الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Réf:…………

جامعة محمد خيضر بسكرة
كلية العلوم و التكنولوجيا
قسم:الهندسة الكهربائية
المرجع: ………..

Thèse présentée en vue de l'obtention
du diplôme de

Doctorat en sciences

Option : Electronique

Soft Computing in Control and modelling

Présentée par : **MEGHERBI Hassina**

Soutenue le : **10 Juin 2014**

Devant le jury composé de :

Djedi Noureddine	Professeur	Université de Biskra	Président
Benmahammed Khier	Professeur	Université de Setif	Rapporteur
Harmas Mohamed Nadjib	Professeur	Université de Setif	Examinateur
Taleb-Ahmed Abdelmalik	Professeur	Université de Valenciennes(France)	Examinateur
Taibi Mohamoud	MC ''A''	Université de Annaba	Examinateur
Ouafi abdel Krim	MC ''A''	Université de Biskra	Examinateur

Année universitaire : 2013-2014

ملخص الأطروحة

نعرض في هذه الأطروحة تطبيق خوارزمية التطور ذات الترميز الطبيعي (IEA) لتحسين بنك المعلومات الغامض الخاص بنظام تحكم يعمل بالمنطق الغامض(FLC)، و التخلص بذلك من الحاجة إلى خبير في مرحلة التصميم. الخوارزمية المقترحة عدلت و كُيِفت لتركز البحث في جوار المنطقة التي يوجد بها الحل الأمثل من خلال مرحلة الاستغلال أو التركيز. وعند أخذنا بعين الاعتبار التغير في جهد المدخل للمحركات ذات التيار المستمر كمركبة لدالة الهدف تحصلنا على سلوك أملس كفاية عند مخرج نظام FLC. لضمان مظهر التمام في التجزئة المبهمة بدون خسارة مظهر التمييز، نقترح استراتيجية تشفير خاصة أين تكون التداخلات بين دالات الأنتماء المتجاورة (MFs) مشفرة في الكروموسوم و تقوم الخواريزمية التطورية بتطويرها. نقوم أيضا بالبحث عن معاملات (MFs) في مجالات تتعلق بمعاملات ال(MFs) المجاورة، مما يجعل عملية التشفير متسلسلة. الدافع وراء طريقة البحث المقترحة هو تصميم نظام تحكم مبهم مفهوم و أملس يقوم بتحكم متابعة دقيق و سريع لمحرك تيار مستمر. نتائج المحاكات المتحصل عليها تظهر أن خصائص التمام و التميز مضمونة و أن ال(FLC) المطور أظهر أداءً عاليا في تحكم المتابعة البطيء و السريع مقارنة مع نضام التحكم (PD) الكلاسيكي.

نعرض أيضا في هذه الأطروحة تقنية بحث تطوري ذات طابقين للتصميم الأوتوماتيكي لنظام تحكم مبهم قطاعي (SFC). في الطابق الأول، تقوم ال(EA) المقترحة بتحسين ال(SFC) من أجل نموذج اسمي للنظام المراد التحكم فيه (أي بدون تشويشات أو تغيير معاملات). الهدف الرئسي من الطابق الثاني هو تعزيز متانة ال(SFC) المطور في الطابق الأول. بالتحديد يقوم ال(EA) بالبحث في جوار أحسن (SFC) المتحصل عليه في الطابق الأول و الذي يعطي أحسن وسطية بين أداء التحكم للنموذج الإسمي و للنموذج المشوش. تم تكييف المميزات القطاعية في البحث التطوري عبر عملية خاصة لتحديد معاملات بنك القواعد المبهمة (FRB) و (MFs) ل (SFC)، عامل تصليح و تهيئة أولية خاصة للجزء الخاص بال(FRB) من الكروموسوم. و لقد وجد أن ال(SFC) المطور بطريقة التصميم المقترحة أعطى أداءً جد مرضٍ تحت عدة أنواع من التشويش. و قد تم أيضا دراسة الوسطية بين أداء الدقة و أداء المتانة أثناء العملية التطورية .

الكلمات المفتاحية

نظام التحكم المبهم نوع Mamdani ، نظام التحكم المبهم القطاعي ،التصميم الأوتوماتيكي لبنك المعلومات المبهم، الخوارزميات التطورية، تفسير التجزئة المبهمة، التصميم المتين، محرك ذو التيار المستمر، الاستكشاف/الاستغلال.

Résumé

Dans cette thèse, on décrit l'application d'un algorithme évolutionnaire à codage entier (IEA) pour l'optimisation de la base de connaissances d'un contrôleur flou (FLC); éliminant de la sorte le besoin d'un expert-humain dans la phase de conception. L'IEA proposé est étendu pour concentrer la recherche dans la région de voisinage de l'optimum de l'espace de recherche en adoptant une phase dite d'exploitation. En considérant la variation de la tension d'entrée des actionneurs DC comme une composante de la fonction objectif, on a obtenu un comportement suffisamment lisse à la sortie du contrôleur conçu. Pour garantir l'aspect de complétude de la partition floue sans perdre celui de distinction, on propose une stratégie de codage spéciale où les chevauchements entre les fonctions d'appartenances (MFs) adjacentes sont codés dans le chromosome et évolués par l'IEA. On a aussi recherché les paramètres des MFs dans des intervalles dépendant sur ceux des MFs adjacentes précédentes, ce qui rend le processus de codage hiérarchique. La motivation de la méthode de recherche proposée est la conception d'un contrôleur interprétable et lisse pour accomplir un contrôle de poursuite précis et rapide pour les actionneurs à entrainement direct. Les résultats de simulation montrent que l'interprétabilité de la partition floue est garantie et que le FLC évolué a manifesté de hautes performances dans le contrôle de poursuite lent et rapide par rapport au contrôleur PD conventionnel.

Cette thèse présente aussi une méthodologie de recherche évolutionnaire à deux étages pour concevoir automatiquement un contrôleur flou sectoriel (SFC). Dans le premier étage, l'EA proposé optimise, le SFC pour un model nominal (i.e., sans bruit additive ou variation de paramètres). L'objective principale du 2ème étage est le renforcement de la robustesse de SFC résultant du 1ère étage. Plus précisément, l'EA proposé cherche dans le voisinage du meilleur SFC trouvé dans le 1ère étage en vue de trouver un SFC qui fournit un compromis entre les performances de contrôle pour un modèle nominal et un model perturbé. Les propriétés sectorielles sont accommodées dans la recherche évolutionnaire à travers une paramétrisation spéciale de la base de règles floues (FRB) et les MFs, un opérateur de réparation et une initialisation spéciale de la partie réservée pour la base des règles. Le SFC obtenu avec la méthodologie de conception proposée a fourni des performances très satisfaisantes sous différents types de perturbations. Le compromis entre les performances de précision et ceux de robustesse sont aussi analysé lors du processus d'évolution.

Mots-clés : Contrôleur flou de type Mamdani, contrôleur flou sectoriel, algorithme évolutionnaire, conception automatique de la base de connaissances floues, Interprétabilité de la partition floue, conception robuste, moteur DC à entrainement directe, exploitation/exploration.

Abstract

In this thesis, we describe the application of an integer-coded evolutionary algorithm (IEA) for fuzzy knowledge base optimization of a fuzzy logic controller (FLC), eliminating in such a way the need of an expert-human in the design phase. The proposed IEA is extended to concentrate the search into optimum vicinity region of the overall search space by adopting exploitation or focusing phase. By considering the variation of the input voltage of the DC actuators as components of the fitness function, we get a satisfactory smooth behavior at the evolved FLC output. To guarantee the completeness aspect of fuzzy partitions without losing the distinguishability one, we propose a special encoding strategy where the overlappings between the adjacent membership functions (MF) are coded in the chromosome and evolved by the IEA. We also evolve the MF parameters in ranges depending on the parameters of the previous adjacent MF parameters which make the decoding process hierarchical. The motivation behind the proposed search method is to design a smooth interpretable fuzzy controller to achieve rapid and accurate tracking control for direct drive. Simulation results show that fuzzy partition interpretability is guaranteed and the evolved FLC exhibits high performances in slow and fast tracking tasks as compared with the conventional PD controller.

We also present in this thesis a two stages evolution search methodology to automatically design a sectorial fuzzy controller (SFC). At first stage, the proposed EA optimises the SFC for disturbance-free model of the plant to be controlled. The principal aim of the second stage is the robustness enhancement of the evolved SFC resulting from the former stage. Specifically, the proposed EA looks in the vicinity of the best SFC found in the first stage for a SFC that provide the best compromise between the control performance for a disturbance-free model and for disturbed model. The sectorial properties were accommodated in the evolutionary search through a special parameterization of the fuzzy rule base (FRB) and the membership functions (MFs) of the SFC, repairing operator and special initialization of FRB chromosome part. The evolved SFC with the proposed design methodology found to provide very satisfactory performance under different types of disturbances. The trade-off between the accuracy performance and the robustness performance is also analysed during the evolution process

Keywords : Mamdani fuzzy logic controller, sectorial fuzzy controller, fuzzy knowledge base automatic design , evolutionary algorithm, fuzzy partition interpretability, robust design, direct drive DC motors, exploitation/exploration.

Contents

List of Tables

List of Figures

List of abbreviations

SC	Soft Computing
FLC	Fuzzy Logic Controller
MF	Membership Function
EA	Evolutionary Algorithm
GA	Genetic Algorithm
FKB	Fuzzy Knowledge Base
FRB	Fuzzy Rule Base
FDB	Fuzzy Data Base
RMSE	Root Mean Squared Error
Sum|ΔEa|	Sum of the absolute variation of the input voltage
IEA	Integer-coded Evolutionary Algorithm
IEA-1	mono-phase IEA having as a fitness function the RMSE
SFC	Sectorial Fuzzy Controller
ObjN	Objective function for the Nominal case
ObjD	Objective function for the Disturbed case
Obj	final Objective function

To my parents, brothers and sisters

To my dear husband, and lovely kids

Acknowledgements

I convey my gratefulness to my teacher and supervisor Prof. BENMAHAMMED Khier for his support, guidance and encouragement throughout the research period.

I wish to express my deep sense of gratitude to the faculty and staff members of Electronic Department of Ferhat Abess University – Setif for their kind hospitality and help in my post-graduate studies and research activity.

I extend my humble thanks to the faculty and staff members of Electrical Department of Mohamed kheider University – Biskra for accepting me as a part of this institution (as a teacher and a student).

I would like to thank my brother Megherbi Ahmed chauki and my sister Megherbi Najla for their co-operation and help in my research activity.

I want to express my sincere thanks to my husband for helping me directly and indirectly to complete the thesis.

I extend my humble thanks to my parents, brothers and sisters for the constant moral support provided by them.

I would like to thank all my colleagues and friends whose company and encouragement helped me a lot to work hard.

Last, but not the least, I thank ALLAH, for giving me the strength during the course of this research work.

Introduction

Modern technologies are confronted today with increasing sever performance demands which necessarily requires a high-performance controller. The use of the existing conventional control approaches cannot meet such demands, because they are unable to cope with the uncertainty, imprecision, discontinuity, irregularity, time variance, and nonlinearities inherent to the plant to be controlled. In the contrary, plant operator is able to cope with nonlinearities and time variance. He is also able to act in presence of complex sets of noisy observations and poorly specified constraints and satisfy multiple subjective-based performance criteria. The emerging field interested in incorporating these attributes and others related to biological/natural systems into control systems or more general into computer science is the so-called Soft-Computing field.

1. LITTLE BIT ON SOFT-COMPUTING

The term of soft-computing (SC) has been first introduced by Lotfi Zadeh as : “ *In traditional - hard -computing, and rigor. By contrast, the point of departure in soft-computing is the thesis that precision and certainty carry a cost and that computation, reasoning, and decision making should exploit - whenever possible – the tolerance for imprecision and uncertainty*” [1].

At first SC works appear within different disciplines: artificial intelligence, computer science, applied Mathematics …etc. In the last decade, it become more and more separate discipline, self-sustaining field with its own professional society, conferences, journals and meetings and then it is referred as "*Intelligent control"* since it was applied first in control system. After the increasing of the radius of its applications, it takes the name of "*computational intelligence*" in parallel with "*soft computing*". But the science community has some trouble to define this discipline from intelligence point of view, because there is no standard concept of intelligence, that's why the researchers tend to use the qualification of "*soft computing*".

CS refers to a set of emerging computational paradigms, arises as generalization and complementation of hard (conventional) computing. It aims to capture and emulate the Mother Nature and human being tasks including adapting, searching, learning, granulation of information and reasoning that tolerate imprecision, uncertainty, partial truth and approximation.

The best studied of SC paradigms to date have been fuzzy logic, neural networks and evolutionary computing. Each of these paradigms provides effective conceptual frameworks for dealing with real-world problems and offers different advantages, specifically:

- Fuzzy logic enables the direct incorporation of linguistic and qualitative knowledge of an expert about the problem to be solved into reasoning systems.

- Neural networks have shown real promise in learning from examples of input-output pairs and adapting in response to changes in process parameters or environment.

- Evolutionary computing involves learning capability, global and local search features. It covers several population-based search paradigms, such as evolutionary strategy, evolutionary programming, genetic algorithm and genetic programming, which offer valid approaches to optimization problems requiring efficient and effective search.

It can be easily observed that SC paradigms have distinct and complementary natures in the way of tackling real world problems. To reap benefit from this fact, a combination of them into a hybrid system have been done. As results, the drawbacks and limits that characterize each paradigm are overcome and the SC-based system performances are enhanced further. The hybrid soft computing system can be: neural-fuzzy, neural-genetic, fuzzy-genetic, or neural-fuzzy-genetic system.

2. AIM OF THE THESIS

The development of hybrid soft computing methods has attracted considerable research interest over the past decade. They are applied to important fields such as control, signal processing, and system modelling. Although hybrid soft computing methods have shown great potential in these areas, they share some common shortcomings that hinder them from being used more widely.

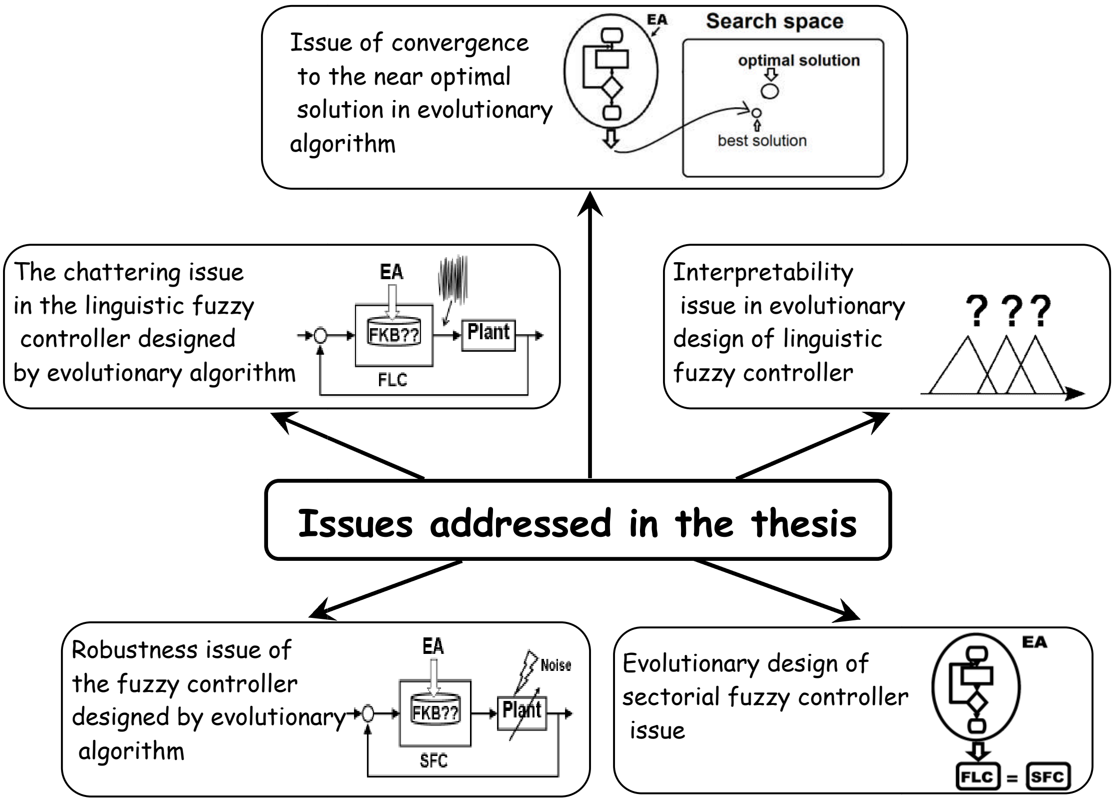


Fig.0.1 Schematic representation of the issues addressed in the thesis

The general aim of this thesis is to explore and investigate one particular hybrid soft computing method, namely the fuzzy genetic methods so that new and enhanced methods can be put forward. Specifically, we establish a novel conceptual framework for the automatic evolutionary design of linguistic fuzzy logic controller (FLC) and propose some methodologies that address several issues and limitations of the methodologies available so far. These issues and limitations are presented in Fig.0.1. They are briefly described as follows:

1. **Issue of convergence to the near optimal solution in evolutionary algorithm:**
   The most remarkable strength of evolutionary algorithm (EA) is that they could locate quickly a well approximating solution in highly complex search space where the fitness function is discontinuous, multimodal, noisy or changes over time. However they may take a relatively long time to reach the optimal solution or at least a very good one.
2. **Interpretability issue in evolutionary design of linguistic fuzzy controller:**
   The fuzzy control was implemented for the first time by Mamdani in [2] where he used a new approach proposed by L. Zadeh in 1965 that provides an effective

means of controlling systems which are too complex or too ill-defined to admit the use of conventional control. The fuzzy control systems present two distinguishable and valuable features: (1) the use of linguistic variables instead or in addition to numerical variables, and (2) the description of the relationship between input and output variables by conditional fuzzy statements (fuzzy rules) which formalize the behaviour of the fuzzy control system in human understandable way. The researcher in the evolutionary linguistic fuzzy control system design, have usually focused on the improvement of the control performance without paying special attention to its interpretability. As results, the EA designs the input/output fuzzy partitions and the fuzzy rules for the linguistic fuzzy control system without any associated meanings. Furthermore, the fuzzy partitions are usually incomplete and indistinguishable.

3. **The chattering issue in the linguistic fuzzy controller designed by EA:**
   It has been acknowledged that fuzzy controller work like a sliding mode controller [3], [4]. It uses one particular control structure for one particular state. From a state to another, the control structure is changed according to some fuzzy rules. A well designed fuzzy controller must provide smooth transition between adjacent structures. However, in evolutionary design of such controller this fact is not taken into account. Subsequently, the designed linguistic fuzzy controller exhibits excessive control activity, i.e., high-frequency switching of the control signal known as ”chattering” which is a serious drawback for technical systems.

4. **Evolutionary design of sectorial fuzzy controller issue:**
   Sectorial fuzzy controller is a linguistic FLC that fulfil a number of sectorial properties. It is evident that designing such system with evolutionary algorithm requires some arrangement and consideration in the structure of the EA which should be working toward preserving the sectorial properties during the evolutionary process.

5. **Robustness issue of the FLC designed by EA:**
   The EAs used for fuzzy controller design use the nominal model of the plant to be controlled which can be quantitative or qualitative (neural, fuzzy or neuro-fuzzy model). The resultant fuzzy controller can have disastrous consequences once put to work in real world application since it is subject to wide range of uncertainties and disturbances.

The direct-drive DC motor is an example of the class of uncertain and non linear dynamical systems, which the proposed SC-based methodologies are intended to control.

3. CONTRIBUTIONS AND OUTLINE OF THE THESIS

This dissertation is organized into four chapters, in addition to an introduction and a conclusion.

Chapter1 introduces the state of the art of the fuzzy set concept, properties and operations together with a number of concepts related to fuzzy logic and fuzzy systems.

Chapter2 describes the basic terminology and principles of the genetic and EAs and summarizes the limits and benefits of this class of methodology.

Chapter 3 highlights the application of EA for Mamdani FLC design. At first, we present a description of a direct-drive DC motor as a system to be controlled. Then, the details of the structure of the Mamdani FLC are provided followed by its parameterization in the chromosome. In the context of parameterization, we propose to exploit symmetry, if exist, of the considered system in reducing the chromosome size. This is the case of most electrical drives. For the sake of simplicity in the simulation and the discussion of the results, the presentation of the work is divided into two parts. In part 1, an interpretable chattering-free Mamdani FLC design is discussed. Only two issues are considered: the chattering and the interpretability issues. The basic idea of taking into account the chattering phenomenon during the optimization process is the introduction of the sum of variation of the control signal as optimization criterion. Doing so will ensure that the designed FLC provides just enough voltage to get the control job accomplished. This contribution is presented in [5]. The interpretability contribution consists in the encoding strategy where overlappings between the adjacent MFs are coded in the chromosome and evolved by the bi-phase IEA. Doing so, the completeness aspect is guaranteed, and there is no need for measuring it and using the multiobjective search. Another consideration in this issue is that all the searching ranges of the MF parameters are dependent on the adjacent MF parameters. This gives the bi-phase IEA the ability to evolve only valid distinguishable fuzzy partitions. Part2 investigates the application of the bi-phase IEA in FLC design. The purpose of the bi-phase scheme is the improvement of the solution issued from an exploratory evolutionary process by exploiting of the best exploratory solution. The idea of the exploitation proposed is based on creep mutating the integer encoding of the best solution and disposing of the crossover and integer mutation whilst adopting the elitism strategy. The interpretability contribution and the bi-phase scheme are introduced in [6].

Chapter 4 describes the evolutionary optimisation framework of the sectorial fuzzy controller as well as its robustness enhancement. The challenge of the sectorial fuzzy controller design by EA consists primarily in maintaining the sectorial properties during the evolution process. For this purpose, a number of considerations are taken in some components of the EA namely, the initial population, the system parameterization and representation on the chromosome. Moreover, a reparation operator is proposed to recover the monitonicity property that can't be preserved by the proposed strategies as described in [7]. The robustness enhancement issue is addresses by a two stage search strategy as described in [8], [7] and [9]. At the first stage, the chromosomes are evaluated on the sole criterion of accuracy. While at the second stage the evaluation of the chromosomes are done on both robustness and accuracy criteria.

Chapter I: Fuzzy Logic Systems

I.1 INTRODUCTION

In recent years, supercomputers play an important role in scientific computations and in simulation of large-scale systems. This is especially true for application in meteorology, in nuclear physics, in modelling of large economic systems, in the solution of partial differential equations and in simulation of complex phenomenon like turbulence, fluid flow, etc. In the real world, there are many applications that can't be implemented with the availability of these supercomputers, for example, the pattern recognition, the natural language processing, and the inference from the information resident in a large knowledge base –especially when this information is imprecise, uncertain, incomplete, or not totally reliable. This had led to an increasing number of uncertainty theories and to numerous attempts to modify the existing formal methods such that they correspond more to reality and human mental behaviour.

A pioneer in this direction was a polish mathematician by the name of J. Lukasiewiecz who first devised a three-valued logic in 1920. Later in the 1930's, he extended it to n-truth valued logic or multi-valued logic. However, even though this multi-valued logic has been available for some times, it has not been used to any significant extent in linguistic, in psychology, and in other fields where human cognition plays an important role, and this is where fuzzy logic enters the picture.

Fuzzy logic is a powerful problem solving technique emerged from fuzzy set theory developed by Lotfi A. Zadeh in 1965 [1] to bridge the wide gap between the precision of classical logic and the imprecision of the real world. Its major feature is the use of linguistic rather than numerical variables by *fuzzy* conditional statement.

In this chapter, we will present the most fundamental concept in fuzzy set theory useful in fuzzy systems and fuzzy control.

I.2 SHORT PRIMER ON FUZZY SETS

I.2.1 Fuzzy sets and membership functions

Roughly speaking a fuzzy set is a class of objects in which the transition from membership to non-membership is gradual rather than abrupt [10], [11], [12], [13], [14]. A more precise definition may be stated as follows:

If U is a collection of objects or values denoted generically by "u", then the fuzzy set F in U is defined by a set of ordered pairs:

$$F : \{(u, \mu_F(u)) \,/\, u \in U\} \qquad \text{(I.1)}$$

Where : $\mu_F(u)$ is called membership function that characterizes completely the fuzzy set F and provides a measure of the degree of membership of an element in U to the fuzzy set F.

U is referred to as the "universe of discourse" or "universal set", and it may contain either discrete or continuous values.

F is commonly defined as:

$$F : \int_U \mu_F/u \qquad \text{if } U \text{ is continuous.} \qquad \text{(I.2)}$$

$$F: \sum_{u_i \in U} \mu_F(u_i)/u_i \qquad \text{if } U \text{ is discrete.} \qquad \text{(I.3)}$$

In these expressions integral and summation sign do not denote integration or arithmetic addition, respectively, but denote the collection of all points $u \in U$ with associated membership function $\mu_F(u)$.

The fuzzy sets can be classified in several types. The most common are the ordinary fuzzy sets, also known as type 1 fuzzy sets, interval-valued fuzzy sets, and type 2 fuzzy sets [15]. For ordinary fuzzy sets, the membership function takes on precise values in the interval [0,1]. However, for the interval-valued fuzzy sets the membership function assigns to each element of the universe of discourse an interval of values. While for the type 2 fuzzy sets, the membership function value is a fuzzy number.

Both interval valued fuzzy system and type 2 fuzzy system have offered more adequate representation of expert knowledge with respect to type 1 fuzzy sets. However, their widespread use is severely limited because of the high degree of computational complexity. For this reason, we choose to use the ordinary fuzzy sets in the present work.

I.2.2 Properties of fuzzy sets

In this section, the most important properties for fuzzy sets will be presented. Let F be a fuzzy set defined in U and described by its membership function $\mu_F(u)$.

I.2.2.A Height of fuzzy set

The height of a fuzzy set F is equal to the largest membership degree. It is denoted by hgt(F), and defined as :

$$\text{hgt}(F) = \ sup_{u \in U} \mu_F(\text{u}) \qquad \text{(I.4)}$$

A fuzzy set F is called "*normal*", if hgt(F) = 1, and "*subnormal*" if hgt(F) < 1; Fig. I.1.

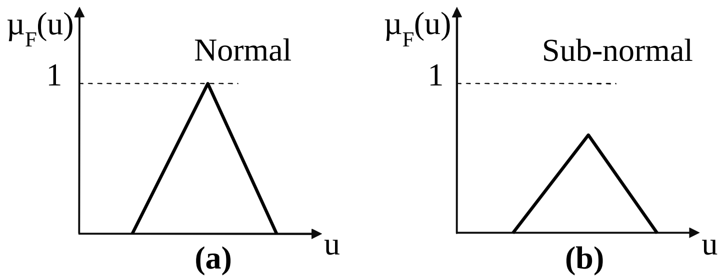


Fig. I.1 An example of a normal and sub-normal fuzzy set.

I.2.2.B Convexity of fuzzy set

A fuzzy set is called convex if its membership function is strictly monotonically increasing, monotonically decreasing or monotonically increasing then decreasing, Fig. I.2. Formally, a fuzzy set F is convex if and only if :

$$\forall u_1, u_2 \in U, \forall \lambda \in [0,1] : \ \ \mu_F(\lambda u_1 + (1-\lambda) u_2) \geq \min\,(\mu_F(u_1), \mu_F(u_2)) \qquad \text{(I.5)}$$

Note:

In fuzzy logic applications, it is usual to deal only with convex and normal fuzzy sets.

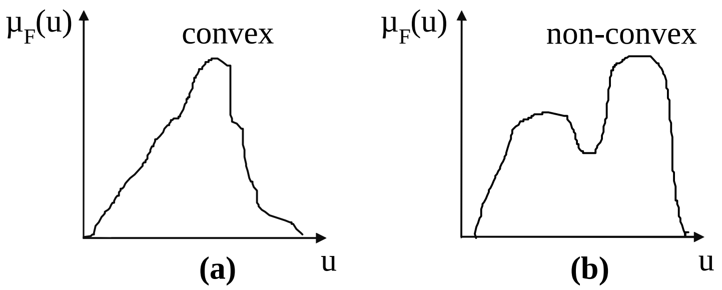


Fig. I.2 An example of a convex and non-convex fuzzy set.

I.2.2.C Support of fuzzy set

We call support of a fuzzy set *F* in *U* a crisp set of all points *u* in *U* such that $\mu_F(u) > 0$, Fig. I.3. It is denoted by *S(F)* and formally defined as :

$$S(F) = \{ u \in U \mid \mu_F(u)>0\} \tag{I.6}$$

The fuzzy set whose support is a singleton point in *U* is called "*fuzzy singleton*".

I.2.2.D Crossover point

The point *u* in *U* at which $\mu_F(u)$=0.5, is called the "crossover point", Fig. I.3.

I.2.2.E Nucleus of fuzzy set

The nucleus or the core of a fuzzy set *F* is the crisp set that contains all the values of the universe of discourse having the membership degree equal to unity, Fig. I.3. Formally, the nucleus of the fuzzy set *F*, is defined by :

$$nucleus(F) = \{ u \in U \mid \mu_F(u)=1\} \tag{I.7}$$

If there is only one point with membership degree equal to 1, then this point is called the "*peak value*" of *F*.

I.2.2.F Boundary of fuzzy set

We call boundary of a fuzzy set *F* in *U* a crisp set of all points *u* in *U* such that: $0 < \mu_F(u) < 1$, Fig. I.3. The boundary is denoted by *B(F)* and formally defined as :

$$B(F) = \{ u \in U \mid 0 < \mu_F(u) < 1\} \tag{I.8}$$

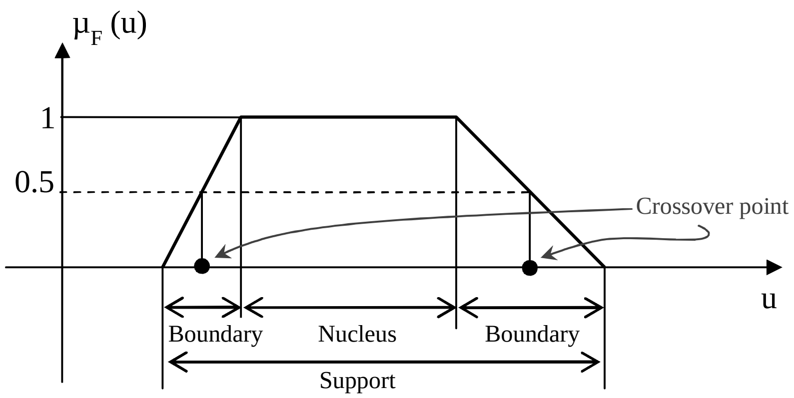


Fig. I.3 Boundary, support, nucleus, crossover point of a fuzzy set F.

I.2.3 Formulation of membership function

There is a variety of basic types or shapes of membership function that can be used. The triangular, trapezoidal and Gaussian, generalized bell membership functions are the most popular in the engineering applications. The graphical representation of these MFs is illustrated on Fig. I.4 and their mathematical formulation is given in what follows.

A ***triangular membership function*** is parameterized by three parameters *{a, b, c} (with a<b<c)* and given by:

$$\mu(u) = \begin{Bmatrix} 0, & u \le a \\ \frac{u-a}{b-a}, & a \le u \le b \\ \frac{c-u}{c-b}, & b \le u \le c \\ 0, & c \le u \end{Bmatrix} = max\left\{min\left\{\frac{u-a}{b-a}, \frac{d-u}{d-c}\right\}, 0\right\} \qquad (I.9)$$

The ***trapezoidal MF*** is parameterized by four parameters *{a,b,c,d}* (with $a < b \le c < d$) and defined as:

$$\mu(u) = \begin{Bmatrix} 0, & u \le a \\ \frac{u-a}{b-a}, & a \le u \le b \\ 1, & b \le u \le c \\ \frac{d-u}{d-c}, & c \le u \le d \\ 0, & d \le u \end{Bmatrix} = max\left\{min\left\{\frac{u-a}{b-a}, \frac{d-u}{d-c}, 1\right\}, 0\right\} \qquad (I.10)$$

The parameters *a, b, c* and *d* used in the two above shapes represent the x-coordinates of the corner points of the underling shape (triangular or trapezoidal).

The ***Gaussian membership function*** is parameterized by the two parameters {c,σ} and given by:

$$\mu(u) = e^{-\frac{1}{2}\left(\frac{u-c}{\sigma}\right)^2} \qquad (I.11)$$

Where the parameter c locates the centre of the peak and σ controls the width of the function.

A ***generalized bell membership function*** is parameterized by three parameters {a,b,c} and defined as:

$$\mu(u) = \frac{1}{1 + \left|\frac{u-c}{a}\right|^{2b}} \qquad (I.12)$$

The parameter a and b influence the width of the function and the parameter c represents the centre of the peak. The parameter b should be positive, otherwise, the shape of this MF becomes an upside-down bell.

Every type of membership function has shown some advantages and disadvantages. For instance, triangular and trapezoidal membership functions are characterized by the simplicity of implementation and computational efficiency in real-time based systems. However, it is very difficult to adjust those functions adaptively using statistical learning functions because of their discontinuity in their mathematical formulation. For Gaussian MF, the computational time is higher than the former types, but its exponential term allows naturally their adaptive adjustment in statistical model.

Until now, there are no general rules that can determine which membership function shape is most suitable for a given system or application. Usually, the choice is based more on personal preference than any mathematical justification.

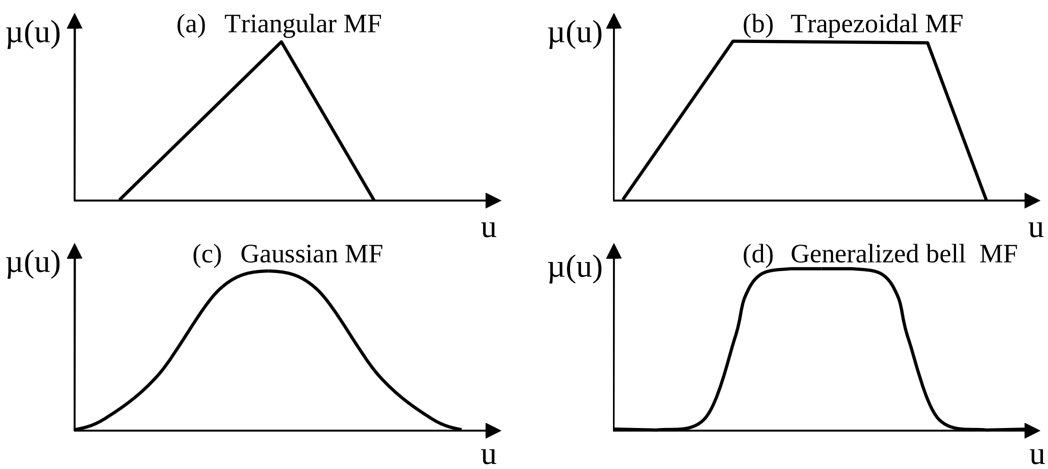


Fig. I.4 Example of MF shapes.

I.2.4 Standard operations on fuzzy sets

In this section we briefly summarize the basic operations defined on the fuzzy sets. These operations are defined in terms of their membership functions [16],[11],[12].

Let fuzzy sets A, B, and C in U described by their membership functions $\mu_A(u)$, $\mu_B(u)$, and $\mu_C(u)$, respectively.

The "*Union*" of A and B, denoted as $A \cup B$, is defined by :

$$\mu_{A\cup B}(u) = \max\left(\mu_A(u), \mu_B(u)\right) \quad \text{(I.13)}$$

The "*intersection*" of *A* and *B*, denoted as $A \cap B$, is defined by :

$$\mu_{A\cap B}(u) = \min\left(\mu_A(u), \mu_B(u)\right) \quad \text{(I.14)}$$

The "*complement*" of *A*, denoted as $\bar{A}$, is defined by :

$$\mu_{\bar{A}}(u) = 1-\mu_A(u) \quad \text{(I.15)}$$

The "*Cartesian product*" of *A*, *B*, and C, denoted as $A \times B \times C$, is defined by :

$$\mu_{A\times B\times C}(u_1, u_2, u_3) = \min\left(\mu_A(u_1), \mu_B(u_2), \mu_C(u_3)\right) \quad \text{(I.16)}$$

Or

$$\mu_{A\times B\times C}(u_1, u_2, u_3) = \mu_A(u_1) \times \mu_B(u_2) \times \mu_C(u_3) \quad \text{(I.17)}$$

In addition to the basic operations just defined, there are other operations that are useful in the presentation of linguistic hedges. Some of these will be briefly described:

The "*concentration*" of *A*, denoted as con(*A*), is defined by :

$$\mu_{con(A)}(u) = \mu_A^2(u) \quad \text{(I.18)}$$

Because the most used membership functions are normal, it is clear that the operation of concentration leads to a membership function that lies within the membership function of the original function, thus the term concentration.

The "*dilatation*" of *A*, denoted as dil(*A*), is defined by :

$$\mu_{dil(A)}(u) = \sqrt{\mu_A(u)} \quad \text{(I.19)}$$

This operation leads to a membership function that lies outside of the membership function of the original set, thus the term dilatation.

I.2.5 Triangular norms

The over mentioned standard fuzzy operations known in classic set theory are not the uniquely defined operations. The general classes of operations that can implement the fuzzy intersection (conjunction) and fuzzy union (disjunction) are represented by triangular norm (T-norm) and triangular conorm (T-conorm or S-norm), respectively.

The triangular norm is a class of functions *T* defined from $[0,1] \times [0,1]$ to $[0,1]$ satisfying the following criteria for $a, b, c, d \in [0,1]$:

- Monotonicity : $T(a, b) \leq T(c, d)$, whenever $a \leq c$, $b \leq d$

- Commutiativity : $T(a, b) = T(b, a)$
- Associativity : $T(T(a, b), c) = T(a, T(b, c))$
- One identity: $T(a,1) = a$

The triangular conorm is a class of functions S defined from [0,1] × [0,1] to [0,1] satisfying the following criteria for $a, b, c, d \in [0,1]$:

- Monotonicity: $S(a, b) \leq S(c, d)$, whenever $a \leq c$, $b \leq d$
- Commutiativity: $S(a, b) = S(b, a)$
- Associativity: $S(S(a, b), c) = S(a, S(b, c))$
- Zero identity: $S(a,0) = a$

Some typical t-norm and S-norm operations are described for $a,b \in [0,1]$ in Table. I.1.

	Operation	**Description**
T-norm	Intersection	*min (a,b)*
	Algebraic product	*a b*
	Bounded product	*max(0, a+b-1)*
	Drastic product	*b if a = 1* *a if b = 1* *0 if a, b <1*
S-norm	Union	*max(a,b)*
	Algebraic sum	*a+ b-a.b*
	Bounded sum	*min(1, a+b)*
	Drastic sum	*a si b=0* *b si a=0* *1 si a,b > 0*
	Disjoint sum	*max[min[a,1-b], min[1-a,b]]*

Table. I.1 The main operations of triangular norms.

I.2.6 Linguistic and fuzzy variables

A fuzzy variable (e.g., color) is a variable whose values are terms or words in natural language (red, blue, green, yellow, etc) [17]. More generally, the values may be sentences in specified language, in which case, we say that the variable is linguistic. The sentences in question are formed from: words or terms, negation "*not*", connective "*and*" and "*but*", hedges like very, somewhat, quite, more or less. For example, the variable height might be expressible as tall, not tall, somewhat tall, very tall, not very tall, very very tall, tall but not very tall, quite tall, more or less tall. Therefore, the variable height as defined above is a linguistic variable.

I.2.7 Fuzzy relation

Fuzzy relation provides a measure of a degree of presence of association, interaction, or interconnection or more generally a specific common property between the elements of two or more fuzzy sets [18].

Let U_1, U_2, U_3, …, U_n be n-universes of discourse. A fuzzy relation R is a fuzzy set in the Cartesian product space $U_1 \times U_2 \times U_3 \times \ldots \times U_n$ and is expressed as:

$$R_{U_1 \times \ldots \times U_n} = \left\{ \left((u_1, \ldots, u_n), \left(\mu_R(u_1, \ldots, u_n) \right) \right) \middle| (u_1, \ldots, u_n) \in U_1 \times \ldots U_n \right\} \quad (I.20)$$

Where μ_R is the membership function of the fuzzy relation which measures the degree by which the elements $u_1, u_2, \ldots, u_n$ is related to each other. It can be represented by formulas, matrices, mappings, and directed graphs. The formula representation is usually used for infinite fuzzy relations, while the others are suitable to represent finite fuzzy relations. The most used representation is the relational matrix.

