NAIST-IS-MT1451126

Master's Thesis

Metaheuristic Based Design of Finite-level Dynamic Quantizers for Networked Control Systems

Juan Esteban Rodriguez Ramirez

March 14, 2016

Department of Information Science
Graduate School of Information Science
Nara Institute of Science and Technology

A Master's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
MASTER of ENGINEERING

Juan Esteban Rodriguez Ramirez

Thesis Committee:

Professor Kenji Sugimoto	(Supervisor)
Professor Shoji Kasahara	(Co-supervisor)
Associate Professor Takamitsu Matsubara	(Co-supervisor)
Assistant Professor Yuki Minami	(Co-supervisor)

Metaheuristic Based Design of Finite-level Dynamic Quantizers for Networked Control Systems*

Juan Esteban Rodriguez Ramirez

Abstract

Quantization, that transforms continuous-valued signals into discrete-valued ones, plays a crucial role in the operation of networked control systems (NCSs). There are two issues associated to the quantization of signals: the quantization errors and the communication rate constraint of the channel. In order to diminish the effects of these problems, this thesis focuses on the design of dynamic quantizers that minimize the performance degradation due to the quantization errors under the communication rate constraint. The quantizer design problem is formulated as a nonconvex optimization problem, then, it cannot be solved by conventional optimization methods. Thus, in this thesis, design methods based on metaheuristics are proposed. By numerical examples the effectiveness of the proposed methods is verified and a comparison is carried out among the design methods based on different metaheuristics. Besides, an optimal expression for the design of the quantizer is numerically studied and an event-based switched-type dynamic quantizer is introduced.

Keywords:

Networked Control System (NCS), Dynamic Quantization, Communication Constraints, Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES), Differential Evolution (DE)

---

*Master's Thesis, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1451126, March 14, 2016.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Networked Control Systems

In recent years, the networked control systems or NCSs have been receiving a lot of attention from researchers and manufactures due to the their many advantages and potential applications. In a NCS the plants, controllers, sensors and actuators are physically separated and connected to each other via communication networks as it is shown in Figure 1.1. A characteristic feature of the NCSs is that the communication network is shared and multiple control systems can be operating over it at the same time. Moreover, the NCS can be connected to the companies intranet or even to the internet allowing any type of data to go through them, not only control signals.

The NCSs have several advantages. The elimination of the unnecessary wiring and the use of network technologies reduce the system's complexity and make it very easy to add or withdraw controllers, sensors or actuators. Thus, NCSs make the systems very flexible and scalable. Additionally, by using a network to connect the elements of the control system, the costs of installation and maintenance are greatly reduced. The NCSs connect the cyber space to the real world making the tele-operation of systems very easy to implement. Examples of NCSs are found in industrial automation, control systems for automobiles and aircraft and the control of large distributed systems like smart grids, transportation and water distribution networks. They have a lot of potential applications like disaster prevention systems, hazardous environments monitoring, domestic robots, space and terrestrial exploration, tele-operated systems and many others.

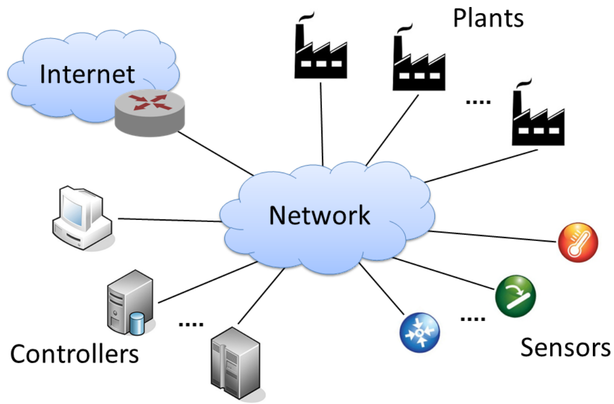


Figure 1.1: General structure of an NCS.

Although, NCSs bring many advantages to the systems, they have several problems associated to them, like the delay in the transmission of data, jitter, package looses, quantization errors and limited information bandwidth [1, 2, 3]. This thesis focuses on the last two of these problems.

In NCSs, the control and sensor signals are transmitted through the communication network. These signals need to be quantized in order to be sent through a digital channel. The quantization is the process in which the continuous-valued signals are transformed into discrete-valued ones. The device that performs the quantization of the signals is called quantizer. In general a quantizer is composed by a certain *number of quantization levels M* and a constant distance between the quantization levels called *quantization interval d*. Figure 1.2 shows an example of quantizer. When the effects of quantization are considered in the design and operation of a control system, we are in the presence of a quantized control system. Figure 1.3 shows how an open loop quantized system works.

There exists limitations in the amount of data that can be transmitted per unit of time in the networked control systems, these limitations are known as *data rate constraints*. Since the quantizer is supposed to send data to the plant in each sampling time, the data rate constraint of the channel limits the number of quantization levels in the quantizer. Each quantized signal is transformed into a chain of bits by a encoder in order to be transmitted. The amount of bits generated by the encoder depends of the amount of quantization levels that the quantizer has. These bits should be sent before

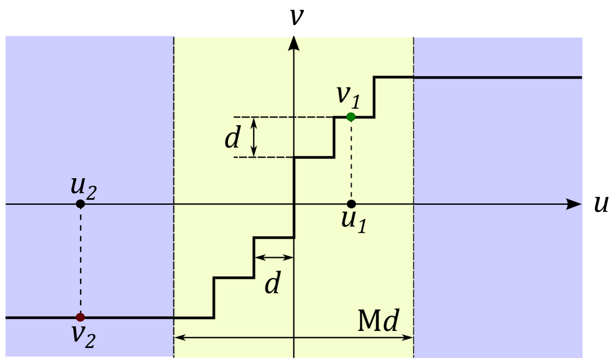


Figure 1.2: Example of a finite-level quantizer ($M = 6$). Where u is the continuous-valued input and v is the discrete-valued output. The yellow area represents the *range of the quantizer* and the blue one is the *saturation area*. In order the quantizer to work properly, $u(t)$ should be always inside the range of quantizer. For example the value of u_1 gives a quantized value v_1 inside the quantizer's range, but the value u_2 inside the saturation area gives a quite smaller quantized value v_2.

another sampled signal arrives to the encoder. Then, the system will work properly only if the if the amount of bits that can be sent through the channel during a sampling time interval is more or at least equal to the amount of bits generated by the encoder due to one quantized signal.

Besides, a finite number of quantization levels may cause the saturation of the amplitude of the quantized signal. The saturation happens when the input signal's amplitude is bigger than the maximum value that the quantizer can transformed. What happens in this case is usually that the quantizer assigns to the quantized output the maximum value that the quantizer can produce. In consequence there may be a big difference between the input and the output of the quantizer. Such saturation problem has the potential to destabilize the systems [4, 5].

Since nowadays there are networks that transmit data in gigabits per second, the data rate constraints may not look like a big problem. But remember that the NCSs are shared networks and the effects of congestion and bottle necks will affect the performance of the system [6, 7].

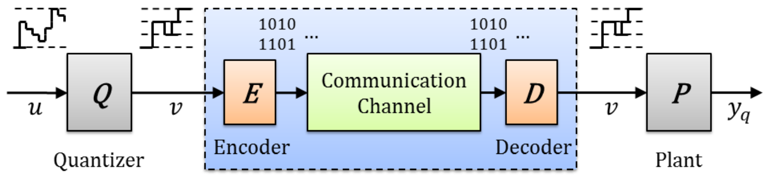


Figure 1.3: Structure of an open loop quantized control system. The quantizer Q transform the control signal u in the quantized signal v. Then, v is transformed in a chain of bits by the encoder E in order the be sent trough the channel and it is reconstructed on the destination by a decoder D.

1.2. Quantizer Design for Networked Control Systems

The difference between a signal and its quantized version is known as quantization error. The quantization errors are reflected in the output of the plant causing the degradation in the performance of the system. The performance degradation due to quantization might lead to instability. Several studies have been carried out aiming to reduce the performance degradation due to quantization, and they have shown that the use of feedback type dynamic quantizers is an effective way to do so [8, 9, 10, 11, 12].

There are several types of quantizers. They can be divided into two categories static and dynamic. A dynamic quantizer is a device that maps the continuous-valued input signal into discrete-valued outputs by a feedback and filtering of the past quantization errors. In this study we consider a feedback type of dynamic quantizer Q that is composed of a digital filter followed by a static quantizers q. Figure 1.4 shows a representation of this quantizer.

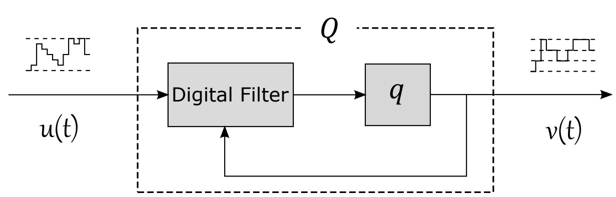


Figure 1.4: Feedback type dynamic quantizer.

The limited amount of quantization levels in the quantizer may cause the quantization errors to get bigger. For example, when the range of the quantizer is big in comparison with the range of the input signal and the number quantization levels is small. The quantization errors and, in consequence, the performance degradation can be reduced greatly by the proper design of the quantizers. In the dynamic quantizer design the goal is usually to make the quantization errors as small as possible in order to approximate the quantized system to the ideal system without quantization.

In past studies the authors focuses only in the design of the quantizer's digital filter assuming that the quantization interval d is given and the quantizer has an infinite number of quantization levels M. For example, in [8] Azuma and Sugie (2008) found a closed form expression with respect to the plant parameters to evaluate the performance of a class of dynamic quantizers for LTI systems. Besides, based on the performance analysis, they provided an analytic expression of the optimal quantizer and its performance. In 2014, Minami and Murokami [13] used differential evolution (DE) for the design of fixed-order decentralized dynamic quantizers for discrete-valued input control. In these studies the effects of the channel's data rate constraints were not taken into account.

Then, in order consider those effects it is necessary to design the digital filter and the static quantizer at the same time. In this case the static quantizer has a finite number of quantization levels. Some studies have tackled already this type of problem [6, 7, 14, 15] and developed methods for the design of the finite-level dynamic quantizer. However, these methods are not perfect because they may not give optimal solutions to the quantizer design problems.

In [6], Okajima et al. (2010) proposed a design method for finite-level dynamic quantizers for SISO systems. They derived a design method of the quantizer's smallest quantization interval. Based on the invariant set analysis the design method is derived as a linear matrix inequality (LMI) problem. In [7], Sawada et al. (2011) extended the previous study to consider MIMO systems. In these two methods they solved a relaxed version of the original quantizer design problem and because of that the solution might be conservative. In [14], Yoshino et al. (2014) presented a design method for a dynamic quantizer under data rate constraints. The design is carried out by optimizing an objective function using a metaheuristic known as particle swarm optimization (PSO). This method gives a smaller performance degradation than the method proposed by Okajima et al., but the rate of success in finding the optimum values of the quantizer is very low. The low success rate is explained by the existence of local minima in the objective function and the tendency of PSO of getting trapped on it.

Additionally, Rodriguez et al. in [15] (2015) tackled the finite-level dynamic quantizer design problem by using DE. This publication is part of this thesis and the design method will be explained in detail in the chapters that follow.

1.3. Motivation and Objective

NCSs are a natural step in the evolution of the control systems, they have a lot of advantages and the potential to become the standard in automation. They are easy to implement and cheap, thus, very likely to be implemented even in small industries. Thereby, it is important to increase the reliability of NCSs reducing the performance degradation caused by quantization. A way to do this is the use of dynamic quantizers properly designed to minimize quantization errors.

Moreover, the constraints in the communication is an issue that needs to be addressed. Even if the channel has a big bandwidth the effects of congestion and bottleneck cannot be ignored, since they make the system unable to operate properly. Then, the quantizers should be designed considering the constraints in the communication channel and in order to put as small traffic as possible in the network.

For these reasons the main objective of this thesis is:

> To develop easy-to-use and powerful computational methods for the design of finite-level dynamic quantizers that:
>
> i Minimize the system's performance degradation due to quantization, and
>
> ii Satisfy the channel's data rate constraints.

The finite-level dynamic quantizer design problem considered in this thesis is not easy to solve. It leads to the optimization of a non-linear and non-convex function, that cannot be solved by algebraic methods or conventional numerical optimization methods such as linear programming or quadratic programming. Thus, the use of metaheuristics seems a reasonable option. Three different metaheuristics were selected to implement the quantizers design method, namely, covariance matrix adaptation (CMA-ES), differential evolution (DE) and firefly algorithm (FA). The effectiveness of the design methods developed is verified by means of numerical examples, and their efficiency is compared among each other and with previous developed design methods based on particle swarm optimization (PSO) [14]. It is revealed that the proposed methods based on CMA-ES and DE achieve better performance than the others in terms of precision, convergence and execution time.

Besides, making use of the design methods developed, this thesis extends the study about dynamic quantization into two different topics. The first one studies the reduction of design parameters for the finite-level dynamic quantizer, which is an important topic in order to make the system analysis easy and to save time in the design process. The design parameter reduction analysis is made by means of numerical examples. The second extended topic is about the development of a type of dynamic quantizer that operates in a event-based manner. The use of this type of quantizers is oriented to systems in which network resources like bandwidth and energy are very limited. The event operated dynamic quantizer will help to preserve these resources by sending data to the plant only when is needed. This type of quantizers should make a trade-off between the performance of the system and the consumption of resources in the network.

1.4. Contributions and Organization

The main contributions and achievements of this work are summarized as follow:

- The development of practical dynamic quantizer design methods based on CMA-ES and DE.
- The verification of the proper operation of the quantizers designed with the proposed methods.
- The comparison of the proposed design methods with a previously developed design method and the evidence of their superior performance.
- The finding of a quasi optimal analytic expression for the finite-level dynamic quantizer.
- The development of a type of data rate driven event-triggered quantizers and its design method.

This thesis is divided into six chapters. In chapter 2, the finite-level quantizer design problem is formulated. The considered system is described as well as the considerations and simplifications made. The performance index, and the minimum quantization interval are described. Finally the design problem and constraints are formally stated. Chapter 3 describes the metaheuristics employed to implement the quantizer design methods. After a brief description of them the algorithms are explained in detail. In Chapter 4, the metaheuristic based quantizer design method is described along with the methodology and settings to implement it. After that, the effectiveness of the proposed design methods are verified by means of numerical examples and a comparison is carried out among them and previously developed methods. Chapter 5 discuses some topics related to the finite-level quantizer design, like the reduction of parameters in the design and the extension of the study to consider event-based actuated quantizers. Lastly, in Chapter 6, the final conclusions are presented and some perspectives for future expansions of the current work are given.

