Digital redesign of uncertain interval systems based on extremal gain/phase margins via a hybrid particle swarm optimizer

Chen-Chien Hsu a,*, Wern-Yarng Shieh b, Chun-Hwei Gao c

a Department of Applied Electronic Technology, National Taiwan Normal University, 162 He-Ping East Rd., Section 1, Taipei, 10610, Taiwan
b Department of Computer and Communication Engineering, St. John's University, 498, Sec. 4, Tam King Rd., Tamsui, Taipei County, 25135, Taiwan
c Department of Electrical Engineering, Tamkang University, 151 Ying-Chuan Rd., Tamsui, Taipei County, 25137, Taiwan

ARTICLE INFO

Article history:
Received 20 March 2008
Received in revised form 14 June 2009
Accepted 23 August 2009
Available online 29 August 2009

Keywords:
Digital redesign
Particle swarm optimization
Interval plants
Uncertain systems
Gain margin
Phase margin
NM simplex search
Frequency response
Discretization
Digital control
Extremal systems

ABSTRACT

In this paper, a hybrid optimizer incorporating particle swarm optimization (PSO) and an enhanced NM simplex search method is proposed to derive an optimal digital controller for uncertain interval systems based on resemblance of extremal gain/phase margins (GM/PM). By combining the uncertain plant and controller, extremal GM/PM of the redesigned digital system and its continuous counterpart can be obtained as the basis for comparison. The design problem is then formulated as an optimization problem of an aggregated error function in terms of deviation on extremal GM/PM between the redesigned digital system having an interval plant and its continuous counterpart, and subsequently optimized by the proposed optimizer to obtain an optimal set of parameters for the digital controller. Thanks to the performance of the proposed hybrid optimizer, frequency-response performances of the redesigned digital system using the digital controller evolutionarily derived by the proposed approach bare a far better resemblance to its continuous-time counterpart in comparison to those obtained using existing open-loop discretization methods.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Most practical systems, such as flight vehicles, electric motors, and robots, are formulated in continuous-time uncertain settings. The uncertainties in these systems arise from unmodelled dynamics, parameter variation, sensor noises, actuator constraints, etc., of the systems. These variations of the uncertain parameters generally do not follow any of the known probability distributions and are most often quantified in terms of bounds [1,2]. Hence, practical systems or plants are most suitably represented by continuous-time parametric interval models [3,4]. Simple as they might be, uncertain models in the form of interval systems have provided a convenient way in constructing mathematical models for physical systems, based on which feasible design can be achieved for use in industry.

On the other hand, digital control is getting more and more popular because of the fast advances in digital technologies and computers [5,6]. The current trend toward digital control of dynamic systems is mainly due to the availability of low-cost digital computers and the advantages found in working with digital signals. It is also well known that digital control provides many advantages over the analogue control in terms of reliability, flexibility, cost, performance, etc. As a result, digital controllers are increasingly being used to implement control systems. The main stream in designing a suitable digital controller is to discretize the continuous-time plant first, and then to determine a digital controller for the discretized plant, which is called the direct digital design approach [7]. Another efficient approach to design a digital controller is called digital redesign [8,9] where an analog controller is first designed and then converted to an equivalent digital controller in the sense of state-matching [10]. For those who are familiar with analog controller design, the digital redesign approach is a great advantage. All they have to do is to design an analog controller by using continuous-time techniques, so that the overall system meets the performance specifications. The analog controller is then converted into a digital one by using any of the conventional discretization techniques [11], for example, Tustin transform, step invariant transform, Euler’s method, back-
ward integral, Boxer–Thaler transformation, etc. However, this kind of ‘open-loop’ design philosophy, where a sufficiently small sampling time is assumed, ignores the fact that the analog system is actually operating in a feedback configuration whose output is determined by the closed-loop system. As a result, poor suitability of the redesigned digital system is inevitable if open-loop design approaches are only considered. If a better system performance is required, closed-loop discretization methods [12–14] are required to derive the digital controllers. Unfortunately, the closed-loop discretization approaches deal with deterministic mathematical models only. There is no systematic approach available to effectively solve the problems of digital redesign of uncertain continuous-time interval systems. As an attempt to improve the performance of redesigned digital systems having an interval plant, this paper formulates the design problem to obtain a desired digital controller for the redesigned digital system as an optimization problem of an aggregated error function in terms of deviation on extremal gain margin (phase margin) (GM/PM) between the redesigned digital system and its continuous counterpart. Proceeding to the design of a desired digital controller for the redesigned digital system, however, we found that the objective functions formulated for optimization to derive a set of optimal parameters for the digital controller are generally non-convex [15] with serious nonlinear couplings of the uncertain plant parameters. Critical difficulties, therefore, arise because conventional gradient-based optimization algorithms cannot efficiently solve this problem.

Particle swarm optimization (PSO), with its capabilities of directed random search for global optimization [16–20] has provided a promising alternative to address the above-mentioned problems and difficulties. However, conventional PSO and its variants do not provide consistent and satisfactory performance for the problem under investigation. This motivates the development of a hybrid evolutionary algorithm [21–23] which incorporates particle swarm optimization (PSO) with the use of a center particle [24] in a swarm and an enhanced NM simplex search method [25–27] to improve the optimization performance to design a desired digital controller for redesigned digital systems having an interval plant.

The rest of the paper is organized as follows. Section 2 introduces the uncertain interval systems and establishes the design problem to be solved. Derivation of extremal GM/PM for interval systems is introduced in Section 3. A hybrid particle swarm optimizer incorporating an enhanced NM simplex search scheme is given in Section 4. Section 5 proposes an evolutionary approach using the hybrid PSO optimizer to derive digital controllers for uncertain interval systems based on resemblance of extremal gain/phase margins (GM/PM) between the redesigned digital system and its continuous counterpart. Two illustrated examples are demonstrated in Section 6. Conclusions are drawn in Section 7.

2. Problem description

Consider an uncertain interval plant in Fig. 1 given by [4]:

\[
G_p(s, a, b) = \frac{b_0 + b_1 s + b_3 s^2 + \cdots + b_{n-1} s^{n-1}}{a_0 + a_1 s + a_2 s^2 + \cdots + a_{n-1} s^{n-1} + s^n} = \frac{N(s)}{D(s)},
\]

where coefficient vectors \( a = (a_0, a_1, a_2, \ldots, a_{n-1}) \) and \( b = (b_0, b_1, b_2, \ldots, b_{n-1}) \) lie in the \( n \)-dimensional boxes

\[
A = \{ a : a_i \in [a_i^L, a_i^U], \quad \forall i = 0, 1, 2, \ldots, n - 1 \}
\]

and

\[
B = \{ b : b_i \in [b_i^L, b_i^U], \quad \forall i = 0, 1, 2, \ldots, n - 1 \},
\]

respectively.

