International Journal of Innovative Computing, Information and Control
Volume **12**, Number **3**, June **2016**
ICIC International ©2016 ISSN 1349-4198
pp. 795–808

DESIGN OF FINITE-LEVEL DYNAMIC QUANTIZERS BY USING COVARIANCE MATRIX ADAPTATION EVOLUTION STRATEGY

Juan Esteban Rodriguez Ramirez, Yuki Minami and Kenji Sugimoto

Graduate School of Information Science
Nara Institute of Science and Technology
8916-5, Takayama-cho, Ikoma, Nara 630-0192, Japan
{ juan.rodriguez.jj4; minami; kenji }@is.naist.jp

Received November 2015; revised March 2016

Abstract. *In networked control systems, there are two issues due to communication rate constraints. One is the quantization errors, and the other is the signal saturation. In order to tackle these issues, this paper focuses on the quantizer design problem which finds a dynamic quantizer that minimizes the performance degradation caused by the quantization errors and satisfies the signal saturation constraint. For the problem, this paper proposes an easy-to-use quantizer design method based on covariance matrix adaptation evolution strategy (CMA-ES), which is a state-of-the-art metaheuristic algorithm. Then, the effectiveness of the proposed method is verified by numerical experiments.*
Keywords: Networked control system, Dynamic quantization, Covariance matrix adaptation evolution strategy, Differential evolution, Particle swarm optimization

1. Introduction.

Networked control systems are dynamical systems where plants, controllers, sensors, and actuators are connected to each other via communication channels. Examples of networked control systems are found in industrial automation systems, large distributed systems such as smart grids, and so on [1, 2, 3]. In recent years, the networked control systems have received the attention of researchers and manufacturers because of their many advantages such as the increase of the flexibility and the scalability of the systems, and the reduction of the costs for installation and maintenance.

In the networked control systems, since control/sensor signals are transmitted over communication channels, continuous-valued signals are quantized into discrete-valued ones. The quantization error, which is the difference between the continuous-valued signal and its quantized version, leads to the performance degradation of the control systems. Thus, several studies have been carried out in order to minimize the performance degradation due to the quantization, and they have shown that an effective method is the use of dynamic quantizers, where the quantization error is fed back and filtered [4, 5, 6, 7].

Besides, there exists a limitation for the amount of data that can be transmitted per unit of time in the networked control systems, which means that the number of quantization levels is limited. A finite number of quantization levels may cause the saturation of the amplitude of the quantized signal. Such saturation problem has the potential to destabilize the systems [8, 9]. To overcome this, the design of dynamic quantizers with a finite number of quantization levels has been tackled in [10, 11, 12, 13]. However, there is no perfect method to design quantizers because the existing methods in [10, 11, 12, 13] may not give optimal solutions to the quantizer design problems. The reasons are that the relaxed problem of the original design problem solved in [10, 11] might give conservative solutions, and the metaheuristic algorithms used in [12, 13] may give local minima.

Motivated by the above background, this paper considers the design problem of finite-level dynamic quantizers that (i) minimize the system's performance degradation and (ii) satisfy the channel's data rate constraints. Then, we propose a reliable and easy-to-use quantizer design method. Since the design problem is formulated as the optimization of a non-linear and non-convex function, it cannot be directly solved by algebraic methods or conventional numerical optimization methods. Thus, one reasonable approach is the use of metaheuristic methods which are powerful tools to explore feasible solutions to optimization problems [14, 15]. This paper focuses on the covariance matrix adaptation evolution strategy (CMA-ES) algorithm [16, 17]. It is a state-of-the-art metaheuristic algorithm, and it shows very good performance in the optimization of multimodal functions. In addition, CMA-ES is an easy-to-use algorithm since most of the heuristic rules and parameters in the CMA-ES algorithm are automatically chosen and determined. For these reasons, CMA-ES is employed as a reliable optimization tool instead of particle swarm optimization (PSO) in [12] and differential evolution (DE) in [13].

The contributions of this paper are the following. First, the CMA-ES based design method of finite-level dynamic quantizers is proposed. Then, through numerical experiments, it is shown that the proposed method gives satisfactory dynamic quantizers. Furthermore, the CMA-ES based method is compared with other dynamic quantizer design methods based on PSO [12] and DE [13], and it is verified that the proposed method achieves better performance in terms of precision and convergence speed. Finally, this paper presents a novel application of the CMA-ES algorithm since there are a few applications in the control and systems field [18, 19, 20, 21, 22], and CMA-ES has not been used for quantizer design problems.

This paper is organized as follows. First, the quantizer design problem is formulated in Section 2, and the CMA-ES algorithm is introduced in Section 3. Then, the effectiveness of the CMA-ES based quantizer design method is verified with numerical experiments in Sections 4 and 5. Finally, Section 6 concludes this paper.

Notation: Let $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{N}$ denote the set of real numbers, the set of the positive real numbers, and the set of the natural numbers, respectively. For the matrix $\mathbf{A} := \{A_{ij}\}$, let abs($\mathbf{A}$) be abs($\mathbf{A}$) $:= \{|A_{ij}|\}$ and when $\mathbf{A}$ is a square matrix, let $\Lambda_i(\mathbf{A})$ represent the i^{th} eigenvalue of $\mathbf{A}$. For a vector $\boldsymbol{v}$, the expression $\|\boldsymbol{v}\|$ represents the euclidean norm of $\boldsymbol{v}$. Finally, $\mathbf{I}$ is the identity matrix, $\mathbf{0}$ is the null matrix, and $\mathsf{E}\|\mathbf{P}\|$ is the expected value of some probability distribution $\mathbf{P}$.

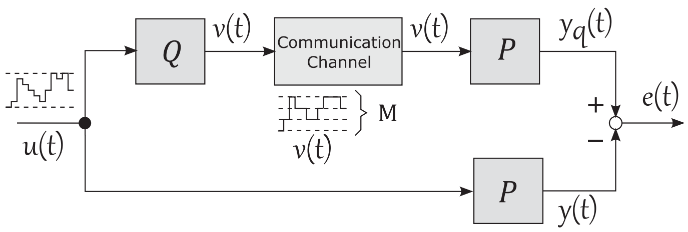


Figure 1. Error system

2. Problem Formulation.

Consider the error system shown in Figure 1, composed of the plant P, the quantizer Q, and the communication channel.

The discrete-time SISO plant P is given by

$$P : \begin{cases} \boldsymbol{x}(t+1) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}u(t), \\ y(t) = \mathbf{C}\boldsymbol{x}(t), \end{cases} \tag{1}$$

where $t \in \{0\} \cup \mathbb{N}$ is the discrete time, $\boldsymbol{x} \in \mathbb{R}^{n_P}$ is the state of P, $u \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ is the output, and $\mathbf{A} \in \mathbb{R}^{n_P \times n_P}$, $\mathbf{B} \in \mathbb{R}^{n_P \times 1}$ and $\mathbf{C} \in \mathbb{R}^{1 \times n_P}$ are constant matrices. The initial state is given by $\boldsymbol{x}(0) = \boldsymbol{x}_0$ for $\boldsymbol{x}_0 \in \mathbb{R}^{n_P}$, and we assume that all eigenvalues of $\mathbf{A}$ are inside the unit circle in the complex plane.

