[Mansoura Engineering Journal](https://mej.researchcommons.org/home)

Volume 40 | Issue 1 Article 1

7-5-2020

A New Method of Particle Swarm Optimization Approach for Tuning PID Controller.

Amal Moharam
Computers Engineering and Control systems Dept., Faculty of Engineering, Mansura University

Mostafa Elhosseini
Department of Computer engineering &Control Systems,, Mansoura University, Egypt,
melhosseini@gmail.com

Hisham Khalifa
Computers and Systems Engineering - Faculty of Engineering - Mansoura University,
h_arafat_ali@mans.edu.eg

A New Method of particle swarm optimization approach for tuning PID controller

استخدام خوارزمية جديدة لسرب الجسيمات لضبط معاملات المتحكم التناسبى التكاملى التفاضلى

Amal Moharam, Mostafa A. El-Hosseini, Hesham A. Ali
Computers Engineering and Control systems Dept., Faculty of Engineering, Mansura University.

الملخص

يعرض هذا البحث خوارزمية جديدة لتحديد معاملات المتحكم التناسبى التكاملى التفاضلى باستخدام خوارزمية سرب الجسيمات وقد تم استخدام المتحكم المقترح فى منظومة للتحكم فى مستوى السائل. وتم مقارنة المتحكم المقترح ببعض الخوارزميات الاخرى وجد انه الاكثر كفاءة واستطاع تحقيق افضل النتائج .

Abstract

This paper outlines a design method for tuning of PID controller using new algorithm based on Particle Swarm Optimization. The tuned PID controller is used in liquid level system control. Compared to different algorithms; the proposed method was more efficient. It gets the best results.

Keywords:

Particle swarm optimization (PSO); PID controller.

1. Introduction

In industrial control system, several control methods such as adaptive control, neural control, and fuzzy control have been used. However, the best known controller is proportional-integral-derivative (PID) controller. PID controller is widely used in the industry because of its simple structure and robust performance.

However, the process of setting the optimal gains of PID controllers is very difficult because many industrial plants have often many problems such time delays, and nonlinearities. Several methods have been proposed for the tuning of PID controllers such as Ziegler and Nichols (Z-N) method [1] and Cohen–Coon method. Z-N method is the oldest method and simplest one.

However, it is often hard to determine optimal or near optimal parameters of PID with the Z-N. formula in many industrial plants. Z-N method fails to provide an acceptable performance because it is always provides a large overshoot and settling time, so that the values of the PID parameters are often subsequently refined in accordance with the operator's experience [37].

Recently, many evolutionary computation methods such as genetic algorithm (GA),

Received: 26 January, 2015 - Accepted: 1February, 2015

many evolutionary computation methods such as genetic algorithm (GA), differential evolution (DE) and particle swarm optimization (PSO) have been employed to tune PID controller in various plants [2-7].However, the simplicity of PSO (i.e. it is straight forward and has less parameters to be tuned) and its low computational cost with high performance make it commonly used in the industrial applications [43].

PSO is an evolutionary computation technique proposed by Kennedy and Eberhart in [8].The original intent of PSO is to simulate the social interaction behavior of birds flocking. In PSO, each particle p in the flock is initialized with randomly chosen velocity Vp and position Xp in an n-dimensional search space. The PSO concept consists of, at each time step, accelerating each particle towards its own historically best position ($pbest_p$) and the best position found by the entire swarm ($gbest$)[9]. The update equations in the PSO are:

$$\left.\begin{aligned} V_{pd}(t+1) &= \omega V_{pd}(t) + C_1 r_1\,(pbest_{pd} - X_{pd}(t)) + C_2 r_2\,(gbest_d - X_{pd}(t)) \\ X_{pd}(t+1) &= X_{pd}(t) + V_{pd}(t+1) \end{aligned}\right\} \quad (1)$$

Where r_1, r_2 are random numbers uniformly distributed in [0,1]. C_1, C_2 are called acceleration constants, ω is the inertia weight, and d (d = 1, 2, . . . , n) represents the dth dimension of the search space.

Although PSO has fast convergence behavior, there is some deficiency in PSO performance. This is due to that all particles learn from best particle in updating velocities and positions. If best particle located at a local optimum the whole swarm may trap and this would lead to premature convergence.

Various attempts have been made to improve the performance of PSO. One of these attempts is hybridizing PSO with other search techniques [10-17] to benefit from advantages of each algorithm. The hybridization between DE [18] and PSO looks a promising optimizer. This hybridization tries to benefit from good global search capability of DE and high speed convergence of PSO.

DE is a simple evolutionary algorithm for global optimization proposed by Price and Storn. The DE-variants perturb the current generation population members with the scaled differences of randomly selected and distinct population members. Therefore, no separate probability distribution has to be used for generating the offspring [19]. Easy methods of implementation and negligible parameter tuning made the algorithm quite popular.

In the past decade, numerous hybrids of DE and PSO have been made [20-23]. Particle swarm optimization with differentially perturbed velocity (PSO-DV) is considered one of embedded hybrid algorithms [21]. In this algorithm a differential operator (borrowed from differential evolution) has been tightly coupled with the velocity update scheme of PSO. This differential operator can be considered as additional mutation to guarantee more diversity to PSO and hence avoid local optimum problem. Unlike the PSO scheme, a particle is shifted to a new location only if the new location yields a better fitness value. Due to this restriction, some of the particles may get stagnant in the search space. PSO-DV algorithm tries to solve this problem by replacing these stagnated particles with randomized particles.

In PSO-DV algorithm, DE is embedded in PSO. The operating manners of DE and PSO optimizers cannot be separated explicitly. We cannot separate their contribution to fitness improvement since DE and PSO are integrated into a hybrid optimizer. So, PSO-DV usually implies a higher risk of design failure [20].

Recently, a new PSO approach has been introduced based on aging concept [24]-[28]. It applies the age index on a particle (or a group of particles) in the swarm. If the particle cannot contribute effectively in searching space (i.e. it gets old), it should be replaced. Thus, the age of the particle is considered as an important indicator of its quality. The way of adjusting the life span of a particle and selecting a new one is an important step.

PSO with an aging leader and challengers (ALC-PSO) [28] is an age-based PSO algorithm. ALC-PSO is an attempt to simulate the aging concept in the nature. As the leader of the colony becomes old, it gives the opportunity to another individual to lead the colony. In this algorithm, when the swarm falls in local optima, this means that *gbest* gets old and cannot lead the swarm. ALC-PSO searches about another promising particle (called a challenger) to replace *gbest*. To do that, it travels randomly in one way direction for a predetermined number of attempts to select a challenger. The most important feature of ALC-PSO is keeping fast convergence of PSO. However, it is often hard to get acceptable diversity to search more and more promising area of search space.

