International Journal of Electrical and Computer Engineering (IJECE)
Vol. 7, No. 6, December 2017, pp. 3454~3466
ISSN: 2088-8708, DOI: 10.11591/ijece.v7i6.pp3454-3466

VHDL Based Maximum Power Point Tracking of Photovoltaic Using Fuzzy Logic Control

Doaa M. Atia[1], Hanaa T. El-madany[2]
Department of Photovoltaic Cells, Electronics Research Institute

Article Info

Article history:

Received Oct 18, 2016
Revised May 30, 2017
Accepted Jun 15, 2017

Keyword:

Maximum power point tracking (MPPT)
Photovoltaic (PV)
Fuzzy logic controller (FLC)
Very high speed integrated Circuit hardware description language (VHDL)

ABSTRACT

It is important to have an efficient maximum power point tracking (MPPT) technique to increase the photovoltaic (PV) generation system output efficiency. This paper presents a design of MPPT techniques for PV module to increase its efficiency. Perturb and Observe method (P&O), incremental conductance method (IC), and Fuzzy logic controller (FLC) techniques are designed to be used for MPPT. Also FLC is built using MATLAB/ SIMULINK and compared with the FLC toolbox existed in the MATLAB library. FLC does not need knowledge of the exact model of the system so it is easy to implement. A comparison between different techniques shows the effectiveness of the fuzzy logic controller techniques. Finally, the proposed FLC is built in very high speed integrated circuit description language (VHDL). The simulation results obtained with ISE Design Suite 14.6 software show a satisfactory performance with a good agreement compared to obtained values from MATLAB/SIMULINK. The good tracking efficiency and rapid response to environmental parameters changes are adopted by the simulation results.

Corresponding Author:

Hanaa T. El-madany,
Departement of Photovoltaic cells,
Electronics Research Institute,
ElTahrir Str., Dokki, Giza, Egypt.
Email: hanaa_tolba@eri.sci.eg

1. INTRODUCTION

At present worldwide have an energy crisis so that renewable energy is a necessary solution to conventional resources problems [1]. Photovoltaic (PV) energy is highly recommended in electrical power applications. It is crucial to operate the PV systems near the maximum power point to increase the PV system efficiency.However, the nonlinear nature of PVarray depends on the array terminal operating voltage. Therefore, the tracking control of the maximum power point is a complicated problem. To overcome these problems, many conventional tracking control strategies have been proposed such as perturb and observe [2],[3], incremental conductance [4], parasitic capacitance [5], constant voltage [5], the hill climbing strategy [6]. Intelligent MPPT controller techniques, such as neural network (NN) and fuzzy logic (FL) methods, have been developed [7]-[18].

Using Fuzzy logic methods for maximum power point determination of a solar array has a good stability and a high response rate. Nowadays, fuzzy based [6],[7] researches have been published due toits good performance andhigh accuracy. Field programmable gate arrays (FPGAs) are digital integrated circuits that contain several millions of programmable logic blocks connected together with configurable interconnections. FPGA is used for hardware implementation for many applications especially FLC because of the required high speed and parallel processing of fuzzy logic applications [19],[20].

The paper is organized as follows. In section 2, the stand-alone PV system is presented which describes the PV system and the DC–DC boost converter mathematical modeling and design. Then, proposed

MPPT control strategies are reported in section 3 which explains the perturb and observe, incremental conductance, and fuzzy logic control approach. The simulation results and discussion are illustrated in section 4 and section 5 respectively. Finally, Section 6 contains a conclusion of this work.

2. MPPT IN STAND-ALONE PV SYSTEMS

The PV stand-alone system comprises of PV module, MPPT control unit, DC/DC converter and the required load as shown in Figure 1. The PV module under test has 75W as a maximum power, 4.4A as a maximum current and 17.7V as a maximum voltage. The DC/DC boost converter is used where the proposed system operates at 48 DC voltage. Perturbation and observation, Incremental conductance, and Fuzzy logic control method are known as MPPT algorithms for PV systems which are widely used.

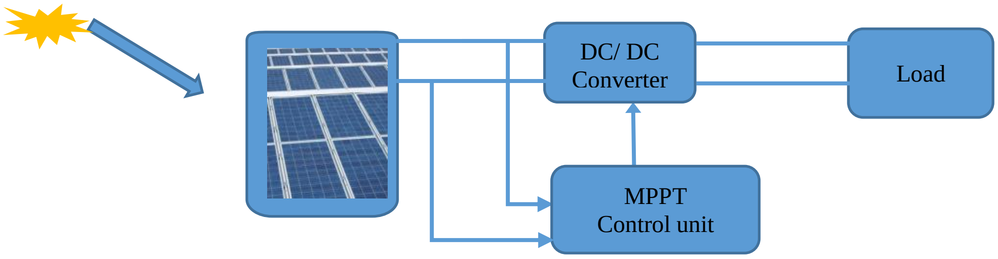


Figure 1. PV stand-alone system scheme

2.1. Photovoltaic generator model

A solar cell converts energy in the sunlight photons into electricity. PV modules consist of group of PV cells connected in series. PVmodules are interconnected in a series-parallel configuration to form PV arrays. The equation that descripes the I-V charateristics of PV array can be expressed as follws [21]:

$$I_{PV} = I_L - I_o\left(\exp\frac{(V+IR_S)}{V_t} - 1\right) \tag{1}$$

where I_{PV} is the output current of PV (A), I_L is the light generated current (A) , I_o is the revese saturation current at operating temperature (A),V is the output voltage of PV (V) and V_t is the thermal voltage. The solar insolation variation effect on PV module is studied in Figure 2.

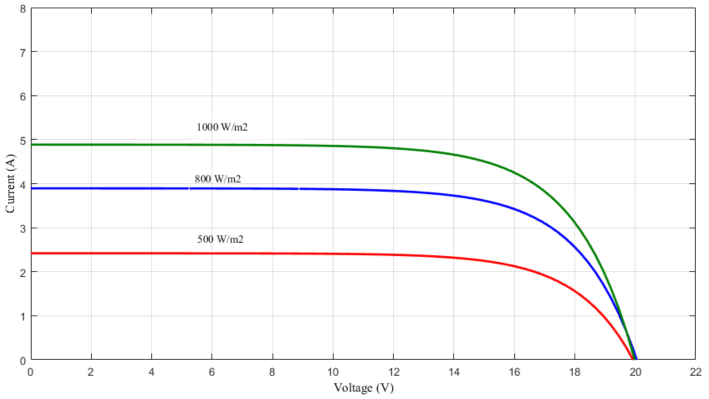


Figure 2. IV characteristic curve of PV module

2.2. Boost Converter

For PV systems, DC-DC converter is used to regulate DC voltage. The regulation is normally achieved by pulse width modulation technique using MOSFET or IGBT as a switching element. By adjusting the PWM duty cycle, maximum power will be changed and the maximum power point is reached. The mathematical modelling of the converetr in a continuous conduction mode is given by [22]:

$$\frac{V_o}{V_i} = \frac{1}{1-D} \tag{2}$$

$$L = \frac{V_i \; D}{f_s \; CRF} \tag{3}$$

$$C = \frac{I_O \, D}{f_s \; VRF} \tag{4}$$

where *CFR* is the current ripple factor, *VRF* is the voltage ripple factor, $f_s$is the switching frequency, and *D* is the duty cycle. By designing of proposed DC-DC boost converter under the maximum power 75W, the value of inductor and capacitor are 5 mH and 100 µF respectively.

