negative Small* (NS), *Zero* (ZE), *Positive Small* (PS), and *Positive Big* (PB). The FLC has twenty five rules whose output membership functions are defined as:

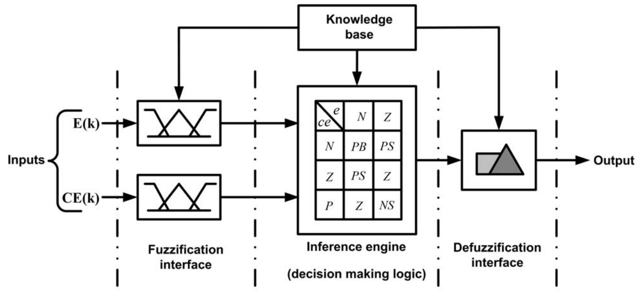

Fig. 5. Basic configuration of a fuzzy logic controller

$$\mu D(k) = aE(k) + bCE(k) + c \tag{3}$$

where a, b, and c are constants.

IV. Fuzzy controller design using particle swarm optimization

The conventional design of the membership functions and the rule base of a fuzzy inference system is based on the experience of the operator or the designer of the system [10]. Various techniques have been proposed for improving the design and performance of fuzzy inference systems. These include simple heuristic methods [11], neuro-fuzzy (ANFIS) [12], [13], genetic algorithms [5], and particle swarm optimization (PSO) [14]–[16]. Before starting the design of the PSO based fuzzy controller, the basic principle of PSO is presented.

A. Principle of particle swarm optimization

Particle swarm optimization is an evolutionary optimization technique based on the movement and intelligence of swarms. It applies the concept of social interaction to problem solving. The algorithm was developed by Kennedy and Eberthart in 1995 for simulating the flight patterns of birds, which is governed by three factors: avoiding collision, matching the velocity, and flock centering [17]. The developers observed that the bird flocking behaviour can be applied in optimization using a population of potential solutions called particles that are flown through an *n*-dimensional search space. The instantaneous position of each particle is identified by the vector

$$X(k) = [X_1(k), X_2(k), \ldots, X_n(k)] \tag{4}$$

where $X_n(k)$ represents a parameter of the problem that has to be optimized. Initially, each particle position is randomly generated and the particle then moves with a random velocity subject to the boundary conditions. At the time step $k+1$, the velocity of i_{th} particle is given by

$$\begin{aligned} V_i(k+1) \quad &= wV_i(k) + C_1 rand() \tfrac{\left(P_{i(best)}(k) - X_i(k)\right)}{\Delta t} + \\ &\quad C_2 rand() \tfrac{(G_{best}(k) - X_i(k))}{\Delta t} \end{aligned} \tag{5}$$

where w is the inertia factor; C_1 and C_2 are the social and cognitive rate, respectively; $P_{i(best)}$ is the best position ever found by the particle during its motion; G_{best} is the best position discovered by the entire swarm; *rand()* is a uniform random number generator between 0 and 1 ; and Δt is the time step. The new position of each particle is then determined by,

$$X_i(k+1) = X_i(k) + V_i(k)\Delta t \tag{6}$$

If a particle position violates the side constraints, the current velocity is set to zero and a new velocity is evaluated as,

$$\begin{aligned} V_i(k) \quad &= C_1 rand() \tfrac{\left(P_{i(best)}(k) - X_i(k)\right)}{\Delta t} + \\ &\quad C_2 rand() \tfrac{(G_{best}(k) - X_i(k))}{\Delta t} \end{aligned} \tag{7}$$

and a new position is re-evaluated using (6).