Assume a fixed-structure controller described by a rational transfer function:

\[
C(s, p, q) = \frac{q_0 + q_1 s + q_2 s^2 + \cdots + q_m s^m}{p_0 + p_1 s + p_2 s^2 + \cdots + p_m s^m}
\]

is designed so that the closed-loop system in Fig. 1 satisfies the performance specifications, where \( p = [p_0, p_1, \ldots, p_m] \) and \( q = [q_0, q_1, \ldots, q_m] \) designate the vectors of the controller parameters, which are given and known \textit{a priori}. When the controller \( C(s, p, q) \) is placed in series with plant \( G_p(s, a, b) \) and closed under unity feedback as shown in Fig. 1, the transfer function of the closed-loop system becomes:

\[
G_{cl}(s, p, q, a, b) = \frac{1 + C(s, p, q) \cdot G_p(s, a, b)}{1 + C(s, p, q) \cdot G_p(s, a, b)}
\]

\[
= \frac{N(s, p, q, a, b)}{D(s, p, q, a, b)}
\]

\[
= \frac{n_0 + n_1 s + n_2 s^2 + n_3 s^3 + n_4 s^4 + \cdots}{d_0 + d_1 s + d_2 s^2 + d_3 s^3 + d_4 s^4 + \cdots}
\]

When the continuous plant is subject to digital control, the sampled-data system of the interval plant preceded by a zero-order hold (ZOH) [28] is illustrated in Fig. 2, where the discrete equivalent plant of \( G_p(s, a, b) \) can be further represented in the purely discrete-time domain as:

\[
G_{cl}(z, a, b) = \frac{1 - z^{-n}}{s} \frac{G_p(s, a, b)}{s}
\]

To this end, the redesigned digital control system for the continuous system having an interval plant \( G_p(s, a, b) \) is illustrated in Fig. 3, where \( G_{cl}(z, a, b) \) is the digital controller to be designed. Combining the digital controller \( C_d(z, u, v) \) and the discrete-time equivalent plant \( G_{cl}(z, a, b) \) in Fig. 3, we have the closed-loop transfer function in the discrete-time domain as:

\[
G_{cl}(z, u, v, a, b) = \frac{C_d(z, u, v) \cdot G_{cl}(z, a, b)}{1 + C_d(z, u, v) \cdot G_{cl}(z, a, b)}
\]

Now the problem to derive an optimal digital controller for the redesigned digital system having an interval plant can be formulated as: Given interval plant \( G_p(s, a, b) \) and an analog controller \( C(s) \), determine an optimal digital controller \( C_d(z) \), so that frequency performance in extremal gain/phase margins of the
redesigned digital system in Fig. 3 closely match that of its continuous counterpart in Fig. 1.

3. Extremal gain margin and phase margin

Thanks to the perturbation of uncertain plant parameters, there are infinite set of intervals for the responses in the interval system in Fig. 1, and therefore infinite set of gain and phase margins (GM/PMs). By investigating the extremal systems of the uncertain systems in Figs. 1 and 3, extremal GM/PM, i.e., largest and smallest GM/PM, for the redesigned digital system and the original continuous system can be obtained, as will be demonstrated in this section. Based on the extremal GM/PM, we can formulate the design problem as an optimization problem to derive a set of optimal parameters for the digital controller.

3.1. Extremal systems

Consider the interval plant $G_p(s) = N(s)/D(s)$. The boundary of the complex plane set for interval plant $G_p(s)$ can be described as follows.

$$
\partial G_p(s) \subset \left( \frac{K_p(s)}{S_p(s)} \cup \frac{N_p(s)}{D_p(s)} \right) = \frac{N_p(s)}{(1 - \lambda)D_p(s) + \lambda D_p(s)}
$$

where $\lambda = [0,1]$, $(jk) \in \{1,2,3,4\}$, $i \in \{1,2,3,4\}$. The set of points $K_p$ and $D_p$ are the vertices of $N(s)$ and $D(s)$, respectively, and the set of points $S_p$ and $P_p$ are the edges of $N(s)$ and $D(s)$, respectively. $G_p(s)$ is referred to as the so-called extremal systems of $G_p(s)$ [29,41].

With the frequency domain properties [29], it is understood that the boundary of the complex plane set associated with the loop transfer function $C(s)G_p(s)$ can be derived as follows:

$$
\partial (C(s)G_p(s)) \subset C(s) \left( \frac{K_p(s)}{S_p(s)} \cup \frac{N_p(s)}{D_p(s)} \right) = \frac{N_p(s)}{(1 - \lambda)D_p(s) + \lambda D_p(s)}
$$

$$
= C(s)G_p(s) \Rightarrow C(s)G_p(s) = G_p(s)
$$

Alternatively, a set of 32 extremal systems for $G_p(s)$ can be represented as

$$
G_p(s, \lambda), \quad i = 1, 2, 3, \ldots, 32, \quad \lambda = [0,1].
$$

Substituting $s = j\omega$ into the extremal systems $G_p(s)$, we have

$$
G_p(s, \lambda) |_{s=j\omega} = G_p(j\omega, \lambda) = |G_P(j\omega, \lambda)|/G_P(j\omega, \lambda),
$$

where

$$
i = 1, 2, 3, \ldots, 32, \quad \lambda = [0,1].
$$

Based on the 32 extremal systems above, frequency-response envelope of the interval system can be constructed where boundary of the magnitude envelope is calculated from the Kharitonov polynomials and Kharitonov segments, and boundary of the phase envelope is calculated from the Kharitonov polynomials [29].

3.2. Steps to obtain extremal GM/PM for the uncertain interval system

Before the design can be performed, we need to calculate the extremal GM/PM of the uncertain interval system via the extremal systems. Key steps to obtain the extremal GM/PM are outlined as follows:

**[Step 1] (Extremal systems)**

Obtain the loop transfer function $G(s) = G_p(s)C(s)$ as an interval system. 32 extremal systems associated with the interval system $G(s)$ can be obtained via Eq. (11), denoted as $G_{Ei}(s), i = 1, 2, 3, \ldots, 32$. Substituting $s = j\omega$ into the extremal systems $G_{Ei}(s)$, we have

$$
G_{Ei}(s, \lambda) |_{s=j\omega} = G_{Ei}(j\omega, \lambda) = |G_E(j\omega, \lambda)|/G_E(j\omega, \lambda),
$$

where

$$
i = 1, 2, 3, \ldots, 32, \quad \lambda \in [0,1].
$$

**[Step 2] (Extremal GM)**

To obtain the smallest and largest gain margin of the loop transfer function $G(s)$, we calculate the frequency response of the extremal systems $G_{Ei}(s)$ and locate the intersections on the real axis when phase is $-180^\circ$ for each extremal system. Let

$$
f_i(\omega, \lambda) = |G_E(j\omega, \lambda) + 180^\circ| = 0, \quad i = 1, 2, 3, \ldots, 32.
$$