On the other hand, the feedback type dynamic quantizer Q [10, 11] is given by

$$Q : \begin{cases} \boldsymbol{\xi}(t+1) = \boldsymbol{\mathcal{A}}\boldsymbol{\xi}(t) + \boldsymbol{\mathcal{B}}(v(t) - u(t)), \\ v(t) = \mathrm{q}[\boldsymbol{\mathcal{C}}\boldsymbol{\xi}(t) + u(t)], \end{cases} \tag{2}$$

where $\boldsymbol{\xi} \in \mathbb{R}^{n_Q}$ is the state of Q, $v \in \{\pm d, \pm 2d, \ldots, \pm \frac{M}{2} d\}$ is the quantized output, $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_Q \times n_Q}$, $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{n_Q \times 1}$ and $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{1 \times n_Q}$ are constant matrices, and the initial state of the quantizer is given by $\boldsymbol{\xi}(0) = \mathbf{0}$. The static quantizer $\mathrm{q}[\cdot]$ rounds off the continuous-values to the nearest discrete-ones. The parameters of the static quantizer are the number of quantization levels $M \in \mathbb{N}$ and the quantization interval $d \in \mathbb{R}_+$. Figure 2 illustrates the static quantizer $\mathrm{q}[\cdot]$ in the case of $M = 6$. The design parameters of the dynamic quantizer Q are $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}$, $\boldsymbol{\mathcal{C}}$ and d since M is given so as to satisfy the data rate constraints.

For this system, this paper makes the following assumptions: the communication channel has no losses and no delays, and the input signal u is bounded, i.e., $u \in U$ for a given $U = [u_{\min}, u_{\max}]$.

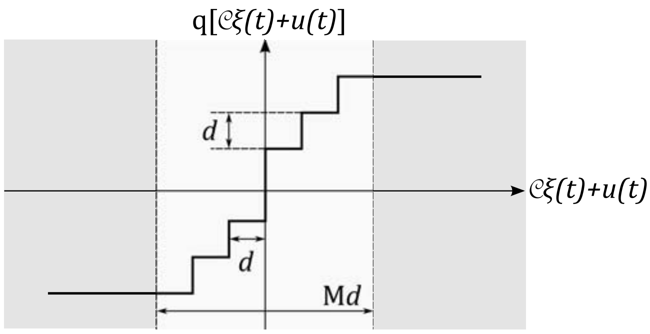


FIGURE 2. Example of a static quantizer function $\mathrm{q}[\cdot]$ ($M = 6$)

Now, the following *performance index* is introduced to evaluate the performance degradation of the system due to the quantization.

$$E(Q) := \sup_{\substack{u \in U \\ t \in \{1,2,\ldots,L\}}} \mathrm{abs}\left(y_q(t) - y(t)\right), \tag{3}$$

where $L \in \mathbb{N}$ is the evaluation interval, y_q indicates the output of the plant P whose input is the quantized signal v, and y is the output of the plant whose input is u. The performance index $E(Q)$ evaluates the maximum absolute value of $e(t) = y_q(t) - y(t)$, which corresponds to the worst case performance of the system. Thus, by minimizing $E(Q)$, the system composed of the plant P and the quantizer Q can be optimally approximated to the plant P, in terms of the input-output relation. Here, the value of $E(Q)$ can be calculated as follows [4].

$$E(Q) = \frac{d}{2} \sum_{t=0}^{L} \mathrm{abs}\left(\begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B}\boldsymbol{\mathcal{C}} \\ \mathbf{0} & \boldsymbol{\mathcal{A}} + \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{C}} \end{bmatrix}^t \begin{bmatrix} \mathbf{B} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} \right). \tag{4}$$

In order to obtain the smallest performance degradation caused by the quantization, it is necessary to make the right hand side of (4) as small as possible by the appropriate design of the dynamic quantizer Q.

Furthermore, the data rate constraint of the communication channel imposes a limitation in the design of dynamic quantizers. When the number of bits that can be transmitted through the channel per sampling time is N_b, the number of quantization levels M should be given under the relation:

$$M \leq 2^{N_b}. \tag{5}$$

Besides, the quantization interval d has to satisfy the following condition derived from Figure 2.

$$\mathrm{abs}\left(\mathcal{C}\boldsymbol{\xi}(t) + u(t)\right) \leq \frac{1}{2}Md. \tag{6}$$

The design of d under (6) is equivalent to a reachable set problem [10], which is not easy to solve. However, the smallest quantization interval d that satisfies (6) for $u \in U = [u_{\min}, u_{\max}]$ is given as follows [10].

$$d^* = (u_{\max} - u_{\min}) \left(M - \frac{\mathrm{abs}(\mathcal{C}\mathbf{T})\mathrm{abs}(\mathbf{T}^{-1}\mathcal{B})\bar{\Lambda}^L}{1-\bar{\Lambda}} - \sum_{t=0}^{L} \mathrm{abs}\left(\mathcal{C}(\mathcal{A}+\mathcal{B}\mathcal{C})^t\mathcal{B}\right) \right)^{-1}, \tag{7}$$

$$\bar{\Lambda} = \max_i \mathrm{abs}\left(\Lambda_i(\mathcal{A}+\mathcal{B}\mathcal{C})\right), \tag{8}$$

where $\mathbf{T} \in \mathbb{R}^{n_Q \times n_Q}$ is the matrix used for the diagonalization of the matrix $(\mathcal{A}+\mathcal{B}\mathcal{C})$.

Based on the above setting, the finite-level dynamic quantizer design problem is formulated as follows.

Problem 2.1. Suppose that the plant P, the number of quantization levels M and the input signal range U are given. Then, find the quantizer parameters $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and d which minimize $E(Q)$, under the conditions:

(i) Q is stable, i.e., $\bar{\Lambda} < 1$,
(ii) The data rate constraint is satisfied, i.e., $d^* > 0$.

The optimization problem considered here is non-linear and non-convex, and thus it cannot be directly solved by conventional methods such as linear programming or quadratic programming. For this reason a metaheuristic approach is used to solve the problem.

However, the metaheuristic algorithm considered in this paper has been designed to solve unconstrained problems. Therefore, the above constrained optimization problem is transformed into the following unconstrained version, based on the method in [23, 24].

