Bitlis Eren Üniversitesi Fen Bilimleri Dergisi
BİTLİS EREN UNIVERSITY JOURNAL OF SCIENCE
ISSN: 2147-3129/e-ISSN: 2147-3188
VOLUME: 12 NO: 3 PAGE: 673-685 YEAR: 2023
DOI:10.17798/bitlisfen.1281022

lnertia Weight-free Particle Swarm Optimization in Optimal Control Design for Vehicle Active Suspension Systems

Hasan BAŞAK[1*], Kadri DOĞAN[2]

[1]*Artvin Çoruh University, Faculty of Engineering, Dept. of Electrical-Electronics Eng. Artvin /TURKEY*
[2]*Artvin Çoruh University, Faculty of Engineering, Dept. of Basic Sciences, Artvin/TURKEY*
(ORCID:0000-0002-3724-6819) (ORCID:0000-0002-6622-3122)

Keywords: Active suspension system, Standard PSO, Inertia-weight-free PSO, LQR, Road disturbance.

Abstract
Vehicle active suspension systems play an important role in ride comfort and driving safety. This study considers the problem of an efficient control scheme design for vehicle active suspension systems. The active suspension systems aim to get more comfortable riding and good handling for random road disturbances. The purpose of this work is to reduce the driver's entire body acceleration and thereby improve ride comfort. The inertial weight-free particle swarm optimization (PSO) method is utilized to obtain weighting matrices of the optimal control namely linear quadratic regulator (LQR) for the active suspension systems. The designed state-feedback controller is applied to the quarter-car suspension system under different road profiles. Simulation results of the inertia weight-free PSO-tuned LQR are compared with the results of the classical-tuned controller and standard PSO-tuned LQR controller to show the effectiveness.

1. Introduction

Active suspension systems in vehicles aim to ensure ride comfort, road holding, and passenger safety for different road irregularities. To utilize the potential of active suspension systems, the control algorithms should deal with changing road profiles. In the literature, various control methods have been designed for active suspension systems such as sliding mode control [1], adaptive control [2], fuzzy H_∞ control [3], fuzzy [4], machine learning-based controller [5], and PID (Proportional-Derivative–Integral) with the genetic algorithm [6]. The control objectives of active suspension systems are passenger comfort, minimum vehicle body acceleration, and road handling. The linear quadratic controller is designed to obtain optimal performance without deteriorating conflict design requirements [7-9]. A state feedback optimal control law namely Linear Quadratic Regulator (LQR) offers guaranteed stability, robustness, and a structured design method for multiple-input multiple-output systems. The LQR approach computes an optimal state-feedback gain by minimizing a quadratic performance index, which consists of the state and input variables penalized by the weighting matrices. In the spite of the potential advantages, one of LQR design difficulties is the optimal selection of the weighting matrices, which does not have an efficient procedure. Bryson's method [10] can be used to obtain the initial selection of weighting matrices. However, this method is a check-test method that is a time-consuming and tiring approach. Hence, we focus on the problem of the selection of weighting matrices for the LQR control design using inertia weight-free particle swarm optimization (PSO) approach.

In literature, several researchers have proposed swarm intelligence methods to find out the optimal weighting matrices. For instance, Kumar et al. [11] use the PSO approach to obtain the weighting matrices of LQR for a two-degrees-of-freedom helicopter system. It is reported that weighting matrices obtained with the PSO make the control law

*Corresponding author: *hasanbasak@artvin.edu.tr*
Received: 11.04.2023, Accepted: 13.08.2023

optimal and also eliminate the tiring manual tuning procedure. A reinforced quantum-behaved PSO is proposed in [12] to optimize the weighting matrices of the LQR controller, which is applied to an inverted pendulum system and a flight landing system. Maghfiroh et al. [13] utilize the PSO algorithm to find optimal weighting matrices for a DC motor system. The controller provides the smallest integral of absolute the error index. Karanki et al. [14] design a PSO-based state feedback controller for a unified power-quality conditioner. Shao et al. [15] propose PSO-based LQR for a model of an overhead crane system. For vehicle active suspension systems, the bat algorithm [16], artificial fish swarm algorithm [17], and adaptive predator–prey optimization algorithm [18] are employed for the selection of optimal weighting matrices of the LQR controller.

The main contribution of this work is to propose the inertia weight-free PSO algorithm tuned optimal LQR controller for an active suspension system. The inertia weight-free PSO algorithm improves the convergence speed and accuracy of the standard PSO algorithm. The simulation studies are conducted to show the effectiveness of the proposed method. The inertia weight-free PSO-tuned LQR controller is compared with the classical tuned LQR controller and the standard PSO-tuned LQR in terms of the vehicle body acceleration, suspension deflection, and tire deflection.

2. Material and Method

2.1. Model of Active Suspension System

This section gives the dynamic equations of a quarter-active suspension system. The quarter active suspension system is depicted in Figure 1. The system has two inputs (control input F and the road surface position, z_r). The vehicle body displacement and the tire displacement are denoted by z_s and z_{us} from the ground respectively.

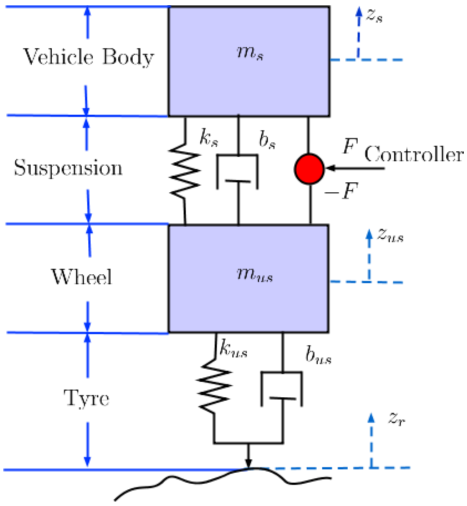

Figure 1. Diagram of the quarter active vehicle suspension system

The quarter active suspension system equation of motion are derived in [16] using the Newton law as follows:

$$\mathrm{m_{us}}\ddot{\mathrm{z}}_{\mathrm{us}} = -\mathrm{b_{us}}\dot{\mathrm{z}}_{\mathrm{us}} - \mathrm{b_s}\dot{\mathrm{z}}_{\mathrm{us}} - \mathrm{F} + \mathrm{b_s}\dot{\mathrm{z}}_{\mathrm{s}} + \mathrm{b_{us}}\dot{\mathrm{z}}_{\mathrm{r}} - (\mathrm{z_{us}} - \mathrm{z_s})\mathrm{k_s} - (\mathrm{z_{us}} - \mathrm{z_r})\mathrm{k_{us}} \tag{1}$$

$$\mathrm{m_s}\ddot{\mathrm{z}}_{\mathrm{s}} = \mathrm{b_s}\dot{\mathrm{z}}_{\mathrm{us}} + \mathrm{F} - \mathrm{b_s}\dot{\mathrm{z}}_{\mathrm{s}} - (\mathrm{z_s} - \mathrm{z_{us}})\mathrm{k_s} \tag{2}$$

