This is a postprint version of the published document at:

Haji Haji, V. y Monje, C. A. (2017). Fractional Order Fuzzy-PID Control of a Combined Cycle Power Plant Using Particle Swarm Optimization Algorithm with an Improved Dynamic Parameters Selection. *Applied Soft Computing*, 58, pp. 256-264.

DOI: <https://doi.org/10.1016/j.asoc.2017.04.033>

© 2017 Elsevier Ltd. All rights reserved.

This article is licensed under a under a [Creative Commons Attribution Non-Commercial No Derivatives License](#) [4.0 International License](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Fractional Order Fuzzy-PID Control of a Combined Cycle Power Plant Using Particle Swarm Optimization Algorithm with an Improved Dynamic Parameters Selection

V. Haji Haji^{a,*}, Concepción A. Monje^b

^a*Young Researchers and Elite Club, Borujerd Branch, Islamic Azad University, Borujerd, Iran*

^b*Systems Engineering and Automation Department, University Carlos III of Madrid, Avenida Universidad 30, 28911 Leganés, Madrid, Spain*

Abstract

The effectiveness of the Particle Swarm Optimization (PSO) algorithm in solving any optimization problem is highly dependent on the right selection of tuning parameters. A better control parameter improves the flexibility and robustness of the algorithm. In this paper, a new PSO algorithm based on dynamic control parameters selection is presented in order to further enhance the algorithm's rate of convergence and the minimization of the fitness function. The powerful Dynamic PSO (DPSO) uses a new mechanism to dynamically select the best performing combinations of acceleration coefficients, inertia weight, and population size. A fractional order fuzzy-PID (fuzzy-FOPID) controller based on the DPSO algorithm is proposed to perform the optimization task of the controller gains and improve the performance of a single-shaft Combined Cycle Power Plant (CCPP). The proposed controller is used in speed control loop to improve the response during frequency drop or change in loading. The performance of the fuzzy-FOPID based DPSO is compared with those of the conventional PSO, Comprehensive Learning PSO (CLPSO), Heterogeneous CLPSO (HCLPSO), Genetic Algorithm (GA), Differential Evolution (DE), and Artificial Bee Colony (ABC) algorithm. The

*Corresponding author. Address:

Tel.: +98 936 54 75 525; Fax: +98 66 42605335.

Email address: vhbhaji@gmail.com (V. Haji Haji)

simulation results show the effectiveness and performance of the proposed method for frequency drop or change in loading.

Keywords: Combined cycle power plant, Fractional order fuzzy-PID, Frequency control, Dynamic Particle Swarm Optimization.

1. Introduction

There has been a continuous development of the Combined Cycle Power Plants (CCPP), which combine a gas turbine and a steam turbine, because of their high energy efficiency and their low emission. In single-shaft combined cycle power plants the fuel supply system and the compressor are both attached to the turbine's shaft. Thus any frequency drop or change in loading have a direct effect on the air and fuel supply, which has a negative effect on the performance and stability of the system [1, 2]. The need for accurate modeling of the gas turbine, parameters estimation, and stability frequency and electrical voltage in presence of sudden changes or variations of the electrical load in the transmission and distribution systems has led to several publications in this area. In [1] the dynamic behavior and stability analysis of a CCPP for frequency drops are investigated. The performance and impact of the gas turbine dynamics and the model of the CCPP are discussed in [2, 3]. Paper [2] combines some dynamic models of the CCPP and builds a new one. The work in [4] focuses on the frequency regulation performance to improve control strategies. The classical controllers of a combined cycle gas turbine (CCGT) plant using Firefly Algorithm (FA) are discussed in [5]. It develops a small signal model for CCGT and applies FA for optimization of classical controller gains. A sensitivity analysis is carried out to show the robustness of the proposed controller.

In recent years, there has been a growing interest in the application of fractional calculus in various fields of science and engineering. Fractional calculus is a generalization of the classical calculus integral and derivative operators with integer order to arbitrary order. It has applied in various fields such as mathematics, physics, engineering, chemistry, computer science and mechanics [6–8]. The use of fractional calculus in the area of control theory has become very popular since the last few years. This application extends the opportunity to enhance the system control performance of conventional controllers. In papers [9–13], several fractional order controllers problems and tuning methods are introduced and considered. Paper [14] deals with

the tuning and auto-tuning of fractional order controllers for industry applications. This paper shows that the fractional order controllers are robust and can outperform standard PID controllers. In [15] a robust controller design based on fractional calculus for a wind turbine generator is presented. The fractional order PID (FOPID) filter controller tuning method based on Internal Model Control (IMC) is investigated in [16].

Evolutionary and Swarm Intelligence (SI) algorithms such as Genetic Algorithm (GA) [17], Particle Swarm Optimization (PSO) [18, 19], and Artificial Bee Colony Algorithm (ABC) [20] techniques have a long history in solving optimization problems of control systems. In [21] the Imperialist Competitive Algorithm (ICA) is used for robust load-frequency control of power systems. The performance of the Fruit fly Optimization Algorithm (FOA) based fractional order fuzzy-PID controller for electronic throttle is investigated in [22]. The multi-area automatic generation control problem is discussed in [23]. The authors in [23] use FA for the optimization of FOPID controller gains. Paper [24] discusses an application of improved Differential Evolution (DE) to the design of FOPID controllers. It shows the superiority of the proposed approach compared to GA and PSO algorithms.

