Particle Swarm Intelligence Based PID Position Control System

SUNDAY ILIYA[1], TIMOTHY AFIAGH[2], OLUROTIMI OLAKUNLE AWODIJI[3]
[1, 2, 3] *University of Jos*

Abstract- ***This paper present a robust and efficient way of tuning PID controller using three variants of swam intelligence algorithms for control of a positioning system. Out of the three variants implemented, toroidal bound comprehensive learning particle swarm optimization (CLPSO) appear to be more promising in addressing this problem with peak overshot of 0.0176, rise tie of 0.01s, setting time of 0.01s and combined cost function of 0.0134 followed by toroidal bound inertia PSO. The results obtained using the swarm intelligence algorithm variants outperform those of Deferential Evolution (DE) variants used in solving the similar problem as presented in [7].***

Indexed Terms- ***Swarm intelligent algorithms, PID controller, Step response, Ziegler–Nichols tuning method, optimization, objective fitness function.***

I. INTRODUCTION

One of the major challenge of any positioning control system is the ability to coup with unpredictable changes resulting from within the system or its environment. Positioning systems are face with different challenges depending on their application and the environment they are designed to be used, among these are disturbance from natural events such as wind, unpredictable change in position of their target for non-stationary target, etc. To mitigate the chances of the system missing its target, we proposed a generalised intelligent control schemes for positioning systems that uses DC motor to track their target in a dynamically changing environment. The schemes presented in this paper is based on swarm intelligent optimization framework using PID controller.

II. OPTIMIZATION OR TUNING ALGORITHMS

A brief description of the optimization algorithms implemented are presented in this section. We explore the advantages of global search capability of population-based Swarm Intelligence Algorithms (SIA) variants to evolve the gains of the PID controller. The complexity of many heuristic controllers becomes increasingly complicated due to meta parameters (free parameters) in the model or controller frame work that govern their behaviour and efficiency in optimizing a given problem. How best a given controller can solve a given problem, depends on the correct choice of the meta parameters. The values of those parameters are problem dependent, thus for each problem, those parameters need to be fined tune to get the optimum or near optimum. The tuning pose another optimization problem. The PID gains of the positioning system depicted in this paper were optimized using population-based randomization optimization algorithms based on swarm intelligent framework.

2.1. Comprehensive Learning Particle Swarm Optimization (CLPSO)

Two algorithms which are based on swarm intelligence framework were implemented for tuning the PID controller. These two-particle swarm optimization (PSO) variants are: the standard PSO with inertia weight [8] and the CLPSO [3]. PSO emulates the swarm behaviour of which each member of the swarm adapts it search path by learning from its own experience and other members' experiences. The velocity update of PSO and CLPSO are giving by Eq. (2) and (1) respectively the particle update for both PSO and CLPSO is given by Eq (3). In the inertia weighted PSO, each of the particles learn from its local best pbest and the global best gbest for all dimension. The parameters C_1 and C_2 are the acceleration constants that reflect the weighting of the stochastic

acceleration term that pull each particle toward pbest and gbest respectively. The inertia weight w is used to facilitate both global and local search. Large w facilitate global search while smaller values favoured local search. In this study w was made to decrease exponentially as the generation progresses. This approach facilitate global search within the early stage (generations) and then start to favour local search as the budget (generation) comes to an end. In standard PSO, all particles learn from its own pbest and gbest for all dimension. Constraining the social learning aspect to only the gbest lead to premature convergence of the original PSO. Since all particles in the swarm learn from the current gbest even if the gbest is very far from the global optimum. Thus all the particles stand the risk of been attracted to gbest and get trapped in a local optimum especially when solving complex problems with numerous local optimums. To circumvent the problem of premature convergence associated with the standard PSO, a CLPSO was proposed [3]. In CLPSO, instead of particles learning from its pbest and gbest for all dimensions, and for all generations, each element (dimension) of a particle can learn from any other particle's pbest including its own pbest. The decision on whether a particle's dimension should learn from its own pbest or other particles' pbest depends on the probability $P_c$called learning probability. Each particle has its own P_c. For every dimension of particle i a random number in the range [0, 1] is generated, if this random number is greater than P_{ci}, the particular dimension will learn from its own pbest otherwise it will learn from another particle's pbest. To ensure that at least one of the dimension of each particle learn from another particle's pbest, if all dimensions happen to learn from its own pbest, one dimension is pick at random and two particles are pick at random from the population, the selected dimension will learn from the corresponding dimension of the particle with the best fitness (pbest). In this study, P_{ci} is given by Eq. (4), [3].