1.5. Notations

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

Chapter 2

Quantizer Design Problem

2.1. Considered System

In this thesis we consider the feedforward system shown in Figure 2.1. This system is known as *error system* and it is used to evaluate the quantized system performance.

The error system is composed of the quantizer Q, the plant P and the channel. The control signal $u(t)$ is applied to the system and it goes through two branches. The lower branch represents the ideal case in which $u(t)$ is applied directly to the plant giving the ideal output $y(t)$. In the upper branch the effects of the channel and quantization are considered, the quantized control signal $v(t)$ is applied to the plant and the output $y_q(t)$ is known as real output. The difference between $y(t)$ and $y_q(t)$ is the error signal $e(t)$, which gives a measurement of the performance of the system.

The assumptions made in this thesis are the following: the communication channel has no losses and no delays, the plant P is stable and the input signal is bounded, i.e., $u(t) \in U$ for $U := [u_{min}, u_{max}]$.

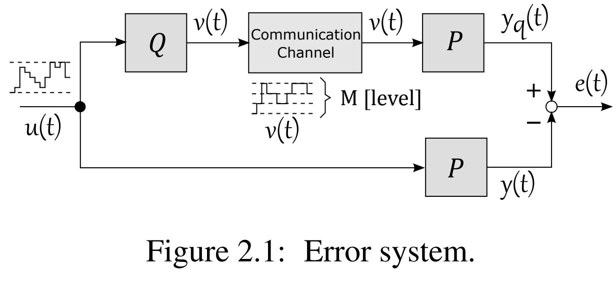


Figure 2.1: Error system.

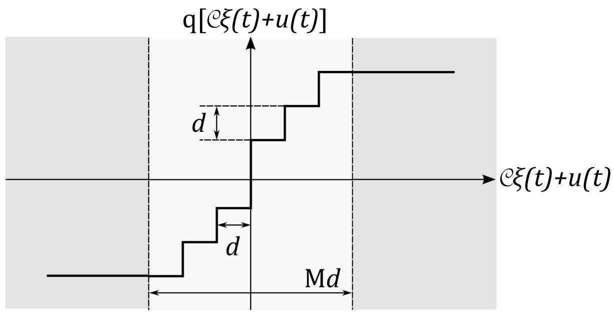


Figure 2.2: Example of a static quantizer function q$[\cdot]$ ($M = 6$).

The plant P is discrete-time and single-input-single-output (SISO) system. Its state space representation is given as follows:

$$P : \begin{cases} \mathbf{x}(t+1) & = \mathrm{A}\mathbf{x}(t) + \mathrm{B}u(t), \\ y(t) & = \mathrm{C}\mathbf{x}(t), \end{cases} \tag{2.1}$$

where $t \in \{0\} \cup \mathbb{N}$ is the discrete time, $\mathbf{x} \in \mathbb{R}^{n_P}$ is the state vector, $u \in \mathbb{R}$ is the control signal, $y \in \mathbb{R}$ is the output signal, $\mathrm{A} \in \mathbb{R}^{n_P \times n_P}$, $\mathrm{B} \in \mathbb{R}^{n_P \times 1}$ and $\mathrm{C} \in \mathbb{R}^{1 \times n_P}$ are constant matrices and the initial state of the plant is $\mathbf{x}(0) = \mathbf{x}_0$ for $\mathbf{x}_0 \in \mathbb{R}^{n_P}$. With the assumption that the plant P is stable we imply that all the eigenvalues of the matrix A are inside the unit circle in the complex plane.

The dynamic quantizer considered in this study is shown in Figure 1.4 and is represented by

$$Q : \begin{cases} \xi(t+1) & = \mathcal{A}\xi(t) + \mathcal{B}(v(t) - u(t)), \\ v(t) & = \mathrm{q}[\mathcal{C}\xi(t) + u(t)], \end{cases} \tag{2.2}$$

where $\xi \in \mathbb{R}^{n_Q}$ is the state vector, $v \in \{\pm d, \pm 2d, \cdots, \pm \frac{M}{2} d\}$ is the quantized output, $\mathcal{A} \in \mathbb{R}^{n_Q \times n_Q}$, $\mathcal{B} \in \mathbb{R}^{n_Q \times 1}$ and $\mathcal{C} \in \mathbb{R}^{1 \times n_Q}$ are constant matrices, and the initial state of the quantizer is $\xi(0) = \mathbf{0}$. The static quantizer q$[\cdot]$ rounds the signals to the nearest quantization level at each time. 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 2.2 shows an example of the function q$[\cdot]$ where $M = 6$.

In this thesis it is assumed that M is given in advance. Thus, the design parameters of the dynamic quantizer Q are $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and d.

An important care that should be taken into account when designing a dynamic quantizer is to make it stable. The stability condition for the quantizer can be derived easily. First lets represent the quantizer's output $v(t)$ given in Equation (2.2) as follows

$$v(t) = \mathcal{C}\xi(t) + u(t) + w(t), \tag{2.3}$$

where $w(t) \in \left[-\frac{d}{2}, \frac{d}{2}\right]$ is the *quantization error*. Then, substituting the new $v(t)$ into Equation (2.2) we obtain the following representation of the quantizer

$$Q : \begin{cases} \xi(t+1) & = (\mathcal{A} + \mathcal{B}\mathcal{C})\,\xi(t) + \mathcal{B}w(t), \\ v(t) & = \mathcal{C}\xi(t) + w(t) + u(t). \end{cases} \tag{2.4}$$

In this representation the quantizer's state $\xi(t)$ depends only of the quantization error $w(t)$ which works as a bounded signal. Thus, the quantizer Q is bounded-input-bounded-output (BIBO) stable if all the eigenvalues of $(\mathcal{A} + \mathcal{B}\mathcal{C})$ are inside the unit circle in the complex plane. This condition can be expressed as

$$\bar{\Lambda} = \max_i \operatorname{abs}\left(\Lambda_i(\mathcal{A} + \mathcal{B}\mathcal{C})\right) < 1, \tag{2.5}$$

where $\bar{\Lambda}$ is the maximum absolute value of the eigenvalues of the $(\mathcal{A} + \mathcal{B}\mathcal{C})$.

2.2. Performance Index

As it was mentioned previously, the error signal $e(t) = y_q(t) - y(t)$ can be used to evaluate the performance degradation of the system. Minimizing $e(t)$, the system composed of the quantizer Q and the linear plant P can be optimally approximated to the linear plant P, in terms of the input-output relation.

The performance degradation of the system due to quantization is measured by a parameter called *performance index* which is defined as follows

$$E(Q) := \sup_{\substack{u \in U \\ t \in \{1,2,\ldots,L\}}} \operatorname{abs}\left(y_q(t) - y(t)\right), \tag{2.6}$$

where $L \in \mathbb{N}$ is the evaluation interval. $E(Q)$ gives the biggest possible value of $e(t)$, i.e., the worst case performance of the system.

To calculate the value of $E(Q)$ we proceed in the following way. Putting together Equations (2.1) and (2.2) we have

$$\begin{cases} \begin{bmatrix} \mathbf{x}(t+1) \\ \xi(t+1) \end{bmatrix} & = \begin{bmatrix} \mathrm{A} & 0 \\ 0 & \mathcal{A} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \xi(t) \end{bmatrix} + \begin{bmatrix} \mathrm{B} \\ \mathcal{B} \end{bmatrix} v(t) - \begin{bmatrix} 0 \\ \mathcal{B} \end{bmatrix} u(t), \\ \\ y(t) & = \begin{bmatrix} \mathrm{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \xi(t) \end{bmatrix}. \end{cases} \tag{2.7}$$

By replacing the quantized output $v(t)$ with Equation (2.3) we can express Equation (2.7) as:

$$\begin{cases} \bar{\mathbf{x}}(t+1) & = \bar{\mathrm{A}}\bar{\mathbf{x}}(t) + \bar{\mathrm{B}}_1 w(t) + \bar{\mathrm{B}}_2 u(t), \\ y(t) & = \bar{\mathrm{C}}\bar{\mathbf{x}}(t), \end{cases} \tag{2.8}$$

where:

$$\bar{\mathbf{x}}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \xi(t) \end{bmatrix}, \quad \bar{\mathrm{A}} = \begin{bmatrix} \mathrm{A} & \mathrm{B}\mathcal{C} \\ 0 & \mathcal{A}+\mathcal{B}\mathcal{C} \end{bmatrix}, \quad \bar{\mathrm{B}}_1 = \begin{bmatrix} \mathrm{B} \\ \mathcal{B} \end{bmatrix}, \quad \bar{\mathrm{B}}_2 = \begin{bmatrix} \mathrm{B} \\ 0 \end{bmatrix}, \quad \bar{\mathrm{C}} = \begin{bmatrix} \mathrm{C} & 0 \end{bmatrix}. \tag{2.9}$$

The expression of the state $\bar{\mathbf{x}}(t)$ for this system is given as follows

$$\bar{\mathbf{x}}(t) = \bar{\mathrm{A}}^t \bar{\mathbf{x}}(0) + \sum_{i=0}^{t-1} \bar{\mathrm{A}}^{t-1-i} \big(\bar{\mathrm{B}}_1 w(i) + \bar{\mathrm{B}}_2 u(i)\big). \tag{2.10}$$

Then, we take Equation (2.10) to Equation (2.8) in order to find the plant output under the effects of quantization, the real output $y_q(t)$. The result is

$$y_q(t) = \bar{\mathrm{C}}\bar{\mathrm{A}}^t \bar{\mathbf{x}}(0) + \sum_{i=0}^{t-1} \bar{\mathrm{C}}\bar{\mathrm{A}}^{t-1-i} \big(\bar{\mathrm{B}}_1 w(i) + \bar{\mathrm{B}}_2 u(i)\big). \tag{2.11}$$

Considering the value $\bar{\mathrm{A}}$ in Equation (2.9), the expresion $\bar{\mathrm{A}}^t$ can be reduced to

$$\bar{\mathrm{A}}^t = \begin{bmatrix} \mathrm{A}^t & \Omega(t) \\ \mathbf{0} & (\mathcal{A}+\mathcal{B}\mathcal{C})^t \end{bmatrix}, \tag{2.12}$$

where $\Omega(t)$ is an expression that increases for each t, in this case its value is not important since it will disappear from the calculation. Effectively, remembering that $\bar{\mathbf{x}}(0) = [\mathbf{x}(0) \quad \xi(0)]^\top = [\mathbf{x}(0) \quad 0]^\top$, the terms of Equation (2.11) are represented by

$$\bar{\mathrm{C}}\bar{\mathrm{A}}^t \bar{\mathbf{x}}(0) = \mathrm{C}\mathrm{A}^t \mathbf{x}(0), \tag{2.13}$$

$$\bar{\mathrm{C}}\bar{\mathrm{A}}^{t-1-i} \bar{\mathrm{B}}_2 u(i) = \mathrm{C}\mathrm{A}^{t-1-i} \mathrm{B} u(i). \tag{2.14}$$

Then, $y_q(t)$ can be rewritten as

$$y_q(t) = \mathrm{C}\mathrm{A}^t \mathbf{x}(0) + \sum_{i=0}^{t-1} \bar{\mathrm{C}}\bar{\mathrm{A}}^{t-1-i}\bar{\mathrm{B}}_1 w(i) + \sum_{i=0}^{t-1} \mathrm{C}\mathrm{A}^{t-1-i}\mathrm{B}u(i). \tag{2.15}$$

For the case in which the control signal $u(t)$ is apply directly to the plant P (ideal case) the plant's output $y(t)$ is

$$y(t) = \mathrm{C}\mathrm{A}^t \mathbf{x}(0) + \sum_{i=0}^{t-1} \mathrm{C}\mathrm{A}^{t-1-i}\mathrm{B}u(i). \tag{2.16}$$

The difference between the Equations (2.15) and (2.16) gives us:

$$y_q(t) - y(t) = \sum_{i=0}^{t-1} \bar{\mathrm{C}}\bar{\mathrm{A}}^{t-1-i}\bar{\mathrm{B}}_1 w(i). \tag{2.17}$$

Taking the absolute value of both members of Equation (2.17) and considering that the quantized error $\mathrm{abs}\,(w(i)) \leq \frac{d}{2}$ we have

$$\begin{aligned} \mathrm{abs}\left(y_q(t) - y(t)\right) &= \mathrm{abs}\left(\sum_{i=0}^{t-1} \bar{\mathrm{C}}\bar{\mathrm{A}}^{t-1-i}\bar{\mathrm{B}}_1 w(i)\right), \\ &\leq \sum_{i=0}^{t-1} \mathrm{abs}\left(\bar{\mathrm{C}}\bar{\mathrm{A}}^{t-1-i}\bar{\mathrm{B}}_1\right) \mathrm{abs}\,(w(i)), \\ &\leq \frac{d}{2}\sum_{i=0}^{t-1} \mathrm{abs}\left(\bar{\mathrm{C}}\bar{\mathrm{A}}^{t-1-i}\bar{\mathrm{B}}_1\right). \end{aligned} \tag{2.18}$$

By replacing the values in Equation (2.9) to Equation (2.18) we obtain

$$\mathrm{abs}\left(y_q(t) - y(t)\right) \leq \frac{d}{2}\sum_{t=0}^{t-1} \mathrm{abs}\left(\begin{bmatrix} \mathrm{C} & 0 \end{bmatrix} \begin{bmatrix} \mathrm{A} & \mathrm{B}\mathcal{C} \\ 0 & \mathcal{A} + \mathcal{B}\mathcal{C} \end{bmatrix}^{t-1-i} \begin{bmatrix} \mathrm{B} \\ \mathcal{B} \end{bmatrix}\right). \tag{2.19}$$

Finally, in [8] it is proven that the equality in Equation (2.19) gives the supremum of $\mathrm{abs}\left(y_q(t) - y(t)\right)$, this means that the performance index is given by

$$E(Q) = \frac{d}{2}\sum_{t=0}^{L} \mathrm{abs}\left(\begin{bmatrix} \mathrm{C} & 0 \end{bmatrix} \begin{bmatrix} \mathrm{A} & \mathrm{B}\mathcal{C} \\ 0 & \mathcal{A} + \mathcal{B}\mathcal{C} \end{bmatrix}^{t} \begin{bmatrix} \mathrm{B} \\ \mathcal{B} \end{bmatrix}\right), \tag{2.20}$$

where the evaluation interval $L \to \infty$, but for practical purposes it can be considered as a big number.

