International Journal of System Dynamics Applications
Volume 11 • Issue 6

MPPT Control of Photovoltaic Systems Using Statechart With Abstraction and Its Comparison With Fuzzy Logic

Venkat Pankaj Lahari Molleti, AU College of Engineering, Andhra University, India
Rama Sudha Kashibhatla, AU College of Engineering, Andhra University, India
Vijaya Santhi Rajamahanthi, AU College of Engineering, Andhra University, India

ABSTRACT

Maximum power point tracking (MPPT) control is an indispensable aspect of photovoltaic (PV) systems. Many MPPT techniques including a few based on soft computing have been employed earlier. The soft-computing techniques include fuzzy-FSMs (finite state machines), which are integration of fuzzy logic (FL) into states or transitions of FSMs which are used for control and modeling of real-time systems. However, FSMs pose certain disadvantages as compared to its advanced variant called 'statecharts'. In this work, statecharts with abstraction layers are proposed for MPPT control of PV system. An abstract-statechart MPPT (ASM) controller is developed and is verified with PV system using co-simulation. A C++ based FL MPPT program is also developed, which is independent of any predefined and simulation-only functions. A conceptual estimation of execution time of such a FL MPPT program is presented and compared with the execution times delivered by proposed ASM controller. It can be observed that the ASM controller gives accurate, fast tracking speeds, along with the advantage of abstraction.

KEYWORDS

Abstraction, FSM, Fuzzy Logic, LabVIEW FPGA, MPPT, Photovoltaic, Statechart

INTRODUCTION

Soft computing techniques for maximum power point tracking (MPPT) of photovoltaic (PV) systems have been already used to achieve better performance of the systems (Amit et al., 2019; Hanane & Elhassan, 2018). These techniques include fuzzy logic control, genetic algorithms, probabilistic computing, etc (Amit et al., 2019; Mohmed et al., 2020). These techniques have also been used in conjunction with another effective tool for implementing real time control and modeling, which is the Finite State Machine (FSM). FSMs have been used as Fuzzy-FSMs(Hamzaoui et al., 2020a; Hamzaoui et al., 2020b; Mohmed et al., 2020; Reyneri, 1997; Speranskii, 2015), Genetic-FSM synthesis(Ali et al., 2004; Bereza et al., 2013), probabilistic-FSM(Li & Tan, 2019), etc. Hence, FSMs have hence established a significant hand-shake with soft computing techniques in various applications.

However, FSMs have certain limitations due to lack of abstraction in representing a complex system with many states which results in explosion of states(Harel, 1987; Lahari et al., 2019; Mierlo & Vangheluwe, 2019). This could pose a problem in efficiently integrating the FSM with a

DOI: 10.4018/IJSDA.20221101.oa1

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

soft-computing technique. To overcome this limitation of FSM, in this work, an extension of FSM, 'statechart' is presented for MPPT control of a PV system. The statechart is developed for an FPGA target and is modeled with abstraction in this work.

Statecharts, that can represent, model and also implement control of complex systems using states, transitions and actions codes (Harel, 1987; Mierlo & Vangheluwe, 2019), have unique features as explained further. Statecharts allow for 'abstraction' in their models, which is to have number of layers of states instead of a flat set of states as in FSMs. This abstraction or layering of states provides compact, simpler and flexible modeling possible. Statecharts also have feature of history, which is, their abstract or layered states can be modeled to have 'memory' of the last inner state that it was active in, before the higher-layered state is left. Also, statecharts method of communication is 'broadcast-communication' through which every state, layered or non-layered, can be communicated of an event occurring at any other state of the system at the same time as the event occurred and an apt control action can thus be achieved. These traits of abstraction, history, 'sensing' of an event by all states at same time (broadcast-communication)(Harel, 1987) make statecharts highly flexible, reliable and apt for systems that require 'softness' instead of 'crisp' methodology in their implementation and modeling. Hence, statecharts can be integrated seamlessly with other soft-computing techniques, as an alternative to FSMs being integrated in fuzzy-FSMs, genetic-FSM synthesis, probabilistic-FSM, etc.

For this reason, abstract-statecharts are proposed in this work, developed for achieving MPPT control of a photovoltaic system, with fast tracking speeds. The statechart is developed for an FPGA as a target and performance of the PV system is verified with the abstract-statechart MPPT(ASM) controller so-developed.

Also developed in this work is a C++ based fuzzy logic(FL) program for achieving MPPT control, so as to present a brief comparative aspect with the developed ASM controller. However, unlike conventional methods of fuzzy logic models using MATLAB fuzzy inference system(FIS) or MATLAB code(Aymen et al., 2018), a C++ based fuzzy-MPPT program is developed and result verified from an online compiler is presented. Implementing FL developed using FIS into hardware digital controllers require specific hardware interfacing modules(Aymen et al., 2018; Sana & Anis, 2018) with MATLAB. Else, they are to be verified in pure simulation which cannot be completely reliable as far as the real-time execution times of the FL are considered. FL developed based on MATLAB code uses pre-defined fuzzy-based function provided by the software which cannot be directly utilized to be deployed into physical embedded targets. When verified using pure simulation, this method also does not accurately provide the execution times of the FL control. To make approximate estimation of execution times of the FL possible even in simulation, FL programs developed around textual languages that can be used to program embedded targets are to be used, which are C and C++, as they can be easily converted into assembly language programs(Sibigtroth, 1996) and based on the target digital controller and its machine cycle frequencies(Atmel, n.d.), the execution times of the FL program can be estimated. Hence, in this work, the FL developed for comparison of its estimated execution time with that of the ASM controller, is based on C++. This requires that every stage in FL development (namely defining fuzzy variables, membership functions, fuzzy rules, aggregation/defuzzification) is to be programmed as the embedded targets do not have pre-defined fuzzy functions like MATLAB simulation provides. In this work, development of code for each part of FL in a textual language without pre-defined fuzzy libraries is discussed and presented. The execution time of FL code so-developed is approximately estimated and compared with the results obtained from the proposed abstract-statechart MPPT(ASM) controller.

