Intelligent Methods for Complex Systems Control Engineering

Rudwan Ali Abolgasim Abdullah

Department of Computing Science and Mathematics
University of Stirling
Stirling, FK9 4LA
Scotland-UK

This thesis has been submitted to the University of Stirling
in partial fulfilment of the requirements for the
degree of Doctor of Philosophy

2007

Declaration

I hereby declare that this thesis has been composed by myself, that the work and results have not been presented for any university degree prior to this and that the ideas that I do not attribute to others are my own.

Rudwan Abdullah

2007

Dedication

To whoever appreciates this work.

Abstract

This thesis proposes an intelligent multiple-controller framework for complex systems that incorporates a fuzzy logic based switching and tuning supervisor along with a neural network based generalized learning model (GLM). The framework is designed for adaptive control of both Single-Input Single-Output (SISO) and Multi-Input Multi-Output (MIMO) complex systems.

The proposed methodology provides the designer with an automated choice of using either: a conventional Proportional-Integral-Derivative (PID) controller, or a PID structure based (simultaneous) Pole and Zero Placement controller. The switching decisions between the two nonlinear fixed structure controllers is made on the basis of the required performance measure using the fuzzy logic based supervisor operating at the highest level of the system. The fuzzy supervisor is also employed to tune the parameters of the multiple-controller online in order to achieve the desired system performance. The GLM for modelling complex systems assumes that the plant is represented by an equivalent model consisting of a linear time-varying sub-model plus a learning nonlinear sub-model based on Radial Basis Function (RBF) neural network. The proposed control design brings together the dominant advantages of PID controllers (such as simplicity in structure and implementation) and the desirable attributes of Pole and Zero Placement controllers (such as stable set-point tracking and ease of parameters' tuning).

Simulation experiments using real-world nonlinear SISO and MIMO plant models, including realistic nonlinear vehicle models, demonstrate the effectiveness of the intelligent multiple-controller with respect to tracking set-point changes, achieve desired speed of response, prevent system output overshooting and maintain minimum variance input and output signals, whilst penalising excessive control actions.

Acknowledgments

I would like to express my extreme thanks and gratitude to my principal supervisor Dr. Amir Hussain for his invaluable guidance, experience and stimulating research ideas. I wish to thank my second supervisor Dr. Bruce Graham and all others in the department of Computing Science and Mathematics who offered helpful advice and support during my stay in the University of Stirling.

I am also grateful to my friend and colleague Dr. Ali Zayed, who is currently labouring at the Seventh April University in Libya, for his kind support during the early stages of my work and for the fruitful technical discussions we had from time to time.

I would also like to especially acknowledge Professor Kevin Warwick, from the University of Reading, for the stimulating discussions on numerous key aspects of intelligent control (during his visit to the University of Stirling in summer 2006), particularly related to his pioneer general input-output nonlinear modelling framework that form the basis of the GLM.

I express my deep thankfulness to Professor Amin Missallati, the founder of the Biruni Remote Sensing Centre (Libya), for providing me the opportunity to study for a PhD degree. I am also equally grateful to the Biruni Centre for funding my study.

My parents and my wife deserve a special mention and thanks for believing in me and for their patience and prayers so that this humble work came to reality.

Abbreviations

ACC	Adaptive Cruise Control
ANN	Artificial Neural Network
AVC	Autonomous Vehicle Control
AVCS	Advances in Vehicle Control and Safety
BP	Backpropagation
CARMA	Controlled Auto-Regressive Moving Average
DARPA	Defence Advanced Research Project Agency
ETC	Electronic Throttle Control
FL	Fuzzy Logic
FLS	Fuzzy Logic System
GLM	Generalized Learning Model
GMV	Generalised Minimum Variance
GMVC	Generalised Minimum Variance Control
IVHS	Intelligent Vehicle Highway Systems
MIMO	Multi-Input Multi-Output
MLP	Multi-Layer Perceptron
NARX	Nonlinear Auto-Regressive model with eXternal
NN	Neural Network
PATH	Partners for Advanced Transit and Highways
PID	Proportional-Integral-Derivative
RBF	Radial Basis Function
SISO	Single-Input Single-Output
SMVC	Simple Minimum Variance Controller

Table of Contents

List of Figures

Figure **Page**

List of Tables

Chapter 1

Introduction

For complex systems with significant nonlinearity and parametric uncertainty, adaptive nonlinear control has evolved as a powerful methodology leading to global stability and tracking results for a class of nonlinear systems [15]. In parallel with the development in adaptive nonlinear control, there has been a tremendous amount of activity in the application of artificial intelligence techniques in control engineering. The area of intelligent control is in fact interdisciplinary, and it attempts to combine and extend theories and methods from areas such as control engineering, computer science and operation research to attain demanding control goals [61]. Intelligent control techniques are nowadays recognised tools in both academia and industry. Methodologies coming from the field of computation intelligence, such as neural networks, fuzzy logic systems and evolutionary computation, can lead to accommodation of more complex processes, improved performance and considerable time savings and cost reductions [13].

An intelligent controller, based on an expert control concept, as a decision-making element in a feedback control loop requires much the same decision making ability as is needed in other expert systems, however there are significant differences. One crucial requirement is the need to provide control signals to the process in real-time. The second requirement is that the intelligent controller should not need human interaction to complete its functions. The third is that the intelligent controller must be interfaced directly to the process and be equipped with the means for applying control to the process [59]. An intelligent controller based on this type of a controller should be able to use several different control algorithms as well as to tune the parameters of each algorithm according to the desired user specifications. It should also autonomously manage the selection between those control algorithms to maintain the control objectives at or near their optimal values for specific process conditions.

1.1 Motivation for the Thesis

Inspired by the advances and the on going research in soft-computing techniques and their applications in intelligent control engineering, the motivations for this thesis are presented in the following sub-sections.

1.1.1 Control of Complex Systems

During the last few decades, nonlinear control became a field of growing interest. The reason is twofold. Firstly, new achievements of nonlinear control theory combining control engineering, computer science and operation research methods strengthened its power. Secondly, new demands for high performance control arose in science and engineering. Nonlinearity plays an especially strong role in control of mechanical systems such as chemical industry, cars, robots, helicopters, ships, etc. Many of those systems are characterized by a high level of complexity: high dimension of the state space, multiple inputs and outputs, parametric uncertainty and un-modelled dynamics [61, 62].

A difficult problem in the control of these complex dynamic systems is due to the inherent nonlinearities of their models and these problems cannot be solved using traditional linear control techniques. The application of linear control theory to these problems relies on the key assumption of a small range of operation in order for the linear model assumption to be valid. When the required operating range is large, a linear controller may not be adequate. For this reason, it seems appropriate to extend linear control to plants with nonlinear models and with plant/model mismatch. A possible way this can be achieved is by incorporating the inherent nonlinearity of the process into the control design process using a so-called learning model. "Learning models" result from a synthesis of learning systems. Learning systems are particularly useful whenever complete knowledge about the environment is either unknown, expensive to obtain or impossible to quantify. When learning systems are synthesized with modelling techniques, so-called learning models emerge. Furthermore, when learning models are used with advanced control methods, they result in learning control systems [63, 64].

Over the last decade or so, there has been much progress in the modelling and control of complex processes, using black-box type learning models. Some of nonlinear input-output representations have appeared, such as Volterra series, Hammerstein (HM), Wiener, Wiener-Hammerstein, Artificial Neural Networks, etc. Neural networks have been shown to be very effective for controlling complex non-linear systems, when there is no complete model information, or when the controlled plant is considered to be a "black box" [65].

Zhu and Warwick [19] have proposed an improved more generalised method for developing nonlinear adaptive control based on a general input-output nonlinear modelling framework. In such designs, which Zayed *et al.* [37] term the Generalized Learning Model (GLM), the process model can be split into two parts, namely linear and non-linear dynamical learning sub-model, so that this special structure allows the linear part of the controller to exploit classical linear theory. In addition, the coupling effects and the other relationships are accommodated in the nonlinear learning sub-model allowing effective compensation. Therefore, this thesis improves the performance of the GLM to better represent complex real-world systems and develop a novel intelligent framework to cover complex dynamic systems.

1.1.2 Intelligent Methods in Control Engineering

Intelligent control is a very active and multi-disciplinary field [66]. The concept of intelligence in control applies to a variety of approaches used for extending classical control theory that include learning, nonlinear control, model-based control, and, in general, control of complex systems that will achieve the desired performance when confronted with unexpected or unplanned situations [67]. There are requirements today that cannot be successfully addressed with the existing conventional control theory. They mainly pertain to the area of uncertainty, which is present because of poor models due to lack of knowledge, or due to high level models used to avoid excessive computational complexity. Normally the plant is so complex that it is either impossible or inappropriate to describe it with conventional mathematical system models such as differential or difference equations. Even though it might be possible to accurately describe some systems with highly complex nonlinear differential equations, it may be inappropriate if this description makes subsequent analysis too difficult or too computationally complex to be useful. The complexity of the plant model needed in design depends on both the complexity of the physical system and on how demanding the design specifications are [61].

There are needs in the control of these complex systems which cannot be met by conventional approaches to control. For instance, there is a significant need to achieve higher degrees of autonomous operation for robotic systems, spacecraft, manufacturing systems, automotive systems, underwater and land vehicles, and others. To achieve such highly autonomous behaviour for complex systems one can enhance today's control methods using intelligent control systems and techniques [68].

Intelligent control systems are typically able to perform one or more of the following functions to achieve autonomous behaviour: planning actions at different levels of detail, emulation of human expert behaviour, learning from past experiences, integrating sensor information, identifying changes that threaten the system behaviour, such as failures, and reacting appropriately. This identifies the areas of Planning and Expert Systems, Fuzzy Systems, Neural Networks, Machine Learning, Multi-sensor Integration, and Failure Diagnosis, to mention but a few, as existing research areas that are related and important to Intelligent Control [68]. Therefore, in this thesis, soft-computing techniques such as neural networks and fuzzy logic will be used to improve the approximation of the GLM for complex systems and introduce autonomy to the multiple controller framework proposed by Zayed [17].

1.1.3 Multiple-Controller Structures and Techniques

Control engineers are sometimes confronted with situations in which they have to design and implement real-time control systems that are composed of a set of controllers in stead of a single control algorithm. These situations occur for example when the control problem to solve is complex of nature, that is, when the problem can be thought of being composed of an interconnection of a set of simpler sub-problems [8]. A common approach to control such complex dynamic systems is to design a set of different controllers, each of which for a particular operating region or performance objective, and then to switch them in real-time to achieve the overall control objective [69]. Some multiple-controller architectures have been reported in the field of control engineering, and these are known under the general name of *multiple model approach* [8].

Many physical systems are hybrid in the sense that they have continuous behaviours and discrete phenomena. A good example of a complex hybrid system is an automobile [69]. Discrete signals are gear ratios, load and road characteristics, driver inputs, and control signals. The continuous parts are often nonlinear dynamics of motion, motor characteristics, sensor signals, and so on. Continuous dynamic characteristics vary according to the state of discrete signals. Dynamics can be changed by the operator input or due to a change in the environment. Therefore, it is required to implement a different controller for each operating condition. An intelligent multiple control system may have the ability to operate in multiple environments by understanding the current operating condition and achieving the various tasks appropriately.

This research will work towards developing a new intelligent multiple-controller framework which incorporates a fuzzy logic based switching and tuning supervisor to provide the system with the choice between deploying the conventional Proportional-Integral-Derivative (PID) self-tuning controller, or the PID structure based (simultaneous) pole and zero placement controller. Both controllers, which were originally proposed by Zayed *et al.* [17, 39, 70] benefit from the simplicity of having a PID structure, operate using the same adaptive procedure and can be selected on the basis of the required performance measure.

1.2 Aims and Objectives of the Research

The general aim of this thesis is to develop a new intelligent multiple-controller framework for controlling complex Single-Input Single-Output (SISO) and Multi-Input Multi-Output (MIMO) systems. Specific objectives include: (1) Explore soft-computing techniques, such as neural networks and fuzzy logic, in order to further develop the multiple-controller framework proposed by Zayed *et al.* [17, 39, 70]. (2) Improve the approximation capability of the nonlinear sub-model in the GLM for more accurate complex SISO and MIMO plant representation. (3) Introduce autonomy to the manual switching mechanism among the candidate controllers, namely conventional PID controller and Pole-Zero Placement controller. (4) Introduce autonomous online tuning of the parameters of the controllers employed. (5) Apply and assess the developed intelligent multiple-controller framework in novel realistic challenging applications.

1.3 Original Contributions of the Thesis

The main contributions of this thesis are:

1.3.1 Radial Basis Function Neural network Based Enhanced GLM

In modelling nonlinear dynamics and disturbances of complex systems using the GLM, the use of the computationally efficient Radial Basis Function (RBF) neural network (NN) sub-model instead of the Multi-Layered Perceptron (MLP) NN based sub-model, which was originally proposed in [16, 19], has been shown to improve the system performance in terms of achieving minimum variance of the output signal and the control input signal, both for tracking changes in the reference signal and for dealing with the nonlinearities and addition of random and constant disturbances. The enhanced RBF based GLM is applied successfully to SISO and MIMO systems.

1.3.2 Fuzzy Logic based Switching between Multiple Controllers

A fuzzy-logic based switching scheme has been developed in order to introduce autonomy to the conventional manual switching mechanism between multiple controller modes, which was originally adopted in [17, 39, 70]. The fuzzy-logic based supervisor operates at the highest level of the system and makes a switching decision, on the basis of the required performance measure, between two candidate nonlinear controllers, namely a PID controller, or a PID structure based zero and pole placement controller. In general, the need for switching stems from the fact that typically no single controller can guarantee the desired behaviour when connected with the poorly modelled process, and particularly so for the case of complex processes exhibiting significant nonlinearity, non-stationarity, uncertainty and/or multi-variable interactions [50].

1.3.3 On-Line Fuzzy Logic based Tuning of Multiple-Controller Parameters

In complex control systems, large changes in the operating state lead to corresponding variations in the parameters of the plant model about these operating states. It is well known that it is not possible therefore to design a controller to operate satisfactorily at one operating state and expect it to perform equally well elsewhere without re-tuning it. Closed loop system performance is degraded since the controller cannot track the changes in the operating states [74]. In order to study the sensitivity of tuning the multiple-controller parameters for achieving the desired performance, an online parameter tuning strategy has been proposed. Therefore, the aim of this work is to apply the fuzzy-logic supervisor to tune the parameters of the multiple-controller online, including the poles and zeros of the (simultaneous) pole-zero placement controller in addition to the PID gains. This novel tuning strategy builds on the conventional fuzzy gain scheduling strategies that have been conventionally employed for only PID controllers [4, 71, 72].

1.3.4 New Intelligent Multiple Controller Framework for Complex Systems

The work in this research is integrated to develop an intelligent multiple-controller framework for the control of SISO and MIMO complex systems. In the proposed approach, a switching and tuning fuzzy-logic supervisor is situated at the highest level of the system in order to govern the selection scheme among the conventional PID and pole-zero placement nonlinear controllers, and to perform the required changes on the parameters of the active controller. To model the behaviour of the plant, the framework incorporates the RBF neural network based GLM. The GLM assumes that the unknown complex plant is represented by an equivalent stochastic model consisting of a linear time-varying sub-model plus a nonlinear RBF neural network learning sub-model.

1.3.5 Novel Application of the Proposed Intelligent Multiple-Controller to Autonomous Vehicle Control

The field of autonomous vehicles is a rapidly growing one with its abundant applications of electronics, sensors, actuators, and microprocessor based control systems to provide improved performance, fuel economy, emission levels, comfort, and safety. Autonomous vehicles represent the intersection of artificial intelligence and robotics, combining decision-making with real-time control [76]. An important component of autonomous vehicle control (AVC) is to design control systems for controlling the throttle, wheel brake and steering systems so that the vehicle can follow a desired path and target speed, which could be a speed response of a leading vehicle, and at the same time keep a safe inter-vehicle spacing under the constraint of comfortable driving [77]. There are though a lot of possible techniques with which to perform AVC. Conventional methods based on analytical control generate good results but exhibit high design and computational costs since the application object is a complex nonlinear element and a complete mathematical representation is impossible. Therefore, other ways of reaching human-like vehicle control have been developed, for example, through the application of artificial intelligence techniques [78].

One important and challenging problem in AVC development from the real-time control applications is vehicle subsystems integration [76]. Dangerous yaw motions of the automobile may result from unexpected yaw-disturbances caused by unsymmetrical car-dynamics perturbations like side-wind forces, unilateral loss of tire pressure or braking on unilateral icy road. One approach for yaw dynamics improvement is to use individual wheel braking, thereby creating the moment that is necessary to counteract the undesired yaw motion. Another approach is to command additional steering angles to create the counteracting moment [79]. Another alternative approach, which is suggested in this work, is to treat the three drivetrain sub-systems (i.e., throttle, brake and steering sub-systems) as one MIMO plant. In this work, the interactions between the vehicle longitudinal and lateral properties, disturbances and nonlinearities are considered in the multivariable MIMO control law and modelled using the MIMO neural network employed in the so called GLM. The application of the proposed intelligent multivariable multiple-controller framework to the autonomous vehicle control problem was shown to manage the control of the longitudinal and lateral complex vehicle model in order to track target speed and path trajectories.

1.4 Publications Arising

The following nine papers have been resulted from the research:

1.4.1 Journal Papers

[1] R. Abdullah, A. Hussain, K. Warwick and A. Zayed, Autonomous Intelligent Vehicle Control using a Novel Multiple-Controller Framework Incorporating Fuzzy-Logic based Switching and Tuning, *Neurocomputing (Elsevier Science)*, 70, *in press*, 2007.

[2] A. Zayed, A. Hussain and R. Abdullah, A Novel Multiple-Controller Incorporating a Radial Basis Function Neural Network based Generalized Learning Model, *Neurocomputing (Elsevier Science)*, 69 (16), 1868-1881, 2006.

1.4.2 Refereed International Conference Proceedings

[1] R. Abdullah, A. Hussain and M. Polycarpou, Fuzzy Logic based Switching and Tuning Supervisor for a Multivariable Multiple-Controller, *IEEE International conference on Fuzzy Systems (FUZZ-IEEE 2007)*, 1644-1649, Imperial College, London, UK, 23-26 July, 2007.

[2] R. Abdullah, A. Hussain and A. Zayed, Novel Intelligent Multiple Controller Framework incorporating Fuzzy Logic based Switching, *In Proceedings: International Control Conference (ICC 2006)*, Glasgow-UK, No. 252, 29 Aug. -1 Sep. 2006.

[3] R. Abdullah, A. Hussain and A. Zayed, A New Radial Basis Function Neural Network Based Multi-variable Adaptive Pole-Zero Placement Controller, *In Proceedings of the IEEE International Conference on Engineering of Intelligent Systems (IEEE ICEIS'2006)*, Islamabad-Pakistan, 22-23 April 2006.

1.4.3 Invited Papers in International Workshops

[1] R. Abdullah and A. Hussain, Intelligent and Nonlinear Control Design for Autonomous Vehicle Systems, *Workshop on Advances in Real-Time Control for Nonlinear Systems*, Glasgow, UK, 5-7 Sep. 2007.

[2] R. Abdullah and A. Hussain , A New Intelligent Multiple Controller Framework for Complex Systems Deploying Fuzzy Supervisory Switching and Tuning, *Workshop on Nonlinear Control Design for Industrial Applications*, Glasgow, UK, 5-6 April 2006.

[3] R. Abdullah A. Hussain, and A. Zayed , New Intelligent Methods for Nonlinear Adaptive Control and System Identification, *Workshop on Introduction to Nonlinear Control Techniques for Industrial Processes*, Glasgow, UK, 3-4 November 2004.

1.4.4 Book Chapters

[1] R. Abdullah, A. Hussain and A. Zayed, A New RBF Neural Network Based Non-linear Self-tuning Pole-Zero Placement Controller, Lecture Notes in Computer Science (LNCS), Springer-Verlag, Chapter p, Vol.3697, 951-956, 2005.

1.5 Thesis Layout

This section provides a chapter-by-chapter guide to the thesis.

Chapter two provides a background to the intelligent control systems. The review includes an overview of the general requirements and structure of intelligent control systems. As related topics to the developments reported in this thesis, the issues of complex systems identification based on neural networks, and using fuzzy-logic in supervisory control will be discussed.

The third chapter is devoted to illustrate the concept of multiple-controller approach in control engineering. The focus will be on the multiple-controller framework proposed in [17, 39, 70]. A detailed discussion regarding the framework's three multiple-controller modes is given, namely: conventional PID controller; pole placement controller; and (simultaneous) pole-zero placement controller. Finally, the design of the GLM is presented.

Chapter four introduces intelligence to the conventional SISO multiple-controller presented in chapter three. The new developments presented here include the enhanced RBF based GLM and the fuzzy-logic based switching and tuning supervisor. These developments will form part of the new intelligent multiple-controller framework for controlling complex system in order to achieve a more effective control action and to over come certain limitations exhibited in the original design of [17, 39, 70].

Chapter five is an extension to the work proposed in the previous chapter as the intelligent multiple-controller for SISO systems is extended to cover complex MIMO plants. The theoretical stability analysis of the proposed intelligent multiple-controller framework is given also.

Chapter six presents results of a series of simulation experiments. The first experiment starts with a common control problem occurring in chemical process industries, namely the control of fluid levels in real-world SISO storage tanks or reaction vessels. In this case, the proposed intelligent framework will work on tracking a reference signal, which represents the target fluid level, in addition to dealing with the systems' nonlinear dynamics and added disturbances. Finally, this experiment is extended to a coupled tanks MIMO system that comprises one container with a centre partition to divide the container into two tanks. In order to compare the performance of the new RBF based GLM, with that proposed in [16, 17], a quantitative measure of variance of the controller output and the control input signals is used to assess the closed-loop system performance. The second experiment carried out in this thesis exhibit the new application of the intelligent multiple-controller framework to the challenging problem of autonomous vehicle control. The proposed methodology is used to simultaneously control the throttle, brake and steering subsystems of a validated complex nonlinear vehicle model. This multivariable problem will demonstrate the effectiveness of the intelligent controller with respect to tracking desired longitudinal and lateral displacements, vehicle speed changes and achieving the desired speed of response, whilst penalising excessive control actions.

Finally, concluding remarks are given in chapter seven together with some recommendations for future work.

Chapter 2

Background to Intelligent Control Systems

2.1 Introduction

Considerable research is being devoted to an understanding and a representation of intelligence, and the development of intelligent learning algorithms that can be applied to technological problems whose complexity defies conventional algebraic model based solutions. To have a resultant intelligence, a computer needs to be able to sense the world in some way and then have the ability to carry out physical actions, or to cause physical actions to be carried out [158]. In intelligent or learning systems a desired mapping f may be determined by either of the following techniques [1]:

1- Indirectly from the set of inputs via self organisation or by the presentation of training examples.

2- A direct specification of $f(.)$ through a functional transform algorithm, a look up table or a rule base.

Intelligent systems are frequently characterised by the estimation of the mapping $f(.)$ from experiential evidence via an associative memory without recourse to mathematical modelling or specification. Intelligent systems naturally involve learning or adaptation of the systems associational structure or functional dependencies in response to changes in process parameters [13]. That is, learning cannot occur, without variation in process parameters, and equally variations in process parameters lead to new experience and different functional dependencies which allow learning.

An important concept within connectionist modules is that of local receptor fields in the input space and its associated response in the output space. It is highly desirable that similar inputs produce similar outputs to minimise sensitivity to input errors, but more importantly to ensure that the behavioural response exceeds the specific inputs. This is called generalisation, allowing a module to generalise beyond specific examples, which is a common feature of intelligent systems [15].

2.2 Intelligent Control Requirements

The ever increasing complexity of dynamical systems coupled with the increasing demands in closed loop performance specification necessitates the use of more complex and sophisticated controllers, yet as systems become more complex uncertainty in modelling increases. Intelligent controllers are enhanced adaptive or self-organising controllers that can accommodate significant changes in the plant and its environment, whilst meeting increasingly stringent controller specifications [73]. Intelligent control systems are defined as those that can operate successfully in a wide variety of situations by detecting the specific situation that exists at any instant and serving it appropriately [21]. External disturbances, changes in sub-system dynamics, parameter variations, etc., are examples of different unknown environments in which the system has to operate. Since environments can change rapidly, the objective is to achieve fast and sufficiently accurate adaptation [2]. Central to intelligent control is the construction of an internal model of the true system processes. Unfortunately many large complex processes are not amenable to mathematical modelling based upon simple physical laws as the process may be [1, 3, 15]:

1- Too complex to understand or present simply.

2- The models are difficult or expensive to evaluate. Variables may not be easily measured or causal variables may not even be known.

3- The process is subject to large unpredictable environment disturbances.

4- The model structure may not be amenable to simple linear time invariant modelling, being distributed, nonlinear and time varying.

Yet many such plants are first regulated or manually tuned by human operators before automatic controllers are installed. The plant operator has few apparent problems with plant nonlinearities or adjusting to slow parametric changes in the plant or with satisfying a set of complex static and dynamic process constraints. The human operator is able to respond to complex sets of observations and constraints, and to satisfy multiple subjective based performance criteria. However, the control actions of human are difficult to analyse as they are variable and subjective, prone to error, inconsistent and unreliable, and in the case of safety critical situations and hazardous processes, potentially dangerous.

The purpose of intelligent control is to incorporate the positive intelligent and creative attributes of human controllers, whilst avoiding the elements of inconsistency, unreliability, temporal instability, fatigue, which are associated with the human conditions. Therefore, intelligent controllers should have the following characteristics [1]:

1- Perform under significant uncertainties in the system and in the environment in which it operates.

2- Able to compensate for system failures without external interventions.

3- Sufficiently adaptable to deal with unexpected situations, new control tasks or changes in the control objectives.

2.3 Intelligent Control Architecture

A variety of architectures that support and integrate intelligent control have emerged [9, 10], amongst which the most natural is the hierarchal and functional architecture shown in figure (2.1). Within this intelligent autonomous control architecture there is an interface to the process involving sensing (e.g., via conventional sensing technology, vision, touch, smell, etc.), actuation (e.g., via hydraulics, robotics, motors, heaters, etc.), and an interface to humans (e.g., a driver, pilot, crew, etc.). Higher levels involve imprecise conceptual qualitative reasoning operating in non-real-time, whereas at lower levels more precise and quantitative algorithmic methods dominate in real-time. Furthermore, there is increasing intelligence accompanied by decreasing precision as one moves from lower to the higher levels [4].

The execution level consists of low-level numeric signal processing and control algorithms. These algorithms could be PID, adaptive, or intelligent control; parameter estimators, failure detection and identification algorithms. The coordination level provides for tuning, scheduling, supervision, and redesign of the execution-level algorithms. Moreover, it provides for crisis management, planning and learning capabilities for the coordination of execution-level tasks, and higher-level symbolic decision making for identification and control management. The management level provides for the supervision of lower-level functions and for managing the interface to the human and other systems. The human interface can be quite complex. It could allow the user to monitor all aspects of operation of the system via graphical user interface, provide the user with information about the overall health of the system, and work the user to specify reasonable and achievable goals for automation [12].

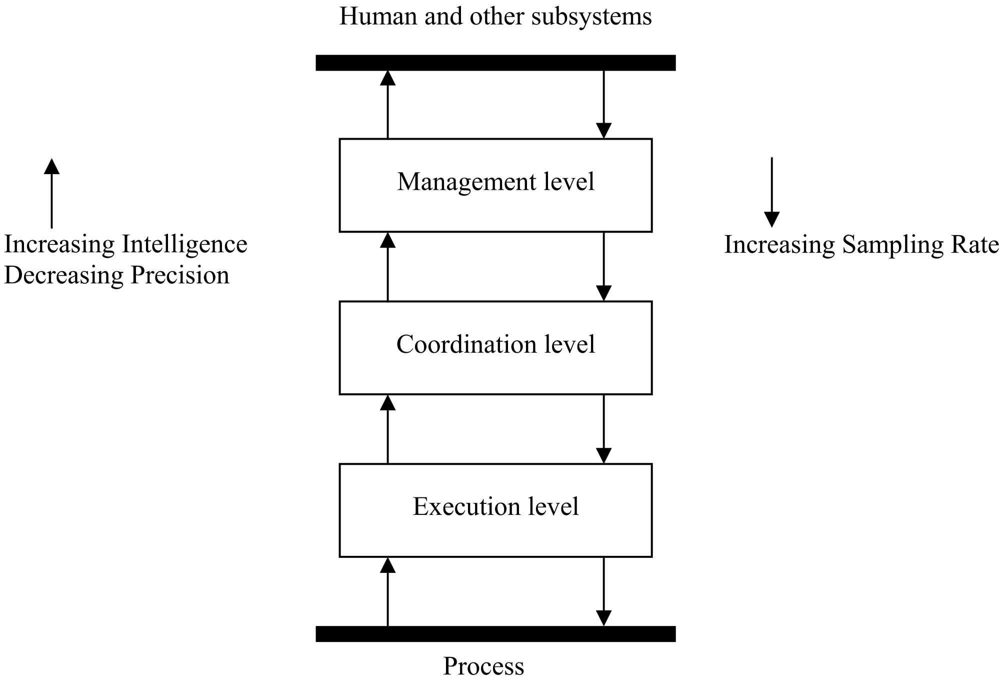


Figure (2.1): Three level intelligent control architecture

2.4 Review of Complex Systems

In engineering and mathematics, a dynamical system is a deterministic process in which a function's value changes over time according to a rule that is defined in terms of the function's current value. In general, there are two kinds of dynamical systems: discrete and continuous. A discrete dynamical system involves step-by-step state changes, and time is measured in discrete steps. If time is measured continuously, the resulting system is considered continuous dynamical system [20].

Most dynamical systems encountered in practice are inherently nonlinear. The control system design process build on the concept of a model. Nonlinearity and model accuracy directly affect the achievable control system performance. Nonlinearity can impose hard constraints on achievable performance. When portions of the plant model are unknown or inaccurately defined, or they change operation, the control performance may need to be severely limited to ensure safe operation. Therefore, there is often an interest to improve the model accuracy. Especially in tracking applications this will typically necessitate the use of complex models [15].

2.4.1 Complex System Identification

The ability to adapt to unknown operating conditions is an important attribute of intelligent systems. Adaptive control is a promising technique to obtain a model of the plant and its environment from experimental data and to design a controller. Adaptive control for a feedback linearizable nonlinear system has attracted much interest among control system designers over several decades [69]. However, the desired level of performance or tracking problems with a sufficiently large operating region may require in which the nonlinearity be directly addressed in the control system design [18]. If the exact knowledge of the system is available, it is possible to transform a nonlinear adaptive control problem into a linear control problem by using a feedback linearization technique. However, in many cases, the plant to be controlled is too complex to obtain the exact system dynamics, and the operating conditions in dynamic environments may be unexpected [69]. Recently, an adaptive control technique has been combined with function approximators such as neural networks, fuzzy inference systems, and series expansion. These types of controllers take the capability of learning unknown nonlinear functions by universal approximation theorem and massive parallel computation. Based on the fact that universal approximators are capable of uniformly approximating a given nonlinear function over a compact set to any degree of accuracy, a globally stable adaptive controller has been developed with an adaptation algorithm [15].

A nonlinear time-invariant dynamic system with single-input, single-output (SISO) can be represented by the following equation [13]:

$$y(t) = S(u(t)), \qquad y(t) \in \mathbb{R}, \qquad u(t) \in \mathbb{R}, \tag{2.1}$$

where t denotes continuous time and S is an operator relating the input signal $u(t)$ to the output signal $y(t)$. The mathematical description of the system itself is unknown, but it is assumed that input-output data are available. The input and the output are sampled at a constant rate, resulting in discrete-time signals denoted by $u(t)$ and $y(t)$. The system described in equation (2.1) can then be approximated by a SISO Nonlinear Auto-Regressive model with eXternal input (NARX) [13, 14]:

$$\hat{y}(k+1) = f(\mathbf{x}(k)), \tag{2.2}$$

where *f* is an unknown approximation function, the hat denotes approximation and $\mathbf{x}(t)$ is the regression vector defined as the collection of previous process inputs and outputs:

$$\mathbf{x}(t) = [y_1(k),...,y_1(k-n_1+1),u_1(k),...,u_1(k-m_1+1)]^{\mathrm{T}}. \tag{2.3}$$

The parameters n_1 and m_1 are integers related to the dynamic order of system (2.1). Further, denote $n = n_1 + m_1$ the dimension of the regression vector. To properly account for noise disturbances, such as sensor noise; process noise; etc., more complicated model structures can be chosen. Some common examples are the nonlinear output error (NOE) model, which involves the past model predictions instead of the process output data:

$$\mathbf{x}(t) = [\hat{y}_1(k),...,\hat{y}_1(k-n_1+1),u_1(k),...,u_1(k-m_1+1)]^{\mathrm{T}}. \tag{2.4}$$

In the Auto-Regressive Moving Average with eXternal input (NARMAX) model the prediction error $e(k) = y(k) - \hat{y}(k)$ and its past values are included in the regression vector as well:

$$\begin{aligned}\mathbf{x}(t) = [&y_1(k),...,y_1(k-n_1+1),u_1(k),...,u_1(k-m_1+1)\\ &e(k),...,e(k-n_e)]^{\mathrm{T}}.\end{aligned} \tag{2.5}$$

The problem of complex system identification is to infer the unknown function *f* in the system equation (2.2) from the sampled data sequences $\{(u(k),y(k))|k = 1,2,\ldots,N\}$. Depending on the type of nonlinearity and the manner that the nonlinearity affects the system, various nonlinear control methods are available [14, 15]. The system designer can use general function approximators such as Artificial Neural Networks (ANNs), Fuzzy Logic systems, splines, interpolated look-up tables, etc. [13]. In particular, ANN-supported complex dynamic system modelling and control has been well advocated in the control engineering community. The introduction of neural networks to nonlinear control system design has significantly relieved the difficulty in resolving mathematical solutions in theory and tuning controller parameters in practice [19].

In this thesis, the complex plant is modelled using a neural network based Generalised Learning Model (GLM) proposed by [16, 17] as part of the Multiple-Controller Framework, which will be discussed in chapter 3.

2.4.2 Complex System Identification with Neural Networks

The aim of systems identification is to determine models from experimental data. The identified models can then be used for different objectives, such as prediction, simulation, optimization, analysis, control, fault detection, etc. Artificial Neural networks, in the context of system identification, are black-box models, meaning that both the model parameters and the model structure are determined from data [13].

ANNs are typically constructed from layers of simple computation nodes, with weighting elements between nodes that define the strength of connection between nodes, which are adapted during learning by some optimisation procedure to yield the appropriate input/output map. ANNs are specified by the network topology, node function characteristics, and the associated learning rules that update the weighting elements [16, 1]. Neural networks are endowed a number of unique attributes which make them potentially suitable for intelligent control [18, 19]:

ANNs learn by experience rather than programming.

They have the ability to generalise, that is, map similar inputs to similar outputs.

They can form arbitrary continuous nonlinear mappings.

Their architectures are distributed, inherently parallel and potentially real time.

For intelligent control or neurocontrol, ANNs require the additional properties of [15]:

Temporal stability, the stability to absorb new information whilst retaining knowledge previously encoded across the network.

Real time adaptation or learning in response to plant variations.

Known or proven learning convergence conditions necessary for process closed loop behaviour prediction or neurocontroller certification.

For neurocontrol, networks that can perform functional approximations are most useful. An important class of neural networks are those in which the input feeds forward through the network layers to the output, these are referred to as feedforward networks. They are able to learn complex input-output functional mappings which are ideal for the purposes of system identification, modelling and control of complex processes [48]. The most popular ANNs in neurocontrol are the Multi-Layer Perceptron (MLP), and the Radial Basis Function (RBF) [21, 22, 23]. Both MLP NNs and RBF NNs are able to adaptively model or identify a dynamical nonlinear Multiple-Input Multiple-output (MIMO) or SISO process/plant on-line while the process is changing [23].

2.4.3 MLP and RBF NNs Function Approximation

Based on Weierstrass approximation theorem [1, 12, 15], for the domain D of a compact space of n-dimensions, let $C(D)$ be a set of continuous functions on D, with metric $\| . \|$, for every $f \in C(D)$ and given error $\varepsilon > 0$ there exists an approximation function $\hat{f}$ such that $\left\| f - \hat{f} \right\| < \varepsilon$.

The MLP may be represented by the network set for m hidden units by

$$NS_1 : \{ f \in C(D) / f(\mathbf{x}) = \sum_{j=1}^{m} a_j \Phi(\sum_{k}^{n} x_k w_{kj} + w_{j_0}) w_{kj}, a_j, w_{j_0} \in \Re \} \tag{2.6}$$

where $\sum_{k}^{n} x_k w_{kj} = \mathbf{x}^T \mathbf{w}_j$; a_j are constants , w_{ki} are adaptive weights, and $\Phi(.)$ are invertible, differentiable squashing functions (such as sigmoid functions). The invertibility and differentiability conditions on $\Phi(.)$ are required to ensure backward error propagation through $\Phi(.)$ to update the weights w_j.

Whereas the associative memory RBF ANN can be represented by the network set

$$NS_2 : \{ f \in C(D) / f(\mathbf{x}) = \sum_{j=1}^{m} w_j \Psi_j \left(\left\| \mathbf{x} - c_j \right\| \right), w_j \in \Re, a_j, c_j \in \Re^n \} \tag{2.7}$$

where c_j are fixed centroids of the basis function $\Psi_j(.)$, and w_j are adaptive weights.

For the network sets NS_1 and NS_2 to be dense in the compact domain D (i.e. any element in D can be approximated by some element from NS_1 and NS_2 with as small an error as desired so that $\| f - \hat{f} \| \leq \varepsilon$ for arbitrary ε) are dependent upon the choice of $\Phi(.)$ and $\Psi(.)$ [1, 18]. The functions in NS_1 and NS_2 can be computed by arbitrary large decaying exponential networks on D.

Since *sinusoids* can be expressed through transforms (such as $cos\,(a+\beta) = cos\ a\ cos\,\beta$ - $sin\ a\ sin\,\beta$) that converts multiplication into addition, and *sinusoids* can be written in terms of exponentials, this makes the MLP NN, which is based on NS_1, dense in the domain D. In this context, the *NS₁*, family of $cosig(.)$ squashing functions

$$\Phi(x) = cos(x) = \begin{cases} 0 & x \leq -0.5 \\ 1+\frac{1}{2}cos(2\pi x) & -0.5 < x < 0 \\ 1 & x \geq 0 \end{cases} \qquad (2.8)$$

are dense in D since the derivative of $cosig(.)$ is zero outside the interval $[-0.5, 0]$, therefore only a small subset of neurons local to this function are updated by back propagation, avoiding temporal instability or learning corruption from previous information.

The Gaussian basis functions are used in RBF NNs

$$\Psi(x) = exp\left(-\frac{1}{2} \frac{(\mathbf{x} - d_j)^2}{\gamma_j^{\ 2}} \right) \qquad (2.9)$$

where $\{d_j\}_{j=1}^N$ are a set of centre locations which are defined for each node j, $(\mathbf{x} - d_j)$ will serve as $\left\| \mathbf{x} - d_j \right\|$ which is the norm of distance from the evaluation point to the j^{th} node centre, and $\{\gamma_j\}_{j=1}^N$ are a set of free parameters that determine the width or region of influence of the kernel functions around the centre of the nodes (i.e. radiuses for the N basis functions) [1] and [15]. Since the approximation of the RBF NNs is linear in the parameters, **w**, any quadratic cost functional used in minimising the approximation error through weights update will have a unique global minimum. In addition, only those weights directly related to new information are updated. It is therefore concluded that the class of artificial neural networks in the form NS_2 provide unique best approximation, which are stable if $\Psi(.)$ have compact support on D.

It is stated that the MLPs are slow, convergence cannot be established, and increasing the state space covered by the training set results in the whole network being retrained. Usually, nonlinear models use nonlinear approximation methods, such as the stochastic approximation algorithm, based upon gradient descent optimisation ensuring convergence to a local minima that contains the initial parameter vector in its attraction basin [48, 49, 107]. The MLP is highly nonlinear in the adjustable weights or parameters, generating a complex cost functional surface in the weight space, with many local minima which traps gradient descent rules. Additionally, MLPs are temporally unstable.

Associative memory single layer networks such as RBF NN satisfy, in some measure, neurocontroller functional conditions 1 to 8 in section mentioned in section 2.4.2 above.

2.5 Review of Neurocontrol: ANN based Control

Multilayer Perceptrons and Radial Basis Functions, trained with the back-propagation algorithm, have been applied successfully in a variety of control applications. To control a system is to make it behave in a desired manner. The behaviour of this system depends on the task to be solved, the dynamics of the system, the actuators, the measurement equipment, the available computational power, and so on. These factors influence the formulation of the desired behaviour as well. There are two basic formulation of the desired behaviour are accepted [24]:

The closed-loop system, consisting of controller and system to be controlled, should follow a prescribed transfer function model. This class of design methods comprises well-known strategies like pole placement and model-reference controllers.

Express the desired behaviour in terms of a quadratic criterion and derive the controller as the minimiser of this criterion. Examples are minimum variance, predictive, and optimal control.

Often it is preferable to formulate the behaviour in terms of time domain characteristics such as; steady state error, degree of overshoot, rise and fall time, variance of the control signal. These types of characteristics can often be handled under linear conditions. Unfortunately, they are in general hard to satisfy for unknown complex dynamic systems.

Neural networks have been used as a tool for modelling complex dynamic systems due to their ability to map complicated output and input nonlinear relationships sufficiently.

Using back-propagation learning algorithm, neural networks have presented a popular architecture in many research fields, including complex system identification and control [16]. Recent results have indicated that neural networks can exactly match input and output signal behaviours [19].

A neural network based system model can be used in the design of a controller or can become a part of a model-based control scheme. The next subsections will present a review of neurocontrol methods.

2.5.1 Gain Scheduling

In classical gain scheduling, slow varying scheduling variables are used to capture nonlinearities and parameter dependencies. The control law is obtained by interpolating a number of locally valid linear controllers. In the context of neural network based control, gain-scheduled control is obtained when using a neurocontroller, usually designed on the basis of an MLP or RBF neural network model of the plant. Applications of this approach can be found in [25] where RBFs are used as a gain-scheduling controller for the lateral motion of a propulsion controlled aircraft, and in [26] where a neural network is used to improve the performance of a classic continuous parameter gain-scheduling controller.

2.5.2 Inverse Control

A straightforward application of neural network based design of a controller for complex process is inverse control [13]. This approach can be explained using the SISO model

$$y(k+1) = f(\mathbf{x}(k), u(k)). \tag{2.10}$$

The vector $\mathbf{x}(k) = [y(k),...,y(k-n_1+1), u(k-1),...,u(k-m_1+1)]^T$ denotes the actual state and thus it does not include the input $u(k)$, such that the system's output at the next sampling instant is equal to the desired (reference) output $r(k+1)$. This can be achieved if the process model (2.10) can be inverted according to:

$$u(k) = f^{-1}(\mathbf{x}(k), r(k+1)). \tag{2.11}$$

Generally, it is difficult to find the inverse function f^{-1} in an analytical form. It can however, always be found by numerical optimization, using the following objective function:

$$J(u(k)) = [r(k+1) - f(\mathbf{x}(k), u(k))]^2. \tag{2.12}$$

The minimisation of J with respect to $u(k)$ gives the control corresponding to the inverse function (2.11) [13]. Given this control scheme, neural controller is considered in order to let the output $y(k)$ of the plant track the reference input $r(k)$ with the above function

$$\min_{w_i^l} J = \sum_k J(u(k)) \tag{2.13}$$

where w_i^l is the i^{th} interconnection weight at layer l of the neural controller [14]. Important neural network application can be found in [21, 27, 28, 29].

2.5.3 Internal Model Control

This class of neurocontrol is model-based control strategy. The difference between the actual plant output and the output of the neural network based model is fed back into the control scheme. In this method it is assumed that the closed loop system is stable [13, 14, 24, 30]. This technique was used for the control of a bioreactor in [31] and in [32] in a design procedure of neural internal controller for stable processes with delay.

2.5.4 Model Based Predictive Control

In model predictive control, a neural network model provides predictions of the future plant response over a specified time horizon. Predictions supplied by the network are passed to a numerical optimization routine in order to minimize an objective cost function subject to the dynamical system model. The system's neural controller is then trained to produce the same control signal for given plant output [14]. Reference [33] used this kind on neurocontrol for chemical process.

2.5.5 Controller Tuning

Neural networks have been used to tune the parameters of different kinds of conventional controllers with a given known structure. References [34, 35] used neural networks to automatically tune the gains of a PID controller.

2.5.6 Generalized Learning Model (GLM) for identification of Complex Plants

Zhu *et al.* [16] introduced a neural network based control structure such that the unknown complex plant is represented by an equivalent model consisting of a simple linear sub-model plus a nonlinear sub-model. The parameters of the linear sub-model are identified by a standard recursive algorithm, whereas the nonlinear sub-model is identified by Back Propagation Neural Network (BPNN), in which the weights are updated based on the error between the plant outputs and the outputs of the linear sub-model. The simple linear dynamic sub-model is used to approximate the dominant dynamics of a wide range of linear and nonlinear dynamic plants around their operating points. As an error agent (nonlinear sub-model), the BPNN is used to learn the errors from the linear sub-model that are due to nonlinearities, uncertainties and disturbances in the controlled plant. The benefits of using a combined model structure come from the fact that the self-tuning control design mechanisms developed from linear model descriptions can be directly expanded to nonlinear dynamic models, and the optimal performance derived from a self-tuning methodology can be directly implanted into the control law [19].

In general, a wide range of complex dynamic plants can be described by a discrete time equation [16]

$$y(t+1) = f(Y,U)\,, \tag{2.14}$$

where $f(Y,U) \to R^n$; $\{Y \in R^{n_y}; U \in R^{n_u}; n = n_y + n_u\}$ is smooth nonlinear function, and $y(t) \in Y$ and $u(t) \in U$ are the plant output and input signals respectively at discrete times $t \in 1,2,\ldots$.To control such a plant, a generalised parametric time-varying plant model is used [16, 17, 37, 38, 39]

$$A(z^{-1})y(t+1) = B(z^{-1})u(t) + f_{0,t}(Y,U) + \xi(t+1)\,, \tag{2.15}$$

where $A(z^{-1})$ and $B(z^{-1})$ are polynomials with orders n_y and n_u, z^{-1} is a one-step backward shift operator. Also, it is considered that the parameters associated with $A(z^{-1})$ and $B(z^{-1})$ are either time invariant or are slow time varying. The system disturbances are represented by the function $\xi(t+1)$ which is an uncorrelated sequence of random variables with zero mean at the sampling instant t. $f_{0,t}(Y,U) \rightarrow R^n$ is potentially a time-varying nonlinear function, and therefore the equivalent model is a combination of a linear time-varying function plus a nonlinear time-varying error agent. Therefore, the overall plant model represented by equation (2.15) above, is termed the Generalized Learning Model (GLM) [37], and can be seen as the combination of a linear sub-model and a non-linear (learning) sub-model as shown in Figure (2.2).

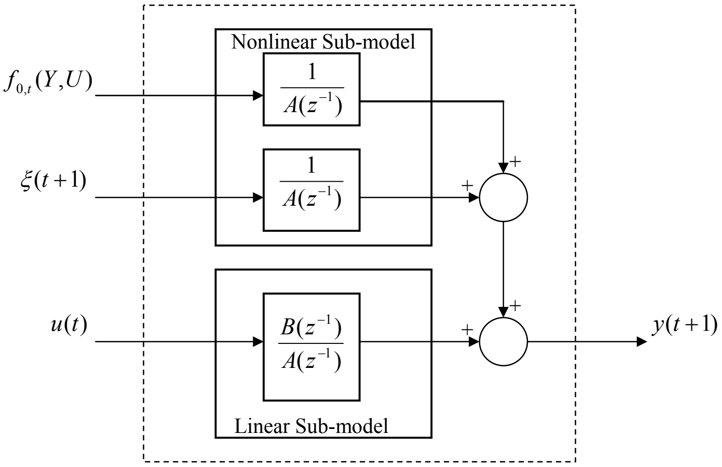


Figure (2.2): Generalized Learning Model for complex systems Identification.

2.6 Review of Fuzzy Logic for Supervisory Control

The idea of using intelligent systems to supervise adaptive control systems was first introduced in [7] and is also reported in [6, 4]. The supervisor can use any available data from the control system to characterize the system's current behaviour so that it knows how to change the controller and ultimately achieve the desired specifications. Moreover, the supervisor can be used to integrate other information into the control decision-making process. It can incorporate certain user inputs, or inputs from other subsystems. For example, in an automotive vehicle control problem, the other subsystem information that a supervisor could incorporate for supervisory control for an automotive vehicle control application could include data from the engine that would help integrate the controls on the vehicle [4].

Most controllers in operation today have been developed using conventional control methods. There are, however, many situations where these controllers are not properly tuned and there is heuristic knowledge available on how to tune them while they are on operation. This knowledge facilitated the opportunity to utilize fuzzy-logic methods as the supervisor that tunes and coordinates the application on conventional controllers [6].

2.6.1 Fuzzy Tuning of PID Controllers

Majority of the controllers in operation today are PID controllers. The popularity of PID controllers is because they are robust to control a wide range of processes, simplicity of their structure, easy to understand and easy to implement [72, 4, 17]. On the other hand, many of the PID loops that are in operation are in continual need of monitoring and adjustment since they can easily become improperly tuned, that is due to plant parameter variations or operation condition changes. Therefore, there is a significant need to develop methods for automatic tuning of PID controllers. In literature, such as [80, 81, 13, 4], there exists many conventional methods for PID auto-tuning. Beginning with Ziegler and Nichols's work [85], various parameters tuning methods for conventional PID controllers have been proposed [86, 87]. Ever since fuzzy theories are proposed by Zadeh in [88], fuzzy logic has gradually adopted as one of major approaches for controller design [86]. There have been numerous articles investigating different schemes of applying fuzzy logic to the design of PID controllers, which are generally termed as fuzzy PID controllers. Basically, a fuzzy supervisor is working to recognise when the controller is not properly tuned and then seeks to adjust the PID gains to obtain improved performance. The design of the PID auto-tuner (upper-level supervisor) may be implemented via simple tuning fuzzy rules where the premises of the rules form part of the behaviour recogniser and the consequents form the PID tuner. Some candidate rules for such a *Mamdani* model based fuzzy system may include the following [4]:

IF steady-state-error IS large THEN increase the proportional-gain.

IF the response IS oscillatory THEN increase the derivative-gain.

IF the steady-state-error IS too-big THEN adjust the integral-gain.

IF the overshoot IS too-big THEN decrease the proportional-gain.

The approach of fuzzy-logic based auto-tuning is recently used by Chang and Shyu in [82] for the application of active noise cancellation, and used by Abdul-Mannan *et al.* in [83] for PI controller for high-performance induction motor drive.

2.6.2 Fuzzy Gain Scheduling

Conventional gain scheduling involves using extra information from the plant, environment, or users to tune, via schedules, the gains of a controller (i.e., schedule controller gains). The design provides a set of gains for the controller at each operating condition over the entire operation envelop. A gain schedule is simply an interpolator that takes as inputs the operating condition and provides values of the gains as its outputs [4]. In the context of fuzzy systems, gain scheduled control is obtained when using *Takagi-Sugeno* model based fuzzy controller represented by the following rule set [13]:

$$\text{IF } z(k) \text{ IS } A_i \text{ THEN } u(k) = C_i y(k),\ i = 1,2,\ldots,k$$

where $z(k)$ is the vector for the scheduled variables, A_i is the antecedent linguistic terms (such as ‘small’, ‘large’, etc.), represented by fuzzy sets, k is the number of rules, $u(k)$ is the control input, $y(k)$ is the systems output signal, and C_i is a linear time-invariant controller. In [72], the PID gains K_p, K_i, and K_d were respectively calculated through fuzzy logic based on the error signal and the first difference of the error signal.

This general gain scheduling approach is widely used in aircraft industry and engine control [4]. Brdys and Littler [84] used this technique for nonlinear servo tracking where the servo controls two elements of a tracker mounted on a ship at sea.

2.6.3 Fuzzy Supervisory Control

Fuzzy supervisor control approach can offer more general functionality than only tuning of gains. They also can provide the capability to completely switch which controllers are operating at the lower level. That is, they can switch between linear and nonlinear controllers, controllers of different order or different structure [4]. Garcia-Benitez *et al.* [89] proposed a two level hierarchical control strategy to achieve accurate end-point position of a two-link robot with flexible members. The upper level consists of a fuzzy logic based supervisor, whereas the lower level consists of three conventional controllers, all involved in shaping of the control input in order to achieve satisfactory performance. The fuzzy supervisor chooses within these three control strategies and tunes their parameters according to the commanded manoeuvre speed and robot arm configuration. In [90], Jia *et al.* suggested that fuzzy switching controller can be used to achieve smooth control input signal in multiple model approach control. Based upon the arguments presented in this section (2.7), this thesis proposes a fuzzy logic switching and tuning supervisor for the conventional multiple-controller proposed by Zayed *et al.* in [17, 39, 70] and will be discussed in the next chapter.

2.7 Summary

This chapter started by explaining that intelligent or learning systems are characterised by the estimation of a desired mapping and they naturally involve learning or adaptation of the systems associational structure or functional dependencies in response to changes in process parameters. The ever increasing complexity of dynamical systems coupled with the increasing demands in closed loop performance specification necessitates the use of more complex and sophisticated controllers, yet as systems become more complex uncertainty in modelling increases. Intelligent controllers are enhanced adaptive or self-organising controllers that can accommodate significant changes in the plant and its environment, whilst meeting increasingly stringent controller specifications. After recognising the importance of intelligent control, the chapter then moved onto the issue of complex systems and how they can be identified using neural networks. It was assumed that if the exact knowledge of the system is available, it is possible to transform a nonlinear adaptive control problem into a linear control problem by using a feedback linearization technique. However, in many cases, the plant to be controlled is too complex to obtain the exact system dynamics, and the operating conditions in dynamic environments may be unexpected. Therefore, adaptive control has been combined with function approximators such as neural networks. These types of controllers take the capability of learning unknown nonlinear functions by universal approximation theorem and massive parallel computation. The approximation theory of neural networks was discussed in section (2.4.3). More details regarding MLP and RBF neural networks function approximation was illustrated in section (2.4.4).

Using back-propagation learning algorithm, neural networks have presented a popular architecture in many research fields, including complex system identification and control. Recent results have indicated that neural networks can exactly match input and output signal behaviours. Because of this, a neural network based system model can be used in the design of a controller or can become a part of a model-based control scheme. Therefore, section (2.5) was dedicated to present a review of the main application of neural networks in control engineering.

In section (2.6), the chapter ends by reviewing the capabilities of fuzzy logic in supervising adaptive control systems. The fuzzy logic based supervisor can use any available data from the control system to characterize the system's current behaviour so that it knows how to change the controller and ultimately achieve the desired specifications. It was shown that fuzzy supervisor control approach can offer the functionality of tuning controller gains. Moreover, they also can provide the capability to completely switch which controllers are operating at the lower level. That is, they can switch between linear and nonlinear controllers, controllers of different order or different structure.

The advantages of neural networks in approximating the nonlinear dynamics of complex systems along with the capabilities of fuzzy logic in tuning controller gains and switching between conventional controllers, which were summarised above, represent the tools that will be used, in this thesis, to improve modelling complex plants and to bring autonomy to the conventional multiple-controller discussed in the next chapter.

Chapter 3

Review of Multiple-Controller Architectures and Algorithms

3.1 Introduction

Control engineers are sometimes confronted with situations in which they have to design and implement real-time control systems that are composed of a set of controllers in stead of a single control algorithm. These situations occur for example when the control problem to solve is complex of nature, that is, when the problem can be thought of being composed of an interconnection of a set of simpler sub-problems [8]. A common approach to control such complex dynamic systems is to design a set of different controllers, each of which for a particular operating region or performance objective, and then to switch them in real-time to achieve the overall control objective [69]. Some architectures of multi-controller have been reported in the field of control engineering, and these are known under the general name of *multiple model approach* [8], which is an appealing approach to adaptive control with the potential to include complex systems [11].