$$V_i^d = w_i.V_i^d + C_1.rand1_i^d\left(pbest_i^d - X_i^d\right) + C_2.rand2_i^d(gbest_i^d - X_i^d) \quad (1)$$

$$V_i^d = w_i.V_i^d + C.rand_i^d\left(pbest_{f_i(d)}^d - X_i^d\right) \quad (2)$$

Where $pbest_{f_i(d)}^d$ is any particle's pbest including particle i pbest. $f_i = [f_i(1), f_i(2), \ldots, f_i(D)]$ defined which particles' pbests the particle i should learn from (follow). D is particle dimension, $rand^d_i$ is a random number in the range [0, 1], and each particle dimension d has its own $rand^d_i$. X referred to particles' positions (potential solutions) while C is the acceleration pull.$V_i$is the velocity of particle i.

$$X_i^d = X_i^d + V_i^d \quad (3)$$

$$P_{ci} = 0.05 + 0.45\left(\frac{e^{\frac{10(i-1)}{SS-1}} - 1}{e^{10} - 1}\right) \quad (4)$$

Where SS is the swarm size (number of particles).

1.2. Selection process

When the particles are updated, the fitness of each updated particle $f(X_i)$ is compared with the fitness of its local best $f(pbest_i)$, to determine the next generation local bests. Iff(X_i) < $f(pbest_i)$the updated particle X_i will replaced its local best $pbest_i$ in the next generation,otherwise the local best will be allowed to continue in the next generation. This scheme is based on the principles of survival of the fittest. The fitness of each local best $f(pbest_i)$is further compared with that of the global best f(gbest). If $f(pbest_i) < f(gbest_i)$ the global best gbestwill be replaced by the particular local best $pbest_i$otherwise it will be maintained in the next generation. The final global best is used to control the system. The tuning fitness function used in this research is the weighted sum of the overshot (over or under shot), rise time and the settling time when a unit step input command is used.

1.3. Fitness Function Evaluation

The optimization problem presented in this paper is a multi-objective optimization problem since there are three cost functions we want to minimise i.e. the maximum overshot (M_o), rise time (T_r) and settling time (T_s). In order to get a robust controller gains, the problem is converted to single objective problem with one cost function consisting of the weighted sum of the three objective functions, Eq (5). The weights depends on the important or cost of risk resulting from that particular performance index. This approach is robust because different models can be evolved by just changing the weight to meet up with setting system performance specifications.

$$\gamma = \alpha_o M_o + \alpha_r T_r + \alpha_s T_s \quad (5)$$

Where: γ is the fitness function, M_o is the maximum overshot, $T_r$is the rise time and T_s settling time, while

α_o, α_r, and α_s are their weights respectively. For this research, after a manual tuning, the following values were used with M_o having the highest priority, α_o =1, α_r=0.5, and α_s=0.4. Note the maximum value the weight can take for this application is 1.

III. PROPORTIONAL PLUS INTEGRAL PLUS DERIVATIVE (PID) CONTROLLER

It is interesting to know that nearly half of the industrial controllers used today are PID or modified PID or derivatives of PID controllers. Some intelligent controllers e.g. Fuzzy logic or adaptive fuzzy logic are derivatives of basic PID i.e. they make use of the error and its derivative (rate of change of the error). There are different variant of the PID controller, the one used in this research is given by Eq. (6) while the transfer function $G_c(s)$ of the controller is depicted by Eq. (7) [2][1][4][5]. A proportional controller will have the effect of reducing the rise time, but will not eliminate the steady-state error. Because of the present of pole at the origin introduced by the integral controller, the integral controller will have the capability of eliminating the steady-state error, but it may make the transient response worse. The derivative controller will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. The derivative controller predict future error using the rate at which the error is changing while the integral captured the cumulative effects of past errors to improve the system performance.

$$PID = K_p(e(t) + \frac{1}{T_i}\int_{t_0}^{t} e(t)\,dt + T_d\frac{de(t)}{dt}) \quad (6)$$

$$G_c(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \quad (7)$$

Where: t is time, e(t) is present error at time t, K_p is the proportional gain while T_i and $T_d$are integral and derivative time constants respectively, s is Laplace complex notation.

