Energy dispatch fuzzy controller for a grid-independent photovoltaic system

Richard L. Welch, Ganesh Kumar Venayagamoorthy *

Abstract

This paper presents the development of an optimized fuzzy logic based photovoltaic (PV) energy dispatch controller using a swarm intelligence algorithm. The PV system considered is grid-independent and consists of a fuzzy logic controller (FLC), PV arrays, battery storage, and two types of loads: a constant critical load and a time-varying non-critical load. The swarm intelligence algorithm applied in this paper is the particle swarm optimization (PSO) algorithm, and it is used to optimize both membership functions and rule set of the FLC. By using the PSO algorithm, the optimized FLC is able to maximize energy to the system loads while also maintaining a higher average state of battery charge. This optimized FLC is then compared with the standard energy dispatch controller, referred to as the “PV-priority” controller. The PV-priority controller attempts to power all loads and then charge the battery resulting on lesser number of days of power to critical loads unlike the optimized FLC.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

As the cost of traditional fossil fuels continues to rise, the cost of electricity generated by traditional means also increases. However, as technology and manufacturing processes improve the cost of alternative energy sources such as solar and wind energy decreases. This narrowing gap between the rising cost of traditional energy sources and lowered cost of renewable energy is driving demand growth for renewable energy to unprecedented levels. Even so, the difference in cost of electricity generated by wind (and especially solar) and that generated by conventional sources is not insignificant, thus making some optimal control of a renewable energy source a good way to make the overall system more economical. By using a smart or optimal energy dispatch controller, more of the load can be met than by using a traditional controller with the same sized system. Additionally, others have employed a decentralized topology when utilizing large numbers of photovoltaic (PV) arrays in order to better match individual components, but even these technologies can benefit from optimal control. Researchers have reported on improving the efficiency of photovoltaic systems by carrying out optimal control of PV systems with fuel cells. In this case, the excess solar energy is stored by transforming it into hydrogen.

Traditionally, PV energy dispatch controllers have been simple devices that do not assign priority to various loads. Instead, they attempt to power all loads all of the time, and if there is any excess energy, then that excess energy is used to charge the batteries. Previous work by other researchers has reported on the development of non-optimized fuzzy logic controllers for large solar thermal plants, as well as small grid-independent systems. Additionally, one effort produced a fuzzy logic controller for a solar thermal plant with a rule base optimized using a genetic algorithm. The authors have reported optimizing the membership functions of a fuzzy logic controller.

In this paper, a fuzzy logic controller (FLC) is developed to assign priority to the installed system loads such that all critical loads receive a higher priority than the non-critical loads, and so when there exists a shortage of available energy the critical loads are first met before attempting to power the non-critical loads. This energy dispatch controller is also optimized to maintain a higher battery charge so that the controller is better able to power critical loads during an extended period of unfavorable weather conditions or low solar insolation. In this study, the simultaneous optimization of the membership functions and rule base of a fuzzy logic controller is carried out. A simulation study is carried out using Matlab with hourly data from the typical meteorological year 2 (TMY2) database [9]. The TMY2 database maintains hourly meteorological data for many cities, and most important for this study is the solar radiation or insolation received at each location. In this paper, three locations are chosen that represented varying degrees of average insolation. For the low average insolation case, Caribou, ME is selected; for the average case, Omaha, NE is chosen; and finally, for the high average insolation case Las Vegas, NV is selected.

The rest of the paper is organized as follows. Section 2 describes the grid-independent photovoltaic solar energy system model. Sections 3 and 4 describe the PV-priority controller and fuzzy logic controller design, respectively. Finally, Section 5 presents the results and conclusions.

* Corresponding author.
E-mail address: gkumar@ieee.org (G.K. Venayagamoorthy).
2. Grid-independent photovoltaic solar energy system

The complete photovoltaic system model is composed of the PV array, maximum power point tracker, energy dispatch controller, battery charge controller, batteries, inverter, critical loads and non-critical loads. The critical load consists of loads that need to be powered all the time (such as refrigeration and emergency radio communication), while the non-critical load contain items which are non-essential (television, etc.).