The multiple model approach appears in different forms. Hilhorst *et al.* in 1994 [91] used multiple controllers to control a plant whose behaviour can be described by a limited set of so-called modes. A similar approach is used by Narendra and Balakrishnan in 1997 [73] who exploit multiple controllers to handle plant faults, such as sensor or actuator fall-outs. Other techniques that use multiple controllers are Gain-Scheduling Controllers [92], Tagaki-Sugeno Fuzzy Models introduced by Takagi and Sugeno in 1985 [93] and Logic-based Switching Controllers [94]. An important category of such systems are those consisting of a process to be controlled, a family of fixed-gain or variable-gain candidate controllers, and an event-driven switching logic called a supervisor whose job is to determine in real-time which controller should be applied to the process. Major reasons for introducing logic and switching are to deal with communication, actuator and sensor constraints, with model uncertainty, with unforeseen events or to avoid performing difficult tasks e.g., precise equipment calibration which might otherwise be necessary were one to consider only conventional controls [94].

3.2 Multiple-Controller Approach General Architecture

Perhaps the general architecture for a feedback system employing a family of controllers is that reported by [94] and depicted in Figure (3.1). That is, the measured output y of a process to be controlled drives a bank of controllers, each controller generating a candidate feedback signal u_i. The control signal applied to the process at each instant of time is in the form

$$u \equiv u_{\eta}\,, \tag{3.1}$$

where η is a switching signal. The generation of such a switching signal is typically carried out by some type of hybrid dynamical system which is called supervisor [97]. The key factor for the simplicity of this structure is that at any instant of sampling time t only one of the constituent controllers is to be applied to the process, which Johansen and Murray-Smith [95] termed as the local controller. Because of this, at each time t it is only necessary to generate one candidate control signal [94]. Switching controllers is needed for reacting to rapidly changing plant characteristics and avoiding catastrophic failures [98].

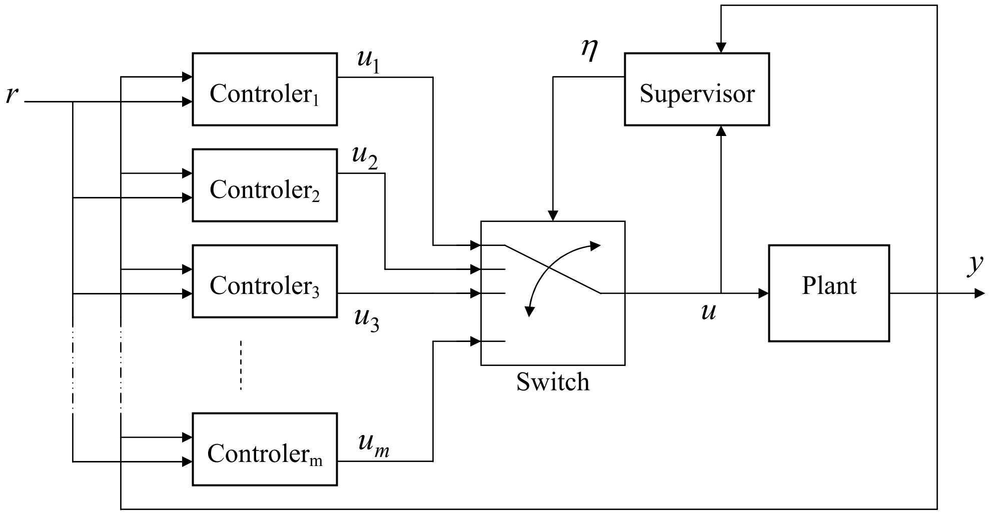


Figure (3.1): Multiple-controller general architecture, r is the reference signal, u is the control input and y is the output signal.

In this approach, the complex mechatronic plant system is modelled as a physical process that is operating in a limited set of operating regimes. With conventional methods it might be possible to design one robust controller that controls the plant in all operating regimes, but it will not work optimally for the current operating regime [96]. Parameter adaptive controllers can be used, but they may respond too slowly to abrupt changes of the plant's dynamic behaviour. In the above multiple-controller architecture, the plant is controlled by a set of controller modules (u_m), each optimized for a special operating regime of the plant. The supervisor is able to switch between the controller modules to determine the active module. The decision to switch from one u_m to the next is made on basis of measurements of physical values of the plant. The strategy of the supervisor can vary from manual switching to simple functions of the measurements to agent-based techniques [5, 96].

3.3 Complex Systems Modelling for Multiple Controllers

As mentioned above, a variety of multiple controller techniques exist. Although they share common tasks, such as decomposing the overall control problem and integrating individual solutions (i.e., controllers), they all deal with these aspects in their own unique way [5, 8]. There is no general abstract framework for discussing the relevant issues related to the design of complex control systems that are composed of local controllers. Such an abstract framework would enhance the divide-and-conquer strategy, as it would allow the use of heterogeneous multiple model techniques while using a uniform design method. By considering the design of a multi-controller as a general distributed problem, the automation capabilities provided by the field of Artificial Intelligence can be integrated with the concepts and techniques from this field to the multiple model approach of designing control systems may be advantageous, from a practical perspective, to solve complex control problems.

Recently in 2005, Narendra mentioned in [98] that the control of complex systems is considered difficult is due to four reasons (i.e., complexity, uncertainty, nonlinearity, and time-variation). He further stated that adaptation and learning can deal with uncertainty, while neural networks help to cope with complexity and nonlinearity [101]. For the time-variations problem, adaptive control theorists have been interested in adaptation in changing environments. Based on the success of employing adaptive control on *time-invariant* systems with unknown parameters, it would also prove satisfactory when the parameters varied with time, provided that the variation occurred on a relatively slow time scale [98, 99].

To control such a complex system, Zhu *et al*. [16, 19] proposed a neural network based plant model which combines a linear time invariant or slowly time varying sub-model plus a nonlinear time-varying sub-model (or an error agent), which have been collectively termed the Generalised Learning Model (GLM) [37, 100]. The linear sub-model is used to approximate the dominant linear dynamics of the complex plant around its operating point. On the other hand, the 'learning' error agent is used to learn the errors from the linear sub-model that are due to nonlinearities, uncertainties, disturbances and model mismatch in the controlled plant. This methodology was considered by Zayed *et al.*[17, 39, 70] in their proposed multiple controller framework for controlling complex systems, which will be discussed in the next section.

3.4 Review of the conventional Multiple-Controller Framework

The multiple-controller proposed by Zayed *et al.* [17, 39, 70] builds on the concept of minimum variance control which was originally introduced by Aström and Wittenmark [102], which had as its target the minimisation of the variance of the plant output. On the other hand, it had considerable limitations in that the control objective was only appropriate for minimum phase systems and excessive control input might be obtained if Simple Minimum Variance Controller (SMVC) was used. These limitations were overcome by the modification of Clarke and Gawthrop [103] in what is known as the Generalised Minimum Variance Control (GMVC). As an extension to this work, the GMVC was modified by Allidena and Hughes [104] to achieve pole-placement control.

There are two main reasons behind using pole-placement control. Firstly, in the regulator case, it provides a means for overcoming the restriction to minimum phase systems of the original minimum variance self-tuner of [102]. Secondly, in the servo case, it gives the ability of directly introducing bandwidth and damping ratio as tuning parameters. However, the modified controller has considerable drawbacks in that the arbitrary zeros, which may be used to reduce excessive control, are not considered in the design and the controller design involves the solution of a Diophantine equation, which in some applications may lead to excessive computational and numerical instability problems.

Recently, more attention was given to the zeros since they can be used to achieve better set point tracking and they also help reduce the magnitude of the control action [47, 105]. The generalised minimum variance controller was extended to achieve pole-placement control by Hussain *et al*. [105], and Allidina and Hughes [104] but the zeros were not considered in these initial designs. To further develop and extend the GMVC technique in order to achieve zero-pole placement control, Zayed *et al.*[46] proposed pole-zero placement controller that overcomes the main problems associated with the original generalised minimum variance pole-placement controller of [104].

It is a fact that PID controllers are popular for their robustness in a wide range of operating conditions, the simplicity of their structures, as well as the familiarity of designers and operators with PID algorithms. Also, they are easy to implement using analogue or digital hardware and they are inexpensive to implement and reasonably sufficient for many industrial control needs [72, 4, 17]. For these reasons the generalised minimum variance controller for the SISO case (Clarke and Gawthrop, [103]) was then modified for the first time by Cameron and Seborg [106] in order to combine the advantages of the conventional PID controllers with that of adaptive regulators.

Aiming towards bringing together the advantages of the adaptive controllers with those of the PID and pole-zero placement controllers, and benefiting form the neural-network enhanced generalised minimum variance adaptive controller for nonlinear discrete-time systems introduced by Zhu *et al*. [16], the multiple-controller framework was therefore proposed in [17, 39, 70]. This multipurpose controller provides the user with a choice of using either a conventional PID, a PID based pole-placement or a PID based pole-zero placement adaptive nonlinear controller. All these three controllers operate using the same adaptation procedure. Effectively, the overall action of the design is that of a controller which can be implemented as an adaptive PID controller, as a pole-placement controller or as pole-zero placement controller through the use of a simple switch. Details of the design of this conventional multiple-controller are presented in the rest of this chapter.

3.4.1 Derivation of the Multiple Controller Control Law

Consider the following Controlled Auto-Regressive Moving Average (CARMA) representation for a complex plant model [16, 17, 37, 38, 39]:

$$A(z^{-1})y(t+k) = B(z^{-1})u(t) + f_{0,t}(Y,U) + \xi(t+k)\,, \tag{3.2}$$

where $y(t)$ is the measured output, $u(t)$ is the control input and $\xi(t)$ is uncorrelated sequence of random variables with zero mean at the sampling instant $t = 1,2,\ldots$, and k is the time delay of the process in the integer-sample interval. The term $f_{0,t}(Y,U)$ in equation (3.2) above, is potentially a nonlinear function (which accounts for any unknown time-delays, uncertainty and nonlinearity in the complex plant model). The overall plant model represented by equation (3.2) above, is termed the Generalized Learning Model (GLM), and can be seen as the combination of a linear sub-model and a nonlinear (learning) sub-model as shown in Figure (3.2) next.

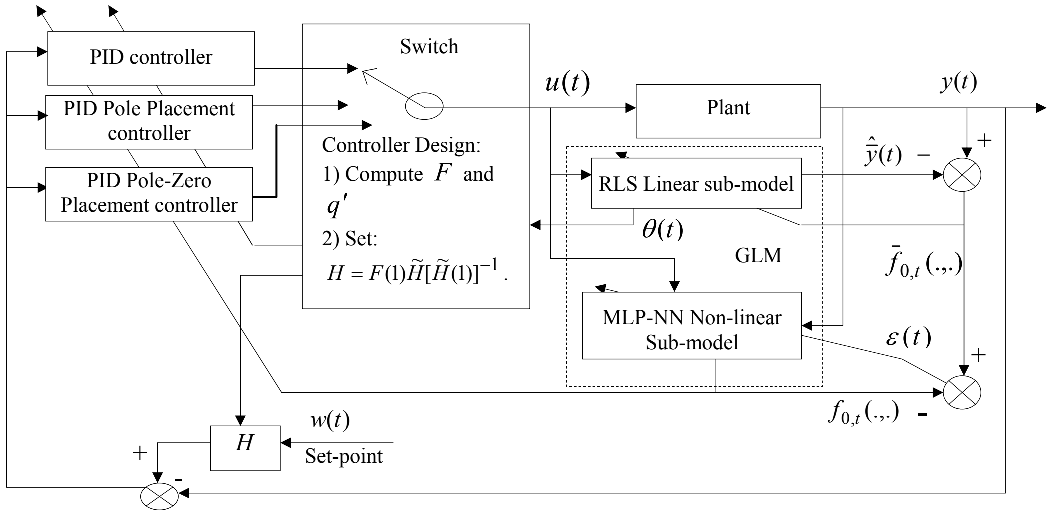


Figure (3.2): Conventional multiple-controller incorporating the GLM.

Also, in equation (3.2) above, we define $y(t) \in Y$, and $u(t) \in U$; $\{Y \in R^{n_a}; U \in R^{n_b}\}$ and $A(z^{-1})$ and $B(z^{-1})$ are polynomials with orders n_a and n_b, respectively, which can be expressed in terms of the backwards shift operator, z^{-1} as:

$$A(z^{-1}) = 1 + a_1 z^{-1} + \ldots + a_{n_a} z^{-n_a}, \tag{3.3a}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + \ldots + b_{n_b} z^{-n_b}, \; b_0 \neq 0. \tag{3.3b}$$

In order to simplify the analysis, the time delay is taken as $k = 1$ [16, 39]. For this case the non-linear system represented by equation (3.2) can be written as [39]:

$$A(z^{-1})y(t) = z^{-1}B(z^{-1})u(t) + z^{-1}f_{0,t}(Y,U) + \xi(t). \tag{3.4}$$

The generalised minimum variance controller of interest minimises the following cost function [16]:

$$J_N = E\{[\phi(t+1)]\}, \tag{3.5}$$

where

$$\phi(t+1) = [P(z^{-1})y(t+1) + Q(z^{-1})u(t) - R(z^{-1})w(t) - H_N(z^{-1})f_{0,t}(.,.)], \qquad (3.6)$$

where $w(t)$ is a bounded set point and $P(z^{-1}) = [P_d(z^{-1})]^{-1}P_n(z^{-1})$, $Q(z^{-1})$, $R(z^{-1})$ and $H_N(z^{-1})$ are user-defined transfer functions in the backward shift operator z^{-1} and $E\{.\}$ is the expectation operator.

Next, we can introduce the following identity [17]:

$$P_n(z^{-1}) = A(z^{-1})E(z^{-1})P_d(z^{-1}) + z^{-1}F(z^{-1}), \qquad (3.7)$$

where $E(z^{-1})$, $F(z^{-1})$, $P_n(z^{-1})$ and $P_d(z^{-1})$are polynomial in z^{-1}.

where $P_n(z^{-1})$ and $P_d(z^{-1})$ are the numerator and denominators of the polynomial matrix $P(z^{-1})$.

The orders of the polynomial matrices $E(z^{-1})$, $F(z^{-1})$ and $P_n(z^{-1})$ in the equations (3.6) are specified as follows:

$$\left.\begin{aligned} n_e &= k-1 \\ n_f &= (n_{P_d} + n_a - 1) \\ n_{p_n} &= \max(n_a + n_{p_d} + n_e,\ k + n_f) \end{aligned}\right\}, \qquad (3.8)$$

where, n_f, n_{p_n} and n_{p_d} represent the degrees of $P_d(z^{-1})$, $P_n(z^{-1})$ and $P_d(z^{-1})$ respectively.

Multiplying equation (3.4) by $P_d(z^{-1})E(z^{-1})$ and substitute for $(E(z^{-1})A(z^{-1}))$ from equation (3.7) gives:

$$[P_d]^{-1}P_n y(t+1) = F(z^{-1})y(t) + B(z^{-1})E(z^{-1})u(t) + Ef_{0,t}(.,.) + E\xi(t+1)\,. \qquad (3.9)$$

Adding $Q(z^{-1})u(t) - R(z^{-1})w(t) - H_N(z^{-1})f_{0,t}(.,.)$ to both sides of equation (3.9) and using equations (3.6) and (3.7), yields:

$$\begin{aligned}\phi(t+1) = {} & [P_d(z^{-1})]^{-1}F(z^{-1})y(t) + \\ & (Q(z^{-1}) + B(z^{-1})E(z^{-1}))u(t) - R(z^{-1})w(t) + \\ & (E(z^{-1}) - H_N(z^{-1}))f_{0,t}(.,.) + E(z^{-1})\xi(t+1).\end{aligned} \qquad (3.10)$$

In the rest of this section, the argument z^{-1} will be omitted from various polynomials and transfer functions in order to simplify the notation and will only be used where required for clarification purposes.

Now we can define the optimal predictor $\phi^*(t+1|t)$ and the prediction error $\tilde{\phi}(t+1\,|\,t)$ as follows:

$$\phi_y^*(t+1|t) = [P_d]^{-1}Fy(t) + (Q+EB)u(t) + [E - H_N]f_{0,t}(.,.) - Rw(t)\,, \qquad (3.11)$$

$$\tilde{\phi}(t+1|t) = E\xi(t+1)\,. \qquad (3.12)$$

If we set $\phi^*(t+1|t) = 0$ in equation (3.11) and after some arrangement, the generalised minimum variance control law for non-linear systems is obtained as:

$$P_d(EB+Q)u(t) = [P_d Rw(t) - Fy(t) + P_d(H_N - E)f_{0,t}(.,.)]\,. \qquad (3.13)$$

Now, if we set:

$$\left. \begin{aligned} R &= [P_d]^{-1} H_0 \\ H_N &= ([P_d]^{-1} \Delta H'_N + E) \end{aligned} \right\}. \tag{3.14}$$

If we set the transfer function $Q(z^{-1})$ such that the following relation is satisfied [17, 37]:

$$P_d(EB + CQ) = v^{-1} \Delta q', \tag{3.15}$$

then, equation (3.13) becomes:

$$\Delta q' u(t) = [vH_0 w(t) - vFy(t) + \Delta v H'_N f_{0,t}(.,.)], \tag{3.16}$$

where v is a user defined gain [40, 41] and q' is a polynomial in z^{-1} having the following form:

$$q'(z^{-1}) = 1 + q'_1 z^{-1} + ... + q'_{n_{q'}} z^{-n_{q'}}, \tag{3.17}$$

where $n_{q'}$ is the degree of the polynomial q'.

We can see clearly from equations (3.15) and (3.16) that the controller denominator has now conveniently been split into two parts:

An integrator action part (Δ) required for PID design, where $\Delta = (1 - z^{-1})$.

An arbitrary compensator (q') that may be used for pole (only) placement and simultaneous pole and zero placement designs.

It can be seen from equation (3.15) that the polynomial matrix $q'(z^{-1})$ and the gain matrix v are user-defined parameters since they depend on the user transfer function $Q(z^{-1})$. It is also clear from equation (3.14) that H_0 and H'_N are user-defined parameters because they depend on the transfer functions $R(z^{-1})$ and H_N respectively.

Now, if we set:

$$H_0 = \tilde{H}[\tilde{H}(1)]^{-1}F(1)\,, \tag{3.18}$$

and combine equations (3.18) and (3.16), then we can readily obtain:

$$\Delta q'u(t) = v\tilde{H}[\tilde{H}(1)]^{-1}F(1)w(t) - vFy(t) + \Delta vH'_N f_{0,t}(.,.)\,, \tag{3.19}$$

where $\tilde{H}$ in equation (3.19) is a user-defined polynomial which can be used to introduce arbitrary closed loop zeros for explicit pole-zero placement controller and has the following form:

$$\tilde{H}(z^{-1}) = 1 + \tilde{h}_1 z^{-1} + \ldots + \tilde{h}_{n_{\tilde{h}}} z^{-n} \;. \tag{3.20}$$

The above equation (3.19) represents the final expression of the control law for the proposed multiple controller

3.4.2 Multiple Controller Mode 1: Non-Linear PID Controller

In this mode, the multiple controller operates as a conventional self-tuning PID controller, which can be expressed in the most commonly used velocity form [41, 42] as:

$$\Delta u(t) = K_I w(t) - [K_P + K_I + K_D]y(t) - \\ [-K_P - 2K_D]y(t-1) - K_D y(t-2). \tag{3.21}$$

If we assume that the degree of $F(z^{-1})$ is equal to 2, therefore,

$$F(z^{-1}) = f_0 + f_1 z^{-1} + f_2 z^{-1}, \tag{3.22}$$

Then, if we switch off, both the pole-placement polynomial q' given by equation (3.17) and zero-placement polynomial $\widetilde{H}$ given by (3.20), by setting:

$$\left.\begin{array}{l} q'(z^{-1}) = 1,\ (i.e.\ q'_1 = q'_2 = ..,.. = q_{n_{q'}} = 0) \\ \widetilde{H} = 1,\ (i.e.\ \widetilde{h}_1 = \widetilde{h}_2 = ...,... = \widetilde{h}_{n_{\widetilde{h}}} = 0) \end{array}\right\}, \tag{3.23a}$$

And, next, if we set:

$$H'_N = -[B(1)v]^{-1} q'(1), \tag{3.23b}$$

then an adaptive controller with PID structure is obtained, where

$$\Delta u(t) = [vF(1)w(t) - v(f_0 + f_1 z^{-1} + f_2 z^{-1})y(t) + \Delta v H'_N f_{0,t}(.,.)], \tag{3.24}$$

$$K_P = -v[f_1 + 2vf_2], \tag{3.25a}$$

$$K_I = v[f_0 + f_1 + f_2], \tag{3.25b}$$

$$K_D = vf_2 . \tag{3.25c}$$

It can be seen from the above equations (3.24), (3.25a), (3.25b) and (3.25c) that the PID control parameters K_P, K_I and K_D depend on the polynomial matrix $F(z^{-1})$ and the gain v [43, 41]. In this case the parameters of the polynomial matrix $F(z^{-1})$, f_0, f_1 and f_2 are computed directly from equation (3.7) by selecting suitable user-defined polynomials P_d and P_n which are selected in trial and error basis. It can also clearly be seen from equation (3.8) that the order of $F(z^{-1})$ which indicates the type of the controller (PI or PID) is governed by the polynomial A and P_d [43, 41].

As stated above, the multiple controller mode 1 described by equations (3.21)-(3.25c) is tuned by a selection of the polynomials P_n and P_d, and the gain v. However, the main disadvantage of many PID self-tuning based minimum variance control designs (see for example [43, 41]) is that the tuning parameters must be selected using a trial and error procedure. Alternatively, these tuning parameters can be automatically and implicitly set on line by specifying the desired closed loop poles [44, 16].

3.4.3 Multiple Controller Mode 2: Non-Linear PID Based Pole (only) Placement Controller

Substituting for $u(t)$ given by equation (3.19) into the process model described by equation (3.4), the closed loop system is obtained as:

$$\begin{aligned}(\widetilde{A}q' + z^{-1}\widetilde{B}F)y(t) = z^{-1}Bv\widetilde{H}[H(1)]^{-1}F(1)w(t) + \\ \Delta(z^{-1}BvH'_N + q')f_{0,t} + \Delta q'\xi(t)\end{aligned}, \tag{3.26}$$

$$\left.\begin{aligned}\widetilde{A} = \Delta A \\ \widetilde{B} = vB\end{aligned}\right\}, \tag{3.27}$$

We can now introduce the identity:

$$(q'\widetilde{A}+z^{-1}F\widetilde{B})=T\ , \tag{3.28}$$

where T is the desired closed loop poles and q' is the controller polynomial. For equation (3.28) to have a unique solution, the order of the regulator polynomials and the number of the desired closed loop poles can be set as [40, 43, 45, 46]:

$$\left.\begin{array}{l} n_{f'}=n_{\widetilde{a}}-1=n_a \\ n_{q'}=n_{\widetilde{b}}+k-1 \\ n_t\le n_{\widetilde{a}}+n_{\widetilde{b}}+k-1 \end{array}\right\}, \tag{3.29}$$

where $n_{\widetilde{a}}$, $n_{\widetilde{b}}$, and $n_{q'}$ are the orders of the polynomials $\widetilde{A}$, $\widetilde{B}$ and q', respectively, and n_t denotes the number of desired closed loop poles. Also, $n_{\widetilde{b}}=n_b$ and $n_{\widetilde{a}}=n_a+1$. Combining equations (3.26) and (3.28), gives:

$$Ty(t)=z^{-1}Bv\widetilde{H}[H(1)]^{-1}F(1)w(t)+\Delta(z^{-1}BvH'_N+q')f_{0,t}+\Delta q'\xi(t)\ . \tag{3.30}$$

If the explicit zero placement polynomial given by (3.20) is switched off by setting:

$$\widetilde{H}=1,\ (i.e.\ \widetilde{h}_1\ =\widetilde{h}_2=...=\widetilde{h}_{n_{\widetilde{h}}}=0)\ . \tag{3.31}$$

And, if we set:

$$H'_N=-[B(1)v]^{-1}q'(1)\ , \tag{3.32}$$

then the closed loop function of equation (3.30) becomes:

$$Ty(t) = z^{-1}BvF(1)w(t) + \Delta(z^{-1}Bvq'(1)[-(B(1)v)^{-1}] + q')f_{0,t} + \Delta q'\xi(t) . \qquad (3.33)$$

In this case

$$T(z^{-1}) = 1 + t_1 z^{-1} + t_2 z^{-2} + \ldots + t_{n_t} z^{-n_t} , \qquad (3.34)$$

where $n_{\tilde{h}}$ and n_t in equations (3.31) and(3.34) represent orders of the polynomials $\widetilde{H}(z^{-1})$ and $T(z^{-1})$ respectively.

It can be seen from equation (3.33) that the closed loop poles are placed at their desired positions which is pre-specified by the user through the use of the polynomial $T(z^{-1})$.

3.4.4 Multiple Controller Mode 3: Non-Linear PID Based (simultaneous) Pole Zero Placement Controller

In this controller mode, an arbitrary desired zeros polynomial can be used to reduce excessive control action, which can result from set point changes when pole placement is used.

If the zero-placement polynomial ($\widetilde{H}$) given by equation (3.20) is switched on then the closed loop given by equation (3.26) is again obtained and can be simplified such that the closed loop function of equation (3.33) becomes:

$$\begin{aligned} Ty(t) = z^{-1}Bv[\widetilde{H}(1)]^{-1}\widetilde{H}F(1)w(t) + \\ \Delta(z^{-1}Bvq'(1)[-(B(1)v)^{-1}] + q')f_{0,t} + \Delta q'\xi(t). \end{aligned} \qquad (3.35)$$

Note that in practice, the order of $T(z^{-1})$ and $\widetilde{H}(z^{-1})$ are most of the time selected to equal 1 or 2 [43, 46, 47].

It can be seen from equations (3.35) that the closed loop poles and zero are placed at their desired positions which are pre-specified by using the polynomials $T(z^{-1})$ and $\widetilde{H}(z^{-1})$.

It can also be clearly seen from equations (3.7)-(3.8) and (3.13)-(3.17) above, that the user defined transfer functions P, Q, R and H_N must change at every sampling instant in order to satisfy the conditions specified by equations (3.22), (3.23), (3.25a), (3.25b) and (3.25c) for achieving self-tuning PID control (multiple controller mode 1). On the other hand, the above user-defined transfer functions must change in order to satisfy equations (3.28), (3.29), (3.31), (3.32) and (3.34) for achieving pole (only) placement control (multiple controller mode 2). Finally, for achieving simultaneous pole and zero placement control (multiple controller mode 3) these user-defined transfer functions must change automatically in order to satisfy equations (3.28), (3.29), (3.32) and (3.20). However, note that it is not necessary to explicitly calculate these user defined design transfer functions [39, 40, 41, 43, 46]. This does, of course, suggest that the cost index has time varying weightings in this problem.

3.4.5 GLM based Identification of the Complex Plant Model

This sub-section will provide details regarding the GLM model discussed in section 2.5.6 and included in Figure (3.2) above, where a recursive least squares algorithm is initially used to estimate the parameters A and B (equation (3.2)) of the linear sub-model, and a neural network based learning model is subsequently used to approximate the non-linear part $f_{0,t}$.

It can be seen from equation (3.4) that the measured output $y(t)$ can be obtained as follows:

$$y(t) = \varphi^T(t)\theta\ (t) + f_{0,t}(y,u)\,, \tag{3.36}$$

where θ is the parameter vector and $\varphi \in \Re^m$ is the data vector as follows:

$$\left.\begin{array}{l} \theta(t) = [-a_1,...,-a_{n_a}, b_0,...,b_{n_b}]^T \\ \varphi^T(t) = [y(t-1),...,y(t-n_a), u(t-1),...u(t-n_b)] \end{array}\right\}. \tag{3.37}$$

Equations (3.4) and (3.36) can also be presented as:

$$y(t) = \hat{\bar{y}}(t) + \bar{f}_{0,t}(.,.)\,. \tag{3.38}$$

It can be seen from Figure (3.2) that $\hat{\bar{y}}(t) = \varphi^T(t)\hat{\theta}(t)$ is the linear sub-model output and $\bar{f}_{0,t}(.,.) = y(t) - \hat{\bar{y}}(t)$ is the difference between the actual output $y(t)$ and the linear sub-model output $\hat{\bar{y}}(t)$.

From Figure (3.2) we can also see that $\bar{f}_{0,t}(.,.)$ can be expressed as:

$$\bar{f}_{0,t}(.,.) = f_{0,t}(.,.) + \varepsilon(t)\,. \tag{3.39}$$

Using the above equation (3.39), an MLP neural network is used for estimating the non-linear function $f_{0,t}(.,.)$. The identification error $\varepsilon\ (t)$, with the error-correction learning rule (delta rule), is used to update the weights and thresholds of the learning MLP neural network model [16, 19]. The schematic diagram of the MLP neural network is shown in Figure (3.3), and the non-linear function $f_{0,t}(.,.)$ is adaptively estimated by using the following equations [39]:

$$\left.\begin{aligned} \hat{f}_{0,t}(.,.) &= \frac{1}{1+\exp[-\beta'_{3_i} layer_{\text{hidden}_2}]} \\ layer_{\text{hidden}_2} &= \sum_{n=1}^{r} w_{3_{n,i}} o_{2,n} + b_{3_i} \end{aligned}\right\}, \tag{3.40}$$

$$\left.\begin{aligned} o_{2,r}(t) &= \frac{1}{1+\exp[-\beta'_{2_r} layer_{\text{hidden}_1}]} \\ layer_{\text{hidden}_1} &= \sum_{n=1}^{j} w_{2_{n,r}} o_{1,n} + b_{2_r} \end{aligned}\right\}, \tag{3.41}$$

$$\left.\begin{aligned} o_{1,j}(t) &= \frac{1}{1+\exp[-\beta'_{1_j} layer_{\text{input}}]} \\ layer_{\text{input}} &= \sum_{n=1}^{l} w_{1_{n,j}} x_n + b_{1_j} \end{aligned}\right\}, \tag{3.42}$$

where $w_{1_{l,j}}$ and β'_{1_j} are the weights and activation factors between the input layer and the first hidden layer, $w_{2_{j,r}}$ and β'_{2_r} are the weights and activation factor between first and second hidden layers, and $w_{3_{r,i}}$ and β'_{3_i} are the weights and activation factor between second hidden layers and the output layer. The parameters b_{1_j}, b_{2_r}, and b_{3_i} are the threshold values for the j^{th}, r^{th}, and i^{th} neurons in the first hidden layer, second hidden layer, and output layer respectively.

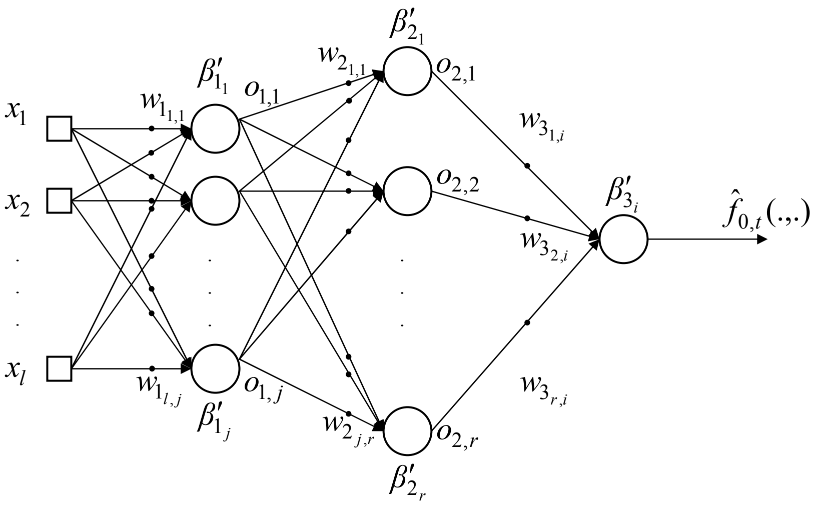


Figure (3.3): BP MLP neural network to approximate nonlinear function $f_{0,t}(.,.)$.

Note that for SISO case $i = 1$ is used in equations (3.40)-(3.42).

It is reported in the literature that the most popular neural networks in neurocontrol are the MLP and RBF NNs. Both networks are able to adaptively model or identify a dynamical complex (MIMO or SISO) process online while the process is changing [1, 21, 22, 23]. The RBF NNs ability to uniformly approximate smooth functions over compact sets is well documented (see for example [48, 49, 107]). From mathematical prospective, RBF NNs represent one class of linear in the weight approximators. Compared to the MLP network, the RBF network is simpler to implement, needs less computational memory, converges faster, and global minimum convergence is achieved even when operating conditions change or fault occurred during testing with frozen weights. The RBF NN also required less training time to converge and fewer computational complexities to train the network online [23].

In conclusion, opposed to the MLP, the RBF NNs improve the system damping and dynamic transient stability more effectively than the MLP NNs. Control performance could be improved if unknown nonlinear portion of the model are more accurately modelled [15]. Therefore, the RBF should be preferred to the MLP networks for the online identification of complex systems.

3.5 Summary

Complex control systems can be thought of being composed of an interconnection of a set of simpler sub-problems. A common approach to control such complex dynamic systems is to design a set of different controllers, each of which for a particular operating region or performance objective, and then to switch them in real-time to achieve the overall control objective. Some architectures of such multi-controller, which are known under the general name of multiple model approach, have been reported and shown to be an appealing approach to adaptive control with the potential to include complex systems.

The chapter discussed about a general architecture for a feedback system employing a family of controllers for controlling a complex mechatronic plant system which is modelled as a physical process operating in a limited set of operating regimes. Control of such complex systems is considered difficult due to their complexity, uncertainty, nonlinearity, and time-variation. Then it was concluded that, those difficulties could be dealt with by considering control adaptation and learning and soft-computing techniques in the design of a multi-controller for complex systems control.

The discussion focused on the recently developed multiple-controller framework which incorporated a neural network based generalised learning model (GLM) for modelling and control of complex systems. The unknown complex plant was represented by an equivalent model composed of a simple linear sub-model plus a non-linear sub-model. The parameters of the linear sub-model are identified by a standard recursive least squares algorithm, whereas the nonlinear sub-model was approximated using an MLP BP neural network. The controllers employed in this design were built on the concept of adaptive generalised minimum variance control. This methodology provided the designer with the choice of using a conventional PID adaptive controller, a PID pole placement controller or the PID pole-zero placement controller. All of these controllers operate using the same adaptive procedure. The switching (transition) decision between these different fixed structure controllers was achieved manually.

The chapter also discussed the limitations of the multiple-controller of Zayed *et al.* in modelling and control of complex systems. It was concluded that, these drawbacks were caused through the use of MLP neural networks to approximate the nonlinear dynamics and disturbances of the complex system. In addition, the framework lacks the automated switching among the available controllers. Moreover, the controllers design parameters (i.e. PID gain, poles and zeros) were fixed during the whole control operation without any online tuning. Switching controllers is needed for reacting to rapidly changing plant characteristics and avoiding catastrophic failures. Tuning of the controller is desirable for gradually improving the performance of the system. Consequently, both switching and tuning play important roles in the adaptive control of dynamical systems using multiple controller approach [98].

The next chapter presents the new intelligent multiple-controller framework which builds on the multiple-controller of Zayed *et al*. [17, 39, 70]. The new design uses an RBF based GLM form modelling the complex systems and incorporates a fuzzy-logic supervisor for controllers' switching and tuning. Introducing logic based switching and tuning are to deal with communication, actuator and sensor constraints, with model uncertainty, with unforeseen events or to avoid performing difficult tasks e.g., precise equipment calibration which might otherwise be necessary were one to consider only conventional controls.

Chapter 4

New Intelligent Multiple-Controller Framework Incorporating a Fuzzy Logic Based Switching and Tuning Supervisor: SISO Case

4.1 Introduction

The intelligent controller should be able to use several different control algorithms as well as to tune the parameters of each algorithm according to the desired performance specifications. It should also automatically manage the selection between those control algorithms to maintain the control objectives at or near their optimal values for specific process conditions. In emergency situations where major elements in a system break down, an intelligent controller may manage the reconfiguration of the control algorithm or switch to another more appropriate or robust control algorithm [59]. The intelligent controller knowledge base consists of experiential knowledge about the process along with facts and rules that are used to infer which control algorithm to apply and what the current parameter settings for that algorithm should be. By periodically applying identification algorithms for monitoring the results, the intelligent controller could accumulate more and more information about a given process in order to find the best control law [4].

Intelligent control as defined by Aström *et al.* [7, 74] involves the construction of a composite control structure for complex process including supervisory function, adaptive control algorithms and low-level control laws all managed by an expert system which monitor process parameters and control system performance. In this context, it is desirable to combine fuzzy logic with control systems to achieve better overall control performance [59]. Multiple controllers switching and tuning is a methodology that provides a natural framework for the design of intelligent control systems [2]. Switching controllers is needed for reacting to rapidly changing plant characteristics and avoiding catastrophic failures. Tuning of the controller is desirable for gradually improving the performance of the system. Consequently, both switching and tuning play important roles in the adaptive control of dynamical systems using multiple controller approaches [98].

The rest of this chapter is organised as follows:

Section (4.2) presents the bases for the design of the proposed intelligent multiple-controller. These bases will include: the complex plant model along with conventional multiple-controller control law; the criteria for assessing the performance of the multiple-controller; and a discussion for the multiple-controller switching decisions. Section (4.3) will give details about the design of the proposed fuzzy-logic based switching and tuning supervisor including the behaviour recogniser, the switching logic subsystem and the tuning logic subsystem. In section (4.4), the improved RBF based

GLM will be shown. Section (4.5) illustrates the general structure design of the proposed intelligent multiple-controller. The two control modes (the conventional PID controller and the Pole-zero Placement controller) will be addressed as well as the algorithm of the control procedure. Summary of the chapter will be provided in section (4.6).

4.2 Multiple-Controller Framework for Complex SISO Systems

Considering the Controlled Auto-Regressive Moving Average (CARMA) representation for a complex plant model [16, 17]:

$$A(z^{-1})y(t+k) = B(z^{-1})u(t) + f_{0,t}(Y,U) + \xi(t+k), \tag{4.1}$$

where $y(t)$ is the measured output, $u(t)$ is the control input and $\xi(t)$ is an uncorrelated sequence of random variables with zero mean at the sampling instant t=1,2,…, and k is the time delay of the process in the integer-sample interval. The term $f_{0,t}(Y,U)$ in equation (4.1) above, is potentially a non-linear function which accounts for any unknown time-delays, uncertainty and non-linearity in the complex plant model, and is conveniently represented by an MLP neural network [16, 17]. The overall plant model represented by equation (4.1) above, is termed the Generalized Learning Model (GLM) [37], which was discussed in sections 2.5.6 and 3.5, can be seen as the combination of a linear sub-model and a non-linear (learning) sub-model. Also, in equation (4.1) above, we define $y(t) \in Y$, and $u(t) \in U$; $\{Y \in R^{n_a}; U \in R^{n_b}\}$ and $A(z^{-1})$ and $B(z^{-1})$ are polynomials with orders n_a and n_b, respectively, which can be expressed in terms of the backwards shift operator, z^{-1}. as:

$$A(z^{-1}) = 1 + a_1 z^{-1} + \ldots + a_{n_a} z^{-n_a} . \tag{4.2a}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + \ldots + b_{n_b} z^{-n_b} \text{ , } b_0 \neq 0 . \tag{4.2b}$$

The multiple-controller derived control law for the above complex plant is meanwhile given in the next equation:

$$u(t) = \frac{[v\widetilde{H}[H(1)]^{-1} F(1) w(t) - vFy(t) + \Delta v H'_N f_{0,t}(.,.)]}{\Delta q'} , \tag{4.3}$$

where *w*(*t*) is the system set point, $f_{0,t}(.,.)$ is a non-linear function representing the plant non-linear dynamics with unknown time-delays, uncertainty and disturbances. The variable *v* is a user-defined gain, Δ is the integral action required for the PID design, $\widetilde{H}$ is a user-defined polynomial which can be used to introduce arbitrary closed loop zeros for an explicit pole-zero placement controller, *H*(1) is the value of $\widetilde{H}$ at system output steady state, *F* is a polynomial derived from the linear parameters of the controlled plant, *F*(1) is the value of *F* at the steady state, H'_N is a user-defined polynomial. The parameter q' is a transfer function used to bring the closed loop system poles in the stability unit disc.

It can be seen from the control law in equation (4.3) above that the controller denominator is split into two parts, namely: an integrator action part (Δ) required for

PID design; and, an arbitrary compensator (q') that may be used for simultaneous pole and zero placement designs.

4.2.1 Control System Performance Assessment

An essential part of supervision of complex system control is focused on to detect the criteria for measuring system behaviour in order to provide information about the controller performance [51]. The interest from both academia and industry in control performance monitoring has surged tremendously in the last decade. The recent survey papers by Jelali [110] and Qin and Yu [109] provide a very good collection of recent development in the control performance monitoring area for SISO and MIMO control problems.

In order to make a switching decision among multiple controllers based on the user specifications, a performance assessment criteria has been defined. Both heuristics and quantitative measures are considered in order to make a decision for the control algorithm selection. Therefore, performance of each controller is evaluated based on a number of factors. However, the user specifications about transient accuracy, usually given in terms of required overshoot, rise time and settling time can give a good measure about which control algorithm will give the best performance as compared with the others [4, 108, 110].

Another important widespread criterion considered for controller performance assessment is the variance of the system output, because of its direct relationship to process performance, product quality, and profit [110, 111, 112]. A straightforward extension of the signal variance is by considering control action penalisation. Both output variance and control action are useful when more information on controller performance, such as how much can the output variance be reduced without significantly affecting the controller output variance, is needed [110]. The price to be paid is that more information on the process is required, i.e., measurement of the manipulating variable(s).

Figure 4.1 depicts the performance criteria which are specified in terms of control action, output variance and meeting user specifications [59].

Using the performance measurements, one can assess the performance of a control loop and make statements on the potential of improvements resulting from re-tuning of controller parameters or switching to another controller [112].

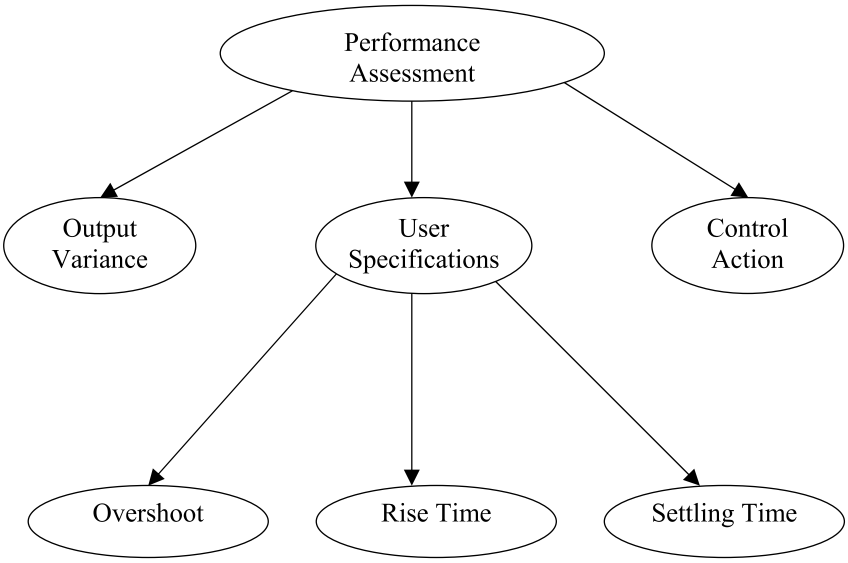


Figure (4.1): Controller Performance Assessment Criteria.

4.2.2 Discussion for the Multiple-Controller Switching Decision

Due to the robustness, simplicity of structure, ease of implementation, and remarkable effectiveness in regulating a wide range of processes (assuming correctly tuned), the conventional adaptive PID controller (multiple controller mode 1) presented in section (3.4.2) should normally be the first choice to obtain satisfactory closed-loop system performance. If however, a better closed-loop performance (based on say, a desired damping ratio, rise time, settling time overshoot or bandwidth) is required, or if the system to be controlled is difficult to tune using a conventional adaptive PID controller, then the PID structure based (simultaneous) pole and zero placement controller (multiple controller mode 3) discussed in section (3.2.4) can be used as a second choice. This mode conserves the advantages of the pole (only) placement controller, discussed in section (3.2.3), in addition to including the placement of the zeros in the control action. However this will be at the expense of a greater computational effort required for implementing this (simultaneous) pole and zero placement controller.

In the situation where an excessive control action results from set point changes, added disturbances and/or plant nonlinearity, then the multiple controller can be switched to operate in the PID structure based (simultaneous) pole and zero placement controller, to obtain a more efficient control action, at the expense of an even greater computational requirement [17].

4.3 New Fuzzy Logic Based Switching and Tuning Supervisor

The fuzzy logic based switching and tuning supervisor is situated at the highest level of the multiple-controller framework. Following [4, 8, 12, 50], the fuzzy supervisor can use any available data from the control system to characterise the system current behaviour so that it knows which controller to choose, which parameters to tune, and the tuning value for each parameter that is required to ultimately achieve the desired specification. The main idea behind the fuzzy-logic supervisor approach here is to employ logic-based tuning and switching between the family of the candidate controllers. The need for switching stems from the fact that typically no single controller can guarantee the desired behaviour when connected with a poorly modelled process, and particularly so for the case of complex processes exhibiting significant nonlinearity, non-stationarity, uncertainty and/or multi-variable interactions [37]. Such switching schemes can provide an alternative to more traditional continuously tuned adaptive control algorithms.

The supervisor employed in this work comprises three subsystems: a behaviour recogniser, a switching logic and a tuning logic, each of which are discussed next. Based on the system performance criteria presented in subsection (4.2.1), the supervisor aims to recognise when the system requires selection of another controller, or when a selected controller is not properly tuned, and then seeks to switch to the candidate controller and/or adjust the controller parameters to obtain improved performance. The whole supervisor is implemented using simple fuzzy logic based switching and tuning rules where the premises of the rules form part of the behaviour recogniser and the consequent form the switching and tuning decision. In this way, a simple fuzzy system is used to implement the entire supervisory control level.

4.3.1 Behaviour Recogniser Subsystem

The behaviour recogniser seeks to characterise the current behaviour of the control system in a way that will be useful to the switching and tuning logic subsystems. The proposed behaviour recogniser subsystem benefits from the system performance criteria discussed in section (4.2.1). The behaviour of the system is characterized through the online measurements of parameters listed below. In order to prevent the so called shattering problem (Zeno behaviour) [8], which can result in an infinite number of switching between controller modules, the average of the current and last two values of the measurements are used as output parameters from the behaviour Recogniser.

- Overshoot (ζ_y) of the closed-system output signal [51]:

$$\zeta_y(t) = \frac{y_{\max} - y_{\infty}}{y_{\infty}} 100 \, , \tag{4.4}$$

where $y_{\max}$ is the amplitude maximum value reached at the output signal, and y_{∞} is the steady state value of the output signal.

- Rise and fall time of the system output signal (ρ_y):

The output signal rise and fall times represent the amount of time for a signal to change state. To measure rise time, the behaviour recogniser uses 10% to the 90% point of the output signal, or vice versa for the output signal fall time.

- Output signal settling time (τ_y):

which is the time required for the measured process variable $y(t)$ to first enter and then remain within a band Δ_y whose width is computed as $\pm 5\%$ of the total change in $y(t)$ [113].

- Steady state error (e_∞):

which is the difference between the desired output $w(t)$ and the actual output $y(t)$ as time goes to infinity (i.e. when the output reached its steady state) [114]. The steady state error formula can be expressed as [115]:

$$e_\infty(t) = \lim_{t\to\infty}(w(t) - y(t)) \cong 0\,. \tag{4.5}$$

- The variance of the system output signal (V_y):

The variance of sampled population of the output signal $y(t)$ is the mean squared deviation of the individual values y_i of $y(t)$ from the population mean [116]. The mean is considered to be the steady state value y_∞. Therefore, V_y is computed as follows [116]:

$$V_y(t) = \frac{\sum_{i=1}^{N}(y_i - y_\infty)^2}{N-1}\,, \tag{4.6}$$

where N denotes the size of the sampled population of the output signal $y(t)$.

- Magnitude of control input signal (control action $u(t)$).

- Changes in reference signal state (Π_w):

By comparing the current set-point with the previous one, the behaviour recogniser will check the state of the reference signal whether it is increasing, decreasing or remaining as it was in the last state.

$$\Pi_w(t) = w(t) - w(t-1)\,. \tag{4.7}$$

The behaviour recogniser subsystem output is contained in the vector $\Xi(t)$ and expressed as:

$$\Xi(t) = [\zeta_y(t), \rho_y(t), \tau_y(t), e_\infty(t), V_y(t), \Pi_w(t)]\,, \tag{4.8}$$

The contents of the variable $\Xi(t)$ will be used by both fuzzy-logic based switching and tuning subsystems.

4.3.2 Fuzzy Logic Based Switching Logic Subsystem

The switching logic subsystem is designed according to the multiple-controller switching criteria presented in section (4.2.2) above, and used by Abdullah *et al.* in [100]. The key task of the switching logic subsystem is to generate a switching signal which determines, at each instant of time, the candidate controller module that is to be activated [8, 50]. The switching logic is implemented using fuzzy logic rules where the premises of the rules use four variables from the output of the behaviour recogniser

$\Xi(t)$. These variables will represent the input parameters of the switching logic, which can expressed as:

$$\Xi_s(t) = [\zeta_y(t), e_\infty(t), V_y(t), \Pi_w(t)], \quad \Xi_s \subseteq \Xi. \tag{4.9}$$

The consequents of the fuzzy rules form the controller selection decision (i.e. output parameter) which is symbolized as C_η.

$$C_\eta = [q_\eta, H_\eta]\cdot \tag{4.10}$$

The switching parameter C_η can be set to $[1,1]$ for a Pole-Zero Placement controller or $[0,0]$ for a conventional PID controller. Based on the system performance, the supervisor will set q_η and H_η to switch either to the conventional PID controller, or to the PID structure based (simultaneous) pole and zero placement controller.

4.3.2.1 Fuzzy Sets for the Switching Logic Parameters

In order to build the switching subsystem which represents a relationship between the fuzzy-logic supervisor inputs and output, each variable must first be decomposed into a set of regions and the output or solution variable then redefined into a set of fuzzy regions. There are four inputs and one output for the fuzzy switching subsystem. The inputs and the output are defined as fuzzy regions (sets) in a fuzzy logic system as shown in the following Figures (4.2a-d) and (4.3) respectively.

The membership functions (MFs) used in the fuzzy supervisor play a crucial role in the final performance of the switching and tuning subsystems. Therefore, selection of the appropriate functions is an important design problem. So, in order to design an optimal fuzzy supervisor the proper membership functions are searched by using several simulation experiments. As shown in Figures (4.2a-d and 4.3), the switching subsystem input variables are characterized by three fuzzy membership functions and the fuzzy output variable is characterized by two membership functions. From the point of view of simplicity and computational complexity [4], the fuzzy values are represented by triangular (TriMF) and trapezoidal (TrapMF) membership functions with not more than two membership functions overlapping. But, fuzzy membership functions can have different shapes and sizes depending on the designer’s preference or experience [157].

The TriMF curve is a function of a vector, x, and depends on three scalar parameters a, b, and c, as given by

$$f(x,a,b,c)=\left\{\begin{matrix} 0, & x\le a \\ \dfrac{x-a}{b-a}, & a\le x\le b \\ \\ \dfrac{c-x}{c-b}, & b\le x\le c \\ 0, & c\le x \end{matrix}\right\}. \tag{4.11}$$

The TrapMF curve is a function of a vector, x, and depends on four scalar parameters a, b, c and d, as given by

$$f(x,a,b,c,d)=\left\{\begin{matrix} 0, & x\le a \\ \dfrac{x-a}{b-a}, & a\le x\le b \\ 1, & b\le x\le c \\ \dfrac{d-x}{d-c}, & c\le x\le d \\ 0, & d\le x \end{matrix}\right\}. \tag{4.12}$$

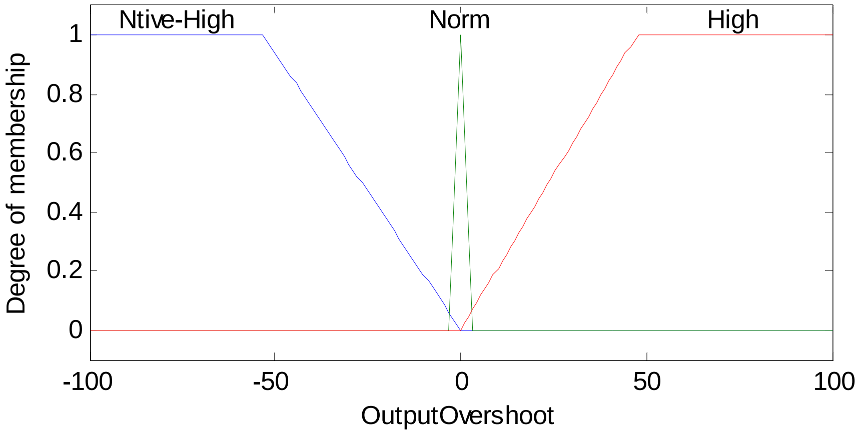


Figure (4.2a) Switching Logic input parameter: overshooting of the output signal $\zeta_y(t)$.

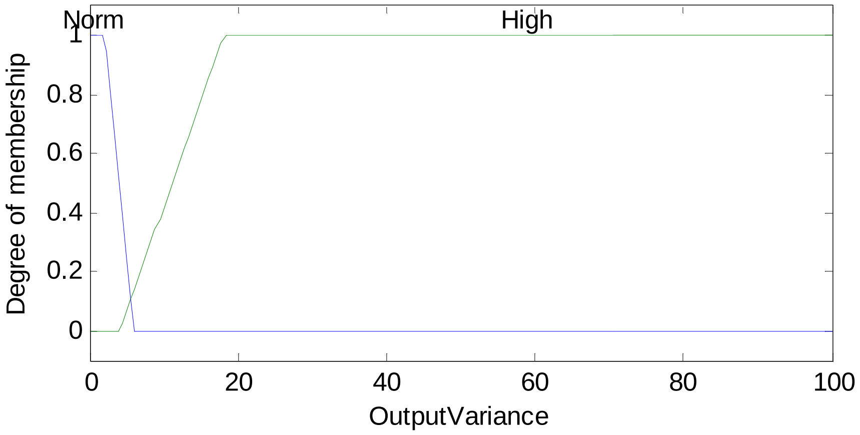


Figure (4.2b) Switching Logic input parameter: variance of the output signal $V_y(t)$.

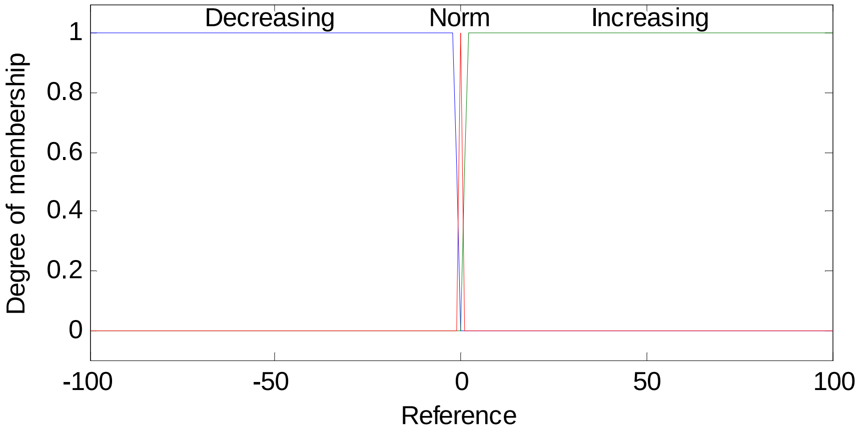


Figure (4.2c) Switching Logic input parameter: reference signal state $\Pi_w(t)$.