In order to obtain the smallest performance degradation, it is necessary to make $E(Q)$ as small as possible by the appropriate design of the dynamic quantizer Q. Then, the dynamic quantizer design is reduced to an optimization problem, where the objective function is given by $E(Q)$ in Equation (2.20) and the design variables are the parameters of the quantizer Q.

2.3. Minimum Quantization Interval

The data rate constraint of the communication channel imposes limitations in the design of dynamic quantizers. Considering that N_b is the number of bits that can be transmitted through the channel per sampling time, the number of quantization levels M should satisfy the following condition

$$M \leq 2^{N_b}. \tag{2.21}$$

In this thesis it is assumed that M is given so as to satisfy Equation (2.21).

On the other hand, Equation (2.20) states that d is directly proportional to $E(Q)$. Then, in order to minimize $E(Q)$ it is necessary to make d as small as possible. However, d and M are subjected to the following condition derived from Figure 2.2.

$$\mathrm{abs}\left(\mathcal{C}\xi(t) + u(t)\right) \leq \frac{1}{2}Md. \tag{2.22}$$

The minimization of d under this condition is equivalent to a reachable set problem not easy to solve.

Okajima et al. in [6], [16] and [17] found an expresion for the minimum quantization interval that satisfies the data rate constrains d^*. To design d, we need to consider the values of $\xi(t)$ and $u(t)$. Although the range of $u(t)$ is given by $U = [u_{min}, u_{max}]$ the range of $\xi(t)$ is unknown. Thus, the range of $\xi(t)$ should be estimated for a given Q.

The range of the quantizer $\bar{U} = [\bar{u}_{min}, \bar{u}_{max}]$ can be characterize by the ranges of $u(t)$ and $\psi = \mathcal{C}\xi$ as follows

$$\bar{U} = \left[u_{min} + \psi_{min}, u_{max} + \psi_{max}\right], \tag{2.23}$$

Considering Equation (2.3) a normalized version of the quantization error $w(t)$ is given by

$$\bar{w}(t) = \frac{w(t)}{d/2} = \frac{2}{d}\left(v(t) - \mathcal{C}\xi - u(t)\right), \tag{2.24}$$

where $\bar{w}(t) \in [-1,1]$ and, in consequence, $|\bar{w}(t)| \leq 1$. From this normalized error we build the following system

$$H : \begin{cases} \xi(t+1) & = (\mathcal{A}+\mathcal{B}\mathcal{C})\,\xi(t) + \frac{d}{2}\mathcal{B}\bar{w}(t), \\ \psi(t) & = \mathcal{C}\xi(t), \qquad |\bar{w}(t)| \leq 1, \end{cases} \tag{2.25}$$

by making the following change in the coordinates $\xi = \frac{d}{2}\bar{\xi}$ we can represent the system H as

$$\bar{H} : \begin{cases} \bar{\xi}(t+1) & = (\mathcal{A}+\mathcal{B}\mathcal{C})\,\bar{\xi}(t) + \mathcal{B}\bar{w}(t), \\ \bar{\psi}(t) & = \mathcal{C}\bar{\xi}(t), \qquad |\bar{w}(t)| \leq 1, \end{cases} \tag{2.26}$$

where $\bar{\psi} = 2\psi/d$. Notice that the system $\bar{H}$ is independent of d.

The reachable set problem is reduced into finding the $\bar{\psi}_{min}$ and $\bar{\psi}_{max}$ subjected to the condition that

$$\bar{\psi}_{min} \leq \mathcal{C}\bar{\xi}(t) \leq \bar{\psi}_{max}, \qquad \forall \bar{\xi}(t) \in \Xi, \tag{2.27}$$

where Ξ is the reachable set of $\bar{\xi}(t)$ to $\bar{w}(t)$.

From the Equations (2.22) and (2.23) the following conditions can be derived

$$\begin{aligned} \mathcal{C}\xi(t) + u(t) \leq \quad & u_{max} + \psi_{max} & = u_{max} + \frac{d}{2}\bar{\psi}_{max}, \\ \mathcal{C}\xi(t) + u(t) \geq \quad & u_{min} + \psi_{min} & = u_{min} + \frac{d}{2}\bar{\psi}_{min}, \end{aligned}$$

and the range of the quantizer is subjected to the following condition

$$\begin{aligned} \bar{u}_{max} - \bar{u}_{min} & \leq \quad Md, \\ \left(u_{max} + \frac{d}{2}\bar{\psi}_{max}\right) - \left(u_{min} + \frac{d}{2}\bar{\psi}_{min}\right) & \leq \quad Md, \\ (u_{max} - u_{min}) + \frac{d}{2}\left(\bar{\psi}_{max} - \bar{\psi}_{min}\right) & \leq \quad Md, \\ \frac{(u_{max} - u_{min})}{M - \frac{1}{2}\left(\bar{\psi}_{max} - \bar{\psi}_{min}\right)} & \leq \quad d, \end{aligned} \tag{2.28}$$

By using the optimal values of $\bar{\psi}_{min}$ and $\bar{\psi}_{max}$ the minimium quantization interval d^{opt} is given by

$$d^{opt} = \frac{(u_{max} - u_{min})}{M - \frac{1}{2}\left(\bar{\psi}^{opt}_{max} - \bar{\psi}^{opt}_{min}\right)}, \tag{2.29}$$

as it is shown in [6]. If $M - \frac{1}{2}\left(\bar{\psi}^{opt}_{max} - \bar{\psi}^{opt}_{min}\right) \leq 0$, there is no d that satisfies the data rate constraints and the quantizer should be redesign.

Considering the system in Equation (2.26), we can obtain the value of $\bar{\psi}(t)$ by evaluating the system from $\tau = 0, 1, 2, \ldots, t-1$ and having in account that $\bar{\xi}(0) = \frac{2}{d}\xi(0) = 0$. The value of $\bar{\psi}(t)$ is given by

$$\bar{\psi}(t) = \sum_{\tau=0}^{t-1} \mathcal{C}(\mathcal{A} + \mathcal{B}\mathcal{C})^{t-1-\tau}\mathcal{B}\bar{w}(\tau), \tag{2.30}$$

and the reachable set boundaries are given as follow

$$\bar{\psi}^{opt}_{min} = -\sum_{t=0}^{\infty} \text{abs}\left(\mathcal{C}(\mathcal{A} + \mathcal{B}\mathcal{C})^{t}\mathcal{B}\right), \tag{2.31}$$

$$\bar{\psi}^{opt}_{max} = +\sum_{t=0}^{\infty} \text{abs}\left(\mathcal{C}(\mathcal{A} + \mathcal{B}\mathcal{C})^{t}\mathcal{B}\right), \tag{2.32}$$

In reality we cannot evaluate the series in Equations (2.31) and (2.32) until ∞. Thus, we approximate the values of $\bar{\psi}^{opt}_{min}$ and $\bar{\psi}^{opt}_{min}$ by evaluating the equations until a finite number L, the evaluation interval. This give us

$$\bar{\psi}^{L}_{min} = -\sum_{t=0}^{L} \text{abs}\left(\mathcal{C}(\mathcal{A} + \mathcal{B}\mathcal{C})^{t}\mathcal{B}\right), \tag{2.33}$$

$$\bar{\psi}^{L}_{max} = +\sum_{t=0}^{L} \text{abs}\left(\mathcal{C}(\mathcal{A} + \mathcal{B}\mathcal{C})^{t}\mathcal{B}\right), \tag{2.34}$$

Then, Okajima et al. in [16] defined $\hat{\psi}$ as follows

$$\begin{aligned} \hat{\psi} &:= \frac{\text{abs}(\mathcal{C}\mathrm{T})\text{abs}(\mathrm{T}^{-1}\mathcal{B})\bar{\Lambda}^{L}}{1 - \bar{\Lambda}}, \\ &= \sum_{t=L}^{\infty} \text{abs}(\mathcal{C}\mathrm{T})\text{abs}(\mathrm{T}^{-1}\mathcal{B})\bar{\Lambda}^{t}, \\ &\geq \sum_{t=L}^{\infty} \text{abs}\left(\mathcal{C}(\mathcal{A} + \mathcal{B}\mathcal{C})^{t}\mathcal{B}\right), \end{aligned} \tag{2.35}$$

where $\bar{\Lambda}$ is given by Equation (2.5) and T is the matrix used for the diagonalization of $(\mathcal{A} + \mathcal{B}\mathcal{C})$, which means that its columns are the eigenvectors of $(\mathcal{A} + \mathcal{B}\mathcal{C})$. $\hat{\psi}$ satisfies these relations

$$\bar{\psi}^{L}_{min} - \hat{\psi} \leq \bar{\psi}^{opt}_{min} \leq \bar{\psi}^{L}_{min}, \tag{2.36}$$

$$\bar{\psi}^{L}_{max} \leq \bar{\psi}^{opt}_{max} \leq \bar{\psi}^{L}_{max} + \hat{\psi}, \tag{2.37}$$

Finally replacing $\bar{\psi}_{min}^{opt}$ and $\bar{\psi}_{max}^{opt}$ in Equation (2.29) for $\left(\bar{\psi}_{min}^{L} - \hat{\psi}\right)$ and $\left(\bar{\psi}_{max}^{L} + \hat{\psi}\right)$ respectively we have

$$d^* = \frac{(u_{max} - u_{min})}{M - \hat{\psi} - \frac{1}{2}\left(\bar{\psi}_{max}^{L} - \bar{\psi}_{min}^{L}\right)}, \tag{2.38}$$

which is the approximated version of Equation(2.29). Is fair to notice that

$$d^{opt} = \lim_{L \to \infty} d^*. \tag{2.39}$$

Putting all the together Equations (2.33), (2.34) and (2.35) into Equation (2.38) we have

$$d^* = \frac{(u_{max} - u_{min})}{M - \dfrac{\mathrm{abs}(\mathcal{C}\mathrm{T})\mathrm{abs}(\mathrm{T}^{-1}\mathcal{B})\bar{\Lambda}^L}{1 - \bar{\Lambda}} - \sum_{t=0}^{L} \mathrm{abs}\left(\mathcal{C}(\mathcal{A} + \mathcal{B}\mathcal{C})^t \mathcal{B}\right)}, \tag{2.40}$$

The evaluation interval L should be selected as a big number to reduce the error due to the approximation. Therefore, d^* is used for the design of the finite-level dynamic quantizer.

2.4. Design Problem and Constraints

The finite-level dynamic quantizer design problem is formulated as follow:

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

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

This quantizer design problem is subjected to two constraints that define its space solution. Then, since the metaheuristic algorithms considered in this thesis were designed to solve unconstrained problems, the above constrained optimization problem is transformed into an unconstrained one, by using the method in [18] and [19]. In this method an optimization problem subjected to multiple constraint conditions is formulated as follows

$$\underset{\mathbf{x}\in\mathbb{F}}{\text{minimize}}\, f_{org}(\mathbf{x}), \qquad \mathbb{F} := \{\mathbf{x}|p_1(\mathbf{x}) < 0, p_2(\mathbf{x}) < 0, \ldots, p_m(\mathbf{x}) < 0\}, \tag{2.41}$$

where $f_{org}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ represents the original objective function, $\mathbf{x} \in \mathbb{R}^n$ is the design variable, $p_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is a linear/nonlinear hard constraint function and $\mathbb{F}$ represents the feasible region, i.e., the set of solutions satisfying all constraint conditions. It is assumed that $\mathbb{F}$ is not empty.

First, it is necessary to find a function $f_v(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ that satisfies simultaneously the following conditions:

(C1) $f_v(\mathbf{x}) < 0$ holds for any $\mathbf{x} \in \mathbb{F}$.

(C2) $f_v(\mathbf{x}_a) < f_v(\mathbf{x}_b)$ holds whenever $f_{org}(\mathbf{x}_a) < f_{org}(\mathbf{x}_b)$ is satisfied.

The function $f_v(\mathbf{x})$ is known as *virtual objective function* and it always exists. Notice that if $f_{org}(\mathbf{x})$ already satisfies the first condition (C1), the virtual function can be simply set as $f_v(\mathbf{x}) = f_{org}(\mathbf{x})$.

Once that $f_v(\mathbf{x})$ is selected, the constrained optimization problem in Equation (2.41) is transformed into the following unconstrained one

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}}\, f(\mathbf{x}), \qquad \text{for} \qquad f(\mathbf{x}) := \begin{cases} p(\mathbf{x}) & \text{if } p(\mathbf{x}) \geq 0, \\ f_v(\mathbf{x}) & \text{otherwise,} \end{cases} \tag{2.42}$$

where

$$p(\mathbf{x}) := \max[p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x})]. \tag{2.43}$$

The function $p(\mathbf{x})$ is called here *penalty factor*, because this function is applied when the constraints of the problem are not satisfied. Then, the constraints of the problem act as variable penalty factors. A fixed penalty factor does not provide information about how far a candidate solution is out from $\mathbb{F}$, and it usually happens that when all the candidate solutions are initially out of the feasible region there is no way for the metaheuristic to make them go inside $\mathbb{F}$. The method described above solves this problem by providing the information about how far a search point is out of $\mathbb{F}$, this information is given by $p(\mathbf{x})$ in Equantion (2.43).

In order to implement the handling constraints method to the design of finite-level dynamic quantizers it is necessary to represent the constrained optimization problem in the form shown in Equation (2.41). The original objective function is

$$f_{org}(\mathbf{x}) = E(\mathbf{x}) \quad | \quad d = d^*(\mathbf{x}), \tag{2.44}$$

where $E(\mathbf{x})$ is the performance index in Equation (2.20), d^* is the quantization interval given by Equation (2.40) and the candidate solutions $\mathbf{x}$ are composed of the n unknown elements of the matrices $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. Then, the constraints of the problem take the form

$$\begin{array}{llcl} \text{Stability constraint:} & \bar{\Lambda}(\mathbf{x}) < 1 & \Rightarrow & p_1(\mathbf{x}) = \bar{\Lambda}(\mathbf{x}) - 1, \\ \text{Data rate constraint:} & d^*(\mathbf{x}) > 0 & \Rightarrow & p_2(\mathbf{x}) = -d^*(\mathbf{x}), \end{array} \tag{2.45}$$

where $\bar{\Lambda}(\mathbf{x})$ is given by Equation (2.5).

There are many ways to select $f_v(\mathbf{x})$ satisfying the conditions (C1) and (C2). In this study we use the following virtual function

$$f_v(\mathbf{x}) = \arctan[f_{org}(\mathbf{x})] - \pi/2, \tag{2.46}$$

since it probed to be effective in the solution of the problems presented in [19].