In this paper, an effective approach called hybrid Differential Evolution and Particle Swarm Optimization with an Aging Leader and Challengers (ALC-PSODE) is presented. ALC-PSODE modifies PSO-DV by using another mutation borrowed from ALC-PSO (ashelping agent) to solve stagnation of particles problem and improve the performance of PSO-DV. Using one dimensional mutation of ALC-PSO can prevent *gbest* moving to the position of randomly mutated particle. This can protect the swarm from unacceptable divergence. To validate ALC-PSODE algorithm, it first tested on five benchmark functions. Then it is also used for tuning PID controller.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 develops the ALC-PSODE algorithm in detail. Section 4 experimentally validates the ALC-PSODE and compares it with five evolutionary algorithms on five benchmark functions. Section 5 describes the linearized model of liquid level system. Section 6 describes the application of ALC-PSODE to tune the PID controller and test it against various controllers. Finally, section 7 concludes this paper.

2. Related work

To solve the premature convergence problem of PSO, many algorithms that are based on adding a new mutation to PSO have been introduced as PSO-DV [21] and ALC-PSO [28]. In the PSO-DV algorithm [21], for each particle p in the swarm two other distinct particles, say p1 and p2 ($p \neq p1 \neq p2$), are selected randomly. The difference between their positional coordinates is taken as a difference vector δ:

$$\delta = X_{p2} - X_{p1}$$

Then the d-th velocity component ($1 < d <$ number of dimensions) of the target particle p is updated as:

If rand (0, 1) ≤ CR
$V_{pd}(t+1) = \omega V_{pd}(t) + ß\delta_d + C_2 r_2\,(gbest_d - X_{pd}(t))$
Else $V_{pd}(t+1) = V_{pd}(t)$
Endif

Where CR is the crossover probability, ß is a scale factor in [0,1] and $\delta_d$is the d-th δcomponent .The aim of using vector differential operator is to produce some additional exploration capability. The next step is creating a new trial location Tr_p for the particle:

$$Tr_p = X_p(t) + V_p(t+1)$$

Then the algorithm selects between the trial vector Tr_p and position vector $X_p(t)$. It selects the vector which has better fitness value. Due to this selection strategy, some particles may not be able to find better position (i.e. they get stagnant). The algorithm tackles this stagnation by shifting these particles to a new locationusing a random mutation. However, this means slow and unstable convergence especially if there are a large number of stagnated particles.

Some PSO-DV variants have been proposed to improve its performance. The simplest one is APSO-DV algorithm (particle swarm optimization with differentially perturbed velocity hybrid algorithm with adaptive acceleration coefficient) [29].The main objective of APSODV is to use adaptive acceleration coefficient for updating the positions of the particles of PSODV to accelerate the search for the global solution. In general, tuning the parameters can improve the performance of PSO-DV, but this is not a general case.

Another modified PSO-DV algorithm is evolving ant direction particle swarm optimization with differentially perturbed velocity (EADPSODV) [30]. In this approach, ant colony search is utilized by the EADPSODV algorithm to find a suitable mutation operator for PSODV. Genetic algorithm method is employed to evolve the ant colony parameters.

ALC-PSO [28] is a new technique to tackle the shortcoming of PSO. Its key idea is based on invoking a new mutation notion. When $gbest$ fails to lead the swarm, all swarm particles learn from another particle (i.e. a challenger). Hence, the diversity can be restored again after it had been lost. The steps of ALC-PSO algorithm can be summarized as follows:

1. First, initialize randomly the position and velocity of all particles ($p = [1,2, \dots . M]$) in the swarm, set $gbest$ as a *Leader*, initialize the age of the Leader θ = 0, set the lifespan of the Leader to an initial value (e.g.lifespan=30).
2. In the beginning of the run; in each generation, updating the velocity, position, $pbest$ and $Leader$ as in the PSO algorithm.
3. Adjust life span as will be discussed later and increase the age of the Leader θ by 1.
4. Check if the Leader become aged (i.e. age of the Leader> lifespan), then go to step 5. Else, go to step 7.
5. Generate a challenger which is different from the previous Leader in only one dimension (i.e. to keep some good characteristics of it).
6. Test the *challenger*, if it can improve at least one $pbest$; then replace the *Leader* by the *challenger*. Set the age of the Leader θ = 0; set the lifespan of the Leader to an initial value. If the *challenger* fails in testing, then the old leader remains leading the swarm.
7. Check if stop condition is satisfied, then terminate the algorithm .Else, go to step 2.

The main component of ALC-PSO that determines its performance is life span. The lifespan is adjusted according to the leader's ability to improve the swarm. To determine leader's ability, ALC-PSO observes 3 indexes during a *Leader*'s lifetime (the decision tree for the lifespan controller is shown in Fig. 1) [1]:

1- Improvement of $gbest$.
If $f(gbest(\theta)) - f(gbest(\theta - 1)) < 0$; then the current Leader have a good leading power and it is expected to have the ability to improve the swarm in the following generations. So the lifespan of the Leader is elongated (i.e. lifespan is increased by 2).

2- Improvement of collective $pbest_p$.
If $\sum_{p=1}^{M} f(pbest_p(\theta)) - \sum_{p=1}^{M} f(pbest_p(\theta - 1)) < 0$; this means that the historically *gbest* is not improved but the Leader is still able to improve at least one $pbest_p$position. This situation implies that the current Leader still has the ability to improve the swarm (but at less degree) in the following generations. So the lifespan of the Leader is increased by 1.

3- Improvement of $Leader$.
If $(Leader(\theta)) - f(Leader(\theta - 1)) < 0$;this means that the current *Leader* fails to lead any of the particles in the swarm toward better positions. However, the *Leader* still has the ability to improve itself. Hence, the ability of the Leader to improve the swarm in the following generations is dubious. Therefore, thelifespan of the Leader remains unchanged.

If there is no improving in the above three indexes, then the current Leader is expected to have no ability to improve the swarm in the following generations. So the lifespan of the Leader is shortened (i.e. lifespan is reduced by 1).Indeed, this decreasing in lifespan means that the leader is getting old by higher rate (i.e. the moving towards replacing the Leader).