In order to simplify the simulation and focus on the controller aspect of this system, all of the supporting system components (such as the inverter, maximum power point tracker, wiring, batteries, etc.), are assumed to operate at 100% efficiency; in the real world however, these components are non-trivial, with several maximum power point tracking algorithms being recently proposed [10–14]. Likewise, new energy storage systems [15,16] may eclipse the traditional lead-acid batteries in terms of performance. Also, the efficiency of the PV array model is taken as 11% to account for various non-optimal conditions (such as array misalignment, dust on the arrays, temperature, etc.). This value is representative of the current commercially available range of efficiencies for PV arrays. Generally, PV panels vary in efficiency from 6% to 30%; although the high efficiency panels are generally reserved for spacecraft usage because of their high radiation tolerances and higher power-to-weight ratio. A rough equivalent to the PV arrays being simulated in this paper would be an array of eight Kyocera KC200GT panels. These panels are over 16% efficient and are non-essential (television, etc.).

Due to insufficient PV energy during winter months in several locations and no PV energy being available at night, a control system is required to decide the amount of energy to be dispatched to the different loads, including charging of the battery. The schematic diagram form of the system studied is shown Fig. 1 (energy flow depicted by arrows).

The system states consist of the PV energy (see nomenclature for complete list of system abbreviations that are used for this paper), critical load, non-critical load and current battery charge. The control signals are energy dispatch to the critical load, energy dispatch to the non-critical load and finally energy dispatch to the battery.

Additionally, a map showing the total solar resource availability for the United State of America is shown in Fig. 2. The average output from the PV array over a year at three different selected locations is shown in Fig. 3 in order of increasing average insolation (Caribou, ME, then Omaha, NE, and finally Las Vegas, NV).

3. PV-priority controller

The standard controller, called the “PV-priority” controller [18], is an energy dispatch controller which always tries to meet the loads (the critical and then the non-critical) before charging the battery. At any one time, if there is not enough energy from the PV array to supply the loads then the remaining is drawn from the battery. If instead there is an excess, then whatever is left over after meeting the loads is dispatched to charge the battery. In this way, the controller attempts to power all loads and charge the battery as best it can, without any considerations given to the time-varying states of the system.

This controller works well when there is an abundant supply of PV energy. However, when there is insufficient PV energy, then the battery will not be fully recharged and the loads will not be met. The weather and use loads are stochastic in nature; therefore there is no one definitive model at all times. PV-priority control is not an optimal strategy. Thus, it is logically to look at intelligent model-free learning methods of controlling such a system optimally. The flowchart for operation of this PV-priority controller is illustrated in Fig. 4.

4. PV fuzzy logic controller design

4.1. Fuzzy logic controller

The PV fuzzy logic controller consists of three main modules: the fuzzification process, the inference engine, and the defuzzification process. The relationship between these three main components is shown in Fig. 5, which shows a block diagram of the
fuzzy logic controller. Each of the main components is discussed below.

4.1.1. Fuzzification process

The input membership functions take the inputs to the controller (after they have been normalized by some value suitable for the membership functions) and produce a degree of membership for each fuzzy set in the membership function. This value is usually designated by the symbol $\mu$. For example, the membership function shown in Fig. 6 takes as inputs the current loads and assigns to that a degree of membership for each fuzzy set in the graph. In this example, “Z” represents the “Zero” fuzzy set, “VS” represents “Very Small”, “S” is “Small”, “M” is “Medium”, “L” is “Large”, and finally “VL” is “Very Large”. In this case, an input of 0.5 would give the following degrees of membership for each fuzzy set:

\[
\begin{align*}
\mu(Z) &= 0, \\
\mu(VS) &= 0, \\
\mu(S) &\approx 0.5, \\
\mu(M) &\approx 0.5, \\
\mu(L) &\approx 0.5, \\
\mu(VL) &= 0.
\end{align*}
\]

For this study, all three input membership functions and two of the output membership functions use the above fuzzy sets (as shown in Fig. 6, in the pre-optimized form). The third output (the output for determining the energy dispatch to the battery) uses a membership function containing only five sets: “LD” for “Large Discharge”, “SD” for “Small Discharge”, “Z” for “Zero”, “SC” for “Small Charge” and finally “LC” for “Large Charge”.

