Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

New maximum power point tracking for wind turbines

Alnufaie Lafi *

Department of Electrical Engineering, College of Engineering, Shaqra University, Shaqraa, Saudi Arabia

ARTICLE INFO

Article history:
Received 25 April 2019
Received in revised form
6 August 2019
Accepted 7 August 2019

Keywords:
Maximum power point tracking
Fuzzy controller
Wind speed
Wind energy

ABSTRACT

In this paper, we deal with the maximum power point tracking problem for wind generator in a standalone installation. The aim is to develop a new algorithm to get the value of optimal power. So, we intend to use fuzzy logic to exploit its robustness in presence on not certainties and its simplicity of design. In order to improve human expertise, we aim to use two evolutionary algorithms as Particle Swarm Optimization and Genetic Algorithms. Lots of simulation results are introduced to show the obtained improvement.

1. Introduction

An excellent way to get green energy is Wind turbines. In spite of the high cost associated, a lot of time, energy and money has been invested to reduce that cost, and make them more affordable, and more efficient in their function. A lot of parameters affect the amount of energy produced, so an adaptation unit based on a DC-DC converter is needed between the turbines and the load. The Maximum Power Point (MPP) is tracked using either a direct or an indirect approach (Hui and Bakhshai, 2008; Moor and Beukes, 2004; Xia et al., 2011; Yaoqin et al., 2002).

The indirect method is unsuitable for our aim because it needs large amounts of memory for the storage of environmental data, which are easily to get. However; the direct method has no need for information about climatic conditions (Ram et al., 2017; Zou et al., 2011), and it based on direct measures of power from the generator. This action is usually done with Perturb and Observe (Camblong et al., 2006; Meghni et al., 2017), through perturbation of the duty cycle to move the function point on the generator characteristics curve i.e. the voltage is adjusted by a small amount and the power is measured; then, provided that the power increases, further adjustments are made in the same direction until power no longer increases. Unfortunately, this method creates a constant oscillation around the maximum power point.

The fuzzy logic adoption gives us a suitable solution to many of the difficulties presented by P&O, thus creating a system is so only fast in calculation, but more robust, yielding more precise results. However human expertise is still an essential need for setting the rules (Ahmed et al., 2008; Belmokhtar et al., 2014; Eltamaly and Farh, 2013). To solve this problem, we intend to use two optimization algorithms: particle swarm (PSO) and genetic algorithms (GA). Several simulation results are introduced to show the improvement due to using evolutionary algorithms.

2. System description

Fig. 1 shows the block diagram of the wind energy system which used in our research, where a PMSG is driven by a Variable-speed wind turbine to feed the extracted power from wind resources to the dc-link, using a rectifier and DC-DC converter. This DC-DC converter is used to get the largest amount of energy show Table 1.

3. Maximum power point tracking approach

Based on Fig. 2, we notice that each wind speed, is found in a specific point in the wind generator output power versus rotating-speed characteristic where the output power reach to its maximized point. Its control loads result in a variable-speed wind generator operation, this maximum power always comes from the wind (MPPT control).

In the next, we propose to approaches to attain the maximum power point.

* Corresponding Author.
Email Address: lalnufaie@su.edu.sa
https://doi.org/10.21833/ijaas.2019.10.012
Corresponding author's ORCID profile:
https://orcid.org/0000-0003-2453-2637

3.1. Fuzzy logic controller

It was developed by Zadah (1965), it is mathematical form of reasoning depend on degrees of truth. Instead of traditional 'Boolean logic' that only allows for 'True' or 'False, using fuzzy logic makes space for partial truths, for example 'quite true' or 'quite false'. It is possible to imitate human reasoning, by interpreting the knowledge of human experts into rules that can be interpreted by the computer, managing it to issue commands based on this information.

Table 1: Wind turbine characteristics

Power range	1kw
Start wind	3m/s
Nominal power	1000w
Rotor dimension	2.5m
Blade control	fixed
Nominal speed	11m/s