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathrm{minimize}}\, f(\boldsymbol{x}), \qquad f(\boldsymbol{x}) := \begin{cases} h_{\max}(\boldsymbol{x}) & \text{if } h_{\max}(\boldsymbol{x}) \geq 0, \\ f_v(\boldsymbol{x}) & \text{otherwise,} \end{cases} \tag{9}$$

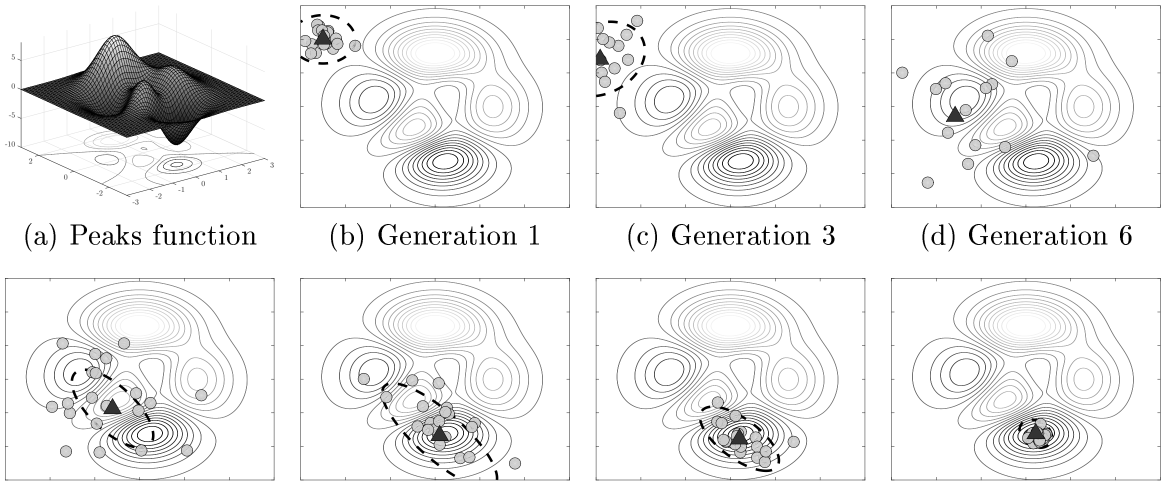
where

$$f_v(\boldsymbol{x}) := \arctan[E(\boldsymbol{x})] - \pi/2 \quad \text{with} \quad d = d^*(\boldsymbol{x}), \tag{10}$$

$$h_{\max}(\boldsymbol{x}) := \max\left[\bar{\Lambda}(\boldsymbol{x}) - 1,\ -d^*(\boldsymbol{x})\right]. \tag{11}$$

In these equations, the $E(\boldsymbol{x})$, $d^*(\boldsymbol{x})$ and $\bar{\Lambda}(\boldsymbol{x})$ correspond to (4), (7) and (8) respectively. Note that the design variable $\boldsymbol{x} \in \mathbb{R}^n$ is constructed with the n unknown elements of the matrices $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$.

3. **Covariance Matrix Adaptation Evolution Strategy (CMA-ES).** In this paper, the covariance matrix adaptation evolution strategy (CMA-ES) algorithm [16, 17] is adopted to solve the quantizer design problem. The CMA-ES algorithm is an evolutionary algorithm used for solving black-box optimization problems in continuous domains. The main advantage of CMA-ES over other metaheuristics is that most heuristic rules and control parameters are automatically chosen and determined. In fact, the only parameter left to the user is the number of candidate solutions N. By increasing N, the exploration



(a) Peaks function (b) Generation 1 (c) Generation 3 (d) Generation 6

(e) Generation 9 (f) Generation 12 (g) Generation 15 (h) Generation 18

FIGURE 3. Example of the operation of CMA-ES over a two dimensional search space. It shows how the $N = 20$ search points (gray dots) and the mean (black triangle) move in the search space through the generations. The break line represents a contour of the multivariate normal distribution with probability $p_1 = 0.1$, that helps to see the adaptation of the covariance matrix.

capabilities and robustness of CMA-ES are usually improved, while the convergence time increases.

In the CMA-ES algorithm, the candidate solutions, called search points, are generated randomly according to a multivariate normal distribution with mean $\boldsymbol{m}$ and covariance matrix $\boldsymbol{\Sigma}$. The initial value of $\boldsymbol{m}$ is provided by the user or it can be selected randomly inside the search space, and the initial value of $\boldsymbol{\Sigma}$ is given by $\boldsymbol{\Sigma} = \mathbf{I}$. Then, in each iteration of the algorithm, the best points are selected and the parameters of the normal distribution are updated. Thus, the mean $\boldsymbol{m}$ goes toward the best solution. In the next iteration the search points are generated randomly according to the normal distribution with the new parameters. An example of the operation of CMA-ES is shown in Figure 3 for a simple two dimensional optimization problem.

This paper uses the $(\mu/\mu_W, \lambda)$ CMA-ES version shown in Algorithm 1, which has been presented in [25].

Algorithm 1: $(\boldsymbol{\mu/\mu_W}, \boldsymbol{\lambda})$ CMA-ES

Initialization: Given $N \in \mathbb{N}$, $k_{\max} \in \mathbb{N}$, $\boldsymbol{m} \in \mathbb{R}^n$, the step size $\sigma \in \mathbb{R}_+$ and the initial search space $S = [x_{\min}, x_{\max}]^n$. Initialize $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$, $\boldsymbol{p}_\sigma \in \mathbb{R}^n$ and $\boldsymbol{p}_c \in \mathbb{R}^n$ as $\boldsymbol{\Sigma} = \mathbf{I}$, $\boldsymbol{p}_\sigma = \mathbf{0}$ and $\boldsymbol{p}_c = \mathbf{0}$, respectively. Set the values of the parameters c_c, c_σ, c_1, c_μ, d_σ, μ_{eff} and w_i $(i = 1, 2, \ldots, N)$ to their default values shown in Appendix A. Then, $k = 0$.

Step 1 (Sample new population): N search points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ are generated randomly from the multivariable normal distribution $\mathcal{N}(\boldsymbol{m}, \sigma^2\boldsymbol{\Sigma})$ as follows

$$\boldsymbol{x}_i = \boldsymbol{m} + \sigma\boldsymbol{y}_i, \quad \boldsymbol{y}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \quad \text{for } i = 1, 2, \ldots, N. \tag{12}$$

Step 2 (Selection and recombination): The objective function $f(\boldsymbol{x}_i)$ is evaluated for each $\boldsymbol{x}_i$, then the sets $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ and $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}$ are ordered based on the fitness value of $\boldsymbol{x}_i$. The ones with the best fitness go at the beginning. The first μ search points are the parents of the next generation. They are combined with each other to generate the new mean as follows

$$\boldsymbol{m} = \sum_{i=1}^{\mu} w_i \boldsymbol{x}_i = \boldsymbol{m} + \sigma \boldsymbol{y}_w, \tag{13}$$

$$\boldsymbol{y}_w = \sum_{i=1}^{\mu} w_i \boldsymbol{y}_i. \tag{14}$$

Step 3 (Step size control):

$$\boldsymbol{p}_\sigma \leftarrow (1 - c_\sigma)\boldsymbol{p}_\sigma + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\text{eff}}}\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{y}_w, \tag{15}$$

$$\sigma \leftarrow \sigma \times \exp\left[\frac{c_\sigma}{d_\sigma}\left(\frac{\|\boldsymbol{p}_\sigma\|}{\mathsf{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1\right)\right], \tag{16}$$

where $\mathsf{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\| \approx \sqrt{n}\,(1 - 1/4n + 1/21n^2)$.

