Using particle swarm optimization to enhance PI controller performances for active and reactive power control in wind energy conversion systems

M Taleb¹, M Cherkaoui² and M Hbib¹

¹ Ph.D. Student, Mohamed V University, EMI, Rabat, Morocco
² Professor, Mohamed V University, EMI, Rabat, Morocco

Abstract. Recently, renewable energy sources are impacting seriously power quality of the grids in term of frequency and voltage stability, due to their intermittence and less forecasting accuracy. Among these sources, wind energy conversion systems (WECS) received a great interest and especially the configuration with Doubly Fed Induction Generator. However, WECS strongly nonlinear, are making their control not easy by classical approaches such as a PI. In this paper, we continue deepen study of PI controller used in active and reactive power control of this kind of WECS. Particle Swarm Optimization (PSO) is suggested to improve its dynamic performances and its robustness against parameters variations. This work highlights the performances of PSO optimized PI control against classical PI tuned with poles compensation strategy. Simulations are carried out on MATLAB-SIMULINK software.

1. Introduction
In real life, many optimization problems are difficult to solve by exact optimization methods, due to properties, such as dimensionality, multimodality, epistasis (parameters interactions), and non-differentiability. Hence, approximate algorithms are an alternative approach for these problems. They try to find near-optimum solutions to hard optimization problems, mimicking intelligent processes of behaviours observed from nature, sociology, thinking, and other disciplines [1]. Among these approximate algorithms, Nature-inspired algorithms (NIA) [2], [3] are currently the most exciting research areas, and they are continuously demonstrating exceptional strength in solving complex real-life problems. The two most popular ones are Genetic Algorithm (GA) [4] and PSO [5] that have been found to be highly competitive for solving a wide variety of optimization problems. In spite of both of them are nature-inspired algorithms, GA is an evolutionary algorithm [6] inspired by genetic evolution process based on best-to-survive criteria, while PSO is a swarm intelligence algorithm [7] that simulates the social behaviour of swarms such as flocks of birds or schools of fish. PSO and GA can both obtain high quality solutions, yet the computational effort required by PSO is less than the corresponding effort required by GA. In addition, PSO is easy to implement and there are few parameters to adjust [8]. That is why we expected in our previous paper [9] to use PSO approach to remedy to PI controller low transient time performances and weak robustness.
PI controller has been used to control active and reactive power of the most popular wind energy conversion system based on Doubly Fed Induction Generator (DFIG) for which stator is connected directly to the grid while the rotor is connected to grid through double bi-directional converters separated by a DC-Link [9] and sized to allow flowing only 30% of power via the rotor winding as
illustrated in figure 1. Bulk and cost, as well as copper losses are then reduced accordingly in comparison with other topologies [10], [11].

This paper is organized in four major sections. In section 2, PSO paradigm is summarized and objective function to optimize is formulated. PSO optimized PI is presented in section 3. Finally, conclusions are given in last section (section 4).

2. PSO paradigm

PSO paradigm was first introduced by James Kennedy, a social psychologist, and Russell C. Eberhart, an electrical engineer, in 1995 [5]. Now, it is widely used in various fields because of its easy implementation, low computational cost and good robustness [12], [13].

PSO evolves populations or swarms or individuals called particles which represents candidate solutions in search space. Each particle is assigned an initial random position vector and an initial random velocity. Then it updates iteratively its trajectory, namely position $x_i$ and velocity $v_i$, according to its best experience $p_i$ (personal best) and the best experience of the whole swarm $g$ (global best).

Updating rule is very simple and requires only primitive mathematical operators as defined on following equations (1) and (2):

$$
v_i(t+1) = w v_i(t) + c_1 r_1 (p_i(t) - x_i(t)) + c_2 r_2 (g(t) - x_i(t))
$$

$$
x_i(t+1) = x_i(t) + v_i(t+1)
$$

The three values that effect the new search direction, namely, current motion, particle own memory, and swarm influence, are incorporated via a summation approach as shown in equation (1) with three weight factors, namely, inertia factor ($w$), self-confidence factor ($r_1 c_1$) and swarm confidence factor ($r_2 c_2$), respectively. These latter two terms have randomized amplitudes due to random numbers $r_1$ and $r_2$ within $[0,1]$ that are generated at each iteration for each particle. Thus, different particle has different random numbers and then may have different strategies. Search space is then well explored and entrapment in local optima is avoided. Furthermore, to avoid big dispersion of the swarm that can lead to algorithm divergence, velocity magnitude should be controlled by setting a threshold $[v_{min}, v_{max}]$ on its absolute value. Finally, notion of “best” is evaluated, at each iteration, by the fitness function to be optimized (minimized or maximized). In present work, evaluation is made according to objective function given in equation (3).