Example:

If U = {bank, shop}, and V = {chemist's, museum}, then the fuzzy relation R : proximity can be defined as :

R = {((bank, chemist's),0.4), (bank, museum), 0.8), (shop, chemist's), 0.5), ((shop, museum), 0.2)}

The relational matrix representation of the fuzzy relation R is as follows:

$$\mu_R(u,v) = \begin{array}{r} \\ bank \\ shop \end{array} \begin{array}{c} \quad Chemist's \quad museum \\ \begin{bmatrix} 0.4 & 0.8 \\ 0.5 & 0.2 \end{bmatrix} \end{array}$$

I.2.8 Compositions on fuzzy relations

There are two types of compositions on fuzzy relations: relation-relation composition and set-relation composition.

I.2.8.A Relation-relation composition

For this type of composition, two cases are considered: the first where all the fuzzy relations are defined in the same product space; the second where fuzzy relations are defined in different product space but share one set.

- Let *R* and *S* be two relations defined in the same Cartesian product space $U \times V$ and their associated membership functions be μ_R and μ_S. The composition of these two relations could be a union or an intersection. It is defined for $(u,v) \in U \times V$ as follows:

$$\mu_{R \cup S}(u,v) = \mu_R(u,v) \dot{+} \mu_S(u,v) \qquad \text{(I.21)}$$

$$\mu_{R \cap S}(u,v) = \mu_R(u,v) * \mu_S(u,v) \qquad \text{(I.22)}$$

Where "*" is the notation of an operator of T-norm class, and "$\dot{+}$"is the notation of an operator of S-norm class.

- Let *R* and *S* be two relations defined in $U \times V$ and $V \times W$, respectively, and their associated membership functions be μ_R and μ_S. The composition of these two relations is a fuzzy relation in $U \times W$, denoted by $R°S$ and defined for $(u,w) \in U \times W$ as follows:

$$\mu_{R°S}(u,v) = sup_{v \in V}[\mu_R(u,v) * \mu_S(v,w)] \qquad \text{(I.23)}$$

Where "*" denotes an operator of T-norm class.

This decomposition is called max-star composition. The most used compositions are the max-min and max-product composition.

I.2.8.B Set-relation composition

Let *F* be a fuzzy set in *U* and *R* be a fuzzy relation in $U \times V$. The max-star composition of the fuzzy set *F* and the fuzzy relation *R* is denoted by $F°R$, and defined for $(u,v) \in U \times V$ as follows:

$$\mu_{F°R}(u,v) = sup_{u \in U}[\mu_F(u) * \mu_R(u,v)] \qquad \text{(I.24)}$$

Where "*" denotes an operator of T-norm class.

I.3 FUZZY LOGIC

Fuzzy logic as its name implies is the logic underlying fuzzy or approximate reasoning. By approximate or fuzzy reasoning we mean the processes by which possibly imprecise conclusion is deduced from a collection of imprecise premises [19], [20].

From another perspective, fuzzy logic may be viewed as a generalization of multi-valued logic, in that it provides a wider range of tools for dealing with uncertainty and imprecision in knowledge representation, inference, and decision analysis. In particular, fuzzy logic allows the use of:

- fuzzy predicates (e.g., small, young, nice, …, etc),
- fuzzy quantifiers (e.g., most several, many, few, more, …, etc),
- fuzzy truth values (e.g., quite true, very true, mostly false, …, etc),
- fuzzy probabilities (e.g., quite possible, almost impossible, …, etc),
- predicate modifiers (e.g., very, more or less, quite, extremely, …, etc) .

Fuzzy logic is a powerful problem solving technique with wide spread applicability, especially in the area of control and decision making. It is most useful when a mathematical model of the plant do not exist or exist but too difficult to encode, or is too complex to be evaluated fast enough for real time operation, or involves too much memory on the designed chip architecture, and when experienced human operator is available for providing qualitative rules underlying the system behaviour in terms of vague and fuzzy sentences. Fuzzy logic are also supposed to work in situations where there is large uncertainties or unknown plant parameters and structures, and when high ambient noise level must be dealt or when it is important to use inexpensive sensors and/or low precision micro-controllers.

I.4 FUZZY INFERENCE MECHANISM

Most fuzzy statements can be written under the form “x is A”, which mean that the linguistic variable x takes the linguistic value (term or label) A, associated to a fuzzy set on a certain universe of discourse.

In fuzzy logic, the degree of truth of fuzzy statement “y is B`” is inferred from the degree of truth of a given fuzzy statement “x is A`” and a given implication “(x is A)→(y is B)”. This fact allows inferring a non-trivial conclusion even in case of imperfect match of the available statement with the antecedent part of the implication. This inference process proposed by Zadeh in [16] can be represented as follows:

x is A`	Premise
x is A→y is B	Implication
y is B`	Conclusion

It is known as the *generalized modus ponens*, since the classical modus ponens is recovered when the statements are non-fuzzy with A` = A and B` = B.

I.5 FUZZY LOGIC SYSTEMS

A fuzzy logic system represents a functional mapping from a set of input variables to one or more output variables[21], [22]. This functional mapping is described by means of fuzzy IF-THEN rules of the following general form:

IF *fuzzy antecedent proposition* ***THEN*** *fuzzy consequent proposition*

The fuzzy antecedent proposition is always a simple fuzzy proposition of the type "*x is A*", or a compound fuzzy proposition of the type "x_1 *is* A_1 *and* x_2 *is* A_2 *and ... and* x_n *is* A_n". Depending on the form of the fuzzy consequent proposition, two major types of fuzzy logic systems are distinguished:

- ***Mamdani (or Linguistic) fuzzy logic system*** deals with fuzzy IF-THEN rules where both the antecedent and the consequent parts are fuzzy propositions.
- ***Takagi–Sugeno (TS) fuzzy logic system*** deals with fuzzy IF-THEN rules where the antecedent part is a fuzzy proposition and the consequent part is a crisp function.

Our interest in this thesis is essentially focused on the Mamdani fuzzy logic system.

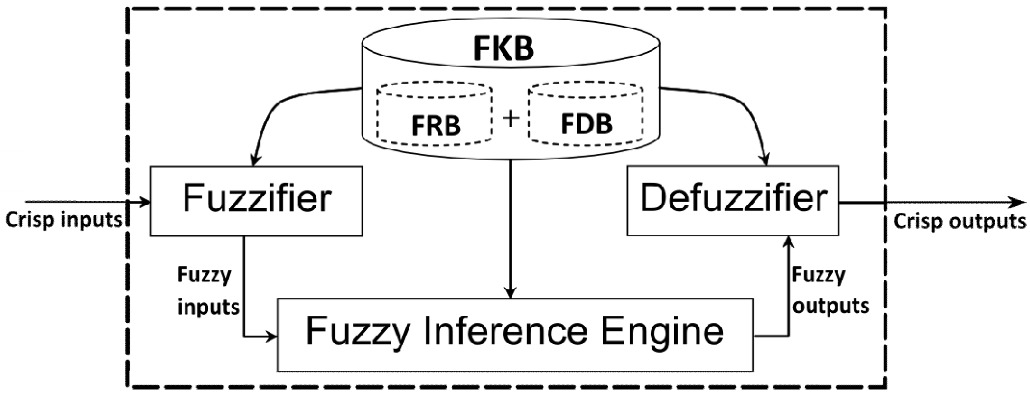


Fig. I.5 Basic structure of fuzzy logic system

I.6 MAMDANI FUZZY LOGIC SYSTEM

As illustrated in Fig. I.5, the basic structure of Mamdani fuzzy logic system consists of four main components:

- Fuzzifier;
- Fuzzy Knowledge base;

- Fuzzy inference engine;
- Defuzzifier.

Each of these components will be the subject of a detailed description in what follows.

I.6.1 Fuzzifier

The fuzzifier converts the crisp inputs $\underline{u}_0 = (u_{01}, u_{02},\ldots, u_{0n})^T \in U$ to a fuzzy set $F_x=F_{x0}\times F_{x1}\times\ldots\times F_{xn}$ defined in U, with n is the number of input variables . This unit is needed because in practical applications the observed data are crisp while in fuzzy system the manipulation of data is based on the fuzzy set theory. At least there are two choices of this conversion:

Singleton fuzzification where the crisp input $\underline{u}_0 \in U$ is converted to a fuzzy singleton F_x in U defined in term of MF as follows:

$$\mu_{F_X}(\underline{u}) = 1 \quad \text{if } \underline{u} = \underline{u}_0 \qquad \text{(I.25)}$$

$$\mu_{F_X}(\underline{u}) = 0 \quad \text{if } \underline{u} \neq \underline{u}_0 \qquad \text{(I.26)}$$

This strategy is largely used in fuzzy control applications due to its simple implementation.

Non-singleton fuzzification in which the MF value $\mu_{F_X}(\underline{u})$ is equal to unity if $\underline{u} = \underline{u}_0$ and decreases from 1 as $\underline{u}$ moves away from $\underline{u}_0$. For example, $\mu_{F_X}(\underline{u})=\exp(-(\underline{u} - \underline{u}_0)^T. (\underline{u} - \underline{u}_0)/\sigma^2)$ where σ is a parameter characterizing the shape of μ_{F_X}. The non-singleton fuzzifier may be useful if the inputs are corrupted by noise. The shape of the function can be an arbitrary but must suits the expert in term of simplicity, and computational efficiency.

I.6.2 Fuzzy knowledge base

The fuzzy knowledge base consists of a fuzzy data base and a fuzzy rule base. The fuzzy data base is a collection of concepts related to definition of the fuzzy variables of the fuzzy logic system, such as the boundaries of the universes of discourse, the number of membership function distributed within these universes, the shape of membership functions (e.g., triangular, trapezoidal or Gaussian) and its descriptive parameters (e.g., the width and the center if the shape is symmetric triangular).The fuzzy rule base is a set of fuzzy IF-THEN rules which defines the relation between the observation (or antecedent) and the action (conclusion). Each of these rules is generally expressed for a typical multiple input single output (MISO) fuzzy logic system as:

$$R^{(l)} : \textbf{\textit{IF}}\ (\ x_1 \text{ is } A_1^l \text{ and } x_2 \text{ is } A_2^l \text{ and } \ldots\ldots\ldots\ldots \text{and } x_n \text{ is } A_n^l\ )$$
$$\textbf{\textit{THEN}}\ (\ y \text{ is } C^l\ ) \qquad \text{(I.27)}$$

Where x_i and y are linguistic variables. They correspond generally to the state variable. A_i^l and C^l are linguistic terms associated to the fuzzy sets F_i^l *and* G^l, with $i=1,2,...,n$ and $l=1,2,...,M$. M is the size of the fuzzy rule base that depends on the number of input/output variables and on the number of fuzzy sets associated with each variable.

Each fuzzy rule defines a fuzzy implication which is simply a fuzzy relation defined as:

$$R^l = F_1^l \times F_2^l \ldots \times F_n^l \to G^l = \{\ ((\underline{u}, v), \mu_R^l(\underline{u}, v)) \ \ |\ \ \underline{u} \in U, v \in V\ \} \qquad \text{(I.28)}$$

Where $\mu_R^l(\underline{u}, v)$ is known as *fuzzy implication rule*. The Cartesian product between the input fuzzy sets implements the ***'and'*** connector which interpret the fuzzy conjunction in the If part.

The main fuzzy implication rules used in fuzzy logic are given in Table II., where $F^l = F_1^l \times F_2^l \ldots \times F_n^l$.

Name	DESCRIPTION
Rule of operation min. (Mamdani)	$min[\mu_F^l(\underline{u}), \mu_G^l(v)\ ]$
Rule of operation product (Larsen)	$\mu_F^l(\underline{u}) \cdot \mu_G^l(v)$
Arithmetic rule (Lukasiewicz)	$min[\ 1, 1 - \mu_F^l(\underline{u}) + \mu_G^l(v)\ ]$
Max-min rule (Willmot)	$max[\ min[\mu_F^l(\underline{u}), \mu_G^l(v)], 1-\mu_F^l(\underline{u})\ ]$
Fuzzy implication rule of the standard sequence (Rescher-Gaines)	$1 \quad if \quad \mu_F^l(\underline{u}) \le \mu_G^l(v)$ $0 \quad if \quad \mu_F^l(\underline{u}) > \mu_G^l(v)$
Booleen fuzzy implication rule	$max[\ 1-\mu_F^l(\underline{u}), \mu_G^l(v)\ ]$
Goguen fuzzy implication rule (Goguen)	$1 \qquad if \quad \mu_F^l(\underline{u}) \le \mu_G^l(v)$ $\mu_G^l(\underline{u}) / \mu_F^l(v) \quad if \quad \mu_F^l(\underline{u}) > \mu_G^l(v)$

Table. I.2 The principle fuzzy implication rules.

I.6.3 Fuzzy inference engine

Based on the fuzzy rules and the compositional rule max-star, the fuzzy inference engine derives from each fuzzy rule an output fuzzy set B^l defined in V from the input fuzzy set F_x defined in U in the following manner:

Each fuzzy rule described by a fuzzy implication R^l, determines a fuzzy set $B^l = F_x \circ R^l$ in V such that:

$$\mu_{B^l}(v) = \mu_{F_x \circ R^l}(v) = sup_{u \in U} \{ \mu_{F_x}(\underline{u}) * \mu_{R^l}(\underline{u}, v) \} \qquad (I.29)$$

Example:

Suppose that the fuzzy rule base of a fuzzy system contains the following two rules:

$R^{(1)}$: ***IF*** *(* x_1 *is* A_1^1 *and* x_2 *is* A_2^1 *)* ***THEN*** *(* y *is* C^1 *),*

$R^{(2)}$: ***IF*** *(* x_1 *is* A_1^2 *and* x_2 *is* A_2^2 *)* ***THEN*** *(* y *is* C^2 *).*

Let choose the triangular shape for the MFs, a singleton fuzzifier, the composition max-min, fuzzy conjunction min. The fuzzy inference process could be interpreted graphically on Fig. I.6(a) and (b) using the fuzzy implication rule of Mamdani (rule of operation min) and fuzzy implication rule of Larsen (Rule of operation product), respectively. In both cases, the output fuzzy set of each fuzzy rule is given by: $B^l = F_x \circ R^l$ with $l=1,2$; its corresponding membership function is expressed as:

$$\mu_{B^l}(v) = \max_{\underline{u} \in U} \{ \min\{ \mu_{F_x}(\underline{u}), \mu_{R^l}(\underline{u}, v)\} \} \qquad (I.30)$$

$$= \max_{\underline{u} \in U} \{ \min\{ \mu_{F_x}(\underline{u}), (\mu_{F^l}(\underline{u}) * \mu_{G^l}(v)) \} \} \qquad (I.31)$$

$$= \mu_{F^l}(u_{01}) * \mu_{G^l}(v) \qquad (I.32)$$

$$= \min\{ \mu_{F_1^l}(u_{01}), \mu_{F_2^l}(u_{02}) \} * \mu_{G^l}(v) \qquad (I.33)$$

$$\mu_{B^l}(v) = \alpha^l * \mu_{G^l}(v) \qquad (I.34)$$

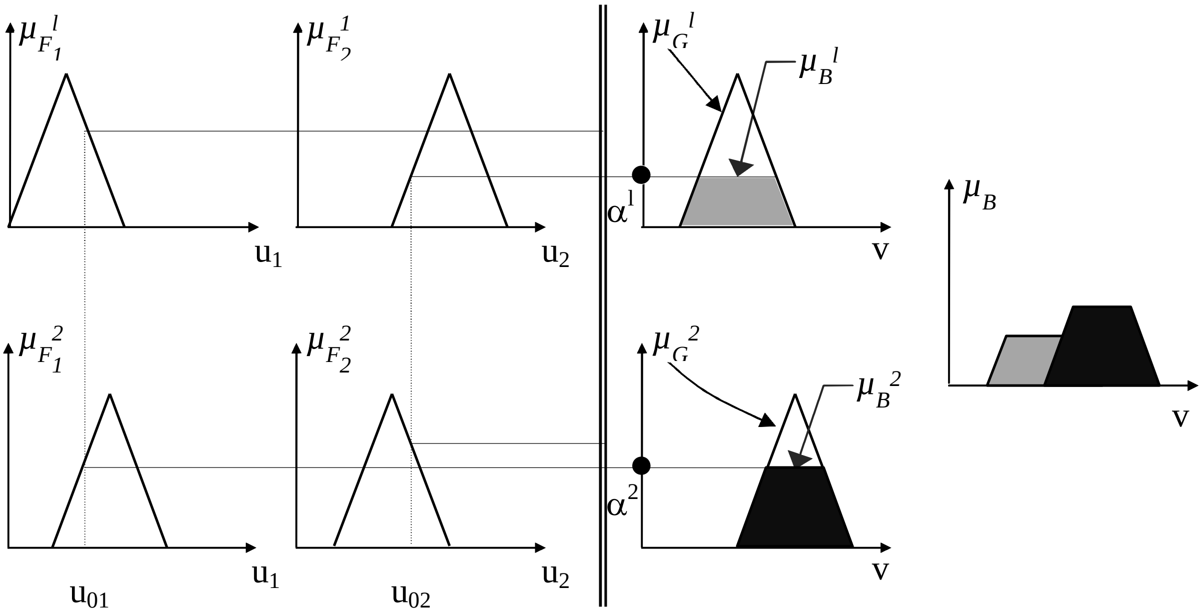
where $\alpha^l = \min\{ \mu_{F_1^l}(u_{01}), \mu_{F_2^l}(u_{02})\}$, and * design the operation min or product depending on the case.

I.6.4 Defuzzifier

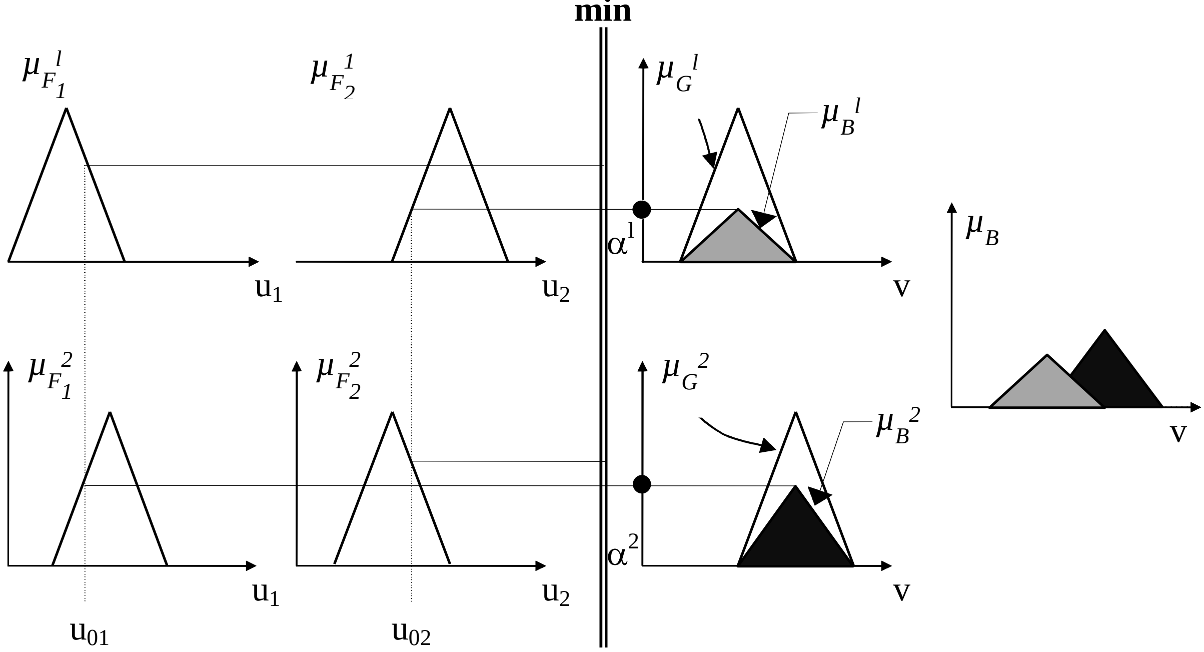
The defuzzifier provides a crisp value based on the fuzzy sets issued from the fuzzy inference engine. Usually, there are two approaches: defuzzifying without aggregating approach and aggregating without defuzzifying approach.

I.6.4.A Defuzzifying without aggregating approach

The basic idea of this approach is to exploit the information inferred from each rule directly in the process of defuzzification. Example of the defuzzification strategies included in this category are height defuzzification and modified height defuzzification.



(a)



(b)

Fig. I.6 Graphical interpretation of the fuzzy inference based on: (a) Mamdani implication rule, (b) Larsen implication rule

I.6.4.A.(a) Height defuzzification (center-average)

Let v^l be the center of the membership function, i.e. the point with the largest membership value of a fuzzy set B^l associated to the activation of the l^{th} rule. The defuzzifier with this strategy compute $\mu_B{}^l(v)$ at v^l then compute the output as:

$$y = \frac{\sum_{l=1}^{M} v^l \mu_{B^l}(v^l)}{\sum_{l=1}^{M} \mu_{B^l}(v^l)} \qquad \text{(I.35)}$$

This strategy is simple and efficient because the centres of the used MFs are usually known ahead. However, it doesn't take into a count if the MF support is either large or narrow.

I.6.4.A.(b) Modified height defuzzification

Just like for the height defuzzification, let v^l be the center of the membership function of a fuzzy set B^l associated to the activation of the l^{th} rule. The defuzzifier computes first $\mu_B{}^l(v)$ at v^l then computes the output as:

$$y = \frac{\sum_{l=1}^{M} \frac{v^l \mu_{B^l}(v^l)}{\sigma^{l^2}}}{\sum_{l=1}^{M} \frac{\mu_{B^l}(v^l)}{\sigma^{l^2}}} \qquad \text{(I.36)}$$

Where σ^l measures the MF support of the l^{th} rule. For triangular and trapezoidal shape, σ^l represents the base of the triangle and the trapezoid, respectively. While for the Gaussian MFs, σ^l is the standard deviation.

I.6.4.B Aggregating before defuzzifying approach

In this approach, the fuzzy sets issued from all the fuzzy rules are first aggregated to get a final fuzzy set $B= F_x \ o \ \{R^1, R^2, ..., R^M\}$. This aggregation uses the fuzzy disjunction which interprets the connector '***also***' of the fuzzy rules. The MF of the final fuzzy set is defined as:

$$\mu_B(v) = \mu_{F_x \circ R^1}(v) \dot{+} \mu_{F_x \circ R^2}(v) \dot{+} \dots \dot{+} \mu_{F_x \circ R^M}(v) \qquad \text{(I.37)}$$

where $\dot{+}$ denotes an operation of the S-norn class.

As a second step, the final fuzzy set is defuzzified by one of the following defuzzification strategies.

I.6.4.B.(a) Maxima strategies

Maxima strategies consider values *v* for which the membership function value $\mu_B{}^l(v)$ is maximum. To resolve the conflict in multiple maxima case, different maxima methods were proposed, e.g., *first of maxima (FOM), last of maxima (LOM)* and *mean of maxima (MOM).*

I.6.4.B.(b) Center of gravity strategy

The center of gravity defuzzification is the most used defuzzification method. It is defined as:

$$y = \frac{\sum_{i=1}^{N} v_i \mu_B(v_i)}{\sum_{i=1}^{N} \mu_B(v_i)} \qquad \text{(I.38)}$$

Where, *N* is the number of the discrete points in the output *y*.

I.6.4.B.(c) General defuzzification strategies

The basic idea underlying all these strategies is to perform some transformation of the membership function to a possibility distribution according to an automatically generated set of parameters [23].

The crisp output of the defuzzifier unit can be written as:

$$y = \frac{\sum_{i=1}^{N} \mu_i T_i v_i}{\sum_{i=1}^{N} \mu_i T_i} \qquad \text{(I.39)}$$

Where, *T* is a transformation function. Some examples of such functions are given in Table. I.3 where,

$V=\{v_1 , ..., v_N\}$,

$B=\{(v_i , \mu_B(v_i)= \mu_i) \quad | \quad v_i \in V \}$,

$\mu_m = max(\mu_i)$,

$M=\{i \quad | \quad \mu_i = \mu_m , i\in\{1,....N\}\}$,

$H=\{i \quad | \quad \mu_i \geq \alpha , i\in\{1,....N\}\}$,

$L=\{i \quad | \quad \mu_i < \alpha , i\in\{1,....N\}\}$,

α, β and γ are the transformation parameters such that :

$\alpha \in [0, \mu_m]$, $\beta\in[0,1]$ and $\gamma \in [0,\infty]$.

Transformation function	**Defuzzification strategy**
$T_i \quad (i=1, ..., N)$	Center of gravity defuzzification
$T_i = \begin{cases} 0 & if \quad i \in M \\ 1 & if \quad i \notin M \end{cases}$	Mean of maxima defuzzification
$T_i = (\mu_i)^{\gamma-1}$	Basic Defuzzification Distribution strategy (BADD)[24]
$T_i = \begin{cases} 1-\beta & if \ i \in L \\ 1 & if \ i \in H \end{cases}$	Semi Linear DEfuzzification strategy (SLIDE) [25]
$T_i = \begin{cases} 1-\beta & if \quad i \notin M \\ 1+\beta \dfrac{\sum_{j \notin M} \mu_j}{m\mu_m} & if \quad i \in M \end{cases}$	Modified Semi Linear DEfuzzification strategy (M-SLIDE) [23]
$T_i = exp(-\beta\,(\mu_i-\mu_m)^2\,)$	Gaussian distribution Transformation based Defuzzification strategy (GTD) [23]
$T_i = \left[\sum_{j=0}^{N} \beta_j (\mu_i - 0.5)^j \right]^2$	Plynomial Transformation based Defuzzification strategy (PTD) [23]

Table. I.3 Examples of transformation function and the corresponding defuzzification method.

I.7 DESIGN OF FUZZY LOGIC SYSTEM

The design process of the fuzzy logic system involves several steps, which can be summarized as follows:

I.7.1 Identifying the system variables

The first step in the design process is defining the fuzzy logic system in term of input and output variables. In control applications, the input variables are determined by the type of the controller to be used. For example, if the fuzzy controller is fuzzy PD-like controller, the input variables are the error and the error change of the state variable; if it is fuzzy PI-like controller, the input variables are the error and the integral of the error of the state variable. The output variables represent, in control applications, the control actions or the variation of the control actions to be applied to the system under control [26].

I.7.2 Establishing the fuzzy knowledge base

The next step is to set up the identified fuzzy variables on the appropriate universes of discourse. Then, each variable is associated with several fuzzy sets which must be labelled according to the problem to be solved. The membership functions characterising the fuzzy sets must also be chosen. The adjacent membership functions must be overlapped, generally with 20% to 100% of the adjacent MF boundary. Finally, it is important to note that a properly choice of the fuzzy rules is very critical in this step. In the earlier applications, it depends strongly on the experience and knowledge of the operator. For PD-like linguistic controller, MacVicar-Whelan proposed some general heuristics guidelines [27] that Yager and Filv called in [28] '*meta-rules'* to derive a standard template FRB also called Micar-Whelan fuzzy control matrix. The Micar-Whelan meta-rules are expressed as follow [29] :

1) If both the error and change in error are zero, then change in output is zero.
2) If the error is tending to zero at a satisfactory rate, then change in output is zero.
3) If the error is not self-correcting, then change in output is not zero and depends on the sign and magnitude of the error and change in error.

Micar-Whelan fuzzy control matrix defines a reasonable set of fuzzy rules that can be adjusted and adapted to fit the specificity of the control problem. Table. I.4 shows an example of such matrix for a fuzzy PD-like controller having the input and output variables fuzzified into seven fuzzy sets. The fussy sets are associated to the following labels: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM) and Positive Big (PB).

		Error change						
		NB	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
error	**NB**	NB	NB	NB	NB	NM	NS	Z
	NM	NB	NM	NM	NM	NS	Z	PS
	NS	NB	NM	NS	NS	Z	PS	PM
	Z	NB	NM	NS	Z	PS	PM	PB
	PS	NM	NS	Z	PS	PM	PM	PB
	PM	NS	Z	PS	PM	PM	PM	PB
	PB	Z	PS	PM	PB	PB	PB	PB

Table. I.4 Example of MacVicar-Whelan fuzzy control Matrix

The performances of the fuzzy logic systems in fact depend drastically on the design of the fuzzy rules as well as the MFs. Usually, this step is performed on the basis of expert heuristic knowledge or trial and error. More recently, a number of automatic generation methods of FKB have been suggested such as self-tuning and optimization methods. These methods are described in detail in the next section.

I.7.3 Defining the structure of the fuzzy logic system

The choice of the structure of fuzzy logic system includes the choice of fuzzification and defuzzification types, the operators that implement the fuzzy conjunction ‘and’, the fuzzy disjunction ‘also’, the fuzzy implication rule and the max-star composition.

I.7.4 Validation of the designed fuzzy logic system

The goal of this step is to evaluate the designed fuzzy logic system behaviour with respect to its response from a set of predefined experimental inputs. These inputs are generated by the developers or the target system experts. If the fuzzy logic system fails to meet the expected performance, we have to iterate on the above design steps.

Note: It should be noted that the success of these design steps strongly relies on the problem at hand, the soundness of the knowledge acquisition techniques and the amount and quality of the available expert knowledge. For some problems, the fuzzy logic system design may lead quickly to efficient systems, while for others it may be a very time-consuming and inefficient procedure.

I.8 GENERATION OF THE FUZZY KNOWLEDGE BASE

The crucial problem in fuzzy system design is the generation of the FKB just as for the expert systems. A large number of approaches have been developed to overcome this problem. They can be classified according to the used method into four categories: direct approaches, approaches based on classical identification algorithms, approaches based on self learning methods and approaches based on optimization methods.

I.8.1 Direct approaches

In this category, the FKB is directly generated from the expert’s *a priori* knowledge. In fact the process of the FKB generation can be performed in different ways. On simple way is by interrogating the human expert or a skilled operator using a carefully formulated questionnaire. Another way is by observing the skilled operator manipulating the system. These approaches are the first used to build the FKB in the earlier fuzzy system applications [2], [30]. There is no general methodology for implementing these approaches which is more an art of intuition and experience than precise theory [31].

I.8.2 Approaches based on classical identification algorithms

The principle behind these approaches is that the fuzzy logic system is considered as a special type of non linear system that could be estimated by the classical non linear identification methods. The research studies using this kind of approaches use, for example, non linear least-square parameter estimation [32], orthogonal least-squares [33], gradient descent [34], quasi-Newton [35], Levenberg-Marquardt [36], and auto-regressive modeling [37].

I.8.3 Approaches based on self learning methods

Self-learning fuzzy systems -also known in the literature as self organizing, self tuning or adaptive- is a fuzzy system adopted with self-learning capability to facilitate the heuristic adjustment of the FKB and also to cope with the time varying systems. In general, the learning process in fuzzy systems could be done off line or on line the real-time application. Mamdani and Procyk have proposed the first self learning fuzzy system in [38]. This paper was a seminal article of that period because it reports a major breakthrough in introducing the adaptation for fuzzy controllers. Afterwards, some additional works have been reported in [39] and [40]. Unfortunately those methods are efficient only in set point control and behave poorly in tracking control. To overcome this drawback, Layne and Passino proposed a fuzzy model reference learning control algorithm based on model reference adaptive control (MRAC) in [41] and [42]. Since then, various self-learning approaches were developed and they were successfully used for a wide variety of applications [43]. An interesting approach among them is the implementation of the fuzzy system into a neural network and the application of the adaptive algorithm for connection weights adjustment such as the back-propagation gradient descent algorithm[44], [45] and adaptive resonance based algorithms[46]. The systems based on this approaches are usually referred to as neuro-fuzzy systems and usually represented as special multilayer feed-forward neural networks, for example systems ANFIS [47], FuNe [48], Fuzzy RuleNet [49], GARIC [50], and NEFCLASS, NEFPROX and NEFCON [51].

I.8.4 Approaches based on optimization methods

The FKB generation can be considered as an optimization problem where part or all of the parameters of the FKB constitute the design parameters. Theses parameters are found to influence the performance of the fuzzy system in unknown and co-dependent manner. Both of these facts make the search space of this problem large and complex. Since the impressive success achieved by GAs in FKB generation ([52], [53], [54]) the optimization community has shown a growing interest in this issue as can be seen through the multiple contributions reported in literature, e.g., [55], [56], [57], [58], [59], [60] and [61] just to mention a few. These contributions make use of different meta-heuristic and soft computing techniques such as tabu search [55], [56], EAs [57], [58], simulated annealing [59], particle swarm optimization algorithm [60] and ant colony algorithm [61]. The remarkable thing is that the genetic/evolutionary algorithms are continuing to dominate the research on this issue which is still on the upswing. This is due in fact to the appealing capability of the EAs to deal with the optimization problem on large and complex search space.

I.9 CONCLUSION

The ultimate goal of this chapter is to give a comprehensive overview of theoretical foundations of the fuzzy set theory and its use in fuzzy logic system. A short primer on fuzzy set theory was first introduced. Then, we presented the principle of fuzzy reasoning which forms the basis of fuzzy logic. Next, the definition of fuzzy logic was given followed by the enumeration of the situations in which the fuzzy logic is recommended. Based on the type of fuzzy rules, two types of fuzzy logic were identified: Mamdani fuzzy logic system and Takagi-Sugeno fuzzy logic system. A detailed description of the structure of Mamdani fuzzy logic system was presented and followed by a brief description of the designing steps of the fuzzy logic system. Finally, a classification of the FKB generation approaches is drawn up based on the type of the used method.

Chapter II: Genetic and Evolutionary Algorithms

II.1 INTRODUCTION

There have been an increased interest in the methodologies for solving optimization problems which involve the determination of a set of parameters that optimize (i.e., minimize/maximize) a given function with respect to some finite set of constraints. This kind of problems is encountered in one form or other in almost every field, in particular, the engineering domain. Accordingly, variants of techniques were proposed in the literature, including calculus-based techniques, enumerative techniques and stochastic techniques. Most of them are based on the gradient for finding the direction, which impose the existence of the derivative function. But, in practice, a large number of functions to be optimized are non-differentiable everywhere and even discontinuous, which make such techniques inefficient in finding the global optimal solution in the real problems. Another major deficiency arises when the function to be optimized is multi-modal (i.e., multiple peaks). In this case, the extremes reached are optimum only if the starting point is in the vicinity. So, it is obvious that starting near a local optimum, the search process will converge to this point and it will be considered as a global optimum.

In the past decade, a new optimization technique biologically motivated has received a great deal of attention regarding their capability to reach rapidly the near-optimal solution for complex optimization problems. This technique — called genetic algorithm (GA) — is a search procedure inspired by biological paradigm of natural selection and genetics [62], [63], [64], [65], [66]. The GAs have demonstrated their power as optimizer in different applications ranging from mathematics and engineering to finance and management [67], [68], [69], [70].

In this chapter we introduce the fundamentals concepts of standard genetic algorithm and we underline its working mechanism. We also give the genetic algorithm variants grouped in a class of search methods referred as EAs and belong to the evolutionary computation discipline. Some aspects related to evolutionary optimization are also presented.

II.2 OVERVIEW OF STANDARD GENETIC ALGORITHM

GAs are powerful exploratory search and optimization algorithms founded on the mechanism of natural selection and genetics. They were first introduced by Holland [71], and subsequently they were extensively studied and explored by Goldberg [66] and Davis [72].

Fig. II.1 shows a general description of a standard genetic algorithm (S-GA). This later evolves a population of encoding of the potential solutions of the problem to be solved to explore the search space. These encodings are called chromosomes, individuals, or genotypes. To determine how well each chromosome solves the problem, S-GA calculates a "fitness" function (objective function or cost function) which measures the profit, the utility or the quality to be optimized. Along the generations, the S-GA tends to improve the fitness of the population by selecting chromosomes (parents) according to the basic criteria of "survival of the fittest", and then applying the genetic operators which are the crossover and mutation, examples of their application is depicted in Fig. II.2. These operators serve for the generation of the new chromosomes (children or offspring) by recombining parts of the selected parents in a random manner using crossover operator, and by random alteration of one gene in the chromosomes using the mutation operator. The genetic operators are in more details described in Section III. . Thus, S-GAs are able to use historical information as a guide through the search space. The resulting chromosomes are again evaluated and transformed using such probabilistic operators. This genetic process is repeated until a termination criterion is satisfied. The most commonly used termination criteria are:

- a suitable solution is found (a solution that solves the problem, within a specified tolerance);
- user-specified maximum of number of generations is reached;
- user-specified maximum computation time (runtime) is reached;
- the best fitness function of the current generation reach a user-specified fitness threshold;
- The best fitness function of the current generation has not improved for a certain number of consecutive generations;

- Human judgment and inspection can also be used in some more subjective cases.
- Combinations of the above criteria.

- *Randomly generate an initial population ;*
- *Evaluate each chromosome in the population ;*
- *Do {*
  - *Select the chromosomes of the new population ;*
  - *Generate new chromosomes by applying genetic operators : crossover and mutation ;*
  - *Evaluate the new chromosomes of the population ;*

  } while (termination criterion is not satisfied)

Fig. II.1 Abstract description of a standard genetic algorithm.

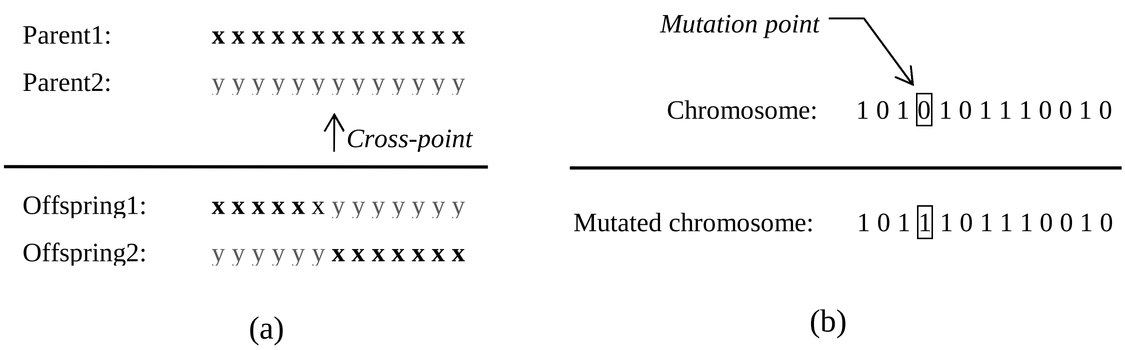


Fig. II.2 Example of basic genetic operators. (a) crossover operator, (b) mutation operator.

II.3 EVOLUTIONARY ALGORITHMS

Around the S-GA algorithm a lot of new population-based optimization methods have been proposed to improve its performance and extend its applicability to a wide variety of domains. This is basically obtained by introducing some modifications. The resulting methods lead to the emergence of new discipline referred as evolutionary computation. From this class of computational approaches, very interesting algorithms have been appeared such as evolutionary programming (EP), evolutionary strategies (ES), genetic programming (GP). It should be noted that the modifications made in to the S-GA can affect:

- **The chromosome encoding** which can severely limit the window by which the algorithm observes its world ;
- **The genetic operators** that introduce new chromosomes ;
- the way to create **the initial population** ;
- **the fitness function** which measure how close the associated solution is to the optimum one ;
- **the setting of the parameters**— commonly called control parameters— that GA uses, such as population size, probabilities of applying the genetic operators, etc.

An overview of these modifications will be given in the following sections.

II.4 OPTIMIZATION PROBLEMS SOLVED BY THE EVOLUTIONARY ALGORITHMS

Depending on the number of variables to be optimized, the number of objectives to be satisfied simultaneously and the existence of some constraints imposed upon the search space, the problems solved by the EAs can be classified as follows:

- Single and multi-variable problems;
- Single and multi-objective problems;
- Constrained and unconstrained problems.

II.4.1 Single and multi-variable problems:

For multi-variable problems, the potential solutions corresponding to the design variables are coded and concatenated to form one chromosome, see the example in Fig. II.3.

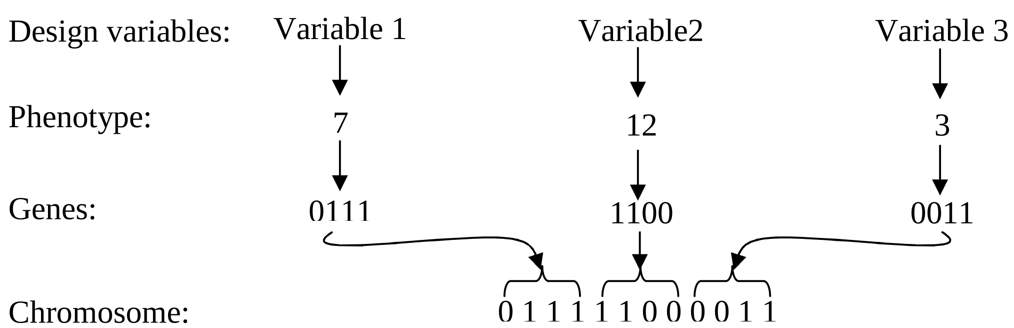


Fig. II.3 Binary encoding example for multi-variable optimization.