Background

Statecharts and fuzzy logic can have a good integration with each other, as already indicated by successful applications of fuzzy-FSMs (Mohmed et al., 2020; Speranskii, 2015). However, the hand-shake between fuzzy logic and statecharts would be more efficient due to the duality between the two and also because statecharts address the limitations posed by FSMs(Harel, 1987). It can be understood

from the various attributes of statecharts and fuzzy logic techniques. Fuzzy logic uses 'linguistic' variables like 'negative big', 'positive big', etc instead of crisp numbers and ranges. Statecharts also provide this flexibility in the form of event-based 'state actions' like 'increase', 'decrease', 'move up', 'move down', etc. Fuzzy logic utilizes fuzzy rules based on 'if-then' clauses. Statecharts also utilize these types of clauses in the form of 'transition guard' codes, which if true, allow for a state transition. Where fuzzy logic uses fuzzy variables, the inputs is system are converted into state variables in statecharts. While fuzzy logic can introduce 'fuzziness' in the control, statecharts can introduce 'abstraction' in the control. However, a significant advantage of statecharts over fuzzy logic control is that they can also generate control events along with producing a control variable output. Also fuzzy logic does not cater for 'feedback' or 'memory' in its control, while statecharts can incorporate 'memory' or 'history' in their control implementation.

There are several MPPT techniques for PV systems, among which a few are integrated with fuzzy logic widely, like the Perturb & Observe algorithm. Also, the most widely used MPPT algorithm are P&O and Incremental Conductance. So, in this work, one of them, P & O algorithm is modeled and verified via statecharts. Stateflow of MATLAB/ Simulink, which is based on FSM, had been used for MPPT control in renewable energy systems in Ahmed et al.(2014), Ali et al. (2016), Maher et al. (2019), Shourov et al.(2018). In Ali et al. (2016), it has been used to model incremental conductance algorithm. However, the verification is done in purely offline simulation and the model developed is not for any targeted embedded controller. In Shourov et al. (2018) also, stateflow based MPPT verification is performed using offline simulation and the MPPT algorithm for which stateflow is used is P & O algorithm. This stateflow based model is also not targeted for any embedded controller. In Maher et al. (2019) and Ahmed et al. (2014), stateflow-based P & O algorithms were implemented in digital signal processors (DSPs) and MPPT tracking is observed. However, statecharts have a unique feature of concurrency which couldn't be implemented by sequential processors like DSPs. Hence, in this work, FPGA, which is capable of parallel processing and hence which is suitable for statechart implementation, is chosen as the target embedded controller for modeling the statechart. In Lahari et al.(2019), offline statechart models have been developed for P & O and Incremental Conductance algorithms and MPPT tracking is achieved. However, they are also not for any targeted embedded controller. More importantly, in all the works mentioned above, the element of 'abstraction' of statecharts, which is required for integration with other soft-computing techniques, was not utilized. In this work, along with developing the statechart model for an FPGA, there is also abstraction of states included in the model.

PV SYSTEM WITH ABSTRACT-STATECHART MPPT(ASM) CONTROLLER

PV System

Figure 1 shows the block diagram of the PV system under consideration. The simulation model developed in NI Multisim gives behavior of a PV panel BP MSX 120 interfaced to a resistive load via a boost converter. The statechart developed provides the control output, dutycycle (D) of the boost converter, due to which the MPP is tracked under varying system conditions.

The parameters I, V, R denote current, voltage, resistance and the subscripts PV, O, IN denote that the parameters are of PV panel, load, input side respectively. Table 1 shows the specifications of the PV panel under consideration (BP Global Solar Marketing, n.d.). Also, in the work, the default load resistance value taken is 1 kΩ.

Abstract-Statechart MPPT(ASM) Controller

In this work, P & O based abstract-statechart MPPT (ASM) controller is developed for a PV system, using LabVIEW FPGA. Abstract statecharts imply that there are states which are layered. In the developed ASM controller, there is an overall state in abstraction layer-1 which includes abstraction

Figure 1.Block Diagram of PV system with Abstract-Statechart MPPT controller

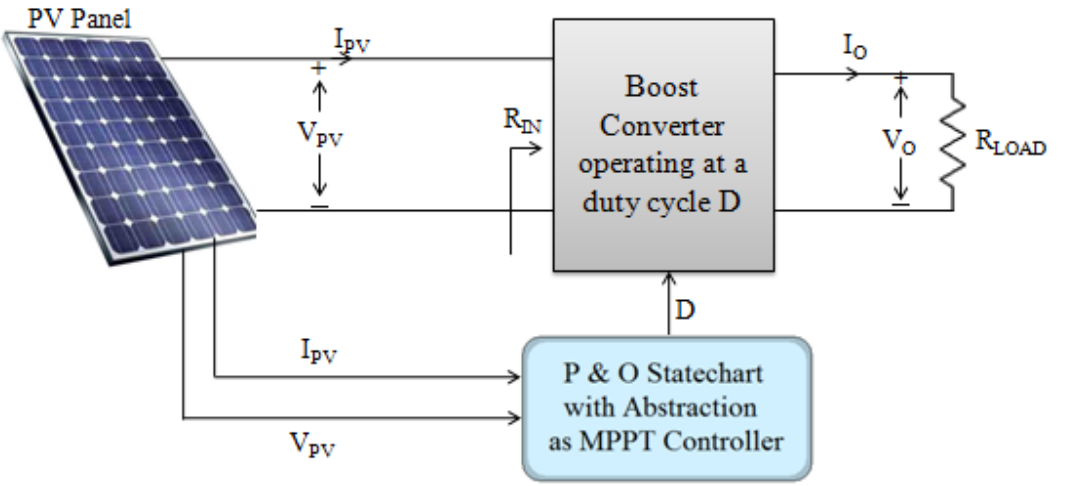


Table 1. Specifications of PV Panel

Parameter	Value	Units
Maximum Power	120	W
Standard Irradiance	1000	W/m^2
Standard Temperature	25	ºC
Short Circuit Current	3.87	A
Open Circuit Voltage	42.1	V