3. MPPT CONTROL STRATEGY

There are many MPPT algorithms have been developed and implemented by researchers [2]-[18]. The MPPT algorithms used in this work are Perturb and Observe method (P&O), incremental conductance method (IC), and fuzzy logic method. These methods will be discussed below in detail.

3.1. MPPT control with perturb and observe method

The main concept of perturb and observe method is to drive the system to operate in the direction of increasing PV output power. If the output power increases, then the perturbation of the voltage is made in the same direction otherwise perturbation against the original direction should be made. Figure 3 shows the flowchart of P&O technique. The following equation describes the strategy of the P&O technique [2].

$$if \quad \frac{dP}{dV} > 0; \qquad D = D_i + \mathrm{dD} \tag{5}$$

$$if \quad \frac{dP}{dV} < 0; \qquad D = D_i - \mathrm{dD} \tag{6}$$

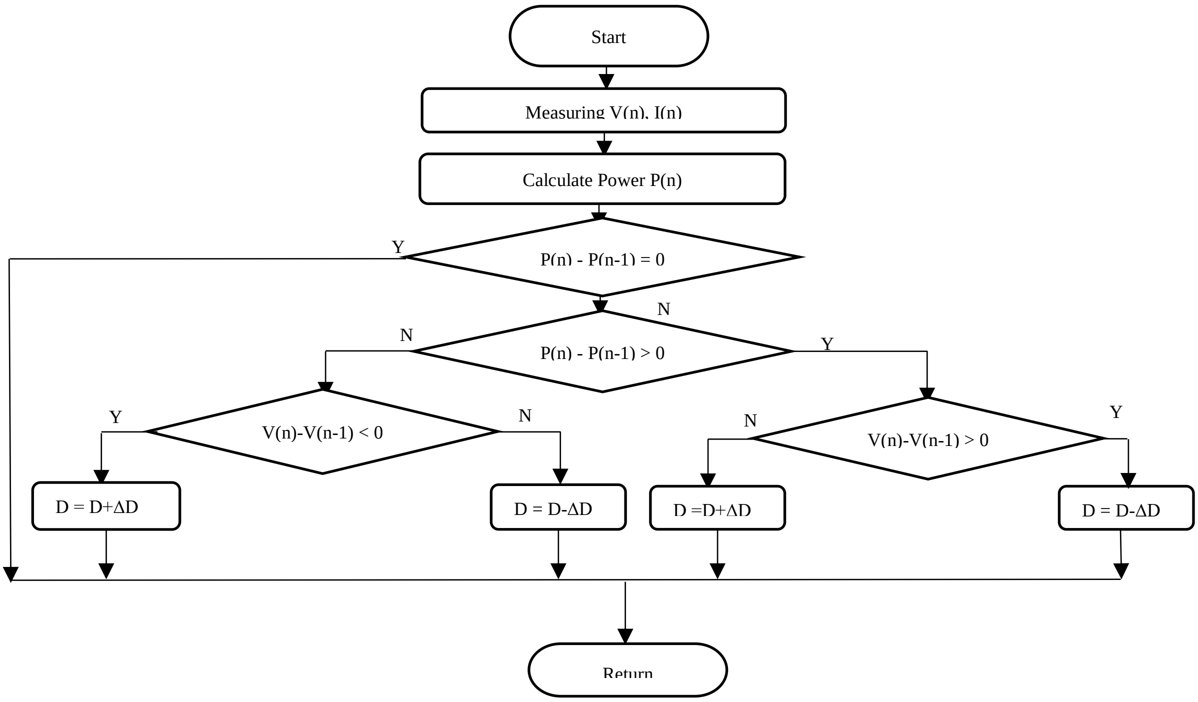


Figure 3. Flowchart of P&O algorithm

3.2. MPPT control with incremental conductance

The incremental conductance MPPT algorithm flowchart is shown in Figure 4. The incremental conductance algorithm as cleared in Eq. (7) is based on the PV power differentiation with respect to PV voltage and setting the result equal to zero as given below [4]:

$$\frac{dP}{dV} = 0; \qquad at \text{ MPP} \tag{7}$$

$$\frac{dI}{dV} = -\frac{I}{V}; \qquad at \text{ MPP} \tag{8}$$

$$if \quad \left(\frac{dI}{dV} > -\frac{I}{V}\right); \qquad D = D_i + \mathrm{dD} \tag{9}$$

$$if \left(\frac{\mathrm{dI}}{\mathrm{dV}} < -\frac{\mathrm{I}}{\mathrm{V}}\right); \qquad \mathrm{D} = \mathrm{D_i} - \mathrm{dD} \tag{10}$$

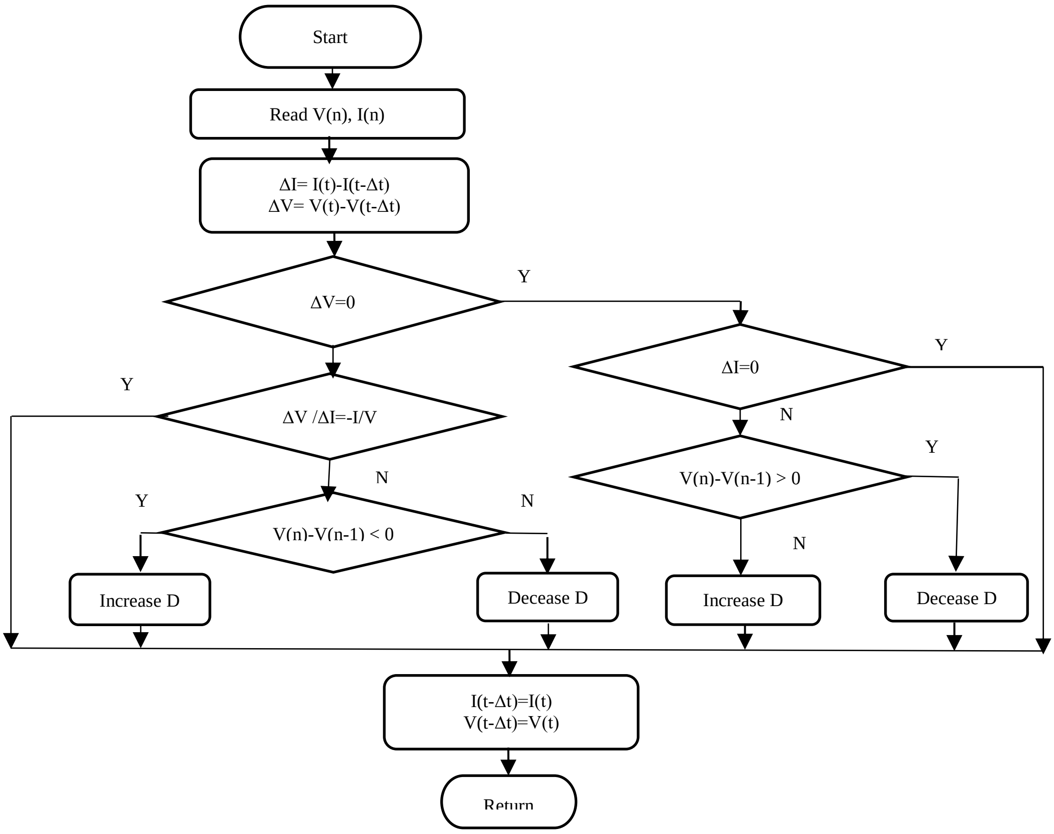


Figure 4. Flowchart of IC algorithm

3.3. MPPT control with fuzzy logic

Fuzzy logic based MPPT control contains mainly from three steps which are fuzzification, rule-base, inference and defuzzification. Two simulation methods are proposed to build FLC in MATLAB/SIMULINK. The first one focused on FLC toolbox in MATLAB library; on the other hand, the mathematical modeling of FL technique is built using MATLAB/SIMULINK. The proposed FL MPPT control has two inputs and one output. The two inputs of FLC are the error *E (k)* and change of error *CE (k)* as defined below [18]:

$$E(k) = \frac{P(k)-P(k-1)}{V(k)-V(k-1)} \tag{11}$$

$$CE(k) = E(k) - E(k-1) \tag{12}$$

The FLC input variables are represented by seven fuzzy sets; Negative Large (NL), Negative Medium (NM), Negative Small (NS), zero (Z), Positive Small (PS), Positive Medium (PM) and Positive Large (PL). The rule base is presented in Table 1. Figure 5 gives the membership function of FLC input and output.