The PSO algorithm is inspired by natural concepts such as bird flocking and fish schooling, and shows significant success in solving different optimization problems. Several variants of this algorithm are available in different references and can be classified according to fuzziness, accordance, attraction, activity, grouping, mobility, divisibility, interaction, uncertainty, continuity, topology, hierarchy, restriction, cooperation, objective, recursion, etc [25]. A variant of PSO algorithm which is named Comprehensive Learning PSO (CLPSO) is introduced in [26]. This algorithm tries to benefit from all particles' best experiences to update a particle's velocity, and hence prevent from a premature convergence. In [27], authors develop the CLPSO algorithm to the Heterogeneous CLPSO (HCLPSO), in which the swarm population is divided into two heterogeneous subpopulations to enhance exploration and exploitation performances, and therefore maintain the diversity of the algorithm. The PSO algorithm has been used in different areas of science and engineering, including classification, sensor networks, swarm robotic, and optimization problems [28]. In [29] the application of a fractional order PID controller to an Automatic Voltage Regulator (AVR) is discussed. This paper employs a PSO algorithm to tune the gains of the FOPID controller. The Automatic Generation Control (AGC) of the two areas thermal power system with craziness based PSO is presented and studied in [30]. A hybrid

PSO fuzzy-PID controller is proposed and implemented for speed control of a gas turbine in [31]. This paper evaluates the performance of the proposed algorithm and compares the main characteristics of the step response, such as rise time, settling time, and overshoot during startup and operating condition of the gas turbine, with other controllers. In [32] a new algorithm based on hybridizing between DE and PSO algorithms is presented to find the optimal parameters of a PID controller. A detailed review of the PSO concepts and its application can be found in [28, 33, 34].

In this paper, a new PSO algorithm based on dynamic control parameters selection is proposed. The DPSO algorithm uses a new mechanism and selects the best performing combinations of acceleration coefficients, inertia weight, and population size. A fractional order fuzzy-PID controller based on the DPSO algorithm is proposed to improve the performance of a single-shaft combined cycle power plant. The proposed controller is used as a speed (frequency) controller of the CCPP. The simulation results show the dynamic performance of the fuzzy-FOPID with optimum gains during frequency drop or change in loading based on two cost functions: Integral Time Absolute Error (ITAE) and Integral Time Squared Error (ITSE). These results are compared with those obtained using conventional PSO, CLPSO, HCLPSO, GA, DE, and ABC algorithms.

This paper is organized as follows: In Section 2, the combined cycle plant model is presented. The basic concept of fuzzy-FOPID controller is described in Section 3. Section 4 provides a brief overview of the PSO algorithm. In this section the main idea of the DPSO algorithm is discussed. Simulation results are presented and discussed in Section 5 and finally, the paper is concluded in Section 6.

2. Combined Cycle Power Plant Model

A CCPP is the combination of a gas turbine generator (Brayton cycle) and a steam turbine generator (Rankine cycle) for the production of electric power. The plant consists of several components, such as a gas turbine, a combustor, a compressor, a waste heat recovery boiler, a steam turbine and a generator. The compressed air is sent to the combustor by the compressor. The air burns with fuel, and makes a high temperature and high pressure combustor gas. The gas turbine is driven by this combustor high pressure gas. The energy of the exhaust high temperature gas is collected by the waste heat recovery boiler. This collected energy is used to drive the steam

turbine. The plant power output is the sum of the gas turbine generator and the steam turbine generator power outputs [2]. Since the air is adiabatically compressed, the compressor discharge temperature T_d and the ratio of input-output temperature x for isentropic compression are given as follows:

$$T_d = T_i \left(1 + \frac{x - 1}{\eta_c} \right), \quad (1)$$

$$x = (P_{r0} W_a)^{\frac{\gamma-1}{\gamma}}, \quad (2)$$

where T_i and η_c are the ambient temperature and the compressor efficiency, respectively, P_{r0} is the nominal compressor pressure ratio, W_a is the airflow, and γ is the ratio of specific heat. From the energy balance equation in the combustion chamber, the relationship between T_d and the gas turbine inlet temperature T_f is described as follows:

$$T_f = T_d + (T_{f0} - T_{d0}) \frac{W_f}{W_a}, \quad (3)$$

where W_f is the fuel flow, T_{f0} is the gas turbine inlet temperature, and T_{d0} is the compressors discharge temperature. The relationship between T_f and the gas turbine exhaust temperature T_e can be obtained by the following equation:

$$T_e = T_f \left[1 - \left(1 - \frac{1}{x} \right) \eta_t \right], \quad (4)$$

where η_t is the turbine efficiency. The energy supplied to the gas turbine and the steam turbine are defined as:

$$E_g = K_0 \{ (T_f - T_e) - (T_d - T_i) \} W_a, \quad (5)$$

$$E_s = K_1 T_e W_a, \quad (6)$$

where K_0 is the gas turbine output coefficient, and K_1 is the steam turbine output coefficient. The dynamic model and main signals of the CCPP are shown in Fig. 1 and Table 1. The parameters used for the CCPP are the same as those used in [1,2]. The plant model consists of two control loops. The first one is the frequency (speed) control loop, which changes the fuel flow to balance the difference between generation and load in frequency (speed) deviation (annotated in red). Our approach will be applied to this particular loop. The second one is the temperature control loop and consists of

Table 1: Main signals of the CCP model.