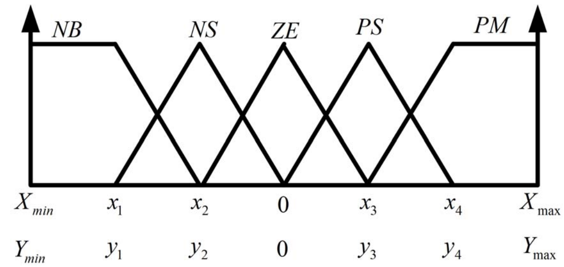

Fig. 6. Particles of the input membership function

TABLE I
FLC rule base

E	***CE***	NB	NS	ZE	PS	PB
NB		MF_1	MF_2	MF_3	MF_4	MF_5
NS		MF_6	MF_7	MF_8	MF_9	MF_{10}
ZE		MF_{11}	MF_{12}	MF_{13}	MF_{14}	MF_{15}
PS		MF_{16}	MF_{17}	MF_{18}	MF_{19}	MF_{20}
PB		MF_{21}	MF_{22}	MF_{23}	MF_{24}	MF_{25}

B. Structure of the particles

Particle swarms are used to determine the optimal input and output membership functions. The particles for the input membership functions (MFs) for error, E, and change of error, CE, are assigned as shown in Fig. 6, where X_i and Y_i corresponds to E, and CE respectively. The output MF particles are given by $MF_j = [z_{ja}, z_{jb}, z_{jc}]$ for each of the output MFs in Table I. For a swarm S_i in a population U having N members (where $i \in [1, N]$), the particles are arranged as,

$$S_i = [X_i, Y_i, Z_i] \tag{8}$$

where,

$$X_i = [x_1, x_2, x_3, x_4] \tag{9}$$

$$Y_i = [y_1, y_2, y_3, y_4] \tag{10}$$

$$Z_i = [z_{i,1a}, z_{i,1b}, z_{i,1c}, \ldots, z_{i,25a}, z_{i,25b}, z_{i,25c}] \tag{11}$$

Thus the swarm S_i consists of 83 particles encoded as real numbers with the boundary conditions shown in Table II. The values were derived by observing the parameter variations during simulation. The velocity of the particles V_i has the same dimensions as S_i and also encoded as real numbers.

TABLE II
BOUNDARY CONDITIONS

Particle	Minimum	Maximum
x_i	-50	50
y_i	-10	10
z_a	-0.5	0.5
z_b	-0.05	0.05
z_c	-0.01	0.01

TABLE III
INITIALIZATION OF PSO PARAMETERS

Parameter	Value
Population (N)	30
Number of iterations (I_{max})	50
Inertia factor (w)	0.5
Social rate (C_1)	0.9
Cognitive rate (C_2)	2.5

C. Initialization of the algorithm

The size of the population N, the maximum number of iterations I_{max}, the inertia factor w, social rate C_1, and the cognitive rate C_2, were initialized as shown in Table III. The initial velocity and position of each particle was randomly generated.

D. Optimization criterion

The goal of maximum power tracking is to minimize the error between the PV module output power P_m, and the maximum possible power, P_{max}. The fitness of each swarm is evaluated using (12).

$$f(S_i) = \sqrt{(P_{max} - P_m)^2} \tag{12}$$

E. Simulation steps

The tuning of the FLC parameters can be summarized as follows:

- Step 1: Encode the MF particles for inputs and outputs as presented in Fig.6 and Table I.
- Step 2: Initialize the swarm position vector S_i and corresponding velocity V_i for all swarms in the population. All particles are randomly generated (i.e no experienced particles are used).
- Step 3: Update the velocity of each particle in the swarms using (5), and the position using (6).
- Step 4: Decode each swarm (S_i) into a fuzzy inference system (FIS) structure and output the results to the MPPT controller. The fitness of each FIS structure is then evaluated and used to update P_{best} and G_{best}.
- Step 5: Repeat Steps 3 and 4 for the set number of iterations.

F. Optimal FLC structure

The system converges to an optimal solution which is the FIS structure obtained from the best swarm at the end of the simulation. The input MFs of the best solution before and after simulation are presented in Fig. 7 and Fig. 8. The rule surface before and after optimization are as shown in Fig. 9 and Fig. 10 respectively.