Equations (1)-(2) can be given in the state–space realization as:

$$\begin{aligned} \dot{\mathrm{x}}(\mathrm{t}) &= \mathcal{A}\mathrm{x}(\mathrm{t}) + \mathcal{B}\mathrm{u}(\mathrm{t}) \\ \mathrm{y}(\mathrm{t}) &= \mathcal{C}\mathrm{x}(\mathrm{t}) + \mathcal{D}\mathrm{u}(\mathrm{t}) \end{aligned} \tag{3}$$

where the state variable vector is $x = [(z_s - z_{us})\ \dot{z}_s\ (z_{us} - z_r)\ \dot{z}_{us}]^T$ and the input vector is $u = [\dot{z}_r\ F\ ]^T$ and the output vector is $y = [\ (z_s - z_{us})\ \ddot{z}_s\ ]^T$. $(z_s - z_{us})$ and $(z_{us} - z_r)$ are the suspension and tire deflections, $\dot{z}_s$ and $\dot{z}_{us}$ are the body and the tire vertical velocities respectively. Matrices $\mathcal{A}, \mathcal{B}, \mathcal{C}$and $\mathcal{D}$ are obtained as follows:

$$\begin{aligned} \mathcal{A} &= \begin{bmatrix} 0 & 1 & 0 & -1 \\ \frac{-\mathrm{k_s}}{\mathrm{m_s}} & \frac{-\mathrm{b_s}}{\mathrm{m_s}} & 0 & \frac{\mathrm{b_s}}{\mathrm{m_s}} \\ 0 & 0 & 0 & 1 \\ \frac{\mathrm{k_s}}{\mathrm{m_{us}}} & \frac{\mathrm{b_s}}{\mathrm{m_{us}}} & \frac{-\mathrm{k_{us}}}{\mathrm{m_{us}}} & -\frac{(\mathrm{b_s}+\mathrm{b_{us}})}{\mathrm{m_{us}}} \end{bmatrix}, \\ \mathcal{B} &= \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{\mathrm{m_s}} \\ -1 & 0 \\ \frac{\mathrm{b_{us}}}{\mathrm{m_{us}}} & -\frac{1}{\mathrm{m_{us}}} \end{bmatrix}, \end{aligned} \tag{4}$$

$$\mathcal{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{-k_s}{m_s} & \frac{-b_s}{m_s} & 0 & \frac{b_s}{m_s} \end{bmatrix}, \ \mathcal{D} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{m_s} \end{bmatrix}$$

Table 1 gives the parameters of the quarter active suspension system.

Table 1. Model parameters [16].

Symbol	Value	Definition
$\mathbf{m_s}$	2.45 kg	Sprung mass
$\mathbf{m_{us}}$	1 kg	Unsprung mass
$\mathbf{k_s}$	900 N/m	Suspension stiffness
$\mathbf{k_{us}}$	1250 N/m	Tire stiffness
$\mathbf{b_s}$	7.5 Ns/m	Suspension damping coefficient
$\mathbf{b_{us}}$	5 Ns/m	Tire inherent damping coefficient

2.2. Performance Requirements

The performance requirements of the active suspension system are given in [8, 9] as:

1) Ride comfort: The vehicle body acceleration, $\ddot{z}_s$ must be reduced by the active suspension system.
2) Suspension deflection: The active suspension system has to maintain the suspension deflection within the allowable interval to avoid vehicle damage. $|(z_s - z_{us})| \leq \bar{z}$, $\bar{z}$ is the greatest acceptable suspension deflection.
3) Road handling: The wheel assembly has to stay in firm contact with the road to ensure passenger safety. Therefore, the tire's dynamic load has to be smaller than its static load ($|k_s(z_{us} - z_r)| \leq (m_s + m_{us})g$).

2.3. Problem Description

Consider the following linear time-invariant system:

$$\begin{aligned} \dot{x}(t) &= \mathcal{A}x(t) + \mathcal{B}u(t), \qquad x(0) = x_0 \\ y(t) &= \mathcal{C}x(t) + \mathcal{D}u(t) \end{aligned} \tag{5}$$

in which $x(0)$ is the initial condition. The purpose is to find the optimal control law, $u(t)$ which can drive the state variables of the dynamics to demand state by optimizing the following quadratic objective function:

$$J = \int_0^\infty x^T(t)Qx(t) + u^T(t)\mathcal{R}\,u(t)dt, \tag{6}$$

Here Q is the positive semi-definite state and $\mathcal{R}$ is the positive-definite weighting matrices respectively. Diagonal weighting matrices are generally selected. The order of Q and $\mathcal{R}$ matrices equals the number of states and the number of inputs. Assume that $(\mathcal{A}, \mathcal{B})$ is stabilisable and $(\mathcal{A}, \mathcal{C})$ is observable, then the LQR controller computes as follows:

$$u(t) = -Kx(t) \tag{7}$$

in which K is the optimal state-feedback gain computed by $K = \mathcal{R}^{-1}\mathcal{B}^T\mathcal{P}$ that is called the Lagrange multiplier based on optimization. The positive definite-matrix, $\mathcal{P}$ is obtained from the solution of the following algebraic Riccati equation:

$$\mathcal{A}^T\mathcal{P} + \mathcal{P}\mathcal{A} + Q - \mathcal{P}\mathcal{B}\mathcal{R}^{-1}\mathcal{B}^T\mathcal{P} = 0 \tag{8}$$

The design of the LQR control approach has a challenging issue in selecting the weighting matrices. The selection of weighting matrices Q and $\mathcal{R}$ affect the speed of state variables and control effort [17]. Hence, the weighting matrices can be determined by using the inertial weight-free PSO in this work. The model of the active suspension system is given as a fourth-order system. Therefore, the weighting matrix Q is set to be a 4x4 diagonal semidefinite matrix ($Q = \mathrm{diag}([q_1\ q_2\ \ q_3\ \ q_4])$. The system has a control input, F thereby a scalar, $\mathcal{R}$ is considered as $\mathcal{R} = r_1$. The corresponding J to be minimized is given as follows:

$$\begin{aligned} J &= \int_0^\infty \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}^T \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + u_1 r_1 u_1 dt \\ &= \int_0^\infty (q_1{x_1}^2 + q_2{x_2}^2 + q_3{x_3}^2 + q_4{x_4}^2 + r_1{u_1}^2)dt \end{aligned} \tag{9}$$

in which q_1, q_2, q_3 and q_4 are scalar weights of state variables. r_1 is the scalar weight of the controlled force. LQR weighting matrices are optimized using the PSO algorithm whose fitness function is given by the following integral of the time-weighted absolute error (ITAE):

$$\mathrm{ITAE} = \int_0^\infty t|e|dt \tag{10}$$

where $e = \boldsymbol{e}_{(z_s - z_{us})} + \boldsymbol{e}_{\dot{z}_s} + \boldsymbol{e}_{(z_{us} - z_r)} + \boldsymbol{e}_{\dot{z}_{us}}$ and $\boldsymbol{e}_{(z_s - z_{us})}$, $\boldsymbol{e}_{\dot{z}_s}$, $\boldsymbol{e}_{(z_{us} - z_r)}$ and $\boldsymbol{e}_{\dot{z}_{us}}$ are tracking errors of state variables for the given road profile.