Signals	Definition
P_g	gas turbine generated power
P_s	steam turbine generated power
E_g	thermal power converted by the gas turbine
E_s	thermal power input of the heat recovery boiler
T_e	exhaust temperature
T_e'	measured exhaust temperature (pu)
T_e''	measured temperature at radiation shield (pu)
T_f	gas turbine inlet temperature
T_r	reference temperature (pu)
T_c	overheat control signal
F_d	fuel demand signal
N	rotor speed (frequency)
W_a	airflow
W_f	fuel flow
T_{off}	temperature offset

Figure 2: Steady state diagram in a CCP

where $\lambda, \mu > 0$. Also, K_P , K_I and K_D are the proportional, integral, and derivative gains, respectively. The $PI^\lambda D^\mu$ controller output $u(t)$ in time domain is:

$$u(t) = K_P e(t) + K_I D^{-\lambda} e(t) + K_D D^\mu e(t), \quad (8)$$

where $e(t)$ is the error signal. In our case, the fractional values for λ and μ are between 0 and 1, that is to say, $0 < \lambda < 1$ and $0 < \mu < 1$, corresponding to a standard FOPID.

Figure 3 shows a fractional order fuzzy-PID controller scheme, where $r(t)$ is the input speed reference, $y(t)$ is the rotor speed, and $u(t)$ is the control signal. In this work, the parameters K_e , K_d , K_α , K_β , λ , and μ for the CCP speed control are determined using the dynamic PSO algorithm. The Fuzzy Logic Controller (FLC) inputs are the speed (frequency) deviation and its derivative and the FLC output signal is the change in fuel demand. The fuzzy rule bases of the fractional order PID controller are shown in Table 2, where the fuzzy linguistic values NL, NM, NS, ZO, PS, PM, PL represent Negative Large, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium, and Positive Large, respectively. Figure 4 shows the membership functions for the error inputs and the FLC output. The nonlinear surface of the fuzzy logic controller is depicted in Fig. 5.

Figure 3: Structure of the fractional order fuzzy-PID controller.

Table 2: Control rules.

$\begin{matrix} e \\ \hline \frac{d^\mu e}{dt^\mu} \end{matrix}$	NL	NM	NS	ZO	PS	PM	PL
PL	ZO	PS	PM	PM	PL	PL	PL
PM	ZS	ZO	PS	PS	PM	PL	PL
PS	NM	NS	ZO	PS	PM	PM	PL
ZO	NM	NM	NS	ZO	PS	PS	PM
NS	NL	NM	NM	NS	ZO	PS	PM
NM	NL	NL	NM	NS	NS	ZO	PS
NL	NL	NL	NL	NM	NM	NS	ZO

Figure 4: Membership functions for error, rate of error, and FLC output.

4. Particle Swarm Optimization Algorithm

The PSO algorithm was induced by Kennedy and Eberhart in the field of computing and artificial intelligence for solving problems mostly in the domain of optimization [35]. A pseudo code of the PSO is summarized in Fig. 6. The PSO algorithm starts with a random population (particles), and uses two cognitive aspects, individual learning and learning from a social group to move each particle in a D dimensional search space toward the solution. The position and velocity of each particle are represented by vectors $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ and $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, respectively. In the case of fractional Fuzzy-PID controller, $D = 6$ and each particle x_i is defined as $\{K_e, K_d, K_\alpha, K_\beta, \lambda, \mu\}$. The best position of each particle is recorded in a vector $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, where i is the index of that particle. The best

Figure 5: Surface plot for the fuzzy logic controller.

solution achieved so far among all the neighbors of a particle is stored as a vector $p_g = (v_{g1}, v_{g2}, \dots, v_{gD})$. At each time step, the velocity and position of each particle can be updated according to the equations given below:

$$v_{id}(t+1) = wv_{id}(t) + c_1r_1(p_{id} - x_{id}(t)) + c_2r_2(p_{gd} - x_{id}(t)), \quad (9)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \quad d = 1, \dots, D \quad (10)$$

where c_1 and c_2 are the cognitive learning rate and the social learning rate which influence the convergence speed of each particle, r_1 and r_2 are uniform random numbers in $(0, 1)$, and w is an inertia weight. The inertia weight w plays a role of balancing the local and global search and often decreases linearly from w_{max} to w_{min} during the iterations.

4.1. Dynamic Particle Swarm Optimization Algorithm

In solving any optimization problem, the choice of the search operators and right combination of control parameters (cognitive coefficient c_1 , social coefficient c_2 , inertia weight w , and population size PS) plays a pivotal role in the performance of the PSO algorithm. In paper [36] a dynamic parameter selection mechanism is proposed to dynamically select the best performing combination of parameters (amplification factor, crossover rate, and the population size) for the DE algorithm. This paper demonstrates that the dynamic mechanism shows better performance over the state-of-the-art algorithms.

Particle swarm optimization algorithm

```
Define  $c_1$ ,  $c_2$ , and  $\omega$ 
Objective function  $f(x)$ 
Generate particles with random position  $x_i = \{K_e, K_d, K_\alpha, K_\beta, \lambda, \mu\}$ 
and velocity  $v_i (i = 1, 2, \dots, n)$ 
Evaluate fitness value for each particle  $x_i$  by  $f(x_i)$  and update global best
while ( $t < \text{MaxGeneration}$ )
  for  $i = 1 : n$ 
    Update the particle velocity according to equation (9)
    Update the particle position according to equation (10)
    Evaluate the fitness of particle  $x_i$ 
    Compare and update the personal best  $p_i$  and global best  $p_g$ 
  end for  $i$ 
end while
```

Figure 6: Pseudo code of the PSO algorithm.