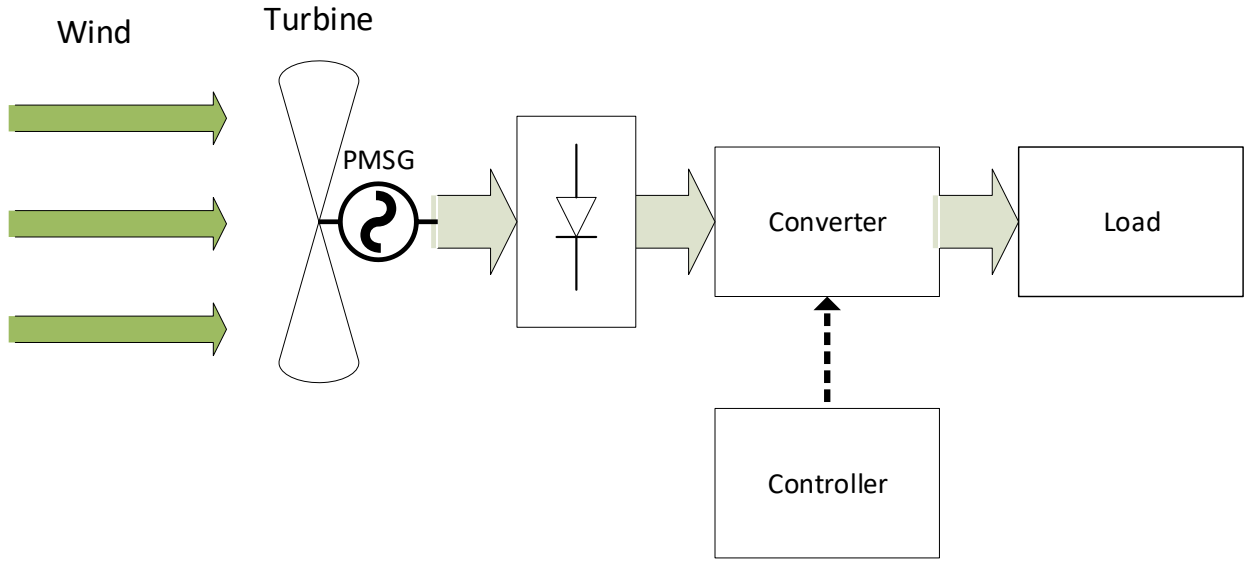


Fig. 1: System architecture

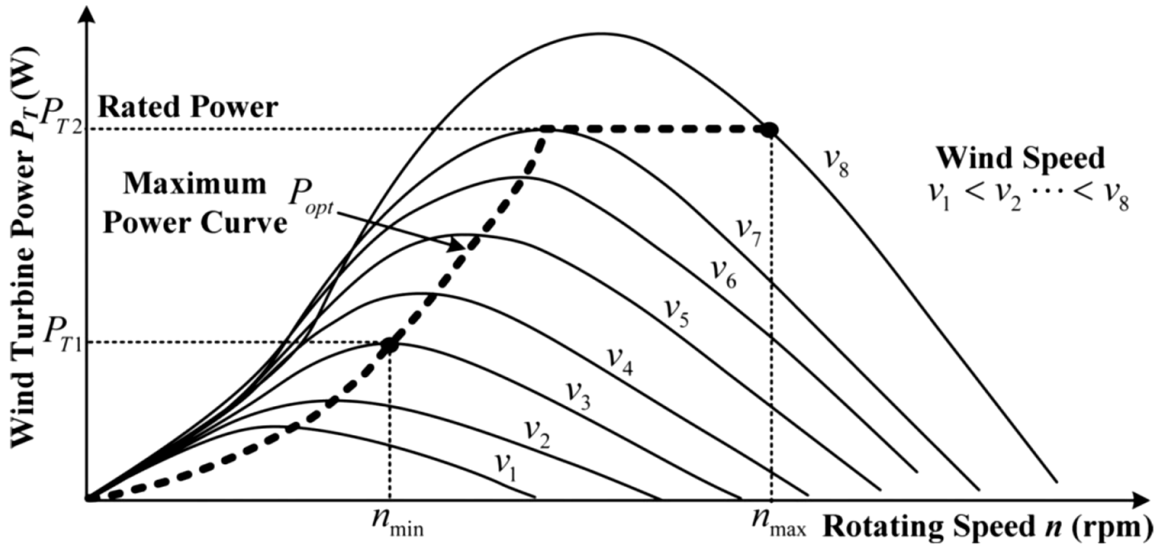


Fig. 2: Wind generator power curves at various wind speeds

However, it is only possible to use fuzzy logic if the exact functioning of the system used is understood, meaning expert knowledge is basic when conceiving a flexible and robust controller. The FLC system consisted of three steps: They are the fuzzification, Inference and defuzzication:

- The Fuzzification: The numeric is transformed into thee linguistic
- Inference: Human expert rules are applied
- Defuzzification: Numeric is switched to linguistic.

3.1.1. Fuzzification

This is the process of moving from real to Fuzzy, for which the enrollment degree of an input changeable must be persistent, for a membership function. The simplest of the many types of function is the triangular form, where two functions will be effective for each input at any moment given to it. In this method the calculation times of all parameters are limited, and the command is simplified. The number of functions used is important, a higher

number of functions means the controller will be more sensitive and have a higher set of rules. Normalizing the discourse universe by using the interval -1, +1 further simplifies the system, with the extreme membership functions being fixed to these limits Fig. 3 (Kazmi et al., 2010; Kumar and Chatterjee, 2016; Zeng et al., 2008).

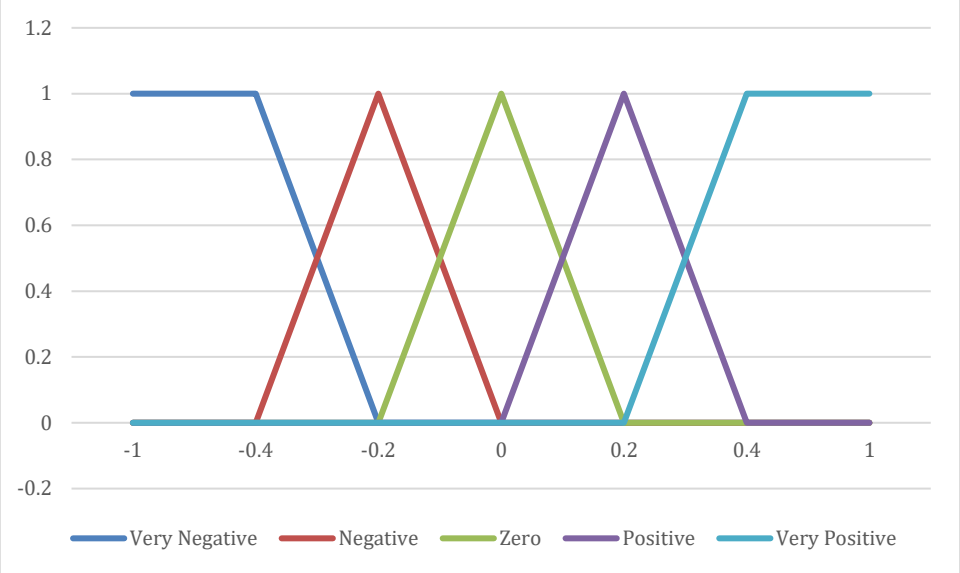


Fig. 3: Membership function

3.1.2. Inference

This type has to be set before the rules; in this case it is Takagi-Sugeno. The human expertise is translated to set the rules, according to the variation of the maximum power point. Using this type of system allows a crisp output, and equipment calculation, making the both rapid and precise command show Table 2 (Lin et al., 2010; Varzaneh et al., 2014).