4.1.2. Inference engine

Once the degrees of membership for each fuzzy set have been determined for a particular input, they are presented to the inference engine. The inference engine takes these fuzzy set memberships and determines which rules should be evaluated. A typical fuzzy rule is of the form “If A, B, and C then D, E and F”. As an example, one of the rules for this fuzzy controller might be: If (PV energy is “Large”) and (Current state of charge of the battery is “Large”) and (Current loads are “Large”) then (Energy to the critical load is “Very Large”) and (Energy to the non-critical load is “Medium”) and (Energy to the battery is “Small Charge”).

The rules to be evaluated are selected based on non-zero memberships of the input values. To extend the previous example, the shown rule would only be selected if all of the inputs had a membership value other than 0. Once this rule (and any others meeting these criteria) is found, the output degrees of membership would be asserted according to the membership’s values of the inputs. Once the output degrees of membership are found, they are defuzzified to produce the dispatch signals as explained below.

4.1.3. Defuzzification

Once the degrees of membership of the outputs have been found via the inference engine, the defuzzification process takes these values and translates them into an output dispatch signal. This is done much like the fuzzification process but a reverse process. And, in the case where multiple rules have been asserted (and hence multiple degrees of membership for the outputs), the center of mass of the weighted outputs is found. This output value is then multiplied by a normalizing value to return it to the level of real world outputs.

In addition to the three main modules above, a check on the outputs of the controller is made so that no excess energy is dispatched to any of the three outputs (i.e., the loads could not be over supplied nor could the battery be overcharged). Additionally, checks are put into place to verify that no more energy is being dispatched than is available at any given time.
4.2. Optimized fuzzy logic controller

In order to optimize the fuzzy logic controller, particle swarm optimization (PSO) [19,20] is used to optimize both the membership functions in the fuzzification and defuzzification modules, and rule set of the inference engine. PSO is an iterative algorithm that represents possible solutions to a given problem with a series of multidimensional vectors. Each vector is called a *particle* and contains one complete solution. Each dimension of each particle represents one parameter of a solution to be optimized. In this paper, 30 particles are chosen to represent each possible controller with the following parameters:

Each fuzzy set in each membership function (besides the first and last fuzzy set in each membership function) is represented by three values: the first one for the left-most point, the second one for the right-most point, and the third one for the middle point. The end values had a fixed leg (either the left or right, depending on which end of the membership function the fuzzy set occupies), so they are only specified by two values. Since there are five membership functions with six fuzzy sets (three inputs and two outputs) and one membership function with five fuzzy sets (one output), this equates to 93 parameters just to represent the membership functions.

Each rule is represented by three values (one for each output). Since there are three inputs and each can take on as many as six values, there are 216 rules. Since each is represented by three values, this adds another 648 parameters.

Summing each of these values up, it can be seen that each solution is represented with 741 parameters, this means each PSO particle has 741 dimensions. PSO optimizes these values by using a process based on social interaction, much like a flock of birds or school of fish. In PSO, a collection of particles takes on values that represent a possible solution. As the swarm of particles moves about (according to a defined velocity determined by how well each is doing), the particles’ values change. As they change, a record of each particle’s best position (called *pbest*) is kept as well as the global overall best position (called *gbest*). The equations to determine velocity and position updates are shown below in (1) and (2), respectively. In each, the index *i* ranges over the number of particles.

