A Novel Sequential Switching Quadratic Particle Swarm Optimization Scheme with Applications to Fast Tuning of PID Controllers

Yuqiang Luo[a,b], Zidong Wang[c,d], Hongli Dong[e], Jingfeng Mao[f], Fuad E. Alsaadi[g]

[a]*Informatization Office, University of Shanghai for Science and Technology, Shanghai 200093, China.*
[b]*Shanghai Key Lab of Modern Optical System, Department of Control Science and Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China.*
[c]*College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China.*
[d]*Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom.*
[e]*Artificial Intelligence Energy Research Institute, Northeast Petroleum University, Daqing 163318, China.*
[f]*School of Electrical Engineering, Nantong University, Nantong 226019, China.*
[g]*Communication Systems and Networks Research Group, Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia.*

Abstract

In this work, a sequential switching quadratic particle swarm optimization (SSQPSO) scheme is investigated, where the velocity update mechanism is improved to enhance the convergence performance. Considering the sequential characteristics (related to evolution factors) of the evolution process, a Markov chain with special probability transition matrix is employed to characterize the switching of evolution state. With the help of the mean distance, the concept of population density is first put forward in the dynamic search region enclosed by all particles. Then, taking into account the change of the population density in different generations, two quadratic acceleration terms are introduced into the velocity update model based on the Hadamard product, where four evolution-state-dependent acceleration coefficients are also adopted. The positivity or negativity of the quadratic acceleration terms is retained by resorting to the matrix sign functions. Several widely utilized benchmark functions (including two unimodal and multimodal functions) are employed to evaluate the search capability of the studied SSQPSO scheme. The experimental consequences illustrate that the performance of the developed SSQPSO scheme is superior to that of some popular particle swarm optimization (PSO) schemes. To further demonstrate the effectiveness in practical engineering, the addressed SSQPSO scheme is successfully applied to achieve the fast parameter tuning of the proportional-integral-derivative controller in a spring-mass-damper system.

Keywords: Particle swarm optimization; Markov chain; sequential switching; quadratic acceleration; proportional-integral-derivative controller; parameter tuning.

Email addresses: `abel_luo_me@163.com` (Yuqiang Luo), `Zidong.Wang@brunel.ac.uk` (Zidong Wang), `shiningdhl@vip.126.com` (Hongli Dong), `mao.jf@ntu.edu.cn` (Jingfeng Mao), `falsaadi@kau.edu.sa` (Fuad E. Alsaadi)

1. Introduction

The past few decades have witnessed increasing research interest devoted to the nonlinear optimization problem from many communities such as control, economics, aerospace, computer science, and so forth [44, 18, 37, 43, 42, 25, 19, 41, 26]. As is well known, the conventional optimization schemes (e.g. gradient descent method, Newton method, and conjugate gradient method) are usually time-consuming, especially for the high-dimensional nonlinear constrained problems [40, 34, 33, 48, 20]. To this end, the evolutionary computation, inspired by the process of biological evolution, has stood out as a powerful family of optimization approaches [6, 22, 11]. Some representative evolutionary computation algorithms include, but are not limited to, genetic algorithms (GAs), evolution strategies, and evolutionary programming. Particularly, the PSO scheme, first put forward in [12], is well known to be a stochastic iterative and population-based heuristic approach.

Owing primarily to the distinctive advantages such as easy implementation, satisfactory performance, and guaranteed diversity, the PSO algorithm has attracted increasing research attention in recent years, see e.g. [39, 15, 35, 36, 29, 16, 2, 38] and the references therein. Nevertheless, as pointed out in [46], the PSO scheme suffers from the problems of computational inefficiency (with respect to the required quantity of function evaluations) and trapping local optima especially in multimodal problems. Therefore, great effort has been made to develop new PSO variants that are able to guarantee an acceptable convergence rate and avoid the possible local optimum, such as the randomly occurring distributedly delayed PSO [14] and the switching delayed PSO [26].

Generally speaking, the convergence speed of PSO is dependent on various factors such as population initialization and parameter selection. For example, an adaptive PSO with an improved search speed has been proposed in [46] by adaptively controlling the acceleration coefficients and the inertia weight. It is noticeable that in the standard PSO, the movement speed of each individual particle is directly related to its distances to two positions, i.e., the local best-position achieved through the individual itself and the global best-position discovered through a specified neighborhood. Obviously, the speed adjustment based on the distance between the individual and the best positions might be helpful for the convergence acceleration. Nevertheless, such a problem has not acquired enough research attention yet, and this stimulates the current research.

It has been commonly recognized that identifying the states of evolutionary process can accelerate search course and improve the efficiency of evolutionary algorithms. In this regard, the concept of evolution state has been introduced in [47, 45]. To be specific, four training states (i.e. initial state, submaturing state, maturing state, and matured state) have been considered in [47] to adaptively adjust the probabilities of the operations of crossover and mutation in GA. In [45], the same four states have been utilized to control the acceleration coefficients of PSO in an adaptive manner. Considering that the variation of the population distribution of PSO depends on not just the number of generation but the evolution state as well [46], it makes sense to determine the evolution state by using the information of population distribution. Accordingly, a systematic estimation scheme has been developed in [46] for four evolution states (i.e., exploration, exploitation, convergence, and jumping-out) by resorting to the clustering analysis and fuzzy classification techniques. Recently, the switching PSO (SPSO) variants have been reported in [31, 17, 26] based on the evolution factor.

Over decades, as a result of the simple structure and the robustness to disturbances, the proportional-integral-derivative (PID) control technique has found successful applications in a wide range of realms such as process industry, cruise control, and servo system [21]. The main mechanism of the so-called PID control is to achieve the desired performance of control system by adjusting the parameters of the proportional loop, integral loop, and derivative loop according to their respective practical meanings. Accordingly, the parameter tuning has been playing a paramount role in the implementations of PID controller. Nevertheless, this is a nontrivial task in reality due to the following two challenges: 1) there usually exist certain conflicts between the system characteristics and application requirements; and 2) it is often the case that multiple indexes (e.g. short transient and high stability) need to be simultaneously satisfied. Consequently, many approaches have been developed for the parameter tuning, such as the Ziegler-Nichols method [32], Cohen-Coon method [9], and Tyreus-Luyben technique [23]. However, these schemes are computationally expensive, especially for the system with a long loop time. As such, it is imperative to develop an online and automatic tuning method to improve the performance of the PID design, which leads to another motivation of this paper. Fortunately, the PSO-based method is capable of meeting the demand of online computation without requiring detailed information about the system parameters.

Based upon the aforementioned analysis, our primary objective is to design a sequential switching quadratic particle swarm optimization (SSQPSO) algorithm with practical applications on the parameter tuning of PID controllers. The main novelties of our paper can be summarized as follows: 1) by resorting to a special transition probability matrix, a Markov chain jumping in a finite state space is introduced to characterize the sequential switching process of evolution states; 2) the concept of population density is proposed based on the mean distance and the evolution factor, and a new velocity update strategy is developed to improve the evolutionary speed; 3) by means of the sign function and Hadamard product, a novel SSQPSO algorithm is constructed with enhanced convergency speed and accuracy; and 4) the developed SSQPSO scheme is successfully employed to achieve the fast parameter tuning of PID controllers for a spring-mass-damper mechanical system.

The rest organization of this work is outlined as follows. In Section 2, the standard PSO scheme and its some well-known variants are reviewed. In Section 3, based on the evolution states and population density, a novel SSQPSO algorithm is put forward with an improved velocity update mechanism. Some benchmark tests and performance comparisons with other PSO algorithms are given in Section 4. Section 5 provides a case study on the parameter tuning of PID controller. Some concluding remarks are drawn finally in Section 6.

2. Standard PSO Algorithm and Its Variants

2.1. Standard PSO Algorithm

Inspired by the swarm behaviors of bird flocking [5] and fish schooling [10], the standard PSO algorithm turns out to be an effective way of finding the globally optimal solution with simple mathematical operations and inexpensive computation cost [49]. In the standard PSO scheme, a swarm composed of $\mathscr{P}$ individuals (also called particles) moves with varying speed in a given D-dimensional hyperspace (also called solution space) to search the global optimum for the object function (also known as fitness function), and each particle in the swarm is deemed as a candidate solution of the concerned optimization problem [46, 31]. Specifically, as the first step, each particle is randomly initialized within the specified region, and then they adjust their positions toward the global optimum according to the individual experience and the real-time interaction among particles [4, 12]. In the course of evolution, there are two kinds of particle positions that need to be stored in each iteration. The first position, denoted by $pBest_i$, is the best position encountered so far by the ith individual, and the second one, denoted by $nBest$, is the best position in a specified neighborhood with n particles. In particular, $nBest$ is said to be $gBest$ if $n = \mathscr{P}$ and $lBest$ if $n \leq \mathscr{P}$, where g and l are, respectively, the abbreviations of "global" and "local". The corresponding PSO is referred to as the local-version PSO (LPSO) (or the global-version PSO (GPSO)) when $gBest$ (or $lBest$) is used [4, 46, 27].