II.4.2 Single and multi-objective problems:

Quite often, the real-world optimization problems require that multiple objectives must be satisfied simultaneously. The application of EAs to multi-objective problems has become very popular in recent years. Although the earlier EAs are designed for single optimization problems which need to achieve a single objective, they can be used to deal with multi-objective problems. The conceptual approaches to do this are mainly concerned with fitness assignment or the chromosome evaluation for the selection operator. One of these approaches consists in lumping together all the different objectives in a single objective function through a weighting scheme. Another approach entails setting all the objectives except one of them as constraints in the optimization process. The objectives considered as constraints are assigned different levels of attainment of their respective objective functions. Single-objective EAs used for multi-objective problems yield one optimum solution. On the contrary, in multi-objective EAs, there is no single optimum solution, but a set of alternative solutions with different trade-off between the different objectives. This set of solutions is largely known as the compromised, trade-off, non-dominated, non-inferior or Pareto-optimal solutions. To get this set of solutions, several runs are performed with the single-objective EAs specifying in each run different weights or levels of attainment for each objective. However, the use of the multi-objective EAs provides wider range of Pareto-optimal solutions in just a single run which promotes the roles of the analysts (modellers) and the decision makers in the optimization process.

II.4.3 Constrained and unconstrained problems:

The constraints in EAs are usually handled with different strategies that can be grouped into four classes[73], [74].

- ***Rejection-based class:*** The unfeasible chromosomes are rejected and discarded from the population during the evolution. It is the first proposed and the simplest way to deal with the unfeasible chromosomes. As results, the population size decreases and the exploration of the search space is not done effectively, especially when the initial population consists of only unfeasible chromosomes.

- ***Penalty-based class:*** the constrained optimisation problem is converted to an unconstrained problem by penalizing the unfeasible chromosomes. Although, the principle of this class is conceptually very simple, in the implementation it is quite difficult to design or formulate the penalty functions for effective search.

- ***Repairing based class:*** The unfeasible chromosomes are repaired and converted to feasible ones. This class is based on additional function evaluations.
- ***Modified evolutionary components based class:*** The feasibility of the chromosomes is maintained by problem-specific evolutionary components (decoder, evolutionary operator, fitness function, etc). In such class of strategies, the unfeasible chromosomes are never generated. However, faster convergences and better solutions could be found in the unfeasible regions [74].

II.5 CHROMOSOME ENCODING

Various encoding methods have been proposed for particular problems in order to represent the potential solution in the population. Based on the types of the alphabets or symbols used as the alleles of a gene, three key types of encoding strategy are possible.

II.5.1 Binary encoding

Binary encoding is certainly the first encoding strategy used in EAs. Using a binary alphabet {0,1}, the solution is encoded in a binary string of a particular length, defined by the user and depends on the desired precision. This type of encoding offers several advantages including minimum number of alphabet {0,1}, ease in implementing genetic operators, and the existence of theoretical foundation (schemata theory). However, it shows some deficits:

- For large scale problems requiring high precision, the binary-coded GA presents poor and unsatisfactory results, as demonstrated in [75].
- The Hamming distance between two adjacent numbers in phenotypic representation could be very large in genotypic representation. For example, the integers 7 and 8 corresponding to codes :0111 and 1000, respectively, have a distance of hamming equal to one in decimal representation and four in binary representation. This phenomenon is called the Hamming-Cliff problem which could possibly lead to a convergence but not to the optimal solution [76].

II.5.2 Real encoding

Real encoding is a natural and adequate representation of real numbers in continuous domains. It can be simply defined as a direct mapping between the real parameter (phenotype) and the code (genotype). Hence, the real parameter is used directly in the chromosome evaluation without coding or decoding process. This implies a significant reduction of the computational time. Moreover, real encoding seems naturally having the capability of fine local search which is crucial for high precision optimization problems.

II.5.3 Integer/permutation encoding

To tackle the optimization problems that have integer variables whose values are unrestricted (all digits) or restricted to a finite set of digits- for example, {0,1,2,3} or {North, East, South, West}-, the integer encoding is more suitable. In this type of encoding, the genes forming the chromosome take as alleles digits in a specific base of numeral system. When the order of the genes is significant, the integer encoding became a permutation encoding. This particular issue is encountered in combinatorial optimization problems where a combination of some items is searched to meet some constraints.

Based on the structure of encoding, it can be classified into two categories, namely, one dimensional and multi-dimensional. The one-dimensional encoding strategy is the most used in EAs where the potential solutions are represented by a vector or an array of genes. In the case of optimization problems that have solutions with multi-dimensional structures, it is natural to choose a multi-dimensional encoding strategy to represent those solutions.

The encoding strategies can also be classified according to the type of the contents encoded into the chromosome which could be:

- the solution of the problem at hand alone;
- the solution and some control parameters.

The control parameters involved in the chromosome could be the EA parameters such as the crossover and/or mutation probabilities or parameters characterizing the solution itself.

II.6 GENETIC/EVOLUTIONARY OPERATORS

To generate new and better possible solutions, a set of genetic operators is applied to the population chromosomes. These operators are probabilistic and they strongly rely on the types of the alphabets used in the encoding strategy and the data structure adopted to represent the chromosomes. Three basic genetic operators are in common use: crossover, mutation and selection operators. In the sequel we present the most frequently used for one-dimensional encoding strategies. More kinds of genetic operators related to different encodings and different problems could be found in [77] and [78].

II.6.1 Crossover operator

Crossover operator is analogous to that occurring in the natural systems. It is the genetic operator that has the potential to breed significant amounts of new chromosomes (offsprings). This operator works independently of the alphabets type. According to the building block hypothesis [66], crossover operator attempts to create an offspring that is more fit than either of the two or more parents by performing different exchanges. Besides the basic one-point crossover operator there are many crossover operators, most of which are representation and problem specific. n-point crossover and uniform crossover are the most used.

Parent1: **x x x x x x x x x x x x x**

Parent2: y y y y y y y y y y y y y

Cross-points

Offspring1: **x x** y y y y **x x x** y y y y

Offspring2: y y **x x x x** y y y **x x x x**

Fig. II.4 Example of 3-point crossover.

II.6.1.A n-point crossover operator

The n-point operator also called multi-point crossover constitutes a generalization of one-point crossover operator where n cross-points are chosen randomly along the chromosome. Between each cross-point one or the other of the parent's fragment is copied and alternated for the next cross-point. Fig. II.4 show an example of the application of this operator.

II.6.1.B Uniform crossover operator

The logical extension of the n-point operator is referred to as uniform crossover [79]. At first, a bitmask is chosen randomly. It is simply a binary string with the same length as the chromosome. For each bit of this bitmask, the exchange of the genes occurs if the value of this bit is one, and if it is zero, the chromosome parents keep their genes. An example of the application of this operator is illustrated in Fig. II.5.

Parent1: **x x x x x x x x x x x x x**
Parent2: y y y y y y y y y y y y y
Bit mask: 0 1 1 0 0 0 1 0 1 0 0 1 1
Exchange of the chromosome genes

Offspring1: **x** y y **x x x** y **x** y **x x** y y
Offspring2: y **x x** y y y **x** y **x** y y **x x**

Fig. II.5 Example of uniform crossover.

II.6.2 Mutation operator

Unlike crossover operator, mutation operator acts only on one chromosome, and introduces minor modifications to the genes of this chromosome. It is implemented by altering one or more genes selected randomly. The researchers argued that the application of the mutation operator enables the recovery of genes, which are lost from the current population. Furthermore, it prevents from rapidly converging on a local optimum as it provides the system with a way to avoid getting trapped in local optima. Since mutation operator also has a destructive effect as well it is only applied relatively rarely, i.e., with low probability, as in the biological systems.

For different encoding strategy, different mutation operator types are suitable:

II.6.2.A Mutation for Binary encoding

The mutation operator in this case simply consists in flipping the selected gene, i.e., if the gene value is 1, it is changed to 0 and vice versa.

II.6.2.B Mutation for integer encoding

There are two principle types of mutation operator: random resetting (or random choice) operator and creep mutation operator. Random resetting operator chooses the new value of the selected gene from the set of permissible values. It is mostly applied for cardinal attributes. Creep mutation operator is proposed for ordinal attributes. A small random (positive or negative) integer is added to the selected gene value by the creep mutation operator. This integer random value is taken from a parametric distribution which is symmetrical around zero.

II.6.2.C Mutation for real encoding

In real-coded EAs the mutation operators used change gene value randomly within its specific range. The type of the distribution of the random changes defines the type of the mutation operator. There are two types of mutation operator: uniform mutation operator and non-uniform mutation operator. With the uniform mutation operator the changes affecting the selected gene is chosen at random from continuous uniform distribution. While a normal or Gaussian distribution with zero mean and user-specified standard deviation is used to obtain the random changes in the non-uniform mutation.

II.6.2.D Mutation for permutation encoding

As mentioned in section III.5, the locus and the order of the genes are important in permutation encoding. The available types of mutation operator take this fact into account for such representation. The most used are briefly described in what follow and examples are given in Fig. II.6.

- ***Swap mutation operator*** selects two genes at random and swap their positions. This fact preserves most of adjacency information (4 links broken), but disrupts more the genes order.

- ***Insert mutation operator*** selects randomly two genes at first. Then move the second to follow the first, and shift the rest genes along to make room. This operator preserves most of the order and the adjacency information

- ***Scramble mutation operator*** selects a subset of consecutive genes at random and rearrange randomly the genes in those positions. The adjacency information is not preserved, and the genes order is strongly disrupted.

- ***Inversion mutation operator*** selects two genes at random and then inverts the substring between them. Doing so, this operator preserves most of the adjacency information (only breaks two links) but disrupts the genes order.

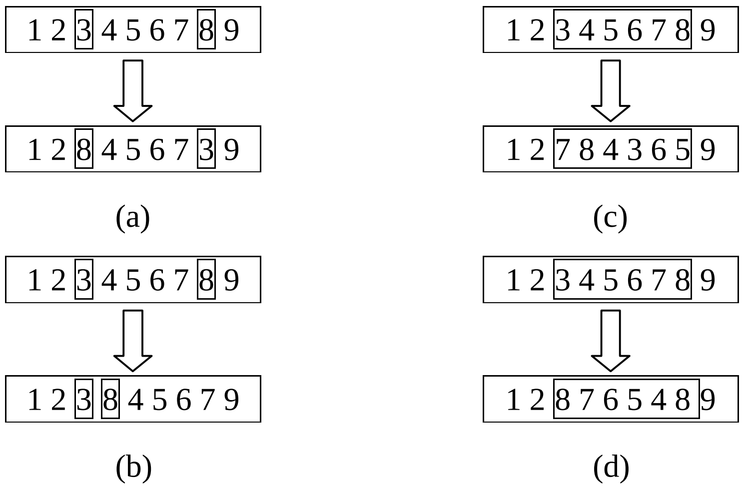


Fig. II.6 Example of mutation operator for permutation encoding. (a) swap mutation, (b) insert mutation, (c) scramble mutation, (d) inversion mutation.

II.6.3 Selection operator

The aim of the selection operator is to favor by a random process the contribution of the good chromosomes in the breeding of the new population. The selection operator is particularly useful in preventing good chromosomes from being lost. It is based on the principle of the natural evolution theory known as "survival of the fittest" in which the best individual of the population should survive and create new offspring. The roulette wheel selection operator was the first selection operator used by Holland [71]. There are many other types that differ primarily in the probability function assigned to the chromosomes. The probability function that determines the selection process can be associated to the actual fitness function value, a scaled value or the rank.

In this subsection, we present roulette wheel selection operator, tournament selection and rank-based selection.

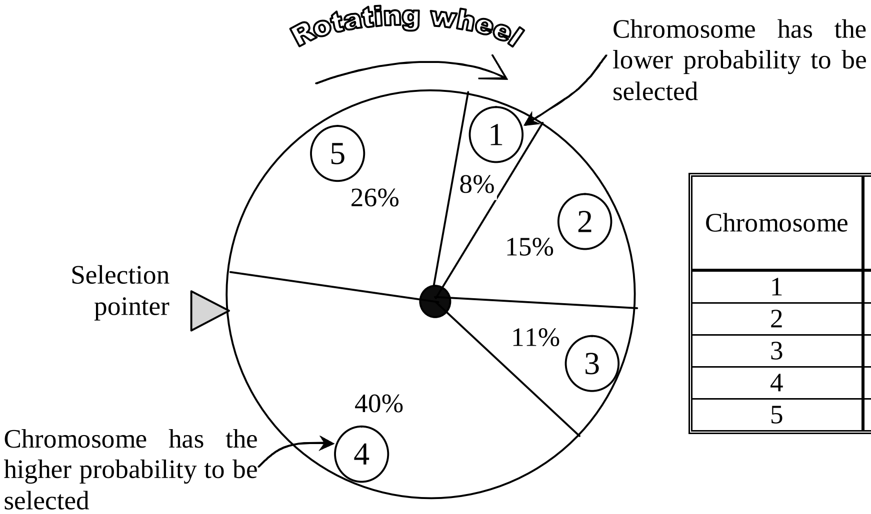
II.6.3.A Roulette wheel selection operator

The rationale behind this operator is that each chromosome is allocated an area proportional to their relative fitness function value randomly ordered around a virtual roulette wheel. The wheel is spun and the chromosome in front of the pointer when the wheel stops is selected as shown in Fig. II.7. This roulette wheel selection is implemented by first evaluating all the chromosomes of the population of size N by computing their fitness function noted as f_i. Then a relative fitness value f_r is calculated for each of these chromosomes as:

$$f_{ri} = \frac{f_i}{\sum_{j=1}^{N} f_j} \quad \text{(II.1)}$$

After that a random number *r* is generated in the range [0 1] and compared to the cumulative fitness of the chromosomes successively. The cumulative fitness of a chromosome *l* (f_{cl}) is defined as:

$$f_{cl} = \sum_{j=1}^{j=l} f_{ri} \quad \text{(II.2)}$$



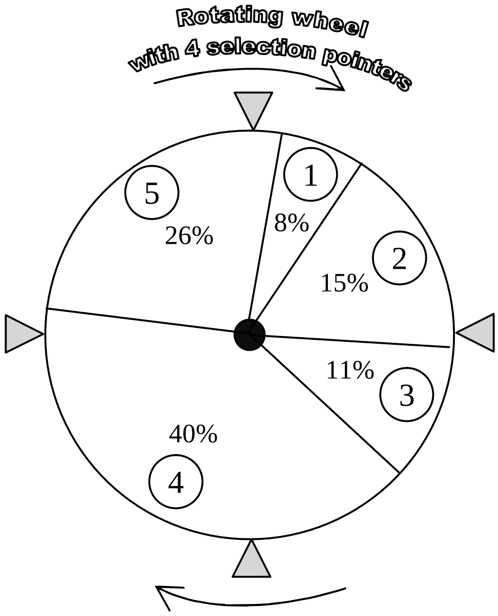
Chromosome	Relative fitness function expressed as percentage
1	8%
2	15%
3	11%
4	40%
5	26%

Fig. II.7 Example of roulette wheel

The first chromosome whose cumulative fitness is superior or equal to the random number *r* is selected to be a chromosome parent.

Among the available selection operators, the roulette-wheel selection operator is still the most widely used selection operator. However, it suffers from one drawback; it has a strong tendency to select the best chromosome several times which yields a loss of diversity and hence the efficiency of the evolutionary process. To remedy this problem, the mechanism of the roulette wheel was modified. This modification consists in removing from the wheel the chromosome once it is selected.

Another variant of this type of operator is the stochastic universal sampling developed by Baker [80]. Instead of repeating the process of spinning the wheel and picking a chromosome, this operator use a number of pointers equal to the number of chromosomes to select and equally spaced around the wheel. The wheel is spun and the chromosomes in front of the pointers are selected. This minimizes the bias and drift connected with the repeated spinning of the wheel, see Fig. II.8.



Chromosome	Relative fitness function expressed as percentage
1	8%
2	15%
3	11%
4	40%
5	26%

Fig. II.8 Example of stochastic universal sampling selection.

II.6.3.B Rank-based selection operator

The rank-based selection operator first sorts the population chromosomes according to their fitness function from best to worst. Then the rank 1 is assigned to the worst chromosome, the rank 2 to the second worst one, … etc, and the rank N is assigned to the best chromosome, where *N* is the population size. A parent chromosome is selected with a probability proportional to its rank rather than the fitness function value. As a sorting of chromosomes population is required with this operator, the evolution is slow but the diversity is preserved.

II.6.3.C Tournament selection

Tournament selection operator takes randomly a group of chromosomes from the population (two or more) for competition. The fittest of those chromosomes is selected as chromosome parent. The competition is often held between pairs of chromosomes. An example of possible tournament competition between the chromosomes is presented in Fig. II.9. In fact, the tournament selection, while slower and more complicated, can create more diversity in the population than the roulette wheel selection.

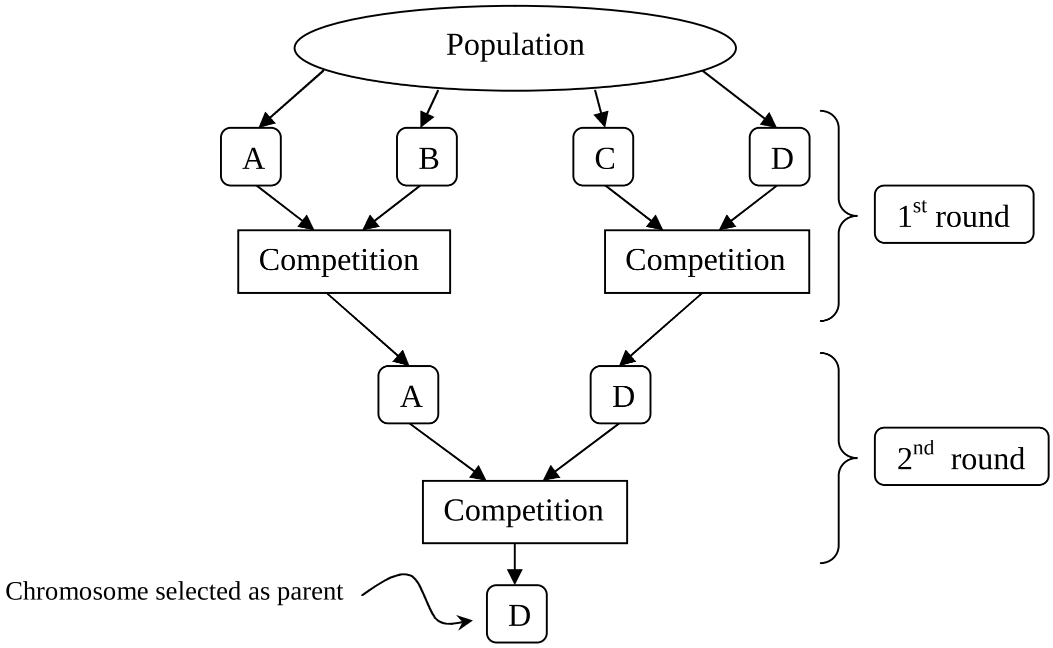


Fig. II.9 Example of tournament selection process.

II.7 POPULATION INTIALIZATION

The initial population is the first pool of chromosomes generated before the evolution process is started. It is preferable that the initial population has a good uniform coverage. This means that the solutions are well spread out to cover the whole search space, and they do not form clusters or leave relatively large regions of the search domain unexplored. Usually, the initial population is generated randomly. Especially, when there is no *a priori* information about the problem domain. In practice, genuine random (truly independent) numbers cannot be generated numerically, and instead, pseudo random numbers and quasi random numbers are used. Pseudo random numbers imitate genuine random numbers and quasi random sequences are designed to produce numbers that maximally avoid each other. The well-established pseudo random number generator are classified into congruential and recursive generators. The congruential generators include linear, quadratic, inversive, additive and parallel linear congruential generators [81] and [82]. The recursive generators include multiplicative recursive, lagged Fibonacci, multiply-with-carry-generator, add-with-carry and substract-with-borrow generators [81]. There are also pseudo random vector generators, which produce sequences of vectors instead of scalars. Examples of those are feedback shift register generator [81] and SQRT generator [83]. The frequently used quasi random sequence generators include Van der Corput, Hammersley, Halton, Faure, Sobol' and Niederreiter generators. The heuristic could be included in the initial population generation for finding high-quality initial starting solutions [84] and [85]. This technique is known as seeding and may help the genetic algorithm to find better solutions quickly.

II.8 FITNESS FUNCTION

Fitness function is an important concept in EAs. It provides a measure of how well the chromosome solves the optimization problem at hand. The value of fitness function of a chromosome is the main criterion in the selection process within the EAs. The usual way to choose the fitness function of an optimization problem is to use the objective function. The main disadvantage of this method is that few best chromosomes may dominate the population at the later stages. To maintain a certain level of competition between the chromosomes throughout evolution process, scaling mechanisms could be applied. Several scaling techniques exist including linear, exponentially scaling and sigma truncation [66].

II.9 REPLACEMENT SCHEMES

The new population containing the new chromosomes (offspring) can either replace the previous population entirely (generational replacement) or partially (steady-state replacement). In the first type of replacement, the populations are often referred to as non-overlapping populations and there is a chance that the EA will lose the best chromosome found so far. In the second type the populations are known as overlapping populations for which the chromosomes to be removed and those to be inserted are defined. When the best chromosome is chosen to survive to the next generation, the strategy is called the elitism strategy.

II.10 EXPLOITATION/EXPLORATION BALANCE

Many researchers suggest that the remarkable success achieved by EAs in solving a variety of complex problems is due to their adequate trade off between the exploration and the exploitation. The exploration allows interesting regions to be identified and the exploitation refines these regions. Too much exploration can result in very slow convergence towards the optimum solution, while an intense exploitation in the earlier generations of EAs can lead to a premature convergence (early convergence to a suboptimal solution). There are several factors that affect the exploration/exploitation balance. Those that promote the exploitation include:

- using a small population size.
- using the recombination, or choosing both parents based upon their fitness.
- not allowing for mutation.
- not selecting both an offspring and its compliment.
- immediately replacing the weakest chromosome of the population with an offspring.

The factors that retard exploitation and promote exploration are:

- using a large population size.
- choosing one parent randomly.
- allowing for significant mutations.
- selecting both an offspring and its compliment.
- allowing a chromosome of the current population to recombine before it is removed.

II.11 BENEFITS OF EAs

- Suitable for complex, multi-dimensional, non-differential, non-continuous, and even non-parametrical problems.
- EA concepts are very easy to understand and it practically does not demand the knowledge of mathematics.
- Supports multi-objective optimization
- EAs always give a solution ; and this solution gets better with time
- The concept of population in EAs makes their parallel implementation and distribution on a network of independent CPUs easy.
- Good for “noisy” environments
- Many ways to speed up and improve an EA-based application as knowledge about problem domain is gained.
- Easy to exploit previous or alternate solutions.

II.12 COMMON DIFICULTIES OF EAs

Despite the successfully application of EAs in solving a wide range of optimization problems in various domains with only little available knowledge about the problem domain, there are some fundamental problems noticed. The most common problems with EAs are highlighted in what follows.

II.12.1 Heuristic principle:

There is no guarantee that the EAs will find the global optimum. The precision of the solution obtained in a limited amount of computation cannot be also guaranteed or predicted. These facts make the application of EAs in on-line or real-time optimization very limited.

II.12.2 Difficult adjustment of parameters:

A large number of options and parameters need to be adjusted. For example, the type of genetic operators: selection, crossover and mutation operators, the settings of the control parameters of EAs such as the population size the crossover and mutation rates, the form of the fitness function, etc. This setting is required because every optimization problem has specific characteristics and must be solved in a special way. The correct setting of those options has a crucial effect on the performance of the EA. However, it is difficult to determine them primarily because of the nonlinear and complex interaction between them. In practice the successful EAs applications are often the results of the lengthy and tremendous trial-and-error procedure for particular class of problem or even problem instance.

II.13 CONCLUSION

The purpose of this chapter was to give first a description of the S-GA which is the basis of all the EAs developments, followed by a classification of the problems that could be solved by the EAs. Subsequently, we reviewed the common encoding strategies, some popular genetic/evolutionary operators, and other important ingredients of EAs. We concluded this chapter with the identification of the benefits and the common difficulties of these techniques.

Chapter III: Evolutionary Design of Interpretable Mamdani Fuzzy Controller

III.1 INTRODUCTION

FLCs have been successfully applied in several industrial areas [86], [87]. Their performance relies substantially on the components of the fuzzy knowledge base (FKB), which is traditionally obtained using painstaking iterative trial and error method. The research works on generating automatically the FKB started around 1990 [52], [53], [54], [88]. Since then, new approaches were continually developed and refined. These design approaches fall into four major classes. In the first one, fuzzy data base (FDB) is optimised with fixed fuzzy rule base (FRB) set by trial and error [52], [89]. In the second class the situation is reversed, it means the FRB is generated while the FDB is fixed [54], [88], [90]. The third class consists in generating both FRB and FDB but in stages. It involves determining the FRB considering a predefined FDB as first stage using the methods of the former class or others. Then, optimising the FDB in a second stage while using the FRB found in the previous stage [91], [92]. The fourth class is based on the fact that the ingredients of the FKB are co-dependent, so their simultaneous optimisation is more appropriate [93], [94]. Among the available optimization and learning methods, EA is considered as the most suitable candidate to tackle such multi-parameter optimisation problem. At the end of the twentieth century, some research studies began to challenge the interpretability issues besides the accuracy one in the automatic fuzzy system design [95], [96], [97]. The most notable of these issues are complexity-based interpretability and semantic-based interpretability. The approaches dealing with complexity-based interpretability issue are devoted for decreasing the complexity of designed linguistic fuzzy system through the reduction of the number of variables, the number of fuzzy sets, the number of fuzzy rules, the number of premises, the shape of MFs, etc. The semantic-based interpretability issue is associated with some properties such as completeness, distinguishability, consistency of the FRB, the number of the fuzzy rules fired at the same time, etc. The crucial interpretability issue faced in fuzzy control design is the semantic-based interpretability, because in practical fuzzy control applications the number of inputs, fuzzy sets and conditions in the antecedent part of the rule is usually quite small. The semantic-based interpretability is taken into account by the existing evolutionary-based approaches in two different ways. Some research works considered measures quantifying the related interpretability aspects such as distinguishability, completeness, etc, and use them as additional objective to be maximized or minimized [98], [99], [100]. The search problem in this case becomes multiobjective. Other works imposed constraints on the components of the fuzzy knowledge base [101], [102],[103]. This fact means that the potential FKBs that do not verify the constraints are discarded, repaired or a penalty value is associated to them.

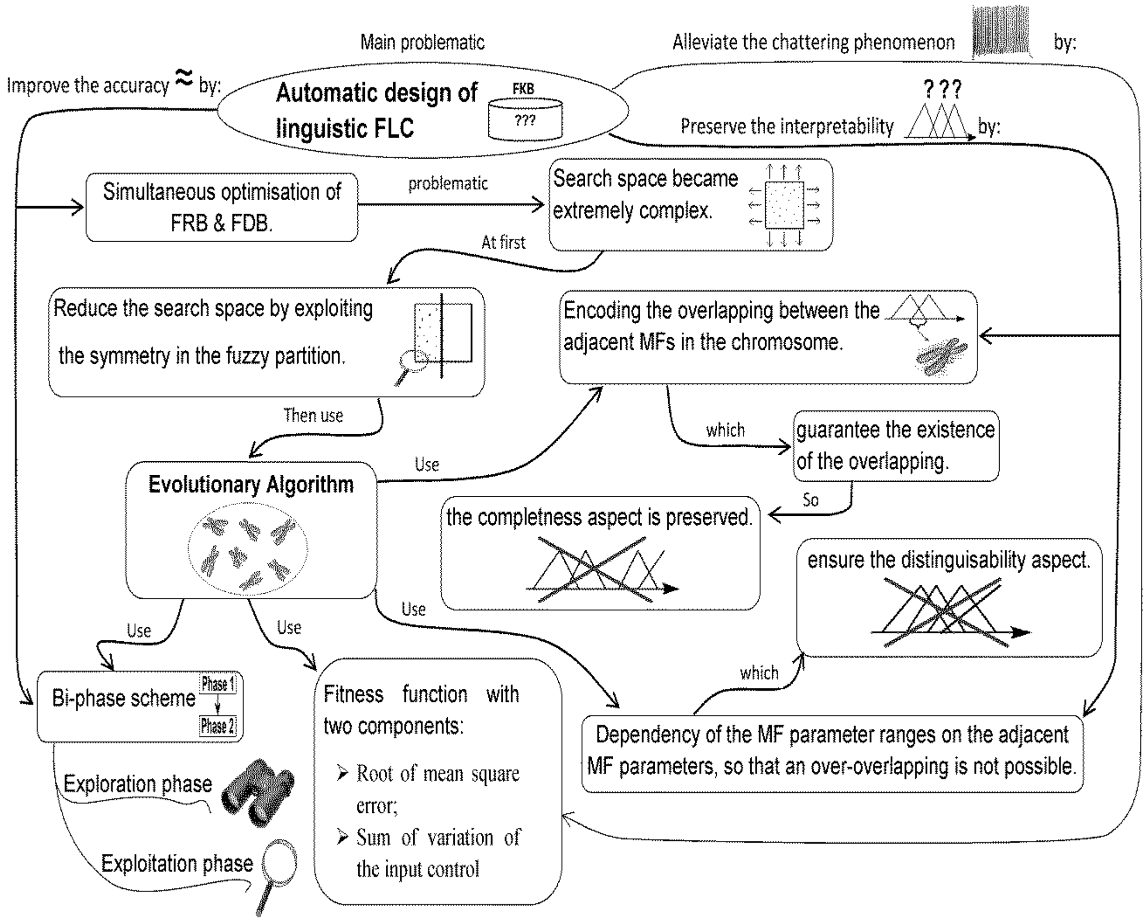


Fig. III.1Graphical representation of research motivations and contributions.

In this chapter, we investigate the use of integer-coded IEA to simultaneously optimize the FRB and the FDB of a Mamdani FLC. Our basic research motivations and contribution are summarized in Fig. III.1. The integer coding is used because it has the advantage in reducing the Hamming Cliff effects associated with binary coding and accelerates the convergence, since the length of the chromosome is further reduced compared to the binary one.

In practice, the evolutionary designed FLCs are not involved directly in the control process and their remarkable potentials are far from being fully exploited. This is due in large part to the chattering phenomenon that can damage the controlled plants. This problem can be avoided by a suitable FLC design. The basic idea of taking into account the chattering phenomenon during the optimization process is the introduction of the sum of variation of the control signal as optimization criterion. Doing so will ensure that the designed FLC provides just enough voltage to get the control job accomplished.

The concept of a bi-phase scheme for IEA is introduced to improve the accuracy performance of the evolved FLC. It consists of an exploration phase and an exploitation phase. In the exploration phase, the standard genetic process is performed to explore globally the overall search space. The IEA in the exploitation phase performs exploitation of favourable regions of the search space around the neighbourhood of the near optimum solution found by the former phase.

In the FDB, the triangular and symmetric MFs, which is the most used shape in control applications, is used. Concerning the fuzzy partition interpretability, the proposed evolutionary design technique distinguishes itself from previous works in its encoding strategy where overlappings between the adjacent MFs are coded in the chromosome and evolved by the bi-phase IEA. Doing so, the completeness aspect is guaranteed, and there is no need for measuring it and using the multiobjective search. Furthermore, all the searching ranges of the MF parameters are dependent on the adjacent MF parameters. This gives the bi-phase IEA the ability to evolve only valid distinguishable fuzzy partitions.

III.2 DIRECT-DRIVE DC MOTOR

The system to be controlled is a direct-drive DC motor. The main characteristic of this type of motors is that the load is directly driven without motion transfer mechanism such as belt, chain, ball screw or gearbox. In fact, the motion transfer mechanisms are known to be the source of some undesirable nonlinear effects such as vibration, friction, backlash, and elasticity. Direct drive motor, however, need a more precise controller. This is due to its significant sensitivity to any low variation in load parameters or external disturbances since they are directly reflected on the motor dynamic. The dynamic equations of the used direct-drive DC motor are given by:

$$E_a = R_a . I_a + L_a . \frac{dI_a}{dt} + K_e . \dot{q} \tag{III.1}$$

$$T_m = K_T . I_a \tag{III.2}$$

$$I_n . \ddot{q} = T_m - D . \dot{q} - T_l \tag{III.3}$$

Where q, $\dot{q}$, and $\ddot{q}$ denotes the angular position, angular velocity and angular acceleration of the motor shaft. E_a the input voltage, I_a the armature current, T_m the generated torque, and T_l the load torque. The other parameters and their numerical values are given on Table. III.1

Parameter	Notation	Value	Unit
Rated input voltage	E_{ar}	24	V
Rated output power	P_r	17	W
Rated output torque	T_{mr}	5.29	N.m
Viscous friction constant	D	1.74	N.m.s/rad
Motor inertia moment	I_n	0.0974	N.m.s^2/rad
Torque constant	K_T	0.54	N. m/A
Voltage constant	K_e	5.44	V/rad/sec
Armature resistance	R_a	2.8	Ω
Armature inductance	L_a	1.1	mH

Table. III.1: Electrical and mechanical parameters of the direct-drive DC motor.

III.3 MAMDANI FUZZY CONTROLLER TO BE EVOLVED

As many FLCs set to work nowadays, we have chosen the inputs of our FLCs to be the error (x_1) and the change error (x_2) on the angular position of the motor shaft. At the output, the FLC provides the input voltage (E_a) that excites the DC motor and brings it in the desired angular position. This choice makes the FLC to be evolved by the proposed IEA a PD-like fuzzy controller, which is the most suitable in direct-drive DC motor. This is due to its fast response and its ability to predict the future error of the actuator response.

The FLC used in our application can be viewed as a mapping from crisp inputs $\underline{x} = (x_1,x_2)^T \in U \subset IR^2$ to crisp output $y=E_a \in V \subset IR$, and this mapping can be expressed quantitatively as $y=f(\underline{x})$ where f is non-linear. Let the universe of discourse be $U=U_1 \times U_2$, where $U_1=U_2=[U_{min}, U_{max}]= [-0.05, 0.05]$, and V=$[-24, 24]$.

The structure of the used FLC, which is illustrated in Fig. III.2, consists of the following components:

A ***singleton fuzzifier*** converts a crisp value $\underline{x} \in U$ into a fuzzy singleton A_x within U.

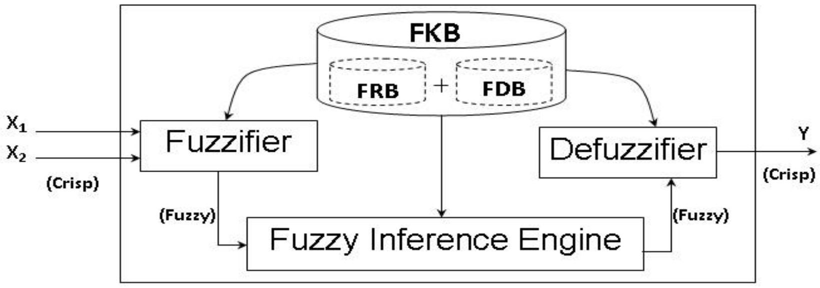


Fig. III.2 Structure of the fuzzy logic control.

The ***fuzzy data base***: The space of x_1 is partitioned into three triangular and symmetric membership functions associated to the following labels: negative (N), zero (Z) and positive (P). The space of the second input x_2 and the output y are partitioned into seven membership functions associated to the following labels: negative big (NB), negative medium (NM), negative small (NS), zero (Z), positive big (PB), positive medium (PM), and positive small (PS).

The ***fuzzy rule base*** consists of a collection of fuzzy IF-THEN rules expressed as:

$$R^{(l)}:\ IF\ (x_1\ is\ A_1^l\ and\ x_2\ is\ A_2^l\ )\ THEN\ (\ v\ is\ C^l\ ) \qquad \text{(III.4)}$$

Where A_i^l and C^l are terms associated to the fuzzy sets F_i^l and G^l defined in U_i and V, respectively, with $l = 1,2,.....,M$. M is the number of rules in the FRB. Here we have chosen $M = 3x7 = 21$ to account for every possible combination of input fuzzy sets.

Each fuzzy IF-THEN rule defines a fuzzy implication:

$$R^l = F_1^l\ x\ F_2^l \rightarrow G^l \qquad \text{(III.5)}$$

$$R^l = \{\ ((\underline{u},v),\ \mu_R^l(\underline{u},v))\ \mid\ \underline{u} \in U, v \in V\ \} \qquad \text{(III.6)}$$

Where $\mu_{R^l}(\underline{u},v)$ is defined by the following Larsen's fuzzy implication rule:

$$\mu_{R^l}(\underline{u},v) = \mu_{F_1^l x F_2^l}(\underline{u})\ .\ \mu_{G^l}(v) \qquad \text{(III.7)}$$

$$\mu_{R^l}(\underline{u},v) = (\ \mu_{F_1^l}(u_1)\ .\ \mu_{F_2^l}(u_2)\ )\ .\ \mu_{G^l}(v) \qquad \text{(III.8)}$$

The ***fuzzy inference engine*** derives from each fuzzy rule of the FRB an output fuzzy set, in the following way:

Each fuzzy rule of (IV.6), described by a fuzzy implication R^l, determines a fuzzy set $B^l = Ax\ o\ R^l$ in V such that:

$$\mu_{B^l}(v)=\mu_{Ax\,o\,R^l}(v) \tag{III.9}$$

$$\mu_{B^l}(v)=\max_{\underline{u}\in U}\{\mu_{Ax}(\underline{u})\cdot\mu_{R^l}(\underline{u},v)\} \tag{III.10}$$

The ***defuzzifier*** used in our fuzzy controller is the modified height defuzzifier. Let v^l denote the center of the fuzzy set B^l, which is associated with the activation of the l^{th} fuzzy rule. This defuzzifier evaluates $\mu_{B^l}(v^l)$ at v^l, and then computes the output of the FLC as:

$$y=\frac{\sum_{l=1}^{M} v^l\,\frac{\mu_{B^l}(v^l)}{\delta^l}}{\sum_{l=1}^{M}\frac{\mu_{B^l}(v^l)}{\delta^l}} \tag{III.11}$$

Where δ^l is the support's length of the triangular membership function of the consequent for the l^{th} fuzzy rule.

With this components, the FLC is called "*fuzzy system as expansion of FBF: Fuzzy Basis Function*" [104] .

III.4 MAMDANI FLC PARAMETERS TO BE EVOLVED

To use the IEA, it is vital to define first the parameters to be optimized and then code it as some finite-length strings or chromosomes "*Ch*". Two elements must be optimized for the fuzzy controller: the FRB and the FDB.

The FRB part of the chromosome involves the consequent labels (linguistic terms) of the fuzzy rules. The labels associated to the output fuzzy sets from NB to PB used in the consequent part of the fuzzy rules are coded by integers from 1 to 7.

The FDB part of the chromosome contains the descriptive parameter set of the input/output MFs. In the fuzzy system applications, it is used to define the MFs as separate functions. In this work, we propose to define the MFs with respect to their adjacent MFs. The relationship between the adjacent MFs is measured by the overlapping. There are four possible overlapping situations for adjacent MFs: over-overlapping MFs, fully-overlapping MFs, partially-overlapping MFs, and non-overlapping MFs as depicted in Fig. III.3. The fully-overlapping MFs are also called complementary MFs. The overlapping percentage for the partially-overlapping MFs must be at least 10% of the full overlapping. The distinguishability and completeness aspects of the different situations are also depicted in

Fig. III.3. The over-overlapping MFs are not distinguishable and satisfy the completeness aspect. On the contrary, the non-overlapping MFs are distinguishable and do not satisfy the completeness aspect. These facts made both of them not interpretable. However, the remaining situations, i.e., the fully-overlapping MFs and the partially-overlapping MFs are at the same time distinguishable and satisfy the completeness aspect, which means that both overlapping MFs situations form interpretable fuzzy partitions. So, to preserve the fuzzy partition interpretability, the adjacent MFs must be either fully-overlapping or partially-overlapping. Most of the evolutionary design methods in the literature restrict adjacent MF to fully overlap, because allowing partial overlappings during evolution requires the test of the existence of an overlapping between the adjacent MFs in all the chromosomes. In our design method, the overlappings are coded in the chromosome and evolved by the IEA. This fact enforces the partial overlapping between the MFs, and the unfeasible chromosomes are avoided during the crossover and mutation operations. Since the shape of the MFs is assumed to be triangular and symmetric, then we need only two parameters for its description. These parameters are elements of {center (C), width (W), overlapping (O)}. It is obvious that the MFs located at the extremes are defined by the center and the width; while the parameters of the others are the center and the overlapping.

III.5 IEA FOR MAMDANI FLC DESIGN

Fig. III.4 shows the structure of the evolutionary fuzzy control system for DC motor that includes the IEA, and a FLC. By genetic operators, the IEA creates character-strings or chromosomes, whose characters termed alleles are integer values. Every such "genotype" defines the FKB which is used by the FLC to track the desired trajectory and at the same time to calculate the fitness function that measures its performance.

