IJE TRANSACTIONS C: Aspects Vol. 37 No. 09, (September 2024) 1716-1735

International Journal of Engineering

Journal Homepage: www.ije.ir

A New Optimization Algorithm Based on Particle Swarm Optimization Genetic Algorithm and Sliding Surfaces

M. J. Mahmoodabadi*[a], A. R. Nemati[a], N. Danesh[b]

[a] *Department of Mechanical Engineering, Sirjan University of Technology, Sirjan, Iran*
[b] *Department of Mechanical and Aerospace Engineering, University of Texas at Arlington, Arlington, USA*

PAPER INFO

Paper history:
Received 06 September 2023
Received in revised form 04 February 2024
Accepted 10 February 2024

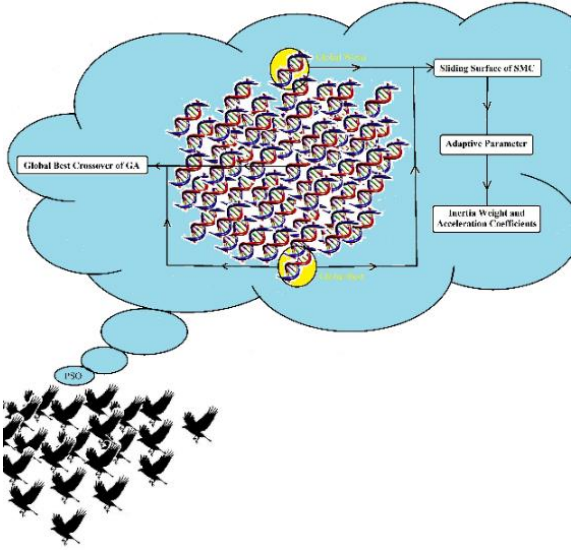
Keywords:
Particle Swarm Optimization
Genetic Algorithm
Sliding Surface
Optimal Control
Nonlinear Quadcopter System
Inelastic Compression Member

ABSTRACT

To enhance the performance of meta-heuristic algorithms, the development of new operators and the efficient combination of various optimization techniques are valuable strategies for discovering global optimal solutions. In this research endeavor, we introduce a novel optimization algorithm called PGS (Particle Swarm Optimization-GA-Sliding Surface). PGS combines the strengths of particle swarm optimization (PSO), genetic algorithm (GA), and sliding surface (SS) to tackle both mathematical test functions and real-world optimization problems. To achieve this, we adaptively tune the weighting function and learning coefficients of the PSO algorithm using the sliding mode control's SS relation. The global best particle discovered through the PSO method serves as one of the parents in the GA's crossover operation. This new crossover operator is then probabilistically integrated with an improved particle swarm optimization algorithm, enhancing convergence speed and facilitating escape from local optima. We evaluate the proposed algorithm's performance on both uni-modal and multi-modal mathematical test functions, considering un-rotated and rotated cases, thereby testing its effectiveness and efficiency against other prominent optimization techniques. Furthermore, we successfully implement the PGS algorithm in optimizing the state feedback controller for a nonlinear quadcopter system and determining the cross-section for an inelastic compression member.

doi: *10.5829/ije.2024.37.09c.02*

Graphical Abstract


Global Worst
Sliding Surface of SMC
Adaptive Parameter
Global Best Crossover of GA
Inertia Weight and Acceleration Coefficients
Global Best
PSO

*Corresponding Author Email: *mahmoodabadi@sirjantech.ac.ir* (M. J. Mahmoodabadi)

Please cite this article as: Mahmoodabadi MJ, Nemati AR, Danesh N. A New Optimization Algorithm Based on Particle Swarm Optimization Genetic Algorithm and Sliding Surfaces. International Journal of Engineering, Transactions C: Aspects. 2024;37(09):1716-35.

NOMENCLATURE			
$\vec{X}_{\mathrm{i}}$	Randomly selected particle	A_g	Cross-section area
$\vec{X}_i'$	Off-spring	b_f	Flange width
$\vec{X}_{\mathrm{gbest}}$	Global best particle	t_f	Flange thickness
F	Fitness function	d	Section depth
$\vec{X}_{gworst}$	Global worst particle	t_w	Web thickness
$\max t$	Maximum number of iterations	K_y	Effective length factor
E	Error signal	L_y	Laterally unsupported length
$\dot{E}$	Derivative of error signal	F_y	Specified minimum yield stress
N	Population size	r_y	Gyration radius
b	Adaptive parameter	F_{ey}	Euler stress
Tr	Trace of matrix	I_y	Moment of inertia
$\mathrm{J}_{\frac{1}{2}}$	Bessel function type I	**Greek Symbols**	
erf	Error function	Λ	Diagonal elements of the slope matrix
φ	Golden ratio	τ	Lattice relaxation time
ɯ	Inertia weight	ζ	Maximum number of iterations
ɯ$_{min}$	Minimum inertia weight	Ω_i	Angular velocities
ɯ$_{max}$	Maximum inertia weight	$\vec{M}$	Vector of external torques
C_1, C_2	Acceleration coefficients	ω_x, ω_y, ω_z	Angular velocities of the quadcopter around
psd	Popular Standard Deviation	T_i	Motor thrust torques
D	Dimension size	τ_i	Blade torques
c	Thrust parameter	γ	Density of steel
d	Drag parameter	λ_y	Slenderness ratio
K	Design vector	λ_e	Limiting value of slenderness ratio

1. INTRODUCTION

In the recent decades, there is a growing interest in the design of optimization algorithms and their applications in the fields of sciences and technologies (1-4). In this way, many optimization techniques have been presented to solve various real-world problems by researchers in the recent years (5-7). Among them, evolutionary algorithms have attracted more considerations for linear/non-linear, convex/non-convex and constrained/non-constrained problems (8-10). In which, the Genetic Algorithm (GA) (11) and Particle Swarm Optimization (PSO) (12) as the most prominent and well-known approaches have been broadly implemented in the scientific research studies (13-15).

The genetic algorithm as a metaheuristic method was initially inspired by the process of natural selection based upon the concept of Darwin's evolution theory which was proposed by Holland (11). The success of this algorithm has been demonstrated by numerous studies in the literature. For instance, Jozaghi et al. (16) suggested optimum design of alumina-forming austenitic stainless-steel composition through thermodynamic stability and oxidation model considerations based on the genetic algorithm to have the minimum number of alloying additions. Rath et al. (17) designed a hybrid navigational controller by the genetic algorithm and neural networks for path planning of a humanoid robot in given cluttered environments. At first, a genetic algorithm controller was used to generate an initial turning angle for a robot, and then it was hybridized with a neural network controller to create the final turning angle (17). In another study, optimum design of a four-layer laminated composite plate with quasi-square cut-out was presented for maximization of the thermal buckling load by Mahdavi et al. (18). They used the genetic optimization algorithm to achieve optimum variables such as cut-out orientation, fiber angle, bluntness of cut-out corners, cut-out size, plate size ratio and stacking sequence. Further, Ehsani and Rezaeepazhand (19) studied stacking sequence optimization of laminated composite grid plates for maximization of the axial buckling load by employing the genetic algorithm. In this regard, the calculated elastic buckling load was considered as a fitness function using the first-order shear deformation and classical laminated plate theories while the pattern gird and orientation of the layer were taken as the design variables. Le-Manh and Lee (20) proposed optimal fiber orientations of imperfect laminated composite plates based on the genetic algorithm for maximization of the strength by investigating the bending, buckling and post-buckling. Moreover, the optimum design of the fiber orientation angle and number of layers was investigated for a laminated composite submerged cylindrical pressure hull through the genetic algorithm by Imran et al. (21). Wei et al. (22) developed the genetic algorithm coupled with the finite element method for optimally design of symmetrical laminated composite cylindrical shells under hydrostatic pressures. They solved a stacking sequence optimization problem for searching the best laminations with the maximum buckling pressure. Moreover, the genetic algorithm was employed to find the ideal proportional-integral-derivative gains for speed control of a brushless DC motor by Ibrahim et al. (23). Furthermore, Sun et al. (24) suggested an optimum automatic design method of convolutional neural networks based on the genetic algorithm to impressively address the image classification tasks.

Besides, the particle swarm optimization as another population-based algorithm was originally intended to simulate the behaviors of bird flocks and fish schools by Kennedy and Russell (12). After that, this scheme has been widely utilized in different research works to name but, Zamani et al. (25) used it to design a fractional-order proportional-integral-derivative controller considering five parameters and a cost function. Ezzeddine (26) suggested an improved scheme for reactive power analysis and frequency control of a self-excited wind induction generator by employing the particle swarm optimization algorithm. Hamed et al. (27) optimized critical nonuniform axial compressive and shear loads of sandwich functionally graded beams with the porous core using a particle swarm optimization scheme. They applied the parabolic higher-order shear deformation theory and generalized variational principle to determine the kinematic displacement field and equilibrium governing equations, respectively. Furthermore, an effective hybrid optimization procedure based on an improved particle swarm optimization algorithm was proposed by Keshtegar et al. (28) to maximize the buckling load of laminated composite plates subjected to uniaxial and biaxial compressive loads. Further, for buckling maximization design of composite panels, Huang et al. (29) applied a multimodal particle swarm optimization approach. The solutions of structural optimization problems with complex design spaces via directly performing the feasibility of constraints were found utilizing the particle swarm optimization method by Jansen and Perez (30). In another study, Nguyen et al. (31) ameliorated the performance of an adaptive neuro-fuzzy inference system for calculating and optimizing the buckling capacity of circular opening steel beams. They investigated the most effective parameters of the particle swarm optimization procedure, namely the population size, the number of iterations, acceleration coefficients, initial weight and velocity limits, on the results. To achieve more reasonable pricing for designing cold-formed steels, Ye et al. (32) used the particle swarm optimization scheme by considering a crucial factor as finding a solution in the constrains of the fabrication and construction industries. Besides, to forecast the freezing possibility of wind turbine blades, Xu et al. (33) considered a particle swarm optimization scheme in the kernel function for providing the optimal parameters of a support vector machine.

As it was mentioned, falling in local optima and low convergence speed are two main issues related to the evolutionary algorithms that can be resolved by combining two or more methods of them (34). In this regard, the following hybridizations could be noted: firefly algorithm, genetic algorithm, ant colony optimization algorithm (35), particle swarm optimization and cultural algorithm (36), particle swarm optimization and ant colony (37), genetic algorithm and particle swarm optimization (38), competitive swarm optimization and imperialist competitive algorithm (39), grey wolf optimization algorithm, particle swarm optimization (40), particle swarm optimization and simulated annealing method (41), particle swarm optimization and whale optimization algorithm (42) whale optimization approach and genetic algorithm (43), genetic algorithm and teaching-learning-based optimization (44), etc.

As the contribution of this paper, a new combination of the particle swarm optimization with the genetic algorithm as well as the sliding surfaces is suggested. In fact, the adjustable parameters of the PSO algorithm are tuned by the SS concepts, while the crossover operator of the GA is improved by employing the best solution of the population. Hybridization of these strategies, introduced as the PGS algorithm, is utilized to find the optimum solutions of the mathematical benchmark functions. Finally, the performance of the PGS optimization approach is challenged to solve two practical problems, i.e., tuning the control gains of a nonlinear 4DOF quadcopter system and finding the cross-section sizes of an inelastic compression member.

The rest of this paper is organized as follows. Section 2 represents the preliminary tools to design the considered methodology including the genetic algorithm, particle swarm optimization and sliding surface relations. The proposed PGS optimization algorithm is discussed in section 3 in details. Analysis of the algorithm behavior on the benchmark functions is presented in section 4. The performance of the PGS on the real-world design problems is challenged in sections 5 and 6. Finally, section 7 briefly concludes the paper.

2. PRELIMINARY TOOLS

Recently, genetic algorithms and particle swarm optimization technique have attracted attentions of researches in the science and engineering fields. The GA has been accepted due to its intuitiveness and ease of implementation, while the mechanism of the PSO is inspired by the swarming behavior of biological systems. Because the two techniques are presumed to solve same problems but utilize different polices, it would be appealing to compose their formulations.

2. 1. Genetic Algorithm

The genetic algorithm, as a population-based optimization method, mainly utilizes three operators i.e., reproduction, mutation, and crossover to enhance the situations of the members (45). The reproduction operator employs the best members to produce the new child. The mutation operator suddenly changes the situations of some randomly selected parents to escape from local minima. Finally, the crossover operator picks up the parents from the mating pool and combines them to present the new off-springs. The traditional crossover operator is commonly formulized as follows (45).