In the standard PSO scheme, the velocity and position of the ith individual at the $(k+1)$th iteration are governed by the following model:

$$
\begin{cases}
v_i(k+1) = wv_i(k) + c_1 r_1 (pBest_i(k) - x_i(k)) \\
\qquad\qquad + c_2 r_2 (nBest(k) - x_i(k)), & \text{(1a)} \\
x_i(k+1) = x_i(k) + v_i(k+1), & \text{(1b)}
\end{cases}
$$

where $i \in \mathbb{P} = \{1, 2, \cdots, \mathscr{P}\}$, $k \in \mathbb{Z}^+$ denotes the current iteration number with $\mathbb{Z}^+$ being the set of all nonnegative integers. $w \in \mathbb{R}$ denotes the inertia weight. The constants c_1 and c_2 stand for acceleration coefficients. r_1 and r_2 denote two uniformly distributed random quantities sampled from $[0, 1]$. $\boldsymbol{v}_i(k) = [v_{i1}(k), v_{i2}(k), \cdots, v_{iD}(k)]^T \in \mathbb{R}^D$ is the velocity vector of the ith individual and the jth component $v_{ij}(k) \in [0, v_{i\max}(k)]$, where $v_{i\max}(k)$ is the specified maximum velocity. $\boldsymbol{x}_i(k) = [x_{i1}(k), x_{i2}(k), \cdots, x_{iD}(k)]^T \in \mathbb{R}^D$ is the position vector of the ith individual and the jth component $x_{ij}(k) \in [x_{\min,j}, x_{\max,j}]$, where $x_{\min,j} \in \mathbb{R}$ and $x_{\max,j} \in \mathbb{R}$ are, respectively, the lower and upper bounds of the jth-dimensional search/solution region [31]. For convenience, the flowchart of the standard PSO algorithm is shown in Fig. 1.

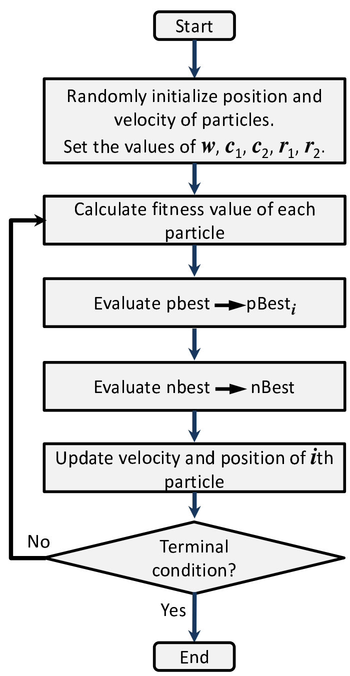


Figure 1: Flowchart of standard PSO algorithm.

2.2. History and Variants of PSO Algorithm

Ever since the pioneering work by Kennedy and Eberhart in 1995 [12], many variants of PSO scheme have been proposed in the literature to improve the performance from a variety of perspectives. For example, in order to improve the search course and control the exploration capacity

[1], the maximum velocity of the individuals is limited by Eberhart and Kennedy in 1995 [4]. In 1998, the parameter of inertia weight w has been introduced by Shi and Eberhart [27] to control the velocity of particles, which can regulate the ability of exploration and exploitation and gives rise to the standard PSO algorithm. As pointed out in [1, 27], a large (or small) inertia weight would strengthen the global (or local) search capability. Since the inertia weight has a crucial impact on the balance between exploration and exploitation, the corresponding dynamic regulation has become an important direction to improve the search power of PSO approach. Up to now, many regulation mechanisms on inertia weight have been put forward in the literature, such as the adaptive strategy [46], random strategy, linearly decreasing strategy, and exponentially decreasing strategy [1]. In particular, the PSO approach with a linearly decreasing inertia weight (PSO-LDIW) has been presented in [28, 27]. Instead of utilizing the inertia weight, the PSO with constriction factors (PSO-CK) has been investigated in [3] to assure the convergence. On the other hand, some variants have focused on adjusting the acceleration coefficients. For example, the PSO scheme with time-varying acceleration coefficients (PSO-TVAC) has been studied in [24] to achieve the efficient control on the local search and convergence to the global optimum.

3. A New SSQPSO Algorithm

In this section, by analyzing the evolution states of PSO, an innovative SSQPSO scheme is offered to further improve the evolution process and the search performance. The main novelties of the concerned SSQPSO algorithm are twofold: 1) a special transition probability matrix is developed to characterize the sequential evolution process, which can effectively describe the real features of the evolution behaviors of particles; and 2) two quadratic acceleration terms are introduced in the velocity update equation, which is capable of accelerating the convergence of the searching process.

3.1. Evolution States

For the purpose of improving the search efficiency, the GPSO algorithm with the evolution state estimation strategy has been investigated in [46], where the evolution state is determined by utilizing the information of population distribution. In fact, the information of population distribution can be evaluated based on the average distance of the ith individual to all other particles with the following Euclidian metric:

$$d_i = (\mathscr{P} - 1)^{-1} \sum_{l=1, l \neq i}^{\mathscr{P}} \left(\sum_{j=1}^{D} (x_{ij} - x_{lj})^2 \right)^{\frac{1}{2}}. \tag{2}$$

Then, the evolution state of GPSO can be clearly exposed by checking the dynamic trajectories of the so-called evolution factor

$$f_E = \frac{d_{\mathrm{g}} - d_{\min}}{d_{\max} - d_{\min}}, \tag{3}$$

where d_{g} is the average distance of the global best individual to all other particles in the population. $d_{\min}$ and $d_{\max}$ are, respectively, the minimum and maximum of d_i. According to the value of f_E, the evolution process of GPSO can be divided into four types of states (i.e., exploration, exploitation, convergence and jumping-out), which is in essence sequential and periodical. In this paper, instead of using the fuzzy approach in [46], a simple yet effective method for the classification of evolution states is adopted as follows:

$$\text{Evolution State} = \begin{cases} \text{Exploration}, & 0.5 < f_E \leq 0.7; \\ \text{Exploitation}, & 0.1 < f_E \leq 0.5; \\ \text{Convergence}, & 0 < f_E \leq 0.1; \\ \text{Jumping-out}, & 0.7 < f_E \leq 1. \end{cases} \tag{4}$$

The evolution state information reflected by the evolution factor further reveals the periodic and sequential switching characteristics of the evolution state [46].

In order to characterize the switching process of the four evolution states (as illustrated in Fig. 2), the exploration, exploration, convergence and jumping-out state are, respectively, denoted as S_1, S_2, S_3 and S_4. Let $\theta(k)$ be a Markov chain jumping in a finite space $\mathbb{S} = \{1, 2, 3, 4\}$ with the switching probability assigned by

$$\text{Prob}\{\theta(k+1) = r | \theta(k) = s\} = \pi_{sr}, \quad r, s \in \mathbb{S} \tag{5}$$

where "1", "2", "3" and "4" represent, respectively, the four evolution states S_1, S_2, S_3 and S_4. Then, the probability transition matrix is the following:

$$\mathscr{T} = \begin{bmatrix} \pi_{11} & 1-\pi_{11} & 0 & 0 \\ 0 & \pi_{22} & 1-\pi_{22} & 0 \\ 0 & 0 & \pi_{33} & 1-\pi_{33} \\ 1-\pi_{44} & 0 & 0 & \pi_{44} \end{bmatrix}.$$

Different from the descriptions in [31, 17], in this paper, only two nonzero elements exist in each row of the probability transition matrix, thereby better reflecting the engineering practice.

