**Design PID controller using PSO for electrical vehicle driven by BLDC Motor**

**Nasseer K. Bachache (member IEEE), Ali Muhssen Abdulsada**

**University College of Humanity Studies**

**Al Najaf, Iraq**

**Abstract**

In this project the swarm designer is applied to PID control system and grows its optimizer. PSO strategy has been applied to tune the parameters of PID controller for brushless DC motor. PSO is a valuable tool to tune PID based on a cost function that captures the user-specified performance criteria. The PSO tuning process is able to decrease the time cost function that measures how quickly controller stabilizes nonlinear systems which have been studied. The implementation of particle swarm optimization was implemented using MATLAB/ m-file program and linked with the system simulation program in MATLAB/ SIMULINK to calculate the fitness function.The proposed controller have designed to govern the output power of BLDC Motor not only the speed to save the life of battery in electrical vehicle The simulation results demonstrate that the system's behavior didn’t need further adaptation due to the robustness of the optimized PID controller.

Keywords— Brushless DC motor; PID controller; Particle Swarm Optimization PSO.

# **Introduction**

 It is an axiomatic in various countries have some goals that are more important than others. While the need of reducing carbon emissions has become an urgent global priority to mitigate climate change, but cost-efficient still the important goal. A numerous industries and electric vehicles EV need to a high speed variable application works with a high quality controller. This paper has been organized to produce Particle Swarm Optimization PSO that adjusts the parameters of PID Controller for industrial DC-drive works in electric vehicles EVs. The DC drive has been one of the most important strategies in speed control. There are many papers deal with adjusting parameters of PID controller parameters[1]. but there methodologies still need an effective method for tuning. Nowadays, several new intelligent optimization techniques have been emerged, such as Genetic Algorithms (GA), exploiting the ideas of Darwinian evolution, Simulated Annealing (SA), Ant Colony Optimization (ACO) and Bacteria Foraging Optimization (BFO) among these nature-inspired strategies the Particle Swarm Optimization (PSO) algorithm is relatively novel [2], PSO has received great attention in a control system. In paper [3] the fuzzy controller parameters generated by PSO for an AC-Drive speed controller. In paper [4] designed AC-Drive system depends on the reference model (RF) on line all these research didn’t care to calculate the power dissipated or the losses. In this paper, we proposed the tuning of PID controller adjusted at an optimal parameters which gives the lower power dissipated

# Particle Swarm Optimization

## Definitions of Particle swarm optimization (PSO)

 Particle swarm optimization (PSO) has been enormously successful within little more than a decade hundreds of papers have reported successful applications of PSO. In fact, there are so many of them, that it is difficult for PSO practitioners and researchers to have a clear up-to-date vision of what has been done in the area of PSO applications. Particle swarm optimization has been used across a wide range of applications. Universally we can say that PSO has particular promise include multimodal problems and problems for which there is no specialized method available or all specialized methods give unsatisfactory results. PSO applications are so numerous and diverse that a whole would be necessary just to review the most paradigmatic ones, assuming someone could identify them among the many hundreds of applications reported in the literature a really enormous task [5].

 The general purpose optimization method known as Particle Swarm Optimization (PSO) was first introduced by Dr. Russell C. Eberhart and Dr. James Kennedy in 1995 [6]. This theory works by maintaining a swarm of particles that move around the search-space influenced by the improvements discovered by the other particles. The advantage of using an optimization method such as PSO is that it does not rely explicitly on the gradient of the problem to be optimized, so the method can be readily employed in a host of optimization problems. This is especially useful when the gradient is too laborious or even impossible to derive [7]. Then (PSO) has been used across a wide range of applications.

## PSO Algorithms (Heading 2)

 A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

 The choice of PSO parameters can have a large impact on optimizing performance. Selecting PSO parameters that yield good performance has therefore been the subject of much research [8-10]. The PSO algorithm consists of just three steps, which are repeated until some stopping condition is met [11]:

(1) Evaluate the fitness of each particle

(2) Update individual and global best fitness’s and positions

(3) Update velocity and position of each particle

The first two steps (Particle Swarm Optimizers) are fairly trivial and share common with other algorithms like Genetic Algorithms and Fitness evaluation is conducted by supplying the candidate solution to the objective function. Individual and global best fitness’s and positions are updated by comparing the newly evaluated fitness’s against the previous individual and global best fitness’s, and replacing the best fitness’s and positions as necessary. The velocity and position update step is responsible for the optimization ability of the PSO algorithm.

##  Implementing PSO for Adapting PID Controller

The implementation of particle swarm optimization in this work is same what complex, because the performance of the system must be examined in each iteration and particles position during the optimization algorithm. Therefore, the optimization algorithm is implemented by using MATLAB m-file program and linked with the system simulation program in MATLAB SIMULINK, to check the performance of the system during the iterations.

Generally, In most intelligent optimization algorithms, there are commonly performance criteria such as: Integrated Absolute Error (IAE), the Integrated of Square Error (ISE), and Integrated of Time weight Square Error (ITSE). That can be evaluated analytically in frequency domain.

Each criterion has its own advantage and disadvantage. For example, disadvantage of IAE and ISE criteria is that its minimization can result in a response with relatively small overshot but a long settling time, because the ISE performance criteria weights all errors equally independent of time. The ITSE performance criterion can overcome the disadvantage of ISE criterion. The IAE, ISE, and ITSE performance criterion formulas are as equations 1,2,3 respectively:

 (1)

 (2)

 (3)

In this paper, the integrated of time weight square error ITSE is used for evaluating the accuracy performance of the fuzzy controller. A set of good control parameters can yield a good step response that will result in performance criteria minimization in the time domain, this performance criterion is called Fitness Function (FF) which can be formulated as equation 4:

 (4)

Where:

 MP is a maximum overshot.