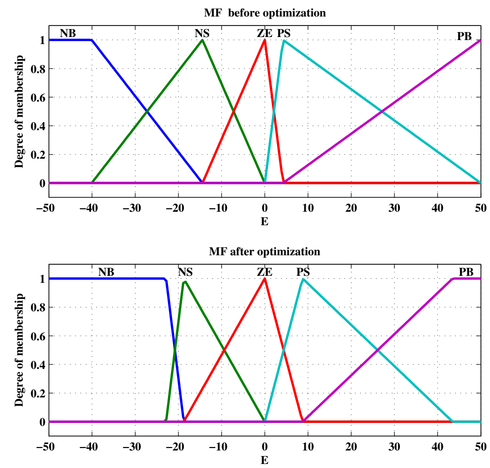

Fig. 7. Membership functions for error E before and after optimization

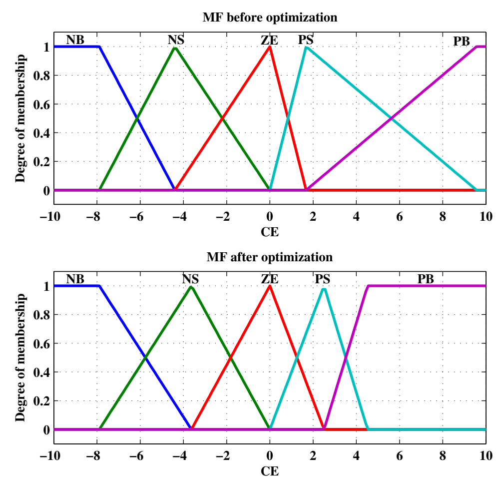

Fig. 8. Membership functions for change of error CE before and after optimization

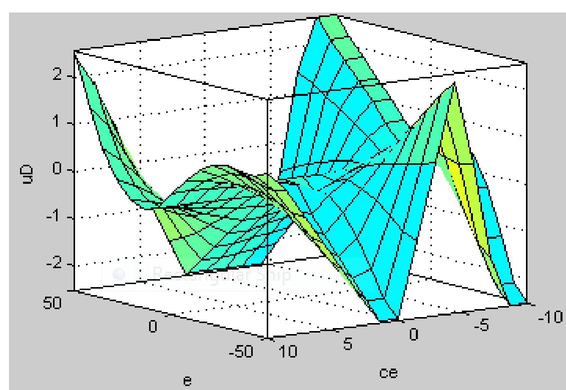

Fig. 9. Rule surface before optimization

V. SIMULATION RESULTS

The performance of the particle swarm optimized fuzzy controller (OFLC) was validated using simulations under dif-

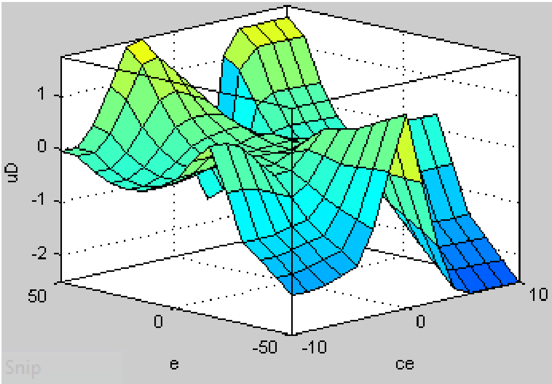

Fig. 10. Rule surface after optimization

ferent PV module operating conditions using the Simulink model presented in Fig. 4. The conventional FLC and the OFLC were simulated under the following test conditions:

- Constant temperature of 25^oC and constant solar radiation of $1000W/m^2$ with 90% step change in load
- Constant temperature of 25^oC and constant load with 50% fast change in solar radiation
- Constant solar radiation of $1000W/m^2$ and constant load with 50% fast change in temperature

The simulation results are presented in Figs. 11-16. The results show that an optimized fuzzy logic controller has improved performance and is more robust than the conventional controller.