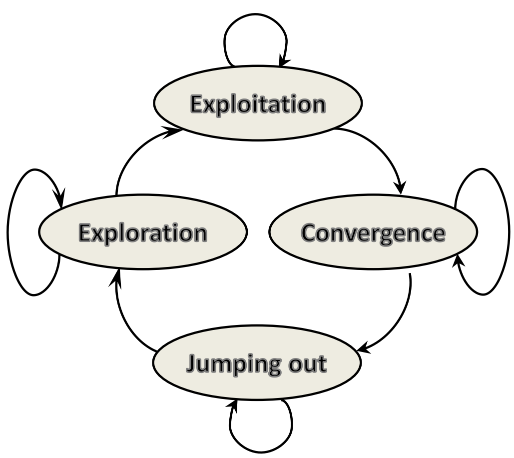


Figure 2: Switching path of the four evolution states.

Remark 1. *It is worth pointing out that the Markov chain is not the only model to characterize the switching process of the evolution states. For example, let* $\eta_i(k)$ $(i = 1, 2, 3, 4)$ *be a set of random white sequences which are mutually independent and take values of* 0 *or* 1*, and*

$$\text{Prob}\{\eta_i(k) = 1\} = \alpha_i,\ \text{Prob}\{\eta_i(k) = 0\} = 1 - \alpha_i,$$

where $\alpha_i \in [0, 1]$ *are given scalars. Then, the state switching process can be characterized as follows:*

$$\eta_1(k) = \begin{cases} 1\text{: switching to } S_1 \text{ from } S_4; \\ 0\text{: no switching.} \end{cases}$$

$$\eta_2(k) = \begin{cases} 1\text{: switching to } S_2 \text{ from } S_1; \\ 0\text{: no switching.} \end{cases}$$

$$\eta_3(k) = \begin{cases} 1\text{: switching to } S_3 \text{ from } S_2; \\ 0\text{: no switching.} \end{cases}$$

$$\eta_4(k) = \begin{cases} 1\text{: switching to } S_4 \text{ from } S_3; \\ 0\text{: no switching.} \end{cases}$$

3.2. Improved Velocity Update Model

As pointed out in [27], the particles in the PSO algorithm evolve based on the cooperation and competition with the individuals themselves. Generally speaking, the particles are randomly distributed within the search area at the initial stage, and gradually cluster to the global optimum through generations. In this sense, it is clear that the population distribution would change as the iteration proceeds. As an example, the aggregation process of particles in the optimum search of the Sphere function ($D = 3$) is visualized in Fig. 3, where the population distributions at three different generations are shown.

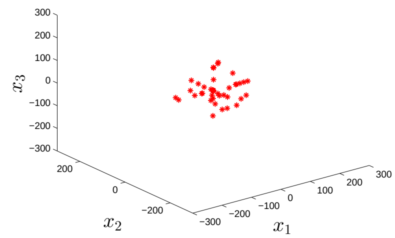
In order to further characterize the aggregation of particles, the following population density D_s is defined in the dynamic search region where all particles are enclosed

$$D_s = \frac{\mathscr{P}}{\prod_{j=1}^{3} \max_{i \in \mathbb{P}, i \neq l} |x_{ij} - x_{lj}|}. \tag{7}$$

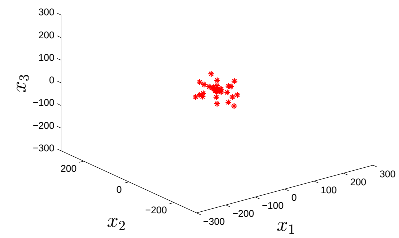
Note that the trajectory of the population density is not fully monotonic. As shown in Fig. 4, there exists a drastic oscillation at a later stage of the iterations. In fact, such an oscillation occurs at the fine-searching stage.

It should be pointed out that a fine-searching in the case of sparse population density at the incipient stage is superfluous and even harmful to the global search speed. For the purpose of expediting the convergence speed of the PSO search, it is suggested to let the particle with large d_i stride with a high speed. Meanwhile, it makes practical sense to adopt a fine-searching with a slow speed at the later stage. Unfortunately, the aforementioned search mechanism is not considered in the standard model (1a)-(1b). As such, in this paper, we introduce the following speed strategy:

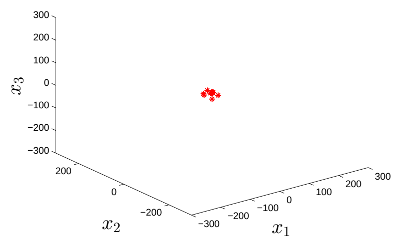
$$\text{Particle speed} \begin{cases} \mathit{Fast}, & d_i \text{ is long;} \\ \mathit{Low}, & d_i \text{ is short.} \end{cases} \tag{8}$$



(a) Generation = 35



(b) Generation = 44



(c) Generation = 55

Figure 3: Population distributions at different generations.

Accordingly, the velocity and position in our new SSQPSO scheme are dynamically governed as follows:

$$
\begin{cases}
v_i(k+1) = wv_i(k) + c_1(\theta(k))r_1(pBest_i(k) - x_i(k)) \\
\qquad\qquad +c_2(\theta(k))r_2(gBest(k) - x_i(k)) \\
\qquad\qquad +c_3(\theta(k))r_3\mathfrak{S}_3(k)(pBest_i(k) - x_i(k)) \\
\qquad\qquad \circ(pBest_i(k) - x_i(k)) \\
\qquad\qquad +c_4(\theta(k))r_4\mathfrak{S}_4(k)(gBest(k) - x_i(k)) \\
\qquad\qquad \circ(gBest(k) - x_i(k)), & \text{(9a)} \\
x_i(k+1) = x_i(k) + v_i(k+1), & \text{(9b)}
\end{cases}
$$

where r_1, r_2, r_3 and r_4 are random numbers of uniform distribution which are independently sampled from $[0,1]$. $c_1(\theta(k))$, $c_2(\theta(k))$, $c_3(\theta(k))$ and $c_4(\theta(k))$ denote evolution-state-dependent acceleration coefficients. $\mathfrak{S}_3(k) = \mathrm{diag}(\mathrm{sign}(pBest_i(k) - x_i(k)))$ and $\mathfrak{S}_4(k) = \mathrm{diag}(\mathrm{sign}(gBest(k) - x_i(k)))$ are matrix sign functions, where $\mathrm{diag}(x)$ denotes a square matrix with elements of n-dimensional vector x placed on the main diagonal, and $\circ$ denotes the Hadamard product of matrices.

Remark 2. *What is noteworthy is that the particle individuals would gradually gather together during the evolutionary process, which means that the population density D_s will gradually increase in general. Considering such a trend, it is reasonable to execute a coarse-searching with high particle speed in the case of low density, and a fine-searching with slow particle speed in the case of large*

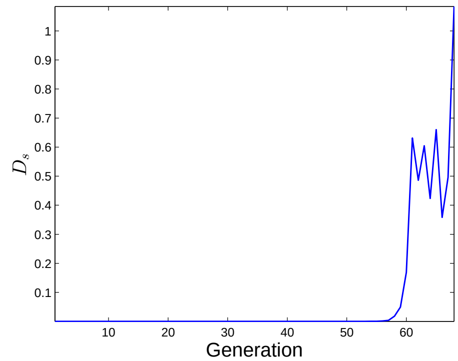
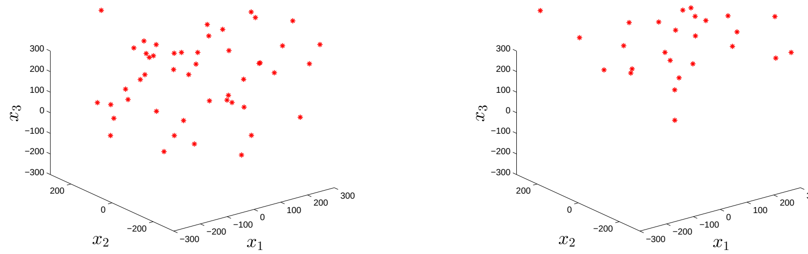


Figure 4: Population density in generations.

population density. From the standard PSO model (1a)-(1b), it is obvious that the velocity update heavily relies on two acceleration terms, i.e., $(pBest_i(k) - x_i(k))$ *and* $(gBest(k) - x_i(k))$*, where the former relates to the distance between the* i*th particle and the historically individual best position, while the latter is the distance between the* i*th particle and the historically global best position. By further analyzing the velocity update model, we see that a quadratic manipulation is positively associated to the mean distance* d_i*, which motivates us to put forward the so-called quadratic PSO algorithm realized by Hadamard products.*