Table 1. Rule base of fuzzy logic controller

		Change of error (Ce)						
		NL	NM	NS	ZE	PS	PM	PL
Error (e)	NL	NL	NL	NL	NL	NM	NS	ZE
	NM	NL	NL	NL	NM	NS	ZE	PS
	NS	NL	NL	NM	NS	ZE	PS	PM
	ZE	NL	NM	NS	ZE	PS	PM	PL
	PS	NM	NS	ZE	PS	PM	PL	PL
	PM	NS	ZE	PS	PM	PL	PL	PL
	PL	ZE	PS	PM	PL	PL	PL	PL

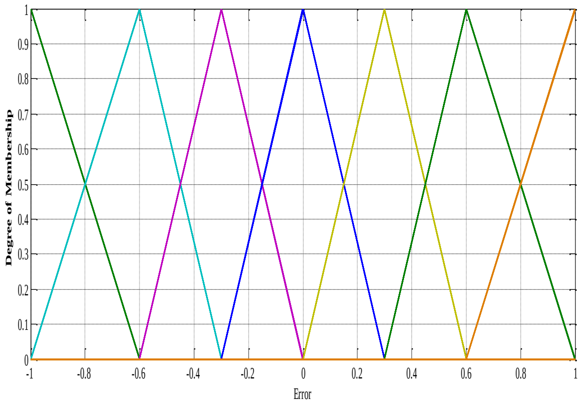


Figure 5. Membership function for fuzzy inputs and output

The mathematical modeling of triangle membership function is given by [7]:

$$\mu M(x) = max(min\left(\frac{x-x_1}{x_T-x_1}, \frac{x_2-x_1}{x_2-x_T}\right), 0) \tag{13}$$

The fuzzification step consists of seven membership tringle function connected together and min/ max opereator is used to achieve this step. Due to the seven membership tringle function there are 49 rules in total. The defuzzification step used the center of the area method. The mathematical expression of this method is shown by Eq.14 [18].

$$z = \frac{\sum_{j=1}^{n} \mu(z_j).\, z_j}{\sum_{j=1}^{n} \mu(z_j)} \tag{14}$$

4. SIMULATION RESULTS

In order to show the effectiveness of the MPPT, a comparison between the different control techniques has been carried out. The value of solar radiation variation levels which used for different controllers is the same. Figure 6 shows the total system design using MATLAB SIMULINK. The system consists of PV unit, P&O, IC and FLC MPPT unit using toolbox, FLC subsystem using Simulink and boost converter unit. The PV power and voltage are the input signals to MPPT unit and duty cycle is the output signal. The detailed block diagrams of control subsystem using different MPPT control technique such as P&O, IC and FLC MPPT unit using toolbox, FLC subsystem using Simulink are shown in Figure 7 to Figure 10 respectively. Figure 11 clarify the detailed model of the fuzzy subsystem. The Simulink model of tringle membership function is shown in Figure 12. The detailed simulink model of the fuzzification stage is presented in Figure 13. It consists of seven membership tringle functions connected together and min/ max opereator is used to achieve the fuzzification step. Due to the seven membership tringle function there are 49 rule base in total as seen in Figure 14.