3.Proposed ALC-psodealgorithm

In the beginning of the run, ALC-PSODE is as PSO-DV in updating the velocity and position. If the swarm traps,a simple mutation is borrowed from ALC-PSO to tackle PSO-DV deficiency. This mutation is done by making the swarm learning from another particle which differs from the old leader in one dimension. This one dimension mutation can introduce the diversity again to PSO-DV and moreover keeping acceptable convergence. The updated velocity equation is changed to:

$$V_{pd}(t+1) = \omega V_{pd}(t) + ß\delta_d + C_2 r_2\left(Leader_d - X_{pd}(t)\right) \quad (2)$$

The steps of ALC-PSODE algorithm for minimizing the function f ()is shown in Fig. (2).

4.Test functions and experimental results

Real problems nowadays are more and more complex. Their objective functions are often multimodal with peaks, valleys, channels, and flat hyper planes of different heights. Solving these types of problems, which are classified as global optimization problems, to optimality undoubtedly, becomes a true challenge. Test functions have many characteristics for simulating the complexity of most real applications. For example; multimodal functions are used to test the ability of an algorithm to escape from any local minimum. If the exploration process of an algorithm is poorly designed, then it cannot search the function landscape effectively. This, in turn, leads to an algorithm getting stuck at a local minimum.

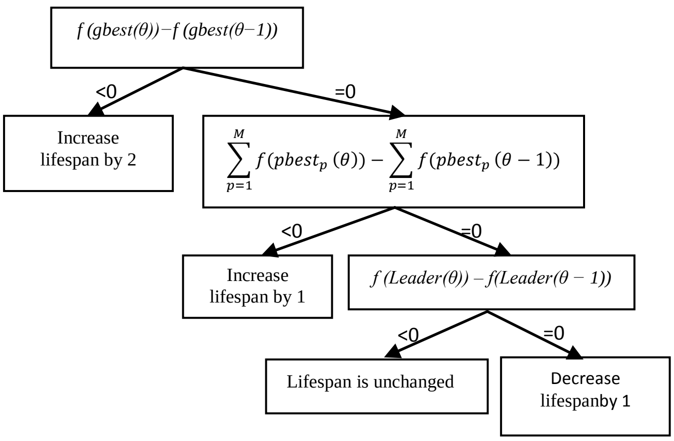


Fig.1. Decision tree for the lifespan controller to adjust the lifespan (The Age of the leader $\theta = 1,2,\ldots.,lifespan$)

```
Begin
    Randomly initialize the position and velocity of all particles in the swarm
    Evaluate the objective value for each particle
    Set Leader = gbest
    Set the age of the Leader θ = 0.
    Set the lifespan of the Leader = initial value.
  Repeat
    For each particle p in the swarm
        Select two other particles p1 and p2 randomly
        Construct the difference vector δ
        Perform crossover between updated velocity vector {eq. (2)} and velocity vector of the previous iteration for the particle p with probability CR
        Create trial vector Tr_p
        Select the particle for next iteration based on competing between trial vector and old position vector ( i.e. select the vector which has less evaluation value )
    End for
        Update Leader to be the best position in the iteration as in conventional PSO, but the Leader represents the best solution generated by particles during the Leader's lifetime.
        Adjust lifespan, as shown in Fig.1
        Increment the age of the Leader
        Test lifespan if it has been exhausted or not
          If lifespan is exhausted
            Then   generate a challenger
            For T iteration
            For each particle
            Update velocity and position using a challenger as a Leader
              Evaluate the objective value for each particle
                Update a challenger
            End for
                If  any position is improved    // a challenger succeed
              Then   replacethe Leader by the challenger;
                Set the age of the Leader θ = 0.
                    Set the lifespan of the Leader = initial value
                    Return
                End if
              End for
            Else // a challenger failed
              Leader is not changed and old status is resumed
              Set the age of the Leader= lifespan-1.
    End if
 Until maximum iteration is reached
    gbest is the solution
End
```

Fig. 2. Pseudo code of ALC-PSODE algorithm

4.1. Benchmark and experimental settings

ALC-PSODE is tested against five well-known benchmarks functions to evaluate the performance of the proposed algorithm. These test functions are shown in Table 1. The first two test functions are unimodal, having only one minimum. The others are multimodal with many local optima.

The proposed algorithm has been tested against the canonical PSO,GA, DE (rand/1/bin), ALC-PSO and PSO-DV, algorithms. The number of dimensions of all the test functions is set to n = 30.*The population size =100.*

In the experiment, the parameters of PSO, ALC-PSO and ALC-PSODE are set as follows: the acceleration coefficients $c_1 = c_2 = 2.0$, the inertia weight ω = 0.4 [28], the initial value of lifespan = 30, the number of steps T for evaluating the leading power of a challenger is set to T = 3 and the legal velocity range is set to 50% of the search range. In the case of PSO-DV and ALC-PSODE, we choose the crossover constant CR = 0.9 and the scale factor ß is linearly varying from 0.9 at the beginning of the search to 0.4 at the end of the search. For DE, the crossover constant CR = 0.9 and the scale factor F = 0.8. In the case of GA, roulette wheel selection operator, single point crossover and adaptive mutation were employed [43,44].

To reduce statistical errors, each test is repeated 50 times independently. During each run, a maximum number of 10000 function evaluations (FEs) are used. The reliability of search is reflected by the "success%" in Table 2, which stands for the percentage of the successful runs that acceptable solutions are found. Each run is considered to be success if and only if the best solution found by an algorithm achieves the predetermined accuracy level in Table 1.

Table 1[*]

Test Functions

Test Function		Range of search	Optimum ±accuracy
Unimodal Functions (2)			
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	[−100,100]	0±0.1
Rosenbrock's	$f_2(x) = \sum_{i=1}^{n-1} [100\,(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[−10,10]	0±100
Multimodal Functions (3)			
Rastrigin	$f_3(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[−5.12,5.12]	0±100
Ackley	$f_4(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + e$	[−32,32]	0±0.1
Griewank	$f_5(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x}{\sqrt{i}}\right) + 1$	[−600,600]	0±0.1

[*]For more details about test functions browse: http://www3.ntu.edu.sg/home/EPNSugan/

4.2. Comparison results

The following performance measures are used for our comparative study:

- Success rate (number of successful runs/total runs).
- Quality of the final solution.
- Speed of convergence towards the optimal solution.