Tuning of the PID gains (K_p, T_i and T_d) Ziegler–Nichols

The process of selecting the controller parameters K_p, T_i and T_d to meet a given performance specifications is known as controller tuning. Different variants of population based swarm intelligence algorithms (SIA) were used to evolve the PID gains. One of the major challenge is to define the decision search space i.e. the range within which each of the meta parameters (K_p, T_i and T_d) of the controller should be searched. To address this problem, Ziegler–Nichols tuning method was used to obtain the centre of the radius of the search space. The Ziegler–Nichols reference gains were obtained using the mathematical model of the positioning system shown in Fig. (2). The centre of the radius for search of the gains K_p, T_i and T_d are given by equations (8), (9) and (10) respectively [2].

$$K_p = 0.6K_{cr} \quad (8)$$

$$T_i = 0.5P_{cr} \quad (9)$$

$$T_d = 0.125P_{cr} \quad (10)$$

Where K_{cr} and P_{cr} are the critical gain and critical frequency for self-sustained oscillation of the system. The decision search space for each of the gains were obtained as follows:

$$K_{p(space)} = [\alpha_{min}K_p,\ \alpha_{max}K_p] \quad (11)$$

$$T_{i(space)} = [\beta_{min}T_i,\ \beta_{max}T_i] \quad (12)$$

$$T_{d(space)} = [\mu_{min}T_d,\ \mu_{max}T_d] \quad (13)$$

K_p, T_i and T_d are given by equations (8), (9) and (10) respectively while after a manual tuning, the minimum and maximum values of α, β and μ were obtained as follows:

$\alpha_{min} = 0.4,\quad \beta_{min} = 0.2,\ \mu_{min} = 0.2,\ \alpha_{max} = 5,\quad \beta_{max} = 4, \mu_{max} = 4$

3.1. Mathematical model of the positioning system

The rotation of the positioning system to meet up with a given target specifications is achieved using DC motor.

$$V = R_a I_a + L_a\frac{dI_a}{dt} + E_b \quad (14)$$

$$T = J\frac{dw}{dt} + Fw \quad (15)$$

$$E_b = K_b w \quad (16)$$

$$T = K_t I_a \quad (17)$$

$$w = \frac{d\theta}{dt} \quad (18)$$

Where V is motor terminal supply voltage, R_a armature resistance, L_a is armature inductance, I_a is armature current, E_b is back emf (electromotive force), T is the torque, w is the angular speed in rad/s, J is the inertia constant while F is the viscose constant, K_b is

the back emf constant, t is time and Θ is angular position in rad.

The block diagram shown in Fig. 1 was obtain using equations (14) to (18) along with the controller, where Θ_R is the command reference input angle while Θ is the actual output.

IV. RESULTS

Each of the swarm intelligence variant is run for 500 generations consisting of 20 potential candidate solutions (particles). At the end of the generation, the must fitted (best) candidate is used to set the PID gains. The fitness function used during the training is the weighted sum of the maximum overshot, rise time and settling time, Eq. (5). The evolved best candidate was used to control the positioning system using three different approaches, i.e. the system was tested using standard ram and parabolic input command. Thirdly a real world scenario was modelled as a command input to see how the output of the system can track the target input. The performance index used to evaluate the accuracy of the system in tracking the command input is the root mean square error (RMSE) given by Eq. (19) [6]. It is interesting to note that the fact that the system depicted good performance for standard ram and parabolic input with low RMSE does not necessarily mean that the system will perform well when subjected to real world scenario. This is revealed when the untune controller obtain directly using Ziegler–Nichols method was used. The RMSE of ram and parabolic command using untune PID and for toroidal bound comprehensive learning particle swarm optimization (CLPSO) shown in Fig. 5 and Fig. 4 are 0.0095 and 0.0477 respectively while for the tuned PID are 0.0017 and 0.0149 respectively. But when the tuned and the untune PIDs were tested using real world command input, the untune PID perform poorly with RMSE of 3.9376 while the tuned PID followed the command input closely with RMSE of 1.6527 as shown in Fig. 3. This research also validate that PID gains obtained using Ziegler–Nichols method may not be the optimum but is a useful tool for obtaining the radius of the search space within which the optimum or near optimum are likely to be found. The details of the numeric results obtained from the three SIA variants implemented in this research are shown in table 1.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\Theta_{Ri} - \Theta_i)^2} \qquad (19)$$

Where: RMSE is the root mean square error, N is the number of simulation time steps, Θ_{Ri} and Θ_i are the command input and the actual output at time index i respectively.