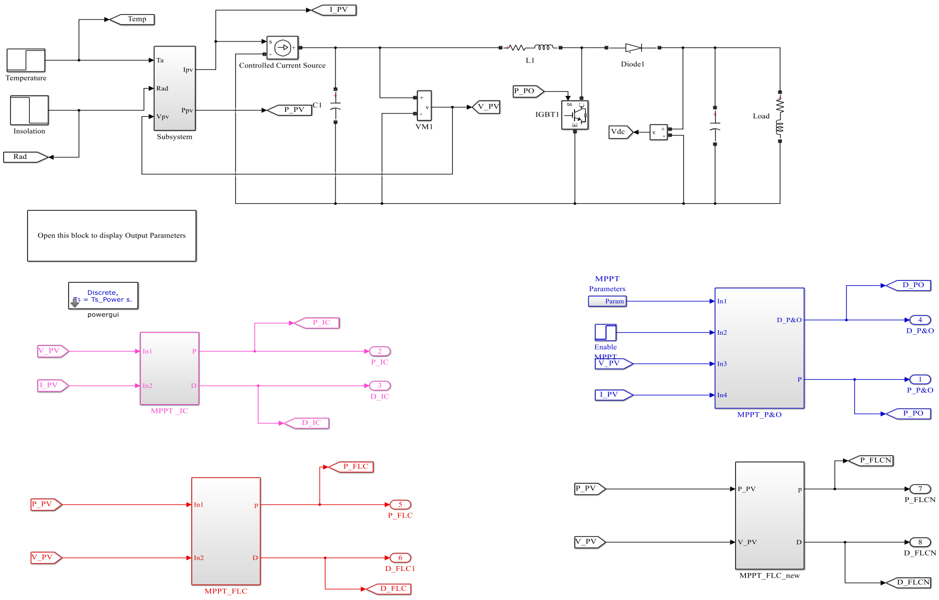


Figure 6. Total system SIMULINK

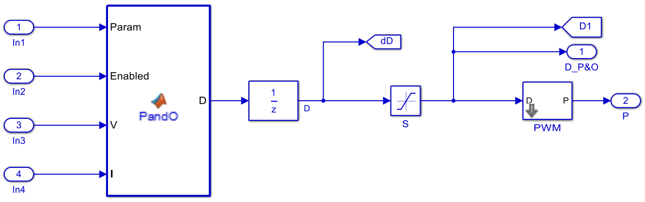


Figure 7. Control subsystem using P&O

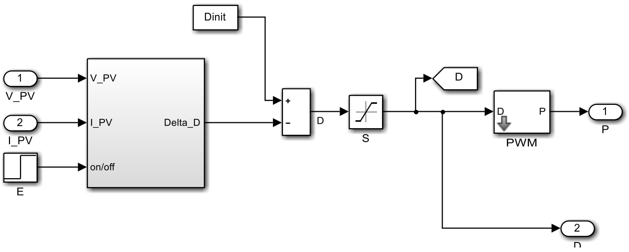


Figure 8. Control subsystem using IC

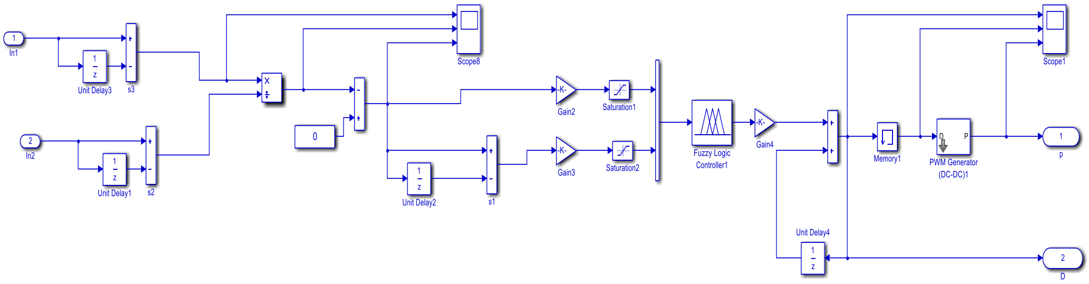


Figure 9. Control subsystem using FLC toolbox

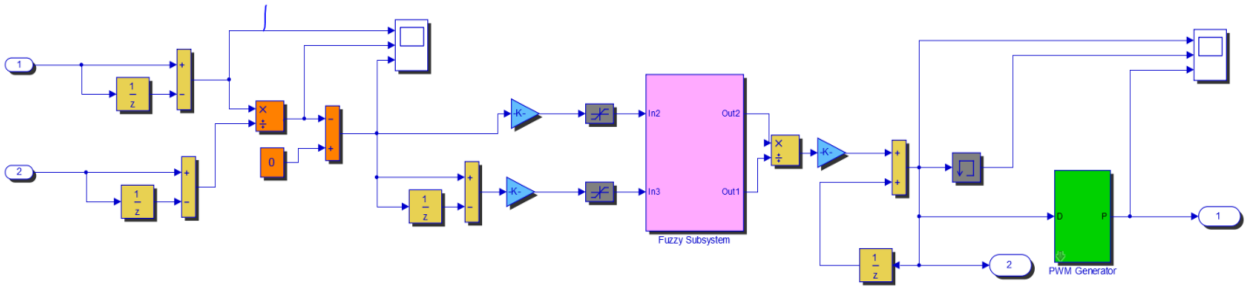


Figure 10. Control subsystem using FLC

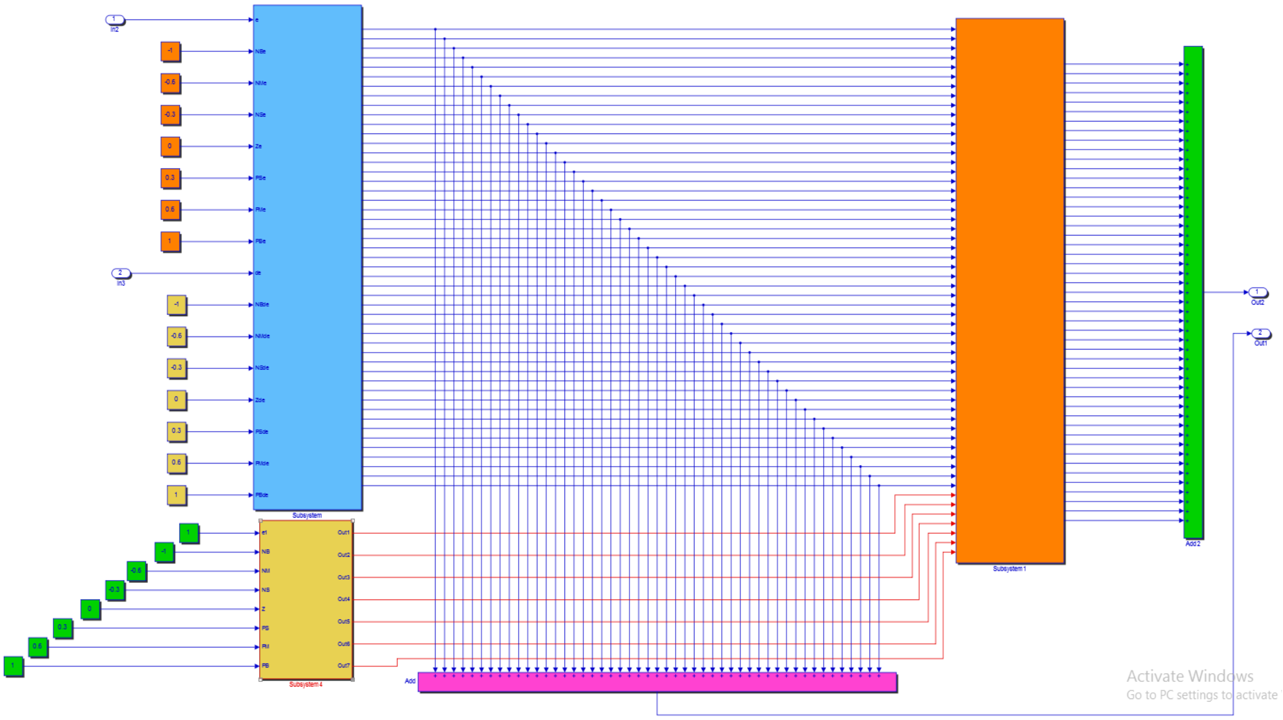


Figure 11. Fuzzy subsystem implementation in MATLAB

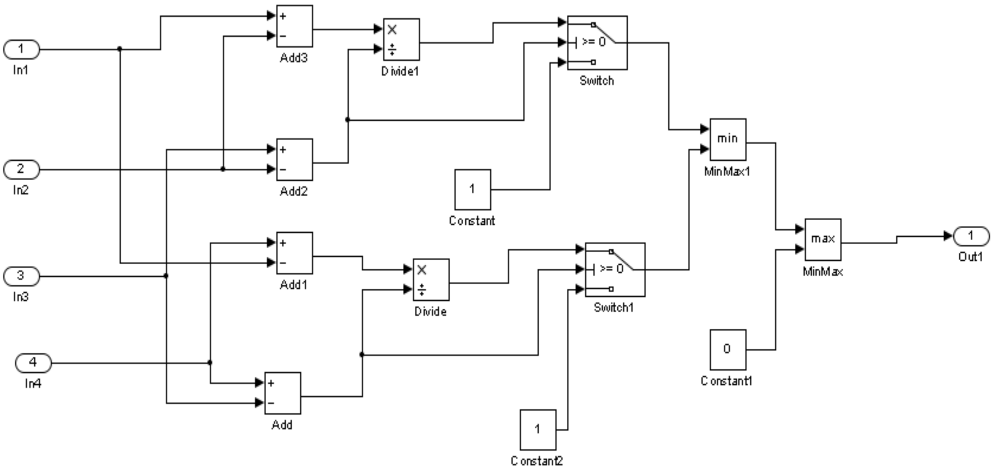


Figure 12. Tringle Membership function SIMULINK

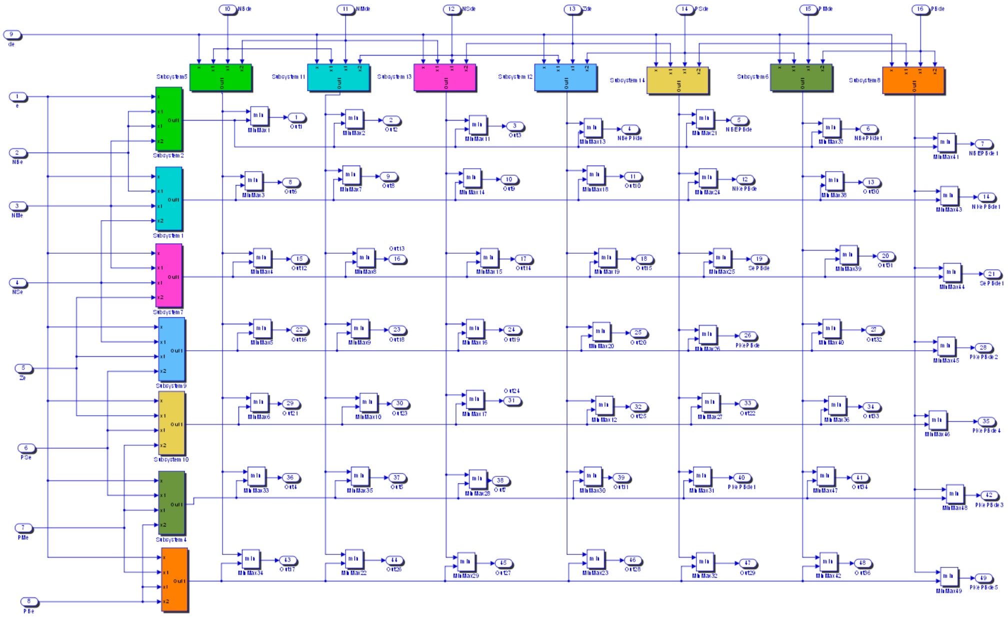


Figure 13. Fuzzification subsystem

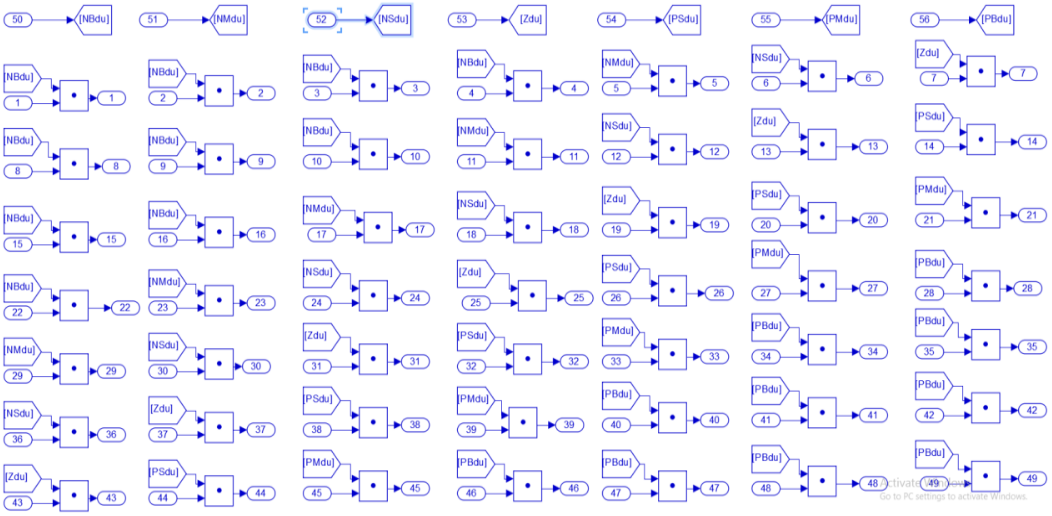


Figure 14. Rule base simulation using MATLAB

5. RESULTS AND DISCUSSION

The control unit is tested at variable solar radiation and constant air temperature degree as given in Figure 15. Simulation is carried out by giving an irradiation changes from 500 W/m^2 to 1000 W/m^2 and constant temperature at 25°C. The different MPPT techniques are tested to check its ability to track the PV maximum power. Figure 16 gives the PV power, current and voltage waveforms simulated using P&O algorithm. The performance of IC technique is cleared in Figure 17. The two different FLC techniques give the output power of PV module equal to 74.4 W at 1000 W/m^2 and 34 W at 500 W/m^2 as shown in Figure 18 and Figure 19.