layer-2 with two states – parent state and the state Sf. Sf is a final state. The parents state in turn comprises of another abstraction layer-3 with three states whose actions are to 'initialize', ' increase' and 'decrease' This layer actually contains the control logic for MPP tracking. Figure 2 shows the abstraction in the developed ASM controller in this work. The ASM controller so-developed is for an FPGA target, which is ensured by proper configuration settings and accurate fixed-point definitions of the state variables which are current, voltage of PV panel and duty cycle. Figure 3 shows the configuration setting for the FPGA-based ASM controller. There are various targets for which statecharts can be modeled, like for offline simulation (desktop target), FPGA target, real-time processor target, touchpad target, etc. Since FPGA is a digital controller capable of parallel processing and is hence more apt for statecharts as they inherently possess concurrency, in this work, FPGA is chosen as the target.

The inputs to ASM controller are current, voltage values from the PV system, receiving which, it modifies the duty cycle value so as to track the MPP point.

Results with Abstract-Statechart MPPT(ASM) Controller

The performance of simulated PV system model is tested with the FPGA-based ASM controller by co-simulation and its performance is observed. The PV system performance with ASM controller is observed in three cases namely constant irradiance and load condition, variable irradiance condition and variable load condition (the temperature is maintained at standard value in all cases). Also, the

Figure 2. Developed Abstract-Statechart MPPT(ASM) Controller

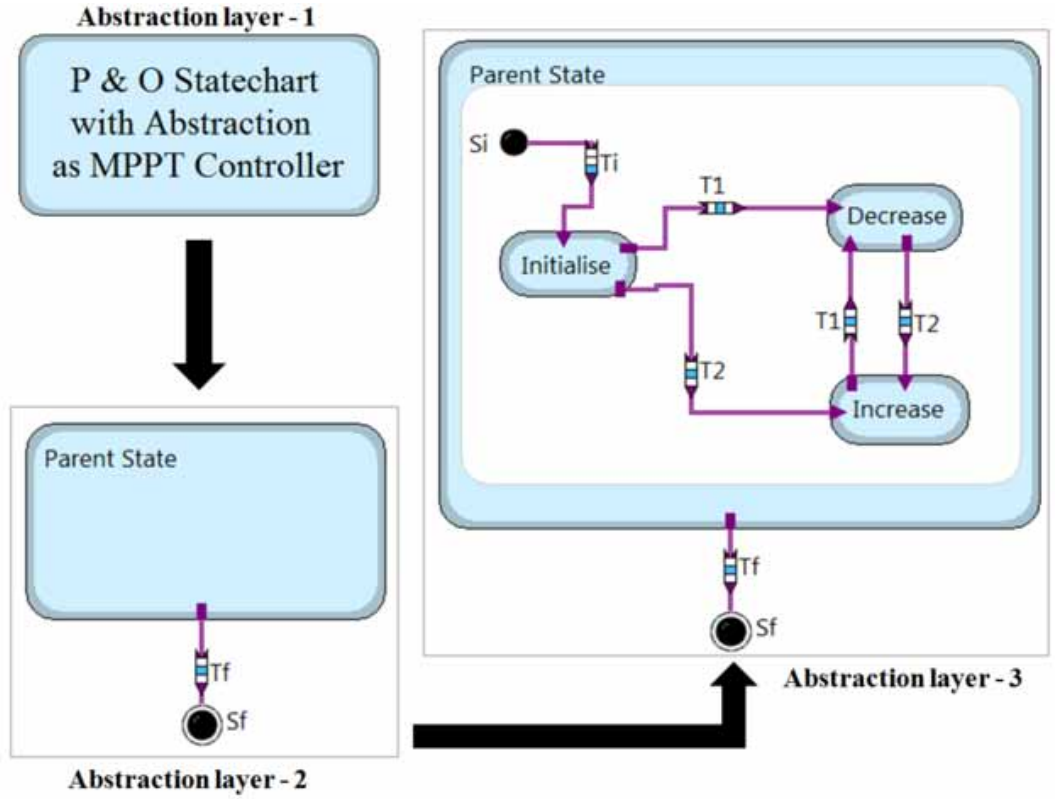
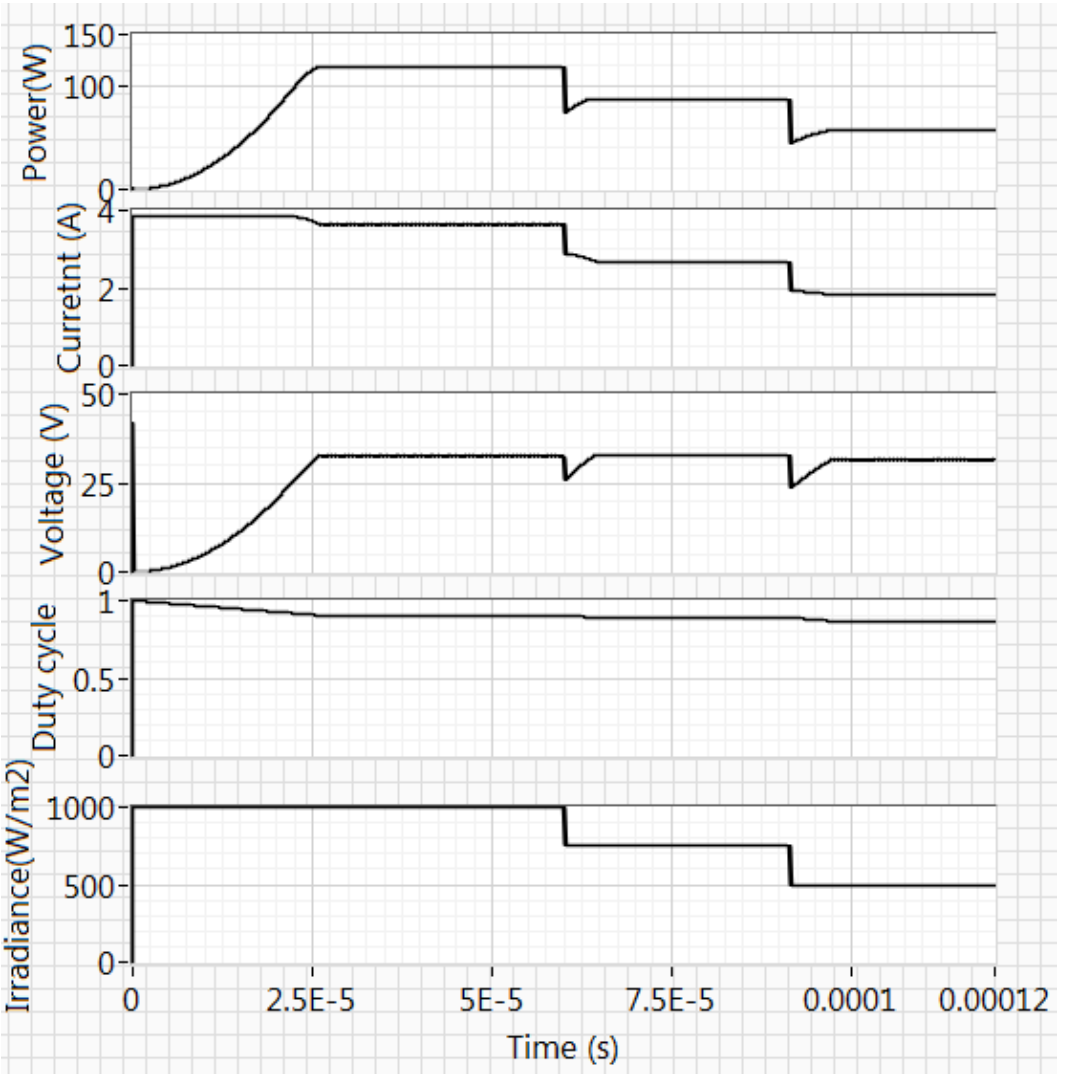