Remark 3. *Considering the varying population density in different generations, the Hadamard product operations are introduced in (9) to ensure the efficiency of the quadratic acceleration. Unfortunately, the quadratic acceleration terms would inevitably result in the absolutely positive shifts at all the dimensions, which may cause deviations from the optimal solution and even ceaseless oscillations. As an example, Fig. 5 shows the population distribution during the search process of the Sphere function* ($D = 3$) *in the absence of* $\mathfrak{S}_3(k)$ *and* $\mathfrak{S}_4(k)$ *in (9), which is obviously nonconvergent. Therefore, it is necessary to take into account the positivity or negativity of* $(pBest_i(k)-x_i(k))$ *and* $(gBest(k)-x_i(k))$ *in the quadratic terms for convergence assurance of the established SSQPSO scheme. As such, we construct the matrix sign functions* $\mathfrak{S}_3(k)$ *and* $\mathfrak{S}_4(k)$ *in (9). In the update rule (9a), the two quadratic terms realized by Hadamard product will, respectively, enforce 1) the coarse-searching with high particle speed in case of low density and 2) a fine-searching with slow particle speed in case of large population density. Note that one needs to be careful about the use of the Hadamard product because more Hadamard products may lead to more computational load which, in turn, cause undesired oscillations in the fine-searching stage.*

As pointed in [31], it is reasonable to consider the dynamic adjustment of acceleration coefficients according to the evolution process. To this end, following the similar line of [31], the values of $c_1(\theta(k))$, $c_2(\theta(k))$, $c_3(\theta(k))$ and $c_4(\theta(k))$ corresponding to $\theta(k)$ are selected in Table 1.



(a) Generation = 27

(b) Generation = 63

Figure 5: Population distributions at different generations without $\mathfrak{S}_3(k)$ and $\mathfrak{S}_4(k)$.

Table 1: Values of acceleration coefficients corresponding to $\theta(k)$

Evolution State	$\theta(k)$	$c_1(\theta(k))$	$c_2(\theta(k))$	$c_3(\theta(k))$	$c_4(\theta(k))$
Initiation	4	2	2	2	2
Exploration	1	2.2	1.8	2.3	1.7
Exploration	2	2.1	1.9	2.0	1.9
Jumping-out	3	1.8	2.2	1.7	2.3
Convergence	4	2	2	2	2

4. Benchmark Tests

In this section, several widely used benchmark functions are chosen to test the efficacy of the processed SSQPSO scheme, and the comparisons of different PSO algorithms are also provided.

4.1. Benchmark Functions and Test Configuration

In this work, we consider nine benchmark functions composed of both unimodal and multimodal cases for testing and comparison. Specifically, six benchmark functions are employed for the unimodal case. In the meantime, three benchmark functions are selected for the multimodal case, including the functions named Ackley, Griewank, and Rastrigin. It is worth mentioning that all the chosen benchmark functions have been extensively adopted in the community of evolutionary computation [8]. The detailed information on these benchmark functions is available in Table 2, and other relevant parameters are configured in Table 3. Moreover, the optimization process is terminated when the obtained fitness value is below a threshold, and 30 independently repeated tests are conducted to acquire mean fitness of each benchmark function.

4.2. Testing Analysis

For the purpose of verifying the superiority of the interested search capability, the efficacy of the addressed SSQPSO scheme is compared to that of five popular PSO algorithms including the APSO, PSO-LDIW, PSO-CK, PSO-TVAC and SPSO. The mean fitness values of these six PSO algorithms in every iteration are shown in Figs. 6-12, and the statistical information (e.g. the best fitness value and the standard deviation) is summarized in Table 4. It is clear that the proposed SSQPSO algorithm exhibits superiority over the other five modified PSO approaches for both unimodal and multimodal problems. It should be emphasized that although the optimality of the SSQPSO algorithm on the Rosenbrock function is inferior to other PSO algorithms, the proposed algorithm has fast search speed at the preliminary stage (see Fig. 11). Additionally, the optimal search in the SSQPSO algorithm is capable of reaching the threshold of the unimodal function Schwefel 1.2 at iteration step $k = 2680$, while other PSO methods do not terminate until the maximum iteration $k = 3000$ is achieved.

According to the performance ranking of the derivative-free optimizers in [13], the PSO algorithm is competitive to many direct search methods (e.g. simplex search, pattern search, and Powell conjugate search) and heuristic schemes (e.g. simulated annealing and GA). Nevertheless, we can find from Table 4 that the proposed SSQPSO algorithm exhibits a fast convergence speed but a poor performance in terms of the optimality and accuracy defined in [13]. In fact, this is caused by the introduced quadratic acceleration terms in the velocity update equation (9a).

5. Case Study: Parameter Tuning of PID Controller

In this part, the investigated SSQPSO scheme is applied in the parameter tuning problem of the PID controller to demonstrate its effectiveness in practical engineering.

Consider a PID control scenario for a spring-mass-damper mechanical system as shown in Fig. 13 [7], where M is the mass of the free body, κ is the elastic coefficient, c is the damping coefficient, and $f(t)$ is the external force. According to Newton's second law, the motion equation is established as follows:

$$f(t) - \kappa y - c\dot{y}(t) = M\ddot{y}(t). \tag{10}$$

Define $y(t)$ and $\dfrac{dy(t)}{dt}$ by q_1 and q_2, respectively. Then, we can rewrite (10) as

$$\begin{cases} \ddot{q}_2(t) = -\dfrac{cq_2}{M} - \dfrac{\kappa q_1}{M} + \dfrac{f(t)}{M}, \\ \dot{q}_1 = q_2. \end{cases}$$

Subsequently, the state-space model is readily formulated as

$$\begin{cases} \dot{q}(t) = \mathscr{A} q(t) + \mathscr{B} u(t), \\ y(t) = \mathscr{C} q(t), \end{cases}$$

where

$$q(t) = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}, \ u(t) = f(t), \ \mathscr{A} = \begin{bmatrix} 0 & 1 \\ -\kappa/M & -c/M \end{bmatrix},$$
$$\mathscr{B} = \begin{bmatrix} 0 \\ 1/M \end{bmatrix}, \ \mathscr{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

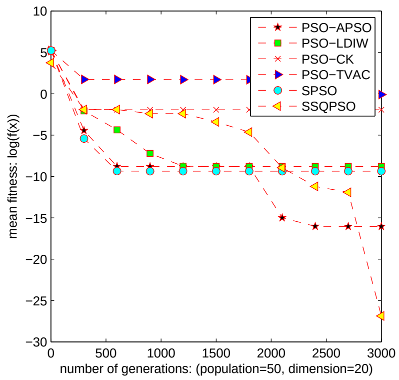


Figure 6: Efficacy comparisons of PSO scheme for 10-D f_1.

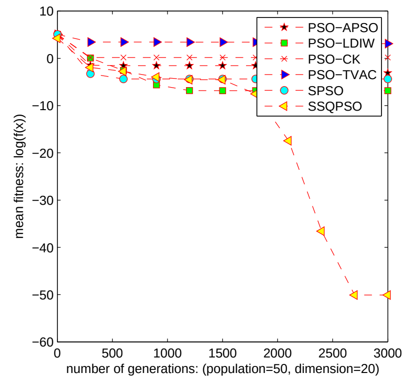


Figure 7: Efficacy comparisons of PSO scheme for 10-D f_2.

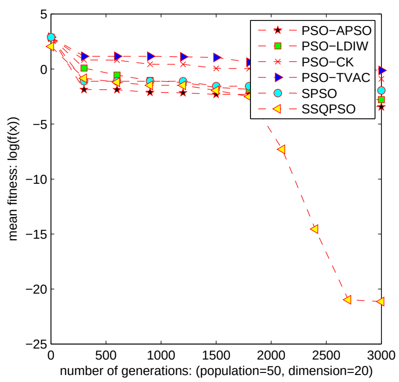


Figure 8: Efficacy comparisons of PSO scheme for 10-D f_3.

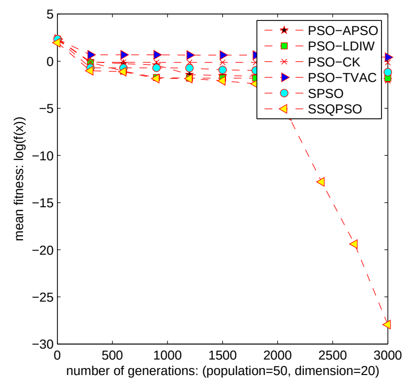


Figure 9: Efficacy comparisons of PSO for 10-D f_4.