Finally, the dynamic quantizer design problem is expressed by the unconstrained optimization problem given by Equation (2.42) with the following setting

$$f_v(\mathbf{x}) = \arctan[E(\mathbf{x})] - \pi/2 \quad \text{with} \quad d = d^*(\mathbf{x}), \tag{2.47}$$

$$p(\mathbf{x}) = \max[\bar{\Lambda}(\mathbf{x}) - 1,\ -d^*(\mathbf{x})]. \tag{2.48}$$

Chapter 3

Metaheuristics

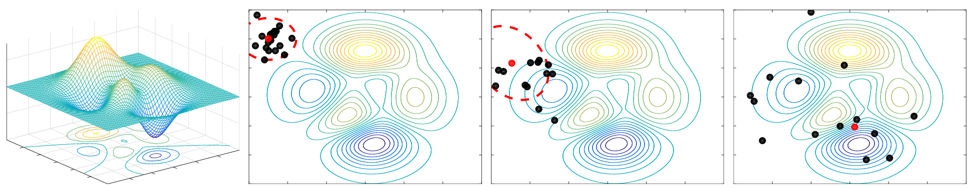
Metaheuristics are high level strategies, often nature-inspired, for exploring feasible solutions to optimization problems. The metaheuristics have several advantages: they do not make assumptions of the problem to be solved; they do not need the gradient or Hessian matrix of the function to be optimized, and many of them are easy to implement and computationally inexpensive. As expected, they have some drawbacks as well such as there is not guarantee that an optimal solution is ever found, some are very sensitive to the tuning of the hyperparameters and the possibility of getting trapped into local minima [20, 21, 22].

The metaheuristics used for the quantizers design are covariance matrix adaptation evolution strategy (CMA-ES), differential evolution (DE) and firefly algorithm (FA). The main reason to chose these metaheuristics is that they show a very good performance in the optimization of multimodal functions with local minima. Additionally, they are easy to implement and require very few hyperparameters to be tuned. A brief description of these metaheuristics and their algorithms is presented below.

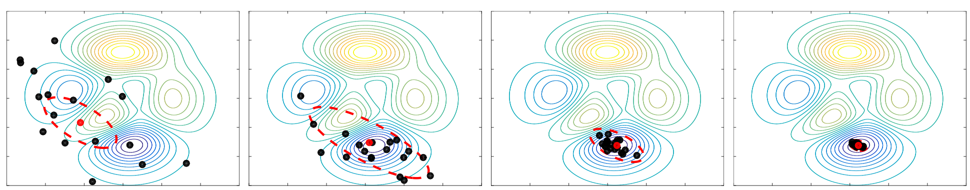
3.1. Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES)

The covariance matrix adaptation evolution strategy (CMA-ES) is an evolutionary algorithm used for solving non-linear non-convex black-box optimization problems in continuous domains. It is considered as a state-of-the-art in evolutionary computation.

In CMA-ES the possible solutions, known as search points, are generated randomly according to a multivariate normal distribution with mean $\mathbf{m}$ and covariance matrix Σ. The initial value of $\mathbf{m}$ is provided by the user or it can be selected randomly inside the search space and initially $\Sigma = \mathrm{I}$. Then, in each iteration of the algorithm, the best search points are selected and the parameters of the normal distribution are updated. Thus, the mean $\mathbf{m}$ goes toward the best solution. In the next iteration the search points are generated randomly according to the normal distribution with the new parameters [23, 24]. An example of the operation of CMA-ES is shown in Figure 3.1 for a two dimensional optimization problem.



(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 3.1: Example of the operation of CMA-ES over a two dimensional search space. It is shown how the $N = 20$ search points (black dots) and the mean (red dot) move in the search space through the generations. The red break line helps to see the adaptation of the covariance matrix, it represents a contour of the multivariate normal distribution with constant probability $p_1 = 0.1$.

The main advantage of CMA-ES over the other metaheuristics is that it does not require the tunning of almost any parameter. The only parameter left to the user is the number of search points N. By increasing N from its default value, the exploration capabilities and robustness of CMA-ES are usually improved, while the convergence time increases. Other advantages are that it has several invariance properties and it shows a very good performance in solving non-separable and ill-conditioned problems.

This algorithm has been evolving since its first development around 1995. The version applied in this study to the dynamic quantizer design is the $(\mu/\mu_W, \lambda)$ CMA-ES strategy, described in [25]. 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, although, in this thesis λ is represented by N for comparison purposes with the other algorithms. The $(\mu/\mu_W, \lambda)$ CMA-ES algorithm is shown in Algorithm 1.

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

Initialization: Given $N \in \mathbb{N}$, $k_{max} \in \mathbb{N}$, $\mathbf{m} \in \mathbb{R}^n$, the step size $\sigma \in \mathbb{R}_+$ and the initial search space $S_0 = [x_{min}, x_{max}]^n$. Initialize $\Sigma \in \mathbb{R}^{n\times n}$, $\mathbf{p}_\sigma \in \mathbb{R}^n$ and $\mathbf{p}_c \in \mathbb{R}^n$ as $\Sigma = \mathrm{I}$, $\mathbf{p}_\sigma = \mathbf{0}$ and $\mathbf{p}_c = \mathbf{0}$ 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 given in Appendix A. Then, $k = 0$.

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

$$\mathbf{x}_i \;\; = \mathbf{m} + \sigma\mathbf{y}_i, \quad \mathbf{y}_i \sim \mathcal{N}(\mathbf{0}, \Sigma) \quad \text{for } i = 1, 2, \ldots, N\,. \tag{3.1}$$

Step 2 (Selection and recombination): The objective function $f(\mathbf{x}_i)$ is evaluated for each $\mathbf{x}_i$, then the search points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and their respective $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ are ordered based on the fitness value of $\mathbf{x}_i$. 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

$$\mathbf{m} \;\; = \;\; \sum_{i=1}^{\mu} w_i\mathbf{x}_i = \mathbf{m} + \sigma\mathbf{y}_w, \tag{3.2}$$

$$\mathbf{y}_w \;\; = \;\; \sum_{i=1}^{\mu} w_i\mathbf{y}_i\,. \tag{3.3}$$

Step 3 (Step size control):

$$\mathbf{p}_\sigma \;\; \leftarrow \;\; (1 - c_\sigma)\mathbf{p}_\sigma + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\text{eff}}}\,\Sigma^{-\frac{1}{2}}\mathbf{y}_w, \tag{3.4}$$

$$\sigma \;\; \leftarrow \;\; \sigma \times \exp\left[\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma\|}{\mathsf{E}\|\mathcal{N}(0,\mathrm{I})\|} - 1\right)\right], \tag{3.5}$$

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

Step 4 (Covariance matrix adaptation):

$$h_\sigma = \begin{cases} 1 & \text{if} \frac{\|\mathbf{p}_\sigma\|}{\sqrt{1-(1-c_\sigma)^{2(k+1)}}} < \left(1.5+\frac{1}{n-0.5}\right)\mathsf{E}\|\mathcal{N}(0,\mathrm{I})\|, \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

$$\mathbf{p}_c \leftarrow (1-c_c)\mathbf{p}_c + h_\sigma\sqrt{c_c(2-c_c)\mu_{\text{eff}}}\mathbf{y}_w, \tag{3.7}$$

$$\Sigma \leftarrow (1-c_1-c_\mu)\Sigma + c_1\left(\mathbf{p}_c\mathbf{p}_c^{\mathrm{T}} + (1-h_\sigma)c_c(2-c_c)\Sigma\right) + c_\mu\sum_{i=1}^{\mu} w_i\mathbf{y}_i\mathbf{y}_i^{\mathrm{T}}\ . \tag{3.8}$$

Step 5 (Check stop condition): If the stop condition is not satisfied $k \leftarrow k+1$ and go to **Step 1**. Otherwise, terminate the algorithm and return $\mathbf{m}$ (or $\mathbf{x}_1$).

The default values of the CMA-ES' control parameters are shown in Appendix A. Meanwhile, the initial values of $\mathbf{m}$ and σ are problem dependent and should be chosen appropriately. A criteria for the selection of the initial $\mathbf{m}$ is to sample it uniformly within a region of the search space where the user expects to find the global optima $S_0 = [x_{min}, x_{max}]^n$. If the global optima is not inside S_0, the CMA-ES algorithm will still be able to find it, but the convergence speed will be slower and the probability of the algorithm to get trapped into a local minimum will increase. On the other hand, the initial step size σ should not be too small since it will reduce the algorithm performance on multimodal functions. A rule of the thumb empirically found is to make $\sigma = 0.3(x_{max} - x_{min})$.

The stop criteria of the CMA-ES algorithm is problem dependent too and it should be selected by the user. In this case the algorithm will terminate if the maximum number of iterations k_{max} is reached or if the condition number of the covariance matrix Σ exceeds 10^{14}. In order to compare the performance of CMA-ES against other metaheuristics different values of k_{max} were used.

3.2. Differential Evolution (DE)

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

Some advantages of DE are that it is very easy to implement and has only two tuning parameters: the scale factor F and the crossover constant H, apart from the number of search points N and the maximum number of iterations k_{max}. Besides, DE shows very good exploration capacities and converge very fast to the global optima. The DE algorithm is shown in Algorithm 2

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 search space $S_0 = [x_{min}, x_{max}]^n$. Set $k = 0$ then select randomly N search points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ in the search space.

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

$$l_k \quad = \arg\min_{i \in \{1,2,\ldots,N\}} f(\mathbf{x}_i). \tag{3.9}$$

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

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

$$\mathbf{M}_i = \mathbf{x}_{base} + F(\mathbf{x}_{\tau_{1,i}} - \mathbf{x}_{\tau_{2,i}}), \tag{3.10}$$

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 $\mathbf{x}_i$ and $\mathbf{M}_i$ a trial vector $\mathbf{L}_i$ is generated by:

$$\mathbf{L}_{i,j} = \begin{cases} \mathbf{M}_{i,j} & \text{if } \rho_{i,j} \leq H \text{ of } j = j_{rand}, \\ \mathbf{x}_{i,j} & \text{otherwise,} \end{cases} \tag{3.11}$$

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:

$$\mathbf{x}_i \leftarrow \begin{cases} \mathbf{L}_i & \text{if } f(\mathbf{L}_i) \leq f(\mathbf{x}_i), \\ \mathbf{x}_i & \text{otherwise,} \end{cases} \tag{3.12}$$

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

3.3. Firefly Algorithm (FA)

The firefly algorithm is a population based metaheuristic that is inspired in the flashing behavior of tropical fireflies. It was first introduced in [27], and it has been evolving since then. The algorithm is based on the following three idealize rules about the fireflies behavior:

1. Each firefly will be attracted to other fireflies regardless of their sex.
2. The attractiveness between two fireflies is proportional to their brightness and it decreases as the distance between them increases. The less brighter firefly will move towards the brighter one and the brightest in the swarm will move randomly.
3. The brightness of a firefly is determined by the landscape of the objective function.

Some advantages of the FA algorithm over the other metaheuristics are the automatic subdivision of the set of search points, the ability to deal with multimodality and the online tuning of the parameters to control the randomness in the update law [28].

The FA algorithm has the following parameters: the initial randomness scaling factor α_0, the initial attractiveness β_0, the absorption coefficient γ and the cooling factor δ. The FA implementation used in this study is shown in Algorithm 3.

Algorithm 3 : FA

Initialization: Given $N \in \mathbb{N}$, $k_{max} \in \mathbb{N}$, α_0, β_0, $\gamma \in [0,\infty)$ (in practice $[0.1,10]$), $\delta \in (0,1)$ and the initial search space $S_0 = [x_{min}, x_{max}]^n$. Set $k = 0$, then select randomly N search points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ within S_0.

Step 1: The objective function $f(\mathbf{x}_i)$ is evaluated for each $\mathbf{x}_i$, then the search points are sorted in ascending order based on their fitness value. If $k = k_{max}$ then $\mathbf{x}_N$ is the solution of the algorithm, if not go to **Step 2**.

Step 2: Find the limits of the search space. Considering that $\mathbf{L}_b = \{l_1, l_2, \ldots, l_n\}$ and $\mathbf{U}_b = \{u_1, u_2, \ldots, u_n\}$, then for $j = \{1, 2, \ldots, n\}$:

$$l_j = \min_{x_j \in \{x_{i,j} | i=1,2,\ldots,N\}} x_j, \tag{3.13}$$

$$u_j = \max_{x_j \in \{x_{i,j} | i=1,2,\ldots,N\}} x_j, \tag{3.14}$$

where $x_{i,j}$ is the j^{th} component of the search point $\mathbf{x}_i$.
Step 3: Reduce the randomness:

$$\alpha = \alpha_0 \delta^k, \tag{3.15}$$

Step 4: For $i = \{1, 2, \ldots, (N-1)\}$ the following update law is applied to each search point:

$$\begin{aligned} \mathbf{x}_i &\leftarrow \mathbf{x}_i + \sum_{j=i+1}^{N} \left[\beta_0 \exp\left[-\gamma \left\|\mathbf{x}_j - \mathbf{x}_i\right\|_2^2\right] \left(\mathbf{x}_j - \mathbf{x}_i\right) + \alpha \varepsilon_j\right], & (3.16) \\ \varepsilon_j &= \rho_j (\mathbf{U}_b - \mathbf{L}_b), & (3.17) \end{aligned}$$

where $\rho_j \in [-0.5, 0.5]$ is a random number uniformly distributed. Then, make $k \leftarrow k+1$ and go to **Step 1**.

Chapter 4

Finite-level Quantizer Design Based on Metaheuristics

In this chapter it is explained the methodology and settings for the implementation of the finite-level dynamic quantizer design method. Then, by numerical examples it is verified that the designed quantizers work properly. After that, the methods based on different metaheuristics are compared among each other and with previously developed methods in order to evaluate their performance.

4.1. Methodology and Settings

The quantizer design method consists in finding the values of $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and d of the dynamic quantizer given by the Equation (2.2) that solve the unconstrained optimization problem given by Equation (2.42). The optimization is carried out by using one of the metaheuristics described in Chapter 3 (CMA-ES, DE or FA). The objective function for the optimization is $f(\mathbf{x})$ in Equation (2.42) with $f_v(\mathbf{x})$ and $p(\mathbf{x})$ given by Equations (2.47) and (2.48) respectively. In there, the design variable $\mathbf{x} \in \mathbb{R}^n$ is constructed with the n unknown elements of the matrices $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$.