Figure 3. Configuration of the ASM controller for FPGA target

ASM controller's performance is observed with variable step-sizes of duty ratio. Figures 4, 5 show the performance plots of PV system with ASM controller under variable irradiances (load constant at 1kΩ) with two different step-sizes. Figures 6,7 show the performance plots of PV system with ASM controller under variable loads (irradiance constant at 1000W/m^2) with two different step-sizes respectively. It can be observed that in every case, the MPP has been tracked by the abstract-statechart MPPT(ASM) controller.

For the considered PV panel, the MPP values at irradiances of 1000W/m^2, 750W/m^2, 500W/m^2 are 118.7W, 87.92W, 57.24W respectively (BP Global Solar Marketing, n.d.; Lahari et al., 2019). When the load is varied, since the irradiance is maintained constant at 1000W/m^2, at every change in load

Figure 4. PV system plots with ASM controller under varying irradiances (step-size 0.0001)



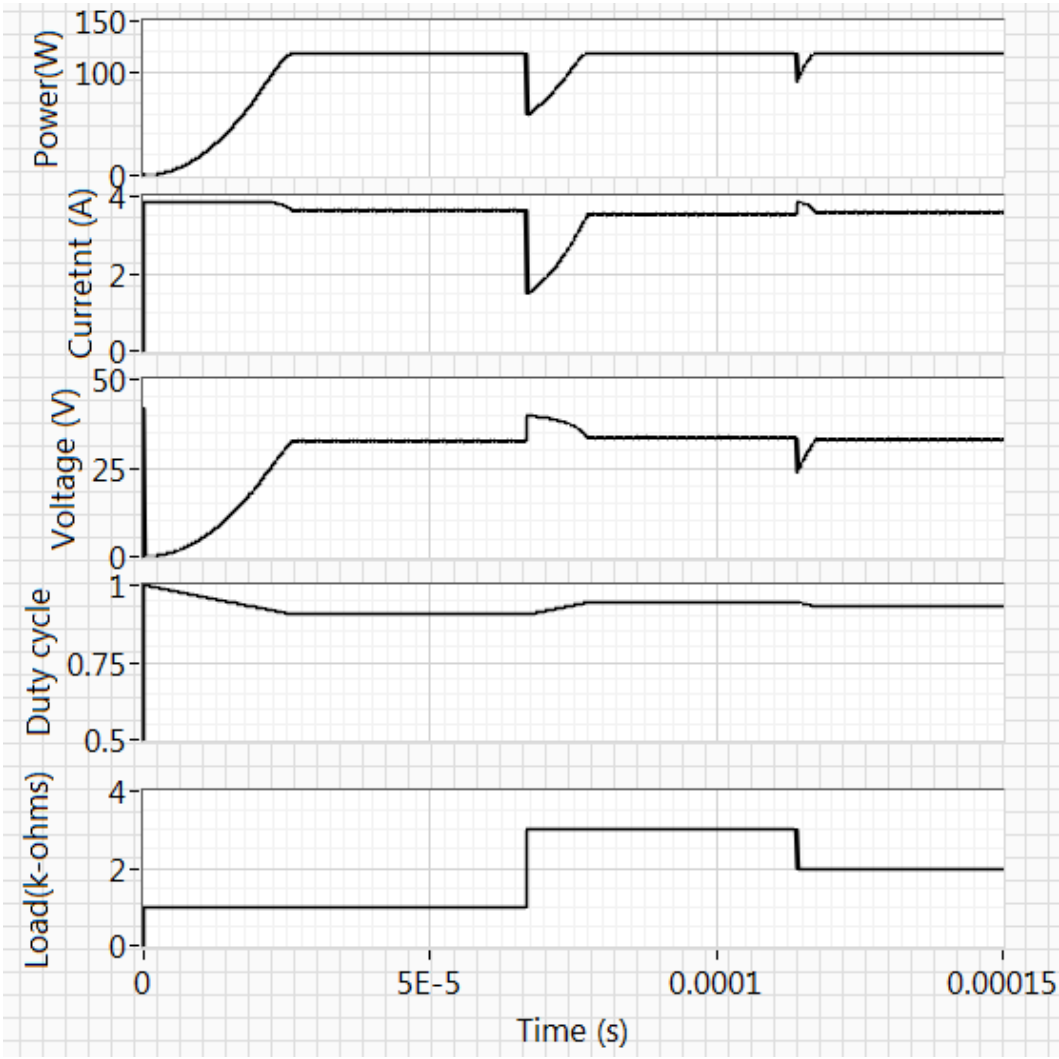
the MPP point is still 118.7W which is to be attained by the MPPT controller (corresponding duty cycle is 0.9023). It can be observed from the figures that in every case, the MPP tracking by ASM controller has been done in a few microseconds. The exact execution times of the ASM controller in tracking MPP in every case is tabulated in summary section of the paper.