In our approach a dynamic selection parameter mechanism is developed and implemented to choose the right combination of control parameters, cognitive coefficient c_1 , social coefficient c_2 , inertia weight w , and population size PS in PSO based optimization problems. The PSO algorithm starts with a random position and velocity for particles, and uses velocity coefficients (c_1, c_2) and inertia weight (w) to move particles. Various studies have been published with different values for each parameter c_1 , c_2 , and w . Mouayad *et al.* [37] used tuned values $c_1 = c_2 = 2.0$, Swarm size $PS = 30$ to design a PID controller for an automatic voltage regulator (AVR) application. In this paper, inertia weight w is decreased by a step size of 0.014 from 0.9 to 0.4. Paper [32] used the values $c_1 = c_2 = 2.0$, Swarm size of $PS = 100$, and inertia weight $w = 0.4$. Parer [38] used values $c_1 = c_2 = 2.05$, Swarm size $PS = 10$, and inertia weight $w = 0.76$ to study the performance of a FOPID controller for robot trajectory control. Control parameters $c_1 = 0.12$, $c_2 = 1.2$, $PS = 50$, and w from 0.4 to 0.9 are used in paper [19] for solving fractional programming problems. Paper [39] carried out a detailed study for parameters selection of PSO algorithm. The acceleration coefficient c_1 can be decreased from 2.5 to 0.5 whereas c_2 can be increased from 0.5 to 2.5 over

Dynamic Particle Swarm Optimization Algorithm

Objective function $f(x)$
Generate particles with random position $x_i = \{K_e, K_d, K_\alpha, K_\beta, \lambda, \mu\}$
and velocity $v_i(i = 1, 2, \dots, n)$
Define the parameters, $i_{pop} = nps, \omega_{set}, c_{1,set}, c_{2,set}$, and PS_{set}
Evaluate fitness value for each particle x_i by $f(x_i)$ and update global best
while ($t < MaxGeneration$)
Assign a random combination (y) of parameters
for $i = 1 : n$ *particles*
Update the particle velocity according to equation (11)
Update the particle position according to equation (10)
Evaluate the fitness of particle x_i
Compare and update the personal best p_i and global best p_g
if new vector is better than its previous vector, then $SR_y = SR_y + 1$;**end if**
end for i
 $period = period + 1$;
 $PS_{prd} = PS_{prd} + 1$;
if $mod(period, CS) = 0$ *and* $period < (\eta * CS)$
Select the best half combination based on the rankings using equation (12) and update y_{set}
else if $mod(period, \eta * CS) = 0$
Set each $SR_y = 0$ and $period = 0$
end if
if $mod(PS_{prd}, CS) = 0$ *and* $i_{pop} > 0$
Calculate $Rank_{PS_{i_{pop}}}$ using (13)
 $set\ i_{pop} = i_{pop} - 1$
if $i_{pop} \sim 0$
Archive the worst $(PS_{i_{pop}+1} - PS_{i_{pop}})$ individuals
Set $PS = PS_{i_{pop}}$
end if
end if
if $i_{pop} = 0$ *and* $PS_{prd} = nps * CS$
Set PS to the population size with the best ranking
Use require individuals from the archive
end if
if $PS_{prd} = \eta * CS$
Set $PS_{prd} = 0$, $i_{pop} = nps$, and $PS = PS_{i_{pop}}$
Use require individuals from the archive
Clear the archive
end if
end while

Figure 7: Pseudo code of the dynamic PSO algorithm.

the whole run. The number of particles can be limited to the range 20-60.

The pseudo code of the dynamic PSO algorithm is presented in Fig. 7. The DPSO starts with a random position, velocity, and combination of c_1 , c_2 , and w for each particle in the population. The best combinations for a certain number of generations (CS) are selected and used for a number of subsequent generations. New vectors are generated as follows:

$$v_{id}(t+1) = w_i v_{id}(t) + c_{1i} r_1 (p_{id} - x_{id}(t)) + c_{2i} r_2 (p_{gd} - x_{id}(t)), \quad (11)$$

where $w_i \in w_{set} = \{w_1, w_2, \dots, w_{nw}\}$, $c_{1i} \in c_{1set} = \{c_{11}, c_{12}, \dots, c_{1,nc_1}\}$, $c_{2i} \in c_{2set} = \{c_{21}, c_{22}, \dots, c_{2,nc_2}\}$, and $PS \in PS_{set} = \{PS_1, PS_2, \dots, PS_{nps}\}$. Here, nw , nc_1 , nc_2 , and nps refer to the cardinality of the set inertia weight, cognitive learning rate, social learning rate, and the population size PS , respectively, and the variables of the fractional order Fuzzy-PID controller to be searched are $x_i = \{K_e, K_d, K_\alpha, K_\beta, \lambda, \mu\}$.