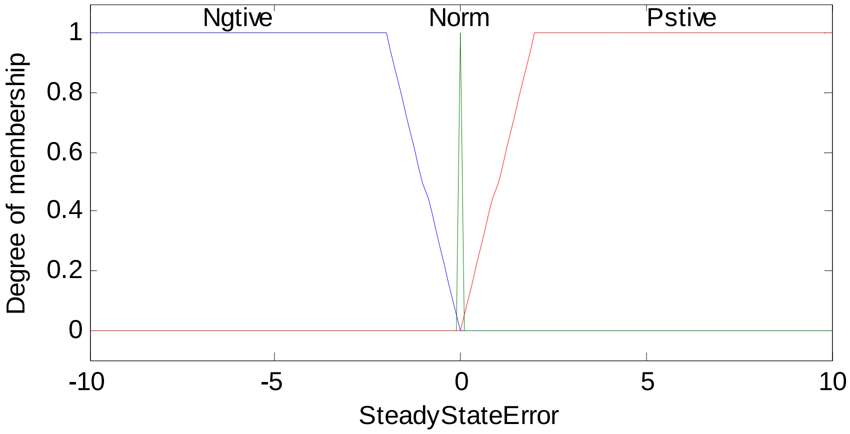


Figure (4.2d) Switching Logic input parameter: the steady state error $e_{\infty}(t)$.

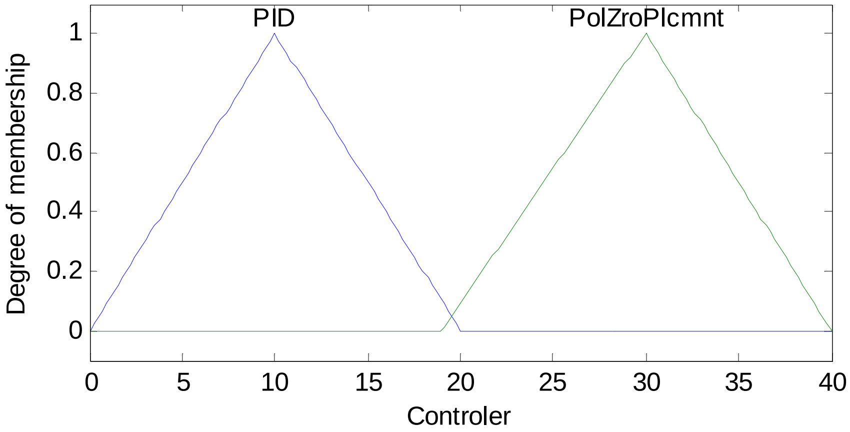


Figure (4.3): Switching Logic output parameter: controller selection $C_{\eta}(t)$.

4.3.2.2 Fuzzy Rules for the Switching Decision

Fuzzy logic is used mostly to handle high-level control functions that traditional control methods do not address such as fuzzy supervisory control for selecting discrete control actions [151]. Fuzzy switching was applied to guarantee stable switching control of a radio-controlled hovercraft [150]. Wang *et. al.* in [149] and Hu and woo [101] used fuzzy logic based supervisor for switching between sliding-mode controller and a fuzzy controller. In [90] the so-called fuzzy switching multiple model was proposed to provide smooth switching and improve the performance of the controller.

The fuzzy supervisor in the proposed multiple-controller framework is a performance-oriented supervisory control. Therefore, the role of the switching logic rule base is to determine the appropriate controller based on the system performance signals received from the behaviour recogniser. The rule base is determined by the experimental consideration of the influence of each of the switching logic input parameters: namely, overshoot of the output signal, variance of the output signal and steady state error; that is to select the required controller (output parameter). Rule 1 and Rule 2, below, activate the Pole-Zero Placement controller in the cases when the system output signal exhibits undershooting or overshooting in addition to high variance in the output signal. These two rules will work on avoiding output signal overshoot and high variance which could take place due to random and/or constant disturbances [17, 46, 70]. When the system output signal reaches low degree of overshooting and minimum variance, Rule 3 will activate the PID controller. In order to prevent output signal overshooting at any set-point change [59], Rule 4 is designed to ensure that the Pole-Zero Placement controller will be activated after each set-point change. When the output signal reaches low steady state error, Rule 5 works to activate the PID controller.

The complete set of rules for the fuzzy logic based switching subsystem is given below:

- Rule 1: IF $\zeta_y(t)$ IS Ntive-High AND $V_y(t)$ IS High THEN $C_\eta(t)$ is Pole-Zero-Placement

- Rule 2: IF $\zeta_y(t)$ IS High AND $V_y(t)$ IS High THEN $C_\eta(t)$ IS Pole-Zero-Placement
- Rule 3: IF $\zeta_y(t)$ IS Norm AND $V_y(t)$ IS Norm THEN $C_\eta(t)$ IS PID
- Rule 4: IF $\Pi_w(t)$ IS NOT Norm THEN $C_\eta(t)$ IS Pole-Zero-Placement
- Rule 5: IF $e_\infty(t)$ IS Norm THEN $C_\eta(t)$ IS PID

Depending on the fuzzified value of the input parameters, the switching logic subsystem will switch either to the conventional PID controller, or PID structure based (simultaneous) pole and zero placement controller. The above fuzzy logic based switching logic is applied to SISO water vessel system presented in chapter 6.

4.3.2.3 Fuzzy Inference Procedure for the Switching Logic

At each sampling time the switching logic input parameters are computed and compared to their desired values using their fuzzy sets. So assume that the output signal overshoot ζ_y is 2.35%, the variance of the output V_y is 5.9, the reference signal is not changing (i.e. $\Pi_w = 0$) and the steady-state-error e_∞ is 0.01 as shown in Figure (4.4). Then all the input parameters are within the range of Norm. The combination causes rules 2, 3 and 5 to fire. The three rules have somehow to be combined to form a single switching output ($C_\eta(t)$). The idea is that the premises connected by an AND are combined by taking the degree of membership of the lesser of the two as the value of combination. On the other hand, the active rules are combined together with an OR to take the larger of the three values as the value of the combination at each point on the horizontal axis. Therefore, from the combined region, one of the several techniques of defuzzification can be applied to produce final switching decision C_η. In this case, Middle-of-the-Max defuzzification approach is used as shown in Figure (4.4) where the Conventional PID controller is selected. The Middle-of-the-Max defuzzification approach grantees the selection of one controller each time. Figure (4.5), next, illustrates the case when the overshoot of the control system output signal ζ_y and its variance V_y have Norm memberships, but the reference signal is changing (i.e. decreasing since $\Pi_w = -12.1$). The combination causes Rule 3 and Rule 4 to fire. The two rules are combined and defuzzified to form the selection of the Pole-Zero Placement controller.

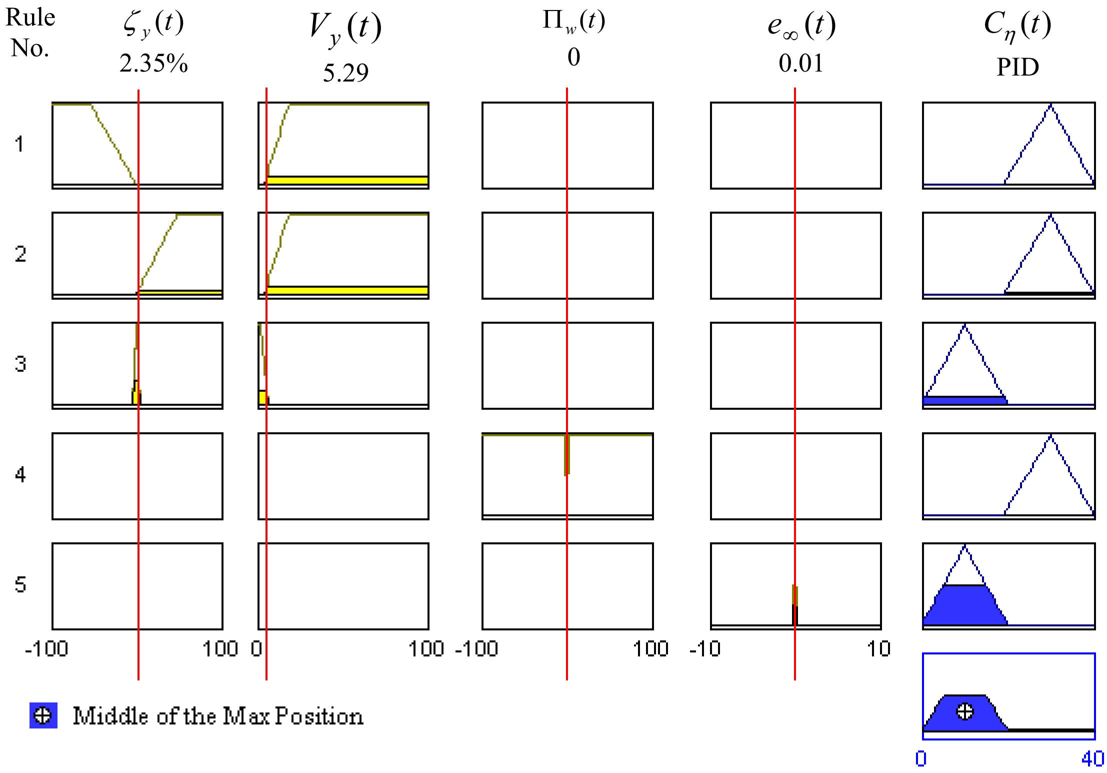


Figure (4.4): Controller selection procedure using the fuzzy-logic based switching logic subsystem: example for PID controller selection.

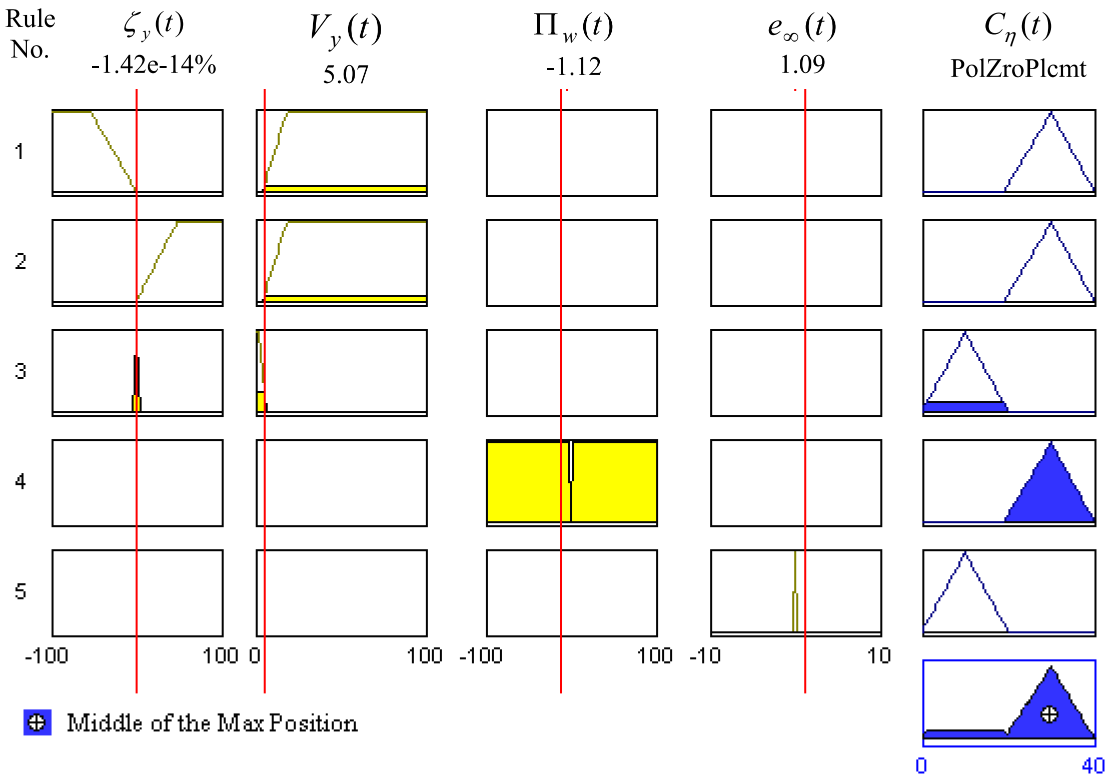


Figure (4.5): Pole-Zero Placement controller selection.

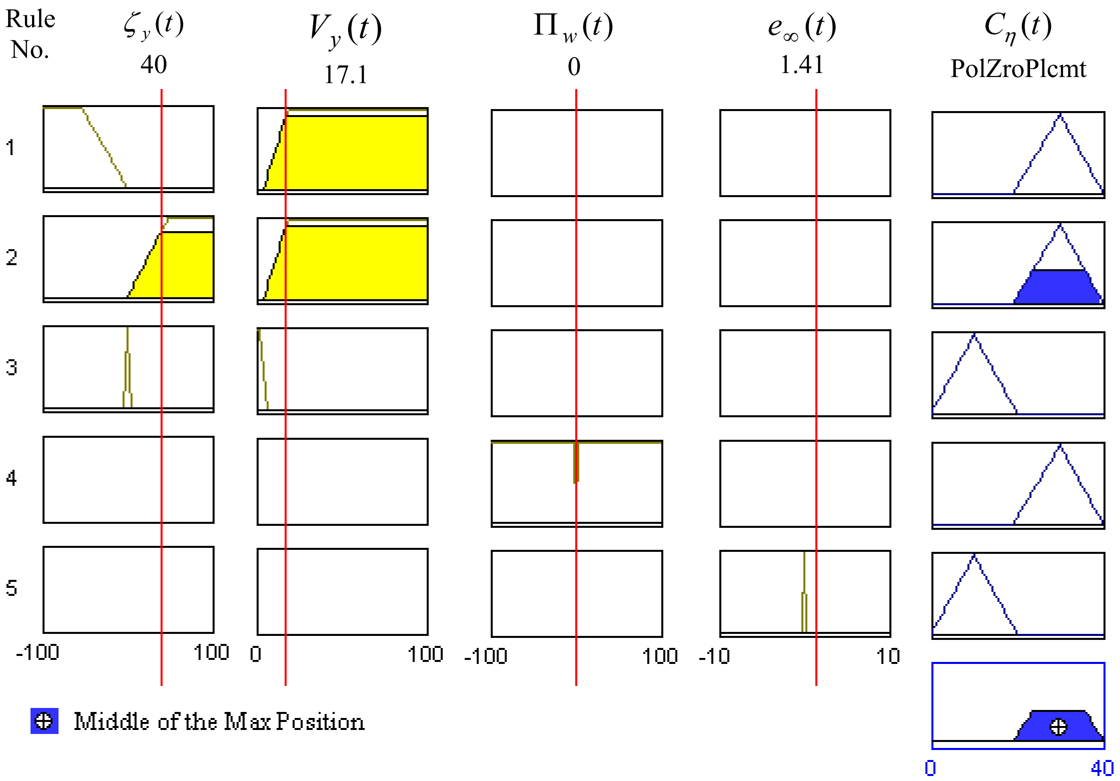


Figure (4.6): Controller selection procedure using the fuzzy-logic based switching logic subsystem: example for Pole-Zero Placement controller selection.

Figure (4.6), above, illustrates the case when the overshoot of the control system output signal ζ_y and the its variance V_y have High memberships, reference signal is not changing (i.e. $\Pi_w = 0$) and the steady-state-error is Positive. This situation could be taking place when high disturbances introduced to the control system. The combination causes Rule 2 to fire. The defuzzification forms the selection of the Pole-Zero Placement controller.

4.3.3 Fuzzy Logic Based Tuning Subsystem

The second task of the switching and tuning logic supervisor is to tune the parameters of the multiple-controller on-line, including poles and zeros of the (simultaneous) pole-zero placement controller in addition to the PID gain v. The tuning facility aims to make the system achieve a desired speed of response and/or minimise the control action.

The input parameters of the fuzzy tuning subsystem are contained in Ξ_t, which includes: part of the measurements supplied by the behaviour recogniser $\Xi(t)$, the output of the switching logic C_η (i.e. the active controller), and the control action $u(t)$.

$$\Xi_t(t) = [\zeta_y(t), \rho_y(t), \tau_y(t), e_\infty(t), \Pi_w(t), C_\eta(t), u(t)].$$

Using the fuzzy sets and fuzzy rules forming the tuning logic subsystem, the supervisor will specify the tuning values for the parameters of the active/selected controller. The new tuning values are contained in the tuning signal $C\tau$, which represents the output parameters of the fuzzy tuning subsystem and expressed as:

$$C\tau(t) = [T_\tau(t), \widetilde{H}_\tau(t), v_\tau(t)],$$

where $T_\tau(t)$ and $\widetilde{H}_\tau(t)$ respectively represent the tuning values for the poles and zero of (simultaneous) Pole-zero Placement controller and $v_\tau(t)$ is the tuning value for the PID gain. The fuzzy sets and rules designed for the multiple-controller parameters' tuning are given next.

4.3.3.1 Fuzzy Sets for the Parameters of the Tuning Subsystem

There are seven input and three output parameters for the fuzzy tuning subsystem. The inputs and the outputs are defined as fuzzy regions (sets) in a fuzzy logic system as shown in the following Figures (4.7a-g) for the inputs and (4.8a-c) for the outputs.

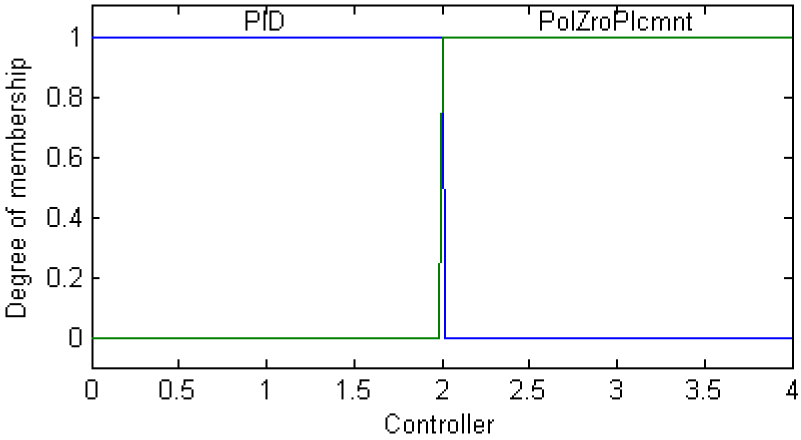


Figure (4.7a) Tuning Logic input parameter: current active controller $C_\eta(t)$.

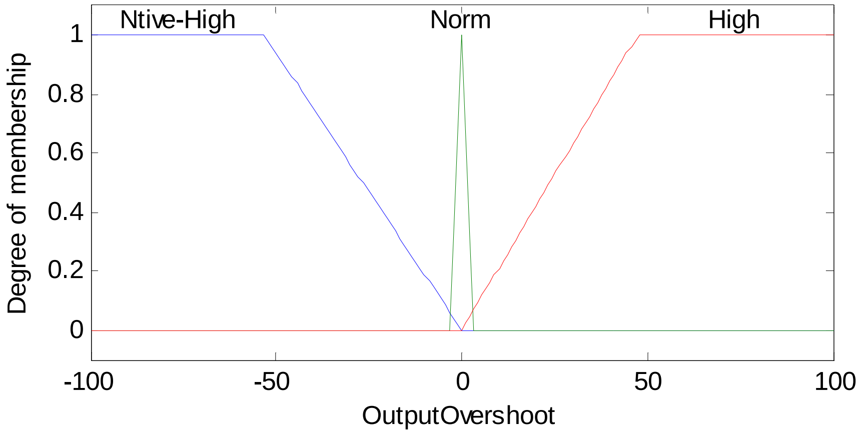


Figure (4.7b) Tuning Logic input parameter: overshooting of the output signal $\zeta_y(t)$.

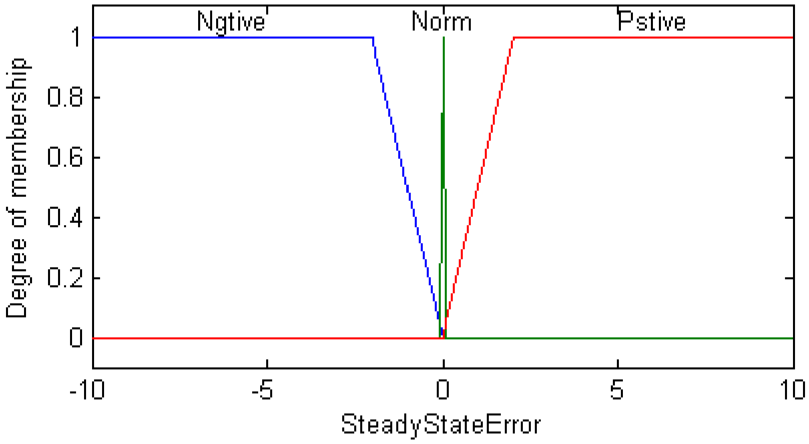


Figure (4.7c) Tuning Logic input parameter: the steady state error $e_\infty(t)$.

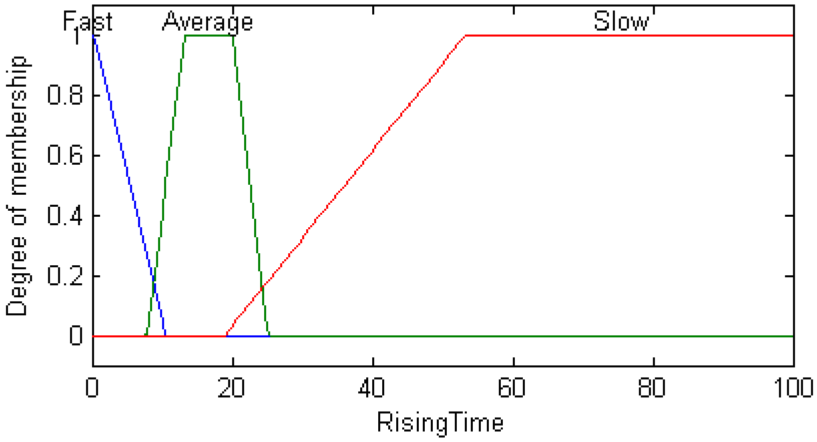


Figure (4.7d) Tuning Logic input parameter: rising time of the output signal $\rho_y(t)$.

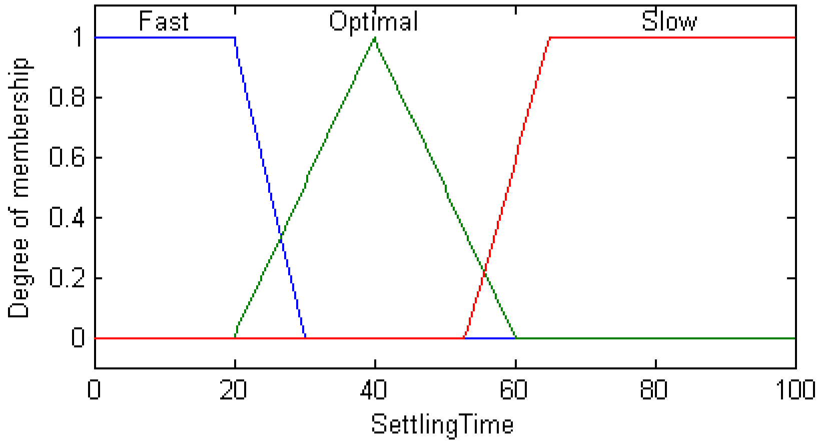


Figure (4.7e) Tuning Logic input parameter: settling time of the output signal $\tau_y(t)$.

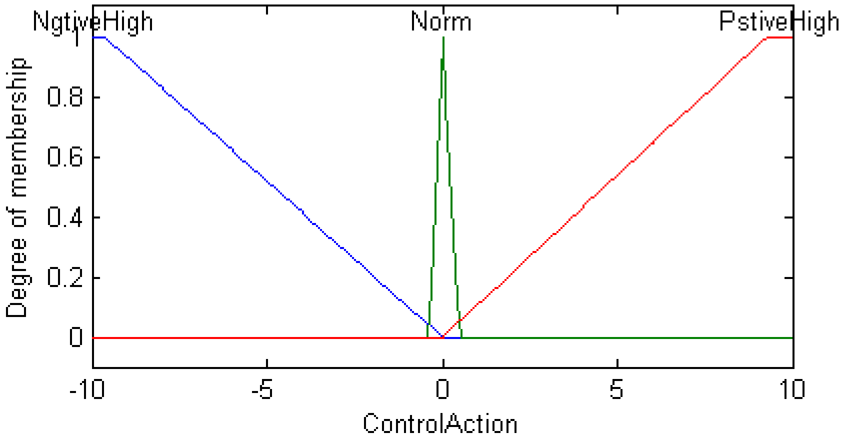


Figure (4.7f) Tuning Logic input parameter: control action signal $u(t)$.

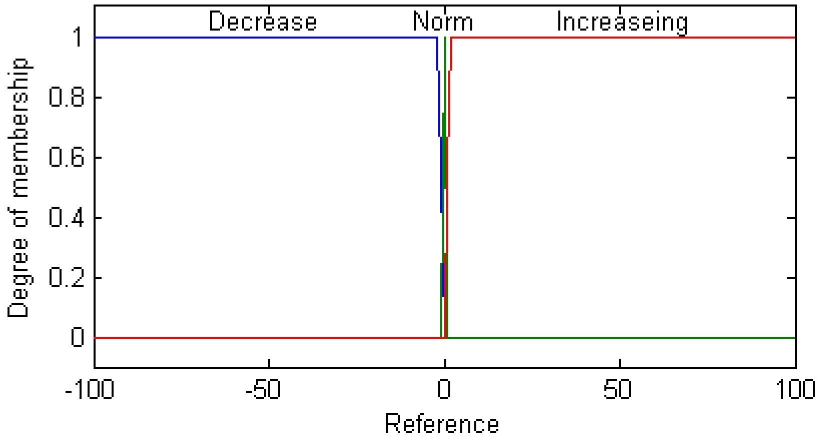


Figure (4.7g) Tuning Logic input parameter: reference signal state $\Pi_w(t)$.

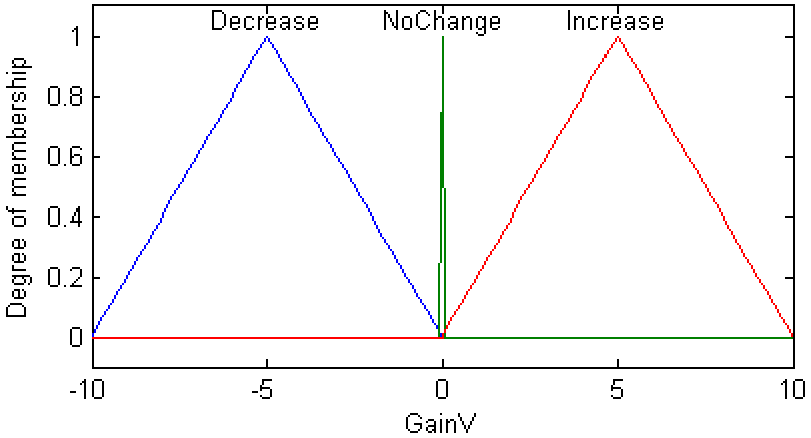


Figure (4.8a) Tuning Logic output parameter: tuning value $v_\tau(t)$ for the PID gain.

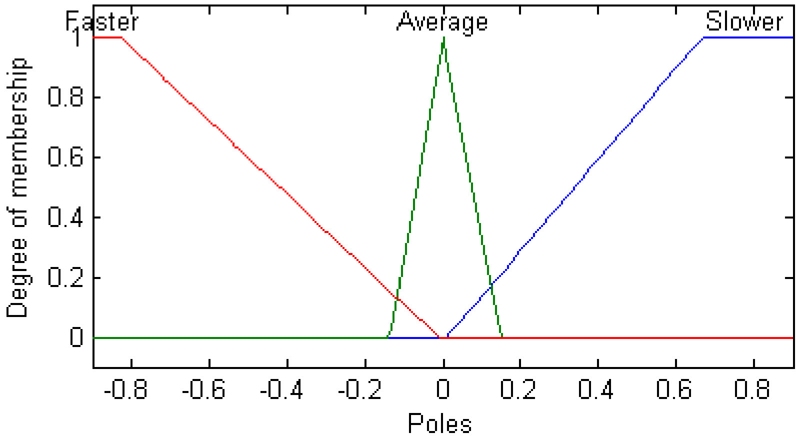


Figure (4.8b) Tuning Logic output parameter: tuning value $T_\tau(t)$ for the Poles of the Pole-Zero Placement controller.

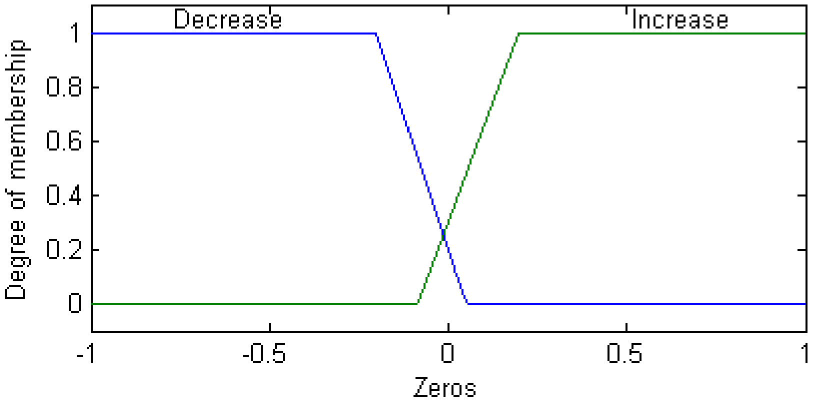


Figure (4.8c) Tuning Logic output parameter: tuning value $\widetilde{H}_\tau(t)$ for the Zeros of the Pole-Zero Placement controller.

4.3.3.2 Fuzzy Rules for the Tuning Decision

Many works have shown an interest in applying fuzzy theory to auto-tuning of active controllers. In 1994 Moudgal *et. al.* [152] used a high level fuzzy supervisor for monitoring and adjusting fuzzy controller in order to reduce the overshoot and oscillation problems in an endpoint control of a two-degree-of-freedom robot with very flexible links. The approach of fuzzy-logic based auto-tuning is also recently used by Chang and Shyu in [82] for the application of active noise cancellation, and used by Abdul-Mannan *et al.* in [83] for PI controller for high-performance induction motor drive. In [72], the PID gains K_p, K_i, and K_d were respectively calculated through fuzzy logic based on the error signal and the first difference of the error signal. The fuzzy gain scheduling approach also is widely used in aircraft industry and engine control [4]. Brdys and Littler [84] used this technique for nonlinear servo tracking where the servo controls two elements of a tracker mounted on a ship at sea.

The fuzzy rule base designed for the proposed tuning subsystem is based on fuzzy rules for controller tuning used by [4, 72, 83] for PID controllers tuning, and on the simulation experiments on the SISO water vessel system presented in chapter 6. In the list of tuning rules below, rules 1, 2 and 3 are designed for tuning the active PID controller in order to prevent the output signal overshoot and oscillation problems, as well as preserve low steady state error. Rules 4, 5 and 6 are used to tune poles of the Pole-Zero Placement controller for preserving the required output signal rise and fall times. Rules 7 and 8 tune the poles of the Pole-Zero Placement controller in order to reach the desired output signal settling time. Based on the magnitude of the control input $u(t)$ and the state of the reference signal, rules 9 to 12 are designed to tune the zeros of the Pole-Zero Placement in order to prevent excessive control actions.

Since the fuzzy sets have been defined, the fuzzy logic based tuning subsystem is completed by writing the rules that will describe the tuning value for the active controller's parameters. The complete set of 12 rules for the fuzzy logic based tuning subsystem is as follows:

- Rule 1: IF $C_\eta(t)$ IS PID AND $\zeta_y(t)$ IS Norm THEN $v_\tau(t)$ IS No-Change
- Rule 2: IF $C_\eta(t)$ IS PID AND $\zeta_y(t)$ IS NOT Norm THEN $v_\tau(t)$ IS Decrease
- Rule 3: IF $C_\eta(t)$ IS PID AND $e_\infty(t)$ IS NOT Norm THEN $v_\tau(t)$ IS Increase
- Rule 4: IF $C_\eta(t)$ IS PolZroPlcmt AND $\rho_y(t)$ IS Slow THEN $T_\tau(t)$ IS Faster
- Rule 5: IF $C_\eta(t)$ IS PolZroPlcmt AND $\rho_y(t)$ IS Average THEN $T_\tau(t)$ IS Average
- Rule 6: IF $C_\eta(t)$ IS PolZroPlcmt AND $\rho_y(t)$ IS Fast THEN $T_\tau(t)$ IS Slower
- Rule 7: IF $C_\eta(t)$ IS PolZroPlcmt AND $\tau_y(t)$ IS Slow THEN $T_\tau(t)$ IS Faster
- Rule 8: IF $C_\eta(t)$ IS PolZroPlcmt AND $\tau_y(t)$ IS Fast THEN $T_\tau(t)$ IS Slower
- Rule 9: IF $C_\eta(t)$ IS PolZroPlcmt AND $u(t)$ IS PstiveHigh THEN $\widetilde{H}_\tau(t)$ IS Increase
- Rule 10: IF $C_\eta(t)$ IS PolZroPlcmt AND $u(t)$ IS NgtiveHigh THEN $\widetilde{H}_\tau(t)$ IS Decrease
- Rule 11: IF $C_\eta(t)$ IS PolZroPlcmt AND $\Pi_w(t)$ IS Increasing THEN $\widetilde{H}_\tau(t)$ IS Decrease

- Rule 12: IF $C_{\eta}(t)$ IS PolZroPlcmt AND $\Pi_{w}(t)$ IS Decreasing THEN $\widetilde{H}_{\tau}(t)$ IS Increase

Depending on the fuzzified value of the input parameters, the tuning logic subsystem will employ the centre of gravity defuzzification procedure to generate the tuning value for the parameters of the active controller which could be either for the gain v of the PID controller, the poles T and/or zeros $\widetilde{H}$ of the PID structure based (simultaneous) pole and zero placement controller. The tuning values will be contained in the tuning signal $C\tau(t) = [T_{\tau}(t), \widetilde{H}_{\tau}(t), v_{\tau}(t)]$. The above tuning logic subsystem is applied to the SISO water vessel system presented in chapter 6.

4.3.3.3 Fuzzy Inference Procedure for the Tuning Logic

The required tuning values for the parameters of the active controller are produced through the operations of fuzzification, inference and defuzzification. The input fuzzy sets are used to quantify the tuning logic input parameters (contained in Ξ_{t}) in the rule-base, and the inference mechanism operates the rules that are relevant to the current situation in order to produce the membership of the fuzzified inputs to the output fuzzy sets. Then, the centroid defuzzification procedure is applied to generate the controller tuning values $C\tau$. Figures (4.9) and (4.10) next, respectively illustrate an example for tuning the gain v of the PID controller, and another case where the poles and zeros of the Pole-zero placement controller are to be tuned. In the case shown in Figure (4.0), only the Rules 1, 2 and 3 are stimulated due the fact that the active controller was the conventional PID controller, at that sampling time.

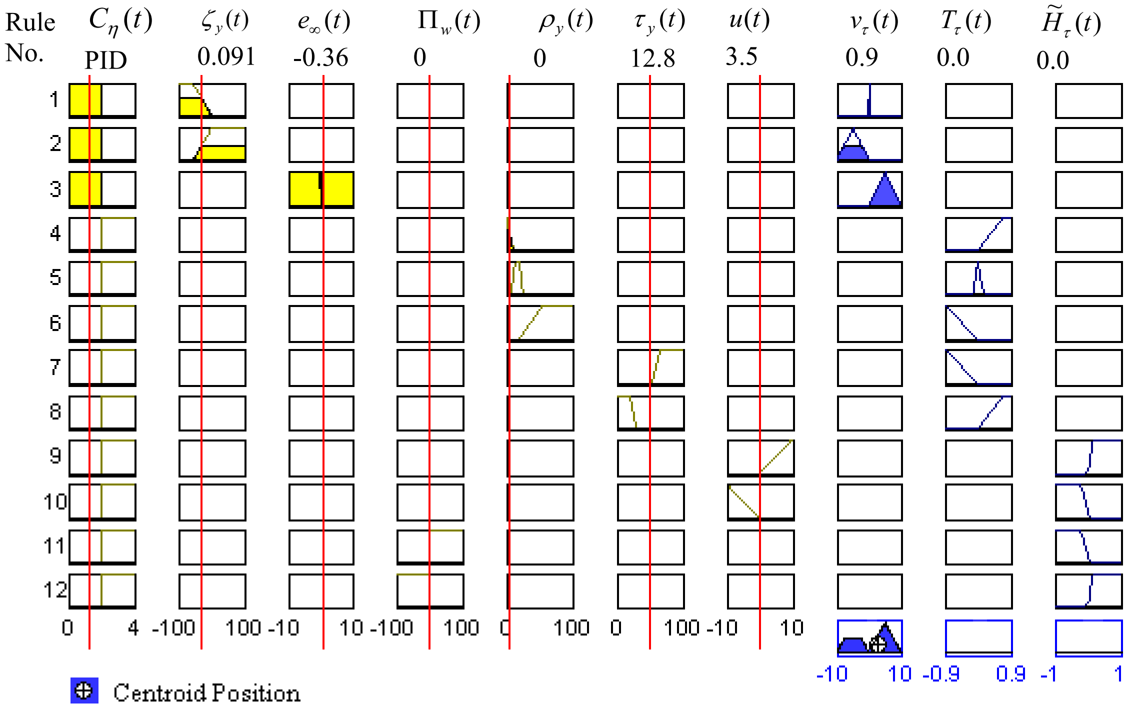


Figure (4.9): Controller parameters' tuning procedure:
tuning value for the gain *v* of the active PID controller.

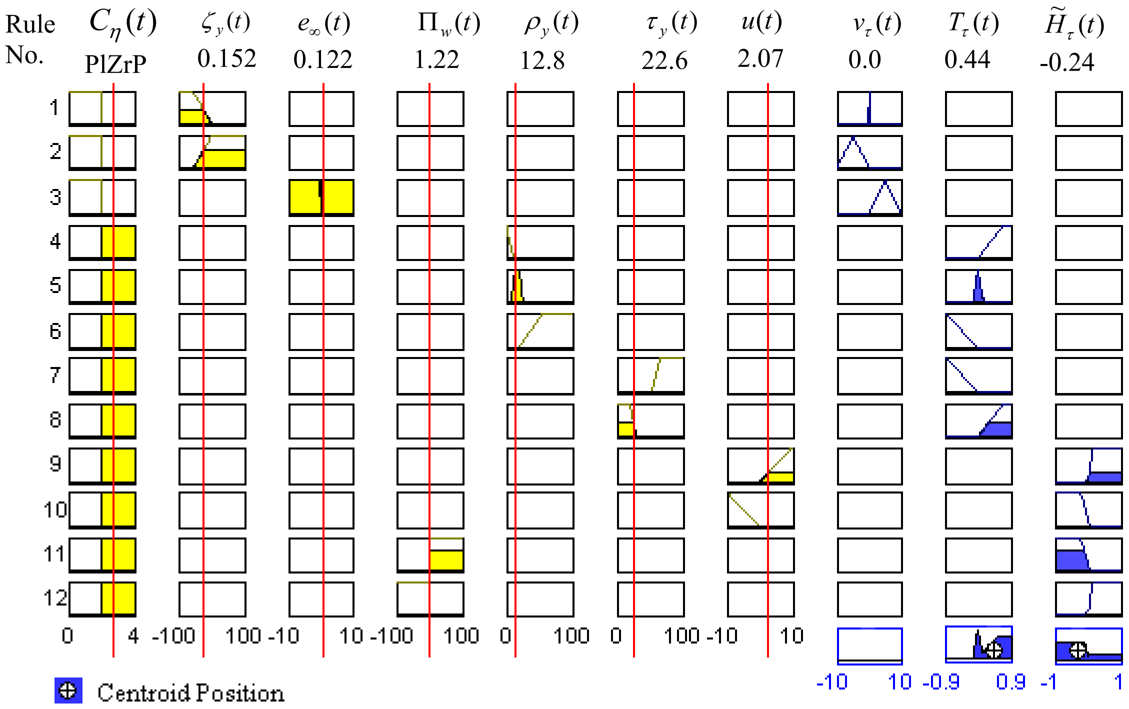


Figure (4.10): Controller parameters' tuning procedure:
tuning value for the poles and zeros of the active Pole-Zero Placement controller.

In Figure (4.10) above, rules 5, 8, 9 and 11 fired and the fuzzy sets of the Poles and Zeros output fuzzy sets were defuzzified to provide the new tuning values for the poles and zeros of the active controller, which was the Pole-Zero Placement controller.

4.4 RBF Based GLM for Complex SISO Systems Representation

As it has been discussed in chapter 2 section (2.4.4) and chapter 3 section (3.5), opposed to the MLP, the RBF NNs improve the system damping and dynamic transient stability more effectively than the MLP NNs. Control performance could be improved if the unknown nonlinear portion of the model is more accurately modelled [15]. Therefore, the RBF should be preferred to the MLP networks for the online identification of complex systems.

An RBF neural network based learning model is subsequently used to approximate the nonlinear part $f_{0,t}(.,.)$ and the disturbances $\xi(t)$. It can be seen from equation (4.1) that the measured output $y(t)$ can be obtained as follows [16]:

$$y(t+1) = \varphi^T(t)\theta + f_0(y,u)\,, \tag{4.13}$$

$$\left.\begin{array}{l} \theta(t) = [-a_1,...,-a_{n_a}, b_0,...,b_{n_b}]^T \\ \varphi^T(t) = [y\ (t-1),...,y\ (t-n_a), u\ (t-1),...,u\ (t-n_b)] \end{array}\right\}. \tag{4.14}$$

From Figure (4.12), next, we can also see that $\bar{f}_0(.,.)$ can be expressed as:

$$\bar{f}_0(.,.) = f_0 + \varepsilon(t)\,. \tag{4.15}$$

Using the above equation (4.15), and as shown in Figure (3.2), a neural network can be effectively used for estimating the nonlinear function f_0, with the identification error $\varepsilon(t)$ being used to update the weights and thresholds of the learning neural network model. The schematic diagram of the RBF neural network is shown in Figure (4.10). The network has four inputs representing the last and current states of the control input signal (i.e. $u_1(t-1)$ and $u_1(t)$) and the last and current states of the system output signal (i.e. $y_1(t-1)$, $y_1(t)$).

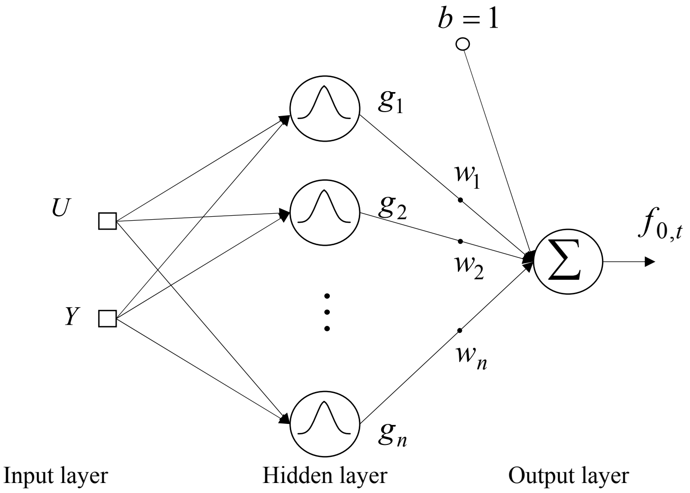


Figure (4.11): RBF neural network used in the representation of the nonlinear sub-model in the GLM.

The nonlinearity and disturbances function $f_0(.,.)$ is adaptively estimated by using the following equations [37]:

$$f_0(.,.)=\sum_{j=1}^{n} w_j g_j b\,, \tag{4.16}$$

$$\delta_w = \eta g_{j,t} f_{0,t}[\bar{f}_{0,t} - f_{0,t}][1-\bar{f}_{0,t}], \tag{4.17a}$$

$$w_j(t) = w_j(t-1) + \delta_w\,, \tag{4.17b}$$

$$g_{0,t}(t) = \exp\left(-\sum_{i=1}^{l} \left\| x_i - c_j^i \right\|^2 /(2\sigma_j^i)^2 \right), \qquad (4.18)$$

where w_j is the hidden layer weights, b is the output layer threshold, δ_w is the change in weights, η is the learning rate, x_i is the inputs, c_j^i is the centre of Gaussian basis function of the j^{th} hidden unit, l is number of inputs, and g_j is the output of the hidden layer. The variance of the Gaussian units σ_j^i is dependent on the input dimension because the RBF inputs are scaled differently [49, 12]. The Gaussian density function is in the hidden layer as an activation function as recommended by [23, 155] where they concluded that the RBF NN should be preferred to MLP NN for online system identification.

4.4.1 RBF Neural Network Parameters Setting

The parameters setting and tuning of the RBF based nonlinear sub-model is done in two stages: offline stage for selecting the number of neurons including tuning their centres and widths, and online stage for tuning the network weights. Using a trial and error approach, the RBF NN for the nonlinear sub-model of the SISO water vessel system presented in chapter six was designed with five units in the hidden layer two of which had fixed centres and the remaining three units were selected to have adaptive centres. The two fixed centres represent the smallest and the largest inputs of the training data set. The three adaptive centres were fine tuned using the back-propagation method proposed in [153] and used in [117]:

$$\Delta c_j^i = 2\frac{\alpha}{\sigma_j^i}(x_i - c_j^i)z_j[(T - f_0)w_j], \tag{4.19}$$

$$z_j = \exp\left(-\sum_l \frac{(c_j^l - x_l)^2}{(\sigma_j^l)^2}\right), \tag{4.20}$$

where $\alpha > 0$ [154] which is set to 0.02, T is the desired output and f_0 is the network output. The width of the RBF units (σ_j^i) strongly affects the performance of the RBF NNs, and in practice it is difficult to estimate the appropriate value of the RBF width [130]. Unfortunately, the literature of RBF NNS lacks the theory regarding tuning the unit width, which remains a challenging task in using RBF NNs. Haykin in [18] used the Delta rule to adjust the width of the RBF hidden layer units. In [130] used the relation $\sigma_i = \beta d_i$ where σ_i is the width of i^{th} neuron, β is a positive scalar and d_i is the minimum of distances from the i^{th} centre to its neighbours. This thesis proposes a new and simple technique for adjusting the neurons' width (σ_j^i) of RBF NN of the GLM

$$\sigma_j^i = \beta c_j^i, \tag{4.21}$$

where σ_j^i is the width of the j^{th} neuron in the RBF hidden layer, c_j^i is the position of the centre of the j^{th} neuron and β is a positive scalar. The proposed technique gives promising results in improving the RBF based nonlinear sub-model in approximating hard nonlinearities and sharp disturbances, as will be presented in simulation results in chapter (6) section (6.2.3).

The output layer weights were updated online using the Delta rule. The RBF weights can be adapted using various algorithms, a good overview of which can be found in [18].

4.5 Proposed Novel Intelligent Multiple-Controller Framework for Complex SISO Systems

The work conducted in this thesis was directed toward developing an affective and efficient intelligent control system. The new proposed intelligent multiple-controller framework for controlling complex SISO systems incorporates the fuzzy-logic based supervisor to automatically govern the selection scheme between the adaptive nonlinear PID controller and the pole-zero placement nonlinear controller, and to perform the required tuning on the parameters of the selected controller. The switching and tuning fuzzy-logic supervisor is situated at the highest level of the control system to act according to the data received from the control system and the environment, as well as the information supplied by the user. Therefore, the switching and tuning decisions are made on the basis of the closed-loop system performance measurements and the user desired performance.

In order to improve the approximation of the complex system model and consequently accomplish more accurate SISO plant representation, the proposed approach incorporates the RBF neural network based GLM. The GLM assumes that the unknown complex plant is represented by an equivalent stochastic model consisting of a linear time-varying sub-model plus a nonlinear RBF neural network learning sub-model. The block diagram of the proposed intelligent multiple-controller for SISO complex systems is shown if Figure (4.12) next. The transfer function for the proposed intelligent multiple-controller, show in Figure (4.12) above, can be expressed as follows:

$$u(t) = \frac{v(v_\tau)[H(H_\eta, \widetilde{H}_\tau)F(1)w(t) - Fy(t) + \Delta H'_N f_{0,t}(.,.)]}{\Delta q(q_\eta, T_\tau)}, \tag{4.22}$$

Where $w(t)$ is the system set point, $f_{0,t}(.,.)$ is a nonlinear function representing the nonlinear dynamics and disturbances of the complex SISO system under control, Δ is the integral action required for the PID design, F is a polynomial derived from the linear parameters of the controlled plant and includes the desired closed loop poles, $F(1)$ is the value of F at the steady state, H'_N is a user-defined polynomial. The transferred poles q were derived through the following Diophantine equation which is used to place poles of the system in the required position:

$$(q'\Delta A + z^{-1}FvB) = T, \tag{4.23}$$

where T represents the desired closed loop poles and q' is the controller polynomial.

In the multiple-controller above, $v(v_\tau)$ is the tuned PID gain and $q(q_\eta, T_\tau)$ represent the transferred poles of the control system transfer function as a function of the switching parameter q_η and the tuning parameter T_τ and can be expressed as:

$$q(q_\eta, T_\tau) = 1 + q_\eta q_1(z^{-1}) + q_\eta q_2(z^{-1}), \tag{4.24}$$

where the tuned poles $q_1(t+1) = q_1(t) + t_{\tau_1}(t)$ and $q_2(t+1) = q_2(t) + t_{\tau_2}(t)$.

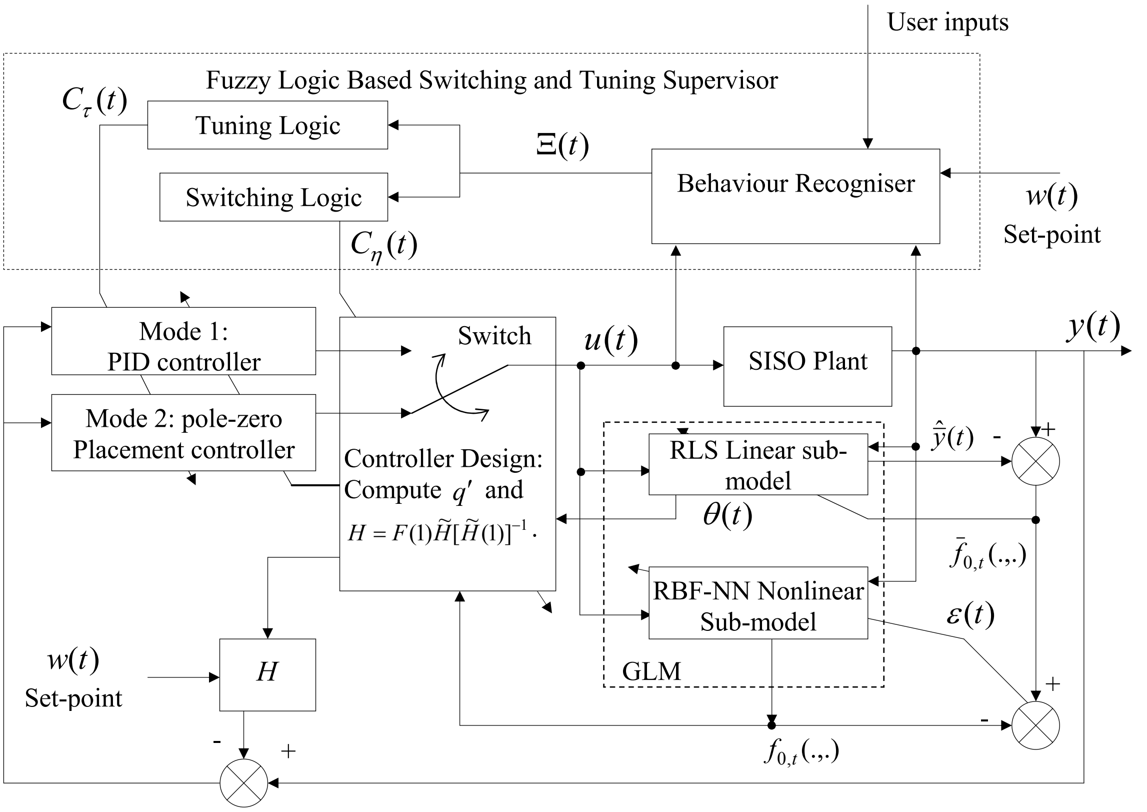


Figure (4.12): Intelligent Multiple-Controller Framework for SISO complex systems.

The term $H(H_\eta, \tilde{H}_\tau)$ represent the transfer function zeros as a function of the autonomous switching and tuning parameters, respectively, H_η and $\tilde{H}_\tau$ as it can be seen the next equation:

$$H(H_\eta, \tilde{H}_\tau) = \tilde{H}(H_\eta, \tilde{H}_\tau)[\tilde{H}(1, H_\eta, \tilde{H}_\tau)]^{-1}, \tag{4.25}$$

where $h_1(t+1) = h_1(t) + \tilde{h}_{\tau_1}(t)$ and $h_2(t+1) = h_2(t) + \tilde{h}_{\tau 2}(t)$ are the tuned zeros, $\tilde{H}(H_\eta, \tilde{H}_\tau) = 1 + H_\eta h_1(z^{-1}) + H_\eta h_2(z^{-1})$, and $\tilde{H}(1, H_\eta, \tilde{H}_\tau) = 1 + H_\eta h_1 + H_\eta h_2$.

The following subsections will outline the various multiple-controller modes which are derived from a single minimum variance based control law. The switching between these modes is performed autonomously in an integral manner by modifying the common control law using the fuzzy-logic supervisor. The automatic switching feature is a key distinction of the new proposed intelligent multiple-controller compared to other classical multiple-controllers switching paradigms which seek to combine a switch between different controllers which may lead to undesirable transfer problems during switching actions [8, 17]. The switching mechanism introduced in this work guarantees bumpless switching between the different controlling scenarios, i.e., switching that does not induce a large transient because of the compatible "initial conditions" of the controllers connected to the plant [100, 118].

4.5.1 Multiple-Controller Mode 1: Self-tuning PID Controller

In this mode, the multiple-controller operates as a conventional adaptive PID controller, which is expressed in the velocity form as:

$$u(t)=\frac{K_I w(t)-[K_P+K_I+K_D]y(t)-[-K_P-2K_D]y(t-1)-K_D y(t-2)}{\Delta}, \quad (4.26)$$

To obtain an adaptive PID controller, the degree of $F(z^{-1})$ is initially set to 2 so that

$$F(z^{-1})=f_0+f_1 z^{-1}+f_2 z^{-2}, \quad (4.27)$$

and both the pole-placement polynomial q and zero-placement polynomial $\widetilde{H}$ are switched off when the fuzzy logic switching and tuning supervisor sets the switching parameter $C_\eta=[q_\eta, H_\eta]$ to equal $[0, 0]$. Consequently,

$$\left.\begin{array}{ll} q(z^{-1})=1, & (i.e.\ q_1=q_2=..,..=q_{n_q}=0) \\ \widetilde{H}(z^{-1})=1, & (i.e.\ \widetilde{h}_1=\widetilde{h}_2=..,..=\widetilde{h}_{n_{\widetilde{h}}}=0) \end{array}\right\}. \tag{4.28}$$

Therefore, the adaptive PID controller is structured as follows:

$$u(t)=\frac{vF(1)w(t)-v(f_0+f_1z^{-1}+f_2z^{-2})y(t)+\Delta vH'_Nf_{0,t}(.,.)}{\Delta}, \tag{4.29}$$

The PID gains K_p, K_i and K_d will be as:

$$K_p=-v[f_1+2vf_2], \tag{4.30}$$

$$K_I=v[f_0+f_1+f_2], \tag{4.31}$$

$$K_D=vf_2. \tag{4.32}$$

4.5.2 Multiple-Controller Mode 2: Pole-Zero Placement Controller

To switch multiple-controller control law in equation (4.22) to the simultaneous pole and zero placement controller with a PID structure, the switching and tuning supervisor will set the autonomous switching parameter $C_\eta=[q_\eta,H_\eta]$ to be $[1,1]$. Consequently,

$$\left.\begin{array}{l} q(z^{-1})=1+q_1(z^{-1})+q_2(z^{-1}) \\ \widetilde{H}(z^{-1})=1+h_1(z^{-1})+h_2(z^{-1}) \end{array}\right\}, \tag{4.33}$$

where q_1 and q_2 are the tuned poles placed in their position, h_1 and h_2 are the tuned zero placed in their position. Therefore, the pole-zero placement controller will be in the form:

$$u(t) = \frac{v(v_\tau)[H(H_\eta, \widetilde{H}_\tau)F(1)w(t) - Fy(t) + \Delta H'_N f_{0,t}(.,.)]}{\Delta q(q_\eta, T_\tau)}, \tag{4.34}$$

Note that in practice, the order of $q(z^{-1})$ and $\widetilde{H}(z^{-1})$ are most of the time selected to equal 1 or 2 [17, 41, 43]. The PID gains K_p, K_i and K_d will be set as in the multiple-controller mode 1 above for a PID structured pole-zero placement controller.