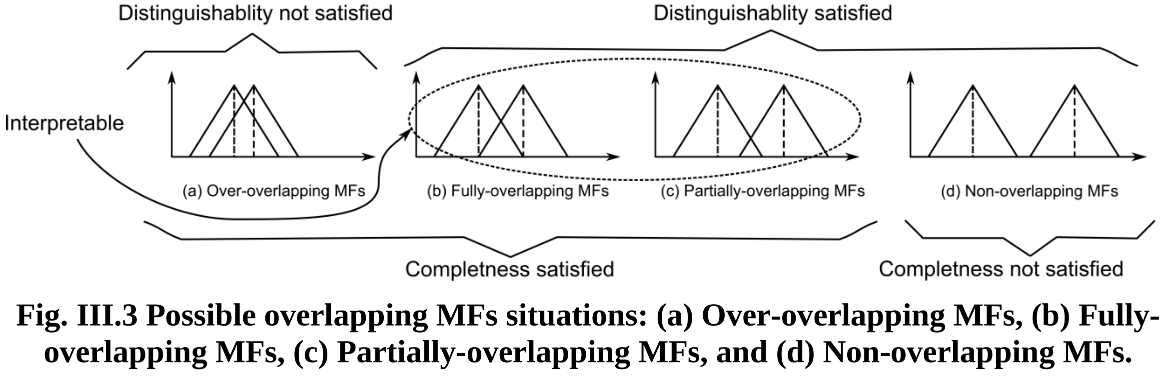


Fig. III.3 Possible overlapping MFs situations: (a) Over-overlapping MFs, (b) Fully-overlapping MFs, (c) Partially-overlapping MFs, and (d) Non-overlapping MFs.

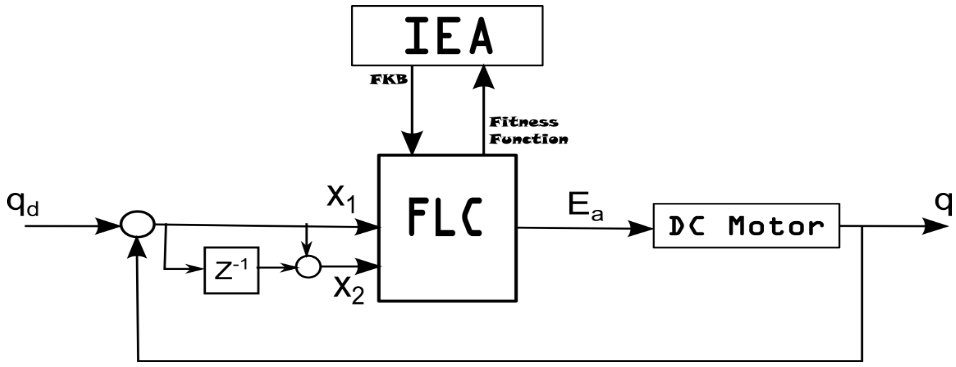


Fig. III.4 Configuration of the evolutionary fuzzy control system for direct-drive DC motor.

III.5.1 Reduction strategy of the chromosome size

Usually EAs are initialized randomly, but if we want to incorporate some knowledge about the problem, we need to introduce that in the initial population of the EA. The most fuzzy controllers set in real world applications are basically characterized by the following properties:

- The fuzzy partitions along the universe of discourse for the input and output variables are symmetric;
- If the inputs are zero, the output should be zero too;
- If the inputs of the two fuzzy rules are symmetric, the outputs of these rules should also be symmetric.

In our work, these properties constitute the implicit knowledge about the motor FLC design to be incorporated. Instead of introducing them in the initial population, we propose to make use of them in reducing the chromosome size and so the convergence time.

Using the first piece of knowledge about the symmetrical aspect of the fuzzy partitions, just the MFs located in either the positive or negative part of the universe of discourse and the MF centred at zero need to be coded in the chromosome, Fig. III.5. Furthermore, it is obvious that the MF associated to the zero term for each variable must have the center fixed at zero.

The second knowledge gives already one fuzzy rule -if x_1 is *Z* and x_2 is *Z* then E_a is *Z*- which must be discarded from evolution. So there's no need to encode it in the chromosome. The last fact implies that we have to search only the half of the FRB and then deduce the other half by symmetry, Fig. III.6.

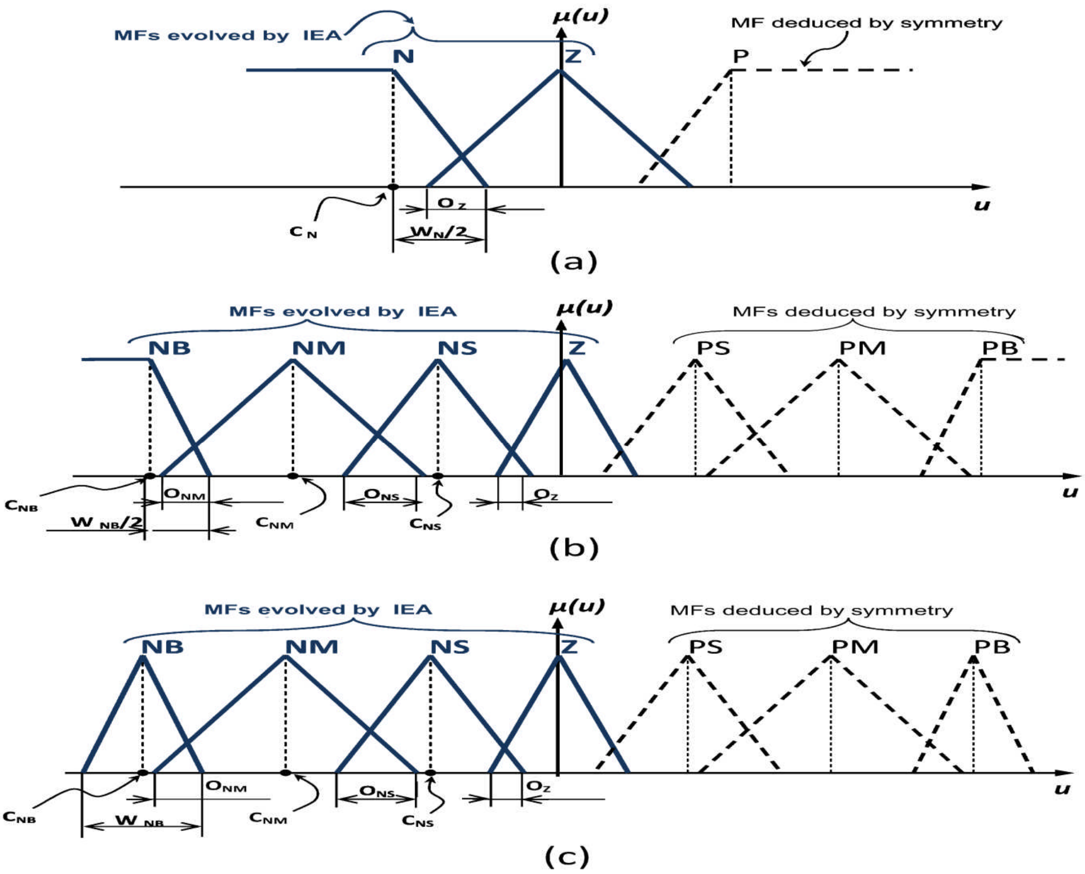


Fig. III.5 FDB parameters to be evolved by IEA

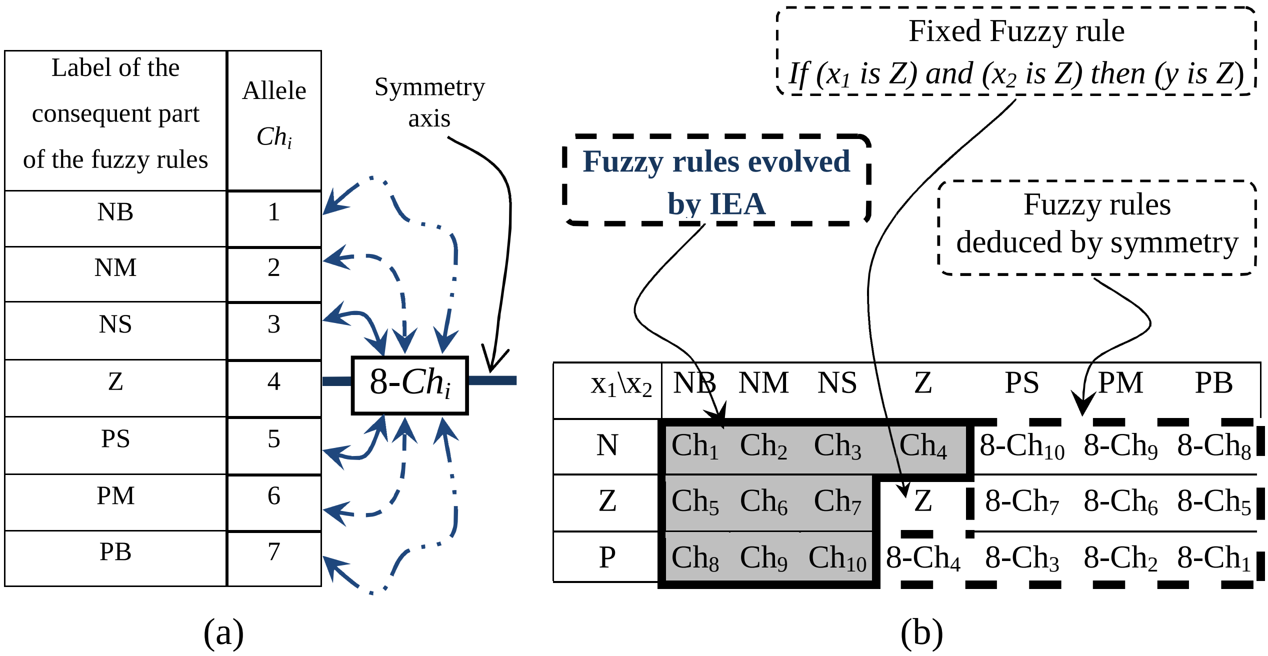


Fig. III.6 (a) Symmetry mechanism of labels in the consequent part of fuzzy rules, (b) FRB parameters to be evolved by IEA.

III.5.2 Genotype

With a multi-parameter, concatenated and integer encoding, the FKB parameters described above are coded on the same chromosome "*Ch*" of 44 genes, Fig. III.7. The first ten genes of the chromosome encode the FRB and take values from 1 to 7. The remaining 34 fragment genes are used to compute the MF parameters which form the FDB. Their values vary between 1 and 9.

Each MF parameter (X) is coded into two-integer subchromosome (Ch_i Chi_{+1}) representing a percentage of a specific range I_X. The general decoding relationship that calculates the numerical MF parameter from its representative genes and the corresponding searching range is given by:

$$X = \frac{Ch_i + 10.Ch_{i+1}}{100} I_x \qquad \text{(III.12)}$$

The possible percentage values are always between 11% and 99% of the searching range lengths. As one can see, the proposed encoding strategy avoids zero percentage to ensure that all the MFs are overlapped and distinguished. The searching range lengths (I_X) of the MF parameters are given in Table. III.2.

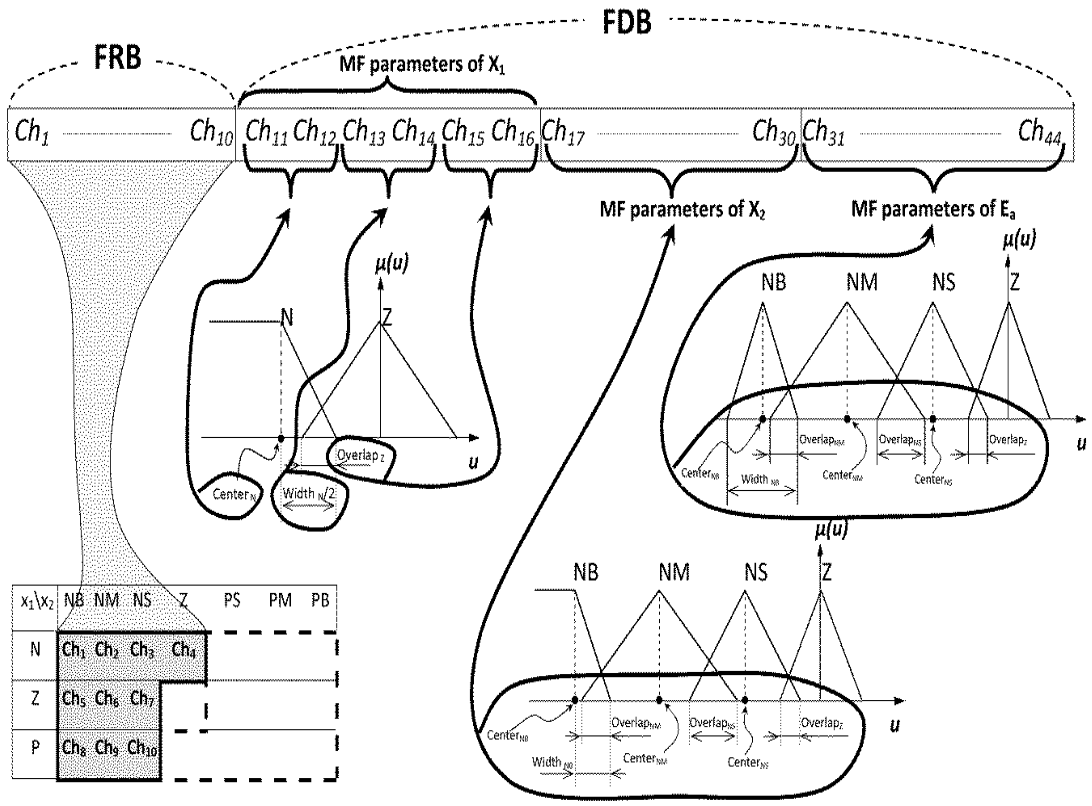


Fig. III.7 Schematic representation of the FKB parameters on a chromosome.

In the previous evolutionary design methods, it is used to optimise the position of the characteristic points the MFs within independent ranges fixed off-line. However, it is well known that these parameters are dependent among themselves and among those of the adjacent MFs. To take into account these parameter interdependency, the boundaries of the searching ranges of the MF parameters are dependent on the previously calculated parameters of the adjacent MF. They are computed during the evolution in particular during the decoding process. A typical example of FDB decoding process and the representation of the searching ranges of the MFs parameters are represented on Fig. III.8 and Fig. II.9. Obviously, every searching range depends on one or two of the previous adjacent MF parameters. The resulting fuzzy partition is subsequently always valid.

III.5.3 Chromosome Initialization

A hybrid chromosome initialization is adopted in this algorithm. Specifically, the MF part of the chromosome is generated randomly within the corresponding ranges. For the FRB, we used the MacVicar-Whelan rule base model. This later is suitable for motor drive applications and more appropriate for PD-like FLCs as it is the case with our work. A detailed description of the FRB generation process according to MacVicar Whelan approach is given in [27], [28].

III.5.4 Evolutionary Operators

Our algorithm uses roulette wheel selection with replacement to select parents for reproduction. The crossover operator is two-point crossover which refers to selecting randomly two sites on one of the chromosomes. Then, the fragment between the two sites is exchanged with the corresponding fragment of a second chromosome. As mentioned in the above section, the chromosome is integer based instead of binary based and each allele of this chromosome has an integer range according to the FLC parameter it represents. For example, alleles representing FRB have an integer range from 1 to 7, and those encoding the MF parameters have an integer range from 1 to 9. The mutation operator thus changes the allele randomly inside its range.

Variable	Parameter (X)	Searching range length(I_X)
Input (x_1)	C_N	U_{min}
	W_N	$- 2*C_N$
	O_Z	W_N
Input (x_2) and output (E_a)	C_{NB}	U_{min}
	C_{NM}	$C_{NB}/2$
	W_{NB}	$2*(C_{NM}- C_{NB})$
	O_{NM}	$W_{NB}/2$
	O_{NS}	W_{NM}
	C_{NS}	$(C_{NM}+W_{NM}-O_{NS})/2$
	O_Z	W_{NS}

Table. III.2: Searching range length (I_X) for MF parameters.

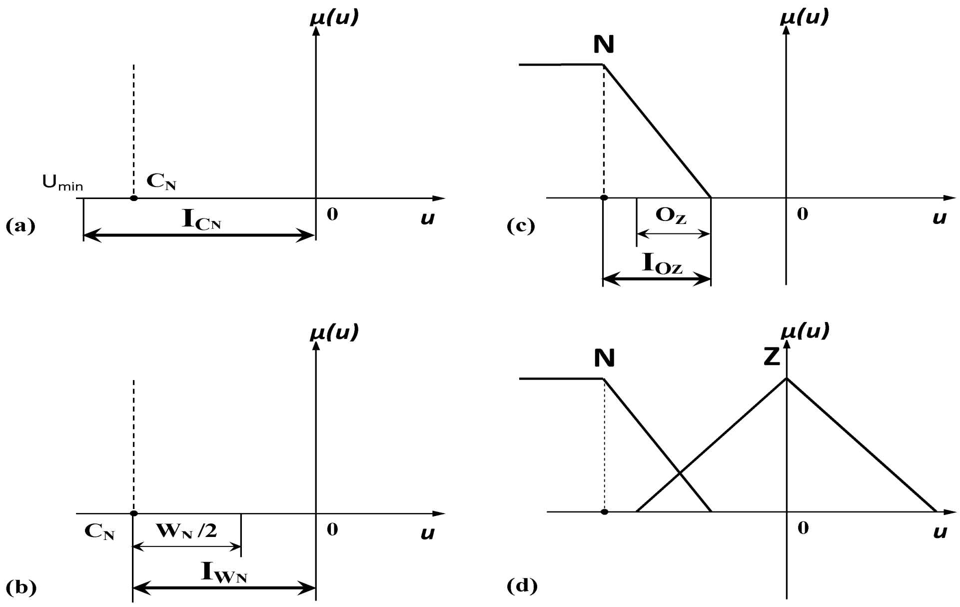


Fig. III.8 Example of FDB decoding process and representation of searching ranges of the MF parameters for the input x_1. (a) CenterNB, (b) CenterNM, (c) widthNB, (d) OverlapNM, (e) OverlapNS, (f) CenterNS, (g) overlapZ, (h) Resulting fuzzy partition.

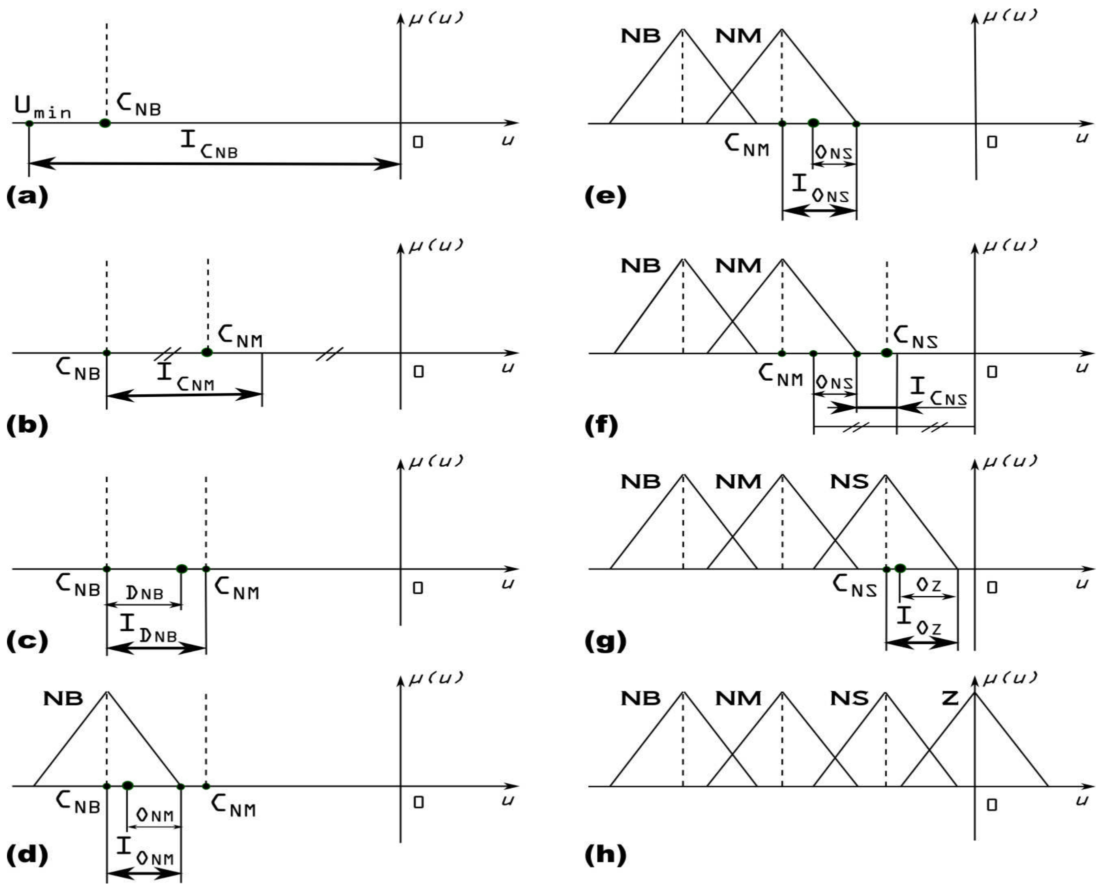


Fig. III.9 Example of FDB decoding process and representation of searching ranges of the membership function parameters for the input x_2 and the output Ea. (a) Center$_N$, (b) Width$_N$, (c) Overlap$_Z$, (d) Resulting fuzzy partition.

III.5.5 Fitness function

The IEA requires that each chromosome of the population be assigned a fitness function value. This value reflects the extent to which the FKB represented in the chromosome produces the expected FLC behaviour over the reference signal. In particular, we seek Mamdani FLC that has a good trajectory tracking and smooth behaviour in control action. That is why the fitness function is chosen to have two components:

- Root of mean square error (*RMSE*) representing the accuracy objective defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(q_i - q_{d_i})}{N}} \qquad \text{(III.13)}$$

Where $q_i$and q_{d_i} are the actual and the desired angular position, respectively, at the ith sampling time. N is the sampling size.

Sum of variation of the input voltage variable ($\mathrm{Sum}|\Delta \mathrm{E_a}|$) representing the smoothness objective defined as:

$$Sum|\Delta E_a| = \sum_{i=1}^{N} |E_{a_i} - E_{a_{i-1}}| \qquad \text{(III.14)}$$

Where $\mathrm{E_{a_i}}$ is the input voltage value at the ith sampling time.

These measures are weighted and summed up so that they form a final quality value:

$$c_1.RMSE + c_2.Sum|\Delta E_a|$$

The parameters c_1 and c_2 are weights used to stress the relative importance of the different fitness function components. Currently, there is no systematic method available at the time for identifying these weights. Usually the empirical methods are used or optimized in the same time as the design parameters. Since our problem has only two objectives, it seems feasible to determine the weights by trial and error. The numerical values used are c_1=1 and c_2= 10^{-7}.

III.5.6 Bi-Phase Scheme

EA is a stochastic search method based on exploration search strategy and exploitation search strategy. The exploration strategy performs a random search without use of any information about the problem domain. The exploitation strategy is a search strategy guided along the generations with the best search direction found so far. The evolutionary operators responsible of the exploration and exploitation are the selection, mutation, and crossover operators. Some studies [105], [106] and [107] tend to suggest that none of these operators is exclusively an exploitation or exploration operator. Furthermore, even the control parameter settings contribute in affecting the exploratory power of the EA.

With an appropriate choice, found by trial and error, of the evolutionary operators and control parameter settings, EA can stress the exploration/exploitation balance towards one strategy or another.

The proposed bi-phase IEA consists of an exploration phase and an exploitation phase. In the exploration phase, the initial population is generated randomly and the standard genetic process is performed to explore globally the search space.

To increase the exploration power in this phase, we have chosen the roulette wheel selection operator known for its high selection pressure. This fact gives to the crossover operator enough time to properly recombine the individuals before the convergence of the population to the near optimal solution.

The factors used to promote the exploitation in the second phase are the use of creep mutation and elitism strategy while disposing of the crossover and mutation operators. The creep mutation in integer representation alters a single allele, but in small increments. In this work, the creep mutation increments or decrements by 1 the allele within the corresponding range. The choice between the incrementation and the decrementation is done randomly. This creep mutation operator is technically the responsible for the exploitation by shifting the mutated chromosome to its vicinity region.

In the exploitation phase, the initial population is generated by creep mutating the best chromosome obtained from the exploration phase, Fig. III.10. If a new best chromosome is found in the newly formed population, we reinitialize the population with the same manner as described above but using the new best chromosome. We repeat this process until the termination criterion, which is a specific maximum number of generations, is satisfied. This phase acts as a hill-climbing search method by looking at the best chromosome vicinity through the decrementation and incrementation of one allele at a time. Such phase can be referred as focusing phase.

III.6 SIMULATION RESULTS

III.6.1 Design of chattering-free Mamdani FLC by a mono-phase IEA

In this section, we investigate the mono-phase IEA (i.e., IEA with exploration phase) in chattering-free Mamdani FLC design for tracking control of direct-drive DC motor. The goals of the simulations are: (1) to reveal the influence of taking into account the objective of smoothness besides the accuracy objective; (2) to show that the proposed IEA can design chattering-free Mamdani FLC effectively; (3) to compare the tracking and robustness performances of the designed chattering-free Mamdani FLC with the conventional PD controller.

Fig. III.11 shows the overall structure of the mono-phase integer evolutionary fuzzy control system flow used to automatically generate FKB for chattering-free Mamdani FLC. Starting from random initialization of the chromosome population, the mono-phase IEA decode the chromosomes into potential FKBs. Mamdani FLC use each of these decoded FKBs to track the desired trajectory and in the same time to compute the fitness function value that measures the tracking performance end the variance of the control signal. Based on these fitness function values, chromosomes are selected by roulette wheel selection operator to be mutated or recombined by integer mutation operator and 2-point crossover operator, respectively. The new resulting chromosomes are evaluated and the evolutionary process repeat until the satisfaction of the stopping criterion.

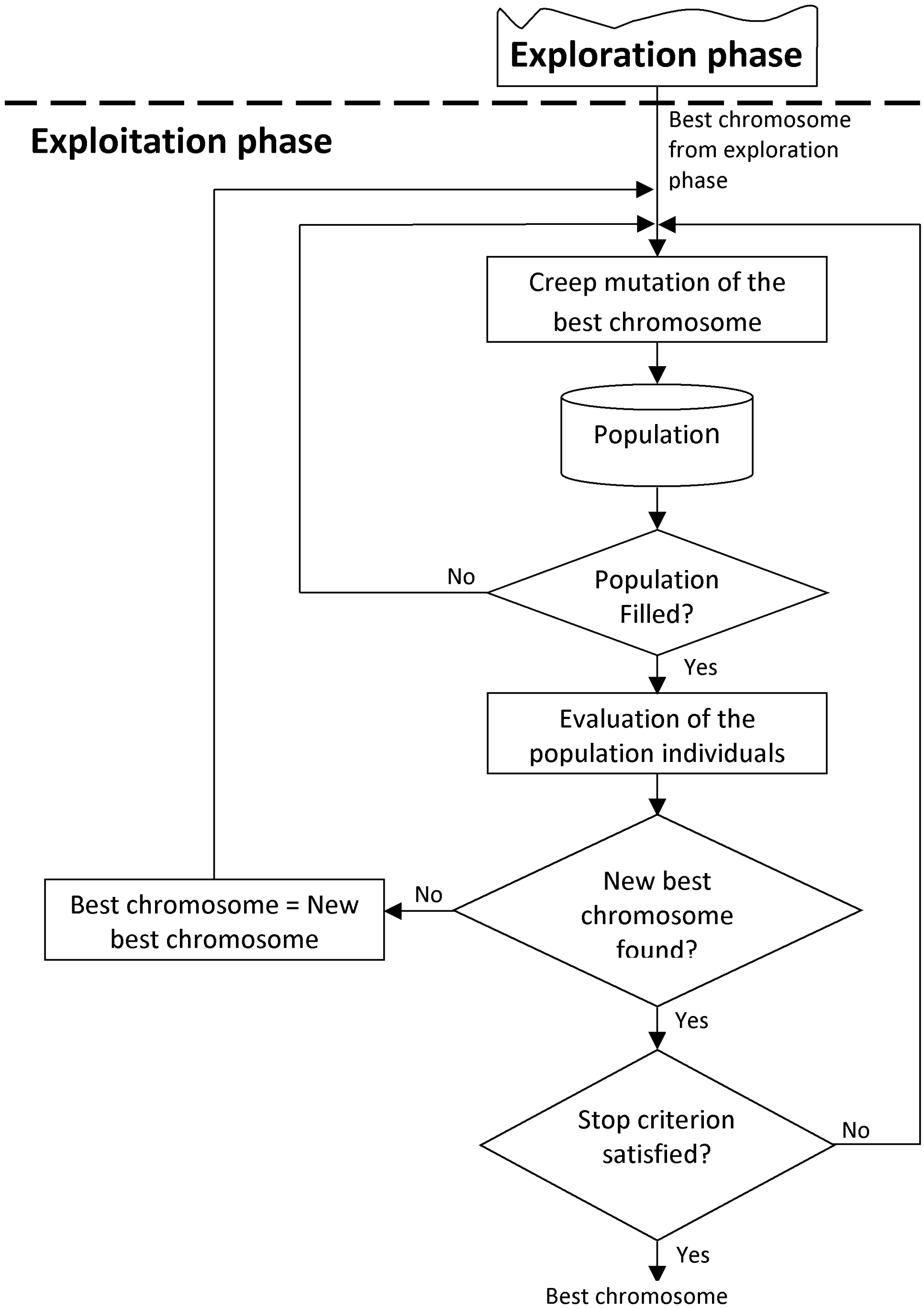


Fig. III.10 The proposed exploitation framework for FLC design.

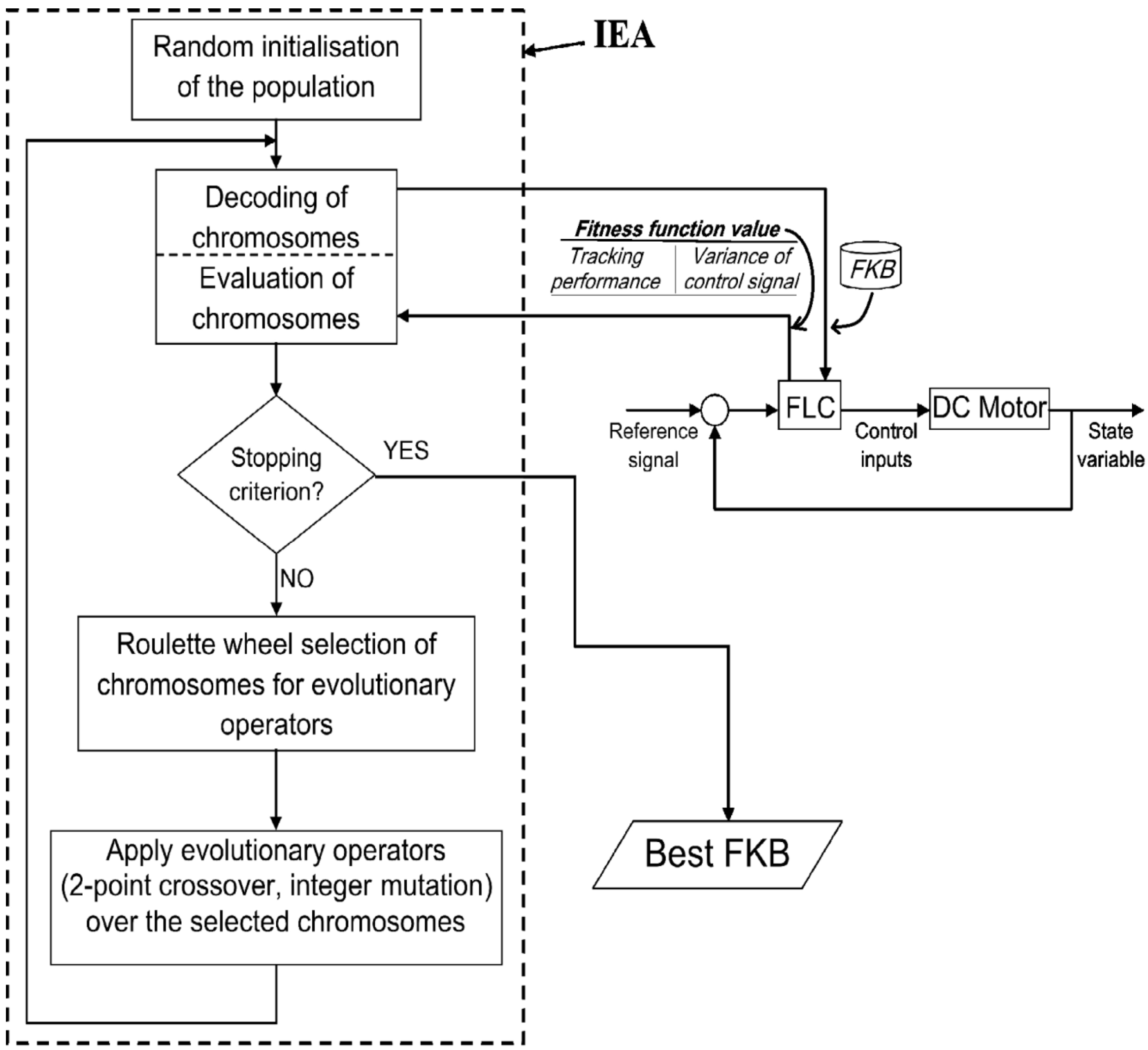


Fig. III.11: Mono-phase integer evolutionary fuzzy control system configuration.

The objective of chattering-free Mamdani FLC design is to make the direct-drive DC motor position tack a reference trajectory defined as:

$$q = 0.75\big(1 - cos(0.25.\pi.t)\big)\ [rad] \qquad \text{(III.15)}$$

The initial states are given by: $q = 0[rad]$, and $\dot{q} = 0\ [rad.s^{-1}]$.

The population size, the mutation rate and the crossover probability were set at 50, 0.1, and 0.8, respectively. Since IEA is stochastic algorithm, it is run ten times using different random number generator seeds producing in such a way different initial populations. The best FKB found by the IEA in each of the ten runs was recorded, and each of these runs was stopped after 100 fitness evaluations.

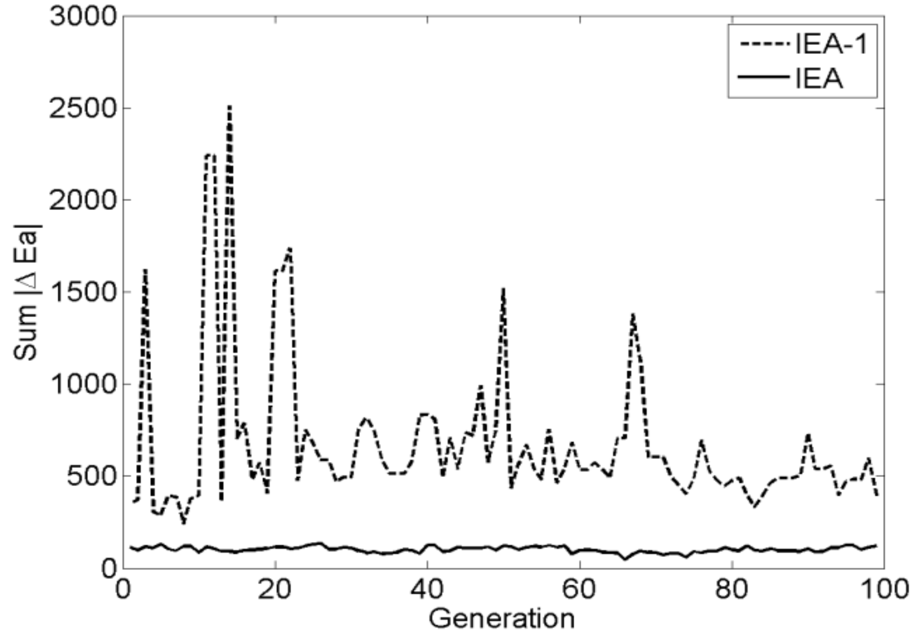


Fig. III.12 Evolution of the smoothness objective (Sum|ΔE_a|) during the design phase for IEA and IEA-1.

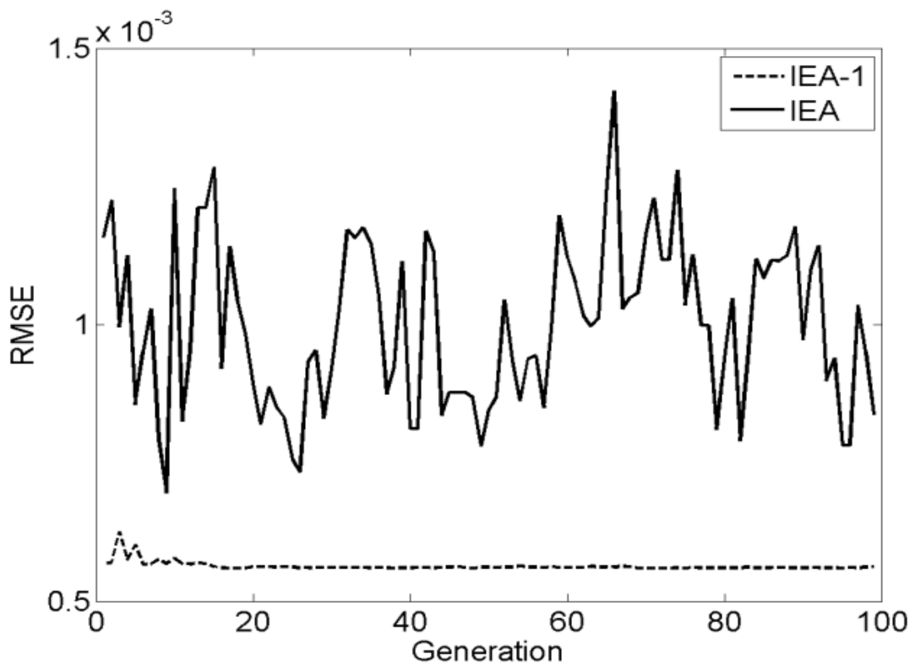


Fig. III.13 Evolution of the tracking accuracy objective (RMSE) during the design phase for IEA and IEA-1.

To investigate the impact of the introduction of the second objective in the design phase, we consider another EA noted as IEA-1 for comparison. IEA-1 is similar to the mono-phase IEA except that the fitness function to be minimized is equal to only the RMSE.

Fig. III.12 and Fig. III.13 show the evolution of RMSE and Sum$|\Delta E_a|$, respectively, for IEA and IEA-1 over the number of generations. Fig. III.12 demonstrates clearly that the IEA succeeds to minimize the Sum$|\Delta E_a|$ greatly compared to IEA-1. On the contrary, in Fig. III.13, it is the IEA-1 that has less RMSE than IEA. This leads to note that the RMSE and Sum$|\Delta E_a|$ are two concurrently objectives, i.e. the amelioration of one objective implies the deterioration of the other one. The IEA thus tends to optimize the FKB over the generations by finding a tradeoff between the two objectives: MSRE and Sum$|\Delta E_a|$.

In order to highlight the effectiveness of the evolved fuzzy controller by IEA, we compare its performances to Mamdani FLC designed by IEA-1 and a conventional PD control. The gains of PD controller are given as: $K_P = 400$; $K_D = 3$. They are determined according to the Ziegler-Nichols tuning method based on the step response of the plant.

The performances of the different controllers are compared for two cases:

- Nominal case: It is a disturbance-free case where the nominal model of the DC motor described in section 2 is used without inducing any disturbances.
- Disturbed case: To perform a qualitative assessment of the robustness of the designed FLC, the motor is supposed to be affected by two types of disturbances: load disturbance, and measurement noise.

The ***load disturbance*** models various external forces that affect the inertia during the interaction with the environment, e.g., forces due to material processing in tool machines or forces due to the impact, for example at spot welding. In the simulations the moment of inertia of the motor shaft is varied while the motor is in motion as:

- $t < 2$s, I_n=0.0974 N.m.s^2/rad (nominal value);
- 2< t <5s, In=0.2922 N.m.s^2/rad (three times of nominal value);
- 5< t < 6s, I_n=0.0974N.m.s^2/rad (reduced inertia to nominal value);
- 6< t < 8s, I_n=0.5844N.m.s^2/rad (six times of nominal inertia);
- 8< t <12s, I_n=0.0974 N.m.s^2/rad (reduced inertia to initial value).

The ***measurement noise*** is introduced in the output signals of the system model to simulate noise-corrupted sensors. It is modelled as zero mean White Gaussian noise with 0.01 deg standard deviation.

The control task is to control the angular position of the motor shaft to track the following trajectory:

$$q = 0.75\big(1 - cos(0.25.\pi.t)\big)\ [rad] \qquad \text{(III.16)}$$

The initial states are given by: $q = 0[rad]$, and $\dot{q} = 0\ [rad.s^{-1}]$.