If the new vector is better than its parent, it updates personal best p_i and the success rate of a combination (SR_y) is increased by one, where y is the combination of all w_{set} , c_{1set} , and c_{2set} . If the new vector is the best position found in the entire swarm, it replaces the global best. After CS generations the number of population size reduces to PS_{nps-1} and $PS_{nps} - PS_{nps-1}$ individuals are archived. Here, PS_{nps} is assumed to be larger than PS_{nps-1} . Also based on Eqn. (12) the success rate of each combination is recorded and the numbers of combinations are reduced to the half:

$$Rank_y = \frac{SR_y}{\text{Number of individuals } ith \text{ used a combination } y}. \quad (12)$$

The combinations with higher $Rank_y$ are chosen for the next generation. At the end of $CS \times nps$ generation, using Eqn. (13) the best population size is selected and used for the $(\eta - nps) \times CS$ next generation,

$$Rank_{PS_i} = \frac{\sum_1^{CS} \sum_{k=1}^{TC} SR_y}{PS_k}, \quad (13)$$

where TC is the total combination and η is calculated as follows:

$$\eta \approx \frac{\log(TC)}{\log(2)}. \quad (14)$$

After $\eta \times CS$ generations, the process restarts with all combinations of control parameters.

5. Simulation Results

The considered fuzzy-FOPID is used in speed control loop of the CCP. In this paper, FOMCON Matlab toolbox has been used to implement the fractional order PID controller [40]. The parameters $\{K_e, K_d, K_\alpha, K_\beta, \lambda, \mu\}$ of the speed controller are optimized using DPSO, PSO, CLPSO, HCLPSO (obtained using the codes available from <http://www.ntu.edu.sg/home/epnsugan>), GA, DE, and ABC algorithms. In this DPSO based optimization, the maximum number of iterations is 100, $CS = 30$, $\eta = 3$, $w \in \{0.2, 0.7, 0.99\}$, $c_1 \in \{1, 1.5, 2\}$, $c_2 \in \{1, 1.5, 2\}$, and $PS \in \{6, 4\}$. The parameters settings of the GA, DE, ABC, PSO, CLPSO and HCLPSO algorithms are listed in Table 3. In order to demonstrate the robustness of the DPSO algorithm proposed and discuss on its convergence, two cost functions ITSE and ITAE are used for tuning the gains of the speed controller in the power plant as follows:

$$J_1(K_e, K_d, K_\alpha, K_\beta, \lambda, \mu) = ITSE = \int_0^\infty t * (N^2(t) + P^2(t))dt, \quad (15)$$

$$J_2(K_e, K_d, K_\alpha, K_\beta, \lambda, \mu) = ITAE = \int_0^\infty t * (|N(t)| + t * |P(t)|)dt, \quad (16)$$

where P and N are power and speed, respectively. For the sake of a fair analysis of the convergence and in order to present statistical results, all the algorithms have been run 10 times. The convergence characteristics of the DPSO, PSO, CLPSO, HCLPSO, DE, GA, and ABC based fuzzy-FOPID for time indices ITSE and ITAE are compared in Fig. 8 and Fig. 9, respectively. It is obvious that the fuzzy-FOPID based DPSO, in terms of objective function minimization, outperforms other algorithms. Table 4 shows the optimum values of the FOPID gains obtained via minimizing J_1 and J_2 using the different evolutionary algorithms proposed, and Table 5 shows the corresponding time indices parameters and objective functions.

As can be seen in Table 4, the values of λ and μ are between 0 and 1 for all the algorithms tested and for both ITSE and ITAE cost functions. For instance, for the case of DPSO algorithm and ITSE cost function, $\lambda=0.7889$ and $\mu=0.5828$, which implies that both integral and derivative control actions are fractional and that the controller obtained is therefore a fractional order PID. The same happens for the rest of cases: the resulting controller is always a fractional order PID.

Table 3: Parameters settings in simulation.

Algorithm	Parameters value
GA	Population size = 6, Mutation probability = 0.2, Crossover probability = 0.8
DE	Population size = 6, Scaling Factor = 0.9, Crossover probability = 0.5
PSO	Population size = 6, $c_1 = c_2 = 2$, Inertia weight = 1
ABC	Population size = 10, Limit = 30
CLPSO	Population size = 6, $c_1 = c_2 = 1.49$, Inertia weight = 0.9-0.2
HCLPSO	Population size = 6, $c_1 = 2.5-0.5$, $c_2 = 0.5-2.5$, $c = 3-1.5$, Inertia weight = 0.99-0.2

Table 6 reports the results obtained by the different algorithms in 10 runs. In this comparison, the DPSO provides better average and best objective function. Besides, the dynamic parameters selection mechanism can improve the DSPO execution time compared to the PSO, CLSPO and DCLPSO. As shown in Fig. 8 and Table 5 for the ITSE case, it is clear that the DPSO provides a very low cost function. In the case of ITAE, from Fig. 9 and Table 5, the PSO algorithm has the ability to provide fast convergence, but fails in cost function aspect compared to the DPSO algorithm. As Fig. 9 shows, the ABC algorithm fails in cost function aspect compared to other algorithms.

The dynamic power responses of the CCP controlled with the different fuzzy-FOPIDs based on the DPSO, PSO, CLPSO, HCLPSO, DE, GA, and ABC algorithms by minimizing ITSE and ITAE are obtained and shown in Fig. 10 and Fig. 11, respectively. From Fig. 10 and Table 5 for the ITSE case, it is obvious that the fuzzy-FOPID based DPSO algorithm provides a very low objective function, while the overshoot, settling time, and the rise time are bigger than those with the fuzzy-FOPID controller based HCLPSO algorithm. The fuzzy-FOPID based GA algorithm provides very low overshoot and rise time but fails in the minimization of the objective function and settling time. It is also evident for the ITSE case that the CLPSO based controller provides the best rise time. From Fig. 11 and Table 5 for the ITAE case, it is clear that the fuzzy-FOPID based DPSO algorithm provides a very low objective function. However, the overshoot and the settling time are bigger than those with the DE and CLPSO algorithms. Although the GA based controller provides a low rise time, it fails in the minimization of the overshoot, the settling time, and the objective function compared to the other algorithms. It is also clear that the DE based controller provides the best overshoot.