Then, given the plant P with order n_p, the first step is to choose the order and form of the quantizer. The selection of the quantizers' order can be subjected to the precision wanted or the computational capacity available to implement it.

Once the order of the quantizer is chosen, it is time to set the form of the quantizer. Of course, it is possible to consider all the elements of $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ as variables for the optimization problem, in that case we will have $n = 2n_q + n_q^2$ variables in total. However, this setting is not optimal since it is possible to represent the same quantizer into its equivalent *canonical controllable* form. Thus, it will be better to express the quantizer directly into its canonical form because in that case the amount of variables will be less ($n = 2n_q$) and the metaheuristic will found the solutions faster and more precisely. With this setting the form of the quantizer is given by

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1 & x_2 & x_3 & \cdots & x_{\frac{n}{2}} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} x_{\frac{n}{2}+1} & x_{\frac{n}{2}+2} & x_{\frac{n}{2}+3} & \cdots & x_n \end{bmatrix} \tag{4.1}$$

Thereby, the form of the design variable or search points for the metaheuristic is $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$. Note that, although d is regarded as a design parameter as well, its value is a function of $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, since $d = d^*$ and d^* is given by the Equation (2.40).

4.2. Numerical Examples

With the purpose of verifying the effectiveness of the proposed metaheuristic based quantizer design method, we performed a series of simulations. The metahueristics used are CMA-ES, DE, FA and PSO. The last one, PSO, was employed in a previous study [14] and we used it here for comparison purposes. Appendix B shows the PSO algorithm.

Considering the system in Figure 2.1, two different plants P were selected. One is the second order system:

$$P_1(s) = \frac{s+20}{s^2+3s+2}, \tag{4.2}$$

and the other is the third order system:

$$P_2(s) = \frac{s+10}{s^3+6s^2+9s+10}. \tag{4.3}$$

For both plants the sampling time is $\Delta t = 0.1[s]$, and the parameters for the quantizer are: $U = [1, -1]$, $M = 2$ and $L = 150$. For P_1 and P_2, the form of the quantizer constant matrices are given by

$$\mathcal{A}_1 = \begin{bmatrix} 0 & 1 \\ x_1 & x_2 \end{bmatrix}, \quad \mathcal{B}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathcal{C}_1 = \begin{bmatrix} x_3 & x_4 \end{bmatrix}, \tag{4.4}$$

$$\mathcal{A}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ x_1 & x_2 & x_3 \end{bmatrix}, \quad \mathcal{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathcal{C}_2 = \begin{bmatrix} x_4 & x_5 & x_6 \end{bmatrix}, \tag{4.5}$$

respectively. Then, the form of the search points for the metaheuristic algorithms are $\mathbf{x}^{(1)} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^\top$ for P_1 and $\mathbf{x}^{(2)} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{bmatrix}^\top$ for P_2, which implies that the dimension of the optimization problems are $n_1 = 4$ and $n_2 = 6$, respectively.

The hyperparameters of the metaheuristics employed are given in the Table 4.1. The metaheuristics' performance is subjected to the value of the hyperparameters. In this case we chose the control parameters based on the recommended values of previous studies. For DE we used the same parameter as in [13], for FA we follow the recomentaion in [28] and for PSO in order to compare the results we used the same hyperparameter as in [14]. In the case of CMA-ES the hyperparameters are the initial values of σ and $\mathbf{m}$ but they can be generated automatically from the initial search interval S_0 according to [25].

Table 4.1: Hyperparameters used for the simulations.

CMA-ES	DE	FA	PSO
$\sigma_0 = 0.3$	$F = 0.6$	$\alpha_0 = 0.25$	$\chi_0 = 0.9$
	$H = 0.9$	$\beta_0 = 1$	$\chi_1 = 1$
		$\gamma = 0.1$	$\chi_2 = 1$
		$\delta = 0.97$	

For DE, FA and PSO the initial search points (members of the first generation) are generated randomly inside the initial search space $S_0 = [-1, 1]^n$ by an uniform probability distribution. Additionally, for PSO the initial velocities of the particles are randomly initialized in the range $V_0 = [0, 1]^n$. For CMA-ES the initial mean $\mathbf{m}_0$ is selected randomly using a uniform probability distribution within $S_0 = [-1,1]^n$ and the initial search points are sampled from the normal probability distribution $\mathcal{N}\left(\mathbf{m}, \sigma^2 \mathrm{I}\right)$.

The simulations are performed by trying $N_{run} = 50$ runs of the algorithms for each combination of the parameters $N = \{50, 100, 500\}$ and $k_{max} = \{50, 100, 200, 500, 1000\}$. The results of the simulations are summarized in the Tables 4.2, 4.3, 4.4 and 4.5. In the tables, for each plant and for each combination of N and k_{max}, the best $E_{min}(Q)$, the mean and the standard deviation of $E_{min}(Q)$, the success rate (SR) in percentage $[\%]$ and the average execution time in seconds $[s]$ are shown. The success rate, which is used as a measurement of the algorithm performance, is the ratio of the number of runs with the best solution to the total number of runs of the algorithm N_{run}.

Table 4.2: 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	Time	Best	Mean	St. dev.	SR	Time
50	50	0.5091	0.5870	0.1059	66	5.9	0.0719	0.0948	0.0335	0	8.8
50	100	0.5091	0.5395	0.0814	86	12.2	0.0691	0.0797	0.0302	82	18.2
50	200	0.5091	0.5800	0.1121	70	18.5	0.0691	0.0751	0.0238	94	37.8
50	500	0.5091	0.5438	0.0861	86	19.6	0.0691	0.0733	0.0200	92	91.6
50	1000	0.5091	0.5736	0.1089	74	17.5	0.0691	0.0814	0.0328	86	173.9
100	50	0.5091	0.5152	0.0353	98	12.7	0.0692	0.0704	0.0009	38	18.3
100	100	0.5091	0.5091	0.0000	100	26.7	0.0691	0.0691	0.0000	100	38.1
100	200	0.5091	0.5140	0.0347	98	29.8	0.0691	0.0691	0.0000	100	78.1
100	500	0.5091	0.5140	0.0347	98	29.0	0.0691	0.0691	0.0000	96	190.4
100	1000	0.5091	0.5190	0.0486	96	28.4	0.0691	0.0691	0.0000	98	388.6
500	50	0.5091	0.5091	0.0000	100	82.2	0.0691	0.0691	0.0000	98	86.0
500	100	0.5091	0.5091	0.0000	100	141.1	0.0691	0.0691	0.0000	100	180.0
500	200	0.5091	0.5091	0.0000	100	142.5	0.0691	0.0691	0.0000	100	356.5
500	500	0.5091	0.5091	0.0000	100	143.5	0.0691	0.0691	0.0000	100	911.3
500	1000	0.5091	0.5091	0.0000	100	146.5	0.0691	0.0691	0.0000	100	1715.2

The simulation results show that the design methods based on CMA-ES and DE are quite superior than the ones based on FA and PSO. The design method based on CMA-ES gave the best results for both plants, with the smallest performance index value and the highest success rate. The results of the design method based on DE are very good as well. Both methods CMA-ES and DE overcome the design method previously developed based on PSO. And the method based on FA show not to be

Table 4.3: 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	Time	Best	Mean	St. dev.	SR	Time
50	50	0.5091	0.6051	0.1197	62	7.8	0.0692	0.1113	0.0461	4	7.9
50	100	0.5091	0.5885	0.1158	68	17.2	0.0691	0.0949	0.0419	30	16.2
50	200	0.5091	0.5984	0.1191	64	34.7	0.0691	0.0963	0.0433	32	33.9
50	500	0.5091	0.5896	0.1153	68	95.1	0.0691	0.0966	0.0432	44	86.0
50	1000	0.5091	0.5741	0.1086	74	181.9	0.0691	0.0838	0.0345	62	175.3
100	50	0.5091	0.5641	0.1026	78	16.4	0.0692	0.0871	0.0358	28	16.0
100	100	0.5091	0.5885	0.1158	68	33.5	0.0691	0.0817	0.0324	68	33.3
100	200	0.5091	0.5666	0.1023	78	67.6	0.0691	0.0853	0.0366	80	69.1
100	500	0.5091	0.5488	0.0910	84	170.8	0.0691	0.0832	0.0347	84	175.5
100	1000	0.5091	0.5438	0.0861	86	349.9	0.0691	0.0852	0.0367	84	350.9
500	50	0.5091	0.5190	0.0486	96	69.8	0.0692	0.0745	0.0198	76	77.5
500	100	0.5091	0.5438	0.0861	86	139.9	0.0691	0.0711	0.0140	98	174.6
500	200	0.5091	0.5425	0.0813	86	286.1	0.0691	0.0712	0.0140	96	352.5
500	500	0.5091	0.5438	0.0861	86	686.1	0.0691	0.0791	0.0300	90	907.0
500	1000	0.5091	0.5438	0.0861	86	1408.7	0.0691	0.0691	0.0000	100	1730.8

appropriated for the design of the finite-level dynamic quantizer. In Section 4.3 these results will be discussed in more detail.

Next, we evaluate the performance of the designed quantizer in the system in Figure 2.1. We take the parameters of the quantizer from the simulations performed, in particular we chose the quantizers that gave the smallest performance indices for each plant. Both quantizers were found for the design method based on CMA-ES, and their values for P_1 and P_2 are shown as follow:

$$\begin{aligned} \mathcal{A}_1 &= \begin{bmatrix} 0 & 1 \\ -0.7413 & 1.7241 \end{bmatrix}, \quad \mathcal{B}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ \mathcal{C}_1 &= \begin{bmatrix} 0.6532 & -1.1619 \end{bmatrix}, \quad d_1 = 3.1082. \end{aligned} \tag{4.6}$$

$$\begin{aligned} \mathcal{A}_2 &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.1391 & -1.1584 & 2.0022 \end{bmatrix}, \quad \mathcal{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \\ \mathcal{C}_2 &= \begin{bmatrix} -0.2078 & 0.9785 & -1.1550 \end{bmatrix}, \quad d_2 = 3.2337. \end{aligned} \tag{4.7}$$

Table 4.4: Simulation results for the second and third order plants by FA ($N_{run} = 50$ trials).

		Second order system, P_1					Third order system, P_2				
N	k_{max}	Best	Mean	St. dev.	SR	Time	Best	Mean	St. dev.	SR	Time
50	50	0.6469	0.9025	0.1486	0	3.2	0.1744	0.2102	0.0233	0	5.8
50	100	0.7562	0.7958	0.0698	0	6.4	0.1546	0.1763	0.0148	0	12.4
50	200	0.7322	0.7852	0.0989	0	12.5	0.1624	0.1696	0.0031	0	26.0
50	500	0.5320	0.7449	0.0487	4	30.6	0.1614	0.1722	0.0078	0	66.6
50	1000	0.5324	0.7575	0.0922	4	59.3	0.1479	0.1716	0.0132	0	134.9
100	50	0.7690	0.9171	0.1569	0	5.8	0.1711	0.2063	0.0247	0	12.0
100	100	0.7497	0.8222	0.1311	0	11.5	0.1644	0.1735	0.0052	0	26.6
100	200	0.6737	0.7753	0.0587	0	22.8	0.1533	0.1699	0.0047	0	55.8
100	500	0.5783	0.7625	0.0442	2	56.7	0.1579	0.1692	0.0032	0	144.9
100	1000	0.5340	0.7629	0.0670	2	113.3	0.1438	0.1715	0.0090	0	285.7
500	50	0.7633	0.8128	0.0964	0	28.9	0.1668	0.1980	0.0167	0	81.2
500	100	0.6695	0.7821	0.0431	0	56.8	0.1479	0.1723	0.0061	0	170.2
500	200	0.7329	0.7620	0.0190	0	113.7	0.1341	0.1689	0.0057	0	349.3
500	500	0.6152	0.7672	0.0724	0	282.4	0.1200	0.1671	0.0087	0	880.0
500	1000	0.5684	0.7440	0.0533	8	577.9	0.1546	0.1697	0.0037	0	1776.7

The selected input signal, which has range $U = [-1, 1]$, is the following:

$$u(t) = 0.7\sin(0.3t) + 0.3\sin(0.4t). \tag{4.8}$$

The results are shown in Figure 4.1, the upper figures show the time responses for the second order system and lower ones for the third order system. Notice that in both cases the quantized input signal $v(t)$ has only two levels since it was specified that $M = 2$. For the second order system Figure 4.1b shows how the real output $y_q(t)$ follows closely the desired output $y(t)$ and the error between them is very small. In fact, the maximum error registered is $\max_{t\in\{1,2,\dots,150\}} \text{abs}\left[y_q(t) - y(t)\right] = 0.3462$. Then, since for the respective designed quantizer $E(Q) = 0.5091$, it is verified that the maximum error in this example is less than the performance index $E(Q)$. The same situation happens for the third order system , where, $\max_{t\in\{1,2,\dots,150\}} \text{abs}\left[y_q(t) - y(t)\right] = 0.0319$ and $E(Q) = 0.0691$.

Table 4.5: 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	Time	Best	Mean	St. dev.	SR	Time
50	50	0.7291	0.7910	0.0271	0	6.4	0.1386	0.2052	0.0340	0	5.6
50	100	0.6467	0.7612	0.0196	0	13.4	0.1297	0.1814	0.0157	0	11.2
50	200	0.7437	0.7577	0.0027	0	28.2	0.0867	0.1650	0.0182	0	23.5
50	500	0.7015	0.7556	0.0080	0	76.8	0.0713	0.1579	0.0300	0	66.6
50	1000	0.5731	0.7518	0.0274	2	159.4	0.0706	0.1529	0.0333	0	151.7
100	50	0.7300	0.7815	0.0211	0	10.7	0.1047	0.1867	0.0195	0	11.4
100	100	0.6960	0.7593	0.0101	0	22.8	0.1556	0.1734	0.0063	0	23.7
100	200	0.6160	0.7491	0.0316	0	49.3	0.0789	0.1557	0.0293	0	48.5
100	500	0.5150	0.7476	0.0429	4	140.6	0.0716	0.1430	0.0405	0	135.7
100	1000	0.6261	0.7535	0.0194	0	306.0	0.0729	0.1510	0.0362	0	299.5
500	50	0.5204	0.7550	0.0366	2	52.8	0.0920	0.1660	0.0209	0	56.6
500	100	0.5561	0.7447	0.0436	4	110.7	0.0774	0.1505	0.0342	0	115.0
500	200	0.5114	0.7458	0.0477	4	234.5	0.0712	0.1491	0.0374	0	249.2
500	500	0.5097	0.7025	0.0964	22	566.6	0.0709	0.1357	0.0426	0	698.6
500	1000	0.5091	0.6777	0.1132	32	1108.1	0.0701	0.1305	0.0453	0	1477.3

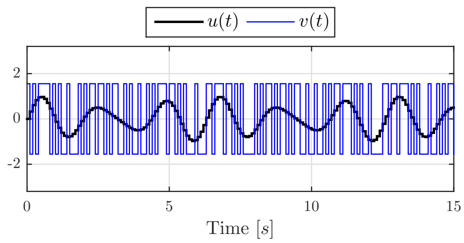
4.3. Comparison Among Metaheuristics

In this section we discuss in detail the results from the developed simulations. A comparison among the design methods based on CMA-ES, DE, FA and PSO is developed in terms of the success rate, convergence behavior and execution time. The data for this analysis can be found in the Tables 4.2, 4.3, 4.4 and 4.5.