\[
\text{Velocity}(i) = \frac{1}{c_3} \cdot \text{rand} \cdot \text{Velocity}(i) + \frac{2}{c_3} \cdot \text{rand} \cdot (\text{pbest}(i) - \text{Position}(i)) \quad (1)
\]
\[
\text{Position}(i) = \text{Position}(i) + \text{Velocity}(i) \quad (2)
\]

The quality of solution for each particle is measured by the *fitness function* when evaluated at the particle’s current position. In this paper, the fitness function is given by (3). Each term in (3) is a percentage calculated over the entire year, where $A_1$, $A_2$, and $A_3$ are defined to be 30/23, 15/23, and 13/23, respectively. These values give emphasis to meeting the critical load over the other objectives, and were empirically determined.

\[
\text{Fitness} = A_1 \times (\text{Critical Load Satisfied}) + A_2 \times (\text{Average Battery State Of Charge}) + A_3 \times (\text{Noncritical Load Satisfied}) \quad (3)
\]

![Energy from PV array for (a) Caribou, ME, (b) Omaha, NE, and (c) Las Vegas, NV.](image)
A higher fitness function value (or just called the fitness) results from a better performing individual. As the algorithm progresses, it is expected that the best solution should continue to improve over time (which is shown by an improving fitness over time). The flowchart of the optimization process is shown in Fig. 7.

One modification to the standard PSO algorithm that is made is adding some limit checks on the instantiated individual, and its velocity through space. At no time could the velocity of a particle be greater than 0.1 in any dimension. In the case of the instantiated individual, checks are also put in place to make sure that no
entry in any membership function spans a distance of less than 0.1. Otherwise the membership function entry is widened to a width of 0.1. Also, some checks are put into place to verify that the entire width of the input space is mapped to a fuzzy set. That is to say that no possible input could fall outside of a membership function entry. Lastly, checks are placed on each dimension of the final instantiated object to always be within a boundary that made physical sense. That is, all dimensions having to do with optimizing the membership functions had to be within a range of 0–1 (inclusive); and all dimensions having to do with optimizing rules had to be within a similar range so that all rules associated with an instantiated FLC could be mapped to an appropriate value.

5. Results and discussion

5.1. Results

The results from this study is much improved from the authors’ previous results of optimizing a fuzzy logic controller [8], in which only the membership functions were optimized. In this study, the FLC’s rule base is also simultaneously optimized such that the performance of the entire FLC is superior to that of the FLC where just the fuzzy membership sets were optimized. Also in this study, one FLC is optimized using PSO with data from one city (Caribou), and then the performance of this energy dispatch controller is tested at two other locations (Omaha and Las Vegas). The training city is selected because it receives less insolation that the other two cities. After running the PSO optimization for 50 iterations, the final gbest fitness is 2.086250. This can be seen from Fig. 8 showing how the best particle’s (gbest) fitness increases throughout the simulation:

The actual results (as well as the results of the PV-priority and un-optimized fuzzy logic controllers) for all three cities are listed in Tables 1–3. The Relative Performance row is calculated by evaluating (4) for each controller and city combination. This gives an objective method of comparing relative controller performance.

Relative performance = \( \frac{1}{365} \sum_{day=1}^{365} \left( \frac{EPV}{P_{\text{max}}} - \frac{EPV}{P_{\text{wasted}}} \right) \times \frac{24 \times (A_1 + A_2 + A_3)}{\sum_{hour=1}^{24} (EPV - EPV_{\text{max}})} \) (4)