Step 4 (Covariance matrix adaptation):

$$h_\sigma = \begin{cases} 1 & \text{if } \frac{\|\boldsymbol{p}_\sigma\|}{\sqrt{1-(1-c_\sigma)^{2(k+1)}}} < \left(1.5 + \frac{1}{n-0.5}\right)\mathsf{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|, \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

$$\boldsymbol{p}_c \leftarrow (1 - c_c)\boldsymbol{p}_c + h_\sigma\sqrt{c_c(2 - c_c)\mu_{\text{eff}}}\boldsymbol{y}_w, \tag{18}$$

$$\boldsymbol{\Sigma} \leftarrow (1 - c_1 - c_\mu)\boldsymbol{\Sigma} + c_1\left(\boldsymbol{p}_c\boldsymbol{p}_c^{\mathrm{T}} + (1 - h_\sigma)c_c(2 - c_c)\boldsymbol{\Sigma}\right) + c_\mu\sum_{i=1}^{\mu} w_i \boldsymbol{y}_i \boldsymbol{y}_i^{\mathrm{T}}. \tag{19}$$

Step 5 (Check stop condition): If $k < k_{\max}$, $k \leftarrow k+1$ and go to **Step 1**; otherwise, terminate the algorithm and return $\boldsymbol{m}$ (or $\boldsymbol{x}_1$).

In the $(\mu/\mu_W, \lambda)$ strategy, μ is the number of parents of the next generation, μ_W indicates a weighted recombination of the parents and λ is the number of search points. Note that λ is represented by N in this paper. In addition, the default setting of the control parameters has been given in [28], and the parameters $\boldsymbol{m}$, $\boldsymbol{\Sigma}$, and σ are automatically updated in the algorithm. Thus, it is expected that the algorithm works well without the careful tuning of the initial values of the parameters. This makes the CMA-ES an easy-to-use algorithm.

4. **Numerical Experiments.** Consider the system in Figure 1. Then, we consider two continuous-time plants:

$$P_1(s) = \frac{s + 20}{s^2 + 3s + 2}, \tag{20}$$

$$P_2(s) = \frac{s + 10}{s^3 + 6s^2 + 9s + 10}. \tag{21}$$

Note that the discrete-time plants in the form of (1) are obtained from $P_1(s)$, $P_2(s)$, and the sampling time $\Delta t = 0.1[s]$.

For P_1 and P_2, the forms of the matrices of the quantizer are given by

$$\mathcal{A}_1 = \begin{bmatrix} 0 & 1 \\ x_1 & x_2 \end{bmatrix}, \quad \mathcal{B}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathcal{C}_1 = \begin{bmatrix} x_3 & x_4 \end{bmatrix}, \tag{22}$$

$$\mathcal{A}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ x_1 & x_2 & x_3 \end{bmatrix}, \quad \mathcal{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathcal{C}_2 = \begin{bmatrix} x_4 & x_5 & x_6 \end{bmatrix}, \tag{23}$$

respectively, which are called the canonical controllable form. Then, the search points are formed like $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^{\mathrm{T}}$ for P_1, and $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{bmatrix}^{\mathrm{T}}$ for P_2.

Thus, the dimensions of the optimization problems are four and six, respectively. The initial values for the CMA-ES algorithm are $\sigma = 0.3$, $S = [-1, 1]^n$ and $\boldsymbol{m}$ is selected uniformly randomly within S.

The simulations are performed by trying $N_{run} = 50$ runs of the algorithm for each combination of the parameters $N = \{50, 100, 500\}$ and $k_{\max} = \{50, 100, 200, 500, 1000\}$. After performing the simulations with $U = [-1, 1]$, $M = 2$ and $L = 150$, it was found that the optimal quantizers for P_1 and P_2 are given by

$$\mathcal{A}_1 = \begin{bmatrix} 0 & 1 \\ -0.7413 & 1.7241 \end{bmatrix}, \quad \mathcal{B}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathcal{C}_1 = \begin{bmatrix} 0.6532 & -1.1619 \end{bmatrix}, \quad d_1 = 3.1082, \tag{24}$$

and

$$\mathcal{A}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.1391 & -1.1584 & 2.0022 \end{bmatrix}, \quad \mathcal{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathcal{C}_2 = \begin{bmatrix} -0.2078 & 0.9785 & -1.1550 \end{bmatrix},$$
$$d_2 = 3.2337. \tag{25}$$

The results of the simulations are summarized in Table 1. For each plant and each combination of N and $k_{\max}$, the best value of $E(Q)$, the mean and the standard deviation of $E_{\min}(Q)$, and the success rate SR in percentage [%] are shown. The success rate is the ratio of the number of runs with the best solution to the total number of runs of the algorithm N_{run}.

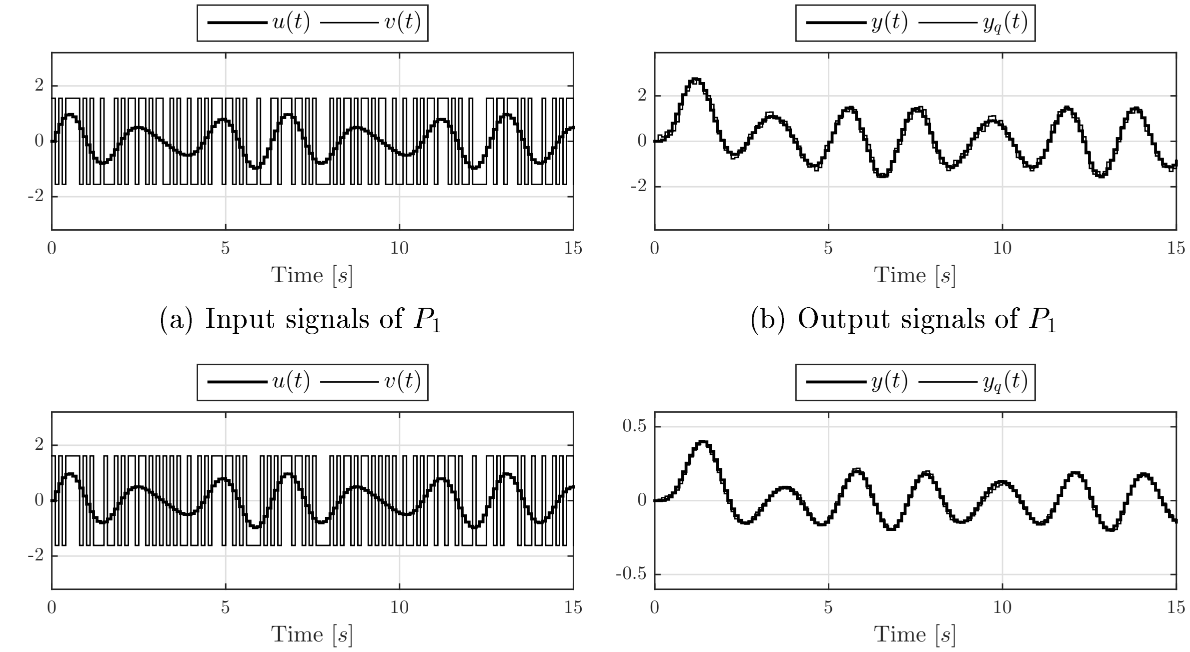
Table 1. Simulation results for the second and third order plants by CMA-ES ($N_{run} = 50$ trials)