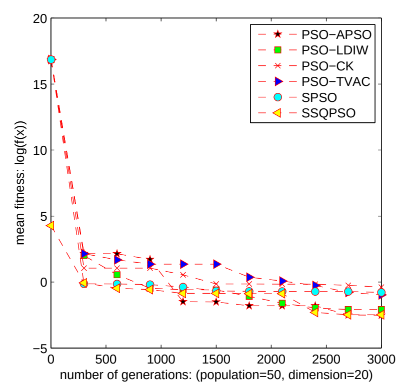


Figure 10: Efficacy comparisons of PSO scheme for 10-D f_5.

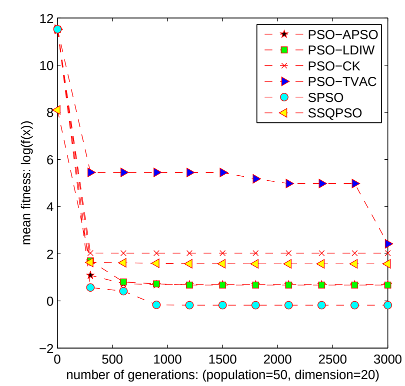


Figure 11: Efficacy comparisons of PSO scheme for 10-D f_6.

Table 2: Benchmark functions

	Function Name	No.	Separable	Convex	Minimum
Unimodal	Sphere	f_1	Yes	Yes	$f(0,\cdots,0)=0$
	Schwefel 1.2 [30]	f_2	Not	Yes	$f(0,\cdots,0)=0$
	Schwefel 2.20	f_3	Yes	Yes	$f(0,\cdots,0)=0$
	Schwefel 2.21	f_4	Yes	Yes	$f(0,\cdots,0)=0$
	Schwefel 2.22	f_5	Not	Yes	$f(0,\cdots,0)=0$
	Rosenbrock	f_6	Not	Not	$f(1,\cdots,1)=0$
Multimodal	Ackley	f_7	Not	Not	$f(0,\cdots,0)=0$
	Griewank	f_8	Not	Not	$f(0,\cdots,0)=0$
	Rastrigin	f_9	Yes	Not	$f(0,\cdots,0)=0$

Function Expression

$$f_1(\mathbf{x}) = f(x_1,\cdots,x_D) = \sum_{j=1}^{D} x_j^2,$$

$$f_2(\mathbf{x}) = f(x_1,\cdots,x_D) = \sum_{j=1}^{D}\left(\sum_{i=1}^{j} x_i\right)^2,$$

$$f_3(\mathbf{x}) = f(x_1,\cdots,x_D) = \sum_{j=1}^{D} |x_j|,$$

$$f_4(\mathbf{x}) = f(x_1,\cdots,x_D) = \max_{j=1,\cdots,D} |x_j|,$$

$$f_5(\mathbf{x}) = f(x_1,\cdots,x_D) = \prod_{j=1}^{D} |x_j| + \sum_{j=1}^{D} |x_j|,$$

$$f_6(\mathbf{x}) = f(x_1,\cdots,x_D) = \sum_{j=1}^{D-1}\left(100\Big(x_{j+1}-x_j^2\Big)^2 + (x_j-1)^2\right),$$

$$f_7(\mathbf{x}) = f(x_1,\cdots,x_D) = -20e^{\left(-0.2\sqrt{\frac{1}{D}\sum_{j=1}^{D} x_j^2}\right)} - e^{\left(\frac{1}{D}\sum_{j=1}^{D}\cos(2\pi x_j)\right)} + 20 + e^{(1)},$$

$$f_8(\mathbf{x}) = f(x_1,\cdots,x_D) = 1 + \frac{1}{4000}\sum_{j=1}^{D} x_j^2 - \prod_{j=1}^{D}\cos\left(\frac{x_j}{\sqrt{j}}\right),$$

$$f_9(\mathbf{x}) = f(x_1,\cdots,x_D) = 10D + \sum_{j=1}^{D}\left(x_j^2 - 10\cos(2\pi x_j)\right).$$

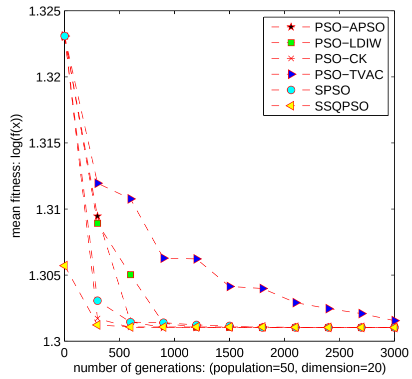


Figure 12: Efficacy comparisons of PSO scheme for 10-D f_7.

Table 3: Test configuration

Algorithm	Search Region (x_j)	$v_{i\max}$	Population Size ($\mathscr{P}$)	Dimension (D)	Maximum Iteration	Threshold	Test Number
APSO	$[-200, 200]$	5	50	10	3000	10^{-50}	30
PSO-LDIW	$[-200, 200]$	5	50	10	3000	10^{-50}	30
PSO-CK	$[-200, 200]$	5	50	10	3000	10^{-50}	30
PSO-TVAC	$[-200, 200]$	5	50	10	3000	10^{-50}	30
SPSO	$[-200, 200]$	5	50	10	3000	10^{-50}	30
SSQPSO	$[-200, 200]$	5	50	10	3000	10^{-50}	30

Table 4: Efficacy comparisons of PSO schemes on nine benchmark functions

	Function No.	Index	APSO[46]	PSO-LDIW [28]	PSO-CK [3]	PSO-TVAC [24]	SPSO[31]	SSQPSO
Unimodal	f_1	Mean	2.39×10^{-14}	2.00×10^{-07}	3.94×10^{-03}	$1.89\times10^{+00}$	3.76×10^{-06}	4.53×10^{-28}
		Best Value	2.43×10^{-17}	1.65×10^{-09}	3.80×10^{-07}	8.04×10^{-01}	4.40×10^{-10}	3.96×10^{-51}
		Std. Dev.	3.37×10^{-14}	1.42×10^{-07}	5.54×10^{-03}	$1.05\times10^{+00}$	5.31×10^{-06}	6.41×10^{-28}
	f_2	Mean	2.97×10^{-04}	2.00×10^{-08}	7.73×10^{-01}	$4.67\times10^{+02}$	2.93×10^{-04}	2.07×10^{-23}
		Best Value	1.72×10^{-08}	1.46×10^{-07}	1.34×10^{-01}	$3.02\times10^{+01}$	4.34×10^{-05}	7.13×10^{-51}
		Std. Dev.	3.68×10^{-04}	1.49×10^{-04}	5.58×10^{-01}	$5.60\times10^{+02}$	3.17×10^{-04}	2.93×10^{-23}
	f_3	Mean	5.25×10^{-03}	2.00×10^{-09}	7.93×10^{-02}	8.43×10^{-01}	1.92×10^{-02}	2.41×10^{-22}
		Best Value	3.62×10^{-04}	2.61×10^{-04}	4.19×10^{-02}	4.15×10^{-01}	1.13×10^{-02}	1.52×10^{-37}
		Std. Dev.	3.47×10^{-03}	2.63×10^{-03}	3.63×10^{-02}	3.94×10^{-01}	1.02×10^{-02}	3.40×10^{-22}
	f_4	Mean	2.51×10^{-01}	2.00×10^{-10}	$1.18\times10^{+00}$	$2.00\times10^{+00}$	9.68×10^{-02}	3.04×10^{-07}
		Best Value	1.06×10^{-02}	2.55×10^{-05}	7.35×10^{-01}	$1.67\times10^{+00}$	4.77×10^{-02}	1.15×10^{-28}
		Std. Dev.	3.24×10^{-01}	1.24×10^{-02}	3.23×10^{-01}	4.54×10^{-01}	5.53×10^{-02}	3.17×10^{-07}
	f_5	Mean	1.29×10^{-02}	2.00×10^{-11}	5.68×10^{-01}	8.37×10^{-01}	1.45×10^{-01}	1.98×10^{-03}
		Best Value	3.04×10^{-03}	8.18×10^{-03}	2.76×10^{-01}	1.04×10^{-01}	2.16×10^{-02}	6.36×10^{-04}
		Std. Dev.	1.39×10^{-02}	7.15×10^{-02}	3.21×10^{-01}	$1.03\times10^{+00}$	9.39×10^{-02}	1.09×10^{-03}
	f_6	Mean	$6.51\times10^{+00}$	2.00×10^{-12}	$5.60\times10^{+01}$	$2.28\times10^{+05}$	$4.69\times10^{+00}$	$1.25\times10^{+01}$
		Best Value	$4.93\times10^{+00}$	$2.57\times10^{+00}$	$8.43\times10^{+00}$	$1.48\times10^{+02}$	6.60×10^{-01}	6.10×10^{-08}
		Std. Dev.	$2.07\times10^{+00}$	$2.06\times10^{+00}$	$4.00\times10^{+01}$	$3.22\times10^{+05}$	$3.20\times10^{+00}$	$1.75\times10^{+01}$
Multimodal	f_7	Mean	$2.00\times10^{+01}$	2.00×10^{-13}	$2.00\times10^{+01}$	$2.00\times10^{+01}$	$2.00\times10^{+01}$	$2.00\times10^{+01}$
		Best Value	$2.00\times10^{+01}$	$2.00\times10^{+01}$	$2.00\times10^{+01}$	$2.00\times10^{+01}$	$2.00\times10^{+01}$	$2.00\times10^{+01}$
		Std. Dev.	2.10×10^{-04}	2.04×10^{-03}	6.53×10^{-03}	1.08×10^{-02}	2.30×10^{-05}	3.78×10^{-04}
	f_8	Mean	2.62×10^{-01}	2.00×10^{-14}	2.08×10^{-01}	2.75×10^{-01}	1.48×10^{-01}	7.12×10^{-02}
		Best Value	9.10×10^{-02}	2.07×10^{-01}	1.80×10^{-01}	2.11×10^{-01}	1.03×10^{-01}	4.24×10^{-02}
		Std. Dev.	1.78×10^{-01}	2.04×10^{-04}	1.99×10^{-02}	9.07×10^{-02}	4.48×10^{-02}	2.09×10^{-02}
	f_9	Mean	$1.59\times10^{+01}$	2.00×10^{-15}	$3.60\times10^{+01}$	$9.87\times10^{+01}$	$5.51\times10^{+01}$	$1.00\times10^{+00}$
		Best Value	$1.19\times10^{+01}$	$7.96\times10^{+00}$	$1.84\times10^{+01}$	$4.21\times10^{+01}$	$3.58\times10^{+01}$	1.15×10^{-03}
		Std. Dev.	$3.25\times10^{+00}$	2.04×10^{-05}	$2.24\times10^{+01}$	$6.55\times10^{+01}$	$1.64\times10^{+01}$	7.42×10^{-01}