Solving Eq. (13) and substituting the solution obtained into $|G_E(j\omega, \lambda)|$, we have the intersections $g_j, j = 1, 2, 3, \ldots$ on the real axis. Thus, the intersection closest to point $(-1,0)$ on the real axis for every extremal system $G_{Ei}(s)$ can be obtained as:

$$
\rho_i = \max\{g_j\}, \quad j = 1, 2, 3, \ldots
$$

and the intersection farthest from point $(-1,0)$ on the real axis for every extremal system $G_{Ei}(s)$ can be identified, respectively. Take

$$
\alpha = \max(\rho_i) \quad \text{and} \quad \beta = \min(\sigma_i)
$$

for all extremal systems $G_{Ei}(s), i = 1, 2, 3, \ldots 32$ Therefore, the smallest gain margin $GM_{\text{lower}}$ and largest gain margin $GM_{\text{upper}}$ associated with the controller $C(s)$ can be obtained as:

$$
GM_{\text{lower}} = 20\log\frac{1}{\alpha}
$$

and

$$
GM_{\text{upper}} = 20\log\frac{1}{\beta},
$$

respectively.

**[Step 3] (Extremal PM)**

To obtain the smallest and largest phase margin of the loop transfer function $sG(s)$, we need to locate the intersections on the unit circle of the frequency response for each extremal system $G_{Ei}(s)$. Let $h_i(\omega, \lambda) = |G_E(j\omega, \lambda)| - 1 = 0, \quad i = 1, 2, 3, \ldots, 32$.

Solving Eq. (19) and substituting the solution obtained into $|G_E(j\omega, \lambda)|$, we have the intersections $p_j, j = 1, 2, 3, \ldots$ on the unit circle for each extremal system $G_{Ei}(s)$. Thus, the smallest angle between $-180^\circ$ and intersections $p_j$

$$
q_i = \min\{p_j\}, \quad j = 1, 2, 3, \ldots
$$

and the largest angle between $-180^\circ$ and intersections $p_j$

$$
\psi_i = \max\{p_j\}, \quad j = 1, 2, 3, \ldots
$$

for each extremal system $G_{Ei}(s)$ can be identified, respectively. Take

$$
\kappa = \min(q_i) \quad \text{and} \quad \chi = \max(\psi_i)
$$

for all extremal systems $G_{Ei}(s), i = 1, 2, 3, \ldots 32$ Therefore, the smallest phase margin $PM_{\text{lower}}$ and largest phase margin $PM_{\text{upper}}$ associated with the controller $C(s)$ can be obtained as:

$$
PM_{\text{lower}} = \kappa + 180^\circ.
$$

$$
PM_{\text{upper}} = \chi + 180^\circ.
$$
and

$$PM_{\text{c,u,per}} = \chi + 180^\circ,$$  \hspace{1cm} (24)

respectively.

### 3.3. Steps to obtain extremal GM/PM for the redesigned digital system

The derivation of the extremal GM/PM associated with the discrete loop transfer function can be obtained in a similar way. By discretizing the 32 extremal systems \( G_{np}(s) \) associated with the interval system, a set of 32 extremal systems in the discrete-time domain can be obtained for incorporation with a digital controller to respectively form a loop transfer function, based on which frequency responses can be calculated to derive the extremal GM/PM for the redesigned digital systems via procedures similar to Steps 2 and 3 in Section 3.2. Because of the similarities, steps to obtain the extremal GM/PM for the redesigned digital systems will not be re-iterated.

### 4. Hybrid particle swarm optimizer incorporating an enhanced NM simplex search scheme

#### 4.1. Particle swarm optimization

Particle swarm adaptation has been shown to successfully optimize a wide range of continuous functions [16,17]. The algorithm, which is based on a metaphor of social interaction, searches a space by adjusting the trajectories of individual vectors, called “particles” as they are conceptualized as moving points in multidimensional space [18]. As an evolutionary technique, the PSO is a population-based algorithm, formed by a set of particles representing potential solutions for a given problem. Each particle moves through a \( n \)-dimensional search space, with an associated position vector \( x_i(t) = [x_{i1}(t), x_{i2}(t), ..., x_{in}(t)] \) and velocity vector \( v_i(t) = [v_{i1}(t), v_{i2}(t), ..., v_{in}(t)] \) for the current evolutionary iteration \( t \). The individual particle in PSO flies in the search space with velocity which is dynamically adjusted according to its own flying experience and its companions’ flying experience [19]. The former was termed cognition-only model and the latter was termed social-only model [20]. By integrating these two types of knowledge, the particle behavior in a PSO can be modeled by using the following equations:

\[
v_i(t + 1) = w \times v_i(t) + c_1 \times \text{rand} \times (p_{\text{best}}_i - x_i(t)) + c_2 \times \text{rand} \times (G_{\text{best}} - x_i(t)) \tag{25}
\]

\[
x_i(t + 1) = x_i(t) + v_i(t + 1) \tag{26}
\]

where \( c_1, c_2 \): acceleration constants (Eberhart and Shi [30] suggested using 0.2–2 with a typical value of 2 [21]); \( \text{rand} \): random number between 0 and 1; \( x_i(t) \): the position of particle \( i \) at iteration \( t \); \( v_i(t) \): the velocity of particle \( i \) at iteration \( t \); \( w \): weight factor (Hu and Eberhart [31] suggested using 0.4–0.9); \( G_{\text{best}} \): best previous position among all the particles; \( p_{\text{best}}_i \): the best previous position of particle \( i \).

Note that the first term on the right-hand side of the velocity-updating rule in Eq. (25) represents the previous velocity, which provides the necessary momentum for particles to roam across the search space. The second term, known as the “cognitive” component, represents the personal thinking of each particle, which encourages the particles to move toward their own best positions found so far. The third term is known as the “social” component, which represents the collaborative effect of the particles in finding the global optima.

**Fig. 4** shows the flow chart of a typical PSO algorithm, in which a population of particles is initialized with random position \( x_i \) and velocity \( v_i \). Fitness of particles is evaluated by calculating the objective function \( f(x_i) \). The current position of each particle is set as \( p_{\text{best}}_i \). The \( p_{\text{best}}_i \) with best value in the swarm is set as \( G_{\text{best}} \). As evolution continues, next position for each particle is evaluated by using Eqs. (25) and (26). If a better position is achieved by an agent, the \( p_{\text{best}}_i \) value is replaced by the current value. If a new \( G_{\text{best}} \) value is better than the previous \( G_{\text{best}} \) value, the \( G_{\text{best}} \) value is replaced by the current \( G_{\text{best}} \) value. Iterations repeat until a predetermined iteration number is reached.

