Investigations into New Algorithms for Self-Organising Fuzzy Logic Control using Type-1 and Type-2 Fuzzy Sets

Mohamed Ehtiawesh

Supervised by Prof Mahdi Mahfouf

The University Of Sheffield.

Submitted to the University of Sheffield for the degree of Doctor of Philosophy in the faculty of Engineering

Department of Automatic Control and Systems Engineering

University of Sheffield

April 2016

Acknowledgments

I wish to express my gratitude to all the people who supported and gave me help throughout the years of my studies.

I would like to express my sincere thanks to my first supervisor, Prof Mahdi Mahfouf, who offered me guidance and encouragement whilst giving me the room to work in my own way. I thank him deeply for his generosity and kindness, and for everything he has done to make the completion of this project possible. I could not ask for better qualified or friendlier supervisor.

Also, I am grateful to all the ACSE research students in the Intelligent Systems Laboratory for the interesting, constructive and enlightening discussions, from which I have benefited immensely.

Finally, I would like to take this opportunity to thank my family for their continuous encouragement and support.

Abstract

The number of applications of intelligent control systems has grown significantly over the last few decades, and today they are used in various challenging industrial application domains, where they provide particularly useful solutions. The term ‘intelligent controllers’ describes a field where control approaches are represented by mechanisms similar to those used by the human brain. These characteristics include, for example, learning, modification, adaption, effective working with high levels of uncertainty and coping with large amounts of data.

Intelligent control systems are particularly useful for complex systems such as biomedical and chemical plants, which are expected to work under optimal conditions. A good example of an intelligent controller is the so called Self-Organising Fuzzy Logic Control (SOFLC) proposed by Procyk and Mamdani in the late 1970s. The SOFLC scheme involves a control policy that allows its structure to be adapted based on the environment in which it operates.

The SOFLC combines a conventional fuzzy logic controller with a supervisory layer which monitors and regulates the performance of the system. In this thesis, new architectures are proposed for single input single output (SISO) and multi-input multi-output (MIMO) structures to improve on the original SISO SOFLC design in terms of performance and robustness, as well as extend the analysis and design issues relating to such algorithms to the MIMO case using hybrid approaches. The work proposed in this thesis includes: 1. A new development of type-1 and type-2 Self-Organising Fuzzy Logic Control with a Dynamic Supervisory Layer (SOFLC-DSL) for the SISO case: In this part of the thesis, the work is mainly focused on designing a sophisticated SOFLC algorithm by combining a type-1 fuzzy system with a new Particle Swarm Optimisation (PSO) algorithm, so as to make the SOFLC scheme more flexible and effective in terms of responding to changes in the process to be controlled or the environment surrounding it. A new on-line PSO algorithm is developed by using the idea of credit assignment and fitness estimation to allow the optimisation of the consequent parts of the performance index (PI) table on-line. The proposed scheme is tested on a non-linear and uncertain

Muscle Relaxation Model. Computer results demonstrate that the proposed algorithm achieve satisfactory performance, and is superior to the standard SOFLC scheme. In order to enhance the capabilities of the controller to deal with environments where the level of uncertainties and noise are high, both interval and zSlice type-2 fuzzy sets are deployed. Simulation results show that the performance of the SOFLC-DSL algorithm improves in terms of set-point tracking properties and the smoothness of the generated control signals. 2. A new extension of the SOFLC-DSL to the multivariable case: The proposed SOFLC-DSL algorithms are applied as the dominating controllers within multivariable control architectures. In order to deal with the effects of interactions between the input and output channels, both the relative array gain matrix as well as a linguistic switching mode compensator are considered. The proposed algorithms are tested on a 2×2 drug dynamic process, and the results show they have good control abilities in terms of maintaining the desired set-points with smooth control effort, as well as in handling the interaction between different control channels.

Table of contents

Chapter 1 – Introduction 1
1.1 Overview 1
1.2 Type-1 and Type-2 Fuzzy Logic Controllers 2
1.3 Bio-inspired Optimisation Algorithms 4
1.4 Supervisory Intelligent Control Systems 5
1.5 Thesis Overview 6
Chapter 2- Background 9
2.1 Introduction 9
2.2 Type-1 Fuzzy Sets and Systems 10
2.2.1 Type-1 fuzzy sets 10
2.2.2 Operations on type-1 fuzzy sets 11
2.2.3 Type-1 fuzzy logic systems 12
2.3 Type-2 Fuzzy Sets and Systems 16
2.3.1 General type-2 fuzzy sets 18
2.3.2 Interval Type-2 fuzzy sets 20
2.3.3 Type-2 fuzzy set operations 21
2.3.4 Type-2 fuzzy logic systems 22
2.4 Fuzzy control 26
2.5 Self-organising fuzzy logic controller 28
2.5.1 Description of the controller 29
2.5.2 Controller parameters setting 32
2.5.3 Adding and deleting control rules 33
2.6 Supervisory intelligent Control Systems 34
2.6.1 Fuzzy controllers using performance index tables 35
2.6.2 Fuzzy controllers using performance measure functions 36
2.6.3 Fuzzy controllers using fitness function 37
2.7 Summary: 38
Chapter 3- Self-Organising Fuzzy Logic Control with a Dynamic Supervisory Layer 40
3.1 Introduction 40
3.2 Particle Swarm Optimisation 41
3.2.1 Introduction 41
3.2.2 Classical PSO algorithm 42

3.2.3 On-line implementation of PSO 43
3.3 Self-Organising Fuzzy Logic Control with a Dynamic Supervisory Layer 45
3.3.1 The PSO process encoding 45
3.3.2 The on-line PSO algorithm operations 46
3.3.3 A Summary of the SOFLC-DSL Algorithm 50
3.4 Simulation on a Muscle Relaxant Model 51
3.4.1 Sensitivity of the SOFLC-DSL to sudden disturbances 53
3.4.2 Controller sensitivity to scaling factors 55
3.4.3 System robustness to model parameter variations 60
3.4.4 Robustness in the stochastic case 62
3.5 Summary 64
Chapter 4 - Type-2 Fuzzy Sets for the Self-Organising Fuzzy Logic Control with a Dynamic Supervisory Layer 66
4.1 Introduction 66
4.2 Intelligent Control of the Depth of Anaesthesia 67
4.3 Particle Swarm Optimization of Interval Type-2 Fuzzy Systems 68
4.4 Comparisons of Type-1 and Type-2 Fuzzy Logic Systems 70
4.5 Introduction to zSlice General Type-2 Fuzzy Sets 73
4.5.1 From interval type-2 fuzzy systems to zSlices 74
4.5.2 Mamdani zSlice-based general type-2 fuzzy logic control 75
4.6 Type-2 SOFLC-DSL algorithm 78
4.7 Experiments and Results 80
4.7.1 System's robustness to fuzzy scaling factors and model parameters 82
4.7.1 Robustness in the stochastic case 88
4.8. Summary 94
Chapter 5 – Self-Organising Fuzzy Logic Control of Multivariable Systems with a Dynamic Supervisory Layer 97
5.1 Introduction 97
5.2 An Overview of Multivariable Fuzzy Logic Control 98
5.2.1 The process reference model 99
5.2.2 Fuzzy decoupled control of multivariable systems 103
5.3 Multivariable System Control Using SOFLC-DSL 108
5.3.1 Decoupling structure and compensating scalars based on RGA 108
5.3.2 Switching mode linguistic compensator 112
5.4 Simulation on a Multivariable Dynamic Process 115
5.4.1 Robustness of the proposed algorithm to scaling factors 117
5.4.2 The system's robustness to model parameter variations 117

5.4.3 Robustness in the stochastic case 118
5.5 Summary 137
Chapter 6 – Type-2 Fuzzy Sets for Self-Organising Fuzzy Logic Control of Multivariable Systems with a Dynamic Supervisory Layer 139
6.1 Introduction 139
6.2 Type-2 Fuzzy Sets for Multivariable Systems 140
6.3 Particle Swarm Optimisation of Multivariable Fuzzy Systems 144
6.4 Multivariable Type-2 SOFLC-DSL Algorithm 146
6.5 Experiments and Results 146
6.5.1 Robustness of the proposed scheme to varying scaling factors 148
6.5.2 Robustness of the proposed scheme to varying system dynamics 148
6.5.3 Robustness of the proposed scheme to additive noise 149
6.5.4 Robustness to the compensator design 150
6.6 Summary 170
Chapter 7 – Conclusions and Future Work 173
7.1 Conclusions 173
7.2 Future Work 178
References 180

Publications

Ehtiawesh, M., and Mahfouf, M. (2014). **Self-Organising Fuzzy Logic Control with a New On-Line Particle Swarm Optimisation-based Supervisory Layer**. Proceedings of the International Conference on Fuzzy Computation Theory and Applications, 95–102.

Ehtiawesh, M., and Mahfouf, M. (2015). **Interval type-2 fuzzy sets for Self-Organising Fuzzy Logic based control with on-line PSO Optimisation**. In 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) (pp. 1–8).

List of Symbols

BPNGG	Back propagation network
CMACG	Cerebellar model articulation controller
CO	Cardiac Output
COEMG	Control output error method
CPNGG	Counter propagation network
E	Fuzzified tracking error
FARMA	Fuzzy auto-regressive moving average model
FLCGG	Fuzzy logic control
FLS	Fuzzy logic system
FMNR	Fixed Maximum number of rules
FOU	Footprint of uncertainty
GA	Genetic algorithm
GC	Fuzzy logic scaling factor for the change of tracking error
GE	Fuzzy logic scaling factor for the tracking error
GT	Fuzzy logic scaling factor for the controller output
ISE	Integral of squared error
ITAE	Integral of time and absolute error
MAP	Mean arterial pressure
MF	Membership function
MGAGG	Micro genetic algorithm
MIMO	Multi-input and multi-output
MISO	Multi-input single-output
MISOG	Multi-input and single-output
MSE	Mean square error
PIDGG	Proportional-integral-derivative controller
PIGGG	Performance index table

RBFN	Radial basis function network
RGA	Relative gain arrays
SISO	Single-input and single-output
SOFLC	Self-organizing fuzzy logic control
SOFLC-DSL	Self-organizing fuzzy logic control using a dynamic supervisory layer
TSK	Takagi-Sugeno Kang
$\dot{e}$	Predicted tracking error
e	Tracking error
u	Control signal
u_{ii}	Dominating control signal for channel i in multivariable systems
u_{ij}	Compensating signal used to counteract the interaction imposed by channel j on channel i
y	Plant output
U	Fuzzified control signal
ϑ	RGA matrix
ʎij	Compensating scalar used to counteract the interaction imposed by channel j on channel i

Chapter 1 – Introduction

1.1 Overview

Traditional control systems (e.g. PID controllers) require models of the processes to be controlled in order to define the relationship between the inputs and outputs of these systems. A major drawback of such control systems is that they often respond to the process being controlled either as linear or as behaving in a linear manner in certain operating regions.

With the availability of models that accurately capture the dynamics of the process, conventional PID controllers generally produce satisfactory performance in different environments. However, real-world applications tend to be non-linear, noisy and mathematically ill-defined, and it can therefore be very challenging to design an optimal conventional controller. Even in cases where a relatively accurate model can be developed, external factors such as noise and sudden disturbances can influence the performance of these systems. Model-free methods, such as fuzzy logic controllers (FLC), offer a preferred alternative solution in such cases. FLCs convert a linguistic control strategy based on the knowledge of an engineer/operator into an automatic control mechanism. The fuzzy controllers can therefore produce satisfactory results even

when the models are not available. In addition, fuzzy controllers can be easily designed and are generally computationally cheap to implement.

1.2 Type-1 and Type-2 Fuzzy Logic Controllers

Type-1 fuzzy controllers have been successfully applied to a wide range of problems in various applications (Al-Dunainawi and Abbod, 2016; Baumann *et al.,* 2000; Hu *et al.,* 2010*).* However, lack of knowledge, and the existence of noise and uncertainty affecting certain problems, makes the reliance on expert knowledge for constructing the fuzzy controller an unreliable approach for some applications (Passino and Yurkovich, 1997)

Biomedical processes represent a good example of such applications. The regulation of the muscle relaxation system proves to be a challenging task due to the associated uncertainty and non-linearities associated with such systems. Moreover, as the complexity of the system increases, it becomes difficult for the operator to design efficient control rules that can effectively respond to this complexity while keeping computational costs at a minimum. Therefore, optimisation approaches are needed. Optimisation techniques can help model real-world applications that are very difficult for engineers/operators to understand and handle.

Type-1 fuzzy logic systems are the most commonly known and used representations of fuzzy logic systems in the literature and industry today. They handle the uncertainty surrounding real-world problems by membership grades in the range [0, 1] (Mendel, 2001). However, humans interpret words differently, and so it can be difficult to ensure that type-1 fuzzy sets provide satisfactory performance in certain complex applications. This limitation was overcome with the introduction of type-2 fuzzy sets (Zadeh, 1975).

As the levels of uncertainty increase, type-1 fuzzy systems become unable to handle them fully. General type-2 fuzzy sets offer a better mechanism for reducing uncertainty, as their third dimension allows for the representation of uncertainty in the form of membership grades that are themselves fuzzy, rather than merely a crisp value as in type-1 fuzzy sets. The third dimension of type-2 fuzzy sets gives them an extra degree of freedom to handle higher levels of uncertainty.

Generally, the computational costs and complexity of the general type-2 fuzzy systems tend to be higher than those of type-1 (Mendel, 2001). However, recent developments and contributions in the field of type-2 fuzzy theory, which have made the construction of these systems simpler and easier to understand and implement, have motivated many researchers to study type-2 fuzzy theory and apply it to different problems in various areas (Mendel, 2000; Karnik and Mendel, 1998). Interval type-2 fuzzy sets are a special case of general type-2 fuzzy sets, and are today the most commonly used form of type-2 fuzzy sets used in industry and discussed in the literature. This is due both to their simplicity and their significantly reduced computational cost in comparison to the general type-2 fuzzy sets (Mendel, 1998).

Type-2 fuzzy systems have been applied to different domains, including intelligent control systems (Hagras, 2004; Hassani and Zarei, 2015; Wu and Tan, 2006), intelligent manufacturing (Castillo and Melin, 2007; Dereli *et al.*, 2011; Zarandi, 2009) and pattern recognition (Sepulveda *et al.*, 2007).

Figure 1.1 shows a representation of type-1, interval type-2 and general type-2 fuzzy sets. As can be observed, the third dimension in the interval representation of the fuzzy set is fixed to normality throughout the entire fuzzy set. In contrast, the same third dimension in the general type-2 representation is fuzzy and can take any value between [0, 1]. It can also be seen how the membership values in the case of a type-1 fuzzy set are described merely by a crisp value between [0, 1].

In general, type-2 fuzzy systems can be constructed using two main methods. The first approach assumes that an optimal type-1 fuzzy set has already been designed, which can then be extended to a type-2 fuzzy system (Mendel, 1998). This design is popular in the literature, and has produced good results when tackling various levels of problems. The second approach makes use of the experimental data to design type-2 fuzzy systems directly (Mendel, 1998; Wu and Mendel, 2007, Wu, 2013). In these two methods, the behaviours of the systems can be improved by using optimisation methods.

The tuning of fuzzy controllers implies the optimisation of certain parameters in order to obtain the desired response with minimal error. Various gradient and non-gradient

optimisation approaches such as quasi-Newton's method, Rosenbrock method, and Simplex method can be used to tune the parameters of type-1 and type-2 fuzzy sets and systems, including those of a bio-inspired nature.

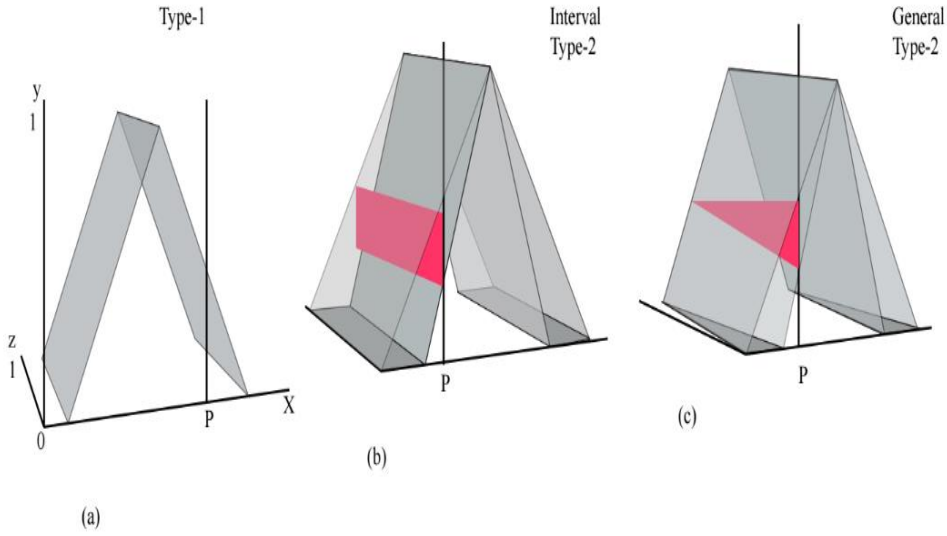


Figure 1.1: An example of the three types of fuzzy sets (Mendel, 2001).

1.3 Bio-inspired Optimisation Algorithms

The use of bio-inspired approaches can help both type-1 and type-2 fuzzy systems solve very complex problems in a diverse range of applications (Mahfouf, 2002; Zhang *et al*., 2012). These methods have been used in various applications to tune and find appropriate values and a suitable structure for fuzzy logic controllers. Genetic algorithms, particle swarm optimisation and ant colony optimisation are the three most common paradigms that are considered for the development of optimal fuzzy systems; (Juang *et al*., 2008; Martinez *et al.*, 2010; Oh *et al.*, 2011). Genetic algorithms have been heavily used with fuzzy logic controllers (Herrera *et al.*, 1995; Lu and Mahfouf, 2006; Pelusi, 2011), whereas fewer contributions have been made using sophisticated versions of particle swarm optimisation to tune the parameters of type-1 and type-2 fuzzy sets and systems.

Despite the success of type-1 and type-2 fuzzy controllers in various industrial applications, their design process, which is based on the use of the knowledge of the operator/engineer, can encounter certain challenges, especially when these controllers

are applied in practical applications. First, the structure of these controllers must be defined a priori. Second, it can be very difficult to design some of the parameters of the fuzzy controller, including determining suitable membership functions, defining a suitable rule- base size, and deriving effective fuzzy control rules, especially when the controllers are applied to Multi-Input Multi-Output (MIMO) systems (Layne and Passino, 1996; Mendel, 1998).

In order to resolve these issues and work with less dependency on pre-defined design information provided from human operators, Procyk and Mamdani (1979) developed the so-called self-organising fuzzy logic controller (SOFLC). The SOFLC belongs to the category of supervisory expert controllers, and includes a self-organising mechanism that modifies the structure of a conventional fuzzy controller and automatically performs the whole design task based on the environment in which it operates.

1.4 Supervisory Intelligent Control Systems

As industrial applications grow larger and become more complex, their control systems become more and more difficult to utilise. Forming precise models for complex processes is demanding and time-consuming. Supervisory intelligent controllers can deal effectively with this increased complexity, and can minimise the reliance on the knowledge of humans in both controller design and process operation (Abbod and Linkens, 1992; Nie and Linkens, 1999).

The supervisory mechanisms that intelligent controllers include rely on either the knowledge obtained during their operation or from previous experiences to improve their capabilities. They allow control systems to be more flexible in terms of modifying their structure to effectively cope with the processes and all the changes in the surrounding environment. They monitor the performance of the processes being controlled, and use the obtained information to take corrective actions to maintain behaviour to be as close to the desired set-points as possible (Mahfouf and Linkens, 2002; Nie and Linkens, 1999).

1.5 Thesis Overview

The contents of the thesis are divided into a total of seven chapters as follows:

Chapter two provides an overview of all the methods, concepts and operations that are needed and used throughout the thesis. The chapter begins by introducing type-1 and type-2 fuzzy systems, carefully defining and explaining various terms that are associated with them. Thereafter, the chapter outlines the basics of the traditional fuzzy logic controller, and then discusses the original SOFLC introduced by Procyk and Mamdani in 1979. Furthermore, a variety of methods utilised in the literature for the design of SOFLC algorithms are introduced.

Chapter three reports on a newly proposed SOFLC algorithm with a dynamic supervisory layer, referred to in this thesis as a SOFLC-DSL algorithm. First, the basic concepts of the standard PSO algorithm are introduced. Thereafter, an on-line PSO algorithm used to adapt the consequent part of the performance index table of the single variable SOFLC scheme is reviewed in detail. The proposed SOFLC-DSL algorithm is then applied to a non-linear, mathematically ill-understood and uncertain muscle relaxation process. The performance of the proposed scheme is then studied extensively to test its robustness to on-line changes in scaling factors and model variables as well as other environments when it is applied in the stochastic case or experiences a sudden disturbance. In all these experiments, the obtained results are compared to the results obtained from a type-1 SOFLC scheme.

Chapter four is mainly concerned with the design of type-2 SOFLC-DSL algorithms using both the interval and zSlice type-2 fuzzy sets. It begins with a comparative review of type-1 and interval type-2 fuzzy controllers, and how PSO algorithms have been used to enhance the capabilities of the latter controller in the literature. The concept of the zSlice type-2 fuzzy system is also defined. Thereafter, the structures of the proposed type-2 SOFLC-DSL algorithms are introduced and their performances assessed in various environments, similar to those outlined in Chapter 3. All the obtained performance data are compared to those of type-1 SOFLC and type-1 SOFLC-DSL algorithms.

Chapter five describes the extension of the proposed SOFLC-DSL algorithm to the MIMO case. First, a solid introduction is provided to various methods that are used to control MIMO processes. This review also includes the way in which the interaction between the input and output loops of these systems can be handled. Thereafter, a decoupled control architecture for a MIMO system that uses the type-1 SOFLC-DSL algorithm, reported in Chapter 3 as dominating controllers, is defined. The interactions between the control loops are handled using a linguistic switching compensator and RGA compensator. The proposed decoupled controller is successfully applied to a 2×2 drug dynamic process, and the simulation results are investigated and discussed.

Chapter six reports on another decoupled design of multivariable architectures in which type-2 fuzzy sets are deployed to replace those of type-1, utilised in chapter five. Both the interval and zSlice fuzzy sets are considered in this chapter. The chapter begins with an overview of different applications reported in the literature in which type-2 fuzzy sets have been used with MIMO systems. Another short review is provided on how PSO-based systems have been used to improve the performance of MIMO systems. Thereafter, the proposed decoupled controller, which is equipped with type-2 fuzzy sets, is detailed and applied to a multivariable drug process. The robustness of the architecture is extensively studied.

Chapter seven summarises all the work conducted in the thesis and discusses some of the results achieved. Furthermore, future work is suggested based on the findings and observations from previous chapters.

Relationships between all the chapters of this thesis is shown in Figure 1.1.

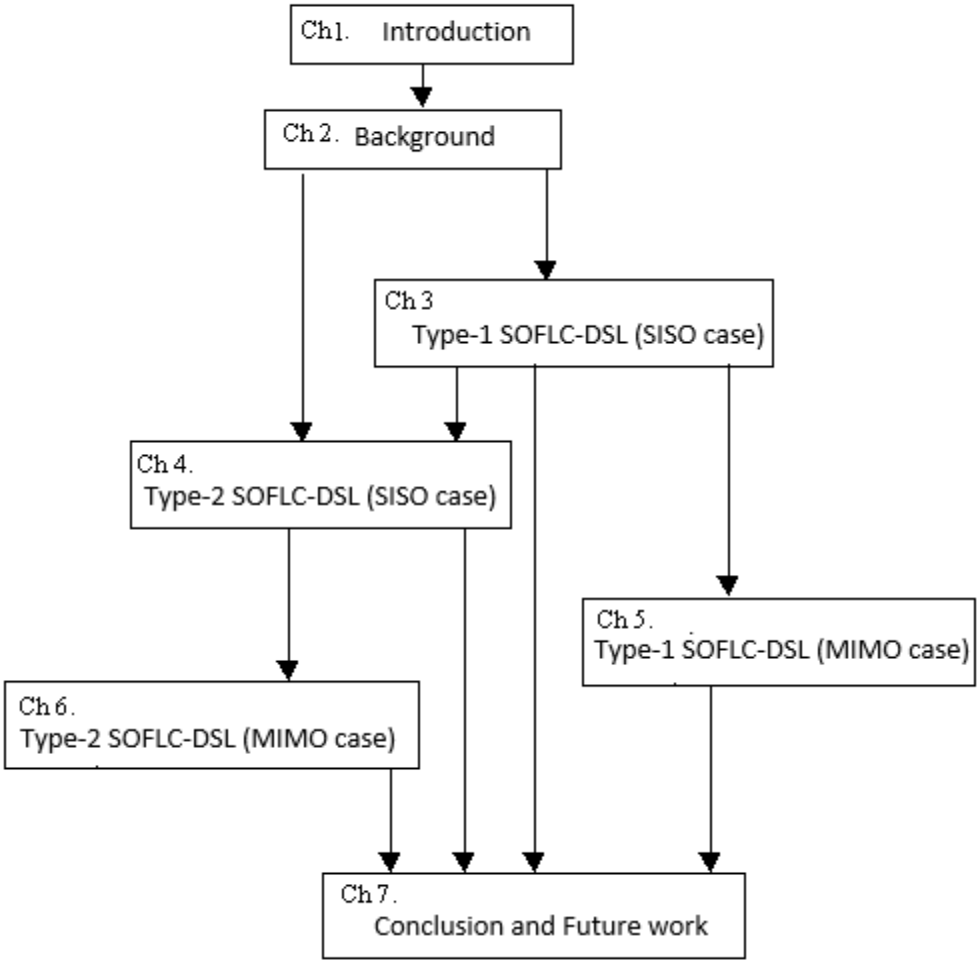


Figure 1.1: Relationships and dependencies between chapters.

Chapter 2- Background

2.1 Introduction

This chapter explains the basic general concepts that are used throughout the chapters of this thesis. It covers the most relevant aspects and methods concerning both type-1 and type-2 fuzzy logic systems, fuzzy logic control, and the self-organising fuzzy logic controllers. It begins by outlining and reviewing the basic aspects and terminologies related to type-1 fuzzy set theory, such as type-1 fuzzy sets and type-2 fuzzy systems, which are extensively used in Chapters 3 and 5. It then explains some major aspects related to the theory of type-2 fuzzy sets and systems that are needed in Chapters 4 and 6. The reasons which, generally, make type-2 produce better results in many applications are also discussed. Next, the basics of fuzzy logic control are outlined before providing a detailed description of the structure of the self-organising fuzzy logic controller that represents the key element of the thesis. In addition, the applications of this scheme in different areas are identified. The last section explains the different

approaches used for designing supervisory intelligent control systems and their application domains.

2.2 Type-1 Fuzzy Sets and Systems

Uncertainty affects different aspects of our lives and appears in a wide range of forms. For example, in decision-making, the concept of uncertainty normally refers to the state or situation when there exist vagueness, imprecision, or a lack of information. Fuzzy logic has emerged as an effective tool to measure uncertainty and to minimise its effect via the so called type-1 fuzzy sets.

2.2.1 Type-1 fuzzy sets

Fuzzy sets were first proposed by Zadeh (1965) as a way of handling uncertainty and vagueness. In a classical crisp set, elements can only be regarded as being part of the set or not and are only represented in the form of 0 (*"false"*) and 1 (*"true"*). For example, assume one has a collection of elements x that make up a universe of discourse X. A crisp set, say A, can be formed by combining various elements in the universe. In such a case, any element x can either be a member of the crisp set A or not and is defined as:

$$\mu_A(x)=\begin{cases} 1 & \text{If } x \in X \\ 0 & \text{Otherwise} \end{cases}$$

In contrast to crisp sets, fuzzy set theory permits the partial belonging to sets where elements can take values in the interval [0, 1]. For example, the characteristics of a fuzzy set A in a universe of discourse X is defined as follows:

$$\mu_A(x):\ X \rightarrow [0,1]$$

Alternatively, fuzzy sets can be denoted in the discrete case as:

$$A = \sum_{i=1}^{n} \mu_i / x_i \tag{2.1}$$

or

$$A = \{\mu_1/x_1 + \cdots + \ \mu_n/x_n\} \tag{2.2}$$

On a continuous universe, the fuzzy set is defined as follows:

$$A = \int_U \mu_A(x_i)/x_i \tag{2.3}$$

where x_i represents the i^{th} number of the universe of discourse, μ_A represents the strength of membership of element x_i and U is the universe of discourse. In equations 2.1 and 2.2, the summation and addition symbols do not represent algebraic summations but rather demonstrate the collection of each element, while in equation 2.3, the integral symbol is not an algebraic integration but rather a continuous function-theoretic aggregation operator.

An example of a type-1 fuzzy set is illustrated in Figure 2.1 which is called 'numbers close to 25'. As shown, the membership values are given based on the location of the input on the universe of discourse. When $x = 25$, for example, the highest membership value is achieved $\mu_A(25) = 1$, while this value decreases as we get away from the centre. For example, $\mu_A(30) = 0.82$, $\mu_A(40) = 0.17$.

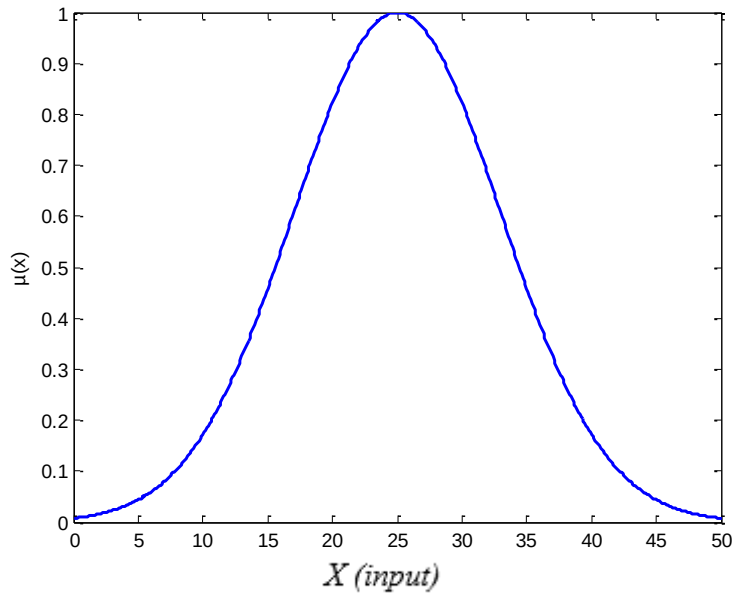


Figure 2.1: A possible membership function to characterise 'numbers close to 25'

2.2.2 Operations on type-1 fuzzy sets

Fuzzy set theory contains a wide range of operations that can be performed on fuzzy sets. These basic operations are a generalization of those used in the crisp set theory and

are defined in terms of the membership functions of fuzzy sets. The most widely used operations include: fuzzy unions, fuzzy intersections, and fuzzy complements. For example, for two fuzzy sets, A and B, which are described by the membership functions $\mu_A(x)$ and $\mu_B(x)$, the union is a fuzzy set in the discourse *U*, defined by $A \cup B$, with a membership function denoted as:

$$\mu_{A\cup B}(x) = max[\mu_A(x), \mu_B(x)] = \mu_A(x) \vee \mu_B(x)$$

The interaction of sets A and B is a fuzzy set $A \cap B$, whose membership function is denoted as:

$$\mu_{A\cap B}(x) = min[\mu_A(x), \mu_B(x)] = \mu_A(x) \wedge \mu_B(x)$$

It is also defined with a product operator as:

$$\mu_{A\cap B}(x) = Prod[\mu_A(x), \mu_B(x)] = \mu_A(x) * \mu_B(x)$$

The complement of the fuzzy set A is defined as:

$$\mu_{A'}(x) = 1 - \mu_A(x)$$

2.2.3 Type-1 fuzzy logic systems

Type-1 fuzzy logic system represents a non-linear mapping of an input data vector into a scalar output (Mendel, 2000). It is also known as the fuzzy rule-based system, fuzzy expert system, and the fuzzy model. The most widely used ones in literature are Mamdani Fuzzy Logic System (FLS) and Takagi-Sugeno-Kang (TKS) FLS. Both Mamdani and TSK FLSs share the same antecedent structure and are characterised by IF-THEN rules. Their difference lies in the structure of their consequent, which is in the form of a fuzzy set in the Mamdani system while the TSK system uses a function. As can be seen in Figure 2.2, the Mamdani FLS, which is the most widely used system in engineering applications, has four components: fuzzifier, rules, inference, and defuzzifier.

2.2.3.1 Fuzzifier

The fuzzifier is the component in the FLS that decides how the input signal will be mapped into a fuzzy signal for further processing. This is carried-out via membership functions. The singleton fuzzifier is often the one used in literature due to its simplicity and reduced computational cost. In this fuzzifier, the crisp inputs $x = (x_1, x_2, .., x_n)$ are connected and converted into fuzzy sets by being evaluated based on the antecedent part of the rules, each input is assigned a membership grade whose values depend on the corresponding fuzzy set. Another common approach is the non-singleton fuzzifier, which is frequently used in cases where noise is present in the data. The non-singleton fuzzifier uses a fuzzy set known as the variability set that all inputs are mapped into. The membership grade for any x is the centre of the fuzzy set while the values of the neighbours get smaller as they move away from x. Figure 2.3 shows an example of the two fuzzifiers.

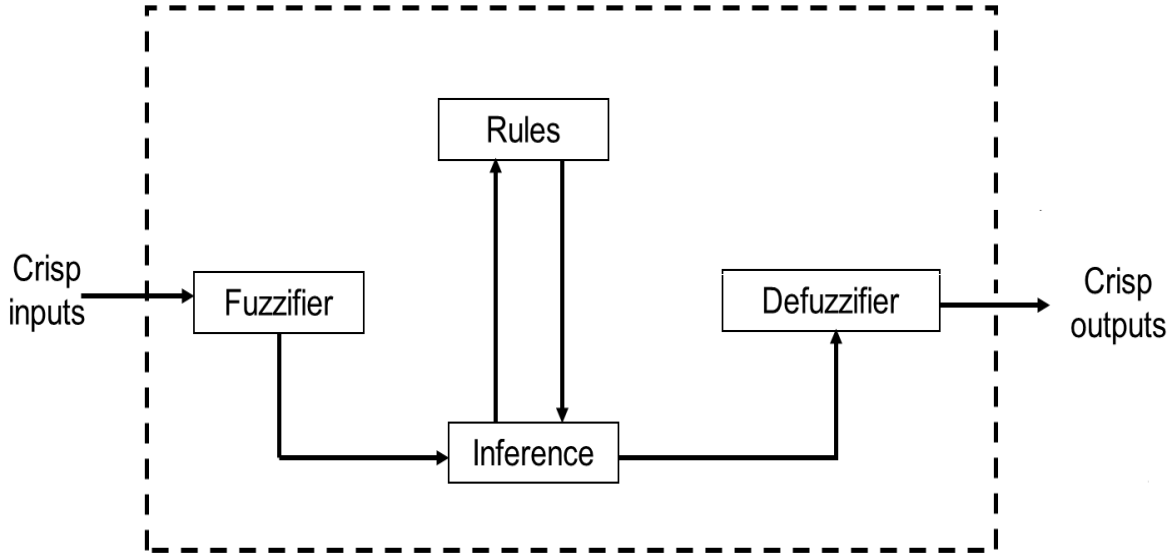


Figure 2.2: Type-1 Fuzzy logic system.

2.2.3.2 Rules

Fuzzy rules are a key tool used to express pieces of knowledge in the form of conditional statements represented in the IF-THEN form and are defined in terms of two parts: the IF part, which is also known as the antecedent of the rule, and the THEN part,

widely known as the consequent part. For example, for a Mamdani type-1 system with *S* inputs, $x_1 \in X_1, \ldots, x_s \in X_s$ and one output $y \in Y$, famously known as multiple input signal output (MISO) system. It is assumed that it has *N* rules and the antecedents are connected by the operation *'and'* (t- norm operator). The l^{th} rule in this case is given by:

$$R^l : \text{IF } x_1 \text{ is } F_1^l \text{ and } \ldots \text{ and } x_p \text{ is } F_p^l, THEN \text{ y is } G^l \quad l=1,\ldots,N \tag{2.4}$$

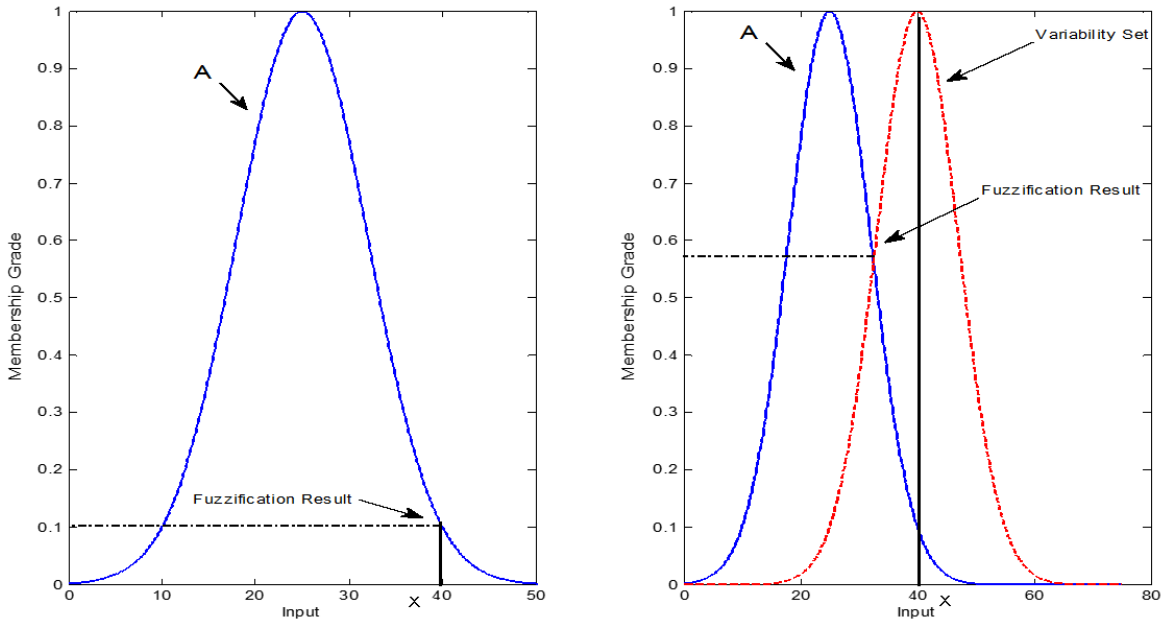


Figure 2.3: The fuzzification of a crisp input *X* using singleton (left) and non-singleton (right) fuzzification on a fuzzy set *A*.

2.2.3.3 Inference engine

The fuzzy inference engines use the principles of fuzzy logic to formulate mapping from a given input to an output. The Mamdani system uses fuzzy sets in both the antecedent and consequent parts. Equation 2.4 can be expressed as $F_1^l \times \ldots \times F_p^l = A^l$. Then, it can be re-written as:

$$R^l: F_1^l \times \ldots \times F_p^l \rightarrow G^l = A^l \rightarrow G^l \quad l=1,\ldots,N. \tag{2.5}$$

The fuzzy rule R^l is defined by the membership function $\mu_{R^l}(x,y) = \mu_{R^l}(x_1, x_2, \ldots, x_p, y)$, such that:

$$\mu_{R^l}(x,y) = \mu_{A^l \rightarrow G^l}(x,y) \tag{2.6}$$

Equation 2.6 can also be written as follows:

$$\mu_{R^l}(x,y) = \mu_{A^l \to G^l}(x,y) = \mu_{F_1^l \times \ldots \times F_p^l \to G^l}(x,y)$$
$$= \mu_{F_1^l \times \ldots \times F_p^l}(x) * \mu_{G^l}(y)$$
$$= \mu_{F_1^l}(x_1) * \ldots * \mu_{F_p^l}(x_p) * \mu_{G^l}(y) \quad (2.7)$$
$$= T_{i=1}^{P} \mu_{F_i^l}(x_i) * \mu_{G^l}(y)$$

It has been assumed here that a Mamdani system is used and the antecedents of the rules are connected by *'and'* (t-norms) and *T* represents the t-norm. The *S*-dimensional input to the fuzzy rule $R^l$are defined by a fuzzy set A_x whose membership grade is represented by:

$$\mu_{A_x}(x) = \mu_{x_1}(x_1) * \ldots * \mu_{x_p}(x_p) = T_{i=1}^{P} \mu_{x_i}(x_i) \quad (2.8)$$

Each fuzzy rule R^l determines a fuzzy set in Y, $B^l = A_x$ o R^l

where

$$\mu_{B^l}(y) = \mu_{A_x o R^l}(y) = sup_{x \in X}\left[\mu_{A_x}(x) * \mu_{A^l \to G^l}(x,y)\right], y \in Y \quad (2.9)$$

Equation 2.9 represents the relationship between the fuzzified inputs that activate the inference engine and the resultant fuzzy output set, as shown in Figure 2.2. This sup-start composition forms a non-linear mapping between the input signal x and the scalar value produced from the output fuzzy set $\mu_{B^l}(y)$.

Substituting equation 2.7 and 2.8 into equation 2.9 results in

$$\mu_{B^l}(y) = \mu_{G^l}(y) * \{ \left[sup_{x_1 \in X_1}\ \mu_{x_1}(x_1) * \mu_{F_1^l}(x_1)\right] *$$
$$\ldots\ldots * \left[sup_{x_p \in X_p}\ \mu_{x_p}(x_p) * \mu_{F_p^l}(x_p)\right], y \in \text{Y} \quad (2.10)$$

The output fuzzy set B is produced by all fuzzy rules and can be determined by combining B^l and its membership grades $\mu_{B^l}(y)$ for all $l = 1..,N$

$$B = A_x\ o[R^1,..,R^m] = \cup_{i=1}^{M} A_x\ o\ R^i \quad (2.11)$$

The fuzzy rules can be combined via different compositions such as sup-min or sup-product compositions (Lee, 1990).

2.2.3.4 Defuzzifier

The defuzzifier is the last stage of the fuzzy logic system and it is responsible for producing crisp output values from the output fuzzy sets generated by the inference engine. There are different methods used in the literature. Since computational complexity can be an obstacle that prevents the usage of fuzzy systems in practical applications, this motivated different researchers to propose numerous methods for defuzzification, including, for example: centroid, centre of sets, mean of maxima height, modified height, and centre of sums. The centroid defuzzifier, as an example, uses the union (t-conorm) to combine the output type-1 fuzzy sets and then finds the centroid of this set. For a fuzzy set B with an associated membership function $\mu_B(y)$.

$$B = \cup_{l=1}^{M} B^l \tag{2.12}$$

Then, the centroid deffuzifier is defined as:

$$Y_B(x) = \frac{\sum_{i=1}^{N} y_i \mu_B(y_i)}{\sum_{i=1}^{N} \mu_B(y_i)} \tag{2.13}$$

In equation 2.13, the membership function of the set B that represents the output set is discretised into N points. A different output value $Y_B(x)$ is obtained for every FLS input x.

2.3 Type-2 Fuzzy Sets and Systems

As already stated, the primary motivation behind the introduction of fuzzy sets was to allow uncertainty, imprecision, and partial truth to be expressed in a mathematical form. In addition, fuzzy set theory provides tools, e.g. type-1 fuzzy logic systems, which can handle vagueness and uncertainty relating to many real world problems. Type-1 fuzzy logic systems have been successfully applied in a wide range laboratory studies, as well

as complex industrial applications. Despite this success, these systems are unable to completely model and minimise the effect of numerical or linguistic uncertainties in environments that are dynamic and unconstructed (Mendel, 2001). This difficulty lies in the way type-1 fuzzy sets are constructed, which, despite having a connotation of uncertainty, their membership grades are completely certain. They represent uncertainty via crisp values in the range [0, 1]. In cases when an entity such as measurement is uncertain, exact values become very hard to determine, and the usage of type-1 fuzzy sets in such situations would lead to better results than the traditional crisp sets. However, as the degree of uncertainty increases, type-1 fuzzy sets fail to perform well. This issue was addressed by Zadeh in 1975 and led to the discovery of a more sophisticated type of fuzzy sets, which included the general type-2 fuzzy set (Zadeh, 1975). Type-2 fuzzy sets are better tools to reduce uncertainty in systems due to the third dimension, which enables them to represent uncertainty via membership grades, which are themselves fuzzy and not merely crisp values as in type-1 fuzzy sets.

From the view point of control applications, high degrees of uncertainty can be in the form of noise in the measurements received from the output resulting from the sensors used in the control loop, loss of information or simple due to a change in the conditions surrounding the process.

The most common sources of uncertainty in fuzzy logic systems in the literature are (Mendel, 2001):

- Uncertainty that is associated with the way humans interpret ‘words’ in fuzzy rules.
- Uncertainty that can appear in the antecedent or consequent parts of the fuzzy rule.
- Measurement uncertainty.
- Parameter tuning data uncertainty.

The first two are result of uncertainty about the way fuzzy sets are constructed, while the last two are associated with the measurement of the data used in the fuzzy logic system. Type-2 fuzzy systems have the ability to model and reduce the effect of all these types of uncertainty and normally give better results than the type-1 fuzzy system when such circumstances are faced. When the sources of such noise and uncertainty disappear,

type-2 fuzzy sets should reduce to type-1 fuzzy sets. In such cases, type-2 fuzzy sets perform in similar way as the type-1 fuzzy sets.

2.3.1 General type-2 fuzzy sets

If a type-1 fuzzy set is blurred to the right and to the left, as shown in Figure 2.4, then a type-2 fuzzy set is constructed. In this scenario, for an input *x'*, different values are taken by the membership function *u'* and are weighted differently. Therefore, one can assign different grades to all the points. If the same is done to all $x \in X$, then a third dimension is constructed and this forms the general type-2 fuzzy set.

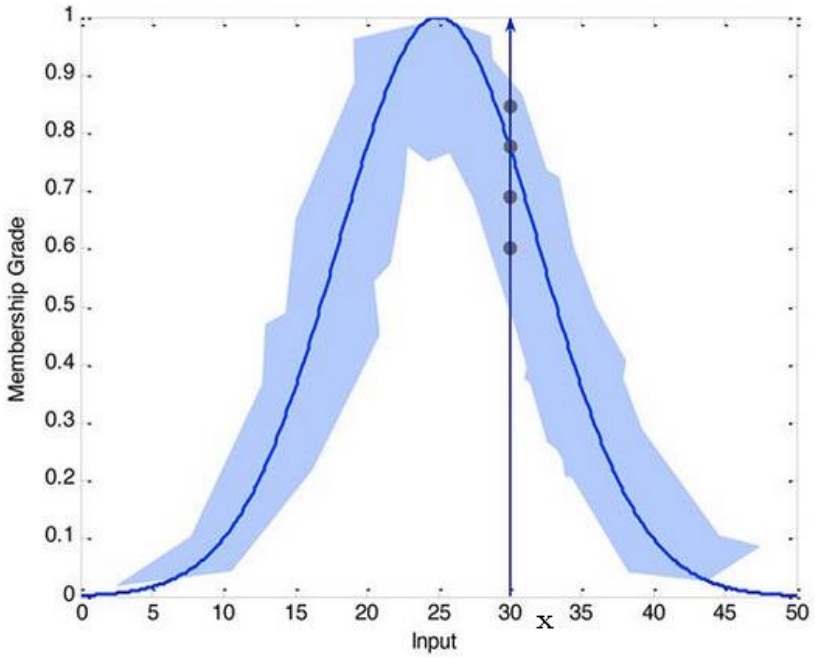


Figure 2.4: Blurred type-1 membership function.

The general concepts of type-2 fuzzy sets, as stated above, were first defined by Zadeh in a number of contributions (Zadeh, 1975). These papers outlined the basic operations of type-2 fuzzy sets. Yet, the implementation of type-2 in fuzzy logic systems remained unpopular in the following years due to a number of obstacles, which included:

- Difficulty in characterising type-2 fuzzy sets.
- Difficulty in performing operations on type-2 fuzzy sets.

- Difficulty in finding a way to inference with type-2 fuzzy sets.
- Difficulty in finding a way to move from the output fuzzy set into a defuzzified value.

Karnik and Mendel (1998) were the first researchers to overcome these obstacles and expand type-2 fuzzy set theory to a point that it can be implemented practically. Other researchers that have contributed to this field include Mizumoto and Tanaka (1976), Dubois and Prade (1980) and Mendel and John (2002).

The type-2 fuzzy set is denoted by $\tilde{A}$ and is characterised by membership functions that are themselves fuzzy, i.e. each point of this set is described by a fuzzy set in the range [0, 1]. A type-2 fuzzy set is denoted by (Mendel and John, 2002):

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) \mid \quad \forall x \in X \quad \forall u \in J_{\chi} \subseteq [0, 1] \} \tag{2.14}$$

where $\mu_{\tilde{A}}(x, u)$ is a membership function in which $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$, while $J_{\chi} \subseteq [0, 1]$ is called the primary membership of x and $\mu_{\tilde{A}}(x, u)$ represents a type-1 fuzzy set known as the secondary set. Hence, the membership grade of type-2 fuzzy set can take any value in the range [0, 1], which may define the primary membership. For each primary membership, there is also a secondary membership, which is also defined in the range [0, 1]. The uncertainty in type-2 fuzzy sets is represented by the union of all primary memberships and is known as the footprint of uncertainty (FOU) (Mendel, John, 2001).

The type-2 fuzzy set, $\tilde{A}$, can also be denoted as follows:

$$\tilde{A} = \int_{x \in X} \int_{x \in J_{\chi}} \mu_{\tilde{A}}(x, u)/(x, u), \; J_{\chi} \subseteq [0, 1] \tag{2.15}$$

where $\iint$ represents union over all x and u (Mendel, 2001)

When the universe of discourse is discrete, $\tilde{A}$ is described as follows:

$$\tilde{A} = \sum_{x \in X} \sum_{x \in J_{\chi}} \mu_{\tilde{A}}(x, u)/(x, u), \; J_{\chi} \subseteq [0, 1] \tag{2.16}$$

Figure 2.5 shows an example of the general type-2 fuzzy set. As can be seen, the membership values for every input x is no longer represented by a single crisp value as experienced with type-1 fuzzy sets, rather, it is now defined by an independent membership function.

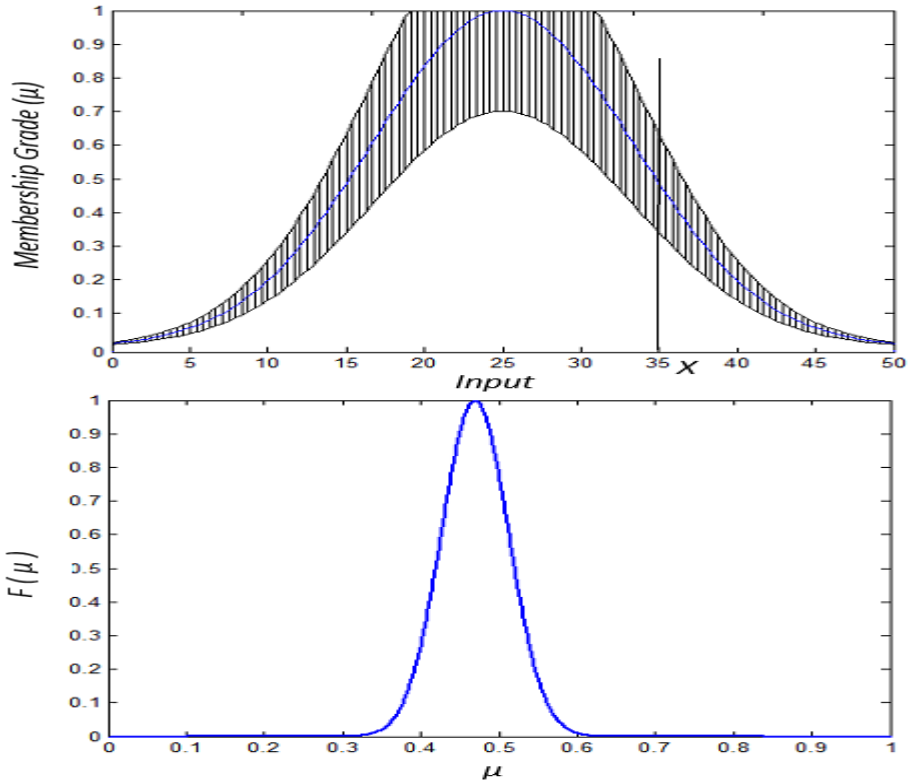


Figure 2.5: General type-2 fuzzy set 'numbers close to 25'.

2.3.2 Interval Type-2 fuzzy sets

Despite the obstacles the general type-2 fuzzy sets have overcome, extending type-1 FLS to type-2 FLS remained impractical due to the high computational cost of type-2 fuzzy logic systems. With the introduction of the so-called interval type-2 fuzzy sets, which are less complex and easier to compute, this obstacle was finally overcome (Mendel, 2001). The characteristic of an interval type-2 fuzzy set is defined as:

$$\tilde{A} = \int_{x \in X} \int_{x \in J_x} 1/(x,u),\ J_x \subseteq [0,1] \qquad (2.17)$$

As it can be seen from equation 2.17, all the secondary membership functions $\mu_{\tilde{A}}(x,u)$ are now equal to 1, which makes it a special case of the general type-2 fuzzy set. Figure 2.6 shows an example of the interval type-2 fuzzy set.

The footprint of uncertainty in the interval type-2 fuzzy sets is the same as that of the general type-2 fuzzy set and is defined by the union of all primary memberships and it represents the entire interval type-2 fuzzy set. The interval type-2 fuzzy set can be

defined in terms of a lower membership function $\underline{\mu}_{\tilde{A}}(x)$ and an upper membership function $\overline{\mu}_{\tilde{A}}(x)$. The middle solid Gaussian curve in Figure 2.6 is known as the principal membership function and is formed by the union of all primary membership mid-points. The type-2 fuzzy set reduces to its principle membership function when all levels of uncertainty of the membership function disappear.

2.3.3 Type-2 fuzzy set operations

In order to perform the set operations on type-2 fuzzy sets, the binary operations of maximum and minimum (or product) and the operation of negation needed to be extended from crisp values to type-1 fuzzy sets, as in type-2 fuzzy sets, uncertainty is represented by a function. The tool for carrying this extension is Zadeh's extension principle (Zadeh, 1975). Different tools were used to define the extension principle. For example, Zadeh used the minimum t-norm and maximum t-conorm while Mizumoto and Tanaka (1976) and Dubios and Prade (1980) used other t-norms and t-conorms.

For two type-2 fuzzy sets $\widetilde{A}$ and $\widetilde{B}$ in universe X

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x)/x = \int_{x \in X} \left[\int_{u \in J_x^u} f_x(u)/u \right] /x,\ J_x^u \subseteq [0,1] \tag{2.18}$$

and

$$\tilde{B} = \int_{x \in X} \mu_{\tilde{B}}(x)/x = \int_{x \in X} \left[\int_{v \in J_x^v} g_x(v)/v \right] /x,\ J_x^v \subseteq [0,1] \tag{2.19}$$

The type-2 fuzzy set operations can be performed as (Mizumoto and Tanaka, 1976; Karnik and Mendel, 2001b):

- Union

The union of the secondary functions $\tilde{A}$ and $\tilde{B}$ is expressed as:

$$\mu_{\tilde{A} \cup \tilde{B}}(x) = \int_{u \in J_x^u} \int_{v \in J_x^v} f_x(u) \bigstar g_x(v)/(u \vee v) = \mu_{\tilde{A}}(x) \coprod \mu_{\tilde{B}}(x),\ x \in X \tag{2.20}$$

where $\vee$ defines maximum and $\bigstar$ defines minimum or product t-norm. $\coprod$ denotes the join operation. In order to perform the union operation between the secondary memberships of the two sets $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$, the operation $u \vee v$ must be carried-out for every possible pairing of u and v such that $u \in J_x^u$ and $v \in J_x^v$.

- Intersection

While the interactions of $\tilde{A}$ and $\widetilde{\mathrm{B}}$ is given by:

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \int_{u\in J_x^u} \int_{v\in J_x^v} f_x(u) \bigstar g_x(v)/\,(u \wedge v) = \mu_{\widetilde{A}}(x) \prod \mu_{\widetilde{B}}(x),\ x \in X \qquad (2.21)$$

where $\wedge$ denotes the minimum or product, while $\bigstar$ defines minimum or product t-norm, and $\prod$ denotes the meet operation. To perform the intersection between the secondary memberships of the two sets $\mu_{\widetilde{A}}(x)$ and $\mu_{\widetilde{B}}(x)$, the operation, $u \wedge v$ must be carried-out for every possible pairing of u and v such that $u \in J_x^u$ and $v \in J_x^v$.

2.3.4 Type-2 fuzzy logic systems

The structure of type-2 fuzzy logic system is similar to that of type-1 fuzzy logic system. The major difference lies in the extra component known as the type-reducer. Figure 2.7 shows the structure of the type-2 fuzzy logic system.

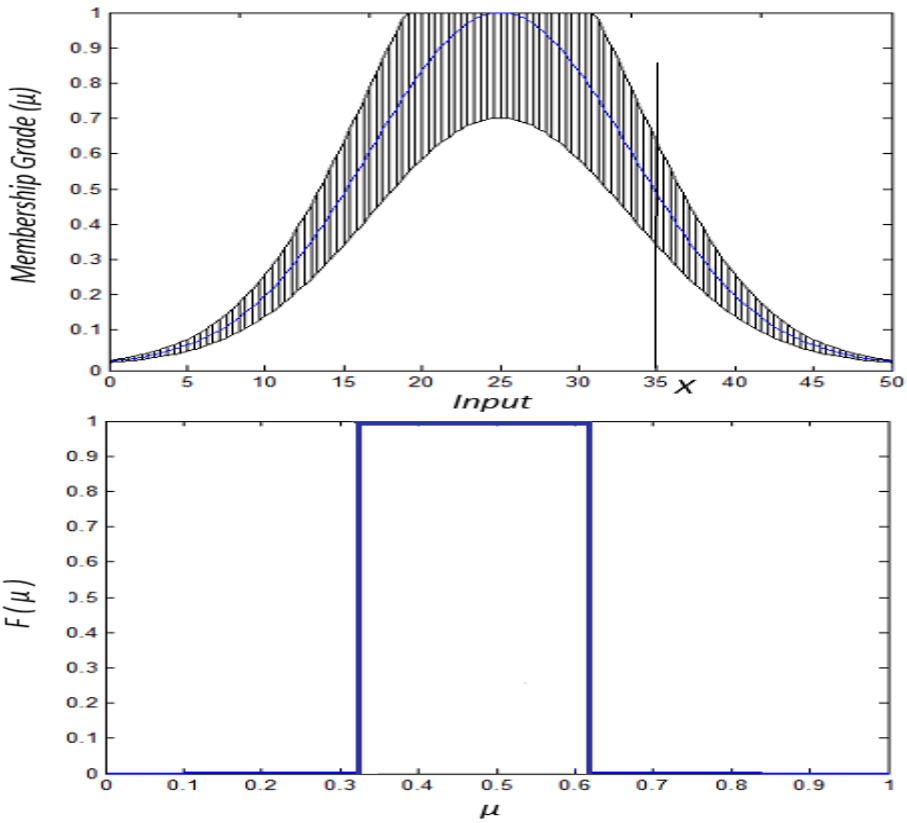


Figure 2.6: Interval type-2 fuzzy set 'numbers close to 25'.

The type reducer block reduces the output of the inference engine, which is a type-2 fuzzy set into a type-1 fuzzy set, so it can be defuzzified and converted into a crisp output. The five components that type-2 FLS has are: fuzzifier, rules, inference, type reducer and defuzzifier.