FUZZY LOGIC BASED MPPT PROGRAM FOR THE PV SYSTEM

For developing FL based MPPT program (Appendix I shows the developed program), the fuzzy linguistic variables and their possible values, their membership functions, the fuzzy rules, aggregation from the fuzzy rules or de-fuzzification are all to be defined and incorporated. In this case of MPPT, the inputs would be error, e and change in error, h defined as follows.

$e = (P_{now}-P_{pre})/(V_{now}-V_{pre})$ (i)

Figure 5. PV system plots with ASM controller under varying loads (step-size 0.0001)



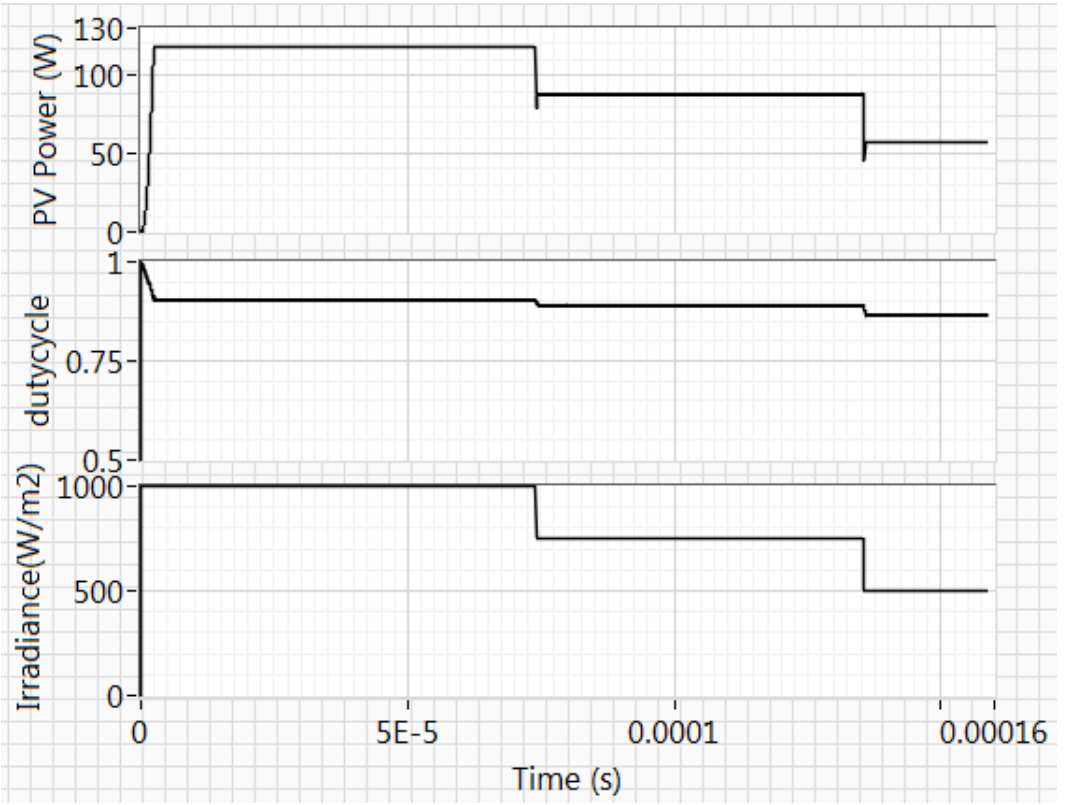
$h = e_{now} - e_{pre}$ (ii)

where P and V stand for voltage and power of the PV system and the subscripts 'pre', 'now' stand for present and previous iteration values respectively.

With e and h being the linguistic variables (defined with range [-1 0 1]), their number of linguistic values are also to be fixed in the program which is taken to be 3 (as zero - Z, positive - P, negative - N). Also, the output variable is dutycycle d (with range [0 to 1], and it has five linguistic values (negative small- NS, negative big -NB, zero -ZE, positive big- PB, positive small- PS).

The next segment of the program is to define the membership functions of the linguistic variables (Figure 8), so that their membership with each linguistic value can be calculated by the program. It is to be recollected that in MATLAB, there are pre-defined functions for membership functions as well and hence the programmer would not have to build these membership functions but needs to simply

Figure 6. PV system plots with ASM controller under varying irradiances (step-size 0.001)



use them. But in the practical programs intended to be deployed in target embedded controllers, this part is also to be programmed.

The membership functions taken are triangular for all the linguistic variables and they are developed by the mathematical equations of the line segments formed from the points on x and y axis of the membership functions (Sibigtroth, 1996). As the linguistic values taken for both the inputs are three, two arrays (one for each linguistic input variable), are defined with size 3 and their equations are formed based on their points on the x and y axis. For example, in between -1 and 0, for the part of ZE, the mathematical equation can be found from its two points (-1,0) and (1,0).