The simulation results illustrating the tracking performance and control activities of the Mamdani FLC designed by IEA-1, the Mamdani FLC designed by IEA, and the conventional PD controller under the two cases are shown in Fig. III.14, Fig. III.15, Fig. III.16, Fig. III.17, Fig. III.18 and Fig. III.19, respectively.

According to Fig. III.14-(b), Fig. III.16-(b), and Fig. III.18-(b), the Mamdani FLC designed by IEA-1 yields the smallest tracking errors. After the disturbances are induced, Mamdani FLC designed by IEA shows the best tracking performance, while for Mamdani FLC designed by IEA-1 it is substantially deteriorated. The tracking errors for PD controller are still in acceptable tolerance.

As one can see in Fig. III.15-(a), Fig. III.17-(a), and Fig. III.19-(a), the effects of the added measurement noise are clearly evident in the input voltage signal for Mamdani FLC designed by IEA and PD controller, but there is no undesirable chattering phenomenon. Contrary to the Mamdani FLC designed by IEA-1 for which the chattering level is quite large.

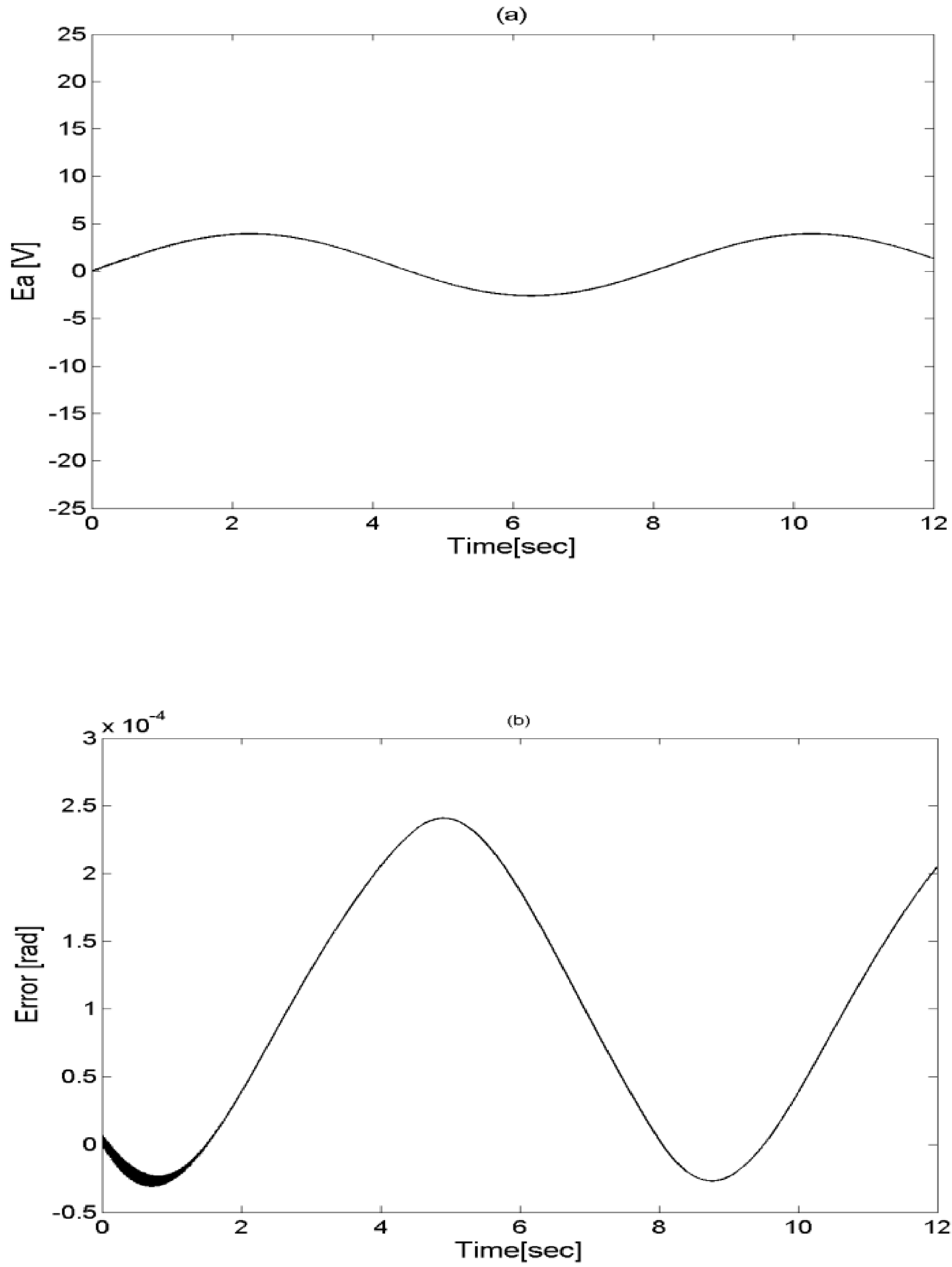
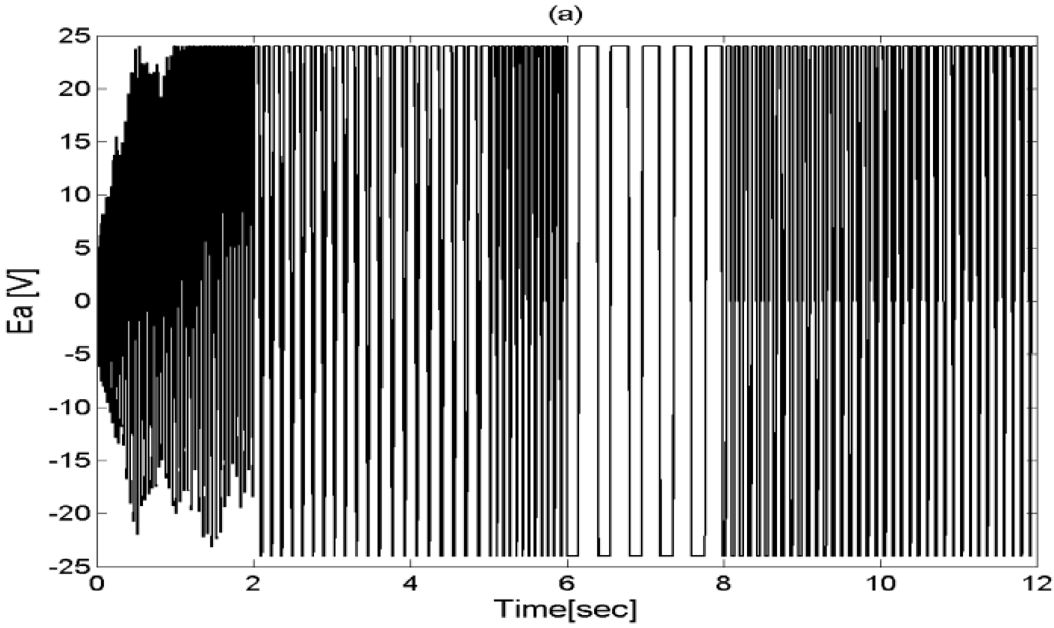


Fig. III.14. Tracking performances and control activities in nominal case of Mamdani FLC evolved by IEA-1.



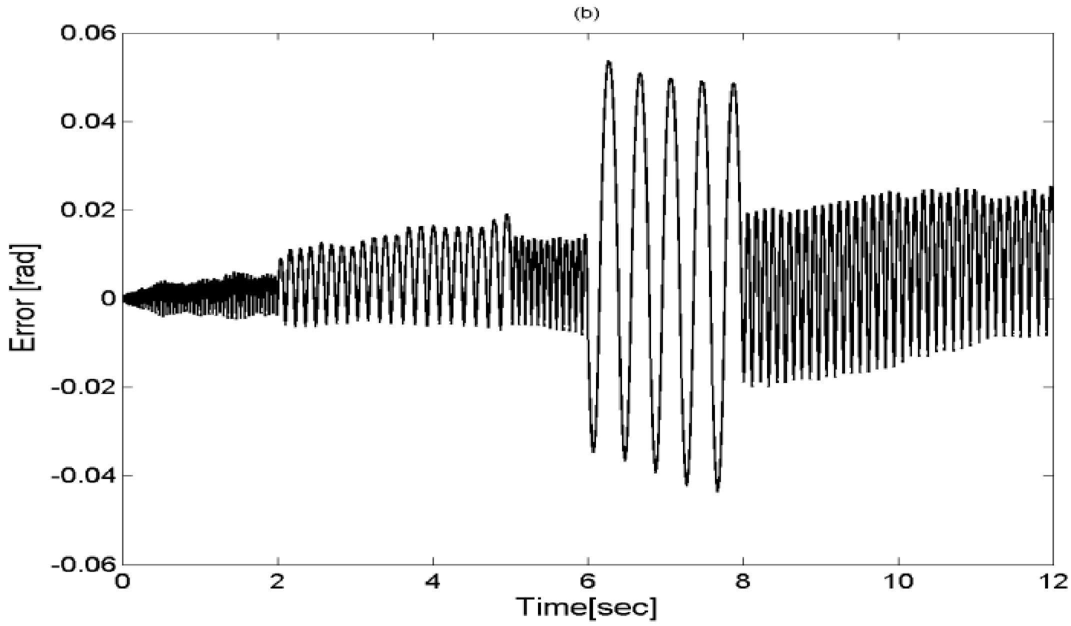


Fig. III.15. Tracking performances and control activities in disturbed case of Mamdani FLC evolved by IEA-1.

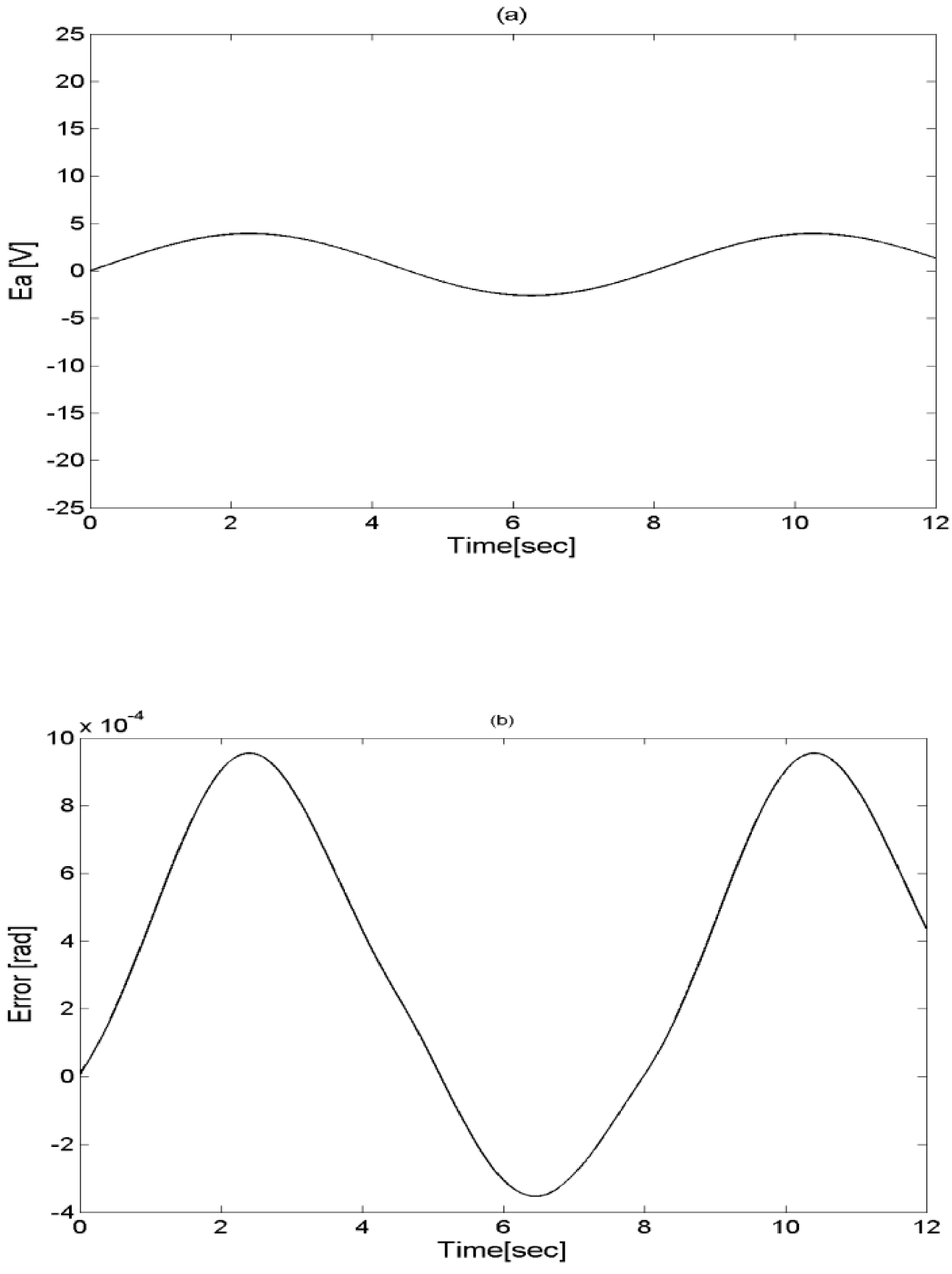
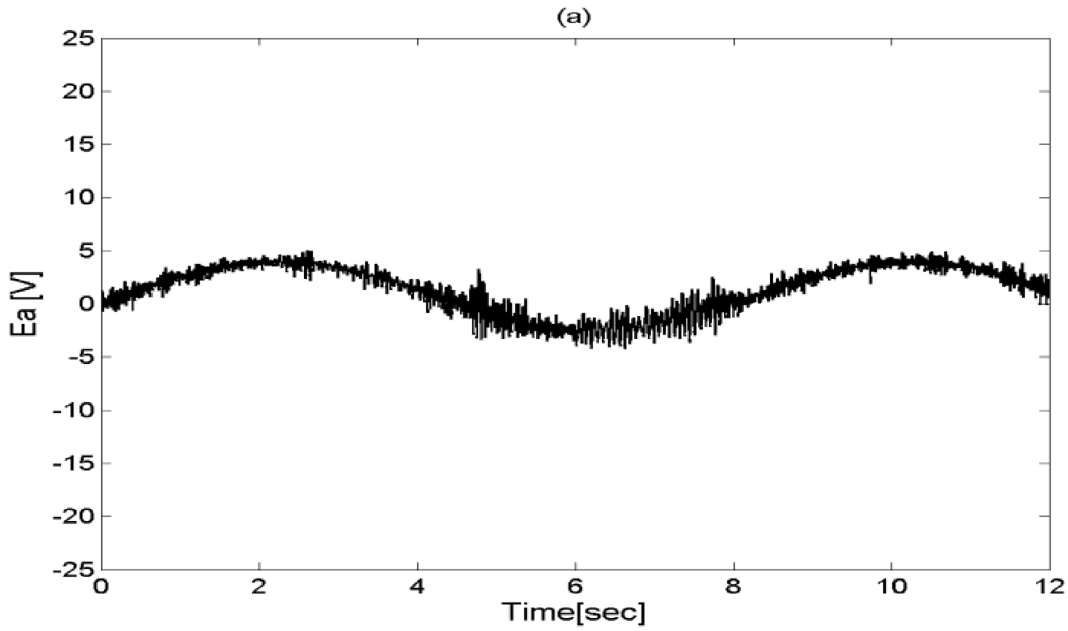


Fig. III.16. Tracking performances and control activities in nominal case of Mamdani FLC evolved by IEA.



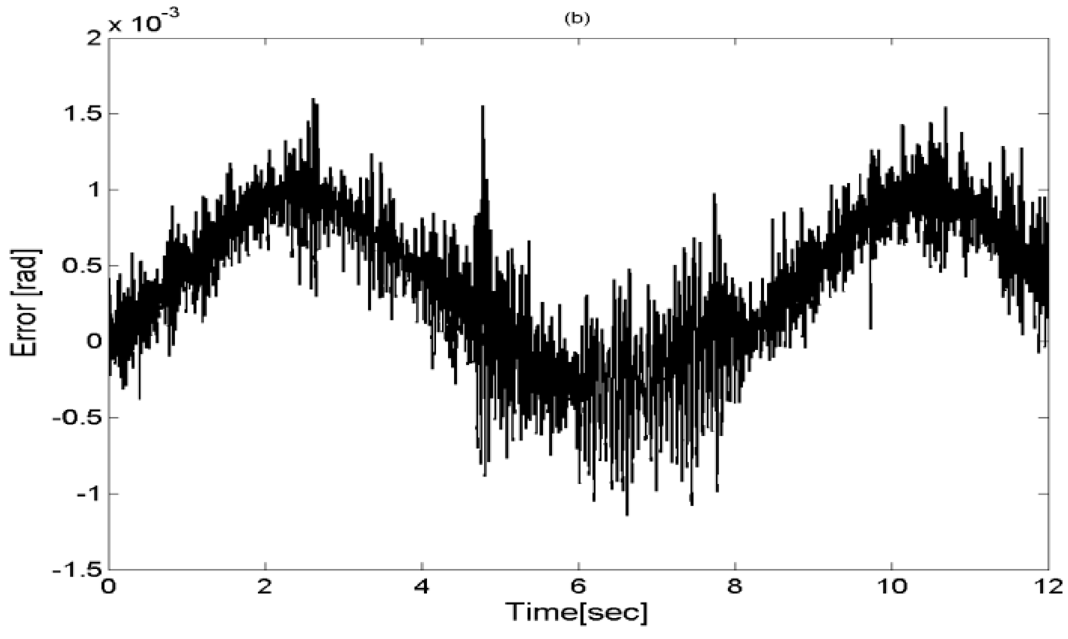


Fig. III.17. Tracking performances and control activities in disturbed case of Mamdani FLC evolved by IEA.

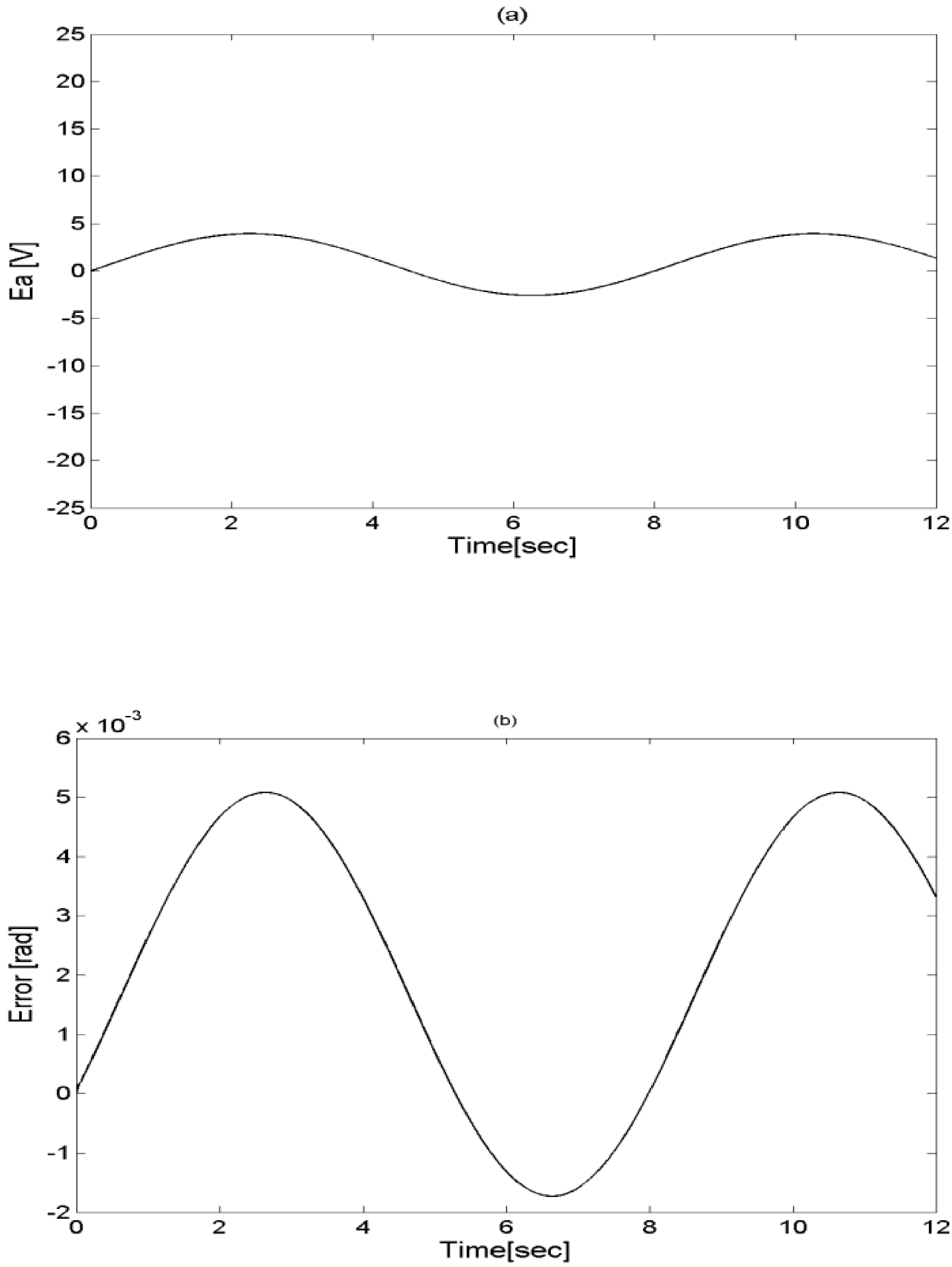
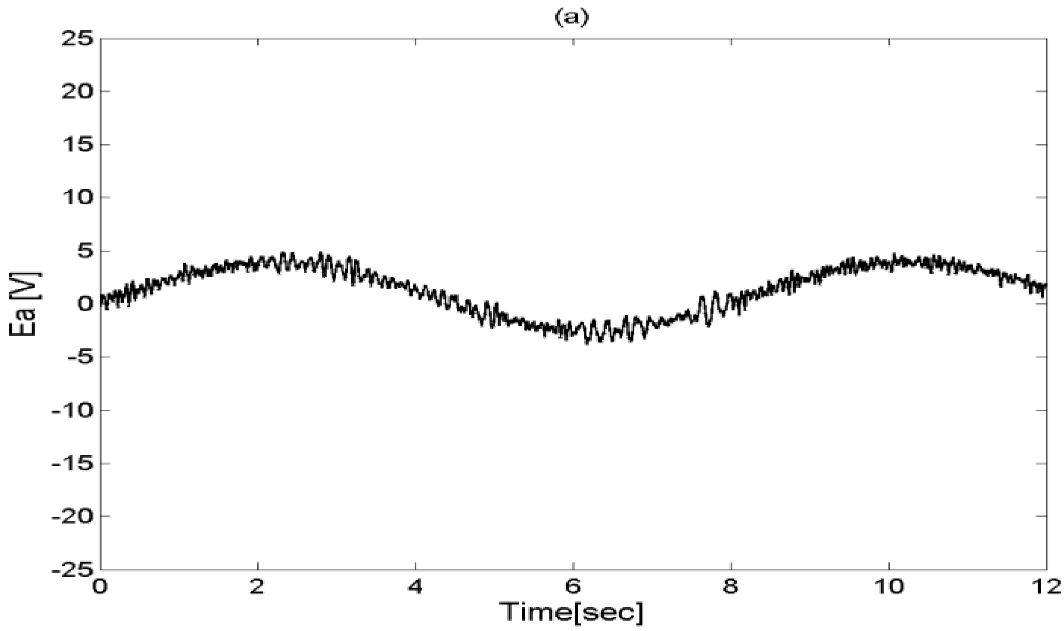


Fig. III.18. Tracking performances and control activities of PD controller in nominal case.



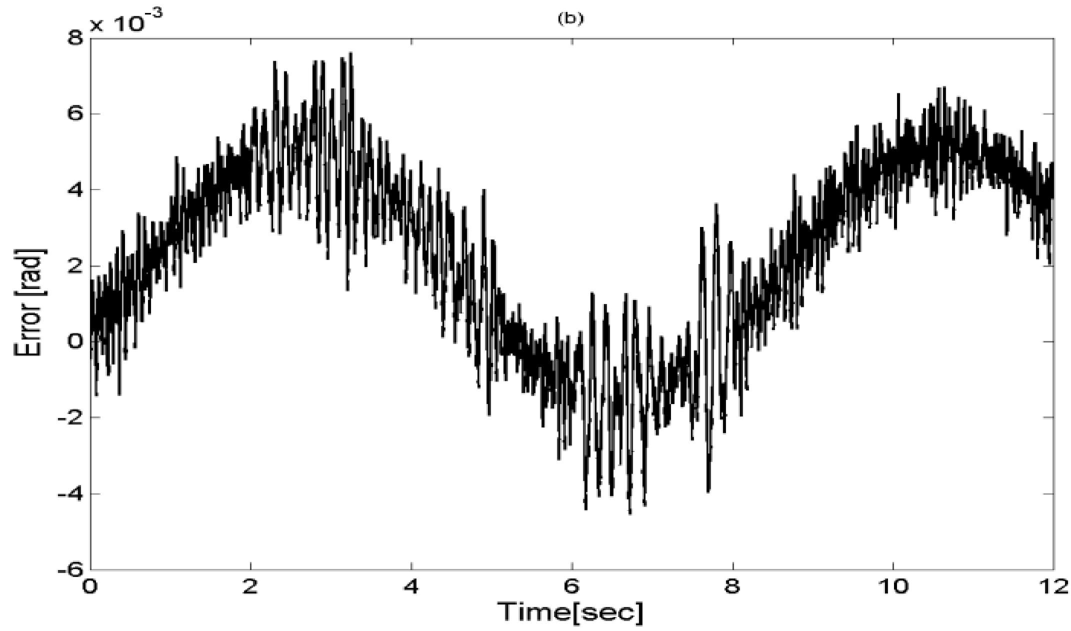


Fig. III.19. Tracking performances and control activities of PD controller in disturbed case.

III.6.2 Design of accurate Mamdani FLC by a bi-phase IEA

In this section, we investigate the bi-phase IEA in FLC design for tracking control of direct-drive DC motor.

The purpose of the simulations is two-fold. The first is to compare the bi-phase IEA with mono-phase IEA in terms of convergence time and tracking performance of the evolved fuzzy controller. The second is to compare the tracking and robustness performance of the conventional PD control with the fuzzy controller evolved by the proposed bi-phase IEA.

III.6.2.A Design setup and specifications

The control task in the design phase is to track the following trajectory:

$$q_d = \begin{cases} 1 & t \leq 2 \\ 0.5(1+\cos(\pi t)) & t > 2 \end{cases} \quad \text{(III.17)}$$

The initial states are given by: q=0 [rad], and $\dot{q}$=0 [rad.s^{-1}]

The population size, the mutation rate, the crossover probability, and the number of exploration phase were set at 50, 0.1, 0.8, and 45, respectively. Since IEA is stochastic algorithm, it is run ten times using different random number generator seeds producing in such a way different initial populations. The best FKB found by the IEA in each of the ten runs was recorded, and each of these runs was stopped after 150 fitness evaluations.

III.6.2.B The best FLC evolved by bi-phase IEA

The FKB that produce the best final objective value is illustrated in Fig. III.20 and Table. III.3. Fig. III.20 depicts the MFs of the input/output variables optimised by the bi-phase IEA including those deduced by symmetry. It is evident that their fuzzy partitions are effectively distinguishable and complete.

The entire FRB of the best FLC is included in Table. III.3. Clearly, there is symmetry of linguistic terms with respect to the fixed fuzzy rule base -if x_1 is Z and x_2 is Z then E_a is Z- and monotonic increase in linguistic terms from left to right and from top to down.

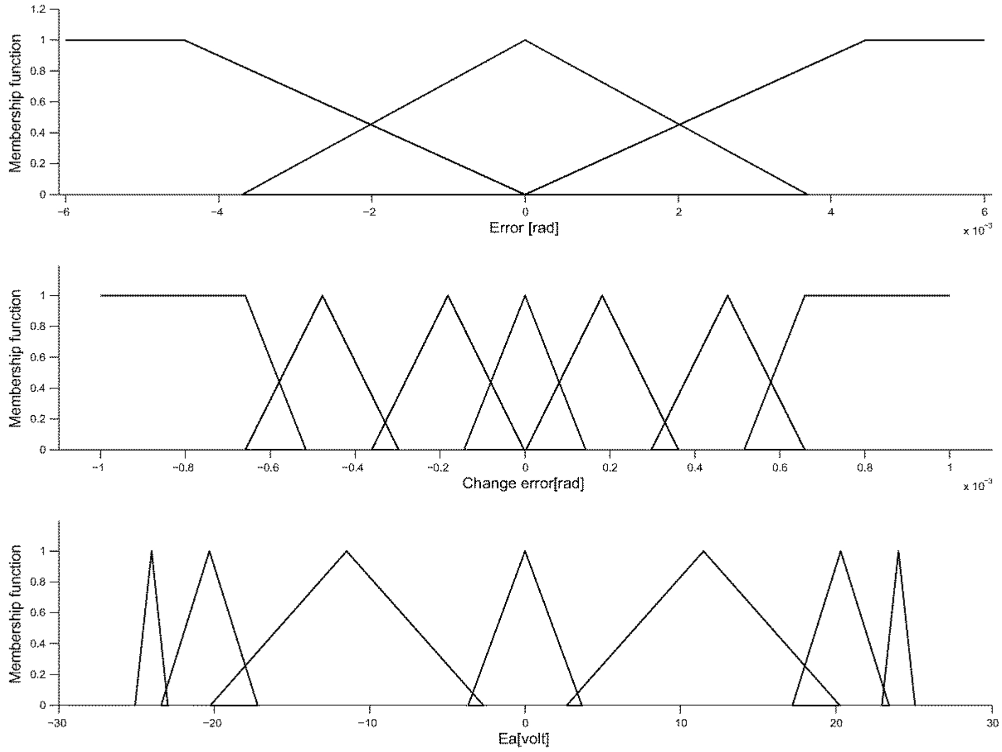


Fig. III.20 The best membership functions evolved by bi-phase IEA.

$x_1 \backslash x_2$	NB	NM	NS	Z	PS	PM	PB
N	NB	NB	NB	NM	NS	NS	NS
Z	NB	NS	Z	Z	Z	PS	PB
P	PS	PS	PS	PM	PB	PB	PB

Table. III.3 The best FRB evolved by bi-phase IEA for DC motor control.

III.6.2.C Bi-phase IEA vs mono-phase IEA

To assess the usefulness of the exploitation phase, a mono-phase IEA is considered for comparison. Mono-phase IEA consists of only the exploration phase. To compare the performance, we measure how fast an algorithm designs the best FLC using the same initial population and the same control parameters settings. Fig. III.21 shows the best fitness function values achieved along the genetic generations by mono-phase IEA and bi-phase IEA. During the first 30 generations, i.e., the exploration phase, both algorithms act identically and have the same fitness function values. After the 30th generation, it is clear that the bi-phase IEA finds better optimized FLC faster than the mono-phase IEA. The potential of the exploitation phase is, therefore, justified.

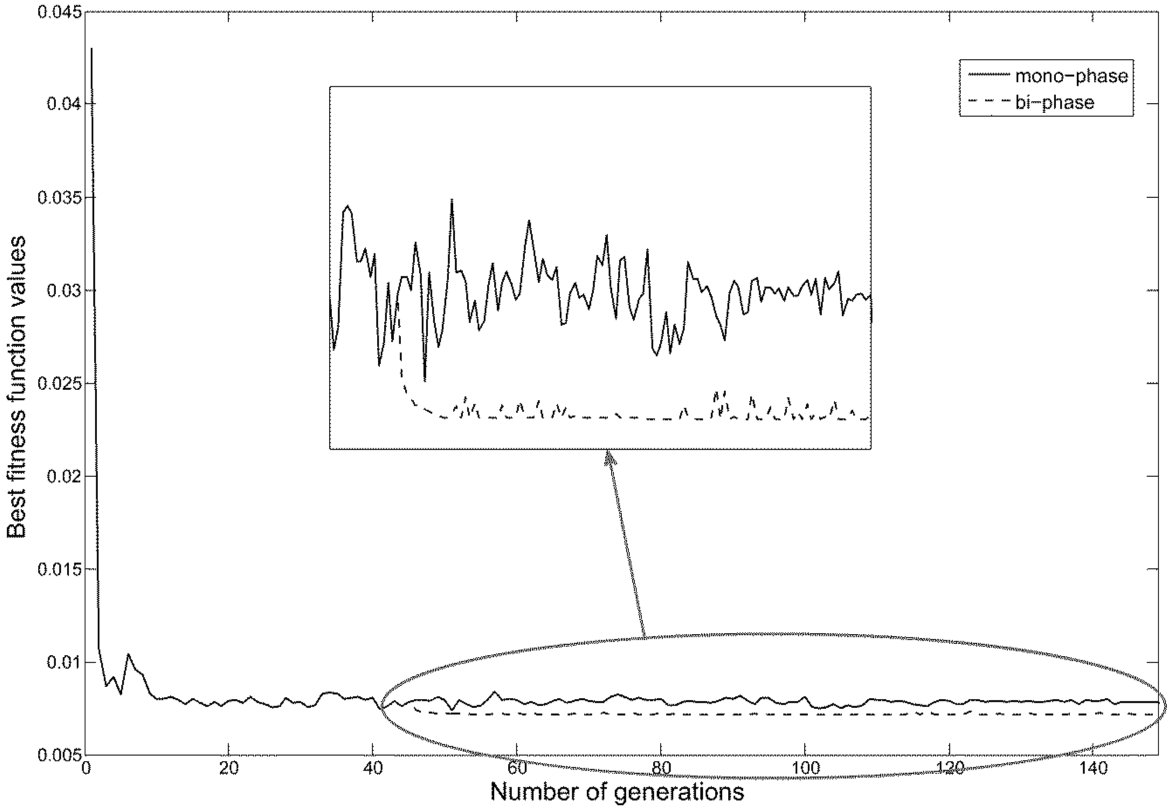


Fig. III.21 Comparison of mono-phase and bi-phase IEAs.

III.6.2.D Controller comparative simulation

In order to highlight the effectiveness of the evolved fuzzy controller, we compare its performances to the conventional PD control. The gains of PD controller are given as : K_P=400, K_D=3. They are determined according to the Ziegler-Nichols tuning method based on the step response of the plant.

The performances of the evolved fuzzy controller are compared against those of the PD controller for two cases:

Nominal case: It is a disturbance-free case where the nominal model of the DC motor is used without inducing any disturbances.

Disturbed case: To perform a qualitative assessment of the robustness of the designed FLC, the motor is supposed to be affected by the following types of disturbances: load disturbance, friction disturbance and motor torque disturbance.

- *The motor torque disturbance* corresponds to internally generated ripple disturbances due to the design of the motors. These disturbances have frequency components proportional to the motors speed and can cause significant position errors in some frequency regions. It is given by :

$$T_l = 1.47\sin(\dot{q}) + 1.4\sin(t) \tag{III.18}$$

- *The load disturbance* models various external forces that affect the inertia during the interaction with the environment, e.g., forces due to material processing in tool machines or forces due to the impact, for example at spot welding. In the simulations the moment of inertia of the motor shaft is varied while the motor is in motion as:

  a) $t \le 2s$, $I_n=0.0974$ N.m.s^2/rad (nominal value);

  b) $2s < t \le 5s$, $I_n=0.2922$ N.m.s^2/rad (three times of nominal value);

  c) $5s < t \le 6s$, $I_n=0.0974$N.m.s^2/rad (reduced inertia to nominal value);

  d) $6s < t \le 8s$, $I_n=0.5844$N.m.s^2/rad (six times of nominal inertia);

  e) $8s < t \le 12s$, $I_n=0.0974$ N.m.s^2/rad (reduced inertia to initial value).

- *The friction disturbance* is a complex phenomenon, but its most important aspects can be captured by the viscous and coulomb effects. In this study, they are given as:

$$T_f = 0.5\dot{q} + 0.16sgn(\dot{q}) \tag{III.19}$$

Where *sgn* denotes the sign function.

The control objective is to control the angular position of the motor shaft to track the following trajectory:

$$q_d = 0.6(1 - \cos(0.5\pi t)) \tag{III.20}$$

The simulation results of the designed FLC and the conventional PD controller under the two cases are shown in Fig. III.22 and Fig. III.23, respectively. For both controllers, and at all cases, the motor torques and the input voltages are well within the ranges of allowable value: [-5.29 5.29][N.m] for motor torque and [-24 24][V] for the input voltage. One can see in Fig. III.22(d)-(e) and Fig. III.23 (d)-(e), the controllers have produced a sinusoid-like variations to counter act the motor torque disturbances. Damped oscillatory behavior can also be seen at the instants of the abrupt change of the inertia (2s, 5s, 6s and 8s) but without deteriorating the tracking performance.

At the nominal case, the maximum tracking error of the PD controller is about of 0.36 degrees while it is about of 0.01 degrees for the designed FLC. Actually, the PD controller shows a maximum tracking error of 36 times larger than that of the designed FLC. At the disturbed case, the maximum tracking error of the PD controller has now increased from 0.36 to 0.63 degrees which is still acceptable. While for the designed FLC, it is increased from 0.01 to 0.03 degrees, which is very small considering the large sudden inertia and the presence of the motor torque and friction disturbances.

It can be concluded that the performance of the control system can be improved greatly by using the linguistic FLC designed by the proposed EA.

III.6.2.E High speed tracking

Additional simulations were performed to see the performance of the evolved controller in high speed tracking. We set to the DC motor another reference trajectory, where the velocity is increased. This trajectory is described by:

$$q_d = 0.6(1 - \cos(\pi t)) \qquad \text{(III.21)}$$

The simulation results of the designed FLC and the conventional PD controller, under the same conditions as for the previous trajectory, are presented in Fig. III.24 and Fig. III.25. It is quite evident, from the typical results shown in Fig. III.24 (c) and Fig. III.25 (c), the designed FLC gives better tracking performance than the conventional PD control as a maximum tracking error of about 0.01 is observed for the FLC versus 0.62 degrees for PD controller.

The results depicted in Fig. III.25 (d)-(f) show motor torque and voltage saturation when the inertia is increased 6 times; and only a torque saturation as effect of the coulomb friction at about 10 sec. The maximum error tracking of the PD controller reaches 4.18 degrees. Therefore, we can realize that the PD controller fails to fulfil the quite demanding control challenge imposed by the fast tracking trajectory with the different types of disturbances.

While the effects of the added disturbances are clearly evident in the angular position errors, the designed FLC successfully maintains the position error in a very satisfactory tolerance [-0.02 0.025] degrees without voltage or torque saturation, Fig. III.24(d)-(f). The designed FLC again prove its superiority on PD control.

III.7 Conclusion

In this chapter, we have proposed an integer IEA for simultaneous optimization of FRB and FDB optimization of chattering-free and interpretable Mamdani-type-1 fuzzy controller. The main characteristics of our evolutionary design technique are :

- Consideration of the variation of the control input as components of the fitness function;
- the use of a bi-phase scheme to improve the accuracy of the designed FLC;
- the encoding of the overlapping parameter in the chromosome;
- the use of dependent searching ranges for MF parameters to ensure the evolving of valid interpretable FKBs.

The simulation results presented here, have demonstrated the effectiveness of the proposed IEA to design smooth and robust Mamdani FLCs capable of controlling direct-drive DC motor to track a desired trajectory. The evolved Mamdani type FLC was shown to be robust to measurement noise and load perturbations without significant chattering in the control input.

More simulations were conducted to assess the validity and usefulness of the bi-phase IEA. The results obtained suggest that the proposed bi-phase scheme does its job of accelerating the IEA convergence and improving the best fitness function. They also show the excellent dynamic performance of the evolved FLC for different operating conditions which reflects the nonlinear character of the designed controller.