		second order system, P_1				third order system, P_2			
N	$k_{\max}$	Best	Mean	St. dev.	SR	Best	Mean	St. dev.	SR
50	50	0.5091	0.5870	0.1059	66	0.0719	0.0948	0.0335	0
50	100	0.5091	0.5395	0.0814	86	0.0691	0.0797	0.0302	82
50	200	0.5091	0.5800	0.1121	70	0.0691	0.0751	0.0238	94
50	500	0.5091	0.5438	0.0861	86	0.0691	0.0733	0.0200	92
50	1000	0.5091	0.5736	0.1089	74	0.0691	0.0814	0.0328	86
100	50	0.5091	0.5152	0.0353	98	0.0692	0.0704	0.0009	38
100	100	0.5091	0.5091	0.0000	100	0.0691	0.0691	0.0000	100
100	200	0.5091	0.5140	0.0347	98	0.0691	0.0691	0.0000	100
100	500	0.5091	0.5140	0.0347	98	0.0691	0.0691	0.0000	96
100	1000	0.5091	0.5190	0.0486	96	0.0691	0.0691	0.0000	98
500	50	0.5091	0.5091	0.0000	100	0.0691	0.0691	0.0000	98
500	100	0.5091	0.5091	0.0000	100	0.0691	0.0691	0.0000	100
500	200	0.5091	0.5091	0.0000	100	0.0691	0.0691	0.0000	100
500	500	0.5091	0.5091	0.0000	100	0.0691	0.0691	0.0000	100
500	1000	0.5091	0.5091	0.0000	100	0.0691	0.0691	0.0000	100

To evaluate that the designed quantizer works properly, the control signal:

$$u(t) = 0.7 \sin(3t) + 0.3 \sin(4t) \tag{26}$$

is applied to the error system in Figure 1. Note that $u \in U = [-1, 1]$. The results are shown in Figure 4: (a) and (b) show the time responses for the second order system P_1, and (c) and (d) show the results for the third order system P_2. Notice that in both cases the quantized input signal $v(t)$ has only two levels since it was specified that $M = 2$. Figure 4 shows how the quantized output $y_q(t)$ follows closely the desired



(a) Input signals of P_1 (b) Output signals of P_1

(c) Input signals of P_2 (d) Output signals of P_2

FIGURE 4. System responses when the control signal $u(t)$ is applied to the plants P_1 and P_2. The black thick lines represent the signals of the system without quantization. Meanwhile, the blue thin ones are the signals when the quantization is performed.

output $y(t)$ and the error between them is very small. In fact, the maximum error is $\max_{t\in\{1,2,\ldots,150\}} \mathrm{abs}\left(y_q(t) - y(t)\right) = 0.3462$. Then, since $E(Q) = 0.5091$ for the designed quantizer, it is verified that the maximum error in this example is less than the value of the performance index $E(Q)$. The same situation happens for the third order system, where $\max_{t\in\{1,2,\ldots,150\}} \mathrm{abs}\left(y_q(t) - y(t)\right) = 0.0319$ and $E(Q) = 0.0691$.

5. **Comparison with DE and PSO.** In order to evaluate the performance of the proposed finite-level dynamic quantizer design method, which is based on CMA-ES, further simulations were carried out. In those simulations, the quantizer design optimization problems in Section 4 are solved by using DE and PSO, since the previous studies [12, 13] have proposed the DE and the PSO based quantizer design methods.

Differential evolution (DE) is a population based metaheuristic algorithm inspired in the mechanism of biological evolution [26]. In this algorithm, the objective function $f(\boldsymbol{x})$ is evaluated iteratively over a population of search points $\boldsymbol{x}_i$, known as target vectors in the DE literature. In each iteration, the search points improve their values and move toward the best solution. Finally the search point with the best fitness value in the last iteration is regarded as the optimal solution. The DE algorithm is shown in Appendix B.

On the other hand, particle swarm optimization (PSO) is a population based metaheuristic algorithm that is inspired in the behavior of biological communities like swarms of bees and flocks of birds [27]. In the PSO algorithm, a search point is called *particle* and the set of all particles is called *swarm*. The PSO implementation used in this study is shown in Appendix B.

The only changes performed to the quantizer design problems in Section 4 are the parameters of the metaheuristic algorithms. In the case of DE, the control parameters are $F = 0.6$ and $H = 0.9$; in the case of PSO, the parameters are $\chi_0 = 0.9$, $\chi_1 = 1$ and $\chi_2 = 1$. These parameters are the same as those in the previous studies [12, 13].

The results are summarized in Table 2 for DE and Table 3 for PSO.

The comparison among design methods based on CMA-ES, DE and PSO is given in terms of the success rate and convergence behavior. All these data can be found in the Tables 1, 2 and 3.

For the second order system the best solutions found by CMA-ES and DE are the same, namely, $E(Q_{CMA\text{-}ES}) = E(Q_{DE}) = 0.509057$, while the best solution found by

Table 2. Simulation results for the second and third order plants by DE ($N_{run} = 50$ trials)

		second order system, P_1				third order system, P_2			
N	$k_{\max}$	Best	Mean	St. dev.	SR	Best	Mean	St. dev.	SR
50	50	0.5091	0.6051	0.1197	62	0.0692	0.1113	0.0461	4
50	100	0.5091	0.5885	0.1158	68	0.0691	0.0949	0.0419	30
50	200	0.5091	0.5984	0.1191	64	0.0691	0.0963	0.0433	32
50	500	0.5091	0.5896	0.1153	68	0.0691	0.0966	0.0432	44
50	1000	0.5091	0.5741	0.1086	74	0.0691	0.0838	0.0345	62
100	50	0.5091	0.5641	0.1026	78	0.0692	0.0871	0.0358	28
100	100	0.5091	0.5885	0.1158	68	0.0691	0.0817	0.0324	68
100	200	0.5091	0.5666	0.1023	78	0.0691	0.0853	0.0366	80
100	500	0.5091	0.5488	0.0910	84	0.0691	0.0832	0.0347	84
100	1000	0.5091	0.5438	0.0861	86	0.0691	0.0852	0.0367	84
500	50	0.5091	0.5190	0.0486	96	0.0692	0.0745	0.0198	76
500	100	0.5091	0.5438	0.0861	86	0.0691	0.0711	0.0140	98
500	200	0.5091	0.5425	0.0813	86	0.0691	0.0712	0.0140	96
500	500	0.5091	0.5438	0.0861	86	0.0691	0.0791	0.0300	90
500	1000	0.5091	0.5438	0.0861	86	0.0691	0.0691	0.0000	100