$$ISE = \int \left[ (P_{ref} - P)^2 + (Q_{ref} - Q)^2 \right] dt$$

ISE is the Integral Squared Error that can be replaced by ITAE (Integral Time Absolute Error) or IAE (Integral Absolute Error).
PSO tunes parameters of the controller to minimize the ISE expressed on both active and reactive powers basis. Other approaches can be investigated, depending in situations, such as minimizing ISE for active power only or reactive power only.

The three steps of velocity update, position update, and fitness calculations are repeated until a desired convergence criterion is met. In our case, algorithm stops when specified maximum iterations number is reached. The flowchart is given in figure 2.

![Flowchart of PSO algorithm](image)

**Problem definition**
- Fitness (function to optimize): $f(x)$
- maximum iterations: $It_{\text{max}}$
- number of particles: $N_{\text{pop}}$
- Particle size: Number of variables = $N_{\text{var}}$
- Search space: $\text{Var}_{\text{min}}$ and $\text{Var}_{\text{max}}$

**PSO parameters**
- coefficients: $w$, $c_1$, $c_2$
- maximum velocity: $V_{\text{max}}$

**Random initialisation**
- Position and velocity of each particle
- fitness of these particles
  - $p_i(0) = x_i(0) = g(0)$

**Move particles according to following equations**

$$v_i(t + 1) = w v_i(t) + c_1 (p_i(t) - x_i(t)) + c_2 (g(t) - x_i(t))$$

$$x_i(t + 1) = x_i(t) + v_i(t + 1).$$

**Calculation of new fitness**
Update $p_i(t)$ and $g(t)$

**Stop criteria**
Yes

**Solution**
$g(t)$

3. PSO optimized PI controller

3.1. PI implementation

We have seen in [9] Direct and Indirect configurations of PI, and we highlighted the performances of indirect approach against the Direct one, mainly in term of overshooting rates. However, it was pointed out the poor capability of both configurations to deal with system uncertainties which may be caused by external disturbances and parameters variation.

We choose to continue deepen study for PI controller and we consider the Direct configuration only that we call back in figure 3.
3.2. PSO parameters

Compensation approach leads to same PI controller for both active and reactive powers. However, with PSO approach, we consider a general case where the two controllers can be totally different. Thus, four parameters will be considered: $K_{pp}$ and $K_{pq}$ the gains for active power PI control, $K_{pq}$ and $K_{iq}$ the gains for reactive power PI control.

Consequently, PSO algorithm is defined as follow:

- particles are the quadruples $(K_{pp}, K_{pq}, K_{pq}, K_{iq})$,
- the swarm consists of 10 particles,
- each particle is randomly initialized in the interval $[0, 100]$,
- the cost function is defined by equation (3),
- the maximum number of iterations is 50.

Standard PSO is adopted for which parameters are $c_1 = c_2 = 2$ and $w = 1$. However, $w$ is initialized to its standard value $w = 1$ and then gradually decreased (1% at each iteration) to avoid local optimum and stagnation problem due to fixed velocity step size [14].

3.3. Simulations and results

Even if in practice variations of $P_{ref}$ and $Q_{ref}$ are not instantaneous but made according to defined ramps, our simulations were made with profiles depicted in figure 4 where stiff variations have been imposed to the system. Results obtained in such hard conditions will be better in real case. $P_{ref}$ is always negative (generator convention) while $Q_{ref}$ can be positive (absorbed) or negative (delivered).

Figure 4. References $P_{ref}$ and for $Q_{ref}$ active and reactive power respectively.
Optimization is achieved with $K_{pp} = K_{pp} = 0.1$, $K_{ip} = 0.0393$, and $K_{iq} = 3 \times 10^{-7}$. ISE is reduced by 26% compared to classical PI (with poles compensation parameters).

Optimized PI provides good tracking without ripples that are due to residual coupling between active and reactive power. The high ripples observed during starting time (at less than 0.5s) for classical PI are also eliminated with PSO optimized PI.