4.5.3 Intelligent Multiple-Controller Algorithm Summary: SISO Case

The proposed intelligent multiple-controller algorithm can now be summarised into the following steps:

Step 1: Select the initial desired closed-loop system poles and zeros polynomials T and $\widetilde{H}$ respectively.

Step 2: The system current controller is initially set to work with the Pole-Zero Placement controller (i.e. $C_\eta = [1, 1]$) in order to avoid high control action, at the start of the control process, and consequently prevent output signal overshooting [100].

Step 3: Select F and the initial value for the gain v for the desired PID control structure.

Step 4: Read the current values of $y(t)$ and $w(t)$.

Step 5: Compute the control input $u(t)$ using (4.34) when the current controller is pole-zero placement, or using equation (4.29) when the controller is the conventional PID controller.

Step 6: Estimate the process linear parameters $\hat{A}$ and $\hat{B}$ using the least squares algorithm of the linear sub-model in the GLM.

Step 7: Compute $\bar{f}_0(.,.) = y(t) - \hat{\bar{y}}(t)$, where $\hat{\bar{y}}(t)$ is the output of the linear sub-model.

Step 8: Apply the RBF based nonlinear sub-model of the GLM to obtain $f_0(.,.)$ by using equations (4.16)-(4.21).

Step 9: The behaviour recogniser will assess the current performance of the control system using the system output $y(t)$, the set-point $w(t)$, the control input $u(t)$ and the user requirements.

Step 10: The behaviour recogniser will report the system performance to the switching and tuning subsystems as concluded in $\Xi(t)$.

Step 11: The fuzzy logic switching subsystem will employ $\Xi_s(t)$ to make the switching decision for the next controller to be activated, that is achieved by setting C_η to $[1,1]$ for a Pole-Zero Placement controller or to $[0,0]$ for a conventional PID controller.

Step 12: The fuzzy logic tuning subsystem will decide the tuning values for the current controller parameters' based on the input $\Xi_t(t)$.

Steps 4 to 12 are to be repeated for every sampling instant.

4.6 Summary

A central theme in the study of intelligent control is the modelling and control of complex systems. Every control system, from the simplest (e.g. the thermostat or a simple positioning servo) to the most complex currently in use (e.g. control of unmanned air vehicle) utilize feedback in one form or another. The essence of the concept involves the triad: measurement, comparison, and correction [98]. That is, measurement of relevant variables, comparison with desired values, and using the errors to correct behaviour. The complexity of the control systems used nowadays emphasise the involvement of more sophisticated and intelligent techniques, that is to cope with the measurements, comparisons, and corrections required for the control decision making process. By considering the design of a multi-controller, the automation capabilities provided by the field of Artificial Intelligence can be integrated with the concepts and techniques from this field to the multiple controller approach of designing control systems may be advantageous, from a practical perspective, to solve such complex control problems.

In this chapter a new intelligent nonlinear multiple-controller framework incorporating a fuzzy logic based switching and tuning supervisor is developed to control complex SISO systems. The framework integrates the simple fuzzy rule based supervisor with the benefits of both the conventional PID and PID structured pole-zero placement nonlinear-controllers along with a GLM framework. In the GLM, the unknown complex process to be controlled is represented by an equivalent stochastic model consisting of a linear time-varying sub-model plus a computationally-efficient RBF neural-network based learning sub-model. The proposed methodology provides the designer the choice between the conventional PID adaptive controller, or the PID structure based (simultaneous) pole and zero placement controller. Both controllers (multiple controller modes 1 and 2) benefit from the simplicity of having a PID structure, operate using the same adaptive procedure and can be selected on the basis of the required performance measure.

The switching decision between the two nonlinear fixed structure controllers, along with online tuning of the controller parameters, is made using a fuzzy logic based supervisor operating at the highest level of the system. The proposed intelligent multiple-controller works to adaptively tracking a desired reference signal, achieving the desired output signal performance and penalising excessive control actions, in response to the current performance of the control systems as assessed by the behaviour recogniser. The stability analysis of the proposed intelligent multiple-controller framework for SISO complex systems will be considered as a subset of the general multivariable case in the next chapter.

It is often the case that higher-level knowledge about how to control a process is available along with the lower-level data on which simple control systems operate [156]. In the proposed intelligent framework, the tasks of fuzzy logic based coordination between multiple-controllers and tuning of the controller parameters are based on information about the application operating points, including system transfer function poles and zeros, and the controller PID gains. Accordingly, information acquired about the system to be controlled was essential to design the fuzzy logic high-level supervisor. Taking into account the real-time implementation constraints, such as minimizing the amount of memory used and the time that it takes to compute the fuzzy outputs using the given inputs [4], the fuzzy logic based switching and tuning supervisor is designed with a minimum number of fuzzy rules with minimum input and output parameters.

The next chapter presents the intelligent multivariable multiple-controller framework for the general case of complex MIMO systems.

Chapter 5

Intelligent Multivariable Multiple-Controller Framework Incorporating a Fuzzy Logic Based Switching and Tuning Supervisor: MIMO Case

5.1 Introduction

In practice, most practical systems considered are nonlinear and multivariable in character. The synthesis of multivariable controllers has received more attention in the industrial field and more particularly in the domain of chemical engineering. Even though, it is still common practice, especially in industrial applications, to design a SISO controller for each (I/O) pair of a MIMO plant by simply ignoring the interaction between these pairs. Such SISO controllers may work satisfactorily for some MIMO plants, but advances in performance can only be achieved through the use of MIMO controllers [120]. Moreover, for MIMO complex systems, the control problem is very complicated due to the coupling among various inputs and outputs. It becomes in general very difficult to deal with when there exist uncertain parameters and/or unknown nonlinear functions in the input coupling matrix [122]. Due to these difficulties, it is noticed that in comparison with vast amount of results on controller design for SISO complex systems in the control literature, there are relatively fewer results available for the broader class of MIMO complex systems.

The design of multivariable control systems requires identification of the effects of individual inputs on each of the outputs. In many complex systems whose behaviour is described by a large set of partial differential equations, the solution cannot be implemented in real-time due to the large number of unknown parameters and constraints. Additionally, for fast real-time control, the computationally intensive model, even if available, must either be reduced, leading to approximation errors, or replaced with another model with the same input-output characteristics [119].

In the last two decades, control methodologies employing fuzzy logic systems and neural networks have been a promising way to approach complex control problems. Particularly, fuzzy logic has attracted the control community because of the simple approach it provides to use heuristic control knowledge for complex control problems [121], see chapter (2) section (2.6) for more details. Similarly, there has been tremendous interest in the study of neural networks in modelling and control of uncertain nonlinear systems with unknown nonlinearities, and great achievements have been met both in theory and practical application [122]. As it was discussed in chapter (2) section (2.5), neural networks are mostly used as approximation models for unknown nonlinearities due to their inherent approximation powers. With these capabilities, it is not necessary to spend much effort on system modelling which might be expensive in many cases.

Considering the importance of MIMO systems and the promising potentials of fuzzy logic and neural networks for control engineering, this chapter presents an intelligent multivariable multiple-controller framework for complex MIMO plants, which extend the SISO results shown in the previous chapter. The proposed multiple-controller methodology uses a Mamdani fuzzy system for the supervision of the multivariable multiple-controller and employs an RBF neural network based GLM for MIMO system representation.

This chapter is organised as follows:

The convention control law of the GMVC multivariable multiple-controller is presented in section (5.2). The new proposed fuzzy logic based supervisor for switching and tuning multivariable multiple-controller is discussed in section (5.3). The discussion will detail the design of the switching logic and tuning logic subsystems of the high level supervisor, in addition to be MIMO system behaviour recogniser subsystem. In section (5.4), the MIMO RBF neural network nonlinear sub-model for the MIMO GLM will be presented. The complete framework of the intelligent multivariable multiple-controller for the autonomous control of MIMO complex systems is illustrated in section (5.5). The structure of the developed control law will be given as well as the fuzzy logic based switching and tuning mechanisms to activate and tune the two adaptive multivariable control modes, namely convention PID controller and Pole-Zero Placement controller. Section (5.6) will present the stability analysis of the proposed intelligent framework including the two switching modes and the fuzzy logic switching and tuning mechanisms. Summary of the chapter will be given in section (5.7).

5.2 Control Law Structure of GMVC for Complex MIMO Systems

The original SISO minimum variance strategy of Aström and Wittenmark [102] was extended into a multivariable by Borrison [123] which was a stepping stone to the developments of the multivariable adaptive control theories. In the sequel, Koivo [124], Keviczky and Kumar [125] and Grimble and Moir [126] extended the SISO generalised minimum variance control proposed by Clarke and Gawthrop [103] into MIMO case. The generalised minimum variance control was also modified to have a MIMO PID structure by Yusof *et al.* [41, 43], Zhu and Warwick [19] and Zayed *et al.* [127], and extended to achieve the multivariable nonlinear pole-placement control by Zhu and Warwick [16].

Zayed *et al.* [17, 39, 46] developed a MIMO multiple-controller framework which achieved more effective control action, and combined the advantages of adaptive controller together with those of PID controllers and nonlinear Pole-Zero Placement controllers. As it will be presented in this chapter, the framework of [17, 39, 46,] is now further developed to achieve a desired closed-loop control system performance through the autonomous intelligent tuning and switching between the GMV based multiple-controllers, and the more efficient complex plant approximation using the new RBF based MIMO GLM [128].

The MIMO complex system considered in this thesis is in the following Controlled Auto-Regressive Moving (CARMA) representation which is for n input n output plant model [17]:

$$\mathbf{A}(z^{-1})\mathbf{y}(t+k) = \mathbf{B}(z^{-1})\mathbf{u}(t) + \mathbf{f}_{0,t}(Y,U) + \xi(t+k)\,, \qquad (5.1)$$

where $\mathbf{y}(t)$ is the measured output vector with dimension $(n\text{x}1)$ $\mathbf{y}(t) = [y_1(t) \quad y_2(t) \quad ... \quad y_n(t)]$, $\mathbf{u}(t)$ is the control input vector $(n\text{x}1)$ $\mathbf{u}(t) = [u_1(t) \quad u_2(t) \quad ... \quad u_n(t)]$, $\xi(t)$ is an uncorrelated sequence of random variables with zero mean at the sampling instant $t = 1,2,...$, $\xi(t) = [\xi_1(t), \xi_2(t),...,....\xi_n(t)]$, and k is the time delay of the process in the integer-sample interval. The term $\mathbf{f}_{0,t}(\mathbf{Y},\mathbf{U})$ in equation (5.1) above, is potentially a nonlinear function which accounts for any unknown time delays, uncertainty and nonlinearity in the complex MIMO plant model where $\mathbf{f}_{0,t}(\mathbf{Y},\mathbf{U}) = [f_{0,1}(.,.) \quad f_{0,2}(.,.) \quad ... \quad f_{0,n}(.,.)]$, and is conveniently represented by a Multi-Layered Perceptron [16, 17]. The overall MIMO plant model represented by equation (5.1) above, is termed the MIMO Generalized Learning Model (GLM) [37], and can be seen as the combination of a linear sub-model and a nonlinear (learning) sub-model. Also, in equation (5.1), $\mathbf{y}(t) \in \mathbf{Y}$, $\mathbf{u}(t) \in \mathbf{U}$; $\{\mathbf{Y} \in R^{n_a}; \mathbf{U} \in R^{n_b}\}$, and $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$ are $(n \times n)$ diagonal polynomial matrices with orders n_a and n_b, respectively, which can be expressed in terms of the backwards shift operator, z^{-1} as:

$$\mathbf{A}(z^{-1}) = \mathbf{I} + \mathbf{A}_1 z^{-1} + \mathbf{A}_2 z^{-2} + ... + \mathbf{A}_{n_A} z^{-n_A}\,, \qquad (5.2a)$$

$$\mathbf{B}(z^{-1}) = \mathbf{B}_0 + \mathbf{B}_1 z^{-1} + ... + \mathbf{B}_{n_B} z^{-n_B}\,,\ \mathbf{B}(0) \neq 0\,, \qquad (5.2b)$$

where n_A and n_B are the degrees of the polynomials $\mathbf{A}(z^{-1})$, $\mathbf{B}(z^{-1})$

The multivariable multiple-controller control law for the above complex MIMO plant is meanwhile given in the next equation [17]:

$$\mathbf{u}(t) = \frac{[\mathbf{v}\widetilde{\mathbf{H}}[\mathbf{H}(1)]^{-1}\mathbf{F}(1)\mathbf{w}(t) - \mathbf{v}\mathbf{F}\mathbf{y}(t) + \Delta\mathbf{v}\mathbf{H}'_{\mathrm{N}}\mathbf{f}_{0,t}(.,.)]}{\Delta\mathbf{q}'}, \tag{5.3}$$

where $\mathbf{w}(t)$ is the $(n \times 1)$ system bounded set point vector, $\mathbf{f}_{0,t}(.,.)$ is a nonlinear function representing the complex dynamics of the MIMO plant, $\mathbf{v}$ is a user-defined gain matrix, Δ is the integral action required for the PID design, $\widetilde{\mathbf{H}}$ is a user-defined polynomial which can be used to introduce arbitrary closed loop zeros for an explicit Pole-Zero Placement controller, $\mathbf{H}(1)$ is the value of $\widetilde{\mathbf{H}}$ at system output steady state, $\mathbf{F}$ is a polynomial derived from the linear parameters of the controlled plant and includes the desired closed loop poles, $\mathbf{F}(1)$ is the value of $\mathbf{F}$ at the steady state, $\mathbf{H}'_{\mathrm{N}}$ is a user-defined polynomial for activating the nonlinear function $\mathbf{f}_{0,t}(.,.)$. The parameter $\mathbf{q}'$ is a transfer function used to bring the closed loop system parameters in the stability unit disc, and is a polynomial in z^{-1} having the following form: $\mathbf{q}'(z^{-1}) = \mathbf{I} + \mathbf{q}'_1 z^{-1} + \mathbf{q}'_2 z^{-2} + \ldots + \mathbf{q}'_{n_{\mathbf{q}'}} z^{-n_{\mathbf{q}'}}$ where $n_{q'}$ is the degree of the polynomial $\mathbf{q}'$.

5.3 New MIMO Fuzzy Logic Based Supervisor for Multiple-Controllers Switching and Tuning

The proposed MIMO fuzzy-logic based switching and tuning supervisor for the multivariable multiple-controller operates at the highest level of the control system framework. The fuzzy supervisor monitors the MIMO control system through the available input-output data, and then characterises the system current behaviour so that it knows which controller to choose, which parameters to tune, and the tuning value for each parameter that is required to ultimately achieve the desired specification. The main idea behind the fuzzy-logic supervisor approach here is to employ logic-based tuning and switching between the candidate multivariable controllers.

The supervisor, which is employed for the MIMO control system, comprises three subsystems: a behaviour recogniser, a switching logic and a tuning logic, each of which are discussed next. Based on similar system performance criteria presented in chapter (4) section (4.2), for each control input and system output, the supervisor aims to configure the best controller, or when the current active controller is not properly tuned. Consequently, the supervisor seeks to switch to the candidate controller and/or adjust the controller parameters to obtain improved performance. The whole supervisor is implemented using simple fuzzy logic based switching and tuning rules where the premises of the rules form part of the behaviour recogniser and the consequent form the switching and tuning decisions. In this way, a fuzzy logic system is used to implement the entire supervisory control level of the multivariable multiple-controller system.

5.3.1 Behaviour Recogniser for MIMO Systems

The behaviour recogniser will characterise the current behaviour of the MIMO control system in a way that will be useful to the switching and tuning logic subsystems. The proposed behaviour recogniser subsystem benefits from the system performance criteria discussed in chapter (4) section (4.2). The multivariable system has n control input signals and n system output signals and characterised through the online measurements of following parameters:

- Degree of overshoot ($\boldsymbol{\zeta}_y$) of the MIMO system output signals:

$\boldsymbol{\zeta}_y = [\zeta_{y_1} \quad \zeta_{y_2} \quad \cdots \quad \zeta_{y_n}]$ which is an $n \times 1$ vector where n is the number of system output signals.

$$\zeta_{y_i}(t) = \frac{y_{\max_i} - y_{\infty_i}}{y_{\infty_i}} 100, \tag{5.4}$$

where $y_{\max_i}$ is the amplitude maximum value reached at the i^{th} ($i = 1,2,...n$) output signal of the MIMO plant, and y_{∞_i} is the steady state value of the i^{th} output signal.

- Rise and fall time of the MIMO system output signals ($\boldsymbol{\rho}_y$):

The vector $\boldsymbol{\rho}_y = [\rho_{y_1} \quad \rho_{y_2} \quad \cdots \quad \rho_{y_n}]$ is an $n \times 1$ vector denotes the current rise or fall time of the n system output signals. The output signal rise and fall times represent the amount of time for a signal to change state. To measure rise time, the behaviour recogniser uses 10% to the 90% point of every output signal, or vice versa for the output signal fall time.

- Settling times of the MIMO system output signals ($\boldsymbol{\tau}_y$):

where $\boldsymbol{\tau}_y = [\tau_{y_1} \quad \tau_{y_2} \quad \cdots \quad \tau_{y_n}]$ and τ_{y_i} is the time required for a measured process output $y_i(t)$ to first enter and then remain within a band Δ_{y_i} whose width is computed as $\pm 5\%$ of the total change in $y_i(t)$.

- Steady state error of the output signals ($\mathbf{e}_\infty$):

$\mathbf{e}_\infty = [e_{\infty_1} \quad e_{\infty_2} \quad \cdots \quad e_{\infty_n}]$ where e_{∞_i} is the difference between the desired output $w_i(t)$ and the actual output $y_i(t)$ as time goes to infinity (i.e. when the output reached its steady state) [114]. The steady state error formula can be expressed as [115]:

$$e_{\infty_i}(t) = \lim_{t\to\infty}(w_i(t) - y_i(t)) \cong 0\,. \tag{5.5}$$

- The variance of the output signals ($\mathbf{V}_y$):

The variance of sampled population of the output signal $y_i(t)$ is the mean squared deviation of the individual values y_{i_l} of $y_i(t)$ from the population mean. The mean is considered to be the steady state value y_{∞_i}. Therefore, V_{y_i} is computed as follows [116]:

$$V_{y_i}(t) = \frac{\sum_{l=1}^{N}(y_{i_l} - y_{\infty_i})^2}{N-1}, \tag{5.6}$$

where N denotes the size of the sampled population of the output signal $y_i(t)$ and $\mathbf{V}_y = [V_{y_1} \quad V_{y_2} \quad \cdots \quad V_{y_n}]$.

- Control input signals of the MIMO system (control actions $\mathbf{u}(t)$).

$$\mathbf{u}(t) = [u_1(t) \quad u_2(t) \quad \ldots \; u_n(t)]^T . \tag{5.7}$$

- The state of the reference Signals ($\mathbf{\Pi}_w$):

By comparing the current set-points $w_i(t)$, $i = 1,2,...n$, with the previous ones $w_i(t-1)$, the behaviour recogniser will check the state of the each reference signal whether it is increasing, decreasing or remaining as it was in the last state.

$$\Pi_{w_i}(t) = w_i(t) - w_i(t-1) . \tag{5.7}$$

The tested states of the reference signals are stored as in $\mathbf{\Pi}_w = [\Pi_{w_1} \quad \Pi_{w_2} \quad \cdots \quad \Pi_{w_n}]$.

The behaviour recogniser subsystem output is contained in the matrix $\Xi(t)$ and expressed as:

$$\Xi(t) = \begin{bmatrix} \zeta_{y_1}(t) & \rho_{y_1}(t) & \tau_{y_1}(t) & e_{\infty_1}(t) & V_{y_1}(t) & \Pi_{w_1}(t) \\ \zeta_{y_2}(t) & \rho_{y_2}(t) & \tau_{y_2}(t) & e_{\infty_2}(t) & V_{y_2}(t) & \Pi_{w_2}(t) \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \zeta_{y_n}(t) & \rho_{y_n}(t) & \tau_{y_n}(t) & e_{\infty_n}(t) & V_{y_n}(t) & \Pi_{w_n}(t) \end{bmatrix} . \tag{5.9}$$

The performance measure variable $\Xi(t)$ will be used by both fuzzy-logic based switching and tuning subsystems.

5.3.2 Fuzzy Logic Based Switching Subsystem for Multivariable multiple-controller

The switching logic subsystem for the multivariable multiple-controller is designed according to the switching criteria presented in chapter (4) section (4.2.2), and used by Abdullah *et al.* in [128]. The switching logic generates the switching signal $\mathbf{C}_\eta$ which determines, at each instant of time, the candidate controller module for every subsystem in the MIMO plant. The switching logic is implemented using fuzzy logic rules where the premises of the rules use $4 \text{x} n$ variables from the output of the behaviour recogniser (i.e. $\boldsymbol{\Xi}(t)$). These variables will represent the input parameters of the switching logic, which can expressed as:

$$\boldsymbol{\Xi}_s(t) = \begin{bmatrix} \zeta_{y_1}(t) & e_{\infty_1}(t) & V_{y_1}(t) & \Pi_{w_1}(t) \\ \zeta_{y_2}(t) & e_{\infty_2}(t) & V_{y_2}(t) & \Pi_{w_2}(t) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \zeta_{y_n}(t) & e_{\infty_n}(t) & V_{y_n}(t) & \Pi_{w_n}(t) \end{bmatrix}. \tag{5.10}$$

The consequents of the fuzzy logic rules form the controller selection decision which is symbolized as $\mathbf{C}_\eta$ where

$$\mathbf{C}_\eta = \begin{bmatrix} \mathbf{q}_\eta & \mathbf{H}_\eta \end{bmatrix} = \begin{bmatrix} q_{\eta_1} & H_{\eta_1} \\ q_{\eta_2} & H_{\eta_2} \\ \vdots & \vdots \\ q_{\eta_n} & H_{\eta_n} \end{bmatrix}, \tag{5.11}$$

$\mathbf{q}_\eta = [q_{\eta_1} \quad q_{\eta_2} \quad \dots \quad q_{\eta_n}]^T$, $\mathbf{H}_\eta = [H_{\eta_1} \quad H_{\eta_2} \quad \dots \quad H_{\eta_n}]^T$ and $i = 1,2,\dots n$ (n is the number of control inputs). Based on the size of n and the system performance, the supervisor will switch either to the conventional PID controller, or to the PID structure based (simultaneous) pole and zero placement controller. For instance, $\mathbf{C}_\eta = \begin{bmatrix} 0 & 1 & \dots & 1 \\ 0 & 1 & \dots & 1 \end{bmatrix}^T$ will set the first control input to be controlled by a convention PID controller and the rest of the control inputs to be Pole-Zero Placement controllers. Another example is that, $\mathbf{C}_\eta = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \end{bmatrix}^T$ will set all control inputs to be controlled by Pole-Zero Placement controllers, and $\mathbf{C}_\eta = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}^T$ will set all control inputs to be controlled by a conventional PID controller.

5.3.2.1 Fuzzy Sets for the Switching Logic of the MIMO Systems

To simplify the design of the fuzzy supervisor for the multivariable multiple-controller, the switching logic fuzzy sets of the SISO case, presented in chapter (4) section (4.3.2.1), have been used for the MIMO system. Depends on the MIMO application, different fuzzy set membership functions can be designed for every fuzzy Rule input in $\Xi_s(t)$.

5.3.2.2 Fuzzy Rules for the Switching Decision

The fuzzy switching subsystem for automatic switching between the multivariable multiple-controller is completed by writing the rules that describe the fuzzy output of each combination of the fuzzy input variables. The fuzzy rules for MIMO system multivariable multiple-controllers switching are designed on the bases of the fuzzy rules for the SISO system multiple-controller switching detailed in chapter 5 at section (4.3.2.2). The final derived rules are experimentally implemented on the two MIMO applications given in chapter 6; namely: MIMO water vessel application (section (6.3)) with $n = 2$, and autonomous vehicle control application (section (6.4)) with $n = 3$, where n is the number of inputs and outputs in the MIMO applications. The complete set of fuzzy rules for the fuzzy logic based switching subsystem is as follows:

- Rule 1: IF $\zeta_{y_1}(t)$ OR $\zeta_{y_2}(t)$,…, $\zeta_{y_n}(t)$ IS Ntive-High AND $V_{y_1}(t)$ OR $V_{y_2}(t)$,…, $V_{y_n}(t)$ IS High THEN $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$,…, $C_{\eta_n}(t)$ IS Pole-Zero-Placement

- Rule 2: IF $\zeta_{y_1}(t)$ OR $\zeta_{y_2}(t)$,…, $\zeta_{y_n}(t)$ IS High AND $V_{y_1}(t)$ OR $V_{y_2}(t)$,…, $V_{y_n}(t)$ IS High THEN $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$,…, $C_{\eta_n}(t)$ IS Pole-Zero-Placement

- Rule 3: IF $\Pi_{w_1}(t)$ OR $\Pi_{w_1}(t)$,…, $\Pi_{w_n}(t)$ IS NOT Norm THEN $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$,…, $C_{\eta_n}(t)$ IS Pole-Zero-Placement

- Rule 4: IF $\zeta_{y_1}(t)$ OR $\zeta_{y_2}(t)$,…, $\zeta_{y_n}(t)$ IS Norm AND $V_{y_1}(t)$ OR $V_{y_2}(t)$,…, $V_{y_n}(t)$ IS Norm THEN $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$,…, $C_{\eta_n}(t)$ IS PID

- Rule 5: IF $e_{\infty_1}(t)$ OR $e_{\infty_2}(t)$,…, $e_{\infty_n}(t)$ IS Norm THEN $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$,…, $C_{\eta_n}(t)$ IS PID

Depending on the fuzzified value of the fuzzy input parameters, the switching logic subsystem will switch either to the conventional PID controller, or PID structure based (simultaneous) Pole-Zero Placement controller.

5.3.2.3 Inference Procedure for the Switching Logic of the Multivariable Multiple-Controller

The MIMO system's switching logic works, at each instant of time, to activate the appropriate controllers according the desired performance. The input parameters of this switching logic are computed and compared to desired values using the fuzzy sets. The fuzzy input parameters in the antecedent part of the fuzzy rules (such as ζ_{y_i} and V_{y_i}) will be related to their corresponding fuzzy output parameters in the consequent part (such as C_{η_i}) through the logic operation OR. The fuzzy rule premises' are connected by an AND are combined by taking the degree of membership of the lesser of the two as the value of combination. On the other hand, the active rules are combined together with an OR to take the larger of the output values as the value of the combination at each point on the horizontal axis. The Middle-of-the-Max defuzzification approach is used to finalise the controller selection decision.

In the example shown in Figure (5.1) below for a MIMO system with two inputs and two outputs, the first output signal overshoot $\zeta_{y_1}(t)$ was in Ntive-High fuzzy region, its variance $V_{y_1}(t)$ was in High fuzzy region and NOT Norm steady-state error $e_{\infty_1}(t)$. Whereas for the second output signal, it had very low overshooting $\zeta_{y_2}(t)$, low variance $V_{y_2}(t)$ and Norm steady-state error $e_{\infty_2}(t)$. This combination causes rules 1, 2, 4 and 5 to fire. The four rules were defuzzified to activate a Pole-Zero Placement controller for the first input-output subsystem ($C_{\eta_1}(t)$ is PolZroPlcmnt) and the conventional PID controller for the second input-output subsystem ($C_{\eta_2}(t)$ is PID).

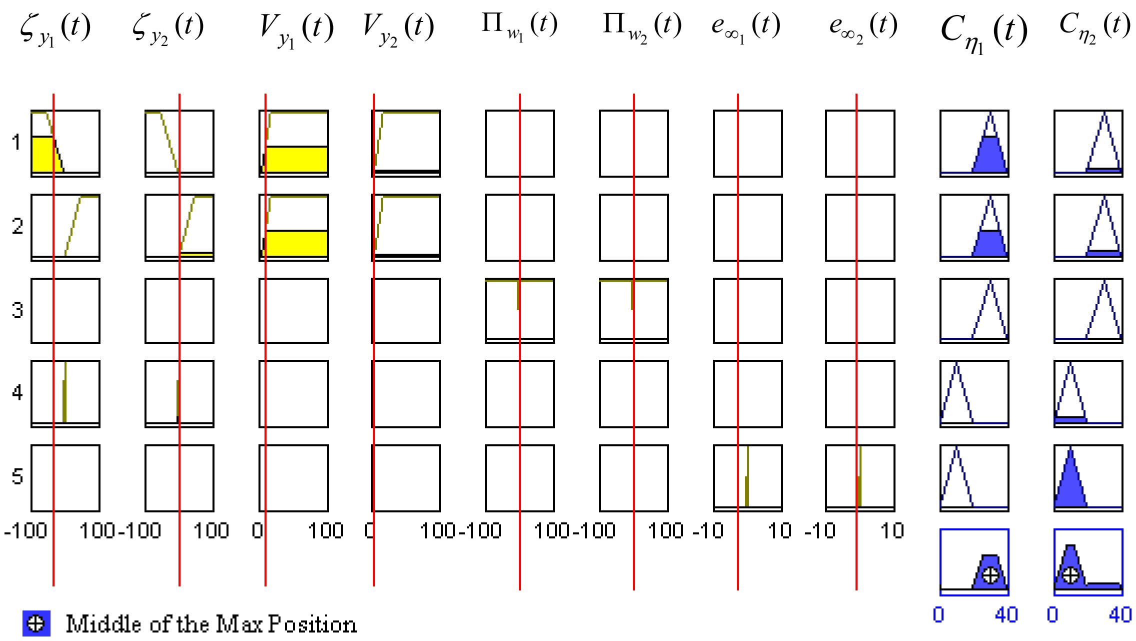


Figure (5.1): Multivariable multiple-controller selection procedure using the fuzzy-logic based switching logic subsystem.

5.3.3 Fuzzy Logic Based Multivariable Multiple-Controller Tuning Subsystem

The input parameters of the fuzzy tuning subsystem are contained in Ξ_t, which includes: part of the measurements supplied by the behaviour recogniser $\Xi(t)$, the output of the switching logic $\mathbf{C}_\eta$ (i.e. the active controllers), and the current control input signals $\mathbf{u}(t) = [u_1(t) \quad u_2(t) \quad . \quad . \quad . \quad u_n(t)]^T$.

$$\boldsymbol{\Xi}_t(t) = \begin{bmatrix} \zeta_{y_1}(t) & \rho_{y_1}(t) & \tau_{y_1}(t) & e_{\infty_1}(t) & \Pi_{w_1}(t) & C_{\eta_1}(t) & u_1(t) \\ \zeta_{y_2}(t) & \rho_{y_2}(t) & \tau_{y_2}(t) & e_{\infty_2}(t) & \Pi_{w_2}(t) & C_{\eta_2}(t) & u_2(t) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \zeta_{y_n}(t) & \rho_{y_n}(t) & \tau_{y_n}(t) & e_{\infty_n}(t) & \Pi_{w_n}(t) & C_{\eta_n}(t) & u_n(t) \end{bmatrix}. \tag{5.12}$$

The tuning logic will specify the tuning values for the parameters of the active/selected controllers. The new tuning values are contained in the tuning signal $\mathbf{C}\tau$, which represents the output parameters of the fuzzy tuning subsystem and expressed as:

$$\mathbf{C}\tau = \begin{bmatrix} T_{\tau_1}(t) & \widetilde{H}_{\tau_1}(t) & v_{\tau_1}(t) \\ T_{\tau_2}(t) & \widetilde{H}_{\tau_2}(t) & v_{\tau_2}(t) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ T_{\tau_n}(t) & \widetilde{H}_{\tau_n}(t) & v_{\tau_n}(t) \end{bmatrix}, \tag{5.13}$$

where $\mathbf{T}_\tau(t) = [T_{\tau_1}(t) \quad T_{\tau_2}(t) \quad \cdots \quad T_{\tau_n}(t)]^T$ and $\widetilde{\mathbf{H}}_\tau(t) = [\widetilde{H}_{\tau_1}(t) \quad \widetilde{H}_{\tau_2}(t) \quad \cdots \quad \widetilde{H}_{\tau_n}(t)]^T$ respectively represent the tuning values for the poles and zeros of the active Pole-Zero Placement multivariable controllers, and $\mathbf{v}_\tau(t) = [v_{\tau_1}(t) \quad v_{\tau_2}(t) \quad \cdots \quad v_{\tau_n}(t)]^T$ is the tuning values for the gains of the active multivariable PID controllers.

5.3.3.1 Fuzzy Membership Functions for the MIMO System Tuning Logic

To simplify the design of the fuzzy supervisor for the multivariable multiple-controller, the tuning logic fuzzy sets of the SISO case, presented in chapter (4) section (4.3.3.1), have been used for the tuning logic of the MIMO control system.

5.3.3.2 Fuzzy Rules for the Tuning Logic of the Multivariable Multiple-Controller

The fuzzy logic based tuning subsystem is completed by writing the rules which will prescribe the tuning values for the parameters of the active controllers. The complete set of rules for the fuzzy logic based tuning subsystem for the multivariable multiple-controller is given below:

- Rule 1: IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$,..., $C_{\eta_n}(t)$ IS PID AND $\zeta_{y_1}(t)$ OR $\zeta_{y_2}(t)$,..., $\zeta_{y_n}(t)$ IS Norm THEN $v_{\tau_1}(t)$ OR $v_{\tau_2}(t)$,..., $v_{\tau_n}(t)$ IS No-Change
- Rule 2: IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$,..., $C_{\eta_n}(t)$ IS PID AND $\zeta_{y_1}(t)$ OR $\zeta_{y_2}(t)$,..., $\zeta_{y_n}(t)$ IS NOT Norm THEN $v_{\tau_1}(t)$ OR $v_{\tau_2}(t)$,..., $v_{\tau_n}(t)$ IS Decrease
- Rule 3: IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$,..., $C_{\eta_n}(t)$ IS PID AND $e_{\infty_1}(t)$ OR $e_{\infty_2}(t)$,..., $e_{\infty_n}(t)$ IS NOT Norm THEN $v_{\tau_1}(t)$ OR $v_{\tau_2}(t)$,..., $v_{\tau_n}(t)$ IS Increase
- Rule 4: IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$,..., $C_{\eta_n}(t)$ IS PolZroPlcmt AND $\rho_{y_1}(t)$ OR $\rho_{y_2}(t)$,..., $\rho_{y_n}(t)$ IS Slow THEN $T_{\tau_1}(t)$ OR $T_{\tau_2}(t)$,..., $T_{\tau_n}(t)$ IS Faster
- Rule 5: IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$,..., $C_{\eta_n}(t)$ IS PolZroPlcmt AND $\rho_{y_1}(t)$ OR $\rho_{y_2}(t)$,..., $\rho_{y_n}(t)$ IS Average THEN $T_{\tau_1}(t)$ OR $T_{\tau_2}(t)$,..., $T_{\tau_n}(t)$ IS Average

- Rule 6: IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t),\ldots,$ $C_{\eta_n}(t)$ IS PolZroPlcmt AND $\rho_{y_1}(t)$ OR $\rho_{y_2}(t),\ldots,$ $\rho_{y_n}(t)$ IS Fast THEN $T_{\tau_1}(t)$ OR $T_{\tau_2}(t),\ldots,$ $T_{\tau_n}(t)$ IS Slower
- Rule 7: IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t),\ldots,$ $C_{\eta_n}(t)$ IS PolZroPlcmt AND $\tau_{y_1}(t)$ OR $\tau_{y_2}(t),\ldots,$ $\tau_{y_n}(t)$ IS Slow THEN $T_{\tau_1}(t)$ OR $T_{\tau_2}(t),\ldots,$ $T_{\tau_n}(t)$ IS Faster
- Rule 8: IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t),\ldots,$ $C_{\eta_n}(t)$ IS PolZroPlcmt AND $\tau_{y_1}(t)$ OR $\tau_{y_2}(t),\ldots,$ $\tau_{y_n}(t)$ IS Fast THEN $T_{\tau_1}(t)$ OR $T_{\tau_2}(t),\ldots,$ $T_{\tau_n}(t)$ IS Slower
- Rule 9: IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t),\ldots,$ $C_{\eta_n}(t)$ IS PolZroPlcmt AND $u_1(t)$ OR $u_2(t),\ldots,$ $u_n(t)$ IS PstiveHigh THEN $\widetilde{H}_{\tau_1}(t)$ OR $\widetilde{H}_{\tau_2}(t),\ldots,$ $\widetilde{H}_{\tau_n}(t)$ IS Increase
- Rule 10: IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t),\ldots,$ $C_{\eta_n}(t)$ IS PolZroPlcmt AND $u_1(t)$ OR $u_2(t),\ldots,$ $u_n(t)$ IS NgtiveHigh THEN $\widetilde{H}_{\tau_1}(t)$ OR $\widetilde{H}_{\tau_2}(t),\ldots,$ $\widetilde{H}_{\tau_n}(t)$ IS Decrease
- Rule 11: IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t),\ldots,$ $C_{\eta_n}(t)$ IS PolZroPlcmt AND $\Pi_{w_1}(t)$ OR $\Pi_{w_2}(t),\ldots,$ $\Pi_{w_n}(t)$ IS Increasing THEN $\widetilde{H}_{\tau_1}(t)$ OR $\widetilde{H}_{\tau_2}(t),\ldots,$ $\widetilde{H}_{\tau_n}(t)$ IS Decrease
- Rule 12: IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t),\ldots,$ $C_{\eta_n}(t)$ IS PolZroPlcmt AND $\Pi_{w_1}(t)$ OR $\Pi_{w_2}(t),\ldots,$ $\Pi_{w_n}(t)$ IS Decreasing THEN $\widetilde{H}_{\tau_1}(t)$ OR $\widetilde{H}_{\tau_2}(t),\ldots,$ $\widetilde{H}_{\tau_n}(t)$ IS Increase

Depending on the fuzzified value of the fuzzy rules input parameters, the tuning logic subsystem will employ the centroid defuzzification procedure to generate the tuning value for the parameters of the active controllers which include the PID gains $\mathbf{v}$ of the multivariable PID controller, the poles in $\mathbf{T}$ and/or the zeros in $\widetilde{\mathbf{H}}$ of multivariable

Pole-Zero Placement controller. The tuning values will be contained in the tuning signal $\mathbf{C}\tau$ in equation (5.13) above.

5.3.3.3 Inference Procedure for the Tuning Logic of the Multivariable Multiple-Controller

The fuzzy logic based tuning subsystem for adjusting the parameters of the multivariable multiple-controller uses the fuzzy input parameter Ξ_t to prescribe the new tuning values (i.e. $\mathbf{C}\tau$) for the parameters of the active multivariable controller. The Mamdani fuzzy system operations of fuzzification, inference and defuzzification were used to design the tuning logic for MIMO plants. The input fuzzy sets are used to quantify the tuning logic input parameters (contained in Ξ_t) in the rule-base, and the inference mechanism operates the rules that are relevant to the current situation in order to produce the membership of the fuzzified inputs to the output fuzzy sets. Then, the centroid defuzzification procedure is applied to generate the controller tuning values $\mathbf{C}\tau$.

Figures (5.2) below shows an example a MIMO system with two control inputs and two system outputs. The tuning logic subsystem employed to tune the poles and zeros of the active Pole-Zero Placement controller simultaneously with the PID gain of the active PID controller. In this case, depending on the selected controller and the fuzzy inputs (premises of the 12 fuzzy rules) only rules 1, 2, 3, 5, 8, 9, and 11 fired. Then, the fuzzified regions combined together with an OR to form the output regions. Therefore, the centroid defuzzification approach was used to produce the final tuning values $T_{\tau_1}(t)$,

$\widetilde{H}_{\tau_1}(t)$ for the poles and zeros of the active Pole-Zero Placement controller ($C_{\eta_1}(t)$) and $v_{\tau_2}(t)$ for the PID gain of the other controller $C_{\eta_2}(t)$ which was a conventional PID controller.

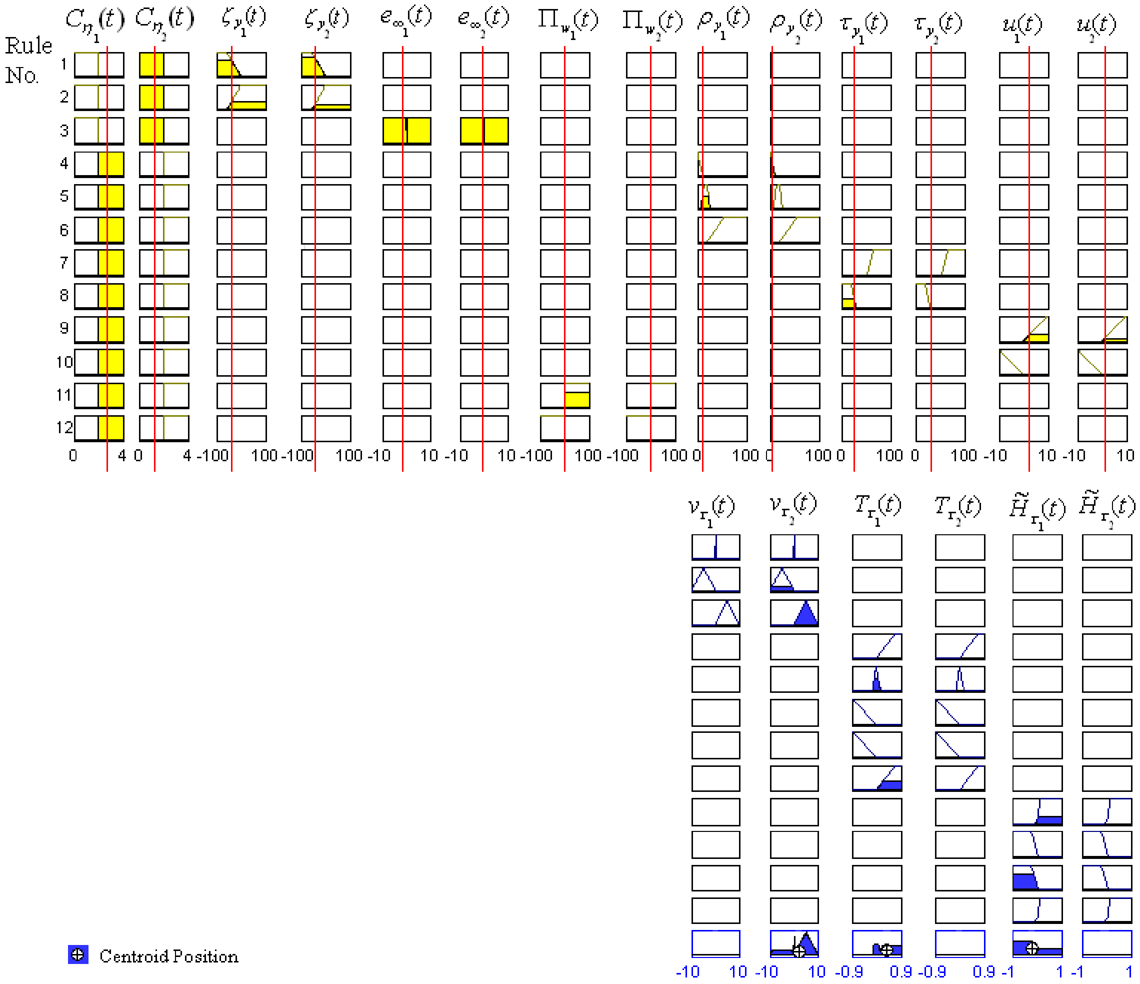


Figure (5.2): Multivariable multiple-controller tuning procedure.

5.4 RBF Based GLM for Complex MIMO Systems Representation

RBF neural network based learning model is subsequently used to approximate the nonlinear part $\mathbf{f}_{0,t}(\mathbf{y},\mathbf{u})$ of equation (5.3).

The RBF based MIMO neural network is effectively used for estimating the non-linear function $\mathbf{f}_0$, with the identification error $\boldsymbol{\varepsilon}(t)$ being used to update the weights and thresholds of the learning neural network model. The neural network model employed in the proposed control scheme is chosen to be the computationally less expensive linear-in-parameters RBF neural network as opposed to the MLP neural network previously used by [16, 17]. The RBF NNs improve the system damping and dynamic transient stability more effectively than the MLP NNs. Also, the RBF requires fewer computational complexities and elapsed time to train the network on-line, than the MLP [23]. The RBF NNs ability to uniformly approximate smooth functions over compact sets is well documented in the literature (see for example [48]).

The schematic diagram of the RBF neural network is shown in Figure (5.3), where the non-linear function $\mathbf{f}_{0,t}(.,.)$ is adaptively estimated by using the following equations [37]:

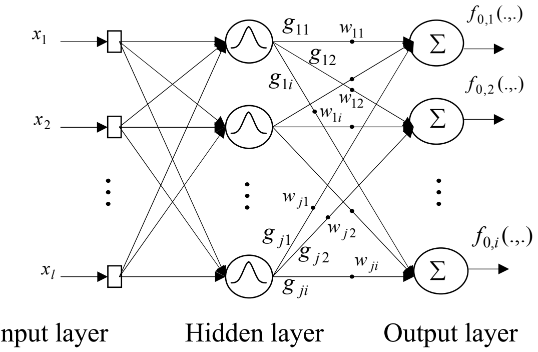


Figure (5.3): RBF Neural network based learning model to approximate the non-linear function $\mathbf{f}_{0,t}(.,.)$.

$$f_{0,i}(.,.) = \sum_{j=1}^{n} w_{ji} g_{ji} \beta , \tag{5.14}$$

$$\delta_i = \eta g_j f_{0,i}[\bar{f}_{0,i} - f_{0,i}][1 - \bar{f}_{0,i}], \tag{5.15}$$

$$w_{ji}(t) = w_{ji}(t-1) + \delta_i, \tag{5.16}$$

$$g_{ji}(t) = \exp \sum_{i=1}^{l} \frac{\left\| x_i - c_j^i \right\|^2}{(2\sigma_j^i)^2}, \tag{5.17}$$

where w_{ji} is the hidden layer weights, β is the output layer threshold, δ_i is the change in weights, η is the learning rate, x_i is the inputs, c_j^i is the centre of Gaussian basis function of the j^{th} hidden unit, l is the number of inputs, and g_{ji} is the output of the hidden layer. The variance of the Gaussian units σ_j^i is dependent on the input dimension because the RBF inputs are scaled differently [49, 12].

5.4.1 RBF Neural Network Parameters Setting

The parameters setting and tuning of the MIMO RBF neural network is achieved using the same procedure applied for the SISO RBF neural network, as mentioned in section (4.4.1) above. The MIMO RBF based nonlinear sub-model is designed with twelve units in the hidden layer four of which had fixed centres and the remaining eight units were selected to have adaptive centres which tuned offline using Equations (4.19 and 4.20) in chapter 4 at section (4.4.1). Two of the fixed centres represent the smallest and the largest inputs of the training data set, and the other two fixed centres are situated in the middle of the input space of the training data set. The output layer weights were updated online using the Delta rule. This MIMO RBF design is used for the simulation experiments of the two inputs and two outputs MIMO water vessel system presented in chapter six. The neural network based nonlinear sub-model of the MIMO GLM had eight inputs and two outputs. The inputs were the last and current of the two control input signals (i.e. $u_1(t-1)$, $u_1(t)$, $u_2(t-1)$ and $u_2(t)$) and the last and current of the two system output signals (i.e. $y_1(t-1)$, $y_1(t)$, $y_2(t-1)$ and $y_2(t)$). On the other hand, the two outputs of the RBF neural network nonlinear sub-model were $f_{0,1}(.,.)$ and $f_{0,2}(.,.)$, which represent the approximation of the nonlinear dynamics and disturbances of the complex MIMO plant that is to be accommodated in the multivariable multiple-control control law.

For the autonomous vehicle control application, given in chapter six in section (6.4), the MIMO RBF neural network had thirteen units in the hidden layer four of which had fixed centres and the remaining nine units were selected to have adaptive centres. The neural network based nonlinear sub-model of this MIMO GLM had twelve inputs and three outputs. The inputs were the last and current of the three control input signals (i.e. $u_1(t-1)$, $u_1(t)$, $u_2(t-1)$, $u_2(t)$, $u_3(t-1)$ and $u_3(t)$) and the last and current of the three system output signals (i.e. $y_1(t-1)$, $y_1(t)$, $y_2(t-1)$, $y_2(t)$, $y_3(t-1)$ and $y_3(t)$). On the other hand, the three outputs of the RBF neural network nonlinear sub-model were $f_{0,1}(.,.)$, $f_{0,2}(.,.)$ and $f_{0,3}(.,.)$.

5.5 Novel Intelligent Multivariable Multiple-Controller Framework for MIMO Complex Systems

In this section the convention multivariable multiple-controller proposed in [17, 39, 46], presented in equation (5.3), is extended to a novel intelligent multiple-controller framework for achieving more efficient control of complex MIMO systems. The new developed methodology combines the advantages of the adaptive conventional PID controller with the Pole-Zero Placement controller through the autonomous tuning and switching between the these multivariable controller and by incorporating the MIMO RBF neural network based GLM. The autonomous switching and tuning actions are performed by the fuzzy-logic supervisor employed at the top of the MIMO control system, as can be seen in Figure (5.4) next, according to the data received from the control system, the environment and the information supplied by the user. Therefore, the switching and tuning decisions are made on the basis of the closed-loop system performance measurements and the user desired performance.

The general transfer function for the proposed multivariable multiple-controller is as follows:

$$\mathbf{u}(t) = \frac{\mathbf{v}(\mathbf{v}_\tau)[\mathbf{H}(\mathbf{H}_\eta, \widetilde{\mathbf{H}}_\tau)\mathbf{F}(1)\mathbf{w}(t) - \mathbf{F}\mathbf{y}(t) + \mathbf{\Delta}\mathbf{H}'_N \mathbf{f}_{0,t}(.,.)]}{\mathbf{\Delta}\mathbf{q}(\mathbf{q}_\eta, \mathbf{T}_\tau)}, \tag{5.18}$$

where $\mathbf{w}(t)$ is an $n \times 1$ vector represents the system set-points, $\mathbf{f}_{0,t}(.,.)$ is a nonlinear function representing the nonlinear dynamics and disturbances of the complex MIMO system under control, $\mathbf{\Delta}$ is the integral action required for the PID design, $\mathbf{F}$ is a polynomial derived from the linear parameters of the controlled plant and includes the desired closed loop poles, $\mathbf{F}(1)$ is the value of $\mathbf{F}$ at the steady state, $\mathbf{H}'_N$ is a user-defined polynomial used for activating the nonlinear function $\mathbf{f}_{0,t}(.,.)$.

The transferred poles $\mathbf{q}$ were derived through the following Diophantine equation which is used to place the poles of the system in the required position:

$$(\mathbf{q}'\Delta\mathbf{A} + z^{-1}\mathbf{F}\mathbf{v}\mathbf{B}) = \mathbf{T}, \tag{5.19}$$

where $\mathbf{T}$ represents the desired closed loop poles and $\mathbf{q}'$ is the controller polynomial.

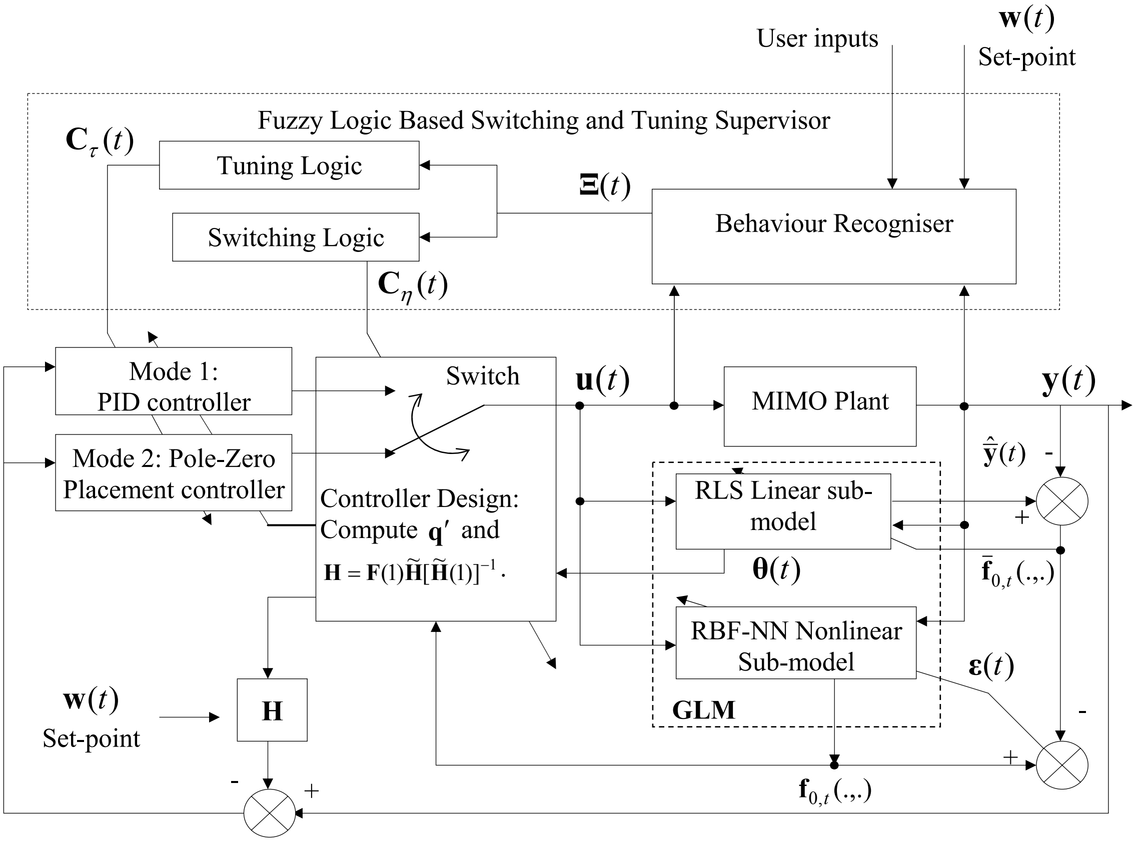


Figure (5.4): Intelligent Multiple-Controller Framework for MIMO complex systems.

In the multiple-controller above, $\mathbf{v}(\mathbf{v}_\tau)$ is the tuned PID gain and $\mathbf{q}(\mathbf{q}_\eta, \mathbf{T}_\tau)$ represent the transferred poles of the control system transfer function as a function of the switching parameter $\mathbf{q}_\eta$ and the tuning parameter $\mathbf{T}_\tau$ and can be expressed as:

$$\mathbf{q}(\mathbf{q}_\eta, \mathbf{T}_\tau) = \mathbf{I} + \mathbf{q}_\eta \mathbf{q}_1 z^{-1} + \mathbf{q}_\eta \mathbf{q}_2 z^{-1} + \ldots + \mathbf{q}_\eta \mathbf{q}_{n_q} z^{-n_q} . \tag{5.20}$$

To simplify the representation of the equation (5.20), since the identity matrix $\mathbf{I}$, $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_{n_q}$ are diagonal matrices, the above equation can be written in the form:

$$\mathbf{q}(\mathbf{q}_\eta, \mathbf{T}_\tau) = diag(1 + \mathbf{q}_\eta q_1^{ii} z^{-1} + \mathbf{q}_\eta q_2^{ii} z^{-2} + \ldots + \mathbf{q}_\eta q_{n_q}^{ii} z^{-n_q}) , \tag{5.21}$$

where n_q is the order of the transferred poles, and the tuned poles are $q_i^{ii} = q_i^{ii} + t_{\tau_i}$ ($i = 1,2,\ldots,n_q$).