The data from Table 1 shows that the FLC that had both the membership functions and fuzzy rule set optimized performed 5.22% better in energy usage than the FLC from [8] that only had the membership functions optimized, and also outperformed the initial un-optimized FLC and PV-priority scheme. Table 2 shows the performance of the optimized FLC for the average insolation case of Omaha, NE. Again in this comparison the FLC that had both the membership functions and fuzzy rule set optimized performed better by 26.13% in energy usage than un-optimized FLC. Finally, Table 3 shows the controller performance for the case of high insolation in Las Vegas, NV. Again, the FLC with the optimized membership functions and rule set is the best performer and is 37.28% better in energy usage than un-optimized controller. Furthermore, the solar insolation in Las Vegas is much more than in Caribou, ME and therefore, the FLC efficiency can be still improved if the FLC is fine tuned custom to Las Vegas, and likewise to each location.
These results show that optimizing the membership functions as well as the rule set allows the fuzzy logic controller to perform significantly better than the un-optimized fuzzy controller and the PV-priority controller. Fig. 9 shows the optimized membership functions for the current load input, while Fig. 10 shows the optimized membership functions for energy dispatched to the critical load output. Figs. 11 and 12 show the unmet critical and non-critical loads, respectively, using data from the Caribou area and the optimized FLC. It is shown that the optimized FLC is able to meet the vast majority of the critical load and most of the non-critical loads, except into the early winter where the insolation falls to inadequate levels. Of course, all the unmet load demand can be supplied with energy by increasing either the size of the battery capacity or increasing the size of the PV output or doing both. Figs. 13–15 show the annual battery state of charge for the Caribou area using the PV-priority controller, initial un-optimized FLC, and

| Table 1 | Summary of controller performance for the Caribou, ME area. |
|------------------------------------------|
| PV-priority | Un-optimized fuzzy | Optimal fuzzy | Optimized membership functions only [8] |
| Percentage critical load met | 84.22 [914.8 kW h] | 93.04 [1011.0 kW h] | 95.51 [1038.0 kW h] | 92.97 [1010.4 kW h] |
| Percentage non-critical load met | 77.21 [778.4 kW h] | 32.16 [324.2 kW h] | 61.8 [623.0 kW h] | 56.07 [565.2 kW h] |
| Average percentage of battery charge | 63.87 [22.07 kW h] | 76.58 [26.47 kW h] | 75.3 [26.02 kW h] | 80.21 [27.72 kW h] |
| Total energy supplied (kW h) | 1715.27 | 1361.67 | 1687.02 | 1603.32 |
| Relative performance | 0.6168 | 0.5632 | 0.6921 | 0.6498 |

| Table 2 | Summary of controller performance for the Omaha, NE area. |
|------------------------------------------|
| PV-priority | Un-optimized fuzzy | Optimal fuzzy |
| Percentage critical load met | 88.66 [963.0 kW h] | 97.08 [1054.0 kW h] | 99.00 [1075.0 kW h] |
| Percentage non-critical load met | 85.85 [865.3 kW h] | 36.03 [363.3 kW h] | 71.44 [720.2 kW h] |
| Average percentage battery charge | 68.98 [23.80 kW h] | 84.30 [29.10 kW h] | 84.36 [29.20 kW h] |
| Total energy supplied (kW h) | 1852.30 | 1446.40 | 1824.40 |
| Relative performance | 0.6170 | 0.5568 | 0.6972 |

| Table 3 | Summary of controller performance for Las Vegas, NV area. |
|------------------------------------------|
| PV-priority | Un-optimized fuzzy | Optimal fuzzy |
| Percentage critical load met | 98.3 [1068.0 kW h] | 99.62 [1082.0 kW h] | 100.00 [1086.0 kW h] |
| Percentage non-critical load met | 97.46 [982.5 kW h] | 35.55 [358.4 kW h] | 89.83 [905.6 kW h] |
| Average percentage battery charge | 82.06 [28.40 kW h] | 97.08 [33.60 kW h] | 92.34 [31.90 kW h] |
| Total energy supplied (kW h) | 2078.90 | 1474.00 | 2023.50 |
| Relative performance | 0.5981 | 0.4415 | 0.6142 |

**Fig. 9.** Optimized membership function for current load input.  
**Fig. 10.** Optimized membership function for energy to critical load output.
optimized FLC, respectively. Fig. 16 shows the relative performance of the controllers at each city. Finally, Table 4 shows a sampling of rules from the fuzzy rule base before and after optimization via PSO. It can be seen that many of the rules are changed after
optimization, especially those with higher rule numbers as these rules deal with system conditions that generally have more energy available to dispatch, and thus have more options to optimize.