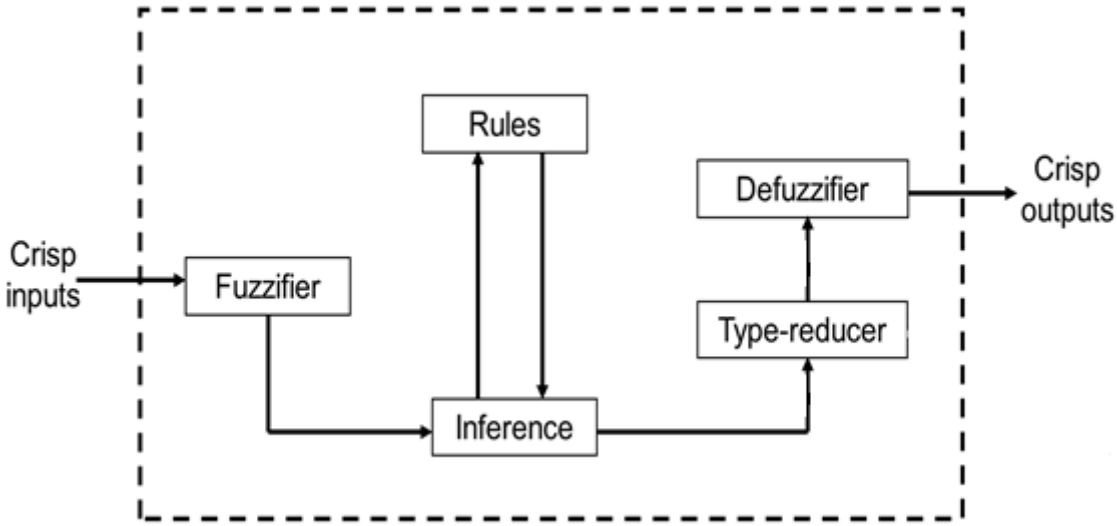


Figure 2.7: Type-2 FLS

2.3.4.1 Fuzzifier

The crisp inputs in the fuzzifier of type-2 FLS, $x = (x_1, x_2, .., x_n)$ are gathered and converted into fuzzy sets by being evaluated based on the antecedent part of the rules. Each input is assigned to its type-2 fuzzy set, say *A*, with its degree of membership in each type-2 fuzzy set. When a general type-2 fuzzy logic system is used, the fuzzified input is a type-1 fuzzy set that represents the secondary membership grades for every input. However, the fuzzified input can only be an interval set when an interval type-2 fuzzy logic system is used.

2.3.4.2 Rules

Fuzzy rules are also represented by IF-THEN statements where the IF part is also known as the antecedent and the THEN part is known as the consequent. They are the same as those of the type-1 fuzzy logic system, but in the former, at least one type-2 fuzzy set is used. For example, for a Mamdani type-2 FLS with *S* inputs, $x_1 \in X_1, \ldots, x_s \in X_s$ and

one output $y \in$ Y, they are also known as multiple-input signal-output (MISO) system. It is assumed that it has *N* rules and the antecedents are connected by the operation *'and'* (t- norm operator). In this case, the l^{th} rule is given by:

$$R^l\text{: IF } x_1 \text{ is } \tilde{F}_1^l \text{ and .. and } x_p \text{ is } \tilde{F}_p^l, THEN \text{ y is } \tilde{G}^l \quad l=1,...,N. \tag{2.22}$$

2.3.4.3 Inference Engine

The fuzzy inference engine uses fuzzy logic principles to convert input type-2 fuzzy sets into output type-2 fuzzy sets, based on the fuzzy IF-THEN rules stored in the rule-base. In Mamdani FLS, the rule in both the antecedent and the consequent part use type-2 fuzzy sets. If $\tilde{F}_1^l \times \ldots \times \tilde{F}_p^l = \tilde{A}^l$, then equation 2.22 can be written as:

$$R^l\text{: } \tilde{F}_1^l \times \ldots \times \tilde{F}_p^l \rightarrow \tilde{G}^l = \tilde{A}^l \rightarrow \tilde{G}^l \quad l=1,...,N. \tag{2.23}$$

The fuzzy rule R^l is described by the membership function

$\mu_{R^l}(x,y)=\mu_{R^l}(x_1, x_2, .., x_p, y)$,

such that:

$$\mu_{R^l}(x,y)= \mu_{\tilde{A}^l \rightarrow \tilde{G}^l}(x,y) \tag{2.24}$$

Equation 2.24 can also be written as follows:

$$\begin{aligned}\mu_{R^l}(x,y)= \mu_{\tilde{A}^l \rightarrow \tilde{G}^l}(x,y) &= \mu_{\tilde{F}_1^l}(x_1) \prod \ldots . \prod \mu_{\tilde{F}_p^l}(x_p) \prod \mu_{\tilde{G}^l}(y) \\ &= \left[\prod_{i=1}^{p} \mu_{\tilde{F}_1^l}(x_1)\right] \prod \mu_{\tilde{G}_p^l}(y)\end{aligned} \tag{2.25}$$

In general, the type-2 fuzzy set ${A^l}_x$ represents the *S*-dimensional input to the fuzzy rule R^l.

The membership function of ${A^l}_x$is defined as follows:

$$\mu_{\tilde{A}_x}(x) = \mu_{\tilde{x}_1}(x_1) \prod \ldots . \prod \mu_{\tilde{x}_p}(x_p) = \prod_{i=1}^{p} \mu_{\tilde{x}_i}(x_1) \tag{2.26}$$

The inputs defined by $\tilde{x}_1 (\, i = 1, .., p)$ are labels of the fuzzy sets.

The fuzzy rule R^l is used by the inference engine to determine the output type-2 fuzzy set $\tilde{\mathrm{x}}^l = \tilde{\mathrm{A}}_x$ o R^l, such that

$$\mu_{\tilde{B}^l}(y)=\mu_{\tilde{A}_x \text{ o } R^l} = \bigsqcup_{x\in X}\left[\mu_{\tilde{A}_x}(x) \sqcap \mu_{R^l}(x,y)\right] y \in Y \quad l = 1, \dots N \tag{2.27}$$

This equation represents the case when a type-2 fuzzy set activates one of the fuzzy rules stored in the rule-base in the inference engine to produce the output fuzzy set.

If the fuzzy logic system uses an interval type-2 fuzzy set and a product t-norm to perform the intersection, the results from the input and antecedent operation, which are contained in the firm set $\sqcap_{i=1,}^{p}\mu_{\tilde{F}_i}(x_i' = F^l(x')$, such as

$$F^l(x') = \left[\underline{F^l}(x'), \overline{F^l}(x')\right] = \left[\underline{F^l}, \overline{F^l}\right] \tag{2.28}$$

where

$$F^l(x') = \underline{\mu}_{\tilde{F}_1^l}(x_1') * \dots * \underline{\mu}_{\tilde{F}_p^l}(x_p') \tag{2.29}$$

$$F^l(x') = \overline{\mu}_{\tilde{F}_1^l}(x_1') * \dots * \overline{\mu}_{\tilde{F}_p^l}(x_p') \tag{2.30}$$

where * stands for production operation.

2.3.4.4 Type reducer

The type reducer converts the output type-2 fuzzy set produced by the inference engine into a type-1 fuzzy set by performing a centroid calculation (Mendel, 2001), which is then converted into a crisp value using the defuzzifier.

Type reduction in general type-2 fuzzy sets: Type reduction for the general type-2 fuzzy set is computationally expensive, which forms the major disadvantage of type-2 fuzzy logic systems. However, the process has a reduced mathematical cost when interval type-2 sets are used. The first known type-reduction method was proposed by (Mendel, 2001). In this method, the union of all the centroids of all the embedded type-2 related to the general type-2 fuzzy set are computed. This method is unpopular in real applications due to the large number of embedded sets involved in type-2 fuzzy sets. Gafa and Coupland (2011) proposed a recursive algorithm to reduce the computations. However, the results still cannot be practically implemented in real applications. A more practical approach known as the geometric defuzzifier was introduced by Coupland and John (2007). In this algorithm, random samples of the embedded sets are used to approximate the exact values (Greenfield, 2005). Another approach that could be practically implemented and does not depend on the concept of embedded sets is the

vertical slice centroid type-reducer (VSCTR), which was initially introduced by (John, 2000) and then developed by (Lucas, 2007). Other methods found in the literature include the usage truth numbers in triangular type-2 fuzzy sets (Starczewski, 2009b), as well as other methods that rely on z-slices (Wanger and Hagras, 2009).

Type reduction in interval type-2 fuzzy sets: As stated above, the reduction process for interval type-2 fuzzy sets is simpler. A commonly used type reducer for interval type-2 fuzzy sets is the Karnik-Mendel (KM) iterative algorithm (Karnik and Mendel, 2001a), which was then improved by the so-called Enhanced KM algorithm (Wu and Mendel, 2009). Another method, known as the collapsing method, is proposed by (Greenfield *et al.*, 2009). Other methods proposed for practical usage are reported in different contributions in literature, which include uncertainty bounds (Wu and Mendel, 2002), Nie-Tan (NT) method (Nie and Tan, 2008).

2.3.4.5 Deffuzifer

The deffuzifer converts the type-reduced into a crisp value so it can be generated into the system. A common natural way is to determine the centroid of the type-reduced set. However, there are other methods available for use, which result in more accurate values (Mendel, 2001).

2.4 Fuzzy control

Control applications represent the field in which fuzzy logic has made great successes. Fuzzy control is involved in various consumer products that are used today. Indeed, Mamdani proposed the world's first fuzzy controller (Mamdani, 1974). His primary motivation was to utilise the knowledge and experience of an operator to construct a controller that would emulate human control behaviour to a certain extent. The success of Mamdani's work opened the door to many other contributions in fuzzy logic control that increased over the years.

Fuzzy controllers have various advantages over traditional control schemes. First, they do not require a mathematical model of the system to be controlled. Second, fuzzy controllers are non-linear and have the ability to control a wide range of inherently

complex, time-varying, and complex systems that involve time delays without the usage of complicated mathematics as they are constructed empirically.

A simple fuzzy controller that is used to maintain the output of process around a given desired-point is shown in Figure 2.8.

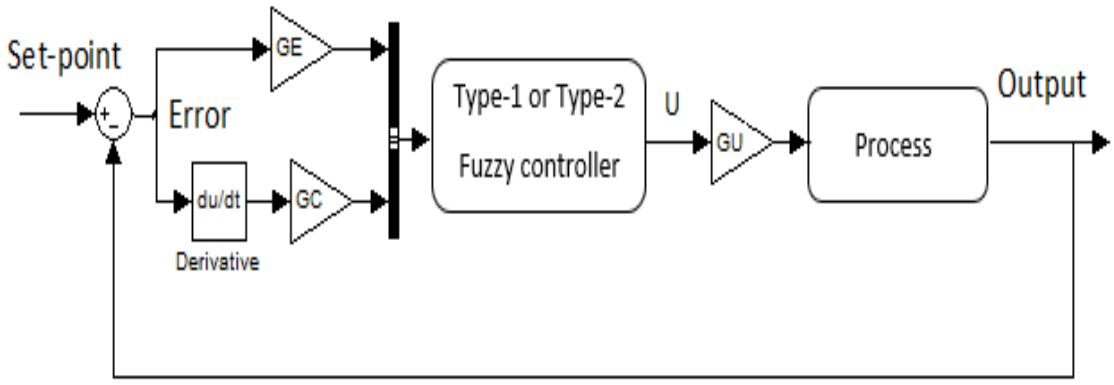


Figure 2.8: A simple fuzzy controller.

The control strategy of fuzzy controllers is governed by linguistic statements expressed as:

'If error is positive big, THEN the controller output needs to be positive medium'

where the controller input is represented by the antecedent of the rule while the consequent represents the output of the controller. The labels such as big and medium are known as the linguistic values and are normally translated into numerical values. The input signals to the controller, expressed as Error *(E)* and change of error *(EC)* as well as the output of the controller (*U*) are scaled by scaling factors (*GE*) and (*GC*) and (*GT*) respectively, which are used to alter the response of these variables.

Despite having been successfully applied to a variety of systems in different areas, the structure of fuzzy controllers, whether they use a type-1 FLS or type-2 FLS, need to be defined beforehand in most of these applications. Some of the usual issues that arise when designing a fuzzy controller include: choosing suitable membership functions, the determination of suitable rule-base size, and obtaining proper control rules. This issue was tackled by the introduction of the so called self-organising fuzzy logic controller (SOFLC).

2.5 Self-organising fuzzy logic controller

The self-organising fuzzy logic controller (SOFLC), which represents the focus of this thesis, was first introduced by Procyk and Mamdani (1979). It involves an amendable control policy that changes according to the dynamics of the process under control as well as the environment in which it operates. The SOFLC uses past experience, which is a combination of the past control rules and the output they produced to evaluate the behaviour of the controller and continuously improve its performance. While operating, the SOFLC performs two tasks:

- Issuing the appropriate control actions while observing the environment that it operates in.
- Improving the control action of the controller through modifying the control rules of the lower-level fuzzy logic rule-base based on evaluation of the system's performance.

The SOFLC has been applied to a wide range of systems. This, for example, includes controlling the water level of a tank. This was achieved by applying the SOFLC to a simulated process of two connected tanks (Yamazaki and Mamdani, 1982). Another popular application of the SOFLC is the Sugeno's fuzzy car (Sugeno and Nishida, 1985), where the controller is used to enable the car to automatically learn how to park. The controller was also used to regulate the attitude of a flexible satellite model (Daley and Gill, 1986). Another researcher has used SOFLC to control an active suspension system (Huang and Lin, 2003). Mahfouf and Abbod (1994) applied the SOFLC to human muscle relaxation in a non-linear muscle-relaxant anaesthesia model. The SOFLC was also used for a sedation control of intracranial pressure (ICP) pattern in an intensive care unit (Shieh *et al*., 2006).

In these applications, the SOFLC has shown its ability to control uncertain, non-linear, time variant, mathematically ill-defined systems, due to its capability to modify its structure to suit the environment in which it operates and also its ability to effectively control systems without the need for explicit mathematical models.

2.5.1 Description of the controller

The SOFLC enables its structure to be modified by adding a performance feedback layer to the traditional fuzzy logic controller as shown in Figure 2.9. As can be seen from the figure, the controller includes two parts. Part '*B*' is a normal Mamdani type system while part '*A*' represents the self-organising mechanism that is responsible for evaluating the performance of part 'B' and modifying its control rules. It consists of the Performance Index (PI) table, the rules modifier, the state buffer, and the process model.

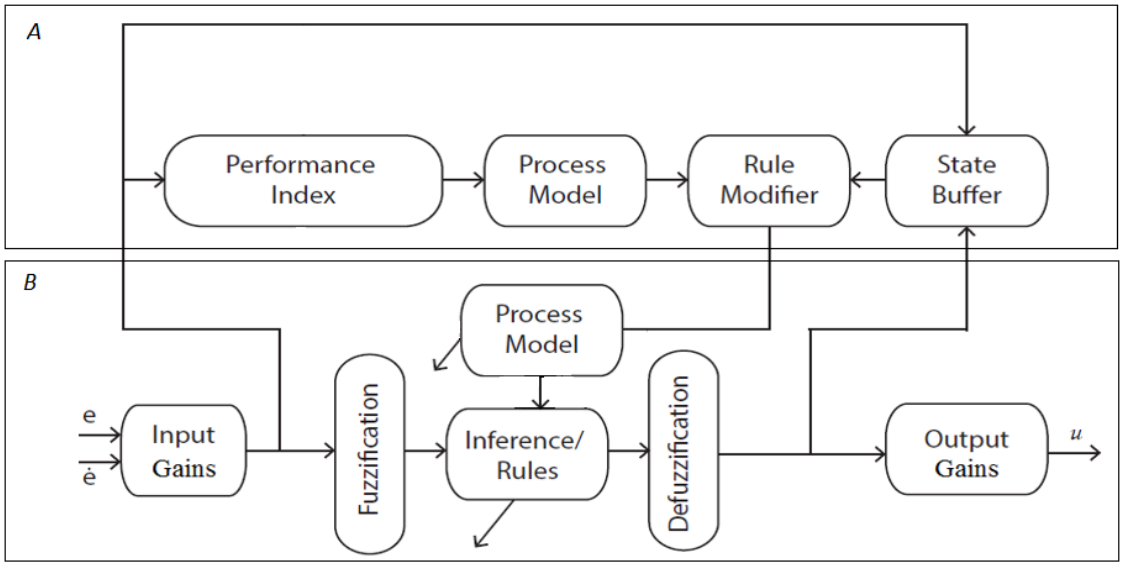


Figure 2.9: The structure of SOFLC.

2.5.1.1 Performance index table

A learning controller can only adapt its control structure to reach a predetermined quality if it manages to effectively evaluate its own performance. The SOFLC uses a PI table to carry- out this task. The PI table evaluates the performance of the system and determines the deviation from the desired trajectory. If required, it issues correction values to bring the output of the process to a desired level. The PI table is developed using the traditional linguistic statements. However, if the inputs of the controller are expressed as fuzzy singletons, then the PI table can also be represented by a 'look up' table (Mamdani and Procyk, 1979), as shown in Table 2.1.

Usually, the SOFLC evaluates the system performance by measuring the system output error (*E*), and change of error (*CE*) and it generates the modification value, *f*, to the lower-level fuzzy logic rule-base at the instance *(nT)*.

$$P_0(nT) = f(E(nT), CE(nT)) \tag{2.31}$$

The PI table is formed based on the knowledge of an operator or an engineer and aims at:

- Achieving fast recovery when the system experiences a disturbance.
- Good damping when the output of the system reaches the desired set-point.
- Confining the output of the system within a certain range around the desired set-point.

E \ CE	NB	NS	ZO	PS	PB
NB	-2.0	-1.5	-1.5	-0.5	0.0
NS	-1.5	-1.0	-1.5	0.0	0.5
ZO	-1.0	-1.0	0.0	1.0	1.0
PS	-0.5	0.0	1.5	1.5	1.5
PB	0.0	0.5	1.5	1.5	2.0

Table 2.1: A typical performance index table (Lu and Mahfouf, 2006).
NB: negative big, NS: negative small, ZO: zero,
PS: positive small, PB: positive big.

2.5.1.2 Rule modifier

So far, it has been shown how the output of the system under control can be used with the aid of the PI table to modify the control strategy of the controller. What is now required is showing how the correction values issued by the PI table are translated into reinforcements to control rules of the lower-level fuzzy logic rule-base. This procedure can be illustrated in the following pattern; it is assumed that for a process with a time lag of *m* samples, the inputs that were generated at *'nT-mT'* are responsible for the present undesirable performance of this process. Therefore, the original procedure,

$$E(nT - mT) \rightarrow CE(nT - mT) \rightarrow \ U(nT - mT) \tag{2.32}$$

should be modified to the following rule:

$$E(nT - mT) \rightarrow CE(nT - mT) \rightarrow \ U(nT - mT) + \ P_i(nT) \tag{2.33}$$

where $P_i(nT)$ is the correction value received from the PI table at (nT).

2.5.1.3 State buffer

As indicated above, in order for the controller to modify its structure, it needs to determine the rules responsible for any undesirable response. The state buffer is a recorder that registers the values of scaled error, called the change of error, as well as the output of the controller before scaling.

2.5.1.4 Process model

Controlling multiple input multiple output (MIMO) systems could be very challenging due to the interaction between inputs and outputs. Usually, each input has influences on all the outputs of the system. The SOFLC uses a model to measure the degree of coupling between the input and output of the process. For a MIMO process, the modification is carried-out in the following pattern:

$$P_i(nT) = M^{-1}P_o(nT) \tag{2.34}$$

where

$$P_i(nT) = \begin{bmatrix} P_i(nT) \\ . \\ . \\ . \\ P_q(nT) \end{bmatrix} \quad , \quad P_o(nT) = \begin{bmatrix} P_i(nT) \\ . \\ . \\ . \\ P_l(nT) \end{bmatrix}$$

$P_o(nT)$ is the correction value issued by the PI table while $P_i(nT)$ is the manipulated input variable to the process and M represents the incremental model of the system. Different studies and investigations for the choices of the process model can be found in different contributions such as (Mamdani and Procyk, 1979; Daley and Gill, 1986). When the SOFLC is used in the SISO case, a simple value of 1 is given to this model.

2.5.2 Controller parameters setting

2.5.2.1 Scaling factors

Generally speaking, fuzzy controllers are normally constructed by defining a set of parameters as discussed above. Issues may arise when choices for the controller structure are not chosen correctly or if the controllers are not properly tuned. Input and output scaling factors are common in fuzzy control systems and are used to tune and improve the performance of these systems. As can be seen in Figure 2.9, the SOFLC uses both input and output scaling factors to regulate its performance. Procyk and Mamdani (1979) discussed the impact of these scaling factors on SOFLC and how they can be used to alter the desired performance of the controller effectively. Daley and Gill (1986) proposed a different approach for determining suitable values for the scaling factors, where small values are initially given to these variables, which increase at different times until the desired response is achieved. This approach enhanced the capabilities of the controller and enabled it to control more complex and high-dimensional systems more effectively in terms of acceptable steady-state accuracy and transient behaviour.

An alternative approach was developed by Linkens and Abbod (1992) who proposed formulas based on a series of trials they conducted on various systems, such as an anaesthesia model and an air heating system. The new formulas resulted in suitable scaling factors which are sufficient for different applications. Chou and Lu (1994) proposed a new mechanism for the output scaling factors to be tuned on-line; this approach was employed in (Rojas *et al.,* 1999). In this strategy, a learning rate that decreases with time exponentially is used; hence, the impact the scaling factors have on the system decreases with time. Simulation results showed how this rule enables the scaling factors to enhance the control action in initial iterations when the control rules are not well designed while their effect decreases when the response approaches the desired point, enabling the controller to run smoothly in this region. Mahfouf *et al.* (2000) added a supervisory layer to the SOFLC in the form of a knowledge-based protocol to govern the scaling factors selection task, this was carried-out based on the system behaviour in terms of different variables such as rise time and the overshoot.

2.5.2.2 Selection of inputs for the performance index table

The performance evaluation task relies solely on the input signals generated to the performance index table; therefore, the selection of these variables play a major role in the how well the controller performs in terms of simplifying its structure and speeding up the response of the system.

Generally, the inputs to the PI table at every sampling instant are taken as the error and change of error (Procyk, Mamdani, 1979; Sugeno, 1984; Mahfouf *et al.*, 2000). However, Kim (2000) developed a new mechanism for the rule modification of fuzzy rules, which uses the so-called sliding mode character s and its derivate $\dot{s}$ as the input variables to the performance index table. Simulation results demonstrated that the proposed algorithm has managed to effectively control different non-linear systems such as circular inverted pendulum systems and a two-link robot manipulator.

2.5.3 Adding and deleting control rules

In order for the SOFLC to improve its performance, it continuously modifies, adds, or deletes control rules depending on the behaviour of the system under control. New rules enable an effective enforcement of control actions while deleting some old rules makes the rule-base able to have enough room to reflect the dynamics of the system under control and also prevents conflicts between rules and rule explosion syndrome.

2.5.3.1 Adding a rule to the lower-level fuzzy logic rule-base

The control rules in the original SOFLC scheme are stored in a rule bank and the criterion for adding rules to the rule bank in the lower-level fuzzy logic rule-base can be stated as: A new rule can be added to a particular cell in the rule bank if this cell does not contain a rule already, otherwise the existing rule will be replaced by the new one (Procyk, Mamdani, 1979). However, there has been a significant amount of work reported in the literature dedicated to the development of new approaches for adding new rules to rules banks. Lee (1990) used the firing strength of control rules as a criterion to govern the rule additions. If at a particular instant, the strength of the rule is smaller than a predefined threshold, then a new rule needs to be added to the rule-base

as the current rules are not enough to enable the controller to perform well. Leng *et al* (2005) utilised the process error that is defined by the absolute difference between the actual output of the process and the reference output as a criteria for adding new rules. In this algorithm, fuzzy rules are added to the lower-level fuzzy logic rule-base or the width of some membership functions are modified if the process error is smaller than a predefined threshold.

2.5.3.2 Deleting rules from the lower-level fuzzy logic rule-base

The SOFLC algorithm avoids the explosion of the number of fuzzy rules by deleting any unimportant rules. In the original SOFLC algorithm, a simple approach is used to detect and delete the unimportant rules, which is based on the criterion that newly generated rules have the priority to survive. In the event of conflicting rules, the existing rule will be replaced by the new one.

Another approach for managing rules is the fixed Maximum Number of Rules (FMNR), which uses the difference between the actual output of the process and the desired output for rule selection when there are two rules for the same cell (Park *et al.*,1995). In this approach, as new input-output data is gathered, the number of rules increases monotonically and converges to a finite number. This approach requires a model of the process and can therefore only be used for off-line applications of the SOFLC. The work of Park was developed by Dias and Dourado (1999) who proposed a new technique which combines both the FMNR and an adaptation mechanism for the parameters. Simulation results show the proposed algorithm performs well with different non-linear systems in both the SISO and MIMO cases.

2.6 Supervisory intelligent Control Systems

Supervisory layers provide integrated control and network management to fuzzy controllers. They enable controllers to rely less on *priori* design information and rather use the dynamics of the system under control to update and modify its structure till the desired response is achieved. This advantage allows controllers to control more satisfactory performance even when they are applied to very complex and highly

uncertain systems. Different approaches are available in the literature that studied and further developed the original SOFLC and also resulted in new schemes.

2.6.1 Fuzzy controllers using performance index tables

In order to improve the capabilities of the original SOFLC in terms of robustness, stability, and control abilities, different techniques have been used to further study and develop the PI table. Layne and Passino (1996) incorporated the idea of conventional adaptive control in the SOFLC to develop the so called fuzzy model reference learning controller (FMRLC). This scheme uses a reference model that defines how the designer would like the process to behave. The input signals of the performance index table are the error $e(nT)$ and change of error $\dot{e}(nT)$, which are defined in terms of the process output $y(nT)$ and reference model output $y_l(nT)$ as follows:

$$e(nT) = y_l(nT) - y(nT) \tag{2.35}$$

$$\dot{e}\,(nT) = e(nT) - e(nT - T) \tag{2.36}$$

When applied to a two-degree of freedom robot manipulator and rocket system, the controller exhibited good system behaviour in terms of good steady-state and transient response. The FMRLC was studied and developed intensively in various contributions and was applied to different systems (Golea *et al*., 2002; Cermen, 2013; Li, 2011).

A new approach for designing a PI based SOFLC was developed by Polkinghorne *et al* (1994) where, rather than using the standard performance index, the controller used enhancement matrices, one for each input variable, which contain the control information of the process under control. The controller was applied on a small vessel with two inputs. The simulation showed the new proposed mechanism provided more precise modification of control rules than the traditional one.

Another interesting technique in the literature uses a multi-stage SOFLC scheme where m-input/n-output SOFLC system is decomposed to many 2-input/1-ouput sets (Chou *et al.*, 2010). This decomposition allowed the performance index tables to be constructed in simple 2-input/1-ouput spaces as those of the original SOFLC scheme. The proposed scheme was designed to control the anaesthesia and muscle relaxation of human bodies.

Simulation results demonstrated the algorithm provided a good performance with reduced steady-state error.

2.6.2 Fuzzy controllers using performance measure functions

An alternative approach to enforce the rule modification in the lower-level fuzzy logic rule-base can be carried-out using a learning algorithm as the one proposed by Yang (1992). In this algorithm, both the error and change of error of the process under control are used to construct a learning algorithm that replaces the performance index table as shown below:

$$\Delta\bar{u}(k) = \gamma[(1-\varsigma)e(k) + \varsigma\dot{e}\,(k)] \tag{2.37}$$

where $e(k)$ is the tracking error and $\dot{e}\,(k)$ is the error change, ς is the weighting distribution value and γ is the learning rate.

The learning rate governs the process by which the correction values are calculated. If large values are given to the learning rate, the algorithm may excessively adapt the control rules and this might result in oscillatory phenomena behaviour during the control process. In contrast, if small values are assigned to the learning rate, the learning law will respond slowly to the changes of the system and will fail to provide the modification needed.

Since inappropriate selection of the design parameters, i.e., the weighting distribution and the learning rate, can lead to learning instabilities, several studies emerged that further studied and developed the algorithm. In one study, a stability analysis was performed and the learning law was modified, both the error and change of error shown in equation 2.37 were replaced by fuzzy values (Huang and Lee, 2000). The idea behind this is to avoid any unnecessary rule modification by forcing e and $\dot{e}$ to reach zero simultaneously. When the modified algorithm was tested on a robotic system, it showed good capabilities in terms stable learning and fast motion control. Li *et al* (2015) added a grey-predictor to the algorithm to enable it to estimate the output of the process, producing a controller capable of pre-correcting control rules to reasonable ones with better abilities to compensate for the dynamic coupling effects between its inputs and

outputs, making it applicable to work in the multivariable case. This work was further developed by Lian (2012) to include what he called the enhanced adaptive grey-prediction algorithm to overcome the stability issues. The experiments on the 6-DOF robot proved the suitability of the proposed algorithm, which provided better control performance than the original.

There were several other studies conducted to propose new techniques to develop the SOFLC schemes using a performance measurement function. Neural networks were expansively used as a self-organising strategy in SOFLC schemes. Lin and Lee (1991) developed a back-propagation network (BPN) based SOFLC that performs the rule modification task by combining both a supervised and an unsupervised learning techniques. When implemented, the proposed algorithm proved to have fast learning capabilities without requiring extensive information of the dynamics of the process. Li and Tan (1994) proposed a three-layer back propagation neural network that optimizes the controller by searching for the optimum points of a vector that determines the shape of the membership functions in the lower-level fuzzy logic rule-base. The weights of the neural network are modified using the Widrow-Hoff learning rule. The algorithm was applied to a two-link manipulator. The simulation results and the experiments proved the effectiveness and the robustness of the new proposed algorithm.

In recent decades, a new neural network based SOFLC was proposed where a radial basis function neural network (RBFN) was used to adjust the parameters of the controller in real-time (Lian, 2011). The proposed algorithm also alleviates the coupling between the inputs and outputs because the coupling weighting of the RNFN has regulation capabilities. Simulation on the 6-DOF robotic system showed that the new scheme provided a good performance even when no prior knowledge of the process was known.

2.6.3 Fuzzy controllers using fitness function

In this category of controllers, the control rule is not directly modified by the performance measurement. Rather, a fitness function that is subjected to the performance measurement is used to optimise the system.

Extensive studies have been conducted in the literature on how genetic algorithms (GA) can be used to tune and design the structure of the SOFLC. Generally, GAs have mainly been used to perform two tasks. First, they were used to optimise the membership functions and control rules of the controller and the second task involved using them as a mechanism to add and delete control rules and eliminate conflicts between these rules.

Pal (2003) proposed a GA-based SOFLC scheme that uses GA to add and delete fuzzy rules. This algorithm included the number of rules in the fitness function to prevent rule explosion and to ensure the size of the rule-base remain within an acceptable limit.

In a new scheme, a multi-objective GA has been used to tune the performance index table of the SOFLC (Mahfouf *et al*., 2000). Simulations on a non-linear muscle-relaxant anaesthesia model, lead to more effective set of parameters and demonstrated the robustness of the algorithm.

Linkens and Nyongesa (1995) developed a SOFLC scheme that has the ability to tune its parameters in real-time using an on-line GA method. The GA algorithm performed this task without the need for a model and this was achieved with the aid of the idea of credit assignment and fitness estimation. Satisfactory performance was demonstrated when the new scheme was apply to a multivariable process. However, due to the poor mechanism used to evaluate the performance of the system, a small yet significant steady-state error was produced. This algorithm was furthered studied and developed by Lu and Mahfouf (2006) where a better mechanism in the form of a ternary representation was used for the evaluation of the process. This led to better results in terms of robustness and control capabilities when it was tested on SISO and MIMO anaesthesia models.

2.7 Summary:

This chapter provided a comprehensive overview of the concepts and methods used and discussed throughout the thesis. Five main areas were described. First, the chapter began by providing a solid background on the theory of type-1 fuzzy sets and fuzzy logic systems .This included defining various fuzzy set operations. Thereafter, as an extension to type-1 fuzzy systems, type-2 fuzzy logic systems were introduced. Both the general type-2 and interval type-2 fuzzy sets were introduced, with particular focus on the concepts and operations associated with the latter. The basics of the traditional fuzzy

logic control was then defined and explained. The fourth area covered in this chapter concerned the structure and design of the SOFLC Algorithm. This included describing the different element of the controller, its operation mechanism and how it differs from the conventional fuzzy controllers. The last part of the chapter outlined the various supervisory intelligent control systems, and the various mechanisms they use to modify the fuzzy control rules.

Chapter 3- Self-Organising Fuzzy Logic Control with a Dynamic Supervisory Layer

3.1 Introduction

The self-organising fuzzy logic controllers described in Chapter 2 represent the extended version of the fuzzy logic controller with an additional control policy that enables them to modify their structure based on the dynamics of the system under control and the environment in which they operate. The SOFLC usually uses two rule-bases: the first is the lower-level fuzzy logic rule-base which is the same as the one included in the basic traditional fuzzy controllers and is used to issue fuzzy control actions, while the second is responsible for evaluating the performance of the controller and altering the control rules until the desirable response is achieved.

The SOFLC schemes have been studied and investigated extensively and generally lead to good performance across a wide range of complex systems. Despite this success, they can also have different disadvantages, such as high computational costs and high storage requirements, particularly when applied to MIMO systems. However, their main drawback lies in the difficulty in constructing the performance index table that remained, in most applications, unchanged since it was first introduced by Procky and Mamdani (1979). The design of the performance index requires a *priori* system information. As a result, when a standard performance index table is used with a wide range of systems that all have different dynamics, this will undoubtedly lead to sub-optimal policies for the alteration of control rules, as well as high computation times as the dimension of the input/output space increases.

An alternative SOFLC architecture with a dynamic supervisory layer, referred to as SOFLC-DSL, that uses an on-line particle swarm optimisation (PSO) mechanism to adapt the consequent part of the performance index table is proposed in this chapter (Kennedy and Eberhat, 1995). This task is accomplished using the idea of fitness estimation and credit assignment. The new scheme is tested on non-linear, mathematically ill-understood, and uncertain systems under different conditions and environments.

3.2 Particle Swarm Optimisation

3.2.1 Introduction

Particle swarm optimization is a bio-inspired intelligence technique that was first introduced by Kennedy and Eberhat (1995). The algorithm is inspired by social behaviour patterns observed in birds flocking, bees or fish, or more specifically, the collective behaviours of simple individuals interacting with each other and the environment around them (Eberhart and Shi., 1998). PSO exploits a population called a ‘swarm’, which includes a set of individuals, named ‘particles’.

Similar to other stochastic methods, the PSO is a population-based optimisation technique (Allaoua *et al*., 2009). When it is applied, the system is initialised with a swarm of random solutions and then begins to search for optimal (or near optimal)

solutions by updating generations. The fitness function to be optimised is used to evaluate all the particles, which all have fitness values as well as velocities that direct their flying within the search space (Wang *et al.*, 2006). In a PSO system, as long as the computational limitations are not exceeded, all the individuals keep updating their positions by flying at a certain velocity in a multi-dimensional search space.

The PSO compared with other methods, can result in better quality solutions within a more stable convergence and faster calculation time characteristics. Furthermore, it can be easily implemented in most programming languages. PSO has been successfully applied in literature for approaches that can be used in a wide variety of applications and has demonstrated its effectiveness in a diverse range of benchmark optimisation problems (Mahfouf *et al*., 2006; Zhang *et al*., 2012; Zhang and Mahfouf, 2011).

3.2.2 Classical PSO algorithm

In a PSO system, a swarm of particles flies around the problem space. Each particle keeps track of its coordinates in the search space through keeping the most successful particles it has so far achieved. This variable is known as the local best solution (*pbest*). Another important value to the algorithm that keeps track of is the so-called global best solution (*gbest*), which denotes the best solution obtained so far from any group. Depending on this information, each particle within the population recognises how well it is performing, and performs in tandem with other particles in the swarm.

Since the PSO was first introduced, different versions of the associated algorithm have been proposed subsequently (Poli *et al*., 2007). The version used in this thesis is the one modified by Shi and Eberhart (1998). Each particle, *i*, is represented in the PSO algorithm by a position vector x_i' as well as a velocity vector v_i', which are both updated as follows:

$$v_i(t+1) = w(t)v_i(t) + c_1 r_1\big(pbest_i(t) - x_i(t)\big) + c_2 r_2(gbest(t) - x_i(t)) \quad (3.1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3.2)$$

where x_i and v_i represent the positions and the velocities of the particles respectively, $pbest$ is the best position to the time *t*, while $gbest$ is the global best position achieved so far; w is the inertia weight, which usually decreases linearly from 0.9 to 0.4; $c_1$and c_2

are called the acceleration coefficients, and are normally set to 2.0; while r_1 and r_2 are random numbers in the range [0, 1]. Algorithm 3.1 includes the details of the original PSO mechanism.

The trajectory analysis of a certain individual within the fitness landscape is illustrated in Figure 3.1. In a PSO system, particles share information with other neighbouring particles. The second and third components of the equation 3.1 are called cognition and social elements, respectively. From the updated equations 3.1 and 3.2, one would realise that PSO systems combine both the cognition component of each individual with the social component of individuals within a group. The social element suggests that on the one hand, particles ignore their own experience and adjust their behaviour based on the previous best individual known so far. While, on the other hand, the cognition element adjusts particles based on their own experience.

Algorithm 3.1 PSO Algorithm

```
For each particle, generate the initial position and velocity randomly;
Assess the fitness of each particle;
repeat
    for each particle i do
        Update position and velocity of particle i according to equations (3.1) & (3.2);
        if f(x_i) < f(pbest_i) then
            pbest_i=x_i;
            if f(x_i) < f(gbest) then
                gbest_i=x_i;
            end if
        end if
    end for
until the stop criterion is satisfied
```

3.2.3 On-line implementation of PSO

Particle swarm optimisation algorithms have been widely used in a variety of applications including control engineering for controller design, system identification and fault diagnosis. However, the PSO algorithms have all been applied off-line in these applications.

PSO has been classically proposed for use in off-line optimisation. In off-line applications, swarms which contain sets of particles that represent different solutions

evolve for a number of generations before they generate the best solution, which is then used to produce system output. For example, when a PSO algorithm is utilised to tune the parameters of a PID controller (Oi *et al*., 2008), the optimisation is carried-out off-line using a mathematical model that represents the dynamics of the system to be controlled. All the particles at each iteration are evaluated via a fitness function to test how effective they are in terms of controlling the process; the optimal particle that produces the best results off-line is then used for the real system.

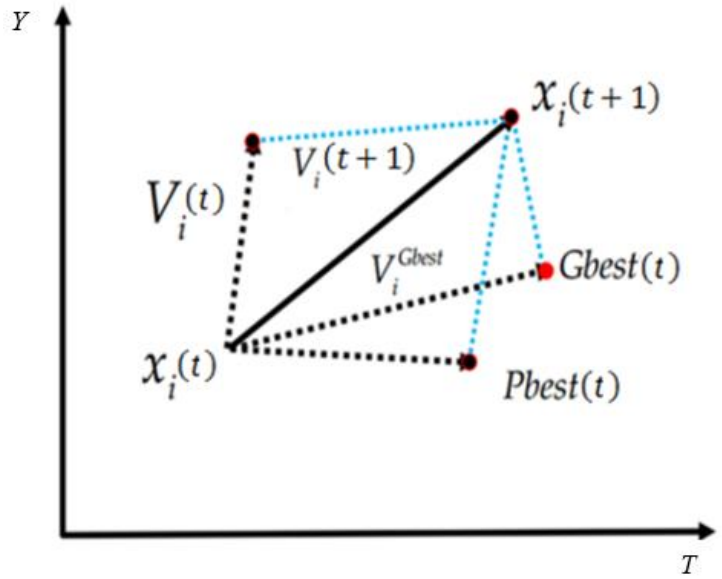


Figure 3.1: Particle trajectory analysis in PSO system.

Various issues normally arise if the PSO algorithm is used to tune the parameters of the PID controller on-line. These include:

- In on-line optimisation, the system cannot use a model to evaluate the performance and assign fitness values to all the particles within the swarm, hence, these fitness values are provided based on noisy feedback signals.
- In on-line implementation of the PSO algorithm, the system must provide an appropriate control action at every sample instant. Therefore, the algorithm can only evaluate one particle in each iteration.
- For the PSO to converge and produce an optimal solution, it normally needs a few iterations; this is sometimes not possible in on-line applications due to the limited amount of computations that can be carried between sampling instants.

3.3 Self-Organising Fuzzy Logic Control with a Dynamic Supervisory Layer

The new proposed architecture uses a new PSO algorithm to modify the consequence parts of the performance index table of the SOFLC at every sampling instant. The structure of the proposed scheme is shown in Figure 3.2.

3.3.1 The PSO process encoding

A performance index table with 25 cells is used in the proposed algorithm. The tracking error and the change of error are taken as the inputs of the performance index table, while the output that the PI table generates is the rule modification value, $P_i(nT)$, of the low-level basic fuzzy logic controller. Independent sets of five particles, are used to optimise each cell of the PI table. 5×25 individuals and velocities are randomly generated in the first generation. In addition, all individuals within all the sets are given the same fitness values. In order to fill all the cells of the performance index table, a randomly chosen individual is chosen.

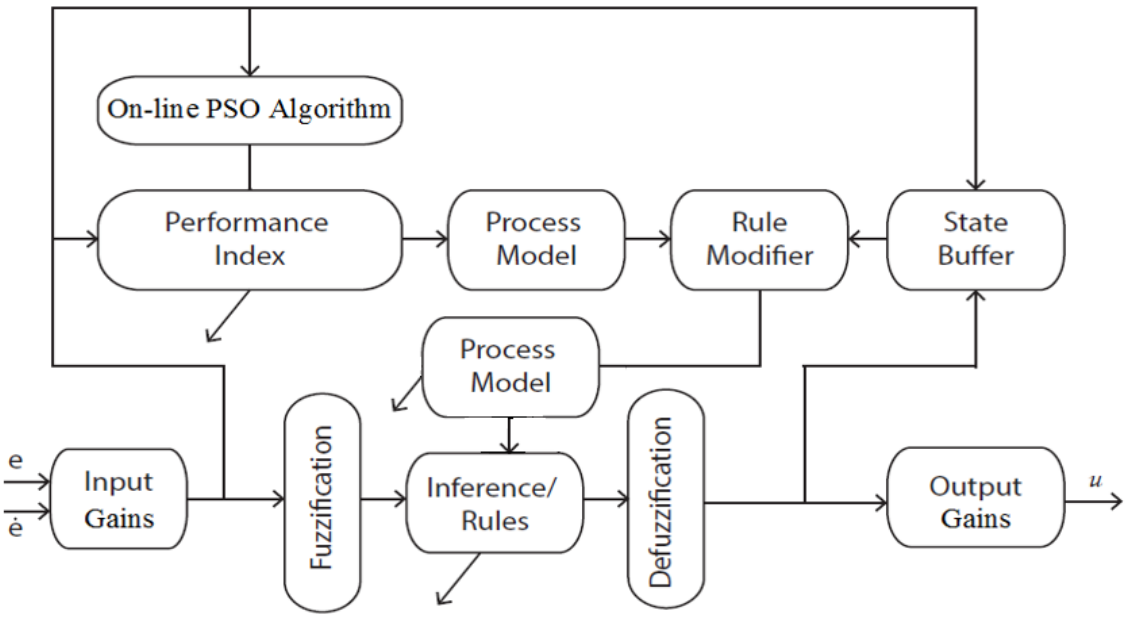


Figure 3.2: The structure behind the SOFLC-DSL algorithm.

An illustration of how the proposed algorithm modifies the fuzzy rules and generates fuzzy rules is shown in Figure 3.3. At the sampling instant '*nT*', the cell *F24*, which was responsible for providing the modification value $P_i(nT - mT)$ that was generated at ' $nT - mT$', is explored again. The five individuals included in cell will experience one iteration using equations 3.1 and 3.2 after being given various rankings based on the estimated fitness values, resulting in new positions and velocities being generated for each particle. If this cell is indexed again, the shaded particle '0.6' in set *B*, for example, which represents the optimal particle with the highest fitness value, will be chosen to generate the modification value. In the meantime, cell *F41* is responsible for generating the modification value $P_i(nT)$ to the lower-level fuzzy logic rule-base. The shaded particle '0.2' in set *A* is the optimal particle and will be chosen to provide the modification value.

3.3.2 The on-line PSO algorithm operations

As stated above, various obstacles are normally faced when the PSO is implemented on-line, some of which are outlined in the previous sections. In order to overcome these constraints, a new version of the PSO is proposed in this thesis. In the new algorithm, only one particle from a set of individuals is measured by the algorithm while the remaining particles within the set are estimated with the assistance of the idea fitness estimation and credit assignment. The estimation of these particles is carried-out based on their relationship with the optimal particle.

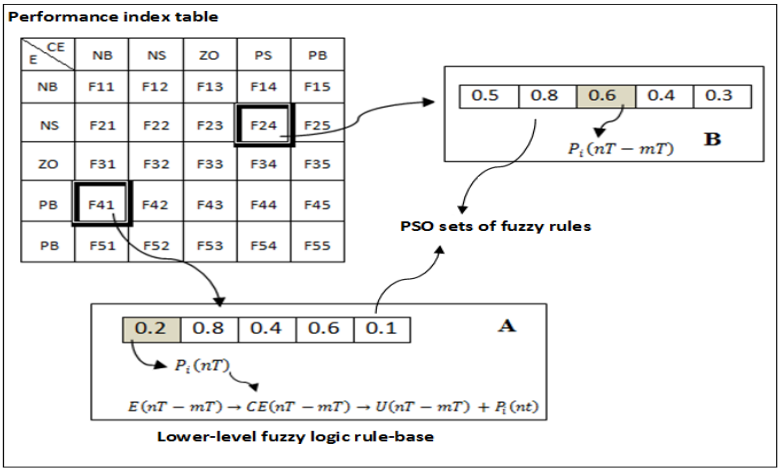


Figure 3.3: The self-organised information flow in the new proposed algorithm.

3.3.2.1 Performance assessment

For the system to improve its performance under various conditions, it needs to modify its structure at each sampling instant; generally speaking, two methods can be used to carry-out this task. First, the global criterion, such as 'Integral Absolute Error (IAE)', which evaluates the performance of the process to be controlled over a complete response trajectory. This method of assessment cannot be used when the PSO is implemented on-line as an accurate evaluation of the contribution of each particle at each sampling instant cannot be obtained. Therefore, this method is only sufficient for off-line performance evaluations.

An alternative method of performance assessment is a local criterion, where, at the current sampling instant, the evaluation of the performance is only carried-out over limited neighbour states. The performance of the applied particle is only assessed by the evaluation index at that particular time, generally in the form of the binary 'good' or 'bad'. A major disadvantage of local performance assessment is that conflicting decisions are likely to be assigned the same particle during different sampling instants.

Linkens and Nyongesa (1995) used predictive error function to predict the future tracking points and provide binary good/bad performance evaluation so that corrective actions are taken in advance to avoid any undesirable deviations from the target. The binary assessment of this method does not provide information on how good or bad the performance is, information that is vital in order to provide effective rules modification. Although satisfactory results are normally achieved with this assessment method, the modified version of this technique proposed by Lu and Mahfouf (2005) was adopted in this thesis. In this assessment technique, a straightforward ternary representation replaces the binary performance evaluation where all the scenarios that the responses of the output can take are classified into three categories, as shown in Figure 3.4. In scenario (*A*), the output response of the system is considered 'satisfactory' at that particular sampling instant if the predicted tracking error is smaller than the current tracking error, regardless of the trend. In scenario (*B*) the output response is considered to be an 'overshoot' if the trajectory passes across the desired target. In scenario (*C*), the output response is referred to as 'moving away' if the response is moving away from the target point. The performance classification of this method is shown in Figure 3.5.

The predictive error function is generally defined by the simple formula as follows (Linkens and Nyongesa, 1995):

$$\hat{e}(nT + kT) = e(nT) + kT\dot{e}(nT) \tag{3.3}$$

where $e(nT)$ and $\dot{e}(nT)$ are the error and the velocity of the process respectively at the sampling instant *'nT'*; *k* is the number of steps predicted ahead.

However, it is worth noting that adding the error acceleration $\ddot{e}(nT)$ to the formula in equation 3.3 results in a more accurate state estimation.

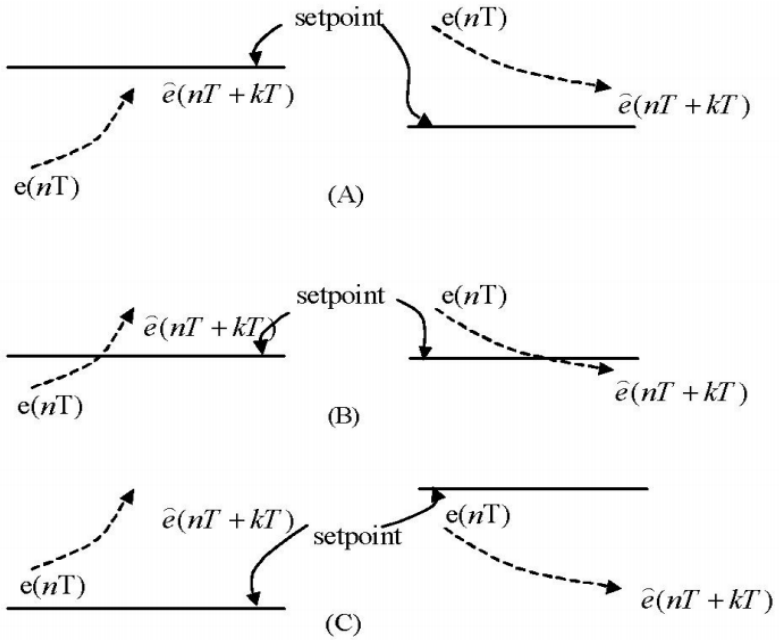


Figure 3.4: Instant state performance categories.

If *sgn* (*e*(*nT*)) = *sgn* ($\hat{e}$(*nT*+*kT*)) *Then*
    If $|e(nT)| > |\hat{e}(nT + kT)|$ Then
        The performance is Satisfactory
    Else
        The response is not tracking the set-point (Moving away)
Else
    The response will overshoot

Figure 3.5: Performance classifications.

3.3.2.2 Credit assignments

The performance assessment illustrated in the previous section represents a measurement of the optimal individual F_i, which is provided by the feedback signals at each sampling instant. In order for the fitness values to be compared in this generation, the algorithm needs to be able to determine the rate of usefulness of the remaining *'N-1'* individuals within the indexed F_{ij} set that cannot be measured. This task is carried-out by the credit assignment using reward-penalty mechanism.

Since different particles within any cell represent different modification values, the possible evaluation (satisfactory, moving away, or overshoot) of these individuals from the optimal particle F_i can be inferred. With such information, a reward or a penalty can be provided to each particle; the degree of reward/punishment of any particle depends on its distance from the optimal particle F_i. The process of the credit assignment task is outlined in Figure 3.6. From this mechanism one can notice that all the rewards and punishments generated in the process work towards forcing the error to converge to 'zero'.

IF System Performance Evaluation is Satisfactory Performance
IF control action is an Increment, THEN punish small
IF control action is a Decrement, THEN punish big
IF Performance Assessment is Moving Away OR overshoot
IF control action is Increment, THEN punish big AND equal and reward small
IF control action is a Decrement, THEN punish small AND equal and reward big

Where
Reward/Punish big: reward/punish those particles giving a larger control action
Reward/Punish small: reward/punish those particles giving a smaller control action
Rewards/Punish equal: reward/Punish those particles giving an equal control action.

Figure 3.6: The credit assignment mechanism used to estimate the PSO individuals.

3.3.3 A Summary of the SOFLC-DSL Algorithm

The operation cycle of the SOFLC DSL that is optimised by an on-line PSO algorithm via fitness estimation and credit assignment is summarised as follows:

1. *N* sets of particles are initially generated to fill all the cells of the performance index table; each of which include *M* particles along with their corresponding velocities. In the initial generation, all the particles are given the same fitness value.
2. In the first sampling instant, each cell of the performance index table is linked with a particle that is selected randomly from the *M* particles provided in each cell. When this cell is visited by the algorithm for the first time, this particle is used to provide the lower-level fuzzy logic rule-base with a modification value, $Pi(nT)$.
3. For a system with *mT* sampling time delay, at the sampling instants *nT*, two input signals *E(nT-mT)/CE(nT-mT),* and *E(nT)/CE(nT)* are forwarded to the SOFLC-DSL and two tasks are then performed.
   a) The on-line PSO algorithm optimises the cell F_{ij} that was activated at the sampling instant '$nT - mT$'. If n<2m, omit this stage and jump to stage b). The particle with the highest fitness value is marked as F_i.
      - The performance classifier shown in Figure 3.5 is used to assess the contribution of the optimal particle F_i. The remaining particles and velocities within that cell are then estimated using the credit assignment mechanism shown in Figure 3.6.
      - Equations 3.1 and 3.2 are then generated in order to produce new particles and velocities.
      - The optimal particle F_i is then linked to the corresponding cell F_{ij} so that it can be generated to the lower-level fuzzy logic rule-base when this cell is visited again.
   b) The new modification value $P_i(nT)$ is provided to the lower-level fuzzy logic rule-base based on the inputs signals *E(nT)/CE(nT)* using the equation 2.33
4. The lower-level fuzzy logic controller is then used to calculate the control action *U(nT)* before it is applied to the process under control to produce new feedback

signals. The algorithm stops if the termination criterion is satisfied, otherwise it returns to step 3.

3.4 Simulation on a Muscle Relaxant Model

During an operation, to obtain a predefined degree of paralysis, patients are normally given a certain dose of muscle relaxant drugs. This task is generally carried-out by an experienced anaesthetist that often fails to maintain a steady level of relaxation required in the process. An alternative and a safer approach is to use a closed loop control system instead.

The overall linear transfer function of muscle relaxant model can be defined as (Mahfouf and Linkens, 1998):

$$G(s) = \frac{X_E(s)}{U(s)} = \frac{k_1(1+T_4 s)e^{-s}}{(1+T_1 s)(1+T_2 s)(1+T_3 s)} \tag{3.4}$$

With the following non-linearity:

$$E_{eff} = \frac{X_E^{2.98}}{X_E^{2.98}+(0.404)^{2.98}} \tag{3.5}$$

Where K_1=1; T_1=34.4 min; T_2=4.8 min; T_3=3.08 min; T_4=10.65 min; X_E is the drug concentration in the blood and E_{eff} is the actual output, which is the muscle relaxation.

The simulation study in this thesis uses a sampling interval of 1 minute and a step length of 0.1. All the initial conditions of the muscle relaxant model are set to zero. Furthermore, the lower-level fuzzy logic controller uses five fuzzy sets, denoted in an equally-partitioned universe of discourse, for both input signals E and CE: negative big (NB), negative small (NS), zero (ZO), positive small (PS), and positive big (PB). In order to ensure the computational cost is kept at a lower level, the performance index table is also divided into 25 cells, each of which corresponds to a particular fuzzy rule. All cells of the the performance index table start empty and are then filled with 25×5 PSO particles in the initial generation.

In this research, a target-point signal of 85%, 65%, and then 85% muscle relaxation is used. The cells of the performance index table were optimised on-line using the algorithm outlined in Section 3.3.3.

The SOFLC-DSL and the standard SOFLC scheme that has a fixed performance index table were both tested and compared as shown in Figure 3.7 and Table 3.2. As stated in the previous sections, the SOFLC-DSL starts with an empty performance index table while Table 3.1 shows the fixed performance index table that standard SOFLC uses.

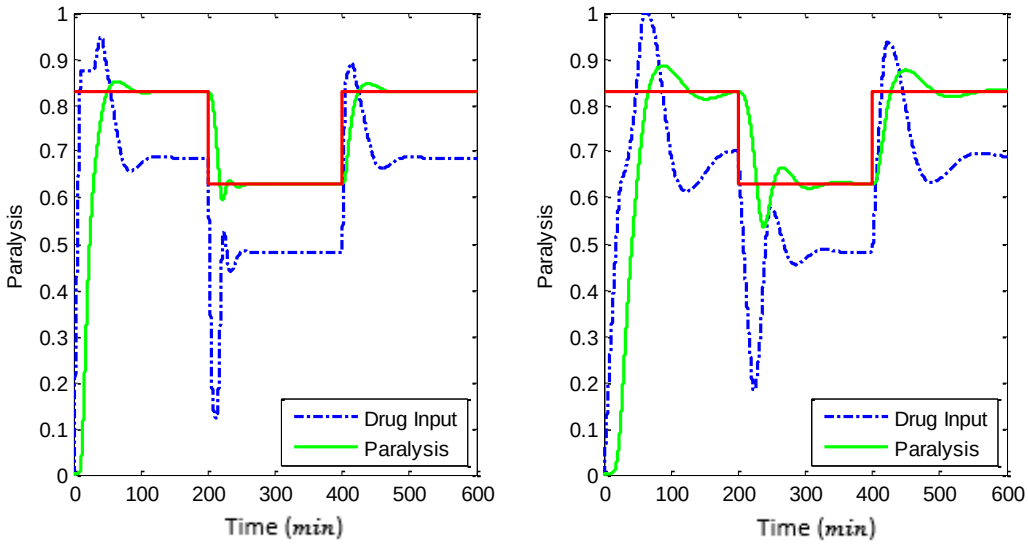
The simulation results reveal that, when compared to the standard SOFLC, the proposed SOFLC-DSL provides better control capabilities in terms of making the system track the target-point more effectively with less undershoot. Furthermore, the self-organising mechanism of the SOFLC-DSL provides the superior system performance with a smaller number of fuzzy rules produced to the low-level fuzzy logic controller, which leads to reduced computations.

The improved performance of the SOFLC-DSL is a result of a more accurate modification mechanism that is attributed to its ability to search a wider search space to achieve a more effective performance index table with a lesser degree of dependency on the knowledge and experience of the expert/operator (an anaesthetist in this case).

E \ CE	NB	NS	ZO	PS	PB
NB	-2.0	-1.5	-1.5	-0.5	0.0
NS	-1.5	-1.0	-1.5	0.0	0.5
ZO	-1.0	-1.0	0.0	1.0	1.0
PS	-0.5	0.0	1.5	1.5	1.5
PB	0.0	0.5	1.5	1.5	2.0

Table 3.1: Performance Index table used by the standard SOFLC scheme (Lu and Mahfouf, 2006).

Notes: E: tracking error, CE: change of the tracking error, NB: negative big, NS: negative small, ZO: zero, PS: positive small, PB: positive big.



System with SOFLC-DSL System with the standard SOFLC

Figure 3.7: Simulation result of the proposed scheme and the standard scheme.

Criteria	SOFLC-DSL	Standard SOFLC
IAE	345.84	423.51
ISE	177.15	199.62
Rule number	15	21

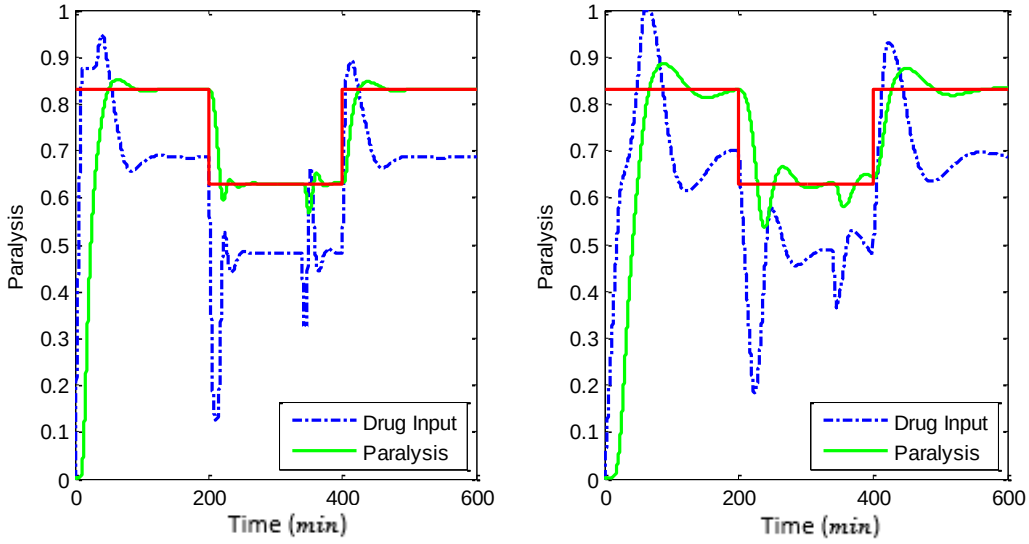
Table 3.2: Summary of performance criteria of the SOFLC-DSL and the standard SOFLC.

3.4.1 Sensitivity of the SOFLC-DSL to sudden disturbances

To investigate the robustness performance of the system and how the SOFLC-DSL responds to on-line parameter changes without the need of re-tuning, sudden disturbances of 10% and 15% , in the form on an impulse that lasted for 5 minutes, were introduced in the system at 350 minutes.