Table 2: Inference parameters

		d^2P/dV^2		
		Negative	Zero	Positive
dP/dV	Negative	Decrease V	Decrease V	Null
	Zero	Null	Null	Null
	Positive	Null	Increase V	Increase V

3.1.3. Defuzzification

It describes the step of moving f the Fuzzy to the real domain. The membership function introducing our output is used to calculate a number of values for the final output. The most common method used to find this value simply and rapid the Gravity center, which calculates an average estimate of the value. Furthermore, we can optimize the fuzzy controller with more fine-tuning to improve MPP tracking, for example by using a Genetic algorithm, from the evolutionary family. It can be used to appoint and form the parameters of the membership tasks in the discourse universe. Optimization algorithms includes:

3.1.4. Particle swarm optimization

Particle swarm optimization or PSO, is an imaginary development technique that developed by Eberhart and Kennedy (1995). Population based, it is stimulated by the social behavior of bird-flocking, where the population is initialized with a collection of irregular solutions and then looks for optima by adding generations. PSO has some stark differences when compared with GA, for example it hasn't any evolutionary operatives, such as crossover or variation. In PSO the wide range of possible solutions are called particles, which fly through the problem space by following the current optimum particles.

There are some benefits in using PSO instead of GA. There are not many parameters to adapt and it has been applied successfully in many areas, from function optimization to neural network training (Lin et al., 2010).

3.2. PSO algorithm

In PSO each single solution or particle in the search space has a fitness value, which is evaluated by the fitness function to be optimized, and a velocity, which directs the flying of the particle. PSO is initialized with a group of random particles (solutions), and then searches for optima by updating generations. In every iteration, each particle is updated by following two best values.

The first one is the best solution (fitness), it has achieved so far. The fitness value is also stored; this value is called Pbest.

The other best value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the population. This best value is a global best, and is called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called l best. After finding the two best values the particle

updates its velocity and position with the following equations:

$$V_i = V_i + 2 * rand(\quad) * (P_{best} - X_i) + 2 * rand(\quad) * (g_{best} - X_i) \quad (1)$$

$$X_i = X_i + V_i \quad (2)$$

where;
X_i: Position vector
V_i: Velocity vector
P_{best}: Each particle has even updated to their best encounter position
g_{best}: Any particle has even updated to their best encounter position

The Particle Swarm Optimization algorithm is shown in Table 3.

Table 3: PSO

For each particle
Initialize particle
END
Do
For each particle
Calculate fitness value
If the fitness value is better than the best fitness value (P_{best}) in history
Set current value as the new P_{best}
END
Choose particle with the best fitness value of all the particles as the g_{best}
For each particle
Calculate particle velocity according to first equation
Update particle position according to the second equation
END
While maximum iterations or minimum error criteria isn't attained
Particles velocities on each dimension are clamped to a maximum velocity V_{max}, which is a parameter specified by the user, then the velocity on the dimension is limited to V_{max}.

3.3. Genetic algorithms

Genetic algorithms were inspired by the work of Charles Darwin, who gradually developed the theory over time through survival of the fittest, by passing on whatever characteristics that best adapted to their environment. In this way a feature that occurred as a result of an unplanned genetic mutation may gradually become more prominent in a population, if it was advantageous to the individual that first had it. This is where GAs come in, the algorithm imitates this random mutation element, working to create better adapted individuals that will help the system to work better. The working of the algorithm is depending on a set of chromosomes that are encoded using a string of possible values, which can be made up of either real values or binary strings (Damousis et al., 2002). A population is consisting of a group of prepared potential solutions. Every member of the group is an individual, encoded using the parameter values from the chromosomes, and applied to the problem. The aim is to achieve an optimal or near-optimal solution, and this determines the fitness. To create the next population, the individuals are selected based on their fitness rate, thereby imitating nature's preference for a better-adapted individual in the wild. Random mutations must also be included, but kept at a low probability, allowing the creation of new individuals and new characteristics.

3.3.1. Selection algorithm

Based on a stochastic sampling with replacement, this algorithm establishes a fitness rate for every chromosome. An individual is selected in proportion to the fitness rate and using a random number generator, i.e., a high fitness rate means a high selection rate and vice versa. This technique has its limits, there is no guarantee that fitter individuals will necessarily be represented in the following generation. This problem may be remedied by the use of another algorithm, which specifically identifies above average chromosomes, thus ensuring those individuals will be selected as their fitness allows (Lin et al., 2010; Thongam and Ouhrouche, 2011; Zeng et al., 2008).

3.3.2. Crossover algorithms

Mixing the strings of two individuals creates new ones. Many algorithms pair individuals for mating, the most basic of which is the single point crossover shown in Table 4.