In Table 2, the best solution, worst solution, mean and standard deviation yielded by different algorithms for 50 independent runs are reported. It can be seen that only ALC-PSODE, ALC-PSO and GA are able to find acceptable solutions with a 100% successful rate. We compare between ALC-PSODE and other two algorithms using two-sample *t*-tests. Two-sample t-test is a hypothesis testing method used for assessing whether the means of two independent samples are statistically different from each other [38, 39]. According to the results of *t*-tests (The difference between two samples is significant at level α=0.05, sample size = 50 and degrees of freedom = 98), ALC-PSODE significantly outperforms ALC-PSO on f_2, f_5. Also, ALC-PSODE significantly outperforms GA on the first four functions. In Fig. 3 we have graphically presented the rate of convergence (for an average run) of all methods for all functions. Although, the performance of all algorithms were comparable to each other regarding to all test bench mark functions, the performance of ALC-PSODE is super exceeded the performance of other algorithms regarding to f_2, f_3 and this is shown in Fig. 3. These results show that the proposed algorithm leads to significant improvement in most cases.

5. A case study: liquid level system control

After validation of ALC-PSODE algorithm on test benchmark functions, it will be used for tuning PID controller. This controller, for simplicity, is called ALCPSODE-PID controller. For testing on real application, ALCPSODE-PID controller is used for controlling 3 tanks liquid level system which is a typical nonlinear complexcontrol system. Liquid level control is very important in many industrial applications as in water purification systems, industrial chemical processing and boilers in all the industries.

5.1. Linearized model of three tanks liquid level system

A simple structure of the water tank is shown in Fig.4. In this device, there are three tanks: tank A, tank B, and tank C. Tank D is the main tank, which provides the water for the pump. The control actuator is an electric valve. The control signal will change the open range of the electric valve from 0% to 100%. Different open range of electric valve means different water flow rate.

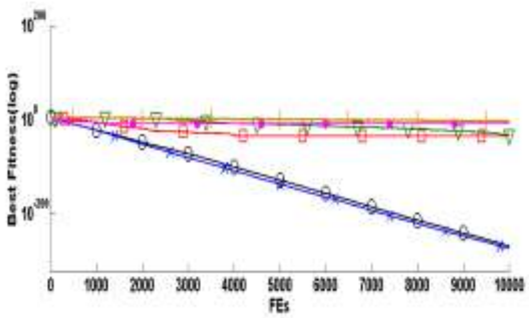
From the simple structure of Fig.4, the main principle of water level system can be explained as following: Driven by pump, the water in tank D is piped to tank A, B and C. Water level will be measured by pressure sensor at bottom of each tank. The difference between actual level and set-point value is calculated. Then the control input can be obtained by PID algorithm. The control input will be feedback to electric valve to change the flow rate, and then the level of water in each tank can be controlled [31].

Table2*

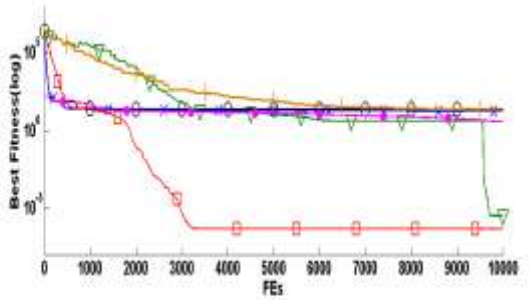
Results obtained for 50 runs

function		PSO	GA	DE	PSO-DV	ALC-PSO	ALC-PSODE
f_1	best	3.2878e-282	1.8600e-009	1.0432e-004	8.6345e-037	**4.4586e-285**	8.8404e-056
	worst	3.3699e-269	1.3500e-008	0.0024	4.7471e-032	**2.7508e-273**	1.5182e-033
	mean	1.0967e-270	5.7207e-009	8.1551e-004	2.7634e-033	**1.6095e-274**	3.5322e-035
	Std deviation	**0**	3.1072e-009	5.1190e-004	8.3100e-033	**0**	2.1480e-034
	Success %	**100**	**100**	**100**	**100**	**100**	**100**
	t test		**-13.0186**			1.1628	
f_2	best	0.1255	4.6300e-005	16.9436	**2.3484e-022**	1.6790e-004	2.4479e-021
	worst	96.9987	6.4701	22.0152	8.3210e-004	77.7372	**1.0906e-005**
	mean	33.4918	1.3229	19.6060	1.6706e-005	17.3852	**3.6305e-007**
	Std deviation	30.3198	2.2989	1.1931	1.1767e-004	15.9752	**1.6084e-006**
	Success %	92	**100**	**100**	**100**	**100**	**100**
	t test		**-4.0690**			**-7.6952**	
f_3	best	16.9143	2.9849	57.0098	**0**	**0**	1.7764e-015
	worst	97.5765	10.9446	99.5367	99.4949	6.8350e-008	**1.2736e-012**
	mean	64.7968	6.4009	82.4987	50.7627	1.3677e-009	**4.7855e-014**
	Std deviation	20.2555	2.3324	12.3051	25.8691	9.6661e-009	**1.7978e-013**
	Success %	90	**100**	34	98	**100**	**100**
	t test		**-19.4054**			-1.0005	
f_4	best	**4.4409e-015**	2.9400e-005	5.2288e-010	1.1546e-014	**4.4409e-015**	7.9936e-015
	worst	**7.9936e-015**	6.7400e-005	3.4389e-008	2.2204e-014	**7.9936e-015**	1.5099e-014
	mean	**7.1794e-015**	5.3303e-005	7.5172e-009	1.7838e-014	7.6383e-015	9.5568e-015
	Std deviation	1.5090e-015	9.3242e-006	1.1973e-008	3.6819e-015	**1.0766e-015**	2.9733e-015
	Success%	96	**100**	14	96	**100**	**100**
	t test		**-40.4227**			4.2	
f_5	best	**0**	4.4400e-011	3.2851e-004	**0**	**0**	**0**
	worst	0.0638	**4.5900e-010**	0.0213	0.0638	0.0515	0.0344
	average	0.0147	**2.3658e-010**	0.0037	0.0123	0.0100	0.0031
	Std deviation	0.0159	**9.5324e-011**	0.0044	0.0135	0.0110	0.0060
	Success %	**100**	**100**	**100**	**100**	**100**	**100**
	t test		3.6534			**-3.8939**	
ALC-PSODE obtains significantly better results			**4**			**2**	
ALC-PSODE obtains significantly worse results			1			1	

*** Best solution, worst solution, mean and standard deviation are calculated for only success runs.**
*** Bold numbers indicate the featured results (i.e. success rate=%100, significantly better results for t test of ALC-PSODE and the minimum (best) results in best, worst, mean and standard deviation).**