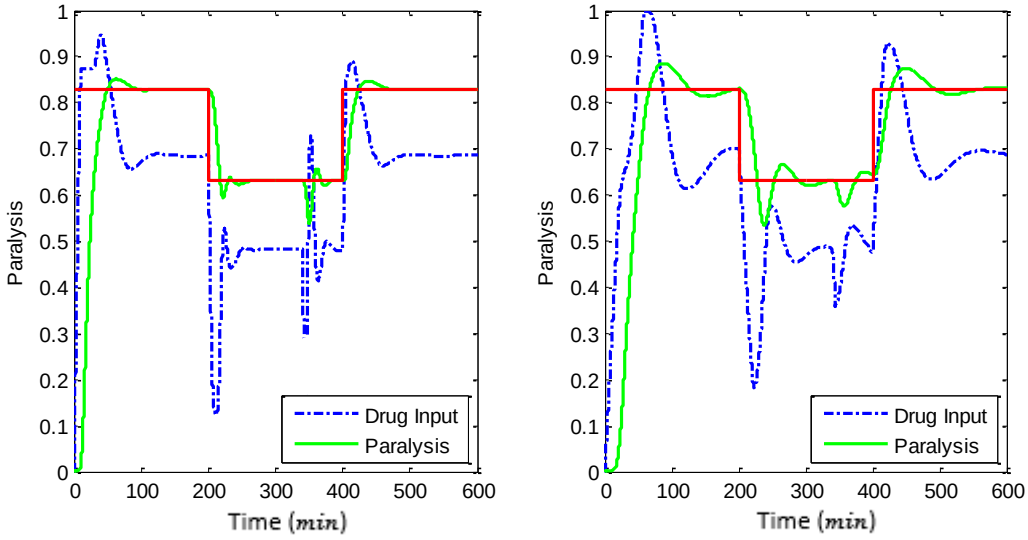
As shown in Figures 3.8 and 3.9, unlike the standard SOFLC that uses a fixed performance index table, the SOFLC-DSL managed to reduce the residual tracking error and bring the system output to effectively track the set-point. However, the standard

SOFLC failed to make the output re-track the set-point after the introduction of disturbance for nearly 350 minutes. Table 3.3 also shows how the SOFLC-DSL performs better under the Integral Absolute Error (IAE) and Integral Square Error (ISE) criteria and how it manages to control the process using a lower number of rules.



System with SOFLC-DSL System with the standard SOLC

Figure 3.8: Simulation result of the proposed scheme and the standard scheme undertaking disturbance of 10%.



System with SOFLC-DSL System with the standard SOLC

Figure 3.9: Simulation result of the proposed scheme and the standard scheme undertaking disturbance of 15%.

	Criteria	SOFLC-DSL	Standard SOFLC
Figure 3.8	IAE	345.84	513.57
	ISE	207.18	214.66
	Rule number	17	23
Figure 3.9	IAE	349.89	518.81
	ISE	210.15	219.69
	Rule number	16	23

Table 3.3: The performance of the SOFLC-DSL with disturbances of 10% and 15%.

3.4.2 Controller sensitivity to scaling factors

The sensitivity of the proposed algorithm to different scaling factors is tested in this section. In doing that, different groups of scaling factors for the error, change of error, and output are used and the results are compared to that of a standard SOFLC scheme. These groups were selected randomly in order to test the control abilities of the controller when it encounters random changes in the environment in which it operates.

Group A: used to test the sensitivity of the SOFLC-DSL to changes in the scaling factors of error, *GE*.

1) *GE= 20.0; GC=220;GT=0.4*
2) *GE= 30.0; GC=220;GT=0.4*

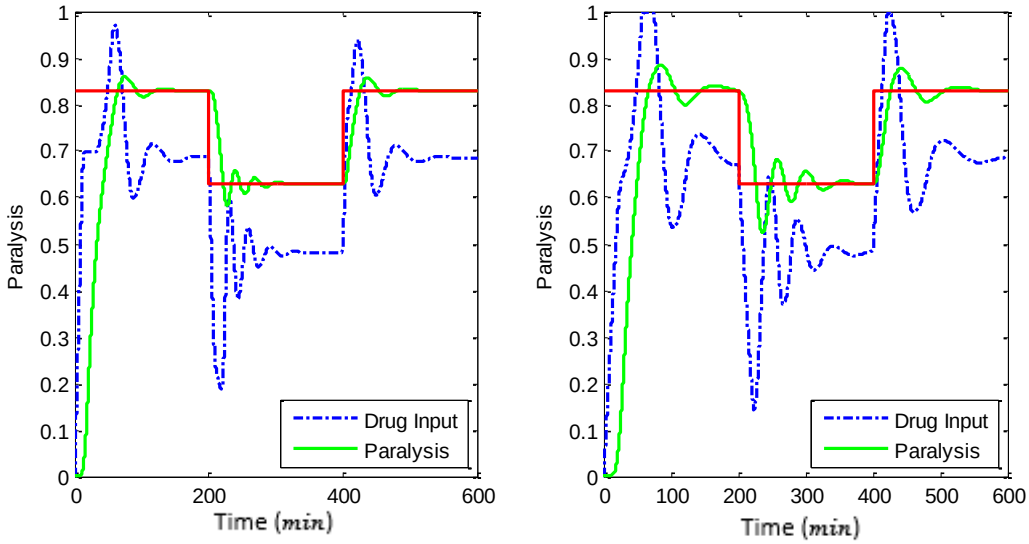
Group B: used to test the sensitivity of the SOFLC-DSL to changes in the scaling factors of change of error, *GC*.

1) *GE= 30.0; GC=250;GT=0.4*
2) *GE= 30.0; GC=400;GT=0.4*

Group C: used to test the sensitivity of the SOFLC-DSL to changes in scaling factors of error, *GT*.

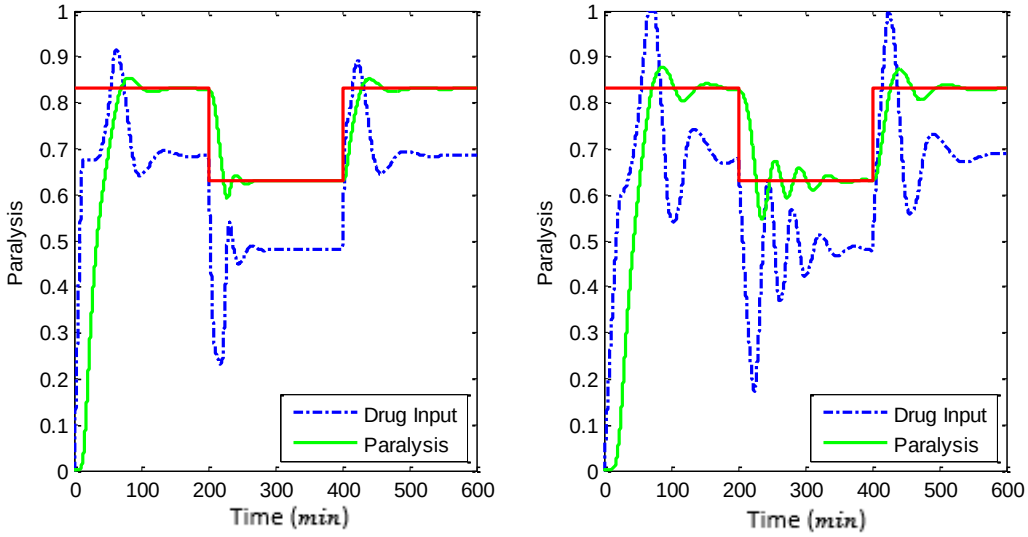
1) *GE= 20.0; GC=300;GT=0.25*
2) *GE= 20.0; GC=300;GT=0.6*

The simulation results of Figures 3.10~3.15 and Table 3.4~3.6 show that the standard SOFLC is very sensitive to the changes in scaling factors and this leads to large fluctuations in its performance. The SOFLC-DSL, however, provides better results, despites the changes in the scaling factors in terms of tracking speed and residual offset.



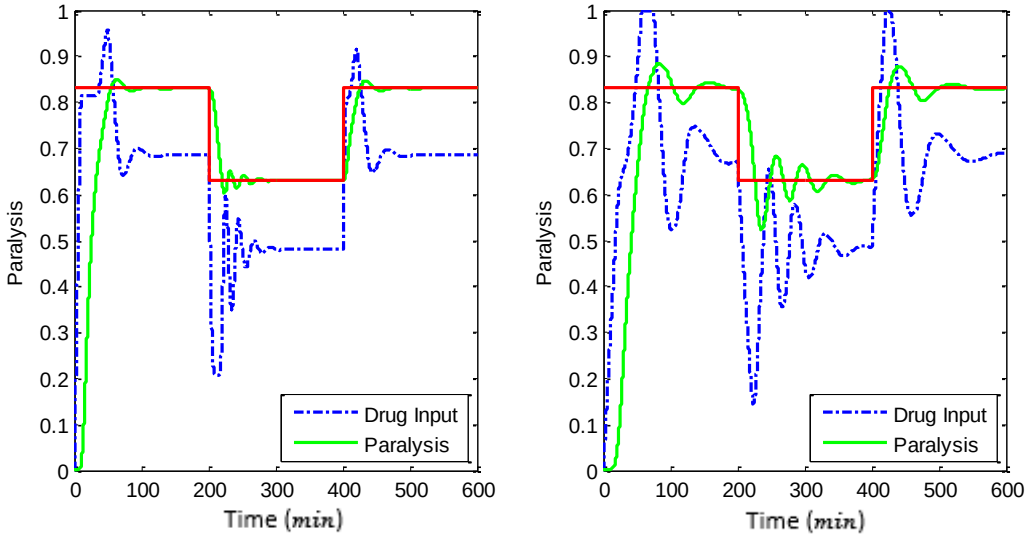
System with SOFLC-DSL System with the standard SOFLC

Figure 3.10: Simulation result of the proposed scheme and the standard scheme with various error scaling factors.



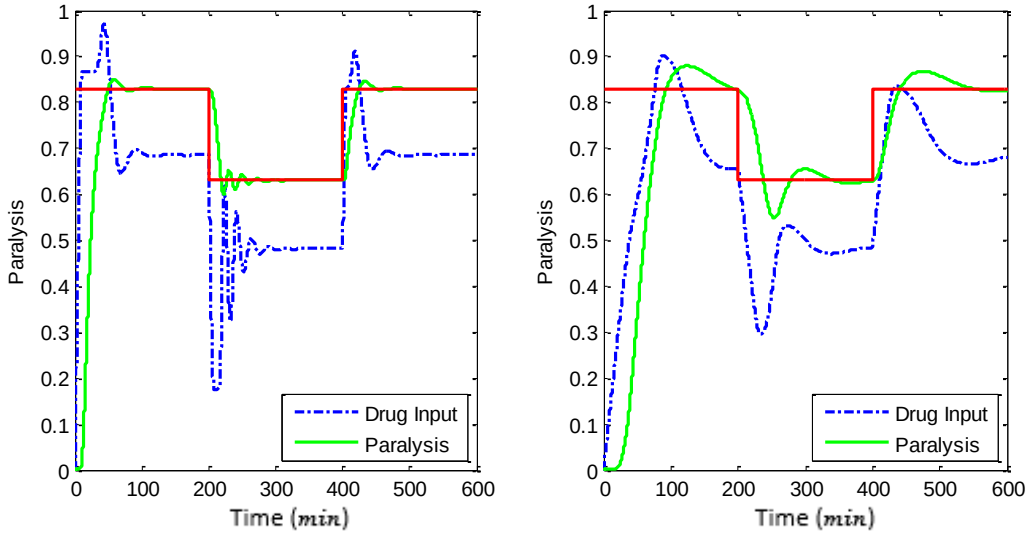
System with SOFLC-DSL System with the standard SOFLC

Figure 3.11: Simulation result of the proposed scheme and the standard scheme with various error scaling factors.



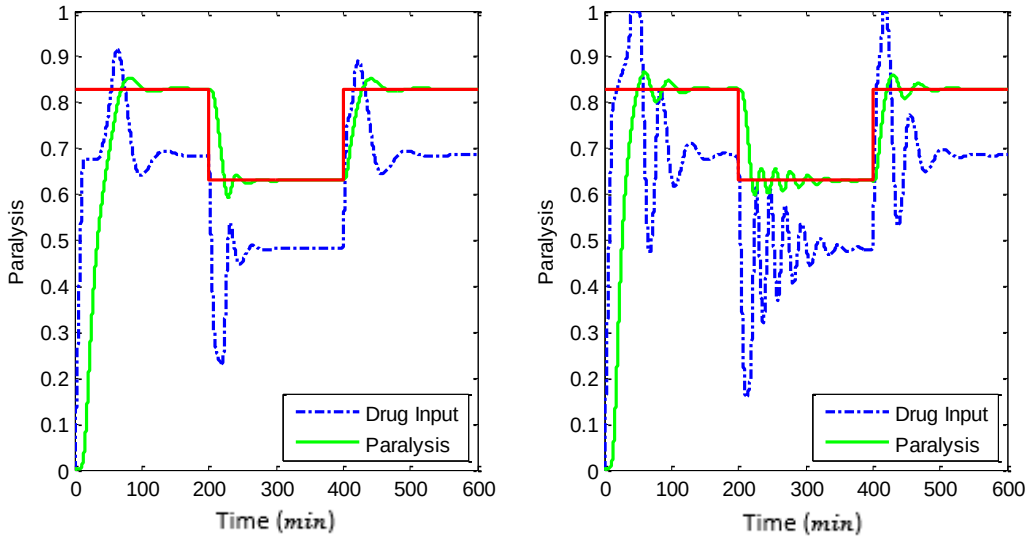
System with SOFLC-DSL System with the standard SOFLC

Figure 3.12: Simulation result of the proposed scheme and the standard scheme with various change of error scaling factors.



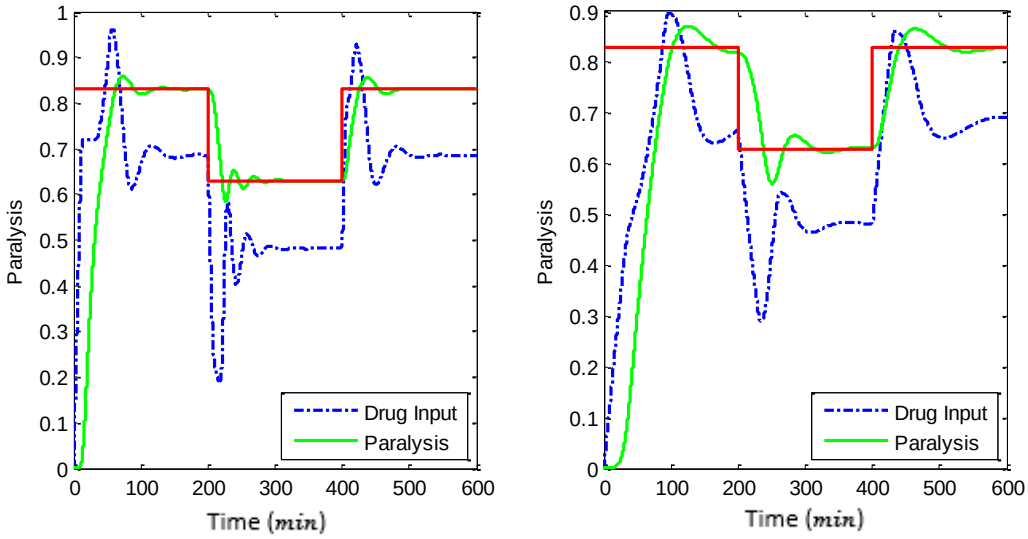
System with SOFLC-DSL System with the standard SOFLC

Figure 3.13: Simulation result of the proposed scheme and the standard scheme with various change of error scaling factors.



System with SOFLC-DSL System with the standard SOFLC

Figure 3.14: Simulation result of the proposed scheme and the standard scheme with output scaling factors.



System with SOFLC-DSL System with the standard SOFLC

Figure 3.15: Simulation result of the proposed scheme and the standard scheme with output scaling factors.

	Criteria	SOFLC-DSL	Standard SOFLC
Figure 3.10	IAE	396.89	633.51
	ISE	201.15	220.92
	Rule number	18	26
Figure 3.11	IAE	386.54	423.51
	ISE	199.15	222.82
	Rule number	18	25

Table 3.4: The performance of the SOFLC-DSL with different error scaling factors.

	Criteria	SOFLC-DSL	Standard SOFLC
Figure 3.12	IAE	395.64	670. 36
	ISE	279.15	237. 94
	Rule number	18	26
Figure 3.13	IAE	389.01	810.51
	ISE	288.73	259.64
	Rule number	19	27

Table 3.5: The performance of the SOFLC-DSL with different change of error scaling factors.

	Criteria	SOFLC-DSL	Standard SOFLC
Figure 3.14	IAE	385.92	653.51
	ISE	278.16	259.82
	Rule number	16	25
Figure 3.15	IAE	395.69	423.51
	ISE	274.16	199.62
	Rule number	17	26

Table 3.6: The performance of the SOFLC-DSL with different output scaling factors.

3.4.3 System robustness to model parameter variations

In biology in general, and biomedics in particular, the patient's parameters normally vary on a ratio of 4:1. Hence, to examine the effectiveness of the SOFLC-DSL to control a wider variety of muscle relaxation processes under changes in the model time constants and gain, the proposed algorithm was applied to various processes which consider such a variation ratio. The model parameters were selected based on the ration stated above. The simulation results of these experiments are shown in Figures 3.16-3.18 and Table 3.7. One can notice how the SOFLC-DSL leads to superior performances versus the standard SOFLC in terms of accurate tracking and efficient fuzzy rule-base elicitation even when new sets of model parameters are used.

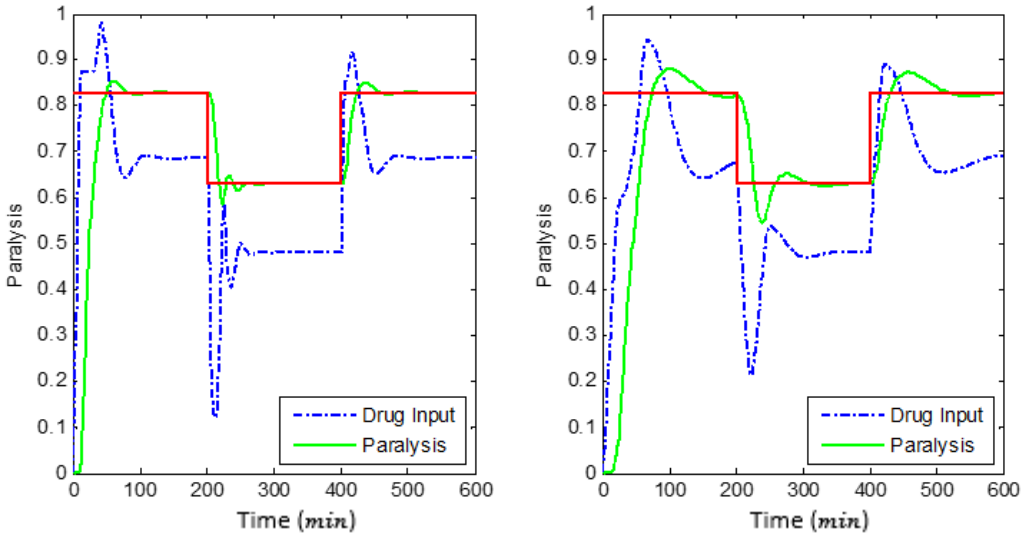
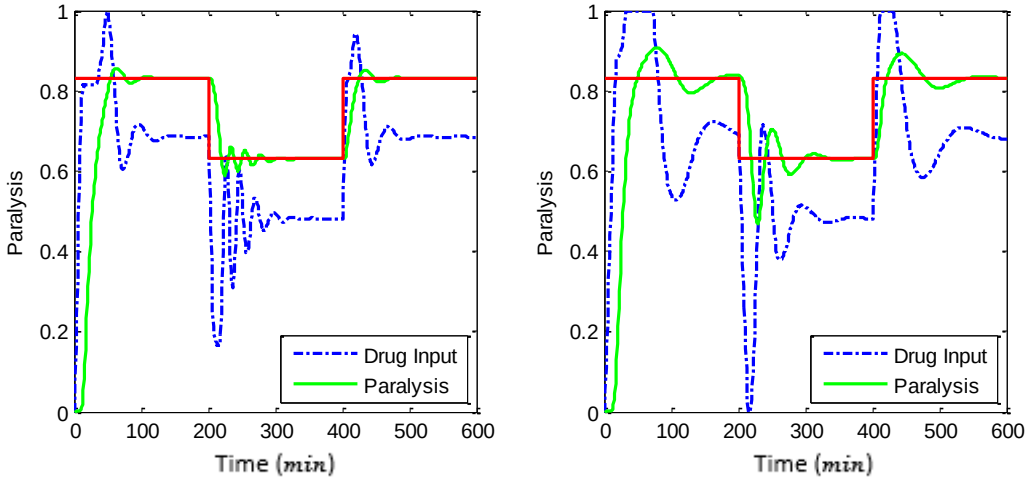
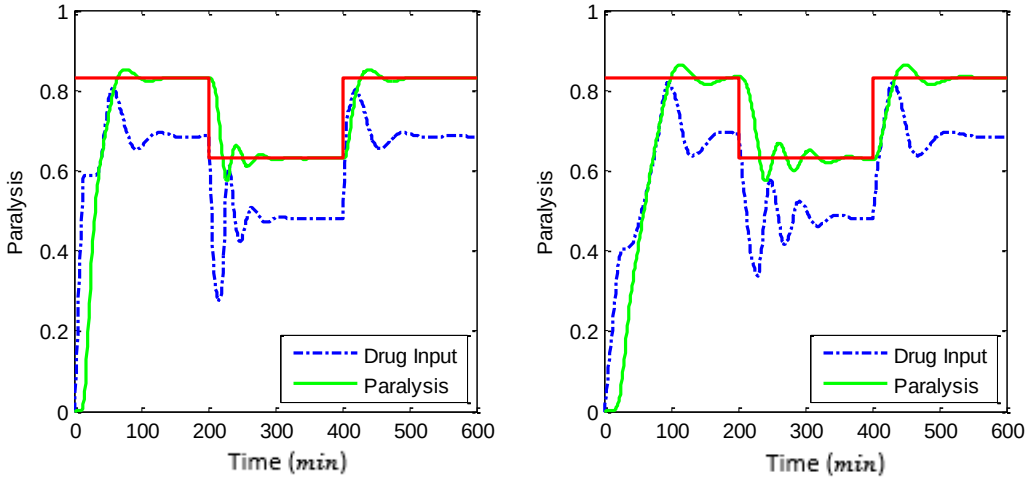


Figure 3.16: Simulation result of the proposed scheme and the standard scheme using a new set of system parameters: K_1=1, T_1=4.8 min, T_2=34.36 min, T_3=3.08 min, T_4=10.65 min.



System with SOFLC-DSL System with the standard SOFLC

Figure 3.17: Simulation result of the proposed scheme and the standard scheme using a new set of system parameters: K_1=1, T_1=5 min, T_2=20 min, T_3=2 min, T_4=7 min.



System with SOFLC-DSL System with the standard SOFLC

Figure 3.18: Simulation result of the proposed scheme and the standard scheme using a new set of system parameters: K_1= 1, T_1=6 min, T_2=22 min, T_3=2.5 min, T_4=6 min.

	Criteria	SOFLC-DSL	Standard SOFLC
Figure 3.16	IAE	381.87	764.56
	ISE	211.25	278.94
	Rule number	17	25
Figure 3.17	IAE	402.32	735.31
	ISE	181.16	267.67
	Rule number	15	23
Figure 3.18	IAE	385.82	785.88
	ISE	187.25	299.62
	Rule number	15	21

Table 3.7: Summary of performance criteria with different system dynamics.

3.4.4 Robustness in the stochastic case

Since the modification of the SOFLC-DSL structure is established based on the feedback signals received from the process at each sampling instant, when these feedback signals are corrupted with noise, the SOFLC-DSL fails to make the output signal track the desired set-point because the modification process of the fuzzy rules will be carried-out based on corrupted signals and will infer the wrong classification. To overcome this issue and improve the performance of the system under noisy conditions, the SOFLC-DSL algorithm included a third-order polynomial filter with a window of 20 samples.

Over a fixed window, this filter uses the principles of least-squares fitting of a polynomial function to the noisy signal. The estimated function takes the following form.

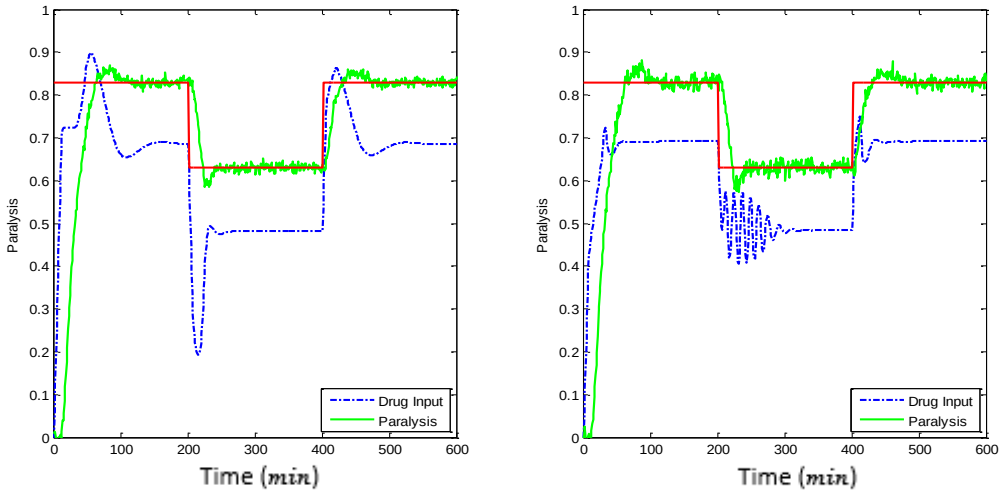
$$p(\mathrm{x}) = \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{c_k}\phi_{\mathrm{k}}(\mathrm{x}) \tag{3.6}$$

where $\phi_k$is the k^{th} order polynomial function and the factors $c_1,c_2,c_3,..,c_m$ can be determined by solving the equation below.

$$\frac{\delta E}{\delta c_j} = \sum_{k=1}^{n} 2[\sum_{i=1}^{m} c_i \phi_i\left(x(k)\right) - y(k)]\phi_j(x(k)) = 0 \tag{3.7}$$

where $E = \sum_{k=1}^{n}[\sum_{i=1}^{m} c_i \Phi_i \left(x(k)\right) - y(k)]^2$

The current smoothed date points are calculated using the previously calculated factors using a window of *n* samples of noisy data *(z(k),k=t-n+1,t-n+2,..,t).* Both the smoothed signal *y(t)* and its derivative *y(t)-y(t-1)* are employed by SOFLC-DSL.



System with SOFLC-DSL System with the standard SOFLC

Figure 3.19: Simulation result of the proposed scheme and the standard scheme when a noise of 5% is added to the system.

The simulation results of Figure 3.19 reveal that the SOFLC-DSL outperforms the standard SOFLC in the noisy environment; this can also be proved by the IAE criterion in Table 3.8 and how the proposed scheme manages to control the process with a lower number of fuzzy rules. From these results one can conclude that the polynomial function has undoubtedly improved the control performance of the SOFLC in the noisy environment while the primary characteristics of the scheme, such as lesser degree of dependency on the operator/expert knowledge and low computations cost are maintained. It can also be noticed that the control signal provided by the standard SOFLC scheme fluctuated significantly in the second stage of the experiment which can pose risk to the patient during surgical procedures.

Criteria	SOFLC-DSL	Standard SOFLC
IAE	412.52	614.56
ISE	254.16	344.64
Rule number	19	26

Table 3.8: Summary of performance criteria with noisy signals.

3.5 Summary

In this chapter, a new SOFLC algorithm with a dynamic layer has been proposed; using fitness estimation and credit assignment, an on-line particle swarm optimisation (PSO) algorithm has been used to make all the cells of the performance index table of the SOFLC adaptable. With such mechanisms, the SOFLC scheme can update both the performance index table and the lower-level fuzzy logic controller at each sampling instant given certain performance criteria. The SOFLC-DSL was validated on non-linear and uncertain muscle relaxation process under different conditions.

The simulation results show that good performances are achieved even when the algorithm begins with an empty performance index table. These results demonstrate that the SOFLC-DSL outperforms the standard SOFLC in terms of accuracy and interpretability. The proposed architecture included the following advantages:

1. The dynamic self-organising mechanism enabled the SOFLC to provide the control action required by the process under control with a simple structure. Unlike the standard SOFLC, the proposed SOFLC-DSL algorithm starts with an empty performance index table and meets the control design requirements despite producing and relying on a significantly small number of fuzzy rules in the lower-level fuzzy logic controller. This leads to reduced computations and low memory storage requirement, a characteristics that is vital for successful implementation of the algorithm in real-time.
2. The SOFLC-DSL performance is superior as compared to the standard SOFLC in terms of fast convergence, system dynamics variations and robustness against disturbances. The proposed scheme also performs better in the transient and stead-state phases.
3. The third-order polynomial filter has improved the capabilities of the proposed algorithm in the stochastic case and assures flexibility of the new control architecture.
4. The SOFLC-DSL is less sensitive to changes in the error, change of error, and output scaling factors, important characteristics that do not make the controller require regular re-tuning when variations in these parameters occur.

The SOFLC-DSL proved to have effective control characteristics while working in various changing environments and conditions.

The following chapter will include improving generalization and enhancing the capabilities of the SOFLC-DSL to handle uncertainties and noises by replacing type-1 sets with type-2 sets.

Chapter 4 - Type-2 Fuzzy Sets for the Self-Organising Fuzzy Logic Control with a Dynamic Supervisory Layer

4.1 Introduction

Based on observations from the previous chapter, the capabilities of the SOFLC-DSL are further enhanced in this chapter by employing type-2 fuzzy sets. This work is carried-out to improve the performance of the controller from different aspects, including:

1. The application of the proposed controller within noise-contaminated environments. In this work, the SOFLC-DSL algorithm is expected to cope well with noise despite relying on limited *a priori* system information.
2. Generalisation of the SOFLC-DSL algorithm to other processes without the need for re-tuning or re-structuring.

3. Further enhancement of the structure of the controller, to enable it, for example, to have a faster response in the transient stage and a smaller tracking error in the steady-state region.

In this chapter, a brief overview of type-2 fuzzy controllers, as well as the concept of zSlices for effective defuzzification, is presented first. This overview is meant to provide a basic introduction to the different applications in which interval type-2 controllers are used, as well as to the general concept needed to understand zSlice general type-2 fuzzy sets and systems. Both an interval and a zSlice general type-2 SOFLC-DSL algorithms, as the basis for the control systems for a muscle relaxant model, are initially constructed, and their performances are then compared with type-1 SOFLC and SOFLC-DSL algorithms developed in Chapter 3. The different SOFLC schemes are tested in different environments to explore their robustness in the presence of parameter changes and noise. Their performances are measured using various approaches, such as structural simplicity, steady-state errors and control stabilities, as well as IAE and ISE criteria.

The experimental simulations carried-out in this study assessed the performance of the SOFLC-DSL algorithms in their ability to regulate the anaesthetic delivery rates for maintaining the desired physiological levels of muscle relaxation during a three-stage surgical procedure.

4.2 Intelligent Control of the Depth of Anaesthesia

Anaesthesia is a sector of biomedical science that is concerned with the management of anaesthetic agents, whose goal is to monitor and control the state of unconsciousness of patients during surgical conditions. Generally speaking, anaesthesia involves muscle relaxation, unconsciousness and analgesia. An anaesthetist is normally responsible for regulating the first two, while the third is related to postoperative conditions (Stoelting *et al.*, 2000). In recent decades, there have been an increasing number of studies on how intelligent systems can be used to monitor and control anaesthetic delivery (El-bardini and El-Nagar, 2011; Mahfouf *et al.*, 2003; Stanski, 1994).

The complexity of the human body in terms of being a multivariable and highly non-linear system with high levels of uncertainty renders the regulation of such system a very challenging task. Some of these issues include:

- The physiology of the human body, which differs from one patient to another (inter- and intra-patient variability) based on age, gender and preoperative health conditions, meaning that the anaesthetic drugs need to be adjusted accordingly in terms of concentration and drug regimens.
- Anaesthetic drugs can have different physiological effects on the body of the patient, hence different bodies require different anaesthetic drug concentrations.
- During surgery, anaesthetists usually monitor complex and sensitive multivariable changes and non-linear interactions via variables such as heart rate, blood pressure (BP), etc.
- The signals that are acquired and provided from the patient's body to regulate the anaesthesiology procedures can be very noisy.

The above challenges represent high levels of complexity, non-linearity and uncertainty, making the design of effective automatic controllers difficult. However, fuzzy logic controllers have been successfully applied in automated drug infusion control, and the obtained results have shown that they are able to deliver satisfactory performance while effectively handling the imprecision and uncertainty faced in the real world (Abbod and Linkens, 1992; Doctor *et al.*, 2005; Esmaeili *et al.*, 2008).

The standard SOFLC algorithm has also been applied to anaesthesia in more than one study. In all of these contributions, the SOFLC algorithm proved able to outperform traditional fuzzy controllers (Mahfouf and Linkens, 1998). In recent years, there has been a growing number of studies on the extraction of fuzzy rules for anaesthesia control using different learning methods, such as genetic algorithms and particle swarm optimisation (Leng, 2005; Lu and Mahfouf, 2006; Mahfouf *et al.*, 2000). However, in most of these contributions the SOFLC schemes use type-1 fuzzy sets only.

4.3 Particle Swarm Optimization of Interval Type-2 Fuzzy Systems

There have been various contributions reported in the literature in which different versions of PSO algorithms are used to optimise interval type-2 fuzzy controllers. In

most of these contributions, the PSO based type-2 fuzzy controllers have led to relative success in different areas of application.

In the work of Cao *et al*. (2008), for example, an adaptive interval-2 fuzzy logic controller was driven by a PSO algorithm to regulate a vehicle's non-linear active suspension system. The controller used interval type-2 membership functions to handle the uncertainty and non-linearity caused by irregular road inputs and other disturbances, as well as to deal with the uncertainty originating from the potential lack of knowledge or experience of the designer/operator. The proposed scheme was provided with an adaptive mechanism in which the active force was self-tuned between the upper and lower membership function of the interval type-2 fuzzy output. When the controller was tested on a quarter active suspension model, it outperformed other conventional controllers and provided effective control performance. In another work, by Martinez *et al.* (2010), a PSO-based mechanism was used to fabricate an optimal controller that minimised the steady-state error of linear processes. The PSO algorithm was used to tune the parameters of the type-2 interval membership functions of the controller to achieve the desired characteristics. When tested on different benchmark plants, and when compared with other controllers obtained by genetic algorithms, the simulation results demonstrated the visibility of the developed algorithm.

In a separate research, Oh *et al.* (2011) proposed an interval type-2 fuzzy controller aided by a PSO algorithm to control a ball and beam system. Ball and beam systems are difficult to regulate and control due to their challenging characteristics, and so the proposed controller drove a servo motor in order to determine the position of the ball. A type-1 membership function with fixed membership grades makes it difficult for the rule-base to fully capture the dynamics of the system, and therefore type-1 controllers tend to be less effective in terms of controlling the system. However, the footprint of uncertainty of type-2 fuzzy sets can effectively improve the control characteristics. The simulation results showed that type-2 fuzzy controllers provided good control abilities and robustness.

A different type-2 fuzzy controller, tuned by a PSO algorithm to control a 2 DOF planar robot, was proposed by Bingul and Karahan (2011). In this Mamdani-type controller, the PSO algorithm used three different cost functions to optimise the centres and widths of interval type-2 Gaussian membership functions that were used for the input and

output. The performance of the developed controller was compared to that of a PID controller, which was also tuned by a PSO algorithm. Different experiments were conducted with different model parameters and in different noise-contaminated environments, and the simulation results demonstrated that in the robot trajectory control, the proposed controller outperformed the PID controller and was more robust.

In a different paper, Martinez *et al.* (2010) explained how a PSO system was used to tune an interval type-2 fuzzy controller to minimise the state error when the controller was applied to linear processes. In doing that, the parameters of the type-2 membership functions of the controller were tuned in order to achieve optimal regulatory performance and stability. When implemented in a Simulink environment, the controller showed good robustness in different environments.

4.4 Comparisons of Type-1 and Type-2 Fuzzy Logic Systems

Type-1 and interval type-2 fuzzy controllers have recently been better understood and explained; their utility has also been compared across a wide range of contributions in the literature, such as in the research conducted by Cazarez *et al.* (2007). The reason for this trend is clear, as discussed in previous chapters, which revealed that when there is a considerable amount of uncertainty associated with the process to be controlled, interval type-2 controllers are more likely to result in better control abilities than type-1 controllers (Aliasghary *et al*., 2012).

A comparative analysis of type-1 and interval type-2 fuzzy controllers in terms of their ability to learn the behaviours of mobile robotics was conducted by Linda and Manic (2010). A type-1 fuzzy controller was first designed and then fine-tuned with a learning algorithm to follow the wall-following movements performed by the designer. This scheme was then extended to a type-2 fuzzy controller by symmetrically blurring all the fuzzy sets used in the type-1 scheme. The controllers were tested in both noise-free and noise-contaminated environments. The experiments showed that the interval type-2 controller had better controller abilities than the interval type-1 controller, in terms of working in uncertain and noise-contaminated environments.

A different comparative study between type-1 and interval type-2 fuzzy controllers, using a more complex robotic football system, was conducted by Figueroa *et al.* (2005).

In this system, moveable robots were placed in a field where they could kick a ball. Different components were used in this complex system, including a high level control system, wireless communication and other software algorithms, to facilitate the movement of these robots. These components introduced a high level of uncertainty and noise, and represented a good platform to test the effectiveness of type-2 fuzzy systems. Simulation results demonstrated that, as expected, the interval type-2 controller provided better control abilities than that of type-1.

Other comparative studies are reported in the literature in which type-2 fuzzy controllers performed better than type-1 fuzzy controllers despite both working under the same circumstances and experimental set-ups (Sepulveda *et al.*, 2007). As outlined by Wu (2010), a potential reason for this superiority is the continuity inherent in the characteristics of interval type-2 fuzzy systems, which contrasts with type-1 systems, which have discontinuous properties, as shown in the control surfaces obtained from Wu's work. Wu concluded that the control surfaces of interval type-2 fuzzy controllers are continuous across all regions, whereas a clear discontinuity was apparent at certain points in type-1 control surfaces. In other words, this means that there are certain situations in which type-1 fuzzy controllers fail to map input to output due to their discontinuous control surface, and as a result become unable to calculate the required output signal needed for the process under control. In contrast, the continuous control surface of interval type-2 controllers allows them to relate all the inputs to the outputs, and gives them the ability to calculate the outputs required by the system in all regions and at all times.

A different argument for the superiority of interval type-2 controllers is reported in the work by Cara *et al.* (2011), where a servo system was used as a basis for the comparative analysis. These researchers argue that the considerably greater number of parameters that the interval type-2 controllers use gives them more flexibility, which leads to better control abilities than the singleton type-1 and non-singleton type-1 fuzzy controllers, which were both used in the work of Cara, especially as the uncertainty levels increase. Other observations of the study revealed that when non-singleton fuzzification was employed, the flexibility of type-1 controllers was further extended, reaching a level closer to that of interval type-2 fuzzy controllers. These results were reached by applying the three controllers – singleton type-1, non-singleton type-1 and

interval type-2 – to a non-linear and mathematically ill-defined servo system. For each controller, three experimental set-ups with different levels of uncertainty were used. The simulation results showed that when the uncertainty level was low, the singleton type-1 system produced the best results. However, as the level of uncertainty increased, the performance of both the non-singleton type-1 and interval type-2 fuzzy controllers improved, even though the latter still provided the best performance. The authors argued that this was due to the FOU enabling the variation and uncertainty to flow through the entire inference system. In contrast, the non-singleton type-1 controller used membership grades to handle these variations. One shortcoming of the contribution of this experiment was that it was only applied to a simple SISO servo system, making it hard to generalise from these findings to other types of systems.

Mendez *et al.* (2007) used a large industrial system as a platform to compare type-1 and type-2 fuzzy controllers. Even though the interval type-2 fuzzy controller was demonstrated to be superior in different circumstances, the author questioned whether it was worth upgrading a system if a type-1 fuzzy controller provided acceptable performance, especially given the lower level fuzzy logic controller's lesser computational cost. Various attempts have been made to make the interval type-2 fuzzy controller less computationally expensive. Wu and Mendel (2002), for example, proposed a technique through which the type reduction stage could be eliminated; despite the need for such a method, however, theirs has remained unpopular in the literature, as other methods, such as the Karnik-Mendel (KM) iterative algorithm for type-reduction (outlined in Chapter 2), have been found to be less complex, and simpler to implement. In addition, the results that type-1 controllers produce are normally within an acceptable range. A different approach for reducing the computational complexity of interval type-2 fuzzy systems was introduced by Coupland and John (2005), which is based on making the join-and-meet operations faster.

Despite the simplicity of interval type-2 fuzzy systems and their success in a wide range of applications, many researchers argue that general type-2 fuzzy sets, such as the zSlice type-2 fuzzy sets, provide better mechanisms to handle imprecision and uncertainty, as their third dimensional secondary membership function is modelled as a continuous fuzzy set whose support is in the range [0-1].

4.5 Introduction to zSlice General Type-2 Fuzzy Sets

Mamdani interval type-2 fuzzy logic control, as shown throughout the previous chapters, is based on interval type-2 fuzzy sets. As outlined in Chapter 2, interval type-2 fuzzy sets are a simplification of what is now known as general type-2 fuzzy logic systems. A considerable amount of research has been conducted in order to find alternatives to interval type-2 fuzzy sets and propose new versions of general type-2 fuzzy sets with improved capabilities, while keeping the computation costs within an acceptable range, allowing for their real-time application (e.g. Coupland and John, 2007; Liu, 2008; Wagner and Hagras, 2008). More recently, the concept of zSlices was introduced (Wagner and Hagras, 2010), which enabled the construction of general type-2 fuzzy systems in the form of modified interval type-2 fuzzy sets, or 'slices'. This discovery enabled the real-time implementation of general type-2 fuzzy control with affordable computational complexity.

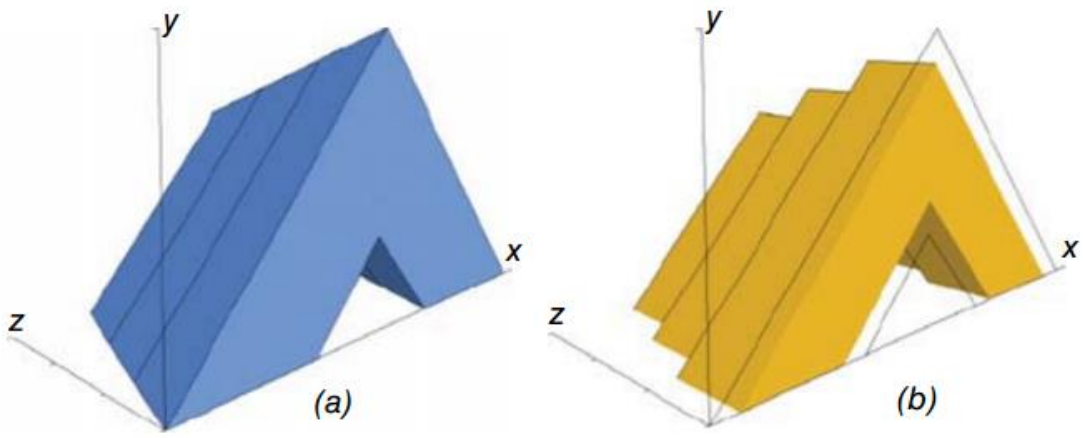


Figure 4.1: (a) Illustration of the side view of a general type-2 fuzzy set, (b) side view of zSlice based general type-2 fuzzy set (Wagner and Hagras, 2010).

From a practical point of view, zSlice-based general type-2 fuzzy sets are provided with the advantage of working flexibly in three dimensions, unlike the interval type-2 fuzzy sets where the third dimension is always unified. For illustration purposes, Figure 4.1 shows the side view of the transition from the standard general type-2 fuzzy set to a zSlice-based general type-2 fuzzy set with three zSlices. The zSlice-based general type-

2 fuzzy sets are normally generated, in practice, as a collection of zSlices, which together construct the general type-2 fuzzy set.

4.5.1 From interval type-2 fuzzy systems to zSlices

zSlices represent the core of zSlice-based general type-2 fuzzy sets. As explained by Wagner and Hagras (2010), a zSlice can be constructed by slicing a general type-2 fuzzy set z at level z_i . An interval set with height z_i in the third dimension is produced by this slicing action. The formed zSlice $\tilde{Z}_i$is equivalent to the interval type-2 fuzzy set with the expectation that, in the third dimension, its membership grades are not fixed to 1 but equal to z_i, where $0 \leq z_i \leq 1$.

Hence, a zSlice $\tilde{Z}_i$ is expressed as follows:

$$\tilde{Z}_i = \int_{x \in X} \int_{y \in \overline{y_i^x}} z_i/(x, y) \tag{4.1}$$

As shown in Figure 4.2(a), zSlicing creates an interval set with height z_i and support $\overline{y_i^x}$ within the range $[l_i, r_i]$ at each x value, as illustrated in Figure 4.2(b). Further, $0 \leq i \leq 1$, where i defines the number of zSlices (excluding $\tilde{Z}_0$), and normally $z_{i=i/I}$. Since $\tilde{Z}_0$ is special case with height $z=0$, it can be discounted (Mendel, Liu and Zhai, 2009). A general type-2 fuzzy set $\tilde{Z}_i$ can be defined as a collection of an infinite number of zSlices, as follows:

$$\tilde{Z}_i = \int_{x \in X} \int_{y \in [l_i, r_i]} z_i/(x, y) \tag{4.2}$$

Note that,

$$\tilde{Z}_0 = \int_{x \in X} \int_{y \in \overline{y_0^x}} 0/(x, y) \tag{4.3}$$

$\tilde{Z}_0$ is not considered to be part of the number of zSlices constructing the zSlice-based general type-2 fuzzy set, as its secondary membership grades are all zero and therefore do not contribute to any computations (Mendel, Liu and Zhai, 2009).

A general type-2 fuzzy set $\tilde{F}$ can be defined as a collection of an infinite number of zSlices, as follows:

$$\tilde{F} = \int_{0 \le i \le 1} \tilde{Z}_i, \; I \to \infty \tag{4.4}$$

In the discrete case, (4.4) can be rewritten as follows:

$$\tilde{F} = \sum_{i=0}^{I} \tilde{Z}_i \tag{4.5}$$

where the summation sign in the equation 4.5 does not define the arithmetic addition, but rather the operation of the union set-theoretic.

Having outlined the main concepts of the zSlices and the zSlice-based general type-2 fuzzy sets, one can proceed to define the construction of zSlice-based general type-2 fuzzy logic control.

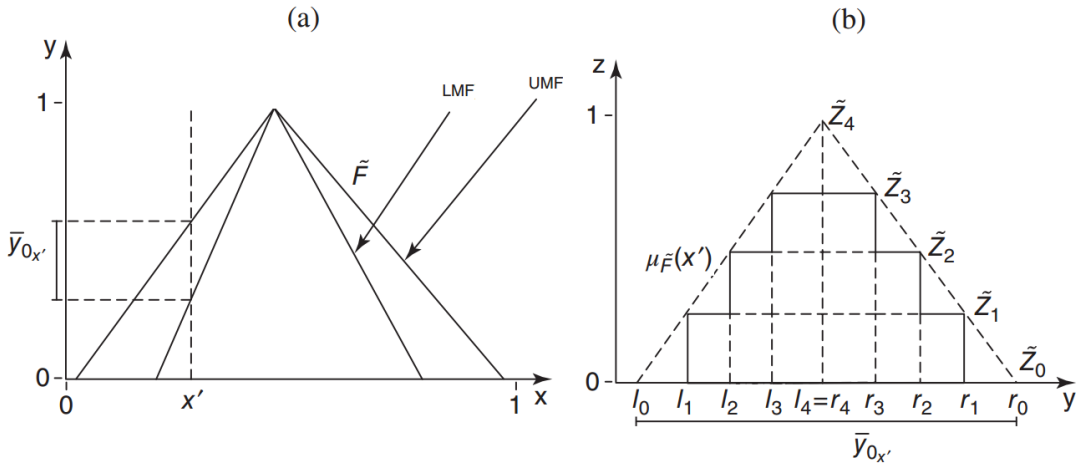


Figure 4.2: (a) Illustration of the front view of a general type-2 fuzzy set, (b) The third dimension at x' of zSlice-based general type-2 fuzzy set (Wagner and Hagras, 2010).

4.5.2 Mamdani zSlice-based general type-2 fuzzy logic control

The structure of the zSlice-based general type-2 fuzzy logic system is similar to its interval type-2 counterpart, i.e. it consists of a fuzzifer, an inference engine, a fuzzy rule-base, a type-reducer and a defuzzifier. The only difference is that the fuzzy logic system of the former combines the type-reducer and defuzzifier in one component, and employs and also processes the zSlice-based general type-2 fuzzy sets.

The zSlice-based general type-2 fuzzy logic control can be defined directly as the weighted combination of the interval type-2 fuzzy logic control calculated at each zlevel, where the weighting is represented by the secondary membership at each zlevel. In other words, the implementation of the zSlice-based general type-2 fuzzy logic control can be achieved by constructing a series of interval type-2 fuzzy logic controls, where each of them is associated with a particular zlevel. This process is illustrated in Figures 4.3 and 4.4.

The defuzzification process is straightforward in the zSlice-based general type-2 fuzzy logic system; it is carried-out based on the individual centroids produced by each 'contained interval type-2 fuzzy logic control', as shown in Figure 4.4. For an output set $\widetilde{N}$ of an interval type-2 fuzzy control, the centroid $C_{\widetilde{N}}$ is an interval type-1 fuzzy set expressed by its right and left points, as shown below:

$$C_{\widetilde{N}}(x) = 1/[u_l(x), u_r(x)] \tag{4.6}$$

The centroid of the output set $\widetilde{N}$ for a given zlevel z_i is defined as:

$$C_{\widetilde{N}}(x) = z_i/[u_l(x), u_r(x)] \tag{4.7}$$

The union of the centroids of a 'contained interval type-2 fuzzy logic control' can express the overall centroid C of the zSlice-based general type-2 fuzzy logic control, which can be written as follows:

$$C(x) = \bigcup_{i=1}^{I} C_{\widetilde{N}}(x) \tag{4.8}$$

An example centroid of the zSlice-based general type-2 fuzzy logic control with three zlevels is shown in Figure 4.5.

The produced deffuzifed value of the zSlice-based general type-2 fuzzy logic control can be calculated by either determining the centroid dezzifier for $C(x)$ or by computing the weighted average of each z_i, as shown below (Mendel *et al.*, 2014):

$$y_C(x) = \frac{z_1\left[\frac{u_{l_1}(x)+u_{r_1}(x)}{2}\right]+z_2\left[\frac{u_{l_2}(x)+u_{r_2}(x)}{2}\right]+\cdots+z_I u_I(x)}{z_1+z_2+\cdots+z_I} \tag{4.9}$$

It is worth noting from (4.9) that the values associated with $\tilde{Z}_0$are excluded from the calculation as they have no impact on the final output of the zSlice-based general type-2 fuzzy logic control.

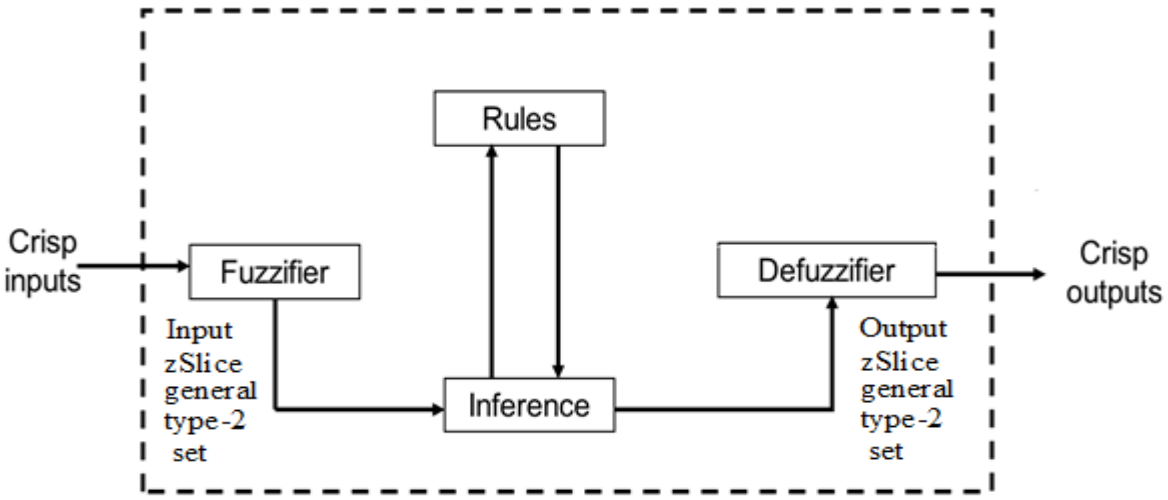


Figure 4.3: zSlice-based general type-2 fuzzy logic system.

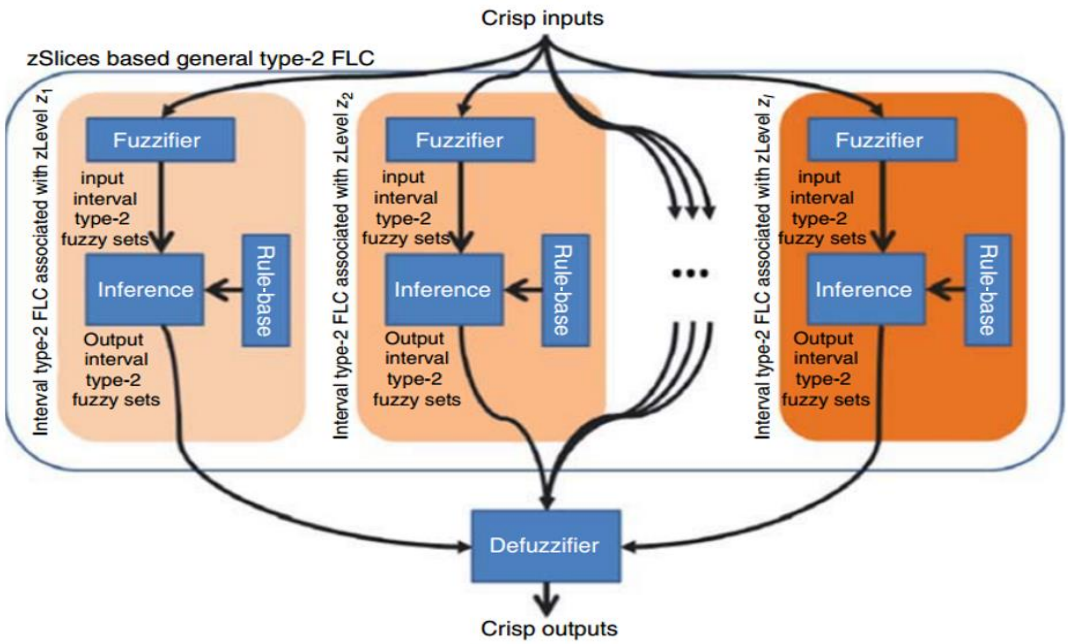


Figure 4.4: The process within the zSlice-based general type-2 fuzzy logic control (Mendel *et al.*, 2014).

In the literature, zSlice-based type-2 fuzzy controllers have been used successfully in a wide range of applications. Wagner and Hagras (2009) developed a navigation control for a two-wheeled, mobile robot using the concept of a zSlice-based general type-2 fuzzy controller. Simulation results showed that the proposed controller had effective control ability, and, when compared to type-1 and interval type-2 controllers, it provided very promising results.

Bosco *et al*. (2012) proposed a zSlice-based type-2 fuzzy controller for the autonomous navigation of a Robotino Omni-directional mobile robot (ROMR). The robot moved in a crowded and unfamiliar environment, where it faced unpredictable obstacles and events. The controller equipped the robot with a motion planning system and local path modification. Experimental results showed how the navigation controller enabled the robot to reach its desired destination, avoiding both static and mobile obstacles in this difficult and previously unknown environment.

In a different paper, a self-tuning, zSlice-based general type-2 fuzzy proportional-integral controller (zT2-FPI) was proposed (Kumbasar and Hagras, 2015), where an on-line mechanism was used to adjust the antecedent general type-2 fuzzy set's secondary membership functions. The controller investigated the impact that the secondary membership functions had on the closed-system control performance and the improvements they could make to the system. Several simulation studies, and a real-time implementation of a non-linear PIONEER 3-DX mobile robot, demonstrated that the proposed mechanism could enhance the transient state and disturbance rejection capabilities of the system.

4.6 Type-2 SOFLC-DSL algorithm

The third dimension of type-2 fuzzy sets, as outlined in Chapter 2, enables them to effectively model high levels of uncertainty and noise. Type-2 fuzzy sets are therefore suitable for simulating uncertain processes, such as anaesthesia control systems. Furthermore, type-2 fuzzy systems can handle more complex non-linear input-output control relationships than type-1 fuzzy systems (Du and Ying, 2010), and they can therefore present a more suitable mechanism for non-linear biomedical control processes.

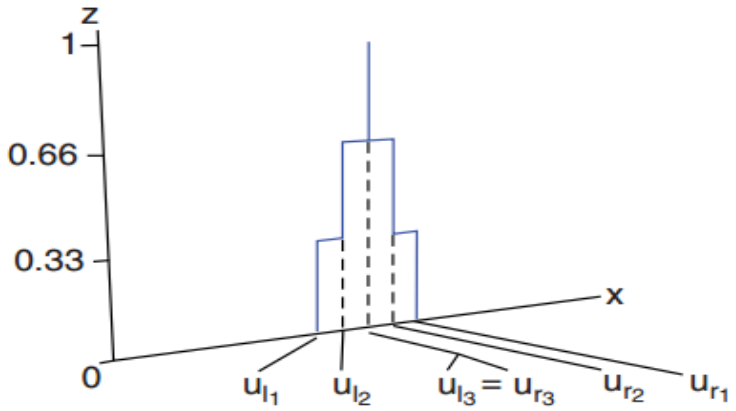
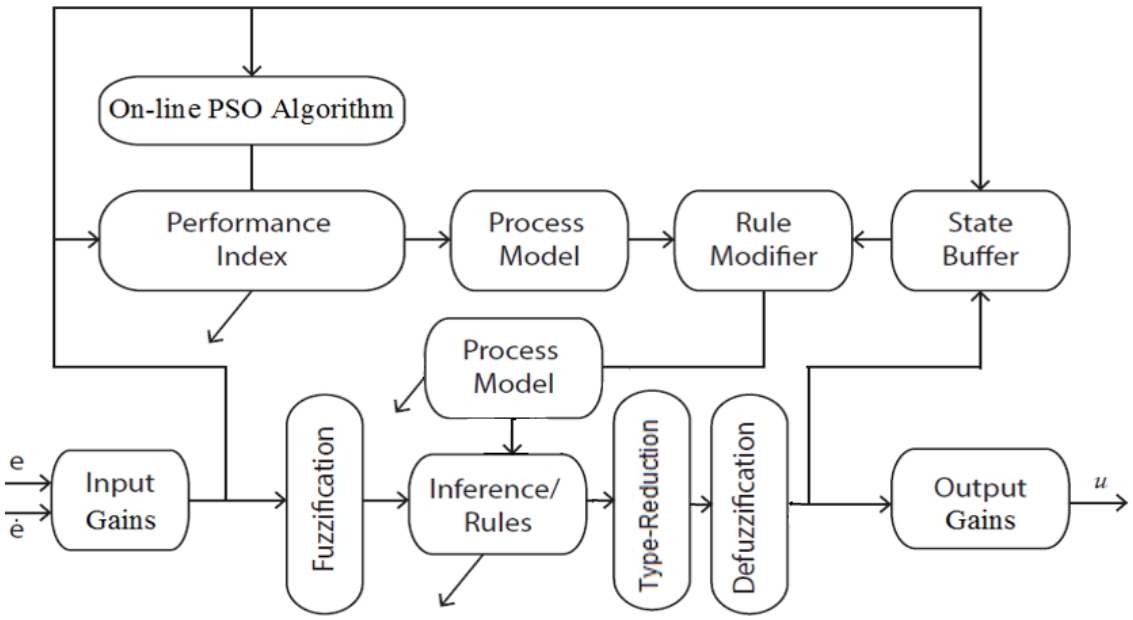


Figure 4.5: An example centroid of the zSlice-based general type-2 fuzzy logic control with three zlevels.