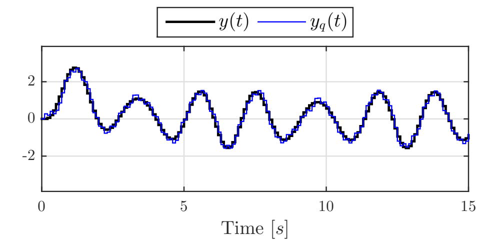
4.3.1 Success Rate Comparison

For the second order system the best solution found for CMA-ES and DE is the same $E_{min}(Q_{CMA-ES}) = E_{min}(Q_{DE}) = 0.509057$, for PSO is slightly bigger $E_{min}(Q_{PSO}) = 0.509073$ and for FA is quite bigger $E_{min}(Q_{FA}) = 0.531969$. For the third order system this relation is the same $E_{min}(Q_{CMA-ES}) = E_{min}(Q_{DE}) = 0.069132$, $E_{min}(Q_{PSO}) = 0.070130$ and $E_{min}(Q_{FA}) = 0.119999$.

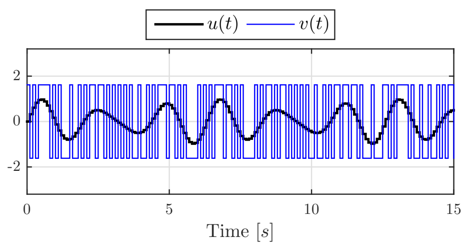
The best solutions are provided by CMA-ES and DE. These solutions are the same in both cases indicating that they both have the exploration capacities to solve the quantizer design problem. Now, in the case of PSO for the second order system, the



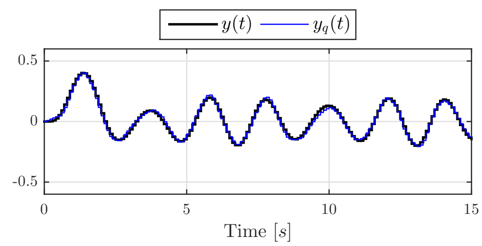
(a) Input signals of second order plant P_1.



(b) Output signals of second order plant P_1.



(c) Input signals of third order plant P_2.



(d) Output signals of third order plant P_2.

Figure 4.1: System responses when the control signal $u(t)$ is applied to the plants P_1 and P_2. The black lines represent the signals of the system without quantization. Meanwhile, the blues ones are the signals when the quantization is performed.

best solution is very close the one obtained by CMA-ES and DE, while for the third order system, the difference between the results is not negligible. In contrast, the solutions provided by FA are relatively big in comparison with the others.

A more important point is the success rate when comparing these algorithms. The success rate gives an idea about how effective is an algorithm. A low success rate indicates that the algorithm gets trapped into local minima, and that the probability to find the global minimum is small. Thus, an algorithm with high success rate is reliable. The success rates of the quantizer design algorithms are shown in Figure 4.2 for the second order system and in Figure 4.3 for the third order system, this values are the same as in the tables.

They show how the performances of CMA-ES and DE are quite better than the performances of PSO and FA for the design of the quantizer. For the second order system the success rates of CMA-ES and DE are always over 60%. Meanwhile, the

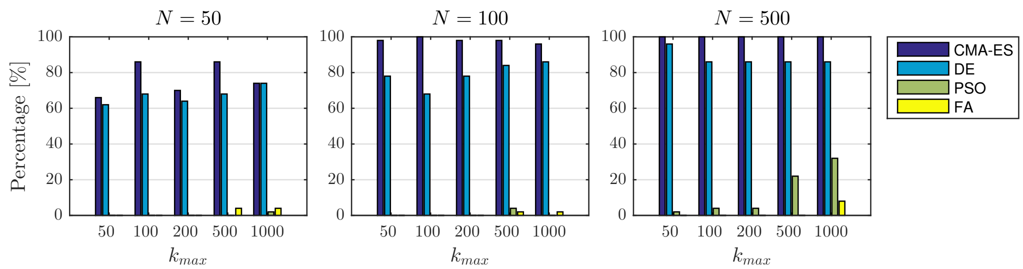


Figure 4.2: Success rate for the second order plant (P_1).

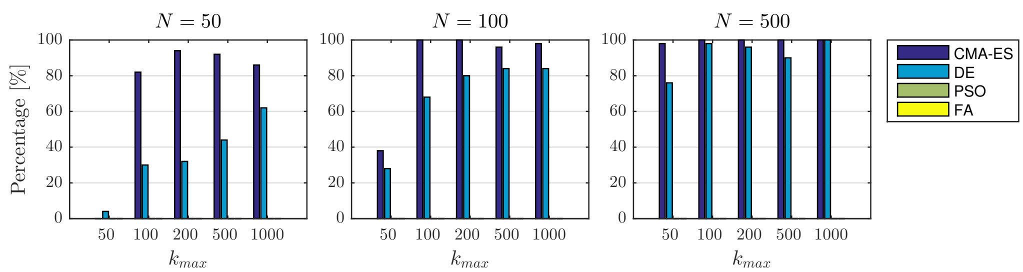


Figure 4.3: Success rate for the third order plant (P_2).

success rates of PSO are less than 40% and for FA less than 10%. For the third order system the success rates of PSO and FA are 0% in all the cases. Thus, it is verified that the quantizer design based on CMA-ES and DE are reliable methods while the ones based on PSO and FA are not. Now, for both plants the success rates obtained with CMA-ES are better than the ones obtained with DE, specially when the number of search points N and (or) the maximum number of generations k_{max} are small, for bigger values of N and k_{max} their success rates are very close to each other. Then, it is fair to say that the design method based on CMA-ES is better and more reliable than the one based on DE.

An interesting result from the tables is the values of the mean. The means were taken over the $N_{run} = 50$ trials of the algorithm. For the CMA-ES based method the mean for both order systems is very close to the best value or $E_{min}(Q)$, same for DE, but not as close as in CMA-ES. In contrast, for PSO the best value and the mean are different for the second order system, since the mean is closer to the local minima, in the third order case they are both close to the local minima. The behavior of FA is similar to PSO. Another remarkable fact taken from the tables is that the standard deviation in the CMA-ES based method is very small, almost zero, indicating how close the results are from the mean that at the same time is almost as the best result. The success rates are a reflection of these results.

4.3.2 Convergence Time Comparison

The convergence behavior of the algorithms is shown in Figure 4.4 for the second order system and in Figure 4.5 for the third order system by sequences of $E(Q)$ against the generation number k. In these figures there were selected examples of trials of the algorithms for each value of N and $k_{max} = 500$, although in the figures the sequences are shown until $k = 50$. The only criteria used for the selection of this sequences among the respective 50 trials was that the initial $E(Q)$ should be close to $E(Q) = 1$. In each case the figures show that the methods based on CMA-ES and DE converge fast to the global optima, and that the design methods based on PSO and FA converge more slowly to the local optima.

In terms of execution time PSO and FA shows to be faster than DE for both systems, considering the results of Table 4.3, 4.4 and 4.4. However, comparing PSO and FA execution times, it is notable that for the second order system the FA algorithm is quite faster than PSO but for the third order system FA is slower than PSO. This is due to the time complexity of the algorithms, thus, increasing the dimension of the problem will make FA even more slower than PSO. For CMA-ES the situation is a little ambiguous. For the second order system, it shows to be quite faster than the others methods, moreover the execution time does not depend on k_{max} after a certain value. On the other hand, for the third order system, CMA-ES is slower than the other algorithms, and the execution time increases with k_{max}. This behavior is explained with the double termination condition of the CMA-ES algorithm that are implemented in this study. The first one is that the generation number k reaches k_{max}, this is the only termination condition for DE, FA and PSO, and this explains why for the third order system the execution time depends on k_{max}. The second termination condition of CMA-ES is that the condition number of the covariance matrix Σ exceeds 10^{14}, clearly this condition is responsible for the behavior in the case of the second order system. The CMA-ES policy is that the algorithm should be stopped whenever it becomes a waste of CPU-time to continue. Then, it is possible to add more termination conditions for the CMA-ES algorithm, and in this way reduce its execution time.

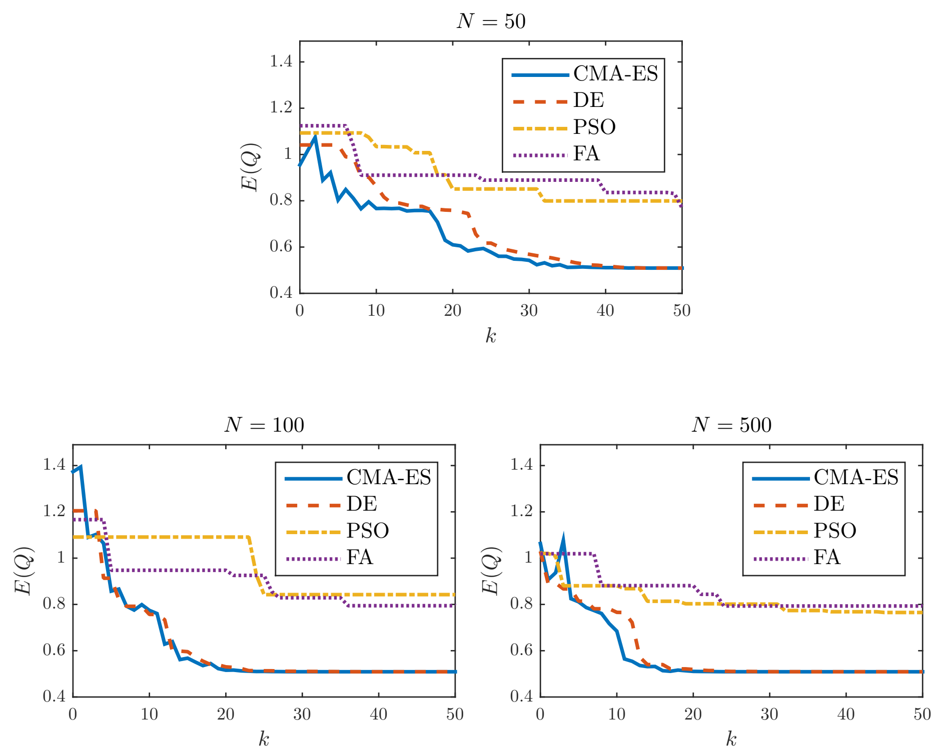


Figure 4.4: Convergence behavior of the different metaheuristics for the second order plant P_1.

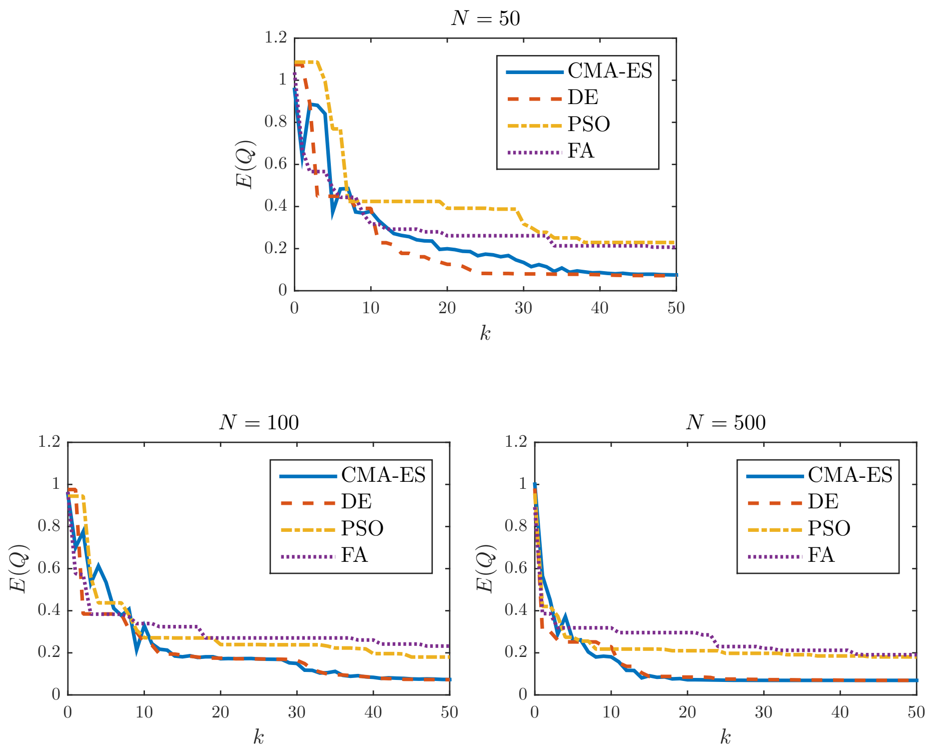


Figure 4.5: Convergence behavior of the different metaheuristics for the third order plant P_2.

Chapter 5

Discussion

This chapter presents a discussion about two topics. The first topic is about the reduction of the parameters in the design of the finite-level dynamic quantizer. The second topic is about the development of a novel type of dynamic quantizer that operates in an event-based manner. In the analysis of these topics the quantizer design method presented in Chapter 4 is extensively used.

5.1. Reduction of Design Parameters for Finite-level Dynamic Quantizers

Although the proposed quantizer design methods are effective, it is still needed an analytic expression of the optimal dynamic quantizer. In fact, the analytic expression will make easy the theoretical analysis of the system with the quantizer. Even if it is not possible to find an analytic expression for the optimal finite-level dynamic quantizer, at least it will be helpful to reduce the number of parameters for its design.