2.4. Inertia Weight-free Particle Swarm Optimization

Many realistic optimization issues demand costly computation-based assessments in order to find the optimal solution. The optimization method should be carried out speedily and it should not be overly complicated [17] due to various limits in research such project time needs and computer resource constraints. Performances of optimization techniques are evaluated using different benchmark fitness functions. These algorithms often provide acceptable results by utilizing their unique information transmission methods in conjunction with a variety of first-candidate solutions in various fitness evaluations. These procedures frequently run-time and resource-intensive computer resources since they evaluate each potential resolution. The study and creation of effective optimization algorithms for assessing a small number of functions is therefore an expanding research topic. Several novel ideas have been proposed and published recently. These techniques with constrained function evaluations have produced some pleasing results [17-19].

Wilson [20] first put out the swarm idea in 1975. Each member of a swarm may use the discoveries and experiences of the others to escape from predators and find food. Each bird in a swarm can identify where it is within the swarm. Every individual will observe neighbouring individuals' flight motions to modify their own flight trajectory, giving the impression that a single entity is in charge of the whole swarm. The positions of two or three of its neighbours, as well as the flight path of the entire swarm, are the three features that each bird must see [21]. Reynolds [22] established a distributed behavioural model in light of this.

Kennedy and Eberhart [23] first developed the stochastic population-based technique known as particle swarm optimization (PSO), which was inspired by some animals' intelligent group behaviour. This technique is a unique evolutionary technique, first motivated by the specific social behaviours of fish schools and bird flocks. PSO blends evolutionary calculations with social psychology concepts from socio-cognition agents. It uses a swarm of particles to represent the potential solutions to the objective issue when applied for optimization processes. Each particle will move in the direction of the problem's probable solution after a search has started, based on its own and the partner particles' investigations. The easy implementation and the limited number of adjustable parameters of PSO are its two main benefits. Inertia weight (w), one of PSO's parameters, to strike a balance between the features of exploration and exploitation. Since the parameter's inception, several ideas for various approaches to calculating the value of inertia weight throughout the duration of a run have been made.

The standard PSO contains the following four items:

1. Determine the objective function.

2. Set parameters.

The basic parameters of the PSO include:

(i) Space dimension

(ii) Particle swarm size

(iii) Location constraint

(iv) Velocity constraint

(v) Number of iterations

(vi) Inertia weight

(vii) Learning factor: The ranges of the independent variables should be considered while determining the learning factor. Particle and particle swarm learning factors are the two different categories of learning factors. Typically, a value between 0 to 5 can be used.

3. Initialize particle swarm.

4. Update velocity and location.

Updating velocity and position is the essence of the standard PSO. The function velocity and position, which is called the PSO algorithm, is as follows:

$$\begin{aligned} v^{k+1}(m,n) = {} & wv^k(m,n) \\ & + r_1 c_1\left(xp^k(m,n) - x^k(m,n)\right) \\ & + r_2 c_2\left(xg^k(n) - x^k(m,n)\right) \end{aligned} \tag{11}$$

$$x^{k+1}(m,n) = x^k(m,n) + v^{k+1}(m,n)$$

Where $v^{k}(m,n)$ is the velocity of the m^{th} particle for the n^{th} dimension at the k^{th} iteration, $x^{k}(m,n)$ is the current position of the m^{th} particle for the n^{th} dimension at the k^{th} iteration. $xp(m,n)$ represents the position of the best solution that the m^{th} particle has achieved so far, for the n^{th} dimension of the problem. $xg(n)$ is the current global best obtained so far by the particle swarm optimization for the n^{th} dimension of the problem. w, c_1, c_2 and r_1, r_2 are defined as inertia weight, single particle's learning factor, particle swarm's learning factor and random values in [0,1], respectively [24]. The hybrid optimization algorithm is an effective combination of a metaheuristic optimization algorithm with another optimization algorithm that can exhibit more stable behaviour and greater flexibility against complicated and difficult problems. Local search algorithms use a well-specified neighborhood mechanism to recursively explore the search space for a better answer than an already existing one. Metaheuristics are made up of iterative processes that successfully integrate many sub-heuristics to find a search space. To locate global optimal areas, certain learning algorithms are employed. Natural approaches known as population-based metaheuristics investigate the search space by manipulating the population, and the outcomes heavily depend on these particular manipulative techniques. Compared to other trajectory approaches, which are easily impacted by local optima, population-based metaheuristics methods are better at characterizing local optima. Because of this, metaheuristic hybrids that effectively combine the advantages of population-based and trajectory approaches are typically quite effective and successful [25-26].

During the early search phase, the standard PSO technique often converges quickly before slowing down. It frequently has slow convergence and becomes locked in local minima. Moreover, the inertia weight, w and the pair of learning factors (c_1, c_2) are important variables affecting the standard PSO convergence. The way each particle updates are the main difference between the inertia weight-free PSO and the standard PSO algorithm. In this work, the inertia weight-free means that PSO algorithm does not have adjustable parameters of w, c_1 and c_2.The following equations show the calculation of particle positions and velocity in the inertia weight-free PSO algorithm [27]:

$$v^{k+1}(m,n) = (2r_1 - 0.5)v^k(m,n) + (2r_2 - 0.5)\left(xp^k(m,n) - x^k(m,n)\right) + (2r_3 - 0.5)(xg^k(n) - x^k(m,n)) \quad (12)$$

$$u^{k+1}(m,n) = (2r_4 - 0.5)\left(xg^k(n) - xp^k(m,n)\right) + (2r_5 - 0.5)\left(xg^k(n) - x^k(m,n)\right) \quad (13)$$

$$x^{k+1}(m,n) = xp^k(m,n) + (2r_6 - 0.5)v^{k+1}(m,n) + (2r_7 - 0.5)u^{k+1}(m,n) \quad (14)$$

where, r_1, r_2 , r_3 r_4 r_5 r_6 and r_7 are defined as random values in [0,1]. Figure 1 illustrates the flowchart of the inertia weight-free PSO algorithm.