#### 4.2. The Nelder–Mead simplex search method

The Nelder–Mead simplex (NM simplex) search method was proposed by Nelder and Mead [25] in 1965 as a local search method for unconstrained optimization problems without using gradient information. Basically the method solves the optimization problem by continuously rescaling a \( N \)-dimensional simplex formed by \( N + 1 \) vertex points at each iteration based on the landscape and local behavior of the function by using four basic procedures: reflection, expansion, contraction, and shrinkage. Through these procedures, the simplex continuously improves itself and gets closer to the optimum \( X^* \) as iterations continue. To illustrate the operations of the method, the original NM simplex
procedures are outlined below by using a two-dimensional case \((N = 2)\) as an example.

### 4.2.1. Initialization

For a function of \(N\) variables, create \(N + 1\) vertex points to form an initial \(N\)-dimensional simplex. Evaluate the function value at each extreme point (or vertex) of the simplex.

### 4.2.2. Reflection

Determine \(X_{\text{high}}, X_{\text{sechi}}, X_{\text{low}}\) corresponding to \(f_{\text{high}}, f_{\text{sechi}}, f_{\text{low}}\) of the highest, the second highest, and the lowest objective function values, respectively. Find the center \(X_{\text{cent}}\) of the simplex excluding \(X_{\text{high}}\) (in the minimization case). A new vertex \(X_{\text{refl}}\) can be obtained by reflecting the worst point according to the following equation (see Fig. 5a)

\[
X_{\text{refl}} = (1 + \alpha)X_{\text{cent}} - \alpha X_{\text{high}},
\]

where \(\alpha (\alpha > 0)\) is the reflection coefficient. Nelder and Mead [25] suggested that \(\alpha = 1\) is chosen. If \(f_{\text{flow}} \leq f_{\text{refl}} \leq f_{\text{sechi}}\), accept the reflection by replacing \(X_{\text{high}}\) by \(X_{\text{refl}}\).

### 4.2.3. Expansion

If the reflection produces a objective function evaluation smaller than \(f_{\text{flow}}\) (i.e. \(f_{\text{refl}} < f_{\text{low}}\)), an expansion is performed to extend the search space in the same direction for further function improvement by the following equation (see Fig. 5b)

\[
X_{\text{exp}} = \gamma X_{\text{refl}} + (1 - \gamma)X_{\text{cent}},
\]

where \(\gamma\) is the expansion coefficient \((\gamma > 1)\). Nelder and Mead [25] suggested \(\gamma = 2\). If \(f_{\text{exp}} < f_{\text{low}}\), the expansion is accepted by replacing \(X_{\text{high}}\) with \(X_{\text{exp}}\), otherwise, \(X_{\text{exp}}\) replaces \(X_{\text{high}}\).

### 4.2.4. Contraction

When \(f_{\text{refl}}\) lies between \(f_{\text{high}}\) and \(f_{\text{sechi}}\), replaces \(X_{\text{high}}\) with \(X_{\text{refl}}\). Contraction (outward contraction) is performed between \(X_{\text{cent}}\) and \(X_{\text{high}}\) to identify \(X_{\text{cont}}\) (see Fig. 5c). If \(f_{\text{refl}}\) is larger than \(f_{\text{high}}\), \(X_{\text{refl}}\) does not replace \(X_{\text{high}}\), and an attempt is tried to locate \(X_{\text{cont}}\) between \(X_{\text{cent}}\) and \(X_{\text{high}}\) (see Fig. 5d). The contraction vertex is calculated by the following equation:

\[
X_{\text{cont}} = \beta X_{\text{high}} + (1 - \beta)X_{\text{cent}},
\]

where \(\beta\) is the contraction coefficient \((0 < \beta < 1)\). Nelder and Mead [25] suggested \(\beta = 0.5\). If \(f_{\text{cont}} \leq f_{\text{high}}\), replace \(X_{\text{high}}\) with \(X_{\text{cont}}\).

### 4.2.5. Shrinkage

If \(f_{\text{cont}} > f_{\text{high}}\) in step 4, contraction failed and shrinkage will be the next attempt. This can be done by shrinking the entire simplex (except \(X_{\text{low}}\)) by (see Figs. 5e and 3f)

\[
X_i \leftarrow \delta X_i + (1 - \delta)X_{\text{low}},
\]

where \(\delta\) is the shrinkage coefficient \((0 < \delta < 1)\). Nelder and Mead [25] suggested \(\delta = 0.5\).

### 4.3. Enhanced NM search method

The Nelder–Mead simplex method has been considered as one of the most successful optimization methods based solely on function value comparison. There have been abundant studies devoted to various modifications of the Nelder–Mead simplex method [23, 26, 27, 32]. However, there is still room for further improvements as far as optimization accuracy and convergence rate are concerned.

#### 4.3.1. Continuous expansion

Note that the expansion procedure in the conventional NM simplex search method focuses on further exploration along the promising direction. It is natural that we introduce an extra expansion procedure in a hope that the promising direction can be further explored to improve optimization efficiency. Fig. 6 illustrates a two-dimensional continuous expansion procedure, in which dashed lines stand for the operation of the continuous expansion according to the following equation:

\[
X_{\text{con exp}} = \epsilon X_{\text{exp}} + (1 - \epsilon)X_{\text{cent}},
\]

where \(\epsilon\) is the continuous expansion coefficient \((\epsilon > 1)\), for example \(\epsilon = 2\). When \(X_{\text{exp}}\) is better than \(X_{\text{low}}\), the continuous expansion is repeatedly performed as long as \(X_{\text{con exp}}\) is better than \(X_{\text{low}}\). Let \(X_{\text{con exp}}\) replace \(X_{\text{high}}\).

#### 4.3.2. Perturbed shrinkage

Note that if the simplex after shrinkage operator does not contain the desired optimum, premature convergence generally occurs as a result of subsequent iterations. Consequently, when...
reflection, expansion, and shrinkage operators failed, there is an implication that the optimal solution may not reside inside the simplex currently generated. Further application of the shrinkage operator is not desired. We propose a modification to the shrinkage operator according to the following formula:

\[
X_i \leftarrow X_{\text{old}} + \text{rand} \times (X_{\text{high}} - X_{\text{low}}) \\
X_{\text{high}} = X_{\text{low}} + \rho \times X_{\text{low}}.
\]

where \([X_{\text{high}}, X_{\text{low}}]\) represents the boundary of the search space of the optimization problem, \(\text{rand} \in [0, 1]\), and \(\rho\) is a random noise drawn from normal distribution \(N(0, \sigma)\). For example, \(\sigma = 0.1\). The objective of the perturbed shrinkage is therefore to re-generate new individuals (excluding \(X_{\text{low}}\)) around the best fit individual for further evaluations.