Table 3. Simulation results for the second and third order plants by PSO ($N_{run} = 50$ trials)

		second order system, P_1				third order system, P_2			
N	$k_{\max}$	Best	Mean	St. dev.	SR	Best	Mean	St. dev.	SR
50	50	0.7291	0.7910	0.0271	0	0.1386	0.2052	0.0340	0
50	100	0.6467	0.7612	0.0196	0	0.1297	0.1814	0.0157	0
50	200	0.7437	0.7577	0.0027	0	0.0867	0.1650	0.0182	0
50	500	0.7015	0.7556	0.0080	0	0.0713	0.1579	0.0300	0
50	1000	0.5731	0.7518	0.0274	2	0.0706	0.1529	0.0333	0
100	50	0.7300	0.7815	0.0211	0	0.1047	0.1867	0.0195	0
100	100	0.6960	0.7593	0.0101	0	0.1556	0.1734	0.0063	0
100	200	0.6160	0.7491	0.0316	0	0.0789	0.1557	0.0293	0
100	500	0.5150	0.7476	0.0429	4	0.0716	0.1430	0.0405	0
100	1000	0.6261	0.7535	0.0194	0	0.0729	0.1510	0.0362	0
500	50	0.5204	0.7550	0.0366	2	0.0920	0.1660	0.0209	0
500	100	0.5561	0.7447	0.0436	4	0.0774	0.1505	0.0342	0
500	200	0.5114	0.7458	0.0477	4	0.0712	0.1491	0.0374	0
500	500	0.5097	0.7025	0.0964	22	0.0709	0.1357	0.0426	0
500	1000	0.5091	0.6777	0.1132	32	0.0701	0.1305	0.0453	0

PSO is slightly bigger $E(Q_{PSO}) = 0.509073$. For the third order system, $E(Q_{CMA\text{-}ES}) = E(Q_{DE}) = 0.069132$ and $E(Q_{PSO}) = 0.070130$. The best solutions provided by CMA-ES and DE are the same in both cases, which means that CMA-ES and DE have the exploration ability to solve the considered quantizer design problem.

A more important result is the success rate when comparing these algorithms. A low success rate indicates that the algorithm gives local minima. Thus, an algorithm with high success rate is reliable. The success rates of the quantizer design algorithms are shown in Figure 5 for the second order system and in Figure 6 for the third order system. They show that the performances of CMA-ES and DE are quite better than that of PSO. For the second order system, the success rates of CMA-ES and DE are always over 60%, while the success rates of PSO are less than 40%. For the third order system, the success rate of PSO is 0% in all cases. Thus, we can see that the quantizer designs based on CMA-ES and DE are reliable methods, but the one based on PSO is not. Moreover, for both plants, the success rates obtained by CMA-ES are better than the ones obtained by DE, when the number of search points N and (or) the maximum number of generations $k_{\max}$ are small. Thus, it is fair to say that the design method based on CMA-ES is better and more reliable than the one based on DE.

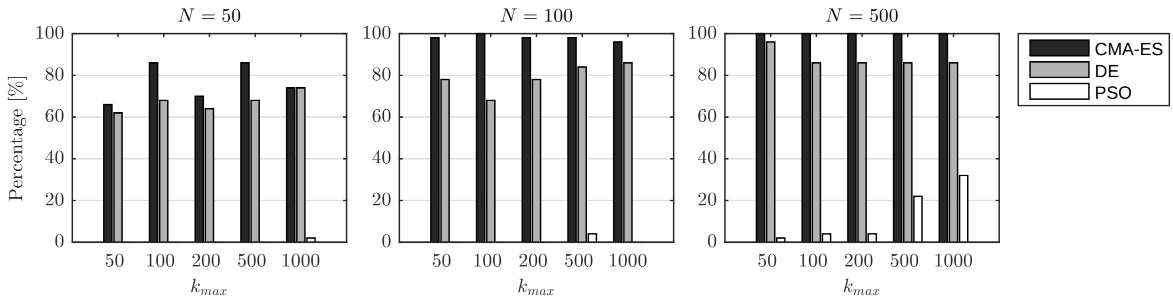


FIGURE 5. Success rate for the second order plant (P_1)

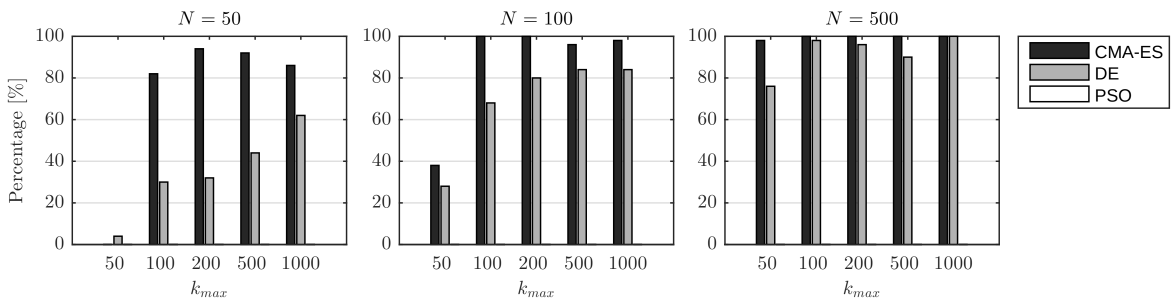


FIGURE 6. Success rate for the third order plant (P_2)

Finally, the convergence behavior of the algorithms is shown in Figure 7 for the second order system and in Figure 8 for the third order system. In these, the cases of (i) $N = 50$ and $k_{\max} = 500$, (ii) $N = 100$ and $k_{\max} = 500$, and (iii) $N = 500$ and $k_{\max} = 500$ are shown. Note that the sequences are shown until $k = 50$ in the figures. In each case, the figures show that the CMA-ES and DE based methods fast converge to the global optima, and that the PSO based design method slowly converges to the local optima.

6. **Conclusion.** In this paper, the finite-level dynamic quantizer design method based on CMA-ES was proposed. Then, through numerical experiments, the effectiveness of

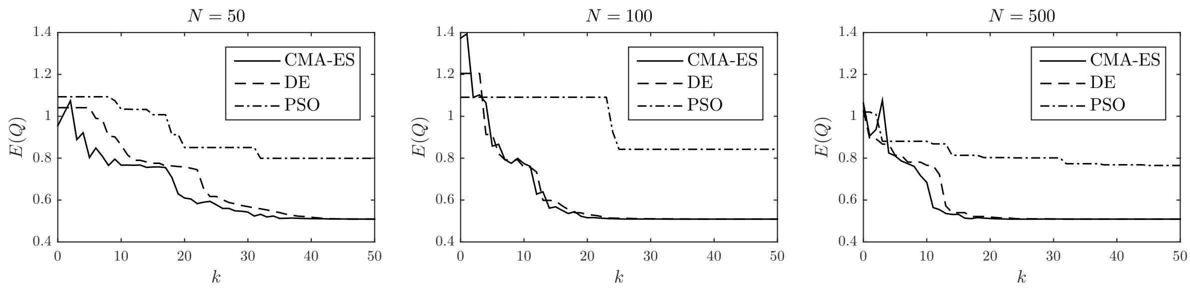


Figure 7. Convergence behavior of the different metaheuristics for the second order plant P_1