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \vec{\kappa}_1(\vec{X}_i(t) - \vec{X}_j(t))$$
$$\vec{X}_j(t+1) = \vec{X}_j(t) + \vec{\kappa}_2(\vec{X}_i(t) - \vec{X}_j(t)) \qquad (1)$$

where, t is the iteration number. $\vec{X}_i$ and $\vec{X}_j$ represent selected members for changing their positions. $\vec{\kappa}_1$ and $\vec{\kappa}_2$ denote vectors having random numbers belong to interval [0,1].

2. 2. Particle Swarm Optimization

The particle swarm optimization modifies position $\vec{X}_\mathrm{i}$ by applying its velocity $\vec{V}_\mathrm{i}$ via the following relations (46).

$$\vec{V}_i(\mathrm{t}+1) = \omega\vec{V}_i(\mathrm{t}) + \mathrm{C}_1\vec{r}_1(\vec{X}_{pbest_i} - \vec{X}_i(\mathrm{t})) + \mathrm{C}_2\vec{r}_2(\vec{X}_{gbest} - \vec{X}_i(\mathrm{t})) \qquad (2)$$
$$\vec{X}_i(\mathrm{t}+1) = \vec{X}_i(\mathrm{t}) + \vec{V}_i(\mathrm{t}+1)$$

where, ω represents the inertia weight parameter that identifies the effect of the previous velocity of the particle on its current velocity. C_1 and C_2 denote the acceleration coefficients. $\vec{r}_1$ and $\vec{r}_2$ are vectors having random numbers belong to [0,1]. $\vec{X}_{gbest}$ is the best solution in the whole swarm. Finally, $\vec{X}_{pbest_i}$ presents the best position found by particle i (46).

2. 3. Sliding Surface Matrix

The sliding mode control approach is a robust and effective methodology to handle uncertainties and disturbances in the nonlinear systems (47-50). Consider a nonlinear dynamical system having the following state-space equations.

$$\dot{X} = f\,(X, U, t) \qquad (3)$$

where, $X \in \mathrm{R}^\mathrm{n}$ is the state vector, $U \in \mathrm{R}^\mathrm{m}$ denotes the control effort vector, and t shows time.

The sliding surface matrix can be written as follows.

$$\mathrm{S}(E,\mathrm{t}) = \left(\frac{d}{dt}I_n + \Lambda\right)^{(n-1)} E = 0 \qquad (4)$$

where, I_n denotes the identity matrix, and sloppe matrix Λ represents a diagonal positive definite one illustrated as follows:

$$\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n), \quad \lambda_i > 0, \; i = 1,2,3,\ldots,n \qquad (5)$$

For a two dimensional ($n = 2$) nonlinear system, the sliding surface is re-written as follows.

$$S = \dot{E} + \Lambda E \qquad (6)$$

As it is clear from this equation, each element of the sliding surface matrix is the weighting summation of the error and its drivative.

3. PROPOSED HYBRID ALGORITHM

In this section, the novel hybrid optimization algorithm, namely PGS, as the combination of the PSO, GA and SS is described. In fact, the particle swarm optimization is considered as the basic approach, and a crossover operator as well as a sliding surface would be implemented to enhance its performance.

In this way, the regarded crossover operator utilizes a randomly selected particle ($\vec{X}_\mathrm{i}(\mathrm{t})$) as well as the global best particle ($\vec{X}_\mathrm{gbest}(t)$) as two parents to produce an off-spring $\vec{X}_i'(\mathrm{t})$ according to the following equation.

$$\vec{X}_i'(t) = (1 + \vec{\mathrm{v}}_1)\vec{X}_{gbest} - \vec{\mathrm{v}}_2\vec{X}_i(\mathrm{t}) \qquad (7)$$

where, $\vec{\mathrm{v}}_1$ and $\vec{\mathrm{v}}_2$ are two vectors having random numbers belong to interval [0,1]. The superior position between $\vec{X}_i(\mathrm{t})$ and $\vec{X}_i'(t)$ should be transferred to the new population as $\vec{X}_\mathrm{i}(\mathrm{t}+1)$.

Moreover, the defined concept for the sliding surface is implemented to introduce new formulae for the inertia weight and acceleration coefficients. To this end, the error signal and its derivative are calculated via the following relations.

$$E = mean\left|F\left(\vec{X}_i(\mathrm{t})\right)\right|$$
$$\dot{E} = mean\left|\frac{\left|F\left(\vec{X}_i(\mathrm{t})\right)\right| - \left|F\left(\vec{X}_i(\mathrm{t}-1)\right)\right|}{t}\right|, \qquad i = 1,2,3,\ldots,N \qquad (8)$$

where, N is the population size, and F represents the fitness function. Moreover, the diagonal elements of the slope matrix are defined as follows.

$$\lambda = \frac{t}{\max t}\left(\left|\vec{X}_{gbest} - \vec{X}_{gworst}\right|\right) \qquad (9)$$

where, max t denotes the maximum number of iterations, and $\vec{X}_{gworst}$ represents the worst solution in the whole swarm. Now, the following strategy is introduced for calculation of adaptive parameter b to find the inertia weight and the acceleration coefficients.

(a) If $Tr(S) < 1$, then the adaptive parameter is defined as follows.

$$b = -\mathrm{J}_{\frac{1}{2}}(Tr(S)) + 1 \tag{10}$$

where, $Tr(S)$ signifies trace of matrix S, and $\mathrm{J}_{\frac{1}{2}}$ represents Bessel function type I with order 0.5 (51).

(b) If condition (a) is not satisfied and $Tr(\mathrm{S}^{-1}) < 1$, then the adaptive parameter is calculated as follows.

$$b = erf(Tr(\mathrm{S}^{-1})) \tag{11}$$

where, erf is the error function.

(c) If conditions (a) and (b) are not satisfied, then the adaptive parameter would be computed via the following equation.

$$b = erf(tan^{-1}(Tr(\mathrm{S}^{-1})) - \varphi + 1) \tag{12}$$

(d) where, φ represents the golden ratio.

(e) When the adaptive parameter is identified, the inertia weight and the acceleration coefficients would be determined by employing the following relations.

$$\begin{aligned} \omega &= \omega_{max} - (\omega_{max} - \omega_{min})b \\ C_1 &= C_{1i} - (C_{1i} - C_{1f})b \\ C_2 &= C_{2\mathrm{i}} - (C_{2\mathrm{i}} - C_{2\mathrm{f}})b \end{aligned} \tag{13}$$

where, ω_{min} and ω_{max} respectively illustrate the minimum and maximum values for the inertia weight, C_{1i} and C_{2i} correspondingly denote the initial values for the first and second acceleration coefficients, while C_{1f} and C_{2f} represent their final values. Figure 1 illustrates the flowchart for computation of the adaptive parameter, inertia weight and acceleration coefficients. Besides, Figure 2 depicts the proposed hybrid optimization algorithm having the modified crossover operator and adaptive parameters.

4. RESULTS ON MATHEMATICAL TEST FUNCTIONS

At first, the searching manners of the PGS algorithm are investigated on Sphere and Griewank test functions. Figures 3(a), 3(b), 4(a) and 4(b) respectively depict changings of the adaptive parameters over iterations for the two test functions. As it is clear from these figures, the adaptive parameter starts from zero and reaches to 0.85 at about iteration 70 and converges to 1 duuring the following iterations. Figures 3(c), 3(d), 4(c) and 4(d) represent changings the inertia weight over iterations for these test functions. It can be seen from these figures that the inertia weight starts from ω_{max} and converges to

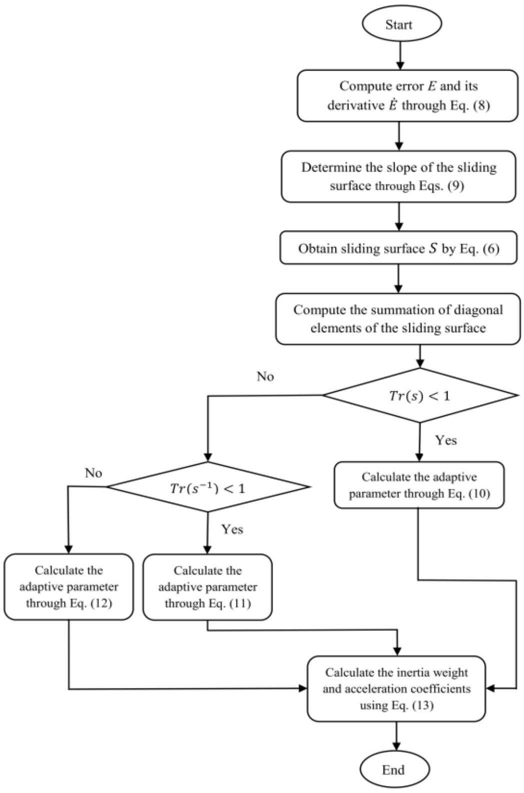

Start
Compute error E and its derivative $\dot{E}$ through Eq. (8)
Determine the slope of the sliding surface through Eqs. (9)
Obtain sliding surface S by Eq. (6)
Compute the summation of diagonal elements of the sliding surface
$Tr(s) < 1$
No
Yes
Calculate the adaptive parameter through Eq. (10)
$Tr(s^{-1}) < 1$
No
Yes
Calculate the adaptive parameter through Eq. (12)
Calculate the adaptive parameter through Eq. (11)
Calculate the inertia weight and acceleration coefficients using Eq. (13)
End

Figure 1. Flowchart for calculation of the parameters related to the PGS algorithm

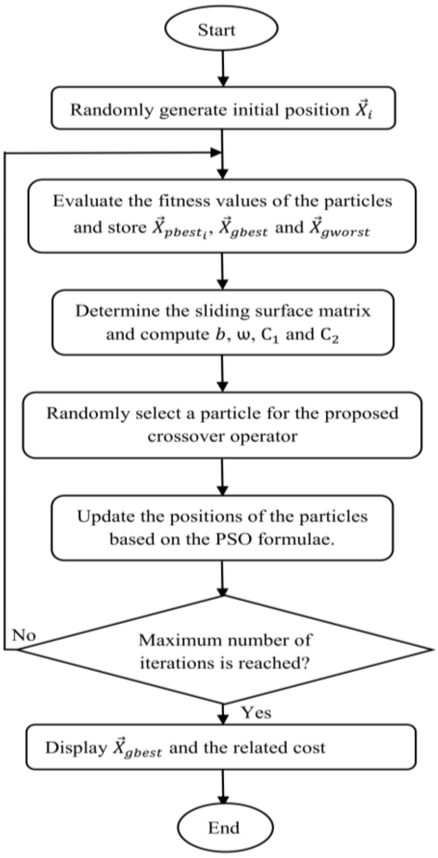
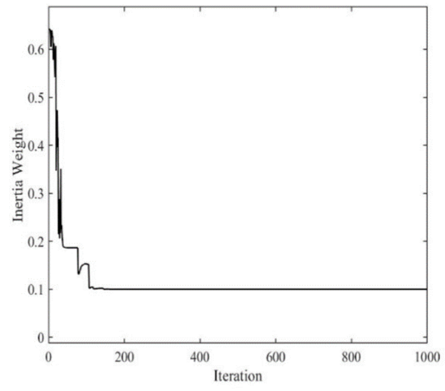
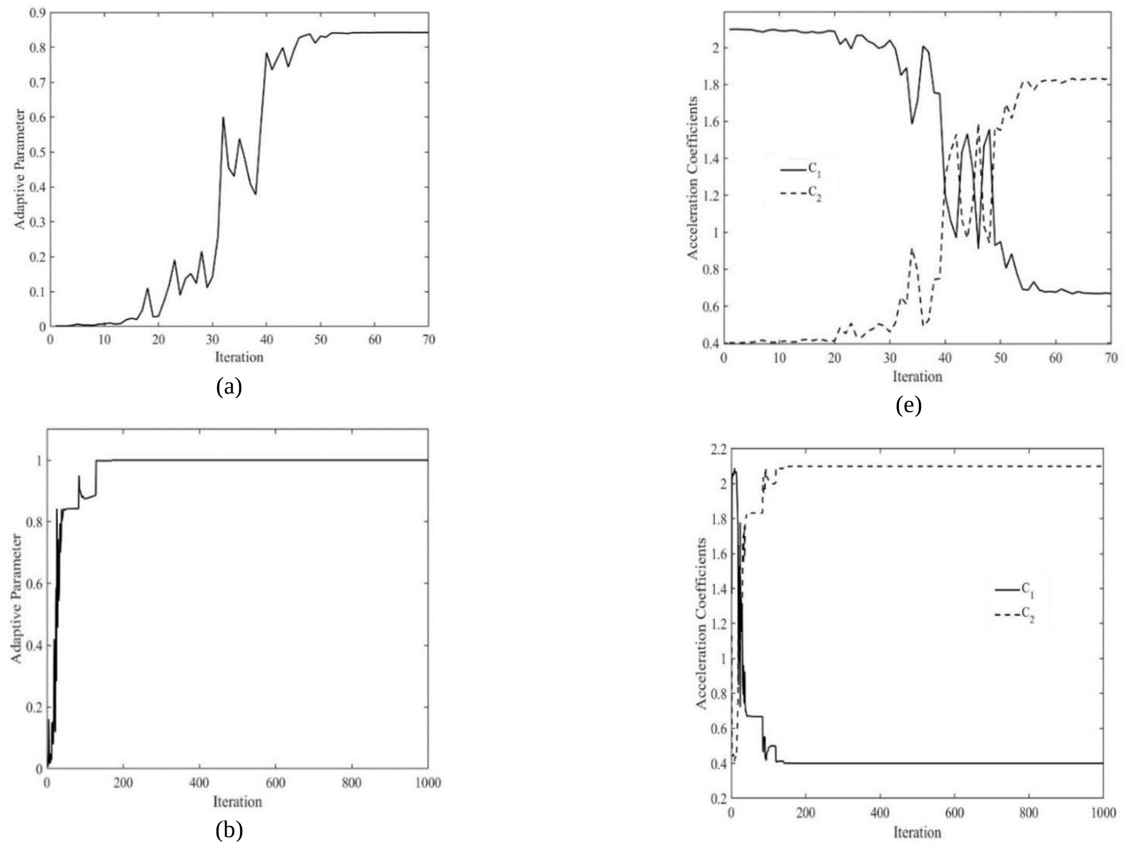