6. Conclusions

The development of an optimal fuzzy logic controller for a grid-independent photovoltaic system has been presented using particle swarm optimization. PSO is able to optimize the membership functions and develop optimal rules for a fuzzy system based controller. Results show more of the critical loads are met most of the time (around 2.5% more in the case of Caribou, ME) after optimizing the FLC, and around 11.3% more when comparing the PV-priority controller to the optimized FLC (again, in the Caribou, ME area). Additionally, the number of complete discharges of the battery is lowest for the optimized FLC (by visual inspection of Figs. 13–15), which serves to lengthen the life of the battery as other researchers have found as well. In this study, the average battery state of charge for each controller can be compared and the expected battery life increase approximated. For the case of Caribou, ME, the average battery state of charge is 75.3% (this leads to an average depth of discharge (DOD) of 24.7%) for the optimal FLC, but only 63.87% (36.13% average DOD) for the PV-priority controller. This leads to approximately a 5% increase in battery life expectancy. Of course, a more realistic evaluation depends on hour-by-hour evaluation of the battery state of charge (not just average depth of discharge).

As a result, it is possible that a smaller (and cheaper) overall PV system utilizing such an optimal energy dispatch controller would be suitable for meeting the same loads as a larger more expensive system not using an optimal controller. The maintenance and replacement cost of the battery is also reduced by approximately the same proportions as the life expectancy increase, since the battery will need to be replaced less often.

Acknowledgement

The authors gratefully acknowledge the support of the U.S. Department of Education Grant: GAANN #P200A070504 and NSF grant EFRI #0836017.

References


Table 4
Sampling of rules from rule base before and after PSO optimization (rules modified by PSO are bolded).

<table>
<thead>
<tr>
<th>Rule number</th>
<th>PV energy</th>
<th>Battery state of charge</th>
<th>Load</th>
<th>Energy to critical load</th>
<th>Energy to non-critical load</th>
<th>Energy to the battery</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>z</td>
<td>z</td>
<td>z</td>
<td>z → z</td>
<td>z → z</td>
<td>z → z</td>
</tr>
<tr>
<td>2</td>
<td>z</td>
<td>z</td>
<td>vs</td>
<td>z → z</td>
<td>z → z</td>
<td>z → z</td>
</tr>
<tr>
<td>3</td>
<td>z</td>
<td>s</td>
<td>z</td>
<td>z → z</td>
<td>z → z</td>
<td>z → z</td>
</tr>
<tr>
<td>4</td>
<td>z</td>
<td>m</td>
<td>z</td>
<td>z → z</td>
<td>z → z</td>
<td>z → z</td>
</tr>
<tr>
<td>5</td>
<td>z</td>
<td>l</td>
<td>z</td>
<td>z → z</td>
<td>z → z</td>
<td>z → z</td>
</tr>
<tr>
<td>6</td>
<td>z</td>
<td>vl</td>
<td>vs</td>
<td>vs → vs</td>
<td>vs → vs</td>
<td>vs → vs</td>
</tr>
<tr>
<td>7</td>
<td>z</td>
<td>vs</td>
<td>z</td>
<td>z → z</td>
<td>z → z</td>
<td>z → z</td>
</tr>
<tr>
<td>8</td>
<td>z</td>
<td>vl</td>
<td>m</td>
<td>z → m</td>
<td>z → m</td>
<td>z → m</td>
</tr>
<tr>
<td>9</td>
<td>z</td>
<td>vs</td>
<td>vl</td>
<td>vl → m</td>
<td>vl → m</td>
<td>vl → m</td>
</tr>
<tr>
<td>10</td>
<td>z</td>
<td>vl</td>
<td>m</td>
<td>m → m</td>
<td>m → m</td>
<td>m → m</td>
</tr>
</tbody>
</table>

...