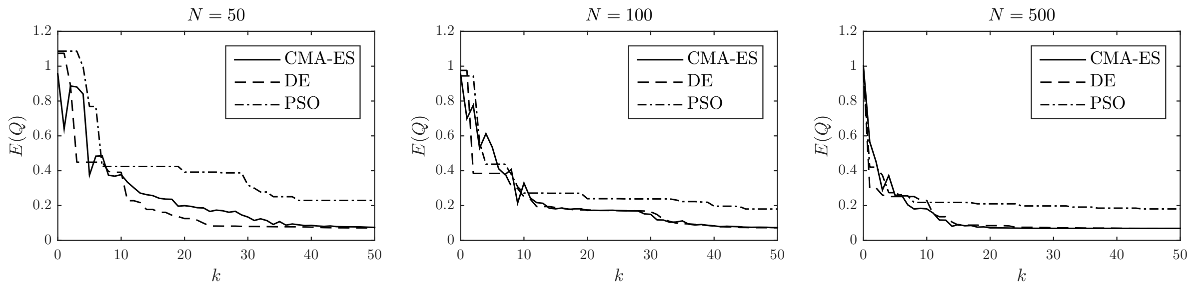


Figure 8. Convergence behavior of the different metaheuristics for the third order plant P_2

the proposed design method was confirmed. The CMA-ES based design method shows a very good performance in terms of success rate and convergence time without the careful tuning of any parameters. Furthermore, compared to the other metaheuristic based design methods, it was verified that the performance of the CMA-ES based method is better than the methods based on DE and PSO. From these results, we can conclude that the CMA-ES based method is very reliable for the design of the dynamic quantizer.

Acknowledgment. The authors would like to thank Associate Professor H. Okajima, Kumamoto University, and Associate Professor T. Matsubara, Nara Institute of Science and Technology for their valuable comments.

REFERENCES

[1] P. J. Antsaklis and J. Baillieul, Special issue on technology of networked control systems, *Proc. of the IEEE*, vol.95, no.1, pp.5-8, 2007.

[2] R. A. Gupta and M. Chow, Networked control system: Overview and research trends, *IEEE Trans. Industrial Electronics*, vol.57, no.7, pp.2527-2535, 2010.

[3] Y.-Q. Xia, Y.-L. Gao, L.-P. Yan and M.-Y. Fu, Recent progress in networked control systems – A survey, *International Journal of Automation and Computing*, vol.12, no.4, pp.343-367, 2015.

[4] S. Azuma and T. Sugie, Optimal dynamic quantizers for discrete-valued input control, *Automatica*, vol.44, no.2, pp.396-406, 2008.

[5] S. Azuma and T. Sugie, Synthesis of optimal dynamic quantizers for discrete-valued input control, *IEEE Trans. Automatic Control*, vol.53, no.9, pp.2064-2075, 2008.

[6] K. Sawada and S. Shin, Synthesis of dynamic quantizers for quantized feedback systems within invariant set analysis framework, *American Control Conference*, pp.1662-1667, 2011.

[7] Y. Minami, S. Azuma and T. Sugie, Optimal decentralized sigma-delta modulators for quantized feedback control, *Nonlinear Theory and Its Applications, IEICE*, vol.3, no.3, pp.386-404, 2012.

[8] W. S. Wong and R. W. Brockett, Systems with finite communication bandwidth constraints. II. Stabilization with limited information feedback, *IEEE Trans. Automatic Control*, vol.44, no.5, pp.1049-1053, 1999.

[9] S. Tatikonda ans S. Mitter, Control under communication constraints, *IEEE Trans. Automatic Control*, vol.49, no.7, pp.1056-1068, 2004.
[10] H. Okajima, N. Matsunaga and K. Sawada, Optimal quantization interval design of dynamic quantizers which satisfy the communication rate constraints, *The 49th IEEE Conference on Decision and Control*, pp.4733-4739, 2010.
[11] K. Sawada, H. Okajima, N. Matsunaga and Y. Minami, Dynamic quantizer design for MIMO systems based on communication rate constraint, *The 37th Annual Conference on IEEE Industrial Electronics Society*, pp.2572-2577, 2011.
[12] R. Yoshino, H. Okajima, N. Matsunaga and Y. Minami, Dynamic quantizers design under data rate constraints by using PSO method, *SICE Annual Conference*, pp.1041-1046, 2014.
[13] J. R. Ramirez, Y. Minami and K. Sugimoto, Design of finite-level dynamic quantizers by using differential evolution algorithm, *SICE Annual Conference*, pp.841-844, 2015.
[14] L. Bianchi, M. Dorigo, L. Gambardella and W. J. Gutjahr, A survey on metaheuristics for stochastic combinatorial optimization, *Natural Computing*, vol.8, no.2, pp.239-287, 2009.
[15] I. Boussaïd, J. Lepagnot and P. Siarry, A survey on optimization metaheuristics, *Information Sciences*, vol.237, pp.82-117, 2013.
[16] N. Hansen and A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, *Evolutionary Computation*, vol.9, no.2, pp.159-195, 2001.
[17] N. Hansen, Nikolaus, S. D. Müller and P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), *Evolutionary Computation*, vol.11, no.1, pp.1-18, 2003.
[18] Y. Wakasa, S. Kanagawa, K. Tanaka and Y. Nishimura, Fixed-structure H_∞ controller synthesis based on the covariance matrix adaptation evolution strategy, *SICE Journal of Control, Measurement, and System Integration*, vol.3, no.4, pp.253-259, 2010.
[19] Y. Wakasa, K. Tanaka and S. Nakashima, Modified virtual reference feedback tuning and its application to ultrasonic motors, *International Journal of Innovative Computing, Information and Control*, vol.9, no.11, pp.4321-4329, 2013.
[20] Y. Wakasa and S. Adachi, Fictitious reference iterative tuning considering input saturation, *IEEEJ Trans. Electrical and Electronic Engineering*, vol.10, no.S1, pp.S159-S161, 2015.
[21] M. W. Iruthayarajan and S. Baskar, Covariance matrix adaptation evolution strategy based design of centralized PID controller, *Expert Systems with Applications*, vol.37, pp.5775-5781, 2010.
[22] K. M. A. Kadhar and S. Baskar, Covariance matrix adaptation evolution strategy based design of fixed structure robust H_∞ loop shaping controller, *Applied Soft Computing*, vol.34, pp.337-348, 2015.
[23] I. Maruta, T.-H. Kim and T. Sugie, Fixed-structure controller synthesis: A meta-heuristic approach using simple constrained particle swarm optimization, *Automatica*, vol.45, no.2, pp.553-559, 2009.
[24] I. Maruta, T.-H. Kim, D. Song and T. Sugie, Synthesis of fixed-structure robust controllers using a constrained particle swarm optimizer with cyclic neighborhood topology, *Expert Systems with Applications*, vol.40, no.9, pp.3595-3605, 2013.
[25] N. Hansen, The CMA evolution strategy: A comparing review, in *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*, J. A. Lozano, P. Larranaga, I. Inza and E. Bengoetxea (eds.), Springer, 2006.
[26] R. Storn and K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, vol.11, no.4, pp.341-359, 1997.
[27] J. Kennedy and E. R. Eberhart, Particle swarm optimization, *IEEE International Conference on Neural Networks*, vol.4, pp.1942-1948, 1995.
[28] N. Hansen, Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed, *GECCO Genetic and Evolutionary Computation Conference*, pp.2389-2396, 2009.