Start
Randomly generate initial position $\vec{X}_i$
Evaluate the fitness values of the particles and store $\vec{X}_{pbest_i}$, $\vec{X}_{gbest}$ and $\vec{X}_{gworst}$
Determine the sliding surface matrix and compute b, ω, C_1 and C_2
Randomly select a particle for the proposed crossover operator
Update the positions of the particles based on the PSO formulae.
Maximum number of iterations is reached?
No
Yes
Display $\vec{X}_{gbest}$ and the related cost
End

Figure 2. Flowchart of the PGS algorithm

ω_{min} respectively set at 0.65 and 0.1. These values for the inertia weight guide the algorithm from a global searching to a local one. Moreover, Figures 3(e), 3(f), 4(e) and 4(f) display the manner of the acceleration coefficients for the considered test functions based on the introduced relations. These diagrams show that the value of C_1 is decreasing while the value of C_2 is increasing versus the iteration. This means that the effect of C_1 is more than that of C_2 at the beginning of the search process. This helps to have the local searching at the first iterations, and the particles utilize the information found by $\vec{X}_{pbest_i}$ instead of that for $\vec{X}_{gbest}$. However, when the process is continued, $\vec{X}_{gbest}$ reaches to the positions near to the global optimum point and could guide the swarm to this situation. Hence, the value of C_2 is more than that for C_1 at the final iterations. It is noticeable that



(d)

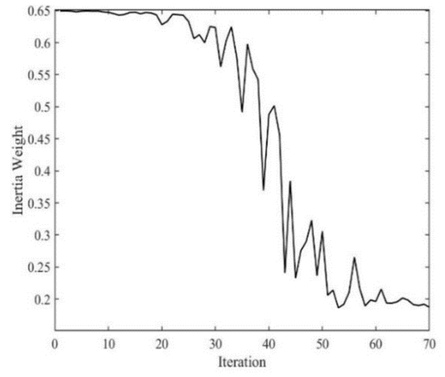


(a)

(e)

(b)

(f)



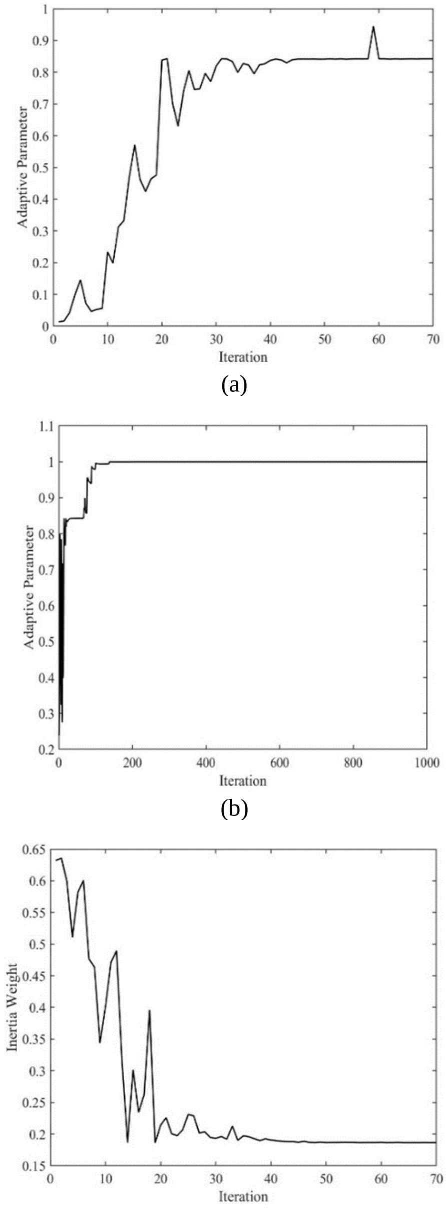
(c)

Figure 3. Searching behavior of the PGS algorithm on Sphere function. Mean value of (a) adaptive parameter during 70 iterations, (b) adaptive parameter during 1000 iterations, (c) inertia weight during 70 iterations, (d) inertia weight during 1000 iterations, (e) acceleration coefficients during 70 iterations and (f) acceleration coefficients during 1000 iterations

acceleration coefficient C_1 is varied between C_{1i} and C_{1f}, respectively regarded as 2.1 and 0.4; while acceleration coefficient C_2 is rising from C_{2i} to C_{2f} correspondingly considered as 0.4 and 2.1.

Furthermore, one of the most important abilities of an optimization algorithm is preservation of the population diversity to avoid falling into the local optimum trap. This capability could be calculated using the Popular Standard Deviation (psd) criteria mentioned as the following relation (52).

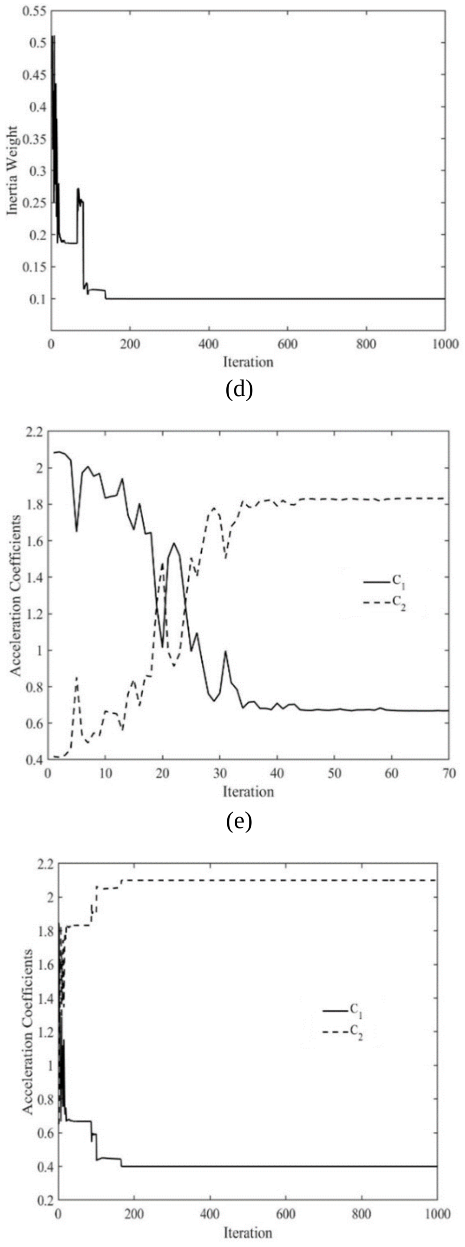
$$psd = \sqrt{\left[\sum_{i=1}^{N}\sum_{j=1}^{D}\left(x_i^j - \bar{x}^j\right)^2\right]/(N-1)} \qquad (14)$$



(a)

(b)

(c)



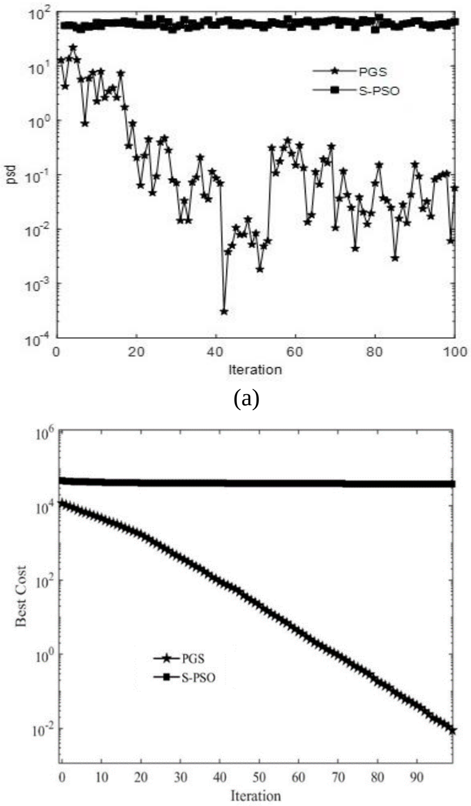
(d)

(e)

(f)

Figure 4. Searching behavior of the PGS algorithm on Griewank function. Mean value of (a) adaptive parameter during 70 iterations, (b) adaptive parameter during 1000 iterations, (c) inertia weight during 70 iterations, (d) inertia weight during 1000 iterations, (e) acceleration coefficients during 70 iterations and (f) acceleration coefficients during 1000 iterations

where, N, D and $\bar{x}$ respectively denote the population size, dimension number and mean position of the whole swarm. Figure 5(a) illustrates the mean values of the psd parameter found for the Griewank function by the introduced method and compares it with that of the Standard PSO (S-PSO) algorithm. It is observable that



(a)

(b)

Figure 5. Performance of the PGS algorithm on Griewank function and its comparison with the standard PSO during 100 iterations, (a) popular standard deviation (b) best cost

the PGS algorithm is able to hold the psd parameter at a higher level in comparison with the S-PSO approach. It is noticeable that the high values for the psd may be due to the weakness of an algorithm to converge to the global optimum solution. Hence, it is suggested that the fitness value of the objective function is also investigated with the psd parameter. Therefore, Figure 5(b) is presented to depict the best cost values of the Griewank test function found by the PGS and S-PSO algorithms. When both diagrams 5(a) and 5(b) are carefully examined, the effectiveness of the introduced strategy in preserving population diversity, escaping fromT local optimum points, and reaching the global minimum solution is clearly evident.