Currently, most type-2 fuzzy systems applications are focused mainly on utilising interval type-2 fuzzy sets, due to their simplicity and reasonable low computational cost. However, both interval and zSlice general type-2 fuzzy sets are proposed in this chapter and then compared with type-1 SOFLC and SOFLC-DSL algorithms in terms of their capacity to manage anaesthesia delivery and keep physiological set points for muscle relaxation. The ability of the proposed controllers to handle both noise-contaminated environments and the body's unpredictability during a three-stage surgical procedure was investigated for this study.

The structure of the proposed Type-2 SOFLC-DSL scheme is illustrated in Figure 4.6; the fuzzy control system in the scheme is the same as the one used in type-1 SOFLC-DSL algorithms. However, the antecedents and consequents in this case are defined in terms of interval and zSlice general type-2 fuzzy sets intended to construct interval and zSlice type-2 SOFLC-DSL algorithms, respectively. Different type reduction methods have been proposed in the literature and used in industry. The enhanced interactive algorithm with stop condition (EIASC) (Wu and Nie, 2011) is used in this study for the interval type-2 SOFLC-DSL algorithm due to its efficiency and simplicity. The other components, shown in Figure 4.6, remain the same as those used in the type-1 SOFLC-DSL scheme outlined in Chapter 3.



. Figure 4.6: The basic structure of interval type-2 SOFLC-DSL.

4.7 Experiments and Results

Simulations in this section were carried-out on the muscle relaxation system described in Chapter 3. The SOFLC algorithm was enhanced by the strategy described in section 4.6.

Figure 4.7 shows the simulation results of the muscle relaxation system and the corresponding control signals obtained when the four fuzzy schemes were applied. It can be seen how the interval type-2 SOFLC-DSL and the zSlice general type-2 SOFLC-DSL schemes performed better in the three stages of the transient period, when the output was still away from the set-point and the control signal, and when the modifications of the fuzzy rules of the performance index tables were rapidly changing. Also, the type-1 SOFLC-DSL scheme, the interval type-2 SOFLC-DSL and the zSlice general type-2 SOFLC-DSL algorithms all tracked the desired set-point in approximately 65 mins, while it took the type-1 SOFLC algorithm until approximately 160 mins to reach the desired point. It can also be seen that the type-1 SOFLC scheme made the output track with set-point with larger overshoots and undershoots than any of the other schemes.

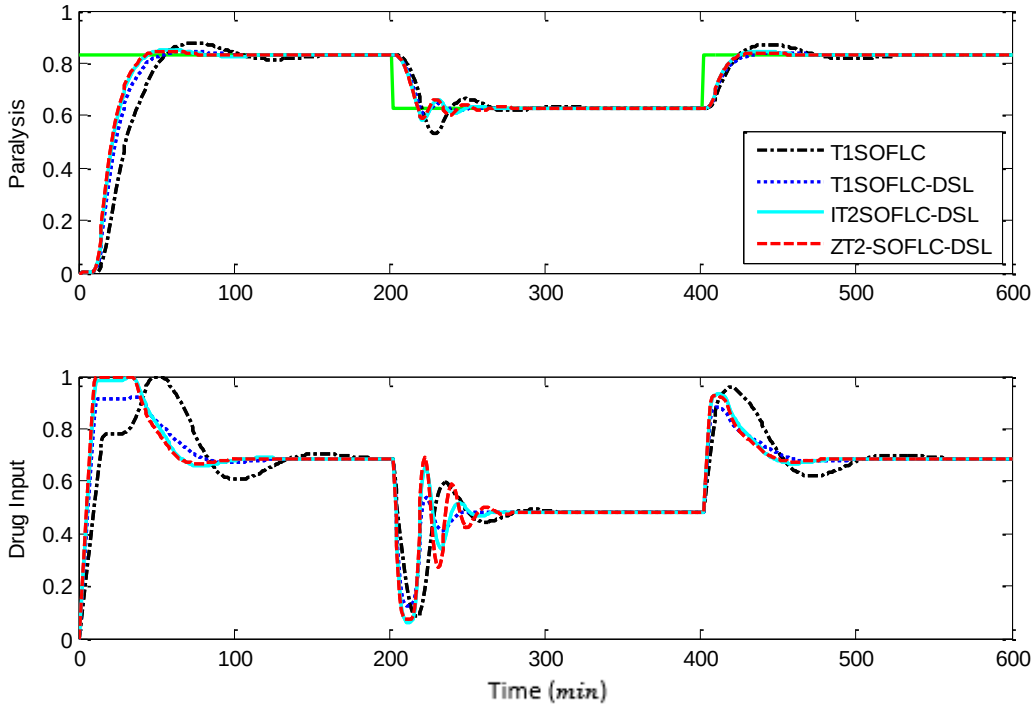


Figure 4.7: Simulation result of muscle relaxation using different controllers.

Criteria	T1SOFLC	T1SOFLC-DSL	T2SOFLC-DSL	ZT2SOFLC-DSL
IAE	352.73	250.34	234.62	228.95
ISE	166.11	129.62	121.41	118.65
Rule number	18	16	14	14

Table 4.1: Summary of performance criteria of different controllers.

It is apparent that the dynamic performance index combined with type-2 fuzzy sets (interval and zSlice) gave the controller better capabilities in the three stages of the closed-loop response in terms of reaching the desired set-point, with faster rise-time and smoother control signals, as is also evidenced in Table 4.1. Indeed, these controllers produced the best results under IAE and ISE criteria and generated smaller number of fuzzy rules in the lower-level fuzzy logic controller. In the current circumstances, both the interval type-2 SOFLC-DSL and the zSlice general type-2 SOFLC-DSL algorithms performed similarly in the three stages of the surgical procedure in terms of speed of rise

time and degree of steady-state error. However, it is apparent that the latter performed better under the IAE and ISE criteria.

4.7.1 System's robustness to fuzzy scaling factors and model parameters

The robustness of the type-2 SOFLC-DSL algorithms compared to that of the type-1 schemes is shown in Figures 4.8~4.10 and Table 4.2. It can be seen that the type-2 fuzzy sets (interval and zSlices) made the controllers less sensitive to a wide range of scaling factor variations, as far as tracking speed, levels of undershoot and residual offsets were concerned. Type-2 fuzzy sets also led to better performances under the IAE and ISE criteria, and resulted in a smaller number of fuzzy rules generated in the lower-level fuzzy logic controller.

It can also be clearly observed that the type-1 SOFLC scheme led to significant variations in performance and fluctuations in drug input, particularly shown in Figure 4.10. As for the type-1 SOFLC-DSL algorithm, even though it performed better than the type-1 SOFLC scheme, it was still outperformed by the type-2 SOFLC-DSL algorithms, whose type-2 fuzzy sets and dynamic performance index gave them better ability to handle the change in scaling factors.

Investigating the robustness performance of the type-2 SOFLC-DSL algorithms, when applied to different models, is of high importance in determining how well the schemes respond to variable system dynamics and ill-defined systems, without the need for continued re-tuning or re-structuring. Figures 4.11~4.13 show the results of the simulations of the four controllers when the parameters of the muscle relaxation model vary.

It can be seen that type-2 SOFLC-DSL algorithms provided better performance in the transient and steady-state phases. These observations are strengthened by the results recorded in Table 4.3, which show that these two controllers performed better under IAE and ISE criteria and succeeded to control the process with a lower number of fuzzy rules. It is, however, not easy to draw comparisons between the interval and zSlice SOFLC-DSL algorithms, as their performances were very similar; and it can be seen that in the two cases the former outperformed the latter, as shown in Table 4.3, while the latter performed better than the former in one case. The type-1 SOFLC-DSL algorithm

gave a satisfactory performance even though the input signal fluctuated in some cases, while the standard type-1 SOFLC algorithm led to an oscillatory control mode and noticeable offset in certain places.

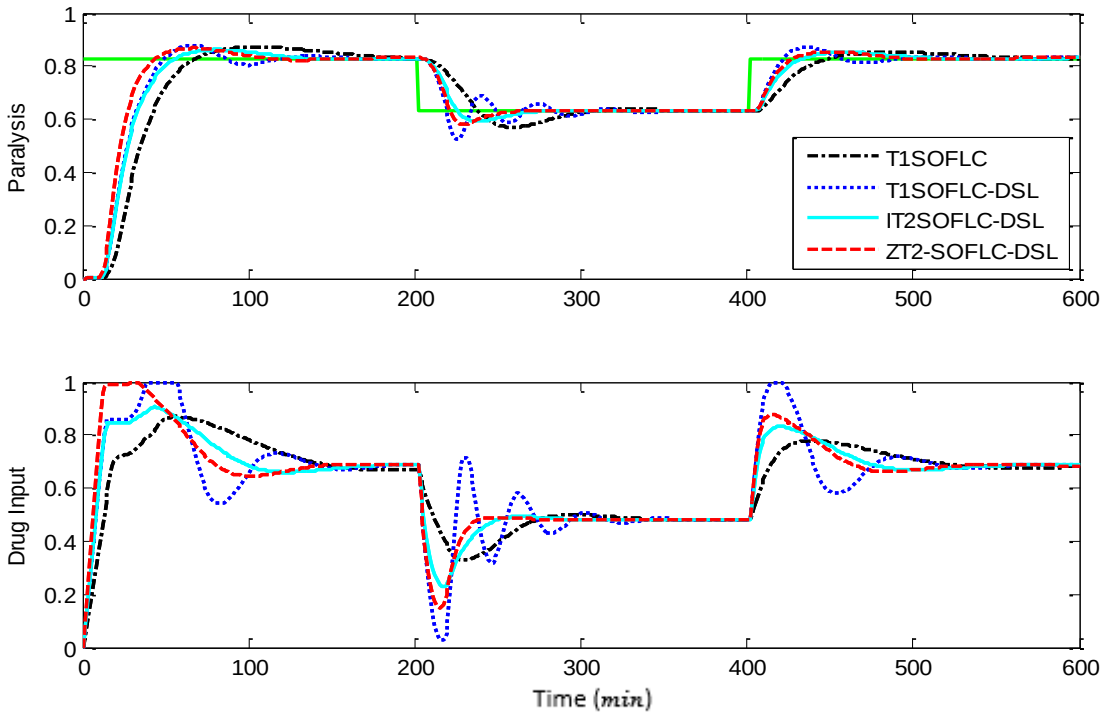


Figure 4.8: Simulation results of muscle relaxation under 50% variations of GE.

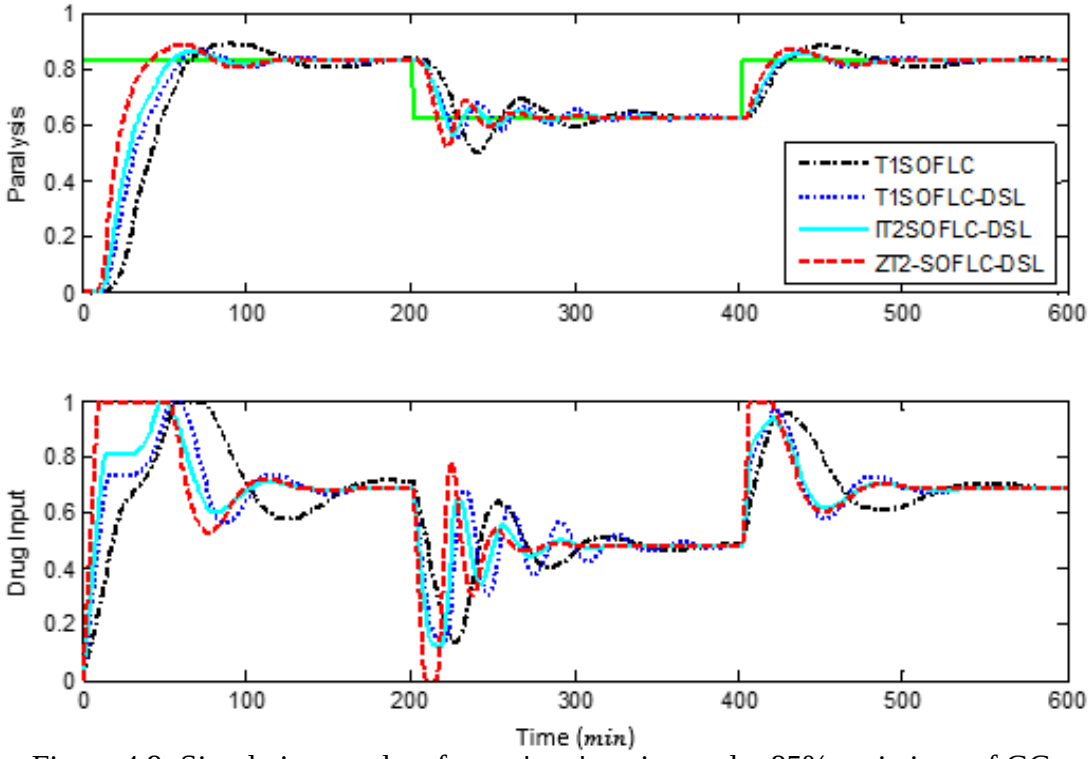


Figure 4.9: Simulation results of muscle relaxation under 85% variations of GC.

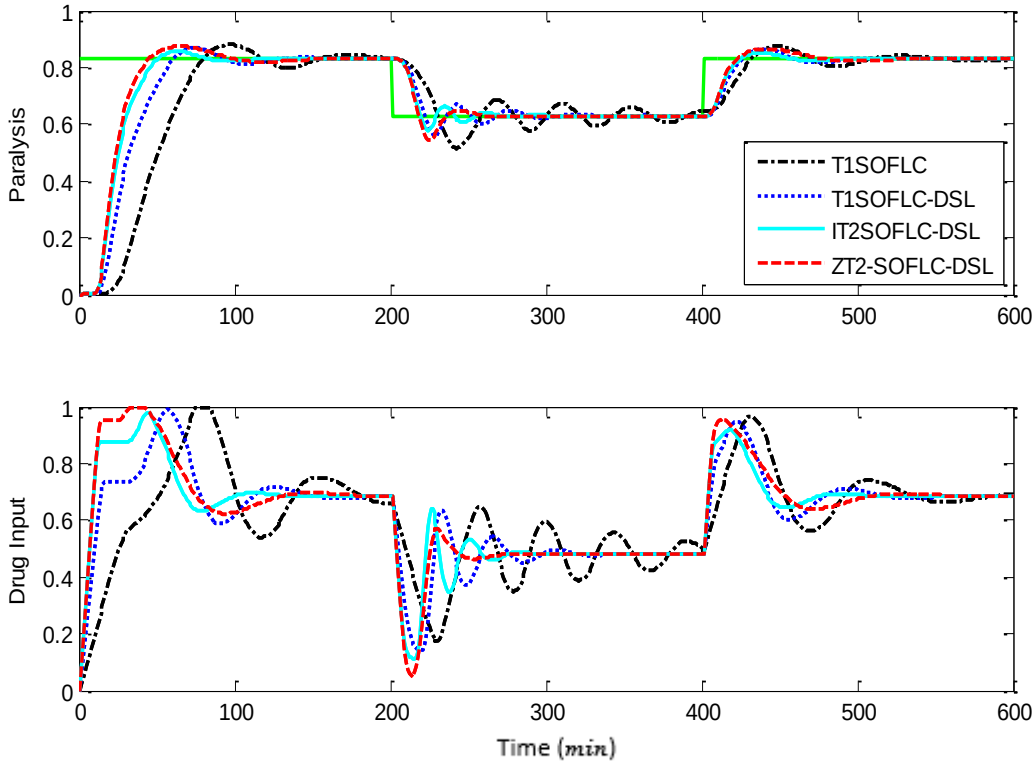


Figure 4.10: Simulation results of muscle relaxation under 50% variations of GT.

	Criteria	T1SOFLC	T1SOFLC-DSL	T2SOFLC-DSL	ZT2SOFLC-DSL
Figure 4.8	IAE	352.92	250.14	223.49	228.34
	ISE	166.41	129.32	121.35	118.35
	Rule number	20	18	16	15
Figure 4.9	IAE	499.52	363.95	308.45	259.23
	ISE	232.82	172.32	151.71	117.77
	Rule number	15	23	18	17
Figure 4.10	IAE	560.12	352.62	272.42	280.60
	ISE	273.90	17195	137.76	133.2
	Rule number	21	19	18	18

Table 4.2: Comparative performances of the SOFLC-DSL with various output scaling factors values.

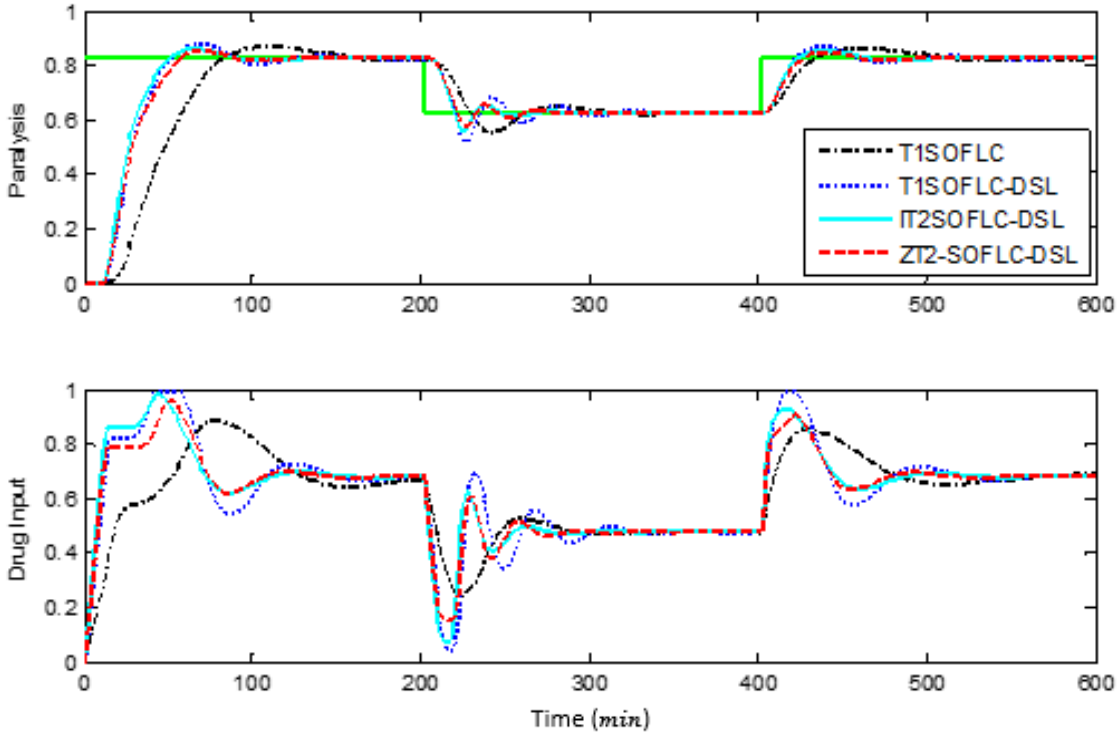


Figure 4.11: Simulation results of the proposed scheme and the standard scheme using a new set of system parameters: K_1=1, T_1=4.8 min, T_2=34.36 min, T_3=3.08 min, T_4= 10.65 min.

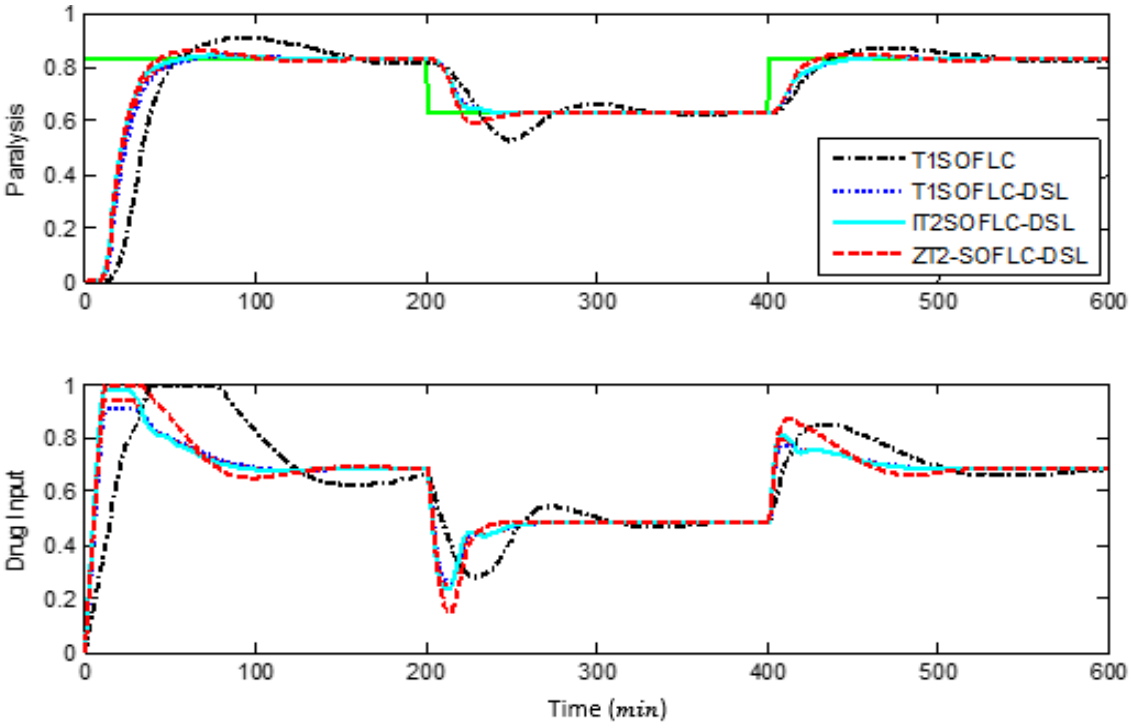


Figure 4.12: Simulation results of the proposed scheme and the standard scheme using a new set of system parameters: K_1=1, T_1=5 min, T_2=20 min, T_3=2 min, T_4= 7 min.

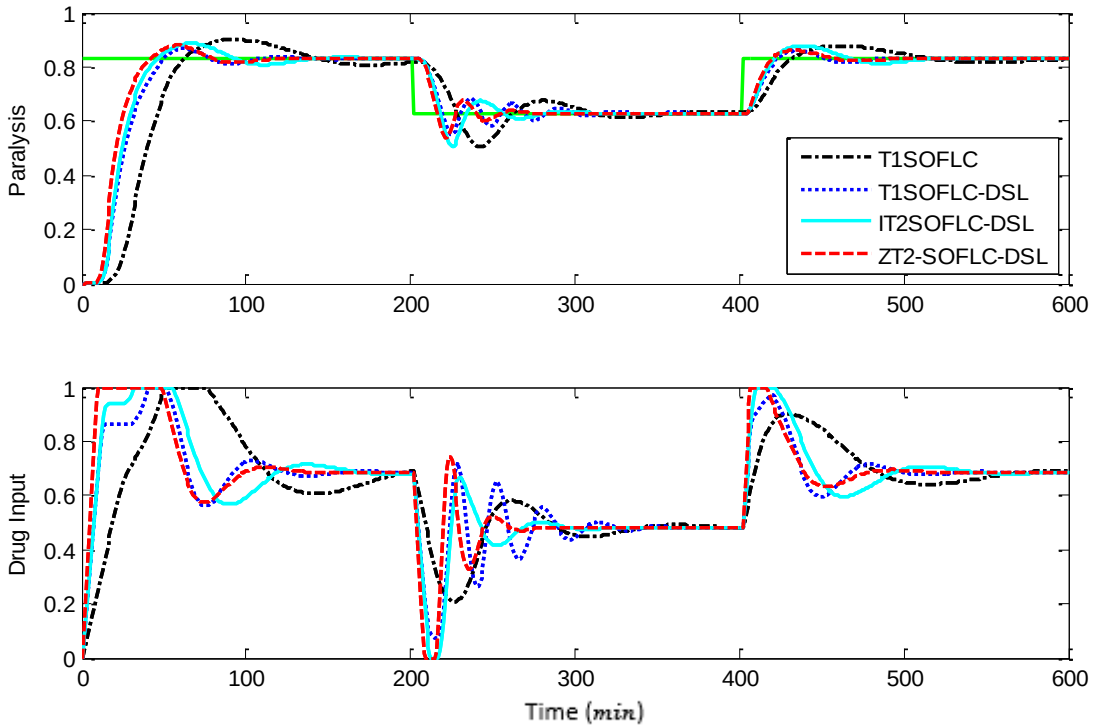


Figure 4.13: Simulation results of the proposed scheme and the standard scheme using a new set of system parameters: K_1=1, T_1=4.8 min, T_2=34.36 min, T_3=3.08 min, T_4= 10.64 min.

	Criteria	T1SOFLC	T1SOFLC-DSL	T2SOFLC-DSL	ZT2SOFLC-DSL
Figure 4.11	IAE	498.25	300.62	308.32	246.21
	ISE	226.84	142.96	139.21	117.21
	Rule number	22	19	17	15
Figure 4.12	IAE	5001.12	338.95	287.74	306.21
	ISE	247.21	156.66	142.54	156.32
	Rule number	21	18	16	18
Figure 4.13	IAE	468.62	270.71	247.51	259.91
	ISE	201.23	135.52	132.12	259.21
	Rule number	22	17	15	16

Table 4.3: Summary of performance criteria with different system dynamics.

4.7.1 Robustness in the stochastic case

Generally speaking, as discussed in previous chapters, one of the main advantages of type-2 fuzzy systems when compared to type-1 fuzzy systems is their ability to perform better in noise-contaminated environments. Unique simulations in which the four controllers were evaluated are presented in this section. In order to compare the performances of the different types of SOFLC, the steady-state error for muscle relaxation and the standard deviation of drug input, calculated over the duration of the entire simulation, were used. Furthermore, since the noise affecting the feedback signals has a direct impact on the results, each simulation experiment was repeated 15 times in order to take into account these effects within our analysis. The responses provided by the four controllers were ranked using the Wilcoxon signed rank test (Sidney Siegel, 1956), which is a technique used to compare paired samples.

In order to effectively evaluate the four controllers, different testing hypotheses were used, based on the assumption interval that type-2 SOFLC controllers tend to provide better capabilities in uncertain environments than type-1 controllers (e.g. type-1 SOFLC and type-1 SOFLC-DSL), and that the zSlice general type-2 SOFLC-DSL schemes would outperform all other three controllers.

Despite obtaining different simulation data in the repeated experiments, due to the introduced noise, the muscle relaxation values converged to similar values among the 15 experiments. Therefore, values that corresponded to 5%, 10% and 15% noise were introduced in the simulations experiments to produce the results used in our analysis.

The simulation results under 5% noise are shown in Figure 4.14, while the corresponding steady-state errors and control stabilities are presented in Table 4.4. We can see how both interval and zSlice type-2 SOFLC-DSL algorithms were maintained at the desired point for the duration of the entire simulation – at all three stages. However, the output produced by the type-1 SOFLC-DSL algorithm was very close to the desired values in stage 1, but became oscillatory at the other two stages. The performance of the type-1 SOFLC scheme, it can be observed, was not as good as the other types, in terms of slow rising time and poorer tracking capabilities at the three stages. It can also be seen from Figure 4.14 and Tables 4.4 and 4.5 that the drug input of the schemes that use

type-1 fuzzy sets included more fluctuations than type-2 SOFLC-DSL schemes, which is undesirable during surgical procedures.

Figures 4.15 and 4.16 show that as the level of noise increased, type-2 controllers demonstrated better control capabilities at the three stages; it can be clearly seen from these figures that type-1 SOFLC scheme still produced the poorest performance in terms of slow rise-time and larger tracking errors than the other three types of controllers.

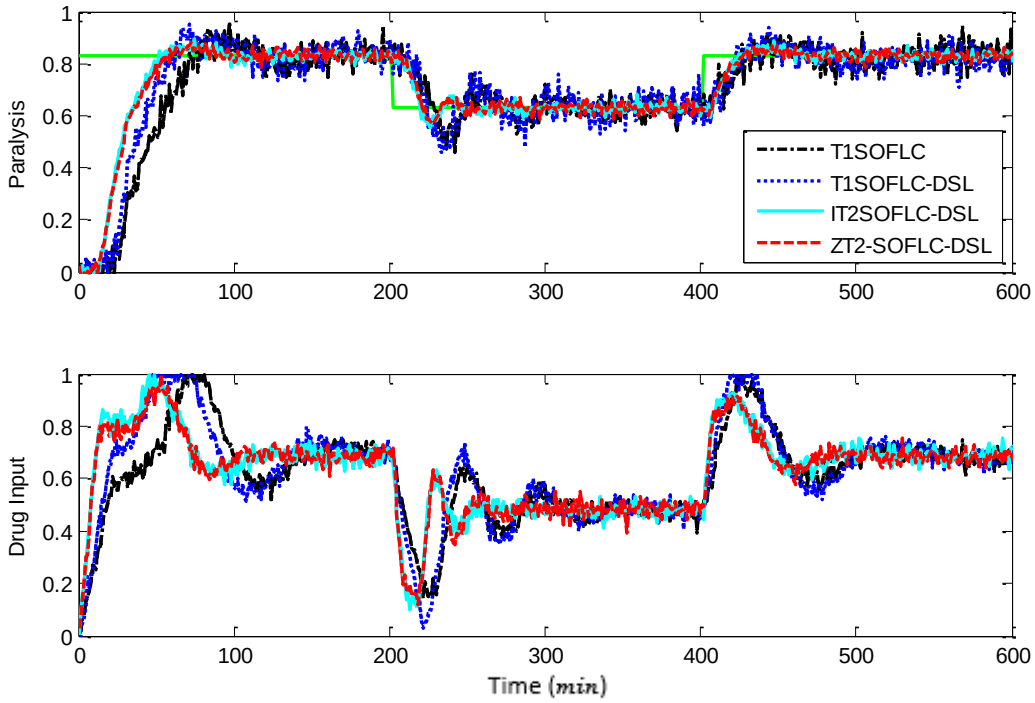


Figure 4.14: Simulation results of muscle relaxation under 5% noise.

It is also apparent that the type-2 SOFLC-DSL scheme (i.e. interval type-2 and zSlice type-2) provided fewer steady-state errors in the three stages and led to more stable control signals than the type-1 SOFLC-DSL algorithm, as shown in Table 4.4. One can easily conclude that type-2 controllers were better than type-1 controllers in this case. However, it is difficult to draw similar conclusions when comparing the zSlice general type-2 SOFLC-DSL with the interval Type-2 SOFLC-DSL. To better evaluate and compare these controllers in the 18 tests, the Wilcoxon signed-ranked test was applied, as can be seen in Tables 4.4 and 4.5.

The four controllers were tested against the 6 hypotheses for the 18 test cases as follows:

- *a): Type-1 SOFLC-DSL would outperform type-1 SOFLC*
- *b): Interval type-2 SOFLC-DSL would outperform type-1 SOFLC*
- *c): Interval type-2 SOFLC-DSL would outperform type-1 SOFLC-DSL*
- *d): zSlice general type-2 SOFLC-DSL would outperform type-1 SOFLC*
- *e): zSlice general type-2 SOFLC-DSL would outperform type-1 SOFLC-DSL*
- *f): zSlice general type-2 SOFLC-DSL would outperform interval type-2 SOFLC-DSL*

The steady-state errors and control activity obtained from the four controllers at the three stages while operating under 5%, 10% and 15% signals are recorded in Tables 4.4 and 4.5. Based on the results, one can see that type-1 SOFLC schemes proved to be the least effective of all the controllers, and that type-2 SOFLCs (i.e. interval type-2 and zSlice type-2) outperformed those of type-1 (i.e. type-1 SOFLC and interval type-1 SOFLC-DSL). Therefore, these results support hypotheses A, B, C, D and E in all the 18 experiments. However, Tables 4.4 and 4.5 also reveal that only 6 of the 18 tests support hypothesis F – that the zSlice general type-2 SOFLC-DSL would outperform the interval type-2 SOFLC-DSL – while the other 12 tests refute the hypothesis. It can be concluded that under the simulation conditions of this study, both the zSlice general type-2 SOFLC-DSL and the interval type-2 SOFLC-DSL provided similar results, and that it is not a straightforward matter to highlight significant differences between the schemes.

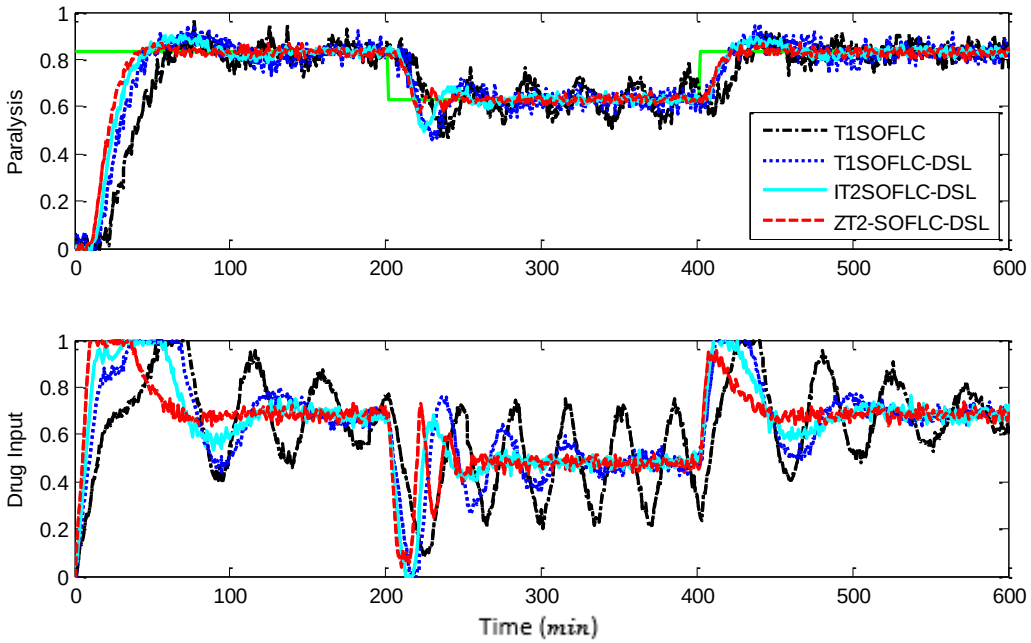


Figure 4.15: Simulation results of muscle relaxation under 10 % noise.

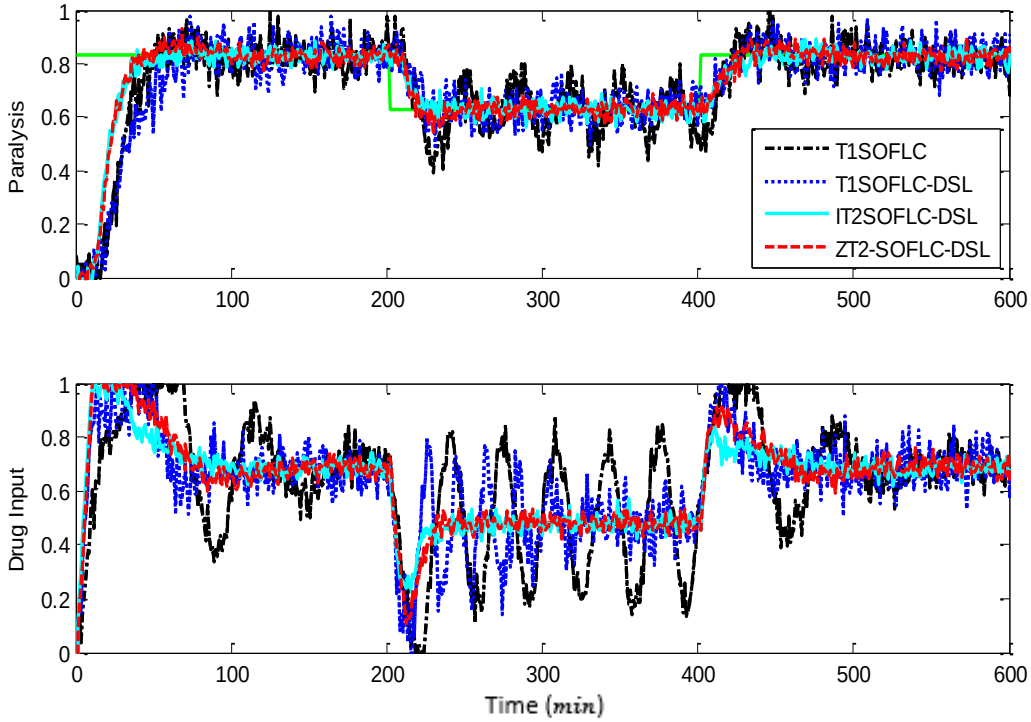


Figure 4.16: Simulation results of muscle relaxation under 15 % noise.

Noise Strength	Type	Stage-1	Stage-2	Stage-3
5 %	T1SOFLC	33.21(±) 25.24	32.85(±) 26.12	30.24(±) 28.19
	T1SOFLC-DSL	30.14(±) 29.35[a]	29.25(±) 28.35[a]	28.45(±) 27.74[a]
	IT2SOFLC-DSL	27.01(±) 26.85[bc]	26.58 (±) 24.92[bc]	25.65(±) 23.55[bc]
	ZT2SOFLC-DSL	27.89(±) 25.92[de]	26.02(±) 24.87[def]	25.21(±) 24.87[de]
10 %	T1SOFLC	36.97(±) 33.19	37.85(±) 32.12	30.24(±) 28.19
	T1SOFLC-DSL	29.94(±) 28.19[a]	33.52(±) 30.84[a]	29.87(±) 26.92[a]
	IT2SOFLC-DSL	27.32(±) 25.85[bc]	25.14(±) 20.92[bc]	24.24(±) 25.44[bc]
	ZT2SOFLC-DSL	25.69(±) 23.32[def]	21.34(±) 19.19[de]	20.65(±) 19.82[de]
15 %	T1SOFLC	39.83(±) 35.65	42.15(±) 32.12	41.99(±) 38.13
	T1SOFLC-DSL	38.14(±) 35.45[a]	38.74(±) 35.64[a]	38.07(±) 32.42[a]
	IT2SOFLC-DSL	28.31(±) 27.81[bc]	26.19(±) 22.44[bc]	29.15(±) 23.14[bc]
	ZT2SOFLC-DSL	29.39(±) 23.85[de]	26.45(±) 45.39[de]	24.88(±) 22.85[de]

Table 4.4: Means and standard deviations of steady-state errors of the four controllers with results of Wilcoxon signed-rank test.

Noise Strength	Type	Stage-1	Stage-2	Stage-3
5 %	T1SOFLC	38.65(±) 36.24	36.45(±) 31.24	33.94(±) 28.19
	T1SOFLC-DSL	36.32(±) 20.21[a]	33.45(±) 28.35[a]	32.42(±) 32.68[a]
	IT2SOFLC-DSL	30.15(±) 28.41[bc]	28.41 (±) 26.66[bc]	27.34(±) 24.87[bc]
	ZT2SOFLC-DSL	29.54(±) 26.74[de]	26.02(±) 26.47[def]	24.94(±) 29.47[de]
10 %	T1SOFLC	45.95(±) 40.199	48.54(±) 48.82	39.45(±) 31.10
	T1SOFLC-DSL	41.67(±) 40.89[a]	44.98(±) 40.88[a]	39.18(±) 36.87[a]
	IT2SOFLC-DSL	37.52(±) 35.67[bc]	35.45(±) 30.12[bc]	34.19(±) 30.56[bc]
	ZT2SOFLC-DSL	35.04(±) 31.85[def]	32.78(±) 28.29[de]	30.84(±) 28.21[def]
15 %	T1SOFLC	51.83(±) 35.65	53.87(±) 50.84	51.41(±) 49.93
	T1SOFLC-DSL	48.14(±) 45.45[a]	49.04(±) 46.79[a]	49.01(±) 44.92[a]
	IT2SOFLC-DSL	45.84(±) 41.18[bc]	42.19(±) 40.29[bc]	47.27(±) 45.87[bc]
	ZT2SOFLC-DSL	42.98(±) 39.89[def]	40.89(±) 38.99[de]	39.48(±) 36.95[de]

Table 4.5: Means and standard deviations of control stabilities of the four controllers with results of Wilcoxon signed-rank test.

4.8. Summary

In this chapter, the capabilities of the SOFLC-DSL algorithm proposed in Chapter 3 were enhanced to deal with uncertainty and noise by substituting its type-1 fuzzy sets with interval and zSlice general type-2 fuzzy sets. These controllers were used to control anaesthesia during three-stage surgical procedures, and were applied in different environments to test their effectiveness in controlling variable system dynamics and scaling factors and measuring their robustness in the stochastic case. The performance of type-1 SOFLC, type-1 SOFLC-DSL, interval type-2 SOFLC-DSL, and zSlice type-2 SOFLC-DSL was evaluated based on different tools, such as the number of produced fuzzy rules and IAE and ISE criteria, as well as steady-state errors and control activity. The simulations were repeated 15 times, and a Wilcoxon signed-rank test was used to compare the performances of the four controllers when they were implemented in noise-contaminated environments.

The results reveal that, in all simulations, the type-1 SOFLC produced the poorest performance of all the controllers. The type-1 SOFLC-DSL algorithm performed well in most simulations in terms of remaining steady at each of the three stages and having fast rise times, even though the drug input it provided in the noise-contaminated environments fluctuated significantly, which can be dangerous during surgical procedures. It was difficult to compare the interval and zSlice SOFLC-DSL with each other, as they both provided similar results in steady-state errors and control activity, although it can be easily seen from the data set out in most of the figures in this chapter that the latter normally reached the set-point faster than the former. The latter also provided the low-level fuzzy logic controller with the fewest fuzzy rules of all the controllers in most experiments, which is an important advantage.

Furthermore, the results produced in the Wilcoxon signed-rank test contradicted the widely accepted notion in the literature that when the levels of uncertainty are high, zSlice general type-2 sets normally provide better control capabilities than interval type-2 fuzzy sets. zSlice general type-2 sets usually lead to a more accurate representation in the third dimension of model uncertainty; this should allow for better responsive control with smoother control characteristics. However, the results in this chapter, particularly in the stochastic case, demonstrate that there were no significant differences in

simulations between the interval type-2 and zSlice general type-2 SOFLC-DSL algorithms. This could be for two main reasons. First, the levels of uncertainly experienced in these simulations may not have been high enough, not reaching the point where interval type-2 fuzzy sets fail to effectively model them. Second, the resolution of the zSlice general type-2 fuzzy sets represented by the number of slices being used may not have high enough. As outlined by Wagner and Hagras (2010), the ability of zSlice general type-2 fuzzy sets to effectively model high levels of uncertainty depends on the number of zSlices they are constructed from.

Figure 4.17 shows a box plot of the four controllers when they were repeated 15 times in different environments and when the IAE values were recorded. The lower limit of the plot represents the best performance; the upper limit corresponds to the worst performance; and the mean performance is represented by the line in between. .

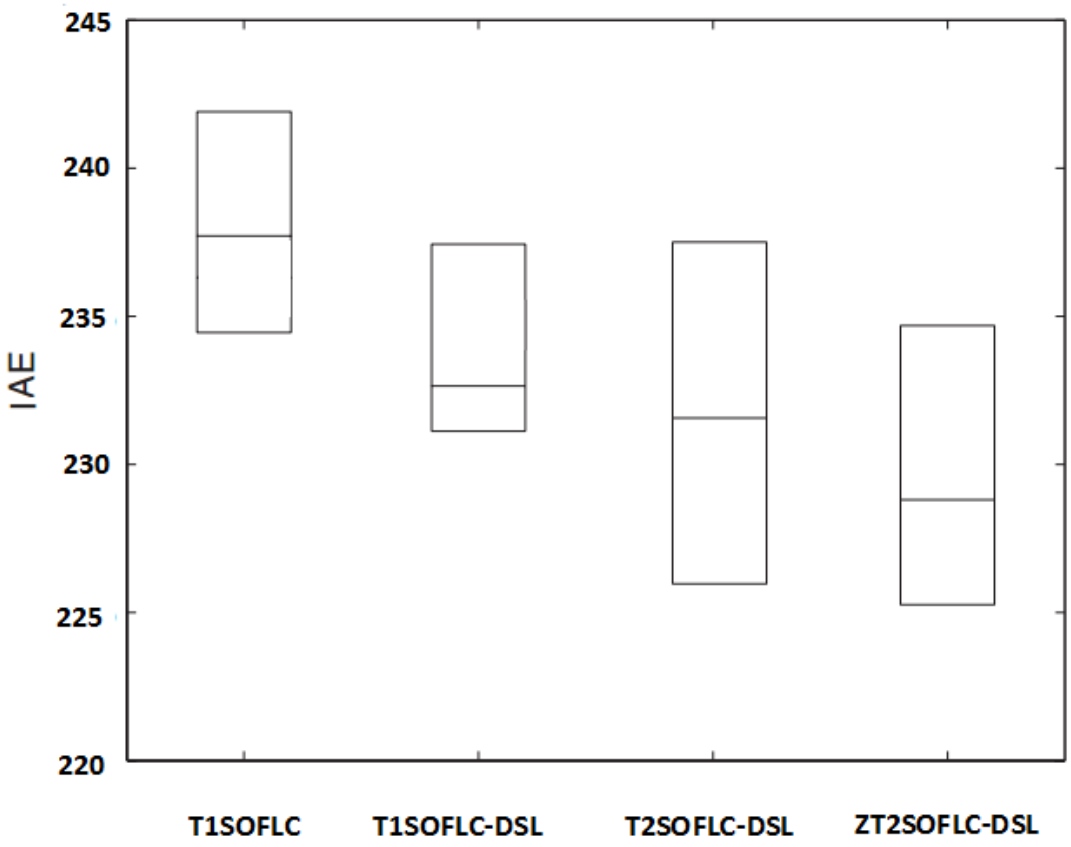


Figure 4.17: The box plot of the four controllers.

It can be observed that the interval and zSlice type-2 SOFLC-DSL algorithms led to similar performance, while they both outperformed the other two controllers; It can also be noted that the mean IAE values for the interval and zSlice type-2 SOFLC-DSL algorithms were lower than the best IAE of the type-1 SOFLC scheme.

The following chapter will include extending the SOFLC-DSL algorithm to deal with multi-input multi-output systems. The SOFLC-DSL algorithm will represent the main architecture of a decoupled multivariable control structure, which will be designed and tested.

Chapter 5 – Self-Organising Fuzzy Logic Control of Multivariable Systems with a Dynamic Supervisory Layer

5.1 Introduction

The design of controllers in cases of multi-input multi-output (MIMO) processes has to be carried-out in such a way that interactions between the input and output loops are carefully considered. In General, scalar controllers do not normally give satisfactory performance when the interactions are strong, and therefore multivariable controllers must be used. Decoupled control of MIMO systems overcomes the challenge of interactions between the control loops. It provides a simplified mechanism for designing fuzzy logic schemes by separating the control task into two main parts: one related to designing independent SISO or MISO systems to control each loop within the multivariable system; the other involves the use of compensators to deal with the interactions between the input and output loops of the system.

Interactions between the different loops in multivariable control can lead to the following effects:

- They can destabilise the closed loop system;
- They can make the tuning of controllers a very challenging task

Relative gain array (RGA) is a powerful tool for measuring the input-output interactions of MIMO processes, and has been adopted in various applications to determine the input-output pairing for decentralised control of multivariable processes.

In this chapter, a solid basis is first provided to explain the different techniques used for controlling multivariable systems and to show how the interaction effects between the input and output loops of these systems can be reduced. A multivariable fuzzy decoupling control architecture is then introduced. The SOFLC-DSL algorithm, proposed in Chapter 3, is incorporated within the decoupled control scheme for target-point tracking for each loop. Moreover, in order to handle the input-output interactions, a steady-state relative gain matrix is utilised. A switching mode compensator is then introduced and incorporated within the SOFLC-DSL algorithm. Furthermore, it is shown how such compensators can be used to improve the control capabilities of the controllers by monitoring the decoupling actions.

The newly proposed algorithms were applied and tested on various MIMO drug processes and their sensitivity in relation to various scaling factors; structure design and noise was also investigated.

5.2 An Overview of Multivariable Fuzzy Logic Control

Traditional fuzzy controllers normally succeed in dealing with complex, non-linear and ill-defined single-input single-output systems. These controllers can be extended and applied to multivariable systems if several single-input single-output loops are used. However, such simplified methods are not always possible in practice due to cross-coupling effects, in which any given input can impact more than any given output. Input-output interactions make it difficult for designers/operators to interpret the control strategies and design suitable rule-bases for achieving the desired performance. Furthermore, by using different SISO loops, the fuzzy rules and computational costs grow dramatically, making it difficult for the controller to be implemented in real-world applications.

The main difference between SISO and MIMO systems control is based on the estimation and compensation of the process interactions between all inputs and outputs. It is clear that in order for the MIMO systems control to provide the desired performance, coupling needs to be considered.

One possible architecture for dealing with this challenge, which is often reported in the literature, is focused on considering the whole system and finding the fuzzy relationships between the *n* inputs and *m* outputs. For such a rule-base, the k^{th} rule can be defined as:

$$\textbf{If } x_1 \text{ is } A_1^k \text{ and } x_2 \text{ is } A_2^k \text{ and .... and } x_n \text{ is } A_n^k,$$

$$\textbf{Then } u_1 \text{ is } B_1^k \text{ and .... and } u_m \text{ is } B_m^k \tag{5.1}$$

As outlined in Chapter 2, when the standard SOFLC scheme is applied to a multivariable system, a model of the process under control is normally required. The model helps the SOFLC scheme relate the modifications introduced by the performance index to the changes of the output of the controllers (Daley and Gill, 1986; Mamdani and Procyk, 1979).

5.2.1 The process reference model

A two-input, two-output system can be characterised by state-space equations, such as:

$$\dot{y}_1 = D(y_1, x_1, x_2) \tag{5.2}$$

$$\dot{y}_2 = E(y_2, x_1, x_2) \tag{5.3}$$

where x_1 and x_2 represent the inputs of the process, while y_1 and y_2 denote the outputs of the process.

The changes in $\dot{y}_1$ and $\dot{y}_2$ can be deduced by partially differentiating equations 5.2 and 5.3 with respect to x_1 and x_2:

$$\delta\dot{y}_1 = \frac{\partial D}{\partial x_1}\delta x_1 + \frac{\partial D}{\partial x_2}\delta x_2 \tag{5.4}$$

$$\delta\dot{y}_2 = \frac{\partial E}{\partial x_1}\delta x_1 + \frac{\partial E}{\partial x_2}\delta x_2 \tag{5.5}$$

After a sampling instant of nT, the changes in the outputs of the process can be obtained via the following assumption:

$$\Delta y_1 \approx T\delta\, \dot{y}_1 = T\frac{\partial D}{\partial x_1}\Delta x_1 + T\frac{\partial D}{\partial x_2}\Delta x_2 \tag{5.6}$$

$$\Delta y_2 \approx T\delta\, \dot{y}_2 = T\frac{\partial E}{\partial x_1}\Delta x_1 + T\frac{\partial E}{\partial x_2}\Delta x_2 \tag{5.7}$$

Equations 5.6 and 5.7 can be written in the form of a matrix as follows:

$$\Delta \mathrm{y} = M\Delta x \tag{5.8}$$

where

$$M = \begin{bmatrix} T\dfrac{\partial D}{\partial x_1} & T\dfrac{\partial D}{\partial x_2} \\ T\dfrac{\partial E}{\partial x_1} & T\dfrac{\partial E}{\partial x_2} \end{bmatrix}$$

As described in Chapter 2, if the required output change is represented by the vector $P_o(nT)$, then the manipulated input vector applied to the process can be represented as:

$$P_i(nT) = M^{-1}P_o(nT) \tag{5.9}$$

Equation 5.9 implies that fuzzy rules in the MIMO system that contribute to the current undesired response must be corrected accordingly.

In order to deal with the interactions between inputs and outputs effectively, the model must accurately represent the dynamics of the process. Alternatively, a static model can be used which defines the degree of interactions between the inputs and outputs in the form of constant parameters. However, this can be difficult to achieve, especially for large scale, complex systems with uncertain dynamics.

There are other available methods for designing a more satisfactory model for control design in the literature. One possible method is to use an indirect logic control approach, which is based on constructing a fuzzy model that captures the dynamics of the process.

Fuzzy rules-based models implement non-linear mapping from an input space to an output space. Fuzzy Sugeno-type models enjoy an important position in the field of modelling, and are commonly applied in a wide a range of applications (Takagi and Sugeno, 1985; Sugeno and Kang, 1988; Sugeno and Tanaka, 1991). The output membership functions of Sugeno-type models can either be linear or constant, unlike fuzzy consequents that are commonly used in Mamdani models (Mamdani, 1976).

Building a Sugeno-type model generally consists of two main parts. First, there needs to be a structural development of the model, which involves deducing how many fuzzy rules, and antecedent variables of these rules, will be needed and used. There are various available design approaches that provide the structure developments, including fuzzy trees (Mendonca *et al.*, 2007), fuzzy grids (Hu *et al.*, 2003) and fuzzy clustering (Sugeno and Yasukawa, 1993; Szabo *et al*., 2005; Yager and Filev, 1995). It is possible to regard the Sugeno-type model as a set of local linear models once the number of fuzzy rules and antecedent variables are deduced. The second phase required in the design of the Sugeno-type model is variable estimation. In linear models, variables can be obtained by pseudo-inversion or by applying the recursive least square approach (Chiu, 1994). Alternatively, these variables can be obtained using classical direct search-for-optimum methods, such as gradient descent and modified newton algorithms or population-based methods, including genetic algorithms, particle swarm optimisation and simulated annealing.

Li and Priemer (2003) proposed a self-learning fuzzy logic system (SFS) algorithm tuned by a random optimisation method aided by a training strategy. The proposed mechanism observed the behaviour of the input and output values of the process and then used them to adapt all the parameters of a Sugeno-type model. Li *et al.* (2004) proposed the concept of pseudoerrors, which are the potential errors that can occur during the process of constructing a self-organising neuro-fuzzy system (SO-NFS). The proposed algorithm employed a Sugeno type fuzzy system and used a clustering algorithm to construct the antecedent part of the rule-base. The pseudoerror-based mechanism provided the controller with all the accurate estimates it needs about the dynamics of the process. When applied to a crane system, the simulation results showed that the SO-NFS can effectively control the crane system in *x*, *y* and *z* directions.

In another research study, Sarimveis and Bafas (2003) developed a fuzzy model that captured the dynamics of the process to be controlled. The Sugeno-type model was then

used to predict future behaviours of the output of the process. The algorithm produced promising results when applied to a chemical reactor even when constraints were imposed on the control action.

In a similar work, by Shieh *et al*. (2006), an adaptive genetic fuzzy clustering algorithm was used to construct a fuzzy model. Generic algorithms were employed to tune the fuzzy rules of the model. The model was used as a basis for controlling anaesthesia. Simulation results revealed the capacity of the model to capture the dynamics of the process. Edwin (2008) proposed a different Sugeno-type model, based on new, incremental learning algorithms. The new algorithms tuned the parameters of both the rule consequents and the incremental learning of the premise parameters found in the fuzzy sets. The training of the model was carried-out both in on-line and off-line modes. The algorithms were applied in three applications, one of which included on-line identification of high-dimensional models at engine test benches. In all three applications, the algorithms performed well.

Different modelling methods, which combine both fuzzy logic and neural networks, have emerged over the last two decades, and have been applied by many researchers to various engineering applications. Neural networks are powerful learning tools which provide fuzzy inference systems with learning abilities from data and have offered a new, promising direction for the control of MIMO systems. Different types of neural networks are used as part of the fuzzy-neural systems, including Radial Basic Functions (RBF), Counter Propagation Networks (CPN) and Cerebellar Model Articulation Controllers (CMAC). Nie and Linkens (1993), for example, developed a fuzzified CMAC mechanism to act as a multivariable adaptive controller. The controller had as a feature self-organising association cells which enabled it to produce acceptable results when it was applied to a multivariable blood pressure control problem. Lin *et al.* (2008) introduced the hybrid fuzzy-CMAC-GA scheme that was used as a basis for an automatic landing system. The genetic algorithms were used to tune the control gains. The fuzzy-CMAC scheme improved the performance of systems in different aspects, such as local generalisation, rapid learning convergence and fuzzy interpretation capability.

Lee *et al*. (1994) employed an RBF network as part of a fuzzy system to construct a self-organising controller. Li *et al*. (2015) proposed what they called a self-organising cascade neural network (SCNN) with random weights. The network consisted of two fundamental phases, a simultaneous structure and a parameter learning mechanism. The SCNN

incorporated the weight optimisation into the architecture design. Simulation experiments demonstrated that the SCNN had fast learning abilities, and its generalisation performance was quite good even for non-linear systems.

Different researchers with an interest in the subject of MIMO control systems have been attracted to the adaptive fuzzy logic control based on Lyapunov theory. Kim and Yun (2000) developed a sliding mode controller to build a direct fuzzy adaptive controller. The controller has showed good robustness even in noisy environments. Similar work was conducted by Chiou and Huang (2005), where an adaptive fuzzy sliding mode controller was developed. An on-line tuning algorithm based on Lyapunov stability theory was used to adjust the consequent parameters of the fuzzy controller. The chattering phenomenon which is inherent in the sliding mode control was eliminated by introducing a boundary layer function. Simulation results showed that the novel scheme had good control abilities, stability and robustness.

In a similar work by Ho *et al.* (2003), an adaptive fuzzy sliding mode controller (AFSMC) was designed; the proposed algorithm used the robust proportional integral (PI) control law for developing SISO non-linear systems with external disturbances and uncertainty. The unknown system model was approximated using a fuzzy logic system, while the chattering action of the control signal was eliminated by the PI control law. The proposed system showed a good tracking performance when applied in simulation environments.

Researchers have incorporated different decomposition techniques as part of the MIMO control system. Decoupled controllers are today considered a promising architecture for dealing with the issue of interaction (or coupling) between inputs and outputs of the process to be controlled.

5.2.2 Fuzzy decoupled control of multivariable systems

As established previously, interactions between input and output channels can destabilise closed-loop systems and make the tuning of controllers very challenging. One way to deal with this challenge is to decouple the highly interacting multivariable systems into sets of independent loops so that a controller can then be assigned for each loop.

Figure 5.1 shows the general architecture of a decoupled control system. As can be seen, the MIMO process is divided into smaller SISO subsystems, each of which is controlled by

an independent controller (e.g.,G_{c1}). Also, in order to compensate for the residual interactions between the various input and output channels, different compensators (e.g.,G_{P11}) are used.

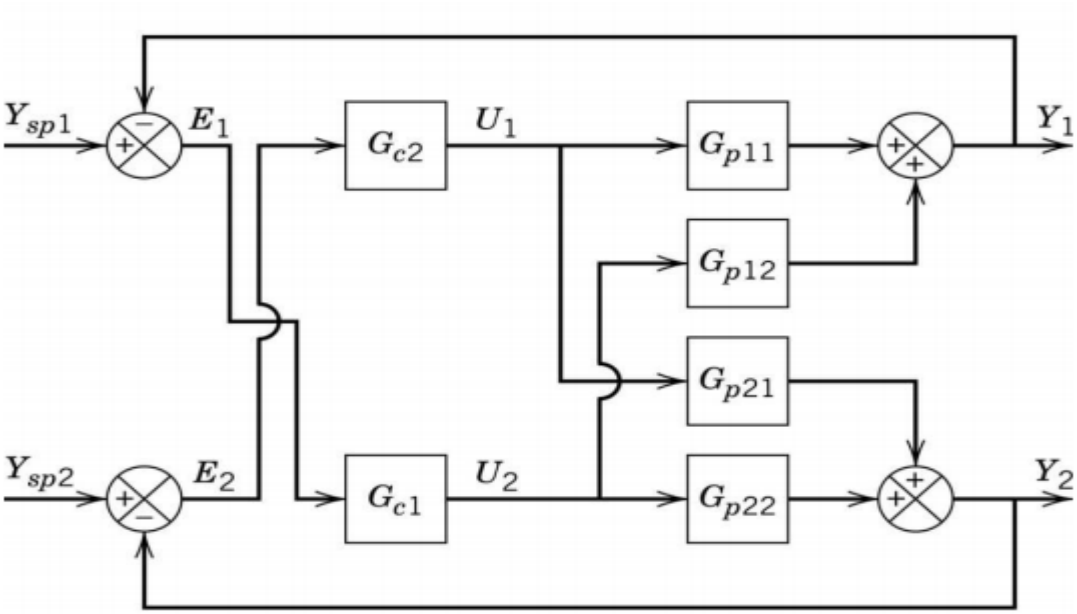


Figure 5.1: A 2 × 2 MIMO system (Linkens and Nie, 1995).

Chou *et al.* (2010) proposed an interesting technique for decomposing a multi-stage SOFLC scheme into many 2-input/1-ouput loops. This decomposition allowed the performance index tables to be arranged into simple 2-input/1-ouput spaces similar to those of the original SOFLC scheme. When the new scheme was utilised to control the general anaesthesia and muscle relaxation of patients, the simulation results revealed that it performed well, with a reduced steady-state error.