For the case in which no data rate constraints are considered in the desing of the quantizer there exists an analytic expresion for the optimal dynamic quantizer. This expresion was presented in [8] and its form is as follow:

$$Q^* : \begin{cases} \mathcal{A} & = \mathrm{A}, \\ \mathcal{B} & = \mathrm{B}, \\ \mathcal{C} & = -(\mathrm{CB})^{-1}\mathrm{CA}, \end{cases} \tag{5.1}$$

where the value of the quantization interval d is given by the designer or can be chosen arbitrarily. An important fact showed in Equation (5.1) is that the structure of the optimal quantizer includes the model of the plant P.

Then, the question is: Does the structure of the optimal finite-level dynamic quantizer have the information of the plant P? In order to answer this question, we consider a possible optimal structure of the quantizer based on the relation in Equation (5.1) and compared the results with the most general structure of the quantizer.

The general structure of the quantizer is as follows

$$\mathcal{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n_q} \\ a_{21} & a_{22} & \dots & a_{2n_q} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_q1} & a_{n_q2} & \dots & a_{n_qn_q} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{n_q1} \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n_q} \end{bmatrix}, \tag{5.2}$$

where all the elements of $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are variables. Clearly, this structure of the quantizer comprehends all the other possible structures, and its solution space encloses all the other possible structures' solution space. Moreover, the solution space of this general structure contains the optimal solution of the quantizer. We refer to the structure in Equation (5.2) as *Case 1*.

On the other hand, the structure that we consider as possible optimal solution is given by

$$\begin{aligned} \mathcal{A} &= \mathrm{A}, \\ \mathcal{B} &= \mathrm{B}, \\ \mathcal{C} &= \begin{bmatrix} c_{11} & \dots & c_{1n_q} \end{bmatrix}. \end{aligned} \tag{5.3}$$

There, the only variables are the elements of $\mathcal{C}$ and we refer to this structure of the quantizer as *Case 2*. In the Case 2, we assume that the optimal quantizer has the model of P in the same way as in Equation (5.1).

Considering the Cases 1 and 2, if $E_1(Q) = E_2(Q)$, the Case 2 has the same optimal solution as Case 1 (the sub-indexes of $E(Q)$ represent the respective case). This will imply that the optimal solution includes the model of P. Figure 5.1 shows our hypothesis represented in terms of the solution spaces.

The correctness or wrongness of the hypothesis previously formulated is verified by means of several numerical simulations. In these simulations, the $E(Q)$ and the

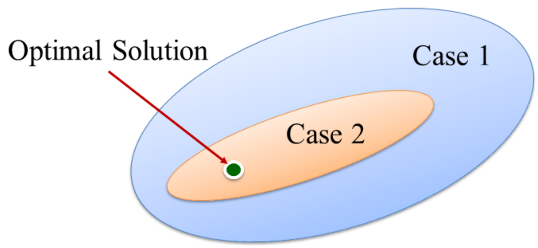


Figure 5.1: Representation of the solution space for the Cases 1 and 2.

parameters of the quantizer are located, by using the finite-level dynamic quantizer design method based on DE for the Cases 1 and 2 and compared the results. For that purpose we made use of 9 different SISO-LTI discrete plants with first, second and third order. They are shown in Table 5.1.

Table 5.1: Set of plants used for the design parameters reduction experiments.

1st order plants	2nd order plants	3rd order plants
$P_{11} = \frac{1}{s+6}$	$P_{21} = \frac{s+20}{s^2+3s+2}$	$P_{31} = \frac{1}{s^3+10s^2+31s+30}$
$P_{12} = \frac{s+2}{3s+10}$	$P_{22} = \frac{2s+25}{s^2+4s+25}$	$P_{32} = \frac{s+10}{s^3+6s^2+9s+10}$
$P_{13} = \frac{s-10}{s+5}$	$P_{23} = \frac{s^2+14s+479}{s^2+7s+74}$	$P_{33} = \frac{3s^2+9s+12}{s^3+5s^2+21s+22}$

These plants were divided in two groups based on the type of system: minimum phase (MP) and non-minimum phase (NMP) systems. A MP system is the system whose all zeros are in the unit circle. Meanwhile, a NMP system has at least one zero which is outside the unit circle. The groups can be seen in the second column of Table 5.2.

The setting of the simulations is as follow: the sampling time is $\Delta t = 0.1[s]$, and the parameters for the quantizer are: $U = [1, -1]$, $M = 2$ and $L = 150$. For the DE algorithm the parameters are $F = 0.6$ and $H = 0.9$. The simulations were made $N_{run} = 50$ times for each combination of parameters $N = \{200, 500, 1000\}$ and $k_{max} = \{200, 500, 1000\}$. The results of these simulations are presented in Table 5.2.

Table 5.2: Results for the design parameters reduction experiments. The quantizer design method use for these experiments is the one based on DE with $N_{run} = 50$ trials.

		Case 1		Case 2	
Plants	Type	$E(Q)$	d	$E(Q)$	d
P_{11}	MP	0.051818263	1.378180841	0.051818263	1.378180841
P_{12}	MP	0.029448186	1.558277203	0.029448186	1.558277203
P_{13}	NMP	0.847100102	1.435266598	0.847100102	1.435266598
P_{21}	MP	0.509057352	3.108213923	0.509117253	3.13869626
P_{22}	MP	0.434145776	2.197980761	0.434250438	2.192987642
P_{23}	NMP	3.586124375	1.552648305	3.599193529	1.56715524
P_{31}	NMP	0.002885156	2.585089764	0.002995108	2.623228209
P_{32}	NMP	0.06913235	3.2337101	0.074049374	3.308945455
P_{33}	MP	0.282408138	1.879615337	0.283939665	1.868429725

The simulations results presented in Table 5.2 can be summarized as follows:

$$E(Q)_1 = E(Q)_2 : 1^{st} \text{ Order MP and NMP,} \tag{5.4}$$

$$E(Q)_1 \approx E(Q)_2 : \begin{cases} 2^{nd} & \text{Order NMP,} \\ 3^{rd} & \text{Order NMP,} \end{cases} \tag{5.5}$$

$$E(Q)_1 < E(Q)_2 : \begin{cases} 2^{nd} & \text{Order MP,} \\ 3^{rd} & \text{Order MP,} \end{cases} \tag{5.6}$$

From this, it is possible to make the following conclusions: First, according to Equation (5.4) for the first order systems our hypothesis is verified and the optimal quantizer is given by Equation (5.3). For the second and third order systems the results show that the performance indices for both cases are different. It is arguable that the metaheuristic couldn't approximate well enough to the proposed optimal quantizer because of the amount of variables. Thus, in order to discard this possibility we considered a *Case 3* in which the quantizer to be designed is in the canonical controllable form as shown in Equation (4.1). The results were the same as in Case 1 (with all the decimals). Therefore, we can say that the hypothesis that we formulated is not true for the second and third order systems, and most probably for higher order systems.

On the other hand, it is still possible that the optimal finite-level dynamic quantizer is relate to the plant somehow, but no directly as we assumed in this section.

5.2. Extension to the Event Triggered Quantizer

A characteristic feature of the NCSs is that the network can be shared. Thus, several control or automation systems can be running over the same network including the company's intranet. Then, the need of optimizing the usage of the network resources is of paramount importance. In addition, energy constraints set severe restrictions on the operation of many types of NCSs, specially the ones in which sensors or actuators are battery powered.

Motivated by the need of generating more efficient systems in terms of bandwidth and energy consumption, here we extend the study developed so far in this thesis to consider a switched-type finite-level dynamic quantizer that is actuated by events. This type of quantizer is known as *event-triggered quantizer*. An event is some type of variation of some characteristics of the system.

Until now, we were considering the situation in which the quantizer is time invariant and the quantization and communication are performed periodically. Figure 5.2 shows the periodic operation of the quantizer.

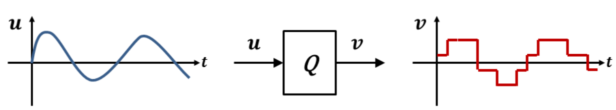


Figure 5.2: Example of a synchronous dynamic quantizer operation.

The event-triggered quantizer is time variant and operates using aperiodic sampling. Figure 5.3 shows an example of the operation of this type of quantizers, where σ is the trigger signal and is provided by an event generator. In this example σ takes only two values ON and OFF. When the value is ON the quantizer sends the data over the network, when the value is OFF no data is sent. The event generator asks for data when necessary, optimizing in this way the usage of the network.

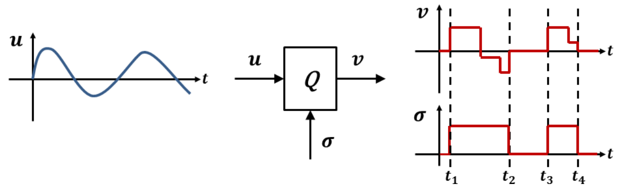


Figure 5.3: Example of an asynchronous dynamic quantizer operation.

The trigger signal σ can be a function of the state of the plant, the time or some other variables in the system. In particular, we consider here an event-triggered quantizer that reacts to variations in the capacity of the channel or more accurately to the availability of bandwidth. The channel capacity is represented in this case by N_b which is the number of bits that can be transmitted in the channel per sampling time. Consequently, we call this quantizer *data rate driven event-triggered quantizer*. Its structure and the way that it operates can be seen in Figure 5.4.

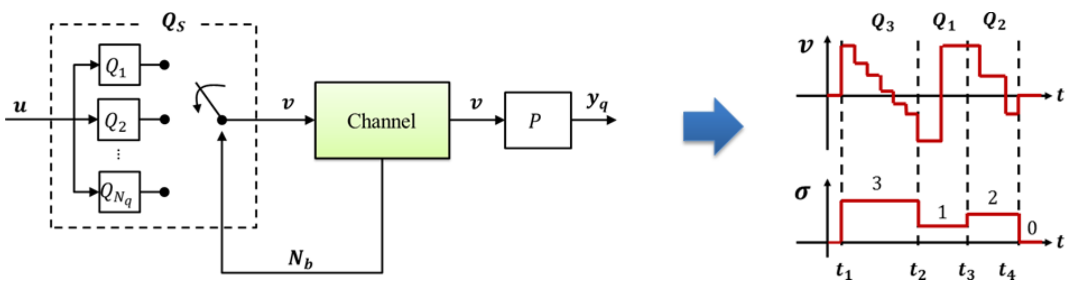


Figure 5.4: Data rate driven event-triggered quantizer structure and operation.

This quantizer Q_S is composed of a set of finite-level dynamic quantizers Q_i (for $i = 1, 2, \ldots, N_q$) followed by a trigger mechanism that is managed by the trigger signal. Each quantizer Q_i is designed to work optimally for a specific value of N_b. In this case σ has several values, each one of them corresponds to a specific quantizer. The trigger signal is given by the channel $\sigma(N_b)$, more precisely by a sensor that measures the traffic in the network. When the network's traffic is small it will be possible to express the signals with more precision, while when the traffic is big the signals will still be able to reach the plant on time but with less precision.

Here we present a design method for this type of quantizer. Given the SISO LTI plant P, the sampling time Δt, the range of the input signal $U = [u_{max}, u_{min}]$ and the values $N_b = \{N_{b1}, N_{b2}, \ldots, N_{bq}\}$. Each finite-level dynamic quantizer Q_i that belongs to Q_S is designed for a specific value of N_b, attending that Equation (2.21) ($M \leq 2^{N_b}$) is satisfied. For example, for $N_b = 1$ we can make $M = 2$, and for $N_b = 2$ we make $M = 4$. Then, for each value of M the quantizer Q_i is designed by using the methods previously developed based on metaheuristics (CMA-ES or DE preferably).

In the data rate driven event-triggered quantizer, the states of each internal quantizer Q_i are updated in each sampling time since the control signal $u(t)$ is applied on them all the time, the output of the quantizer $v(t)$ will be the $v_i(t)$ of the quantizer that is currently connected to the output.

To prove that this type of event-triggered quantizer and its design method work properly, an example is performed. We considered the second order plant given by Equation (4.2), a sampling period of $\Delta t = 0.1$, a control signal with range $U = [-1, 1]$ and a set $N_b = \{1, 2, 3, 4\}$ which in turn produces the set $M = \{2, 4, 8, 16\}$. Then, by using the quantizer design method based on CMA-ES we designed four Q_i. The triggered signal σ takes values on $\{1, 2, 3, 4\}$. In this case, since σ is subjected to variations in the traffic in the network, we assume a random behaviour. Thus, the signal σ can be emitted at any time and with any value.

Once that all the quantizers were found, in order to verify if the event-triggered quantizer works properly the following control signal is applied

$$u(t) = 0.3\sin(0.6t) + 0.4\sin(0.1t) + 0.3\sin(0.3t). \tag{5.7}$$

In this example we assume that initially $N_b = 4$, thus, the switch is connected to the quantizer with $M = 16$. Then, three changes are performed first at $t = 3[s]$, N_b changes to $N_b = 1$ and $M = 2$, at $t = 15[s]$ it changes to $N_b = 3$ and $M = 8$, finally at $t = 24[s]$ it changes to $N_b = 2$ and $M = 4$.

Figure 5.5 shows the resulting signals, in this graphics the lines in black represent the signals without the effects of quantization and the blue ones are the signals with quantization. The upper one is the control signal $u(t)$ and its quantizer version $v(t)$, the graphic in the middle shows the outputs with and without quantization, and the lower graphic shows the error signal, which is the difference between the two outputs considered. Notice that the error signal is relatively small and how the real output $y_q(t)$ follows the desired output $y(t)$ very closely.