Moreover, in order to challenge the performance of the PGS algorithm on the solution accuracy and convergence speed, nine well-known un-rotated test functions are utilized. Table 1 lists the formulations of these functions as well as their searching domains, threshold values and global optimum solutions. The obtained results by the introduced scheme are compared with those of twelve prominent optimization algorithms, i.e. Genetic Algorithm with Traditional Crossover (GA-TC) (53), Genetic Algorithm with Multiple Crossover (GA-MC) (53), Standard Particle Swarm Optimization (S-PSO), Fuzzy combination of Genetic Algorithm and Particle Swarm Optimization (F-GA&PSO) (54), Modified Particle Swarm Optimizer (MPSO) (55), Local-version Particle Swarm Optimization (LPSO) (56), Von Neumann Particle Swarm Optimization (VPSO) (56), Fully Informed Particle Swarm (FIPS) algorithm (57), self-organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients (HPSO-TVAC) (58), Dynamic Multi-Swarm Particle Swarm Optimizer (DMS-PSO) (59), Comprehensive Learning Particle Swarm Optimizer (CLPSO) (60) and Adaptive Particle Swarm Optimization (APSO) (61). For this part of the simulations, the population size is set at $N = 20$, the maximum number of iterations is regarded as $\zeta = 10000$, and the dimension size is considered as $D = 30$. The mean values of thirty independent runs are reported as the final optimum solutions. The settings of the comparative algorithms are set at according to heir literature ((53) throgh (61)) and represented in Table 2. The results for the nine un-rotated test functions obtained by employing the proposed PGS algorithm and the other twelve optimization approaches are reported in Tables 3 and 4. In fact, the mean values of thirty independent runs

TABLE 1. Un-rotated test functions for challenging the optimization algorithms

Name (Comment)	Formulation	Search space
Sphere (Uni-modal)	$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100,100]$
Schwefel's P2.2 (Uni-modal)	$f_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10,10]$
Schwefel's P1.2 (Uni-modal)	$f_3(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100,100]$
Rosenbrock (Uni-modal)	$f_4(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-10,10]$
Step (Uni-modal)	$f_5(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100,100]$
Quadric Noise (Uni-modal)	$f_6(x) = \sum_{i=1}^{D} i x_i^4 + \text{random}[0,1]$	$[-1.28,1.28]$
Ackley (Multi -modal)	$f_7(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right) + 20 - \exp\left(\frac{1}{D} \sum_{i=1}^{D} cos\,(2\pi x_i) \right) + e$	$[-32,32]$
Griewank (Multi -modal)	$f_8(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600,600]$
Generalized Penalized (Multi -modal)	$f_9(x) = \frac{\pi}{D} \{10 sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 \, [1 + 10 sin^2(\pi y_{i+1})]\}$	$[-50,50]$

TABLE 2. Settings of the algorithms applied for the performance comparison

Algorithm	Parameters
GA-TC	$P_r = 0.2, P_{tc} = 0.4, P_m = 0.1, s = 0.05$, tournament method for selection
GA-MC	$P_r = 0.2, P_{mc} = 0.4, P_m = 0.1, s = 0.05$, tournament method for selection
S-PSO	$\omega = 0.9, C_1 = C_2 = 2.0$
F-GA&PSO	$\omega_1 = 0.9, \omega_2 = 0.4, C_{1i} = C_{2f} = 2.5, C_{1f} = C_{2i} = 0.5, \zeta_m = 0.001, \xi_{tc} = \xi_{mc} = 0.2$
PSOM	$\omega: 0.9 - 0.4, C_1 = C_2 = 2.0$
LPSO	$\omega: 0.9 - 0.4, C_1 = C_2 = 2.0$
VPSO	$\omega: 0.9 - 0.4, C_1 = C_2 = 2.0$
FIPS	$x = 0.729, \sum C_i = 4.1$
HPSO-TVAS	$\omega: 0.9 - 0.4, C_1: 2.5 - 0.5, C_2: 0.5 - 2.5$
DMS-PSO	$\omega: 0.9 - 0.2, C_1 = C_2 = 2.0, m = 3, R = 5$
CLPSO	$\omega: 0.9 - 0.4, C = 1.49445, m = 7$
APSO	ω: adaptive relation, C_1 and C_2: fuzzy systems

TABLE 3. Mean values of the best solutions found in 30 independent runs by the PGS and other algorithms for un-rotated test functions

Algorithm	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
GA-TC	3.28×10^{-14}	1.83×10^{-6}	774	133	210	5.10×10^{-2}	15.7	7.16×10^{-1}	13.63
GA-MC	2.56×10^{-18}	8.16	314	82.5	34	8.64×10^{-2}	18	4.86×10^{-1}	3.59×10^{8}
S-PSO	2.25×10^{-98}	3.15×10^{-26}	6.73×10^{-5}	20.4	0.10	1.06	2.81×10^{-1}	2.15×10^{-2}	5.71×10^{7}
FGAPSO	1.58×10^{-119}	9.11×10^{-25}	2.14×10^{-11}	**5.34×10^{-1}**	**0**	3.38×10^{-3}	4.30×10^{-8}	2.03×10^{-2}	3.46×10^{-3}
PSOM	3.16×10^{-52}	2.04×10^{-29}	1.11×10^{-1}	26.93	**0**	8.29×10^{-3}	1.15×10^{-14}	2.79×10^{-3}	2.28×10^{-32}
LPSO	4.77×10^{-29}	2.03×10^{-20}	18.60	21.86	**0**	1.49×10^{-2}	1.85×10^{-14}	1.10×10^{-2}	2.18×10^{-30}
VPSO	5.11×10^{-38}	6.29×10^{-27}	1.44	37.65	**0**	1.08×10^{-2}	1.40×10^{-14}	1.31×10^{-2}	3.46×10^{-3}
FIPS	3.21×10^{-30}	1.32×10^{-17}	0.77	22.54	**0**	2.55×10^{-3}	7.69×10^{-15}	9.04×10^{-4}	1.22×10^{-31}
HPSO	3.38×10^{-41}	6.90×10^{-23}	2.89×10^{-7}	13	**0**	5.54×10^{-2}	2.06×10^{-10}	1.07×10^{-2}	7.07×10^{-30}
DMS-PSO	3.85×10^{-54}	2.61×10^{-29}	47.5	32.3	**0**	1.10×10^{-2}	8.52×10^{-15}	1.31×10^{-2}	**2.05**$\times10^{-32}$
CLPSO	1.89×10^{-19}	1.01×10^{-13}	395	11	**0**	3.92×10^{-3}	2.01×10^{-12}	6.45×10^{-13}	1.59×10^{-21}
APSO	1.45×10^{-150}	5.15×10^{-84}	1.00×10^{-10}	2.84	**0**	4.66×10^{-3}	1.11×10^{-14}	1.67×10^{-2}	3.76×10^{-31}
PGS	**0**	**0**	**1.35×10^{-24}**	17.54	**0**	**9.82×10^{-5}**	**7.67×10^{-15}**	**0**	2.91×10^{-31}

of the functions are represented in Table 3, while the related standard deviation values are demonstrated in Table 4. The bold values in these tables indicate the best found solutions. As it could be seen, the PGS algorithm presents the best results for all unimodal and multimodal mathematical functions except for f_4 and f_9.

Furthermore, in order to provide a more accurate assessment on the performance of PGS, six rotated test functions summarized in Table 5 are regarded to be minimized. In this way, statistical comparisons between the results founed by the PGS and seven famous evolutionary algorithms (S-PSO, SS-BLX (62), SS Arith (62), DE-Exp (63), DE-Bin (63), SaDE (64), HEPSO (65)) on these benchmarks are illustrated in Table 6. These functions are evaluated with dimension 10 with 10 particles, while all algorithms are independently run thirty times having the maximum number of evaluation 100000. The results related to these examinations are reported in Table 6 that illustrate the capability of the PGS to find the global optimum solution on the rotated test functions in comparison with the other optimization approaches.

In order to track the performance of the PGS algorithm, the diagrams in Figure 6 are plotted for dimensions 10, 20 and 30. These diagrams clearly illustrate that the considered scheme could easily scape from local minimum points and converges to the global optimum solutions. Besides, the convergence speeds of the F-GA&PSO, APSO and PGS algorithms are compared with each other on the previous introduced mathematical functions in Table 7. In this table, the mean values of the iterations required to reach the thresholds of the benchmarks are denoted for thirty independent runs. The results reported in this table obviously prove that the introduced optimization algorithm has a very high speed, and the proposed combination of the PSO with the GA and sliding mode concept is a valid method.

5. OPTIMUM CONTROL FOR A QUADCOPTER

Recently, quadcopters have received significant attention from researchers across various fields of science and technology. The quadcopters have four rotors which are directed upwards and located in a square formation with equal distances from the mass center of the quadcopter effective length. These flying robots are controlled through regulating the angular velocities of the rotors that are rotated by electrical motors. Heretofore, various control techniques such as optimal fuzzy controllers (66, 67), PID controllers (68, 69), backstepping approaches (70, 71), nonlinear H_∞ methods (72), linear quadratic

TABLE 4. Standared deviation values of the best solutions found in 30 independent runs by the PGS and other algorithms for un-rotated test functions

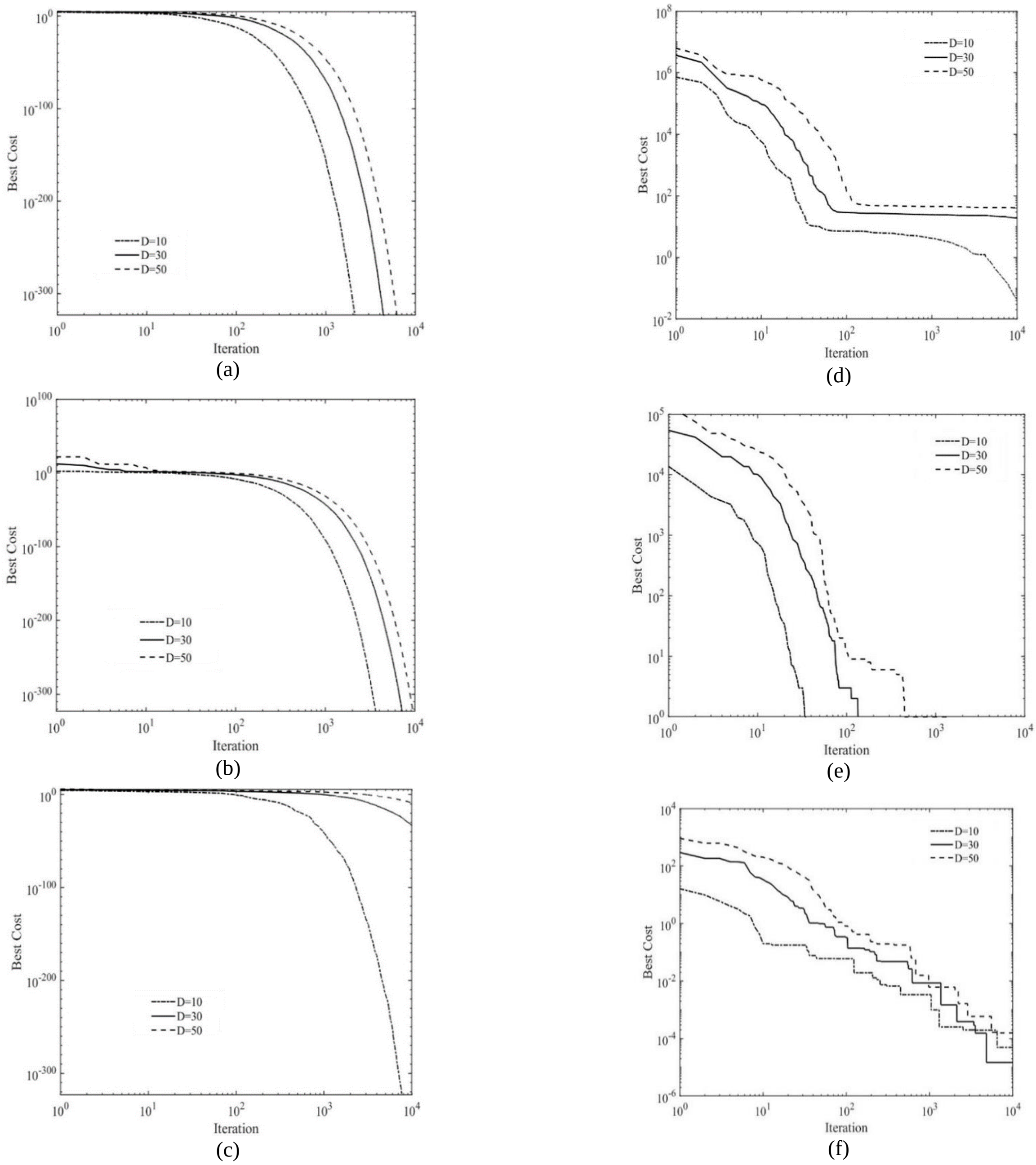
Algorithm	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
GA-TC	4.58×10^{-14}	4.32×10^{-6}	416	132	327	1.88×10^{-2}	1.13	2.87	8.91
GA-MC	1.34×10^{-17}	31.3	199	55.1	132	2.48×10^{-2}	6.23×10^{-1}	1.44	267.20
S-PSO	6.54×10^{-98}	1.73×10^{-25}	9.40×10^{-5}	25.3	0.305	3.12×10^{-1}	5.92×10^{-1}	2.31×10^{-2}	1.86×10^{7}
FGAPSO	8.58×10^{-119}	4.99×10^{-24}	6.91×10^{-11}	1.38	**0**	1.47×10^{-3}	9.90×10^{-8}	2.21×10^{-2}	1.89×10^{-2}
MPSO	6.11×10^{-52}	4.05×10^{-29}	1.27×10^{-1}	30.33	**0**	1.74×10^{-3}	3.55×10^{-15}	4.15×10^{-3}	1.64×10^{-32}
LPSO	1.13×10^{-28}	2.89×10^{-20}	30.71	11.15	**0**	5.66×10^{-3}	4.80×10^{-15}	1.60×10^{-2}	5.14×10^{-30}
VPSO	1.91×10^{-37}	8.68×10^{-27}	1.55	24.94	**0**	3.24×10^{-3}	3.48×10^{-15}	1.35×10^{-2}	1.89×10^{-2}
FIPS	3.60×10^{-30}	7.86×10^{-18}	0.86	**0.31**	**0**	6.25×10^{-4}	9.33×10^{-16}	2.78×10^{-3}	4.85×10^{-32}
HPSO	8.5×10^{-41}	6.89×10^{-23}	2.97×10^{-7}	16.5	**0**	2.08×10^{-2}	9.45×10^{-10}	1.14×10^{-2}	4.05×10^{-30}
DMS-PSO	1.75×10^{-53}	6.60×10^{-29}	56.4	24.1	**0**	3.94×10^{-3}	1.79×10^{-15}	1.73×10^{-2}	$\mathbf{8.12\times10^{-33}}$
CLPSO	1.49×10^{-19}	6.51×10^{-14}	142	14.5	**0**	1.14×10^{-3}	9.22×10^{-13}	2.07×10^{-12}	1.93×10^{-21}
APSO	5.73×10^{-150}	1.44×10^{-83}	2.13×10^{-10}	3.27	**0**	1.70×10^{-3}	3.55×10^{-15}	2.41×10^{-2}	1.20×10^{-30}
PGS	**0**	**0**	$\mathbf{4.33\times10^{-24}}$	0.94	**0**	$\mathbf{5.93\times10^{-5}}$	$\mathbf{9.01\times10^{-16}}$	**0**	1.68×10^{-30}