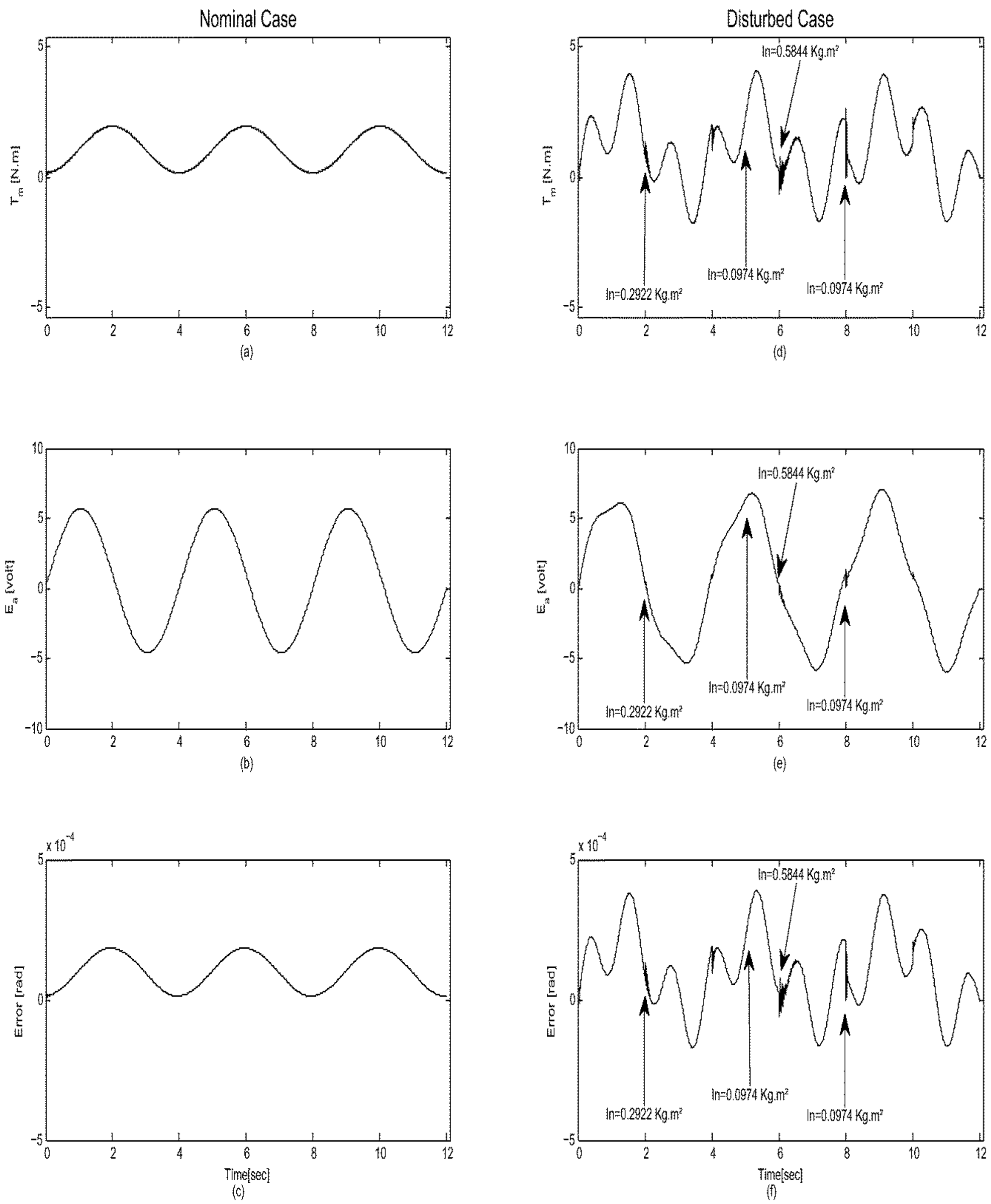


Fig. III.22 Tracking performances and control activities of the designed FLC: (a)-(c) Nominal Case, (d)-(f) Disturbed Case.

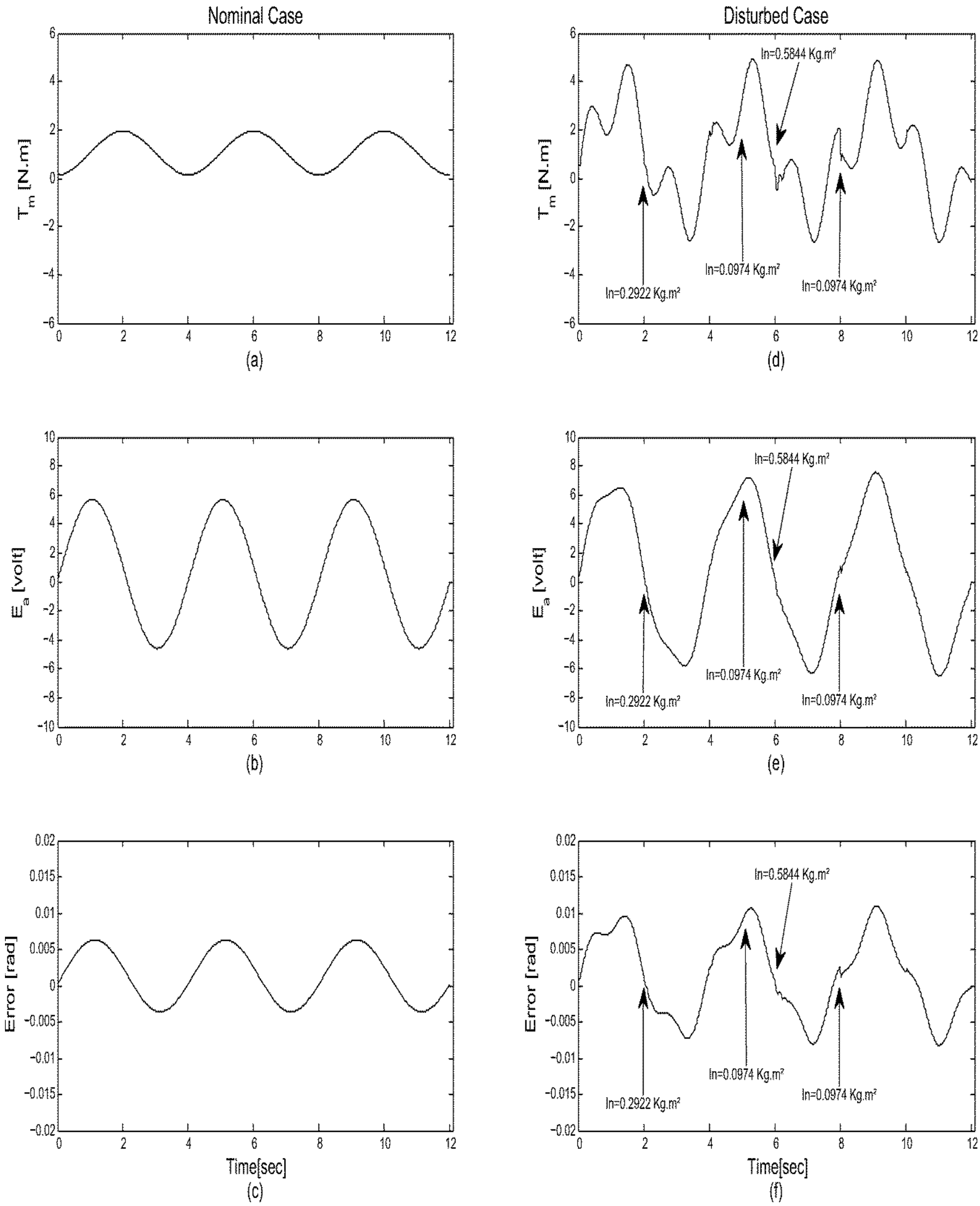


Fig. III.23 Tracking performances and control activities of the PD controller : (a)-(c) Nominal Case, (d)-(f) Disturbed Case.

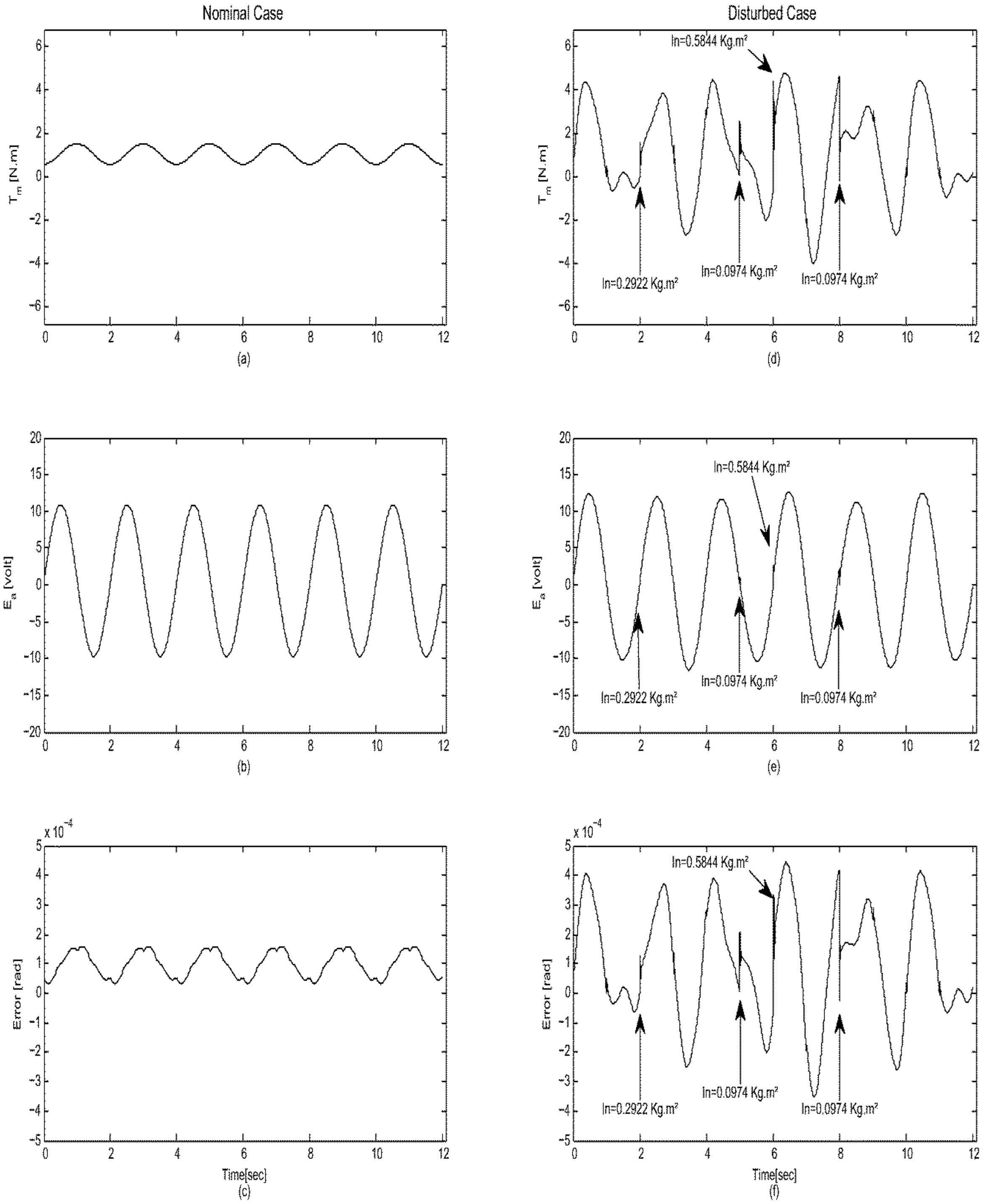


Fig. III.24 Fast tracking performances and control activities of the designed FLC : (a)-(c) Nominal Case, (d)-(f) Disturbed Case.

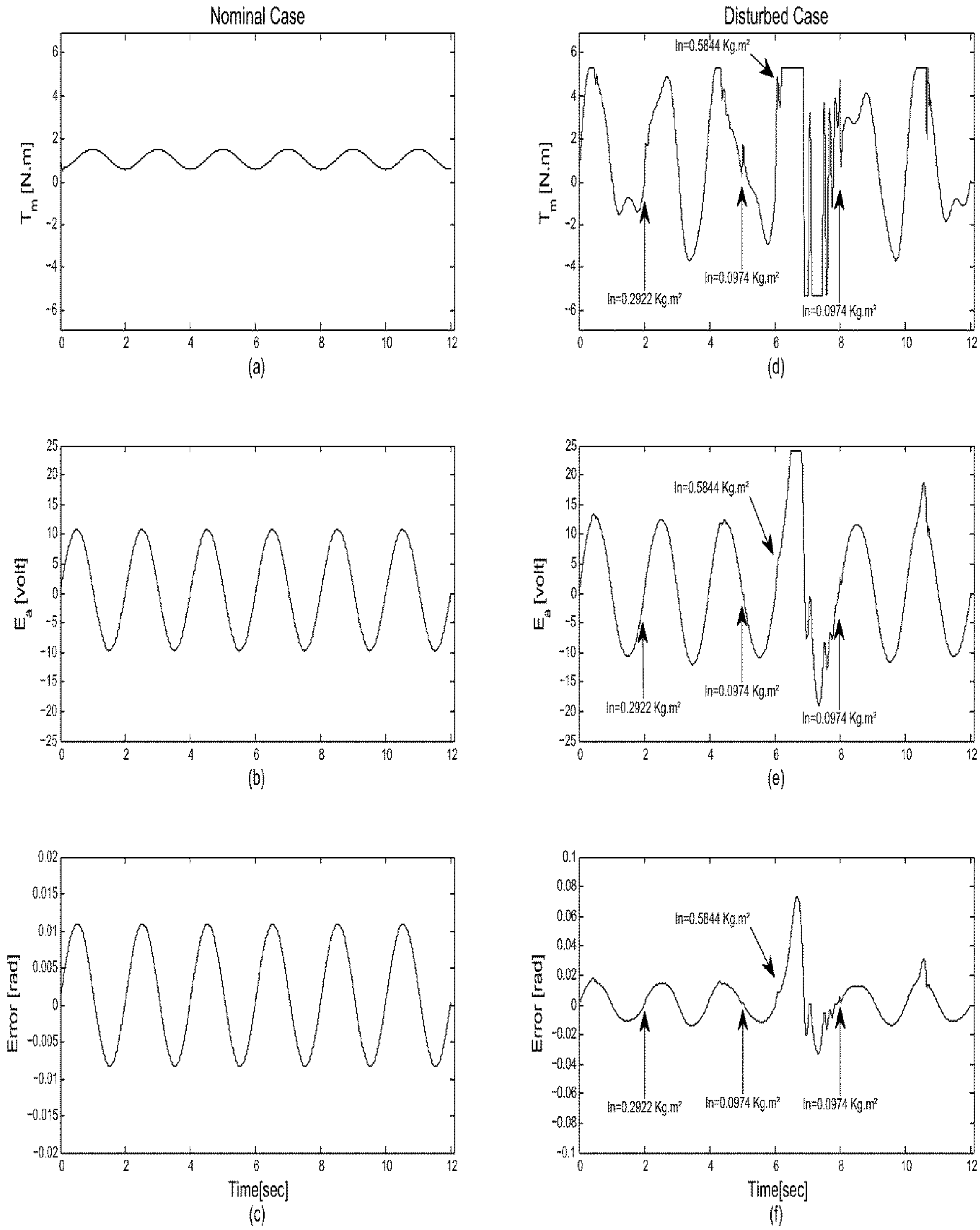


Fig. III.25 Fast tracking performances and control activities of the PD controller : (a)-(c) Nominal Case, (d)-(f) Disturbed Case.

Chapter IV: Design and Robustness Enhancement of Sectorial Fuzzy Controller via Evolutionary Algorithm

IV.1 Introduction

FLC design by EA arises in very broad field of applications and is solved with completely different evolutionary design techniques depending on the particular FLC class and the application specifications [108], [109], [110], [111], [112], [113], [114]. In this chapter, we mainly study an evolutionary design of a very important and widely used class of fuzzy controllers, namely the sectorial fuzzy controllers (SFCs) reported in [115]. SFC is two-input/one-output fuzzy controller viewed as a nonlinear mapping characterized by specific properties characterizing the FRB, the FDB, the defuzzifier and the fuzzy inference engine. The evolutionary design of the SFC is only concerned with the sectorial properties related to FDB and FRB. The most challenging properties among these properties are the monotonicity property associated to the FRB and the complementarity property of the fuzzy partition associated to the FDB. The monotonicity property is usually obtained by implementing MacVicar-Whelan meta-rules in the initial population [27], [28]. According to the permissible values of the output labels, the transition between the adjacent fuzzy rules could be large which deteriorate the smoothness performance of the designed controller. For the complementarity property of the fuzzy partition, most of the proposed methods constraints the characteristic points of the MF to occur within certain fixed ranges in the universe of discourse. This strategy affect the good performance of the optimization method since these parameters are dependent among themselves for each MF and moreover on those of the adjacent MFs.

Design of robust control systems has long been a focus of active research and concern for control and automation community [116], [117], [118], [119]. Robustness property indeed is a primary consideration to take in the assessment analysis of any control system. It consists in small sensitivity of control performance (stability, accuracy, dynamic performance, etc) to inaccurate model, parameter changes and perturbations. FLC is one of the advanced control systems that is commonly known to be robust to plant uncertainties [120], [121], [122], [123], [124], [125], [126]. As stated in [120] and [121], this feature arises from the fact that the fuzzy logic allows to an input data with perturbation to belong to the same fuzzy set as the same data without perturbation but with different membership function value. The support's length of membership function associated to fuzzy sets determines the perturbation level affecting the input data that will be accepted as element of the same fuzzy set. Thus, expanding the membership function's support can increase the robustness to perturbation, while on the other hand it could decrease the accuracy performance. Therefore, a balance must be found during the design between robustness and accuracy. This problem in general is not computationally tractable with conventional design techniques. Robust design methods are the most suitable candidate to tackle such optimization design problems [127], [128]. However, they are rarely applied in control area. This later calls for design methods that integrate only the accuracy criterion in the design process. Probably, this is due because they cannot handle multiple objectives efficiently. To cope with the uncertainties and the trade-off between the robustness and accuracy performance in the design phase, robust EA is an effective and efficient design technique [129], [130], [131], [132], [133], [134] to achieve this job. It is a powerful tool that has already proven its capabilities in several engineering design, specifically in minimizing the effect of uncertainties in a design solution without eliminating the source of uncertainties, which is difficult, if not impossible task. The EAs used for FLC design need model of the plant to be controlled which can be quantitative or qualitative (neural, fuzzy or neuro-fuzzy model). This model in general constitutes a nominal model. However, the controller designed once set to practical use has to deal with the plant affected by structured and unstructured disturbances. Such disturbances are usually modelled as error model. To take into account these disturbances during the design, the whole model is used, i.e., nominal model and error model in the evolutionary design phase.

Our contributions in this work are twofold. First we accommodate the sectorial properties in the evolutionary search through a special parameterization of the fuzzy rule base (FRB) and the membership functions (MFs) of the SFC, repairing operator and special initialization of FRB chromosome part. The second contribution, concerning the robustness enhancement, consists of two-stage search strategy. At the first stage, the accuracy criterion is considered alone, while at the second stage both robustness and accuracy criterion are taken into account as a two-objective optimization problem. The main research motivations and contributions are schematically summarized in Fig. IV.1.

This chapter is structured as follows. In section 2, we describe the direct-drive DC motor to be controlled. In section 3, we give some preliminary concepts on robust evolutionary optimization and SFC. The components of the SFC to be optimised are given in section 4. We present in section 5 the strategy of taking into account the sectorial properties during the evolution. In section 6, we present the implementation details of the EA to SFC design and robustness enhancement. In section 8, simulation results and discussions are given including a comparative controller study.

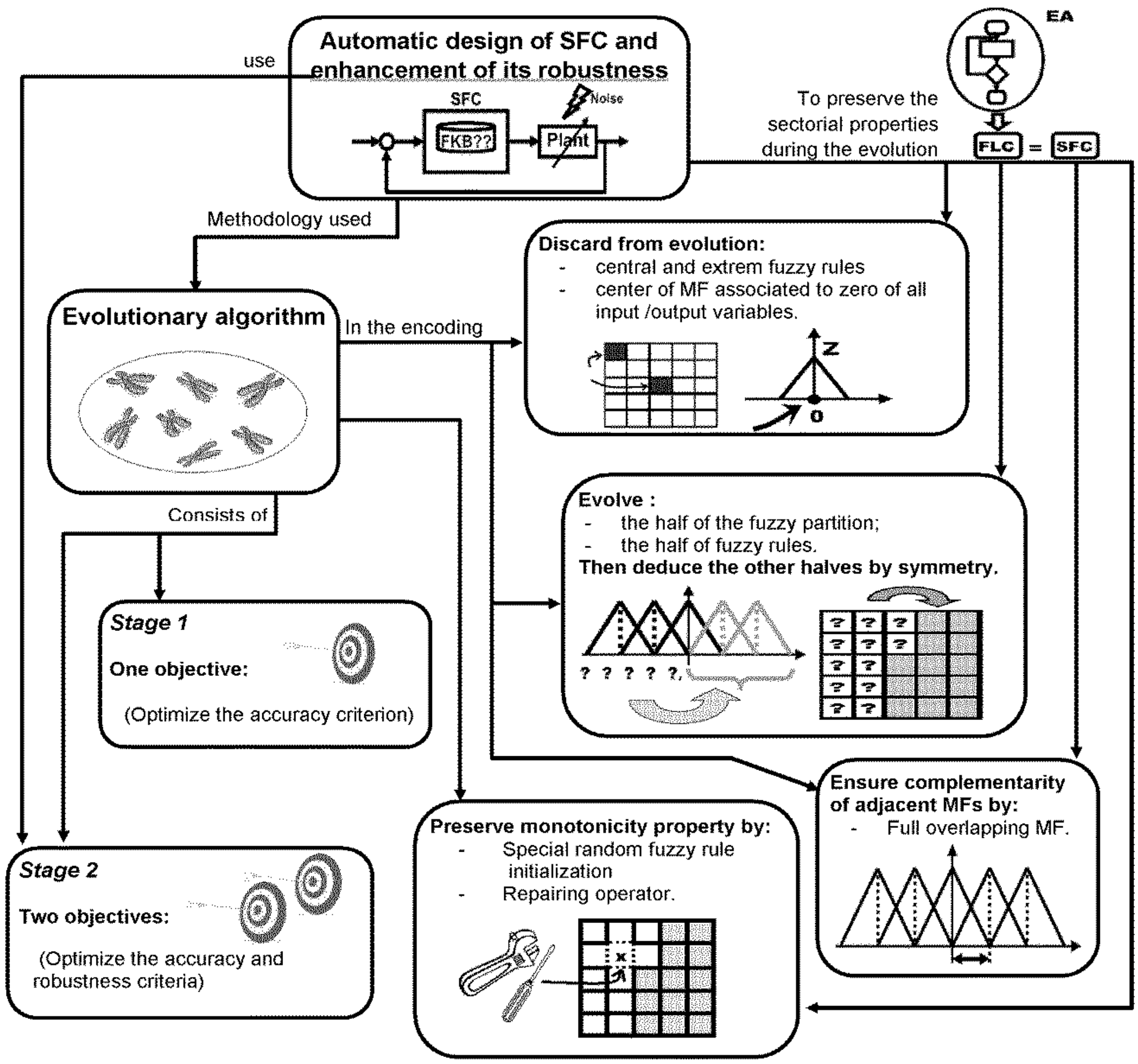


Fig. IV.1. Schematic representation of research motivations and contributions.

IV.2 Preliminaries

IV.2.1 Robust Evolutionary Optimization Design

Engineering design methods can often be cast in terms of optimization design methods, where the objective function is optimized by altering the design parameters or variables while meeting various specifications. However, such methods suffer from the presence of uncertainties as almost all the other disciplines related to engineering. Sources of uncertainties include, to name but a few, physical measurement limitations, the use of stochastic simulation models, complexity of the phenomena to handle, implementation effects (discretization, quantization), and human-machine interaction. As consequences of these practically unavoidable and uncontrollable uncertainties, the optimization design technique yields to a solution design not at the precise point in the design space but somewhere in its neighbourhood. Thus, the resultant design solution can have disastrous consequences once put to work in real world application.

Robust evolutionary optimization design is one way to effectively cope with these uncertainties without eliminating its sources.

EAs are frequently reported to be able to cope well with the uncertainties present in environment, design parameters, and fitness evaluation. In fact, engineering design in presence of uncertainties is considered as a prime application domain for EAs, and that uncertainties can even be helpful in evolutionary search. Indeed, design solutions that are far apart in the design space may have similar objective function values but may have significantly different sensitivities with respect to uncertainties. Thus, allowing for perturbations and parameter variations in the plant during optimization design is potentially the best means of influencing the robust character of the design.

IV.2.2 Sectorial Fuzzy Controller

One of the first researches in investigating the stability analysis of linguistic fuzzy controller based on the passivity theory is reported in [115]. It is pointed out that most fuzzy controllers set in real world applications have some features in common. They are basically two-input/one-output FLCs characterized by sectorial properties [135], [136], classified according to the concerned FLC's component as follows:

- **Fuzzy data base**: The universes of discourse of the input/output variables are symmetrical with respect to zero. They are partitioned into an odd number of fuzzy sets associated to labels assigned arbitrarily. The membership functions for input variables are convex and the adjacent ones are complementary, i.e. the sum of their membership function values is one. At zero, the membership function values for input and output variables are zero.

- **Fuzzy rule base:** The central fuzzy rule has zero labels in both IF part and THEN part corresponding to null output for null inputs. The look-up table of the fuzzy rules is symmetric with respect to the central fuzzy rule, and has a gradual increasing monotonicity in consequent labels within rows from left to right and within column from top to down, Fig. IV.2.
- **Fuzzy inference engine:** Minimum or product inference method is used to derive the output fuzzy set.
- **Defuzzifier:** The crisp output is computed by the centre average defuzzification method.

The FLC that fulfils the aforementioned properties is referred as SFC. The absolute stability for Lagrangian systems driven by this class of FLC was proved in [115] using a passivity approach.

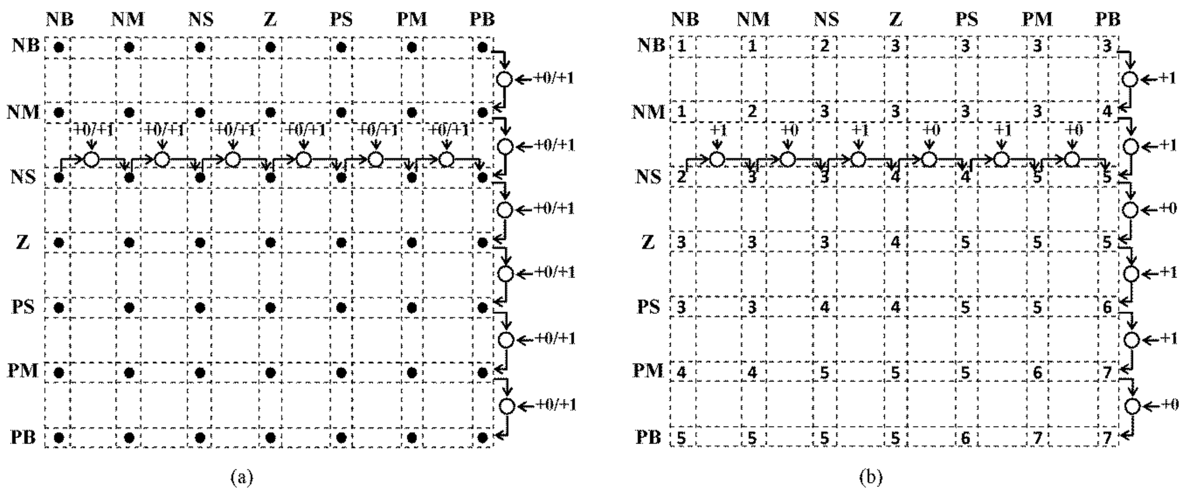


Fig. IV.2. Increase monotonicity property (a) mechanism of increase monotonicity property illustrated in the 3rd row and 7th column of the FRB's look up table; (b) example of FRB respecting the increase monotonicity property.

IV.3 Sectorial Fuzzy Controller to be Evolved

The inputs of the used SFC are the error (x_1) and the change error (x_2) on the angular position of the motor shaft. At the output, the SFC provides the input voltage (E_a)

that excites the DC motor and brings it in the desired angular position. Let the universe of discourse be $U=U_1 \times U_2$, where $U_1=U_2=[U_{min}, U_{max}]= [-0.05, 0.05]$, and V$=[-24, 24]$.

The SFC consists of the following components:

A ***singleton fuzzifier*** converts a crisp value $\underline{x} \in U$ into a fuzzy singleton A_x within U.

The ***fuzzy data base***: The space of the inputs x_1 and x_2 and the output E_a are partitioned into seven membership functions associated to the following labels: negative big (NB), negative medium (NM), negative small (NS), zero (Z), positive big (PB), positive medium (PM), and positive small (PS).

The ***fuzzy rule base*** consists of a collection of fuzzy IF-THEN rules expressed as:

$$R^{(l)}: \text{ IF } (x_1 \text{ is } A_1^l \text{ and } x_2 \text{ is } A_2^l \text{) THEN (} v \text{ is } C^l \text{)} \tag{IV.1}$$

Where A_i^l and C^l are terms associated to the fuzzy sets F_i^l and G^l defined in U_i and V, respectively, with $l = 1,2,.....,M$. M is the number of rules in the FRB. Here we have $M = 7x7 =49$ to account for every possible combination of input fuzzy sets.

Each fuzzy IF-THEN rule defines a fuzzy implication:

$$R^l = F_1^l \, x \, F_2^l \rightarrow G^l \tag{IV.2}$$

$$R^l = \{ ((\underline{u},v), \mu_{R^l}(\underline{u},v)) \mid \underline{u} \in U, v \in V \} \tag{IV.3}$$

Where $\mu_{R^l}(\underline{u},v)$ is defined by the following Larsen's fuzzy implication rule:

$$\mu_{R^l}(\underline{u},v) = \mu_{F_1^l x F_2^l}(\underline{u}) \, . \, \mu_{G^l}(v) \tag{IV.4}$$

$$\mu_{R^l}(\underline{u},v) = (\mu_{F_1^l}(u_1) \, . \, \mu_{F_2^l}(u_2)) \, . \, \mu_{G^l}(v) \tag{IV.5}$$

The ***fuzzy inference engine*** derives from each fuzzy rule of the FRB an output fuzzy set, in the following way:

Each fuzzy rule of (IV.6), described by a fuzzy implication R^l, determines a fuzzy set B^l $=Ax \, o \, R^l$ in V such that:

$$\mu_{B^l}(v)=\mu_{Ax \, o \, R^l}(v) \tag{IV.6}$$

$$\mu_{B^l}(v)=\max_{\underline{u} \in U}\{ \mu_{Ax}(\underline{u}) \, . \, \mu_{R^l}(\underline{u},v) \} \tag{IV.7}$$

The ***defuzzifier*** used in our fuzzy controller is the centre average defuzzifier. Let v^l denotes the point at which μ_{B^l} achieves its maximum, which is associated with the activation of the

l^{th} fuzzy rule. This defuzzifier evaluates $\mu_{B^l}(v^l)$ at v^l, and then computes the output of the SFC as:

$$y = \frac{\sum_{l=1}^{M} v^l \mu_{B^l}(v^l)}{\sum_{l=1}^{M} \mu_{B^l}(v^l)} \qquad \text{(IV.8)}$$

IV.4 Preservation of the Sectorial Properties in the Evolution Process

There are two ways to incorporate any available knowledge about the system to be design by EA. One way is the system parameterization and representation; the other way is the population initialization. In our application, we have to take into account the sectorial properties during the evolution, more specifically those related to FRB and FDB.

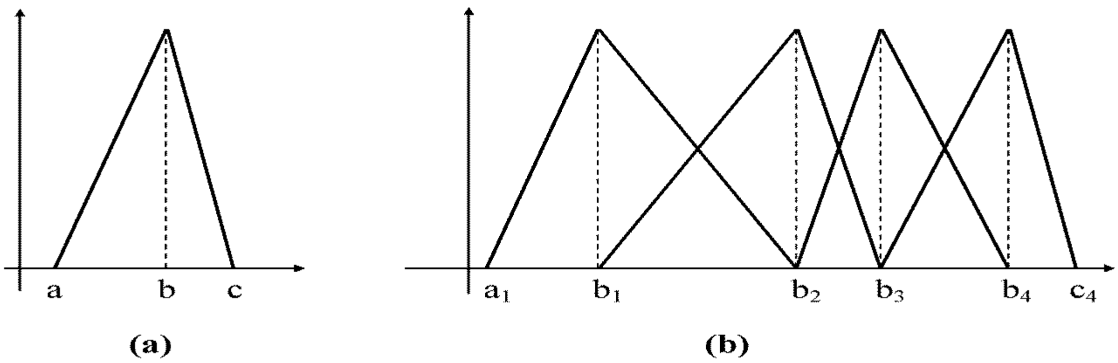


Fig. IV.3. Descriptive parameters of (a) separate triangular MF; (b) triangular MFs in fully-overlapped fuzzy partition.

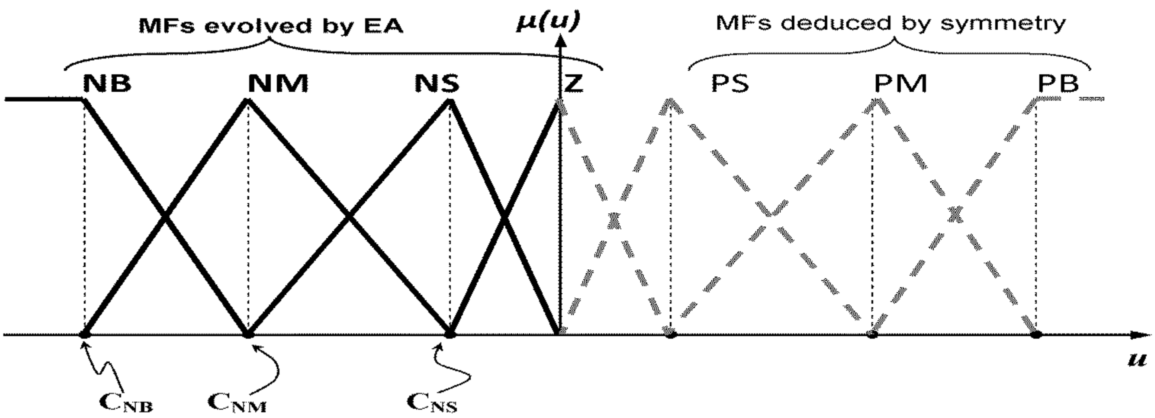


Fig. IV.4. MF parameterization.

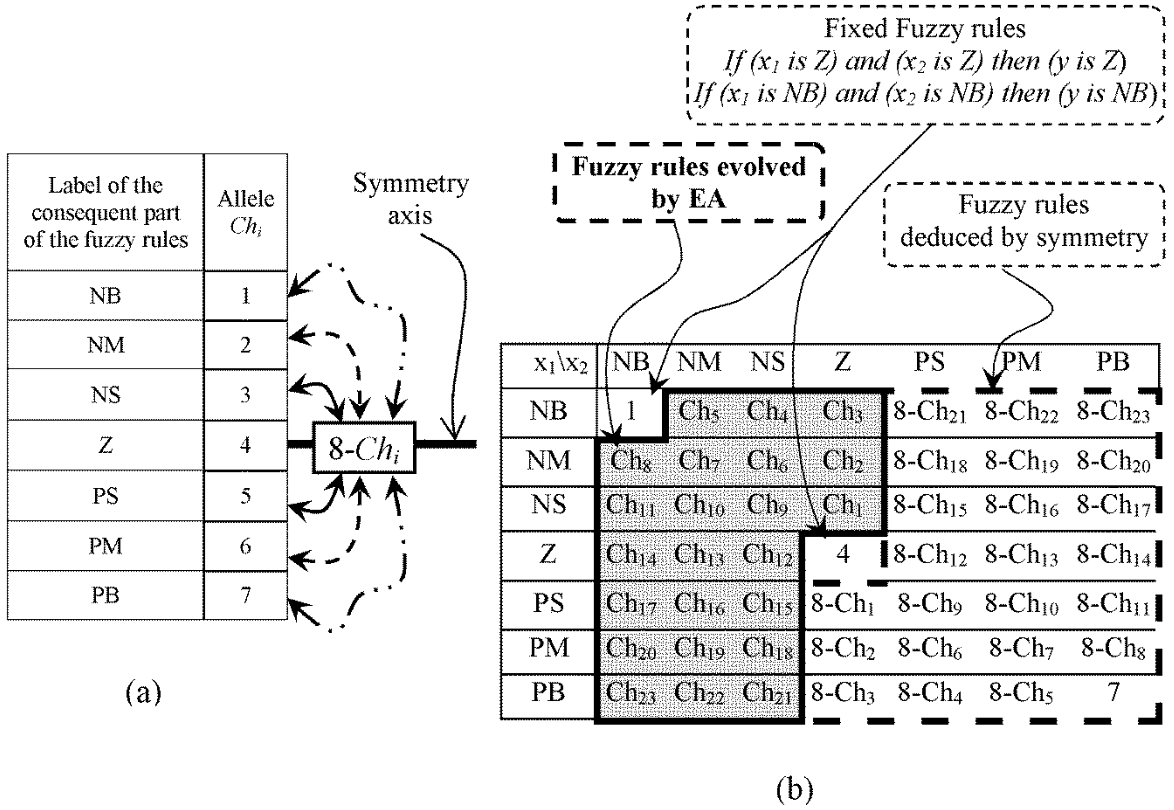


Fig. IV.5. (a) Coding of labels in the consequent part of fuzzy rules and its symmetrical mechanism; (b) FRB parameterization.

IV.4.1 SFC Parameterization and Encoding

A suitable problem representation must be chosen to ensure the sectorial properties in the generated FLC. In our EA, we propose the following considerations within the encoding framework:

- Adopt full overlapping between the adjacent MF to ensure their complementarity. Each separate triangular MF can be defined by three parameters noted in triplet *(a,b,c)* as Fig. IV.3(a) shows. In a fully overlapped fuzzy partition Fig. IV.3(b), only one parameter is needed to define a triangular MF. This is so because the end points *a* and *c* of the MF coincide with the second points *b*, i.e. points directly under the apex, of the adjacent MFs. Of course, an exception is done for the MFs located at the extremes for which one end point has also to be defined.
- Fix or discard from evolution:

a) the central fuzzy rule and the extreme fuzzy rule corresponding to: IF (x_1 is Z and x_2 is Z) THEN (v is Z) and IF (x_1 is NB and x_2 is NB) THEN (v is NB), respectively;

b) center of the MF associated to label Z of all input and output variables.

- Evolve:

  a) the parameters of the MFs located in the negative half of the universe of dicourse, Fig. IV.4;

  b) the half of the fuzzy rule base, Fig. IV.5.

  Then, the other halves of fuzzy rule base and fuzzy partitons are deduced by symmetry.

MF parameters and fuzzy rule labels to be evolved are represented in one finite length chromosome defined as a string or an array (*Ch*) of 41 integer elements or genes, Fig. IV.6. The fuzzy rule labels are coded in the first twenty three genes of the chromosome. Each of these genes takes values from 1 to 7. The remaining eighteen genes take values from 1 to 9 and they are grouped into two-integer sub chromosomes. Each of them represents a percentage of a specific range which is used to compute one of the MF parameters.

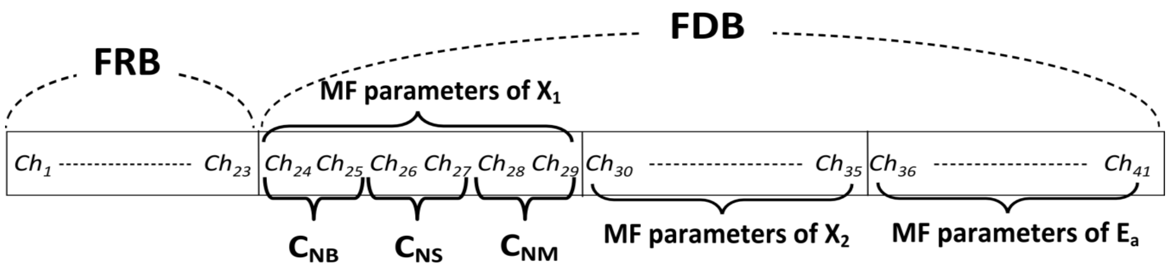


Fig. IV.6. Chromosome structure.

The general decoding relationship that calculates the numerical MF parameter (X) from its representative genes (Ch_i Ch_{i+1}) and the corresponding searching range I_X is given by:

$$X = \frac{Ch_i + 10.Ch_{i+1}}{100.I_x} \qquad \text{(IV.9)}$$

MF Parameter	Searching range length
CNB	Umin
CNM	CNB
CNS	CNM

Table. IV.1. Searching range length for MF parameters.

In the previous evolutionary design methods, it is used to optimise the position of the characteristic points that identify the MFs within independent ranges fixed off-line. However, it is well known that these parameters are dependent among themselves and among those of the adjacent MFs. To take into account these parameter interdependency, the length of the searching ranges of the MF parameters are dependent on the previous calculated parameters of the adjacent MF. They are computed during the evolution and in particular during the decoding process. Table. IV.1 gives the searching range lengths for the MF parameters evolved by the EA.