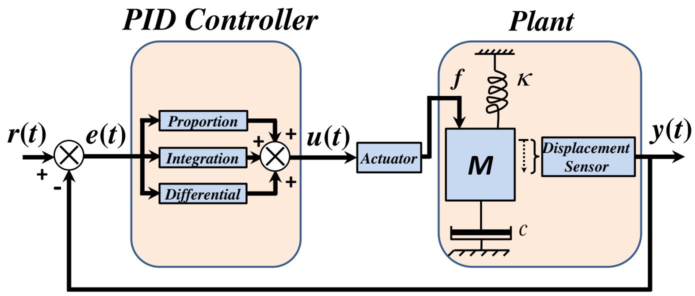


Figure 13: Schematic diagram of PID control system.

Moreover, the transfer function is derived as follows:

$$G(s) = G\left[sI - A\right]^{-1}\mathscr{B} = \frac{1}{Ms^2 + cs + \kappa}.$$

In this example, the PID controller is adopted of the following form:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt},$$

where the parameters K_p, K_i and K_d are three gains of the PID controller to be tuned. The fitness function is chosen as $J(t) = \int_0^\infty t|e(t)|dt$. Let the mass of the free body $M = 1$ (kg), the elastic coefficient $\kappa = 0.03$ (N/m), and the damping coefficient be $c = 1.8$ (N*s/m). Set the maximum search number be 80 times, the population size of particles be $\mathscr{P} = 50$, the inertia weight be $w = 0.6$, and the maximum velocity be $v_{i\max}(k) = 5$ (m/s). Moreover, the initialization of the velocity and position of particles are, respectively, generated by $2 * rand$ and $300 + 600 * rand$. The solution accuracy is 0.00001, and the lower and upper bounds of the search region are not specified. The probability transition matrix $\mathscr{T}$ is set as

$$\mathscr{T} = \begin{bmatrix} 0.3 & 0.7 & 0 & 0 \\ 0 & 0.6 & 0.4 & 0 \\ 0 & 0 & 0.2 & 0.8 \\ 0.1 & 0 & 0 & 0.9 \end{bmatrix}.$$

Let $\boldsymbol{p}(t) = [K_p(t), K_i(t), K_d(t)]^T$ be the parameter vector related to the desired PID controller, which is also deemed as the position vector of the ith particle $\boldsymbol{x}_i(k) = [x_{i1}(k), x_{i2}(k), x_{i3}(k)]^T$, i.e.,

$$\begin{bmatrix} x_{i1}(k) \\ x_{i2}(k) \\ x_{i3}(k) \end{bmatrix} = \begin{bmatrix} K_p(t) \\ K_i(t) \\ K_d(t) \end{bmatrix}.$$

Here, since the number of the parameters (i.e., K_p, K_i and K_d) to be optimized is three, we set the population dimensionality as $D = 3$.

Fig. 14 illustrates the flowchart of the SSQPSO algorithm utilized for the parameter tuning of PID controller, which contains three parts, i.e., the SSQPSO algorithm, interface function and

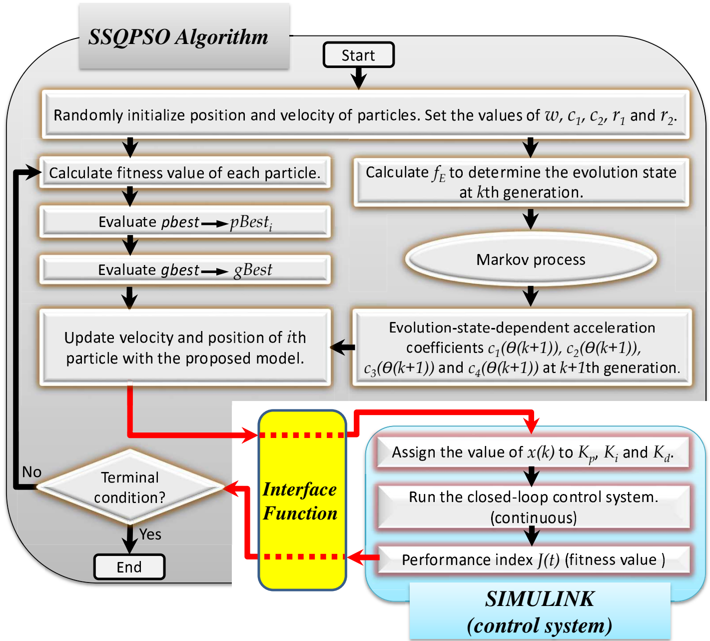


Figure 14: Flowchart of SSQPSO on PID controller.

simulink dynamic component. The function of the SSQPSO scheme is to seek the global optimal positions ($gBest$s), and then the global optimal $x(k)$ is passed to the simulink dynamic system by using the interface function. Then, the interface function feeds the performance index $J(t)$ back to the SSQPSO algorithm to determine whether the search process should be continued or not.

For an over-damping system, the parameter optimization course (i.e., the evolutionary procedure of the $gBest$ value) of the PID law can be examined in Fig. (15a). It is clear that there exists a large variation of $gBest$ before generation $k = 30$, which implies that the particles move with a high speed. After generation $k = 30$, the position fluctuation becomes small. Such a phenomenon is mainly caused by the fact that the particles are close to the optimal position at generation $k = 30$. Fig. (15b) shows the fitness value with respect to $gBest$, from which it can be seen that sharp changes appear before generation $k = 51$, and disappear after entering the stage of fine-searching. Fig. (15c) provides the step function response of the over-damping system in case of the optimized PID strategy.