(a)



(b)

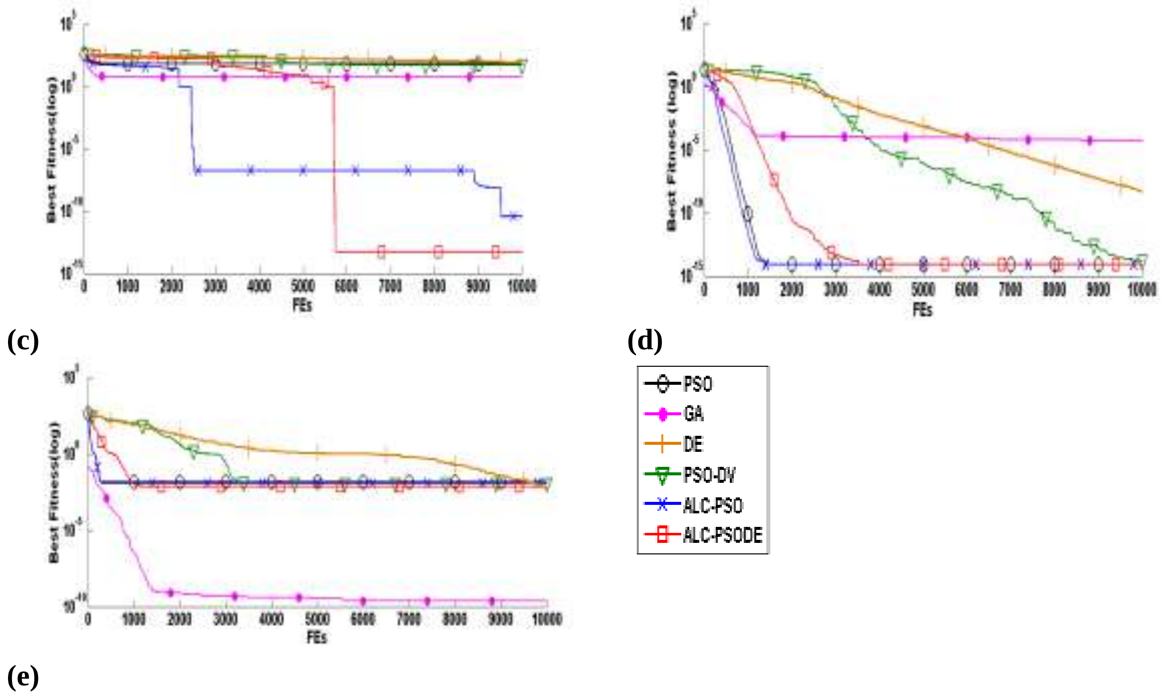


Fig. 3. Convergence graphs on test functions (a) f_1. (b) f_2. (c) f_3. (d) f_4. (e) f_5

To analyze the mechanism of Water Level System, the following symbols should be considered first:

Q_1—inflow of water tank;
Q_2—outflow of water tank;
A— area of tank's section;
VO— valve's opening range;
LH— water level height;
K_1—valve flux proportional coefficient;
K_2—flux proportional coefficient

Here only the dynamic characteristic of Tank C is analyzed. Consider that A,K_1,K_2 are time invariant parameters,VO_0, LH_0 are the input and output value of an equilibrium point, then from Fig.5 we have:

$$Q_2 - Q_1 = A(dLH/dt) \qquad (3)$$

$$Q_1 = K_1 \times VO \qquad (4)$$

$$Q_2 = K_2 \times \sqrt{LH} \qquad (5)$$

From eq.(3),eq.(4) and eq.(5), it can be seen that the water level is a system with nonlinearity. Around the equilibrium point (VO_0, LH_0) , eq.(5) can be rewritten approximately as:

$$\frac{Q_2}{LH} \approx \frac{K_2}{\sqrt{LH_0}} = K_3 \qquad (6)$$

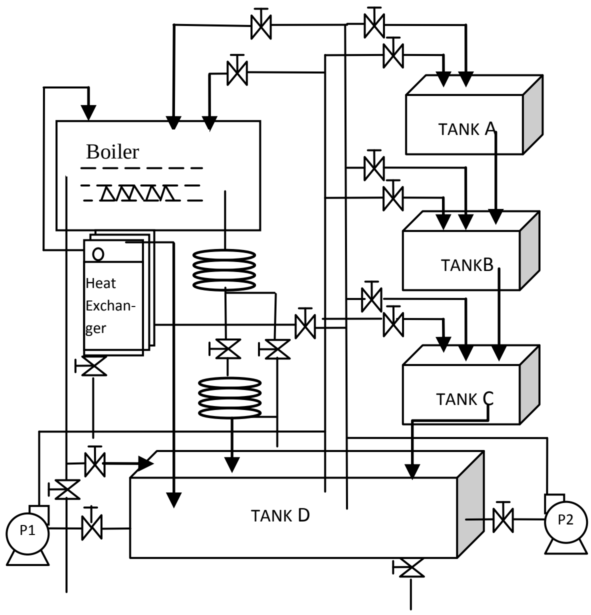


Fig.4. Simple structure of the water level system

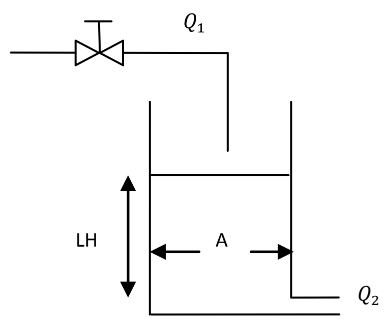


Fig.5. Water level system with control valve

Calculate the Laplace Transformation of eq.(3), eq.(4) and eq.(6) .From the Laplace equations, linear model (around the equilibrium point (VO_0 , LH_0)) with input VO, output LH can be given as below:

$$G(s) = \frac{LH(s)}{VO(s)} = Ke^{-\tau s} /(Ts + 1)$$

where$K = K_1 /K_3$, $T = A/K_3$, τ is the inherent time delay.

Consider that A= 20, $K_1 = 2$, $K_2 = 5, \tau = 0$; the system will come out to be third order dynamics model. Desired transfer function of three tank water level system is [32]:

$$\frac{H(s)}{Q(s)} = \frac{1}{64s^3 + 9.6s^2 + 0.48s + 0.008}$$

5.2. Liquid level system with PID controller

ALC-PSODE algorithm is used to find for the optimal PID parameters [K_p, K_i , K_d] that will minimize the objective function ISE (integral of squared-error). The ISE performance criterion formula is as follows:

$$ISE = \int_0^\infty (r(t) - h(t))^2 = \int_0^\infty e^2(t)\, dt$$

Three Tanks Liquid Level System compensated with a PID controller block diagram is shown in Fig. 6.