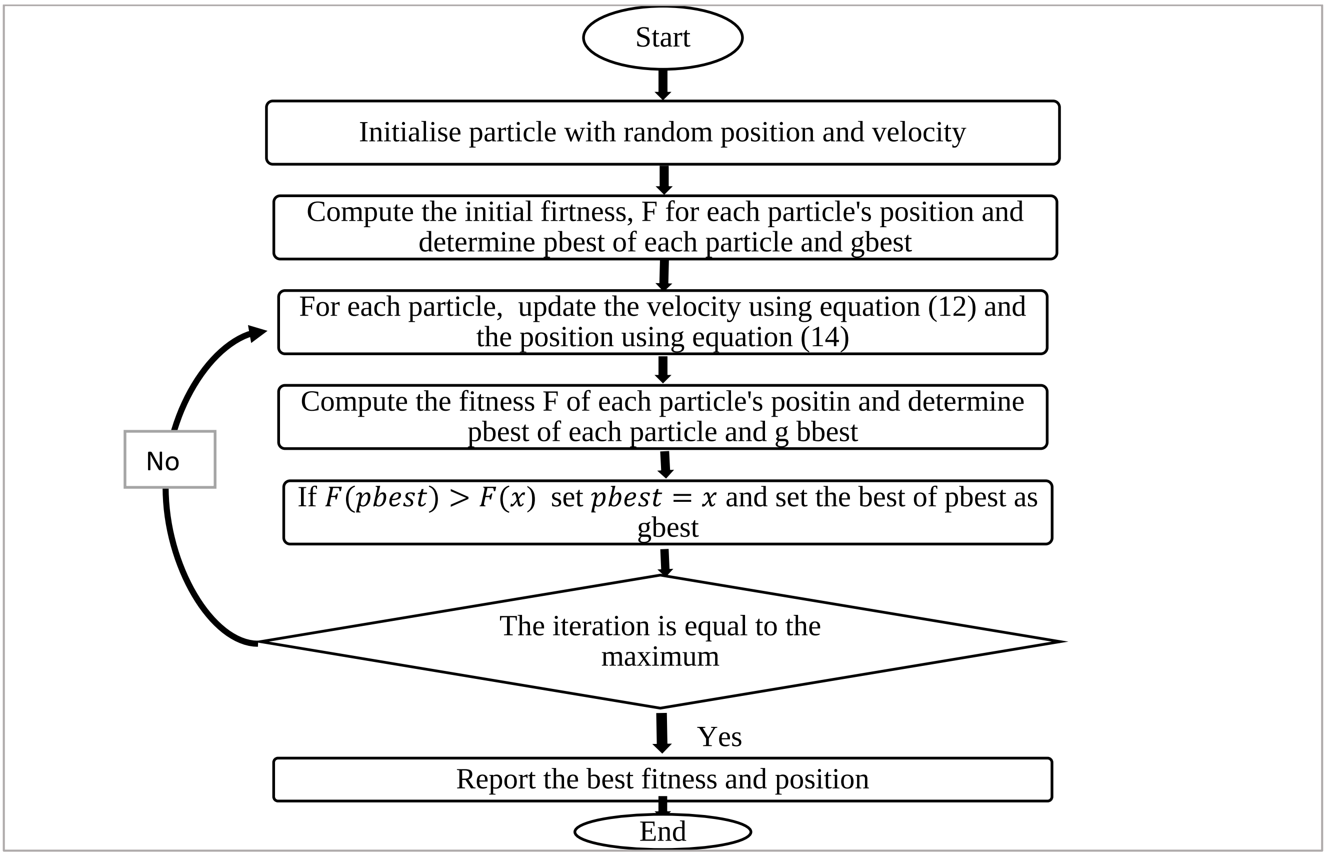

Figure 2. Flowchart of the inertia weight–free PSO algorithm

The process of LQR controller design through the inertia weight-free PSO algorithm can be summarized as follows: A state-feedback gain is computed using with Matlab lqr() command for the pair of system matrices (A,B) from equation (4) and matrices Q and R are found by the inertia weight-free PSO algorithm. For each iteration, the closed-loop system is simulated to calculate ITAE (fitness) given in equation (10). Obtained ITAE values are compared. As a result, the lowest ITAE value and corresponding matrices Q and R are recorded at the final iteration. Simulation results will be given in next section.

3. Simulation Results and Discussion

In this section, simulation results are given to show the effectiveness of the active suspension system with the designed LQR controller against random road disturbances. A square shape (with an amplitude of 0.01 m and frequency of 0.3 Hz) and a bumpy shape (with an amplitude of 0.1 m and 0.175 Hz) signals are taken into consideration as random road disturbances. Classical-tuned LQR, standard PSO-tuned LQR, and inertia weight-free PSO-tuned LQR are tested under random road disturbances.

The parameters of the standard PSO and the inertia weight–free PSO are given in Table 1. The number of the optimized parameter is five which consists of four states and a control variable. The range of weighting matrices Q and R are set between 0.01 and 500. Both the standard PSO and weight–free PSO have 60 particles and 50 iterations. The standard PSO has an inertia weight and a cognitive constant but the weight-free PSO algorithm does not have these critical weight and cognitive constants.

For both the standard PSO and the inertia weight-free PSO, the convergence of the fitness function is shown in Figure 3. It can be seen that the weight-free PSO reaches the minimum fitness value faster than the standard PSO. The standard PSO approach typically converges rapidly during the initial search period and then slows. It has the tendency of being trapped in local minima and slow convergence. Furthermore, inertia weight w, c1 and c2 are critical factors that affect the convergence of the PSO. The inertia weight–free PSO algorithm overcomes these problems. The inertia weight-free PSO takes 27 iterations to find the minimum fitness value whereas the standard PSO obtains the minimum fitness value with 34 iterations. The resulting weighting matrices and gains of state-feedback controllers are reported in Table 2.