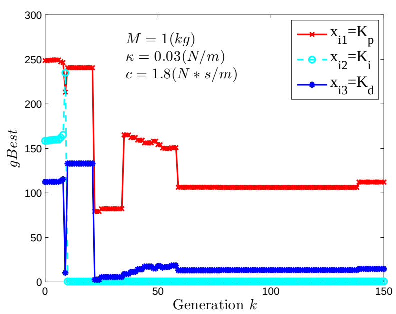
As is well known, a second-order system tends to be stable as long as it is not non-damping or negative-damping. A natural question is whether the feasibility of the search process of the optimal PID controller depends on the stability of the closed-loop system. To investigate this matter, the addressed SSQPSO is realized on the PID controller for four kinds of damping systems (including the negative-damping system, non-damping system, under-damping system and critical-damping system), and the step function responses are displayed in Figs. 16-19. It can be observed from Figs. 16-19 that the developed optimization algorithm is always effective regardless of the stability of the target system. In addition, the evolutions of the state variable q_2 (i.e., dy/dt) for the

Table 5: Relationship between $\mathscr{P}$ and the computational time

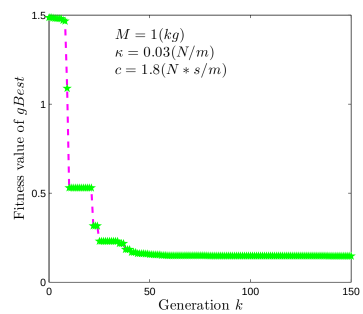
($\mathscr{P}$)	30	40	50	60	70	80	90	100
average cost per generation (seconds)	0.7571	0.9083	1.2006	1.4379	1.6359	1.9150	2.1141	2.4497

aforementioned four systems are shown in Fig. 20, from which we can see that the original systems are well stabilized by the PID controllers under the designed SSQPSO scheme.

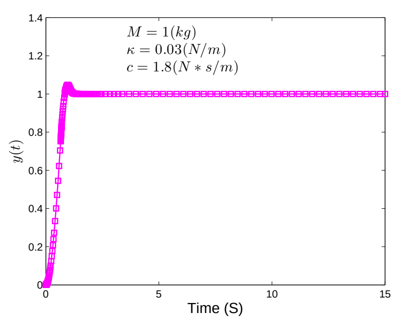
Although the fast tuning of PID is achieved by the studied SSQPSO, the computational time depends heavily on the population size. Generally speaking, a larger size of the particles in population can avoid error propagation to some degree, and therefore provide better position accuracy. In order to figure out the relationship between the computation cost and the particles size, the PID tuning is tested with different population size (see Table 5), and basically each generation takes seconds to implement the optimal evaluation. Table 5 indicates that the computational time increases with the particles size. For this very reason, a tradeoff between the accuracy and computation cost should not be ignored in the choice of $\mathscr{P}$.



(a) Global best position $gBest$



(b) Fitness value of $gBest$



(c) Step function response

Figure 15: Optimizing process of the PID controller and the achieved step response of the overdamping target.

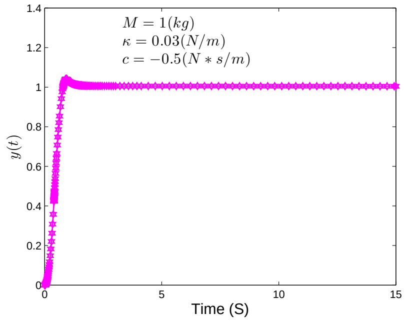


Figure 16: Step function response of the negative-damping target.

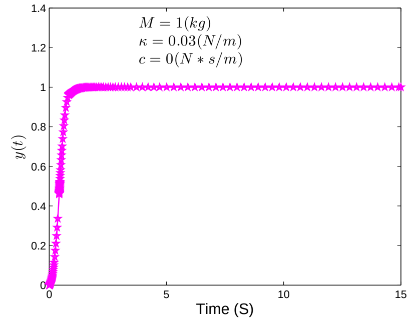


Figure 17: Step function response of the zero-damping target.

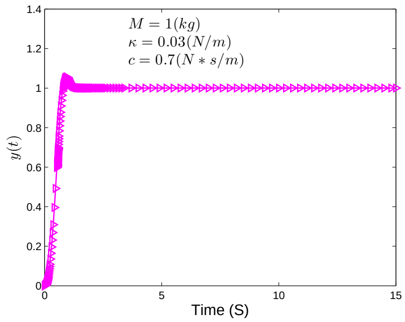


Figure 18: Step function response of the under-damping target.

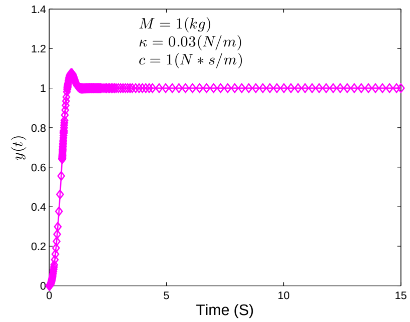


Figure 19: Step response of the critical-damping target.

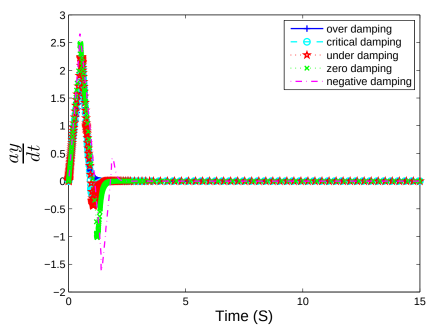


Figure 20: State evolutions.

6. Conclusions

An innovative PSO scheme named the SSQPSO algorithm has been put forward in this work, which aims to elevate the seek efficacy of the traditional counterpart. For the purpose of reflecting the periodic and sequential switching characteristics of four evolution states (i.e., exploration, exploitation, convergence and jumping-out), a Markov chain has been utilized with a special probability transition matrix. By resorting to the matrix sign function and Hadamard product as well as the concept of population density, an improved velocity update model has been established by introducing two quadratic acceleration terms. Then, the coarse-searching with a high speed and the fine-searching with a low speed can be switched according to the information of population density, thereby accelerating the search process. The experimental results have demonstrated the strength of the investigated SSQPSO scheme over other five existing PSO algorithms on the selected benchmark functions of unimodal and multimodal cases. Finally, the practicality of the addressed SSQPSO scheme has been illustrated via the parameter tuning issue of the PID controller in a mechanical systems.

Acknowledgements

This research work was funded by Institutional Fund Projects under grant no. (IFPIP: 33-135-1443). The authors gratefully acknowledge the technical and financial support provided by the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

References

[1] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, Inertia weight strategies in particle swarm optimization, in *Proceedings of 2011 Third World Congress on Nature and Biologically Inspired Computing*, Salamanca, Spain, 2011, pp. 633–640.

[2] M. G. Carneiro, R. Cheng, L. Zhao and Y. Jin, Particle swarm optimization for network-based data classification, *Neural Networks*, vol. 110, pp. 243-255, Feb. 2019.

[3] M. Clerc and J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[4] R. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39–43.

[5] D. Freitas, L. G. Lopes, and F. Morgado-Dias, Particle swarm optimisation: a historical review up to the current developments, *Entropy*, vol. 22, no. 3, art. no. 362, 2020.

[6] F. Fuentes-Hurtado, J. A. Diego-Mas, V. Naranjo, and M. Alcañiz, Evolutionary computation for modelling social traits in realistic looking synthetic faces, *Complexity*, vol. 2018, art. no. 9270152, 2018.

[7] J. Golten and A. Verwer, *Control System Design and Simulation*. New York: McGraw-Hill, USA, 1991.

[8] M. Hellwig and H. G. Beyer, Benchmarking evolutionary algorithms for single objective real-valued constrained optimization–a critical review, *Swarm and Evolutionary Computation*, vol. 44, pp. 927–944, 2019.

[9] F. Isdaryani, F. Feriyonika, and R. Ferdiansyah, Comparison of Ziegler-Nichols and Cohen Coon tuning method for magnetic levitation control system, *Journal of Physics: Conference Series*, vol. 1450, art. no. 012033, 2020.

[10] A. Jamasb, S. H. Motavalli-Anbaran, and K. Ghasemi, A novel hybrid algorithm of particle swarm optimization and evolution strategies for geophysical non-linear inverse problems, *Pure and Applied Geophysics*, vol. 176, no. 4, pp. 1601–1613, 2019.

[11] J. Kennedy, Minds and cultures: particle swarm implications, in *Proceedings of AAAI 1997 Fall Symposium Series: Socially Intelligent Agents*, Cambridge, MA, USA, 1997, pp. 67–72.

[12] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proceedings of IEEE International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942–1948.