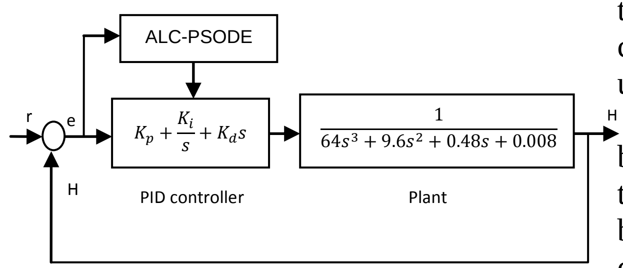


Fig.6. Block diagram of Intelligent PID controller

6. Results

6.1. Performance of ALC-PSODE controller

The following PSO parameters are used for verifying the performance of the various PSO-PID controllers:

Inertia weight ω = 0.4 [28];

Acceleration constant c1 and c2 =2;

Crossover Constant CR = 0.8;

Scale Factor ß = 0.8 [28]

For GA, roulette wheel selection operator, single point crossover and adaptive mutation were employed. In the case of DE, the crossover constant CR = 0.9 and the scale factor F = 0.8 [28].

We performed 30 independent runs for various controllers with different random numbers, different swarm size and different number of generations to observe the variation in their evaluation values. Each run is considered to be success if and only if the best solution (i.e. of the objective function ISE) found by an algorithm is less than 30 [2]. In addition, the best, worst, and average evaluation values were obtained. Results were shown in Table 3. As can be seen, PSO variants can get success rate equal 100% in all cases. GA and DE fail in some runs to get successful evaluation values. This result is due to the internal structure of GA and DE that degrades their performance [33, 34, 41]. In GA; with continue of generations; crossover usually has less effect, and the resulting movements are relatively smaller. This cause a premature convergence and loss of diversity [34]. Also, DE is easy to drop into local optima because of its fast convergence. To solve premature convergence problem of DE, it is desired to increase the population size for increasing the diversity but this will increase the computation time. So, hybrid DE is always used to overcome its drawback [41].

rom the table, all algorithms can get the best solution except PSO. We can observe that ALC-PSO can get betterresult than PSO but it cannot solve the premature convergence problem of PSO completely. This was exactly done when testing benchmark function. The evaluation values of the ALCPSODE-PID controller generated fluctuation in a very small range (std. dev. = 8.2827e-015), thus verifying that the ALCPSODE-PID controller has better convergence characteristic and hence it is the most robust algorithm. The simulation results that showed the best solution (i.e. the comparison has been done using transient-response specifications [40] of the system in addition to ISE) and step response of the three tanks liquid level system for best resultswere summarized in Table 4 and Fig.7 respectively.

6.2.Convergence characteristic

As can be seen from Table 3, only PSO variants can get %100 success rate. So, we will study the convergence characteristics of PSO variants for least population size. For fair comparisons, the same seed of random numbers in initialization of the population has been used. Convergence characteristic of the various PSO-PID controllers is shown in Fig.8. We can see that although conventional PSO has fast convergence characteristics as expected, it cannot get the best result. Fig. 8 shows that PSO-DV, ALC-PSO and ALC-PSODE can get the best result but ALC-PSODE has the best convergence among them. Through about 15 iterations (15 generations), the ALCPSO-DEmethod can achieve fast convergence and obtain good evaluation value.

To study convergence behavior of ALC-PSODE clearly, we observe the variation in its best evaluation values against various successful algorithms for 30 runs. The results are shown in Fig. 9. The results show that ALC-PSODE generated fluctuation in a small range, thus verifying that ALC-PSODE has better convergence characteristic. These discussed results show that the ALCPSODE-PID controller can search optimal PID controller parameters efficiently and quickly.

Table 3
Results obtained for 30 run

Different PID controller	Swarm size	of generatio	best	worst	average	Standard deviation	Success %
PSO	20	50	8.7016	28.020	9.3455	3.5271	100
GA			7.7925	28.0392	8.9846	3.9128	90
DE			7.7922	10.5203	10.1440	0.9574	96.67
PSO-DV			**7.7922**	**7.7922**	**7.7922**	1.49e-07	100
ALC-PSO			**7.7922**	8.7016	8.1605	0.4497	100
ALC-PSODE			**7.7922**	**7.7922**	**7.7922**	**1.8e-010**	100
PSO	30		8.7016	8.7016	8.7016	8.47e-014	100
GA			**7.7922**	10.8067	8.0341	0.6064	100
DE			**7.7922**	10.5203	9.5200	1.3372	100
PSO-DV			**7.7922**	**7.7922**	**7.7922**	3.40e-09	100
ALC-PSO			**7.7922**	8.7016	7.8638	0.2301	100
ALC-PSODE			**7.7922**	**7.7922**	**7.7922**	**2.01e-011**	100
PSO	20	100	8.7016	24.423	9.2256	2.8704	100
GA			**7.7922**	8.3474	7.8358	0.1230	96.67
DE			**7.7922**	10.5203	9.9747	1.1099	100
PSO-DV			**7.7922**	**7.7922**	**7.7922**	2.85e-07	100
ALC-PSO			**7.7922**	8.7016	7.9553	0.3020	100
ALC-PSODE			**7.7922**	**7.7922**	**7.7922**	**5.62e-015**	100
PSO	30		8.7016	8.7016	8.7016	4.26e-014	100
GA			**7.7922**	**7.8157**	**7.7932**	0.0043	100
DE			**7.7922**	10.5203	9.8838	1.1736	100
PSO-DV			**7.7922**	**7.7922**	**7.7922**	2.07e-08	100
ALC-PSO			**7.7922**	8.6602	7.8788	0.2629	100
ALC-PSODE			**7.922**	**7.922**	**7.922**	**8.28e-015**	100

Table 4
Best solution using various controllers

	Ziegler and Nichols	PSO	GA, ,DE, PSO-DV, ALC-ALC-PSODE
K_p	0.03843444	0.0528	**0.0419**
K_i	.0010588	0.0003	**0.0009**
K_d	0.17396084	1	**1**
ISE	32.7064	8.7016	**7.7922**
Overshoot	58.5408	15.4004	**12.4671**
Rise Time	18.1142	**12.0572**	12.7622
Settling Time	476.6251	267.5225	**64.2164**
Peak	1.6262	1.1457	**1.1247**
Peak Time	48.5000	**25**	26.6000