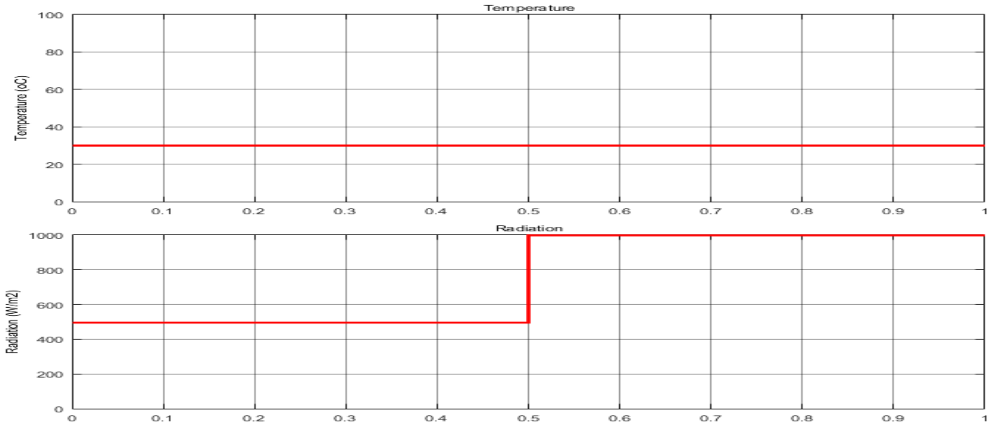


Figure 15. Variable solar radiation and constant air temperature

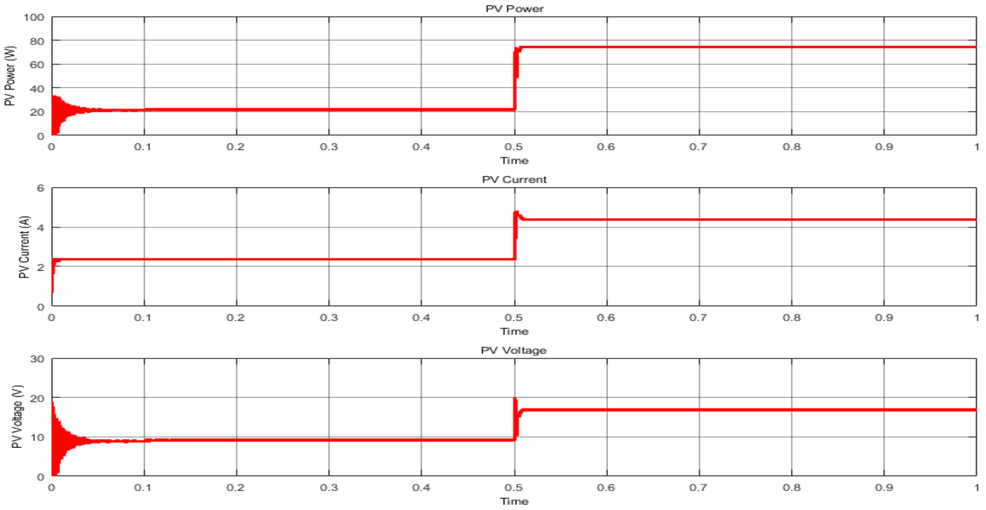


Figure 16. Power, voltage and current of PV for P&O technique

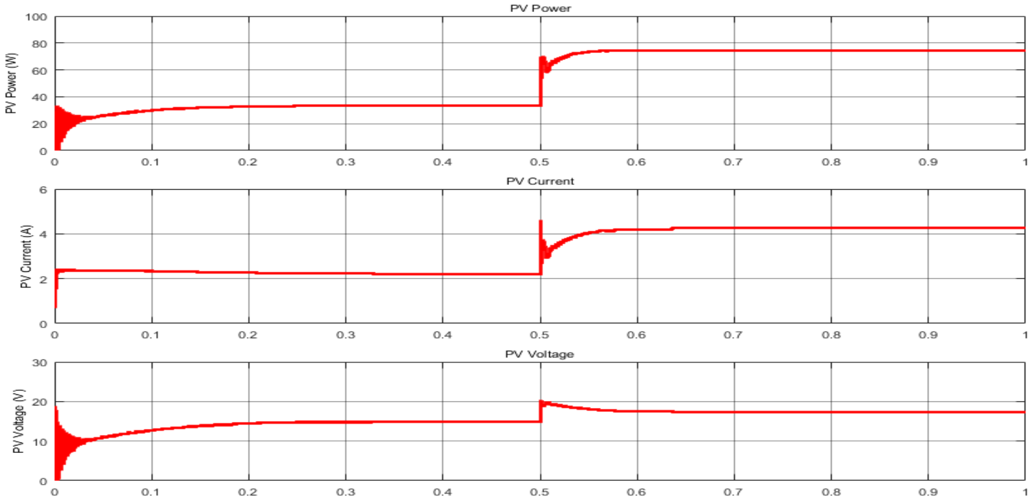


Figure 17. Power, voltage and current of PV for IC technique

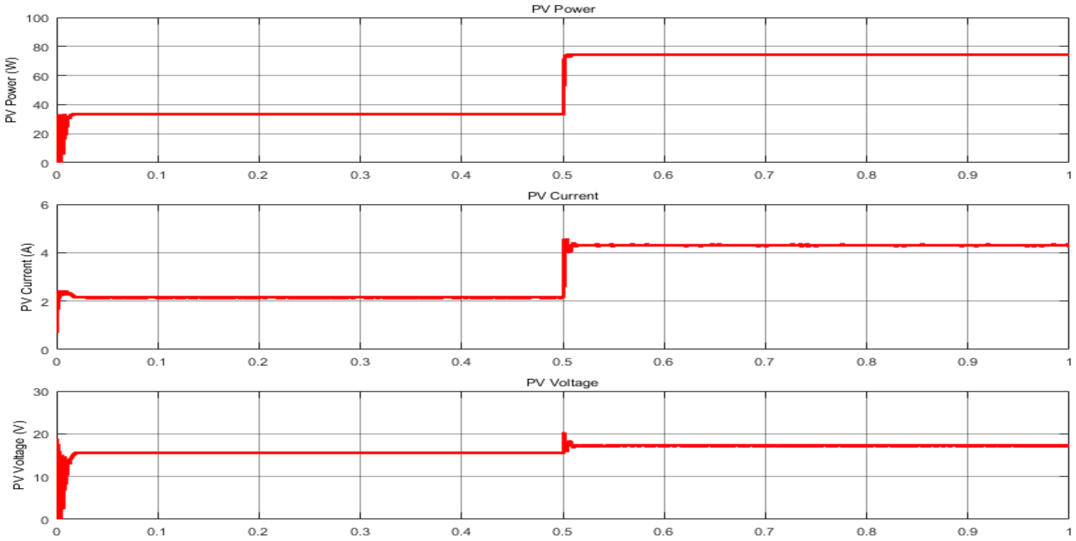


Figure 18. Power, voltage and current of PV for FLC toolbox

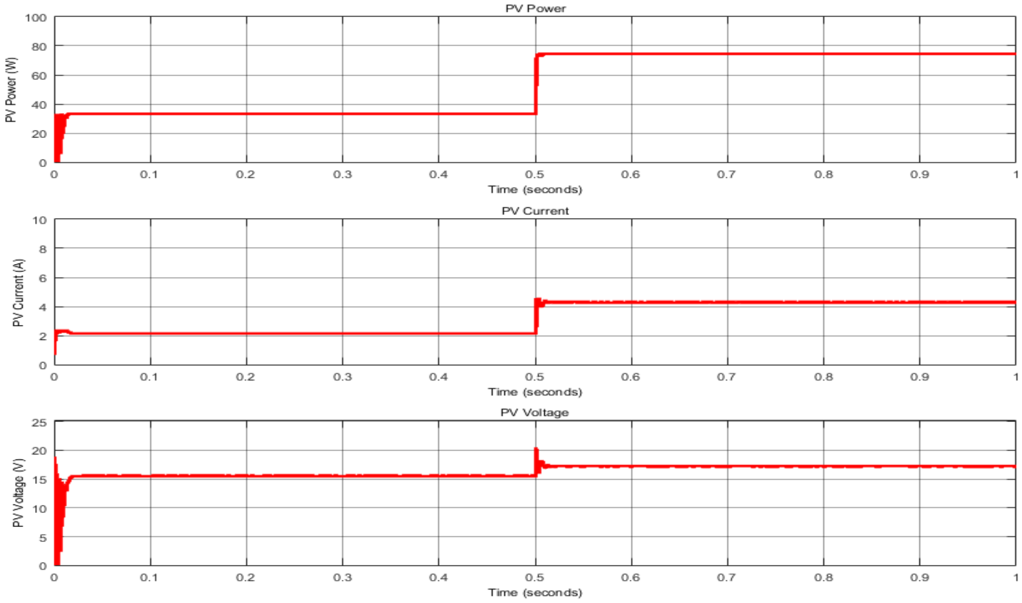
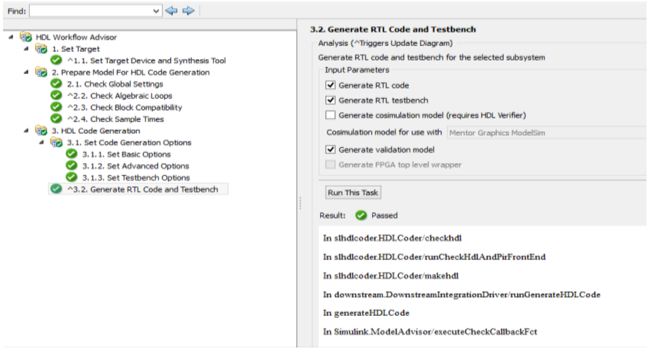


Figure 19. Power, voltage and current of PV for FLC Simulink

The FLC should be selected for its higher performance compared to the other controllers. FLC has better response time, less oscillation and much more accurate tracking at each step. The two different techniques of FLC have the same output response. So the mathematical model representation of FL is selected due to its simplicity for conversion to VHDL code. The first procedure in the conversion process from MATLAB SIMULINK to VHDL code is indicated in Figure 20. Once the FLC block diagram built in MATLAB/SIMULINK, it is converted to VHDL code, the converted VHDL code of the FLC subsystem is obtained in Fig. 21. Also, the VHDL code of the fuzzification, defuzzification, and rule base is shown in Figure 22, Figure 23 and Figure 24 respectively. The ISE Design Suite 14.6 software has been used for simulating the proposed architecture of FLC based on the VHDL as shown in Figure 25. The obtained results has been verified and compared with MATLAB. The simulation results approximately coincide with the simulation MATLAB results. In future the implementation of the proposed architecture will be implemented on FPGA chip.