The main parameters that influence the PI robustness are stator resistance and inductance. We already seen in [15] that robustness against $R_s$ is quite acceptable but it noticed high oscillations amplitudes and a risk of dropping out when $L_s$ vary by 10% to 20%. However, PSO Optimized PI maintains better robustness against same $L_s$ variations as shown in figure 7.

![Figure 5. Comparison between classical PI and PSO optimized PI for active power control.](image)

![Figure 6. Comparison between classical PI and PSO optimized PI for reactive power control.](image)
4. Conclusion
The main contribution of this work is to show how PSO can enhance PI performances for tracking any active and reactive powers profiles dictated by Grids Managers. High precision and fast response time are achieved by tuning PI gains that minimize the quadratic error (ISE). Robustness against parameters variations is also improved. This approach replaces the trial and error method that cannot succeed easily to satisfactory results.
PSO competes with many other nature-inspired algorithms and especially with GA in many fields. In addition, beside its easier implementation, it is generally considered efficient in terms of computational cost and robustness.
As prospect, this algorithm will be implemented in FPGA and DSP chips to appreciate its real-time response.

5. Appendices
DFIG parameters: $S_n = 3 \text{MVA}$; $R_s = 0.01 \Omega$; $R_r = 0.02 \Omega$; $L_s = 0.0137H$; $L_r = 0.0136H$; $M = 0.0135H$; $p = 2$; $J = 0.07\text{Kg.m}^2$; $f = 0.00244\text{N.m.s}^{-1}$. Turbine parameters: $P_w = 4\text{MW}$; Gear-box ratio: $G = 90$.

6. References
[1] K.-L. DU and M. Swamy, Search Optimization by Metaheuristics, Birkhäuser, July 2016.
[2] R. Ribeiro and F. Enembreck, “A sociologically inspired heuristic for optimization algorithms: A case study on ant systems,” Expert Systems with Applications, vol. 40, pp. 1814-1826, 2013.
[3] R. A. Mahale and S. Chavan, “A Survey: Evolutionary and Swarm Based Bio-Inspired Optimization Algorithms,” International Journal of Scientific and Research Publications, vol. 2, no. 12, 2012.
[4] J. H. Holland, “Genetic algorithms and the optional Allocation of trials,” SIAM Journal on Computing, vol. 2, no. 2, pp. 88-105, 1973.
[5] R. Eberhart and J. Kennedy, “A New Optimizer Using Particle Swarm Theory,” in Sixth International Symposium on Micro Machine and Human Science, IEEE, 1995.
[6] G. Rozenberg, T. Bäck and J. N. Kok, Handbook of Natural Computing, Springer-Verlag Berlin Heidelberg, 2012.
[7] S. Keerthi, K. Ashwini and M. Vijaykuma, “Survey Paper on Swarm Intelligence,” International Journal of Computer Application, vol. 115, no. 5, pp. 8-12, April 2015.
[8] A. Layak and L. S. Samrat, “Particle swarm optimization with adaptive neighbourhood search for solving multi-objective optimization problems,” International Journal of Swarm Intelligence, vol. 2, no. 1, 2016.
[9] M. TALEB and M. CHERKAOUI, “Active and Reactive Power Control of Doubly Fed Induction Generator Wind Turbines to Answer Grid Codes Requirements,” Journal of Clean Energy Technologies (JOCET), vol. 6, no. 2, pp. 101-105, 2018.

[10] O. P. Mahela and A. G. Shaik, “Comprehensive overview of grid interfaced wind energy generation systems,” Renewable and Sustainable Energy Reviews, vol. 57, pp. 260-281, 2016.

[11] M. Cheng and Y. Zhu, “The state of the art of wind energy conversion systems and technologies: A review,” Energy Conversion and Management, vol. 88, pp. 332-347, 2014.

[12] M. Taleb, A. Levant and F. Plestan, “Pneumatic actuator control: solution based on adaptative twisting and experimentation,” Control Engineering Practice, 48 (5), 2013.

[13] J. Kennedy and Eberhart RC, “Particle Swarm Optimization,” Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942-1948, 1995.

[14] Y. Shi and R. Eberhart, “A Modified Particle Swarm Optimizer,” in Proceedings of IEEE congress on evolutionary computation (CEC), AK, USA, 1998.

[15] M. TALEB and M. CHERKAOUI, “Active and Reactive Power Robust Control of Doubly Fed Induction Generator Wind Turbine to Satisfy New Grid Codes,” in Advances in Intelligent Systems and Computing 565 (Springer), AECIA-2016, Marrakech, Morocco, 2018.