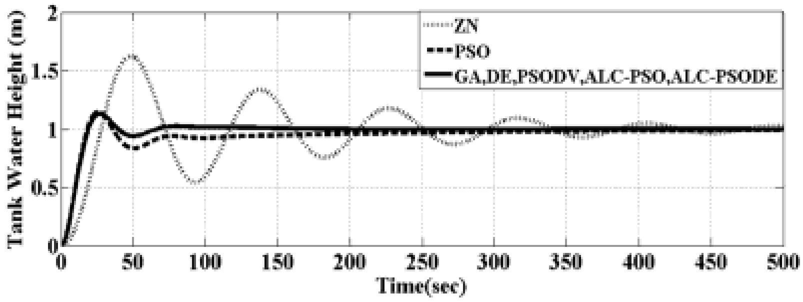


Fig.7 Step response of 3 tanks liquid level system with the various PID controllers for best result.

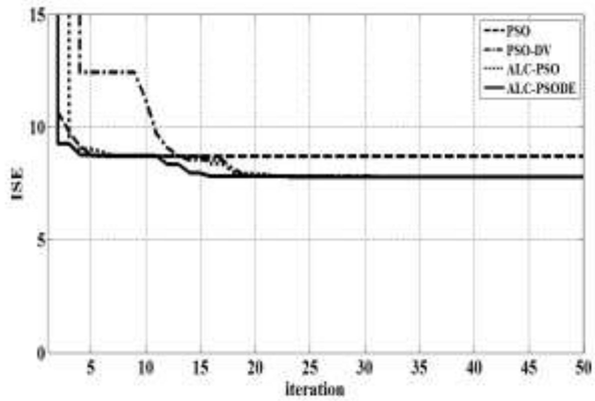
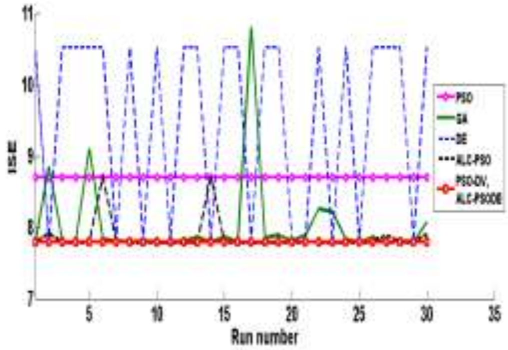
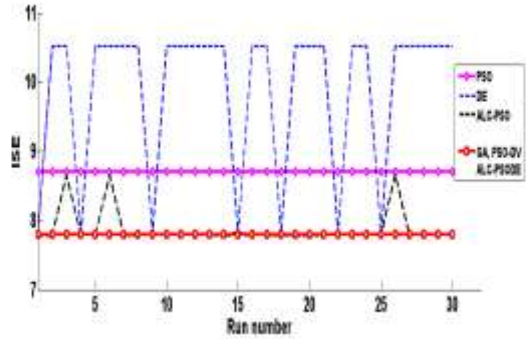


Fig. 8 Convergence tendency of best particle for various PSO-PID controllers (popultion size=20, number of generatons=50)



(a) popultion size=30, number of generatons=50



(b) popultion size=30, number of generatons=100

Fig.9 Comparison of best values of various PID controllers for 30 successful runs

7.Conclusion and future works

In this paper a new PSO variant called ALC-PSODE has been presented and has been shown to improve performance in a statistically meaningful way. The new method has been compared against different PSO variants, GA, DE using five well known benchmarks functions. Then the new algorithm has been used in tuning PID controller to control three tanks liquid level system which is typical nonlinear problem. The ALC-PSODE- PID controller is compared against various PID controllers; the results show that the proposed controller can perform an efficient search for the optimal PID controller parameters. For future research, it would be interesting to test the new algorithm on the multiobjective optimization problems where more diversity is required.

References

[1] J.G. Ziegler , N. B. Nichols, Optimum settings for automatic controllers, ASME Trans. (1942) 759–768.

[2] Zwe-Lee Gaing, A particle swarm optimization approach for optimum design of PID controller in AVR system, IEEE Trans. Energy Conv. 19 (2) (2004) 384-391.

[3] Wang Jingi, Xue Yang, Chen Lei ,The application of GA-based PID parameter optimization for the control of superheated steam temperature, Int. Conf. on Machine Learning and Cyber. (ICMLC), 2012.

[4] L. S. Coelho, M. W. Pessôa, A tuning strategy for multivariable PI and PID controllers using differential evolution combined with chaotic Zaslavskii map, Expert Systems with Applications 38 (11) (2011) 13694-13701.

[5] Queen, A.A. ; Auxillia, D.J. ,Simplified discrete binary PSO tuned multivariable PID controller for binary distillation column plant, Int. Conf. on Circuits, Power and Computing Technologies (ICCPCT), March 2013.

[6] J. Lin, J.Huang , G. Chen, Design of DE-tuning PID controller for linear servo system, IEEE Fifth Int. Conf. Adv. Comput. Intell. (ICACI), Nanjing ,2012,pp. 437 – 440.

[7] Zhang Jun , Zhang Kanyu, A particle swarm optimization approach for optimal design of PID controller for temperature control in HVAC, Third Int. Conf. on Measuring Technology and Mechatronics Automation (ICMTMA),Shangshai, 2011.

[8] J. Kennedy , R. C. Eberhart, Particle swarm optimization, Proc. IEEE Int. Conf. Neural Netw. ,Perth,1995, pp.1942 - 1948.

[9] R. C. Eberhart, y. shi, PSO: developments, applications and resources, Proc. of the 2001 Cong. on Evol. Comput. ,Seoul, 2001, pp.81-86 .

[10] F. Wang ,Xing-shi He ,L. Luo , Y. Wang, Hybrid optimization algorithm of PSO and Cuckoo Search,Int conf. on Art. Intell. Manag. Sci.andElectr. Comm., 2011,pp.1172 – 1175.

[11] W. Xu, X. S. Gu, A hybrid particle swarm optimization approach with prior crossover differential evolution, Proc. 1st ACM/SIGEVO Summit Genetic Evol. Comput.,Shanghai, China, 2009, pp. 671–677.