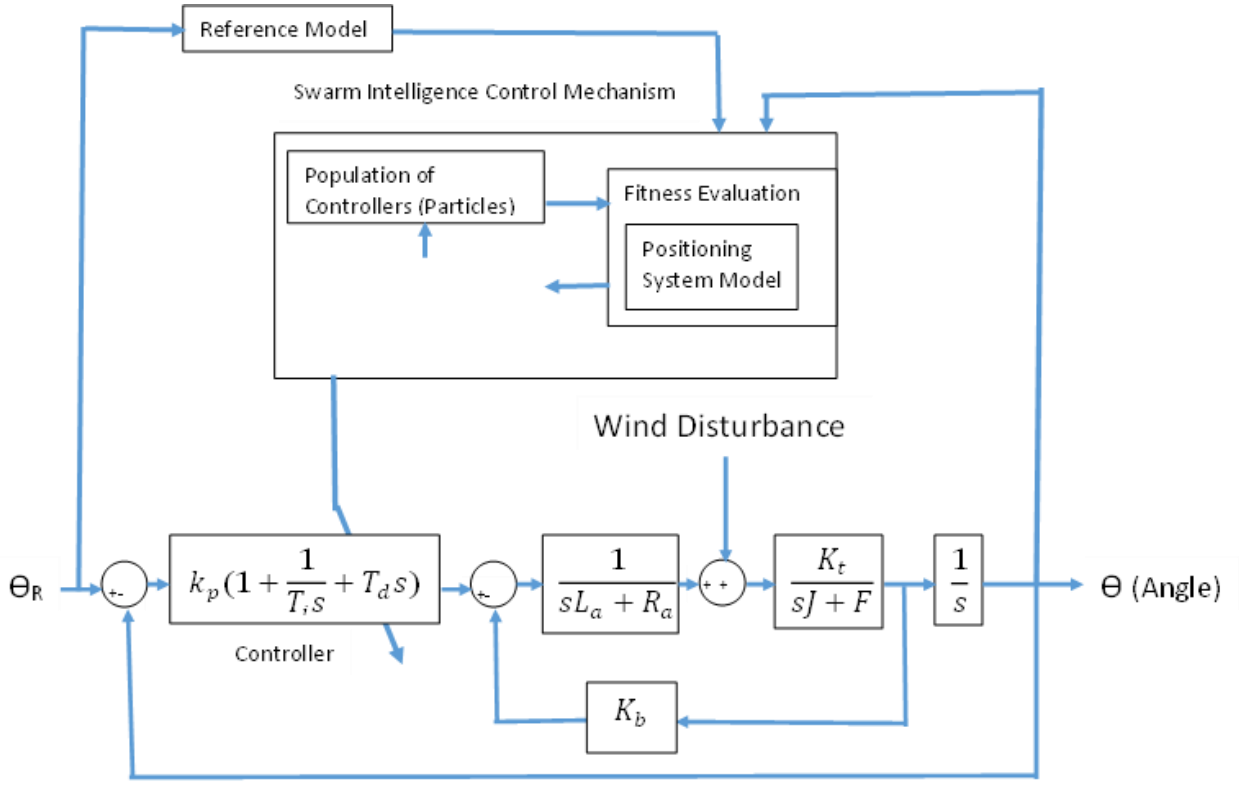

Fig. 1: Block diagram of the control positioning system

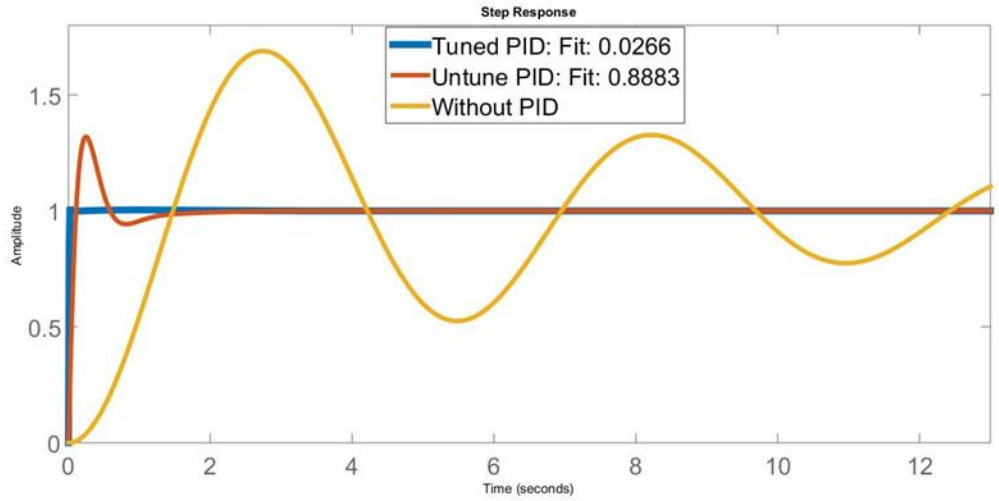

Fig. 2: Unit step response using: tuned PID, untune PID and without PID

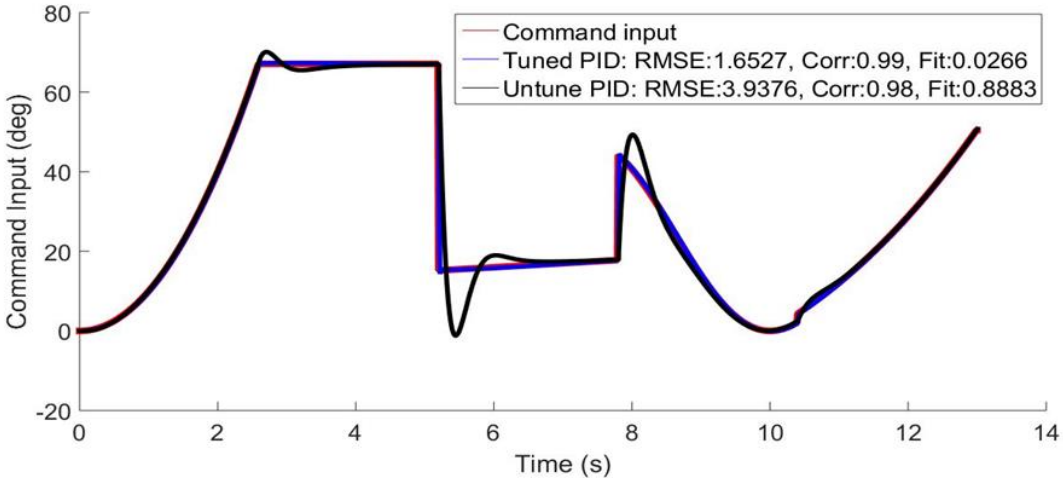

Fig 3: Real world command input using tuned and untune PID CLPSO Toroidal

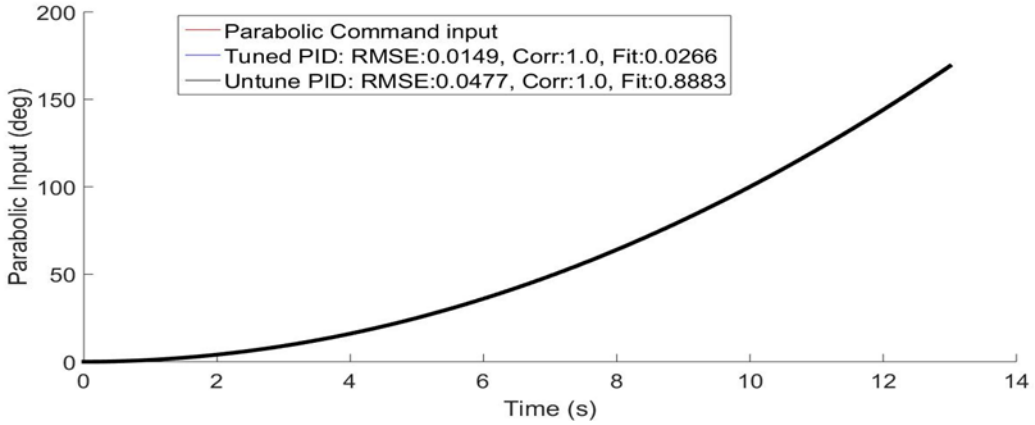

Fig. 4: Parabolic input command

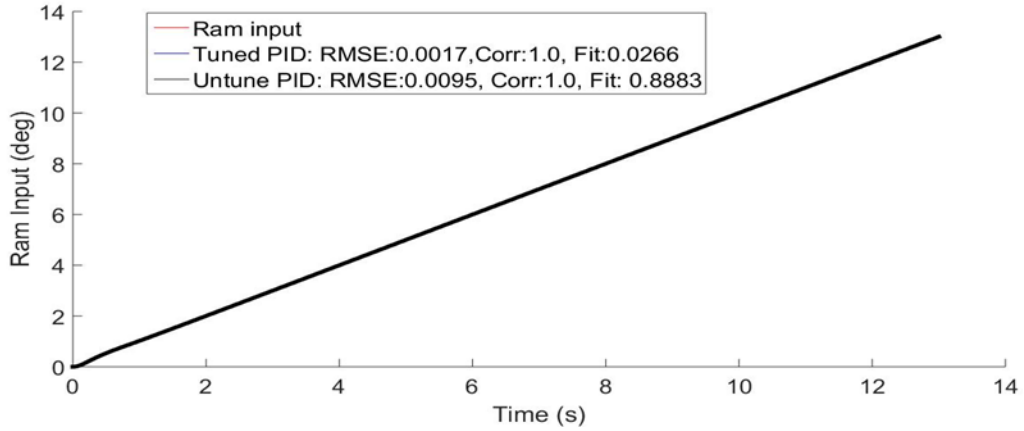

Fig. 5: Ram input command, CLPSO Toroidal

Table 1: Performance of the PSO variants implemented

Algorithms	RMSE for Three Inputs			Max Overshot	Rise Time	Settling Time	Fitness
	Real world	Ram	Parabolic				
CLPSO Toroidal	1.6527	0.0017	0.0149	0.0176	0.0100	0.010	0.0134
CLPSO	1.7085	0.0020	0.0175	0.0055	0.0100	0.0200	0.0185
PSO Toroidal	1.6747	0.0017	0.0145	0.0046	0.0100	0.0100	0.0136

CONCLUSION