We can conclude that the data rate driven event-triggered quantizer and its design

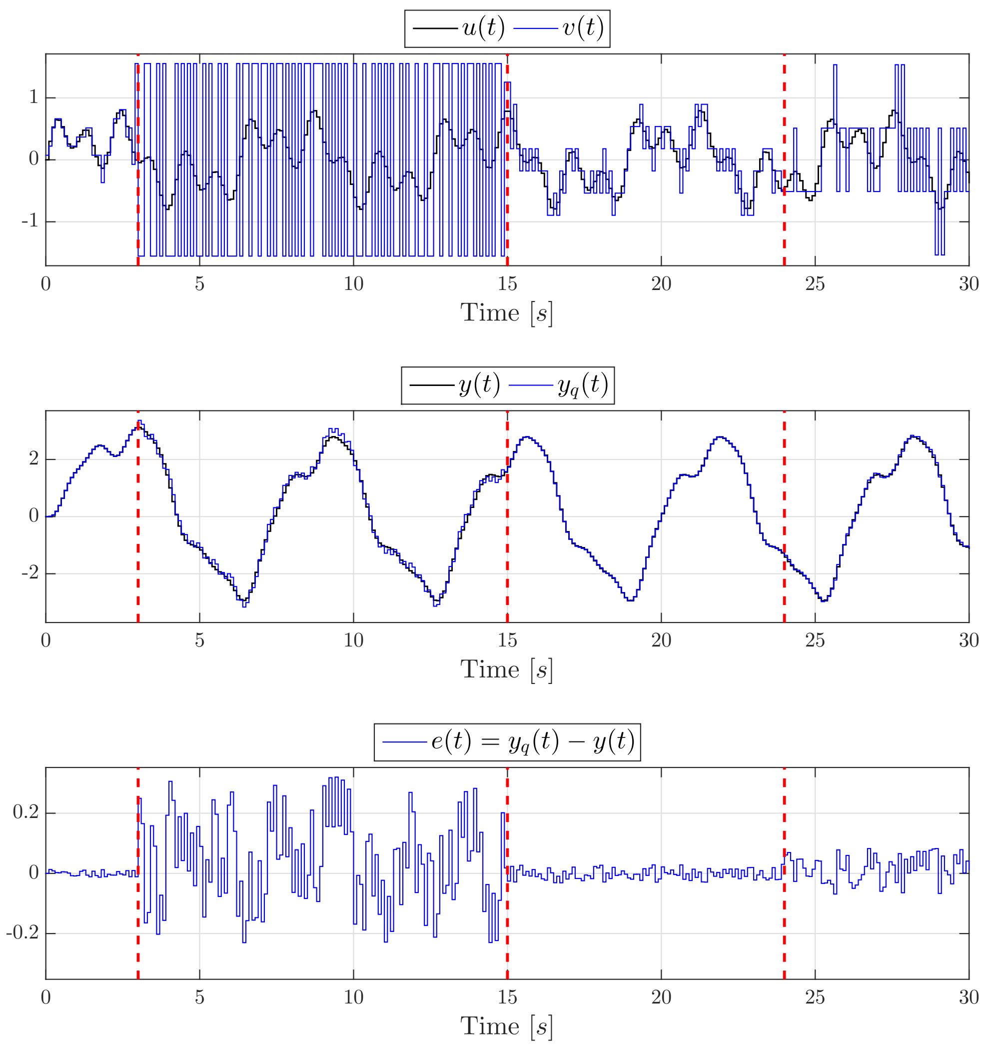


Figure 5.5: Example of the data rate driven event-triggered quantizer operation.

method work properly. By using this quantizer we can reduce the traffic in the network considerably while maintaining a good performance of the system.

Chapter 6

Conclusion

In this thesis the finite-level dynamic quantizer design method based on metaheuristics is proposed. The metaheuristics considered in this study are CMA-ES, DE and FA. By means of numerical simulations the effectiveness of the proposed design methods were confirmed and their efficiency were measured.

The CMA-ES and DE based design methods show in general very good performance in terms of success rate and convergence time. They both have advantages and disadvantages comparing to each other. The method based on CMA-ES shows a slightly better performance than DE, specially when considering a relatively small amount of search points and number of generations, besides the execution time is smaller. Moreover, CMA-ES carries the big advantage that it does not require the tuning of any control parameter. Their values are already optimized and established by default. Nevertheless, it is worth to mention that DE is very easy to implement contrary to CMA-ES.

Compared to the other metaheuristic based design methods, it was verified that the performance of the CMA-ES and DE based methods are quite better than the ones based on PSO and FA. For these reasons, it is possible to say that the methods based on CMA-ES and DE are very reliable for the design of the finite-level dynamic quantizer, but the ones based on FA and PSO are not.

Regarding to the hypothesis that $\mathcal{A} = A$ and $\mathcal{B} = B$ gives the optimal form of the quantizer, it was found that for the design problem in which the stability and data rate constraints are took into account, the hypothesis is true for the first order systems, but for systems of second and third order the result is approximated. Additionally, the even-triggered quantization concept was introduced. As well as a type of data rate driven event-triggered quantizer and its design method. The effectiveness of this type of quantizer was verified with a numerical example.

The work presented in this thesis can be greatly extended. On one hand, it is still needed to find an analytical expression for the optimal finite-level dynamic quantizer. On the other hand the event-triggered quantization offers a fruitful field for research. Many types of event-triggered quantizer can be constructed and the event-control theory should be applied on their analysis and design.

Acknowledgements

I would like to express my sincere gratitude to Professor Kenji Sugimoto and Associate Professor Takamitsu Matsubara for accepting me into their laboratory as a student, for their kindness, thoughtful advice and assistance throughout my study. I would also like to thank Professor Shoji Kasahara for his contribution on the thesis committee and to Associate Professor H. Okajima, Kumamoto University, for his valuable comments. I also would like to thank my parents Gerardo and Norma, my aunt Sofia and my brothers who have been supporting me from the distance and for all their prayers. To my girlfriend Desislava for taking care of me and supporting me in the difficult times. I would also like to thank the Government of Japan through its scholarship program Monbukagakusho for granting me the financial support that enabled me to come to Japan for my education and research. Finally, I wish to specially thank Assistant Professor Yuki Minami for guiding me through the master course, for his wise advice, kindness and patient, for encouraging me to surpass my limits and for being a true friend. Thank you very much Minami-sensei I will never forget all that you did for me.

References

[1] P. Antsaklis and J. Baillieul, "Special issue on technology of networked control systems," *Proceedings of the IEEE*, vol. 95, pp. 5–8, Jan 2007.

[2] R. A. Gupta and M. Chow, "Networked control system: Overview and research trends," *Industrial Electronics, IEEE Transactions on*, vol. 57, pp. 2527–2535, July 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] W. S. Wong and R. W. Brockett, "Systems with finite communication bandwidth constraints. ii. stabilization with limited information feedback," *Automatic Control, IEEE Transactions on*, vol. 44, pp. 1049–1053, May 1999.

[5] S. Tatikonda and S. Mitter, "Control under communication constraints," *Automatic Control, IEEE Transactions on*, vol. 49, pp. 1056–1068, July 2004.

[6] H. Okajima, N. Matsunaga, and K. Sawada, "Optimal quantization interval design of dynamic quantizers which satisfy the communication rate constraints," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 4733–4739, Dec 2010.

[7] K. Sawada, H. Okajima, N. Matsunaga, and Y. Minami, "Dynamic quantizer design for mimo systems based on communication rate constraint," in *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pp. 2572–2577, Nov 2011.

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

[9] S. Azuma and T. Sugie, "Synthesis of optimal dynamic quantizers for discrete-valued input control," *Automatic Control, IEEE Transactions on*, vol. 53, pp. 2064–2075, Oct 2008.

[10] K. Sawada and S. Shin, "Synthesis of dynamic quantizers for quantized feedback systems within invariant set analysis framework," in *American Control Conference (ACC), 2011*, pp. 1662–1667, June 2011.

[11] 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.

[12] K.-Y. You and L.-H. Xie, "Survey of recent progress in networked control systems," *Acta Automatica Sinica*, vol. 39, no. 2, pp. 101 – 117, 2013.

[13] Y. Minami and T. Murokami, "Differential evolution algorithm based design of discrete-valued input systems," in *SICE Annual Conference (SICE), 2014 Proceedings of the*, pp. 333–336, Sept 2014.

[14] R. Yoshino, H. Okajima, N. Matsunaga, and Y. Minami, "Dynamic quantizers design under data rate constraints by using pso method," in *SICE Annual Conference (SICE), 2014 Proceedings of the*, pp. 1041–1046, Sept 2014.

[15] J. Rodriguez Ramirez, Y. Minami, and K. Sugimoto, "Design of finite-level dynamic quantizers by using differential evolution algorithm," in *SICE Annual Conference (SICE), 2015 Proceedings of the*, pp. 841–844, July 2015.

[16] H. Okajima, K. Sawada, J. Kaibe, and N. Matsunaga, "Performance improvement of dynamic quantizer with reachability of output," *Transactions of the Society of Instrument and Control Engineers*, vol. 48, no. 6, pp. 359–361, 2012.

[17] H. Okajima, K. Sawada, and N. Matsunaga, "Dynamic quantizer design under communication rate constraints," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2015.

[18] I. Maruta, T. Kim, and T. Sugie, "Fixed-structure controller synthesis: A metaheuristic approach using simple constrained particle swarm optimization," *Automatica*, vol. 45, no. 2, pp. 553 – 559, 2009.

[19] I. Maruta, T. Kim, D. Song, and T. Sugie, "Synthesis of fixed-structure robust controllers using a constrained particle swarm optimizer with cyclic neighbor-

hood topology," *Expert Systems with Applications*, vol. 40, no. 9, pp. 3595 – 3605, 2013.

[20] L. Bianchi, M. Dorigo, L. Gambardella, and W. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Natural Computing*, vol. 8, no. 2, pp. 239–287, 2009.

[21] X. Yang, "Metaheuristic Optimization," *Scholarpedia*, vol. 6, no. 8, p. 11472, 2011. revision #91488.

[22] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82 – 117, 2013. Prediction, Control and Diagnosis using Advanced Neural Computations.

[23] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[24] N. Hansen, 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.

[25] N. Hansen, "The CMA evolution strategy: a comparing review," in *Towards a new evolutionary computation. Advances on estimation of distribution algorithms* (J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, eds.), pp. 75–102, 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] X. Yang, *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.

[28] X. Yang and X. He, "Firefly algorithm: Recent advances and applications," *Int. J. Swarm Intelligence*, vol. 1, no. 1, pp. 36–50, 2013.

[29] S. Azuma and T. Sugie, "An analytical solution to dynamic quantization problem of nonlinear control systems," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 3914–3919, Dec 2009.

[30] G. Goodwin, K. Lau, and M. Cea, "Control with communication constraints," in *Control Automation Robotics Vision (ICARCV), 2012 12th International Conference on*, pp. 1–10, Dec 2012.

[31] N. Hansen and S. Kern, "Evaluating the CMA evolution strategy on multimodal test functions," in *Parallel Problem Solving from Nature PPSN VIII* (X. Yao *et al.*, eds.), vol. 3242 of *LNCS*, pp. 282–291, Springer, 2004.

[32] N. Hansen, "Benchmarking a bi-population cma-es on the bbob-2009 function testbed," in *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, GECCO '09, (New York, NY, USA), pp. 2389–2396, ACM, 2009.

[33] N. Hansen, D. Arnold, and A. Auger, "Evolution strategies," in *Springer Handbook of Computational Intelligence* (J. Kacprzyk and W. Pedrycz, eds.), pp. 871–898, Springer Berlin Heidelberg, 2015.

[34] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948 vol.4, Nov 1995.

Appendix

A. Default Parameters for the $(\mu/\mu_W, \lambda)$ CMA-ES Algorithm

The default values of these parameters were taken from [32].

i Selection and Recombination:

$$N = 4 + \lfloor 3\ln(n) \rfloor, \quad \mu = \lfloor \mu' \rfloor, \quad \mu' = \frac{N}{2} \tag{6.1}$$

$$w_i = \frac{w'_i}{\sum_{j=1}^{\mu} w'_j}, \quad w'_i = \ln(\mu' + 0.5) - \ln i \quad \text{for } i = 1, \ldots, \mu \tag{6.2}$$

$$\mu_{\text{eff}} = \frac{1}{\sum_{i=1}^{\mu} w_i^2} \tag{6.3}$$

ii Step-size control:

$$c_\sigma = \frac{\mu_{\text{eff}} + 2}{n + \mu_{\text{eff}} + 5} \tag{6.4}$$

$$d_\sigma = 1 + 2\max\left(0, \sqrt{\frac{\mu_{\text{eff}} - 1}{n + 1}} - 1\right) + c_\sigma \tag{6.5}$$

iii Covariance matrix adaptation:

$$c_c = \frac{4 + \mu_{\text{eff}}/n}{n + 4 + 2\mu_{\text{eff}}/n} \tag{6.6}$$

$$c_1 = \frac{2}{(n + 1.3)^2 + \mu_{\text{eff}}} \tag{6.7}$$

$$c_\mu = \min\left(1 - c_1, \alpha_\mu \frac{\mu_{\text{eff}} - 2 + 1/\mu_{\text{eff}}}{(n+2)^2 + \alpha_\mu \mu_{\text{eff}}/2}\right) \quad \text{with} \quad \alpha_\mu = 2 \tag{6.8}$$

B. Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population based metaheuristic that is inspired in the behavior of biological communities like swarms of bees and flocks of birds [34]. In the PSO nomenclature a search point is called *particle* and the set of all the particles is called *swarm*.

PSO is being implemented widely in many applications and in the control systems field. The reasons for such a big development are that it is implemented very easily and has only three control parameters: inertia parameter χ_0, cognition parameter χ_1 and social parameter χ_2. This algorithm is very good for local optimization, but not so much for global optimization. Several variations try to improve its performance and ability to *scape* from local minima. In addition, PSO is very sensitive to the tuning of the control parameters and sometimes it is difficult to find a good set of parameters to make the algorithm effective. The version used in this study is shown in Algorithm 4.

Algorithm 4 : 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 search spaces $S_0 = [x_{min}, x_{max}]^n$ and $V_0 = [v_{min}, v_{max}]^n$. Set $k = 0$, then select randomly N search points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and their velocities $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$ in the corresponding search spaces.

Step 1: The objective function $f(\mathbf{x}_i)$ is evaluated for each $\mathbf{x}_i$. Then, the personal best solutions and the global best solution are selected by:

$$\mathbf{x}_{pbest,i} = \underset{\mathbf{x} \in \{\mathbf{x}_i^{(j)} | j=1,2,\ldots,k\}}{\arg\min} f(\mathbf{x}), \tag{6.9}$$

$$\mathbf{x}_{gbest} = \underset{\mathbf{x} \in \{\mathbf{x}_{pbest,i} | i=1,2,\ldots,N\}}{\arg\min} f(\mathbf{x}). \tag{6.10}$$

where (j) indicates the generation of the respective search point. If $k = k_{max}$ then $\mathbf{x}_{gbest}$ 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.

$$\mathbf{v}_i \leftarrow \chi_0 \mathbf{v}_i + \chi_1 \rho_{1,i} \left(\mathbf{x}_{pbest,i} - \mathbf{x}_i\right) + \chi_2 \rho_{2,i} \left(\mathbf{x}_{gbest} - \mathbf{x}_i\right), \tag{6.11}$$

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i, \tag{6.12}$$

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**.