To show the effectiveness of the enhanced NM (E-NM) search method incorporating the continuous expansion and perturbed shrinkage operators, 18 commonly-used benchmark functions are adopted for optimization [25,32–37]. Simulation results obtained via the enhanced NM search method have demonstrated that successful rate in optimizing these benchmark functions has been dramatically increased via the enhanced NM simplex method, in comparison to the original NM simplex search method.

4.4. Hybridization of PSO and the enhanced NM search scheme

As a global optimizer, PSO has been widely adopted to optimize a wide range of continuous functions. Practical engineering problems are, however, very complex with high dimensions of search space, which makes existing PSO variants perform less satisfactorily in terms of accuracy and convergence. As a matter of fact, the balance between exploration and exploitation has not yet achieved for most PSO variants. It is therefore the objective of this paper to propose a hybrid optimization method which incorporates the E-NM into PSO, fully utilizing their strength in exploitation and exploration search, respectively, with extra help of a center particle to improve the evolution efficiency.

4.4.1. Hybrid NM-PSO-CENTER optimization method

It has been proved that center particle of a swarm of a PSO algorithm plays a critical role during the optimization process, where the position and velocity of the center particle can be calculated according to the following formula [24]:

\[
x_{\text{cd}} = \frac{1}{P} \sum_{i=1}^{P} x_i \\
v_{\text{cd}} = \frac{1}{P} \sum_{i=1}^{P} v_i,
\]

respectively. Because of its significance, the center particle is generally closer to the optimum than \(G_{\text{best}}\) during the search [24]. More importantly, due to frequent appearance as the best particle of swarm, it often attracts other particles and guides the search direction of the whole swarm. Eventually, the swarm statistically converges to the center particle through iterations. Taking advantage of this statistical characteristic, we have the center particle incorporated into the hybrid optimization scheme, NM-PSO-C, as depicted in Fig. 7. The rationale of the proposed approach is that we use PSO for exploration search and the enhanced NM simplex search for exploitation search. A total of \(P\) particles are used by PSO in the proposed approach. Extra \(N\) points are generated around \(G_{\text{best}}\) to generate a simplex of \((N + 1)\) vertex points for the enhanced NM simplex search method as an attempt to offer a thorough exploitation in the promising direction along the best solution \(G_{\text{best}}\) so far obtained. By doing so, we can achieve the objective of using fewer particles to locate the global optima.

1. Initial population: randomly generate \(P\) particles.
2. Ranked population: calculate fitness for each particle for sorting the particles in the swarm according to the fitness in descending order.
3. Center particle: calculate the position and velocity of the center particle and replace particle \(m (m \neq 1)\), e.g., \(m = 5\), in the current swarm by the center particle.
4. Exploitation search: choose the best particle \(G_{\text{best}}\) every \(k\) iterations (e.g., \(k = 10\)). Generate \(N\) vertex points around \(G_{\text{best}}\) so that the enhanced NM simplex search can be applied to obtain a new individual \(G_{\text{best} \text{new}}\) for \(g\) iterations (e.g., \(g = 20\)). If \(G_{\text{best} \text{new}}\) is better than \(G_{\text{best}}\), replace \(G_{\text{best}}\) by \(G_{\text{best} \text{new}}\) for further evolution.
5. Exploration search: based on \(G_{\text{best}}\), new particles can be generated to form a new population.
6. Termination: if terminating condition is satisfied, stop evolution and return the best solution.

4.5. Performance validation

To evaluate the performance of the proposed hybrid optimizer, 18 benchmark functions [25,32–37] given in Appendix A with input dimensions ranging from 2 to 30 are adopted for simulations using control parameters listed in Table 1. Particle number of the proposed approach is chosen as \(P = 10\), while conventional PSO as of Eqs. (25) and (26) and GA adopting float representation, tournament selection, non-uniform mutation, and arithmetic crossover have a population size of 20 for evolution. For each benchmark function, 100 runs are conducted by an individual optimization method. An initial population is randomly generated from within the search space of the optimization problem. Computation environment is based on a PC with Intel® Core™ 2 CPU @ 2.4G and memory capacity of 2GB under Matlab 7.0.1. Simulation results by optimizing the 18 benchmark functions via various approaches, including NM, PSO, GA, and NM-PSO-C, are listed in Table 2, where NM, PSO, and GA have achieved a successful rate of 64%, 80%, and 81%, respectively, for the 18 benchmark functions, while the proposed NM-PSO-C achieves 100% successful rate for all the benchmark functions with fewer function evaluations than the conventional PSO and GA in most
cases. Although the proposed NM-PSO-C requires more function evaluations than conventional PSO does for Fun. 2 and Fun. 18, the successful rate via NM-PSO-C is far better than the PSO. As far as accuracy is concerned, we define it as the deviation between the analytic global optimum and the averaged best objective function value ever searched by the algorithm with success over 100 runs.

Table 2 reveals the averaged best objective function value, successful rate, and the needed function evaluations of the algorithms under the constraint of 10,000 function evaluations. According to Table 2, the performance of the evolutionary algorithms indicated by parenthesis might not be as good as presented in the table. As demonstrated in the table, the proposed hybrid optimization scheme, NM-PSO-C, has better optimization accuracy in most cases, out-performing the other optimization schemes.

### 5. Digital redesign of interval systems via the hybrid particle swarm optimizer

Over the past years, evolutionary algorithms have been widely applied to design controllers to improve system performance with great success [38–41]. In what follows, we will propose an evolutionary approach via the proposed hybrid PSO optimizer to derive an optimal digital controller \( C_d(z) \) based on resemblance of the extremal GM/PM for the redesigned digital system and its continuous counterpart.

#### 5.1. Derivation of digital controllers based on extremal GM/PM

Fig. 8 shows the evolutionary approach to derive an optimal digital controller for redesigned digital systems based on deviation of extremal GM/PM between the redesigned digital system and its continuous counterpart. First of all, 32 extremal systems \( G_{\text{PM}}(s, \lambda) \) shown in Eq. (12) associated with the 32 extremal controlled systems \( G_{\text{PS}}(s) \) and the analog controller \( C(s) \) need to be established for calculating the extremal GM/PM as a basis for comparison. Subsequently, discretize the 32 extremal controlled systems \( G_{\text{PS}}(s) \) as discrete equivalent plants \( G_{\text{PM}}(z) \), based on which a digital controller \( C_d(z) \) is designed via the proposed evolutionary approach. It is hoped that extremal GM/PM for the redesigned digital system and its continuous counterpart are closely matched with the use of the digital controller derived by the hybrid optimizer NM-PSO-C.