In many commonly used multivariable fuzzy control approaches in the literature, the interconnection between the input and output channels is omitted. These fuzzy control methods are generally used to reduce computational costs and ensure that the structure of the controllers remains as simple as possible. Lei and Langari (2000) proposed a hierarchical fuzzy algorithm to regulate and stabilise an inverted pendulum system. Based on human intuition, a supervisory fuzzy compensator was developed to compensate for interactions between the different channels.

With an accurate adjustment of control gains, the proposed controller produced better results than a traditional state feedback controller.

In the work of Lian and Huang (2001), a mixed fuzzy controller was designed based on the physical analysis of the process to be controlled. The mixed controller used a number of regular fuzzy controllers that were designed to deal with the different degrees of freedom in the MIMO system. In order to handle the interactions between the input-output channels, a decoupling fuzzy controller was used.

By using a combination of MISO fuzzy controllers, Tu *et al.* (2005) developed a multi-layer fuzzy compensator. When used to regulate a two-link cart-pole system, the compensator provided satisfactory performance in terms of regulation and stability. Koutb *et al.* (2004) proposed a fuzzy scheme that integrated the input and output parameters in a fuzzy relation used to capture the degree of interaction between the channels of a multivariable process. The proposed controller outperformed conventional multi-loop schemes when it was applied to a non-linear boiler-turbine process.

Based on a fuzzy dynamic model, Sun and Li (2000) introduced a MIMO fuzzy controller. Each degree of freedom of the process was controlled via a conventional fuzzy state feedback control law, while a global fuzzy inference scheme was used to compensate for the multivariable cross-coupling effect. In order to guarantee the system's stability, the concept of a connective subsystem was introduced. Mendonca *et al.* (2003) proposed a fuzzy predictive control for a MIMO system by introducing different constraints within a coordinated fuzzy decision-making architecture. The controller was tested on a container gantry crane and gave an acceptable control performance.

In the reviewed MIMO fuzzy control approaches, the control rules produced depended either on the experience of the operator/designer or on the qualitative analysis of the MIMO processes under control. These fuzzy rules fail to fully capture the relationships between the inputs and outputs due to the presence of vagueness, impreciseness or lack of information, and hence lead to deterioration in the performance of the control system. Furthermore, such structures make it difficult to guarantee the stability of the close loop systems. Therefore, more desirable performance in terms of increased accuracy of the fuzzy rules, and improved overall behaviour can be achieved by using a model-based MIMO fuzzy scheme.

The concept of Relative Gain Array (RGA) was proposed by Bristol (1966) and further studied by different researchers with an interest in the subject of linear multivariable control (Grosdidier *et al*., 1985; Gangopadhyay and Meckl, 2001; Sun and Li, 2000; Persechini *et al*., 2004; Camacho and Rojas, 2000). The RGA is a steady-state compensator, and is today used as powerful tool to eliminate the input-output interactions of MIMO systems using array elements. Only the steady-state gains of the system are required to obtain the relative gain arrays, which are then used to measure the impact of a certain input on a particular output of the multivariable system.

The basic idea behind the way in which a relative gain array is incorporated within multivariable control systems is based on pairing the manipulated parameters with the output of a MIMO system. The array then uses decoupling compensators to decompose the effect of the multivariable interactions. For each single-input/single-output loop within the MIMO system, an independent controller is applied.

Nevertheless, steady-state compensators such as relative gain arrays ignore the system's dynamics and fail to produce direct information about the performance of closed loops systems. One way to improve the control performance is to use off-line, pre-defined methods utilising trial-and-error (Mollov *et al.*, 2004). Linkens and Nyongesa (1996) designed a fuzzy logic scheme for a MIMO system. In this architecture, four independent fuzzy controllers were developed, two of which were the main controllers for the feedback systems while the other two were used to compensate for the interactions between the two inputs and the two outputs. These four controllers were tuned simultaneously using genetic algorithms.

Using a reference signal, Lee (1997) trained a fuzzy controller in the form of the neuron network. In order to cope with a situation where the desired states were not within the desirable range, the linear generalisation method via scaling factors was introduced. No compensators were needed in this algorithm since the information about the interactions between the different channels were fully captured by the reference model. The simulation results when the scheme was tested on a non-linear servomechanism proved the effectiveness of such a decoupling approach.

In order to relate the implied corrections, applied by the performance index table of a SOFLC algorithm, to the actual change of the output of the controller, Dobson and Roskilly (2002) introduced another fuzzy auto regression, moving an average, FARMA-

based SOFLC algorithm to model the system off-line. This model uses inputs that represent the system state's history and the control signals, whereas the one step-head predicted system state was used as the output provided by the model. In order to identify the fuzzy relation matrix, which can be used to predict the output of the process, Abilov *et al.* (2002) used a pseudo-random binary sequence. Similar work was conducted by Mollov and Babuska (2004), which involved designing a Sugeno-type model that they introduced in order to measure the degree of interaction using the output sensitivity function and the relative gain array method.

Nevertheless, these pre-designed interactive schemes can sometimes fail to deal with input-output interactions accurately and efficiently when the multivariable processes to be controlled involve unpredictability or uncertainty. One way to resolve this issue is to include on-line adjusting layers.

Bagheri and Moghaddam (2009) introduced a decoupled adaptive neuro-fuzzy (DANF) sliding mode control system that can be applied in applications where precise system models can be created. The algorithm is equipped with an on-line learning mechanism to tune the parameters of the fuzzy neuro-fuzzy controller. This mechanism is based on Lyapunov stability analysis to ensure that the controller is stable and provides the desired responses, even in the presence of uncertainty and disturbance. In a different work, by Liu *et al.* (2003), an intelligent hierarchical control scheme that integrated fuzzy self-tuning with adaptive control and auto-tuning techniques was designed as a basis for controlling a boiler turbine unit. A steady-state unilateral decoupler was incorporated to compensate for the strong interactions between the control loops. When the scheme was applied, the results showed that it was very robust, even though the structure of the boiler turbine unit was complex.

MA (2007) utilised fuzzy neural networks to develop a new, P–Q decoupled control scheme for the unified power flow controller (UPFC). The controller was introduced to effectively control power systems and reduce the effects of interactions between the parameters of the real and reactive power flow control. Simulation results of the new control scheme showed its superiority over other conventional power flow controllers.

In summary, and in the light of the review provided above, we can conclude that a major challenge facing MIMO control systems is to provide a suitable model of the process to be controlled without increasing the computational cost of the algorithm. This issue is

compounded when the fuzzy control rules of the controllers depend directly or indirectly on these models.

In general, the SOFLC algorithms for multi-input multi-output systems have attracted little interest from researchers due to the dependence of these controllers on inverse models for mapping between the state variables of the systems and the control inputs.

Recently, different optimisation tools and methods have been developed that can be incorporated within multivariable control systems to improve their performance, such as genetic algorithms, particle swarm optimisation and neural networks.

Another interesting conclusion is that decoupled control of MIMO systems is still attractive in terms of its ability to reduce the dimensions of fuzzy rules. A wide variety of compensators is available in the industry and has been proposed in the literature to deal with the interactions between input and output channels, such as the relative gain array. However, in most of these contributions, the use of the interactions of loops for set-point tracking has not been considered.

Adding an adaptation mechanism can help boost the control performance of a MIMO system. However, further work and analysis are still required to ensure that the adaption process is carried-out effectively and in a timely fashion, and that the process does not contribute to the growth of the computation burden. A good way to deal with this challenge is to consider on-line mechanisms that can adjust key parameters of the controller to ensure that the desired response is always achieved.

5.3 Multivariable System Control Using SOFLC-DSL

5.3.1 Decoupling structure and compensating scalars based on RGA

In order to investigate the decupling structure and the compensating scalars based on the relative gain array, let u_j and y_j define the input and output pair for a MIMO process $G(s)$ such that the steady-state gain matrix G is denoted as follows:

$$G = \begin{bmatrix} g_{11} & g_{12}\cdots & g_{1M} \\ g_{21} & g_{22}\cdots & \vdots \\ \vdots & \vdots & \vdots \\ g_{N1} & \cdots & g_{NM} \end{bmatrix} \tag{5.10}$$

Bristol (1966) states that if u_j is used to control y_j, then there will be two extreme cases:

- All other loops open: $u_k = 0, \forall k \neq j$.
- All other loops closed and provide perfect control performance: $y_k = 0, \forall k \neq i$.

Perfect control performance can only be achieved at steady-state. However, it is a good approximation at frequencies within the bandwidth of each loop. The gain for the two extreme cases can be evaluated as follows:

1) Other loops open:

$$\left(\frac{\partial y_i}{\partial u_j}\right)_{u_{k=0,k\neq j}} = g_{ij} \tag{5.11}$$

2) Other loops closed:

$$\left(\frac{\partial y_i}{\partial u_j}\right)_{y_{k=0,k\neq i}} = \bar{g}_{ij} \tag{5.12}$$

where, $g_{ij} = [G]_{ij}$, is the $ij'th$ element of G , while $\bar{g}_{ij}$ is the inverse of the is the $ij'th$ element of G^{-1}

$$\bar{g}_{ij} = \frac{1}{[G^{-1}]_{ij}} \tag{5.13}$$

In order to derive equation 5.13, it can be noted that:

$$y = Gu \Rightarrow \left(\frac{\partial y_i}{\partial u_j}\right)_{u_{k=0,k\neq j}} = [G]_{ij} \tag{5.14}$$

And interchanging the roles of G and G^{-1}, u and y, and i and j to obtain:

$$u = G^{-1}y \Rightarrow \left(\frac{\partial u_i}{\partial y_j}\right)_{y_{k=0,k\neq i}} = [G^{-1}]_{ij}{}^{-1} \tag{5.15}$$

The ratio between the gains in equations 5.11 and 5.12, as argued by Bristol, represents a useful measure of the degree of interactions between the input and output channels of the multivariable system. The ij^{th} gain γ_{ij}can be defined as follows:

$$\gamma_{ij} = \frac{g_{ij}}{\bar{g}_{ij}} = [G]_{ij}[G^{-1}]_{ij} \tag{5.16}$$

The relative array gain is the corresponding matrix of relative gains. From the above equation, one can see that the RGA, $\vartheta = G \times (G^{-1})^T$, where × defines the element-by-element multiplication.

For example, the relative gain array ϑ for a 2 × 2 process is defined as follows:

$$\vartheta = \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix} = G \times (G^{-1})^T = \begin{bmatrix} \frac{g_{11}g_{22}}{\rho} & -\frac{g_{12}g_{21}}{\rho} \\ -\frac{g_{12}g_{21}}{\rho} & \frac{g_{11}g_{22}}{\rho} \end{bmatrix} \tag{5.17}$$

where

$$\rho = g_{11}g_{22} - g_{12}g_{21} \tag{5.18}$$

Based on the matrix above, it can be seen that the parameters u_j and y_j will only be paired if the corresponding gain γ_{ij} is close to 1. This, in other words, means that the gain between u_j and y_j is not affected by closing the other loops. Also, the instability caused by the interactions at low frequencies can be avoided by avoiding pairing with negative steady-state RGA elements (Skogestad and Postlethwaite, 2005).

Figure 5.2 shows a multivariable architecture that uses a predefined RGA matrix for selecting the input and output pairs. It can also be seen that the SOFLC-DSL schemes are incorporated as dominating controllers to regulate each loop.

The steady-state gains were used to compensate for the interactions between the different loops within the system. For the first loop, 1, the applied input u_1^* is formed by combining both the main control inputs u_{11} and the compensating inputs u_{12}.

The actual process control inputs of the other loop are formed using the same reasoning.

The relationship between the control inputs u_{i}, provided by the domination controllers, and the process control inputs u_{i}^*, can be represented as follows:

$$\begin{bmatrix} u_1^* \\ u_2^* \end{bmatrix} = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \Lambda \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (5.19)$$

where $\Lambda = \begin{bmatrix} 1 & -\frac{g_{12}}{g_{11}} \\ -\frac{g_{21}}{g_{22}} & 1 \end{bmatrix}$ and is called the compensation matrix.

The basic operation of the compensator can be used to derive the compensating scalars $\lambda_{12}$and λ_{21}, while other two scalars λ_{11} and λ_{22} can be set to 1 without loss of generality.

Assume that there is in channel 1 a change of Δy_1 that results from the interaction effect from Δu_2 of loop 2. Hence:

$$\Delta y_1 = \Delta u_2 . g_{12} \quad (5.20)$$

Here, g_{12} represents the parameter of the steady-state matrix *G* in the form of equation 5.10.

The interaction effect caused by loop 2 on loop 1 can be reduced by the compensator that generates an alteration within the control input of loop 1 Δu_1, which then results in an equivalent alteration in $-\Delta y_1$.

Therefore,

$$\Delta y_1 = \Delta u_2 \, . \, g_{12} = -\Delta u_1 \, . \, g_{11} \quad (5.21)$$

Or

$$\Delta u_1 = \left(-\frac{g_{12}}{g_{11}}\right) . \Delta u_2 = \lambda_{12} . \Delta u_2 \quad (5.22)$$

$$\lambda_{12} = -\frac{g_{12}}{g_{11}} \quad (5.23)$$

The compassing scalar λ_{22} can be obtained using the same process.

As explained by Nie and Linkens (1995), the variables of the compensators are calculated based on the steady-state gains. Therefore, these compensators remove the interactions

between the input and output channels, not completely but partly, particularly in the transient regions. The remaining number of the effects of this interaction are compensated for by the main controller for each loop.

5.3.2 Switching mode linguistic compensator

In most previous contributions that have studied the interactions between the different variables in multivariable systems, the main concern has mainly been on developing compensators that can deal with these interactions. Little consideration has been paid to how such interactions can impact the different loops.

In the work of Lu and Mahfouf (2006), a compensator-switching strategy was proposed to improve the capabilities of the SOFLC scheme. This strategy is adopted in this thesis.

The way to determine if the interactions can lead to improved error tracking capabilities using this strategy can be estimated as follows.

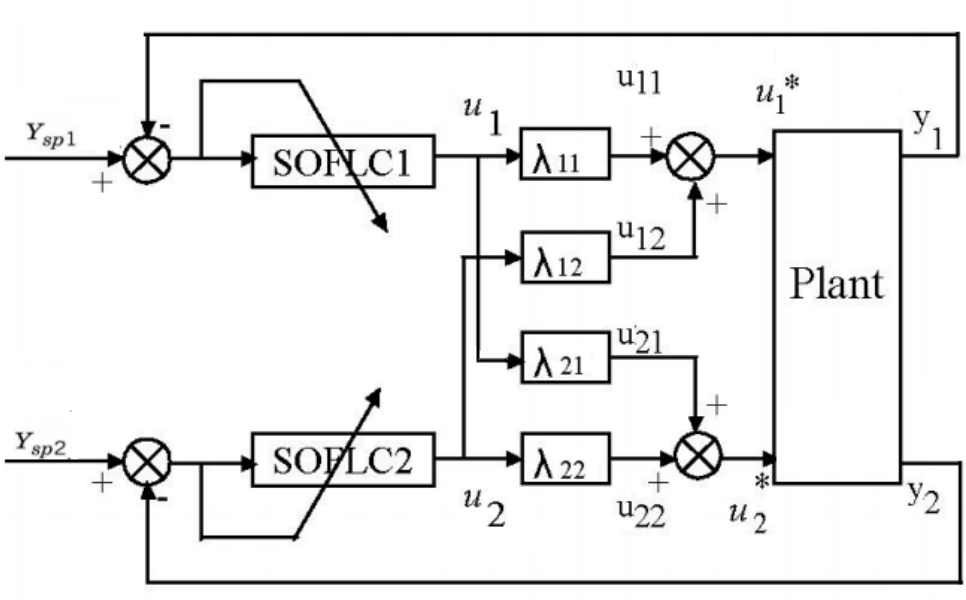


Figure 5.2: Decoupled system using RGA compensators (Lu and Mahfouf, 2006).

First, one can evaluate the output of the dominating controller using the following rule:

IF $\bar{E}_i(nT + kT) < 0$

THEN $u'_{ii}(nT) < u_{ii}(nT)$ is needed for smaller tracking error

ELSE $u'_{ii}(nT) > u_{ii}(nT)$ is required for smaller tracking error (5.24)

where $\bar{E}_i(nT + kT)$ is the predicted tracking error in loop i, k steps ahead of the current sampling instant, which can be calculated by equations 5.25; u_{ii} is the output of the dominating controller in loop i; and u'_{ii} is the output of the ideal estimated controller that makes the output of the process tracks the set-point closer.

$$\bar{E}_i(nT + kT) = E_i(nT) + k \times (E_i(nT) - E_i(nT - T)) \tag{5.25}$$

Second, the mechanism in which the switching strategy works is very simple and can be summarised as follows.

If the dominating control signal u_{jj} in the interacting channel j helps to reduce the difference between u_{ii} and u'_{ii}, then the interaction is allowed . Otherwise, the interaction caused by loop j on loop i is counteracted by the compensating control force of u_{ij}.

When such a strategy is adopted, some of the efforts of the designed compensating controller are utilised to deal with the interactions' effects, while the remaining interactions are made available to the main controller for the purpose of reducing the errors in the process and enhancing the control capabilities of the scheme.

This design strategy is different from that relating to traditional decoupled controllers, which are generally designed to deal only with the interactions of the different loops of the MIMO system without considering the use of the effects of these interactions to enhance the overall performance of the controller.

It is worth noting that, regardless of the status of the current error-tracking values, the compensator remains switched on as long as the system is in steady-state. This is particularly useful when dealing with any case where both loops are in steady-states and, all of sudden, loop i experiences a change. In such instances, the interaction effect can be large enough that it forces the tracking signal of loop j away from the desired values, regardless of the polarity of the interaction.

The described switching strategy of the compensator that is utilised to deal with the interactions caused by loop j on loop i is summarised in Figure 5.3.

IF the system is in the steady-state area

THEN the compensator is switched-on

ELSE

IF $u'_{ii}(nT) < u_{ii}(nT)$ AND $u_{jj} > 0$

THEN the compensator is switched-on

ELSE IF $u'_{ii}(nT) > uii(nT)$ AND $u_{jj} < 0$

THEN the compensator is switched-on

ELSE the compensator is switched-off

Figure 5.3: The switching mode linguistic strategy (Lu and Mahfouf, 2006).

Here, u_{jj} and u_{ii} are the outputs of the dominating controllers, and u'_{ii} is the expected control input provided according to the evaluation of the system performance set out in equation 5.25. It should also be noted that in this equation, the switching strategy is based on the assumption that when u_{jj} and u_{ii} have the same polarity, they both provide the same increase/decrease effect on loop i.

If this is not satisfied, then the following rules are considered:

IF $u'_{ii}(nT) < u_{ii}(nT)$ AND $u_{jj} > 0$, THEN the compensator is switched-on needs be replaced by:

IF $u'_{ii}(nT) < u_{ii}(nT)$ AND $u_{jj} > 0$, THEN the compensator is switched-on.

For the other rule, a similar alternation in the condition part should be generated.

In short, the compensator provides the required control signals as follows:

- The predicted tracking error $\bar{E}_i(nT + kT)$is initially calculated using equation 5.25
- The switching mode can then be determined via the strategy in Figure 5.3
- The current control signal $u_{ii}(nT)$ can be evaluated by applying equation 5.24
- Finally, the output of the compensator can be calculated as follows:

$$u_{ij} = \lambda_{ij} \times u_{jj}$$

$$\lambda_{ij} = \begin{cases} = \dfrac{g_{ij}}{g_{ii}} & if\ compensator\ is\ switched - on \\ 0 & if\ compensator\ is\ switched - off \end{cases}$$

5.4 Simulation on a Multivariable Dynamic Process

In modern pharmacology there are two main categories. First, there is pharmacokinetics (PK), which deals with a given drug's concentration with respect to time and dose schedule. Second, there is pharmacodynamics (PD), which studies drug concentrations and the effect a drug can have on the body of a patient. There are many different variables within the body that need to be regulated during surgical procedures, such as blood pressure and cardiac output. This task is normally carried-out by the use of different drugs, including the vasoactive drug sodium nitroprusside (*SNP*) and the inotropic drug dopamine (*DOP*). The injection of these drugs affects different parameters within the body of the patient, such as Mean Arterial Pressure (*MAP*) and Cardiac Output (*CO*).

The following model represents a drug dynamics process with two-inputs two-outputs (Nie and Linkens, 1995).

$$\begin{bmatrix} \Delta CO \\ \Delta MAP \end{bmatrix} = \begin{bmatrix} 1.0 & -24.76 \\ 0.6636 & 76.38 \end{bmatrix} \begin{bmatrix} \dfrac{k_{11}e^{-\tau_1 s}}{sT_1+1} & \dfrac{k_{12}e^{-\tau_2 s}}{sT_1+1} \\ \dfrac{k_{21}e^{-\tau_2 s}}{sT_2+1} & \dfrac{k_{22}e^{-\tau_2 s}}{sT_2+1} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

$$= \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \tag{5.26}$$

Here, $k_{11} = 8.44$; $k_{12} = 5.275; k_{21} = -0.09; k_{22} = -0.15; \tau_1 = 60s; \tau_2 = 30s; T_1 = 84.1s; T_2 = 58.75s$. I_1 is the infusion rate of the inotropic drug dopamine; I_2 is the infusion rate of the vasoactive drug sodium nitroprusside.

One can notice from the model in 5.26 that there are strong interactions between the different loops and it also has large time delays.

The steady-state gain matrix G in the linear case can simply be represented by $H(s)$

$$G = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = H(0)$$

(5.27)

$$= \begin{bmatrix} 1.0k_{11} - 24.76k_{21} & 1.0k_{12} - 24.76k_{22} \\ 0.6636k_{11} + 76.38k_{21} & 0.6636k_{12} + 76.38k_{22} \end{bmatrix}$$

The relative gain array matrix ϑ is defined as:

$$\vartheta = \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix} = G \times (G^{-1})^T = \begin{bmatrix} \frac{g_{11}g_{22}}{\rho} & -\frac{g_{12}g_{21}}{\rho} \\ -\frac{g_{12}g_{21}}{\rho} & \frac{g_{11}g_{22}}{\rho} \end{bmatrix}$$

(5.28)

$$\vartheta = \begin{bmatrix} 1.1559 & -0.1559 \\ -0.1559 & 1.1559 \end{bmatrix}$$

The control loop for the process shown above, based on the elements of the relative gain array matrix, can be formed by pairing *DOP/CO* and *SNP/MAP*. This makes sense considering that *SNP* is used to reduce the Mean Arterial Pressure (*MAP*) while *DOP* is used to increase Cardiac Output (*CO*).

The SOFLC-DSL based decoupled scheme shown in Figure 5.2 was applied via two primary control routes: *DOP/CO* and *SNP/MAP*. Equations 5.19 and 5.28 were used to calculate the compensating scalars.

Similar to the work conducted in previous chapters, the lower-level fuzzy logic rule-base as well as the performance index table adopted in this chapter use five fuzzy sets, denoted in an equally-partitioned universal discourse, for both input signals, *E* and *CE*. A sampling interval of 30s and square target-point signals with three different modes for *CO* and *MAP* were also used in this simulation study.

The produced outputs when the SOFLC-DSL algorithm with switching mode compensator was applied are shown in Figure 5.4. We can clearly see that the output singles accurately tracked the set-points at all three stages, while the control actions remained smooth throughout. This clearly indicates that the switching mode strategy enabled the system to effectively deal with the interaction and that it enhanced the tracking capabilities, particularly in the transient region.

Figures 5.5 and 5.6 show the response of the system when the SOFLC-DSL algorithm with relative-gain, array-based scalar compensators, and a conventional SOFLC scheme with a fixed performance index table, were used respectively. The results reveal that the SOFLC-

DSL which included the switching mode compensator outperformed the other two schemes for both *CO* and *MAP* in terms of making the system track the target-point with fewer fluctuations.

One more interesting observation is that the SOFLC-DSL algorithm still provided better performance than the standard SOFLC scheme, even when the former was applied without the switching mode strategy.

5.4.1 Robustness of the proposed algorithm to scaling factors

In order to test the performance of the proposed scheme under varied scaling factors, the scaling factors for both the *CO* and *MAP* loops were varied for the SOFLC-DSL using a switching mode compensator, as shown in Figures 5.7~ 5.9.

Table 5.1 data compare the latter scheme with two other schemes: the conventional SOFLC with switching mode compensator and the SOFLC-DPI with RGA-based scalar compensators. The results reveal that SOFLC-DSL with a switching mode compensator had a robust response to scaling factors variations and produced the best results when compared to the other two schemes. It is apparent that the ability of the algorithm to modify its structure on-line through the supervisory layer enhanced its effectiveness and capability to deal with the variation in scaling factors. It is also noted that the standard SOFLC scheme still produced the poorest performance even when it was aided by the switching mode compensator.

5.4.2 The system's robustness to model parameter variations

The ability of the proposed algorithm to deal with the variations in the parameters of the model is vitally important because human bodies have different characteristics, and therefore the constants shown in model 5.26 differ from one person to another.

All the model variables were varied with respect to the original model, as shown in equation 5.26. The proposed algorithm was tested to see how it would behave when applied in a real-world application with different patients, as shown in Figures 5.10 ~ 5.14. The results reveal the proposed decoupled scheme was robust in its response to varying system dynamics in both *CO* and *MAP*. It is interesting to note that, even when the switching mode compensator was not used, the proposed scheme still provided acceptable

results that were better than the standard SOFLC scheme. This is once again due to the capacity of the proposed algorithm to modify its structure on-line, and also its ability to evaluate the performance of the current error tracking and the interaction's effect on that loop.

5.4.3 Robustness in the stochastic case

The 2nd-order polynomial fitting method studied in Chapter 3 was incorporated into the proposed decoupled system to enable the SOFLC-DSL to work within noise-contaminated environments. A good system performance is demonstrated in Figure 5.15 when the proposed scheme was applied under 5% output noise.

5.3.4 Controller's robustness to the compensator design

As described in previous sections, the proposed compensators were developed based on the steady-state gains of the system. It is therefore very important for these controllers to be able to work effectively even when the estimate of steady-state gains is inaccurate. In order to carry-out this task, four different scenarios involving estimates were considered to test robustness.

- Scenario 1: $\lambda_{c12} = 0.6 \times \lambda_{c12}{}', \ \lambda_{c21} = 1.6 \times \lambda_{c21}{}'$
- Scenario 2: $\lambda_{c12} = 1.6 \times \lambda_{c12}{}', \ \lambda_{c21} = 0.6 \times \lambda_{c21}{}'$
- Scenario 3: $\lambda_{c12} = 0.75 \times \lambda_{c12}{}', \ \lambda_{c21} = 0.9 \times \lambda_{c21}{}'$
- Scenario 4: $\lambda_{c12} = 0, \ \lambda_{c21} = 0$

where $\lambda_{cij}{}'$ represents the initial component element utilised to deal with the interactions from loop j to loop i; λ_{cij} represents the value utilised in the simulation. These two values, λ_{c12} and λ_{c21}, altered by a different amount in each case, which represents the deviation between the actual and estimated steady-state gains. The results of the simulation shown in Figures 5.16 ~ 5.19 demonstrate that the proposed decoupled system performed well in the first three cases, and still provided a satisfactory response in the last scenario.

This reveals that the proposed strategy copes effectively even when the estimation of steady-state gains is not accurate.

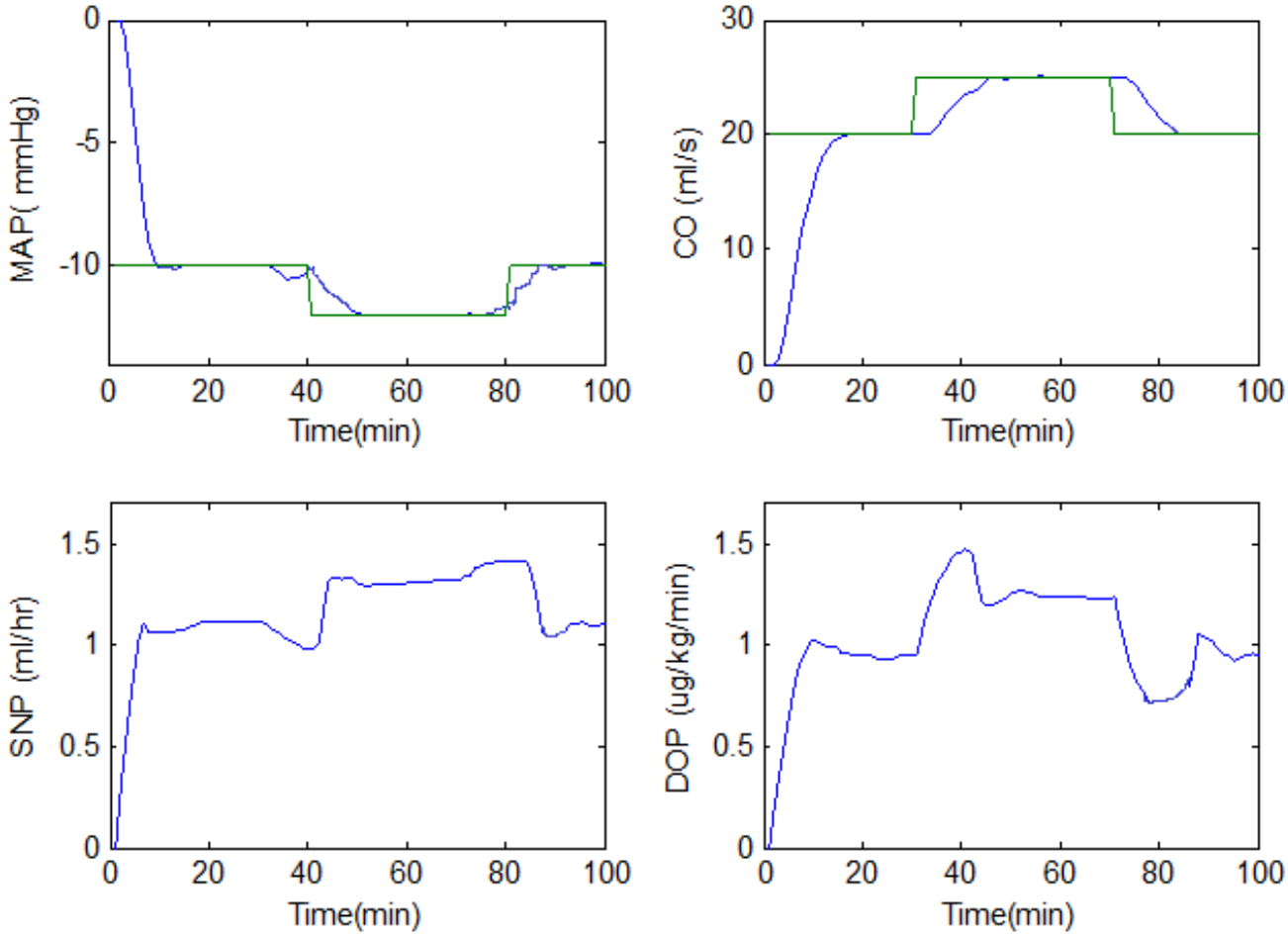


Figure 5.4: System response using the SOFLC-DSL algorithm with switching mode compensator.

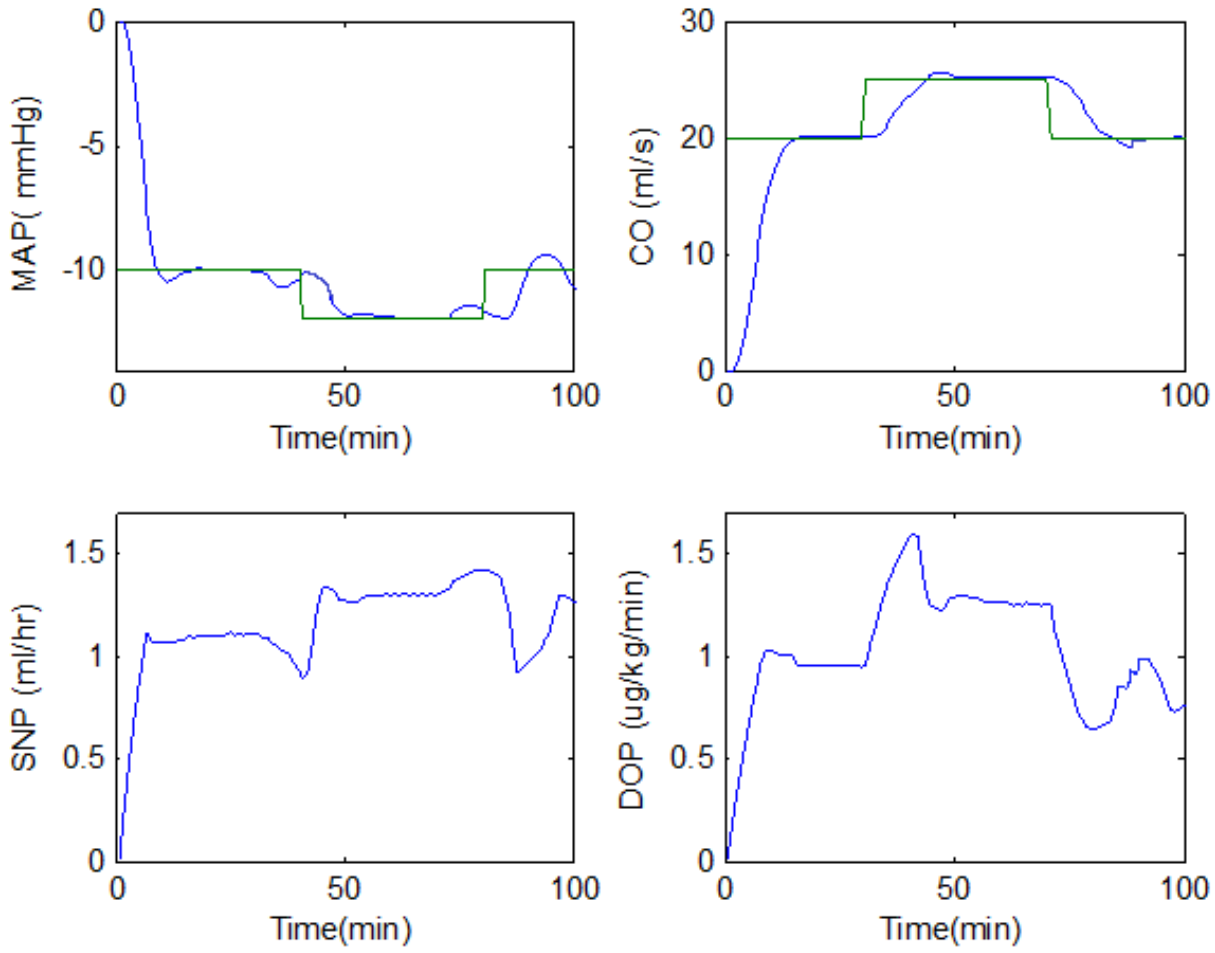


Figure 5.5: System response using the SOFLC-DSL algorithm with RGA based scalar compensators.

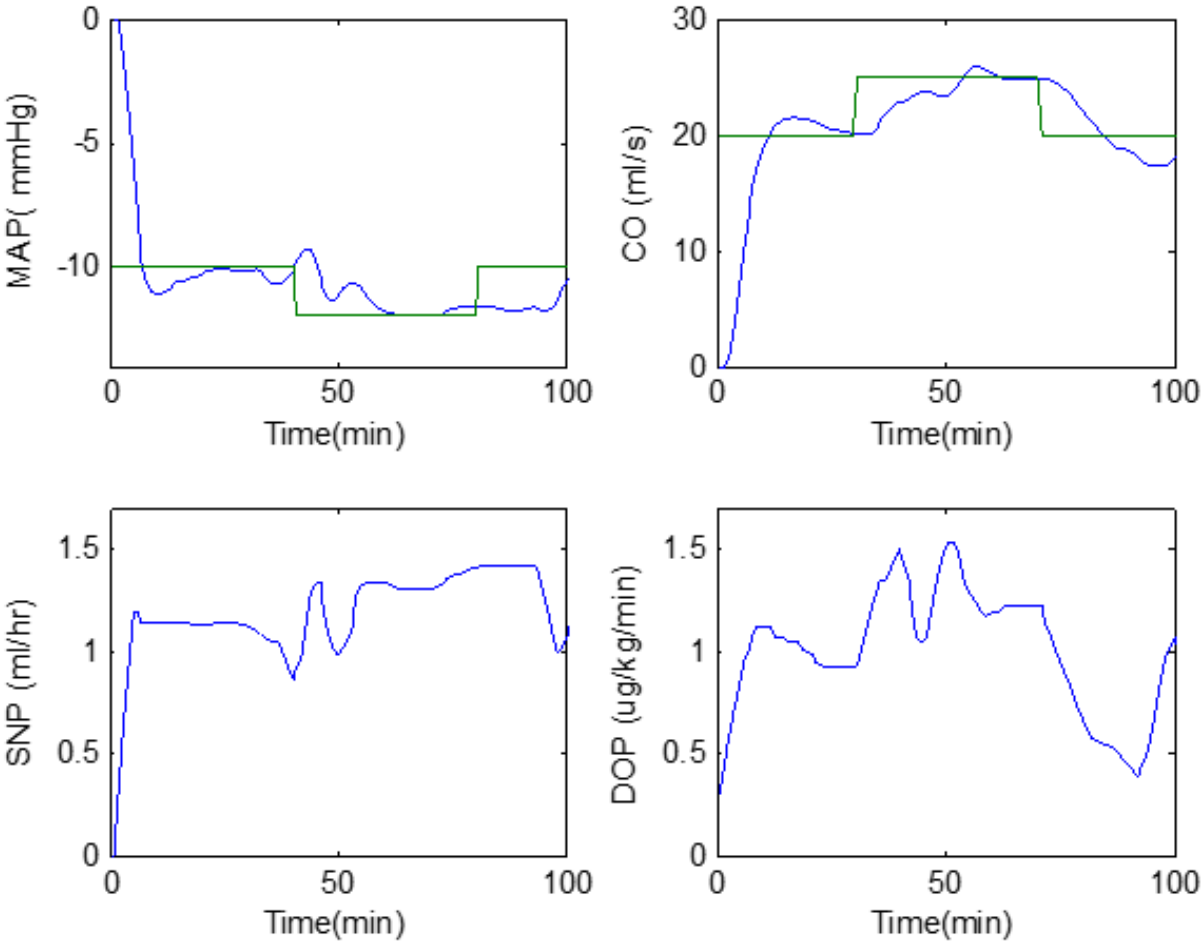


Figure 5.6: System response using the SOFLC algorithm with the switching mode compensator.

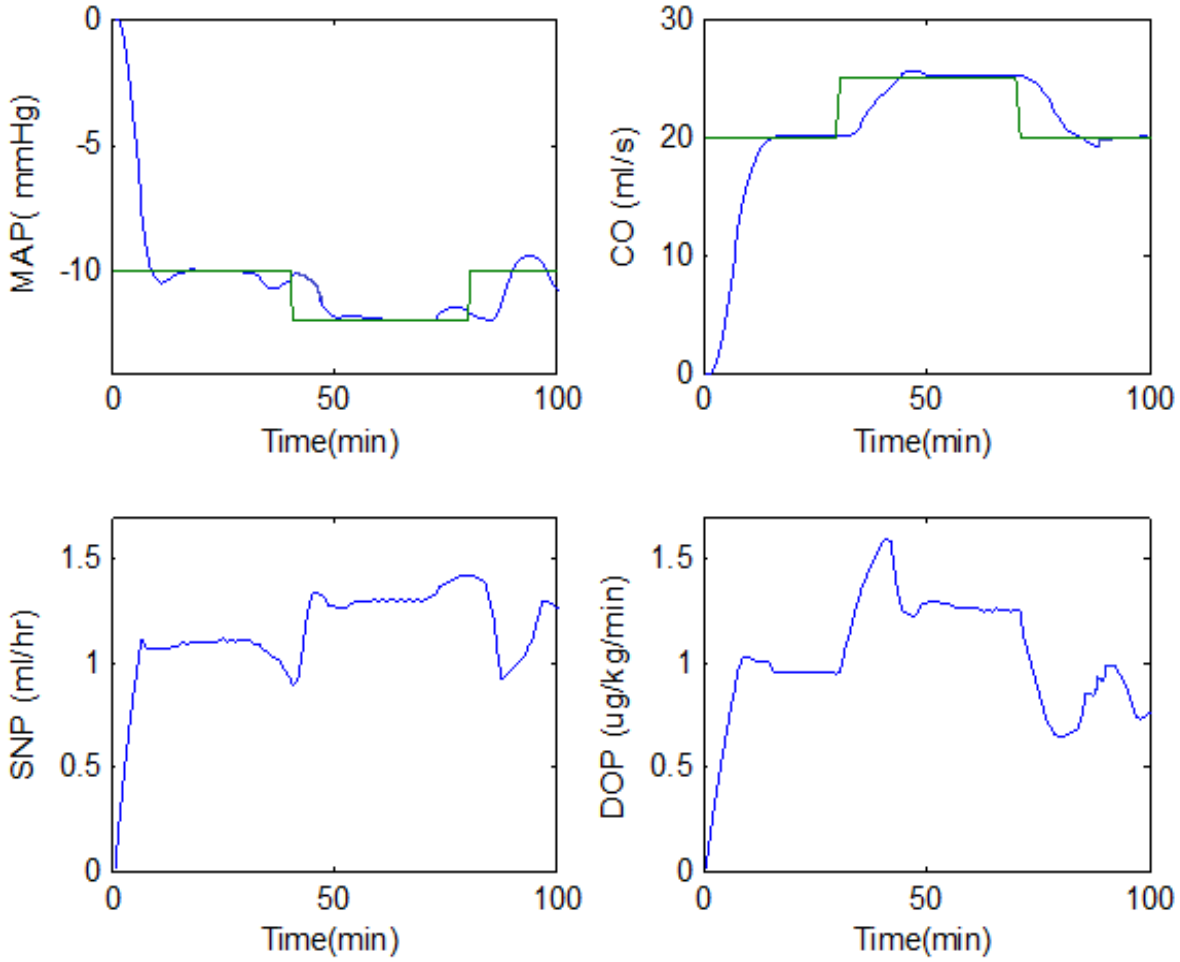


Figure 5.7: System response using the SOFLC-DSL algorithm with the switching mode compensator under 50% variations of *GE* for *CO* and *MAP*.

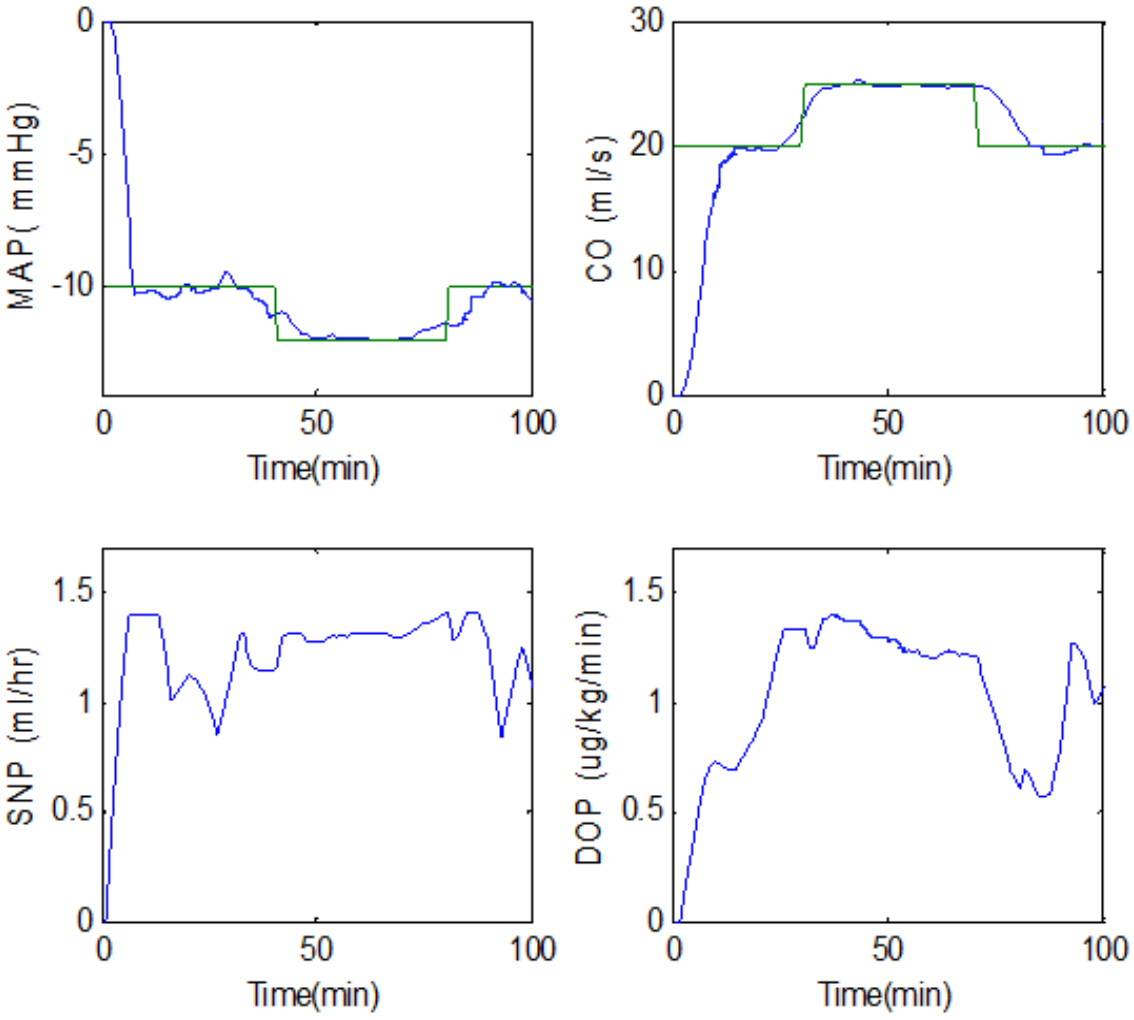


Figure 5.8: System response using the SOFLC-DSL algorithm with the switching mode compensator under 50% variations of *GC* for *CO* and *MAP*.

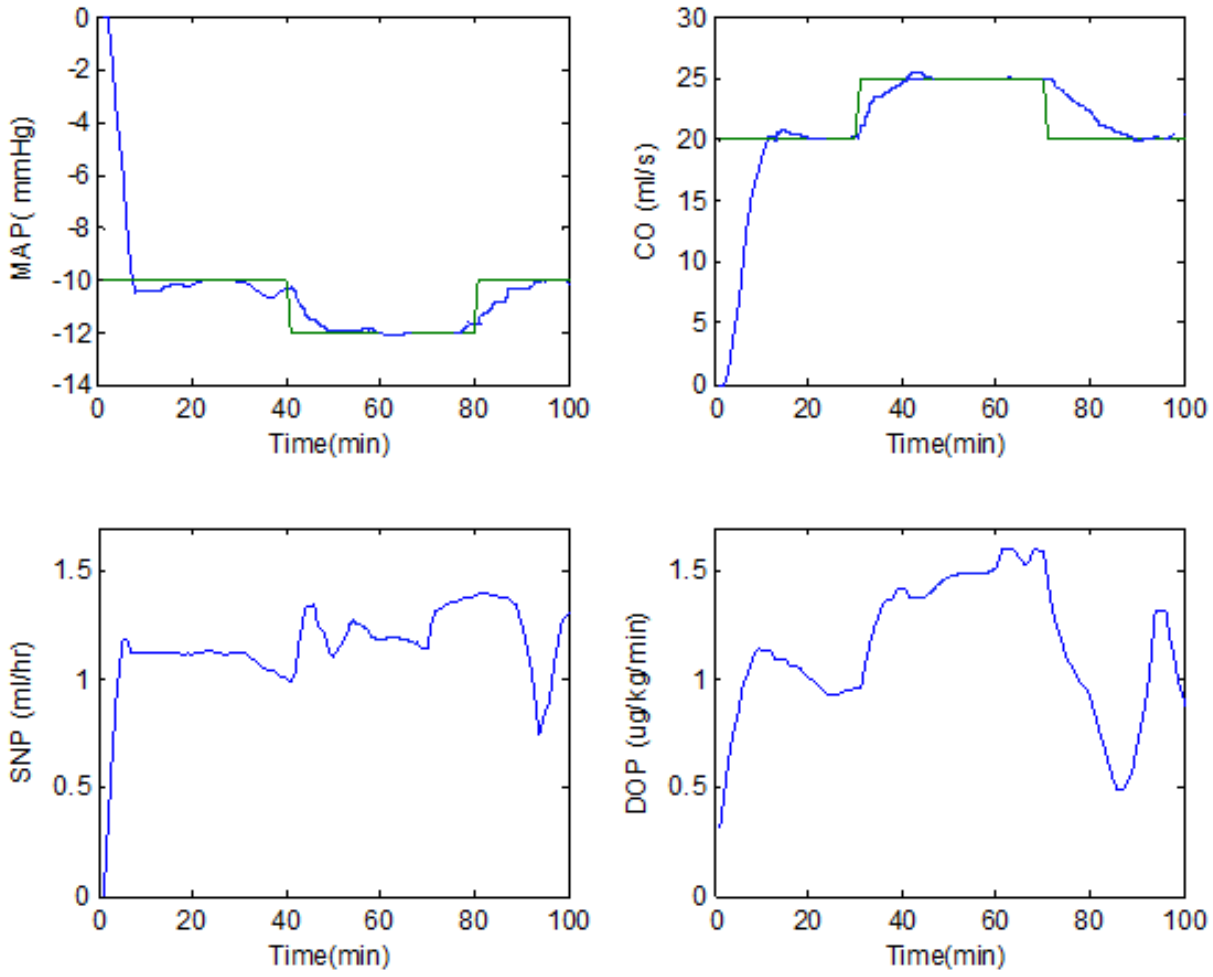


Figure 5.9: System response using the SOFLC-DSL algorithm with the switching mode compensator under 50% variations of *GT* for *CO* and *MAP*.

The highest variations of different scaling factors with respect to the prime values (%).			
	Case 1	Case 2	Case 3
GE/MAP	180	153	90
GC/MAP	210	150	25
GT/MAP	50	20	3
GE/CO	185	87	26
GC/CO	520	310	17
GT/CO	39	27	4

Table 5.1: Variations of scaling factors.

Case 1: SOFLC-DSL algorithm with switching mode linguistic compensator.

Case 2: SOFLC-DSL algorithm with RGA based scalar compensators.

Case 3: SOFLC algorithm with switching mode linguistic compensator.

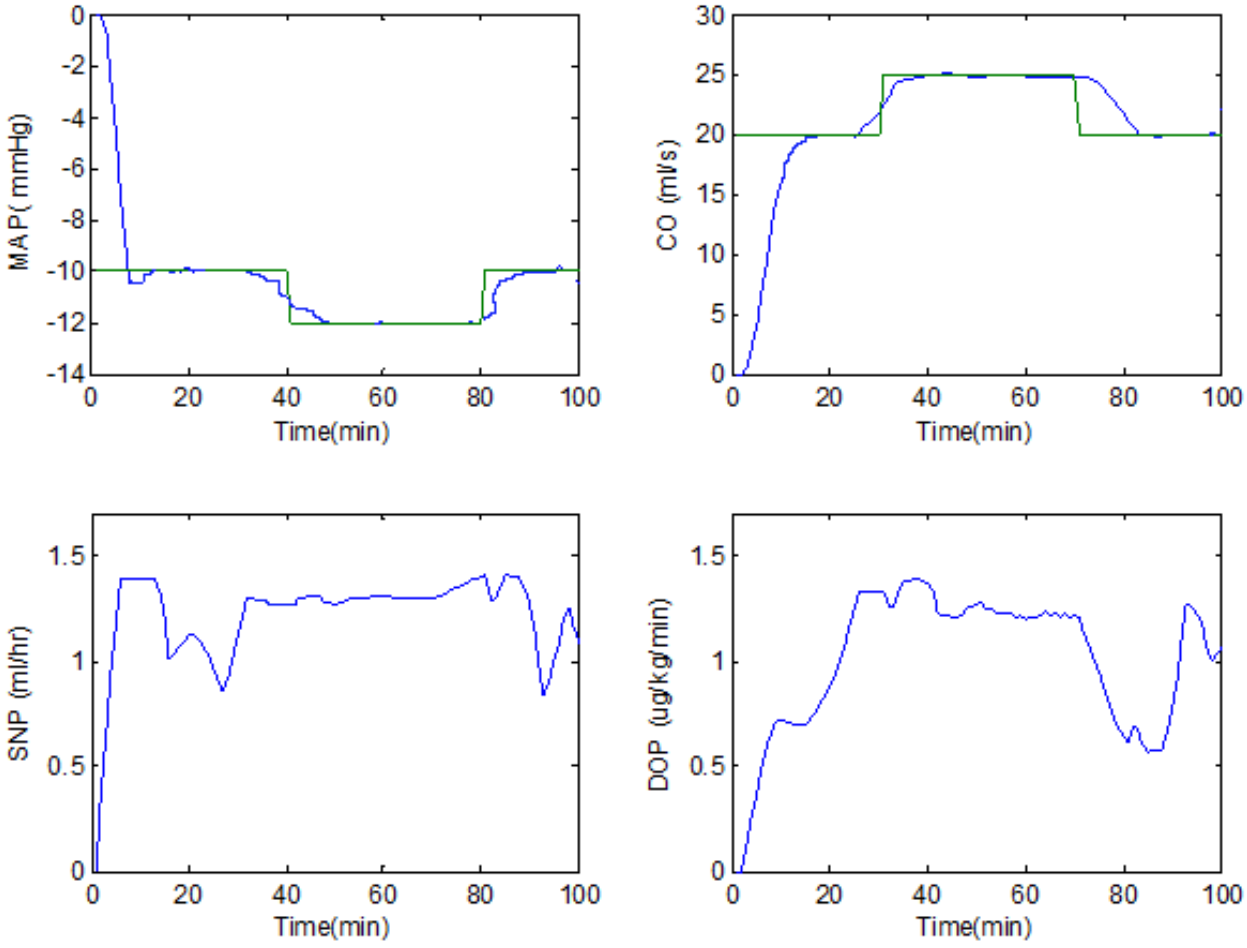


Figure 5.10: System response using the SOFLC-DSL algorithm with the switching mode compensator under 10% variations of system parameters.

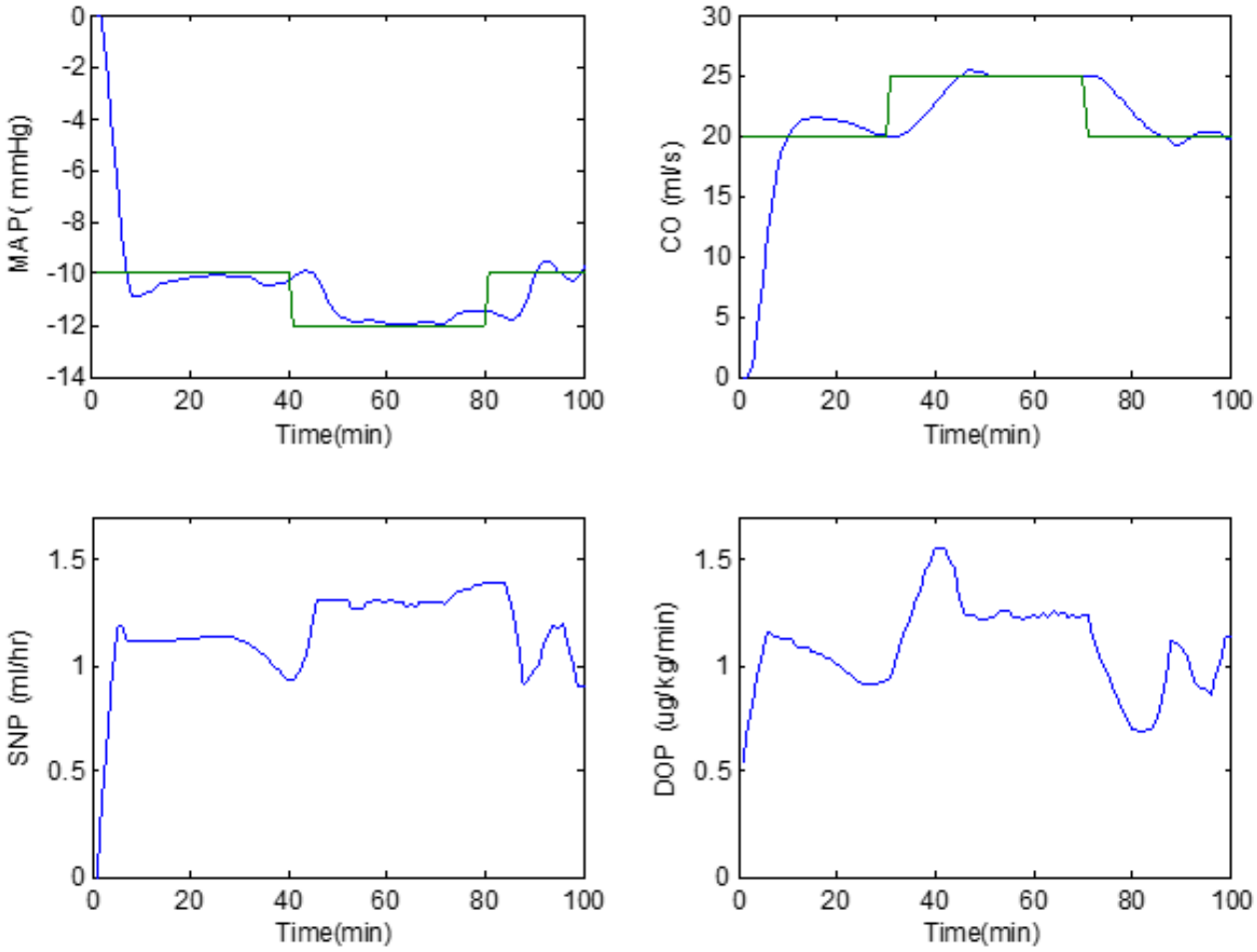


Figure 5.11: System response using the SOFLC-DSL algorithm with RGA based scalar compensator under 10% variations of system parameters.

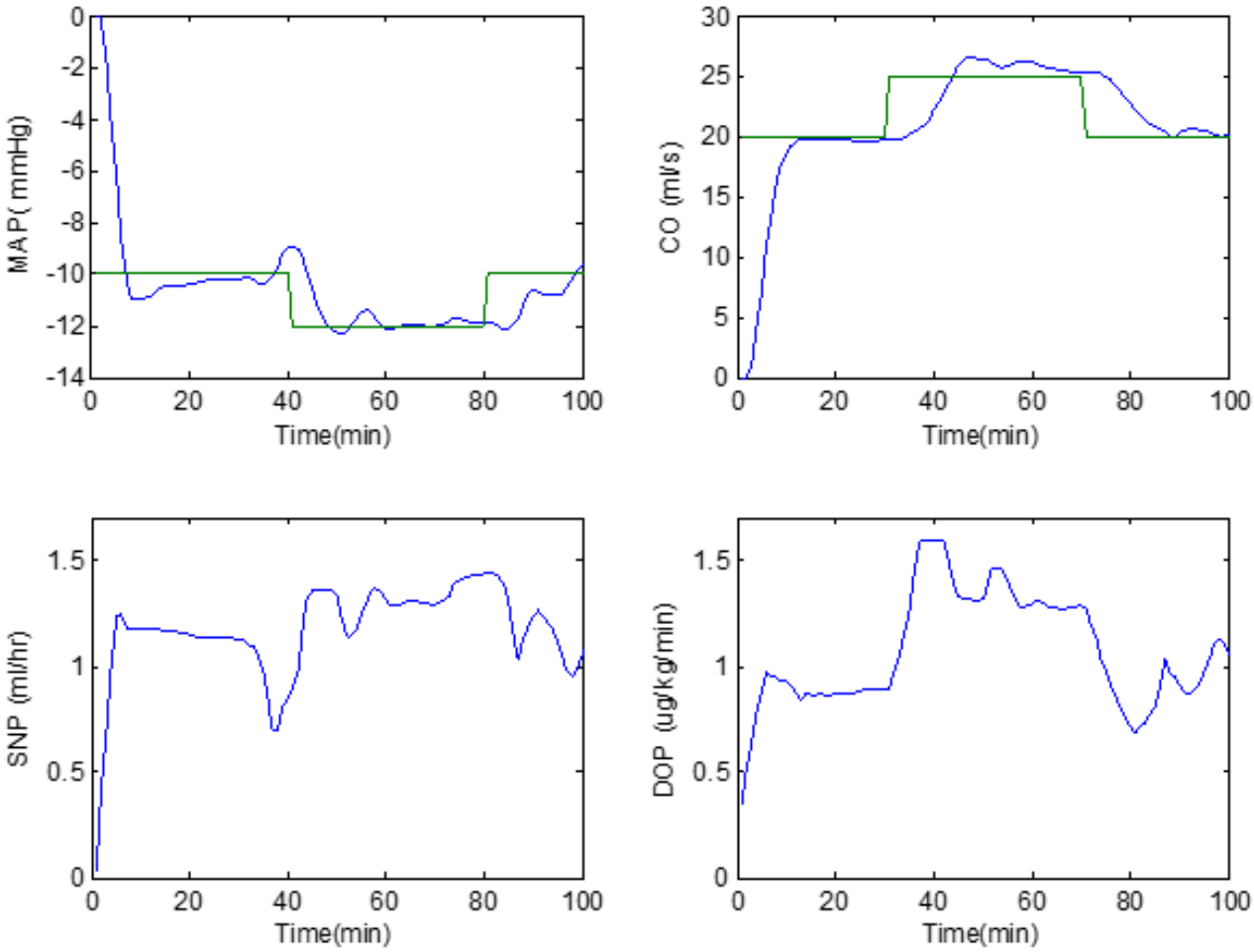


Figure 5.12: System response using the SOFLC algorithm with the switching mode compensator under 10% variations of system parameters.