[12] B. B. Li, L.Wang, B. Liu, An effective PSO-based hybrid algorithm for multiobjective permutation flowshop scheduling, IEEE Trans. Syst. Man Cybern. A, Syst. Humans. 38 (4) (2008) 818–831.

[13] C. F. Huang, A hybrid of genetic algorithm and particle swarm optimization for recurrent network design, IEEE Trans. Syst. Man Cybern. B. 34 (2) (2004) 997–1006.

[14] N. Holden , A. A. Freitas, A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data , Proc. IEEE Swarm Intell. Symp. ,2005,pp.100-107.

[15] P. S. Shelokar, P. Siarry, V. K. Jayaraman, B. D. Kulkarni, Particle swarm and ant colony algorithms hybridized for improving continuous optimization, Appl. Math Comput. 188 (1) (2007) 129–142.

[16] J B. Niu and L. Li, A novel PSO-DE-based hybrid algorithm for global optimization, in: Proc. 4th Int. Conf. Intell. Comput., ,Shanghai, China, 2008, pp.156–163.

[17] H. Zhang, L. Fang, Y. Cen, Comparison among three kinds of hybrid particle swarm optimization algorithms, Cont. and Dec. Conf. (CCDC) ,China,2011, pp.3422 – 3425.

[18] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Global Optimization j. 11(4) (1997) 341-359.

[19] S. Das, P. N. Suganthan,Differential Evolution: A survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4-31.

[20] B. Xin, J. Chen, J. Zhang, H. Fang, Z. Peng, Hybridizing differential evolution and particle swarm optimization to design powerful optimizer:a review and taxonomy, IEEE Trans. Syst.ManCybern. C :application and reviews. 42 (5) (2011)744-767.

[21] S. Das, A. Konar, U. K. Chakraborty, Improving particle swarm optimization with differentially perturbed velocity, in: Proc. Genetic Evol. Comput. Conf. ,Washington, 2005, pp.177-184.

[22] Li Xiao , X. Zuo, Multi-DEPSO: A DE and PSO based hybrid algorithm in dynamic environments, IEEE Cong. Evol. Comput. (CEC), Brisbane , 2012, pp.1-7.

[23] j.Grobler , A.P. Engelbrecht, Hybridizing PSO and DE for improved vector evaluated multi-objective optimization, IEEE Cong. Evol. Comput., Trondheim,2009 , pp. 1255 - 1262

[24] W. Chen, J. Zhang, Y. Lin, N. Chen, Z. Zhan, H. S. Chung, Y. Li, Y. Shi, Particle swarm optimizer with aging operator for multimodal function optimization, Taylor & Francis Int. J. Comput. Intell. Sys. 6 (5) (2013) 862-880.

[25]S. Dehuri, A.Ghosh,R.Mall, Particles with age for data clustering, Int. conf. Inform. Tech. (ICIT), 2006, pp.221 – 224.

[26] R. Kundu, R. Mukherjee, S. Das, Modified Particle Swarm Optimization with Switching Update Strategy, Swarm, Evol., and MemeticComput.,springer,Berlin Heidelberg, 2012, pp 644-652 .

[27] B. jiang, N. Wang, L. Wang, Particle swarm optimization with age- group topology for multimodal functions and data clustering, Comm.. Nonlin. Sci. and Numeric. Sim. 18 (11) (2013) 3134-3145.

[28] W. Chen, J. Zhang, Y. Lin, N. Chen, Z. Zhan, H. S. Chung, Y. Li, Y. Shi, Particle swarm optimization with an aging leader and challengers, IEEE Trans. Evol. Comput. 17 (2) (2013) 241-258.

[29] K.Vaisakh, LR. Srinivas, Adaptive PSODV algorithm for OPF with non-smooth cost functions and statistical analysis, Simulation Modelling Practice and Theory. 19 (9) (2011) 1824-1846.

[30] K.Vaisakh,L.R.Srinivas,KalaMeah, Genetic evolving ant direction PSODV hybrid algorithm for OPF with non-smooth cost function, Springer Electr. Eng. 95 (3) (2012) 185-199.

[31] Xiaoli Li, Ji Li, Longhui Shi, Modelling and Simulation of Water Level System, Proceedings of the IEEE Inter. Conf. on Automation and Logistics ,Qingdao, China, 2008,pp. 2857-2859.

[32] B. Kumar,R. Dhiman, Tuning of PID Controller for Liquid Level Tank System using Intelligent Techniques, IJCST. 2 (4) (2011) 257-260.

[33] D. B. Fogel, Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, second ed., IEEE Press, New York, 2000.

[34] Z.Hao , G. Guo , H. Huang, A Particle Swarm Optimization Algorithm with Differential Evolution, Int. Conf.

Machine Learning and Cyber., Hong Kong, 2007, pp. 1031 – 1035.

[35] N A Selamat, N. A. Wahab, S Sahlan, Particle Swarm Optimization for Multivariable PID Controller Tuning, IEEE 9th Int. Colloquium on Signal Processing and its Applications, Kuala Lumpur, Malaysia, 8 - 10 Mac. 2013.

[36] M. W. I. and S. Baskar, "Evolutionary algorithms based design of multivariable PID controller," Expert Systems with Applications, p. 9, 2009.

[37] A. Visioli, Fuzzy Logic Based Set-Point Weight Tuning of PID Controllers, IEEE Trans. on sys., man, and cybernetics—part a: sys. and humans, 29(6) (1999)587-592.

[38] R. E. Walpole and R. H. Myers, Probability and Statistics for Engineers and Scientist. New York: Macmillan, 1978.

[39] http://www.mathworks.com/help/stats/ttest2.html

[40] K. Ogata, Modern Control Engineering, fifth ed., Prentice Hall, newgersy, 2010.

[41] Y.Jiang, N. R. Chapman and P.Gerstoft , Estimation of Geoacoustic Properties of Marine Sediment Using a Hybrid Differential Evolution Inversion Method, IEEE J. of OCEANIC ENG., (35)(1)(2010) 59-69.

[42] A. El-Gammal, A.El-Samahy, Adaptive Tuning of a PID Speed Controller for DC Motor Drives Using Multi-Objective Particle Swarm Optimization, IEEE UKSim 2009: 11th Int.Conf. on Computer Modelling and Simulation, Cambridge, 2009,pp.398-404.

[43] M. A. Elhosseini, A.Y. Haikal, Modified hybrid algorithm for process optimization, App. Math. andComput., 232 (2014) 1209–1224.

[44] http://www.mathworks.com/help/gads/index.html.