The term $\mathbf{H}(\mathbf{H}_\eta, \widetilde{\mathbf{H}}_\tau)$ represents the control law transfer function zeros as a function of the autonomous switching and tuning parameters, respectively, $\mathbf{H}_\eta$ and $\widetilde{\mathbf{H}}_\tau$ as it can be seen in the next equation:

$$\mathbf{H}(\mathbf{H}_\eta, \widetilde{\mathbf{H}}_\tau) = \widetilde{\mathbf{H}}(\mathbf{H}_\eta, \widetilde{\mathbf{H}}_\tau)[\widetilde{\mathbf{H}}(1, \mathbf{H}_\eta, \widetilde{\mathbf{H}}_\tau)]^{-1}, \tag{5.22a}$$

where, $\widetilde{\mathbf{H}}(\mathbf{H}_\eta, \widetilde{\mathbf{H}}_\tau) = \mathbf{I} + \mathbf{H}_\eta \widetilde{\mathbf{H}}_1 z^{-1} + \mathbf{H}_\eta \widetilde{\mathbf{H}}_2 z^{-2} + \ldots + \mathbf{H}_\eta \widetilde{\mathbf{H}}_{n_h} z^{-n_h}$ and can be written as:

$$\widetilde{\mathbf{H}}(\mathbf{H}_\eta, \widetilde{\mathbf{H}}_\tau) = diag(1 + \mathbf{H}_\eta h_1^{ii} z^{-1} + \mathbf{H}_\eta h_2^{ii} z^{-2} + \ldots + \mathbf{H}_\eta h_n^{ii} z^{-n_h}) , \tag{5.22b}$$

and

$$\widetilde{\mathbf{H}}(1,\mathbf{H}_\eta,\widetilde{\mathbf{H}}_\tau) = diag(1+\mathbf{H}_\eta h_1^{ii} + \mathbf{H}_\eta h_2^{ii} + \ldots + \mathbf{H}_\eta h_{n_h}^{ii}), \tag{5.22c}$$

which denotes the zeros $\widetilde{\mathbf{H}}(\mathbf{H}_\eta,\widetilde{\mathbf{H}}_\tau)$ during the steady state. n_h represents the order of polynomial $\widetilde{\mathbf{H}}$, and the tuned zeros are $h_i^{ii} = h_i^{ii} + \widetilde{h}_{\tau_i}$ and $(i = 1,2,\ldots,n_h)$.

The next subsections will present the two control modes of the intelligent multivariable multiple-controller.

5.5.1 Multiple-Controller Mode 1: Conventional Adaptive PID Controller

In this mode, the multiple-controller operates as a conventional adaptive PID controller, which can be expressed in the most commonly used velocity form [41] as:

$$\Delta\mathbf{u}(t) = \mathbf{K_I}\mathbf{w}(t) - [\mathbf{K_P} + \mathbf{K_I} + \mathbf{K_D}]\mathbf{y}(t) - [-\mathbf{K_P} - 2\mathbf{K_D}]\mathbf{y}(t-1) - \mathbf{K_D}\mathbf{y}(t-2). \tag{5.23}$$

If we assume that the degree of the polynomial $\mathbf{F}(z^{-1})$ is equal to 2

$$\mathbf{F}(z^{-1}) = diag(f_0^{ii} + f_1^{ii}z^{-1} + f_2^{ii}z^{-2}), \tag{5.24}$$

and both the pole-placement polynomial $\mathbf{q}(\mathbf{q}_\eta,\mathbf{T}_\tau)$ and zero-placement polynomial $\widetilde{\mathbf{H}}(\mathbf{H}_\eta,\widetilde{\mathbf{H}}_\tau)$ are switched off when the fuzzy logic supervisor sets the switching parameter $\mathbf{C}_\eta$ as:

$$\mathbf{C}_\eta = [\mathbf{q}_\eta \quad \mathbf{H}_\eta] = \begin{bmatrix} q_{\eta_1} = 0 & H_{\eta_1} = 0 \\ q_{\eta_2} = 0 & H_{\eta_2} = 0 \\ \vdots & \vdots \\ q_{\eta_{n_q}} = 0 & H_{\eta_{n_h}} = 0 \end{bmatrix}. \tag{5.25}$$

Consequently,

$$\mathbf{q}(\mathbf{q}_{\eta},\mathbf{T}_{\tau}) = diag(1+(0)q_1^{ii}z^{-1}+(0)q_2^{ii}z^{-2}+...+(0)q_{n_q}^{ii}z^{-n_q}), \qquad (5.26a)$$

$$\widetilde{\mathbf{H}}(\mathbf{H}_{\eta},\widetilde{\mathbf{H}}_{\tau}) = diag(1+(0)h_1^{ii}z^{-1}+(0)h_2^{ii}z^{-2}+...+(0)h_{n_h}^{ii}z^{-n_h}). \qquad (5.26b)$$

Therefore,

$$\left.\begin{array}{ll} \mathbf{q}(z^{-1}) = diag(1), & (\text{i.e. } q_1'^{ii} = q_2'^{ii} = ..,.. = q_{n_{q'}}^{ii} = 0) \\ \widetilde{\mathbf{H}}(z^{-1}) = diag(1), & (\text{i.e. } \widetilde{h}_1^{ii} = \widetilde{h}_2^{ii} = ...,... = \widetilde{h}_{n_{\widetilde{h}}}^{ii} = 0) \end{array}\right\}, \qquad (5.27)$$

then a multivariable adaptive controller with PID structure is obtained, where

$$\mathbf{u}(t) = \frac{\mathbf{VF}(1)\mathbf{w}(t) - \mathbf{V}(\mathbf{f}_0+\mathbf{f}_1z^{-1}+\mathbf{f}_2z^{-1})\mathbf{y}(t) + \Delta\mathbf{VH}'_N\mathbf{f}_{0,t}(.,.)}{\Delta}, \qquad (5.28)$$

$$\mathbf{K_P} = -diag(v^{ii}f_1 + 2v^{ii}f_2), \qquad (5.29a)$$

$$\mathbf{K_I} = diag(v^{ii}f_0^{ii} + v^{ii}f_1^{ii} + v^{ii}f_2^{ii}), \qquad (5.29b)$$

$$\mathbf{K_D} = diag(v^{ii}f_2^{ii}), \qquad (5.29c)$$

where $i = 1,...n$. It can be seen from the above equations (5.28), (5.29a), (5.29b) and (5.29c) that the PID control parameters $\mathbf{K_P}$, $\mathbf{K_I}$ and $\mathbf{K_D}$ depend on the polynomial matrix $\mathbf{F}(z^{-1})$ and the gain matrix $\mathbf{V}$ [41, 43].

5.5.2 Multiple-Controller Mode 2: Pole-Zero Placement Controller

This control mode works to achieve more effective control actions and combines the advantages of the adaptive PID control with the advantages of the placement poles and zeros [17, 39].

In this multivariable controller mode, the fuzzy logic based switching parameter $\mathbf{C}_\eta$ is set to:

$$\mathbf{C}_\eta = [\mathbf{q}_\eta \quad \mathbf{H}_\eta] = \begin{bmatrix} q_{\eta_1} = 1 & H_{\eta_1} = 1 \\ q_{\eta_2} = 1 & H_{\eta_2} = 1 \\ \vdots & \vdots \\ q_{\eta_{n_q}} = 1 & H_{\eta_{n_h}} = 1 \end{bmatrix}. \qquad (5.30)$$

Consequently,

$$\mathbf{q}(\mathbf{q}_\eta, \mathbf{T}_\tau) = diag(1 + \mathbf{q}_\eta q_1^{ii} z^{-1} + \mathbf{q}_\eta q_2^{ii} z^{-2} + \ldots + \mathbf{q}_\eta q_{n_q}^{ii} z^{-n_q}). \qquad (5.31)$$

For a control system with two control inputs and poles of order two, then $\mathbf{q}(\mathbf{q}_\eta, \mathbf{T}_\tau)$ can be expressed into:

$$\mathbf{q}(\mathbf{q}_\eta, \mathbf{T}_\tau) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + [q_{\eta_1} \quad q_{\eta_2}] \begin{bmatrix} q_1^{00} & 0 \\ 0 & q_1^{11} \end{bmatrix} z^{-1} + [q_{\eta_1} \quad q_{\eta_2}] \begin{bmatrix} q_2^{00} & 0 \\ 0 & q_2^{11} \end{bmatrix} z^{-2}. \qquad (5.32)$$

Appling the fuzzy logic based switching decision in equation (5.30) above, then

$$\mathbf{q}(\mathbf{q}_\eta, \mathbf{T}_\tau) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + [1 \quad 1] \begin{bmatrix} q_1^{00} & 0 \\ 0 & q_1^{11} \end{bmatrix} z^{-1} + [1 \quad 1] \begin{bmatrix} q_2^{00} & 0 \\ 0 & q_2^{11} \end{bmatrix} z^{-2}, \qquad (5.33)$$

which will activate all the MIMO system's transferred poles.

Similarly, with $n_h = 2$ the control system zeros will be included in multivariable control law as follows:

$$\widetilde{\mathbf{H}}(\mathbf{H}_\eta, \widetilde{\mathbf{H}}_\tau) = diag(1 + \mathbf{H}_\eta h_1^{ii} z^{-1} + \mathbf{H}_\eta h_2^{ii} z^{-2} + ... + \mathbf{H}_\eta h_{n_h}^{ii} z^{-n_h}), \tag{5.34}$$

which can be detailed to

$$\widetilde{\mathbf{H}}(\mathbf{H}_\eta, \widetilde{\mathbf{H}}_\tau) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + [H_{\eta_1} \quad H_{\eta_2}] \begin{bmatrix} h_1^{00} & 0 \\ 0 & h_1^{11} \end{bmatrix} z^{-1} + [H_{\eta_1} \quad H_{\eta_2}] \begin{bmatrix} h_2^{00} & 0 \\ 0 & h_2^{11} \end{bmatrix} z^{-2}. \tag{5.35}$$

Appling the fuzzy logic based switching decision in equation (5.50) above, then

$$\widetilde{\mathbf{H}}(\mathbf{H}_\eta, \widetilde{\mathbf{H}}_\tau) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + [1 \quad 1] \begin{bmatrix} h_1^{00} & 0 \\ 0 & h_1^{11} \end{bmatrix} z^{-1} + [1 \quad 1] \begin{bmatrix} h_2^{00} & 0 \\ 0 & h_2^{11} \end{bmatrix} z^{-2}. \tag{5.36}$$

Therefore, an adaptive Pole-Zero Placement controller with PID structure is obtained, where

$$\mathbf{u}(t) = \frac{\mathbf{v}(\mathbf{v}_\tau)[\mathbf{H}(\mathbf{H}_\eta, \widetilde{\mathbf{H}}_\tau)\mathbf{F}(1)\mathbf{w}(t) - \mathbf{F}\mathbf{y}(t) + \mathbf{\Delta H}'_N \mathbf{f}_{0,t}(.,.)]}{\mathbf{\Delta q}(\mathbf{q}_\eta, \mathbf{T}_\tau)}. \tag{5.37}$$

The PID parameters $\mathbf{K_P}$, $\mathbf{K_I}$ and $\mathbf{K_D}$ derived as in mode 1 above.

5.5.3 Multivariable Multiple-Controller Algorithm Summary: MIMO Case

The proposed intelligent multiple-controller algorithm can now be summarised into the following steps:

Step 1: Select the initial desired closed-loop system poles and zeros polynomials $\mathbf{T}$ and $\widetilde{\mathbf{H}}$ respectively.

Step 2: In order to avoid high control action, at the start of the control process, and consequently prevent output signal overshooting, the MIMO system is initially set to work with the multivariable Pole-Zero Placement controller by setting $\mathbf{C}_\eta$ as follows:

$$\mathbf{C}_\eta = [\mathbf{q}_\eta \quad \mathbf{H}_\eta] = \begin{bmatrix} q_{\eta_1} = 1 & H_{\eta_1} = 1 \\ q_{\eta_2} = 1 & H_{\eta_2} = 1 \\ \vdots & \vdots \\ q_{\eta_{n_q}} = 1 & H_{\eta_{n_h}} = 1 \end{bmatrix}.$$

Step 3: Select $\mathbf{F}$ and the initial value for the gain $\mathbf{v}$ for the desired PID control structure.

Step 4: read the current values of $\mathbf{y}(t)$ and $\mathbf{w}(t)$.

Step 5: Compute the control input $\mathbf{u}(t)$ using equation (5.37) when the current controller is Pole-Zero Placement, or using equation (5.28) when the controller is the conventional PID controller.

Step 6: Estimate the process linear parameters $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ using the least squares algorithm of the linear sub-model in the MIMO GLM.

Step 7: Compute $\bar{\mathbf{f}}_0(.,.) = \mathbf{y}(t) - \hat{\bar{\mathbf{y}}}(t)$, where $\hat{\bar{\mathbf{y}}}(t)$ is the output of the linear sub-model.

Step 8: Apply the MIMO RBF based nonlinear sub-model of the GLM to obtain $\hat{\mathbf{f}}_0(.,.)$ by using equations (5.14)-(5.17).

Step 9: The behaviour recogniser will assess the current performance of the control system using the system output vector $\mathbf{y}(t)$, the set-point vector $\mathbf{w}(t)$, the control input vector $\mathbf{u}(t)$ and the user requirements.

Step 10: The behaviour recogniser will report the system performance to the fuzzy logic based switching and tuning subsystems as concluded in $\boldsymbol{\Xi}(t)$.

Step 11: The fuzzy logic switching subsystem will employ $\boldsymbol{\Xi}_s(t)$ to make the switching decision for the next multivariable controller to be activated, that is achieved by reconfiguring the switching matrix $\mathbf{C}_\eta$. For a multivariable Pole-Zero Placement controller $\mathbf{C}_\eta = \begin{bmatrix} 1 & 1 & ... & 1 \\ 1 & 1 & ... & 1 \end{bmatrix}_{2\times n}^T$ (n is number of control inputs), for a multivariable PID controller $\mathbf{C}_\eta = \begin{bmatrix} 0 & 0 & ... & 0 \\ 0 & 0 & ... & 0 \end{bmatrix}_{2\times n}^T$. According to the performance of the MIMO control system, the fuzzy logic switching decision could set different controllers for the various control inputs at the same instant of time t. For instance, the fuzzy switching command $\mathbf{C}_\eta = \begin{bmatrix} 0 & 1 & ... & 0 \\ 0 & 1 & ... & 0 \end{bmatrix}_{2\times n}^T$ will set the second control input to the Pole-Zero Placement controller, but the other control inputs to be PID controllers.

Step 12: Based on Ξ_t, the fuzzy logic tuning subsystem will decide the tuning values for the parameters of the current multivariable controller that is by setting the fuzzy logic tuning output $\mathbf{C}\tau$.

Steps 4 to 12 are to be repeated for every sampling instant.

5.6 Closed Loop Stability Analysis of the Intelligent Multiple-Controller Framework

Stability of the proposed control algorithm is analysed based on the following assumptions:

Assumption A: Given a positive constant $\boldsymbol{\varepsilon}$ and a compact set $\mathbf{S} \subset R^{n_a \times n_b}$, there exist coefficients $\mathbf{W}$ such that $\mathbf{f}_0(.,\mathbf{W})$ approximate the continuous functions $\mathbf{f}_0(.,.)$ with accuracy $\boldsymbol{\varepsilon}$, that is:

$$\exists \mathbf{W}.s.t \ \max\left|\hat{\mathbf{f}}_0(\mathbf{x},\mathbf{W}) - \mathbf{f}(\mathbf{x})\right| \leq \boldsymbol{\varepsilon} < \infty; \ \ \mathbf{x} \in \mathbf{S}. \tag{5.38}$$

Assumption B: $\mathbf{f}_0(.,.)$ is a bounded quantity [52, 16].

Assumption C: the reference signal is bounded.

In order to derive the stability of the overall closed loop system, the stability analysis of each of the two controller switching modes (Pole-Zero Placement controller and PID controller) is discussed separately in sections (5.6.1) and (5.6.2) respectively, whereas the stability of the fuzzy switching and tuning system is discussed in section (5.6.3).

5.6.1 Stability Analysis of the Multivariable Multiple-controller Mode 1 (Pole-Zero Placement controller)

The closed-loop system transfer function at time t and $(t+k)$ can be described by the following Lemma, which is derived from the transfer function of the closed-loop system in Figure (5.5), next, when the Pole-Zero placement controller is selected.

Let $\mathbf{H}'_0 = \widetilde{\mathbf{H}}[\widetilde{\mathbf{H}}(1)]^{-1}\mathbf{F}(1)$.

Lemma 1:

$$\begin{bmatrix} B_{11,t}P_{n_{11,t}} + Q_{11,t}P_{d_{11,t}}A_{11,t} & \cdots & B_{1i,t}P_{n_{1i,t}} + Q_{1i,t}P_{d_{1i,t}}A_{1i,t} \\ \vdots & \ddots & \vdots \\ B_{i1,t}P_{n_{i1,t}} + Q_{i1,t}P_{d_{i1,t}}A_{i1,t} & \cdots & B_{ii,t}P_{n_{ii,t}} + Q_{ii,t}P_{d_{ii,t}}A_{ii,t} \end{bmatrix} \begin{bmatrix} y_1(t+1) & u_1(t) \\ \vdots & \vdots \\ y_i(t+1) & u_i(t) \end{bmatrix} =$$

$$\begin{bmatrix} B_{1,k}.R_{1,t} & \cdots & B_{i,k}.R_{i,t} \\ A_{1,k}.R_{1,t} & \cdots & A_{i,k}.R_{i,t} \end{bmatrix} \begin{bmatrix} w_1(t) \\ \vdots \\ w_i(t) \end{bmatrix} + \begin{bmatrix} A_{11} & \cdots & A_{1i} \\ A_{21} & \cdots & A_{2i} \end{bmatrix} \begin{bmatrix} y_1(t+1) \\ \vdots \\ y_i(t+1) \end{bmatrix} +$$

$$\begin{bmatrix} B_{11} & \cdots & B_{1i} \\ B_{21} & \cdots & B_{2i} \end{bmatrix} \begin{bmatrix} u_1(t) \\ \vdots \\ u_i(t) \end{bmatrix} + \begin{bmatrix} B_{1,k}P_{d_{1,t}}E_{1,t} + Q_{1,t}P_{d_{1,t}} & \cdots & B_{i,k}P_{d_{i,t}}E_{i,t} + Q_{i,t}P_{d_{i,t}} \\ B_{1,k}P_{d_{1,t}}E_{1,t} - P_{n_{1,t}} & \cdots & B_{i,k}P_{d_{i,t}}E_{i,t} - P_{n_{i,t}} \end{bmatrix} \begin{bmatrix} \xi_1(t+1) \\ \vdots \\ \xi_i(t+1) \end{bmatrix} +$$

$$\begin{bmatrix} B_{1,k}.P_{d_{1,t}}H_{N_{1,t}} + Q_{1,t}P_{d_{1,t}} & \cdots & B_{i,k}.P_{d_{i,t}}H_{N_{i,t}} + Q_{i,t}P_{d_{i,t}} \\ A_{1,k}.P_{d_{1,t}}.H_{N_{1,t}} - P_{n_{1,t}} & \cdots & A_{i,k}.P_{d_{i,t}}.H_{N_{i,t}} - P_{n_{i,t}} \end{bmatrix} \begin{bmatrix} \hat{f}_{0,1}(.,.) \\ \vdots \\ \hat{f}_{0,i}(.,.) \end{bmatrix} +$$

$$\begin{bmatrix} B_{1,k}.E_{1,t}P_{d_{1,t}} & \cdots & B_{i,k}.E_{i,t}P_{d_{i,t}} \\ A_{1,k}.E_{1,t}P_{d_{1,t}} & \cdots & A_{i,k}.E_{i,t}P_{d_{i,t}} \end{bmatrix} \begin{bmatrix} f_0{,}_1(.,.) - \hat{f}_0{,}_1(.,.) \\ \vdots \\ f_0{,}_i(.,.) - \hat{f}_0{,}_i(.,.) \end{bmatrix}. \tag{5.39}$$

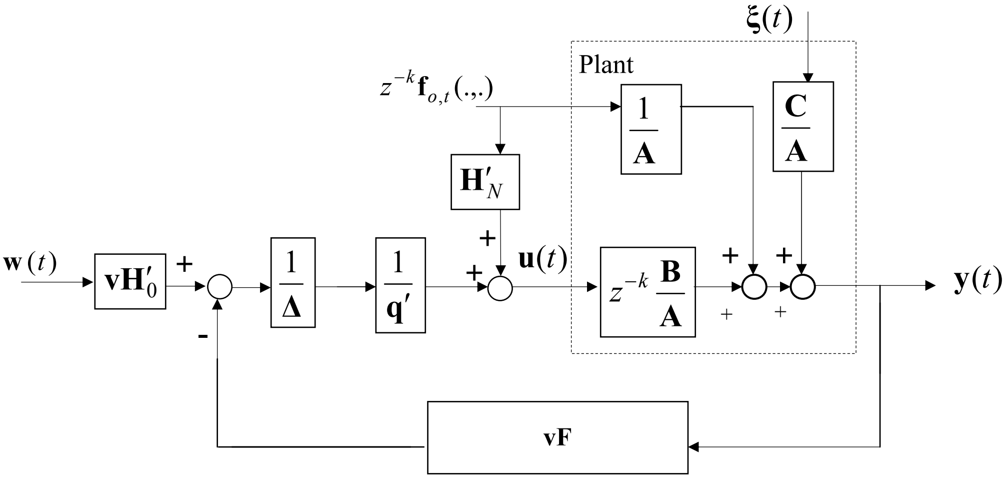


Figure (5.5): Non-linear generalised minimum variance Pole-Zero placement controller.

In Equation (5.39), $i = 1,2,...m$: m is dimension of the MIMO system linear polynomials $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$ with matrices size ($m \times m$),

$$A_{1i} = B_{i,k}.(P_{d_{i,t}} E_{i,t} A_{i,t} - P_{d_{1,t}} E_{i,t}.A_{i,k}) + (B_{i,t} P_{n_{i,t}} - B_{i,k}.P_{n_{i,t}}) , \qquad (5.40a)$$

$$A_{2i} = A_{i,k}.(P_{d_{i,t}} E_{i,t} A_{i,t} - P_{d_{i,t}} E_{i,t}.A_{i,k}) + (A_{i,t} P_{n_{i,t}} - A_{i,k}.P_{n_{i,t}}) , \qquad (5.40c)$$

$$B_{1i} = B_{i,k}.(P_{d_{i,t}} E_{i,t} B_{i,k} - P_{d_{i,t}} E_{i,t}.B_{i,t}) + (P_{d_{i,t}} Q_{i.t} B_{i,t} - B_{i,k}.P_{d_{i,t}} Q_{i,t}) , \qquad (5.40b)$$

$$B_{2i} = A_{i,k}.(P_{d_{i,t}} E_{i,t} B_{i,k} - P_{d_{i,t}} E_{i,t}.B_{i,t}) + (P_{d_{i,t}} A_{i,t} Q_{i,t} - A_{i,k}.P_{d_{i,t}} Q_{i,t}) , \qquad (5.40d)$$

$A_{i,t}$, $A_{i,k}$, $B_{i,t}$ and $B_{i,k}$ denote the estimate values of $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$ at t and $t+k$ moments respectively, i.e.

$$A_{i,t} = \mathbf{A}(t, z^{-1}),\ A_{i,k} = \mathbf{A}(t+k, z^{-1}) , \qquad (5.41)$$

$$A_{i,t}.B_{i,k} = \mathbf{A}(t,z^{-1}).\mathbf{B}(t+k,z^{-1}) \neq B_{i,k}A_{i,t}\,, \tag{5.42}$$

$$A_{i,t}.B_{i,t} = \mathbf{AB}(t,z^{-1}) = B_{i,t}A_{i,t}\,. \tag{5.43}$$

Proof 1: To prove this lemma, reconsider the plant in Figure (5.5):

$$A_{i,k}\mathbf{y}(t+1) = B_{i,k}\mathbf{u}(t) + \mathbf{f}_0\,(.,.) + \mathbf{C}\boldsymbol{\xi}(t+1)\,. \tag{5.44}$$

To simplify the derivation let $\mathbf{C} = \mathbf{I}$.

Multiplying equation (5.44) by $P_{d_{i,t}}E_{i,t}$ we obtain:

$$P_{d_{i,t}}E_t\,.A_{i,k}\mathbf{y}(t+1) = P_{d_{i,t}}E_{i,t}.B_{i,k}\mathbf{u}(t) + P_{d_{i,t}}E_{i,t}.\mathbf{f}_0(.,.) + P_{d_{i,t}}E_{i,t}.\boldsymbol{\xi}(t+1)\,. \tag{5.45}$$

Using the equations (5.42) and (5.43), and the applying the following identity $\mathbf{P}_n(z^{-1}) = \mathbf{A}(z^{-1})\mathbf{E}(z^{-1})\mathbf{P}_d(z^{-1}) + z^{-1}\mathbf{F}(z^{-1})$ yields:

$$\begin{aligned}
&P_{n_{i,t}}\mathbf{y}(t+k) - F_{i,t}\mathbf{y}(t) - P_{d_{i,t}}E_{i,t}B_{i,t}\mathbf{u}(t) - P_{d_{i,t}}E_{i,t}\hat{\mathbf{f}}_{0,t}(.,.) = \\
&P_{d_{i,t}}(E_{i,t}.B_{i,k} - E_{i,t}B_{i,t})\mathbf{u}(t) + P_{d_{i,t}}(E_{i,t}A_{i,t} - E_{i,t}.A_{i,k})\mathbf{y}(t+k) + . \\
&P_{d_{i,t}}E_{i,t}[\mathbf{f}_{0,t}(.,.) - \hat{\mathbf{f}}_{0,t}(.,.)] + P_{d_{i,t}}E_{i,t}.\boldsymbol{\xi}(t+1)
\end{aligned} \tag{5.46}$$

Then using $\mathbf{P}_d(\mathbf{EB}+\mathbf{Q})\mathbf{u}(t) = [\mathbf{P}_d\mathbf{Rw}(t) - \mathbf{Fy}(t) + \mathbf{P}_d(\mathbf{H}_N - \mathbf{E})]\mathbf{f}_{0,t}(.,.)$ [16, 37] and (5.46) gives:

$$\begin{aligned}
&P_{n_{i,t}}\mathbf{y}(t+k) - P_{d_{i,t}}R_{i,t}\mathbf{w}(t) + P_{d_{i,t}}Q_{i,t}\mathbf{u}(t) - P_{d_{i,t}}H_{N i,t}\hat{\mathbf{f}}_{0,t}(.,.) = \\
&P_{d_{i,t}}(E_{i,t}.B_{i,k} - E_{i,t}B_{i,t})\mathbf{u}(t) + P_{d_{i,t}}(E_{i,t}A_{i,t} - E_{i,t}.A_{i,k})\mathbf{y}(t+k) + . \\
&P_{d_{i,t}}E_{i,t}[\mathbf{f}_{0,t}(.,.) - \hat{\mathbf{f}}_{0,t}(.,.)] + P_{d_{i,t}}E_{i,t}.\boldsymbol{\xi}(t+1)
\end{aligned} \tag{5.47}$$

The dynamic relationship between the plant input and output given in (5.39) can subsequently be obtained by multiplying equation (5.47) by $B_{i,k}$ and $A_{i,k}$ respectively, and combining the results with equations (5.41), (5.42) and (5.43).

The stability and convergence of the algorithm are then as stated below:

Theorem 1: If *assumptions* A, B and C are satisfied and hold, the recursive parameter estimation algorithm has the following properties [16, 38]:

$$\lim_{t\to\infty}[y_{i,t}(t)] < \infty\,;\ \lim_{t\to\infty}[u_{i,t}(t)] < \infty\,, \tag{5.48}$$

$$\lim_{t\to\infty}\left|\phi_{i,t}(t+1)\right|^2 < \sigma_{i,t}^2 < \infty\,. \tag{5.49}$$

The boundedness of $\mathbf{y}(t)$ and $\mathbf{u}(t)$ in equation (5.48) can be proven by considering in lemma 1 that the terms in the parentheses of (5.39) tend to zero at $t\to\infty$ subject to assumptions A and B, and the boundedness of $\mathbf{w}(t)$. Therefore, the algorithm stability is proven. From equation (5.47) we have that:

$$\begin{aligned}\lim_{t\to\infty}\left|\phi_{i,t}(t+1)\right|^2 = \lim_{t\to\infty}\Big| & P_{d_{i,t}}(E_{i,t}.B_{i,k} - E_{i,t}B_{i,t})\mathbf{u}(t) + \\ & P_{d_{i,t}}(E_{i,t}A_{i,t} - E_{i,t}.A_{i,k})\mathbf{y}(t+k) + \\ & P_{d_{i,t}}E_{i,t}.\boldsymbol{\xi}(t+1) + P_{d_{i,t}}E_{i,t}[\mathbf{f}_0(.,.) - \hat{\mathbf{f}}_0(.,.)]\Big|^2\end{aligned}\quad , \tag{5.50}$$

$$\lim_{t\to\infty}\left|\phi_{i,t}(t+1)\right|^2 \le (P_{d_{i,t}}E_{i,t}\boldsymbol{\varepsilon})^2 = {\sigma_{i,t}^{2}}^{2} < \infty\,. \tag{5.51}$$

Hence, the convergence of equation (5.49) is proven.

5.6.2 Stability Analysis of the Multivariable Multiple-controller Mode 2 (PID controller)

To prove the stability of the adaptive Nonlinear PID controller, consider the transfer function of the closed-loop system in Figure (5.6).

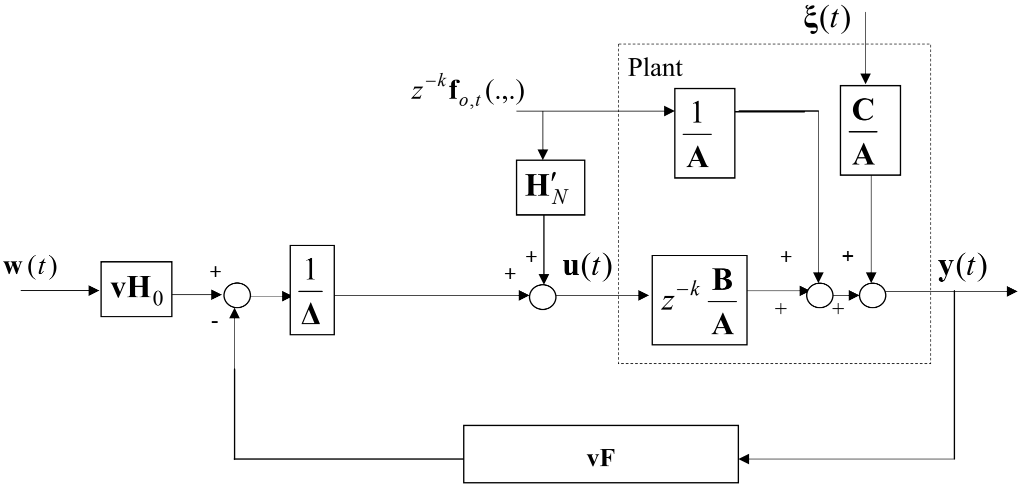


Figure (5.6): Non-linear generalised minimum variance PID controller.

$$\begin{aligned}\mathbf{y}(t) = {} & [1+\frac{1}{\mathbf{\Delta}}z^{-k}\frac{\mathbf{B}}{\mathbf{A}}\mathbf{vF}]^{-1}[(\frac{1}{\mathbf{\Delta}}z^{-k}\frac{\mathbf{B}}{\mathbf{A}})\mathbf{vH}_0]\mathbf{w}(t)+[1+\frac{1}{\mathbf{\Delta}}z^{-k}\frac{\mathbf{B}}{\mathbf{A}}\mathbf{vF}]^{-1}[\frac{\mathbf{C}}{\mathbf{A}}]\xi(t)+ \\ & [1+\frac{1}{\mathbf{\Delta}}z^{-k}\frac{\mathbf{B}}{\mathbf{A}}\mathbf{vF}]^{-1}[\frac{1}{\mathbf{A}}z^{-k}]\mathbf{f}_{0,t}(.,.)+[1+\frac{1}{\mathbf{\Delta}}z^{-k}\frac{\mathbf{B}}{\mathbf{A}}\mathbf{vF}]^{-1}[\frac{\mathbf{H}_N\mathbf{B}}{\mathbf{A}}z^{-k}]\mathbf{f}_{0,t}(.,.)\end{aligned} \quad . \quad (5.52)$$

After manipulating the transfer function in equation (5.52) the resultant equation is, thus,

$$[z^k\mathbf{\Delta A}+\mathbf{BvF}]\mathbf{y}(t)=\mathbf{BvH}_0\mathbf{w}(t)+z^k\mathbf{C\Delta}\xi(t)+[1+\mathbf{H}_N\mathbf{B}]\mathbf{\Delta f}_{0,t}(.,.)\,. \qquad (5.53)$$

If we let $\mathbf{x}_1(t)=\mathbf{C}\xi(t)$, $\mathbf{x}_2(t)=[1+\mathbf{H}_N\mathbf{B}]\mathbf{f}_{0,t}(.,.)$ and let $z^{-1}=d$, then the equation (5.53) above becomes

$$[d^{-k}\mathbf{\Delta A}+\mathbf{BvF}]\mathbf{y}(t)=\mathbf{BvH}_0\mathbf{w}(t)+d^{-k}\mathbf{\Delta x}_1(t)+\mathbf{\Delta x}_2(t)\,. \tag{5.54}$$

Let the transferred poles to be denoted as $\mathbf{G}'$ such that

$$\mathbf{G}'=d^{-k}\mathbf{\Delta A}+\mathbf{BvF}\,. \tag{5.55}$$

The transfer function from the reference input $\mathbf{w}(t)$ to the output $\mathbf{y}(t)$ becomes

$$\mathbf{G}_{wy}=[\mathbf{G}']^{-1}\mathbf{BvH}_0\,, \tag{5.56}$$

the transfer function from the disturbance $\mathbf{x}_1(t)$ to the output $\mathbf{y}(t)$ becomes

$$\mathbf{G}_{x_1y}=[\mathbf{G}']^{-1}d^{-k}\mathbf{\Delta}\,, \tag{5.57a}$$

and the transfer function from the disturbance $\mathbf{x}_2(t)$ to the output $\mathbf{y}(t)$ becomes

$$\mathbf{G}_{x_2y}=[\mathbf{G}']^{-1}\mathbf{\Delta}\,. \tag{5.57b}$$

The poles of the closed loop system are determined by $\mathbf{G}'$ and the zeros are those of the open loop zeros plus additional zeros provided by the term $\mathbf{H}_0$, assuming that no pole zero cancellation occurs providing that the Pole-Zero Placement controller is set offline. The condition for the closed loop stability is then dependent on $\mathbf{G}'$ such that, for stability, $\det(\mathbf{G}')=0$ has all its roots strictly outside the unit circle. The requirement is equivalent to $\mathbf{G}'$ having non-zero eigen values. Therefore, to prove the stability of the closed loop system, it is necessary to prove that $\mathbf{G}'$ is a complex matrix which has an inverse for $|d|<1$ [53, 54, 43]. From the following identity [17, 37]

$$\mathbf{P}_n = \mathbf{AEP}_d + z^{-k}\mathbf{F}, \tag{5.58}$$

we can derive $\mathbf{F}$ as

$$\mathbf{F} = d^{-k}(\mathbf{P}_n - \mathbf{AEP}_d). \tag{5.59}$$

Substituting equation (5.59) into the expression for $\mathbf{G}'$, we obtain

$$\mathbf{G}' = d^{-k}\mathbf{A}[\boldsymbol{\Delta} + \mathbf{A}^{-1}\mathbf{Bv}(\mathbf{P}_n - \mathbf{AP}_d\mathbf{E})]. \tag{5.60}$$

Knowing that $\boldsymbol{\Delta} = (1-d)\mathbf{I}$ and $\mathbf{D} = (d)\mathbf{I}$, then $\mathbf{G}'$ can be written as

$$\mathbf{G}' = d^{-k}\mathbf{A}[\mathbf{I} + (1-d)d\mathbf{P}_d - \mathbf{D}' + \mathbf{A}^{-1}\mathbf{Bv}(\mathbf{P}_n - \mathbf{AP}_d\mathbf{E})]. \tag{5.61}$$

Let

$$\mathbf{G}_1' = (1-d)d\mathbf{P}_d - \mathbf{D}' + \mathbf{A}^{-1}\mathbf{Bv}(\mathbf{P}_n - \mathbf{AP}_d\mathbf{E}), \tag{5.62}$$

then equation (5.55) becomes

$$\mathbf{G}' = d^{-k}\mathbf{A}(\mathbf{I} + \mathbf{G}_1'). \tag{5.63}$$

For the stability, using the result of [55] and [56], $\|\mathbf{G}_1'\|$ must be less than 1 for all $|d| < 1$. A further requirement is that $\mathbf{A}$ is stable. Recall that $\mathbf{P}_d$ is diagonal matrix and letting p_1 be one of the elements of the matrix, we can choose

$$-0.5 \le (d(1-d)p_1 - 1) \le 0.5 \tag{5.64}$$

Therefore, $\left\|(1-d)dP_1 - d\right\|$ is less than 0.5 if we select

$$\frac{(-0.5/d)+1}{(1-d)} \le p_1 \le \frac{(0.5/d)+1}{(1-d)}. \tag{5.65}$$

Finally it is necessary to prove that the remaining term in equation (5.62) has a modulus less that 0.5 with the assumption that the term is bounded. Therefore, we can consider that for stability, $\mathbf{v}$, $\mathbf{P}_n$ and $\mathbf{P}_d$ can be chosen small enough such that the modulus of the remaining term is less than 0.5. Hence, referring to the triangular inequality

$$\left\|\mathbf{G}_1'\right\| \le \left\|(1-d)d\mathbf{P}_d - \mathbf{D}'\right\| + \left\|\mathbf{A}^{-1}\mathbf{B}\mathbf{v}(\mathbf{P}_n - \mathbf{A}\mathbf{P}_d\mathbf{E})\right\|.$$

This makes $\left\|\mathbf{G}_1'\right\|$ less than 1 and the stability of the closed loop system is proven.

5.6.3 Stability of the Fuzzy switching and tuning system

Based on the work of [57, 58], this section will outline the stability of the Fuzzy switching and tuning supervisory system. Let the system state vector at time instant k be $\bar{x}(k) = [x_1(k)...x_n(k)]^T$ where $x_1(k)...x_n(k)$ are the state variables of the system at time instant k, and the controllers state vector at time k be $\bar{u}(k) = [u_1(k)...u_m(k)]$ where $u_1(k)...u_m(k)$ are controller state variables and m is the number of controllers. Then the switching and tuning fuzzy system is defined by the implications below

$$R^i : \text{IF}(\, x_1(k) \text{ is } S_1^i, \text{AND}\ldots\text{AND } x_n(k) \text{ is } S_n^i\,) \text{ THEN}\, \bar{u}(k+1) \text{ is } u_g(k), \tag{5.66}$$

for $i=1...N$, $g=1...M$ and $u_g(k)$ is the new controller state updated by switching and/or tuning.

Here, S_1^i is the fuzzy set corresponding to the state variable x_i and implication R^1. The truth value of the implication R^1 at time instant k denoted by $w_i(k)$ is defined as

$$w_i(k) = \wedge(\mu_{s_1^i}(x_1(k)),...,\mu_{s_n^i}(x_n(k))), \tag{5.67}$$

where $\mu_s(x)$ is the membership function value of the fuzzy set S at the position x and $\wedge$ is an operator satisfying

$\min(l_1,...,l_n) \geq \wedge(l_1,...,l_n) \geq 0$.

Usually $\wedge$ is taken to be the minimum operator which gives the minimum of its operands. Then, at instant k the controllers state vector is updated according to $w_i(k)$ in order to enable the required control state.

A fuzzy system is completely represented by the set of characteristic matrices $\overline{A} = [A_1,..., A_n]$ and the fuzzy sets $S_j^l, l=1..N; j=1...n$. Corresponding to this fuzzy system, the corresponding switching and tuning system is described below.

The state update at time instant k is given as

$$\overline{x}(k+1) = A\overline{x}(k), \tag{5.68}$$

where $A \in \overline{A}$ (i.e., it is one of the matrices $A_1,...,A_n$).

The following is a definition of global asymptotic stability of the switching and tuning system.

Theorem 2: The switching system described in (5.68) is globally asymptotically stable if

$$\bar{x}(k+1) = A(k)\bar{x}(0) \to 0 \text{ as } k \to \infty;\ \forall \bar{x}(0) \in \Re^n, \tag{5.69}$$

$$\text{where } A(k) \in \overline{A}_k \text{. Equivalently } A(k) \to 0 \text{ as } k \to \infty;\ A(k) \in \overline{A}_k. \tag{5.70}$$

The proof of the above *theorem* 2 is presented in [58].

The multiple-controller framework is not founded to exhibit any transfer problems during the switching mode in any of the simulation results [17, 37]. In continuous systems, the problem of the transition between the controller modes can be solved by using a hold circuit. In our case, the system is discrete and the hold circuit is not needed and since the controllers exhibit bumpless switching the stability of the controller is achieved.

5.7 Summary

In the recent years, several control strategies for different process phases have been developed. Most of these control schemes have been manually activated by operators. Often the operators have been overwhelmed by the task of choosing the adequate time instant for activating the several control algorithms and tuning them [129]. Therefore, it was of utmost importance to develop a supervisory control scheme, which is capable of activating and tuning autonomously the different process low-level controllers. Due to the importance of supervisory control and the fact that multivariable controllers are great interest in the industrial and engineering fields, this chapter presented the autonomous intelligent multivariable multiple-controller framework for the control of complex MIMO plants. The proposed multivariable approach extends the SISO results shown in the previous chapter.

The proposed methodology uses a Mamdani fuzzy logic system for the supervision of the low-level multivariable multiple-controller which incorporates a GLM for MIMO system representation. In the GLM, the unknown complex process to be controlled is represented by an equivalent stochastic model consisting of a linear time-varying sub-model plus a computationally-efficient MIMO RBF neural-network based learning sub-model. The employed multivariable adaptive controllers are the conventional PID adaptive controller and the PID structure based (simultaneous) pole and zero placement controller. Both controllers (multiple-controller modes 1 and 2) benefit from the simplicity of having a PID structure, operate using the same adaptive procedure and can be selected on the basis of the required performance measure. The proposed intelligent multiple-controller works by adaptively tracking a desired reference signal, achieving the desired output signal performance and penalising excessive control actions, in response to the current performance of the control systems.

To achieve fast recognition of the MIMO control process phases, the fuzzy logic supervisor employs a behaviour recogniser subsystem to assess the system's performance through the system output signals and control input signals. The tasks of choosing the appropriate control algorithms (either conventional PID or Pole-Zero

Placement) and online tuning of the parameters of the active control algorithm are performed by the other two subsystems of the fuzzy supervisor, namely the switching logic and tuning logic. Both subsystems operate through fuzzy logic rules and fuzzy sets which are designed according the required performance and control system stability restrictions. The switching and tuning decisions are based on the specific performance criteria and continual monitoring of the effectiveness of each multivariable controller in achieving these criteria in response to set-point changes, nonlinear dynamics and the external disturbances in the complex MIMO plant.

The chapter provided the derivation of the stability analysis of the proposed intelligent multivariable multiple-controller framework. The stability of every control mode is presented in addition to the stability of the fuzzy logic based switching and tuning mechanism.

The next chapter presents new applications of the proposed intelligent multiple-controller framework for SISO and MIMO complex systems.

Chapter 6

Applications of the New Intelligent Multiple-Controller

6.1 Introduction

As described in the previous two chapters, this thesis proposes a new intelligent multiple-controller framework for SISO and MIMO complex systems' modelling and control. The techniques that this thesis focus on are the improved approximation of the complex system nonlinear dynamics and disturbances using the RBF based GML, and the autonomous tuning and switching between the conventional PID controller and the Pole-Zero Placement controller using the fuzzy-logic based high level supervisor. Therefore, the simulation experiments conducted in this chapter will seek to highlight the importance of the improved approximation of nonlinear functions $f_o(.,.)$ and $\mathbf{f}_o(.,.)$, which form part of the SISO and MIMO control laws. Furthermore, the capabilities of the fuzzy-logic supervisor in improving the performance of the control system in tracking the target signals, dealing with nonlinearities and disturbances, maintaining low control actions and minimised system output variance. Also, we introduce a new challenging control problem which is the application of autonomous vehicle control.

The chapter will present three simulation applications. First, is a SISO water vessel system that will be applied to the SISO design of the intelligent multiple-controller. Second, is a MIMO water tank system which will be applied to the multivariable design of the intelligent multiple-controller. Last is the application of autonomous vehicle control problem.

6.2 SISO Water Vessel Problem

In chemical process industries, one of the most commonly occurring control problems is that of controlling the fluid levels in storage tanks or reaction vessels. In this example, the proposed intelligent multiple-controller for SISO cases is applied to a real world SISO system model shown in Figure (6.1) and described in [17, 46].

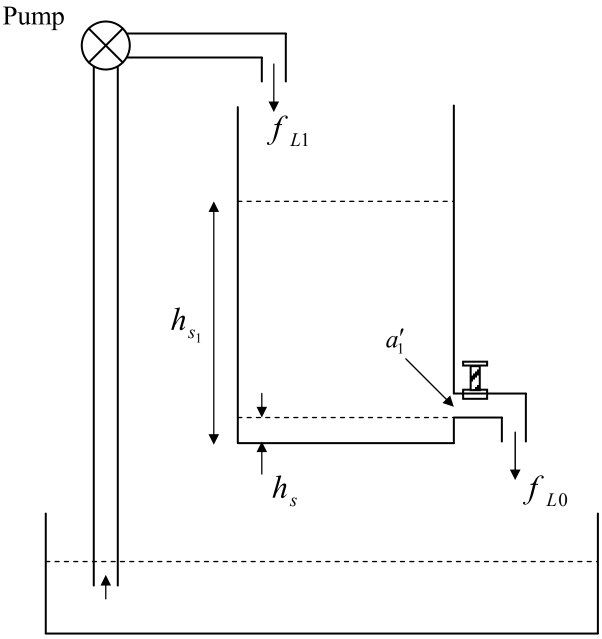


Figure (6.1): Water tank system.

The tank system is 10cm long, 10cm deep and 30cm high. The main objective of the control problem is to adjust the inlet flow f_{L1} so as to maintain the level in the tank h_{s_1} as close as possible to a desired set-point. The fluid flow rate into the tank (f_{L1}) is supplied by a pump. To measure this flow rate, a flow meter is inserted between the pump and the tank. The flow of water from the tank to reservoir (f_{L0}) is controlled by an adjustable tap. The maximum diameter of this tap is $0.70\,\text{cm}$. The depth of fluid is measured using a parallel track depth sensor which is located in the tank.

6.2.1 Model of the SISO Water Vessel System

The non-linear model can be presented as follows [17, 37, 46]:

$$A\frac{dh_{s_1}}{dt} = f_{L1} - a_1'\sigma_1\sqrt{2g(h_{s_1} - h_s)}\,, \tag{6.1}$$

where A is the cross section area of the tank, a_1' is the cross section area of orifice, σ_1 is the discharge coefficient (0.6 for a sharp edged orifice), h_s is the minimum water level and the acceleration gravity is denoted by $g = 9.81\,N/m^2$. The diameter of orifice is adjusted to 0.95cm and drain valve is fully open.

6.2.2 Simulation Setup

In order to demonstrate the closed loop performance of the intelligent multiple-controller for the SISO system, initially, it is arranged that each control mode (namely the conventional PID adaptive controller and the PID based Pole-Zero Placement controller) to work independently in controlling the whole control operation. Therefore, the fuzzy logic switching and tuning supervisor will not be involved in the control process. These simulation examples will illustrate the behaviour of each control mode in tracking a desired reference signal and also minimising the effect of the nonlinearities and disturbances in the complex SISO plant. Moreover, the experiments will present the effectiveness of using the RBF based GLM in approximating the nonlinear function $f_o(.,.)$ compared to the conventional MLP based GLM. Afterwards, in new simulation experiments, the intelligent supervisor will be employed to autonomously switch between the two controlling modes of the multiple-controller, and tune the active controller parameters.

The simulation examples will be performed over 600 sampling times. A first order linear model $([1+\hat{a}_1 z^{-1}]y(t) = z^{-1}\hat{b}_0 u(t))$ is used to identify the parameters of the process using RLS based linear sub-model of the GLM. The initial values for the plant parameters $\hat{a}_1$ and $\hat{b}_0$ were defined as 0.33 and 0.67 respectively [17]. The neural network based nonlinear sub-model is used to approximate the nonlinear function $f_o(.,.)$. The user defined gain and the user-defined polynomials were respectively selected as:

$v = 0.05$, $P_d(z^{-1}) = 1 + p_{d_1} z^{-1}$ and $P_n(z^{-1}) = 1 + p_{n_1} z^{-1}$, where $p_{d_1} = -0.4$, $p_{n_1} = -0.3$. The desired closed loop poles and zeros polynomials were selected as

$T(z^{-1}) = 1 + t_1 z^{-1} + t_2 z^{-2}$ and $\widetilde{H}(z^{-1}) = (1 + \widetilde{h}_1 z^{-1} + \widetilde{h}_2 z^{-2})$.

where, $t_1 = -0.5$, $t_2 = 0$, $\widetilde{h}_1 = 0.95$ and $\widetilde{h}_2 = 0$. Since the polynomial $\hat{A}$ is of order one, therefore a PI controller is obtained.

6.2.3 Fuzzy Supervisor Setup

The application of the SISO water vessel is a system with one control input and one system output. Therefore, the fuzzy sets used in the switching and tuning subsystems were built according to information derived from a single control input signal $u(t)$, a single system output signal $y(t)$ and a single reference signal $w(t)$. For instance, the following sample fuzzy rule, which forms part of the switching logic fuzzy rules given in chapter 4 at section (4.3.2.2), has two input parameters ($\zeta_y(t)$ and $V_y(t)$) determined from the system output signal $y(t)$ to define its overshoot and variance states in order to select a single controller $C_\eta(t)$ that will provide the next control action $u(t)$.

IF $\zeta_y(t)$ IS Ntive-High AND $V_y(t)$ IS High THEN $C_\eta(t)$ is Pole-Zero-Placement

Similarly, the input parameters of the tuning logic fuzzy rules are defined from a single control action and a single system output signal, and the fuzzy output parameter defines the tuning value of one active controller. The following fuzzy rule is a sample rule from among the list of the tuning logic fuzzy rules given in chapter 4 at section (4.3.3.2):

IF $C_\eta(t)$ IS PID AND $\zeta_y(t)$ IS NOT Norm THEN $v_\tau(t)$ IS Decrease

As shown in chapter 4 at sections (4.3.2.1) and (4.3.3.1), from the point of view of simplicity and computational complexity, the fuzzy switching and tuning subsystems' variables were represented by triangular (TriMF) and trapezoidal (TrapMF) membership functions with not more than two membership functions overlapping. For this application, 5% to 10% overlap between neighbouring MFs. Other fuzzy membership functions such as Gaussian and Sigmoid curves with different shapes and sizes were tested and the results were not significantly different.

In the design of fuzzy switching subsystem MFs for online controllers switching for the SISO water vessel problem, four factors are considered: degree of output signal overshoot $\zeta_y(t)$, variance in the output signal V_y, steady state error e_∞ and changes in the reference signal state Π_w. During the optimization process of the MFs of these factors, the TriMF and TrapMF scalar parameters a, b, c and d of Equations (4.11 and 4.12) were experimentally adjusted in order to contain quantitative information about the SISO water vessel system. The final fuzzy sets represent knowledge base about the physical system behaviour by preserving information including: output signal undershooting degree of -53.2%, overshooting degree of 47.4%, output signal variance limit of 18.12, and steady state error between 2% and -2% of the output signal. Details of the switching logic fuzzy sets are given in Table (6.1) and shown in Figures (4.2a-d) in chapter 4.

The switching logic output parameter (i.e. the selected controller) is represented by two TriMFs, one is for the PID controller and the other is for the Pole-Zero Placement controller, with 10% overlap (see Figure (4.3) in chapter 4). Other MFs can be used such as the example given in Figure (6.7a) where two TrapMFs were used with almost 0% overlap. The strategy of winner-take-all is used in the selection of the candidate controller. That is, by applying the Middle-of-Max defuzzification approach, the MF which has the highest amplitude will have its corresponding controller selected. In the case when the two controllers' MFs have equal amplitudes, then the current active controller will be selected.

Fuzzy Parameter -								
Type	**Name**	**Range**	**MF Type**	**MF Name**	**MF Scale Parameters**			
					a	b	c	d
Input	Output signal Overshoot	[-100, 100]	TrapMF	Ntive-High	-100	-100	-53.2	-0.01
			TriMF	Norm	-3.3	0	3.3	
			TrapMF	High	0.01	47.4	100	100
Input	Output Signal Variance	[0, 100]	TrapMF	Norm	0	0	2	6
			TrapMF	High	4.1	18.12	100	100
Input	Reference Signal State	[-100, 100]	TrapMF	Decreasing	-100	-100	-2	0
			TriMF	Norm	-0.01	0	0.01	
			TrapMF	Increasing	0	2	100	100
Input	Steady-State-Error	[-10, 10]	TrapMF	Ngtive	-10	-10	-2	-0.01
			TriMF	Norm	-0.02	0	0.02	
			TrapMF	Pstive	0.01	2	10	10
Output	Controller	[0, 40]	TriMF	PID	0	10	20	
			TriMF	PolZroPlcmnt	20	30	40	

Table (6.1): Switching logic parameters for the SISO water vessel system

The input and output parameters of the fuzzy tuning logic are represented by the TriMF and TrapMF membership functions. The scalar parameters a, b, c and d of these MFs (Equations 4.11 and 4.12) were experimentally adjusted in order to maintain the appropriate tuning values for the active controller parameters. For instance, the rise and fall times of the output signal of the SISO water vessel system are fuzzified through three MFs (Fast , Average and Slow) distributed in a range between $0 Sec$ to $100 Sec$ (see Table (6.2)). Applying the corresponding fuzzy rule, such as the example given next, new tuning values for the controller poles will be derived through the fuzzy set of the poles output parameters.

IF $C_\eta(t)$ IS PolZroPlcmt AND $\rho_y(t)$ IS Slow THEN $T_\tau(t)$ IS Faster

The range of the poles and zeros fuzzy sets represent the dimensions of the stability unit disc. Whereas the range of the PID gain v output fuzzy parameter ranges between -10 and 10, which are found to be the minimum and maximum stability gains for this application [17]. Details of the input and output fuzzy parameters are provided in Table (6.2) next.

Fuzzy Parameter -								
Type	**Name**	**Range**	**MF Type**	**MF Name**	**MF Scale Parameters**			
					a	b	c	d
Input	Controller	[0, 4]	TrapMF	PID	0	0	2	2.001
			TrapMF	PolZroPlcmnt	1.999	2	4	4
Input	Output signal Overshoot	[-100, 100]	TrapMF	Ntive-High	-100	-100	-53.2	-0.01
			TriMF	Norm	-3.3	0	3.3	
			TrapMF	High	0.01	47.4	100	100
Input	Steady-State-Error	[-10, 10]	TrapMF	Ngtive	-10	-10	-2	-0.01
			TriMF	Norm	-0.02	0	0.02	
			TrapMF	Pstive	0.01	2	10	10
Input	Rising Time	[0, 100]	TriMF	Fast	0	0	12.3	
			TrapMF	Average	8.52	10.3	23.21	25.1
			TrapMF	Slow	19.04	52.6	100	100
Input	Settling Time	[0, 100]	TrapMF	Fast	0	0	20	31.2
			TriMF	Optimal	20	40	60	
			TrapMF	Slow	53.4	73.54	100	100
Input	Control Action	[-10, 10]	TrapMF	NgtiveHigh	-10	-10	-9.6	0
			TriMF	Normt	-0.04	0	0.04	
			TrapMF	PstiveHigh	0	8.3	10	10
Input	Reference Signal State	[-100, 100]	TrapMF	Decreasing	-100	-100	-2	0
			TriMF	Norm	-0.01	0	0.01	
			TrapMF	Increasing	0	2	100	100
Output	Gain V	[-10, 10]	TriMF	Decrease	-10	-5	0	
			TriMF	NoChange	-0.01	0	0.01	
			TriMF	Increase	0	5	10	
Output	Poles	[-0.9, 0.9]	TrapMF	Faster	-0.9	-0.9	-0.82	-0.03
			TriMF	Average	-0.17	0	0.17	
			TrapMF	Slower	0.02	0.64	0.9	0.9
Output	Zeros	[-1, 1]	TrapMF	Decrease	-1	-1	-0.18	0.072
			TrapMF	Increase	-0.07	0.18	1	1

Table (6.2): Tuning logic parameters for the SISO water vessel system

6.2.4 RBF and MLP Neural Networks Based GLM Approximation

The simulations presented in this section are used to demonstrate the effectiveness and efficiency of using the RBF NN instead of the MLP NN in representing the nonlinear function $f_o(.,.)$. The examples that will be performed in this regard will focus on the capabilities of the RBF NNs in approximating the soft and hard nonlinearities as well as sharp disturbances, when compared to the MLP NNs. Other features of the RBF NNs outperforming the MLP NNs were discussed in sections (3.5), (4.4), (5.4) and well documented in the literature (see for example [12, 15, 23, 48, 49, 107]). These feature could include: the ability to uniformly approximate smooth functions; simpler to implement; need less computational memory; converge faster; require less training time; and fewer computational complexities to train the network online.