VI. Conclusion

In this work, the optimization of Takagi-Sugeno based FLC for maximum power point tracking in a PV system is presented. An improved PV module model that outputs both the operating power and the optimal power was first developed. A complete MPPT model was then implemented in Matlab/Simulink and used for tuning of the FLC parameters using the intelligence of particle swarms. Simulation results has shown that the proposed optimized FLC is robust and provides fast and accurate tracking of the maximum power point compared to the conventional FLC.

References

[1] E. Koutroulis, K. Kalaitzakis, and N. Voulgaris, "Development of a microcontroller-based, photovoltaic maximum power point tracking control system," *Power Electronics, IEEE Transactions on*, vol. 16, no. 1, pp. 46 –54, jan 2001.

[2] M. Bodur and M. Ermis, "Maximum power point tracking for low power photovoltaic solar panels," in *Electrotechnical Conference, 1994. Proceedings., 7th Mediterranean*, 12-14 1994, pp. 758 –761 vol.2.

[3] J. Enslin and D. Snyman, "Simplified feed-forward control of the maximum power point in pv installations," in *Industrial Electronics, Control, Instrumentation, and Automation, 1992. Power Electronics and Motion Control., Proceedings of the 1992 International Conference on*, 9-13 1992, pp. 548 –553 vol.1.

[4] M. Veerachary, T. Senjyu, and K. Uezato, "Feedforward maximum power point tracking of pv systems using fuzzy controller," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 38, no. 3, pp. 969 – 981, jul 2002.