Table 4: Single point crossover

Parent 1	1	0	1	1	1	1	0	0
Parent 2	1	1	0	1	0	0	1	0
Child 1	1	0	1	1	0	0	1	0
Child 2	1	1	0	1	1	1	0	0

A multiple crossover point works by selecting multiple crossover points, as shown in Table 5.

Table 5: Multiple crossover point

Parent 1	1	0	1	1	1	1	0	0
Parent 2	1	1	0	1	0	0	1	0
Child 1	1	0	0	1	1	1	1	0
Child 2	1	1	1	1	0	0	0	0

Based on a random string of the same length as the parent, a uniform crossover is a more complicated crossover algorithm. It functions in this way: if the bit of the string is 1, it takes a bit from the first parent and gives it to the first child, and if it is zero it gives it to the second child, as shown Table 6.

This is the best performing algorithm shown in Table 6.

Table 6: Uniform crossover point

Parent 1	1	0	1	1	1	1	0	0
Parent 2	1	1	0	1	0	0	1	0
Child 1	1	0	0	1	1	1	1	0
Child 2	1	1	1	1	0	0	0	0

3.3.3. Mutation algorithms

A random number generator creates new individuals with new characteristics through mutation. The mutation method used is simple: for each bit, we generate a random number, and if it is less than the specified mutation probability, flip the bit, if it is 1, change it to zero, and vice versa. It is important to keep the mutation rate low, and constant throughout the lifetime of the GA, since, as in nature, only a small percentage of mutations will be beneficial.

3.3.4. Crossover and mutation rate set-up

There are two different techniques to set the crossover and mutation rates, the first one is using the standard parameters, the most common of which (Zeng et al., 2008) is shown in the Table 7 and the second most common parameter from (Belmokhtar et al., 2014; Lin et al., 2010) is shown in Table 8.

Table 7: Standard parameters for GA

Population size	50
Number of generations	1,000
Crossover type	Typically two point
Crossover rate	0.6
Mutation types	Bit flip
Mutation rate	0.001

Table 8: Standard parameters for GA

Population size	60
Number of generations	Not specified Crossover
Crossover type	Typically two point
Crossover rate	0.9
Mutation types	Bit flip
Mutation rate	0.01

Another technique uses a generic scheme for adapting the crossover and mutation probabilities, in which they are altered as a result of the offspring evaluation. This too has proved efficient and improved the performance of the algorithm (Zeng et al., 2008). The algorithm was run more than 200 times for this work, and we changed the parameters until we had the best result using an 85% crossover rate, and 7% mutation rate, then these parameters were fixed.

4. Set-up simulation and results

4.1. System set-up

4.1.1. Optimization using GA and PSO

The algorithms are introduced in different ways to the Fuzzy Logic, where there is a fixed point simultaneously controlling the size and width of all the membership functions. Secondly, multiple points are used, giving more freedom for the membership functions to move. As shown in the flowchart describing the genetic algorithm execution below, in each iteration the simulation is run for every member of the population, and the energy calculated from the surface. As, we will see later in experimentation and result, different parameters must be considered for the GA and the Fuzzy Logic Controller shown in Fig. 4.

The second chart explains the introduction of PSO to our system with the same set-up as for the GA, and multiple parameters are set. The fact that PSO does not need as much computation as the GA is the main difference between the two set-ups shown in Fig. 5.

4.2. Simulink

The simulations were achieved using Simulink library in MATLAB environment. The system wind turbine diagram is given in Fig. 6.

In order to control the DC-DC converter we should use a fuzzy system with two inputs $d^2P/dV^2$and d^2P/dV^2 and the variation of the duty cycle of the converter ΔD as output. We should think about five fuzzy membership functions for each input, which gives us 25 fuzzy rules summarized in Table 9.

In order to test the robustness of the proposed algorithms, we think about the wind and load variations. The simulation results are given in Fig. 7. It should be noted that the system follows the reference power (theoretical maximum power) but only reaches 87.23% of the production capacity.

4.3. Approach 1: Optimization of fuzzy system with GA

In this approach, we used two methods to optimize the system, with genetic algorithms. The first method consists in using a point of the first membership function to move and thus to modify the width of the other membership functions by calculating all the other points of the functions with respect to the movement of the latter. Indeed, the membership functions used intersect at the same degree of belonging and are symmetrical with respect to zero, the calculation of the motion of the point allows us to keep this same configuration, and thus to deduce the position of the other vertices. For optimization, we have treated the inference table via our genetic algorithm. Since our fuzzy system uses two inputs and 25 fuzzy rules, we will have 27 variables to optimize in this method.

The second method consists of taking 5 points which represent the characteristic points of each membership function. Since the variation of these points can lead to redundancy between fuzzy sets and loss of information, we have imposed that the point of intersection (degree of belonging) between two adjacent functions is fixed.