The results below were obtained after trying different designs for the MLP NN structure such as two hidden layers network and three hidden layers network with various neuron numbers. The best MLP NN design had one input layer with five units, two hidden layers with ten and seven units respectively, and one unit output layer. The five inputs to the MLP NN were the current and last control inputs $[u(t) \quad u(t-1)]$, current and last system outputs $[y(t) \quad y(t-1)]$, and a bias with the value of 1. The Sigmoid transfer function and the Backpropagation learning rule were employed. For the RBF NN structure design, there were four inputs included: the current and last control inputs $[u(t) \quad u(t-1)]$, current and last system outputs $[y(t) \quad y(t-1)]$, and the output layer was a single pure linear unit. The hidden layer had five Gaussian transfer function based neurons (units) with adjustable centres and adjustable units spread width. The weights of the RBF hidden layer and the weights of the MLP layers were randomly selected offline prior to the control operation and then online fine tuned using the delta rule.

Figure (6.2a) next shows the performance of the MLP NN in approximating the nonlinear dynamics of the plant under control. It can be seen that the approximation of the low magnitude target signal was very satisfactory, but when high magnitude signal involved (i.e. hard nonlinear dynamics) after the sampling time 200 the MLP network approximation was inconsistent. Similarly, in figure (6.2b) when low value random disturbances were added to the same nonlinear dynamics, the MLP NN failed to continue the approximation. Figures (6.2c and d) respectively illustrate reasonable performance of the MLP NN in approximation of soft nonlinearities and unstable performance for the estimation of soft nonlinearities mixed with sharp disturbances. On the other hand, Figures (6.3a, b, c and d) show the successful and stable approximation of the RBF NN to emulate soft and hard nonlinearities, correspondingly, when law and sharp disturbances were introduced to the nonlinear dynamics. These sample results showed the power of the RBF NN in the online consistent approximation of the nonlinear and unexpected behaviours of the complex SISO plant when compared to the MLP NN.

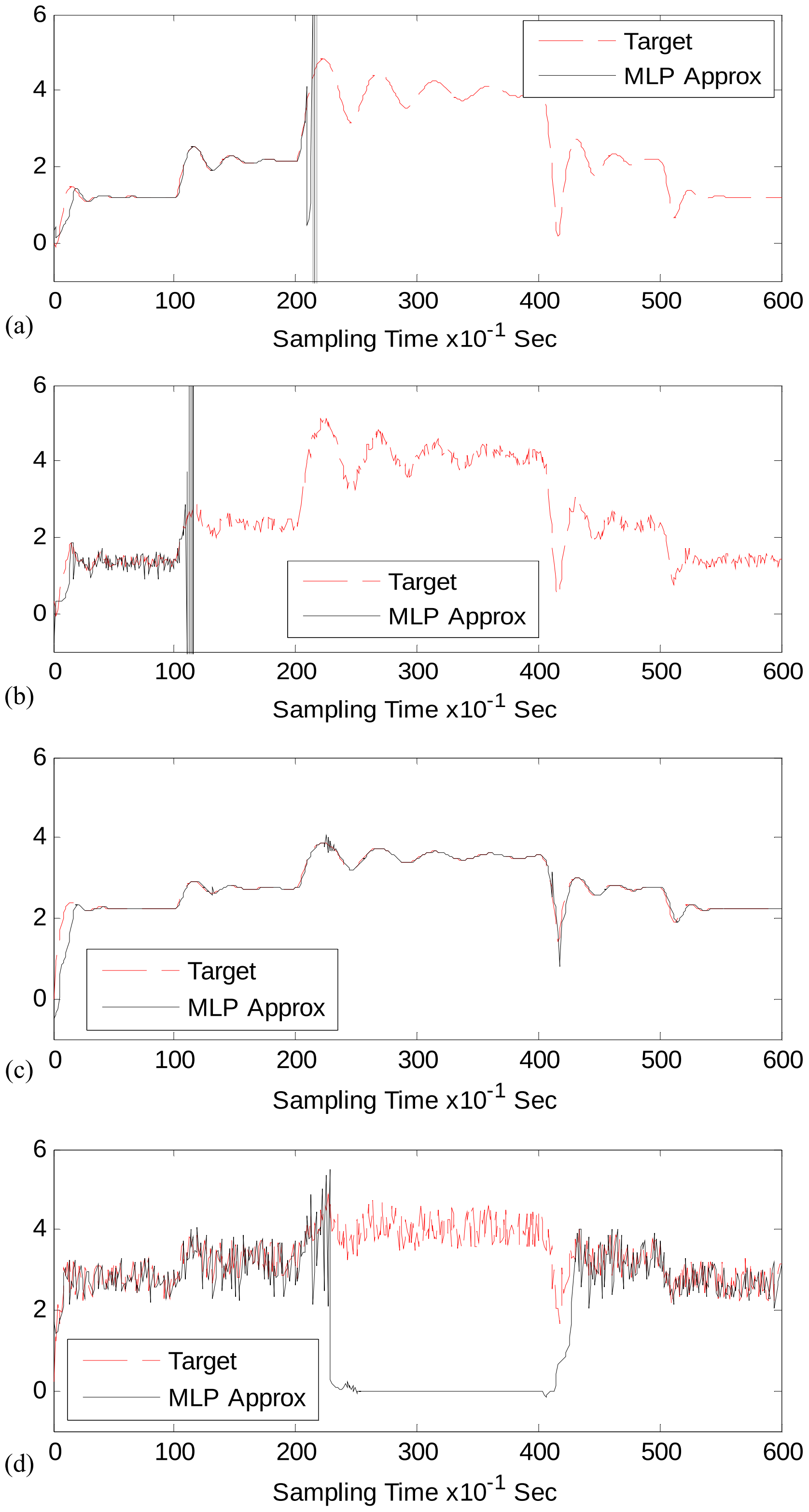


Figure (6.2): MLP NN approximation: (a) hard nonlinearity, (b) hard nonlinearity with disturbances, (c) soft nonlinearity, (d) soft nonlinearity with sharp disturbances.

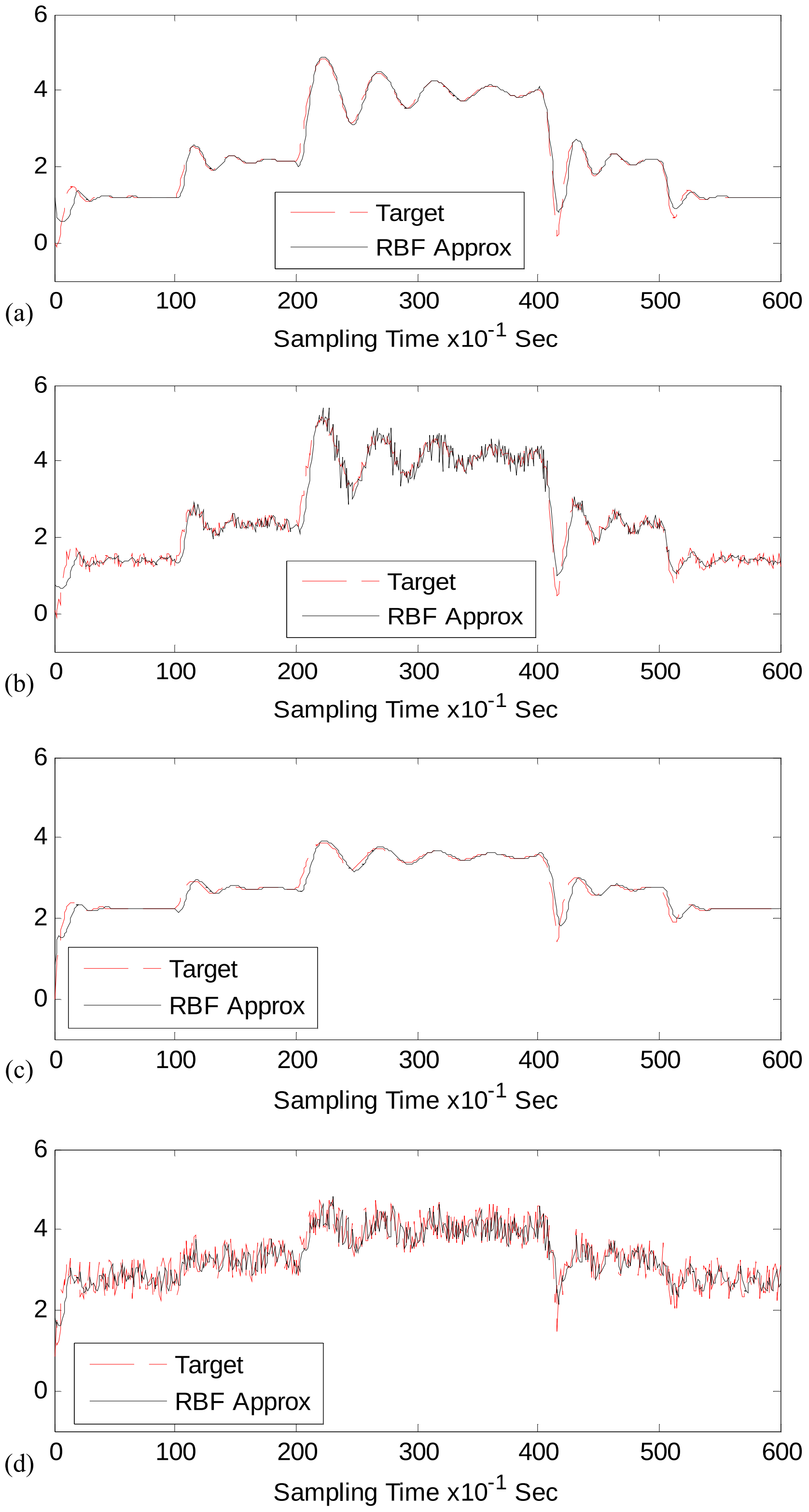


Figure (6.3): RBF NN approximation: (a) hard nonlinearity, (b) hard nonlinearity with disturbances, (c) soft nonlinearity, (d) soft nonlinearity with sharp disturbances.

6.2.5 Effect of the Nonlinear Sub-model in the GLM

To simplify the presentation, the SISO plant was under the control of the conventional PI controller during all the control operation. This experiment will present the importance of deploying the nonlinear learning sub-model of the GLM within the multiple-controller control law in order to approximate the nonlinear function $f_o(.,.)$, which represents the nonlinear dynamics and disturbances introduced to the control system.

In the obtained results below, Figures (6.4a, b and c) respectively illustrate the system output signal h_{s_1}, the control input signal f_{L_1} and the nonlinearities and disturbances affecting the control system when the nonlinear sub-model of the GLM is deactivated by setting polynomial H'_N (in equation (4.22)) to zero. Figures (6.5a, b and c) respectively show the system output signal h_{s_1}, the control input signal f_{L_1} and the nonlinearities and disturbances affecting the control system when the MLP NN is used to represent the nonlinear sub-model of the GLM to approximate the function $f_o(.,.)$. Figures (6.6a, b and c) respectively demonstrate the system output signal h_{s_1}, the control input signal f_{L_1} and the nonlinearities and disturbances affecting the control system when the RBF NN is used to represent the nonlinear sub-model of the GLM to approximate the function $f_o(.,.)$. The performance of the three cases (i.e. deactivating the nonlinear sub-model, employing the MLP NN and employing the RBF NN) is summarised in Table (6.3) next. The numerical obtained results show the great importance of the GLM in modelling the complex plant.

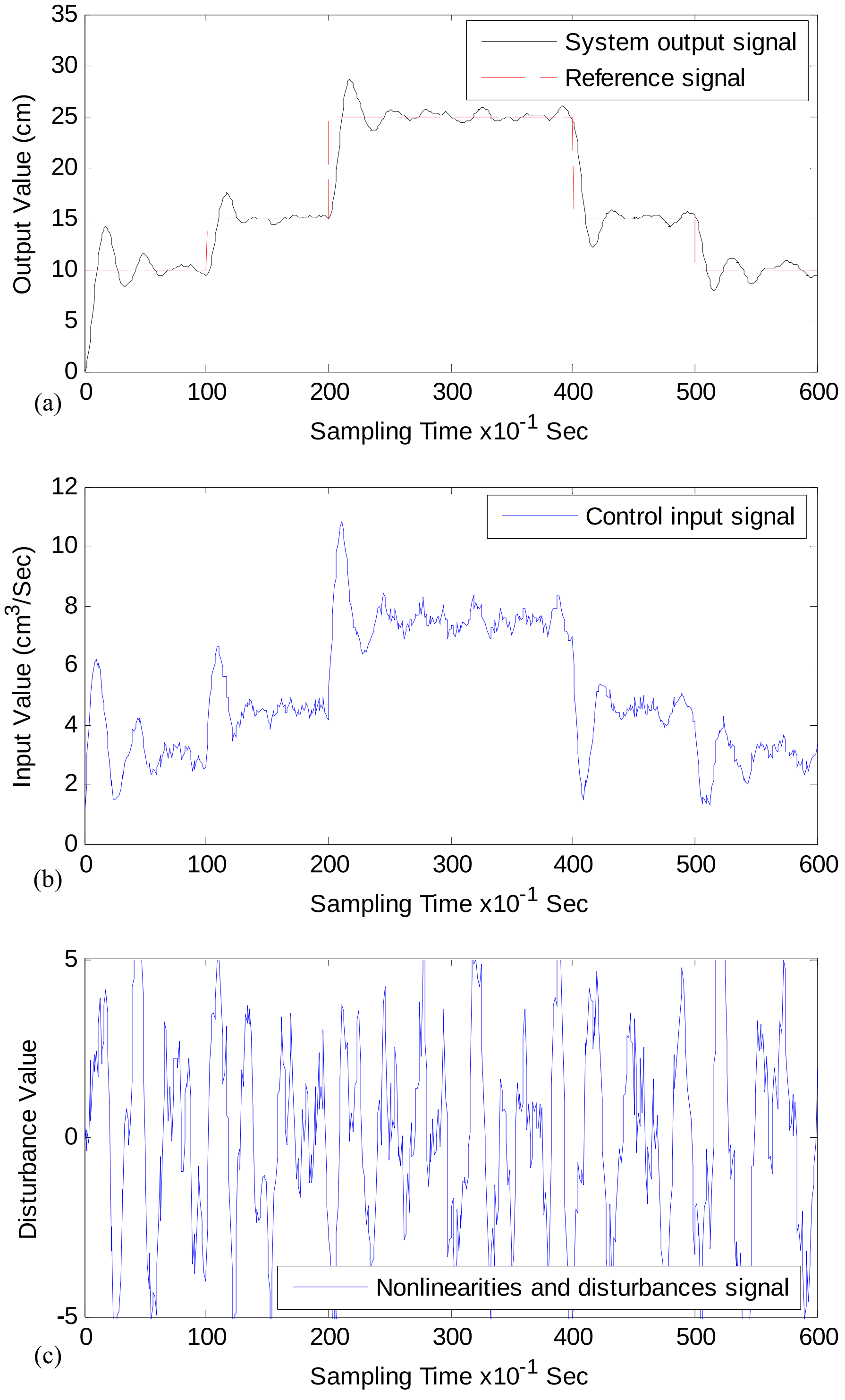
By comparing the computed data in Table (6.3), it can be seen that when the nonlinear function $f_o(.,.)$ was not considered in the control law the variance of the system output signal was clearly high if compared to the two other cases. It is obvious that the performance of the RBF based nonlinear sub-model resulted in minimum variance system output signal, and faster settling time. Similarly, the effect of the nonlinear dynamics and random disturbances was reduced more efficiently than the MLP based nonlinear sub-model. This can be seen by comparing the variance and the magnitude range of the nonlinearity and disturbance signals as listed in Table (6.3) and shown in Figures (6.4c), (6.5c) and (6.6c). It can be concluded that the RBF case reduced 99.2% of the variance and 85.7% of the magnitude range of the nonlinear function $f_o(.,.)$ resulting in smoother system output signal as shown seen in Figure (6.6a).

	GLM Nonlinear Sub-model		
	Not active	**MLP**	**RBF**
Variance of the output signal	0.1400	0.0782	0.0007
Settling time of output signal	7 Sec	7.5 Sec	6.2 Sec
Rising time of output signal	6.6 Sec	7 Sec	7 Sec
Variance of the nonlinearity and disturbances	7.5549	3.4853	0.0554
% Reduction in variance of nonlinearity and disturbances	0.0%	53.86%	99.2%
% Reduction in effect of nonlinearity and disturbances	0.0%	59.7%	85.7%

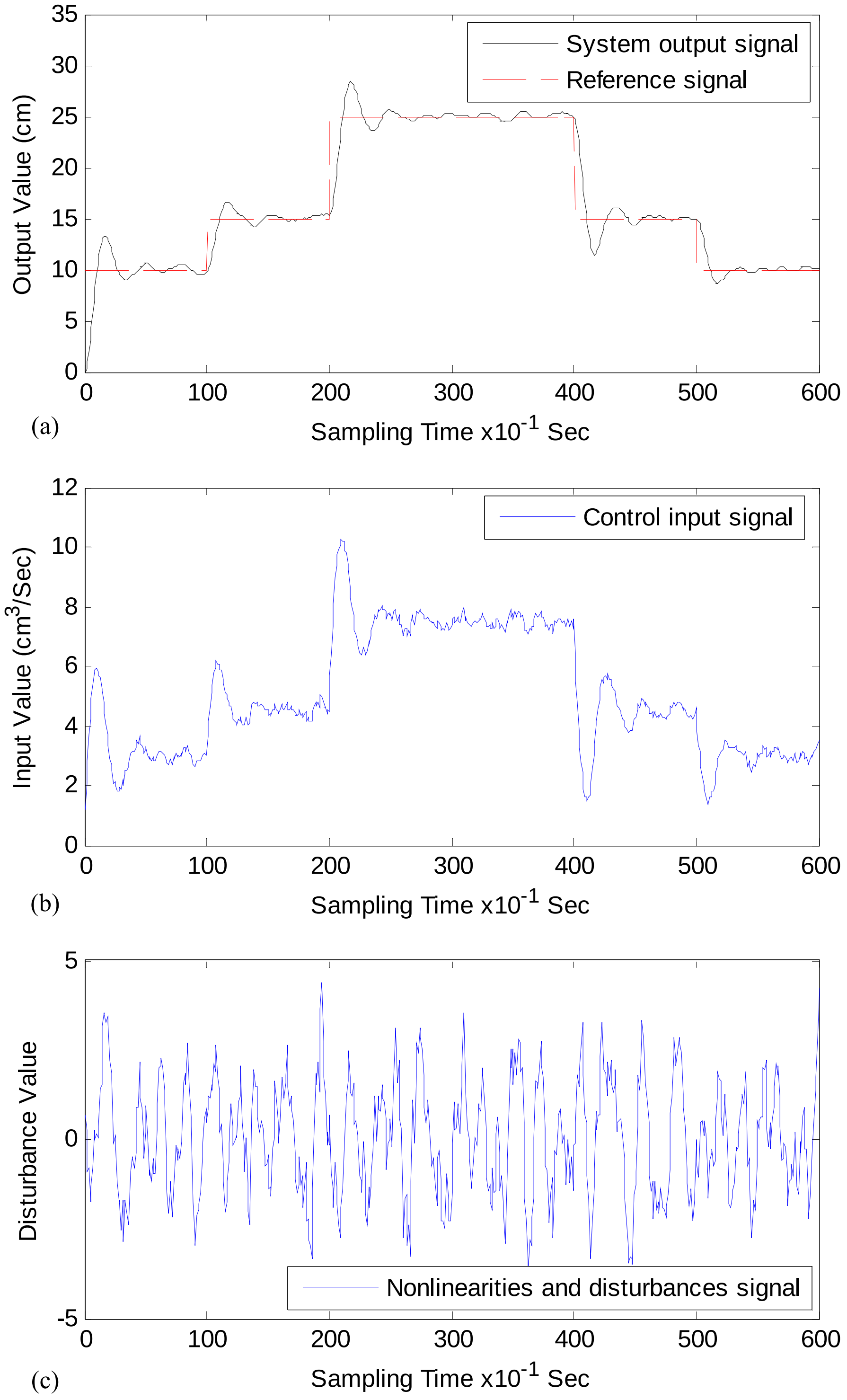
Table (6.3): SISO water vessel system performance factors for comparing the effect of the GLM nonlinear sub-model.

Considering the graphical results shown in Figures (6.4), (6.5) and (6.6), the system output signal in Figure (6.4a) has high oscillating behaviour during the steady state because the nonlinear dynamics and the disturbances of the complex plant were not included in the control law since the nonlinear sub-model of the GLM was deactivated. Consequently, the process nonlinear dynamics and disturbances caused high oscillations in the system output signal and high control actions. In contrast with the system output signal in Figures (6.5a) and (6.6a), where respectively the MLP and RBF neural networks implemented the nonlinear sub-model of the GLM, the steady state had less oscillation. Especially in Figure (6.6a) when the RBF neural network was used to approximate the function $f_o(.,.)$, the steady state was very smooth.

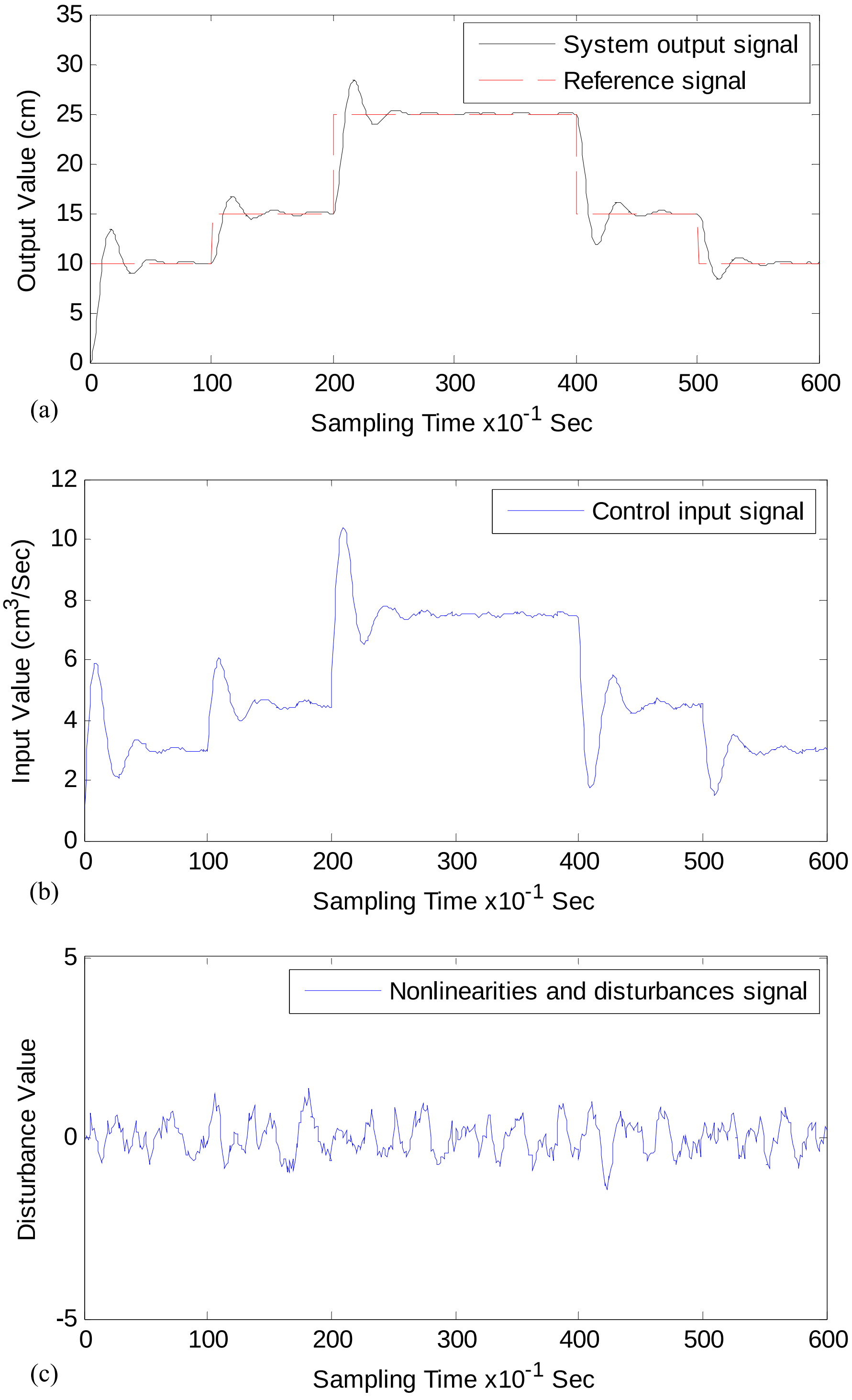
It can be seen from the results illustrated in Figures (6.5) and (6.6) below, that, compared to the case of the conventional nonlinear sub-model based on MLP neural networks, the use of the computationally efficient RBF based nonlinear sub-model can result in improved performance in terms of achieving minimum variance of the output signal and the control input signal, both for tracking changes in the reference signal and for dealing with the nonlinearities and added disturbances.



Figure(6.4): SISO water vessel system Behaviour when the nonlinear sub-model of the GLM is deactivated, (a) system output signal, (b) control input signal and (c) nonlinearities and disturbances affecting the system.



Figure(6.5): SISO water vessel system Behaviour when the MLP NN represents the nonlinear sub-model of the GLM, (a) system output signal, (b) control input signal and (c) nonlinearities and disturbances affecting the system.



Figure(6.6): SISO water vessel system Behaviour when the RBF NN represents the nonlinear sub-model of the GLM, (a) system output signal, (b) control input signal and (c) nonlinearities and disturbances affecting the system.

6.2.6 Control Performance of the Conventional Adaptive PI Only Controller

In this experiment, the multiple-controller is set to control the SISO water vessel system using only the conventional adaptive PI controller (multiple-controller mode 1). In the following experiments, the nonlinear sub-model of the GLM will be represented by the RBF NN. The acquired results, shown in Figures (6.7) to (6.9) below, give understating of the behaviour of this controller, which was part of the base knowledge used to design the proposed fuzzy logic supervisor.

In the simulation results, Figures (6.7a, b), (6.8a, b) and (6.9a, b) respectively illustrate the system output signal $y(t)$, the control input signal $u(t)$ when the gain v of the PI controller was set to 0.01, 0.1 and 1.1. Table (6.4) presents performance measures for the three cases of the PI controller. These measures include: the variance of the systems output signal $y(t)$, percentage of overshooting in the output signal $y(t)$, the settling time of $y(t)$, rise time of $y(t)$ and the maximum control action in the control input signal $u(t)$. The graphical and numerical measures demonstrated next give an idea about the important effect of the gain v on the performance of the PI controller. It can be seen that decreasing the gain v will minimise the variance of the output signal $y(t)$ and preserve low control action, but the settling time and rise time will be slowed down and the steady state error increased. On other hand, increasing the gain v will cause fast settling and rise time, and relatively high overshooting in the system output signal, moreover the variance of the output signal and the control action will increase.

Therefore, it can be concluded that the conventional PI controller preserves smooth steady state signal if the gain v tuned to low value. However, the conventional PI controller is not the best choice to govern the settling and rise time of the output signal that is due to the effect of increasing the gain v on maximising the variance of the output signal and increasing the control actions. Another point is that, the PI controller causes high overshooting and oscillatory performance in the system output signal when the gain v is not properly tuned.

	Conventional PI Controller		
	$v = 0.01$	$v = 0.1$	$v = 1.1$
Variance of the output signal	0.0016	0.1134	0.8622
% of Overshooting in the output signal	0.0%	45.2%	55.1%
Settling time of output signal	9.2 Sec	4.5 Sec	1.3 Sec
Rising time of output signal	5.7 Sec	8 Sec	3 Sec
Maximum control input	7.5 cm^3 / Sec	10.4 cm^3 / Sec	20.4 cm^3 / Sec

Table (6.4): Performance factors of the convention PI controller at different setting for the PID gain v.

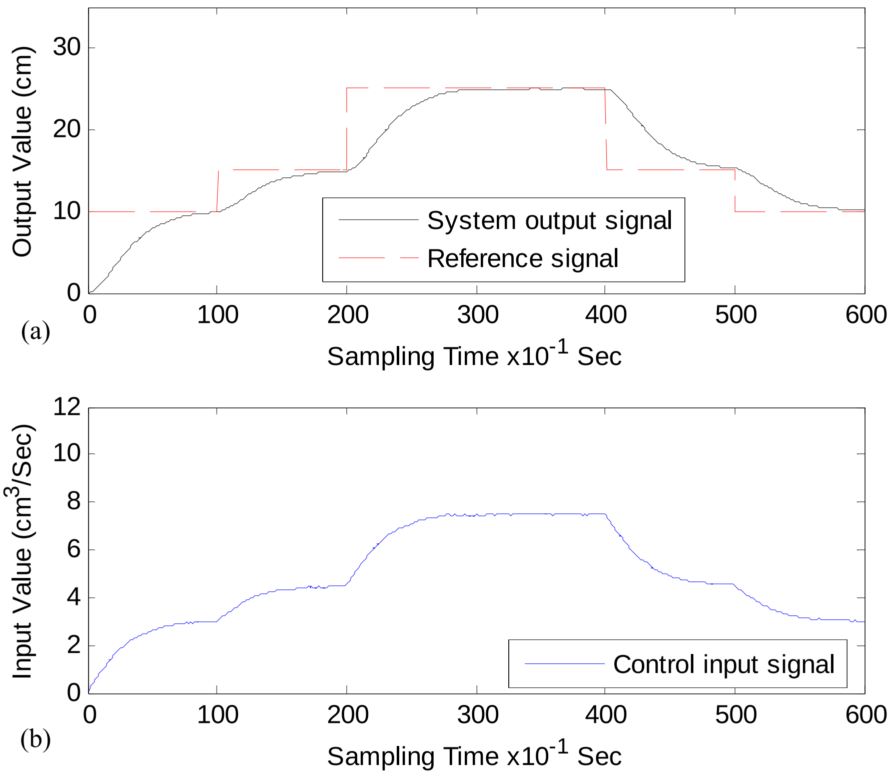


Figure (6.7): PI controller performance at $v = 0.01$, (a) system output signal, (b) control input signal.

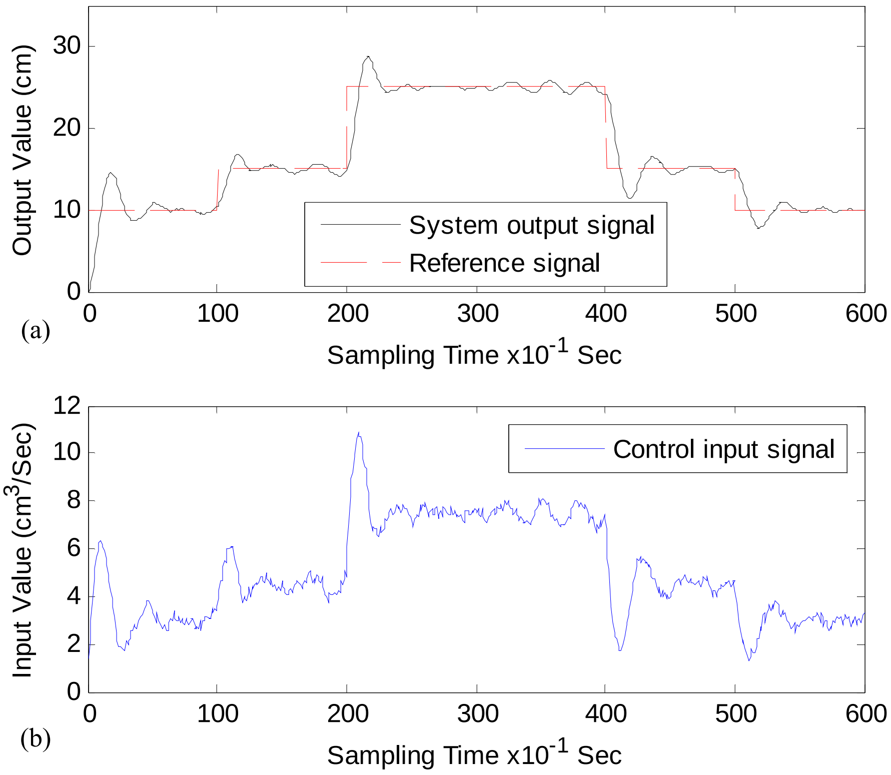


Figure (6.8): PI controller performance at $v = 0.1$, (a) system output signal, (b) control input signal.

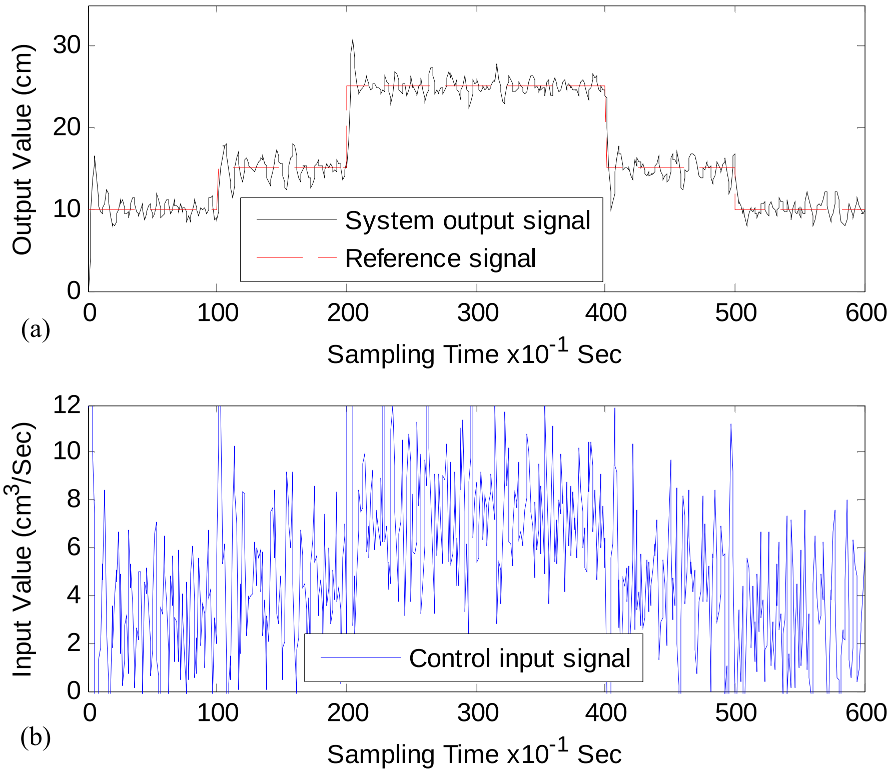


Figure (6.9): PI controller performance at $v = 1.1$,
(a) system output signal, (b) control input signal.

6.2.7 Control Performance of the Pole-Zero Placement Only Controller

The performance of the Pole-Zero Placement controller depends on the position of its poles and zeros [17, 37, 46]. Next, an experiment will be devoted to demonstrating the performance of the Pole-Zero Placement controller through changing the settings of the controller's poles. A following experiment will show the effect of tuning the controller's zeros. Therefore, the multiple-controller is set to control the SISO water vessel system using only the adaptive Pole-Zero Placement controller (multiple-controller mode 2).

6.2.7.1 Effect of the Poles

In this example, the gain v was set to 0.1 and the zero polynomial $\tilde{h}_1 = 0.95$. The results given in Figures (6.10a, b) and (6.11a, b) below, respectively illustrate the system output signal $y(t)$ and the control input signal $u(t)$ when the pole t_1 of the Pole-Zero Placement controller was set -0.5 (case 1) and -0.9 (case 2). Table (6.5) presents the performance measures for the two cases of the pole t_1. The measurements given in Table (6.5) show the variance of the system output $y(t)$, percentage of overshooting in the output $y(t)$, the settling time of $y(t)$, rise time of $y(t)$ and the maximum control action in the control input signal $u(t)$.

	Pole-Zero Placement Controller	
	$t_1 = -0.5$	$t_1 = -0.9$
Variance of the output signal	0.2	0.1
% of Overshooting in the output signal	9.0%	0.0%
Settling time of output signal	0.8 Sec	3.4 Sec
Rising time of output signal	0.5 Sec	2.3 Sec
Maximum control input	16.36 cm^3 / Sec	10.91 cm^3 / Sec

Table (6.5): Performance of the Pole-Zero Placement controller
with different pole settings.

From this experiment, it can be summarised that the settling time of the system output, as well as the rise time, can be manipulated by tuning the value of the poles of the Pole-Zero Placement controller. When fast settling and rise times are required, the system output signal exhibits slight overshooting at the beginning of the control operation, which can be treated by tuning the zeros as will be seen in the next experiment. As can be noticed in the results shown below in Figures (6.10a) and (6.11a), the drawback in the performance of this controller is the continuous and sharp oscillation during the steady state of the output signal $y(t)$. These kind of oscillations could damage or cause harm to the actuator of the control process [131], which is the water pump in this application. Therefore, it can be concluded that the Pole-Zero Placement controller can prevent system output overshooting, which could take place at reference signal changes and/or caused by the disturbances. In addition, tuning the poles of this controller can be useful in managing the settling time and rise time of the system output.

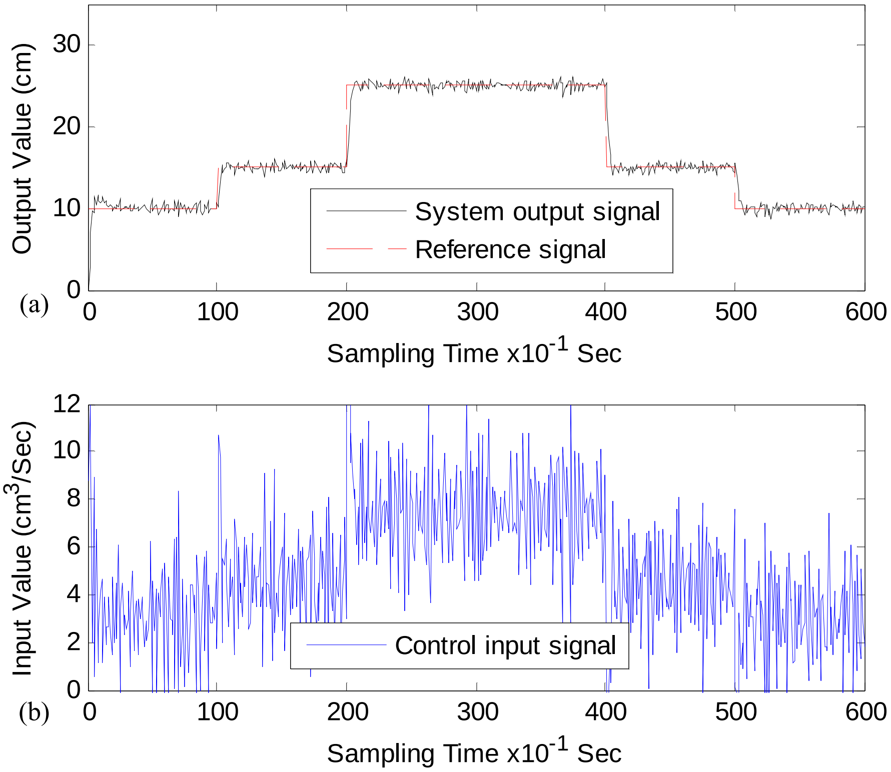


Figure (6.10): Pole-Zero Placement performance at $t_1 = -0.5$,
(a) system output, (b) control input.

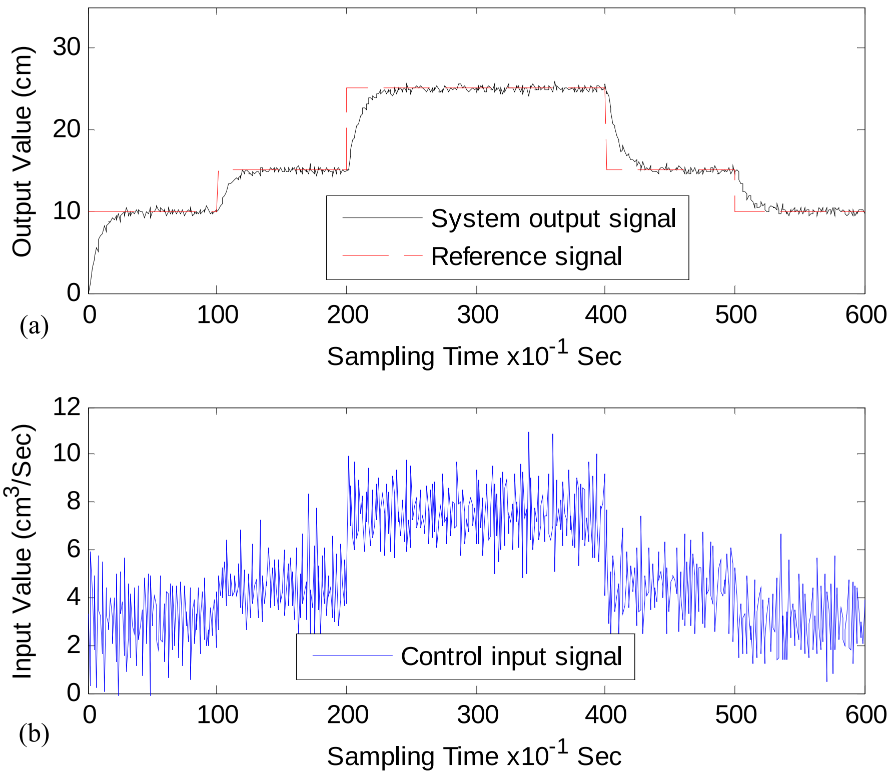


Figure (6.11): Pole-Zero Placement performance at $t_1 = -0.9$,
(a) system output, (b) control input.

6.2.7.2 Effect of the Zeros

The results shown in Table (6.6) and Figures (6.12), (6.13) and (6.14) next are used to illustrate the effect of the zeros of the Pole-Zero Placement controller. The gain v was set to 0.1 and the controller pole $t_1 = -0.7$ throughout the control operation conducted here. The control law zero $\tilde{h}_1$ was set to three different cases such as $\tilde{h}_1 = 0.95$, $\tilde{h}_1 = 0.55$ and $\tilde{h}_1 = 0.35$. The system output signals shown in Figures (6.12a), (6.13a) and (6.14a) expose very low overshooting at start of the control operation and almost zero overshooting during the rest of the process. As can be observed in the data given in Table (6.6), decreasing the zeros will minimise the magnitude of control input, as shown in Figures (6.12b), (6.13b) and (6.14b). Consequently, the variance and overshooting in the system output will be minimised, but the settling time and rise time could be affected. Generally, the zeros can be useful to manipulate the control action of the Pole-Zero Placement controller but in a range that should not cause negative effect on the settling and rise times, which can be managed more consistently by the poles of the controller (as explained in the previous experiment) [37, 100].

	Pole-Zero Placement Controller		
	$\widetilde{h}_1 = 0.95$	$\widetilde{h}_1 = 0.55$	$\widetilde{h}_1 = 0.35$
Variance of the output signal	0.1644	0.1475	0.1240
% of Overshooting in the output signal	2.1 %	1.2 %	0.3 %
Settling time of output signal	1.0 Sec	1.0 Sec	1.9 Sec
Rising time of output signal	0.6 Sec	0.6 Sec	0.85 Sec
Maximum control input	14.3 cm^3 / Sec	13.5 cm^3 / Sec	12.12 cm^3 / Sec

Table (6.6): Performance of the Pole-Zero Placement controller with different zeros' setting.

The observations and conclusions gained from the experimental results here formed the knowledge base for designing the fuzzy logic sets and rules of the fuzzy logic based supervisor for switching and tuning in the proposed intelligent multiple-controller. The next experiment will show the autonomous switching and tuning done by the fuzzy logic supervisor in order to switch between the two control modes and tune the control parameters of the active controller. That is to achieve a required system performance during the whole control operation.

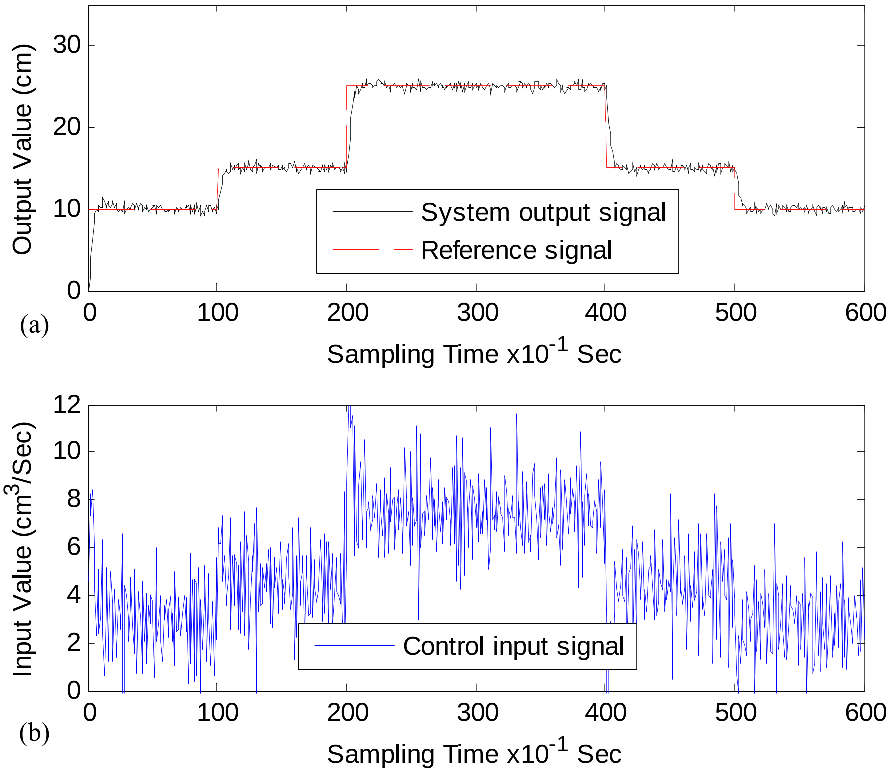


Figure (6.12): Pole-Zero Placement performance at $\tilde{h}_1 = 0.95$, (a) system output, (b) control input.

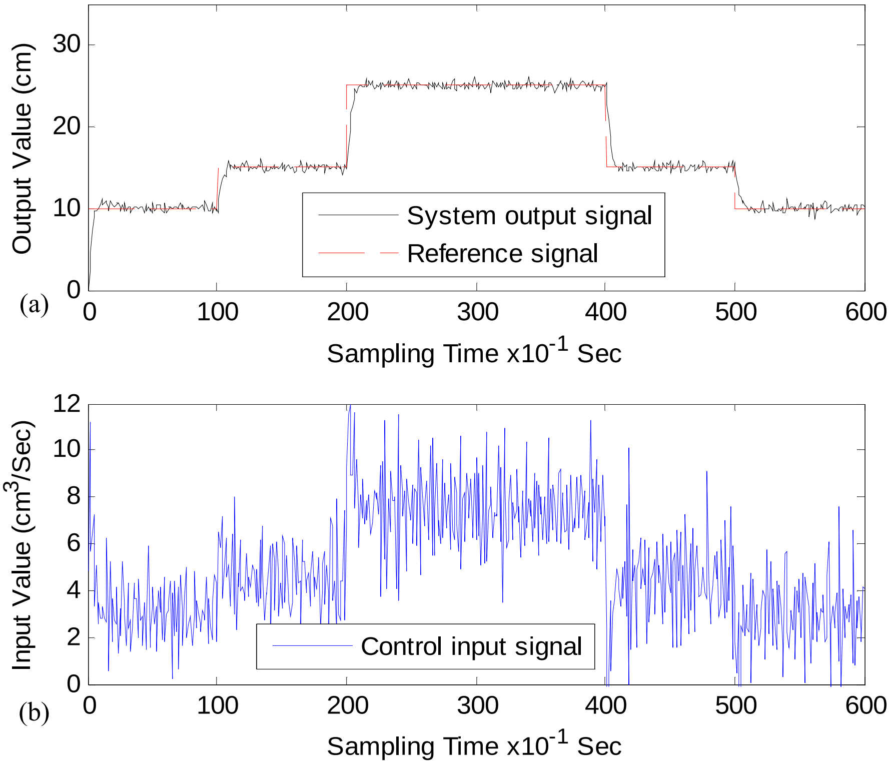


Figure (6.13): Pole-Zero Placement performance at $\tilde{h}_1 = 0.55$, (a) system output, (b) control input.

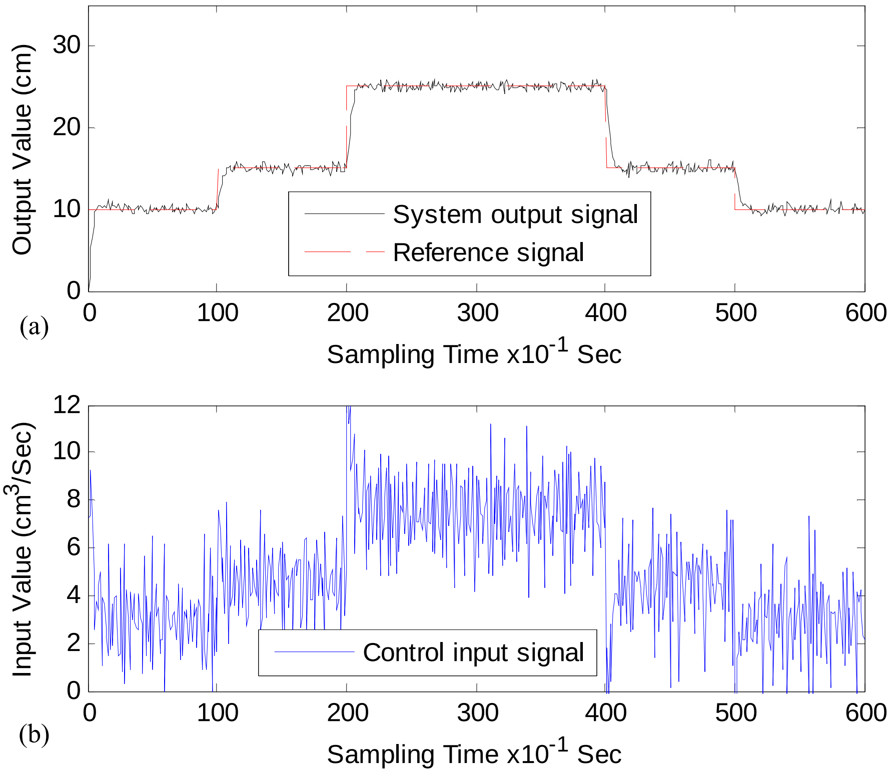


Figure (6.14): Pole-Zero Placement performance at $\tilde{h}_1 = 0.35$, (a) system output, (b) control input.

6.2.8 SISO Water Vessel Control using the Intelligent Multiple-Controller

The problem of regulating the water level of the water tank is divided into four sub-problems: preventing overflow situations (output signal overshooting); normal operation system (smooth steady state); preventing oscillatory output; and maintain low control input actions to keep the system actuator (water pump) [8]. The fuzzy logic based supervisor is designed to switch between the two multiple-controller modes to solve these sub-problems according to the system behaviour detected by the behaviour recogniser. The controller that solves the first sub-problem is the PI structure based Pole-Zero Placement controller (mode 2), whereas the controller that deals best with the second and third sub-problems is the conventional PI controller (mode 1) and finally the

PI-structured based Pole-Zero Placement controller (mode 2) most effectively tackles the fourth sub-problem (at the expense of a relatively greater computational requirement).

Another important aspect of the water storage tanks is the residence time, which is defined as the time necessary to discharge/charge the storage tank [132]. The residence time is one of the most informative characterisations of the flow pattern in chemical reactors since it can provide information on how long the various elements have been in the reactor [133]. Designing a multiple-controller meets such time-domain specifications is an important feature of the new proposed intelligent multiple-controller. In order to demonstrate the effectiveness of the employed fuzzy logic based switching and tuning supervisor, it was arranged for the supervisor to switch between the two controlling modes as well as tuning the parameters of the active controller. Tuning the PID gain v, poles and zeros of the multiple-controller gives the facility to adjust rise and fall times of the system output signal in order to control the water tank residence time.

The results given below present the performance of the intelligent multiple-controller in tracking a reference signal denoting changing water levels. The main targets of this simulation are to preserve minimum variance and smooth steady state, prevent system output overshooting, keep low control actions and maintain user defined rise and fall times. Meeting these goals in the presence of nonlinear dynamics and sharp random disturbances is a challenging task for the proposed framework.

As evident from Table (6.7), which summarises the operation of the fuzzy logic supervisor, the intelligent multiple-controller performed minimum number of switching actions between the conventional PI controller and the Pole-Zero Placement controller (mode 1 and mode 2, respectively), and tuned the active controller parameters to achieve the target goals given above. Figure (6.15a) shows the slow charging of the water tank at the start of the control operation that is to meet the user defined specified 2.5 *Sec* rise time. At the sampling times 100 and 200, the Pole-Zero Placement polynomial t_1 was tuned to -0.53 in order to boost the rise time as required. Where as at the sample 500 t_1 was tuned to -0.91 that is to meet 2.6 *Sec* discharging time for the target 10 *cm* height water level. Tuning of the polynomial $\widetilde{h}_1$, required for low control actions, and tuning of the gain v, for smooth steady state, are give in Table (6.7). Figures (6.15b) and (6.15c) respectively illustrate the control input signal and autonomous switching scheme performed to achieve the system output shown in Figure (6.15a). It can be seen that at the sample 300 the supervisor activated the Pole-Zero Placement controller (mode 2) to prevent system output overshoot due to sudden increase in the random disturbances, then the supervisor switched back to the PI controller (mode 1) to maintain the steady state.

Figures (6.16a, b and c) below show the performance of the multiple-controller in achieving the same previous goals but with out any random disturbances introduced to the system (i.e., the nonlinear sub-model of the GLM represented only the nonlinear dynamics of the SISO plant). These results demonstrate the bumpless switching mechanism between the two control modes as a result of the compatible “initial conditions" of the controllers connected to the plant [100, 118]. Figure (6.16a) shows almost zero overshooting and zero variance system output signal, which could take place if the plant nonlinearities were poorly approximated.

		Controller Parameters				
Sampling Time	**Active Controller**	v	t_1	$\tilde{h}_1$	**Aim of the Switching action**	**Aim of the Tuning action**
0	Mode 2	0.1	-0.7	0.9	Prevent overshooting	No Action
30	Mode 1	0.313	--	--	Smooth and low steady state error	Reduce oscillation
54	Mode 1	0.021	--	--		Smooth steady state
99	Mode 2	--	-0.53	0.9	Prevent overshooting	Maintain requested rise time
110	Mode 1	0.021	--	--	Smooth and low error steady state	--
200	Mode 2	--	-0.53	0.42	Prevent overshooting due to high disturbances	Minimise control action
226	Mode 1	0.021	--	--	Smooth steady state	--
229	Mode 1	0.111	--	--	--	Minimise oscillation
236	Mode 1	0.0232	--	--	--	Maintain smooth steady state
300	Mode 2	--	-0.53	0.32	Prevent overshooting	Minimise control action
320	Mode 1	0.0232	--	--	Smooth steady state	--
500	Mode 2	--	-0.91	0.32	Prevent overshooting	Maintain requested fall time
547	Mode 1	0.433	--	--	Smooth steady state	Prevent oscillation and low steady state error
555	Mode 1	0.0312	--	--	--	Maintain smooth steady state

Table (6.7): Summary of the fuzzy supervisor behaviour during level control of SISO water tank system.