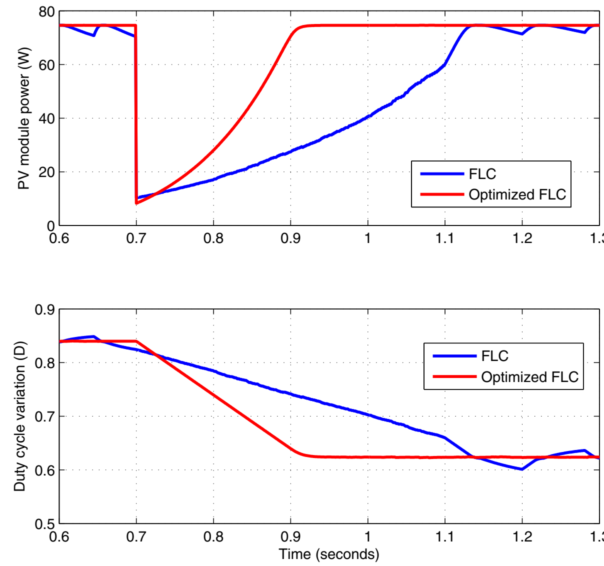

Fig. 11. FLC and OFLC responses with step change in load from 90Ω to 10Ω

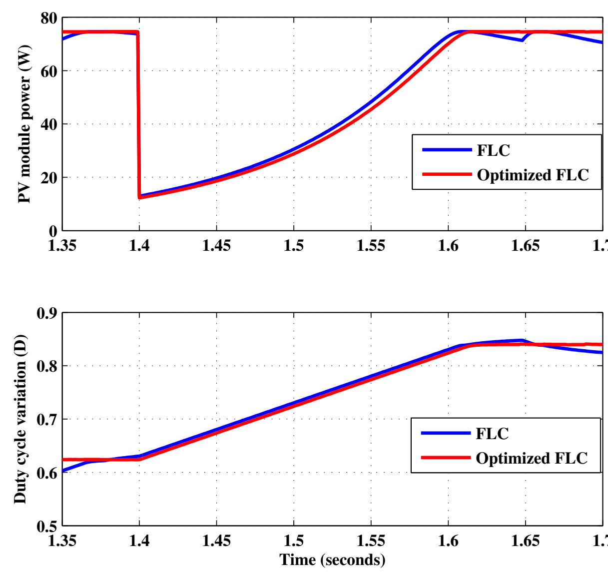

Fig. 12. FLC and OFLC responses with step change in load from 10Ω to 90Ω

[5] C. Larbes, S. A. Cheikh, T. Obeidi, and A. Zerguerras, "Genetic algorithms optimized fuzzy logic control for the maximum power point tracking in photovoltaic system," *Renewable Energy*, vol. 34, no. 10, pp. 2093 – 2100, 2009.

[6] C.-C. Chen and C.-C. Wong, "Self-generating rule-mapping fuzzy controller design using a genetic algorithm," *Control Theory and Applications, IEE Proceedings -*, vol. 149, no. 2, pp. 143 –148, mar 2002.

[7] J. F. Manwell, A. Rogers, G. Hayman, C. T. Avelar, J. G. McGowan, U. Abdulwahid, and K. Wu, *Hybrid2 - A hybrid system simulation model theory manual*, 2006. [Online]. Available: http://ceere.org/rerl/projects/software/hybrid2/Hy2_theory_manual.pdf

[8] ecocalc.com, *BP SX 75 TU: technical sheet.* [Online].

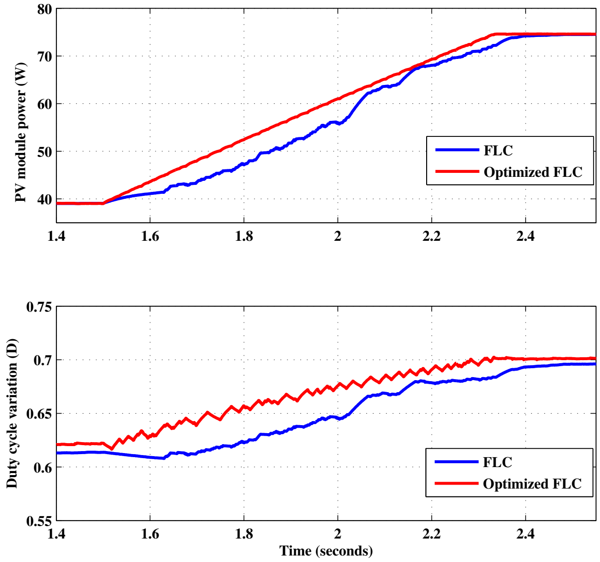

Fig. 13. FLC and OFLC responses with fast change in solar radiation from 50% to 100%

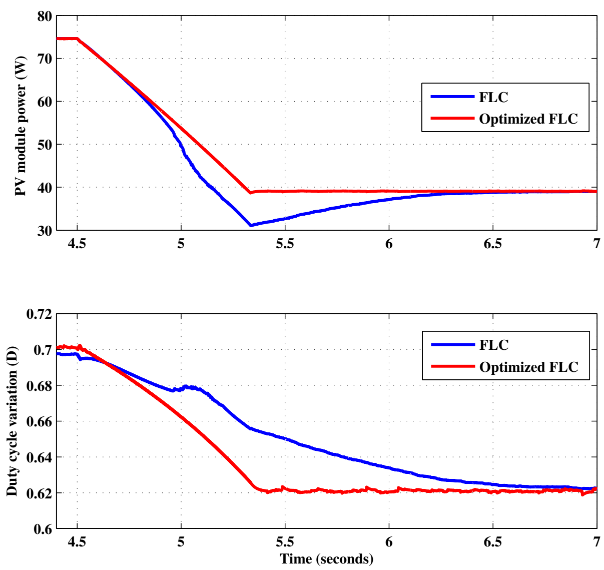

Fig. 14. FLC and OFLC responses with fast change in solar radiation from 100% to 50%