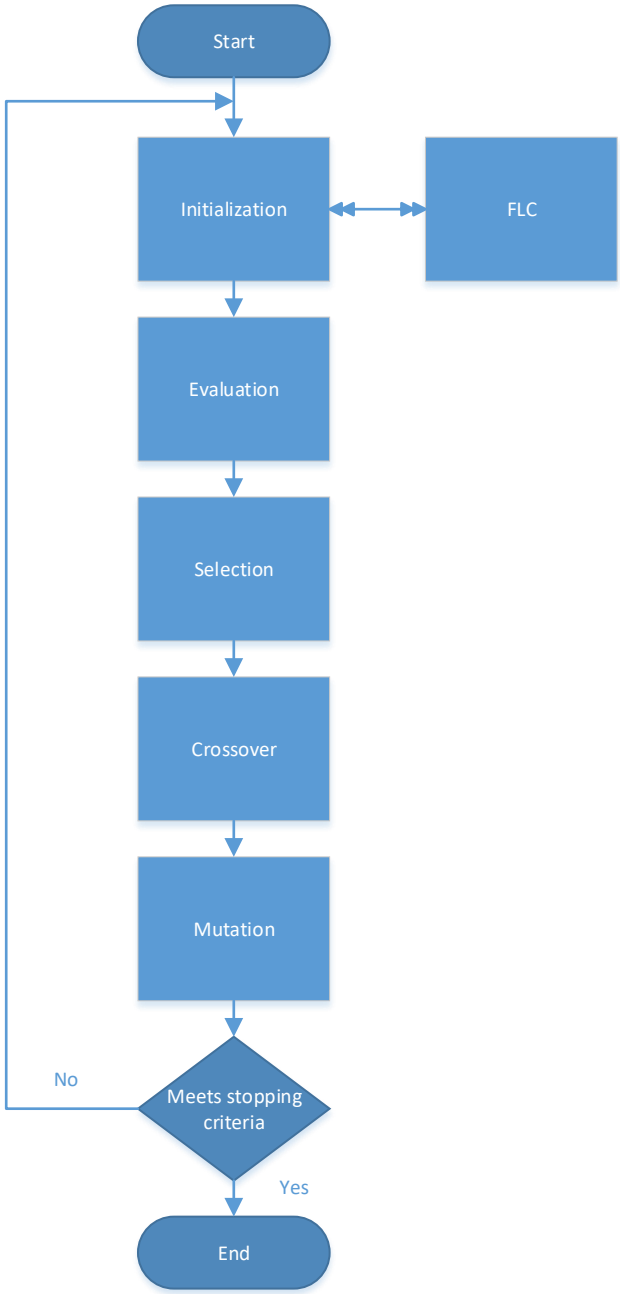


Fig. 4: GA and fuzzy systems flow chart

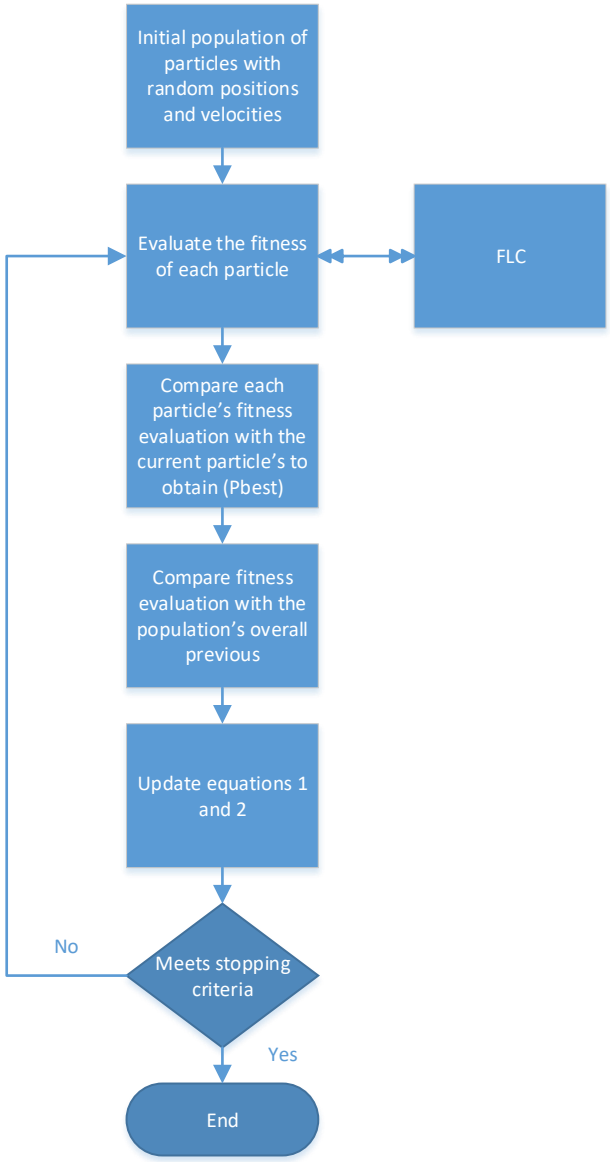


Fig. 5: PSO algorithm flow chart

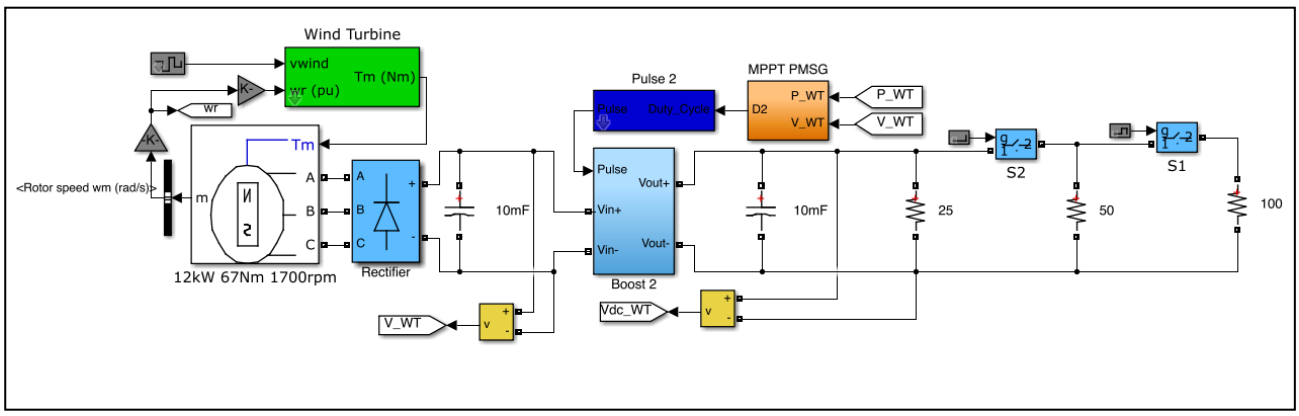


Fig. 6: Simulation of a wind turbine system

Table 9: Table of inference

d^2P/dV^2 / dp/dv		d^2P/dV^2				
		Very Negative	Negative	Zero	Positive	Very Positive
	Very Negative	+3%	+3%	+1%	0%	0%
	Negative	+3%	+1%	+1%	0%	0%
	Zero	0%	0%	0%	0%	0%
	Positive	0%	0%	-1%	-1%	-3%
	Very positive	0%	0%	-1%	-3%	-3%

This choice gives us ten points to manipulate by the genetic algorithm. As in the previous case, we also integrated the rules in our algorithm to optimize them. In total, we will have 45 parameters to optimize. Such an increase will allow us to refine our results and it has no effect on the implementation in real time since the optimization is done off-line shown in Table 10.