Assume that the digital controller has the form:

\[
C_d(z, u, v) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}.
\]

where \( u = [u_0, u_1, \ldots, u_m] \) and \( v = [v_0, v_1, \ldots, v_n] \) represent the coefficients of the numerator and denominator of the digital controller, respectively.

Table 2 Simulation results of NM, PSO, GA and NM-PSO-C for 18 test functions.

<table>
<thead>
<tr>
<th>Test functions</th>
<th>Successful rate (%)</th>
<th>Function evaluations</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NM</td>
<td>PSO</td>
<td>GA</td>
</tr>
<tr>
<td>Fun1</td>
<td>61</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Fun2</td>
<td>31</td>
<td>50</td>
<td>70</td>
</tr>
<tr>
<td>Fun3</td>
<td>100</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td>Fun4</td>
<td>48</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>Fun5</td>
<td>72</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td>Fun6</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Fun7</td>
<td>27</td>
<td>95</td>
<td>100</td>
</tr>
<tr>
<td>Fun8</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Fun9</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Fun10</td>
<td>83</td>
<td>100</td>
<td>55</td>
</tr>
<tr>
<td>Fun11</td>
<td>36</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>Fun12</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Fun13</td>
<td>100</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>Fun14</td>
<td>98</td>
<td>100</td>
<td>98</td>
</tr>
<tr>
<td>Fun15</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Fun16</td>
<td>0</td>
<td>0</td>
<td>80</td>
</tr>
<tr>
<td>Fun17</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Fun18</td>
<td>0</td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td>Avg. successful rate</td>
<td>64%</td>
<td>80%</td>
<td>81%</td>
</tr>
</tbody>
</table>
controller. Let X be a particle representing a set of parameters for the digital controller:

\[ X(u, v) = [u_0, u_1, \ldots, u_m, v_0, v_1, \ldots, v_m]. \]

Initial particles are randomly generated from within the predefined range:

\[ u_j \in [u^-_j, u^+_j], \quad v \in [v^-_j, v^+_j], \quad j = 0, 1, \ldots, m. \]

To accelerate the derivation process, one of the initial particles is replaced by a seeded particle obtained by converting the analog controller \( C(s) \) into \( C_d(z) \) via an existing discretization method.

### 5.2. Fitness function

Based on the extremal GM/PMs, we can now define a fitness function to evaluate particles in a swarm. As shown in Fig. 8, extremal GM/PM of the continuous system \( C(s)G_p(s) \) can be calculated by Eqs. (12)–(24), in which \( GM_{\text{upper}}, PM_{\text{upper}}, GM_{\text{lower}}, \) and \( PM_{\text{lower}} \) represent the largest gain margin, smallest gain margin, largest phase margin, and smallest phase margin, respectively. On the other hand, 32 pulse transfer functions \( G(z) = C_d(z)G_pG(z) \) of the redesigned digital system can be obtained by combining the discretized extremal digital controller and digital controller \( C_d(z, u, v) \). Based on the derived pulse transfer functions \( G(z) = C_d(z)G_pG(z) \), extremal GM/PM of the redesigned digital system can be obtained, in which \( GM_{\text{upper}}, GM_{\text{lower}}, PM_{\text{upper}}, PM_{\text{lower}} \) represent the largest gain margin, smallest gain margin, largest phase margin, and smallest phase margin, respectively. To this end, we can define a set of cost functions based on the deviation between the extremal GM/PMs of the redesigned digital system and its continuous counterpart for a digital controller \( C_d(z, u, v) \):

\[
J_1(u, v) = \begin{cases} 
PM_{\text{upper}} - PM_{\text{upper}}, & PM_{\text{upper}} < PM_{\text{upper}} \\
\kappa \times PM_{\text{upper}} - PM_{\text{upper}}, & PM_{\text{upper}} > PM_{\text{upper}} \\
0, & \text{else}
\end{cases}
\]

\[
J_2(u, v) = \begin{cases} 
PM_{\text{lower}} - PM_{\text{upper}}, & PM_{\text{lower}} < PM_{\text{upper}} \\
\kappa \times PM_{\text{lower}} - PM_{\text{upper}}, & PM_{\text{lower}} > PM_{\text{upper}} \\
0, & \text{else}
\end{cases}
\]

\[
J_3(u, v) = \begin{cases} 
GM_{\text{upper}} - GM_{\text{upper}}, & GM_{\text{upper}} < GM_{\text{upper}} \\
\kappa \times GM_{\text{upper}} - GM_{\text{upper}}, & GM_{\text{upper}} > GM_{\text{upper}} \\
0, & \text{else}
\end{cases}
\]

\[
J_4(u, v) = \begin{cases} 
GM_{\text{lower}} - GM_{\text{lower}}, & GM_{\text{lower}} < GM_{\text{lower}} \\
\kappa \times GM_{\text{lower}} - GM_{\text{lower}}, & GM_{\text{lower}} > GM_{\text{lower}} \\
0, & \text{else}
\end{cases}
\]

Note that \( J_1, J_2, J_3, J_4 \) represent the weighted deviation of the largest phase margin, smallest phase margin, largest gain margin, and smallest gain margin, respectively between the redesigned digital system and its continuous counterpart. A weighting factor \( \kappa \) is included in these cost functions to penalize particles resulting in extremal GM/PM deviating from the desired boundary. Generally, \( \kappa > 1 \) is adopted so that extremal GM/PM boundaries of the redesigned digital system preferably lie within the corresponding extremal GM/PM boundaries of its continuous counterpart. Care must be taken that optimization of Eqs. (37)–(40) is a multi-objective problem with possible conflicting objectives that must be simultaneously achieved in designing the digital controller. To solve this problem, one of the simplest and natural ways is to have it reformulated as a mono-objective optimization problem by means of an aggregating function as:

\[
\min \sum_{j=1}^{4} w_j \cdot J_j(u, v)
\]

(41)

Subject to \( G_d(z, u, v, a, b) \) is robust Schur stable.

where \( w_j \) is the weighting coefficient, \( J_j(u, v) \) is the cost function defined in Eqs. (37)–(40), and \( G_d(z, u, v, a, b) \) is the closed-loop transfer function in the discrete-time domain defined in Eq. (7). Define a fitness function \( \text{Fit}(X) \) for particles \( X \) in a swarm as:

\[
\text{Fit}(X) = \begin{cases} 
1 & \text{particle } X(u, v) \text{ is feasible} \\
0 & \text{particle } X(u, v) \text{ is not feasible}
\end{cases}
\]