TABLE 5. Rotated test functions for challenging the optimization algorithms

Name (comment)	**Formulation**	**Optimum point**	**Search range**
Shifted Sphere (Uni-modal)	$f_{10}(x) = \sum_{i=1}^{D} z_i^2 + f\_bias$ $z = x - o, x = [x_1, x_2, \dots, x_D], o = [o_1, o_2, \dots, o_D]$	$x^* = o,$ $f_{10}(x^*) = f\_bias = -450$	$x \in [-100,100]^D$
Shifted Schwefel (Uni-modal)	$f_{11}(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} z_j\right)^2 + f\_bias$ $z = x - o, x = [x_1, x_2, \dots, x_D], o = [o_1, o_2, \dots, o_D]$	$x^* = o,$ $f_{11}(x^*) = f\_bias = -450$	$x \in [-100,100]^D$
Shifted Rosenbrock (Uni-modal)	$f_{12}(x) = \sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f\_bias$ $z = x - o, x = [x_1, x_2, \dots, x_D], o = [o_1, o_2, \dots, o_D]$	$x^* = o,$ $f_{12}(x^*) = f\_bias = 390$	$x \in [-100,100]^D$
Shifted Rastrigen (Multi-modal)	$f_{13}(x) = \sum_{i=1}^{D} (z_i^2 - 10\,cos(2\pi z_i) + 10) + f\_bias$ $z = x - o, x = [x_1, x_2, \dots, x_D], o = [o_1, o_2, \dots, o_D]$	$x^* = o,$ $f_{13}(x^*) = f\_bias = -330$	$x \in [-5,5]^D$
Shifted Rotated Ackley (Multi-modal)	$f_{14}(x) = -20\,exp(-0.2\sqrt{{}^{1}/_{D} \sum_{i=1}^{D} z_i^2})$ $- exp({}^{1}/_{D} \sum_{i=1}^{D} cos(2\pi z_i) + 20 + e + f\_bias$ $z = (x - o) \times M, x = [x_1, x_2, \dots, x_D], o = [o_1, o_2, \dots, o_D]$	$x^* = o,$ $f_{14}(x^*) = f\_bias = -140$	$x \in [-32,32]^D$
Shifted Rotated Griewank (Multi-modal)	$f_{15}(x) = \sum_{i=1}^{D} \frac{z_i^2}{4000} - - \prod_{i=1}^{D} cos(\frac{z_i}{\sqrt{i}}) + 1 + f\_bias$ $z = (x - o) \times M, x = [x_1, x_2, \dots, x_D], o = [o_1, o_2, \dots, o_D]$	$x^* = o,$ $f_{15}(x^*) = f\_bias = -180$	$x \in [0,600]^D$

TABLE 6. Mean values of the best solutions found in 30 independent runs by the PGS and other algorithms for rotated test functions

Algorithm	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
S-PSO	$1.27\times10^{+03}$	$6.61\times10^{+03}$	$1.32\times10^{+07}$	8.7	2.13×10	2.81×10
SS-BLX	3.40×10	1.73	$1.14\times10^{+02}$	4.19	2.03×10	$1.96\times10^{+03}$
SS-Arith	1.06	5.28	$4.94\times10^{+02}$	5.96	2.03×10	$1.90\times10^{+03}$
DE-Exp	8.26×10^{-9}	8.18×10^{-9}	8.39×10^{-9}	8.15×10^{-9}	2.03×10	$1.26\times10^{+03}$
DE-Bin	7.71×10^{-9}	8.34×10^{-9}	7.95×10^{-9}	4.54	2.03×10	$1.26\times10^{+03}$
SaDE	8.41×10^{-9}	8.20×10^{-9}	1.61×10^{-2}	8.33×10^{-9}	2.03×10	$1.26\times10^{+03}$
HEPSO	**0**	4.02×10^{-04}	4.86×10^{-03}	$\mathbf{2.37\times10^{-9}}$	2.14×10	6.18×10^{-01}
PGS	**0**	$\mathbf{8.21\times10^{-14}}$	$\mathbf{1.77\times10^{-03}}$	4.12	**1.32**	$\mathbf{6.99\times10^{-02}}$



(a)

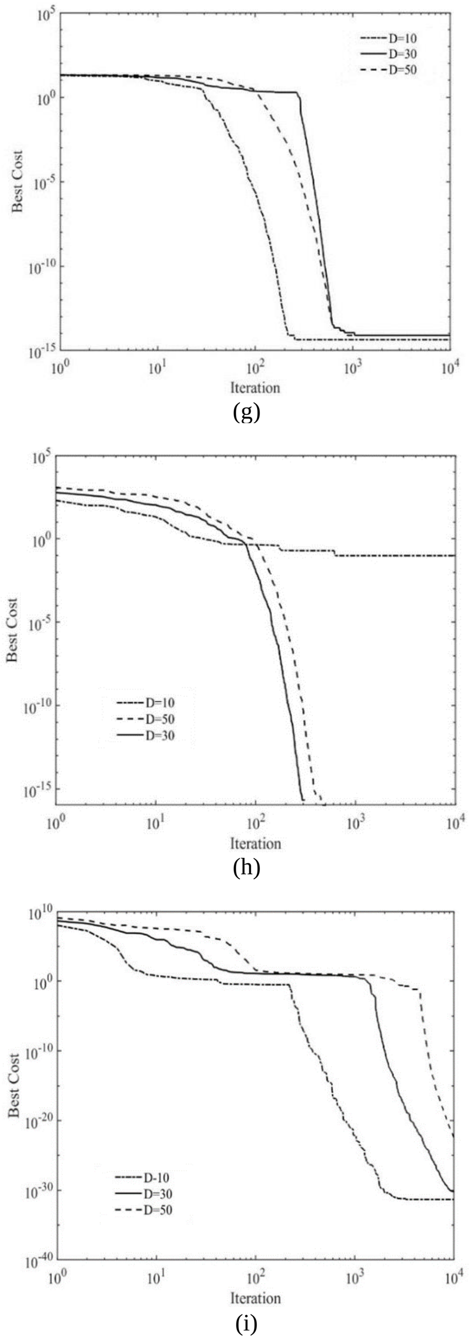
(d)

(b)

(e)

(c)

(f)



(g)

(h)

(i)

Figure 6. Convergence trajectory of the PGS algorithm on the test functions, (a) f_1; (b) f_2 (c) f_3; (d) f_4; (e) f_5; (f) f_6; (g) f_7; (h); (i) f_9.

regulation schemes (73, 74), and nonlinear controllers (75, 76) have been introduced to stabilize these nonlinear unstable systems.

The configuration of the quadcopter regarded in this study is illustrated in Figure 7, and its physical parameters are displayed in Table 8. The following assumptions are considered for its dynamics.

TABLE 7. Mean values of the iteration numbers required to reach the threasholds found in 30 independent runs to illustrate the convergence speed of the PGS, APSO and F-GA&PSO

Test function	PGS	APSO	F-GA&PSO
f_1	**91**	354	2328
f_2	**93**	395	2323
f_3	308	1059	**4**
f_4	**23**	267	2054
f_5	**39**	246	3585
f_6	**699**	3906	6136
f_7	**96**	2037	7244
f_8	**82**	379	2263
f_9	**884**	1077	2841

Figure 7. General view of the investigated quadrotor system

TABLE 8. Numerical values of the physical parameters related to the quadcopter (77)

Physical parameter	Explanation	Value
m	Total mass	0.335 kg
L	Effective length	0.18 m
I_x	Moment of inertia around axis x	0.0018 kgm^2
I_y	Moment of inertia around axis y	0.0018 kgm^2
I_z	Moment of inertia around axis z	0.0047 kgm^2
c	Thrust coefficient	5.7231$\times 10^{-6}$ Ns^2
d	Drag coefficient	1.7169$\times 10^{-7}$ Nms^2
J_r	Rotor inertia	1.85$\times 10^{-5}$ kgm^2

- The mass center and origin of the body coordinate system are coincided on the symmetry center.
- The body coordinate axes are coincided on the principle axes of the quadcopter.
- The thrust forces of the blades are proportional to the square angular velocity.
- The structure of the quadcopter and blades are rigid.

- The structure is symmetric.
- The earth is flat.

The inertia coordinate system XYZ and body coordinate system xyz are illustrated in Figure 8. The difference between the clockwise and counter-clockwise velocities is utilized to calculate the total angular velocity as follows.

$$\Omega = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 \tag{15}$$

where, Ω_i $(i = 1,2,3,4)$ are the angular velocities of the four blades. The vector of external torques $(\vec{M})$ is computed by applying the Newton second law as follows (77):

$$\vec{M} = \begin{bmatrix} I_x\dot{\omega}_x + \omega_y\omega_z(I_z - I_y) + J_r\Omega\omega_y \\ I_y\dot{\omega}_y + \omega_x\omega_z(I_x - I_z) + J_r\Omega\omega_x \\ I_z\dot{\omega}_z + \omega_x\omega_y(I_y - I_x) \end{bmatrix} \tag{16}$$

where, ω_x, ω_y and ω_z are the angular velocities of the quadcopter around x, y and z axes, respectively. I_x, I_y and I_z are the inertia moments of the quadcopter around x, y and z axes, correspondingly. Moreover, J_r denotes the inertia moment of the rotor. The motor thrust forces (T_i) and the torques implemented by air on the blades (τ_i) could be calculated by the following equations.

$$T_i = c\Omega_i^2 \text{ and } i = 1,2,3,4 \tag{17}$$

$$\tau_i = d\Omega_i^2 \text{ and } i = 1,2,3,4 \tag{18}$$

where, c and d respectively represent the thrust and drag parameters. Moreover, torques M_x, M_y and M_z would be determined by applying Equation (19).

$$\vec{M} = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} (T_4 - T_2)L \\ (T_1 - T_3)L \\ \tau_2 - \tau_1 + \tau_4 - \tau_3 \end{bmatrix} = \begin{bmatrix} (T_4 - T_2)L \\ (T_1 - T_3)L \\ \frac{d}{c}(T_2 - T_1 + T_4 - T_3) \end{bmatrix} \tag{19}$$

The following equation is obtained by employing Equations 15 through 19.