Figure 12 and Fig. 13 show the frequency (speed) variations of the CCP

Figure 8: Optimization processes of ITSE for DPSO, PSO, CLPSO, HCLPSO, GA, DE, and ABC algorithms.

Figure 9: Optimization processes of ITAE for DPSO, PSO, CLPSO, HCLPSO, GA, DE, and ABC algorithms.

Figure 10: Comparison of power responses using ITSE based fuzzy-FOPID controllers.

Table 4: Optimum values of FOPID gains obtained via minimizing J_1 and J_2 using different evolutionary algorithms.

Algorithms	K_e	K_β	K_d	λ	μ	K_α
<i>$J_1 = ITSE$</i>						
GA	6.3201	0.0753	1.3914	0.7709	0.5148	7.1593
PSO	6.4405	0.1192	1.7597	0.7732	0.5416	5.6152
ABC	5.1655	0.1097	1.8478	0.7813	0.5977	7.8161
DE	5.1161	0.1277	1.998	0.7793	0.6035	6.8796
CLPSO	6.2130	0.1572	1.8916	0.6929	0.5257	6.1573
HCLPSO	4.3122	0.1503	1.7977	0.8035	0.5804	7.3910
DPSO	5.3301	0.1325	1.9979	0.7889	0.5828	6.5838
<i>$J_2 = ITAE$</i>						
GA	6.2519	0.1112	1.998	0.6830	0.5474	7.3034
PSO	5.6525	0.1589	0.8221	0.7914	0.5080	5.1851
ABC	7.3055	0.0767	1.7755	0.7576	0.5525	6.56
DE	5.6559	0.1059	1.998	0.6512	0.7951	7.992
CLPSO	6.3219	0.1154	0.8885	0.7604	0.7527	5.9463
HCLPSO	5.7656	0.1469	1.9506	0.6916	0.5748	7.1528
DPSO	6.6765	0.1286	1.5621	0.7642	0.5826	5.2145

Table 5: Optimum time indices parameters and objective function for J_1 and J_2 using different evolutionary algorithms.

Algorithms	Percent Overshoot(%)	Rise Time(sec)	Settling Time(sec)	Objective Function
<i>$J_1 = ITSE$</i>				
GA	68.8119	0.3539	8.8946	0.4305
PSO	71.1982	0.3539	7.8232	0.3973
ABC	76.8118	0.3642	7.1958	0.3946
DE	70.5979	0.3871	7.2283	0.3924
CLPSO	76.5512	0.3236	7.6345	0.4052
HCLPSO	71.6258	0.3612	7.2638	0.3883
DPSO	72.4819	0.3628	7.3085	0.3842
<i>$J_2 = ITAE$</i>				
GA	77.3346	0.3233	8.1011	51.4613
PSO	68.3957	0.3882	7.4374	49.6871
ABC	81.1809	0.3326	8.59	51.5261
DE	35.9992	0.8845	4.6206	50.2868
CLPSO	47.3939	0.4893	6.3140	49.3517
HCLPSO	75.4828	0.3501	7.2504	49.6467
DPSO	72.1777	0.3842	6.8752	49.3

Table 6: Optimum objective function and execution time for J_1 and J_2 using different evolutionary algorithms in 10 runs.

Algorithms	Objective Function			Time (minute)		
	Best	Worst	Average	Best	Worst	Average
$J_1 = ITSE$						
GA	0.4305	0.5482	0.4653	9.243	10.342	9.832
PSO	0.3973	0.4294	0.41	12.013	13.544	12.425
ABC	0.3946	0.5179	0.4410	19.898	20.743	20.181
DE	0.3924	0.4812	0.432	12.62	13.809	13.105
CLPSO	0.4052	0.4818	0.4378	10.975	11.911	11.37
HCLPSO	0.3883	0.4466	0.4050	10.970	12.071	11.554
DPSO	0.3842	0.4398	0.4005	9.543	11.231	10.122
$J_2 = ITAE$						
GA	51.4613	58.4613	54.488	9.599	10.345	9.894
PSO	49.6871	51.9465	50.8713	12.105	12.779	12.439
ABC	51.5261	61.3542	54.8061	20.063	21.202	20.539
DE	50.2866	55.6868	52.6749	12.47	13.485	12.999
CLPSO	49.3517	56.6444	52.4964	11.106	12.152	11.676
HCLPSO	49.6467	54.6006	52.6763	10.219	11.984	11.38
DPSO	49.3	52.205	50.401	9.287	11.31	10.351

Figure 11: Comparison of power responses using ITAE based fuzzy-FOPID controllers.

model for the different fuzzy-FOPID controllers tuned. From Fig. 12, it is evident that the DPSO algorithm provides a lower final error than the other algorithms for the ITSE case. In the case of the ITAE cost function, from Fig. 13, it is clear that the DE based controller provides the worst final error.