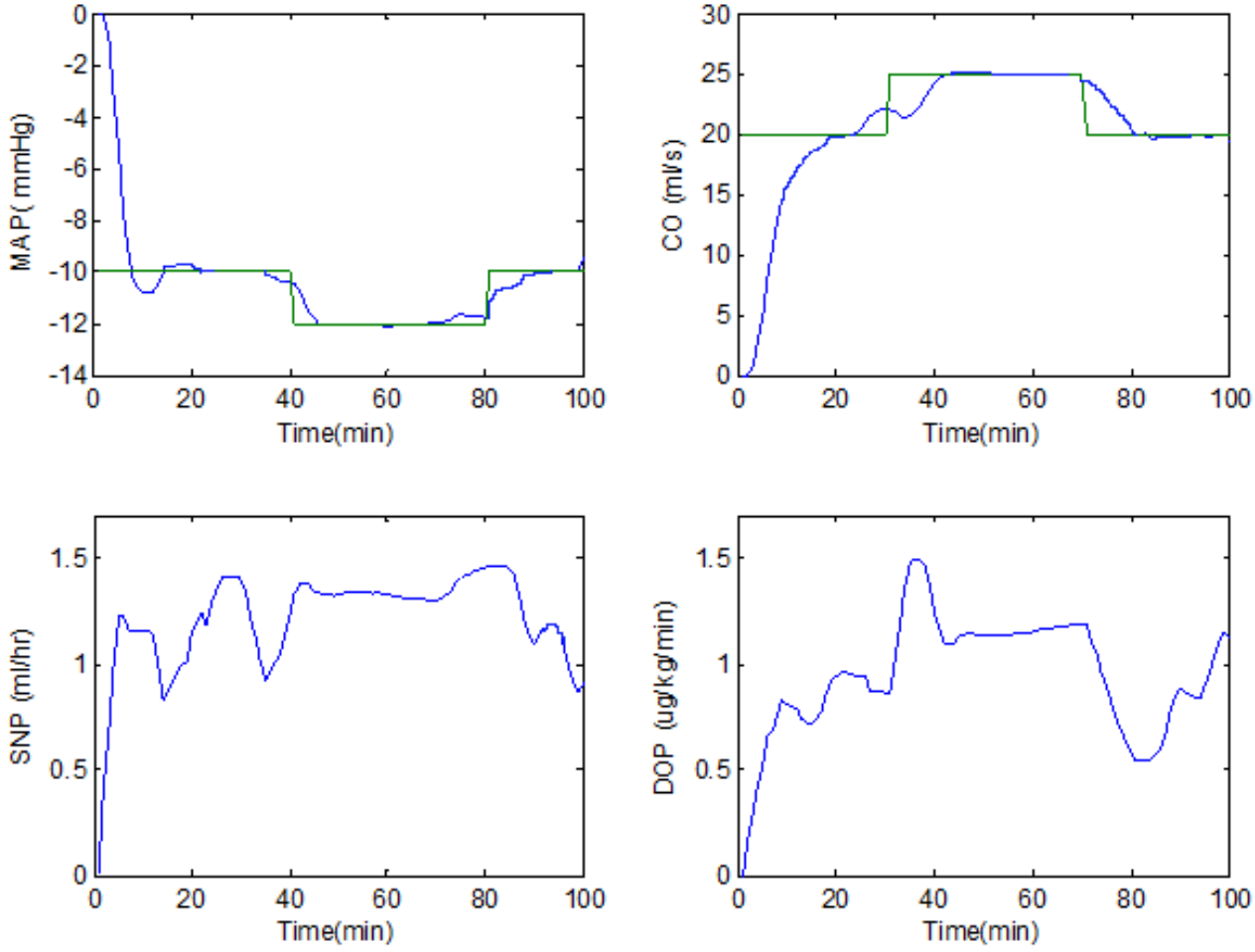


Figure 5.13: System response using the SOFLC-DSL algorithm with the switching mode compensator under 15% variations of system parameters.

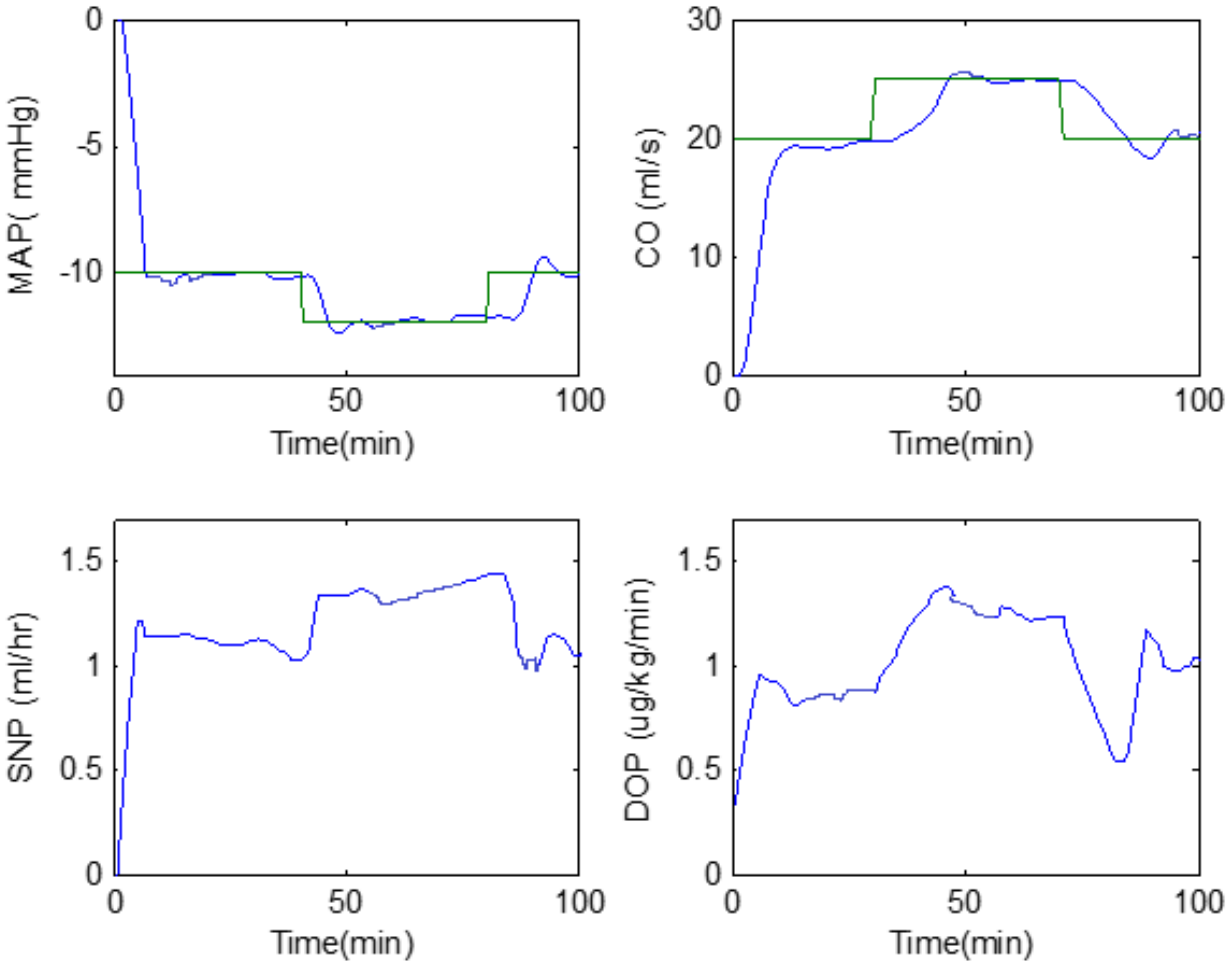


Figure 5.13: System response using the SOFLC-DSL algorithm with RGA based scalar compensator under 15% variations of system parameters.

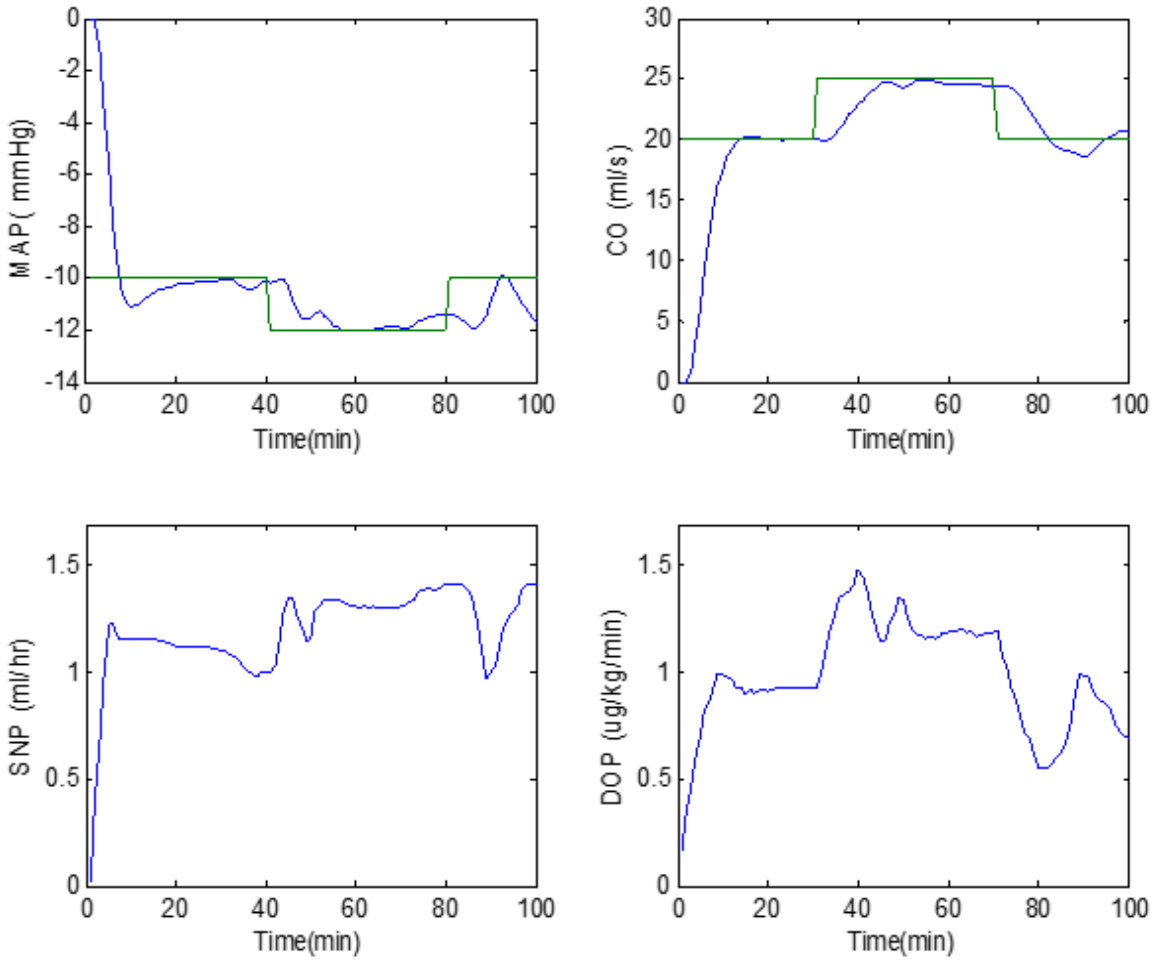


Figure 5.14: System response using the SOFLC algorithm with the switching mode compensator under 15% variations of system parameters.

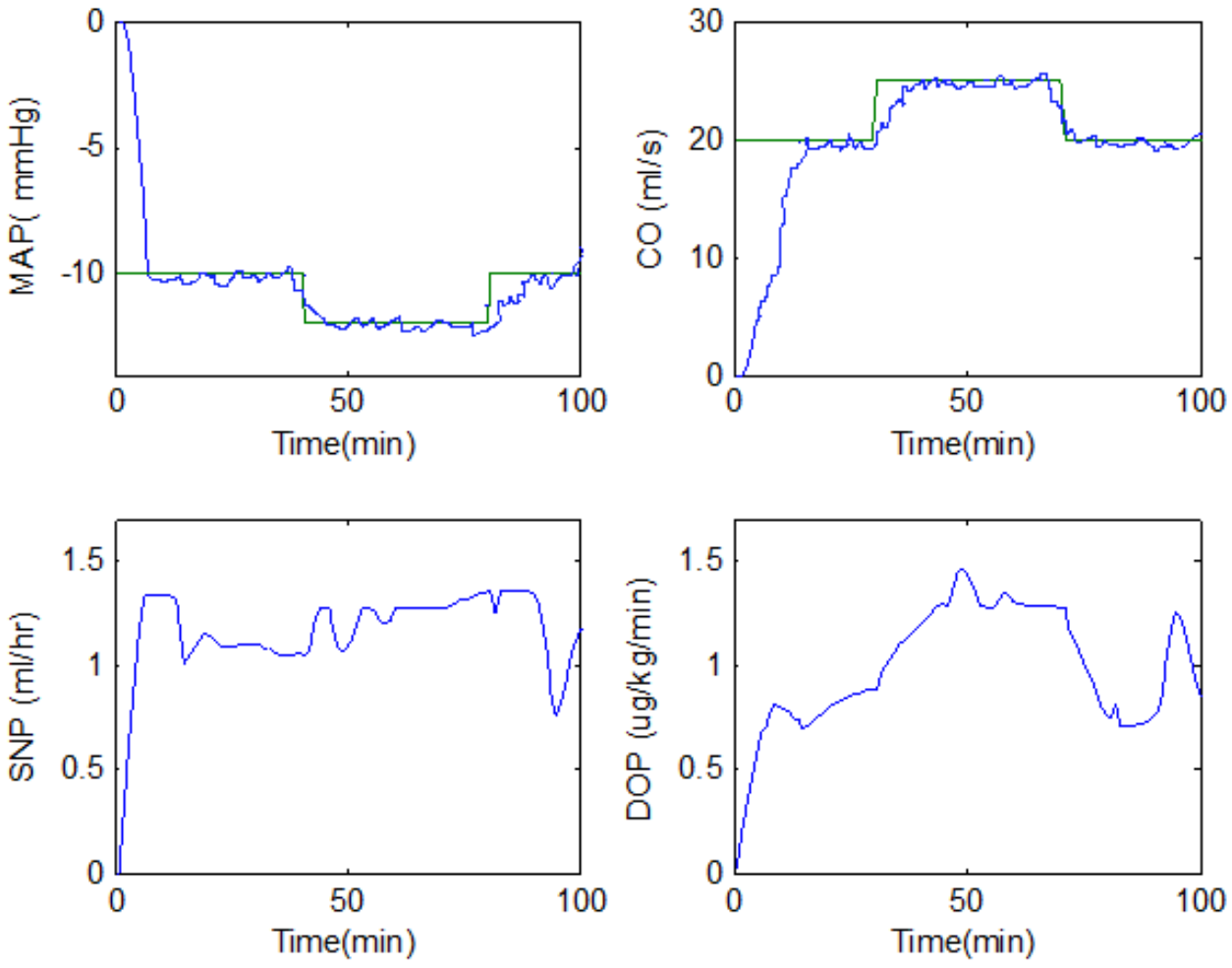


Figure 5.15: System response using the SOFLC algorithm with the switching mode compensator under 5% noise.

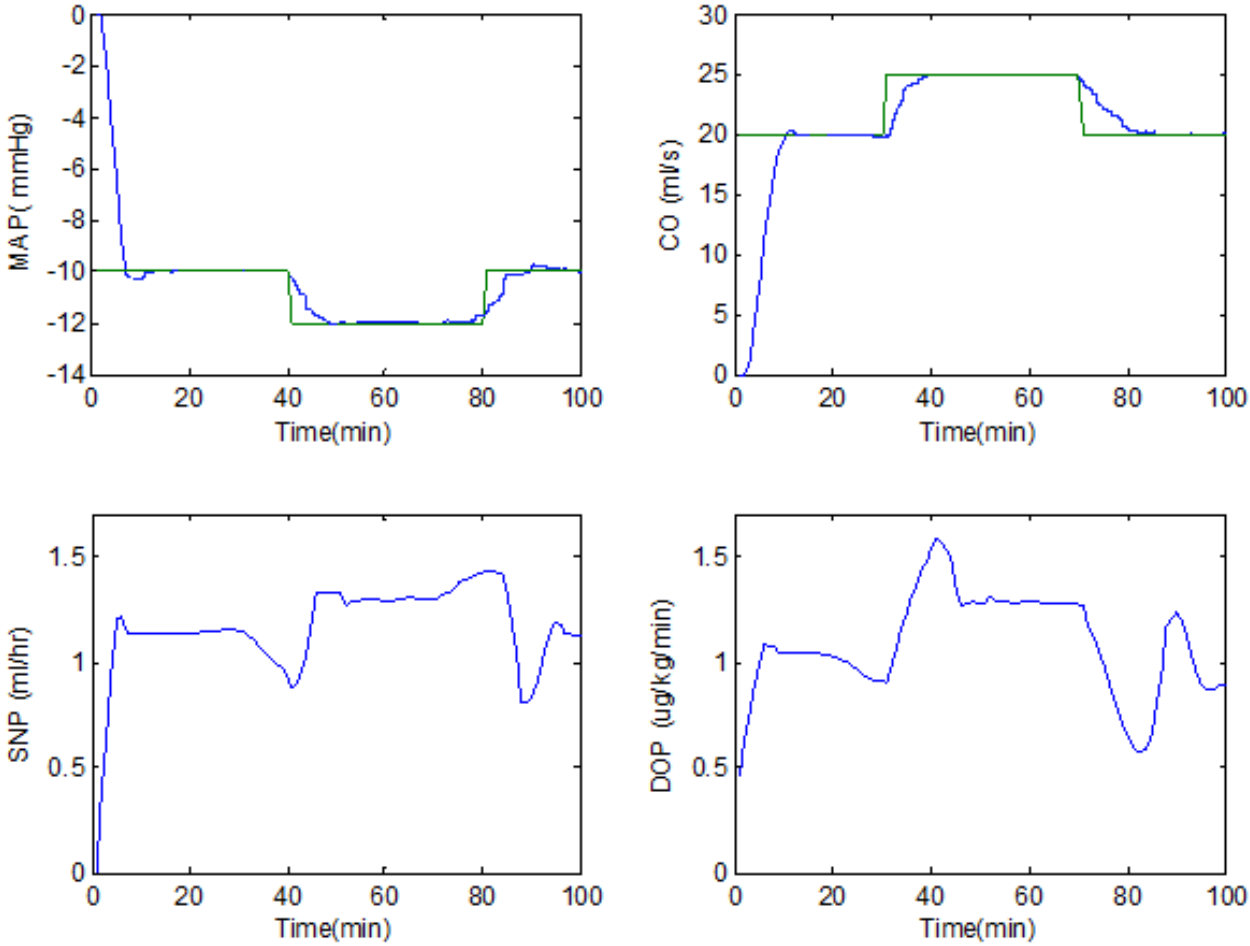


Figure 5.16: Simulation result using $\lambda_{c12} = 0.6 \times \lambda_{c12}'$, $\lambda_{c21} = 1.6 \times \lambda_{c21}'$.

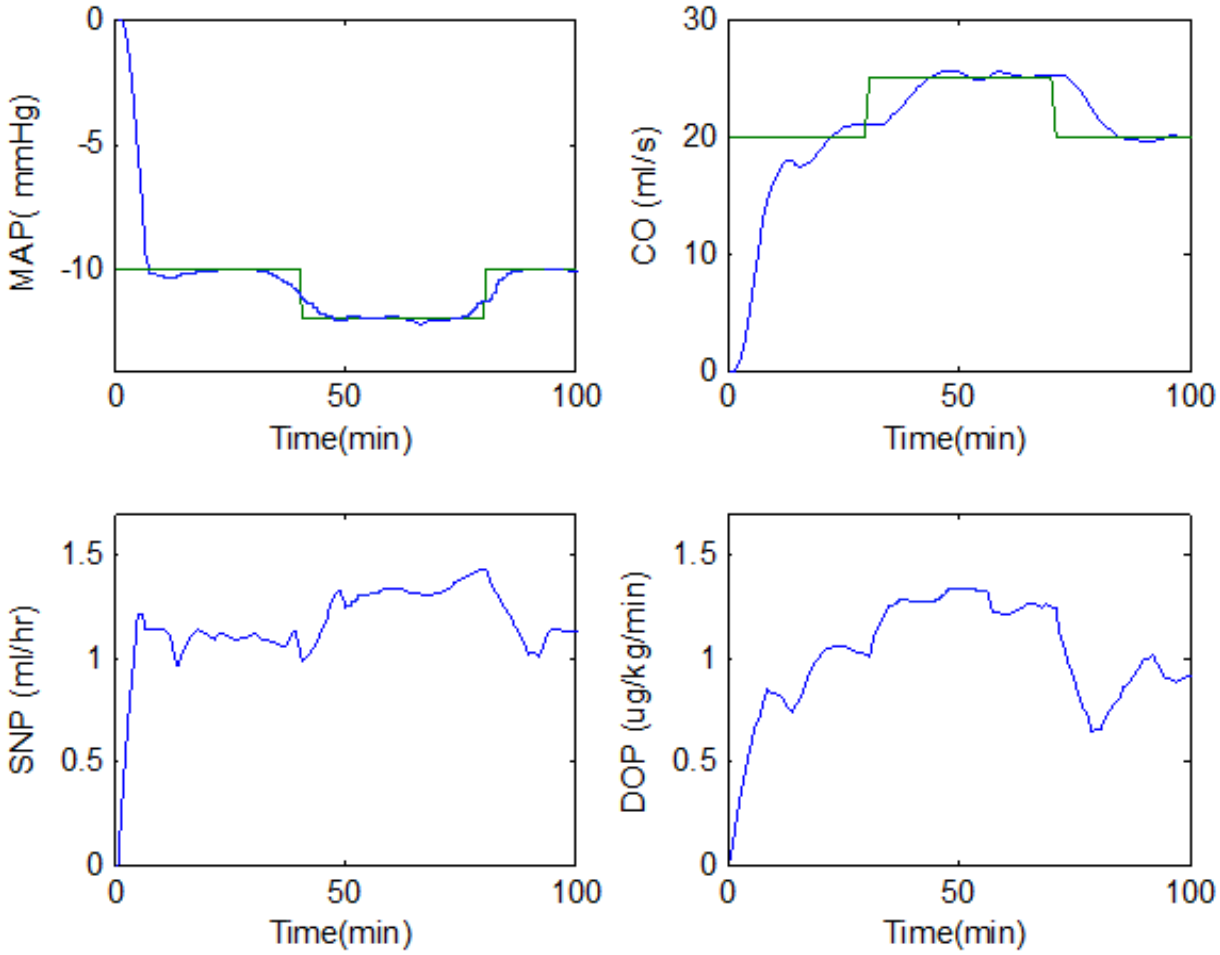


Figure 5.17: Simulation result using $\lambda_{c12} = 1.6 \times \lambda_{c12}'$, $\lambda_{c21} = 0.6 \times \lambda_{c21}'$.

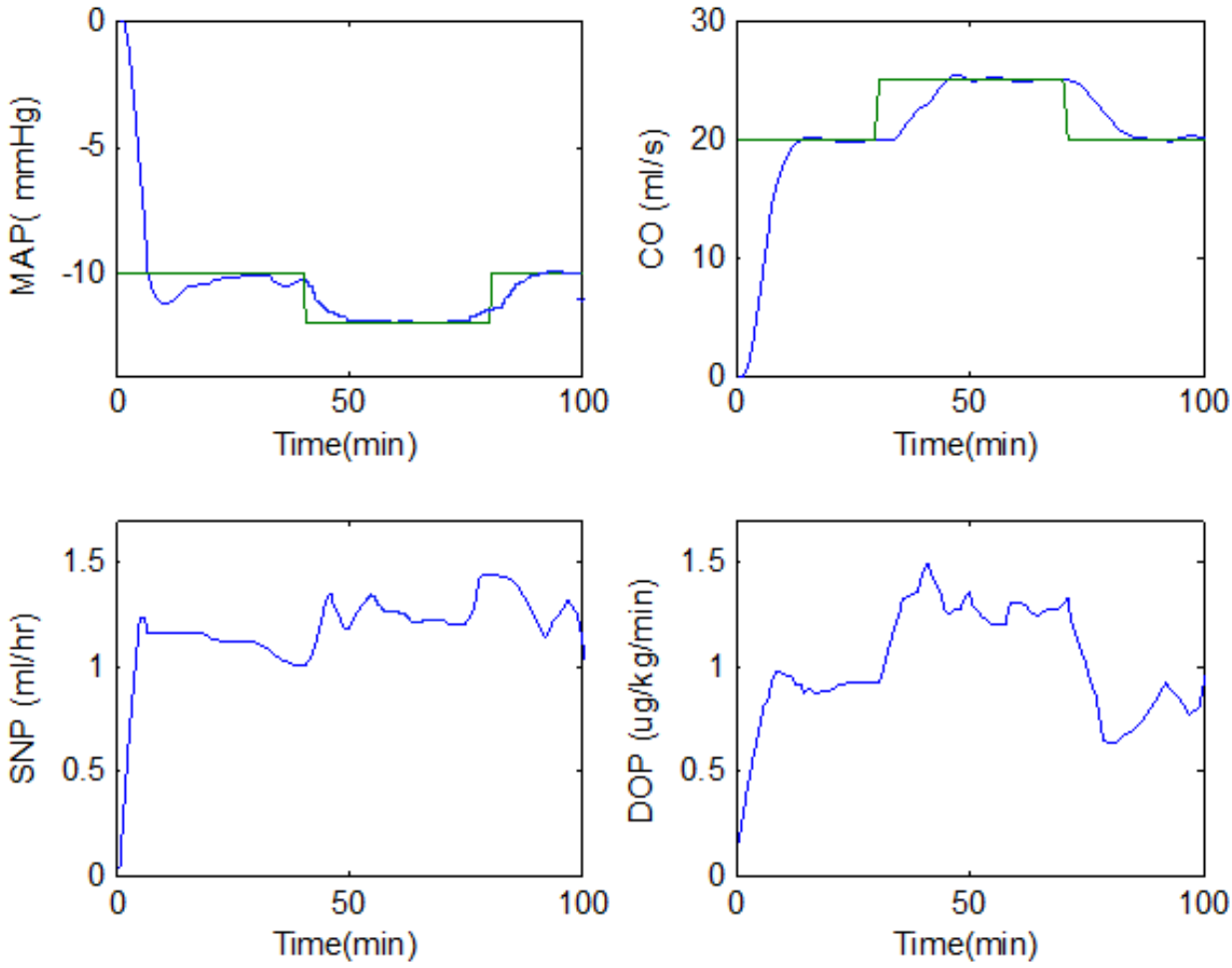


Figure 5.18: Simulation result using $\lambda_{c12} = 0.75 \times \lambda_{c12}'$, $\lambda_{c21} = 0.9 \times \lambda_{c21}'$.

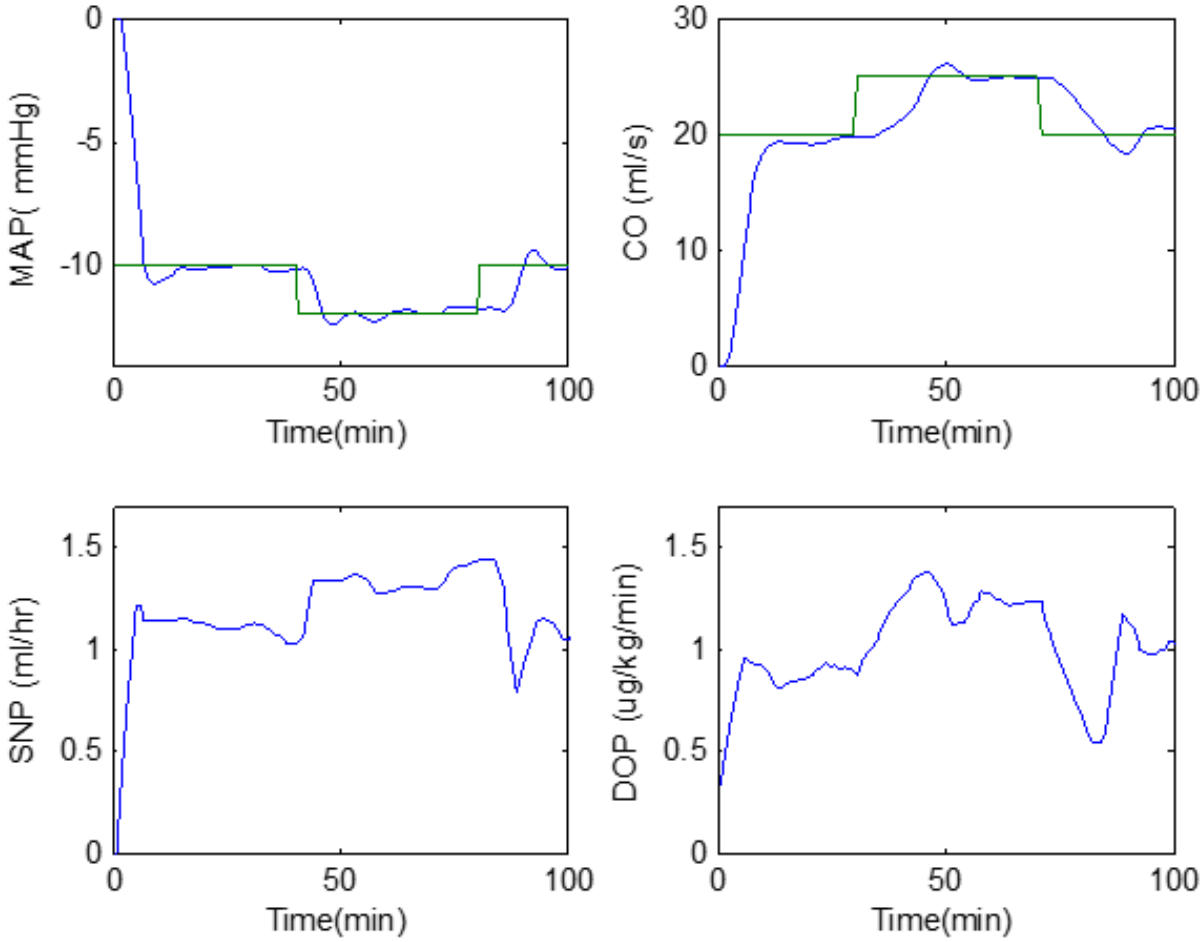


Figure 5.19: Simulation result using $\lambda_{c12} = 0,\ \lambda_{c21} = 0$.

5.5 Summary

The single variable SOFLC-DSL algorithm proposed in Chapter 3 was extended in this chapter to the multivariable case, where forms of adaptation and modification were introduced within the basic algorithm.

The chapter started with a review of multivariable fuzzy logic control systems, with particular emphasis on decoupled systems and how the effects of the input-output interactions can be reduced. Also, a switching mode linguistic compensator based on steady-state gains was described. Finally, the decomposed systems that use the SOFLC-DSL algorithm as a dominant controller to make the output track the set-point signals and the linguistic compensators deal with the interactions within the different channels were applied to a multivariable drug dynamics model.

Similarly, several experiments were carried-out to test the effectiveness of the proposed controller in different environments, including its robustness to process parameter variability, varying system dynamics and noise-contaminated environments. The simulation results revealed that in all experiments, the SOFLC-DSL scheme with the switching mode compensators provided the best performance in terms of fast set-point tracking capabilities in the three regions of the surgical procedures in various conditions and circumstances, where it outperformed both the SOFLC-DSL algorithm with relative gain array-based scalar compensators, and the standard SOFLC scheme that uses a fixed performance index table.

Another interesting observation is that the SOFLC-DSL scheme performed better than the traditional SOFLC algorithm with a fixed index table, even when applied without the switching mode compensators. This is apparently due to the on-line adjustment mechanism that enables it to effectively change its structure, based on the process under control and the environment in which it operates.

It was also observed that the SOFLC-DSL-based system tended to be less sensitive to compensating scalar estimation than any other types of controllers. This is very important, as it essentially means that the controller will depend less on the switching mode linguistic compensator to get rid of the input-output interactions. Furthermore, it was found that

when the switching mode linguistic compensator was used, the system had better resistance to inaccurate estimates of compensating scalars.

The deployment of interval and zSlice type-2 fuzzy sets to the proposed decoupled architecture is considered in Chapter 6.

Chapter 6 – Type-2 Fuzzy Sets for Self-Organising Fuzzy Logic Control of Multivariable Systems with a Dynamic Supervisory Layer

6.1 Introduction

It was observed from the previous chapter that expressing control strategies by relating multi-control rules to multi-situations is not always a trivial task in multivariable control systems. This is mainly due to the reliance of the control performance of these decoupled systems on *a priori* information in the design of suitable compensators. In Chapter 5, the proposed decoupled controller included SOFLC-DSL algorithms and a linguistic switching strategy. This improved the performance of the multivariable system, enabling it to handle the interactions between the different loops and produce a good multivariable control performance in terms of maintaining the desired target points for the various controlled variables. However, the physiological characteristics of the human body can differ from one person to another depending on their health condition, age and gender, making it difficult to design a controller that can work effectively with different patients. This is

mainly due to the effect of the physiological differences on the concentration and duration of the drug required by each patient during surgery.

Furthermore, type-1 fuzzy sets have less degree of freedom than those of type-2, as described in previous chapters, and are therefore less effective in terms of handling real world uncertainty and noise, and in producing accurate and stable control behaviours. In order to deal with this challenge, this chapter shows how both the interval and zSlice type-2 fuzzy sets were tested to enhance the capabilities of the proposed decoupled control of the multivariable system.

The chapter begins with an overview of different applications in which type-2 fuzzy sets are used within multivariable control systems. Another review is then provided on the various PSO-based multivariable systems that are reported in the literature, after which a decoupled architecture of a multivariable system that uses type-2 SOFLC-DSL algorithms as dominating controllers for regulating both the *CO* and *MAP* is introduced. The switching strategy is mainly used to handle the effects of the interactions. However, RGA compensators (see Chapter 4) were also applied in certain simulations.

The performance of the proposed decoupled controller is evaluated by applying it in different environments to test its sensitivity to scaling factors, system dynamics, design structure and additive noise.

6.2 Type-2 Fuzzy Sets for Multivariable Systems

Type-2 fuzzy systems have been heavily used in recent years as part of multivariable systems. Most of these contributions have given promising results, showing that type-2 fuzzy sets are superior to type-1 in fuzzy sets, especially when they are applied in environments similar to those of real-world applications, where signals are associated with high levels of uncertainty and noise.

In the work of Lin (2010), an observer-based indirect adaptive fuzzy controller for non-linear multivariable systems experiencing disturbances was designed. Type-2 fuzzy sets were deployed in order to deal with the training date corrupted by rule uncertainties or noise. An adaptive law, based on the Lyapunov synthesis method, and an output feedback control law were both used to tune the parameters of the adaptive fuzzy scheme. When applied in a simulation environment to control a mass-spring-damper system, the proposed

interval type-2 fuzzy logic system dealt with the noisy training data effectively. In contrast, the adaptive type-1 fuzzy controller required more control effort to handle the uncertainties within the data. Moreover, it was observed that type-2 fuzzy sets enabled the proposed adaptive control scheme to provide better tracking performance than those of type-1 fuzzy sets.

In a similar work, by El-Bardini and El-Nagar (2011), a different direct adaptive interval type-2 fuzzy controller was developed. The proposed scheme was applied to a multivariable anaesthesia system to handle the uncertainty introduced by the varied physiological parameters within the bodies of patients, which differ from one person to another. Simulation results revealed that the proposed scheme provided good robustness when applied to different multivariable anaesthesia models (muscle relaxation and blood pressure). Furthermore, it was observed that the proposed direct adaptive interval type-2 fuzzy controller outperformed the traditional type-1 fuzzy controller for both muscle relaxation and blood pressure in terms of fast and accurate set-point tracking and smoother control efforts.

In a similar work, by Doctor *et al.* (2016), interval and general type-2 SOFLC schemes were developed for the automatic control of anaesthesia during surgical procedures. The biomedical system was modelled using a fuzzy basis function network (FBFN), while a relative gain array of the system was used to handle the effects of interactions. Two orthogonal fuzzy control engines were used to construct the developed fuzzy controller. The control parameters were updated on-line via a horizontal fuzzy control engine in order to tune the control parameters and compensate for all the variations within the unknown system. The horizontal fuzzy control engine for each system input-output pair was provided in the form of a hierarchical structure, whereas the perpendicular fuzzy control engine was introduced to deal with interactions between the input and output channels, and was designed based on the system relative gain array. The new scheme was tested on two simulation problems, where it produced acceptable results, despite the uncertainties of the unknown system and the time-varying variables. Simulation results also showed that the two proposed type-2 controllers were superior to those of type-1 in terms of producing better defined fuzzy rules for set-point tracking and fewer performance errors.

By controlling the concentration of two drugs, Propofol and Remifentanil, Araujo *et al.* (2014) proposed in another recent work a new automatic method for regulating a

Bispectral index (BIS) in the anaesthesia control system. This method involves constructing a model describing both the pharmacokinetics and pharmacodynamics of the patient using real clinical information. Based on this model, an interval type-2 fuzzy PID controller was designed and used to regulate the BIS values. Genetics algorithms were used to optimise the proposed controller. The simulation results demonstrated that the proposed scheme provided the best performance when compared to a standard linear PID controller and a type-1 fuzzy PID controller.

Cervantes and Castillo (2015) developed a control scheme in the form of a hierarchical architecture that combines multiple independent controllers designed to provide accurate and efficient control for complex processes. The hierarchical architecture consists of an individual fuzzy system and a superior controller responsible for adjusting the global performance. Type-2 fuzzy sets were used to implement this controller, which was applied to an airplane flight system. In order to increase the complexity of this system, disturbances were introduced and noise was increased using a joystick. Simulation results proved that the proposed algorithm effectively controlled the complex system and demonstrated its superiority over a traditional fuzzy controller.

In order to control load-frequencies in non-linear power systems, Baydokhty *et al*. (2015) developed a hierarchical type-2 fuzzy controller. This controller was used to overcome the complex dynamics of the process as well as suppress external disturbances with a minimum number of generated fuzzy rules to keep the computational burden low. The parameters of the controller were optimised by the Cuckoo Optimisation Algorithm (COA), which further enhanced its capabilities. The proposed algorithm produced promising results when applied to a two-area power system. It outperformed other traditional controllers, and succeeded in effectively handling the existing uncertainties within the process, as well as improving the overall performance of the system.

In another work, by Li *et al*. (2009), a type-2 fuzzy logic controller based on the Single-Input-Rule-Modules (SIRMs) for controlling non-linear multivariable systems was proposed. In order to tune the parameters and enhance the performance of the proposed algorithm, genetic algorithms were used. The controller was applied to a translational oscillation with a rational proof-mass actuator (TORA). Results revealed the effectiveness of the new algorithm in different environments, including normal, disturbance existing and parameter varying circumstances. Furthermore, it was also observed from the simulations

that the difficulty of designing a fuzzy controller for the complex MIMO TORA system was widely alleviated by incorporating the SIRM, and that the SIMRs-based fuzzy controller was easier to design and understand.

With the assistance of a hierarchical, fair, competition-based genetic algorithm (HFCGA), W.-D.Kim *et al.* (2010) developed a type-2 fuzzy cascade controller for the regulation of a ball and beam system. The cascade controller was formed by outer and inner controllers. The HFCGA, which represents a parallel genetic algorithm, was used to tune the parameters of the controller, and it was selected because of its ability to avoid premature convergence. In a comparative study with a traditional PD cascade controller optimised by a serial genetic algorithm, the proposed scheme produced promising results and outperformed the latter scheme in terms of its effectiveness.

Martinez *et al.* (2009) proposed a type-2 fuzzy controller to regulate a mathematical model representing a unicycle mobile robot set to move around a certain desired path. GA was used to find the optimal values for the parameters of the fuzzy controller, and also to optimise the constants of the trajectory tracking. Stabilising control laws were also designed for the system using a backstopping approach. Both the kinematics and the dynamics of the autonomous mobile robot were taken into account when designing the controller. Computer simulations showed that the proposed scheme provided good tracking performance in relation to different navigation problems.

Liu (2013) used three different types of multivariable SOFLC schemes – type-1, interval type-2 and zSlice general type-2 – as a basis for the regulation of anaesthesia. Rule-extraction methods were used, based on the importance of fuzzy control rules in terms of how many times they were repeatedly generated by the controller. These extracted rules were then applied to drive the type-2 SOFLC (interval and zSlice) to control muscle relaxation. Computer simulations demonstrated that the adopted rule extraction method equipped with the deployment of interval type-2 and zSlice general type-2 sets enabled the proposed multivariable SOFLC scheme to produce better results than those of multivariable type-1 SOFLC schemes.

6.3 Particle Swarm Optimisation of Multivariable Fuzzy Systems

Generally speaking, the output of fuzzy systems is highly non-linear, and this makes using traditional linear optimisation methods to tune such systems a challenging task. Particle swarm optimisation (PSO) has shown itself to be a very powerful optimisation method. It has the capacity to search for solutions within larger solution spaces than any other optimisation method. It therefore represents a better tool for optimising fuzzy logic controllers. The performance of multivariable fuzzy logic schemes depends on their fuzzy control rules and membership functions, and the compensators needed for reducing the effects of interactions. Hence, adjusting these parameters according to the process to be controlled is vitally important.

There have been various contributions, reported in the literature, of PSO algorithms optimising multivariable fuzzy controllers. These works have shown that PSO algorithms have led to relative successes in a wide range of applications when used within fuzzy control architectures.

Shayeghi *et al.* (2008) proposed a particle-swarm-optimisation-based, multi-stage fuzzy (PSOMSF) controller as a solution to the load frequency control (LFC) problem in power systems. In this scheme, the multi-stage fuzzy controller had two rule-bases, and the control law was tuned on-line. In order to achieve the desired level of robust performance with reasonably low control effort, the PSO algorithm was used to accurately generate and tune the membership functions. This structure combined the advantages of the PSO with the flexibility of a fuzzy controller to build a controller with a simple structure and robust performance. When tested on three-area restructured power systems, in three completely different environments, the PSOMSF algorithm produced satisfactory results, outperforming another generic algorithm-based, multi-stage fuzzy algorithm.

In a different paper, by Hou *et al.* (2011), particle swarm optimisation was incorporated within a fuzzy-gain-scheduling, proportional-integral controller designed for a multi-area, interconnected, automatic-generation control (AGC) system. The PSO algorithm was used to optimise the parameters of the proposed fuzzy controller to achieve a global optimum. A new control law, based on area control error (ACE), was adopted, which took into account both the inadvertent interchange accumulation and time error. It was observed from the computer simulations that the PSO system enhanced the performance of the proposed

controller in terms of providing optimal solutions for all the parameters it optimised. In a comparative study with a conventional PI and a fuzzy logic controller, the proposed algorithm demonstrated its superiority in terms of making the output effectively track the set-point, shortening settling time and reducing overshoot.

In order to tune the parameters of a fuzzy logic controller developed for a Delta robot trajectory control, Lu and Liu (2015) proposed a PSO algorithm with dynamic parameters. These dynamic parameters improved the performance of the PSO algorithm in a global search and enabled it to converge faster. The new PSO algorithm used the time-weighted-squared-error (ITSE) as a fitness function to optimise the parameters of the controller. When applied as a basis for controlling a non-linear DELTA robot trajectory, simulation results showed that the proposed controller exhibited some robustness when compared with a PSO-based PID controller. It was also concluded that using the PSO algorithm to tune the controller was simpler and more efficient than using the traditional trial and error methods.

By combining a dynamic-switching-based fuzzy controller and spectral method, Ren *et al.* (2013) developed control architecture for the control of non-linear, disturbed parameter systems (DPSs). Analytical models of the dynamic-switching-based fuzzy controller were used to design the scaling factor of the controller, and also to assess the stability of the control system. A PSO algorithm was used in this scheme to improve the control performance by effectively tuning the scaling parameters. Furthermore, the Lyapunov stability theory was used in combination with the analytical models to analyse the stability of the proposed control in different environments. The proposed scheme was applied to a non-linear rod catalytic reaction process. Simulation results demonstrated the effectiveness of the controller, and showed that its performance in different environments was better than that of a traditional multivariable fuzzy logic controller.

Ali *et al.* (2015) proposed a new, multi-objective predictive controller architecture designed to control a non-linear multivariable process. This architecture was constructed using a new, constrained multi-objective PSO algorithm to optimise a Sugeno fuzzy modelling system, which was used to predict the performance of the non-linear process. By solving constrained multi-objective optimisation problems, a PSO algorithm was used to tune the controller to enable it to introduce suitable control inputs into the system. The

simulation results that were obtained when the new scheme was applied to the multi-inputs multi-outputs quadruple-tank process proved that it had a promising performance.

In a different contribution, by Cheng (2010), a two-stage learning approach combining a PSO-based fuzzy controller with Q-learning fuzzy inference system (QFIS) algorithm was proposed. As the name implies, the PSO was used to tune the parameters of the fuzzy logic controller according to the process to be controlled. Whereas the QFIS algorithm was utilised as a local optimiser, the gradient descent approach was used in the QFIS algorithm to speed-up the convergence to a global minimum. The developed approach was applied to mobile robots in a pursuit/evasion game. The new controller showed a fast learning capability, and succeeded in using the environment in which it operated to generate control actions that produced desirable results.

6.4 Multivariable Type-2 SOFLC-DSL Algorithm

In this section, both the interval and zSlice type-2 fuzzy sets proposed to replace the type-1 fuzzy sets which were used in the decoupled controller considered in the previous chapter are examined. This introduction of type-2 fuzzy sets is carried-out when the proposed decoupled architecture is used with both the switching mode compensators and RGA compensators.

Figure 6.1 shows the general architecture of a decoupled control system using a predefined RGA matrix. The structure is identical to the one described in Chapter 5, with the expectation that the antecedents and consequents of the different SOFLC algorithms would use either interval or zSlice type-2 fuzzy sets.

Due to its simplicity and efficiency, the enhanced interactive algorithm with the stop condition (EIASC) (Wu and Nie, 2001) is used in all the simulations in this chapter.

6.5 Experiments and Results

Similar to the previous chapter, a two-input two-output drug process was used to investigate the performance of the proposed decoupled control systems. For drug dynamics processes, the main control task is to generate an adequate amount of *SNP* and *DOP* so that both the cardiac output and mean arterial pressure are maintained at a steady level, with

less overshoot or downshoot from the set-point levels. The model of the drug dynamic process is represented here as follows:

$$\begin{bmatrix} \Delta CO \\ \Delta MAP \end{bmatrix} = \begin{bmatrix} 1.0 & -24.76 \\ 0.6636 & 76.38 \end{bmatrix} \begin{bmatrix} \frac{k_{11}e^{-\tau_1 s}}{sT_1+1} & \frac{k_{12}e^{-\tau_2 s}}{sT_1+1} \\ \frac{k_{21}e^{-\tau_2 s}}{sT_2+1} & \frac{k_{22}e^{-\tau_2 s}}{sT_2+1} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \quad (6.1)$$

Here, the same configuration used in the previous chapter is adopted again, giving $k_{11} = 8.44; k_{12} = 5.275;\ k_{21} = -0.09;\ k_{22} = -0.15;\ \tau_1 = 60\text{s};\ \tau_2 = 30\text{s};\ T_1 = 8484.1\text{s};\ T_2 = 58.75\text{s}$.

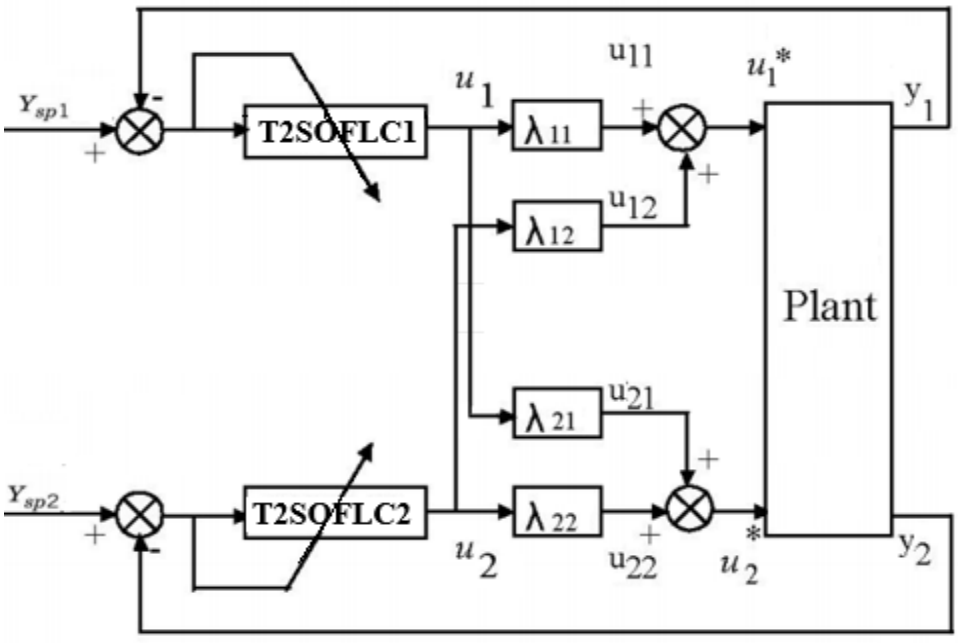


Figure 6.1: Decoupled system using RGA compensators (Lu and Mahfouf, 2006).

Four different SOFLC algorithms were applied as the dominant controllers for each control loop of the multivariable scheme, which are:

- Interval Type-2 SOFLC-DSL algorithm with a switching mode compensator
- zSlice Type-2 SOFLC-DSL algorithm with a switching mode compensator
- Type-1 SOFLC-DSL algorithm with a switching mode compensator
- Type-1 SOFLC algorithm with a switching mode compensator

The system responses under these four algorithms are shown in Figures 6.2 ~6.5. The simulation results prove that the first three controllers successfully maintained both the *CO* and *MAP* at the desired levels in the three stages of the surgical procedure, with short rising time and minimal overshoot. However, poor performance was observed when the type-1 SOFLC algorithm with a switching mode compensator was used, especially in the first and third regions of the surgical procedure. Furthermore, one can also note that the former three controllers produced similar results, even though it is apparent that those which included type-2 fuzzy sets (interval and zSlice) still produced slightly better results in certain regions in terms of having better tracking abilities. Also, as observed in Chapter 4, it remains difficult to differentiate between the controllers that use interval and zSlice type-2 fuzzy sets, as they both produced similarly satisfactory responses.

6.5.1 Robustness of the proposed scheme to varying scaling factors

In this section, the simulation results excluded the type-1 SOFLC algorithm, as it was outperformed by the other three algorithms in all cases. The performance of the other three algorithms on the surgical simulation was assessed based on their ability to maintain the desired set points over the duration of the simulation, with less sensitivity to the configuration of scaling factors. Figures 6.6 ~6.8 provide a comparison of the system performance under 30% variations of *GE*, *GC* and *GT* for both *CO* and *MAP*.

It can be concluded that the type-2 SOFLC-DSL algorithms were able to stabilise quickly, and were successful in making the controlled variables effectively reach the desired set-points in the three regions of the surgical simulation. Moreover, it can be seen that both the interval and zSlice SOFLC-DSL algorithms outperformed the type-1 SOFLC-DSL algorithm, even though it is obvious that the latter controller managed to produce a satisfactory performance. Furthermore, one can observe that the latter scheme required more control effort in terms of producing more fluctuating drug inputs than those of the type-2 controllers.

6.5.2 Robustness of the proposed scheme to varying system dynamics

In this section, the robustness of the system is evaluated by varying the parameters ($k_{11,}k_{12,}k_{21,}k_{22}$) by different percentages with respect to the original model described in

6.1. The corresponding responses of the decoupled multivariable controller are shown in Figures 6.9 ~ 6.13.

Similar to the conclusions observed in the previous section, the SOFLC-DSL algorithms equipped with type-2 fuzzy sets produced better responses than the other schemes. They also succeeded in producing good multivariable control performance in keeping the desired target levels for both *CO* and *MAP* during the three stages of the surgical procedure. Furthermore, it is worth noting that superior performance was achieved, with more stable and smoother drug inputs.

Figure 6.13 shows a comparison of an interval type-2 SOFLC with a switching mode linguistic compensator, and a zSlice type-2 SOFLC with a RGA-based scalar compensator. It can be seen that, even though these controllers were equipped with type-2 fuzzy sets, they still performed poorly in certain regions, especially in the case of the latter scheme. This is clearly due to the fixed performance index tables they both use, which made it hard for these controllers to effectively provide the desired corrective values for the lower-level fuzzy logic rule-base.

6.5.3 Robustness of the proposed scheme to additive noise

The signal strength of the mean arterial pressure (*MAP*) and the cardiac output (*CO*) can be very small when measured. This can be a result of the used high frequency surgical equipment. In order to test the robustness of the proposed algorithms and their ability to produce good performances and maintain the desired set-points over the entire simulation in the stochastic cases, three different scenarios with values of 5%, 9% and 13% white noise were considered.

The simulation results are shown in Figures 5.14 ~ 5.16. One can clearly see how type-2 SOFLC-DSL outperformed the type-1 SOFLC-DSL in terms set-point tracking and control action smoothness. It is obvious that the latter scheme failed, in certain regions, to make both the *CO* and *MAP* track the desired levels, and that it also fluctuated more than the former two algorithms. The type-2 fuzzy sets provided a better mechanism for the SOFLC-DSL algorithms to measure the noise. Also, the dynamic supervisory layer enabled the algorithm to effectively modify the performance index table based on the dynamics of the system under control, and to relate the modifications introduced by the performance index to the changes of the output of the controllers.

It is still very difficult to use these simulations to compare the performances of the interval and zSlice type-2 SOFLC-DSL algorithms, as they both provided satisfactory results.

6.5.4 Robustness to the compensator design

As stated in the previous chapter, the switching mode linguistic strategy depends on the steady-state gains estimation. It is therefore vitally important to evaluate the effectiveness and robustness of such a strategy even with an inaccurate estimation of steady-state gains. In this case, another four different scenarios of estimation were considered.

- Scenario 1: $\lambda_{c12} = 0.8 \times \lambda_{c12}{}', \ \lambda_{c21} = 1.8 \times \lambda_{c21}{}'$
- Scenario 2: $\lambda_{c12} = 1.8 \times \lambda_{c12}{}', \ \lambda_{c21} = 0.8 \times \lambda_{c21}{}'$
- Scenario 3: $\lambda_{c12} = 0.95 \times \lambda_{c12}{}', \ \lambda_{c21} = 1.1 \times \lambda_{c21}{}'$
- Scenario 4: $\lambda_{c12} = 0, \ \lambda_{c21} = 0$

The simulation results shown in Figures 5.17 ~ 5.20 reveal that the proposed decoupled controllers still provided acceptable performance in the four scenarios. This demonstrates that the strategy still coped effectively even when the steady-state gain estimation was inaccurate.

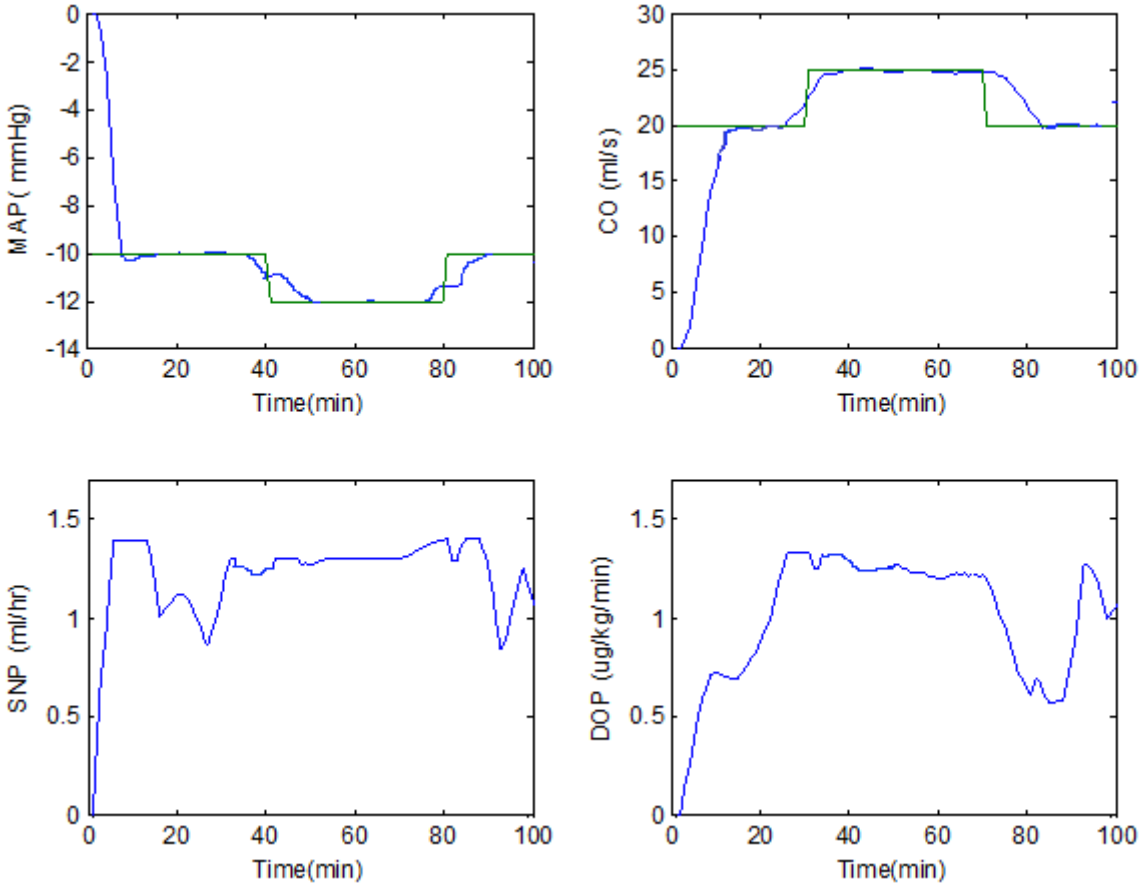


Figure 6.2: System response using the interval type-2 SOFLC-DSL algorithm with a switching mode compensator.