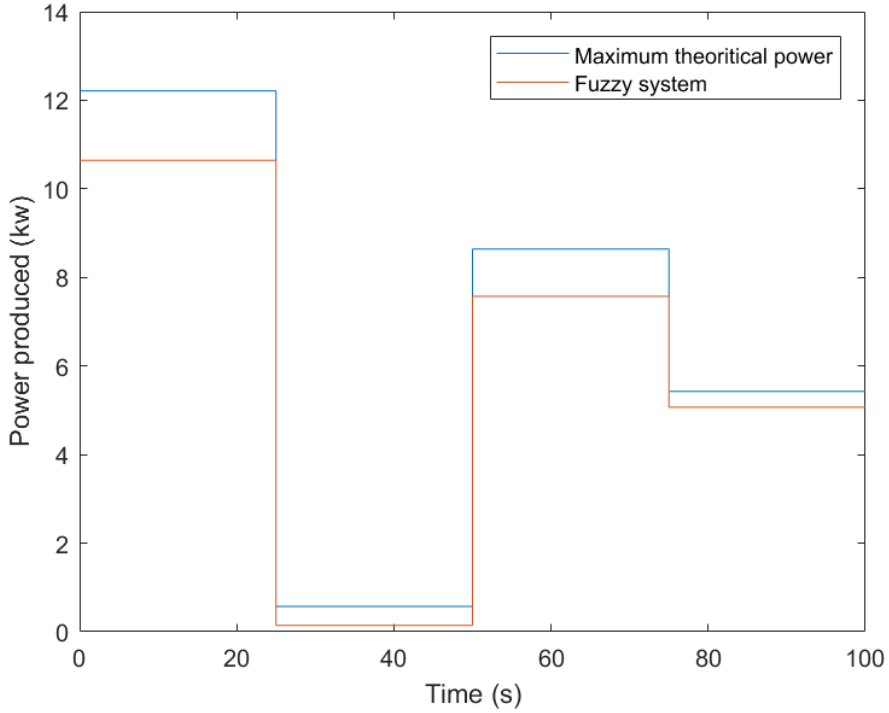


Fig. 7: Results with type 1 fuzzy system

Table 10: GA parameters

	No. of iterations	Population	No. of Variables	Probability of crossover	Mutation rate
Approach 1 method 1	1000	150	27	0.8	0.02
Approach 1 method 2	1000	300	35	0.8	0.02

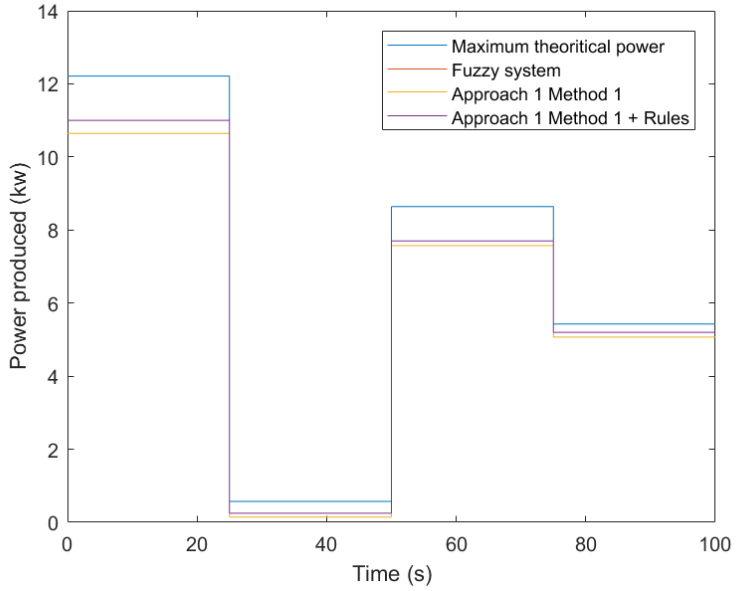
Fig. 8 displays the simulation results. We note that the use of genetic algorithms allows us to improve the efficiency of our installation, especially the second method since we use more parameters and therefore better accuracy. So, the first method allows us to reach 90.2% and the second method 90.95% which represents an increase of 2.97% and 3.72% respectively. In spite of this small increase relatively, it is important because the overall efficiency of a wind turbine does not exceed 70%.