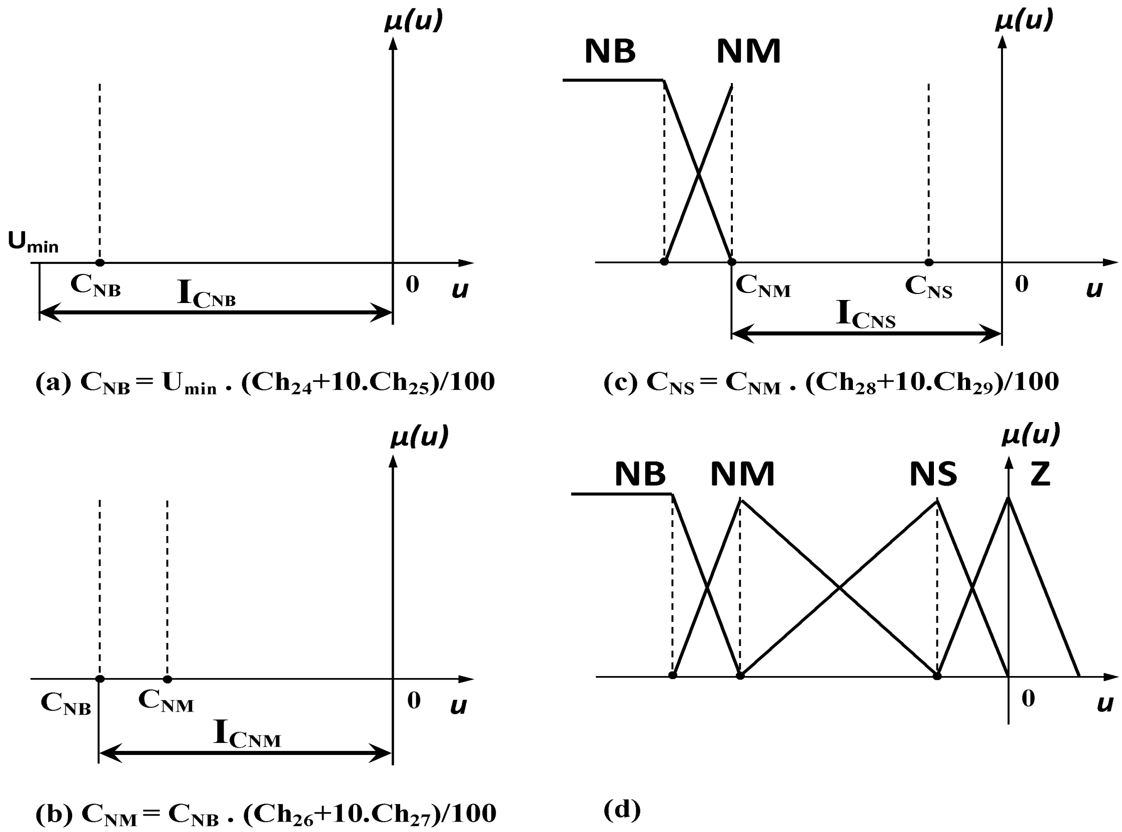


Fig. IV.7. Example of FDB decoding process and representation of searching ranges of the MF parameters for the input/output variables. (a) C_{NB}, (b) C_{NM}, (c) C_{NS}, (d) Resulting fuzzy partition.

A typical example of FDB decoding process and the representation of the searching ranges of the MFs parameters are represented on Fig. IV.7. Obviously, every searching range length depends on the previous adjacent MF parameter. The resulting fuzzy partition is subsequently always valid.

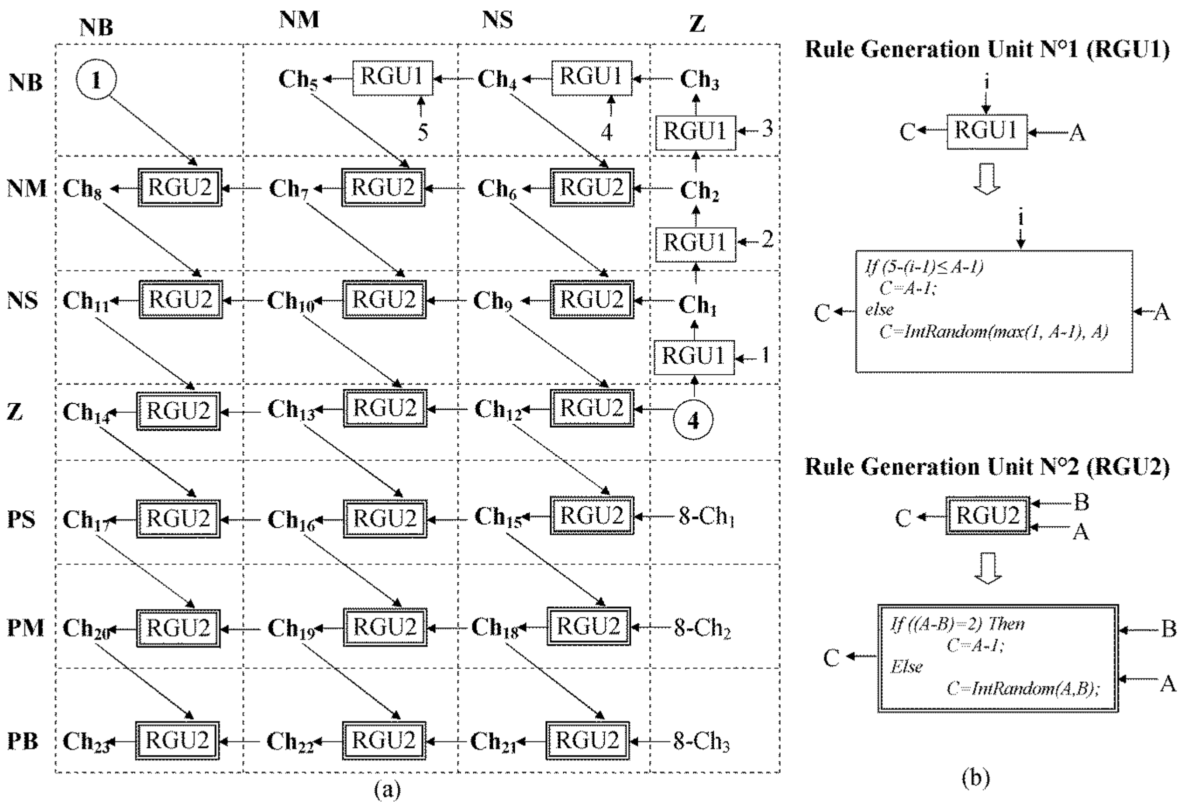


Fig. IV.8. Initialization process for the FRB's chromosome fragment. (a) Initialisation mechanism of fuzzy rule's genes; (b) Configuration of the rule generation units.

	NB	NM	NS	Z
NB	1	1	2	5
NM	1	6	2	3
NS	2	5	3	3
Z	3	4	4	4
PS	7	4	4	
PM	4	5	2	
PB	4	5	5	

Fig. IV.9. Example of the FRB fragment evolved by EA containing incorrect fuzzy rules.

IV.4.2 Population Initialization

Population initialization involves creating the initial population of chromosomes representing potential solutions of the problem at hand. Usually this is done randomly, but sometimes the available knowledge about the problem domain is used to get better solution within less time.

In this work, the purpose of the population initialization is twofold: firstly, to generate randomly the chromosome fragment that codes the MF parameters described in section 0; secondly, to generate randomly the FRB's chromosome fragment respecting the increasing monotonicity of labels within rows and columns from left to right and from top to down, respectively.

The FRB's chromosome fragment initialization process is depicted in Fig. IV.8. Label's genes Ch_1 to Ch_5 are generated successively by the rule generation unit N°1 (RGU1). For the generation of genes Ch_6 to Ch_{23}, the rule generation unit N°2 is used. Both rule generation units as described in Fig. IV.8(b) use the *IntRandom(low, high)* function which generates randomly integer number between *low* value and *high* value.

IV.4.3 Repairing Operator

During the evolution, the issued FRB could contain some incorrect fuzzy rules, as shown in the example depicted in Fig. IV.9 where they are denoted by circles. These fuzzy rules correspond to those that do not respect the monotonicity property. For these fuzzy rules the repairing operator replaces the allele of the fuzzy rule's gene by another allele generated by the corresponding rule generation unit: for Ch_1 to Ch_5 use RGU1, and for Ch_6 to Ch_{23} use RGU2.

The alleles of the following genes are replaced too successively with the same manner. This is because of the hierarchical dependency between them: changing one of them implies changing the following ones.

IV.5 The EA structure

An overview of the proposed EA used for SFC design and robustness enhancement is described in this section. The evolutionary design strategy adopted in this work includes two stages. In Fig. IV.10 and Fig. IV.11, the flowcharts of the evolutionary process at first stage and at second stage are shown, respectively. At the first stage, the population is initially generated as described previously in section IV.4.2. Then, the EA decodes the chromosomes into potential FKBs. The SFC uses each of these decoded FKBs to make the direct-drive DC motor track the desired trajectory and at the same time to compute the fitness function value (*ObjN*) that measures the tracking performance and the variation of the control signal. It is well worthy to note that the direct-drive DC motor model is a nominal one i.e. disturbance-free model. The fitness function at this stage is given by:

$$ObjN = c_1 RMSE + c_2 \sum |\Delta E_a| \qquad \text{(IV.1)}$$

RMSE is the root of mean square error representing the accuracy or tracking objective. $\sum|\Delta E_a|$ is the sum of variation of the input voltage variable that represents the smoothness objective. The parameters c_1 and c_2 are weights used to stress the relative importance of the different fitness function components. The numerical values used are c_1 = 1 and $c_2 = 10^{-7}$ and they are determined by trial and error.

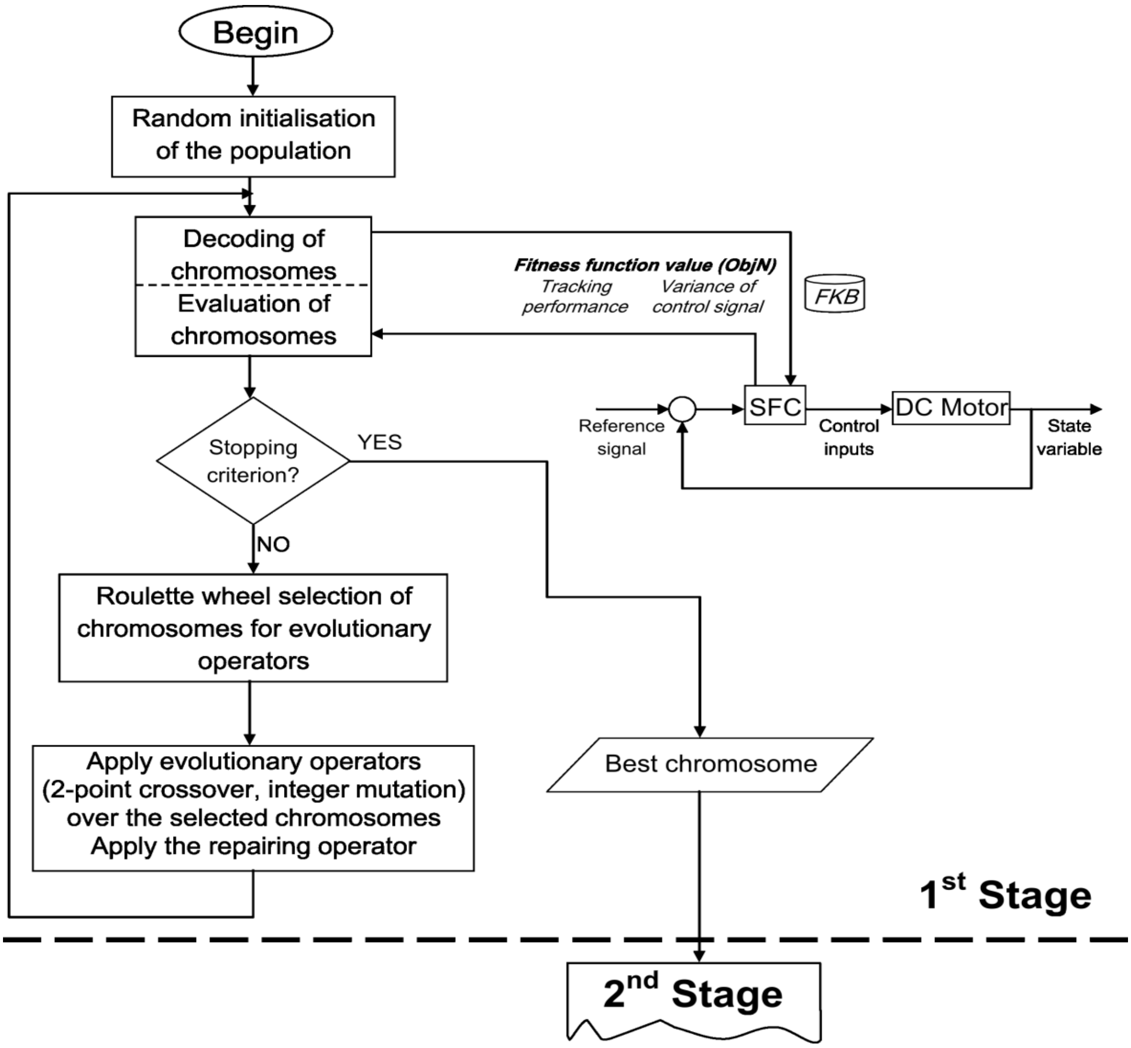


Fig. IV.10. 1st stage framework of the proposed EA.

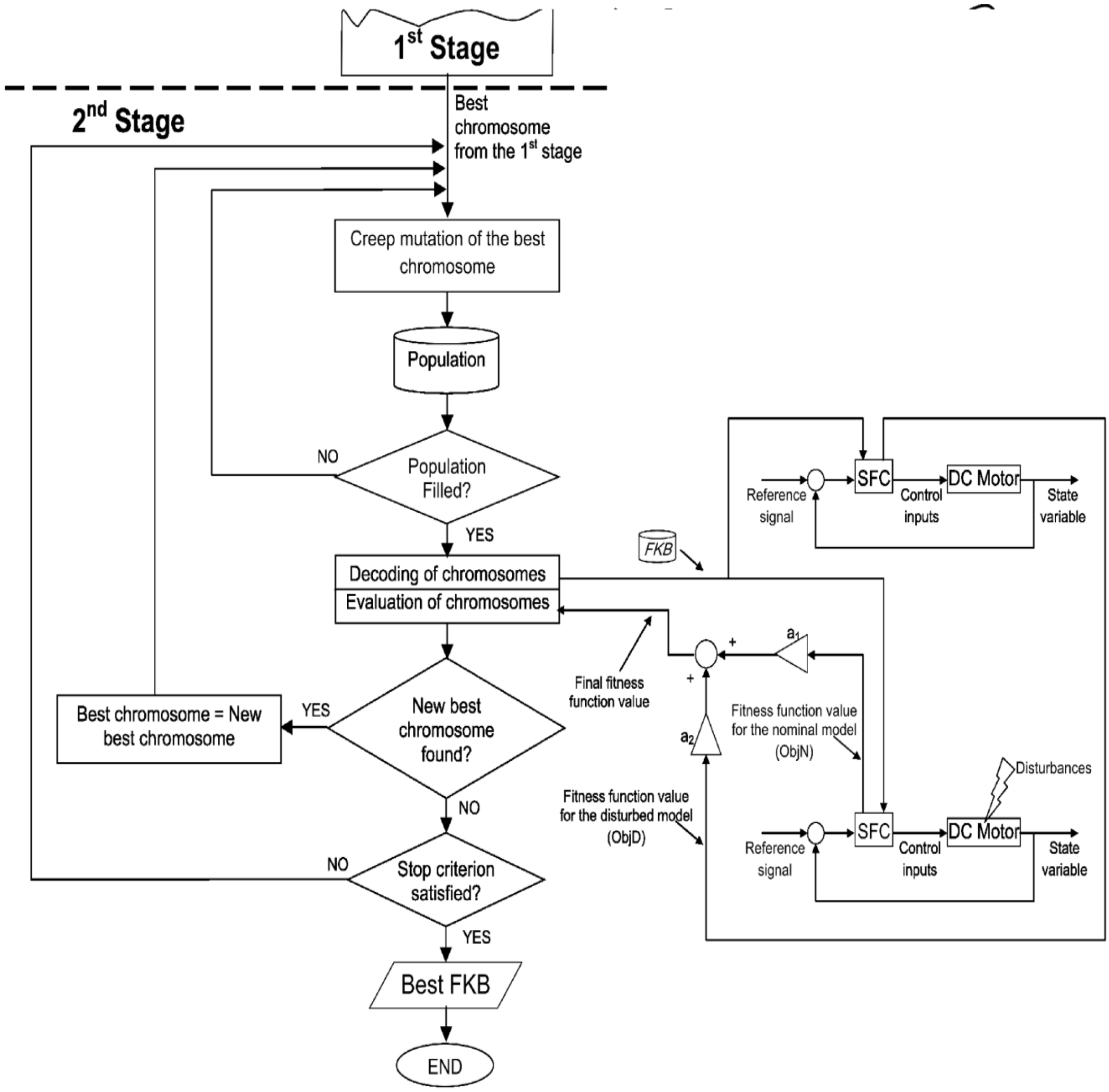


Fig. IV.11. 2nd stage framework of the proposed EA.

In each population generation and based on these fitness function values, a roulette wheel selection operator selects two parent chromosomes from the population for reproduction. The application of the evolutionary operators on these selected chromosomes creates two new chromosomes. The evolutionary operators used are the two-point crossover operator and integer mutation operator. The principle of the crossover operator in integer representation remains the same as for the binary one. The integer mutation operator changes the allele randomly inside the integer range that depends on the FLC parameter it represents. Specifically, the alleles representing FRB have an integer range from 1 to 7, and those encoding the MF parameters have an integer range from 1 to 9. Among the sectorial properties, the monotonicity property is the only one that is not necessarily preserved after the application of the evolutionary operators. To recover this property, the repairing operator is applied on the incorrect FRB fragment of the new generated chromosomes. The overall evolutionary process including the evaluation, the selection, the recombination and the reparation is repeated until the satisfaction of the stopping criterion.

After that, the second stage starts with generating the initial population by creep mutating the best chromosome obtained at the end of the first stage. The decoding and evaluation of chromosomes is then proceeded. The evaluation is done with the nominal model and the disturbed model of the DC motor providing in each case the fitness function values denoted by $ObjN$ and $ObjD$, respectively. These measures are weighted and summed up so that they form a final fitness function value (*Obj*) defined as:

$$Obj = a_1 ObjN + a_2 ObjD \tag{IV.2}$$

where a_1 and a_2 are coefficients determined by trial and error and having the following numerical value : a_1=1 and a_2= 0.01.

If a new best chromosome is found in the newly formed population, we reinitialize the population with the same manner as described above but using the new best chromosome. We repeat this process until the satisfaction of the stopping criterion. In both stages the stopping criterion is a specific maximum number of generations.

The principle aim in the second stage is the robustness enhancement of the evolved SFC resulting from the first stage. The factors used to enhance the robustness are the use of the creep mutation and the elitism strategy while disposing of the crossover and mutation operators. The creep mutation in integer representation alters a single allele, but in small increments. In this work, the creep mutation increments or decrements by 1 the allele within the corresponding range. The choice between the incrementation and the decrementation is done randomly. This creep mutation operator is technically the responsible for the robustness enhancement by shifting the best chromosome to its vicinity region in the sake to find a more robust solution.

IV.6 Simulation results

We demonstrate in this section the feasibility of the proposed EA in SFC design as well as in robustness enhancement.

IV.6.1 Design setup and specifications

In the design phase, the control objective is to make the direct-drive DC motor track the following trajectory:

$$q_d = \begin{cases} 1 & t \leq 2 \\ 0.75(1+\cos(0.5\pi t)) & t > 2 \end{cases} \quad \text{(IV.1)}$$

The initial states are given by: q=0 [rad], and $\dot{q}$=0 [rad.s^{-1}].

The population size, the mutation rate, the crossover probability, and the number of generations at the first and second stage were set at 50, 0.1, 0.8, 30, and 70, respectively. Since EA is stochastic algorithm, it is run ten times using different random number generator seeds producing in such a way different initial populations. The best FKB found by the EA in each of the ten runs was recorded, and each of these runs was stopped after 100 fitness evaluations.

For the robustness enhancement occurring at the second stage, the disturbed model is affected by the following types of disturbances: motor torque disturbance, load disturbance, friction disturbance, and measurement noise. All of them are described in the former chapter. It is supposed that this model is a worst disturbed model to be controlled by the SFC.

IV.6.2 Analysis of evolutionary dynamics

The effectiveness of the robustness enhancement of the proposed EA through the second stage will be demonstrated in the design phase. For comparison, we consider an EA similar to the proposed EA but having ObjN as fitness function in the second stage, which is noted as EA-N. The evolutionary dynamics of the EAs is obtained using the same initial population and the same control parameters settings.

Fig. IV.12 shows the best fitness function values achieved along the evolutionary generations by the proposed EA and the EA-N. It can be seen at the first stage that both algorithms ameliorate the best fitness function values and act identically. At the second stage, the EA continues to ameliorate the best fitness function value given that for EA-N the fitness function is maintained while for the proposed EA it is changed to be Obj. This fact explains the abrupt variation in the fitness function of the proposed EA at 30th generation. The evolution over the number of generations of the performance in nominal case (objN) and the performance in the disturbed case (ObjD) are presented in Fig. IV.13

for the proposed EA and the EA-N. The effect of the robustness enhancement of the proposed EA can be obviously revealed in the second stage through the decreasing of the ObjD value with a slight increase of its compagnion ObjN value.

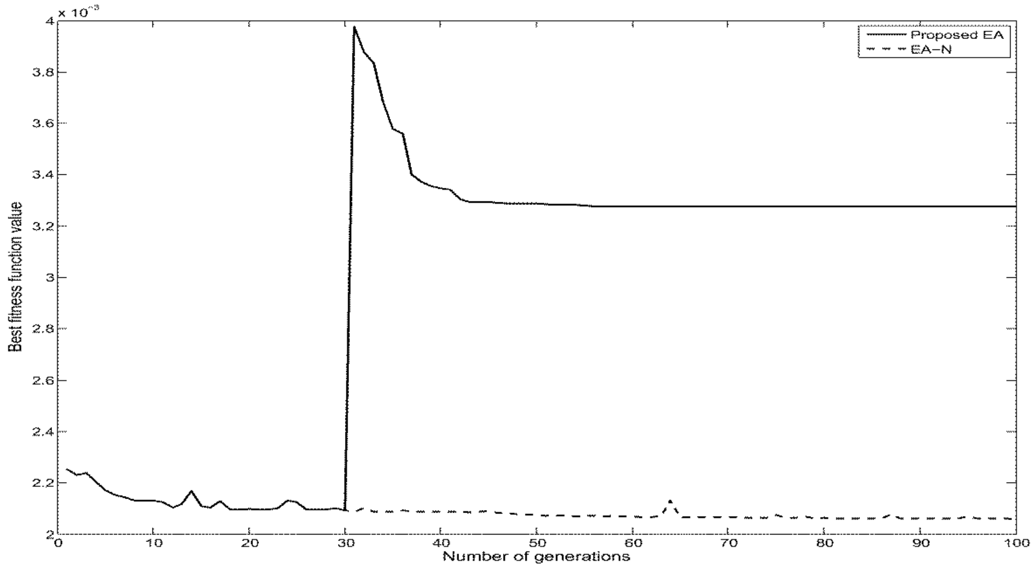


Fig. IV.12. Evolution of the fitness function values for the proposed EA and the EA-N over the number of generations.

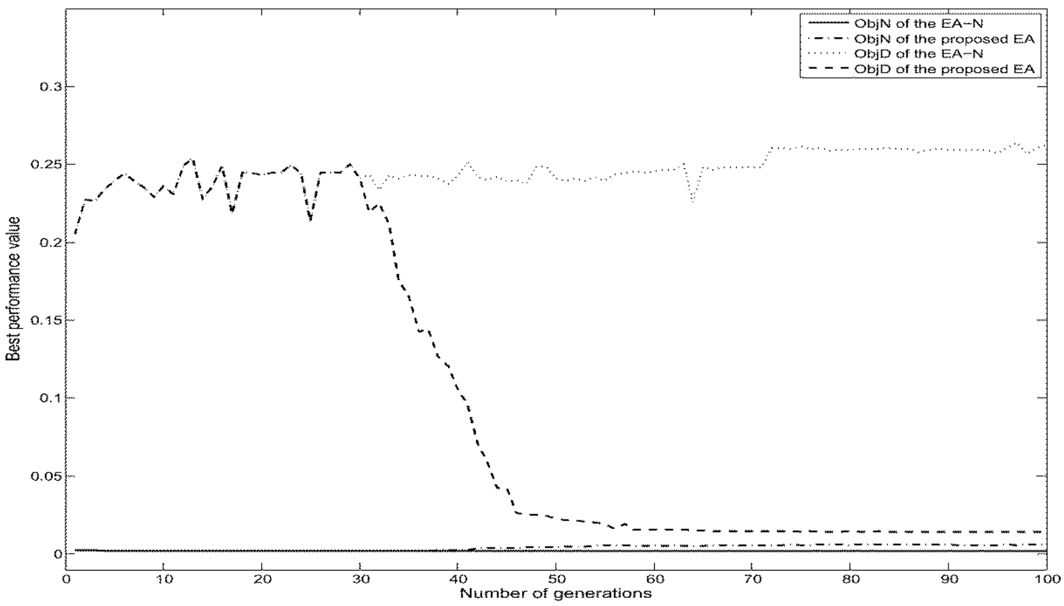


Fig. IV.13. Evolution of ObjN and ObjD for the proposed EA and the EA-N over the number of generations.

IV.6.3 Best SFC evolved by the proposed EA

Fig. IV.14 and Table. IV.2show the best FKB of the SFC that produces the best final objective value. In particular, Fig. IV.14 depicts the fuzzy partitions of the input/output variables optimised by the proposed EA including those deduced by symmetry. It is evident that the sectorial properties related to the MF are effectively meted.

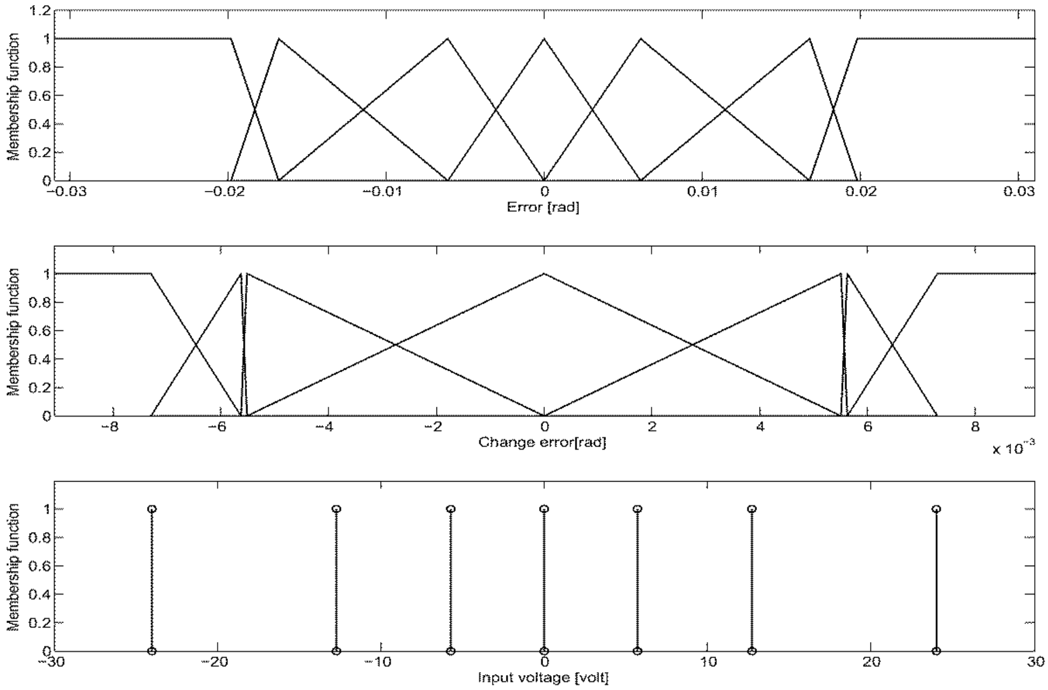


Fig. IV.14. Best fuzzy partitions for the input/output variables of the SFC evolved by the proposed EA.

$x_1 \backslash x_2$	**NB**	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NB	NM	NS	NS
NM	NB	NM	NM	NM	NS	Z	Z
NS	NM	NM	NM	NS	Z	PS	PS
Z	NM	NS	NS	Z	PS	PS	PB
PS	NS	NS	Z	PS	PM	PM	PM
PM	Z	Z	PS	PM	PM	PM	PB
PB	PS	PS	PM	PB	PB	PB	PB

Table. IV.2. Best FRB of the SFC evolved by the proposed EA for DC motor control.

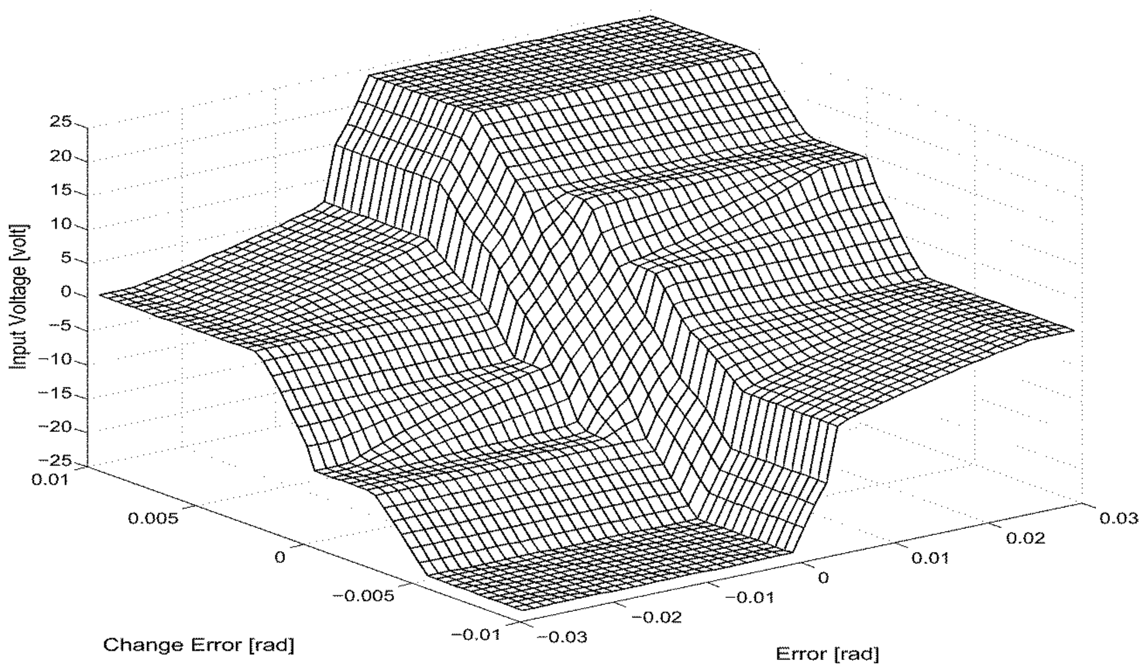


Fig. IV.15. Control surface of the SFC evolved by the EA.

The entire FRB of the best SFC is included in Table. IV.2. Obviously, there is symmetry of linguistic terms with respect to the fixed fuzzy rule base -if x_1 is Z and x_2 is Z then y is Z- and the monotonicity property in linguistic terms from left to right and from top to down is respected.

The fuzzy control surface or fuzzy decision surface of the evolved SFC is shown on Fig. IV.15. They are obtained by plotting the inferred control action E_a for discretized values of x_1 and x_2. The fuzzy control surface represents the overall behavior of the fuzzy controller, which approximates the desired dynamics. In fact, it represents in a compact way all the characteristics of the fuzzy controller including nonlinearity, the energy expended by the controller, the dominant surface region (about the equilibrium point). The shape of this surface is mainly influenced by the fuzzy controller parameters such as the shape and location of membership functions, fuzzy rules, fuzzy operators, fuzzy implication, inference mechanism and defuzzification method.

IV.6.4 Robustness Analysis and Comparative controller study

The control task undertaken for validation is a tracking control of the following trajectory:

$$q_d = 0.6(1 - \cos(0.5\pi t)) \quad \text{(IV.2)}$$

For comparison, we use the conventional PD controller and the best SFC issued at the end of the first stage. The PD controller gains are determined according to the Ziegler-Nichols tuning method based on the step response of the plant. They are given as : K_P = 400 and K_D = 3. The simulations for comparison are carried out for two cases: nominal case and disturbed case.

- The ***nominal case*** is a disturbance-free case where the nominal model of the DC motor is used without inducing any disturbances or changing any parameter.
- In ***disturbed case*** the disturbances are introduced at different instant in order to assess the motor position recovery. They are induced as follows:
    a) $2<t\leq5$s, the three types of disturbances are considered except the measurement noise.
    b) $6< t\leq8$s, all types of disturbances are introduced. The moment of inertia in these time intervals is I_n=0.5844N.m.s²/rad which is six times of the nominal value.
    c) Out of these intervals no disturbances are applied and the parameters of the DC motor take the nominal values.

Fig. IV.16, Fig. IV.17and Fig. IV.18 show the tracking performance and the control activities of the best SFC issued at the end of the first stage, the evolved SFC by the proposed EA, and the PD controller, respectively.

In Fig. IV.16, it is observed that the tracking performance of the PD controller is acceptable (the maximum tracking error is 0.00627 rad in nominal case and 0.0088 rad in disturbed case) but it is poor compared to the two SFCs. The amount of variation of the control signal in Fig. IV.17(c) at interval [2sec 5sec] is improved in Fig. IV.18(c). Owing this to the second stage of the proposed EA. This apparently improvement behaviour in the control effort is obtained at the price of an increase in the tracking error (from 0.00103 to 0.00261rad in the nominal case and from 0.00117 to 0.00282rad in the disturbed case). This can be considered as a consequence of the trade off between the robustness and the accuracy. In Fig. IV.17(c) and Fig. IV.18(c), one can see that both SFCs succeed to recover from the effect of the additive disturbances especially the measurement noise.

In the following, final simulations are conducted and they are concerned with high speed tracking. To this aim, the trajectory to be followed is given by :

$$q_d = 0.6(1 - \cos(\pi t)) \qquad \text{(IV.3)}$$

The simulation results of the best SFC issued at the end of the first stage, the evolved SFC by the proposed EA, and the PD controller are illustrated in Fig. IV.19, Fig. IV.20 and Fig. IV.21, respectively.

At the nominal case, the PD controller produces a acceptable tracking error (less than 0.011 rad) as shown in Fig. IV.19 (b). Whereas in Fig. IV.19 (c)-(d), it is apparent that the behaviour of the PD controller is not satisfactory due to input voltage saturation and overshoots of angular position error occurring when the additive disturbances are applied suddenly; exactly speaking, in the time interval [2 5]sec the tracking error reach 0.12223 rad, and, in the interval [6 8] sec, it is about 0.067rad which is quite large. For either of the SFCs, the tracking performance is still in an excellent level at nominal or disturbed case. What concerns the sensitivity to the induced disturbances, the SFC evolved by the proposed EA presents the best response in control effort with a very acceptable tracking performance. Such a result can be considered very satisfactory if compared with those of the conventional PD control.

IV.7 Conclusion

In this chapter, we present an automatic design methodology of SFC based on an EA. The challenge behind this work is the preservation of the sectorial properties during the evolution. Our contribution in this direction is to adopt a hierarchical representation and a special population initialisation accompanied by a repairing operator. With the aid of a second stage, the SFC design is extended toward the robustness enhancement of the evolved SFC. Therefore, adding the second stage results in more robust SFC with a satisfactory tracking performance. Actually, the evolutionary process proposed at the second stage can be applied to any SFC already implemented to enhance its robustness. Simulations are conducted with a direct-drive DC motor, and the results show the effectiveness of the proposed EA in the design of the SFC and in its robustness enhancement.

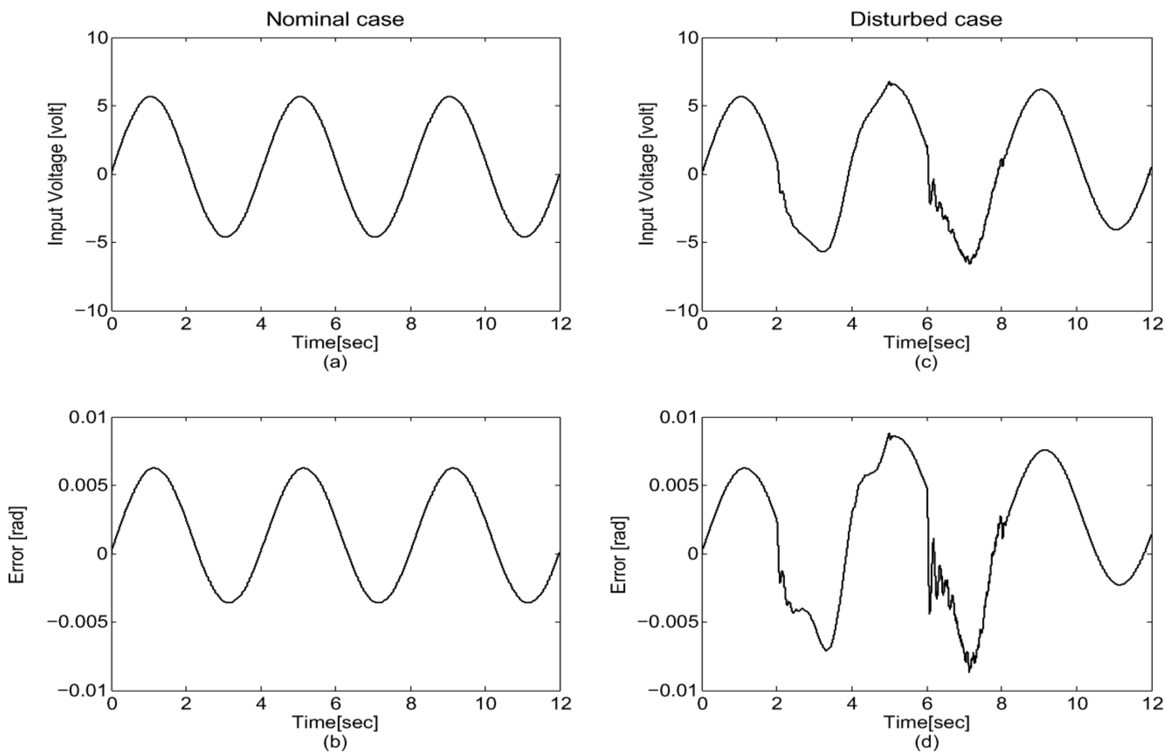


Fig. IV.16. Tracking performance and control activity of PD controller. (a)-(b) nominal case; (c)-(d) disturbed case.

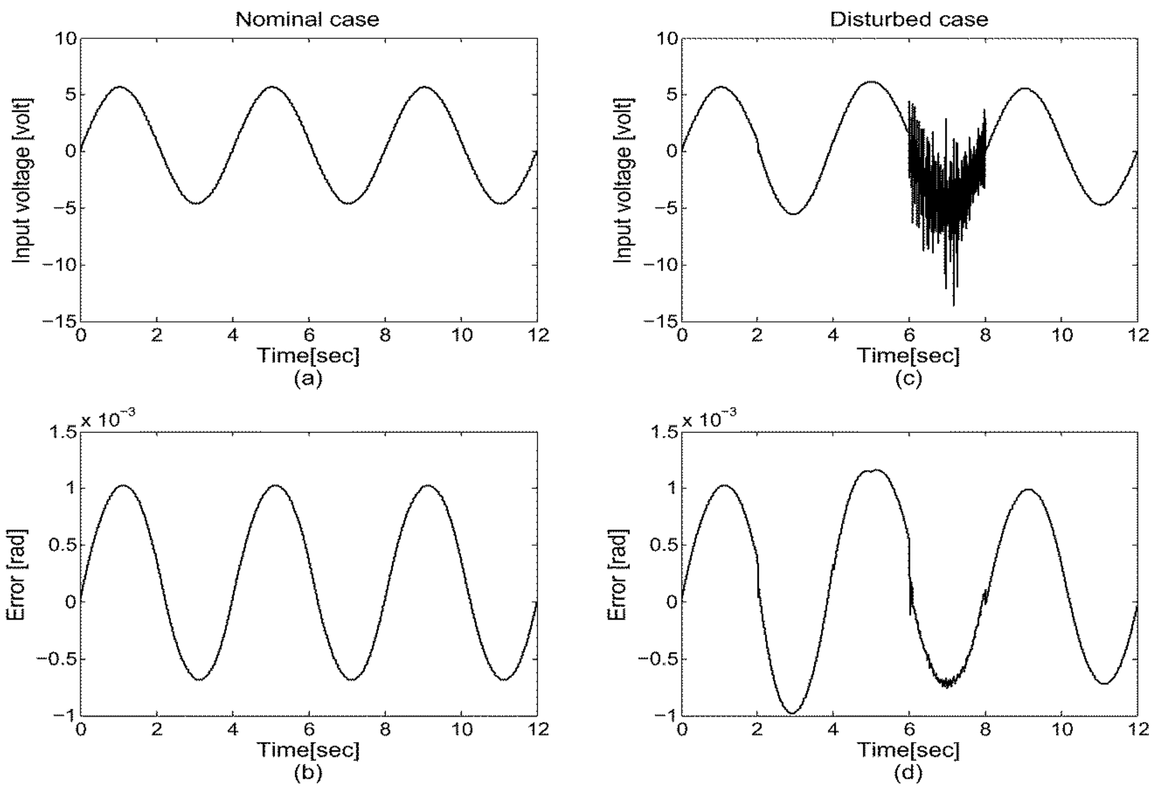


Fig. IV.17. Tracking performance and control activity of SFC evolved at the first stage of the proposed EA. (a)-(b) nominal case; (c)-(d) disturbed case.