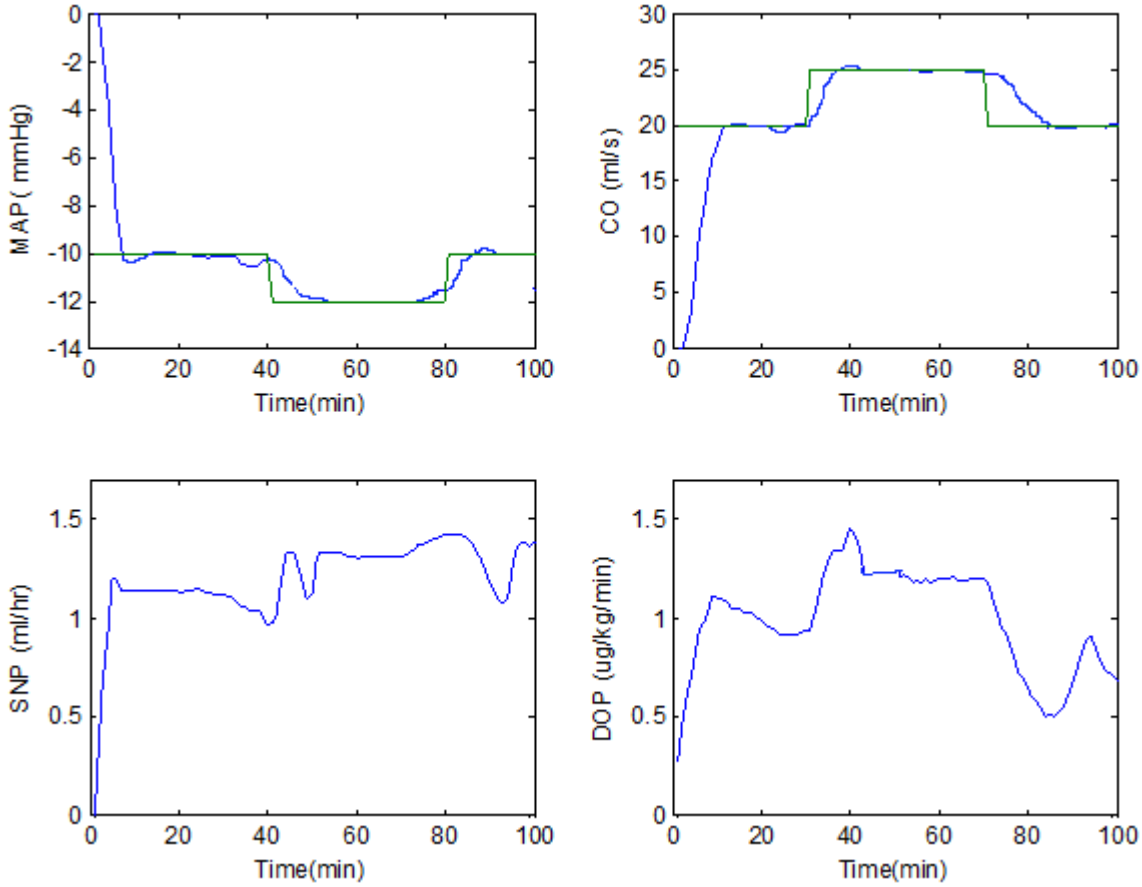


Figure 6.3: System response using the zSlice Type-2 SOFLC-DSL algorithm with a switching mode compensator.

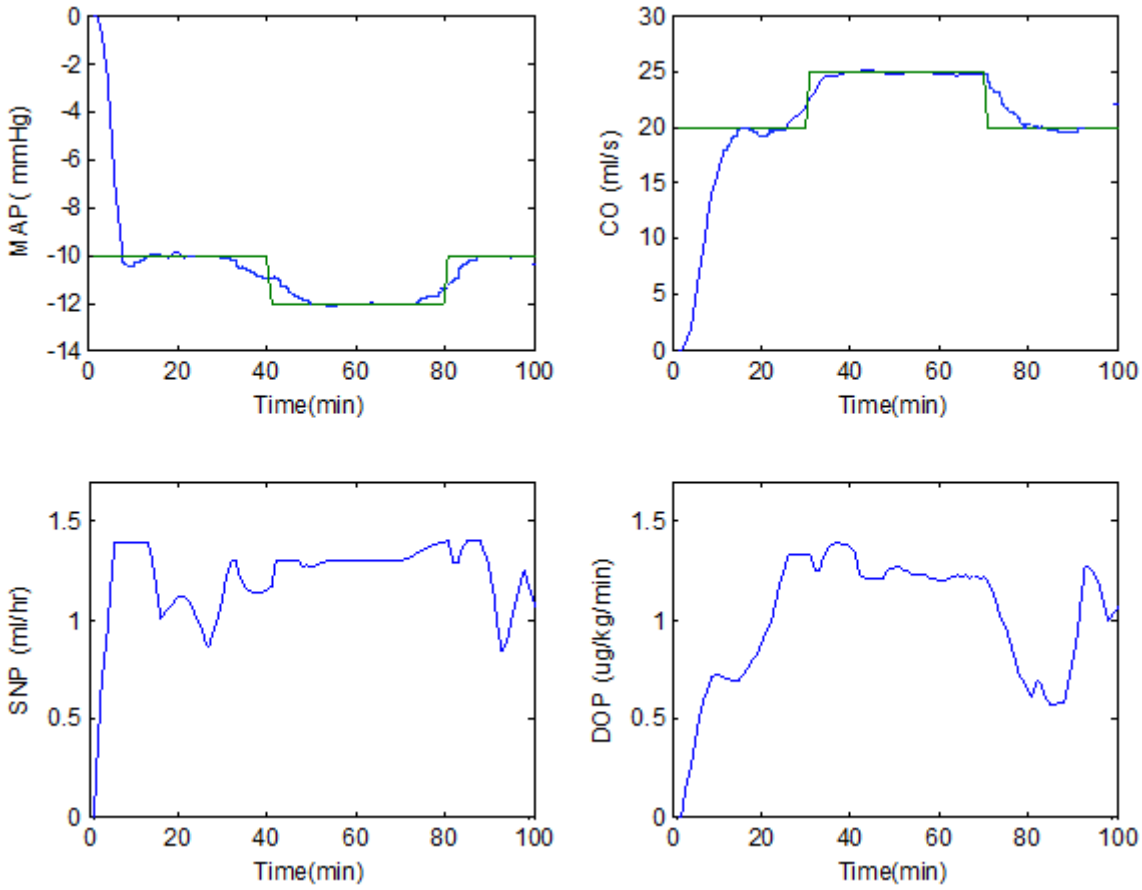


Figure 6.4: System response using type-1 SOFLC-DSL algorithm with a switching mode compensator.

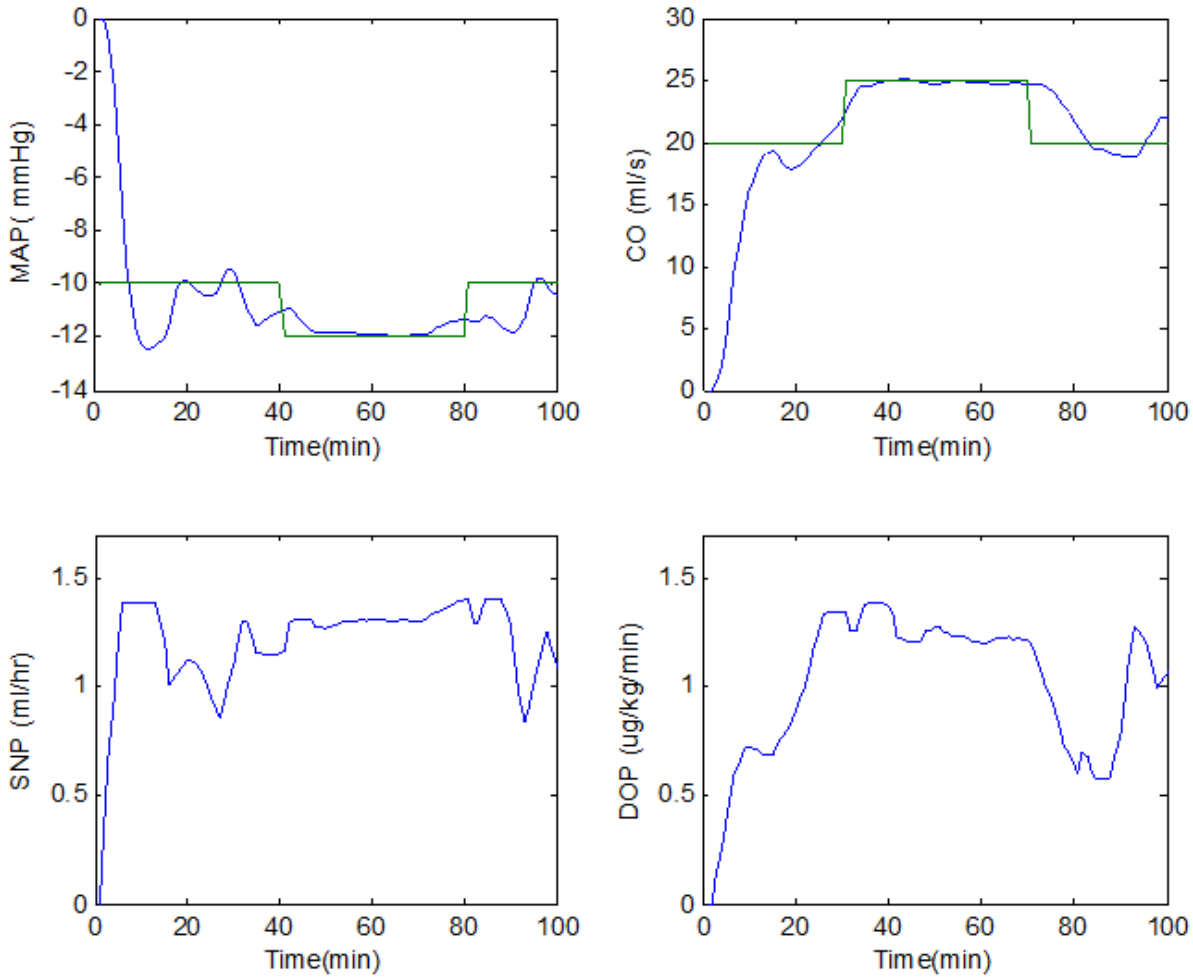


Figure 6.5: System response using the Type-1 SOFLC algorithm with a switching mode compensator.

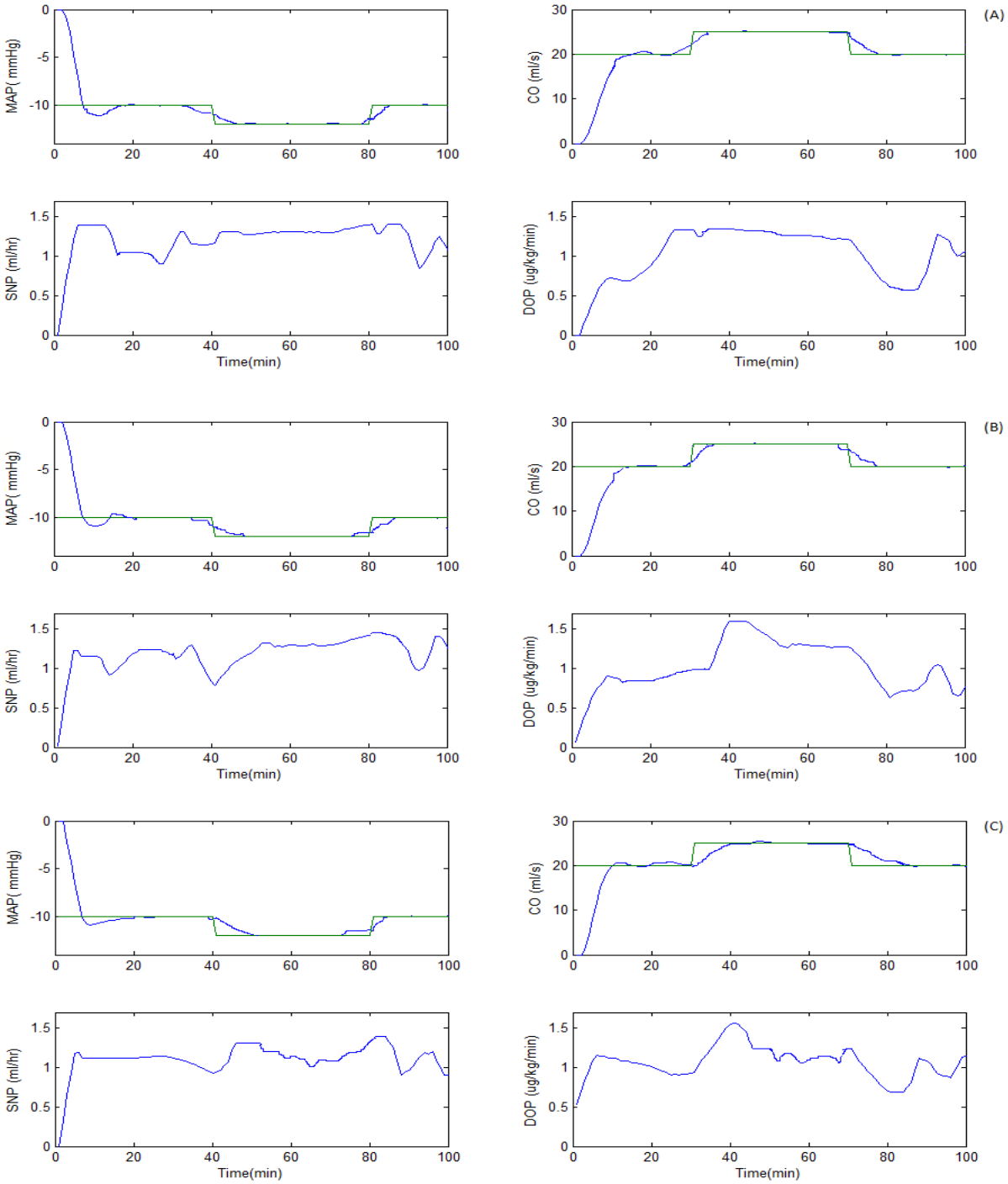


Figure 6.6: System response using various SOFLC algorithms under 30% variations of *GE* for *CO* and *MAP*.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): Interval type-2 SOFLC-DSL with switching mode linguistic compensator
(C): Type-1 SOFLC-DSL with switching mode linguistic compensator

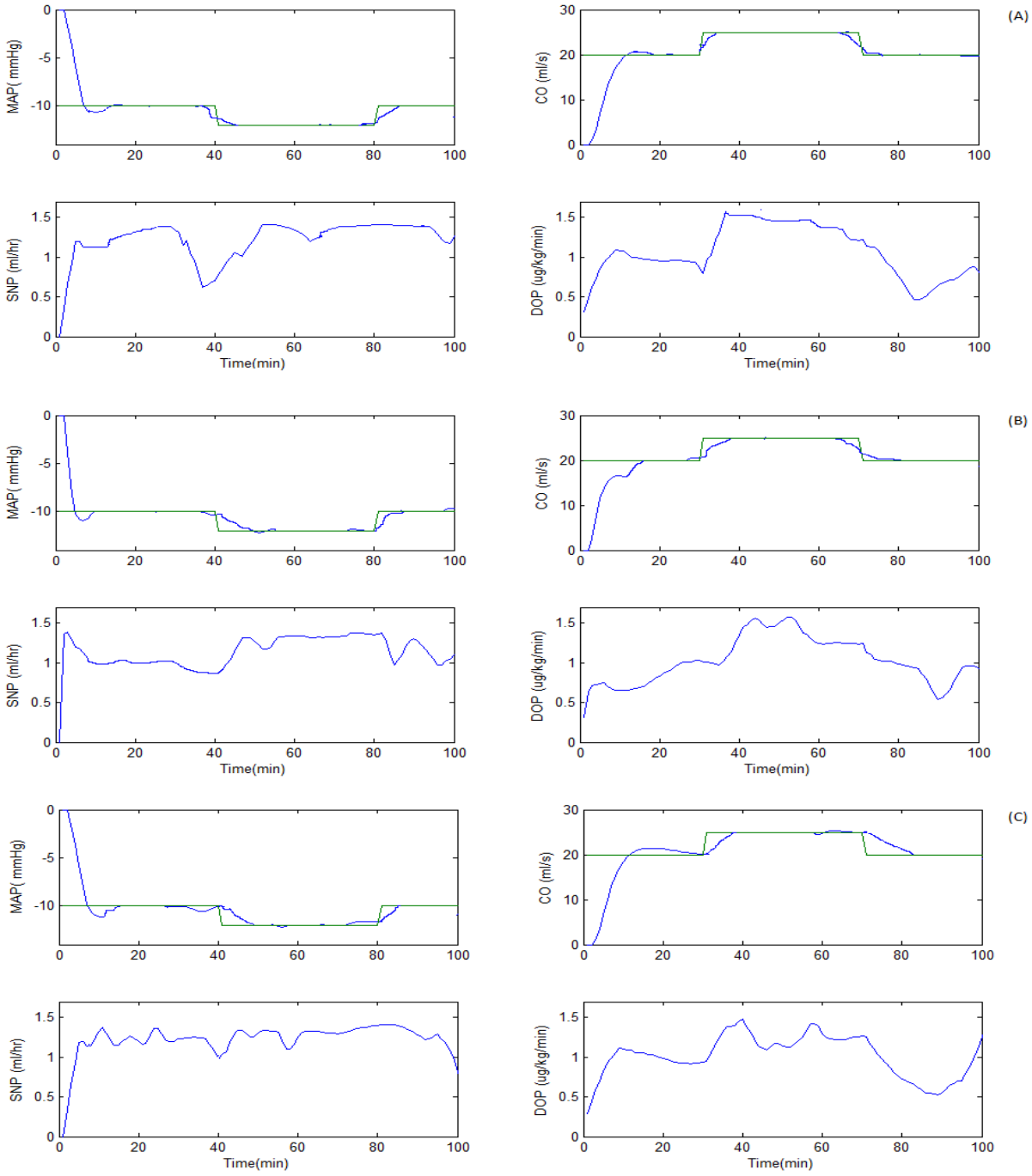


Figure 6.7: System response using various SOFLC algorithms under 30% variations of *GC* for *CO* and *MAP*.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): Interval type-2 SOFLC-DSL with switching mode linguistic compensator
(C): Type-1 SOFLC-DSL with switching mode linguistic compensator

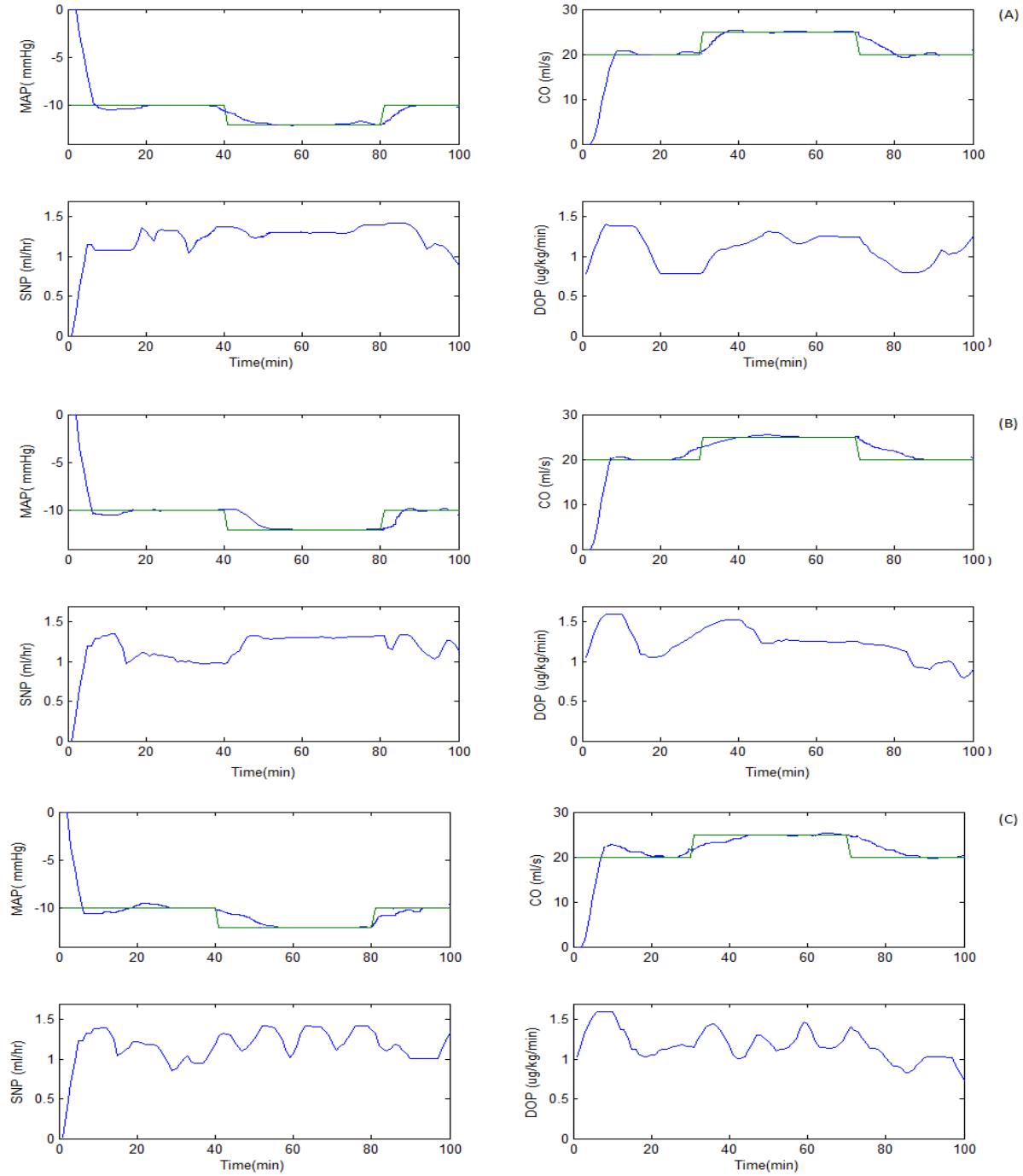


Figure 6.8: System response using various SOFLC algorithms under 30% variations of *GT* for *CO* and *MAP*.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): Interval type-2 SOFLC-DSL with switching mode linguistic compensator
(C): Type-1 SOFLC-DSL with switching mode linguistic compensator

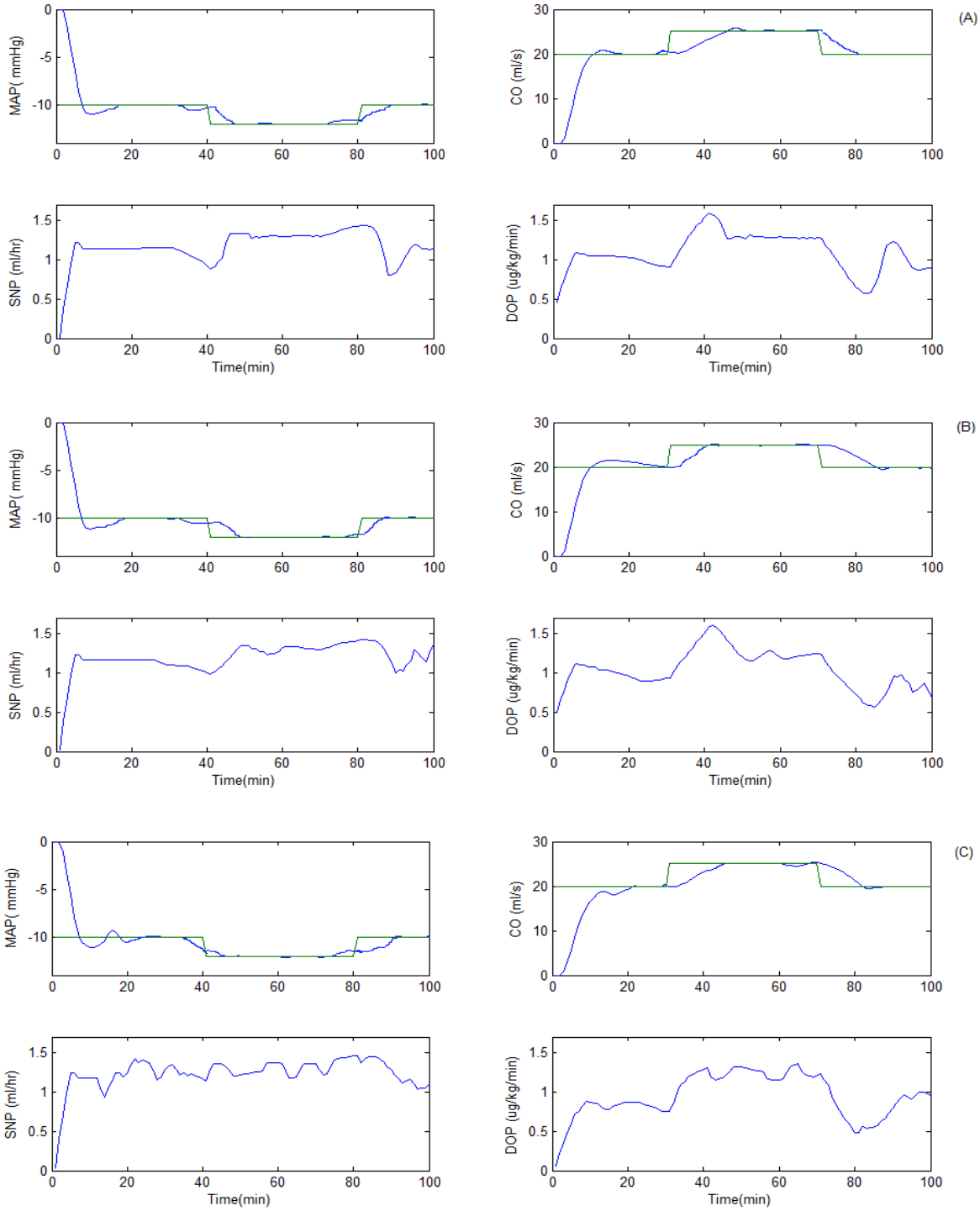


Figure 6.9: System response using various SOFLC algorithms under 4% variations of system parameters.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): Interval type-2 SOFLC-DSL with switching mode linguistic compensator
(C): Type-1 SOFLC-DSL with switching mode linguistic compensator

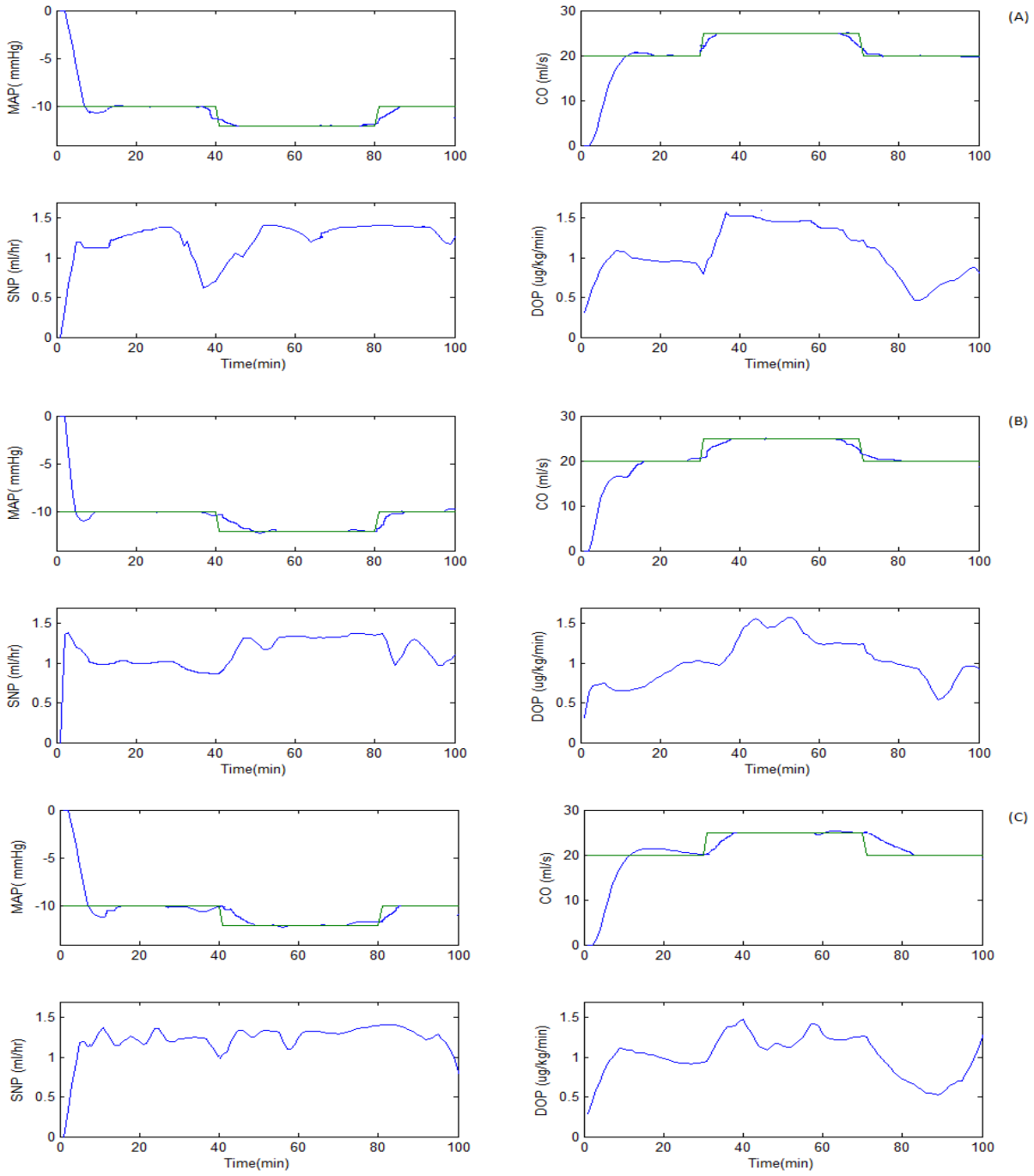
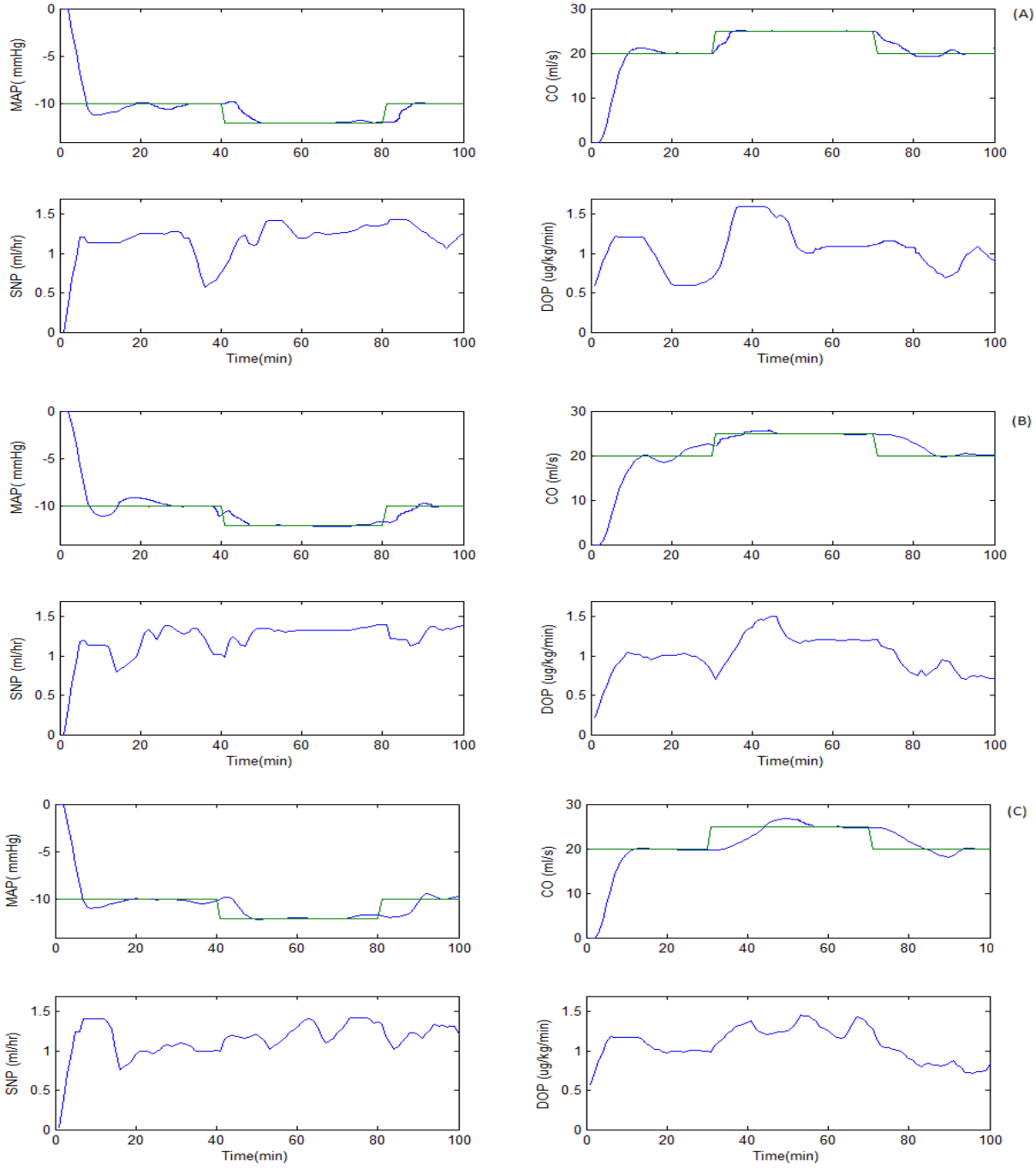


Figure 6.10: System response using various SOFLC algorithms under 7% variations of system parameters.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): Interval type-2 SOFLC-DSL with switching mode linguistic compensator
(C): Type-1 SOFLC-DSL with switching mode linguistic compensator



6.11: System response using various SOFLC algorithms under 12% variations of system parameters.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): Interval type-2 SOFLC-DSL with switching mode linguistic compensator
(C): Type-1 SOFLC-DSL with switching mode linguistic compensator

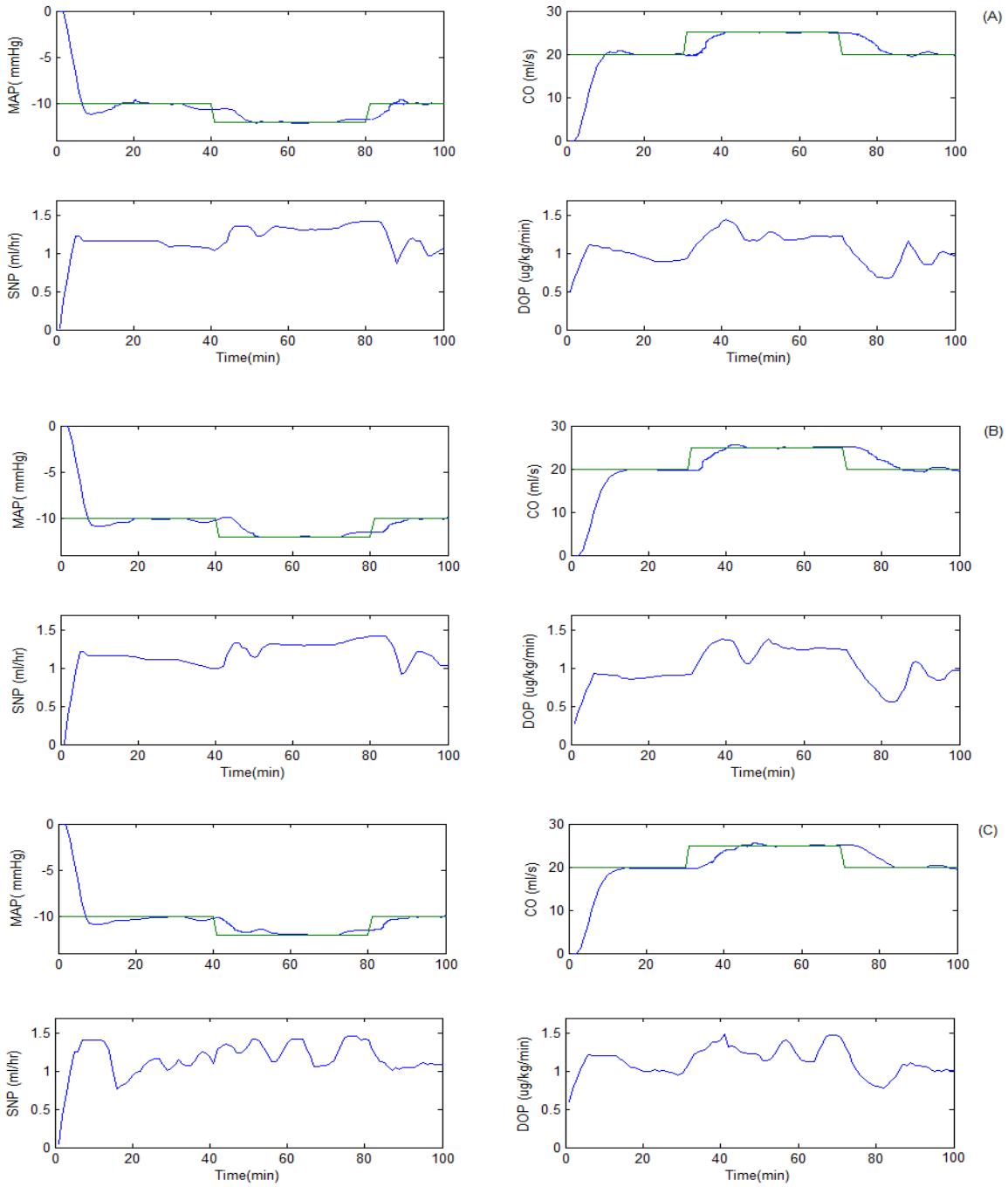


Figure 6.12: System response using various SOFLC algorithms under 15% variations of system parameters.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): Interval type-2 SOFLC-DSL with switching mode linguistic compensator
(C): Type-1 SOFLC-DSL with switching mode linguistic compensator

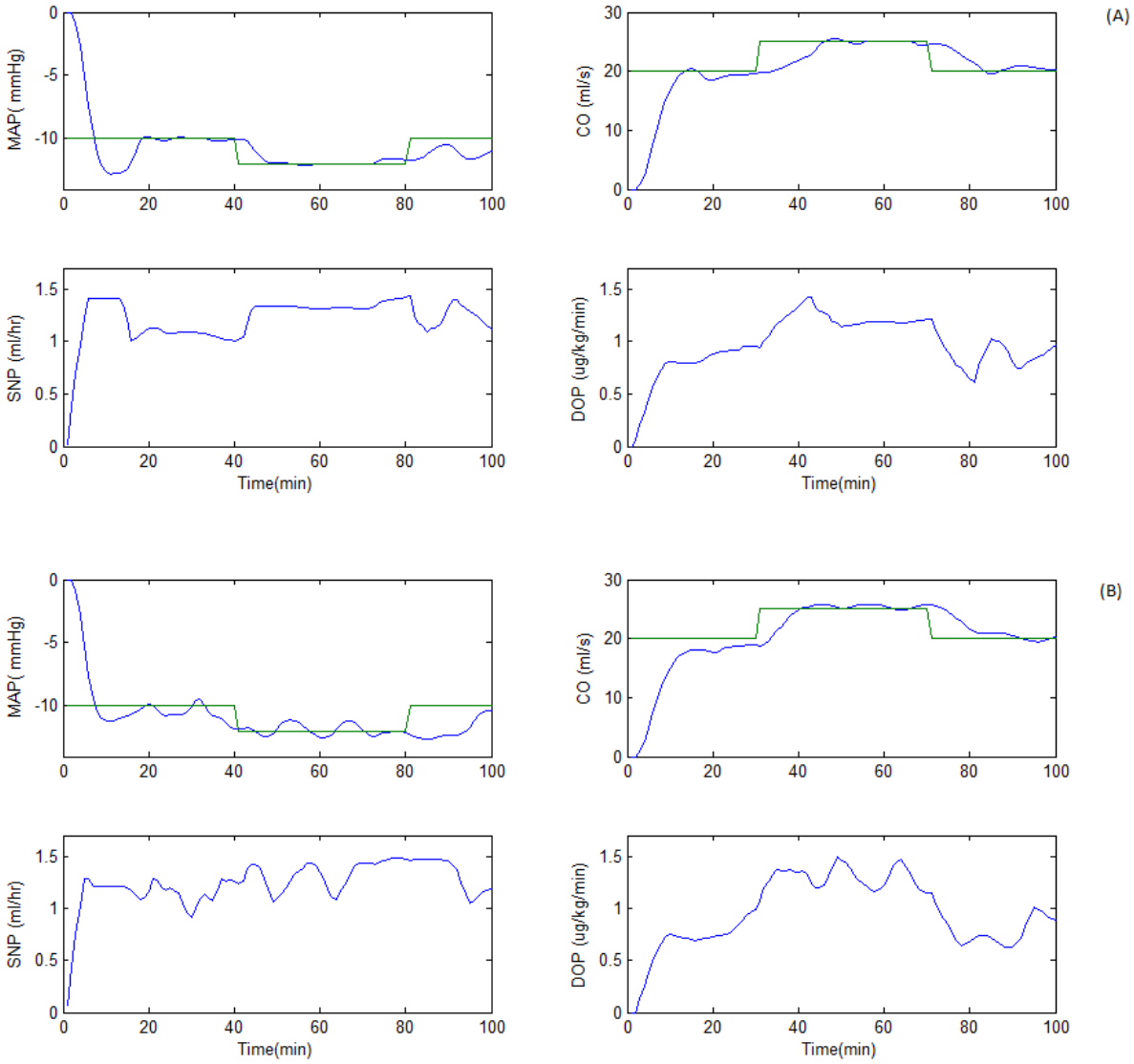
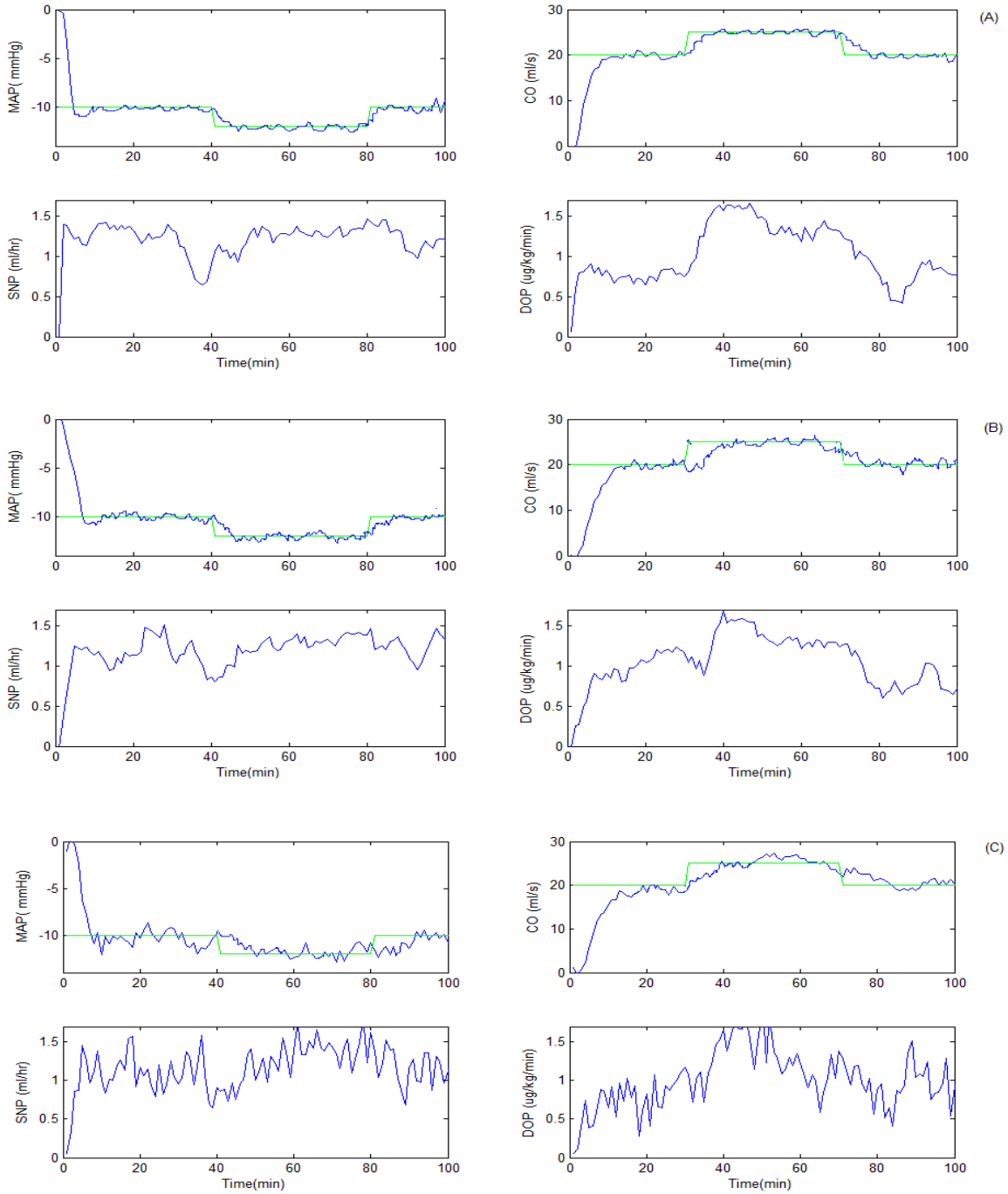


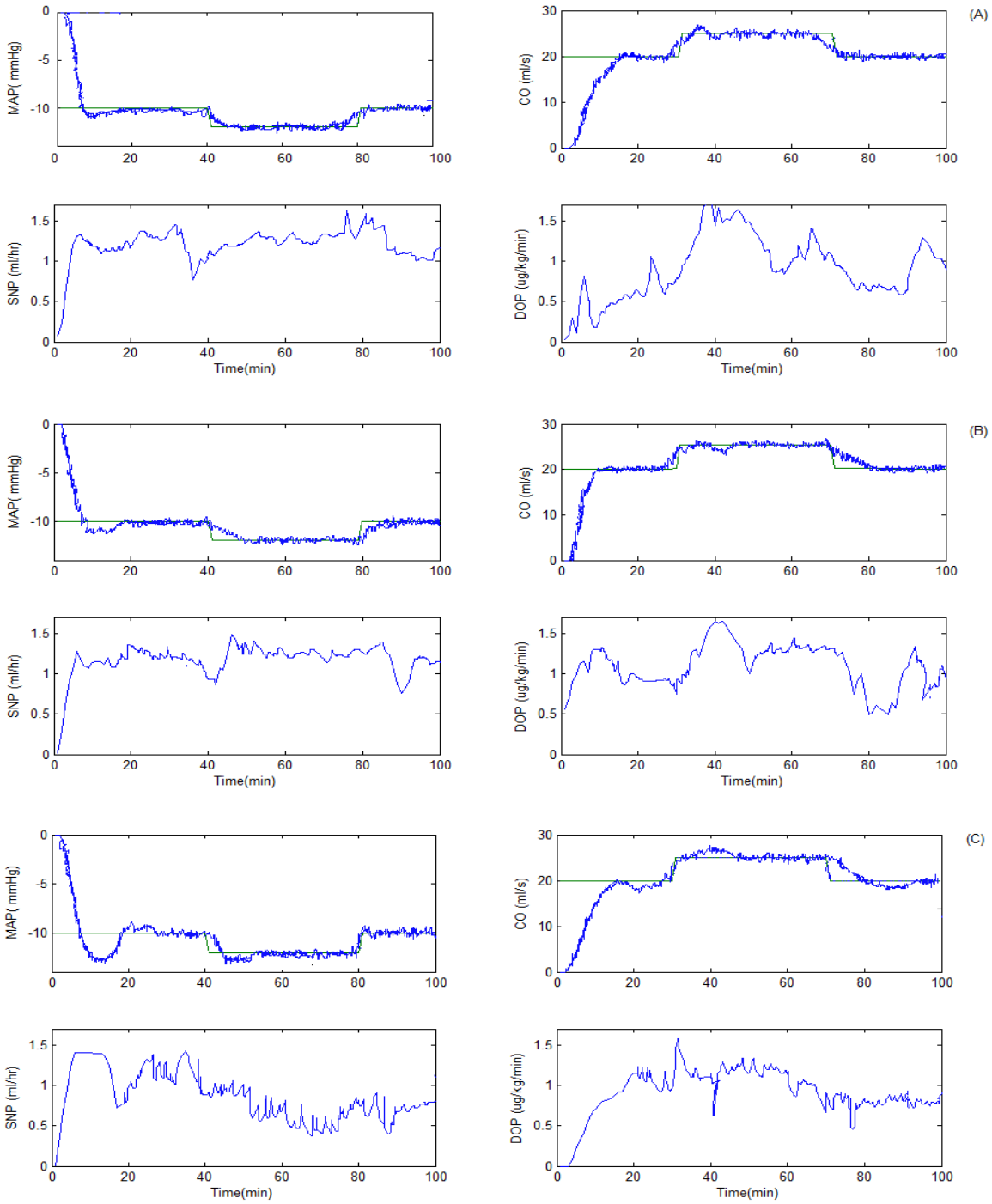
Figure 6.13: System response using two SOFLC algorithms under 15% variations of system parameters.

(A): Interval type-2 SOFLC with switching mode linguistic compensator
(B): zSlice type-2 SOFLC with RGA-based scalar compensator



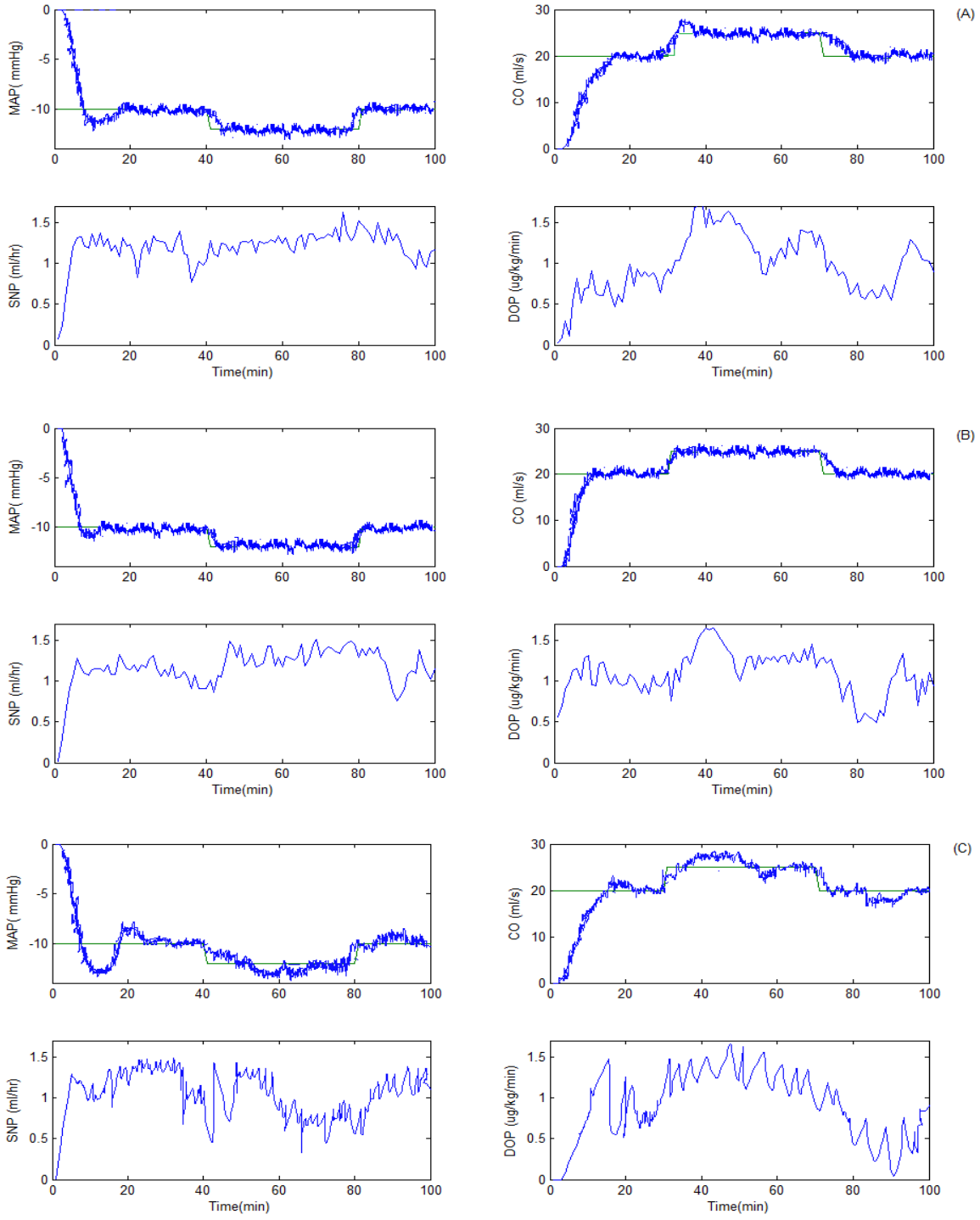
6.14: System response using various SOFLC algorithms under 5% noise.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): Interval type-2 SOFLC-DSL with switching mode linguistic compensator
(C): Type-1 SOFLC-DSL with switching mode linguistic compensator



6.16: System response using various SOFLC algorithms under 9% noise.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): Interval type-2 SOFLC-DSL with switching mode linguistic compensator
(C): Type-1 SOFLC-DSL with switching mode linguistic compensator



6.16: System response using various SOFLC algorithms under 13% noise.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): Interval type-2 SOFLC-DSL with switching mode linguistic compensator
(C): Type-1 SOFLC-DSL with switching mode linguistic compensator

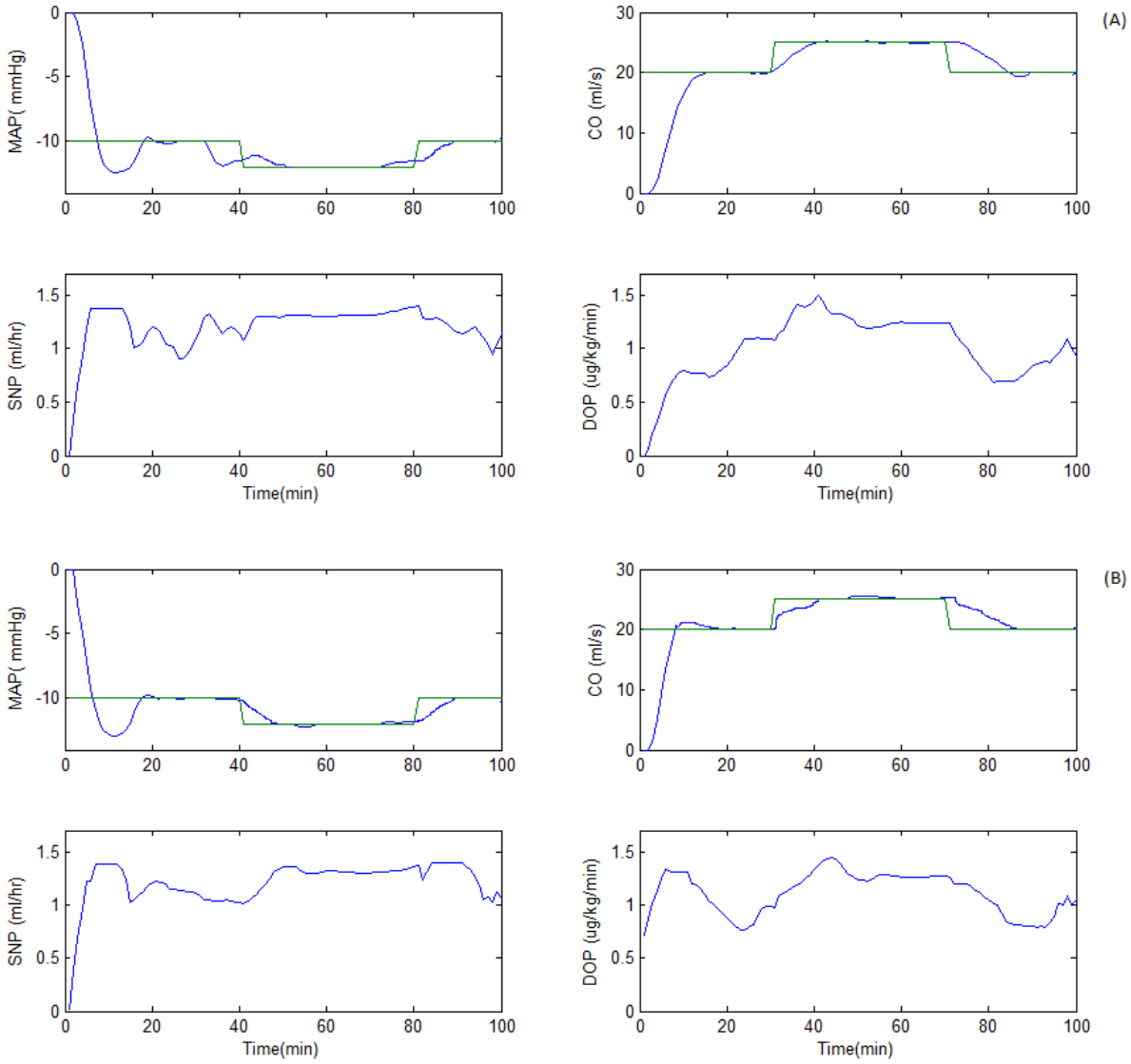


Figure 6.17: System response using two SOFLC algorithms: $\lambda_{c12} = 0.8 \times \lambda_{c12}'$, $\lambda_{c21} = 1.8 \times \lambda_{c21}'$.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): zSlice type-2 SOFLC-DSL with RGA-based scalar compensator

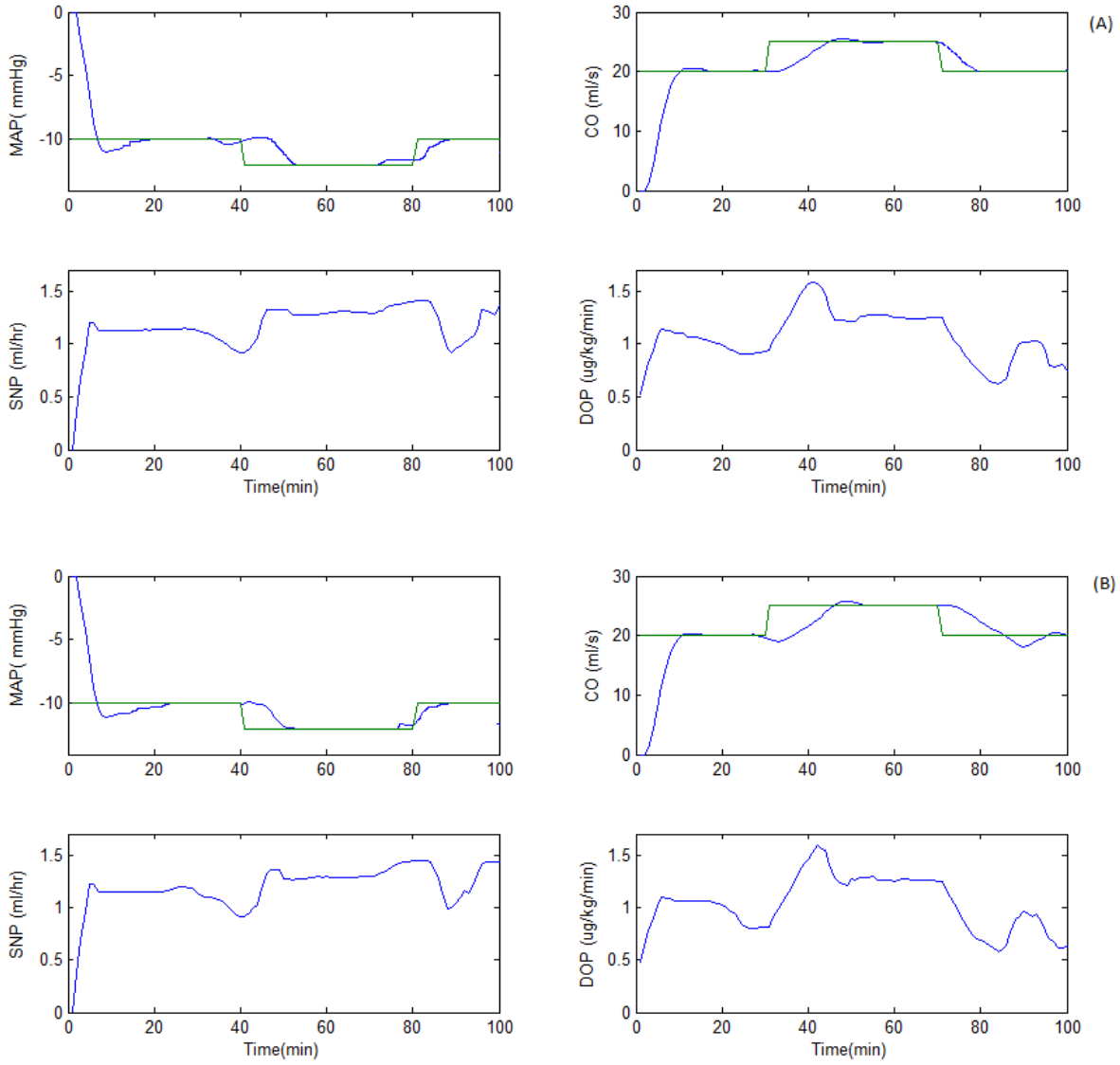


Figure 6.18: System response using two SOFLC algorithms with: $\lambda_{c12} = 1.8 \times \lambda_{c12}'$, $\lambda_{c21} = 0.8 \times \lambda_{c21}'$.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): zSlice type-2 SOFLC-DSL with RGA-based scalar compensator

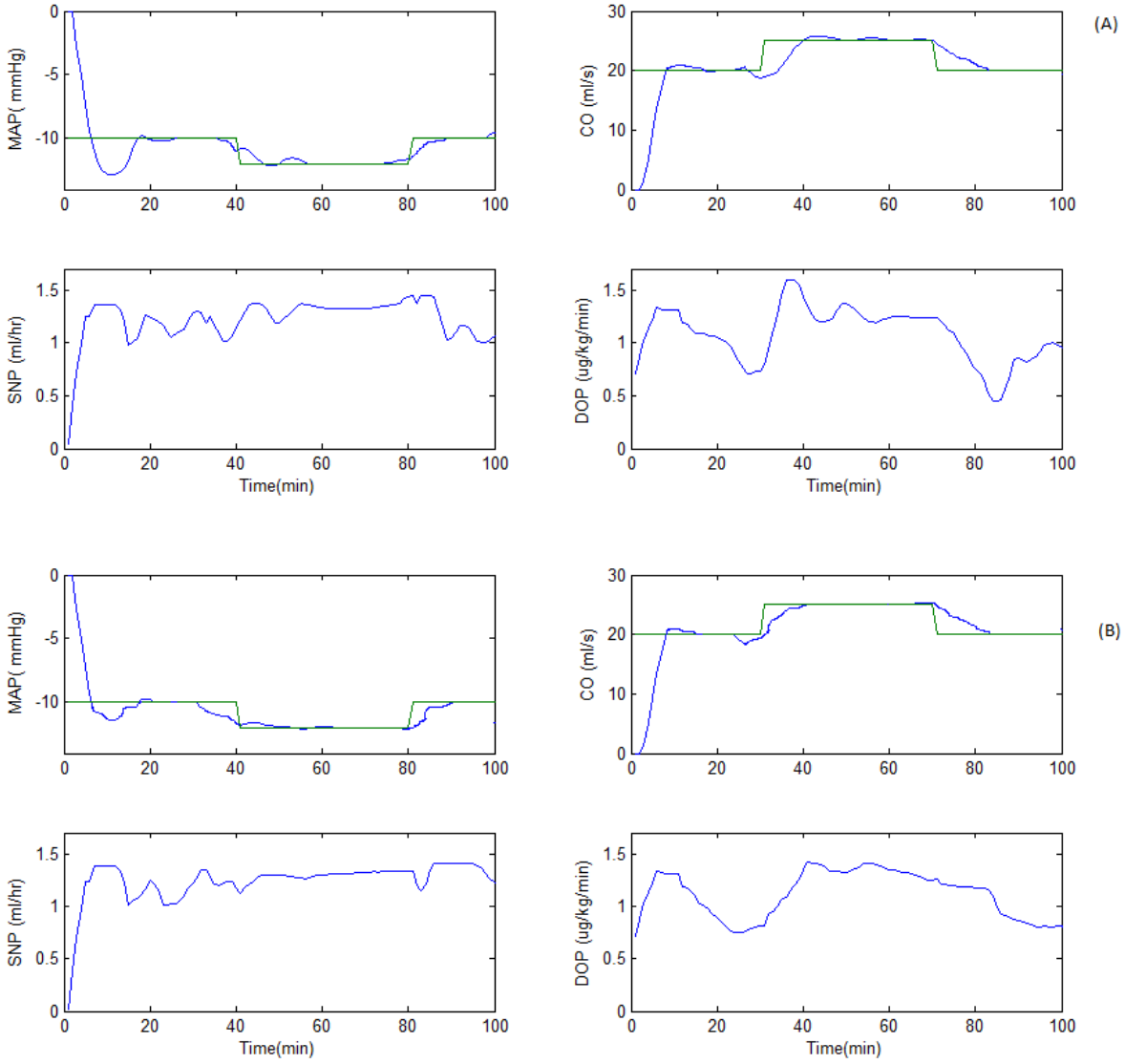


Figure 6.19: System response using two SOFLC algorithms with: $\lambda_{c12} = 0.95 \times \lambda_{c12}'$, $\lambda_{c21} = 1.1 \times \lambda_{c21}'$.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): zSlice type-2 SOFLC-DSL with RGA-based scalar compensator

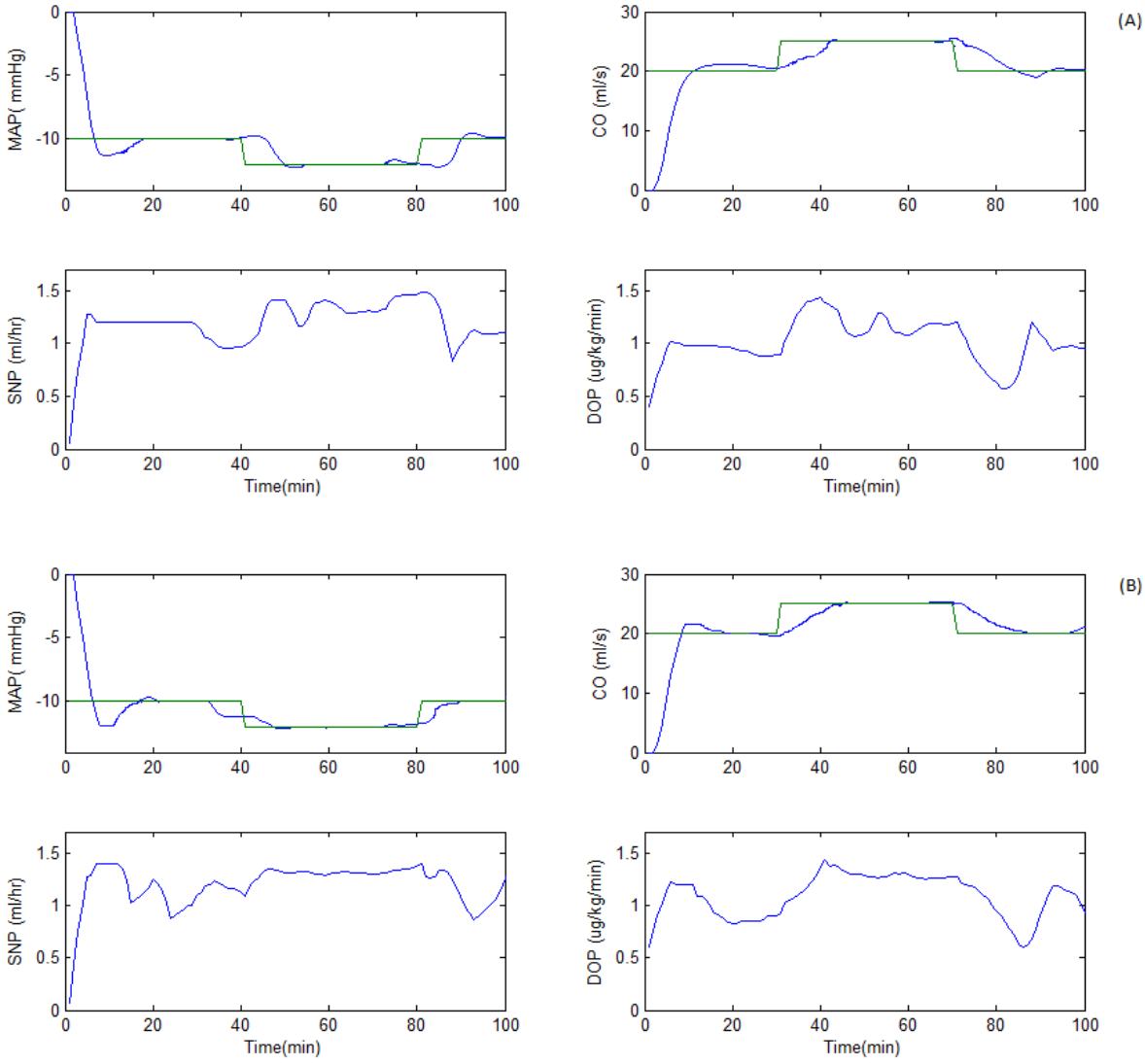


Figure 6.20: System response using two SOFLC algorithms with: $\lambda_{c12} = 0,\ \lambda_{c21} = 0$.