(42)

where \( K_j \) is a dynamic weighting constant for normalizing the individual objectives for preventing significant deviation in magnitudes between gain and phase margins. Without the normalization mechanism, the optimization process might be biased toward one of the objectives, thus preventing the optimization algorithm from obtaining a desired solution. To establish suitable \( K_j \), we first calculate the objective function values of \( J_1, J_2, J_3, J_4 \) for each particle in the current generation to obtain \( s_1 = \sum f_i^+ J_i(X_i), \ s_2 = \sum f_i^- J_i(X_i), \ s_3 = \sum f_i^{+-} J_i(X_i), \) and \( s_4 = \sum f_i^{--+} J_i(X_i), \) where \( P \) is the population size. The dynamic weighting constant \( K_j \) can then be established as follows:

\[
K_j = \frac{1}{M} \times \frac{\sum_{i=1}^{M} s_i}{j}, \quad j = \{1, 2, 3, 4\}
\]

(43)

where \( M \) is the number of objectives, currently \( M = 4 \). Particles result in better fitness whenever \( GM_{\text{upper}}, GM_{\text{lower}}, PM_{\text{upper}}, \) and \( PM_{\text{lower}} \) get closer to \( GM_{\text{upper}}, GM_{\text{lower}}, PM_{\text{upper}}, \) and \( PM_{\text{lower}} \), respectively, and vice versa. Particles resulting in instability are regarded as infeasible during the evolution process, for which an inferior fitness will be given. One of the simplest ways to handle the constraint violation is to define a penalty function \( \phi(u, v) \) as the summation of magnitude of unstable roots \( r_i \) of the characteristic equation in Eq. (7) lying outside the unit circle as:

\[
\phi(u, v) = \sum_{i=1}^{k} (\text{abi}(r_i) - 1)
\]

(44)

where \( k \) is the number of roots outside the unit circle in the discrete-time domain. It is clear that particles resulting in more unstable roots with larger magnitude receive heavier penalties, and vice versa. Based on the evolution scheme of the proposed hybrid optimizer, optimization continues toward a promising search direction to derive an optimal digital controller.

### 6. Illustrated examples

As mentioned earlier, practical systems are most suitably represented by continuous-time parametric interval models, via either analytical derivation or system identification in frequency/time domain. It is therefore natural that a lot of real-world applications, for example, a two-stage operational amplifier [42] considering fluctuation of parameters during the fabrication process, a DC motor driving a viscously damped inertial load [45], a position control system [44], a typical optical disk drive (ODD) tracking-following servo system implemented on a DVD ROM [46], and an Oblique Wing Aircraft (OWA) which has a wing that pivots [4], etc., are modeled as interval plants. Despite their simplicity in formulation, uncertain models in the form of interval systems have provided a convenient way in constructing mathematical models for physical systems, based on which feasible design can be achieved for use in industry. In this paper,
two examples, where the controlled process of the real-world applications is modeled as an interval plant are illustrated to show the performance of the digital controller evolutionarily derived by the proposed hybrid optimizer.

**Example 1.** Consider the feedback control system shown in Fig. 1, where the plant is described by the interval transfer function [46]:

\[ G_p(s) = [0.09, 0.11]s^4 + [0.9, 1.2]s^3 + [0.75, 1.2]s + [0.05, 0.25] \]

An analog controller

\[ C(s) = \frac{0.6804s + 1}{0.0474s + 1} \]

is designed for the interval plant, such that the resulting system has extremal GM/PM of:

\[ [\text{GM}_\text{lower}, \text{GM}_\text{upper}] = [16.104\, \text{dB}, 23.751\, \text{dB}] \]

and

\[ [\text{PM}_\text{lower}, \text{PM}_\text{upper}] = [45.2753^\circ, 66.9909^\circ] \]

respectively. For a sampling time \( T = 0.1 \), determine a digital controller \( C_d(z) = (v_0 + v_1z)/(u_0 + u_1z) \), such that the extremal GM/PM associated with the redesigned digital system in Fig. 3 closely match that of its continuous counterpart.

**[Solution]:**

By using the proposed NM-PSO-C adopting control parameters of population size = 30, max generation = 500, inertia weight \( w = 0.4 \), \( C_1 = C_2 = 2 \), continuous expansion coefficient \( \lambda = 2 \), modified shrink-age \( \sigma = 0.1 \), \( \alpha = 1 \), \( \gamma = 2 \), \( \beta = 0.5 \), \( \delta = 0.5 \), and search space of \([-10,10] \) for each of the controller parameters, an optimal digital controller:

\[ C_d(z) = 9.94318z - 8.167 \]

is evolutionarily obtained for \( T = 0.1 \) via the proposed approach. For comparison purpose, extremal GM/PM of the redesigned digital system associated with various digital controllers are listed in Table 3. As clearly indicated in Table 3, there is a significant deviation on extremal GM/PM between the redesigned digital system and its continuous counterpart by using conventional open-loop discretization methods, for example, ZOH and Tustin methods. The digital controller derived by the proposed hybrid PSO optimizer, on the other hand, results in extremal GM/PM of \([17.8608 \, \text{dB}, 22.7236 \, \text{dB}] \) and \([45.6482^\circ, 66.3518^\circ] \), respectively, for the redesign digital system, which closely match \([16.104\, \text{dB}, 23.751\, \text{dB}] \) and \([45.2753^\circ, 66.9909^\circ] \) of its continuous counterpart, out-performing those obtained via the conventional open-loop discretization methods. For verification, frequency envelope of the redesigned digital system is illustrated in Fig. 9, where the extremal GM/PM coincide with that revealed in Table 3.

**Table 3**

<table>
<thead>
<tr>
<th>Method</th>
<th>ZOH</th>
<th>Tustin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Controller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extremal GM</td>
<td>1.7712 dB, 8.3693 dB</td>
<td>8.5753 dB, 15.4088 dB</td>
</tr>
<tr>
<td>Extremal PM</td>
<td>9.2923, 48.7159]</td>
<td>34.6461, 60.4545</td>
</tr>
<tr>
<td>Digital controller derived by NM-PSO-C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example 2.** Consider the feedback control system shown in Fig. 1, where the plant is described by the interval transfer function [47]:

\[ G_p(s) = [0.0032, 0.005]s^3 + [0.072, 0.1]s^2 + [1.28, 1.305]s \]

An analog controller

\[ C(s) = \frac{0.4801s + 1.4182}{s} \]

is designed for the interval plant, such that the resulting system has extremal GM/PM of:

\[ [\text{GM}_\text{lower}, \text{GM}_\text{upper}] = [5\, \text{dB}, 12.8315\, \text{dB}] \]

and

\[ [\text{PM}_\text{lower}, \text{PM}_\text{upper}] = [31.78^\circ, 43.8199^\circ] \]

respectively. For a sampling time \( T = 0.05 \), determine a digital controller \( C_d(z) = v_0 + v_1z/(u_0 + u_1z) \), such that the extremal GM/PM associated with the redesigned digital system in Fig. 3 closely match that of its continuous counterpart.