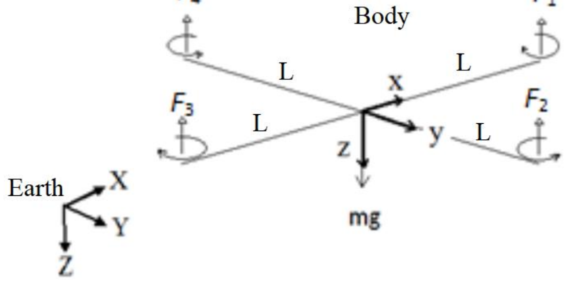


Figure 8. Body and inertial frames for the studied quadrotor system

$$\begin{bmatrix} \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\Psi} \end{bmatrix} = \begin{bmatrix} a_1(\dot{\theta}\dot{\Psi}) - a_4(\dot{\theta} + \varphi\dot{\Psi})\Omega + b_3M_z\theta + b_1M_x \\ a_2(\dot{\varphi}\dot{\Psi}) + a_5(\dot{\varphi} - \theta\dot{\Psi})\Omega - b_3M_z\varphi + b_2M_y \\ a_3(\dot{\theta}\dot{\varphi}) + (a_5\Omega\dot{\varphi} + b_2M_y)\varphi + b_3M_z \end{bmatrix} \tag{20}$$

where,

$$a_1 = \frac{I_y - I_z}{I_x} + 1, a_2 = \frac{I_z - I_x}{I_y} - 1, a_3 = \frac{I_x - I_y}{I_z} + 1,$$
$$a_4 = \frac{J_r}{I_x}, a_5 = \frac{J_r}{I_y}, b_1 = \frac{1}{I_x}, b_2 = \frac{1}{I_y}, b_3 = \frac{1}{I_z} \tag{21}$$

In this research work, the state-space variables for the system are considered as follows:

$$\begin{cases} x_1(t) = \varphi(t) \\ x_2(t) = \dot{\varphi}(t) \\ x_3(t) = \theta(t) \\ x_4(t) = \dot{\theta}(t) \\ x_5(t) = \Psi(t) \\ x_6(t) = \dot{\Psi}(t) \end{cases} \text{and} \begin{cases} U_2(t) = M_x \\ U_3(t) = M_y \\ U_4(t) = M_z \end{cases} \tag{22}$$

Hence, the state-space form of Equation 20 could be rewritten as follows:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1(x_4x_6) - a_4(x_4 + x_1x_6)\Omega + b_3U_4x_3 + b_1U_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = a_2(x_2x_6) + a_5(x_2 - x_3x_6)\Omega - b_3U_4x_1 + b_2U_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = a_3(x_2x_4) + a_5(x_1x_2)\Omega + b_2U_3x_1 + b_3U_4 \end{cases} \tag{23}$$

The following equation is defined for movement in the direction of z axis in the inertia system with regard to the weight and thrust forces.

$$\ddot{z} = -g + \frac{\cos\theta\cos\varphi}{m}U_1 \tag{24}$$

If

$$[x_7, x_8]^T = [z, \dot{z}]^T \tag{25}$$

with considering Equation 22, Equation 24 can be rewritten as follows:

$$\dot{x}_7 = x_8$$
$$\dot{x}_8 = -g + \frac{\cos\theta\cos\varphi}{m}U_1 \tag{26}$$

It is noticeable that U_i, $i = 1,2,3,4$ are the outputs of control systems that should be converted to the dynamical model inputs, i.e. the angular velocities of the propellers $(\Omega_i, i = 1,2,3,4)$. Therefore, the following equations are defined to simply express the control outputs.

$$\begin{cases} U_1 = T_1 + T_2 + T_3 + T_4 \\ U_2 = (T_4 - T_2)L \\ U_3 = (T_1 - T_3)L \\ U_4 = \frac{d}{c}(T_2 + T_4 - T_1 - T_3) \end{cases} \tag{27}$$

Moreover, the following equations are introduced to find the angular velocities of the propellers.

$$\begin{bmatrix} {\Omega_1}^2 \\ {\Omega_2}^2 \\ {\Omega_3}^2 \\ {\Omega_4}^2 \end{bmatrix} = \frac{1}{c}\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \frac{1}{c}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -L & 0 & L \\ L & 0 & -L & 0 \\ -\frac{d}{c} & \frac{d}{c} & -\frac{d}{c} & \frac{d}{c} \end{bmatrix}\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{28}$$

Finally, a control output based on the state feedback formulation can be defined as follows (78):

$$U = -\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & K_7 & K_8 \\ K_1 & K_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_3 & K_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_5 & K_6 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \tag{29}$$

where, design vector K = $[K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8]$ has the positive constant components regarded as the control gains that would be optimally determined by applying the PGS optimization algorithm. Generally, the performance of a closed loop control system could be evaluated via different objectives. In this paper, the summation of integral of absolute values of $\varphi(t), \theta(t), \Psi(t)$ and $z(t)$ is considered as the cost function that must be minimized.

$$cost\ function = \int|\varphi(t)|dt + \int|\theta(t)|dt + \int|\Psi(t)|dt + \int|z(t)|dt \tag{30}$$

Figure 9 depicts the optimization diagrams obtained using the proposed optimization approach for optimum design of the state feedback controller for the quadrotor through five independent runs. The design variables and objective function corresponding to the best optimum design point are listed in Table 9. Figure 10 displays the behavior of the quadrotor and the comparison with the optimum LQR based controller developed by Parhizkar and Naghash (77). It is obvious from these results that the optimal controller of this study is preferred due to its higher convergence and shorter settling time.

6. OPTIMUM DESIGN OF AN INELASTIC MEMBER

Compression members are widely encountered in many practical applications such as structures, columns, frames and trusses. In this example, the objective is minimizing the weight of the compression members to satisfy the AISC manual requirements (79). Different cross-sectional shapes of steel members could be employed for the compression members, such as wide-flange sections (W-shape), angle sections, channel sections, tee sections, hollow circular/square sections, solid circular or square/sections. Here, a W-shape cross-section shown in Figure 11 is considered for optimum design of its sizes with considering inequality constraints. The load for the member is calculated based on its application and in the regarded region while the specified material is ASTM A992 Grade 50 steel.

The objective function is defined as the minimization of the weight per unit length given as the cross-sectional area density:

$$cost\ function = 12\gamma A_g \tag{31}$$

where, γ denotes the density of steel which is equal to 0.283 Ib/in^3, and A_g is the cross-section area formulated as follows:

$$A_g = 2b_f t_f + (d - 2t_f)t_w \tag{32}$$

where, b_f and t_f respectively denote the width and thickness of the flange, d represents the depth of the section, and t_w is the thickness of the web. These four parameters are considered as the design variables having the following upper and lower bounds.

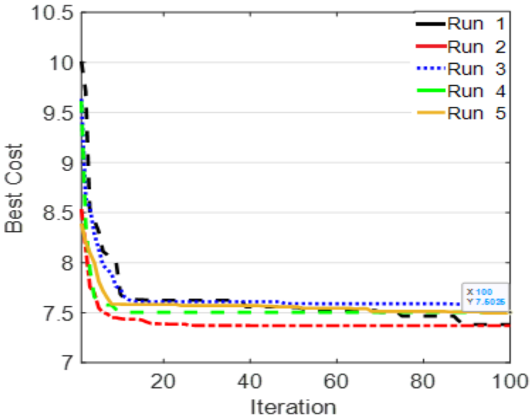


Figure 9. Optimization diagrams for the optimum design of the state feedback controller related to the quadrotor

TABLE 9. Design variables and objective functions corresponding to the optimum design point for control of the quadcopter

Quantity	Optimum value
K_1	0.2896
K_2	0.0329
K_3	0.2971
K_4	0.0265
K_5	0.9844
K_6	0.0822
K_7	1.3082
K_8	0.9038
Objective function by this work	7.3822
Objective function by the method in (77)	10.3707

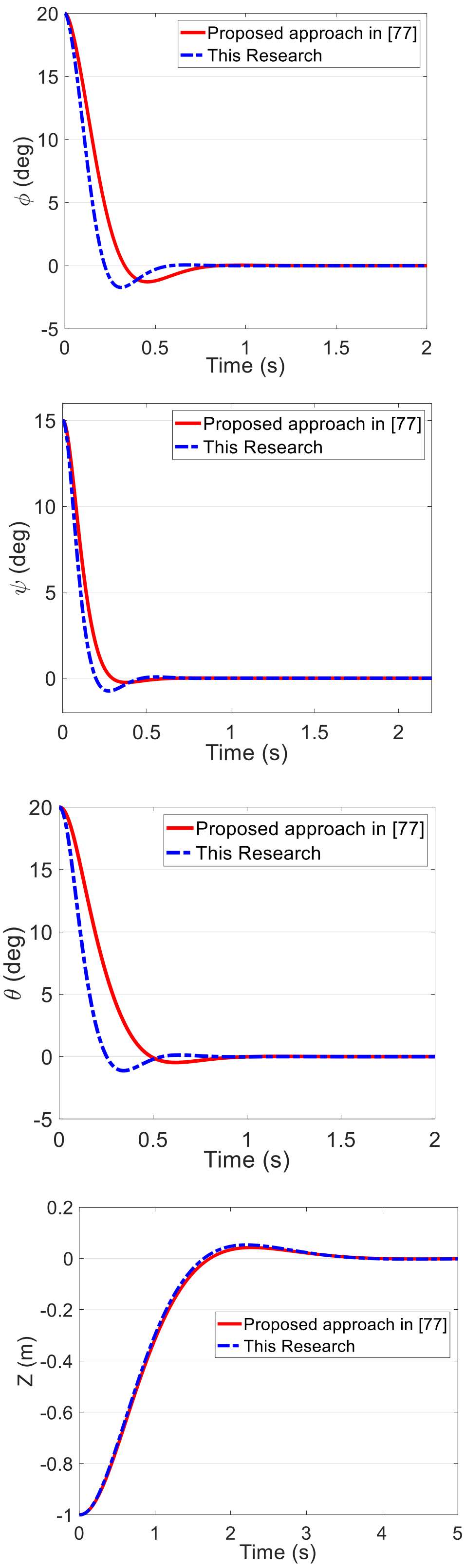


Figure 10. Behavoiur of the quadrotor handled by the optimal state feedback of this research and the comparison with the optimum LQR based controller developed in (77)

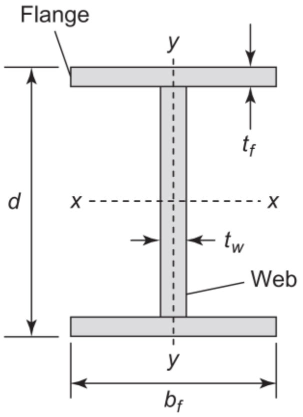


Figure 11. Cross-section and design parameters of the considered inelastic member

$$6.0 \leq b_f \leq 11.7 \tag{33}$$

$$0.425 \leq t_f \leq 2.5 \tag{34}$$

$$17.7 \leq d \leq 21.1 \tag{35}$$

$$0.30 \leq t_w \leq 1.16 \tag{36}$$

Moreover, it is supposed that the buckling of the member occurs about the weak axis (y) that imposes the following inequality constraints:

$$\lambda_y \leq \lambda_e \tag{37}$$

$$P_a \leq 0.6 F_{cry} A_g \tag{38}$$

where, λ_y represents the slenderness ratio for buckling with respect to weak axis y, λ_e denotes the limiting value of the slenderness ratio for inelastic buckling, and F_{cry} is the critical stress. These quantities could be respectively computed by applying the following relations.

$$\lambda_y = \frac{K_y L_y}{r_y} \tag{39}$$

$$\lambda_e = 4.71\sqrt{\frac{E}{F_y}} \tag{40}$$

$$F_{cry} = \left(0.658^{F_y/F_{ey}}\right)F_y \tag{41}$$

where, K_y denotes the effective length factor for the buckling with respect to weak axis y set at 1.0, L_y is the laterally unsupported length of the member for buckling with respect to weak axis y set at 180 (in), E illustrates the modulus of elasticity set at 29,000 (Ksi) , and F_y is the specified minimum yield stress set at 50 (Ksi). Besides, r_y and F_{ey} respectively present the radius of the gyration and the Euler stress for buckling with respect to weak axis y that would be calculated by applying the following equations.

$$r_y = \sqrt{I_y/A_g} \tag{42}$$

$$F_{ey} = \frac{\pi^2 E}{\lambda_y^2} \tag{43}$$

where, I_y denotes the moment of inertia about weak axis y computed as follows.

$$I_y = 2\left(\frac{1}{12}t_f b_f^3\right) + \frac{1}{12}(d-2t_f)t_w^3 \tag{44}$$

Furthermore, to avoid the local buckling for the flange and web, the following inequality constraints should be imposed:

$$\frac{(d-2t_f)}{t_w} \le 0.56\sqrt{\frac{E}{F_y}} \tag{45}$$

$$\frac{b_f}{2t_f} \le 1.49\sqrt{\frac{E}{F_y}} \tag{46}$$

By considering the mentioned descriptions, the optimum design of the inelastic member is performed by utilizing the suggested PGS optimization approach, and the optimization trajectories for five independent runs are depicted in Figure 12. The design variables and objective function corresponding to the best optimum design point are presented in Table 10. It is obvious from these results that the optimum design found by this study is preferred in comparison with data reported in literature (79) due to its minimum weight.