Ess is a steady state error.

is the weight factor can set to be larger than 0.7 to reduce the overshot and steady state error, also can be smaller than 0.7 to reduce the rise time and settling time. The objective function is to minimize the fitness function FF. The PSO algorithm process can be summarized in the flowchart shown in figure 1.



Figure 1 A flowchart of designing PID by PSO

#  Modeling and Simulation of Three Phases Induction Machine

One of the most popular models of DC Motor has derived from its equivalent circuit is Krause’s model is based on a transformation of the stator currents and of the magnetic fluxes of the rotor to the reference frame “d−q” which rotates together with the rotor [12]. While, axis transformation is applied to transfer the three-phase parameters (voltage, current and flux) according to (d-q axes stationary frame), then coupling coefficients between the stator and rotor phase change continuously with the change of rotor position can be solved. Which stator and rotor parameters rotate at synchronous speed and all simulated variables in the stationary frame can consider d.c quantities [13].The per-phase equivalent circuit diagrams of an brushless DC motor in a two-axis synchronously rotating reference frame are illustrated in figure 2.



Figure 2 I.M equivalent circuit is d-q axes components.

Where: “” d-q axes component stator voltage, rotor voltage, stator current and rotor current respectively. () d-q axes components flux linkage of the stator and rotor. “” Stator and rotor winding resistance.() Stator and rotor inductance. Magnetizing inductance. Rotor speed.“” Synchronous speed. From the circuit diagram the equations (5, 6) can be obtained:

 (5)

(6)

 The development torque Te can be obtained by the interaction of air gap flux and rotor current Ir and solve the variables into dq-axes stationary frame to get the equation (7):

 (7)

 The dynamic torque equation of the rotor is formed in equation (8). Where, J is the rotor inertia, and TL is the external load torque [14].

 (8)

 The previous differential equations (5, 6, 7) and dynamic torque were simulated using MATLAB / SIMULINK. The dynamic and static performance of the drive system under different load was tested the SIMULINK has a complete modeling of the system as shown in figure 3. The PSO algorithm gives the optimal parameters of PID-controller as follows {where: Kp=10.2, Ki=1.4, Kd=0.013}..



Figure 3 Simulation of BLDC motor derive with PID.

# Simulation Result

A comparative performance between the proposed PID- PSO method and the ordinary PID controller are illustrated for step response of desired speed shown in figure4. Figure 5 shows the speed desired with different load.



Figure 4 DC Motor speed step response of PID controller,



Figure 5 Different loud derived by optimized PID Controller

# Conclusion

 A successfully generation of PID parameters by PSO is demonstrated in this paper. It is adopted method to give a higher robust controller for this system (perfect speed-tracking and non-sluggish), figure 4 show the optimized PID is more closely with desired input speed and also figure 5 shows a strongly controller of speed with loud variation. For a complex system the easy implementation SIMULINK-MATLAB can used in step “fitness function” of PSO algorithm. Increasing number of particles in PSO is more effective access than the increasing number of iterations to optimize the system for a proposed method.

##### Acknowledgment

We are pleased to thank University College of Humanity Studies and its dean and everyone who give us any supporting.

##### **References**

Wlas, M., Abu-Rub, H., Holtz, J.: "Speed Sensorless Nonlinear Control Of Induction Motor". In The Field Weakening Region. In: IEEE 13th International Power Electronics and Motion Control Conference, pp. 1084–1089 (2008)

Dubey, M.: "Design of Genetic Algorithm Based Fuzzy Logic Power System Stabilizers in Multimachine Power System". In: POWERCON 2008 & Power India Conference, October 12-15. IEEE, New Delhi (2008)

Nasseer K. Bachache،Wen J.W.: "Particle Swarm Optimize Fuzzy Logic Memberships of AC-Drive" ICSI2012 Conference LNCS,vol. 3771, pp. 460–459. Springer, Heidelberg Wang, X., Abbass, H.A. (2012)

A.G.Keskar, K.L.Asanare “Floating Membership Fuzzy Logic Controller For Adaptive Control Of AC Drive” IEEE Catalog Number: 97TH8280 ISIE’97 - Guhariies, Portugal1997

Ozcan Ender, and Chilukuri K. Mohan. "Analysis of a simple particle swarm optimization system." Intelligent engineering systems through artificial neural networks 8 (1998): 253-258.

Kennedy J., and Eberhart R "A discrete binary version of the particle swarm algorithm."IEEE International Conference on Systems, Man and Cybernetics, 1997 Computational Cybernetics and Simulation. 5, (1997): 4104-4108.

Bartz-Beielstein Thomas, Konstantinos E. Parsopoulos, and Michael N. Vrahatis."Analysis of particle swarm optimization using computational statistics." In International Conference of Numerical Analysis and Applied Mathematics (ICNAAM 2004): 34-37.

Trelea I.C. "The Particle Swarm Optimization Algorithm: convergence analysis and parameter selection". Letters Transactions on Industry Applications, 38, no. 3 (2002): 632-63on Information Processing 85 (2003): 317–325.

Eberhart C. Russell and Shi Y. H. "Particle Swarm Optimization." IEEE Transaction on Evolution and Computer 8, no. 3 (2004): 201-203.

Frans van den Bergh. “An Analysis of Particle Swarm Optimizers” PhD thesis submitted to the University of Pretoria, 2001.

Nasri Mehdi, Nezamabadi Pour Hossein, and MaghfooriMalihe "A PSO-based optimum design of PID controller for a linear brushless DC motor." World Academy of Science, Engineering and Technology 26, no. 40 (2007): 211-215.

J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

TallamRangarajan M., Thomas G. Habetler, and Ronald G. Harley. "Transient model for induction machines with stator winding