The variation of best parameters w , c_1 , and c_2 during the whole evolution process is shown in Fig. 14 and Fig. 15 for ITSE and ITAE, respectively. It is clear that no fixed combination is the best during the course of the evolution process. The dynamic speed responses of the CCP for an instantaneous 3 % frequency drop are shown in Fig. 16 and Fig. 17 for ITSE and ITAE, respectively. From Fig. 16, it is evident that all FOPID controllers are able to track the reference signal within a very short time. As Fig. 17 shows, the DE and GA based controllers provide the best final error. Figure 18 and Fig. 19 show the corresponding dynamic power responses for this frequency drop. In Fig. 18 and Fig. 19 the overheat control brach is activated in order to prevent the exhaust gas temperature from exceeding the reference value. Figures 16-19 show that evolutionary and swarm intelligence based FOPID controllers try to bring back the speed and power in a desired interval during the frequency drop within a minimum time.

Figure 12: Comparison of speed responses for ITSE based fuzzy-FOPID controllers.

Figure 13: Comparison of speed responses for ITAE based fuzzy-FOPID controllers.

Figure 14: Best parameter values for w , c_1 , and c_2 using ITSE during the evolution process.

Figure 15: Best parameter values for w , c_1 , and c_2 using ITAE during the evolution process.

Figure 16: Dynamic speed responses for 3 % frequency drop using cost function ITSE.

Figure 17: Dynamic speed responses for 3 % frequency drop using cost function ITAE.

Figure 18: Dynamic power responses for 3 % frequency drop using cost function ITSE.

Figure 19: Dynamic power responses for 3 % frequency drop using cost function ITAE.

6. Conclusions

This paper proposes a newly evolved PSO algorithm with dynamic parameters selection. The dynamic PSO algorithm is based on the PSO algorithm and dynamically selects the best performing combinations of the acceleration coefficients, the inertia weight, and the population size. The analyses based on ITSE and ITAE cost functions shows that the proposed dynamic parameters selection mechanism can improve the algorithm's convergence rate and minimization of fitness function compared to the conventional PSO, CLPSO, HCLPSO, GA, DE, and ABC algorithms.

A fractional order fuzzy-PID controller has been tuned for the speed control loop of a CCPP. The gains of this controller are optimized successfully using DPSO and other algorithms. The simulation studies carried out in this work clearly show that the proposed fuzzy-FOPID controller based DPSO algorithm improves the performance of CCPP responses during frequency drop or change in loading.

In order to continue exploring the potential of the proposed approach, future works will research on the application of this dynamic parameters selection mechanism for enhanced PSO variants, such as HCLPSO, ensemble PSO, CLPSO, Fully Informed Particle Swarm Optimization (FIPS), and others. A comparative analysis will be carried out to check the validity of the results.

References

- [1] J. Mantzaris, C. Vournas, Modelling and stability of a single-shaft combined cycle power plant, *Int. J. Thermo.* 10 (2) (2007) 71–78.
- [2] N. Kakimoto, K. Baba, Performance of gas turbine-based plants during frequency drops, *IEEE Trans. Power Syst.* 18 (3) (2003) 1110–1115.
- [3] G. Lalor, J. Ritchie, D. Flynn, M.J. O'Malley, The impact of combined-cycle gas turbine short-term dynamics on frequency control, *IEEE Trans. Power Syst.* 20 (3) (2005) 1456–1464.
- [4] S. Carmona, S. Rios, H. Peña, R. Raineri, G. Nakic, Combined cycle unit controllers modification for improved primary frequency regulation, *IEEE Trans. Power Syst.* 25 (3) (2010) 1648–1654.

- [5] L.C. Saikia, K.S. Sahu, Automatic generation control of a combined cycle gas turbine plant with classical controllers using Firefly Algorithm, *Int. J. Elec. Power* 53 (2013) 27.
- [6] I. Podlubny, *Fractional Differential Equations*, Academic Press, San Diego, CA, 1999.
- [7] A. Shafieezadeh, K. Ryan, Y. Chen, Fractional order filter enhanced LQR for seismic protection of civil structure, *J. Comput. Nonlinear Dyn.* 3 (2008) 020201.1–021404.7.
- [8] C.A. Monje, Y.Q. Chen, B.M. Vinagre, D. Xue, V. Feliu, *Fractional-order systems and controls. Fundamentals and applications*, Springer, 2010.
- [9] L. Liu, F. Pan, D. Xue, Fractional-order optimal fuzzy control for network delay, *Optik - International Journal for Light and Electron Optics* 125 (2014) 7020–7024.
- [10] H. Li, Y. Luo, Y. Chen, A fractional order proportional and derivative (FOPD) motion controller: Tuning rule and experiments, *IEEE Trans. Control Syst. Technol.* 18 (2010) 516–520.
- [11] S.E. Hamamci, Stabilization using fractional order PI and PID controllers, *Nonlinear Dyn.* 51 (2008) 329–343.
- [12] F. Martín, C.A. Monje, L. Moreno, C. Balaguer, DE-based tuning of $PI^\lambda D^\mu$ controllers, *ISA Trans.* 59 (2015) 398–407.
- [13] J.Y. Cao, B.G. Cao, Design of fractional order controller based on particle swarm optimization, *Int. J. Control, Autom. Syst.* 4 (2006) 775–781.
- [14] C.A. Monje, B.M. Vinagre, V. Feliu, Y.Q. Chen, Tuning and auto-tuning of fractional order controllers for industry applications, *Control Eng. Pract.* 16 (2008) 798–812.
- [15] S. Ghasemi, A. Tabesh, J. Askari-Marnani, Application of fractional calculus theory to robust controller design for wind turbine generators, *IEEE Trans. Energy Conv.* 29 (3) (2014) 780–787.
- [16] B. Maamar, M. Rachid, IMC-PID-fractional-order-filter controllers design for integer order systems, *ISA Trans.* 53 (2014) 1620–1628.