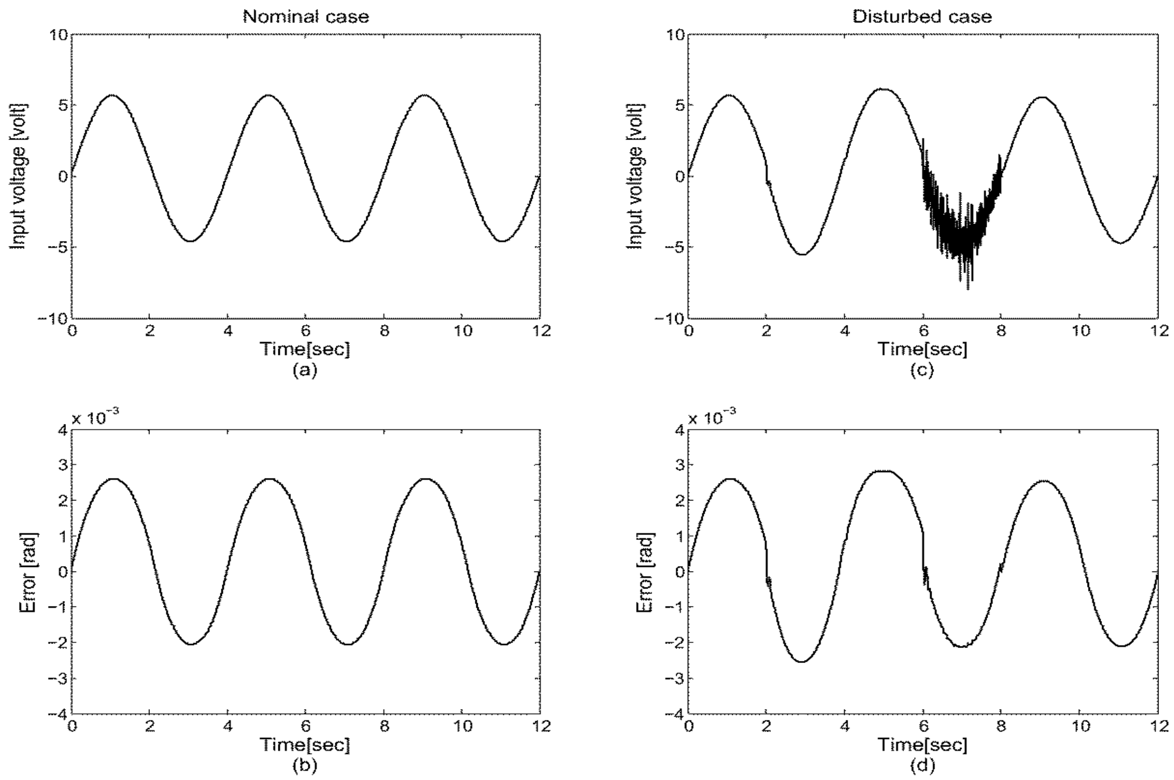


Fig. IV.18. Tracking performance and control activity of SFC evolved by the proposed EA. (a)-(b) nominal case; (c)-(d) disturbed case.

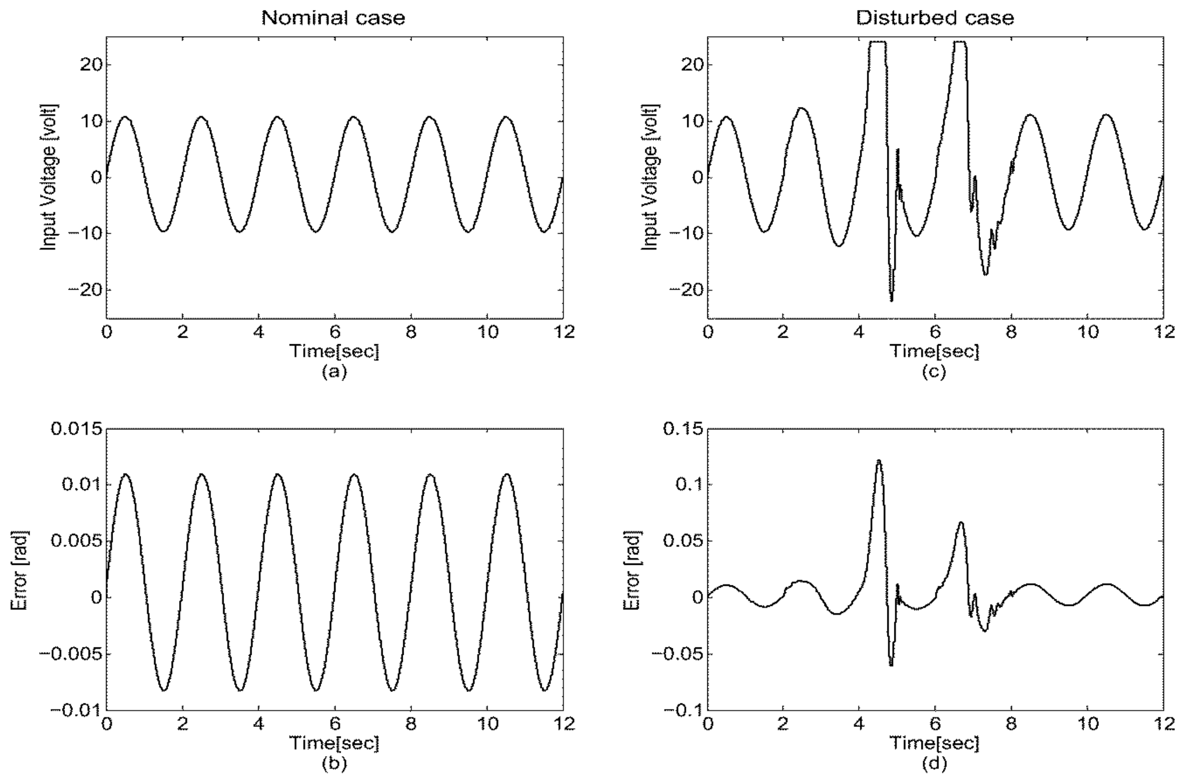


Fig. IV.19. Fast tracking performance and control activity of PD controller. (a)-(b) nominal case; (c)-(d) disturbed case.

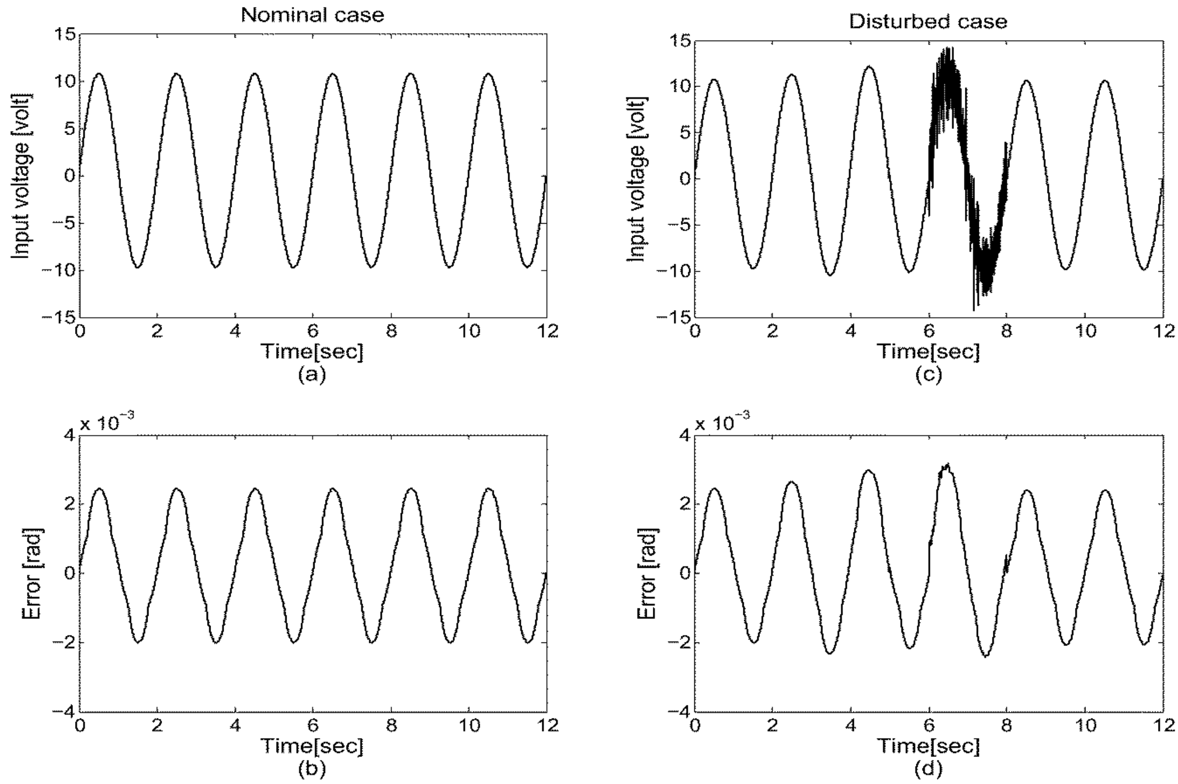


Fig. IV.20. Fast tracking performance and control activity of SFC evolved at the first stage of the proposed EA. (a)-(b) nominal case; (c)-(d) disturbed case.

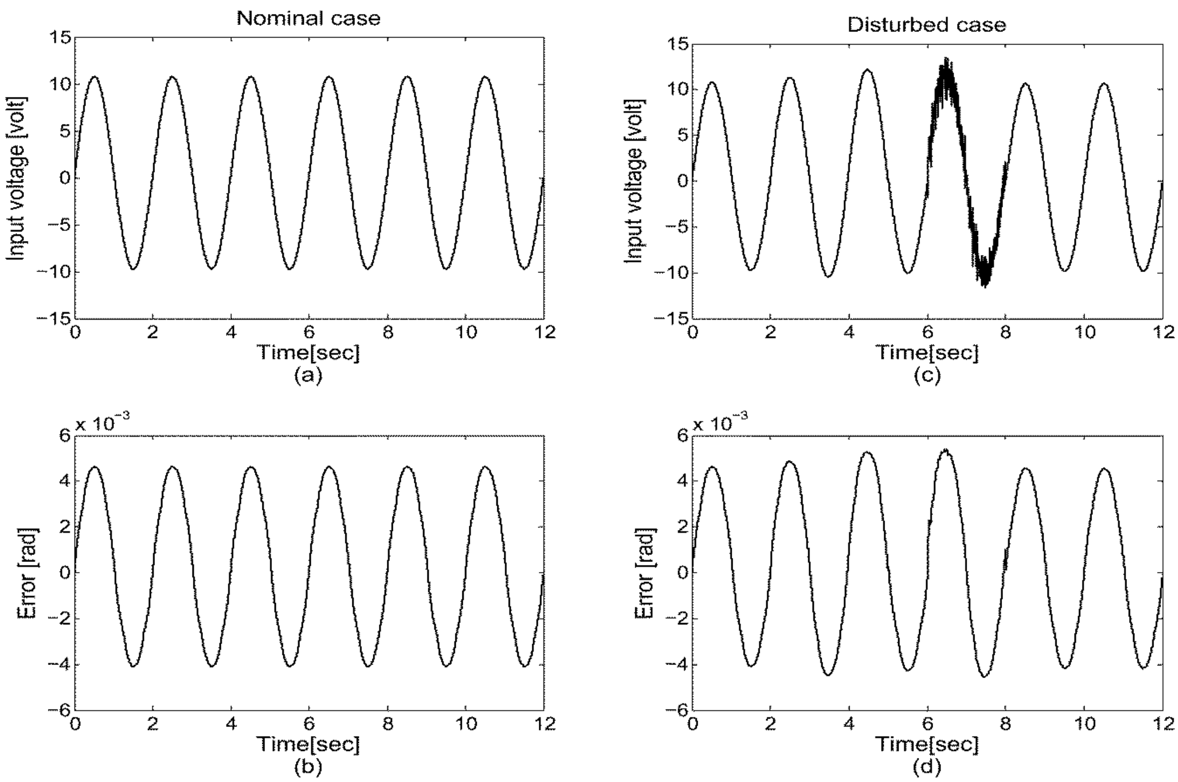


Fig. IV.21. Fast tracking performance and control activity of SFC evolved by the proposed EA. (a)-(b) nominal case; (c)-(d) disturbed case.

Conclusion

Fuzzy logic and evolutionary algorithms are powerful computational assets in the arsenal of Soft Computing. In this dissertation, the research and application of Soft Computing is focused on the use of evolutionary algorithm in the optimization of linguistic fuzzy controller. One of the most important limitation in this research is the loss of the semantic-based interpretability during the evolution. i.e., at the end of the evolution the input/output labels assumed initially are meaningless and the obtained fuzzy partitions are either non complete or indistinguishable. Another major limitation also associated to the automatic design of linguistic fuzzy controller is the chattering phenomenon in the evolved FLC. For what concern the EAs, it is noticed that these EAs have a tendency to quickly find the promising regions of the search space but suffer from excessively slow convergence before providing a reliable and accurate solution.

In this thesis, we have documented an evolutionary automatic design of interpretable linguistic FLC for direct-drive DC motor. The main characteristics of our evolutionary design technique that overcome the over mentioned limitations are:

1) the consideration of the variation of the control input as components of the fitness function to take into account the chattering phenomenon in the design phase;
2) the coding of the overlapping parameter in the chromosome to ensure the completeness of the fuzzy partition ;
3) the use of dependent searching ranges for MF parameters to ensure the evolving of valid interpretable FKBs;
4) the use of a bi-phase scheme to improve and accelerate the accuracy of the designed FLC.

Simulations were conducted to validate the usefulness of the proposed bi-phase IEA. The results obtained suggest that the proposed bi-phase scheme does its job of accelerating the IEA convergence and improving the best fitness function. They also show the excellent dynamic performance of the evolved FLC for different operating conditions which reflects the nonlinear character of the designed controller.

We have also addressed the automatic evolutionary design of SFC. The major challenge in this problem and the first to arise is the accommodation of the sectorial properties in the evolutionary process. These properties are related to the FRB and the FDB. They are accommodated through the population initialisation and in the parameterization and chromosome representation. With these proposed strategies, the monotonicity property of the FRB remains not necessarily ensured during the design process. Accordingly, a repairing operator is proposed.

As mentioned earlier, the FLCs are known to be robust enough to tolerate plant uncertainties. In the sake of widening its operating conditions, we have proposed a robust optimization design methodology of FLC based on two stage EA. Robust design search accommodating presence of uncertainty is possible in this algorithm through the second stage where the robustness and the accuracy criterion are considered simultaneously. Specifically, the robustness to be enhanced is toward load disturbances, motor torque disturbance, load disturbance, friction disturbance, and measurement noise. The enhanced SFC with the proposed EA was found to provide a very satisfactory performances under a very sever operating conditions and to recover successfully from the effects of the additive disturbances.

As prospect for future research, we suggest to apply the evolutionary process proposed at the second stage to the SFC or any FLC already implemented to enhance its robustness and investigate the robustness enhancement against other uncertainty sources. Another possible prospect is the use of Pareto-based multi-objective approach in the second stage. This type of approach is characterized by a large number of trade-off or non-dominated fuzzy controller that could be found simultaneously (i.e., in a single run). The future research also includes developing evolutionary technique around the idea of overlapping encoding in the chromosome in the optimization of Type 2 fuzzy systems. Furthermore, the researcher intends to extend the application of the proposed design methodologies to the fuzzy system modelling.

References

[1] L.A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338-352, 1965.

[2] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, pp. 1-13, 1975.

[3] R. Palm, "Sliding mode fuzzy control ," in *Proceedings of the 1st IEEE International Conference on Fuzzy Systems*, San Diego, CA, 1992, pp. 519-526.

[4] S. Kawaji and N. N. Matsunaga, "Fuzzy control of VSS-type and its robustness," in *Fuzzy Control Systems*, A. Kandel and G. Langholz, Eds. Boca Raton, FL: CRC, 1994, pp. 226-242.

[5] H. Megherbi, N. Megherbi, A. C. Megherbi, and K. Benmahammed, "Evolutionary Optimization of Robust and Chattering-Free Mamdani Type Fuzzy Controller," *Journal Courrier de Savoir*, no. 17, pp. 77-85, 2013.

[6] H. Megherbi, A. C. Megherbi, N. Megherbi, and K. Benmahammed, "Design of a direct fuzzy controller for robot manipulators using bi-phase integer-coded genetic algorithm," in *7th International Symposium on Programming and Systems: ISPS-2005*, Algiers, 9 May 2005.

[7] H. Megherbi, A. C. Megherbi, N. Megherbi, and K. Benmahammed, "Design and Robustness Enhancement of Sectorial Fuzzy Controller via Evolutionary Algorithm," *Journal of Intelligent and Fuzzy Systems*.

[8] H. Megherbi, A. C. Megherbi, K. Benmahammed, and A. Hamzaoui, "H. Megherbi, A. C. Megherbi, K. Benmahammed, A. Hamzaoui, On Enhancing Robustness of Fuzzy Logic Controller Designed by an Evolutionary Algorithm," in *International Conference on Computer Systems and Information Technology (ICSIT'05)*, Algiers, july 19-21, 2005.

[9] H. Megherbi, A. C. Megherbi, N. Megherbi, and K. Benmahamed, "On Enhancing Robustness of an Evolutionary Fuzzy Tracking Controller," *International Journal of Information Technology*, vol. 3, no. 4, pp. 277-284, 2007.

[10] B. Kosko and S. Isaka, "La logique Floue et ses Applications," *Pour la Science*, no. 191, 1993.

[11] D. Dubois and H. Parade, *Fuzzy Sets and Systems*.: Academic Press, 1980.

[12] C. C. Lee, "Fuzzy Logic in Control System: Fuzzy Logic Controller, Part I," *IEEE Transactions on System, Man and Cybernetics*, vol. 20, no. 2, pp. 404-418, 1990.

[13] C. C. Lee, "Fuzzy Logic in Control System : Fuzzy Logic Controller, Part II," *IEEE Transactions on System, Man and Cybernetics*, vol. 20, no. 2, pp. 419-435, 1990.

[14] J. M. Mendel, "Fuzzy Logic System for Engineering: A Tutorial," *Proceedings of IEEE*, vol. 83, no. 3, 1995.

[15] Masahiro Inuiguchi, Shoji Hirano, and Shusaku Tsumoto, Eds., *Rough Sets Theory and Granular Computing*.: Physica-Verlag, 2003.

[16] L. A. Zadeh, "Outline of a New Approach to the Analysis of a Complex Systems and Decision Processes," *IEEE Transactions on Systems, Man and cybernetics*, vol. 3, no. 1, pp. 28-44, 1973.

[17] L. A. Zadeh, "The Concept of a Linguistic Variable and its Application to Approximate Reasoning - Part 3," *Information Sciences*, vol. 9, pp. 43-80, 1976.

[18] J. J. Buckley and E. Eslami, *An Introduction to Fuzzy Logic and Fuzzy Sets*. New York: Physica-Verlag, 2002.

[19] L.A. Zadeh, "A Theory of Approximate Reasoning," *Machine Intelligence*, vol. 9, pp. 149-149, 1979.

[20] L. A. Zadeh, "Fuzzy Logic," *IEEE Computer Magazine*, 1988.

[21] L.-X. Wang, *A Course in Fuzzy Systems and Control*. New Jersy, USA: Prentice-Hall, 1997.

[22] H.T. Nguyen, N.R. Prasad, C.L. Walker, and E.A. Walker, *A first course in Fuzzy and neural control*.: Chapman & Hall/CRC, 2003.

[23] T. Jiang and Y. Li, "Generalized defuzzification strategies and their parameter learning procedures," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 1, pp. 64-71, 1996.

[24] R.R. Yager and D.P. Filev, "A Generalized Defuzzification Method under BAD Distributions," *International Journal of Intelligent Systems*, vol. 6, pp. 689-697, 1991.

[25] R.R. Yager and D.P. Filev, "SLIDE : A Simple Adaptive Defuzzification Method," *IEEE trans. Fuzzy systems*, vol. 1, no. 1, pp. 69-78, 1993.

[26] I. Baturone, A. Jimenez-Fernandez, C. Barriga, D. R. Lopez, and S. Sanchez-Solano, *Microelectronic Design of Fuzzy Logic-Based Systems*. Boca Raton, FL: CRC Press, 2000.

[27] P.J. MacVicar-Whelan, "Fuzzy sets for man-machine interaction," *International Journal of Man-Machine Studies*, vol. 8, pp. 687-697, 1976.

[28] R. R. Yager and D. P. Filev, *Essentials of fuzzy modeling and control*. New York, USA: Wiley-Interscience, 1994.

[29] F. Cheong and R. Lai, "Constraining the optimization of a fuzzy logic controller using an enhanced genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 30, no. 1, pp. 31-46, 2000.

[30] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proceedings IEE*, vol. 121, no. 12, pp. 1585-1588, 1974.

[31] C. Pena-Reyes, "Coevolutionary Fuzzy Modeling," École Polytechnique Fédérale de Lausanne - EPFL, Lausanne, PhD thesis 2002.

[32] Lindskog P., "fuzzy identification from a grey box modeling point of view," in *Fuzzy model identification*, H. Hellendoorn and D. Driankov, Eds., 1997, pp. 3-50.

[33] M. Setnes, "Supervised fuzzy clustering for rule extraction," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 4, pp. 416–424, 2000.

[34] M. O. Efe and O. Kaynak, "On stabilization of gradient-based training strategies for computationally intelligent systems," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 564-575, 2000.

[35] A. Lo Schiavo and A. M. Luciano, "Powerful and flexible fuzzy algorithm for nonlinear dynamic system identification," *IEEE Transaction on Fuzzy Systems*, vol. 9, no. 6, pp. 828-835, 2001.

[36] M. O. Efe and O. Kaynak, "A novel optimization procedure for training of fuzzy inference systems by combining variable structure systems technique and levenberg-rmarquardt algorithm," *Fuzzy Sets and Systems*, vol. 122, no. 1, pp. 153-165, 2001.

[37] B.-S. Chen, S.-C. Feng, and K.-C. Wang, "Traffic modeling, prediction, and congestion control for high-speed networks: A fuzzy AR approach," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 491-508, 2000.

[38] E. Mamdani and T. Procyk, "A linguistic selforganizing proccess controller," *Automatica*, vol. 15, pp. 15–30, 1979.

[39] T. Yamazaki, "An improved algorithm for a self-organizing controller and its experimental analysis," Queen Mary College, Deptement of Electrical and electronic engineering, London, PhD. Thesis 1982.

[40] S. Shao, "Fuzzy self-organizing controller and its application for dynamic processes," *Fuzzy Sets and Systems*, vol. 26, no. 2, pp. 151-164, 1988.

[41] J. Layne and K. Passino, "Fuzzy model reference learning control," in *IEEE Conference on Control*

Applications, Dayton, OH, 1992, pp. 686-691.

[42] J. Layne and K. Passino, "Fuzzy model reference learning control for cargo ship steering," *IEEE Control Systems Magazine*, vol. 13, no. 6, pp. 23-34, 1993.

[43] B. Kovai and S. Bogdan, *Fuzzy Controller Design - Theory and Applications.*.: CRC Press - Taylor and Francis., 2006.

[44] J.-S.R. Jang, "Self-learning fuzzy controllers based on temporal backpropagation," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 714-723, 1992.

[45] M. Brown and C. Harris, *Neurofuzzy Adaptive Modelling and Control*. Englewood Cliffs, NJ: Prentice Hall, 1994.

[46] G.A. Carpenter and S. Grossberg, "Adaptive Resonance Theory," in *The Handbook of Brain Theory and Neural Networks*, Second Edition ed., M. A. Arbib, Ed. Cambridge, MA: MIT Press, 2003, pp. 87-90.

[47] J.-S.R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference Systems," *IEEE Transactions on Systems, Man & Cybernetics*, vol. 23, pp. 665-685, 1993.

[48] S. K. Halgamuge and M. Glesner, "Neural networks in designing fuzzy systems for real world applications," *Fuzzy Sets and Systems*, vol. 65, pp. 1-12, 1994.

[49] N Tschichold-Gürman, "RuleNet -- A New Knowledge--Based Artificial Neural Network Model with Application Examples in Robotics," ETH Zürich, PhD thesis 1996.

[50] H.R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Transactions on Neural Networks* , vol. 3 , pp. 724-740, 1992.

[51] D. Nauck, F. Klawonn, and R. Kruse, *Foundations of Neuro--Fuzzy Systems*. Chichester: Wiley, 1996.

[52] C. Karr, "Genetic algorithms for fuzzy controllers," *AI Expert*, vol. 6, no. 2, pp. 26-33, 1991.

[53] M. Valenzuela-Rendon, "The fuzzy classifier system: A classifier system for continuously varying variables," in *Proceedings of 4th International Conference on Genetic Algorithms*, 1991, pp. 346-353.

[54] P. Thrift, "Fuzzy logic synthesis with genetic algorithms," in *Proceedings of 4th International Conference on Genetic Algorithms*, 1991, pp. 509-513.

[55] H. M. Omar and M. A. Abido, "Enhancement of Integrated Fuzzy-Based Guidance Law by Tabu Search," *Arabian Journal for Science and Engineering*, vol. 37, no. 7, pp. 2035-2046, 2012.

[56] M. Denna, G. Mauri, and A.M. Zanaboni, "Learning fuzzy rules with tabu search-an application to control," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 3, pp. 295-318, 1999.

[57] N. R. Cázarez-Castro, L. T. Aguilar, and O. Castillo, "Fuzzy logic control with genetic membership function parameters optimization for the output regulation of a servomechanism with nonlinear backlash," *Expert Syst. Appl.* , vol. 37, no. 6, pp. 4368-4378, 2010.

[58] O. Castillo, R. Martinez-Marroquin, P. Melin, F. Valdez, and J. Soria, "Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot," *Information Sciences*, vol. 192, pp. 19-38, 2012.

[59] J.M. Garibaldi and E. C. Ifeachor, "Application of Simulated Annealing Fuzzy Model Tuning to Umbilical Cord Acid-Base Interpretation," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 1, pp. 72-84, 1999.

[60] Z. Bingül and O. Karahan, "A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control ," *Expert Systems with Applications* , vol. 38, no. 1, pp. 1017–1031, 2011.

[61] J. Cassillas, O. Cordon, I. F. Viana, and F. Herrera, "Learning cooperative linguistic rules using the best-worst ant system algorithm," *International Journal of Intelligent Systems*, vol. 20, pp. 433-452,

2005.

[62] D. K. Chaturvedi, *Soft Computing: Techniques and Its Applications in Electrical Engineering*. Berlin Heidelberg: Springer-Verlag , 2008.

[63] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing ,Natural Computing Series*.: Springer, 2008.

[64] K.A. De Jong, *Evolutionary computation: a unified approach*. Cambridge MA: MIT Press, 2006.

[65] P. Hingston, L. Barone, and Z. Michalewicz, Eds., *Design by Evolution, Natural Computing Series*.: Springer, 2008.

[66] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. MA: Addison-Wesley, 1991.

[67] E. Sanchez, G. Squillero, and A. Tonda, *Industrial Applications of Evolutionary Algorithms*. Berlin Heidelberg: Springer-Verlag, 2012.

[68] R. Chiong, T. Weise, and Z. Michalewicz, Eds., *Variants of Evolutionary Algorithms for Real-World Applications*. Berlin Heidelberg: Springer, 2012.

[69] D. Dasgupta and Z. Michalewicz, Eds., *Evolutionary Algorithms in Engineering Applications*. Berlin: Springer-Verlag, 1997.

[70] H. Iba and N. Noman, *New frontier in evolutionary algorithms : theory and applications*.: Imperial College Press, 2011.

[71] J.H. Holland, *Adaptation in Natural and Artificial Systems.* Ann Arbor: University of Michigan Press, 1975.

[72] L. D. Davis, *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991.

[73] C. A. C. Coello, "Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art," *Computer Methods in Applied Mechanics and Engineering* , vol. 191, no. 11-12, pp. 1245–1287, 2002.

[74] Z. Michalewicz, "A survey of constraint handling techniques in evolutionary computation methods," in *Proceedings of the 4th Annual Conference on Evolutionary Programming*, 1995 , pp. 135-155.

[75] Z. Michalewicz, *Genetic Algorithms+ Data Structures= Evolution Programs*, 3rd ed. Berlin Heidelberg: Springer, 1996.

[76] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65-85, 1994.

[77] T. Baeck and D. B. Fogel, Eds., *Evolutionary Computation: Advanced Algorithms and Operators*.: Institute of Physics Publishing, 2000.

[78] X. Yu and M. Gen, *Introduction to Evolutionary Algorithms*. London, UK: Springer, 2010.

[79] G. Syswerda, "Uniform crossover in genetic algorithms," in *3th Int. Conf. Genetic Algorithms*, Sam Mateo, CA, 1989, pp. 2-9.

[80] J. E. Baker, "Reducing Bias And Inefficiency in the Selection Algorithm," in *Proceedings of the second international Conference on Genetic Algorithms*, Hillsdale, NJ, 1987, pp. 14-21.

[81] J.E. Gentle, *Random Number Generation and Monte Carlo Methods*. NewYork: Springer-Verlag, 1998.

[82] R.S Wikramaratna, "ACORN—a new method for generating sequences of uniformly distributed pseudo-random numbers," *J. Computat. Phys.*, vol. 83, no. 1, pp. 16–31, 1989.

[83] B. Tuffin, "On the use of low discrepancy sequences in Monte Carlo methods," *Monte Carlo Methods Applications*, vol. 2, no. 4, pp. 295–320, 1996.

[84] C.S. Hiremath and R.R. Hill, "New greedy heuristics for the multiple-choice multidi-mensional knapsack problem," *International Journal of Operational Research*, vol. 2, no. 4, pp. 495–512, 2007.

[85] Z. Michalewicz and D.B. Fogel, *How to Solve It: Modern Heuristics* , 2nd ed. Berlin: Springer–Verlag, 2004.

[86] R. Precupa and H. Hellendoorn, "A survey on industrial applications of fuzzy control," *Computers in Industry*, vol. 62, no. 3, pp. 213-226, 2011.

[87] Y. Bi, H. Zhuang, and D. Wang, *Advanced Fuzzy Logic Technologies in Industrial Applications*. London: Springer, 2006.

[88] D. T. Pham and D. Karaboga, "Optimum design of fuzzy logic controllers using genetic algorithms," *Journal of Systems Engineering*, vol. 1, pp. 114-118, 1991.

[89] C. L. Karr and E. J. Gentry, "Fuzzy control of pH using genetic algorithms," *IEEE Trans. Fuzzy Systems*, vol. 1, pp. 46-53, 1993.

[90] M. Galluzzo, V. Cappellani, and U. Garofalo, "Fuzzy Control of pH using NAL," *Intrernational Journal of Approximate Reasoning*, vol. 5, no. 6, pp. 505-519, 1991.

[91] A. Varsek, T. Urbacic, and B. Filipic, "Genetic Algorithms in Controller Design and Tuning," *IEEE Trans. Systems Man and Cybernetics*, vol. 23, no. 5, pp. 1330-1339, 1993.

[92] J. Kinzel, F. Klawonn, and R. Kruse, "Modifications of Genetic Algorithms for Designing and Optimizing Fuzzy Controllers," in *Proceedings of IEEE Conference on Evolutionary Computation*, Orlando, 1994, pp. 28-33.

[93] A. Homaifar and E. McCormick, "Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms," *IEEE trans. Fuzzy Systems*, vol. 3, no. 2, pp. 129-139, 1995.

[94] D. A. Linkens and H. O. Nyongesa, "Genetic Algorithms for Fuzzy Control. Part I : Offline system development and Application," *IEE Proceedings of Control Theory and Applications*, vol. 142, no. 3, 1995.

[95] Y. Jin, W. Von Seelen, and B. Sendhoff, "On generating FC3 fuzzy rule systems from data using evolution strategies," *IEEE Transactions on Systems Man and Cybernetics*, vol. 29, pp. 829-845, 1999.

[96] M. Setnes, R. Babuska, U. Kaymak, and H. R. Van Nauta Lemke, "Similarity measures in fuzzy rule base simplification," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 28, pp. 376-386, 1998.

[97] J. V. De Oliveira, "Semantic constraints for membership function optimization," *IEEE Transactions Systems, Man, and Cybernitics – Part A: Systems and Humans*, vol. 29, pp. 128-138, 1999.

[98] M. J. Gacto, R. Alcala, and F. Herrera, "Integration of an index to preserve the semantic interpretability in the multi-objective evolutionary rule selection and tuning of linguistic fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 515-531, 2010.

[99] C. Mencar, C. Castiello, R. Cannone, and A. M. Fanelli, "Interpretability assessment of fuzzy knowledge bases: A cointension based approach," *International Journal of Approximate Reasoning*, vol. 52, no. 4, pp. 501-518, 2011.

[100] P. Pulkkinen, J. Hytönen, and H. Koivisto, "Developing a bioaerosol detector using hybrid genetic fuzzy systems," *Engineering Applications of Artificial Intelligence*, vol. 21, pp. 1330-1346, 2008.

[101] P. Pulkkinen, "A multiobjective genetic fuzzy system for obtaining compact and accurate fuzzy classifiers with transparent fuzzy partitions," in *Proceedings of the 8th International Conference Machine Learning and Applications*, Florida, 2009, pp. 89-94.

[102] P. Pulkkinen and H. Koivisto, "A Dynamically Constrained Multiobjective Genetic Fuzzy System for Regression Problems," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 161-177, 2010.

[103] C. Mencar, G. Castellano, and A. M. Fanelli, "Distinguishability quantification of fuzzy sets," *Information Sciences*, vol. 177, pp. 130-149, 2007.

[104] L.-X. Wang and J. M. Mendel, "Fuzzy Basis Function, Universal Approximation, and Orthogonal Least-Squares Learning," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 807-814, 1992.

[105] A. E. Eiben and C. Schippers, "On evolutionary exploration and exploitation," *Fundamenta Informaticae*, vol. 35, pp. 35-50, 1998.

[106] T. Baeck, "Selective pressure in evolutionary algorithms: A characterization of selection mechanisms," in *Proceedings of the 1st Conference on Evolutionary Computing*, 1994, pp. 57-62.

[107] E. Galvan-Lopez, J. Mcdermott, M. Oneill, and A. Brabazon, "Towards an understanding of locality in genetic programming," *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pp. 901-908, 2010.

[108] F. Herrera, "Genetic Fuzzy Systems: Taxonomy, Current Research Trends and Prospects," *Evolutionary Intelligence*, vol. 1, pp. 27-46, 2008.

[109] M. Antonelli, P. Ducange, B. Lazzerini, and F. Marcelloni, "Learning knowledge bases of multi-objective evolutionary fuzzy systems by simultaneously optimizing accuracy, complexity and partition integrity," *Soft Computing: A Fusion of Foundations, Methodologies and Applications*, vol. 15, no. 12, pp. 2335-2354, 2011.

[110] R. Alcala, Y. Nojima, F. Herrera, and H. Ishibuchi, "Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions," *Soft Computing: A Fusion of Foundations, Methodologies and Applications*, vol. 15, no. 12, pp. 2303-2318, 2011.

[111] L. Sanchez, J. Otero, and I. Couso, "Obtaining Linguistic Fuzzy Rule-based Regression Models from Imprecise Data with Multiobjective Genetic Algorithms," *Soft Computing: A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 3, pp. 467-479, 2009.

[112] M.J. Gacto, R. Alcala, and F. Herrera, "Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems," *Soft Computing: A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 3, pp. 419-436, 2009.

[113] D. G. Stavrakoudis, G. N. Galidaki, I. Z. Gitas, and J. B. Theocharis, "Reducing the complexity of genetic fuzzy classifiers in highly-dimensional classification problems," *International Journal of Computational Intelligence Systems*, vol. 5, no. 2, pp. 254-275, 2012.

[114] A. Nebot, F. Mugica, F. Castro, and J. Acosta, "Genetic learning of fuzzy parameters in predictive and decision support modelling," *International Journal of Computational Intelligence Systems*, vol. 5, no. 2, pp. 387-402, 2012.

[115] G. Calcev, "Some Remarks On The Stability Of Mamdani Fuzzy Control Systems," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 3, pp. 436-442, 1998.

[116] M. Sidi, *Design of robust control systems from classical to modern practical approaches*. Florida, USA: Krieger Publishing Co., 2001.

[117] F. Lin, *Robust Control Design: An Optimal Control Approach*.: Wiley--Interscience, 2007.

[118] C.-C. Tsui, *Robust Control System Design: Advanced State Space Techniques*, 2nd ed.: CRC Press, 2003.

[119] L. R. Ray and R. F. Stengel, "Stochastic robustness of linear-time-invariant control systems," *IEEE Transactions on Automatic Control*, vol. 36, pp. 82-87, 1991.

[120] P. J. C. Branco and J. A. Dente, "An experiment in automatic modeling an electrical drive system using fuzzy logic," *IEEE Transactions on Systems, Man and Cybernetics - part: C*, vol. 28, pp. 254-262, 1998.

[121] G. C. Mouzouris and J. M. Mendel, "Dynamic nonsingleton fuzzy logic systems for nonlinear modeling," *IEEE Transactions Fuzzy Systems*, vol. 5, no. 2, pp. 199-208, 1997.

[122] T. J. Ross, *Fuzzy Logic with Engineering Applications*, 3rd ed. Oxford: Wiley-Blackwell, 2009.

[123] R. Jager, "Fuzzy Logic in Control," Delft University of Technology, Department of Electrical Engineering, Control Laboratory, Delft, The Netherlands, Ph.D. thesis 1995.

[124] H. T. Nguyen, C. W. Tao, and W. E. Thompsom, "An emprerical study of robustness of fuzzy systems," in *Proceedings of 2nd IEEE International Conference on fuzzy systems*, 1993, pp. 1340-1345.

[125] D. Hissel, P. Maussion, and J. Faucher, "On evaluating robustness of fuzzy logic controllers through Taguchi methodology," *In Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, pp. 17-22, 1998.

[126] T. Leephakpreeda, "H infinity stability robustness of fuzzy control systems," *Automatica*, vol. 35, no. 8, pp. 1467-1470, 1999.

[127] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*.: Princeton Series in Applied Mathematics, Princeton University Press, 2009.

[128] H.-G. Beyer and B. Sendhoff, "Robust optimization -- A comprehensive survey," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, pp. 3190-3218, 2007.

[129] B. Forouraghi, "A genetic algorithm for multiobjective robust design," *Applied Intelligence*, vol. 12, pp. 151-161, 2000.

[130] S. Tsutsui and A. Ghosh., "Genetic algorithms with a robust solution searching scheme," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 3, pp. 201-208, 1997.

[131] A. Gaspar-Cunha and J. Covas, "Robustness in Multi-Objective Optimization using Evolutionary Algorithms," *Computational Optimization and Applications*, vol. 39, no. 1, pp. 75-96, 2008.

[132] K. Deb and H. Gupta, "Searching for Robust Pareto-Optimal Solutions in Multi-objective Optimization," *Evolutionary Multi-Criterion Optimization Lecture Notes in Computer Science*, vol. 3410, pp. 150-164, 2005.

[133] J. Hu and S. Mehrotra, "Robust and Stochastically Weighted Multiobjective Optimization Models and Reformulations," *Operations Research*, vol. 69, no. 4, pp. 936-953, 2012.

[134] J.-M. Yanga, C.-J. Linb, and C.-Y. Kaob, "A robust evolutionary algorithm for global optimization," *Engineering Optimization*, vol. 34, no. 5, pp. 405-425, 2002.

[135] S. Purwar, I. N. Kar, and A. N. Jha, "Adaptive Control of Robot Manipulators using Fuzzy Logic System Under Actuator Constraints," in *Proceedings of IEEE International Conference on Fuzzy Systems*, Budapest, Hungary, 2004, pp. 1449-1454.

[136] V. Santibaez, R. Kelly, and M. A. Llama, "Global Asymptotic Stability of Tracking Sectorial Fuzzy Controller for robot manipulators," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, vol. 34, no. 1, pp. 710-718, 2004.