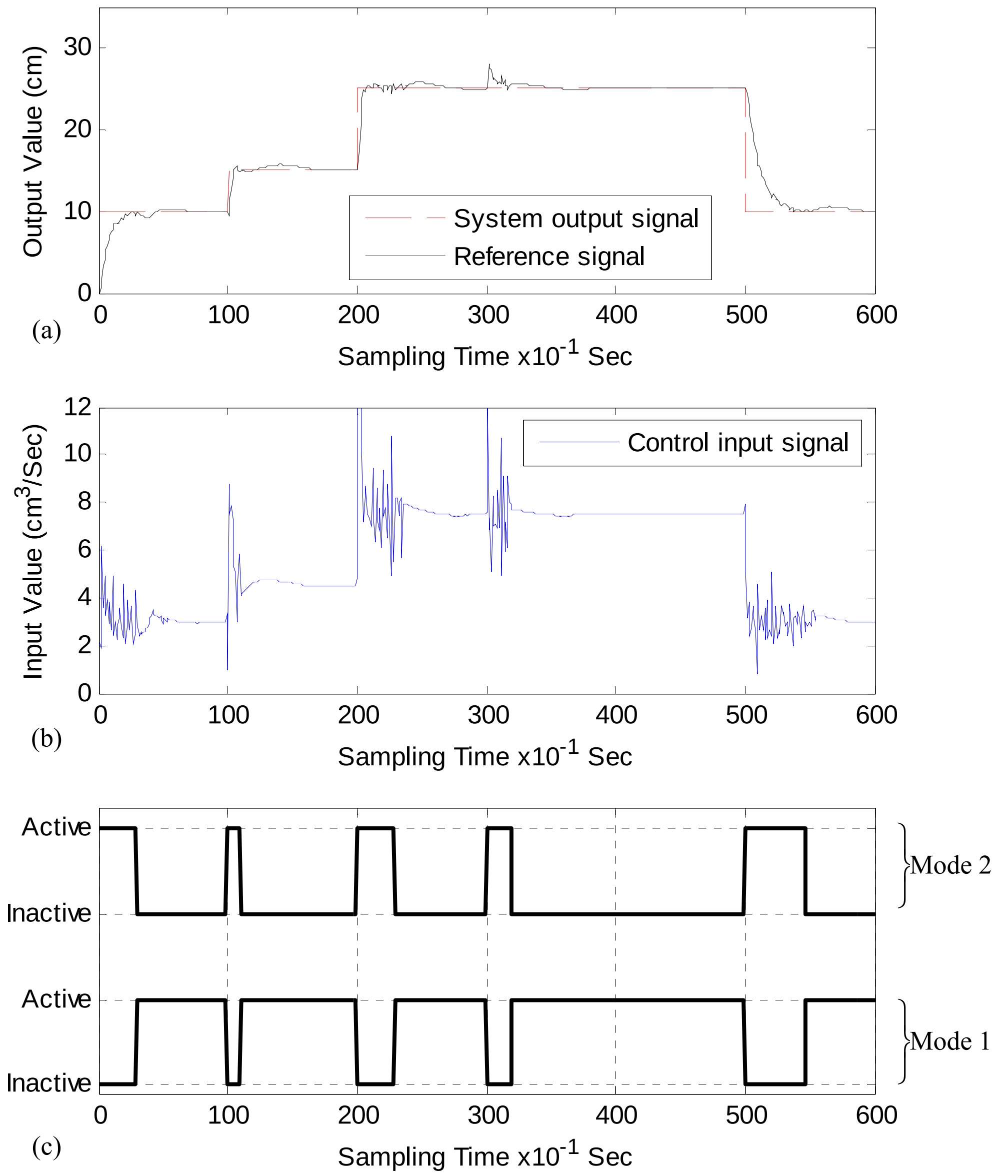


Figure (6.15): Intelligent multiple-controller performance during SISO water vessel control operation, (a) system output signal, (b) control input signal, (c) multiple-controller switching scheme.

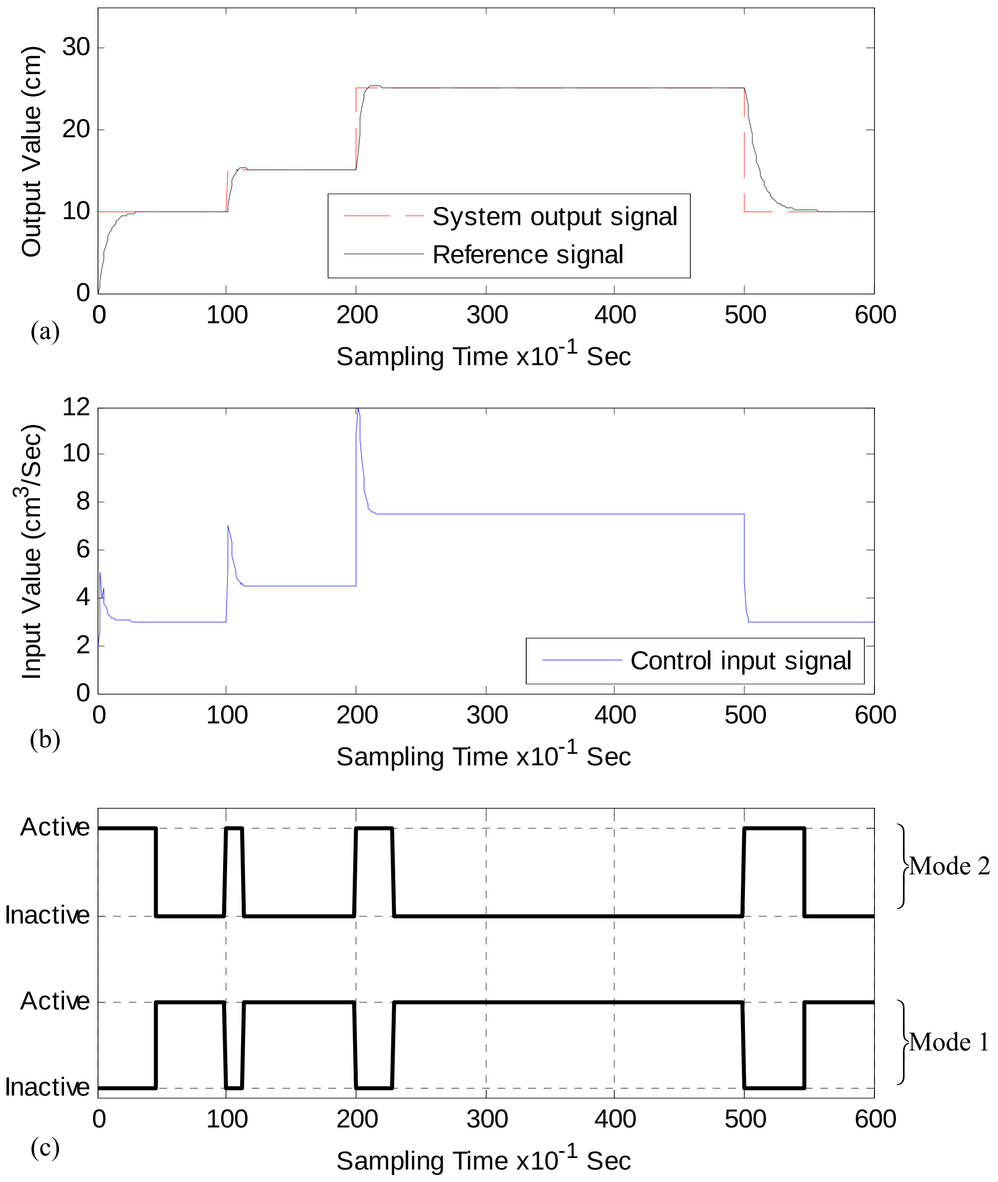


Figure (6.16): Intelligent multiple-controller performance when no random disturbances involved, (a) system output signal, (b) control input signal, (c) multiple- controller switching scheme.

6.3 MIMO Water Vessel Problem

In this example the proposed intelligent multiple-controller was applied to a real world MIMO system model shown in Figure (6.17) and described in [17, 70]. The two-input two-output coupled-tanks system comprises of one container with a centre partition to divide the container into two tanks. Both tanks are 10cm long, 10cm deep and 30cm high. At the base of the partition four holes are provided to allow flow of water between the tanks. These holes are at the height of 3 cm (i.e. $h_s = 3\,\text{cm}$) with different diameters of 1.27cm, 0.95cm, 0.635cm 0.317cm and together form orifice 1, which is adjustable by plugging one or more of the holes. The main objective of the control problem is to adjust the inlet flows f_{L1} and f_{L2} as to maintain the two tank levels (h_{s_1} and h_{s_2}) as close to a desired set-point. The fluid flow rates into tank 1 (f_{L1}) and tank 2 (f_{L2}) are supplied by two pumps. To measure these flow rates, two flow meters are inserted between pumps and tanks. The flow of water from tank 2 to the reservoir (f_{L0}) is controlled by an adjustable tap. The maximum diameter of this tap is $0.70\,\text{cm}$. The depth of fluid is measured using parallel track depth sensors which are located in tanks 1 and 2.

The non-linear model can be presented as follows [17, 70]:

$$A\frac{dh_{s_1}}{dt} = f_{L1} - a_1'\sigma_1\sqrt{2g(h_{s_1} - h_{s_2})}\,, \tag{6.2}$$

$$A\frac{dh_{s_2}}{dt} = f_{L_2} + a_1'\sigma_1\sqrt{2g(h_{s_1} - h_{s_2})} - a_2'\sigma_2\sqrt{2g(h_{s_2} - h_s')}\,, \tag{6.3}$$

where a_1' and a_2', are respectively the cross section area of orifice 1 and cross section area of orifice 2, and A is cross-sectional area of tank 1 and tank 2. σ_1 and σ_2 are the discharge coefficient (0.6 for a sharp edged orifice), and $g = 9.81 N/m^2$. The diameter of orifice 1 is adjusted to 0.95cm and drain valve is fully open.

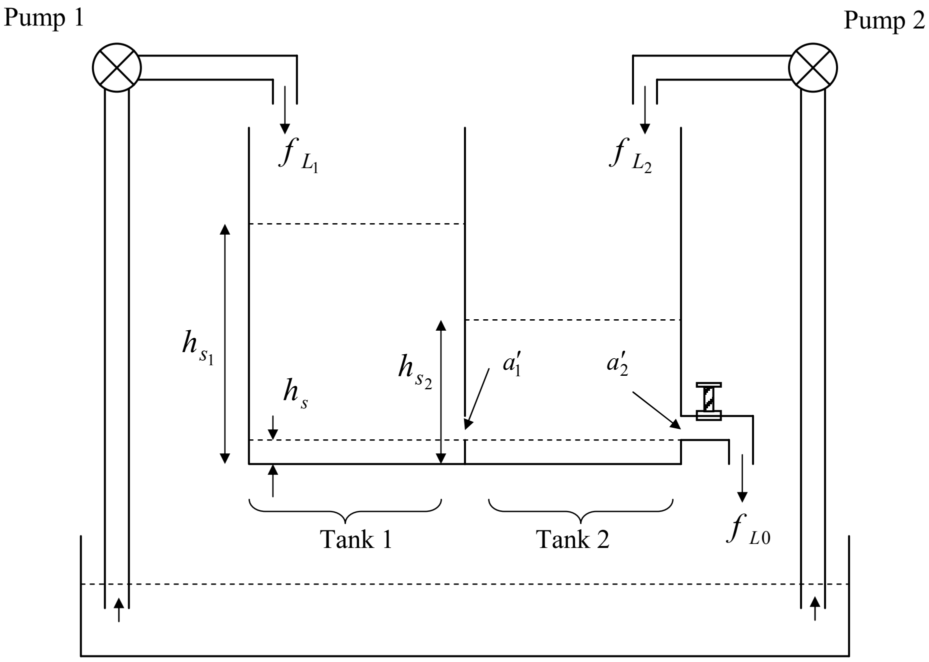


Figure (6.17): Coupled-tanks system.

6.3.1 Simulation Setup

The plant model of the complex MIMO system consists of the first order linear model $[\mathbf{I} + \hat{\mathbf{A}}_1 z^{-1}]\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = z^{-1}\hat{\mathbf{B}}_0 \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$ which used to identify the linear parameters of the process using RLS based linear sub-model of the MIMO GLM, and the nonlinear function $\mathbf{f}_{0,t}(\mathbf{U}, \mathbf{Y})$ which represents the nonlinearities and disturbances in the MIMO plant and approximated by the RBF based nonlinear sub-model of the GLM. The system output signals $y_1(t)$ and $y_2(t)$ denote the MIMO tank water levels h_{s_1} and h_{s_2}

respectively, the control input signals $u_1(t)$ and $u_2(t)$ denote the inlet flows f_{L1} and f_{L2} respectively. The MIMO plant parameters were initially defined as follows:

$\hat{\mathbf{B}}_1 = \begin{bmatrix} 0.33 & 0 \\ 0 & 0.33 \end{bmatrix}$, $\hat{\mathbf{B}}_0 = \begin{bmatrix} 0.67 & 0 \\ 0 & 0.67 \end{bmatrix}$, The matrices $\mathbf{U}$ and $\mathbf{Y}$ represent the inputs to the RBF neural network and denoted as:

$\mathbf{U} = \begin{bmatrix} u_1(t) & u_1(t-1) \\ u_2(t) & u_2(t-1) \end{bmatrix}$, $\mathbf{Y} = \begin{bmatrix} y_1(t) & y_1(t-1) \\ y_2(t) & y_2(t-1) \end{bmatrix}$, where $u_1(t-1)$, $u_1(t)$, $u_2(t-1)$ and $u_1(t))$ are the last and current of the two control input signals and $y_1(t-1)$, $y_1(t)$, $y_2(t-1)$ and $y_1(t))$ are the last and current of the two system output signals. The initial PID gain matrix ($\mathbf{V}$) and the user-defined polynomial matrices were respectively selected as:

$\mathbf{V} = \begin{bmatrix} 1.1 & 0 \\ 0 & 0.63 \end{bmatrix}$, $\mathbf{P}_d(z^{-1}) = \mathbf{I} + \mathbf{P}_{d_1} z^{-1}$ and $\mathbf{P}_n(z^{-1}) = \mathbf{I} + \mathbf{p}_{n_1} z^{-1}$, where $\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$,

$\mathbf{P}_{d_1} = \begin{bmatrix} -0.8 & 0 \\ 0 & -0.9 \end{bmatrix}$ and $\mathbf{P}_{n_1} = \begin{bmatrix} -0.5 & 0 \\ 0 & -0.6 \end{bmatrix}$.

The initial closed loop poles and zeros are respectively selected as:

$$\mathbf{T} = \mathbf{I} + \begin{bmatrix} -0.8 & 0 \\ 0 & -0.8 \end{bmatrix} z^{-1} \text{ and } \tilde{\mathbf{H}} = \mathbf{I} + \begin{bmatrix} 0.95 & 0 \\ 0 & 0.52 \end{bmatrix} z^{-1}.$$

During the 600 simulation samples the intelligent multivariable multiple-controller was set to track the set-point changes $\mathbf{w}(t)$, which denotes the target water level for each of the two tanks, where $\mathbf{w}(t) = [w_1(t) \quad w_2(t)]^T$.

6.3.2 Fuzzy Supervisor Setup

The application of the MIMO water vessel system has two control inputs and two system outputs. Consequently, the fuzzy sets used in the multivariable multiple-controller switching and tuning subsystems were built according to information derived from two control inputs ($u_1(t)$ and $u_2(t)$), two system output signals ($y_1(t)$ and $y_2(t)$), and two reference signal ($w_1(t)$ and $w_2(t)$). Therefore, the fuzzy rules of the switching and tuning subsystems have their fuzzy parameters derived from the two control inputs and two system outputs. The following sample fuzzy rule, which belongs to the fuzzy rules given in chapter 5 at section (5.3.2.2), represents the case when any of the two output signals ($y_1(t)$ and $y_2(t)$) has high undershoot and high variance will have its corresponding controller switched to the Pole-Zero Placement controller.

IF $\zeta_{y_1}(t)$ OR $\zeta_{y_2}(t)$ IS Ntive-High AND $V_{y_1}(t)$ OR $V_{y_2}(t)$ IS High THEN $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$ IS Pole-Zero-Placement

Similarly, the following multivariable multiple-controller tuning fuzzy rule is a sample rule derived from the list of the general tuning logic fuzzy rules given in chapter 5 at section (5.3.3.2):

IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$ IS PID AND $\zeta_{y_1}(t)$ OR $\zeta_{y_2}(t)$ IS NOT Norm THEN $v_{\tau_1}(t)$ OR $v_{\tau_2}(t)$ IS Decrease

The above rule checks the overshoot of both signals $y_1(t)$ and $y_2(t)$ in order to provide tuning values for the gains v_1 or v_2 of the corresponding active PID controller.

The membership functions of the input and output fuzzy parameters used for the MIMO water vessel application are similar to the input and output fuzzy parameters used for the SISO water vessel application above, section (6.2). The reason for applying the same MFs is due to the fact that the tank dimensions and water pump specifications are similar for both the SISO and MIMO applications. In addition, the two applications have similar control input and system output behaviours, such as: limits of the output signals overshoot and variance, steady state error, rise time, poles and zeros stability limits …etc.

6.3.3 Experimental Results

There are two simulation experiments conducted with their obtained results given below. The first example experiment is aimed to show the interactions occurring between the two system's output signals $y_1(t)$ and $y_2(t)$ of the complex MIMO water tank system. These interaction events happen due the nature of the MIMO plant model since each system output is affected by the other, as can be seen in Equations (6.2) and (6.3) above. To demonstrate the significance of the RBF based MIMO GLM employed in the proposed multiple-controller, the nonlinear sub-model representing the nonlinear function $\mathbf{f}_{0,t}(\mathbf{U},\mathbf{Y})$ is deactivated by setting the parameter $\mathbf{H}'_N$ in Equation (5.18) to zero. Random disturbances will not be included in simulations below, that is to simplify the presentation and due to the obvious influence of the unmodelled nonlinearities on the MIMO system behaviour. Another point is that, in the first experiment, there will be no controller switching and no online controller parameter tuning. So, it is set that the output $y_1(t)$ to be controlled by the Pole-Zero Placement controller, and the output $y_2(t)$ to be controlled by the conventional PID controller.

The second experiment will illustrate the performance of the intelligent multivariable multiple-controller in eliminating the defects caused by the interaction accommodated in the complex MIMO system. Activating the nonlinear sub-model of the MIMO GLM and employing the fuzzy logic switching and tuning supervisor will serve the closed-loop MIMO system to prevent system outputs overshooting, achieve minimum variance outputs, penalise high control actions, and attain the desired system outputs rise and fall times.

6.3.3.1 Experiment One

Figures (6.18a, b and c) below, respectively show the system output signals $y_1(t)$ and $y_2(t)$, the control input signals $u_1(t)$ and $u_2(t)$, and the active controllers $C_1 =$ Pole-Zero Placement (Mode 2) and $C_2 =$ conventional PID (Mode 1). It can be clearly noticed in Figure (6.18a) that both outputs $y_1(t)$ and $y_2(t)$ experienced high overshooting and oscillatory behaviours due to the unmodelled nonlinearities and fixed controllers parameters. The two simultaneously operating controllers, Pole-Zero Placement and conventional PID, could not cope with the interactions taking place at the beginning of the control process and at the reference signals alterations. As a consequence, the magnified control inputs (see Figure (6.18b)) have increased the overshooting and variance in system output signals. Table (6.8) below shows high degrees of overshooting in both output signals especially in $y_2(t)$, which could be a result of the untuned gain v of the PI controller and the high nonlinear dynamics at the start of the control process. The variance of the outputs is given in the table below.

	$y_1(t)$	$y_2(t)$
% of Overshooting in the output signal	65%	129%
Variance of the output signal	2.2293	3.2267

Table (6.8): MIMO control system performance measures with unmodelled nonlinearities and no multiple-controller switching or tuning.

The results shown in experiment two next will demonstrate the satisfactory performance of the intelligent multivariable multiple-controller in dealing with the problems experienced in the first experiment above.

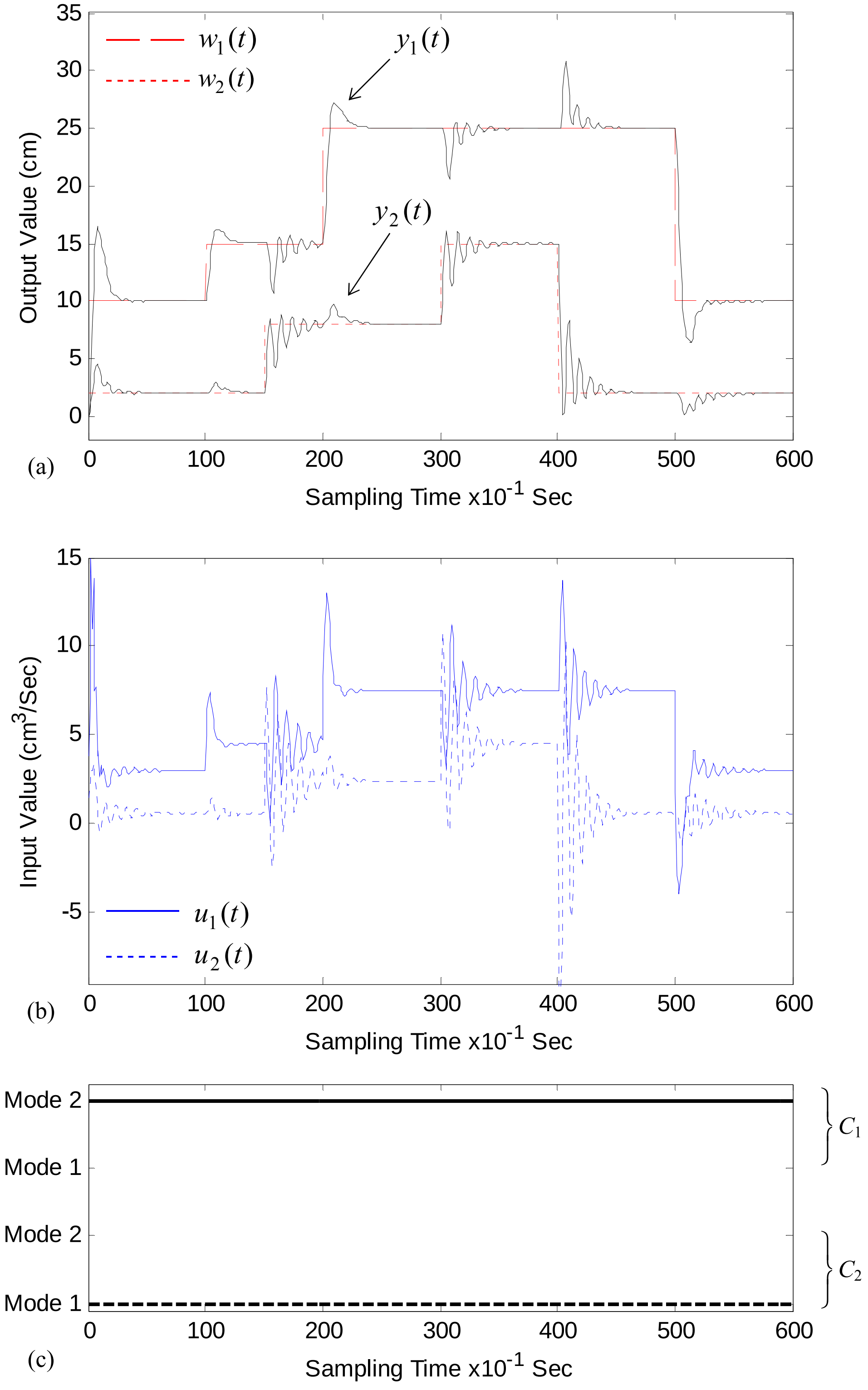


Figure (6.18): Multivariable multiple-controller controls the MIMO water tank system with unmodelled nonlinearities and fixed controller for each system output signal,
(a) two system outputs, (b) two control inputs, (c) active controllers with no switching nor tuning.

6.3.3.2 Experiment Two

In this example, the intelligent multivariable multiple-controller was set to automatically control the coupled tanks system in order to track the two defined water level reference signals $w_1(t)$ and $w_2(t)$. The complex system interactions were approximated using the RBF NN based nonlinear sub-model of the MIMO GML (presented in section (5.4), chapter 5 above). By appropriately designing the switching decision $\mathbf{C}_\eta$ for the multivariable conventional PI controller and multivariable Pole-Zero placement controller (based on the performance measure matrix $\boldsymbol{\Xi}(t)$ supplied to the behaviour recogniser), the fuzzy-logic supervisor worked to prevent any overshooting and minimize the steady state oscillations while controlling the MIMO water tank system. The capability of the fuzzy-logic based supervisor to online tune the multiple-controller parameters also allowed for controlling the rise and fall times of both output signals $y_1(t)$ and $y_2(t)$ according to user-defined requirements.

In the obtained results below, Figures (6.19a, b and c) respectively show the system output signals obtained, the control input signals $u_1(t)$ and $u_2(t)$, the multivariable controller selection scheme between the multiple-controller modes 1 and 2. Compared to the results obtained from the previous experiment, it can be seen from Figures (6.19a and b) that the fuzzy-logic based supervisor effectively managed to prevent output signals overshooting, simultaneously preserved relatively smooth steady state and almost zero output variance. Another point to note is that the excessive control action (resulting from reference set-point changes) is tuned most effectively when the supervisor activated the required controllers.

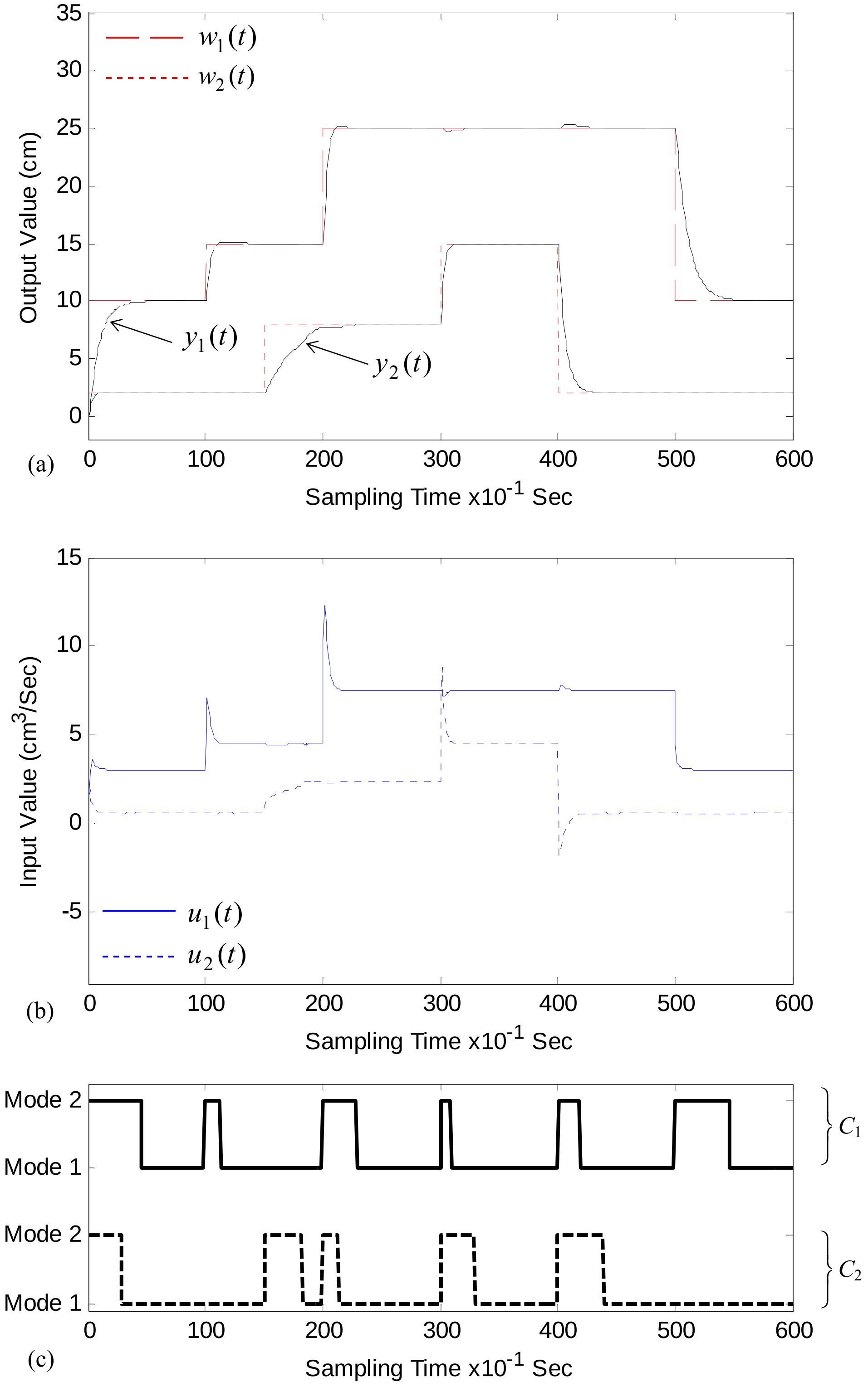


Figure (6.19): Performance of the intelligent multivariable multiple-controller to control coupled water tank system, (a) two system outputs, (b) two control inputs, (c) multiple-controller switching scheme.

Table (6.9) gives a summary of the improved overshooting and variance in the system output signals after the successful switching and tuning done by the fuzzy logic supervisor. The given results can be compared with Table (6.8) in the last experiment.

	$y_1(t)$	$y_2(t)$
% of Overshooting in the output signal	1.24%	0.0%
Variance of the output signal	0.0034	0.0032

Table (6.9): Intelligent multivariable multiple-controller performance measures throughout the control of complex MIMO water tank system.

In order to maintain the desired rise and fall times for both system output signals, the fuzzy supervisor effectively tuned the poles and zeros of the multivariable Pole-Zero placement controller. It can be seen in Figure (6.19a) system output $y_1(t)$, that at the beginning of charging tank 1 the rise time was 2.8*Sec*. A faster tank charging time of 1*Sec* was requested to reach water levels 15*cm* and 25*cm*, which was maintained to 1.12*Sec* with delay error of 0.12*Sec* that could be due to limits of the used plant model (Equations (6.2) and (6.3)). The system output $y_2(t)$ in Figure (6.19a) was successfully controlled to reach 8*cm* water level in 5.6*Sec* and discharge tank 2 to 2*cm* level in 2.4*Sec*. Table (6.10) gives sample tuned poles from the MIMO system control operation.

Time instant	y_1 Target rise/fall time	y_2 Target rise/fall time	Tuned poles
0	2.8*Sec*	1.8*Sec*	$\begin{bmatrix} -0.92 & 0 \\ 0 & -0.64 \end{bmatrix}$
100	1*Sec*	-	$\begin{bmatrix} -0.71 & 0 \\ 0 & -0.64 \end{bmatrix}$
150	-	5.6*Sec*	$\begin{bmatrix} -0.71 & 0 \\ 0 & -0.96 \end{bmatrix}$
500	3.8*Sec*	-	$\begin{bmatrix} -0.91 & 0 \\ 0 & -0.96 \end{bmatrix}$

Table (6.10): Sample pole tuning actions performed by the fuzzy supervisor to achieve user-request rise and fall times for the two system output signals.

6.4 Autonomous Vehicle Control Problem

Today's automobile effectively encompasses the spirit of mechatronic systems with its abundant applications of electronics, sensors, actuators, and microprocessor based control systems to provide improved performance, fuel economy, emission levels comfort, and safety [77, 134]. For almost two decades autonomous systems have been a topic of intense research. Since the mid 1980s, several research programs have been initiated all over the world, including Advances in Vehicle Control and Safety (AVCS) in Asia, Intelligent Vehicle Highway Systems (IVHS), and Partners for Advanced Transit and Highways (PATH) in the United States. Since 2004, the US Defence Advanced Research Project Agency (DARPA) has started to organize the DARPA Grant Challenge to test automatic-vehicle technology [135]. In Europe, the DRIVE and PROMETHEUS projects have aimed at increasing the safety and efficiency in normal traffic and at reducing the adverse environmental effects of the motor vehicle [136]. In France, projects such as Praxitele and "La rout automatiée" focus on driving in urban environments, as do the EU's Cybercars and CyberCars-2 projects. Another European project, Chauffeur, focuses on truck platoon driving [135]. Thus, many research groups are focusing on the development of functionalities for autonomous road vehicles that are able to interact with other vehicles safely and cooperatively [136, 137, 138].

An important component of Adaptive Cruise Control (ACC) is to design control systems for controlling the throttle, brake and steering systems so that the vehicle can follow a desired path and target speed, which could be responses of the leading vehicle and at the same time keep a safe inter-vehicle spacing under the constraint of comfortable driving [139, 140]. There are though a lot of possible techniques with which to perform ACC. Conventional methods based on analytical control generate good results but exhibit high design and computational costs since the application object, a car, is a nonlinear element and a complete mathematical representation is impossible. As a result, other means of reaching human-like speed control have been recently developed, for example, through the application of artificial intelligence techniques [78].

One of the important and challenging problems in ACC relates to dangerous yaw motions of the automobile that may result from unexpected yaw-disturbances caused by unsymmetrical car-dynamics perturbations like side-wind forces, unilateral loss of tire pressure or braking on unilateral icy road. One approach for yaw dynamics improvement is to use individual wheel braking, thereby creating the moment that is necessary to counteract the undesired yaw motion. Another approach is to command additional steering angles to create the counteracting moment [79]. Another alternative approach, which is used in this work, is to treat the three drivetrain sub-systems (i.e., throttle, brake and steering sub-systems) as one MIMO plant. Therefore, the interactions between the vehicle longitudinal and lateral properties, disturbances and nonlinearities are considered in the multivariable control law and modelled using the MIMO neural network based GLM.

6.4.1 Longitudinal and Lateral Vehicle Model for Autonomous Vehicle Control

In general, the process of autonomous vehicle control consists of two stages. During the first stage, the desired path and vehicle speed are determined on the basis of the driving environment. In the second stage, the vehicle is operated with the aim of realising the anticipated path and speeds. Basically, the autonomous vehicle controller is composed of three modules: a driver decision module, co-ordinates' transfer module and driver following module [141]. The driver decision module provides the desired path $Y_{path} = F_{path}(x)$ and speed $V_{ref} = G_v(S_x)$ described in the space domain, where S_x is the distance along the path $Y_{path} = F_{path}(x)$. The co-ordinates' transfer module changes the target path and speed into the time domain as $y_{path} = f_{path}(t)$ and $v_{ref} = g_v(t)$ in order to form the input for the driver-following module. On the other hand, the driver following module outputs the desired control elements which are the steering wheel angle δ_{sw}, throttle angle θ and brake torque T_b [145].The driver decision and the co-ordinate transfer modules are not covered by the discussion in this thesis. Assuming that the desired path and vehicle speed have been provided, the focus is on the approaches of achieving the above desired control elements (δ_{sw}, θ and T_b) in order to follow the target path in the desired speed.

A simplified functional diagram of a MIMO vehicle model control system incorporating the proposed intelligent multivariable multiple-controller is given in Figure (6.20) [138, 141, 145]. Each block can be considered as a subsystem with various inputs and outputs. The throttle, brake and steering subsystems are considered as one MIMO system with the desired throttle angle θ_t, braking torque T_t and steering wheel angle δ_t as inputs, and throttle angle θ, braking torque T_b and steering wheel angle δ_{sw} as outputs, which will deliver the vehicle to the desired path in the desired speed.

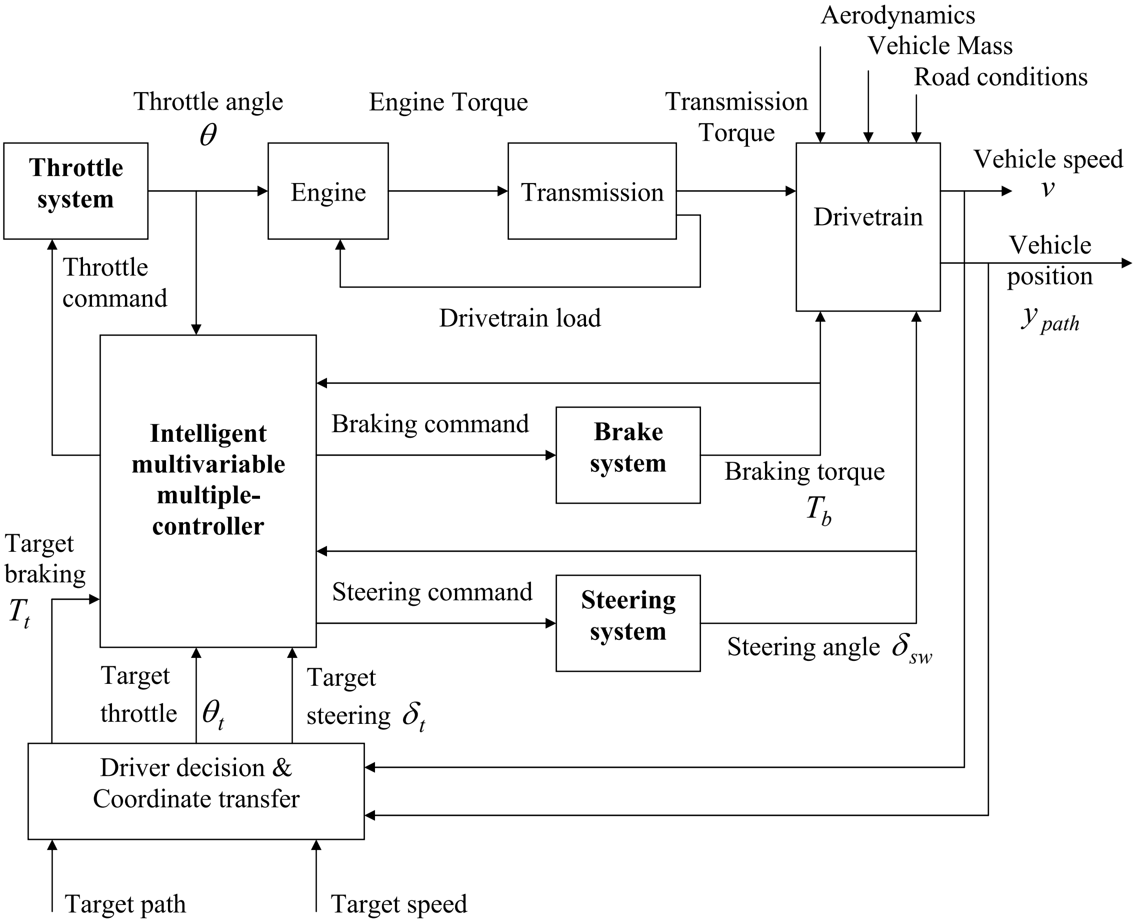


Figure (6.20): Longitudinal and lateral vehicle model incorporating the intelligent multivariable multiple-controller.

A simplified longitudinal and lateral vehicle model can be represented using the following equation [77, 138, 142, 143]:

$$\dot{v}_{x,y} = \frac{1}{m}[-c_v v^2 - c_p v - d_m + f_1(v,T_e) - G_r T_b - C_f \delta_{sw})], \qquad T_e = f_2(\theta) \tag{6.4}$$

where $v_{x,y}$ is the vehicle speed, m is the vehicle mass, c_v is the coefficient of aerodynamics drag, c_p is the coefficient of friction force, d_m is the mechanical drag, the function $f_1(v,T_e)$ is the ideal tire force which is generally measured by steady-state tests [138] and it depends mainly on the vehicle speed and the engine torque T_e. The engine torque T_e itself is a nonlinear mapping from θ to T_e. Finally, $f_2(\theta)$ is the steady state characteristics of engine and transmission systems, and in this thesis it was represented by an RBF neural network with θ as its input, and T_e as the output variable. The RBF design parameters were selected on a trial and error basis. The term $G_r T_b$ represents the braking force, where G_r is effective gear ratio from the engine to the wheel and T_b is the braking torque. The term $C_f \delta_{sw}$ is the cornering force [144], where C_f is corner stiffness coefficient and δ_{sw} is the steering angle.

The following three subsections will present the complex plant models of the throttle, braking and steering subsystems used in this thesis (as in shown Figure (6.20) above). Other model representations can be found in [145].

6.4.2 Electronic Throttle Control (ETC) Subsystem

The ETC system uses a torque motor (DC servo-motor) to regulate the throttle plate angle θ between $0 < \theta < \pi / 2$ radians (i.e., closed to wide-open-throttle) in order to adjust the inlet airflow. The servo-motor is controlled by the applied armature voltage e_a in volts, which represents the control input to the ETC system. The nonlinear model of the ETC can be presented as follows [77]:

$$\theta(t) = \frac{b_0}{1 + a_1 z^{-1} + a_2 z^{-2}} e_a(t) - c_1 \Delta P \cos^2 \theta(t) - \theta_0 , \qquad (6.5)$$

where θ is the ETC system output, θ_0 is pre-tension angle of the throttle spring, the linear process parameters $a_1, a_2, b_0,$ and c_1 are estimated using the linear sub-model of the GLM model, ΔP is the manifold pressure across the throttle plate. The nonlinear function $\Delta P \cos^2 \theta(t)$ is approximated by the RBF neural network in the GLM. The main objective of the control problem is to adjust the throttle plate angular position θ so as to maintain the desired speed v.

6.4.3 Wheel Brake Subsystem

The brake system plant model used in this work is defined as in [142]:

$$T_b(t) = \frac{\hat{b}_0}{1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + \hat{a}_3 z^{-3}} e_b(t) + d(t)\,. \tag{6.6}$$

The zero of the braking process model was experimentally found restricted to $0 \leq \hat{b}_0 < 0.5$. The first two poles of the braking process were restricted to $0.9 \leq \hat{a}_1, \hat{a}_2 < 1.1$ and the third pole (the pole of the torque sensor) was restricted to $0.6 \leq \hat{a}_3 < 0.8$. The term $d(t)$ will include the nonlinear dynamics and disturbances of the process [142]. Based on the amount of wheel slip and other factors, the controller requests a desired braking torque T_b at the wheel. To reach the requested torque, the controller controls the brake line pressure by means of a voltage control e_b at the actuator, that consists of a DC motor and a ball-screw/piston device [142].

6.4.4 Steering Wheel Subsystem

The transfer function from the front wheel steering angle δ_{sw} to the desired vehicle lateral position $f_{path}(t)$ can be computed as [79]:

$$\delta_{sw}(t) = \frac{\bar{b}_0 + \bar{b}_1 z^{-1}}{1 + \bar{a}_1 z^{-1} + \bar{a}_2 z^{-2} + \bar{a}_3 z^{-3}} e_{sw}(t) + M_d(t)\,, \tag{6.7}$$

where e_{sw} is an input voltage applied to the DC servomotor installed in the steering wheel column, $\bar{b}_0$, $\bar{b}_1$, $\bar{a}_1$, $\bar{a}_2$ and $\bar{a}_3$ are approximated using the RLS based linear sub-model in the GLM. The yaw-disturbance M_d is approximated using the RBF based nonlinear sub-model of the GLM.

6.4.5 Simulation Setup

The proposed intelligent multiple-controller framework was applied to the complex longitudinal and lateral vehicle model in order to demonstrate the effectiveness of the framework with respect to tracking desired vehicle speed and path changes and achieving the desired control performance and maintain the required speed of response, whilst penalising nonlinearities and disturbances. The complex vehicle model below was used to identify the parameters of the MIMO process:

$$[\mathbf{I} + \mathbf{A}(z^{-1})][y_1(t), y_2(t), y_3(t)]^T = [\mathbf{B}(z^{-1})][u_1(t), u_2(t), u_3(t)]^T z^{-1} + \mathbf{f}_{0,t}(.,.)\,. \tag{6.8}$$

The variables $y_1(t)$, $y_2(t)$ and $y_3(t)$ represent the output signals θ, T_b and δ_{sw} respectively, $u_1(t)$, $u_2(t)$ and $u_3(t)$ are respectively the control inputs e_a, e_b and e_{sw}.

The nonlinear dynamics $\mathbf{f}_{0,t}(.,.)$ were approximated by the MIMO RBF based nonlinear sub-model of the GLM. The parameters $\mathbf{I}$, $\mathbf{A}$ and $\mathbf{B}$ of the MIMO process above were defined as:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} a_1 & \hat{a}_1 & \bar{a}_1 \\ a_2 & \hat{a}_2 & \bar{a}_2 \\ 0 & \hat{a}_3 & \bar{a}_3 \end{bmatrix} = \begin{bmatrix} -1.7713 & -2.0632 & -1.3251 \\ 0.8107 & 1.0098 & 0.7622 \\ 0 & -0.0444 & -0.0214 \end{bmatrix}.$$

$$\mathbf{B} = \begin{bmatrix} b_0 & \hat{b}_0 & \bar{b}_0 \\ 0 & 0 & \bar{b}_1 \end{bmatrix} = \begin{bmatrix} 0.1122 & 0.1123 & 0.0909 \\ 0 & 0 & 0.0841 \end{bmatrix}$$

The vehicle parameter values are given in Table (6.11) [145].

Parameter	Value
m	$1067kg$
c_v	1.42
c_p	0.3
G_r	2.66
C_f	$-15000N/rad$

Table (6.11): Vehicle parameters.

To demonstrate performance of the intelligent multivariable multiple-controller in controlling the complex autonomous vehicle control application, two simulation experiments will be performed next. In the first experiment, the MIMO control system will work in tracking a reference signal representing a changing target vehicle speed along a longitudinal track with no lateral displacements (i.e., no steering actions required). This example will focus on the interactions between the throttle and wheel brake systems to reach the target speed. The second experiment will illustrate the performance of the MIMO control system in tracking target path along with target speed. So, longitudinal and lateral displacements will be involved in the control process.

6.4.6 Fuzzy Supervisor Setup

The application of the autonomous vehicle control has three control inputs and three system outputs. Consequently, the fuzzy sets used in the multivariable multiple-controller switching and tuning subsystems were built according to information derived from three control inputs ($u_1(t)$, $u_2(t)$ and $u_3(t)$ are respectively the control inputs e_a, e_b and e_{sw}), three system output signals ($y_1(t)$, $y_2(t)$ and $y_3(t)$ represent the output signals θ, T_b and δ_{sw} respectively), and three reference signal ($w_1(t)$, $w_2(t)$ and $w_3(t)$). Therefore, the fuzzy rules of the switching and tuning subsystems have their fuzzy parameters derived from the three control inputs and three system outputs. The following sample fuzzy rule, which belongs to the fuzzy rules given in chapter 5 at section (5.3.2.2), represents the case when any of the three output signals has high undershoot and high variance will have the corresponding controller switched to the Pole-Zero Placement controller.

IF $\zeta_{y_1}(t)$ OR $\zeta_{y_2}(t)$ OR $\zeta_{y_3}(t)$ IS Ntive-High AND $V_{y_1}(t)$ OR $V_{y_2}(t)$ OR $V_{y_3}(t)$ IS High THEN $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$ OR $C_{\eta_3}(t)$ IS Pole-Zero-Placement

Similarly, the following multivariable multiple-controller tuning fuzzy rule is a sample rule derived from the list of the general tuning logic fuzzy rules given in chapter 5 at section (5.3.3.2):

IF $C_{\eta_1}(t)$ OR $C_{\eta_2}(t)$ OR $C_{\eta_3}(t)$ IS PID AND $\zeta_{y_1}(t)$ OR $\zeta_{y_2}(t)$ OR $\zeta_{y_3}(t)$ IS NOT Norm THEN $v_{\tau_1}(t)$ OR $v_{\tau_2}(t)$ OR $v_{\tau_3}(t)$ IS Decrease

The above rule checks the overshoot in the three system output signals $y_1(t)$, $y_2(t)$ and $y_3(t)$ in order to provide tuning values for the gains v_1, v_2 or v_3 of the corresponding active PID controllers.

In the design of the fuzzy MFs for the switching and tuning subsystems for this MIMO application, the TriMF and TrapMF scalar variables a, b, c and d of the switching and tuning fuzzy parameters were experimentally adjusted in order to contain quantitative information about the autonomous vehicle model subsystems (i.e. electronic throttle, wheel brake, and steering wheel subsystems). The final fuzzy sets represent knowledge base about the vehicle model subsystems' behaviour by preserving different information including: output signals degree of overshooting, output signal variance limit, steady state error, and so on. By adjusting MFs scalar variables a, b, c and d (given in Equations 4.11 and 4.12 and Tables 6.1 and 6.2) of each vehicle model subsystem, the overlap between neighbouring MFs will differ between the different input and output fuzzy parameters.

6.4.7 Experiment One

The obtained results presented in Figures (6.21a, b and c) and (6.22a, b and c) show the performance of the intelligent multivariable multiple-controller in simultaneous controlling of the throttle subsystem and wheel braking subsystem while dealing with the nonlinearity and disturbance interactions. Figure (6.21a) shows the closed-loop system output obtained when the multiple-controller framework is used to maintain the desired vehicle speed *v*. Figure (6.21b) shows the selection scheme for the two multiple-controllers C_1 and C_2 (resulting from use of the fuzzy switching and tuning supervisor) which led to effective tracking of the desired speed changes whilst penalising excessive control action, and achieving non-overshooting and minimum variance system outputs. It is interesting to note that at the sample time 700 the brake system was activated in order to reach the new desired vehicle speed of 10m/Sec. The throttle angle was set to its minimum by the throttle controller C_1 (Figure (6.22a and b)) while the brake controller C_2 was activated to slow the vehicle system to the target low speed (Figure (6.22c and d)).

It can be seen through the multiple-controller switching in Figure (6.21b), Figures (6.22a) and (6.22c), for both the throttle and brake subsystems, the fuzzy logic based switching and tuning supervisor activates the Pole-Zero Placement controller at the points of changes in the speed reference signal. As a result of these switching decisions, the nonlinearity effects and overshooting in the throttle and brake subsystems were prevented. When the signals reach the steady state, the PID controller was automatically activated resulting in minimum variance steady states in both subsystems. Also in Figure (6.22a), at sampling time of 500 the poles of the Pole Zero placement controller, were automatically tuned online by the fuzzy supervisor. The tuning logic caused adjustment in the rising time of the throttle angle signal, which in turn affects the turning speed of the throttle plate. The zeros were also simultaneously tuned to minimize the control action (Figures (6.22b and d)).

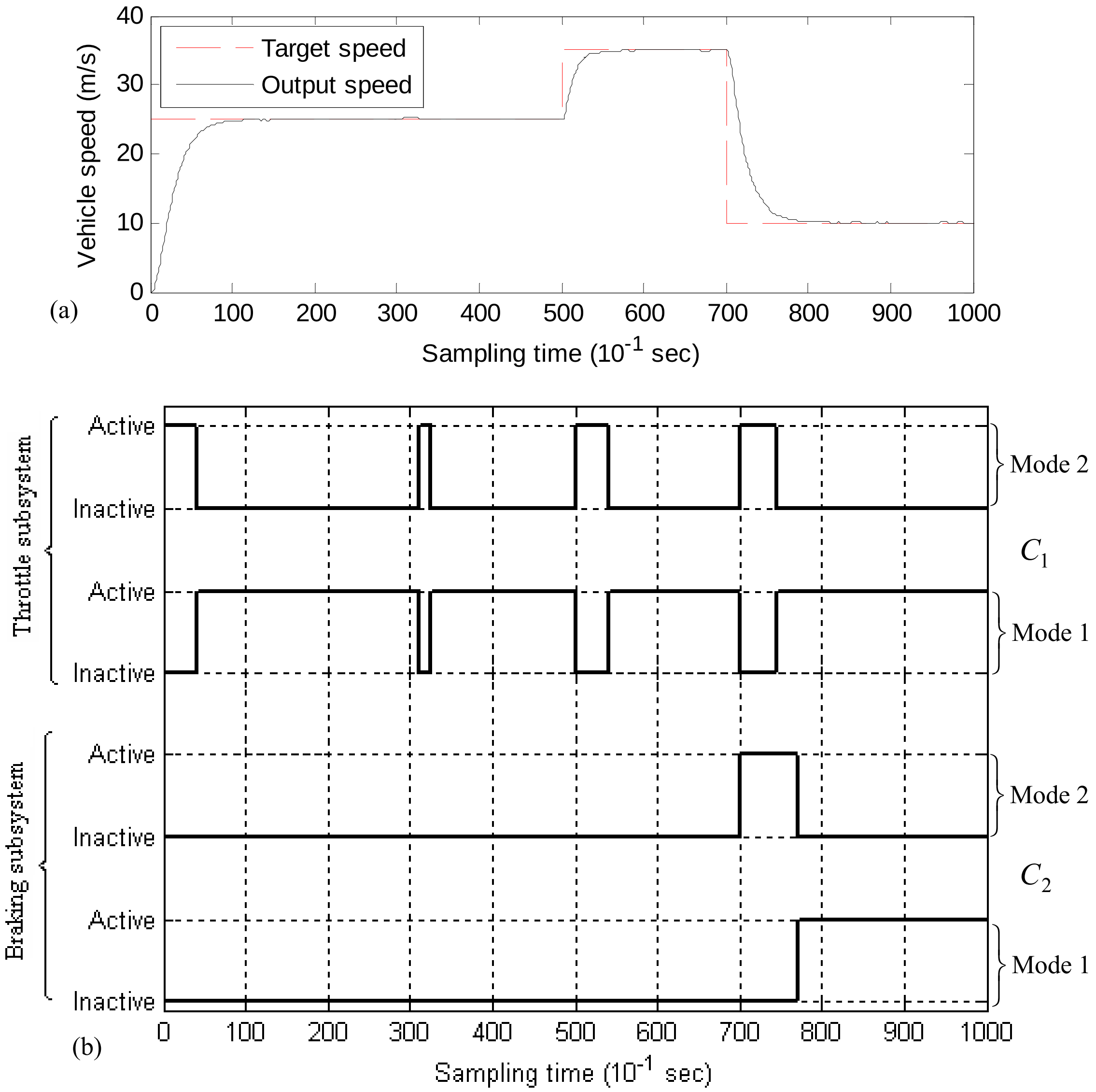


Figure (6.21): Intelligent multiple-controller in tracking target vehicle speed (I): (a) output speed trajectory, (b) multiple-controller switching scheme among throttle and wheel brake subsystems.

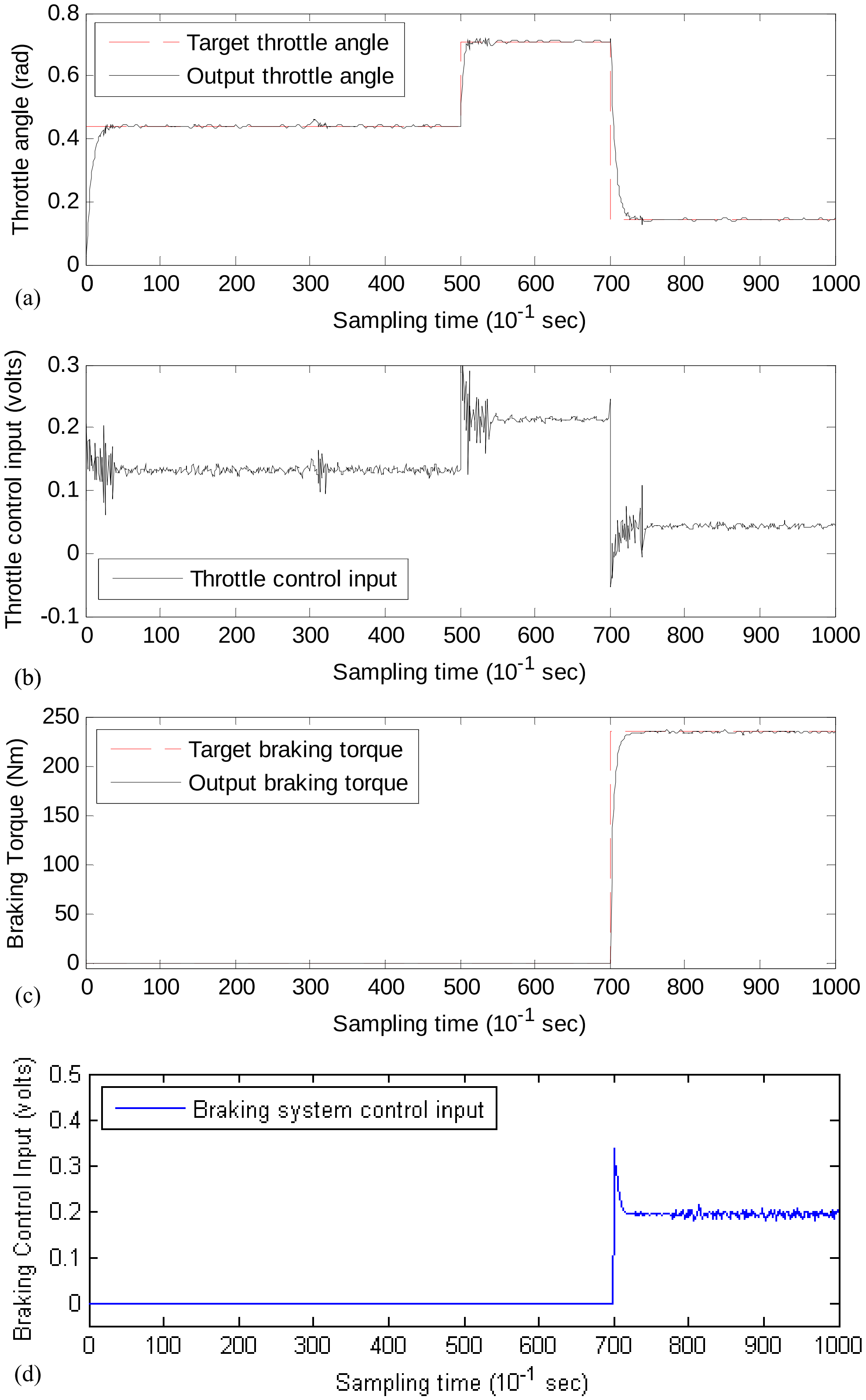


Figure (6.22): Intelligent multiple-controller in tracking target speed (II): (a) the required throttle *θ*, (b) throttle subsystem control input(c) the required braking torque, (d) braking subsystem control input.