Available: http://www.ecocalc.com/manufacturer_pv/790/BP+Solar/BP+SX+75+TU?ep=1&prid=
[9] R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics, Second Edition*. Kluwer Academic Publishers, 2001.
[10] B. K. Bose, *Modern Power Electronics and AC Drives*. Prentice-Hall, Inc., 2002.
[11] S. Abe and M. S. Lan, "A method for fuzzy rules extraction directly from numerical data and its application to pattern classification," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 1, p. 1828, 1995.
[12] S. Mitra and Y. Hayashi, "Neuro-fuzzy rule generation: Survey in soft computing framework," *IEEE Trans. on Neural Netw.*, vol. 11, no. 3, p. 748768, 2000.

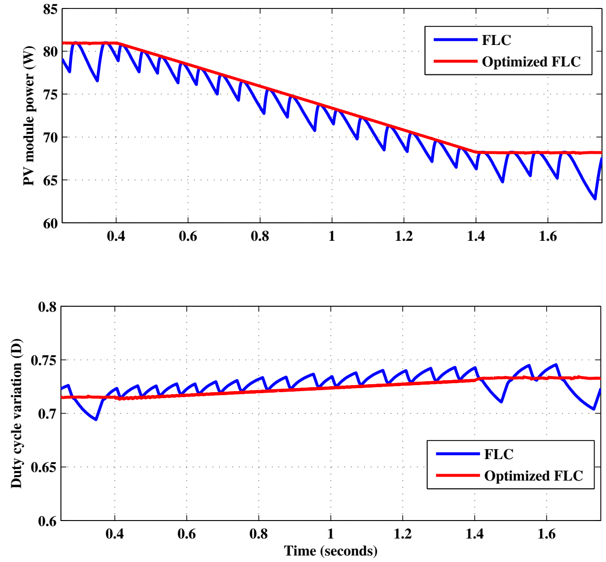

Fig. 15. FLC and OFLC responses with fast change in temperature from 5^oC to 45^oC

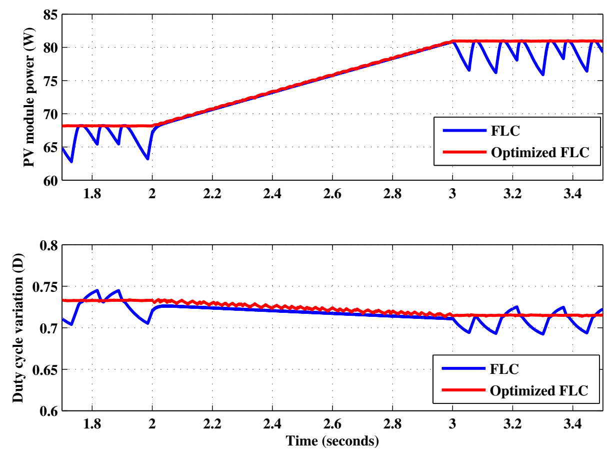

Fig. 16. FLC and OFLC responses with fast change in temperature from 45^oC to 5^oC

[13] D. Chakraborty and N. R. Pal, "A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, p. 110123, 2004.
[14] Z. Chenghui, S. Qingsheng, C. Naxin, and L. Wuhua, "Particle swarm optimization for energy management fuzzy controller design in dual-source electric vehicle," in *Power Electronics Specialists Conference, 2007. PESC 2007. IEEE*, 17-21 2007, pp. 1405 –1410.
[15] A. Khosla, S. Kumar, K. K. Aggarwal, and J. Singh, *Particle Swarm for Fuzzy Models Identification*. Springer Berlin / Heidelberg, 2006, vol. 26/2006, pp. 149–173.
[16] L. Wang, Q. Kang, F. Qiao, and Q. Wu, "Fuzzy logic based multi-optimum programming in particle swarm optimization," in *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE*, 19-22 2005, pp. 473 – 477.
[17] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, nov/dec 1995, pp. 1942 –1948 vol.4.