- HDL Workflow Advisor
  - 1. Set Target
    - ^1.1. Set Target Device and Synthesis Tool
  - 2. Prepare Model For HDL Code Generation
    - 2.1. Check Global Settings
    - ^2.2. Check Algebraic Loops
    - ^2.3. Check Block Compatibility
    - ^2.4. Check Sample Times
  - 3. HDL Code Generation
    - 3.1. Set Code Generation Options
      - 3.1.1. Set Basic Options
      - 3.1.2. Set Advanced Options
      - 3.1.3. Set Testbench Options
    - ^3.2. Generate RTL Code and Testbench

3.2. Generate RTL Code and Testbench

Analysis (^Triggers Update Diagram)

Generate RTL code and testbench for the selected subsystem

Input Parameters

☑ Generate RTL code

☑ Generate RTL testbench

☐ Generate cosimulation model (requires HDL Verifier)

Cosimulation model for use with: Mentor Graphics ModelSim

☑ Generate validation model

☐ Generate FPGA top level wrapper

Result: ✓ Passed

In slhdlcoder.HDLCoder/checkhdl

In slhdlcoder.HDLCoder/runCheckHdlAndPirFrontEnd

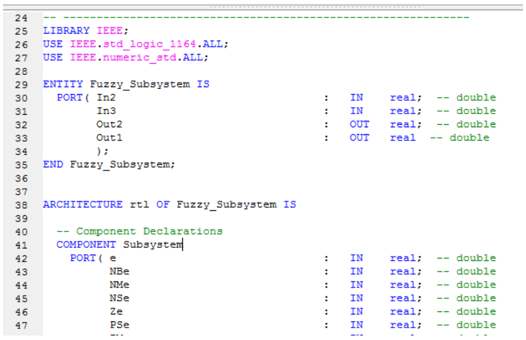
In slhdlcoder.HDLCoder/makehdl

In downstream.DownstreamIntegrationDriver/runGenerateHDLCode

In generateHDLCode

In Simulink.ModelAdvisor/executeCheckCallbackFct

Figure 20. Conversion process from MATLAB SIMULINK to VHDL code



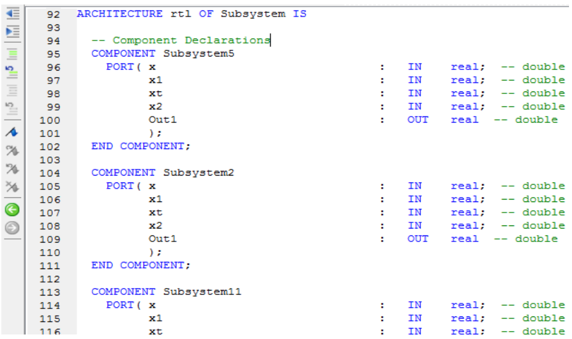
```
-- ---------------------------------------------------------------
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;

ENTITY Fuzzy_Subsystem IS
  PORT( In2                                       :   IN    real;  -- double
        In3                                       :   IN    real;  -- double
        Out2                                      :   OUT   real;  -- double
        Out1                                      :   OUT   real  -- double
        );
END Fuzzy_Subsystem;


ARCHITECTURE rtl OF Fuzzy_Subsystem IS

  -- Component Declarations
  COMPONENT Subsystem
    PORT( e                                       :   IN    real;  -- double
          NBe                                     :   IN    real;  -- double
          NMe                                     :   IN    real;  -- double
          NSe                                     :   IN    real;  -- double
          Ze                                      :   IN    real;  -- double
          PSe                                     :   IN    real;  -- double
```

Figure 21. VHDL code of the global subsystem of the FLC



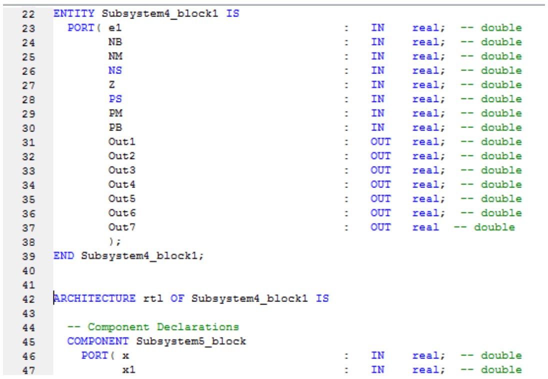
```
ARCHITECTURE rtl OF Subsystem IS

  -- Component Declarations
  COMPONENT Subsystem5
    PORT( x                                       :   IN    real;  -- double
          x1                                      :   IN    real;  -- double
          xt                                      :   IN    real;  -- double
          x2                                      :   IN    real;  -- double
          Out1                                    :   OUT   real  -- double
          );
  END COMPONENT;

  COMPONENT Subsystem2
    PORT( x                                       :   IN    real;  -- double
          x1                                      :   IN    real;  -- double
          xt                                      :   IN    real;  -- double
          x2                                      :   IN    real;  -- double
          Out1                                    :   OUT   real  -- double
          );
  END COMPONENT;

  COMPONENT Subsystem11
    PORT( x                                       :   IN    real;  -- double
          x1                                      :   IN    real;  -- double
          xt                                      :   IN    real;  -- double
```

Figure 22. VHDL code of the Fuzzification process

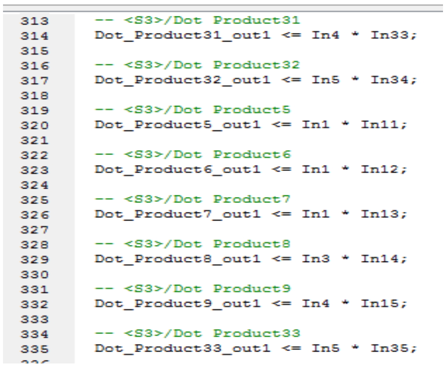


```
ENTITY Subsystem4_block1 IS
  PORT( e1                                        :   IN    real;  -- double
        NB                                        :   IN    real;  -- double
        NM                                        :   IN    real;  -- double
        NS                                        :   IN    real;  -- double
        Z                                         :   IN    real;  -- double
        PS                                        :   IN    real;  -- double
        PM                                        :   IN    real;  -- double
        PB                                        :   IN    real;  -- double
        Out1                                      :   OUT   real;  -- double
        Out2                                      :   OUT   real;  -- double
        Out3                                      :   OUT   real;  -- double
        Out4                                      :   OUT   real;  -- double
        Out5                                      :   OUT   real;  -- double
        Out6                                      :   OUT   real;  -- double
        Out7                                      :   OUT   real  -- double
        );
END Subsystem4_block1;


ARCHITECTURE rtl OF Subsystem4_block1 IS

  -- Component Declarations
  COMPONENT Subsystem5_block
    PORT( x                                       :   IN    real;  -- double
          x1                                      :   IN    real;  -- double
```

Figure 23. VHDL code of the deuzzification process



```
-- <S3>/Dot Product31
Dot_Product31_out1 <= In4 * In33;

-- <S3>/Dot Product32
Dot_Product32_out1 <= In5 * In34;

-- <S3>/Dot Product5
Dot_Product5_out1 <= In1 * In11;

-- <S3>/Dot Product6
Dot_Product6_out1 <= In1 * In12;

-- <S3>/Dot Product7
Dot_Product7_out1 <= In1 * In13;

-- <S3>/Dot Product8
Dot_Product8_out1 <= In3 * In14;

-- <S3>/Dot Product9
Dot_Product9_out1 <= In4 * In15;

-- <S3>/Dot Product33
Dot_Product33_out1 <= In5 * In35;
```