Once membership functions are defined in the program, the fuzzy rules are to be programmed using if-then clauses based on the rules set (as shown in Table 2). Figure 9 shows the code for developing membership functions and rules.

The final step requires aggregation/ defuzzification. For the centre of gravity method, the program can be developed as follows. The membership functions for the linguistic values of output linguistic values can also be determined from the points of the line segments. Hence, for every linguistic value, there exists a crisp number on the x axis of the membership function graph. Hence, the x and y coordinates can be determined for every linguistic output value obtained from the rules set and a product performed with its corresponding x-axis value (denoted by y[0] to y[8] in the program) can give us the numerator in the centre of gravity defuzzification and sum of all the memberships/ linguistic values obtained from the rules set becomes the denominator. Hence, the defuzzified value of the output can be determined.

Figure 7. PV system plots with ASM controller under varying loads (step-size 0.001)

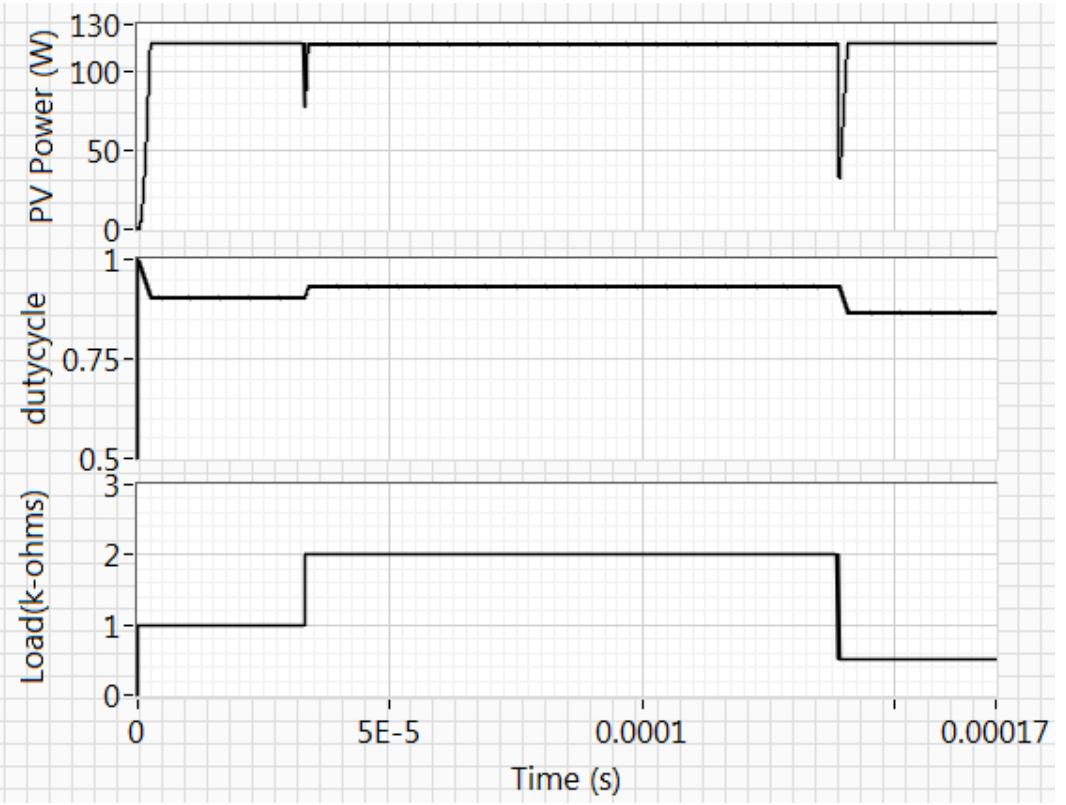


Figure 8. Membership functions of input and output linguistic variables

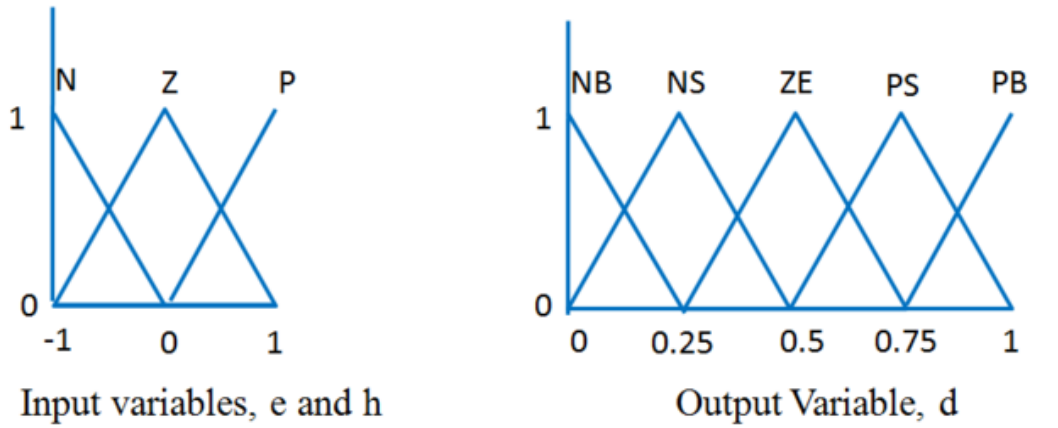


Figure 9. Sample code for Membership functions and rules in FL MPPT program

```
//define membership functions
if (e>-1 && e<0)
{
x0[0]= -e;
x0[1]= e+1;
}
else if (e>0 && e<1)
{
x0[1]=-e+1;
x0[2]=e;
}
if (h>-1 && h<0)
{
x1[0]= -h;
x1[1]= h+1;
}
else if (h>0 && h<1)
{
x1[1]=-h+1;
x1[2]=h;
}
```

```
//rules
if (x0[0] && x1[0]!=0)
x2[2]=x0[0]<x1[0]?x0[0]:x1[0];
if (x0[0] && x1[1]!=0)
x2[4]=x0[0]<x1[1]?x0[0]:x1[1];
if (x0[0] && x1[2]!=0)
x2[4]=x0[0]<x1[2]?x0[0]:x1[2];
if (x0[1] && x1[0]!=0)
x2[3]=x0[1]<x1[0]?x0[1]:x1[0];
if (x0[1] && x1[1]!=0)
x2[2]=x0[1]<x1[1]?x0[1]:x1[1];
if (x0[1] && x1[2]!=0)
x2[1]=x0[1]<x1[2]?x0[1]:x1[2];
if (x0[2] && x1[0]!=0)
x2[0]=x0[2]<x1[0]?x0[2]:x1[0];
if (x0[2] && x1[1]!=0)
x2[0]=x0[2]<x1[1]?x0[2]:x1[1];
if (x0[2] && x1[2]!=0)
x2[2]=x0[2]<x1[2]?x0[2]:x1[2];
```

Table 2. Fuzzy rule set

Down -e/ Across -h	N	Z	P
N	ZE	PB	PB
Z	PS	ZE	NS
P	NB	NB	ZE

Figure 10. Sample output from FL MPPT program

```
//defuzzification

y[0]=-0.25*x2[0]+0.25;
y[1]=0.25*x2[1];
y[2]=-0.25*x2[1]+2;
y[3]=0.25*x2[2]+0.25;
y[4]=-0.25*x2[2]+0.75;
y[5]=0.25*x2[3]+0.5;
y[6]=-0.25*x2[3]+1;
y[7]=0.25*x2[4]+0.75;
cog_num = ((x2[0]*y[0]) + (x2[1]*(y[1]+y[2])) +(x2[2]*(y[3]+y[4])) +(x2[3]*(y[5]+y[6])) +(x2[4]*y[7]) );
cog_den=(x2[0]+x2[1]+x2[1]+x2[2]+x2[2]+x2[3]+x2[3]+x2[4]);
cog=cog_num/cog_den;
cout <<"CRISP OUTPUT VALUE OF DUTY CYLE: " <<cog;
    }

Input
CRISP OUTPUT VALUE OF DUTY CYLE: 0.903727

...Program finished with exit code 0
Press ENTER to exit console.
```

Table 3. ASM Controller execution time for tracking MPPT

Case	Tracking time with Stepsize 0.001	Tracking time with Stepsize 0.0001
Uniform irradiance and load	2.7 μs	26.5 μs
Variable irradiance	0.5 μs	5.4 μs
Variable load	0.8 μs	11.1 μs

With this FL MPPT program executed in online compiler (OnlineGDB, n.d.) and inputs being given as 0.06 and 0.46, the obtained defuzzified output is d=0.903 which is close to the duty cycle corresponding to the MPP of the PV system as already seen in results from ASM controller. Figure 10 shows the sample output.

SUMMARY