Appendix A. Default Parameters for the $(\mu/\mu_W, \lambda)$ CMA-ES Algorithm. The default values of the parameters were taken from [28].

(i) Selection and Recombination:

$$N = 4 + \lfloor 3\ln(n) \rfloor, \quad \mu = \lfloor \mu' \rfloor, \quad \mu' = \frac{N}{2}, \tag{27}$$

$$w_i = \frac{w_i'}{\sum_{j=1}^{\mu} w_j'}, \quad w_i' = \ln(\mu' + 0.5) - \ln i \quad \text{for } i = 1, 2, \ldots, \mu, \tag{28}$$

$$\mu_{\text{eff}} = \frac{1}{\sum_{i=1}^{\mu} w_i^2}. \tag{29}$$

(ii) Step-size control:

$$c_\sigma = \frac{\mu_{\text{eff}} + 2}{n + \mu_{\text{eff}} + 5}, \tag{30}$$

$$d_\sigma = 1 + 2\max\left(0, \sqrt{\frac{\mu_{\text{eff}} - 1}{n+1}} - 1\right) + c_\sigma. \tag{31}$$

(iii) Covariance matrix adaptation:

$$c_c = \frac{4 + \mu_{\text{eff}}/n}{n + 4 + 2\mu_{\text{eff}}/n}, \tag{32}$$

$$c_1 = \frac{2}{(n+1.3)^2 + \mu_{\text{eff}}}, \tag{33}$$

$$c_\mu = \min\left(1 - c_1, \alpha_\mu \frac{\mu_{\text{eff}} - 2 + 1/\mu_{\text{eff}}}{(n+2)^2 + \alpha_\mu \mu_{\text{eff}}/2}\right) \quad \text{with} \quad \alpha_\mu = 2. \tag{34}$$

Appendix B. **DE and PSO Algorithms.** The DE algorithm is shown in Algorithm 2, and the PSO algorithm is shown in Algorithm 3.

Algorithm 2: DE (DE/best/1/bin strategy)

Initialization: Given $N \in \mathbb{N}$, $k_{\max} \in \mathbb{N}$, $F \in [0,2]$, $H \in [0,1]$ and the initial search space $S = [\boldsymbol{x}_{\min}, \boldsymbol{x}_{\max}]^n$. Set $k = 0$ then select randomly N search points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ in the search space.

Step 1: The objective function $f(\boldsymbol{x})$ is evaluated for each $\boldsymbol{x}_i$ and $\boldsymbol{x}_{base} = \boldsymbol{x}_{l_k}$ is calculated by:

$$l_k = \underset{i \in \{1,2,\ldots,N\}}{\arg\min}\ f(\boldsymbol{x}_i). \tag{35}$$

If $k = k_{\max}$ then $\boldsymbol{x}_{base}$ is the final solution, if not go to **Step 2**.

Step 2 (Mutation): For each $\boldsymbol{x}_i$ a mutant vector $\boldsymbol{M}_i$ is generated by:

$$\boldsymbol{M}_i = \boldsymbol{x}_{base} + F(\boldsymbol{x}_{\tau_{1,i}} - \boldsymbol{x}_{\tau_{2,i}}), \tag{36}$$

where $\tau_{1,i}$ and $\tau_{2,i}$ are random indexes subject to $i \neq \tau_{1,i} \neq \tau_{2,i} \neq l_k$.

Step 3 (Crossover): For each $\boldsymbol{x}_i$ and $\boldsymbol{M}_i$ a trial vector $\boldsymbol{L}_i$ is generated by:

$$L_{i,j} = \begin{cases} M_{i,j} & \text{if } \rho_{i,j} \leq H \text{ of } j = j_{rand}, \\ x_{i,j} & \text{otherwise,} \end{cases} \tag{37}$$

where $\rho_{i,j} \in [0,1]$ and $j_{rand} \in \{1, 2, \ldots, n\}$ are generated randomly.

Step 4 (Selection): The members of the next generation $k+1$ are selected by:

$$\boldsymbol{x}_i \leftarrow \begin{cases} \boldsymbol{L}_i & \text{if } f(\boldsymbol{L}_i) \leq f(\boldsymbol{x}_i), \\ \boldsymbol{x}_i & \text{otherwise,} \end{cases} \tag{38}$$

then $k \leftarrow k+1$ and go to **Step 1**.

Algorithm 3: PSO

Initialization: Given $N \in \mathbb{N}$, $k_{\max} \in \mathbb{N}$, $\chi_0 \in [0,1]$, $\chi_1 \in [0,4]$, $\chi_2 \in [0,4]$ and the initial search spaces $S_x = [\boldsymbol{x}_{\min}, \boldsymbol{x}_{\max}]^n$ and $S_v = [\boldsymbol{v}_{\min}, \boldsymbol{v}_{\max}]^n$. Set $k = 0$ then select randomly N search points $\{\boldsymbol{x}_1^0, \boldsymbol{x}_2^0, \ldots, \boldsymbol{x}_N^0\}$ and their velocities $\{\boldsymbol{v}_1^0, \boldsymbol{v}_2^0, \ldots, \boldsymbol{v}_N^0\}$ in the corresponding search spaces.
Step 1: The objective function $f(\boldsymbol{x}_i^k)$ is evaluated for each $\boldsymbol{x}_i^k$. Then, the personal best solutions and the global best solution are selected by:

$$\boldsymbol{x}_{pbest,i}^k = \underset{\boldsymbol{x} \in \{\boldsymbol{x}_i^j | j=1,2,\ldots,k\}}{\arg\min} f(\boldsymbol{x}), \tag{39}$$

$$\boldsymbol{x}_{gbest}^k = \underset{\boldsymbol{x} \in \{\boldsymbol{x}_{pbest,i}^k | i=1,2,\ldots,N\}}{\arg\min} f(\boldsymbol{x}). \tag{40}$$

If $k = k_{\max}$ then $\boldsymbol{x}_{gbest}^k$ is the solution of the algorithm, if not go to **Step 2**.
Step 2: Sequentially the following update laws are applied to each search point.

$$\boldsymbol{v}_i^{k+1} = \chi_0 \boldsymbol{v}_i^k + \chi_1 \rho_{1,i} \left(\boldsymbol{x}_{pbest,i}^k - \boldsymbol{x}_i^k\right) + \chi_2 \rho_{2,i} \left(\boldsymbol{x}_{gbest}^k - \boldsymbol{x}_i^k\right), \tag{41}$$

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k + \boldsymbol{v}_i^{k+1}, \tag{42}$$

where $\rho_{1,i}$ and $\rho_{2,i} \in [0,1]$ are random numbers uniformly distributed. Then make $k \leftarrow k+1$ and go to **Step 1**.