[13] L. Li, A. A. F. Saldivar, Y. Bai, Y. Chen, Q. Liu, and Y. Li, Benchmarks for evaluating optimization algorithms and benchmarking MATLAB derivative-free optimizers for practitioners' rapid access, *IEEE Access*, vol. 7, pp. 79657–79670, 2019.

[14] W. Liu, Z. Wang, X. Liu, N. Zeng, and D. Bell, A novel particle swarm optimization approach for patient clustering from emergency departments, *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 632–644, 2019.

[15] W. Liu, Z. Wang, N. Zeng, F. E. Alsaadi, and X. Liu, A PSO-based deep learning approach to classifying patients from emergency departments, *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 1939–1948, 2021.

[16] Y. Liu, J. Liu and Y. Jin, Surrogate-assisted multipopulation particle swarm optimizer for high-dimensional expensive optimization, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, in press, DOI: 10.1109/TSMC.2021.3102298.

[17] X. Luo, Y. Yuan, S. Chen, N. Zeng and Z. Wang, Position-transitional particle swarm optimization-incorporated latent factor analysis, *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3958-3970, Aug. 2022.

[18] X. Luo, H. Wu, Z. Wang, J. Wang and D. Meng, A novel approach to large-scale dynamically weighted directed network representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9756-9773, Dec. 2022.

[19] X. Luo, Y. Zhong, Z. Wang and M. Li, An alternating-direction-method of multipliers-incorporated approach to symmetric non-negative latent factor analysis, *IEEE Transactions on Neural Networks and Learning Systems*, in press, DOI: 10.1109/TNNLS.2021.3125774.

[20] E. R. Pinto, E. G. Nepomuceno and A. S. L. O. Campanharo, Individual-based modelling of animal brucellosis spread with the use of complex networks, *International Journal of Network Dynamics and Intelligence*, vol. 1, no. 1, pp. 120-129, Dec. 2022.

[21] K. Premkumar, T. Thamizhselvan, M. Vishnu Priya, S. B. Ron Carter, and L. P. Sivakumar, Fuzzy anti-windup PID controlled induction motor, *International Journal of Engineering and Advanced Technology*, vol. 9, no. 1, pp. 184–189, 2019.

[22] I. U. Rahman, Z. Wang, W. Liu, B. Ye, M. Zakarya, and X. Liu, An N-state Markovian jumping particle swarm optimization algorithm, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 6626–6638, 2021.

[23] M. Rahman, Z. C. Ong, W. T. Chong, S. Julai, and X. W. Ng, Wind turbine tower modeling and vibration control under different types of loads using ant colony optimized PID controller, *Arabian Journal for Science and Engineering*, vol. 44, no. 2, pp. 707–720, 2019.

[24] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[25] F. M. Shakiba, M. Shojaee, S. M. Azizi and M. Zhou, Real-time sensing and fault diagnosis for transmission lines, *International Journal of Network Dynamics and Intelligence*, vol. 1, no. 1, pp. 36-47, Dec. 2022.

[26] M. Sheng, Z. Wang, W. Liu, X. Wang and S. Chen, A particle swarm optimizer with multi-level population sampling and dynamic p-learning mechanisms for large-scale optimization, *Knowledge-Based Systems*, vol. 242, art. no. 108382, Apr. 2022.

[27] Y. Shi and R. C. Eberhart, A modified particle swarm optimizer, in *Proceedings of 1998 IEEE International Conference on Evolutionary Computation*, Anchorage, AK, USA, 1998, pp. 69–73.

[28] Y. Shi and R. C. Eberhart, Parameter selection in particle swarm optimization, in *Proceedings of the 7th International Conference on Evolutionary Programming*, San Diego, CA USA, 1998, pp. 591–600.

[29] B. Song, Z. Wang, and L. Zou, An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve, *Applied Soft Computing*, vol. 100, art. no. 106960, 2021.

[30] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, *KanGAL Report*, rep. no. 2005005, 2005.

[31] Y. Tang, Z. Wang, and J.-A. Fang, Parameters identification of unknown delayed genetic regulatory networks by a switching particle swarm optimization algorithm, *Expert Systems with Applications*, vol. 38, no. 3, pp. 2523–2535, 2011.

[32] K.-H. Tseng, M.-Y. Chung, C.-Y. Chang, C.-L. Hsieh, and Y.-K. Tseng, Parameter optimization of nanosilver colloid prepared by electrical spark discharge method using Ziegler-Nichols method, *Journal of Physics and Chemistry of Solids*, vol. 148, art. no. 109650, 2021.

[33] M. Wang, H. Wang and H. Zheng, A mini review of node centrality metrics in biological networks, *International Journal of Network Dynamics and Intelligence*, vol. 1, no. 1, pp. 99-110, Dec. 2022.

[34] X. Wang, Y. Sun and D. Ding, Adaptive dynamic programming for networked control systems under communication constraints: a survey of trends and techniques, *International Journal of Network Dynamics and Intelligence*, vol. 1, no. 1, pp. 85-98, Dec. 2022.

[35] Z.-J. Wang, Z.-H. Zhan, S. Kwong, H. Jin and J. Zhang, Adaptive granularity learning distributed particle swarm optimization for large-scale optimization, *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1175-1188, Mar. 2021.

[36] X. Xia, L. Gui, F. Yu, H. Wu, B. Wei, Y.-L. Zhang and Z.-H. Zhan, Triple archives particle swarm optimization, *IEEE Transactions on Cybernetics*, vol. 50, no. 12, pp. 4862-4875, Dec. 2020.

[37] L. Xu, B. Song and M. Cao, An improved particle swarm optimization algorithm with adaptive weighted delay velocity, *Systems Science & Control Engineering*, vol. 9, no. 1, pp. 188-197, 2021.

[38] P.-Y. Yang, F.-I. Chou, J.-T. Tsai, and J.-H. Chou, Adaptive-uniform-experimental-design-based fractional-order particle swarm optimizer with non-linear time-varying evolution, *Applied Sciences*, vol. 9, no. 24, art. no. 5537, 2019.

[39] F. Yao, Y. Ding, S. Hong and S.-H. Yang, A survey on evolved LoRa-based communication technologies for emerging internet of things applications, *International Journal of Network Dynamics and Intelligence*, vol. 1, no. 1, pp. 4-19, Dec. 2022.

[40] N. Yu, R. Yang and M. Huang, Deep common spatial pattern based motor imagery classification with improved objective function, *International Journal of Network Dynamics and Intelligence*, vol. 1, no. 1, pp. 73-84, Dec. 2022.

[41] Y. Yuan, X. Tang, W. Zhou, W. Pan, X. Li, H.-T. Zhang, H. Ding and J. Goncalves, Data driven discovery of cyber physical systems, *Nature Communications*, vol. 10, no. 1, pp. 1-9, 2019.

[42] Y. Yuan, H. Zhang, Y. Wu, T. Zhu and H. Ding, Bayesian learning-based model-predictive vibration control for thin-walled workpiece machining processes, *IEEE/ASME transactions on mechatronics*, vol. 22, no. 1, pp. 509-520, 2016.

[43] N. Zeng, D. Song, H. Li, Y. You, Y. Liu and F. E. Alsaadi, A competitive mechanism integrated multi-objective whale optimization algorithm with differential evolution, *Neurocomputing*, vol. 432, pp. 170–182, Apr. 2021.

[44] N. Zeng, Z. Wang, W. Liu, H. Zhang, K. Hone and X. Liu, A dynamic neighborhood-based switching particle swarm optimization algorithm, *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9290-9301, 2022.

[45] Z.-H. Zhan, J. Xiao, J. Zhang, and W.-N. Chen, Adaptive control of acceleration coefficients for particle swarm optimization based on clustering analysis, in *Proceedings of 2007 IEEE Congress on Evolutionary Computation*, Singapore, 2007, pp. 3276–3282.

[46] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, Adaptive particle swarm optimization, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 39, no. 6, pp. 1362–1381, 2009.

[47] J. Zhang, H. S.-H. Chung, and W.-L. Lo, Clustering-based adaptive crossover and mutation probabilities for genetic algorithms, *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 326–335, 2007.

[48] Q. Zhang and Y. Zhou, Recent advances in non-Gaussian stochastic systems control theory and its applications, *International Journal of Network Dynamics and Intelligence*, vol. 1, no. 1, pp. 111-119, Dec. 2022.

[49] Y. Zhang, H. Li, E. Bao, L. Zhang, and A. Yu, A hybrid global optimization algorithm based on particle swarm optimization and Gaussian process, *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, pp. 1270–1281, 2019.