Table 1. Parameters of the standard PSO and the inertia weight-free PSO

Parameters	Standard PSO	Inertia weight-free PSO
Iterations	50	50
Particles	60	60
Variables	5	5
Inertia weight	0.9	-
Cognitive constants	2	-

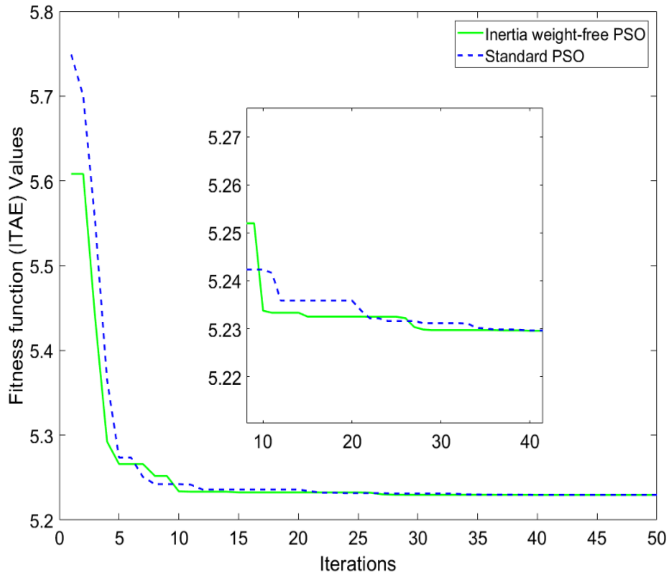

Figure 3. Comparison of fitness function values

Table 2. State feedback gains and weighting matrices

	State-feedback gain	Weighting matrices
Classical-tuned LQR [8]	$K = \begin{bmatrix} 469.4581 \\ 52.2394 \\ -179.95 \\ -12.35 \end{bmatrix}^T$	$Q = \text{diag}(692.52, 0.16, 652.52, 0.16)$ $\mathcal{R} = 0.00065$
Standard PSO-tuned LQR	$K = \begin{bmatrix} 0.0006 \\ 73.96 \\ -347.90 \\ -13.6843 \end{bmatrix}^T$	$Q = \text{diag}(0.0100, 53.5209, 1.2415, 14.0084)$ $\mathcal{R} = 0.01$
Inertia weight-free PSO-tuned LQR	$K = \begin{bmatrix} 0.00060 \\ 116.9200 \\ -494.5004 \\ -13.6274 \end{bmatrix}^T$	$Q = \text{diag}(0.0101, 136.7950, 204.8922, 19.2798)$ $\mathcal{R} = 0.01$

Figure 4 compares the vehicle body acceleration closed-loop responses for the square road profile. The inertia weight-free PSO-tuned LQR controller (black line) provides lower body acceleration than other controllers do. Figures 5 and 6 display the vehicle body and tire positions respectively. It can be seen from plots that the inertia weight-free PSO-tuned LQR controller achieves better road profile tracking. However, the classical-tuned LQR controller (green line) has an oscillatory response that might damage the vehicle. To void structural damage, the absolute value of the suspension deflection should be less than 0.038 m ($|(z_s - z_{us})| \leq 0.038$ m) [8]. The results of suspension deflections with all designed controllers are given in Figure 7. It can be seen that $|(z_s - z_{us})| \leq 0.02$ m which is within the acceptable range. The tire deflection responses are plotted in Figure 8. For road handling requirements, the tire's dynamic load has to be smaller than its static load $(((m_s + m_{us})g = 33.84\,N)$. All controllers satisfy this performance requirement. The active suspension system with the classical tuned LQR controller has tire's dynamic load, $|k_s(z_{us} - z_r)| = 17.16\ N$ and with the standard PSO-tuned LQR controller has tire's dynamic load $|k_s(z_{us} - z_r)| = 17.12\ N$. Although, the inertia weight-free PSO-tuned LQR controller has a less dynamic load

($|k_s(z_{us} - z_r)| = 17.07 \;\; N$) within the permissible range.

Similarly, closed-loop responses are given in Figures 9, 10, 11, 12 and 13 under the bumpy road profile. Vehicle body acceleration plots are given in Figure 9 to evaluate the passenger ride comfort. It is clearly seen that the inertia weight-free PSO-tuned LQR controller improves the ride comfort better than other controllers do. Figures 10 and 11 show that the vehicle body and the tire positions are tracked well with the inertia weight-free PSO-tuned LQR controller so passenger safety is ensured. Figure 12 indicates the suspension deflection under the bumpy road profile. The inertia weight-free PSO-tuned LQR controller increases the suspension deflection to improve ride comfort. The maximum value of the suspension deflection is 0.0124 m with the inertia weight-free PSO-tuned LQR controller, which is less than the permissible travel range of 0.038 m. Lastly, the tire deflection is depicted in Figure 13. The dynamic loads with the classical tuned LQR, the standard PSO-tuned LQR and the inertia weight-free PSO tuned LQR controllers are 3.049 *N*, 2.755 *N*, and 2.62 *N* respectively which are lower than the tire's static load. Furthermore, the closed-loop response with the inertia weight-free PSO-tuned LQR controllers has the best tire deflection amongst all designed controllers. ITAE values under different road profiles are reported in Table 3. Closed-loop response with the inertial weight-free PSO-tuned LQR controller has the lowest ITAE index. The above discussion and figures show that the inertial weight-free PSO-tuned LQR controller outperforms the other controllers and enhances ride comfort without an important deterioration in suspension deflection.

Table 3. ITAE values of controllers in the different road conditions.

Road Profile	Controller	ITAE values
Square road profile	Classical tuned LQR	12.7569
	Standard PSO-tuned LQR	5.2322
	Inertia weight-free PSO-tuned LQR	**5.2296**
Bumpy road profile	Classical tuned LQR	8.9835
	Standard PSO-tuned LQR	8.3998
	Inertia weight-free PSO-tuned LQR	**7.9868**