Swarm intelligence algorithms variants proved to be an efficient optimizer for tuning the PID controller with CLPSO algorithm using toroidal bound for constraint optimization emerging as the best for addressing this particular control problem with RMSE of 1.6527, followed by toroidal bound inertia PSO with RMSE 1.6747. Comparing the performance of the PSO variants with those of differential evolution (DE) algorithms variants presented in [7] for solving similar problem, the PSO outperformed the DE for this particular problem

REFERENCES

[1] A.R. Laware1, V.S. Banda and D.B. Talange. Real Time Temperature Control System Using PID Controller and Supervisory Control and Data Acquisition System (SCADA), *International Journal of Application or Innovation in Engineering & Management, Volume 2, Issue 2, 2013*

[2] I J Narath and M. Gopal: Control system Engineering, fourth Edition, 2006

[3] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar. Comprehensive learning particleswarm optimizer for global optimization of multimodal functions. IEEE Transactionson Evolutionary Computation, 10(3):281–295, 2006.

[4] Katsuhiko Ogata Modern Control Engineering *Fifth Edition* Upper Saddle River 3010

[5] K Smriti Rao, Ravi Mishra Comparative study of P, PI and PID controller for speed control of VSI-fed induction motor, International Journal of Engineering Development and Research, Volume 2, Issue 2, 2014

[6] S. Iliya. Application of Computational Intelligence in Cognitive Radio Network for Efficient Spectrum Utilization, and Speech Therapy. PhD Thesis, 2017.

[7] S. Iliya. Differential Evolution Based PID Antenna Position Control System. *International Journal of Scientific and Engineering Research, July 2017.*

[8] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In Proceedings of theIEEE Congress on Evolutionary Computation, 1998.