6.4.8 Experiment Two

The complex vehicle plant model consists of three-input three-output MIMO system includes interacting nonlinearities from the throttle, braking and wheel steering subsystems, was a challenging task for the intelligent multivariable multiple-controller. Due to complex nature of this task and to simplify the simulation process, there were no random disturbances involved in this experiment. The obtained results in Figures (6.23) to (6.24) illustrate the capability of the proposed framework in producing non-overshooting and smooth steady state system output signals with minimum control input actions, which were results of proper switching and tuning actions performed by the fuzzy logic supervisor.

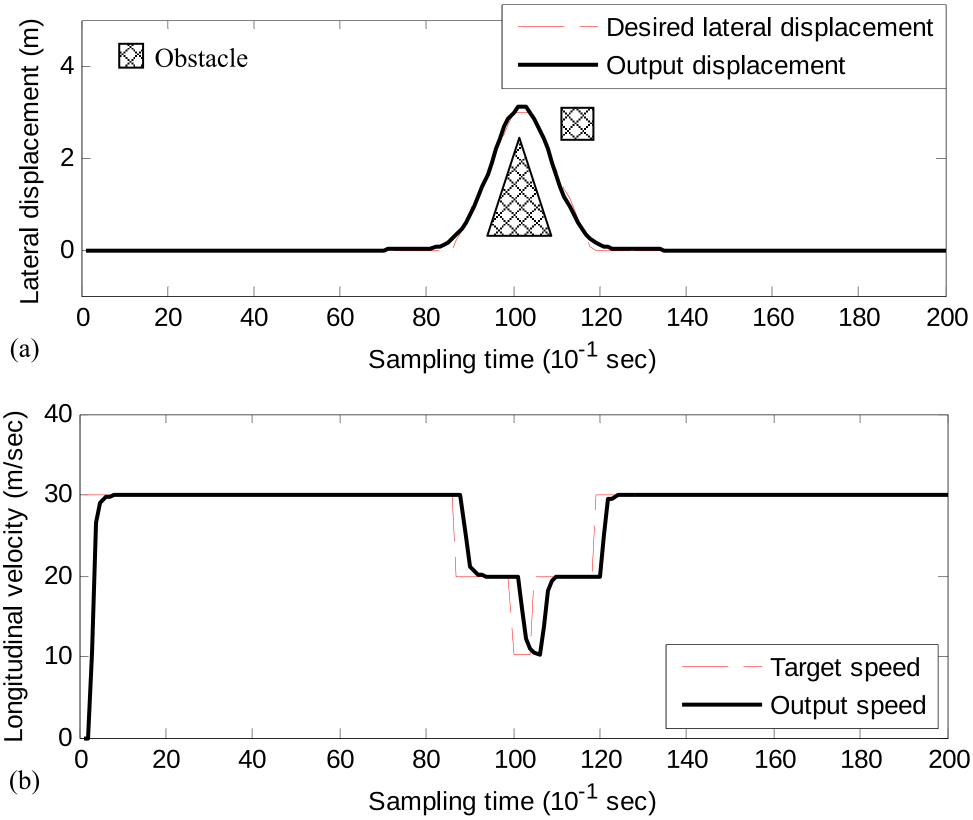


Figure (6.23): Intelligent multiple-controller in tracking target speed and target path (I): (a) tracking of the target path displacements, (b) tracking of the target speed.

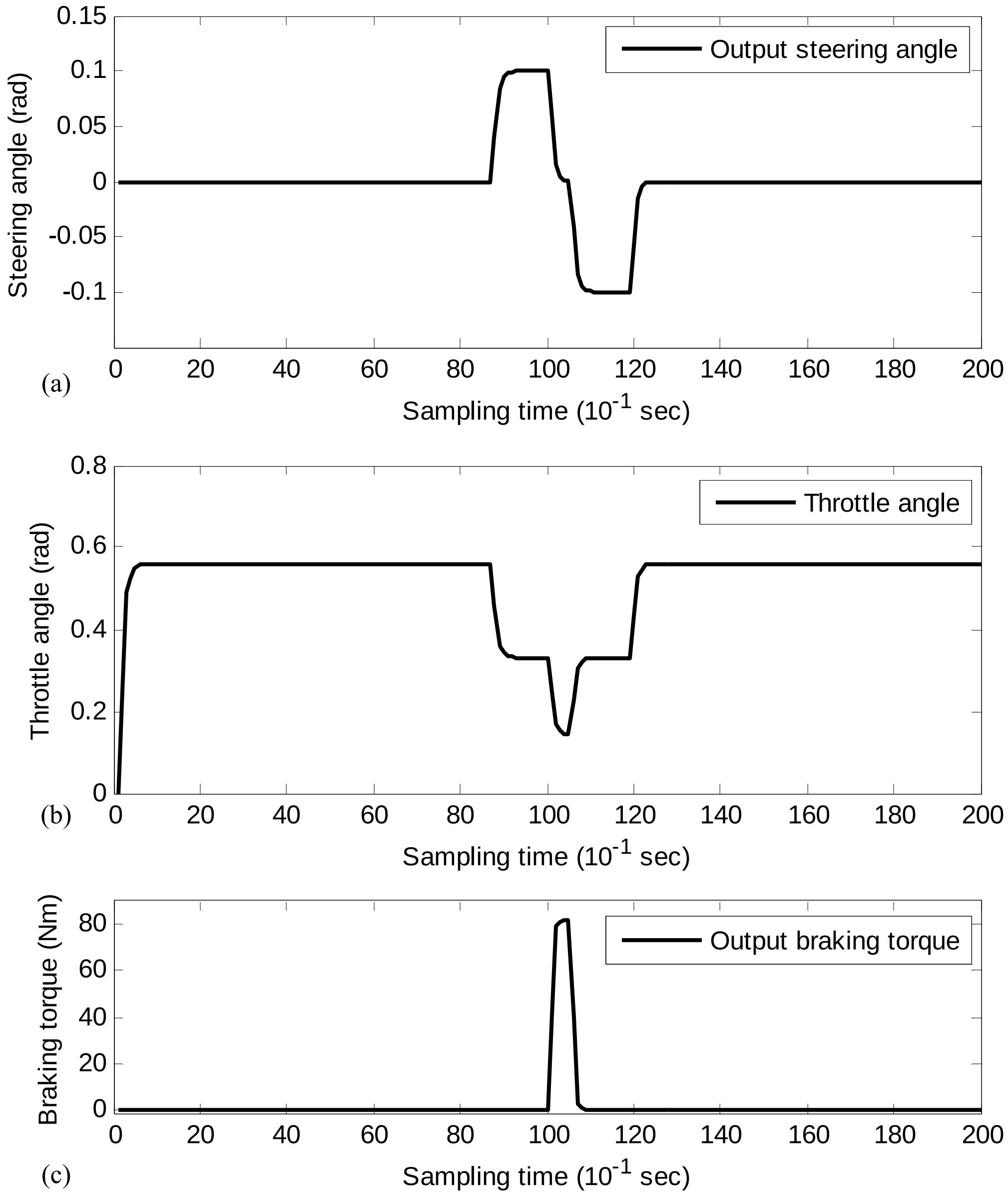


Figure (6.24): Intelligent multiple-controller in tracking target speed and target path (II): (a) maintaining the required steering angle δ_{sw}. (b) maintaining the required throttle angle θ, (c) maintaining the required braking torque T_b.

Figures (6.23) and (6.24) illustrate the results obtained by controlling the throttle, brake and steering wheel systems through the use of the longitudinal and lateral MIMO autonomous vehicle model. The simulation results demonstrate that the proposed methodology is able to follow control decisions within the desired path and speed trajectories. During the sampling times from 0 to 86 the autonomous vehicle had to follow the path with a speed of 30m/sec, then slow down to 20m/sec to follow the left side turn. This was to be followed by slowing down to a speed of 10m/s to track a sharp turn right within the target path, and then speeding up to 20m/sec and lastly attaining the target speed of 30m/sec (as shown in Figures (6.23a and b).

Figure (6.24a) shows the output steering wheel angle during the operation of tracking the target path. As can be seen in Figure (6.24b), during sampling times from 100 to 104 the throttle plate was set to its minimum in order to slow down the speed to 10m/sec. However, as this minimum throttle angle was not sufficient to reach this low target speed, simultaneously the wheel brake system was triggered to produce the required braking torque as shown in Figure (6.24c).

6.5 Summary

The chapter presented a set of experiments to demonstrate the performance of the new proposed intelligent SISO and MIMO multiple-controller frameworks given in chapters four and five respectively. The conducted simulation experiments, which were used to evaluate the proposed methodology, have been performed on benchmark control applications. Each control application can be characterised as a complex control problem, as each was modelled in terms of estimated linear parameters, approximated nonlinear dynamics and added random disturbances. To assess the behaviour of the SISO design of the intelligent multiple-controller, a SISO water vessel plant model was used. Also, for the MIMO design, a coupled water tank system was used. The simulations were ended by introducing the proposed methodology to a new challenging application which was the complex multivariable problem of autonomous vehicle control.

Generally, the experiments were aimed to achieve four goals. Firstly, to justify the choice of employing the RBF neural network based GLM instead of the conventional MLP neural network based GLM for both SISO and MIMO multiple-controllers. Secondly, to demonstrate the stable and bumpless fuzzy logic based switching actions between the conventional PID controller and the Pole-Zero Placement controller. Thirdly, to illustrate the effectiveness of the novel fuzzy logic based online tuning of the multiple-controller parameters including the PID gain of the conventional PID controller, and poles and zeros of the Pole-Zero Placement controller. The switching and tuning tasks were performed autonomously by the proposed fuzzy-logic supervisor to maintain the required closed-loop system performance. Fourthly, to examine the proposed framework on a more complex application namely autonomous vehicle control. This application has more than one actuator with the involvement of nonlinearities and interactions between its subsystems.

The obtained results from the SISO and MIMO case studies showed how the RBF neural network consistently outperformed the sigmoidal MLP neural network through stable learning and a improved approximation of high nonlinearities and sharp disturbances. Due to the online switching and tuning, the fuzzy logic supervisor successfully performed the minimum switching actions required for producing control system signals which preserve low overshooting and minimum variance system outputs, and penalised control inputs. Furthermore, tuning the parameters of the multiple-controller showed satisfactory outcomes in adjusting the rise and fall times of the system output signal, penalising control actions, minimising the steady state errors and minimising output variance.

The output results produced from applying the intelligent multiple-controller framework to the new application of autonomous vehicle control had provided promising outcomes. It was shown that this proposed interdisciplinary control strategy managed to control the longitudinal and lateral complex vehicle model to track target speed and path trajectories, which was achieved through the simultaneous control of the throttle, wheel brake and steering complex subsystems.

The next chapter presents this thesis's concluding remarks together with some recommendations for future work.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Motivated by the demand for the use of more generalised plant modelling methods and sophisticated controllers, which is due to the increasing complexity of dynamical systems coupled with the increasing demands in closed loop performance specification, the research work reported in this thesis was directed towards integrating control engineering tools with soft computing techniques in order to achieve the development of a new intelligent framework for modelling and control of complex systems. The proposed intelligent multiple-controller methodology for SISO and MIMO complex systems employed an RBF neural network based generalised learning model (GLM) for modelling complex plants, and suggested a high level fuzzy logic supervisor for autonomous online tuning and switching between a conventional PID controller and a Pole-Zero Placement controller.

There were two review chapters included in this thesis. Chapter two aimed at highlighting the importance of intelligent control and introducing the issue of complex systems and how they can be dealt with using neural networks and fuzzy logic. Chapter three discussed a general architecture for a feedback system employing a family of controllers for controlling a complex mechatronic plant system which is modelled as a physical process operating in a limited set of operating regimes. The discussion focused on the recently developed multiple-controller framework which incorporated an MLP neural network based GLM for modelling and control of complex systems. The controllers employed in this design were built on the concept of adaptive generalised minimum variance control.

The contributions of this thesis were presented in chapters four, five and six. Chapter four presented the new intelligent multiple-controller framework incorporating a Mamdani fuzzy logic system designed to supervise the multiple-controller switching and tuning processes in order to control complex SISO systems. The complex plant was modelled using an improved GLM based on an RBF neural network. The proposed intelligent multiple-controller operates adaptively for tracking a desired reference signal, achieving the desired output signal performance and penalising excessive control actions, in response to the current performance of the control systems as assessed by a behaviour recogniser. In chapter five, due to the importance of supervisory control and multivariable controllers, an autonomous intelligent multivariable multiple-controller framework for the control of complex MIMO plants was presented. The switching and tuning decisions are based on the specific required performance criteria and continual monitoring of the effectiveness of each multivariable controller in achieving these criteria in response to set-point changes, nonlinear dynamics and added disturbances in the complex MIMO plant. The chapter was concluded by proving the derivation of the stability analysis of the proposed intelligent multivariable multiple-controller framework. Finally, chapter 6 presented the simulation experiments used to assess the behaviour of the SISO and MIMO designs of the intelligent multiple-controller. The simulations concluded by applying the proposed methodology to a new challenging application, namely the complex multivariable problem of autonomous vehicle control.

On the basis of the theoretical discussions and the simulation experiments presented in the previous chapters, the following conclusions can be drawn:

- The GLM based on the RBF neural networks consistently outperformed the sigmoidal MLP neural network based GLM. The RBF neural network's stable learning and its improved approximation of high nonlinearities and sharp disturbances added an advantage to the proposed intelligent multiple-controller. The GLM approach for modelling complex systems simplified the design of control systems for both SISO and MIMO complex plants. Generally speaking, modelling complex systems with nonlinearities, disturbances and uncertainties is a difficult task and traditional solutions involve linearization of system dynamics to establish control techniques imported from linear control systems theory. On the other hand, implementing globally stable nonlinear controllers with adequate performance during all stages of the control process is not an easy task. By using nonlinear PID structure based minimum variance control designs, such as the multiple-controller used in this thesis, coupled with a learning nonlinear plant modelling approach, supported by online controllers switching and tuning, we can successfully control a general class of complex discrete-time systems.

- The proposed fuzzy logic supervisor successfully performed the desired switching actions required for producing control system signals which preserve low overshooting and minimum variance system outputs, and penalised control inputs. Moreover, the selection decisions experienced no conflict in the choice between the two candidate controllers. These switching advantages can be dedicated to the multiple system assessment factors (such as: degree of overshooting; signal variance; settling time; steady state error; …etc.) used in the fuzzy rules and supplied by the supervisor’s behaviour recogniser subsystem. That was also due to the facility of online tuning for the multiple-controller parameters, which had its fuzzy rules given higher priority than the switching fuzzy rules. In this manner, the active controller had the chance for its tuneable parameters to be adjusted before it could be deactivated in order to switch to another candidate controller. Furthermore, tuning the parameters of the multiple-controller showed satisfactory outcomes in adjusting the rise and fall times of the system output signal, penalising control actions, minimising the steady state errors and minimising output variance. Tuning of the PID gain, poles and zeros of the multiple-controller is a novel use of fuzzy logic in online tuning of control parameters, in view of the fact that fuzzy supervisor has conventionally only been employed for tuning PID controllers [4].

- The simulation results demonstrated that the proposed intelligent framework is able to compensate complex system deficiencies caused by subsystems' interactions, nonlinearities and disturbances. Novel application of the intelligent multiple-controller framework to the autonomous vehicle control problem showed promising results. It was shown that this proposed interdisciplinary control strategy managed to control the longitudinal and lateral complex vehicle model to track target speed and path trajectories, which was achieved through the simultaneous control of the throttle, wheel brake and steering complex subsystems.

7.2 Implementation Challenges

In the fuzzy logic based supervisory system proposed for tuning and switching between the multiple-controllers for each of the SISO and MIMO applications, the selection of the supervisor structure involved a series of over two dozens trial-and-error experiments to set up the final structure parameters, which included the following choices:

(a) Input and output variables for the fuzzy switching and tuning subsystems: Details of the input and output fuzzy variables for the SISO and MIMO supervisor systems are given in sections (4.3) and (5.3) respectively.

(b) Number and type of membership functions for the fuzzy variables: The membership functions used for the fuzzy values of fuzzy variables are selected based on experimental observations for each of the SISO and MIMO applications illustrated in chapter 6. From the point of view of simplicity and computational complexity, the fuzzy switching and tuning subsystems' variables were represented by TriMF and TrapMF membership functions with not more than two membership functions overlapping. During the optimization process of the MFs of the fuzzy variables, the MFs scalar parameters a, b, c and d of Equations (4.11 and 4.12) were experimentally adjusted in order to adequately represent quantitative information about the SISO and MIMO applications. The final fuzzy sets represent the knowledge base about the physical system behaviour by preserving information such as: output signal(s) degree of overshooting, output signal(s) variance limit, and steady state error range(s). Details of the fuzzy logic sets are given in Tables (6.1 and 6.2) chapter 6.

(c) Rule base: The rules of the fuzzy switching and tuning supervisor were designed based on the SISO and MIMO controllers' performance assessment criteria given in section (4.2.1) in chapter 4, and on the experimental consideration of the influence of each of the input and output fuzzy variables in the switching and tuning logic subsystems. The final rule bases were implemented with the minimum possible number of rules such that the switching logic and the tuning logic employed 5 rules and 12 rules respectively.

The employed membership functions and rule bases play a crucial role in the final performance of the fuzzy supervisory system. Therefore, selection of the appropriate MFs and fuzzy rules is an important design problem. Depending on the designer's preference or experience, fuzzy membership functions can have different shapes and sizes, fuzzy rules can have different orientations and number of rules, which can in turn lead to an improvement in the system's performance.

Taking into account the real-time implementation constraints, such as minimizing the amount of memory used and the time that it takes to compute the system outputs using the derived control inputs, the fuzzy logic based switching and tuning supervisor is designed with a minimum number of fuzzy rules with minimum input and output parameters. From the point of view of computational complexity, a maximum of three membership functions is considered for both the inputs and outputs with not more than two membership functions overlapping. In order to improve the computation time, the triangular and trapezoidal membership functions are used, which have the advantages of simplicity and require minimum rules. To reduce the memory requirements and to enable online operation (switching and tuning), the fuzzy supervisor is designed to compute the rule-base at each time instant rather than using a stored one. Implementation prospects could be improved by using a state-of-the-art microprocessor or signal processing chip. An alternative would be to investigate the advantages and disadvantages of using a fuzzy processor (i.e., a processor designed specifically for implementing fuzzy controllers) as in [4].

The field of supervisory control is a combination between decision-making and real-time control. The decision-making aspect corresponds to "autonomous" (or independent) supervision and the real-time control corresponds to plant control (or execution). The developers of autonomous intelligent control for real-time systems cannot work on autonomy and computer processing separately from working on the plant system mechanics and specifications. Therefore, integration of these two areas into one physical system presents a significant challenge in itself. Not only does the computer equipment need to be able to physically withstand the operational environment of the plant under control, but it also needs to appropriately connect the algorithms to the incoming sensor data and decide which sensor information is needed in the first place.

In practical implementations of control structures for trajectory control, one difficulty in achieving accurate trajectory tracking is the existence of observation disturbances and plant nonlinear dynamics, which could corrupt the parameters involved in the design of the multiple-controller. To overcome the effects of this situation, which is very likely to be encountered in practice, research is still required in accurate estimation of plant parameters, as well as precise approximation of disturbances and nonlinearities, especially in the area of transfer learning (*i.e.* generalising from a previous example to a novel situation). Full understanding of this excellent generalisation ability of humans and then having it accomplished in computers would improve learning times and high costs in system modelling, that are often the cause of brittle performance. To achieve these goals, the future work section, next, proposes other forms of neurobiologically motivated learning and control perspectives, along with alternative approaches for online neural network construction to further improve the approximation of the nonlinearity and disturbances.

In order to make a switching decision among various controllers and tuning them based on the required specifications, adequate performance criteria have to be defined. The heuristics and quantitative performance measures of each controller have to be evaluated based on a number of factors, such as energy consumption (magnitude of control actions), output variance and meeting user specifications. The user specifications, which are usually given in terms of required overshoot, rise time and settling time, can give a good measure about which control algorithm will give the best performance as compared to others. Therefore, it is important to translate these specifications accurately into supervision (decision-making) parameters represented in the fuzzy sets to guarantee that each controller meets the desired specifications.

7.3 Future Work Recommendations

Now that it has been established that the proposed intelligent multiple-controller can be functionally useful, further work is required to understand the extent of this usefulness. Some proposals in this regard are presented next.

- One main drawback of the new multiple-controller framework for MIMO and SISO cases is that Diophantine equations need to be solved to obtain the control algorithms. This limitation may be overcome by using an implicit Pole-Zero Placement controller as an additional control mode option within the new multiple-controller framework. Another novel option could be a fuzzy logic controller designed on the basis of a minimum variance control scheme. The mathematical structure of fuzzy logic world also offers a promising implementable solution.
- The structure of the proposed intelligent multiple-controller control law, for both SISO and MIMO cases, is built on the basis of generalised minimum variance control with a PID structure. Therefore, the interesting relationship between the control law polynomials (A , P_d and F) and the design of the three term PID controller as a PI, PD or PID controller provides a possibility of adaptively changing the structure of the conventional PID controller (mode 1) into PI, PD or PID according to the system behaviour and the required performance. This means additional controllers would be involved in the multiple-controller framework, particularly PID controllers which are the most commonly applied algorithms in the control industry.

- Within the above suggested control strategies various parameters would need to be chosen based on the required control system performance, which could vary in time according to system configurations and nonlinearities. Nonetheless, the multiple-controller switching activity and parameters tuning could be further developed through fuzzy logic based on the error signals and the first difference of the error signals as in [72].

- Modelling the complex plant with the GLM could be further investigated to ensure whether linear sub-model plus neural network based nonlinear sub-models are adequate to model dynamical systems operating in "critical environments" where, for example, the safety of a crew (e.g., in an aircraft/spacecraft), or environmental issues (e.g., from nuclear power plants or process control) are of concern. Hence, it is both possible, and of significant importance to introduce robust mathematical modelling and analysis techniques to be used in the verification and certification of the behaviour of intelligent control systems. In complex plants or processes of this kind, nonlinearities are present due to the large range of operating conditions and power levels experienced during a typical mission. Also, such systems are restricted due to mechanical, aerodynamic, thermal, and flow limitations. RBF based GLM could be useful because it can explicitly handle the nonlinearities, and both input and state constraints of many variables in a single control formulation. To improve the online approximation capabilities of the RBF NN used in the GLM nonlinear sub-model, the current offline trial-and-error approach used for the design of the network structure could be replaced with the online allocation and tuning methods for RBF units which have been proposed by Platt in [153]. An alternative approach could be that the nonlinear sub-model in the

GLM could be represented by fuzzy logic or a neuro-fuzzy sub-model in order to further improve the approximation of the nonlinearity and disturbances. Alternatively, instead of representing the GLM with two sub-models, it could be interesting to represent the GLM with a single model based on a type-2 Fuzzy Logic System (FLS). The type-2 FLSs have started to emerge as a promising control mechanism for autonomous mobile applications navigating in real world environments. This is because such applications need control mechanisms such as type-2 FLSs which can handle large amounts of uncertainties present in real world environments [146, 147, 148]. There are various types of learning models that could be integrated into the framework to produce an intelligent multiple model switching and tuning framework, in which a controller could be associated with its corresponding optimised learning model.

- As discussed in this thesis, intelligent control is a discipline in which control algorithms are developed by emulating certain characteristics of intelligent biological systems. It is quickly emerging as a technology that may open avenues for significant advances in many areas. In fact fuelled by advancements in computing technology, it has already achieved some very exciting and promising results. Other forms of neurobiologically motivated learning and control perspectives could be incorporated into the framework, such as reward based learning (also known as re-enforcement learning), multiple models, multiple agents and Genetic Algorithms. These techniques can support complex single and multiple autonomous agents (vehicle, robot, or software entity) and make use of the plethora of environmental information they receive to guide purposive and useful behaviour.

References

[1] C. J. Harris, C. G. Moore and M. Brown, Intelligent Control: aspects of fuzzy logic and neural nets, (World Scientific Publishing, Singapore, 1993).

[2] L. Chen and K. Narendra, Intelligent control using multiple neural networks, *International Journal of Adaptive Control and Signal Processing*, 17, 417-430, 2003.

[3] T. Schön, *Identification for predictive control – multiple model approach*, Master's Thesis, Division of Automatic Control, Linköping University, Sweden, 2001.

[4] K. M. Passino and S. Yurkovich, Fuzzy Control, (Addison Wesley Longman, Inc., UAS, 1998).

[5] A. J. van Breemen, *Agent-Based Multi-Controller Systems*, PhD Thesis, University of Twente, Netherlands, 2001.

[6] P. J. Antsaklis and K. M. Passino, An Introduction to Intelligent and Autonomous Control, (Kluwer Academic Publishers, Norwell, MA, 1993).

[7] K. J. Aström, J. J. Anton, and K. E. Arzen, Expert control, *Automatica*, 22(3), 277–286, March 1986.

[8] A. Breemen, T. Vries, An Agent-Based Framework for Designing Multiple Controller Systems, *Proc. Fifth Inter. Conf. The practical Applications of Intelligent Agents & Multi-Agents Technology*, Manchester, UK, 219-235, 10-11 April 2000.

[9] S. J. Corfield, R. J. Fraser and C. J. Harris, Architecture for Real Time Intelligent Control of Autonomous Vehicles, *IEE Journal Computing and Control Engineering*. 2(6), 254-262, 1991.

[10] G. N. Saridis, Analytical Formulation of the Principal of Increasing Precision with Decreasing Intelligence of Intelligent Machines, *Automata*, 25(3), 461-467, March 1989.

[11] B. O. Anderson, Failures of Adaptive Control Theory and their Resolution, *Communications in Information and Systems*, 5(1), 1-20, 2005.

[12] K. Passino, Biomimcry for Optimization, Control, and Automation, (Springer, 2005).

[13] A. E. Ruano, Intelligent Control Systems using Computational Intelligence Techniques, (The Institution of Electrical Engineering, London, UK, 2005).

[14] J. A. K. Suykens, J. P. L. Vandewalle and B. L. R. De Moor, Artificial Neural Networks for Modelling and Control of Non-Linear Systems, (Kluwer Academic Publishers, Dordrecht, Nethrlands, 1996).

[15] J. A. Farrell and M. M. Polycarpou, Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches, (John Wiley & Sons, Inc, New Jersey, USA, 2006).

[16] Q. Zhu, Z. Ma and K. Warwick, Neural network enhanced generalised minimum variance self-tuning controller for nonlinear discrete-time systems, *IEE Proc. Control Theory and Applications*, 146*,* 319-326, 1999.

[17] A. S. Zayed, Novel Linear and Non-linear Minimum Variance Techniques for Adaptive Control Engineering, Ph.D. Thesis, Department of Computing Science & Math, University of Stirling, Stirling, UK, 2005.

[18] S. Haykin, Neural Networks: A Comprehensive Foundation, (Macmillan, USA 1994).

[19] Q. Zhu and K. Warwick, A neural network enhanced generalized minimum variance self-tuning Proportional, Integral and Derivative control algorithm for

complex dynamic systems, Journal of Systems and Control Engineering, *Proceedings of the Institution of Mechanical Engineers Part I*, 126 (3), 265-273, 2002.

[20] M. L. Bujorianu, Stochastic Hybrid Systems: Modelling and Varification, Ph.D. Thesis, Department of Computing Science & Math, University of Stirling, Stirling, UK, 2005.

[21] K. S. Narendra and K. Parathasathy, Identification and Control of Dynamical Systems using Neural Networks, *IEEE Trans. on Neural Networks*, 1 (1), 4-27, 1990.

[22] S. Chen and S. A. Billing, Neural Networks for Nonlinear Dynamic Systems Modelling and Identification, *Int. Journal of. Control*, 2, 319-346, 1992.

[23] J. W. Park, R. G. Harley and G. K. Venayagamoorthy, Comparison of MLP and RBF Neural Networks using Deviation Signals for On-Line Identification of a Synchronous Generator, *IEEE Power Engineering Winter Meeting, New York, USA*, 1, 274-279, 2002.

[24] M. Nørgaard, O. Ravn, N. K. Poulsen and L. K. Hansen, Neural Networks for Modelling and Control of Dynamic Systems, (Springer-Verlag London, 2000).

[25] E. A. Jonckheere, S. R. Yu and C. C. Chien, Gain Scheduling for Lateral Motion of Propulsion Controlled Aircraft using Neural Networks, *Proceedings of 1997 American Control Conference*, Albuquerque, USA, 3321-3325, 1997.

[26] J. T. Jeng and T. T. Lee, A Neural Gain Scheduling Network Controller for Nonholonomic Systems, *IEEE Trans. on Systems, Man and Cybernetics-Part A*, 29 (6), 654-661, 1999.

[27] D. T. Pham and S. J. Oh, Identification of Plant Inverse Dynamics using Neural Networks, *Artificial Intelligence in Engineering*, 13 (3), 309-320, 1999.

[28] J. B. D. Cabrera and K. S. Narendra, Issues in the Application of Neural Networks for Tracking based on Inverse Control, *IEEE Trans. on Automatic Control*, 44 (11), 2007-2027, 1999.

[29] G. L. Plett, Adaptive Inverse Control of Linear and Nonlinear Systems using Dynamic Neural Networks, *IEEE Trans. on Neural Networks*, 14 (2), 360-376, 2003.

[30] K. Hunt and D. Sbarbaro, Neural Networks for Nonlinear Internal Model Control, *IEEE Proceedings on Control Theory and Applications*, 138 (5), 431-438, 1991.

[31] A. Aoyama and V. Venkatasubramanian, Internal Model Control framework for the Modelling and Control of a bioreactor, *Engineering Applications of Artificial Intelligence*, 8 (6), 689-701, 1995.

[32] I. Rivals and L. Personnaz, Nonlinear Internal Model Control using Neural Networks: Applications to Processes with Delay and Design Issues, *IEEE Trans. on Neural Networks*, 11 (1), 80-90, 2000.

[33] P.M. Mills, A. Y. Zomaya and M. O. Tade, Adaptive Control using Neural Networks, *International Journal of Control*, 60 (6), 1163-1192, 1994.

[34] A. E. Ruano, P. J. Fleming and D. I. Jones, A Connectionist Approach to PID Autotuning, *IEE Proceedings on Control Theory and Applications*, 139 (3), 279-285, 1992.

[35] M. J. Lima and A. E. Ruano, Neuro-Genetic PID Autotuning: Time Invariant Case, *IMACS Journal of Mathematics and Computers in Simulation*, 51, 287-300, 2000.

[36] L. Giovanini, A. W. Ordys, M. J. Grimble, Adaptive Predictive Control using Multiple Models, Switching and Tuning, *International Journal of Control, Automation, and Systems*, 4 (6), 669-681, December 2006.

[37] A. Zayed, A. Hussain and R. Abdullah, A novel multiple-controller incorporating a Radial Basis Function neural network based generalized learning model. *Neurocomputing (Elsevier Science B.V.)*, 69 (16-18), 1868-1881, 2006.

[38] A. Zayed and A. Hussain, Stability Analysis of a New pole-zero placement controller Neural Networks, Proc. *7th IEEE international multi-topic Conference (INMIC'2001), Lahore, 28-31 Dec 2003*, 267-271.

[39] A. Zayed and A. Hussain, A New Non-linear Multi-variable Multiple-Controller incorporating a Neural Network Learning Sub-Model, *Proc. International Symposium on Brain Inspired Cognitive Systems (BICS2004)*, Stirling-Soctlan, 29 Aug - 1 Sep 2004.

[40] M. Tokuda and T. Yamamoto, A neural-Net Based Controller supplementing a Multiloop PID Control System, *IEICE Trans. Fundamentals*, E85-A (1), 256-261, 2002.

[41] R. Yusof, S. Omatu and M. Khalid, Self-tuning PID control: a multivariable derivation and application, *Automatica*, 30, 1975-1981, 1994.

[42] A. Zayed, Minimum Variance Based Adaptive PID Control Design, M.Phil Thesis, Industrial Control Centre, University of Strathclyde, Glasgow, U.K., 1997.

[43] R. Yusof and S. Omatu, A multivariable self-tuning PID controller, *Int. J. Control*. 57, 1387-1403, 1993.

[44] Z. Ma, A. Jutan and V. Baji, Non-linear Self-tuning Controller For Hammerstein Plants with application to a pressure tank, *IJCSS*, 1(2), 221-230, 2000.

[45] D. Prager and P. Wellstead, Multivariable pole-placement Self-tuning regulators, *Proc. Inst. Electr. Engineering*, Part D, 128, 9-18, 1980.

[46] A. Zayed, A. Hussain and L. Smith, A New Multivariable Generalised Minimum-variance Stochastic Self-tuning Controller with Pole-zero Placement, *Int. J. of Control & Intelligent Systems*, 32 (1), 35-44, 2004.

[47] R. Sirisena and F. C. Teng, Multivariable Pole-Zero Placement Self-Tuning Controller*, Int. J. Systems Sci*, 17(2), 345-352, 1986.

[48] R. Sanner and Slotine J., Gaussian Networks for Direct Adaptive Control, IEEE Trans. on Neural Netwroks, 3 (6) (1992) 837-863.

[49] Y. Li, S. Qiang, X. Zhuang and O. Kaynak, Robust and Adaptive Backstepping Control for Nonlinear Systems Using RBF Neural Networks, *IEEE Trans. on Neural Networks*, 15(3), 693-701, (2004).

[50] J. Hespanha, D. Liberzon, A. Morse, B. Anderson, T. Brinsmead and D. Bruyne, Multiple model adaptive control. Part 2: switching, *International Journal of Robust and Nonlinear Control*, 11, 479-496, 2001.

[51] C. Pous, J. Colomer, J. Melendez and J. de la Rosa, Fuzzy identification for fault isolation. Application to analog circuits diagnosis, *Artificial Intelligence Research & Development*, 1, 409-420, 2003.

[52] L. Mei-Qin, Stability analysis of neutral-type nonlinear delayed systems: An LMI approach, *Journal of Zhejiang University SCIENCE*, A 7 (2), 237-244, 2006.

[53] D. Cao and P. He , Stability criteria of linear neutral systems with a single delay, *Applied Mathematics and Computation*, 148(1), 135-143, 2004.

[54] K. Najim, Control of Continuous Linear Systems (ISTE Ltd 2006).

[55] P. Lancaster, Theory of Matrices (Academic Press, Inc 1969).

[56] P. Lancaster and M. Tismenetsky, The Theory of Matrices (Academic Press, Orlando, FL, 1985).

[57] K. Narendra and C. Xiang, Adaptive Control of Discrete-Time Systems Using Multiple Models, *IEEE Trans. on Auto. Control*, 45 (9), 1669-1686, 2000.

[58] M. Thathachar and P. Viswanath, On the Stability of Fuzzy Systems, *IEEE Trans. on Fuzzy Systems*, 5 (1), 145-151, 1997.

[59] M. Farsi, K. Karam and H. Abdalla, Intelligent multi-controller assessment using fuzzy logic, *Journal of Fuzzy Sets and Systems*, vol. 79 (1), 25-41, 1996.

[60] K. J. Aström, P. Albertos, M. Blanke, A. Isidori, W. Schaufelberger and R. Sanz, Control of Complex Systems, (London, Springer-Verlag, UK, 2001).

[61] R. J. Antsaklis, Defining Intelligent Control, *IEEE Control Systems Magazine*, June 1994.

[62] A. L. Fradkov, I. V. Miroshnik and V. O. Nikiforov, Nonlinear and Adaptive Control of Complex Systems, (Kluwer Academic Publishers, Dordrecht, 1999).

[63] R. Gomez, E. K. Najim and E. Ikonen, Stochastic learning control for non-linear systems, *Intern. Joint Conf. on Neural Networks*, Honolulu, Hawaii, U.S.A, 12-17, May 2002.

[64] K. Narendra and M. Thathachar, Learning Automata, An Introduction, (Prentice-Hall, 1989).

[65] W. Ya and A Poznyak, Indirect Adaptive Control via Parallel Dynamic Neural Networks, *IEE Proc. Contol Theory and Applications*, 146(1), 25-30, 1999.

[66] L. Chen and K. S. Narendra, Intelligent Control Using Neural Networks and Multiple Models, *Proceedings of the 41st IEEE Conference on Decision and Control*, 2, 1357- 1362, USA, 10-13 Dec. 2002.

[67] J. M. Evans, E. R. Messina, James S. Albus, Knowledge Engineering for Real Time Intelligent Control, *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, Vancouver Canada, 27-30 October, 2002.

[68] S. S. Ge and J. Zhang, Neural-Network Control of Nonaffine Nonlinear System with Zero Dynamics by State and Output Feedback, *IEEE Trans. on Neural Networks*, 14 (4), 900-918, 2003.

[69] C. Y. Lee, Adaptive Control of a Class of Nonlinear Systems Using Multiple Parameter Models, *International Journal of Control, Automation, and Systems*, 4 (4), 428-437, August 2006.

[70] A. Zayed, A. Hussain, M. Grimble, A Non-linear PID-based Multiple Controller incorporating a Multi-Layered Neural Network Learning Sub-model, *International Journal of Control & Intelligent Systems, IASTED / ACTA Press*, 34 (3.201-1499), 2006.

[71] M. G. Papoutsidakis, G. Chamilothoris, F. Dailami, N. Larsen and A. Pipe, Accurate Control of a Pneumatic System using an Innovative Fuzzy Gain-Scheduling Pattern, *Trans. On Eng.*, *Computing and Technology*, 8, 189-192, 2005.

[72] Z. Y. Zhao, M. Tomizuka and S. Isaka, Fuzzy Gain Scheduling of PID Controllers, *IEEE Trans. on Systems*, *Man. and Cybernetics*, 23 (5), 1392-1398, 1993.

[73] K. S. Narendra and J. Balakrishnan, Adaptive Control Using Multiple Models, *IEEE Trans. On Automatic Control*, 42 (2), 171-187, 1997.

[74] K. J. Aström and H. Hagglund, PID controllers: Theory, design, and tuning, (Research Triangle Park, NC: Instrument Society of America, 1995).

[75] L. Ren, G. W. Irwin and D. Flynn, Nonlinear Identification and Control of a Turbogenerator – An On-line Scheduled Multiple Model/Controller Approach, *IEEE Trans. On Energy Conversion*, 20 (1), 237-245, 2005.

[76] J. Connelly, W. S. Hong, R. B. Jr. Mahoney, D. A. Sparrow, Current challenges in autonomous vehicle development, *Unmanned Systems Technology VIII, Proceedings of the SPIE*, 6230, 62300D, 2006.

[77] R. Conatser, J. Wagner, S. Ganta, I. Walker, Diagnosis of Automotive Electronic Throttle Control Systems, *Control Engineering Practice*, 12, 23-30, 2004.

[78] J. Naranjo, C. Gonzalez, J. Reviejo, R. Garcia, T. Pedro, Adaptive Fuzzy Control for Inter-Vehicle Gap Keeping, *IEEE Trans. on Intelligent Transportation Systems*, 4 (3), 132-142, 2003.

[79] B. Güvenç, T. Bünte, D. Odenthal, L. Güvenç, Robust Two Degree-of-Freedom Vehicle Steering Controller Design, *IEEE Trans. Control Systems Tech.*, 12 (4), 627-636, 2004.

[80] M. Brown and C. Harris, Neurofuzzy Adaptive Modelling and Control, (Prentice Hall, New York, 1994).

[81] J. Jang, C. Sun and E. Mizutani, Neuro-fuzzy and Soft-computing; A Computational Approach to Learning and Machine Intelligence, (Prentice Hall, Upper Saddle River, NJ, 1997).

[82] C. Chang and K. Shyu, A self-tuning Fuzzy Filtered-U Algorithm for the Application of Active Noise Cancellation, *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications*, 49 (9), 1325-1333, 2002.

[83] M. Abdul-Mannan, T. Murata and J. Tamura, A Fuzzy-Logic Based Self-Tuning PI Controller for High-Performance Vector Controlled Induction Motor Drive, *Electronic Power Components and Systems*, 34 (4), 471-281, 2006.

[84] M. A. Brdys and J. J. Littler, Fuzzy Logic Gain Scheduling for Nonlinear Servo Tracking, *Int. Journal of Applied Math. and Computer Sci.*, 12 (2), 209-219, 2002.

[85] J. G. Ziegler and N. B. Nichols, Optimum Setting for Automatic Controller, *ASME trans.*, 64 (11), 756-768, 1942.

[86] L. Yao and C.-C. Lin, Design of Gain Scheduled Fuzzy PID controller, *Trans. on Engineering, Computing and Technology*, 1, 432-436, 2004.

[87] C. Jiang, Y. Ma and C. Wang, PID Controller Parameters Optimization of Hydro-Turbine Governing Systems using Deterministic-Chaotic-Mutation Evolutionary Programming (DCMEP), *Energy Conversion and management (Elsevier)*, 47, 1222-1230, 2006.

[88] L. A. Zadeh, Fuzzy Sets, *Information and Control*, 8, 338-353, 1965.

[89] E. Garcia-Benitez, S. Yurkovich and K. M. Passino, Rule-Based Supervisory control of a Two-Link Flexible Manipulator, *Journal of Intelligent and Robotic systems*, 7 (2), 195-213, 1993.

[90] B. Jia, G. Ren and Z. Hiu, Fuzzy Switching Controller for Multiple Model, *Lecture Notes in Computer Science*, 3613, 1011-1014, 2005.

[91] R. A. Hilhorst, J. van Amerongen, P. Lohnberg, and H. J. Tulleken, A Supervisor for Control of Mode-Switch Processes, *Automatica*, 30(8), 1319–1331, 1994.

[92] K. J. Aström and B. Wittenmark, Adaptive Control, (Addison-Wesley, Massachusetts, USA, second edition, 1995).

[93] T. Takagi and M. Sugeno, Fuzzy identification of systems and its application to modelling and control, *IEEE Trans. on Systems*, *Man and Cybernetics*, 15(1), 116–132, 1985.

[94] A. S. Morse, Control using logic-based switching, *Trends in control*, (Springer, London, 1995).

[95] T. A. Johansen and R. Murray-Smith, The Operating Regime Approach to Nonlinear Modelling and Control, (Taylor & Francis, 1997).

[96] K. Danne and C. Bobda, Dynamic Reconfiguration of Distributed Arithmetic Controllers: Design Space Exploration and Trade-off Analysis. *In Proceedings of the 11th Reconfigurable Architectures Workshop (RAW'04)*, *IEEE Computer Society*, 26 - 27 April 2004.

[97] B. D. Anderson, T. Brinsmead, D. Liberzon and A. S. Morse, Multiple Model Adaptive Control with Safe Switching, *Inter. Journal of Adaptive Control and Signal Processing*, 15, 445-470, 2001.

[98] R. Murray-Smith and R. Shorten, Switching and Learning in Feedback Systems, *LNCS 3355*, (Springer-Verlag Berlin Heidelberg, 2005).

[99] R. Marino and P. Tomei, Adaptive Control of Linear Time-Varying Systems, *Automatica*, 39, 651-650, 2003.

[100] R. Abdullah, A. Hussain, K. Warwick and A. Zayed, Autonomous Intelligent Vehicle Control using a Novel Multiple-Controller Framework Incorporating Fuzzy-Logic based Switching and Tuning, *Neurocomputing, (Elsevier Science)*, *in press*, 2007.

[101] H. Hu and P.-Y. Woo, Fuzzy Supervisory Sliding-Mode and Neural-Network Control for Robotic manipulators, *IEEE Trans. on Industrial Electronics*, 53 (3), 929-940, 2006.

[102] K. J. Aström and B. Wittenmark, On Self-tuning Regulators, *Automatica, 9,* 185-199, 1973.

[103] D. W. Clarke and P.J. Gawthrop, Self-tuning Control, *Proc. Inst. Electrical Engineering*, Part D, 126, 633-640, 1979.

[104] A. Allidina and F. Hughes, Generalised Self-Tuning Controller with Pole Assignment, *Proc. Inst. Electrical Engineering, Part D, 127*, 13-18, 1980.

[105] A. Hussain, A. Zayed, and L. Smith, A New Neural Network and Pole-Placement based Adaptive Composite Controller, *Proc. 5th IEEE international multi-topic Conference (INMIC'2001)*, Lahore, 267-271, 28-31 Dec 2001.

[106] F. Cameron and D. E. Seborg, A Self-tuning Controller with a PID Structure, *Int. Journal of Control*, *38*, 401-417, 1983.

[107] S. Seshagiri and H. K. Khalil, Output Feedback Control of Nonlinear Systems using RBF Neural Networks, *IEEE trans. on Neural Networks*, 11 (1), 69-79, 2000.

[108] B. Huang and S. L. Shah, Performance Assessment of Control Loops, (Springer-Verlag, London, 1999).

[109] S. J. Qin and J. Yu, Recent developments in multivariable controller performance monitoring, *Journal of Process Control*, 17, 221–227, 2007.

[110] M. Jelali, An overview of control performance assessment technology and industrial application, *Control Engineering Practice*, 14, 441-466, 2006.

[111] A. Horch and A. Isaksson, A modified index for control performance assessment, *Journal of Process Control*, 9, 475-483, 1999.

[112] J. Schäfer and A. Cinar, Multivariable MPC system performance assessment, monitoring, and diagnosis, *Journal of Process Control*, 14, 113–129, 2004.

[113] R. Jyringi, R. Rice and D. J. Cooper, Opening the Black Box: Demystifying Performance Assessment Techniques, *Proc. ISA Expo 2005*, 495, 25-27 October 2005.

[114] M. H. Khammash and L. Zou, Analysis of steady-state tracking errors in sampled-data systems with uncertainty, *Automatica*, 37, 889-897, 2001.

[115] G.-Y. Tang, S.-M. Zhang and B.-L. Zhang, Optimal tracking with zero steady-state error for time-delay systems with sinusoidal disturbances, *Journal of Sound and Vibration*, 299, 633-644, 2007.

[116] O. L. Davies and P. L. Goldsmith, Statistical Methods in Research and Production, (Oliver and Boyd, Edinburgh-UK, 1972).

[117] C. Panchapakesan, M. Palaniswami, D. Ralph and C. Manzie, Effects of Moving the Centers in an RBF Network, *IEEE Trans. on Neural Networks*, 13 (6), 1299-1307, 2002.

[118] L. Zaccarian and A. Teel, A common framework for anti-windup, bumpless transfer and reliable design, *Automatica*, 38 (10), 1735–1744 2002.

[119] D. Nakazawa and A. Trunov, Model design and Data Analysis for Multi-Input and Multi-Output Systems, *Proc. of Int. Conf. on Integration of Knowledge Intensive Multi-Agent Systems*, Boston-USA, 368-374, 30 Sep.- 4Oct. 2003.

[120] A. Bentayeb, N. Maamri and J.-C. Trigeassou, Design of PID Controllers for Delayed MIMO Plants using Moments Based Approach, *Journal of Electrical Engineering*, 57 (6), 318-328, 2006.

[121] R. Ordónez and K. Passino, Stable Multi-Input Multi-Output Adaptive Fuzzy/Neural Control, *IEEE Trans. on Fuzzy Systems*, 7 (3), 345-353, 1999.

[122] S. S. Ge and C. Wang, Adaptive Neural Control of Uncertain MIMO Nonlinear Systems, *IEEE Trans. on Neural Networks*, 15 (3), 674-692, 2004.

[123] U. Borison, Self-Tuning Regulators for a Class of Multivariable Systems, *Automatica*, 15, 209-215, 1979.

[124] H. N. Koivo, A Multivariable Self-Tuning Controller, *Automatica*, 16, 351-366, 1980.

[125] L. Keviczky and K. S. Kumar, Multivariable Self-Tuning Regulator with Generalised Cost function, *Int. Journal Control*, 33(5), 913-921, 1981.

[126] M. J. Grimble and T. J. Moir, Multivariable Weighted Minimum Variance Self-Tuning Controllers, *Int. Journal. Control*, 42 (6), 1283-1307, 1985.

[127] A. S. Zayed, L. Petropoulakis and M. R. Katebi, An explicit multivariable self-tuning pole-placement PID controller*, Proc.12th International Conf. on Systems Engineering ICSE'97*, Coventry University, UK., 778-785, 9-11 September 1997.

[128] R. Abdullah, A. Hussain and M. Polycarpou, Fuzzy Logic based Switching and Tuning Supervisor for a Multivariable Multiple-Controller, *IEEE International conference on Fuzzy Systems (FUZZ-IEEE 2007)*, Imperial College, London, UK, 1644-1649, 23-26 July, 2007.

[129] C. W. Frey and H.-B. Kuntze, A neuro-fuzzy supervisory control system for industrial batch processes, *IEEE Trans. on Fuzzy Systems*, 9 (4), 570-577, 2001.

[130] N. Mai-Duy and T. Tran-Cong, Solving Biharmonic Problems with Scattered-Point Discretization using Indirect Radial Basis Function networks, *Engineering Analysis with Boundary Elements*, 30, 77-78, 2006.

[131] K. H. Johansson, The Quadruple-Tank Process: A Multivariable Laboratory process with an Adjustable Zero, *IEEE Trans. on Control Systems Ttechnology*, 8(3), 456-465, 2000.

[132] J. F.-Seara, F. Uhia and J. Sieres, Experimental Analysis of a Domestic Electric Hot Water Storage Tank. Part II: Dynamic Mode of Operation, *Applied Thermal Engineering*, 27, 137-144, 2007.

[133] J. G.-Serna, E. G.-Verdugo, J. Hyde, J. F.-Dubreuil, C. Yan, M. Poliakoff and M. Cocero, Modelling Residence Time Distribution in Chemical Reactors: A Novel

Generalised N-Laminar Model, Application to Supercritical CO_2 and Subcritical Water Tubular Reactors, *Journal of Supercritical Fluids*, 41, 82-91, 2007.

[134] C. Toy, K. Leung, L. Alvarez, and R, Horowitz, Emergency Vehicle Maneuvers and Control Laws for Automated Highway Systems, *IEEE Trans. on Intelligent transportation Systems*, 3 (2), 109-118, 2002.

[135] J. Naranjo, C. Gonzalez, R. Garcia, and T. Pedro, Using Fuzzy in Automated Vehicle Control, *IEEE Intelligent Systems*, 22 (1), 36-45, 2007.

[136] R. Gregor, M. Lützeler, M. Pellkofer, K. Siedersberger, E Dickmanns, EMS-Vision: A Perceptual System for Autonomous Vehicles, *IEEE Trans. on Intelligent Transportation Systems*, 3 (1), 48-59, 2002.

[137] A. Girard and J. Hedrick , Real-Time Embedded Hybrid Control Software for Intelligent Cruise Control Applications, *IEEE Robotics and Automotive Magazine, Special Issue on Intelligent Transportation System,* 12, 22-28, 2005.

[138] Y. Zhang, E. Kosmatopoulos, P. Ioannou, Autonomous Intelligent Cruise Control Using Front and Back Information for Tight Vehicle Following Maneuvers, *IEEE Trans. on Vehicular Tech.*, 48 (1), 319-328, 1999.

[139] P. Ioannou and Z. Xu, Throttle and Brake Control Systems for Automatic Vehicle Following, *IVHS Journal*, 1 (4), 345-377, 1994.

[140] L. Kun and P. Ioannou, Modeling of traffic flow of automated vehicles, *IEEE Trans. on Intelligent Transportation Systems*, 5 (2), 99- 113, 2004.

[141] K. Guo, H. Ding, J. Zhang, J. Lu and R. Wang, Development of a Longitudinal and Lateral Driver Model for Autonomous Vehicle Control, *Int. Journal of Vehicle Design*, 36(1), 50–65, 2004.

[142] W. K. Lennon and K. M. Passino, Intelligent Control of Brake Systems, *IEEE Trans. on Control System Technology*, 7(2), 188-202, 1999.

[143] A. Girard, A. Howell and J. Hedrick, Model-Driven Hybrid and Embedded Software for Automotive Applications, *Second RTAS Workshop on Model-Driven Embedded Systems*, pp. 25-28, 2004.

[144] W. El-Messoussi, O. Pages and A. El-Hajjaji, Four-Wheel Steering Vehicle Control using Takagi-Sugeno Fuzzy Models, *IEEE International conference on Fuzzy Systems (FUZZ-IEEE 2007)*, London, UK, 1866-1871, 23-26 July, 2007.

[145] U. Kiencke and L. Nielsen, Automotive Control Systems: For Engine, Driveline, and Vehicle, (Springer-Berlin, 2005).

[146] C. Wagner and H. Hagras, A Genetic Algorithm Based Architecture for Evolving Type-2 Fuzzy Logic Controllers for Real World Autonomous Mobile Robots, *IEEE International conference on Fuzzy Systems (FUZZ-IEEE 2007)*, Imperial College, London, UK, 193-198, 23-26 July, 2007.

[147] J. Mendel, Uncertain Rule-Based Fuzzy Logic Systems, (Prentice Hall 2001).

[148] J. Mendel and R. John, Type-2 Fuzzy Sets Made Simple, *IEEE Transactions on Fuzzy Systems*, 10 (2), 117-127, 2002.

[149] J. Wang, A.B. Rad and P.T. Chan, Indirect Adaptive Fuzzy Sliding Mode Control: Part I: Fuzzy Switching, *Fuzzy Sets and Systems*, 122, 21-30, 2001.

[150] K. Tanaka, M. Iwasaki and H.O. Wang, Switching Control of an R/C Hovercraft: Stabilization and Smooth-Switching, *IEEE Trans. on Systems, Man and Cybernetics, Part B*, 31 (6), 853-863, 2001.

[151] S. Chiu, Using Fuzzy Logic in Control Applications: Beyond Fuzzy PID Control, *IEEE Control Systems Magazine*, 18 (5), 100-104, 1998.

[152] V. Moudgal, K. Passino and S. Yurkovich, Rule-Based Control for a Flexible-Link Robot, *IEEE Trans. on Control Systems Technology*, 2 (4), 392-405, 1994.

[153] J. Platt, A Resource-Allocating Network for Function Interpolation, *Neural Computation*, 3 (2), 213-225, 1991.

[154] K. Gurney, An Introduction to Neural networks, (UCL Press 1997).

[155] J.W. Park, G. Venayagamoorthy and G. Harley, MLP/RBF Neural-Network-Based online Global Model Identification of synchronous generator, *IEEE tarns. on industrial Electronics*, 52 (6), 1685-1695, 2005.

[156] K. Passino, Intelligent Control for Autonomous Systems, *IEEE Spectrum*, 55-62, 1995.

[157] A. Bağiş, Determining Fuzzy Membership Functions with Tabu Search – An Application to Control, *Fuzzy Sets and Systems*, 139, 209-225, 2003.

[158] K. Warwick, QI: The Quest for Intelligence, (Judy Piatkus Limited-UK 2000).