Figure 24. VHDL code of the Rulebase

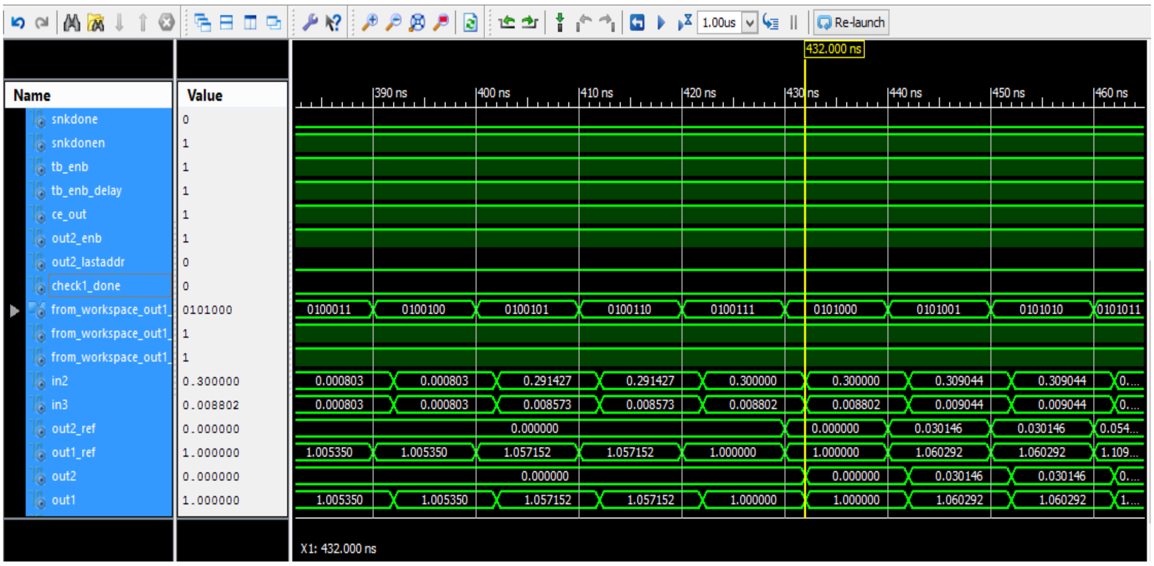


Figure 25. VHDL simulation of FLC subsystem

6. CONCLUSION

Photovoltaic stand-alone system simulation using Matlab/SIMULINK with a maximum power point tracking approach was presented in this paper. Also, several maximum power point tracking (MPPT) controller techniques were presented and implemented in Matlab/Simulink environment such as Perturb and Observe method (P&O), Incremental Conductance method (IC), and Fuzzy logic controller (FLC). The simulation results of FLC provided satisfactory results in extracting the MPP of PV compared with the other proposed controller techniques. The complete FLC was built in ISE design suite 14.6 software using very high speed integrated hardware description language to extract maximum power of the PV module. The simulation results validated the programs before downloading them on the FPGA. In future the implementation of the proposed architecture will be developed and implemented on FPGA chip.

REFERENCES

[1] R. Putri and M. Rifa'I, "Maximum Power Point Tracking Control for Photovoltaic System Using Neural Fuzzy," *International Journal of Computer and Electrical Engineering*, vol/issue: 4(1), 2012.

[2] C. Hua, C. Shen, "Comparative study of peak power tracking techniques for solar storage system," *Applied Power Electronics Conference and Exposition, APEC '98, Thirteenth Annual Proc*, pp. 679–83, 1998.

[3] K. H. Hussein, *et al.*, "Maximum photovoltaic power tracking: an algorithm for rapidly changing atmospheric conditions," *IEEE Proc Generation Transmission Distribution*, vol/issue: 142(1), pp. 59–64, 1995.

[4] A. Brambilla, "New approach to photovoltaic arrays maximum power point tracking," *Proc 30th IEEE Power Electron Specialists Conf.*, pp. 632–7, 1998.

[5] Hohm D. P. and Ropp M. E., "Comparative study of maximum power point tracking algorithm using an experimental, programmable, maximum power point tracking test bed," *Proc 28th IEEE Photovoltaic Specialist Conf*, pp. 1699–702, 2000.

[6] B. N. Alajmi, *et al.*, "Fuzzylogic-control approach of a modified hill-climbing method for maximum power point in microgrid standalone photovoltaic system," *IEEE Transactions on Power Electronics*, vol/issue: 26(4), pp. 1022–1030, 2011.

[7] C. B. Salah and M. Ouali, "Comparison of fuzzy logic and neural network in maximum power point tracker for PV systems," *Electric Power Systems Research*, vol. 81, pp. 43–50, 2011.

[8] A. Chaouachi, *et al.*, "A novel multi-model neuro-fuzzy-based MPPT for three-phase grid-connected photovoltaic system," *Solar Energy*, vol. 84, pp. 2219–2229, 2010.

[9] C. S. Chiu, "T-S fuzzy maximum power point tracking control of solar power generation systems," *IEEE Transactions on Energy Conversion*, vol/issue: 25(4), pp.1123–1132, 2010.

[10] T. L. Kottas, *et al.*, "New maximum power point tracker for PV arrays using fuzzy controller in close cooperation with fuzzy cognitive networks," *IEEE Transactions on Energy Conversion*, vol/issue: 21(3), pp. 793–803, 2006.

[11] T. Esram and P. L. Chapman, "Comparison of photovoltaic array maximum power point tracking techniques," *IEEE Transactions on Energy Conversion*, vol/issue: 22(2), pp. 439–449, 2007.

[12] Y. H. Liu, *et al.*, "Neural-networkbased maximum power point tracking methods for photovoltaic systems operating under fast changing environments," *Solar Energy*, vol. 89, pp. 42–53, 2013.

[13] A. Messai, *et al.*, “Maximum power point tracking using a GA optimized fuzzy logic controller and its FPGA implementation,” *Solar Energy*, vol. 85, pp. 265–277, 2011.

[14] K. Punitha, *et al.*, “Artificial neural network based modified incremental conductance algorithm for maximum power point tracking in photovoltaic system under partial shading conditions,” *Energy*, vol. 62, pp. 330–340, 2013.

[15] K. Punitha, *et al.*, “Development and analysis of adaptive fuzzy controllers for photovoltaic system under varying atmospheric and partial shading condition,” *Applied Soft Computing*, vol/issue: 13(11), pp. 4320–4332, 2013.

[16] R. Rajesh and M. C. Mabel, “Efficiency analysis of a multi-fuzzy logic controller for the determination of operating points in a PV system,” *Solar Energy*, vol. 99, pp. 77–87, 2014.

[17] M. Veerachary, *et al.*, “Neural-network-based maximum-power-point tracking of coupled-inductor interleavedboost-converter-supplied PV system using fuzzy controller,” *IEEE Transactions on Industrial Electronics*, vol/issue: 50(4), pp. 749–758, 2003.

[18] T. F. Wu, *et al.*, “A fuzzy-logic-controlled single-stage converter for PV-powered lighting system applications,” *Industry Applications Conference, Thirty-Fourth IAS Annual Meeting. Conference Record of the 1999 IEEE Proc.*, pp. 287–296, 1999.

[19] P. Balamurugan, *et al.*, “FPGA Implementation of MPPT Hybrid Algorithm for Photovoltaic System,” *International Conference on Engineering Innovations and Solutions Proc*., pp. 32-36, 2016.

[20] M. EL-Moghany, “Sun and Maximum Power Point Tracking in Solar Array Systems Using Fuzzy Controllers Via FPGA,” *The Islamic University*, Gaza, 2011.

[21] H. Yatimi and E. Aroudam, “Mathematical Modeling and Simulation of Photovoltaic Power Source using Matlab/Simulink,” *International Journal of Innovation and Applied Studies*, vol/issue: 16(2), pp. 322-330, 2016.

[22] K. Umadevi and C. Nagarajan, “Photovoltaic System with DC-DC Boost Converter Topology Using Psim Software,” *International Journal of Current Research and Modern Education*, Special Issue, pp. 51-60, 2016.