The FL program developed hence contains many segments and hence would require many number of instructions. However, the program developed is in a high-level language, which when required to be downloaded into a target embedded controller, would have to be converted into a corresponding assembly language program. This conversion is done by compilers. But the drawback is that each instruction in the high-level language might be converted into multiple assembly language instructions(David, 1999), for the program to not lose its logical accuracy. And the number of instructions into which it is converted cannot be determined. The current program has approximately 90 instructions in the high level language, which means it would have atleast 200 instructions in the converted assembly language program(David, 1999). Also, the number of assembly language instructions in the final converted program varies from compiler to compiler and hence is unpredictable. Hence, the execution time taken by the FL program in the embedded controller is quite high(Sibigtroth, 1996) (even if a time as low as 0.06μs is assumed for execution of one machine cycle as in the case of some Atmega controllers of Arduino.) This is because the time taken by each assembly language instruction could in turn require multiple machines cycles for execution(Atmel, n.d.) and hence the final execution speed would be in order of tens and hundreds of microseconds or more.

On the contrary, the execution times obtained by the ASM controller can be observed from Table 3, which shows that it is in the order of micro and sub-micro seconds in all cases. Hence, the ASM controller developed is a fast-tracking and accurate MPPT controller, also providing the advantages of abstraction which can be utilized for its integration with fuzzy-based control techniques and for systems with soft boundaries and uncertain events.

Conclusion

An abstract-statechart MPPT(ASM) controller with three abstraction layers has been developed using LabVIEW FPGA in this work, with the target of programming being FPGA. The developed ASM controller has been verified with the PV system and it is observed that in various system conditions, the developed controller accurately tracked the MPP and with fast-tracking speed.

A C++ based FL MPPT program is also developed and the various stages required for its development are discussed. The importance of such programs which are independent of pre-defined library functions that are common in MATLAB is that they can be deployed into target embedded controllers and these programs allow for initial debugging of the FL developed. A broad approximation of execution time of these programs allows for an understanding that these are slower compared to the developed ASM controller, since all the instructions developed in a high-level language are to be

converted into assembly language instructions which results in programs with unpredictable number of instructions and execution times.

Comparing and observing the results from these two techniques with execution-time point of view, it is clear that ASM controller achieved faster tracking speeds. Hence, this work suggests that ASM controller is an effective MPPT controller for photovoltaic systems.

Also, when implementation into a digital controller is desired, an integration of FL with abstract-statechart would be more effective rather than purely FL owing to the high execution times of the latter. This work also suggests fuzzy-ASM controller over fuzzy-FSM controllers due to the advantages of abstraction, memory and broadcast-communication of statecharts. These aspects would be further explored in future works.