(A): zSlice type-2 SOFLC-DSL with switching mode linguistic compensator
(B): zSlice type-2 SOFLC-DSL with RGA-based scalar compensator

6.6 Summary

The performance of the proposed multivariable control systems that use different SOFLC-DSL algorithms as dominating controllers for each control loop were further enhanced in this chapter. This was carried-out by replacing the type-1 fuzzy sets that these controllers use with interval and zSlice type-2 fuzzy sets.

The chapter started with a review of different applications in which type-2 fuzzy sets were used to enhance the performance of multivariable fuzzy logic control systems. A further review was provided to show how the particle swarm optimisation algorithms were used to tune multivariable fuzzy control systems. Multivariable decoupled controllers that use both interval and zSlice type-2 SOFLC-DSL algorithms were then introduced and applied to the automatic control of a two-input two-output drug dynamic process. The switching mode linguistic strategy described in the previous chapter was also used to deal with the interactions between the different control channels.

These controllers were tested in different simulation tasks, including scaling factor changes, model parameter variations, inaccurate design structures and noisy environments. These two decoupled controllers were compared with other schemes that use the standard SOFLC and type-1 SOFLC-DSL algorithms.

Some of the conclusions that can be drawn from this particular study include:

- The simulation results proved the merits of including type-2 SOFLC-DSL algorithms when controlling the drug process to maintain physiological target points for both *CO* and *MAP*, especially in noisy environments, in comparison to type-1 SOFLC schemes.
- In contrast to their type-1 counterpart, both interval and zSlice SOFLC-DSL algorithms provided smoother control actions, especially in areas around the steady-state regions.
- The simulation results obtained in this chapter support the suggestions made by Karnik *et al.* (1999), that interval type-2 fuzzy systems are generally more adaptive than type-1 fuzzy systems in terms of their ability to handle more complex input-output relationships. The different simulations clearly show that the interval type-2 SOFLC-DSL were able to adaptively control the outputs of the drug process by effectively regulating the delivery of the *SNP* and *DOP*.

- It was also observed that the zSlice type-2 SOFLC-DSL algorithms provided the best performance in various experiments. However, it was also obvious that, in other cases, the interval type-2 SOFLC-DSL algorithms demonstrated their superiority. A further investigation into the number of zSlices used to construct the zSlice general type-2 fuzzy sets might lead to better results.

Figures 6.21 and 6.22 show a box plot that compares four controllers applied 15 times in different environments, and where the IAE was recorded for both *CO* and *MAP*. The best performance is denoted by the lower limit of the plot, while the upper limit corresponds to the worst performance, and the mean performance is represented by the line in between.

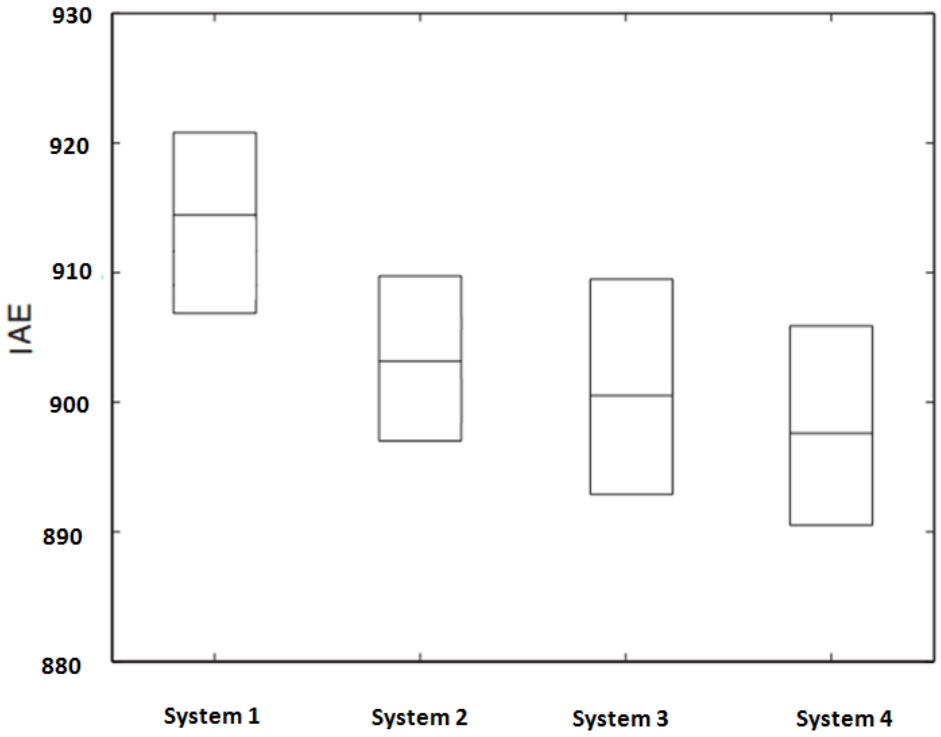


Figure 6.21: The box plot of the four controllers for CO.

System 1: Type-1 SOFLC with switching mode linguistic compensator
System 2: Type-1 SOFLC-DSL with switching mode linguistic compensator
System 3: Interval Type-2 SOFLC-DSL with switching mode linguistic compensator
System 4: zSlice Type-2 SOFLC-DSL with switching mode linguistic compensator

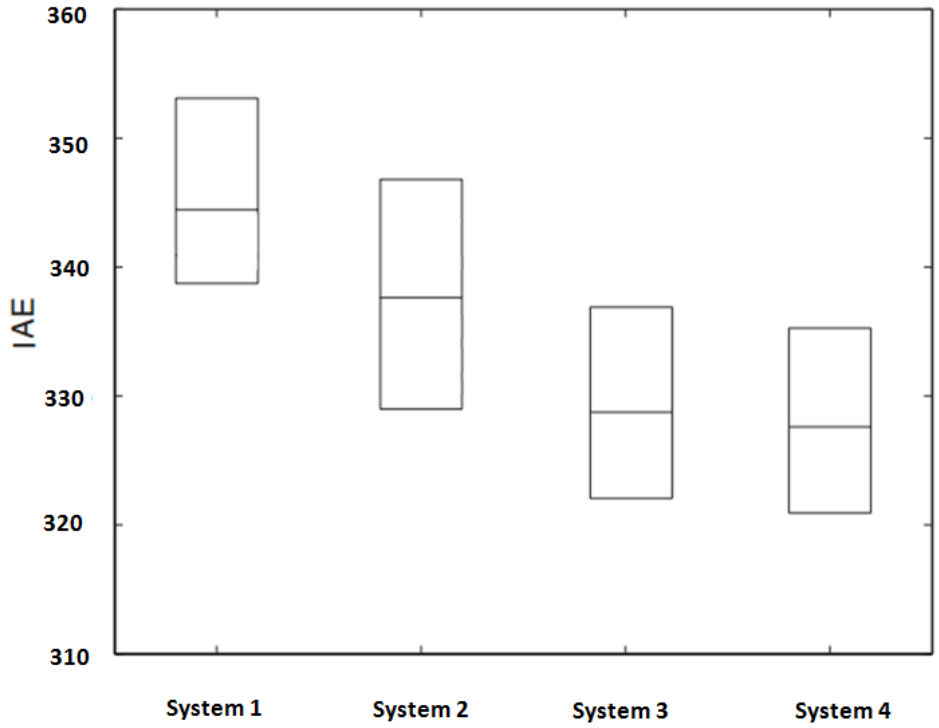


Figure 6.22: The box plot of the four controllers for *MAP*.

System 1: Type-1 SOFLC with switching mode linguistic compensator
System 2: Type-1 SOFLC-DSL with switching mode linguistic compensator
System 3: Interval Type-2 SOFLC-DSL with switching mode linguistic compensator
System 4: zSlice Type-2 SOFLC-DSL with switching mode linguistic compensator

It is also worth noting that the multivariable controllers that use both the interval and zSlice type-2 SOFLC-DSL algorithms to regulate the control loops outperformed the other two algorithms. However, it is also apparent that the former two algorithms produced very similar results, even though the zSlice SOFLC-DSL algorithm is still slightly superior in some simulations. Moreover, it can also be observed that the best IAE for a type-1 SOFLC is still greater than the mean IAE values of the interval and zSlice type-2 SOFLC-DSL algorithms.

Chapter 7 – Conclusions and Future Work

7.1 Conclusions

The work conducted and presented in this thesis is mainly concerned with the development of new Self-Organising Fuzzy Logic Control (SOFLC) algorithms that can effectively control non-linear, time-varying and mathematically ill-defined systems. The standard SOFLC scheme is a hierarchical algorithm which consists of a conventional fuzzy logic controller and self-organising mechanism that monitors the performance of the process under control, and corrects any deviation from the set-point issues by continuously modifying the control rules of the fuzzy controller.

In the newly proposed SOFLC schemes proposed in this thesis, an on-line Particle Swarm Optimisation (PSO) algorithm, combined with the idea of credit assignment and fitness estimation, were used to optimise the consequent parts of the performance index (PI) table on-line. Interval and general type-2 fuzzy logic systems were used, as well as a type-1 fuzzy logic system.

These SOFLC algorithms were applied to biomedical systems in the form of a muscle relaxant model in the single-variable case, and a 2×2 drug dynamic process in the multivariable case. The performance of these systems was measured in terms of the capacity of the proposed controllers to maintain the desired set-points over the duration of the simulation, and their sensitivity to scaling factors, sudden disturbances, structural design and noisy environments.

It has been highlighted that, most applications which involve the use of SOFLC algorithms rely on a *priori* system information and use a fixed performance index table. This restricts the abilities of the controller to effectively monitor and issue the corrective actions required for each process. This issue was solved with the introduction of a dynamic supervisory layer that has the ability to modify its structure based on the feedback received from the process to be controlled. This enriched the capabilities of the SOFLC algorithm in various ways, including the following.

- The proposed self-organising Fuzzy Logic Control with a dynamic supervisory layer (SOFLC-DSL) starts with an empty performance index table, unlike the standard SOFLC scheme. The on-line PSO algorithm has kept modifying the PI table at every sampling instance during the whole simulation. This has led to better set-point tracking properties, and also has made the lower-level fuzzy logic controller rely on fewer fuzzy rules. This is vitally important in terms of simplifying the complexity of the controllers and reducing the computational costs, a characteristic that is crucial for successful implementation of the controller in real-time.
- The simulation results also have shown the superiority of the SOFLC-DSL algorithms in terms of producing a faster response with a reduced control effort, in comparison with the standard SOFLC scheme with a fixed performance index table. This is, once again, a result of the capacity of the on-line PSO algorithm to use the feedback signals obtained from the system to effectively tune the consequence parts of the PI table. Furthermore, the flexibility of the PI table in terms of not being restricted to a certain range of values helped the controller provide reasonable control effort, without, of course, compromising on the speed of the response.

- The proposed SOFLC-DSL algorithms have produced satisfactory performance even when non-optimally designed scaling factors were used for the error, change of error, and output of the controller. To further test the applicability of the proposed algorithm for working with a wide range of processes without the need for re-tuning, it was applied to different muscle relaxant models. In all the simulations, the controller succeeded in producing good performance with minimal errors in terms of desired trajectory and smooth control signals. Similarly, good results were obtained when a sudden disturbance was introduced. All these simulations verified the merits of the supervisory layer, and demonstrated how it enabled the design structure of the controller to be more flexible and effective.

As stated above, the SOFLC-DSL algorithm depends on the feedback signals obtained from the process used to generate the suitable correction values needed by the lower-level fuzzy logic controller. However, when these feedback signals are corrupted with noise, the SOFLC-DSL algorithm should fail to accurately evaluate the performance of the process, and therefore should also fail to produce sufficient control signals. In order to overcome this issue and allow the controller to work effectively in stochastic environments, a third-order polynomial filter with a window of 20 samples was used. This approach enabled the SOFLC-DSL algorithm to use a function that best represents the tracking tendency with the most appropriate shape.

Simulation results demonstrated the robustness of the controller in the stochastic case. However, it was also observed that as the level of the noise increased, the controller started to find it difficult to maintain the desired set-point with smooth control effort. In order to improve the control performance in such conditions and environment, the interval and zSlice type-2 fuzzy sets were introduced.

Type-2 SOFLC-DSL algorithms were applied to the same model in controlling anaesthesia delivery so as to maintain physiological target level for muscle relaxation during a three-stage surgical procedure. Similar to their type-1 counterpart, the proposed interval and type-2 SOFLC-DSL algorithms were tested in different environments and under different conditions. In addition, their performances were compared to type-1 SOFLC and type-1 SOFLC-DSL schemes. In normal circumstances, computer simulations showed that type-1 and type-2 SOFLC-DSL algorithms produced satisfactory results, and that they all outperformed the type-1 SOFLC scheme in terms maintaining the desired set-points, with

faster rising time and reduced overshoots and undershoots. However, as the levels of uncertainty increased in the form of varying the scaling factors and the parameters of the dynamic model as well as introducing noise, it was apparent that those controllers provided with type-2 fuzzy sets outperformed their type-1 counterparts (both SOFLC and SOFLC-DSL algorithms). In the light of these, the following remarks should be noted:

- Type-2 fuzzy sets enabled the SOFLC-DSL to effectively handle uncertainty and produce better tracking performance than those of type-1. This superior performance was achieved with a smooth control effort. Conversely, the control actions of the type-1 schemes produced fluctuating control singles in certain areas, especially when the levels of uncertainty were high. Fluctuating drug delivery can pose risk to patients.

- It was noticed that type-1 and type-2 SOFLC-DSL generated few control rules to make the controlled variables reach the set points.

- Type-1 SOFLC-DSL algorithms tended to perform reasonably well in various environments. However, as the levels of noise increased, the controller failed to maintain the controlled variable at the desired level. Moreover, the control actions provided by the controller fluctuated more in certain regions, particularly when the controlled variable shifted from one set-point stage to another. This was, obviously, due to the failure of type-1 fuzzy sets to fully capture the noise surrounding the process in this environment. Therefore, the system was unable to provide an accurate assessment of the status of the output.

- The results in all the simulations of this chapter, especially in the stochastic case, showed that performances of interval type-2 and zSlice type-2 SOFLC-DSL algorithms were very similar, and that it was difficult to prove that zSlice type-2 fuzzy systems, regarded as forms of a general type-2 system, produced better results than those of interval type-2 fuzzy systems. This could be for two main reasons. First, the levels of uncertainty surrounding the environment experienced in this thesis might not have been high enough, to the point where zSlice type-2 fuzzy sets could produce better results. Second, the number of zSlices representing the resolution of the zSlice type-2 fuzzy set might not have been high enough. Wanger and Hagras (2010) describe how the ability of zSlice type-2 fuzzy sets to handle high levels of uncertainty depends on the number of zSlices of which these fuzzy

sets are formed. Therefore, suitable considerations of the number of zSlices used could lead to significant improvements of the controller.

After successful results were obtained in the single-variable case, the proposed type-1 and type-2 SOFLC-DSL algorithms were extended to the multivariable case. It has been highlighted that, decoupled control of multivariable systems has become an effective tool for handling the challenges that multivariable systems normally experience, such as rule explosion in relation to the increase in the dimensionality of the input space. The success of decoupled controllers in regulating multivariable systems depends on their ability to handle the interaction between the input-output control channels. Two main approaches were used to reduce the effects of the interaction in this thesis, the first of which was the relative gain arrays (RGA) compensator, which is heavily used in the literature. The second approach was the switching mode linguistic compensator, which not only handled the interaction but also used the feedback obtained from the system to improve performance and reduce the tracking errors.

Simulation results show how the switching mode linguistic compensator, coupled with the dynamic supervisory layer, enabled the SOFLC algorithms to produce satisfactory results in various environments in terms of making the controlled variables reach the set-point values, shortening settling time and reducing overshoot. It was also obvious that the decoupled control system with the switching strategy outperformed the decouple control system with the RGA matrix.

As noticed in previous chapters, as the levels of noise increased, type-1 SOFLC-DSL algorithms provided less satisfactory performance, with less smooth control actions than those of type-2 SOFLC-DSL algorithms, especially in noise-containment environments. However, type-1 SOFLC-DSL algorithms still managed to show good robustness under sub-optimum scaling factors, varying system dynamics, and inaccurate estimate of the relative steady-state gains. This was purely due to the on-line learning ability of these algorithms. The standard SOFLC scheme with a fixed performance index table, as in the single-variable case, provided the poorest performance among all controllers.

7.2 Future Work

This thesis has presented new SOFLC algorithms using type-1 and type-2 fuzzy systems. Although several contributions were made and various questions were answered, there are still some areas in which the proposed controllers can be further investigated for potential in improvement in performance. Some of the recommendations for future work include:

- Enhancing the capabilities of the SOFLC-DSL algorithms using other bio-inspired optimisation algorithms, such as ant colony optimisation, which is widely used to search for optimal or nearly optimal solutions and has been successfully applied in many research and application areas. Ant colony optimisation has been proved to produce better results and to have faster convergence properties than other algorithms. These features can be beneficial to the development of the dynamic supervisory layer and the improvement of the performance of the SOFLC-DSL algorithms. Future work should consider the replacement of the PSO algorithm with the ant colony optimisation algorithm.

- In order to further explore the capabilities of zSlice type-2 fuzzy systems, real patient data need to be considered to construct the zSlices and investigate if this leads to any improvement in the control performance of the zSlice type-2 SOFLC-DSL algorithms, to tackle higher levels of uncertainty. Moreover, in this thesis, only five zSlices were used to construct the fuzzy sets involved in all the simulations. Increasing the number of slices should allow the fuzzy sets to have better flexibility to measure uncertainty, and therefore outperform the interval type-2 SOFLC-DSL algorithms.

- All the type-1 and type-2 SOFLC-DSL algorithms proposed in this thesis were applied to biomedical systems. Future work should look at applying the controllers to other industrial applications. Furthermore, the computational cost of the developed dynamic supervisory layer is low, making the proposed algorithm sufficient for practical implementation. Therefore, applying these controllers to real-world, practical systems and in real-time should also be considered in the future.

- The on-line learning mechanism of the proposed controllers proved to be effective in terms of modifying the rule-base of the lower level fuzzy logic controller on-line.

However, there are other available techniques that can be easily employed to further accelerate the learning speed when the controller has limited access to the information of the systems. Some of these approaches include meta-learning, which is a promising field in the area of machine learning. Meta-learning refers to the use of experience to increase the efficiency of the learning system and make the learning process more adjustable, based on the performance of the task under control or study. Meta-learning approaches that 'learn how to learn' and guide the selection of learning models have been used in various domains, including computational intelligence, statistics and intelligent systems. Moreover, learning and improving the control performance of systems from data can sometimes require pre-processing steps, such as selecting appropriate information and choosing suitable methods for classification. Meta-learning approaches can help create optimal models by reusing past experiences from the analysis of other applications or problems, reducing the reliance of system on *priori* design knowledge and improving performance based on these experiences. Meta-learning represents a promising approaching for improving the on-line learning capabilities of the SOFLC-DSL algorithms.

The above approaches and techniques can extend the study of the SOFLC-DSL algorithms and pave the way for future opportunities in this exciting area of research, with the ultimate goal of improving the control performance and the self-learning abilities of the proposed algorithms to make them suitable for more diverse applications.

References

Abilov, A. G., Zeybek, Z., Tuzunalp, O., and Telatar, Z. (2002). Fuzzy temperature control of industrial refineries furnaces through combined feedforward/feedback multivariable cascade systems. Chemical Engineering and Processing: Process Intensification, 41(1), 87–98. http://doi.org/10.1016/S0255-2701(01)00119-2

Ali, T., Anis, S., Faouzi, M. S., Multivariable, A. N., and Description, S. (2015). Design of Multivariable Predictive Controller Based on New constrained Multi-objective PSO And T-S Fuzzy Modeling Approach, (Icmic), 3–7.

Aliasghary, M., Eksin, I., Guzelkaya, M., and Kumbasar, T. (2012). Design of an interval type-2 fuzzy logic controller based on conventional PI controller. In 2012 20th Mediterranean Conference on Control and Automation (MED) (pp. 627–632). IEEE. http://doi.org/10.1109/MED.2012.6265708

Allaoua, B., Gasbaoui, B., and Mebarki, B. (2009). Setting up PID DC motor speed control alteration parameters using particle swarm optimization strategy. Leonardo Electronic Journal of Practices and Technologies, 7(14), 19–32.

Araujo, H., Xiao, B., Liu, C., Zhao, Y., and Lam, H. K. (2014). Design of Type-1 and Interval Type-2 Fuzzy PID Control for Anesthesia Using Genetic Algorithms, (May), 70–93.

Bagheri, A., and Moghaddam, J. J. (2009). Decoupled adaptive neuro-fuzzy (DANF) sliding mode control system for a Lorenz chaotic problem. Expert Systems with Applications, 36(3), 6062–6068. http://doi.org/10.1016/j.eswa.2008.06.123

Baydokhty, M., Zare, A., and Balochian, S. (2016). Performance of optimal hierarchical type 2 fuzzy controller for load – frequency system with production rate limitation and governor dead band. Alexandria Engineering Journal, 55(1), 379–397. http://doi.org/10.1016/j.aej.2015.12.003

Bingül, Z., and Karahan, O. (2011). A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control. Expert Systems with Applications, 38(1), 1017–1031. http://doi.org/10.1016/j.eswa.2010.07.131

Bristol, E. (1966). On a new measure of interaction for multivariable process control. IEEE Transactions on Automatic Control, 11(1), 133–134. http://doi.org/10.1109/TAC.1966.1098266

Cao, J., Liu, H., and Brown, D. (2008). Adaptive fuzzy logic controller for vehicle active suspensions with interval type-2 fuzzy membership functions. In 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence) (pp. 83–89). IEEE. http://doi.org/10.1109/FUZZY.2008.4630348

Cara, A. B., Rojas, I., Pomares, H., Wagner, C., and Hagras, H. (2011). On comparing non-singleton type-1 and singleton type-2 fuzzy controllers for a nonlinear servo system. In 2011 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ) (pp. 126–133). IEEE. http://doi.org/10.1109/T2FUZZ.2011.5949560

Castillo, O., Melin, P., Kacprzyk, J., and Pedrycz, W. (2007). Type-2 Fuzzy Logic: Theory and Applications. In 2007 IEEE International Conference on Granular Computing (GRC 2007) (pp. 145–145). IEEE. http://doi.org/10.1109/GrC.2007.118

Cazarez-Castro, N. R., Aguilar, L. T., and Castillo, O. (2012). Designing Type-1 and Type-2 Fuzzy Logic Controllers via Fuzzy Lyapunov Synthesis for nonsmooth mechanical systems. Engineering Applications of Artificial Intelligence, 25(5), 971–979. http://doi.org/10.1016/j.engappai.2012.03.003

Cerman, O. (2013). Fuzzy model reference control with adaptation mechanism. Expert Systems with Applications, 40(13), 5181–5187. http://doi.org/10.1016/j.eswa.2013.03.014

Cervantes, L., and Castillo, O. (2015). Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control. Information Sciences, 324, 247–256. http://doi.org/10.1016/j.ins.2015.06.047

Cheng, S., and Chiou, J. (2010). Application of the GA-PSO with the Fuzzy controller to the robot soccer, 2083–2087.

Chiou, K.-C., and Huang, S.-J. (2005). An adaptive fuzzy controller for robot manipulators. Mechatronics, 15(2), 151–177. http://doi.org/10.1016/j.mechatronics.2004.07.005

Chou, C.-H., and Lu, H.-C. (1994). A heuristic self-tuning fuzzy controller. Fuzzy Sets and Systems, 61(3), 249–264. http://doi.org/10.1016/0165-0114(94)90168-6

Chou, Y. C., Abbod, M. F., Shieh, J. S., and Hsu, C. Y. (2010). Multivariable fuzzy logic/self-organizing for anesthesia control. Journal of Medical and Biological Engineering, 30(5), 297–306. http://doi.org/10.5405/jmbe.30.5.05

Chunshien Li, Kuo-Hsiang Cheng, Chih-Ming Chen, and Jin-Long Chen. (2004). Soft computing approach to adaptive noise filtering. In IEEE Conference on Cybernetics and Intelligent Systems, 2004. (Vol. 1, pp. 1–5). IEEE. http://doi.org/10.1109/ICCIS.2004.1460377

Coupland, S., and John, R. (2007). Geometric Type-1 and Type-2 Fuzzy Logic Systems. IEEE Transactions on Fuzzy Systems, 15(1), 3–15. http://doi.org/10.1109/TFUZZ.2006.889764

Daley, S., and Gill, K. (1986). Design study for self-organising fuzzy logic controller. Pro. Insinuation Mechanical Engineers, 200C(1),59-69.

Dereli, T., Baykasoglu, A., Altun, K., Durmusoglu, A., and Türksen, I. B. (2011). Industrial applications of type-2 fuzzy sets and systems: A concise review. Computers in Industry, 62(2), 125–137. http://doi.org/10.1016/j.compind.2010.10.006

Dias, J. M., and Dourado, A. (1999). A self-organizing fuzzy controller with a fixed maximum number of rules and an adaptive similarity factor. Fuzzy Sets and Systems, 103(1), 27–48. http://doi.org/10.1016/S0165-0114(97)00192-9

Doctor, F., Hagras, H., and Callaghan, V. (2005). A Fuzzy Embedded Agent-Based Approach for Realizing Ambient Intelligence in Intelligent Inhabited Environments. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 35(1), 55–65. http://doi.org/10.1109/TSMCA.2004.838488

Doctor, F., Syue, C.-H., Liu, Y.-X., Shieh, J.-S., and Iqbal, R. (2016). Type-2 fuzzy sets applied to multivariable self-organizing fuzzy logic controllers for regulating anesthesia. Applied Soft Computing, 38, 872–889. http://doi.org/10.1016/j.asoc.2015.10.014

Du, X., and Ying, H. (2010). Derivation and analysis of the analytical structures of the interval type-2 fuzzy-PI and PD controllers. IEEE Transactions on Fuzzy Systems, 18(4), 802–814. http://doi.org/10.1109/TFUZZ.2010.2049022

Dubois, D., and Prade, H. (1998). Fuzzy sets and systems: theory and applications. Vol. 144, Academic Pr.

D. R. Stanski, Monitoring depth of anesthesia, in Anesthesia, R. D. Miller, Ed. Anesthesia, (Churchill Livingstone, New York, 1994, pp. 1127-1159).

Eberhart, R. C., and Shi, Y. (1998). Evolving artificial neural networks. Int'l Conf. on Neural Networks and Brain Pp. PL5-PL13. Beijing, P. R. China.

Edwin, L. (2008). A Comparative Study of two Approaches for Data-Driven Design of Evolving Fuzzy Systems: eTS and FLEXFIS. International Journal of General Systems, 37, 45–67.

El-Bardini, M., and El-Nagar, A. M. (2011). Direct adaptive interval type-2 fuzzy logic controller for the multivariable anaesthesia system. Ain Shams Engineering Journal, 2(3-4), 149–160. http://doi.org/10.1016/j.asej.2011.08.001

Esmaeili, V., Assareh, A., Shamsollahi, M. B., Moradi, M. H., and Arefian, N. M. (2008). Estimating the depth of anesthesia using fuzzy soft computation applied to EEG features. Intelligent Data Analysis, 12(4), 393–407.

Fazel Zarandi, M. H., Rezaee, B., Turksen, I. B., and Neshat, E. (2009). A type-2 fuzzy rule-based expert system model for stock price analysis. Expert Systems with Applications, 36(1), 139–154. http://doi.org/10.1016/j.eswa.2007.09.034

Figueroa, J., Posada, J., Soriano, J., Melgarejo, M., and Rojas, S. (2005). A Type-2 Fuzzy Controller for Tracking Mobile Objects in the Context of Robotic Soccer Games. In The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05. (pp. 359–364). IEEE. http://doi.org/10.1109/FUZZY.2005.1452420

Gafa, C., and Coupland, S. (2011). A new recursive type-reduction procedure for general type-2 fuzzy sets. IEEE SSCI 2011: Symposium Series on Computational Intelligence - T2FUZZ 2011: 2011 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems, (3), 44–49. http://doi.org/10.1109/T2FUZZ.2011.5949548

Gangopadhyay, A., and Meckl, P. (2001). Multivariable PI tuning for disturbance rejection and application to engine idle speed control simulation. International Journal of Control, 74(10), 1033–1041. http://doi.org/10.1080/00207170110049486

Golea, N., Golea, A., and Benmahammed, K. (2002). Fuzzy model reference adaptive control. IEEE Transactions on Fuzzy Systems, 10(4), 436–444. http://doi.org/10.1109/TFUZZ.2002.800694

Greenfield, S. (2012). Type-2 fuzzy logic: circumventing the defuzzification bottleneck.

Greenfield, S., Chiclana, F., Coupland, S., and John, R. (2009). The collapsing method of defuzzification for discretised interval type-2 fuzzy sets. Information Sciences,

179(13), 2055–2069. http://doi.org/10.1016/j.ins.2008.07.011

Greenfield, S., John, R., and Coupland, S. (2005). A novel sampling method for type-2 defuzzification, in. 'Proceedings of UKCI , London, Pp. 120–127.

Grosdidier, P., Morari, M., and Holt, B. R. (1985). Closed-loop properties from steady-state gain information. Industrial and Engineering Chemistry. Fundamentals - ACS, Vol. 24, No. 2, Pp. 221-235.

Hagras, H. a. (2004). A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE Transactions on Fuzzy Systems, 12(4), 524–539. http://doi.org/10.1109/TFUZZ.2004.832538

Hassani, H., and Zarei, J. (2015). Interval Type-2 fuzzy logic controller design for the speed control of DC motors. Systems Science and Control Engineering, 3(1), 266–273. http://doi.org/10.1080/21642583.2015.1013644

Ho, H. F., Wong, Y. K., and Rad, A. B. (2003). Adaptive fuzzy sliding mode control for SISO nonlinear systems. In The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ '03. (Vol. 2, pp. 1344–1349). IEEE. http://doi.org/10.1109/FUZZ.2003.1206626

Hou, G., Qin, L., Zheng, X., and Zhang, J. (2011). Design of PSO-based fuzzy gain scheduling PI controller for four-area interconnected AGC system after deregulation, 72–76.

Hu, Y.-C., Chen, R.-S., and Tzeng, G.-H. (2003). Finding fuzzy classification rules using data mining techniques. Pattern Recognition Letters, 24, 509–519. http://doi.org/10.1016/S0167-8655(02)00273-8

Huang, S., and Lee, J. (2000). A Stable Self-Organizing Fuzzy Controller for, 47(2), 421–428.

Huang, S.-J., and Lin, W.-C. (2003). A Self-Organizing Fuzzy Controller for an Active Suspension System. Journal of Vibration and Control, 9(9), 1023–1040. http://doi.org/10.1177/107754603030675

Karnik, N., and Mendel, J. (2001). Centroid of a type-2 fuzzy set. Information Sciences 132(1), 195–220.

Karnik, N., and Mendel, J. (2011). Operations on type-2 fuzzy sets. Fuzzy Sets and Systems 122(2), 327–348.

Karnik, N. N., and Mendel, J. M. (1998). Introduction to type-2 fuzzy logic systems. In 1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36228) (Vol. 2, pp. 915–920). IEEE. http://doi.org/10.1109/FUZZY.1998.686240

Karnik, N. N., and Mendel, J. M. (2001). Centroid of a type-2 fuzzy set. Informayion Sciences, 132, 195–220. http://doi.org/10.1016/S0020-0255(01)00069-X

Karnik, N. N., Mendel, J. M., and Liang, Q. (1999). Type-2 Fuzzy Logic Systems, 7(6), 643–658.

Kennedy, J., and Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95 - International Conference on Neural Networks (Vol. 4, pp. 1942–1948). IEEE. http://doi.org/10.1109/ICNN.1995.488968

Kim, S. B., and Yun, C. B. (2000). Sliding mode fuzzy control: Theory and verification on a benchmark structure. Earthquake Engineering and Structural Dynamics, 29(11), 1587–1608. http://doi.org/10.1002/1096-9845(200011)29:11<1587::AID-EQE974>3.0.CO;2-W

Kim, W.-D., Jang, H.-J., and Oh, S.-K. (2010). The design of optimized fuzzy cascade controller: focused on type-2 fuzzy controller and HFC-based genetic algorithms. Transactions of the Korean Institute of Electrical Engineers, 59(5), 972–980.

Koutb, M. A., El-Rabaie, N. M., Awad, R. A., and Hewaidy, S. M. (2004). Multivariable fuzzy control for a non-linear drum boiler process. In International Conference on Electrical, Electronic and Computer Engineering, 2004. ICEEC '04. (pp. 3–9). IEEE. http://doi.org/10.1109/ICEEC.2004.1374365

Kumbasar, T., and Hagras, H. (2015). A Self-Tuning zSlices-Based General Type-2 Fuzzy PI Controller. IEEE Transactions on Fuzzy Systems, 23(4), 991–1013. http://doi.org/10.1109/TFUZZ.2014.2336267

Layne, J. R., and Passino, K. M. (1996). Fuzzy Model Reference Learning Control. Journal of Intelligent and Fuzzy Systems: Applications in Engineering and Technology, 4(1), 33–47.

Lee, Marro, K., Shankland, E., and Mathis, M. (2010). Quantitative 19F imaging using inductively coupled reference signal injection. Magnetic Resonance in Medicine, 63(3), 570–3. http://doi.org/10.1002/mrm.22298

Lee, Nie, J. H., and Lee, M. W. (1997). A fuzzy controller with decoupling for multivariable nonlinear servo-mechanisms, with application to real-time control of a passive line-of-sight stabilization system. Mechatronics, 7(1), 83–104. http://doi.org/10.1016/S0957-4158(96)00036-0

Lee, C. C. (1990). Fuzzy logic in control systems: fuzzy logic controller. II. IEEE Transactions on Systems, Man, and Cybernetics, 20(2), 419–435. http://doi.org/10.1109/21.52552

Lee, T. H., Nie, J. H., and Tan, W. K. (1994). A self-organizing fuzzified basis function network control system applicable to nonlinear servomechanisms. Mechatronics, vol.5,pp.695-713.

Lei, S., and Langari, R. (2000). Hierarchical fuzzy logic control of a double inverted pendulum. In Ninth IEEE International Conference on Fuzzy Systems. FUZZ- IEEE 2000 (Cat. No.00CH37063) (Vol. 2, pp. 1074–1077). IEEE. http://doi.org/10.1109/FUZZY.2000.839201

Leng, G., McGinnity, T. M., and Prasad, G. (2005). An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network. Fuzzy Sets and Systems, 150(2), 211–243. http://doi.org/10.1016/j.fss.2004.03.001

Li, Yi, J., and Zhao, D. (2009). Control of the TORA system using SIRMs based type-2 fuzzy logic. In 2009 IEEE International Conference on Fuzzy Systems (pp. 694–699). IEEE. http://doi.org/10.1109/FUZZY.2009.5277305

Li, C., and Priemer, R. (1999). Fuzzy control of unknown multiple-input-multiple-output plants. Fuzzy Sets and Systems, 104(2), 245–267. http://doi.org/10.1016/S0165-0114(97)00217-0

Li, Y., Ling, L., and Chen, J. (2015). Combined grey prediction fuzzy control law with application to road tunnel ventilation system. Journal of Applied Research and Technology, 13(2), 313–320. http://doi.org/10.1016/j.jart.2015.06.009

Li, Z. (2011). Model Reference Adaptive Controller Design Based on Fuzzy Inference System ⋆. Science And Technology, 9(September), 1683–1693.

Lian, R.-J. (2012). Grey-prediction self-organizing fuzzy controller for robotic motion control. Information Sciences, 202, 73–89. http://doi.org/10.1016/j.ins.2012.03.015

Lian, R.-J., and Huang, S.-J. (2001). A Mixed Fuzzy Controller for MIMO Systems. Fuzzy Sets and Systems, Vol.120, Pp. 73-93.

Lin, C.-J., Lee, J.-H., and Lee, C.-Y. (2008). A novel hybrid learning algorithm for parametric fuzzy CMAC networks and its classification applications. Expert Systems with Applications, 35(4), 1711–1720. http://doi.org/10.1016/j.eswa.2007.08.086

Lin, J., and Lian, R.-J. (2013). Design of a grey-prediction self-organizing fuzzy controller for active suspension systems. Applied Soft Computing, 13(10), 4162–4173. http://doi.org/10.1016/j.asoc.2013.06.003

Linda, O., and Manic, M. (2010). Comparative analysis of Type-1 and Type-2 fuzzy control in context of learning behaviors for mobile robotics. In IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society (pp. 1092–1098). IEEE. http://doi.org/10.1109/IECON.2010.5675521

Linkens, D. A., and Abbod, M. F. (1992). Self-organising fuzzy logic control and the selection of its scaling factors. Transactions of the Institute of Measurement and Control, 14(3), 114–125. http://doi.org/10.1177/014233129201400301

Linkens, D. A., and Nyongesa, H. O. (1995). Genetic algorithms for fuzzy control.1. Off-line system development and application. IEE Proceedings - Control Theory and Applications, 142(3), 161–176. http://doi.org/10.1049/ip-cta:19951766

LIU, F. (2008). An efficient centroid type-reduction strategy for general type-2 fuzzy logic system. Information Sciences, 178(9), 2224–2236. http://doi.org/10.1016/j.ins.2007.11.014

Liu, H., Li, S., and Chai, T. (2003). Intelligent control of power-plant main steam pressure and power output. In Proceedings of the 2003 American Control Conference, 2003. (Vol. 4, pp. 2857–2862). IEEE. http://doi.org/10.1109/ACC.2003.1243756

Liu, Y.-X., Shieh, J.-S., Doctor, F., Fan, S.-Z., and Jen, K.-K. (2013). Multivariable Type-2 Self-Organizing Fuzzy Logic Controllers for Regulating Anesthesia with Rule-Base Extraction. In 2013 Conference on Technologies and Applications of Artificial Intelligence (pp. 217–222). IEEE. http://doi.org/10.1109/TAAI.2013.51

Lu, Q., and Mahfouf, M. (2005). A new efficient self-organising fuzzy logic control (SOFLC) algorithm using a dynamic performance in. In World Congress (Vol. 16, p. 1084).

Lu, Q., and Mahfouf, M. (2006). Multivariable Self-organizing fuzzy logic control (SOFLC) using a switching mode linguistic compensator. In 2006 3rd International IEEE Conference Intelligent Systems (pp. 243–249). IEEE. http://doi.org/10.1109/IS.2006.348425

Lu, X., and Liu, M. (2015). A Fuzzy Logic Controller Tuned with PSO for Delta Robot Trajectory Control, 4345–4351.

Lucas, L. A., Centeno, T. M., and Delgado, M. R. (2007). General Type-2 Fuzzy Inference Systems: Analysis, Design and Computational Aspects. In 2007 IEEE International Fuzzy Systems Conference (pp. 1–6). IEEE. http://doi.org/10.1109/FUZZY.2007.4295522

Ma, T.-T. (2007). P–Q decoupled control schemes using fuzzy neural networks for the unified power flow controller. International Journal of Electrical Power and Energy Systems, 29(10), 748–758. http://doi.org/10.1016/j.ijepes.2007.06.019

Mahfouf, M., and Abbod, M. F. (1994). Self-adaptive and self-organising control applied to nonlinear multivariable anaesthesia: a comparative model-based study. Chapter 4 in "Intelligent Control in Biomedicine", Ed. D.A. Linkens, Taylor and Francis Publishers, Pp 79-132.

Mahfouf, M., Chen, M.-Y., Zhang, Q., and Linkens, D. A. (2006). Adaptive weighted particle Swarm multiobjective optimisation and societal reasoning for the design o. In Applications of Large Scale Industrial Systems (Vol. 1, pp. 130–135).

Mahfouf, M., and Linkens, A. D. (1998). Generalized predictive control and bioengineering. T.J. International Ltd., Padstow (UK).

Mahfouf, M., Linkens, D. A., and Abbod, M. F. (2000). Multi-objective genetic optimisation of GPC and SOFLC tuning parameters using a fuzzy-based ranking method. IEE Proceedings - Control Theory and Applications, 147(3), 344–354. http://doi.org/10.1049/ip-cta:20000345

Martínez, R., Castillo, O., and Aguilar, L. T. (2009). Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. Information Sciences, 179(13), 2158–2174. http://doi.org/10.1016/j.ins.2008.12.028

Martinez, R., Castillo, O., Aguilar, L. T., and Rodriguez, A. (2010). Optimization of type-2 fuzzy logic controllers using PSO applied to linear plants. Stud.Comput.Intell, 181–193.

Martinez, R., Rodriguez, A., Castillo, O., and Aguilar, L. T. (2010). Type-2 Fuzzy Logic Controllers Optimization Using Genetic Algoritms and Particle Swarm Optimization. In 2010 IEEE International Conference on Granular Computing (pp. 724–727). IEEE. http://doi.org/10.1109/GrC.2010.43

Mbede, J. B., Melingui, A., Zobo, B. E., Merzouki, R., and Bouamama, B. O. (2012). zSlices based type-2 fuzzy motion control for autonomous robotino mobile robot. In Proceedings of 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (pp. 63–68). IEEE. http://doi.org/10.1109/MESA.2012.6275538

Mendel, J. M. (2001). Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. (N. Prentice-Hall, Upper Saddle River, Ed.).

Mendel, J. M., Hagras, H., Tan, W.-W., Melek, W. W., and Ying, H. (2014). Introduction to Type-2 Fuzzy Logic Control. Hoboken, NJ, USA: John Wiley and Sons, Inc. http://doi.org/10.1002/9781118886540

Mendel, J. M., and John, R. I. B. (2002). Type-2 fuzzy sets made simple. Fuzzy Systems, IEEE Transactions on, 10(2), 117–127. http://doi.org/10.1109/91.995115

Mendez, G. M., Hernández, A., and Castillo-leal, M. (2007). Interval Type-1 Non-Singleton Type-2 TSK Fuzzy Logic Systems Using the Kalman Filter - Back Propagation Hybrid Learning Mechanism, (Foci), 370–374.

Mendonca, C., Griffiths, J., Ateleanu, B., and Collis, R. E. (2003). Hypotension following combined spinal-epidural anaesthesia for Caesarean section. Left lateral position vs. tilted supine position*. Anaesthesia, 58(5), 428–431. http://doi.org/10.1046/j.1365-2044.2003.03090.x

Mendonça, L. F., Vieira, S. M., and Sousa, J. M. C. (2007). Decision tree search methods in fuzzy modeling and classification. International Journal of Approximate Reasoning, 44(2), 106–123. http://doi.org/10.1016/j.ijar.2006.07.004

Mollov, S., Babuska, R., Abonyi, J., and Verbruggen, H. B. (2004). Effective Optimization for Fuzzy Model Predictive Control. IEEE Transactions on Fuzzy Systems, 12(5), 661–675. http://doi.org/10.1109/TFUZZ.2004.834812

Mollov, S., and Robert, B. (2004). Analysis of interactions and multivariate decoupling fuzzy control for a binary distillation column. International Journal of Fuzzy Systems, vol.6, no.2, pp.53-62.

Nie, J., and Linkens. (1995). Fuzzy-neural control : Principles, algorithms and applications. London; New York : Prentice-Hall.

Nie, J., and Linkens, D. A. (1993). A fuzzified CMAC self-learning controller. In [Proceedings 1993] Second IEEE International Conference on Fuzzy Systems (pp. 500–505). IEEE. http://doi.org/10.1109/FUZZY.1993.327518

Nie, M., and Tan, W. W. (2008). Towards an efficient type-reduction method for interval type-2 fuzzy logic systems. In 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence) (pp. 1425–1432). IEEE. http://doi.org/10.1109/FUZZY.2008.4630559

Nyongesa, H. O., and Linkens, D. A. (1995). Genetic algorithms for fuzzy control.2. Online system development and application. IEE Proceedings - Control Theory and Applications, 142(3), 177–185. http://doi.org/10.1049/ip-cta:19951767

Oh, S.-K., Jang, H.-J., and Pedrycz, W. (2011). A comparative experimental study of type-1/type-2 fuzzy cascade controller based on genetic algorithms and particle swarm optimization. Expert Systems with Applications, 38(9), 11217–11229. http://doi.org/10.1016/j.eswa.2011.02.169

Pal, T., Pal, N. R., and Pal, M. (2003). Learning fuzzy rules for controllers with genetic algorithms. International Journal of Intelligent Systems, 18(5), 569–592. http://doi.org/10.1002/int.10104

Persechini, M. A. M., Peres, A. E. C., and Jota, F. G. (2004). Control strategy for a column flotation process. Control Engineering Practice, 12(8), 963–976. http://doi.org/10.1016/j.conengprac.2003.11.003

Polkinghorne, M. N., Roberts, G. ., and Rurns, R. . (1994). A Self-Organising Fuzzy Logic Autopilot for Small Vessels. PhD Thesis, School of Manufacturing, Materials and Mechanical Engineering, University of Plymouth, UK.

Procyk, T. J., and Mamdani, E. H. (1979). A linguistic self-organizing process controller. Automatica, 15(1), 15–30. http://doi.org/10.1016/0005-1098(79)90084-0

Ren, Y., Duan, X., Li, H., Chen, C. L. P., and Rule, M. (2013). Multi-variable fuzzy logic control for a class of distributed parameter systems. Journal of Process Control, 23(3), 351–358. http://doi.org/10.1016/j.jprocont.2012.12.004

Rojas, I., Pomares, H., Pelayo, F. J., Anguita, M., Ros, E., and Prieto, A. (1999). New methodology for the development of adaptive and self-learning fuzzy controllers in real time. International Journal of Approximate Reasoning, 21(2), 109–136. http://doi.org/10.1016/S0888-613X(99)00008-0

Sarimveis.H, and Bafas.G. (2003). Fuzzy model predictive control of non-linear processes using genetic algorithms. Fuzzy Sets and Systems 139, 59-80.

Sepúlveda, R., Castillo, O., Melin, P., and Montiel, O. (2007). Analysis and Design of Intelligent Systems using Soft Computing Techniques. (P. Melin, O. Castillo, E. G. Ramírez, J. Kacprzyk, and W. Pedrycz, Eds.) (Vol. 41). Berlin, Heidelberg: Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-540-72432-2

Shayeghi, H., Jalili, A., and Shayanfar, H. A. (2008). Multi-stage fuzzy load frequency control using PSO. Energy Conversion and Management, 49(10), 2570–2580. http://doi.org/10.1016/j.enconman.2008.05.015

Shieh, J.-S., Kao, M.-H., and Liu, C.-C. (2006). Genetic fuzzy modelling and control of bispectral index (BIS) for general intravenous anaesthesia. Medical Engineering and Physics, 28(2), 134–48. http://doi.org/10.1016/j.medengphy.2005.04.023

Siegel, S. (1956). Non-parametric statistics for the behavioral sciences. New York: McGraw-Hill, 75–83.

Skogestad, S., and Postlethwaite, I. (2005). Multivariable feedback control: analysis and design. International Journal of Robust and Nonlinear Control, 8(14), 575. http://doi.org/978-0-470-01167-6

Starczewski, J. T. (2009a). Efficient triangular type-2 fuzzy logic systems. International Journal of Approximate Reasoning 50(5), 799–811.

Starczewski, J. T. (2009b). Extended triangular norms. Information Sciences 179(6), 742–757.

Stoelting, R. M. D. (2000). Basics of Anesthesia (4th ed.). Churchill Livingstone.

Sugeno, M., and Kang, G. (1988). Structure identification of fuzzy model. Fuzzy Sets and Systems 26(1), 15–33.

Sugeno, M., and Nishida, M. (1985). Fuzzy control of model car. Fuzzy Sets and Systems, 16(2), 103–113. http://doi.org/10.1016/S0165-0114(85)80011-7

Sugeno, M., and Tanaka, K. (1991). Successive identification of a fuzzy model and its application to prediction of a complex system. Fuzzy Sets and Systems 42, 315–334.

Sugeno, M., and Yasukawa, T. (1993). A fuzzy-logic-based approach to qualitative modeling. IEEE Transactions on Fuzzy Systems, 1(1), 7–31. http://doi.org/10.1109/TFUZZ.1993.390281

Sun, Q., and Li, R. (2000). Fuzzy control of multivariable systems based on fuzzy dynamic

model. In Proceedings of the 3rd World Congress on Intelligent Control and Automation (Cat. No.00EX393) (Vol. 3, pp. 1544–1547). IEEE. http://doi.org/10.1109/WCICA.2000.862723

Szabo, A., Castro, L. N. de, and Delgado, M. R. (2011). The proposal of a fuzzy clustering algorithm based on particle swarm. In 2011 Third World Congress on Nature and Biologically Inspired Computing (pp. 459–465). IEEE. http://doi.org/10.1109/NaBIC.2011.6089630

Takagi, T., and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics, SMC-15(1), 116–132. http://doi.org/10.1109/TSMC.1985.6313399

Tu, K. Y., Lee, T. T., and Wang, C. H. (2005). Design of a New Fuzzy Suction Controller Using Fuzzy Modeling for Nonlinear Boundary Layer. IEEE Trans. on Fuzzy Systems, Vol. 13, No. 5, Pp. 605-616.

Wagner, C., and Hagras, H. (2009). zSlices based general type-2 FLC for the control of autonomous mobile robots in real world environments. In 2009 IEEE International Conference on Fuzzy Systems (pp. 718–725). IEEE. http://doi.org/10.1109/FUZZY.2009.5277383

Wagner, C., and Hagras, H. (2010). Toward General Type-2 Fuzzy Logic Systems Based on zSlices. IEEE Transactions on Fuzzy Systems, 18(4), 637–660. http://doi.org/10.1109/TFUZZ.2010.2045386

Wu Woei Wan Tan, D. (2006). A simplified type-2 fuzzy logic controller for real-time control. ISA Transactions, 45(4), 503–516. http://doi.org/10.1016/S0019-0578(07)60228-6

Wu, D., and Mendel, J. M. (2002). Uncertainty Bounds and Their use in the Design of Interval Type-2 Fuzzy Logic Systems. IEEE Transactions on Fuzzy Systems, 10(5), 622–639. http://doi.org/10.1109/TFUZZ.2002.803496

Wu, D., and Mendel, J. M. (2009). Enhanced Karnik-Mendel algorithms. IEEE Transactions on Fuzzy Systems, 17(4), 923–934. http://doi.org/10.1109/TFUZZ.2008.924329

Wu, D., and Mendel, J. M. (2010). Examining the Continuity of Type-1 and Interval Type-2 Fuzzy Logic Systems. IEEE World Congress on Computational Intelligence, 19(x), 179–192.

Wu, D., and Nie, M. (2011). Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems. 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), (6), 2131–2138. http://doi.org/10.1109/FUZZY.2011.6007317

Yager, R. R., and Filey, D. P. (1995). Essentials of Fuzzy Modeling and Control. New York: John Wiley and Sons.

Yamazaki, T. (1982). An improved algoirthm for a self-organising controller and its experimental analysis. PhD Thesis, Queen Mary College, London, Dept. of Electrical and Electornic Engineering.

Yang C. Z. (1992). Design of real-time Linguistic self-organizing fuzzy controller. Department of Mechnical Engineering . Taiwan: National Taiwan University, 1992.

Young-Moon Park, Un-chul Moon, and K. y. L. (1995). A Self-Organizing Fuzzy Logic Controller for Dynamic Systems Using a Fuzzy Auto-Regressive Moving Average (FARMA) Model. Ieee Tranzaction on Fuzzy Systems, Vol.3.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning-I. Information Sciences, 8(3), 199–249. http://doi.org/10.1016/0020-0255(75)90036-5

Zadeh, L. a. (1996). Fuzzy logic equals Computing with words. Ieee Transactions on Fuzzy Systems, 4(2), 103–111.

Zhang, G., Mahfouf, M., Panoutsos, G., and Wang, S. (2012). A multi-objective particle swarm optimization algorithm with a dynamic hypercube archive, mutation and population competition. In *2012 IEEE Congress on Evolutionary Computation* (pp. 1–7). IEEE. http://doi.org/10.1109/CEC.2012.6256489

Zhang, Q., and Mahfouf, M. (2009). A modified PSO with a dynamically varying population and its application to the multi-objective optimal design of alloy steels. In *2009 IEEE Congress on Evolutionary Computation* (pp. 3241–3248). IEEE. http://doi.org/10.1109/CEC.2009.4983355