4.4. Approach 2: Optimisation of Fuzzy system using PSO

Similarly, the way that we used genetic algorithms to optimize the fuzzy system, we will use the PSOs to try to improve the performance of our wind system. The simulation parameters are given in the following Fig. 9.

5. Conclusion

A lot of approaches were done to maximize the production of a wind system. To improve the power production, we should use the genetic algorithms and the PSO. We can easily see from the results that the particle swarm optimization can converge toward the best solution in a shorter time, even delivering almost the same results. The PSO displays that it achieves its best solution needing less repeat, so it shows a shorter processing time. On the other side the GA indicates its accuracy and the fact that it has a higher robustness value. In next works, we will do the true time implementation to confirm the results we get by simulation.



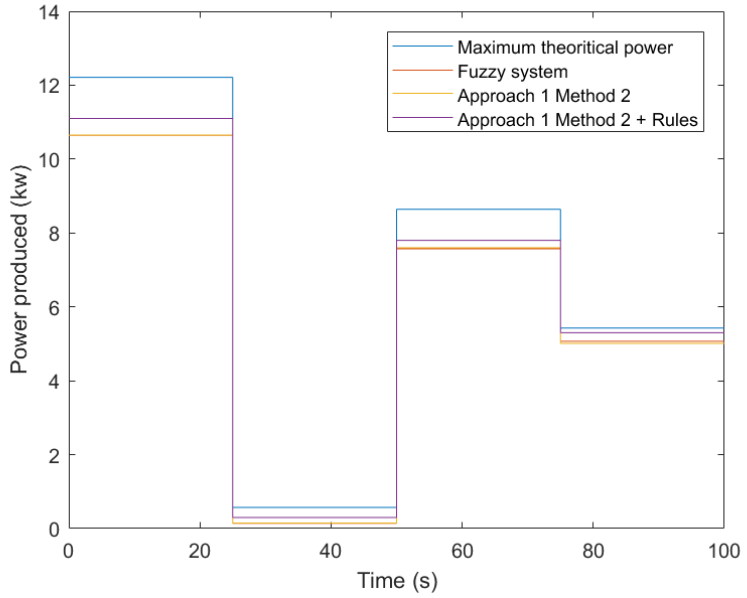
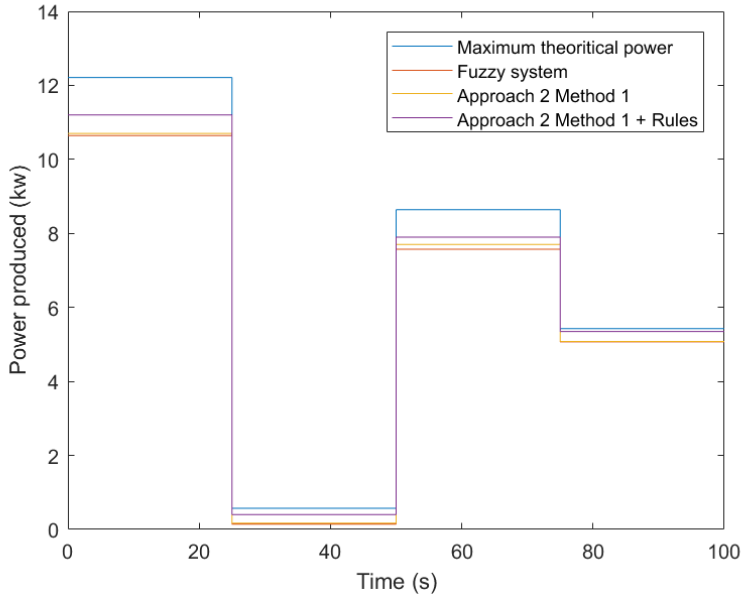


Fig. 8: Results using GA



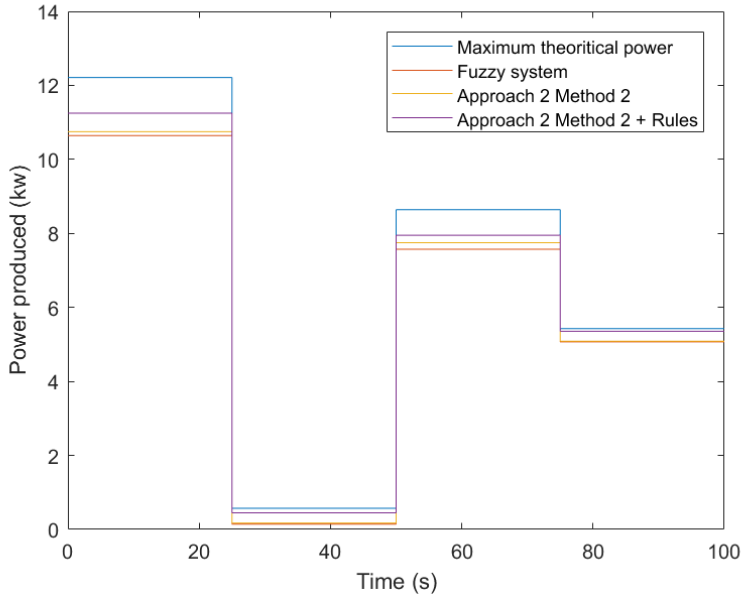


Fig. 9: PSO results

Compliance with ethical standards

Conflict of interest

The authors declare that they have no conflict of interest.