REFERENCES

Ahmed, R., Abdelsalam, A. K., Namaan, A., Dessouky, Y. G., & M'Sirdi, N. K. (2014). Improved performance State-Flow based photovoltaic Maximum Power Point Tracking Technique. In *Proceedings of 3rd IET Renewable Power Generation Conference*. IET. doi:10.1049/cp.2014.0883

Ali, B., Almaini, A. E. A., & Kalganova, T. (2004). Evolutionary Algorithms and Theirs Use in the Design of Sequential Logic Circuits. *Genetic Programming and Evolvable Machines*, *5*(1), 11–29. doi:10.1023/B:GENP.0000017009.11392.e2

Ali, U. S., Veeraraghavulu, D. V., Niveditha, M., Priyadharshini, N., & Sandhiya, P. (2016). Stateflow based incremental conductance MPPT of a P.V. system using Z - source DC - DC converter. In *Proceedings of Int. Conf. of PESTSE*. IEEE.

Amit, K. P., Naruttam, K. R., & Hemanshu, R. P. (2019). MPPT methods for solar PV systems: A critical review based on tracking nature. *IET Renewable Power Generation*, *13*(10), 1–19.

Atmel. (n.d.). *Atmega16.* http://ww1.microchip.com/downloads/en/devicedoc/doc2466.pdf

Aymen, J., Ons, Z., Mohamed, A. H., & Mohamed, N. M. (2018). Hardware Implementation of a Fuzzy Logic Controller for a Hybrid Wind-Solar System in an Isolated Site. *International Journal of Photoenergy*, *2018*(5), 1–16.

Bereza, A., Lyashov, M., & Blanco, L. (2013) Finite state machine synthesis for evolutionary hardware. In *Proceedings of 11th East-West Design and Test Symposium*. EWDTS. doi:10.1109/EWDTS.2013.6673134

David, S. E. (1999). *An Embedded Software Primer*. Addison Wesley.

Hamzaoui, F., Khadhraoui, M. & Messaoud, H. (2020a) A New Design of a Functional H Filter for Linear Singular Systems With an Additional Unknown Input and Bounded Disturbances. *International Journal of System Dynamics Applications, 9*(4), 55-73.

Hamzaoui, F., Khadhraoui, M. & Messaoud, H. (2020b). Design of a Functional H Filter for Linear Singular Systems With an Additional Unknown Input, Bounded Disturbance, and Variable Time Delay. *International Journal of System Dynamics Applications, 9*(4), 24-54.

Hanane, Y. & Elhassan, A. (2018) Standalone Photovoltaic System with Maximum Power Point Tracking: Modeling and Simulation. *International Journal of System Dynamics Applications, 7*(3), 94-111.

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, *8*(3), 231–274. doi:10.1016/0167-6423(87)90035-9

Lahari, M. V. P., Sudha, K. R., & Santhi, R. V. (2019). *Statechart Models of MPPT Controller for a Photo-Voltaic System in Co-Simulation Environment. In TENCON 2019 IEEE Region 10 Conference*. IEEE.

Li, J., & Tan, Y. (2019). A probabilistic finite state machine based strategy for multi-target search using swarm robotics. *Applied Soft Computing*, *77*, 467–483. doi:10.1016/j.asoc.2019.01.023

Maher, R. A., Abdelsalam, A. K., Dessouky, Y. G., & Nouman, A. (2019). High performance state-flow based MPPT technique for micro WECS. *IET Renewable Power Generation*, *13*(16), 3009–3021. doi:10.1049/iet-rpg.2019.0157

Mierlo, S. V., & Vangheluwe, H. (2019) Introduction to Statecharts Modelling, Simulation, Testing, and Deployment. In *Proceedings of 2019 Winter Simulation Conference (WSC)*. IEEE. doi:10.1109/WSC40007.2019.9004771

Mohmed, G., Lotfi, A., & Pourabdollah, A. (2020). Enhanced fuzzy finite state machine for human activity modelling and recognition. *Journal of Ambient Intelligence and Humanized Computing*, *11*(12), 6077–6091. Advance online publication. doi:10.1007/s12652-020-01917-z

OnlineGDB. (n.d.). *Online C++ Compiler*. https://www.onlinegdb.com/online_c++_compiler

Reyneri, L. M. (1997). An introduction to Fuzzy State Automata. In *Proceedings of Biological and Artificial Computation: From Neuroscience to Technology*. Springer. doi:10.1007/BFb0032485

Sana, B. & Anis, S. (2018) Adaptive Neuro-Fuzzy Sliding Mode Controller. *International Journal of System Dynamics Applications, 7*(2), 34-54.

Shourov, E. C., Sajid, A., Sabzehgar, R., & Roshan, Y. M. (2018). A Novel Implementation of the State Flow Approach for Maximum Power Point Tracking of Photovoltaic Systems. In *Proceedings of IEEE International Symposium on Power Electronics for Distributed Generation Systems*. IEEE. doi:10.1109/PEDG.2018.8447577

Sibigtroth, J. M. (1996). Programming fuzzy logic in assembly language. In *Proceedings of IEEE Technical Applications Conference*. IEEE. doi:10.1109/NORTHC.1996.564981

Solar Global Marketing, B. P. (n.d.). *BP MSX 120*. http://www.soltec-solar.com/html/cms/bp/product_msx_120.pdf

Speranskii, D. V. (2015). Experiments with fuzzy finite state machines. *Automation and Remote Control*, *76*(2), 278–291. doi:10.1134/S0005117915020071

Venkat Pankaj Lahari Molleti is a full-time research scholar and guest faculty at Andhra University College of Engineering. Areas of Interest: Embedded Systems, Intelligent control, Automation.

Rama Sudha Kashibhatla is a Professor and Director of the AU Computer Centre, Andhra University, Visakhapatnam, AP, India. Areas of Interest: Fuzzy Logic applications, Digital Logic, Embedded Systems, Power Systems.

Vijaya Santhi Rajamahanthi is an Assistant Professor at the Andhra University College of Engineering. Areas of Interest: Power Systems, Fuzzy Logic applications, Renewable Energy.