**[Solution]:**

By using the proposed NM-PSO-C adopting the same control parameters as those in Example 1, an optimal digital controller:

\[ C_d(z) = -1.12709z + 0.1932 \]

\[ -9.9386z + 7.6243 \]

is derived.
is evolutionarily obtained for $T = 0.005$ via the proposed approach. Simulation results of extremal GM/PM of the redesigned digital system associated with various digital controllers are listed in Table 4, which clearly indicates that digital controller derived by the proposed hybrid optimizer has a better system performance in comparison to the conventional open-loop discretization methods.

7. Conclusions

The performance to digitally redesign a continuous-time system having an interval plant via conventional open-loop discretization methods is generally far from satisfactory. Closed-loop discretization methods, on the other hand, encountered the problem of optimizing non-convex functions with serious nonlinear couplings of the uncertain parameters. To solve this difficult problem, this paper formulates the design problem as an optimization problem subject to robust stability constraint and subsequently solved by a proposed evolutionary approach based on a hybrid optimizer incorporating PSO and an enhanced NM simplex search scheme to derive an optimal digital controller for the redesigned digital system. There is no restrictive condition under which the proposed approach is developed. In general, the optimal controller can be obtained within a moderate number of iterations by using the proposed evolutionary approach. As demonstrated in this paper, the proposed approach provides a simple yet practical way in the design of an optimal digital controller for redesigned digital systems having an interval plant without suffering from the inherent shortcomings of the conventional discretization methods.

Acknowledgement

This work was partially supported by the National Science Council, Taiwan, under Grant NSC 96-2221-E-032-047.

Appendix A. List of test functions

Function 1 (2 variables): BZ function
(a) $f(x) = x_1^2 + x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$;
(b) Search Domain : $-100 < x_1 < 100$, $j = 1.2$;
(c) Global optimum with $f = 0$ at $(0,0)$.

Function 2 (2 variables): Freudenstein–Roth function
(a) $f(x) = [-13 + x_1 + ((5 - x_2)x_2 - 2)x_2^2]
+ [-29 + x_1 + ((x_2 + 1)x_2 - 14)x_2^2]$;
(b) Search domain : $-10 < x_2 < 10$, $j = 1.2$;
(c) Global optimum with $f = 0$ at $(5,4)$.

Function 3 (2 variables): Branin ROOC (RC) function
(a) $f(x) = (x_2 - (5/(4\pi^2))x_1^2 + (5/\pi)x_1 - 6)^2
+ 10(1 - (1/(8\pi))\cos(x_1)) + 10$;
(b) Search domain : $-5 < x_2 < 10$, $0 < x_2 < 15$;
(c) Global optimum with $f = 0.397887$ at $(-\pi, 12.275), (\pi, 2.275), (9.42478, 2.475)$.

Function 4 (2 variables): Eason (ES) function
(a) $f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 + (x_2 - \pi)^2)$;
(b) Search domain : $-100 < x_1 < 100$, $j = 1.2$;
(c) Global optimum with $f = -1$ at $(-\pi, \pi)$.

Function 5 (2 variables): Goldstein and Price (GP) function
(a) $f(x) = [1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2
+ 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_1^2)x_2
+ 12x_1^2 + 48x_1 - 36x_1x_2 + 27x_2^2];$
(b) Search domain : $-2 < x_1 < 2$, $j = 1.2$;
(c) Global optimum with $f = 3$ at $(0, -1)$.

Function 6 (2 variables): Rosenbrock function
(a) $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$;
(b) Search domain : $-5 < x_1 < 10$, $j = 1.2$;
(c) Global optimum with $f = 0$ at $(1,1)$.

Function 7 (2 variables): Shubert (SH) function
(a) $f(x) = \left(\sum_{j=1}^{5} j\cos((j + 1)x_1 + j)\right)
\times \left(\sum_{j=1}^{5} j\cos((j + 1)x_2 + j)\right)$;
(b) Search domain : $-10 < x_1 < 10$, $j = 1.2$;
(c) Global optimum at $f = 0$ at $(0,0)$.

Function 8 (2 variables): Zakharov (Z) function
(a) $f(x) = \left(\sum_{j=1}^{n} x_j^2\right) + \left(\sum_{j=1}^{n} 0.5x_j\right)^2
+ \left(\sum_{j=1}^{n} 0.5x_j\right)^4$;
(b) Search domain : $-5 < x_1 < 10$, $j = 1.2$;
(c) Global optimum with $f = 0$ at $(0,0)$.

Function 9 (3 variables): De Jong function
(a) $f(x) = x_1^2 + x_2^2 + x_3^2$;
(b) Search domain : $-5 < x_1 < 5$, $j = 1.2, 3$;
(c) Global optimum with $f = 0$ at $(0,0,0)$.

Function 10 (3 variables): Hartmann (H3) function
(a) $f(x) = \frac{-4}{3} \sum_{i=1}^{4} c_i \exp\left[-3 \sum_{j=1}^{3} a_i(x_j - p_j)^2\right]$;
(b) Search domain : $0 < x_j < 1$, $j = 1.2, 3$;
(c) Global optimum with $f = -3.863433$ at $(0.1146, 0.055, 0.852)$.

Function 11 (4 variables): Trigonometric function
(a) $f(x) = 4 - \sum_{j=1}^{4} \cos x_1 + i(1 - \cos x_1) - \sin x_2$;
(b) Search domain : $-10 < x_1 < 10$, $j = 1.2, 3, 4$;
(c) Global optimum with $f = 0$ at $(0,0,0,0)$.

Function 12 (4 variables): Variably dimensioned function
(a) $f(x) = \sum_{j=1}^{4} i(x_1 - 1)^2 + \sum_{j=1}^{4} i(x_1 - 1)^2
+ \sum_{j=1}^{4} i(x_1 - 1)^4$;
(b) Search domain : $-10 < x_1 < 10$, $j = 1.2, 3, 4$;
(c) Global optimum with $f = 0$ at $(1,1,1,1)$.

Function 13 (4 variables): Colville function
(a) $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_2^2)^2 + (1 - x_3)^2
+ 10.1(2x_2^2 + x_1 - 1)^2 + (x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$;
(b) Search domain : $-10 < x_1 < 10$, $j = 1.2, 3, 4$;
(c) Global optimum with $f = 0$ at $(1,1,1,1)$.
References
[41] C.-C. Hsu, S.-C. Chang, Tolerance design of robust controllers for uncertain interval systems based on evolutionary algorithms, IET Control Theory & Applications 1 (January) (2007) 244–252 (SCI).