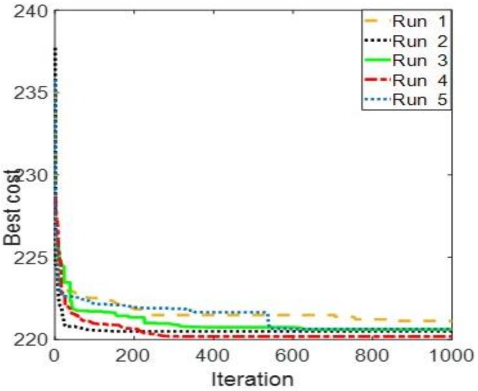


Figure 12. Optimization diagrams for the optimum design of the inelastic member

TABLE 10. Design variables and objective functions corresponding to the best solution for the optimum design of the inelastic member

Quantity	Optimum value
b_f	11.69688 (in)
t_f	2.2123 (in)
t_w	0.9853 (in)
d	17.7003 (in)
Objective function by this work	220.1813
Objective function by the method in (75)	223.8494

7. CONCLUSIONS AND FUTURE WORKS

This paper introduces a new particle swarm optimization algorithm that combines an adaptive inertia weight and acceleration coefficients with a novel crossover operator. Specifically, a second-order sliding surface is employed to dynamically calculate algorithm parameters throughout iterations. The proposed crossover operator utilizes both the global best position of the swarm and the current position of the particle under consideration to generate a new chromosome. The performance of the introduced optimization algorithm is evaluated through various mathematical test functions, and the results are compared against those of other well-known approaches. The analysis demonstrates the superiority of the suggested strategy in terms of solution accuracy and convergence speed. Moreover, the proposed optimization algorithm is successfully applied to designing the state feedback controller for a nonlinear quadrotor system, as well as determining the cross-section dimensions of an inelastic compression beam. The feasibility and efficiency of the PGS strategy are assessed in comparison with recently introduced schemes for these real-world problems. The most important limmitations of the PGS are as: (1) It is not applicable, in this version, to handle the multi-objective problems; (2) Selection proper values for the parameters of the algorithm for different problems.

As part of future research, the following directions are suggested:

- Developing the introduced PGS algorithm for solving multi-objective optimization problems.
- Applying the suggested adaptive parameter for computing coefficients of other algorithms.
- Employing the proposed scenario for combination of other optimization algorithms.
- Utilizing the PGS algorithm for optimum design of other complicated nonlinear constrained problems.

Authors' Contributions

Conceptualization and simulations by A. R. Nemati, text preparation and review by M. J. Mahmoodabadi, and editing by N. Danesh.

Data Availability

The authors confirm that the data supporting the findings of this study are available within the article.

7. REFERENCES

1. Ghafari R, Mansouri N. An Efficient Task Scheduling Based on Seagull Optimization Algorithm for Heterogeneous Cloud Computing Platforms. International Journal of Engineering, Transactions B: Applications. 2022;35(2):433-450. https://doi.org/10.5829/IJE.2022.35.02B.20
2. Sridhar Reddy A, Satish Chembuly VVMJ, Kesava Rao VVS. Collision-free Inverse Kinematics of Redundant Manipulator for Agricultural Applications through Optimization Techniques. International Journal of Engineering, Transactions A: Basics. 2022;35(7):1343-1354. https://doi.org/10.5829/IJE.2022.35.07A.13
3. Mehri F, Mollaei S, Noroozinejad Farsangi E, Babaei M, Ghahramani F. Application of a Novel Optimization Algorithm in Design of Lead Rubber Bearing Isolation Systems for Seismic Rehabilitation of Building Structures. International Journal of Engineering, Transactions C: Aspects. 2023;36(3):594-603. https://doi.org/10.5829/IJE.2023.36.03C.20
4. Kong XY, Yang GH. An Intrusion Detection Method Based on Self-Generated Coding Technology for Stealthy False Data Injection Attacks in Train-Ground Communication Systems. IEEE Transactions on Industrial Electronics. 2023;70(8):8468-8476. https://doi.org/10.1109/TIE.2022.3213899
5. Dhanasekaran B, Siddhan S, Kaliannan J. Ant colony optimization technique tuned controller for frequency regulation of single area nuclear power generating system. Microprocessors and Microsystems. 2020; 73: 102953. https://doi.org/10.1016/j.micpro.2019.102953
6. Cao Y, Huang L, Li Y, Jermsittiparsert K, Ahmadi-Nezamabad H, Nojavan S. Optimal scheduling of electric vehicles aggregator under market price uncertainty using robust optimization technique. International Journal of Electrical Power & Energy Systems. 2020;117:105628. https://doi.org/10.1016/j.ijepes.2019.105628
7. Kaur J, Reddy SRN. Implementation of linux optimization technique for ARM based system on chip. Procedia Computer Science.2020;171:1780-1789. https://doi.org/10.1016/j.procs.2020.04.191
8. Shabani A, Asgarian B, Salido M, Gharebaghi SA. Search and rescue optimization algorithm: A new optimization method for solving constrained engineering optimization problems. Expert Systems with Applications. 2020;161(15):113698. https://doi.org/10.1016/j.eswa.2020.113698
9. Mirrashid N, Alibeiki E, Rakhtala SM. Development and Control of an upper Limb Rehabilitation Robot via Ant Colony Optimization -PID and Fuzzy-PID Controllers. International Journal of Engineering, Transactions B: Applications. 2022;35(8):1488-1493. https://doi.org/10.5829/IJE.2022.35.08B.04
10. Vanaei P, Jalili B, Hosseinzadeh M, Jalili P. Efficiency Optimization Thermal Analysis and Power Output of a Modified Incinerator Plant Using Organic Rankine Cycle. International Journal of Engineering, Transactions A: Basics. 2023;36(7):1300-1309. https://doi.org/10.5829/IJE.2023.36.07A.11
11. Holland J. Adaptation in Natura and Artificia Systems University of Michigan Press. Ann Arbor MI; 1975. https://doi.org/10.7551/mitpress/1090.001.0001
12. Kennedy J, Eberhart RC. Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks IV; 1995. P. 1942-1948. https://doi.org/10.1109/ICNN.1995.488968
13. Mohammadi S, Babagoli M. A Hybrid Modified Grasshopper Optimization Algorithm and Genetic Algorithm to Detect and Prevent DDoS Attacks. International Journal of Engineering, Transactions A: Basics. 2021;34(4):811-824. https://doi.org/10.5829/IJE.2021.34.04A.07
14. Zhou P, Ma X, Zhang Sh, Liu Zh, Meng Zh, Xiang Zh, Wang X, Sun T, Lin X, Li Y. Application of particle swarm optimization in the design of a mono-capillary X-ray lens. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 2020;953(11):163077. https://doi.org/10.1016/j.nima.2019.163077
15. Rashno A, Fadaei S. Image Restoration by Projection onto Convex Sets with Particle Swarm Parameter Optimization. International Journal of Engineering, Transactions B: Applications. 2023;36(2):398-407. https://doi.org/10.5829/IJE.2023.36.02B.18
16. Jozaghi T, Wang C, Arroyave R, Karaman I. Design of alumina-forming austenitic stainless steel using genetic algorithms. Materials & Design. 2020;186(15):108198. https://doi.org/10.1016/j.matdes.2019.108198
17. Rath AK, Parhi DR, Das HC, Kumar PB, Mahto MK. Design of a hybrid controller using genetic algorithm and neural network for path planning of a humanoid robot. International Journal of Intelligent Unmanned Systems. 2020;9(3):2049-6427. https://doi.org/10.1108/IJIUS-10-2019-0059
18. Mahdavi S, Shaterzadeh A, Jafari M. Determination of optimum effective parameters on thermal buckling of hybrid composite plates with quasi-square cut-out using a genetic algorithm. Engineering Optimization. 2019;52(1):106-121. https://doi.org/10.1080/0305215X.2019.1575965
19. Ehsani A, Rezaeepazhand J. Stacking sequence optimization of laminated composite grid plates for maximum buckling load using genetic algorithm. International Journal of Mechanical Sciences. 2016;119:97-106. https://doi.org/10.1016/j.ijmecsci.2016.09.028
20. Le-Manh T, Lee J. Stacking sequence optimization for maximum strengths of laminated composite plates using genetic algorithm and isogeometric analysis. Composite Structures. 2014;116:357-363. https://doi.org/10.1016/j.compstruct.2014.05.011
21. Imran M, Shi D, Tong L, Waqas HM. Design optimization of composite submerged cylindrical pressure hull using genetic algorithm and finite element analysis. Ocean Engineering. 2019;190(15):106443. https://doi.org/10.1016/j.oceaneng.2019.106443
22. Wei R, Pan G, Jiang J, Shen K, Lyu D. An efficient approach for stacking sequence optimization of symmetrical laminated composite cylindrical shells based on a genetic algorithm. Thin-Walled Structures. 2019;142:160-170. https://doi.org/10.1016/j.tws.2019.05.010
23. Ibrahim MA, Mahmood AK, Sultan NS. Optimal PID controller of a brushless DC motor using genetic algorithm. International Journal of Power Electronics and Drive System. 2019;10(2):822-830. https://doi.org/10.11591/ijpeds.v10.i2.pp822-830
24. Sun Y, Xue B, Zhang M, Yen GG, Lv J. Automatically designing CNN architectures using the genetic algorithm for image classification. IEEE Transactions on Cybernetics. 2020;50(9):1-15. https://doi.org/10.1109/TCYB.2020.2983860
25. Zamani M, Ghartemani MK, Sadati N, Parniani M. Design of fractional order PID controller for AVR using particle swarm optimization. Control Engineering Practice. 2009;17(12)1380-1387. https://doi.org/10.1016/j.conengprac.2009.07.005
26. Ezzeddine T. Reactive power analysis and frequency control of autonomous wind induction generator using particle swarm optimization and fuzzy logic. Energy Exploration & Exploitation. 2019;38(3). https://doi.org/10.1177/014459871988637
27. Hamed MA, Abo-Bakr RM, Mohamed SA, Eltaher MA. influence of axial load function and optimization on static

stability of sandwich functionally graded beams with porous core. Engineering with Computers. 2020;36:1929–1946. https://doi.org/10.1007/s00366-020-01023-w

28. Keshtegar B, Nguyen-Thoi T, Troung TT, Pengzhu S. Optimization of buckling load for laminated composite plates using adaptive Kriging-improved PSO: A novel hybrid intelligent method. Defence Technology, available online. 2020;17(1):85-99. https://doi.org/10.1016/j.dt.2020.02.020

29. Huang L, Tai ng C, Sheik AH, Griffith MC. Niching particle swarm optimization techniques for multimodal buckling maximization of composite laminates. Applied Soft Computing. 2017;57:495-503. https://doi.org/10.1016/j.asoc.2017.04.006

30. Jansen PW, Perez RE. Constrained structural design optimization via a parallel augmented Lagrangian particle swarm optimization approach. Computers & Structures. 2011;89(13-14):1352-1366. https://doi.org/10.1016/j.compstruc.2011.03.011

31. Nguyen QH, Ly HB, Le TT, Nguyen TA, Phan VH, Tran VQ, Pham BT. Parametric investigation of particle swarm optimization to improve the performance of the adaptive neuro-fuzzy inference system in determining the bucking capacity of circular opening steel beams. Materials. 2020;**13**(10). https://doi.org/10.3390/ma13102210

32. Ye J, Hajirasouliha I, Becque J, Eslami A. Optimum design of cold-formed steel beams using particle swarm optimization method. Journal of Constructional Steel Research1. 2016;22:80-93. https://doi.org/10.1016/J.JCSR.2016.02.014

33. Xu J, Tan W, Li T. Predicting fan blade icing by using particle swarm optimization and support vector machine algorithm. Computers & Electrical Engineering. 2020;87:106751. https://doi.org/10.1016/j.compeleceng.2020.106751

34. van Hentenryck P, Milano M. Hybrid Optimization: The Ten Years of CPAIOR. Springer Optimization and Its Applications. 2011;45. https://doi.org/10.1007/978-1-4419-1644-0