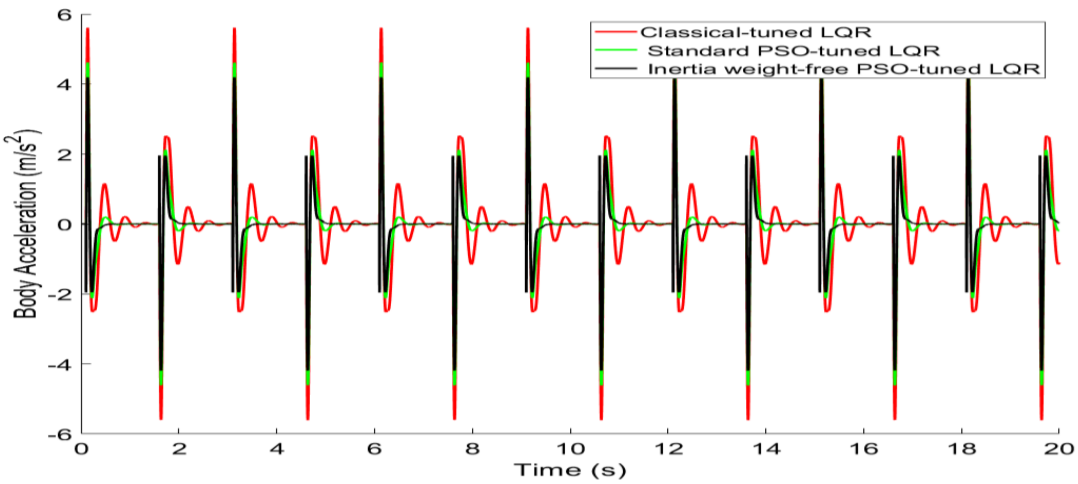

Figure 4. The vehicle body acceleration under the square road disturbance

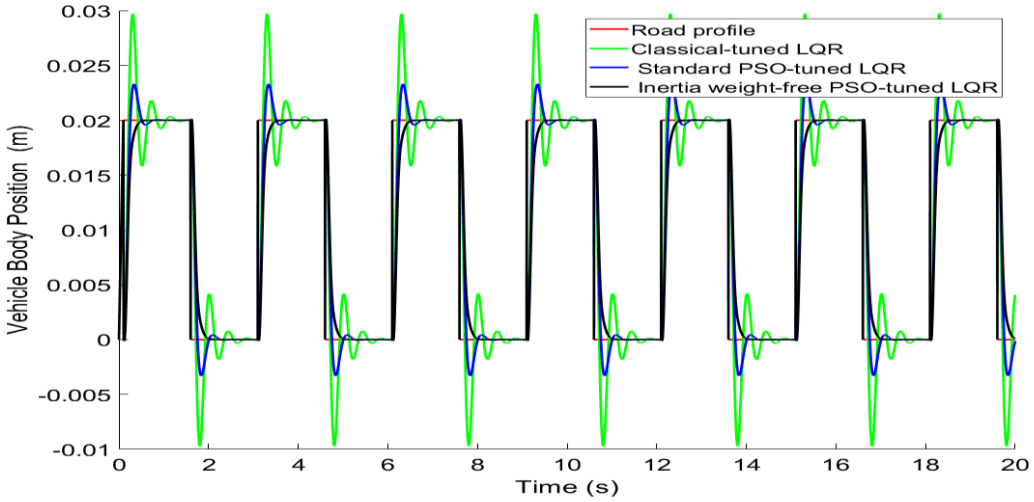

Figure 5. The vehicle body position under the square road disturbance

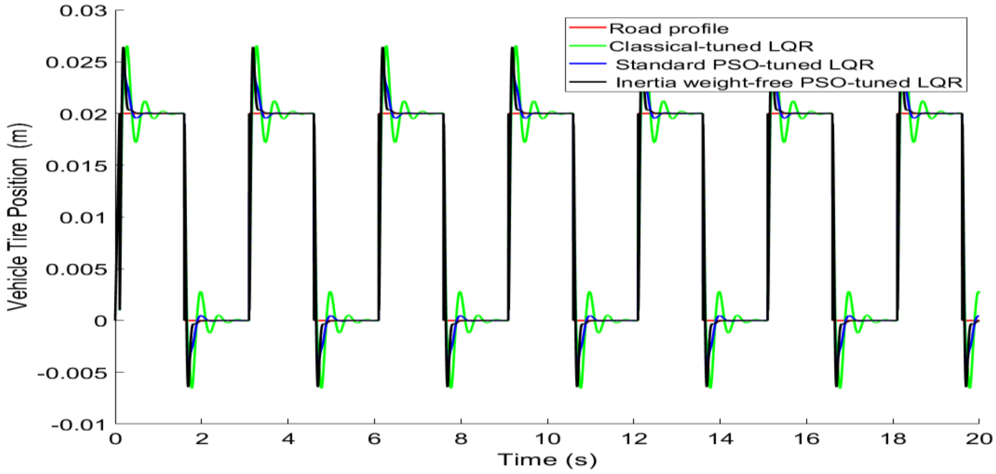

Figure 6. Vehicle tire position under the square road

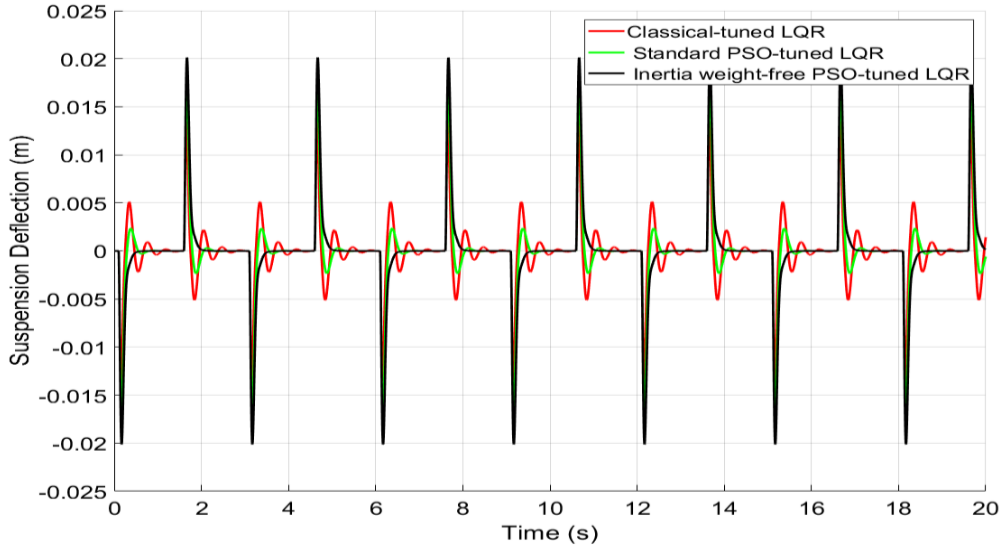

Figure 7. Suspension deflection under the square road

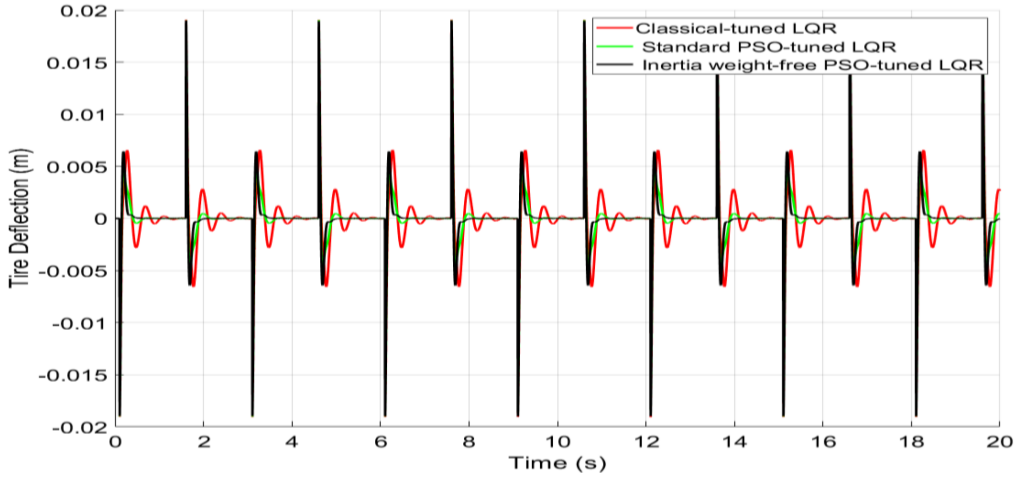

Figure 8. Tire deflection under the square road

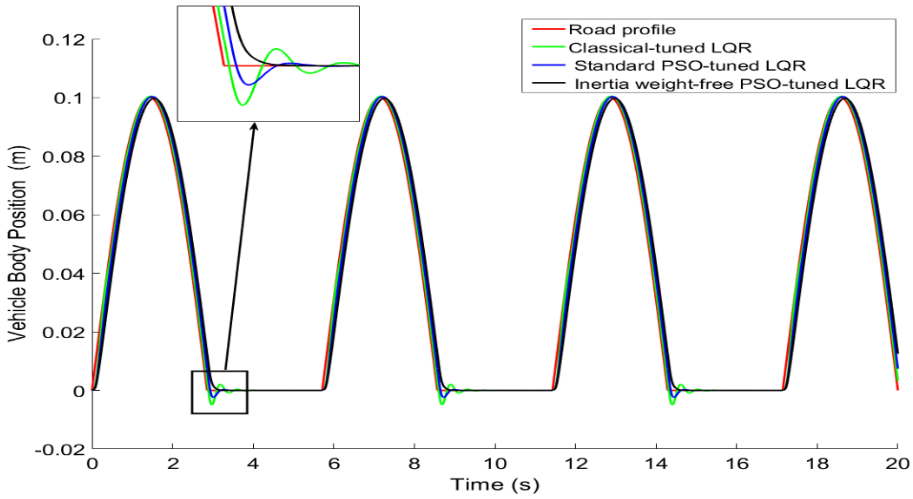

Figure 9. Vehicle body position under the bumpy road

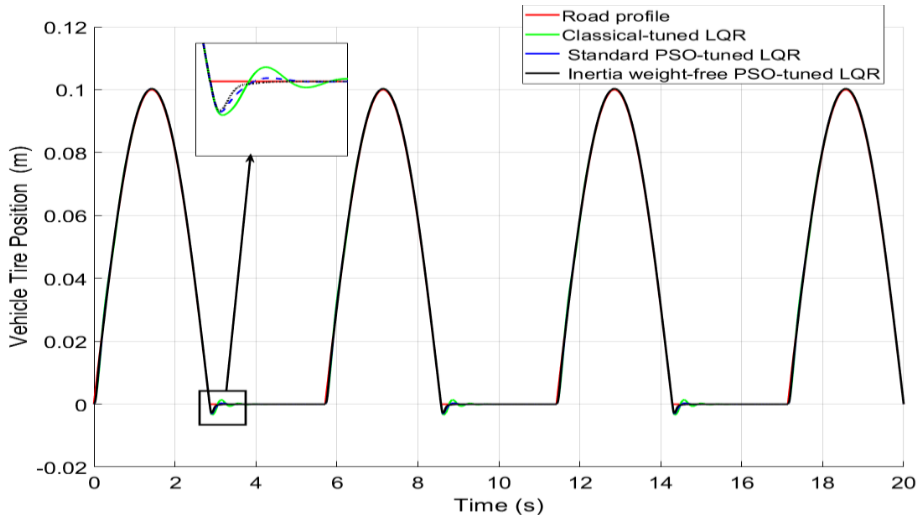

Figure 10. Tire position under the bumpy road

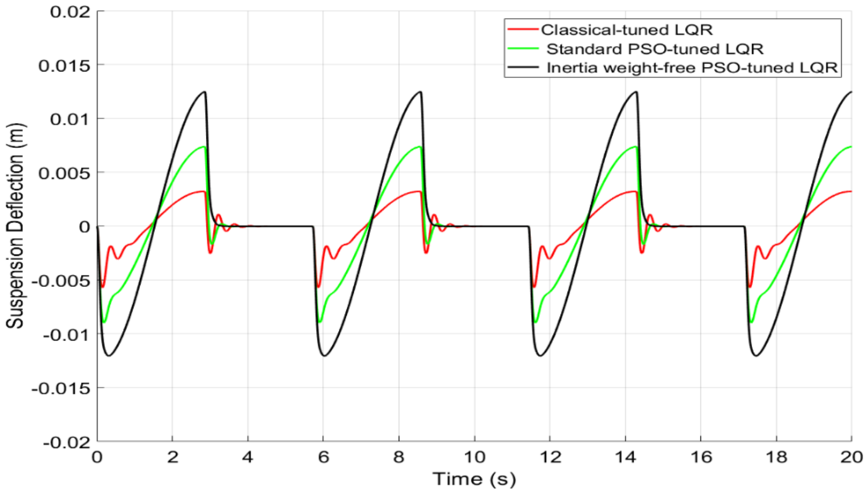

Figure 11. Suspension deflection under the bumpy road

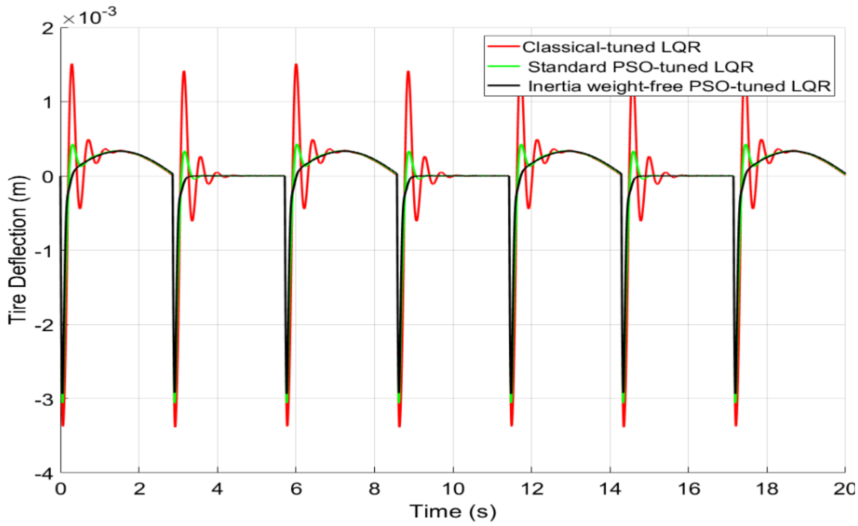

Figure 12. Tire deflection under the bumpy road

4. Conclusions

In this paper, the efficiency of a quarter-active suspension system has been investigated under different road profiles with the optimal control law. To improve ride quality, the inertia weight-free PSO-tuned LQR control law is proposed. Here, the inertia weight-free PSO finds the optimal weighting matrices of the LQR controller so as to satisfy conflicting performance requirements for the quarter-active suspension system. A comparative simulation study is carried out for inertia weight-free PSO-tuned LQR, the standard PSO-tuned LQR controller and the classical-tuned LQR controller. The simulation results demonstrate that the inertia weight-free PSO-tuned LQR controller provides well body and tire position tracking performances and reduces the vehicle body acceleration with the permissible suspension deflection.