- [17] B.K. Sarkar, P. Mandal, R. Saha, S. Mookherjee, D. Sanyal, GA-optimized feedforward-PID tracking control for a rugged electrohydraulic system design, *ISA Trans.* 52 (2013) 583–861.
- [18] S.K. Pandey, S.R. Mohanty, N. Kishor, João P.S. Catalão, Frequency regulation in hybrid power systems using particle swarm optimization and linear matrix inequalities based robust controller design, *Int. J. Elec. Power.* 63 (2014) 887–900.
- [19] O.A. Raouf, I.M. Hezam, Solving fractional programming problems based on swarm intelligence, *J. Ind. Eng. Int.* 10 (2014) 1–10.
- [20] A. Rajasekhar, R. K. Jatoth, A. Abraham, Design of intelligent $PID/PI^\lambda D^\mu$ speed controller for chopper fed DC motor drive using opposition based artificial bee colony algorithm, *Eng. Appl. Artif. Intel.* 29 (2014) 13–32.
- [21] H. Shabani, B. Vahidi, M. Ebrahimpour, A robust PID controller based on imperialist competitive algorithm for load-frequency control of power systems, *ISA Trans.* 52 (2013) 88–95.
- [22] W. Sheng, Y. Bao., Fruit fly optimization algorithm based fractional order fuzzy-PID controller for electronic throttle., *Nonlinear Dyn.* 73 (2013) 611–619.
- [23] S. Debbarma, L.C. Saikia, N. Sinha, Solution to automatic generation control problem using firefly algorithm optimized $I^\lambda D^\mu$ controller, *ISA Trans.* 53 (2) (2014) 358–366.
- [24] A. Biswas, S. Das, A. Abraham, S. Dasgupta, Design of fractional-order PID controllers with an improved differential evolution, *Eng. Appl. Artif. Intel.* 22 (2009) 345–350.
- [25] S. Rana, S. Jasola, R. Kumar, A review on particle swarm optimization algorithms and their applications to data clustering, *Artif. Intell. Rev.* 35 (2010) 211–222.
- [26] J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Trans. Evol. Comput.* 10 (2006) 281–295.

- [27] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, *Swarm Evol. Comput.* 24 (2015) 11–24.
- [28] S. Alam, G. Dobbie, Y.S. Koh, P. Riddle, S. Ur rehman, Research on particle swarm optimization based clustering: A systematic review of literature and techniques, *Swarm Evol. Comput.* 17 (2014) 1–13.
- [29] M. Zamani, M. Karimi-Ghartemani, N. Sadati, M. Parniani, Design of a fractional order PID controller for an AVR using particle swarm optimization, *Control Eng. Pract.* 17 (2009) 1380–1387.
- [30] H. Gozde, M. Cengiz Taplamacioglu, Automatic generation control application with craziness based particle swarm optimization in a thermal power system, *Int. J. Elec. Power Energ. Syst.* 33 (2011) 8–16.
- [31] A. Mansouri Mansourabad, M.T. Hamidi Beheshti, M. Simab, A hybrid PSO-Fuzzy-PID controller for gas turbine speed control, *Int. J. Control, Autom. Syst.* 6 (2013) 13–24.
- [32] A. Moharama, M.A. El-Hosseini, H.A. Ali, Design of optimal PID controller using hybrid differential evolution and particle swarm optimization with an aging leader and challengers, *Appl. Soft Comput.* 38 (2016) 727–737.
- [33] A. Banks, J. Vincent, C. Anyakoha, A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications, *Nat. Comput.* 7 (2008) 109–124.
- [34] A. Khare, S. Rangnekar, A review of particle swarm optimization and its applications in solar photovoltaic system, *Appl. Soft Comput.*, 13 (2013) 2997–3006.
- [35] J. Kennedy, RC. Eberhart, Particle swarm optimization, In *Proceedings of IEEE international conference on neural networks*, 1995, vol 4, pp. 1942–1948.
- [36] R.A. Sarker, S.M. Elsayed, T. Ray, Differential evolution with dynamic parameters selection for optimization problems, *IEEE Trans. Evol. Comput.* 18 (5) (2014) 689–707.

- [37] M.A. Sahib, B.S. Ahmed, A New multi-objective performance criterion used in PID tuning optimization algorithms, *J. Adv. Res.* 7 (2015) 125–134.
- [38] Z. Bingül, O. Karahan, Fractional PID controller tuned by evolutionary algorithms for robot trajectory control, *Turk. J. Elec. Eng. & Comp. Sci.* 20 (2012) 1123–1136.
- [39] S. Das, A. Abraham, A. Konar, Particle swarm optimization and differential evolution algorithms: Technical analysis, applications and hybridization perspectives, in: Y. Liu, A. Sun, H.T. Loh, W.F. Lu, E.P. Lim, *Advances of Computational Intelligence in Industrial Systems*, Springer Berlin Heidelberg, 2008, vol. 116, pp. 1–38.
- [40] A. Tepljakov, E. Petlenkov, J. Belikov, FOMCON: Fractional-Order Modeling and Control Toolbox for MATLAB, in: *Proceedings of 18th International Conference on Mixed Design of Integrated Circuits and Systems*, Gliwice, Poland, 2011, pp. 684–689.