35. Mach JB, Ronoh KK, LangatK. Improved spectrum allocation scheme for TV white space networks using a hybrid of firefly, genetic, and ant colony optimization algorithms. Heliyon. 2023;9(3):13752. https://doi.org/10.1016/j.heliyon.2023.e13752

36. Jafari M, Salajegheh E, SalajeghehJ. Optimal design of truss structures using a hybrid method based on particle swarm optimizer and cultural algorithm. Structures. 2021;32:391-405. https://doi.org/10.1016/j.istruc.2021.03.017

37. DengW, Zhang L, ZhouX, ZhouY, Sun Y, Zhu W, Chen H, Deng W, Chen H, Zhao H. Multi-strategy particle swarm and ant colony hybrid optimization for airport taxiway planning problem. Information Sciences. 2022;612:576-593. https://doi.org/10.1016/j.ins.2022.08.115

38. Divasón J, Pernia-Espinoza A, Martinez-de-Pison FJ. HYB-PARSIMONY: A hybrid approach combining Particle Swarm Optimization and Genetic Algorithms to find parsimonious models in high-dimensional datasets. Neurocomputing. 2023;560:126840. https://doi.org/10.1016/j.neucom.2023.126840

39. Li H, Sun B, Hao J, Zhao J, Li J, Khakichi A. Economical planning of fuel cell vehicle-to-grid integrated green buildings with a new hybrid optimization algorithm. International Journal of Hydrogen Energy. 2022;47(13):8514-8531. https://doi.org/10.1016/j.ijhydene.2021.12.156

40. Deng S, Pan HY, Wang HG, Xu SK, Yan XP, Li CW, Peng MG, Peng HP, Shi L, Cui M, Zhao F. A hybrid machine learning optimization algorithm for multivariable pore pressure prediction. Petroleum Science; 2024. https://doi.org/10.1016/j.petsci.2023.09.001

41. Fontes DBMM, Homayouni SM, Gonçalves JF. A hybrid particle swarm optimization and simulated annealing algorithm for the job shop scheduling problem with transport resources. European Journal of Operational Research. 2023;306(3):1140-1157. https://doi.org/10.1016/j.ejor.2022.09.006

42. Bansal S, Aggarwal H. A Hybrid Particle Whale Optimization Algorithm with application to workflow scheduling in cloud–fog environment. Decision Analytics Journal. 2023;9:100361. https://doi.org/10.1016/j.dajour.2023.100361

43. Anter AM, Elaziz MA, Zhang Z. Real-time epileptic seizure recognition using Bayesian genetic whale optimizer and adaptive machine learning. Future Generation Computer Systems. 2022;127:426-434. https://doi.org/10.1016/j.future.2021.09.032

44. Li Z, Huang J, Wang J, Ding M. Development and application of hybrid teaching-learning genetic algorithm in fuel reloading optimization. Progress in Nuclear Energy. 2021;139:103856. https://doi.org/10.1016/j.pnucene.2021.103856

45. Sivanandam SN, Deepa SN. Introduction to genetic algorithms. Springer Computational Intelligence and Complexity; 2008. book/10.5555/1951762

46. Clerc M. Particle swarm optimization. Wiley-ISTE; 2006. book/10.1002/9780470612163

47. Song C, Fei S, Cao J, Huang C. Robust synchronization of fractional-order uncertain chaotic systems based on output feedback sliding mode control. Mathematics. 2019;7(7):599. https://doi.org/10.3390/math7070599

48. Wang S, Cao Y, Huang T, Chen Y, Li P, Wen S. Sliding mode control of neural networks via continuous or periodic sampling event-triggering algorithm. Neural Networks. 2020;121:140-147. https://doi.org/10.1016/j.neunet.2019.09.001

49. Wang J, Yang C, Shen H, Cao J, Rutkowski L. Sliding-mode control for slow-sampling singularly perturbed systems subject to markov jump parameters. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2020;51(12):1-8. https://doi.org/10.1109/TSMC.2020.2979860

50. Kong X, Zhang T. Distributed Cooperative Sliding Mode Fault-Tolerant Control for Multiple High-Speed Trains Based on Actor-Critic Neural Network. Journal of Mathematics. 2021;9943170. https://doi.org/10.1155/2021/9943170

51. Bowman F. Introduction to Bessel functions. Dover Books on Mathematics; 2010. https://doi.org/10.1021/j150403a019

52. Deb K, Beyer HG. Self-adaptive genetic algorithms with simulated binary crossover. Evoltionary Computation. 2001;9(2):197-221. https://doi.org/10.1162/106365601750190406

53. Wei-Der Chang A. Multi-crossover genetic approach to multivariable PID controllers tuning. Expert Systems with Applications. 2007;33(3):620-626. https://doi.org/10.1016/j.eswa.2006.06.003

54. Mahmoodabadi MJ, Adljooy Safaie A, Bagheri A, Nariman-zadeh N. A novel combination of Particle Swarm Optimization and Genetic Algorithm for Pareto optimal design of a five-degree of freedom vehicle vibration model. Applied Soft Computing. 2013;13:2577-2591. https://doi.org/10.1016/j.asoc.2012.11.028

55. Shi Y, Eberhart RC. A modified particle swarm optimizer, IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence; 1998. p. 69-73. https://doi.org/10.1109/ICEC.1998.699146

56. Kennedy J, Mendes R. Population structure and particle swarm performance. Proceedings of the Congress on Evolutionary Computation; 2002. P. 1671-1676. https://doi.org/10.1109/CEC.2002.1004493

57. MendesR, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better. IEEE Transactions on Evolutionary Computation. 2004;8(3):204-210. https://doi.org/10.1109/TEVC.2004.826074

58. Ratnaweera A, Halgamuge S, Watson H. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Transactions on Evolutionary Computation. 2004;8(3):240-255. https://doi.org/10.1109/TEVC.2004.826071

59. Liang JJ Suganthan PN. Dynamic multi-swarm particle swarm optimizer. Proceedings IEEE Swarm Intelligence Symposium; 2005. P. 124-129. https://doi.org/10.1109/SIS.2005.1501611

60. Liang JJ, Qin AK, Suganthan PN, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation. 2006;10(3):281-295. https://doi.org/10.1109/TEVC.2005.857610

61. Zhan ZH, Zhan JZ, Li Y, Chung HSH. Adaptive particle swarm optimization. IEEE Transactions on System, Man and Cybernetics Part B: Cybernetics. 2009;39(6):1362-1381. https://doi.org/10.1109/TSMCB.2009.2015956

62. Herrera F, Lozano M, Molina D. Continuous scatter search: an analysis of the integration of some combination methods and improvement strategies. European Journal of Operational Research. 2006;169(2):450-476. https://doi.org/10.1016/j.ejor.2004.08.009

63. Price KV, Rainer M, Lampinen JA. Differential Evolution: A Practical Approach to Global Optimization. Springer-Verlag; 2005. book/10.5555/2765832

64. Qin AK, Suganthan PN. Self-adaptive differential evolution algorithm for numerical optimization. in: Proceedings of IEEE Congress on Evolutionary Computation. 2005;2:1785-1791. https://doi.org/10.1109/CEC.2005.1554904

65. Mahmoodabadi MJ, Salahshoor Mottaghi Z, Bagheri A. HEPSO: High exploration particle swarm optimization. Information Sciences. 2014;273:101-111. https://doi.org/10.1016/j.ins.2014.02.150

66. Mahmoodabadi MJ, Babak NR. Fuzzy adaptive robust proportional–integral–derivative control optimized by the multi-objective grasshopper optimization algorithm for a nonlinear quadrotor. Journal of Vibration and Control. 2020;26(17-18). https://doi.org/10.1177/107754631990101

67. Mahmoodabadi MJ, Babak NR. Robust fuzzy linear quadratic regulator control optimized by multi-objective high exploration particle swarm optimization for a 4 degree-of-freedom quadrotor. Aerospace Science and Technology. 2020;97:105598. https://doi.org/10.1016/j.ast.2019.105598

68. Cedroa L, Wieczorkowski K. Optimizing PID controller gains to model the performance of a quadcopter. Transportation Research Procedia. 2019;40:156–169. https://doi.org/10.1016/j.trpro.2019.07.026

69. Abdelhay S, Zakriti A. Modeling of a quadcopter trajectory tracking system using PID controller. Procedia Manufacturing. 2019;32:564-571. https://doi.org/10.1016/j.promfg.2019.02.253

70. Madani T, Benallegue A. Backstepping control for a quadrotor helicopter. IEEE/RSJ International Conference on Intelligent Robots and Systems. 2006;9419317:3255–3260. https://doi.org/10.1109/IROS.2006.282433

71. Chingozha T, Nyandoro O. Adaptive sliding backstepping control of quadrotor UAV attitude. IFAC Proceedings Volumes. 2014;47(3)11043-11048. https://doi.org/10.3182/20140824-6-ZA-1003.01860

72. Raffo GV, Ortega MG, Rubio FR. An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter. Automatica. 2010;46(1):29–39. https://doi.org/10.1016/j.automatica.2009.10.018

73. Carlton Z, Wei W, Cohen K. LQR controller applied to quadcopter system dynamics identification and verification through frequency sweeps. Multi-Rotor Platform-based UAV Systems. 2020;129-152. https://doi.org/10.1016/B978-1-78548-251-9.50007-8

74. Ahmad F, Kumar P, Bhandari A, Patil PP. Simulation of the quadcopter dynamics with LQR based control. Materials Today: Proceedings. 2020;24 Part 2:326-332. https://doi.org/10.1016/j.matpr.2020.04.282

75. Castillo P, Lozano R, Dzul A. Stabilisation of a mini rotorcraft with four rotors. IEEE Control Systems Magazine. 2005;25(6):45–55. https://doi.org/10.1109/TCST.2004.825052

76. Escare˜no J, Salazar-Cruz C, Lozano R. Embedded control of a four-rotor UAV. American Control Conference. 2006;4(11):3936–3941. https://doi.org/10.1109/ACC.2008.4587031

77. Parhizkar N, Naghash A. Comparison of back stepping optimized via pso algorithm and lqr controllers for a quadrotor. Modares Mechanical Engineering. 2017;7(7). 20.1001.1.10275940.1396.17.7.47.9

78. Ghazbi SN, Akbar ALI, Reza M. Quadcopter: Full dynamic modeling, nonlinear simulation and control of attitudes degrees of freedom and the movement. Indian Journal of Scientific Research. 2014;1(2):759-769. 10.1016/j.measurement.2019.106879

79. Arora JS. Introduction to optimum design. Academic Press, 4th Edition; 2016. book/9780123813756

Persian Abstract

چکیده

به منظور افزایش عملکرد الگوریتم‌های فراابتکاری، توسعه عملگرهای جدید و ترکیب کارآمد تکنیک‌های بهینه‌سازی مختلف، استراتژی‌های ارزشمندی برای کشف راه‌حل‌های بهینه سراسری هستند. در این مقاله تحقیقاتی، یک الگوریتم بهینه‌سازی جدید به نام بهینه‌سازی ازدحام ذرات، الکوریتم ژنتیک سطح لغزنده معرفی شده است. این روش نقاط قوت بهینه سازی ازدحام ذرات، الگوریتم ژنتیک و سطح لغزش را برای تحلیل توابع آزمون ریاضی و مسائل بهینه سازی در دنیای واقعی ترکیب می کند. برای رسیدن به این هدف، تابع وزن و ضرایب یادگیری الگوریتم ازدحام ذرات با استفاده از رابطه کنترل حالت لغزشی تنظیم می شوند. بهترین ذره سراسری کشف شده از طریق روش ازدحام ذرات به عنوان یکی از والدین در عملیات ادغام الگوریتم ژنتیک عمل می کند. این اپراتور جدید سپس به طور احتمالاتی با الگوریتم بهینه‌سازی ازدحام ذرات بهبود یافته ادغام می‌شود و سرعت همگرایی آن را افزایش داده و فرار از بهینه محلی را تسهیل می‌کند. عملکرد الگوریتم پیشنهادی در هر دو توابع آزمون ریاضی تک وجهی و چند وجهی، با در نظر گرفتن موارد چرخش نشده و چرخش‌شده ارزیابی می شوند. در نهایت، اثربخشی و کارایی آن در برابر سایر تکنیک‌های بهینه‌سازی معروف و پرکاربرد آزمایش می شود. علاوه بر این، با موفقیت قابل توجهی، الگوریتم پیشنهادی در بهینه‌سازی کنترل‌کننده بازخورد حالت برای یک سیستم کوادکوپتر غیرخطی و تعیین سطح مقطع برای یک عضو تحت فشار غیرالاستیک استفاده شده است.