Conflict of Interest Statement

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The study is complied with research and publication ethics

References

[1] V. S. Deshpande, B. Mohan, P. D. Shendge, and S. B. Phadke, "Disturbance observer based sliding mode control of active suspension systems," *J. Sound Vib.*, vol. 333, no. 11, pp. 2281–2296, 2014.

[2] I. Fialho and G. J. Balas, "Road adaptive active suspension design using linear parameter-varying gain-scheduling," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 1, pp. 43–54, 2002.

[3] X. Shao, F. Naghdy, and H. Du, "Reliable fuzzy H∞ control for active suspension of in-wheel motor driven electric vehicles with dynamic damping," *Mech. Syst. Signal Process.*, vol. 87, pp. 365–383, 2017.

[4] J. Cao, P. Li, and H. Liu, "An interval fuzzy controller for vehicle active suspension systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 885–895, 2010.

[5] A. R. Kaleli̇ and H. İ. Akolaş, "Aktif araç süspansiyon sistemi İçin makine Öğrenimi tabanlı kontrol sisteminin geliştirilmesi," *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, vol. 11, no. 2, pp. 421–428, 2022.

[6] M. P. Nagarkar, Y. J. Bhalerao, G. J. Vikhe Patil, and R. N. Zaware Patil, "GA-based multi-objective optimization of active nonlinear quarter car suspension system—PID and fuzzy logic control," *Int. J. Mech. Mater. Eng.*, vol. 13, no. 1, 2018.

[7] A. Unger, F. Schimmack, B. Lohmann, and R. Schwarz, "Application of LQ-based semi-active suspension control in a vehicle," *Control Eng. Pract.*, vol. 21, no. 12, pp. 1841–1850, 2013.

[8] S. Manna *et al.*, "Ant colony optimization tuned closed-loop optimal control intended for vehicle active suspension system," *IEEE Access*, vol. 10, pp. 53735–53745, 2022.

[9] J. S. David Reddipogu and V. K. Elumalai, "Hardware in the loop testing of adaptive inertia weight PSO-tuned LQR applied to vehicle suspension control," *J. Control Sci. Eng.*, vol. 2020, pp. 1–16, 2020.

[10] A. E. Bryson, "*Control of spacecraft and aircraft* (Vol. 41)". Princeton: Princeton university press, 1993.

[11] H. Maghfiroh, M. Nizam, M. Anwar, and A. Ma'Arif, "Improved LQR control using PSO optimization and Kalman filter estimator," *IEEE Access*, vol. 10, pp. 18330–18337, 2022.

[12] K. Hassani and W.-S. Lee, "Multi-objective design of state feedback controllers using reinforced quantum-behaved particle swarm optimization," *Appl. Soft Comput.*, vol. 41, pp. 66–76, 2016.

[13] E. Vinodh Kumar, G. S. Raaja, and J. Jerome, "Adaptive PSO for optimal LQR tracking control of 2 DoF laboratory helicopter," *Appl. Soft Comput.*, vol. 41, pp. 77–90, 2016.

[14] S. B. Karanki, M. K. Mishra, and B. K. Kumar, "Particle swarm optimization-based feedback controller for unified power-quality conditioner," *IEEE Trans. Power Deliv.*, vol. 25, no. 4, pp. 2814–2824, 2010.

[15] X. Shao, J. Zhang, and X. Zhang, "Takagi-Sugeno fuzzy modeling and PSO-based robust LQR anti-swing control for overhead crane," *Mathematical Problems in Engineering*, vol. *2019*, p. 14, 2019.

[16] T. Yuvapriya, P. Lakshmi, and V. K. Elumalai, "Experimental validation of LQR weight optimization using bat algorithm applied to vibration control of vehicle suspension system," *IETE J. Res.*, pp. 1–11, 2022.

[17] R. R. Das, V. K. Elumalai, R. Ganapathy Subramanian, and K. V. Ashok Kumar, "Adaptive predator–prey optimization for tuning of infinite horizon LQR applied to vehicle suspension system," *Appl. Soft Comput.*, vol. 72, pp. 518–526, 2018.

[18] Quanser, Active Suspension System: User Manual, Quanser Corporation, Ontario, Canada, 2009.

[19] Q. Chen, B. Liu, Q. Zhang, J. J. Liang, P. N. Suganthan, and B. Y. Qu, "Problem definitions and evaluation criteria for CEC 2015 special session on bound constrained single-objective computationally expensive numerical optimization," *Al-roomi.org*. [Online]. Available:https://alroomi.org/multimedia/CEC_Database/CEC2015/RealParameterOptimization/ExpensiveOptimization/CEC2015_ExpensiveOptimization_TechnicalReport.pdf. [Accessed: 05-Apr-2023].

[20] M. R. Tanweer, S. Suresh, and N. Sundararajan, "Improved SRPSO algorithm for solving CEC 2015 computationally expensive numerical optimization problems," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015.

[21] J. L. Rueda and I. Erlich, "MVMO for bound constrained single-objective computationally expensive numerical optimization," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015

[22] E. O. Wilson, *Sociobiology: The New Synthesis*, 25th ed. London, England: Harvard University Press, 2000.

[23] W. Zhang, *Selforganizology: The science of self-organization*. WORLD SCIENTIFIC, 2016.

[24] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” *Comput. Graph. (ACM)*, vol. 21, no. 4, pp. 25–34, 1987.

[25] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, 2002.

[26] W. J. Zhang, “Particle swarm optimization: A Matlab algorithm,” *Iaees.org*. [Online]. Available:http://www.iaees.org/publications/journals/selforganizology/articles/2022-9(3-4)/particle-swarm-optimization-Matlab-algorithm.pdf. [Accessed: 05-Apr-2023].

[27] C. Blum, M. J. B. Aguilera, A. Roli, and M. Sampels, Eds., *Hybrid metaheuristics: An emerging approach to optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[28] I. H. Osman and G. Laporte, “Metaheuristics: A bibliography,” *Ann. Oper. Res.*, vol. 63, no. 5, pp. 511–623, 1996.

[29] M. Jaberipour, E. Khorram, and B. Karimi, “Particle swarm algorithm for solving systems of nonlinear equations,” *Comput. Math. Appl.*, vol. 62, no. 2, pp. 566–576, 2011.