References

Ahmed NA, Miyatake M, and Al-Othman AK (2008). Power fluctuations suppression of stand-alone hybrid generation combining solar photovoltaic/wind turbine and fuel cell systems. Energy Conversion and Management, 49(10): 2711-2719. https://doi.org/10.1016/j.enconman.2008.04.005

Belmokhtar K, Doumbia ML, and Agbossou K (2014). Novel fuzzy logic based sensorless maximum power point tracking strategy for wind turbine systems driven DFIG (doubly-fed induction generator). Energy, 76: 679-693. https://doi.org/10.1016/j.energy.2014.08.066

Camblong H, de Alegria IM, Rodriguez M, and Abad G (2006). Experimental evaluation of wind turbines maximum power point tracking controllers. Energy Conversion and Management, 47(18-19): 2846-2858. https://doi.org/10.1016/j.enconman.2006.03.033

Damousis IG, Satsios KJ, Labridis DP, and Dokopoulos PS (2002). Combined fuzzy logic and genetic algorithm techniques-Application to an electromagnetic field problem. Fuzzy Sets and Systems, 129(3): 371-386. https://doi.org/10.1016/S0165-0114(01)00137-3

Eberhart R and Kennedy J (1995). Particle swarm optimization. In the Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 4: 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

Eltamaly AM and Farh HM (2013). Maximum power extraction from wind energy system based on fuzzy logic control. Electric Power Systems Research, 97: 144-150. https://doi.org/10.1016/j.epsr.2013.01.001

Hui J and Bakhshai A (2008). A new adaptive control algorithm for maximum power point tracking for wind energy conversion systems. In the 2008 IEEE Power Electronics Specialists Conference, IEEE, Rhodes, Greece: 4003-4007. https://doi.org/10.1109/PESC.2008.4592580

Kazmi SMR, Goto H, Guo HJ, and Ichinokura O (2010). Review and critical analysis of the research papers published till date on maximum power point tracking in wind energy conversion system. In the 2010 IEEE Energy Conversion Congress and Exposition, IEEE, Atlanta, USA: 4075-4082. https://doi.org/10.1109/ECCE.2010.5617747

Kumar D and Chatterjee K (2016). A review of conventional and advanced MPPT algorithms for wind energy systems. Renewable and Sustainable Energy Reviews, 55: 957-970. https://doi.org/10.1016/j.rser.2015.11.013

Lin WM, Hong CM, and Cheng FS (2010). Fuzzy neural network output maximization control for sensorless wind energy conversion system. Energy, 35(2): 592-601. https://doi.org/10.1016/j.energy.2009.10.030

Meghni B, Dib D, and Azar AT (2017). A second-order sliding mode and fuzzy logic control to optimal energy management in wind turbine with battery storage. Neural Computing and Applications, 28(6): 1417-1434. https://doi.org/10.1007/s00521-015-2161-z

Moor GD and Beukes HJ (2004). Maximum power point trackers for wind turbines. In the 2004 IEEE 35th Annual Power Electronics Specialists Conference, IEEE, Aachen, Germany, 3: 2044-2049. https://doi.org/10.1109/PESC.2004.1355432

Ram JP, Rajasekar N, and Miyatake M (2017). Design and overview of maximum power point tracking techniques in wind and solar photovoltaic systems: A review. Renewable and Sustainable Energy Reviews, 73: 1138-1159. https://doi.org/10.1016/j.rser.2017.02.009

Thongam JS and Ouhrouche M (2011). MPPT control methods in wind energy conversion systems. In: Carriveau R (Ed.), Fundamental and advanced topics in wind power: 339-360. BoD–Books on Demand, Norderstedt, Germany.

Varzaneh SG, Rastegar H, and Gharehpetian GB (2014). A new three-mode maximum power point tracking algorithm for doubly fed induction generator based wind energy conversion system. Electric Power Components and Systems, 42(1): 45-59. https://doi.org/10.1080/15325008.2013.846437

Xia Y, Ahmed KH, and Williams BW (2011). A new maximum power point tracking technique for permanent magnet synchronous generator based wind energy conversion system. IEEE Transactions on Power Electronics, 26(12): 3609-3620. https://doi.org/10.1109/TPEL.2011.2162251

Yaoqin J, Zhongqing Y, and Binggang C (2002). A new maximum power point tracking control scheme for wind generation. In the International Conference on Power System Technology, IEEE, Kunming, China, 1: 144-148. https://doi.org/10.1109/ICPST.2002.1053521

Zadah LA (1965). Fuzzy Sets. Information and Control Volume 8(3): 338-353. https://doi.org/10.1016/S0019-9958(65)90241-X

Zeng Q, Chang L, and Shao R (2008). Fuzzy-logic-based maximum power point tracking strategy for PMSG variable-speed wind turbine generation systems. In the 2008 Canadian Conference on Electrical and Computer Engineering, IEEE, Niagara Falls, Canada: 405- 410. https://doi.org/10.1109/CCECE.2008.4564566

Zou Y, Elbuluk M, and Sozer Y (2011). Stability analysis of maximum power point tracking (MPPT) method in wind power systems. In the 2011 IEEE Industry Applications Society Annual Meeting, IEEE, Orlando, USA: 1-8. https://doi.org/10.1109/IAS.2011.6074308