DESIGN AND COMPARATIVE ANALYSIS OF MAC ARCHITECTURE USING SOFT COMPUTATIONAL PARADIGMS

A thesis submitted for the award of the Degree of

DOCTOR OF PHILOSOPHY
in
Electrical Engineering

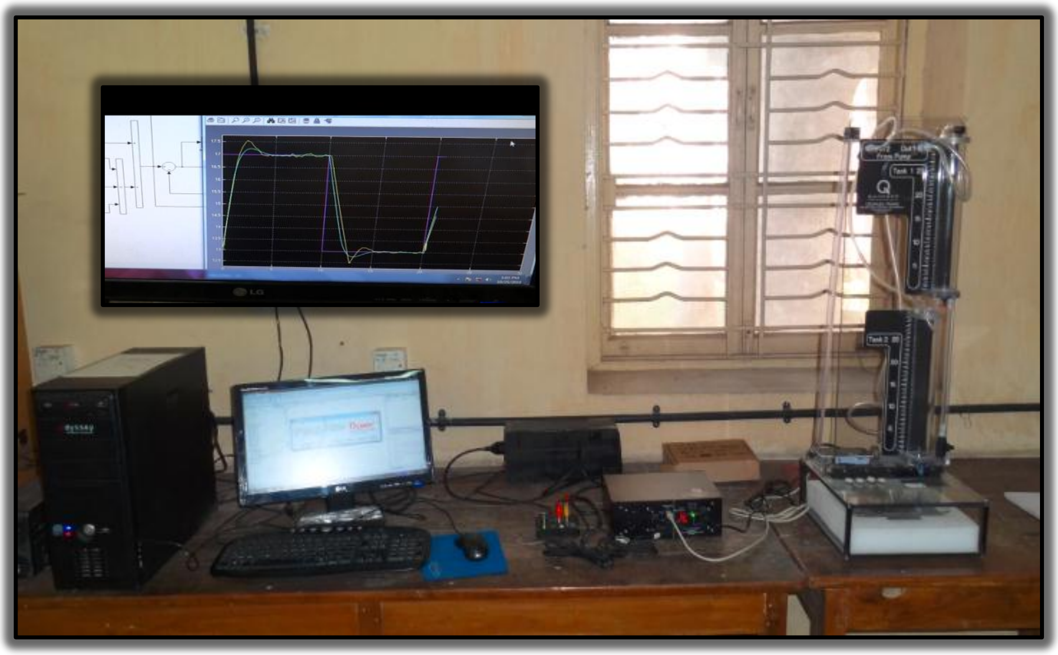
By

Rakesh B. Patel

ELECTRICAL ENGINEERING DEPARTMENT
FACULTY OF TECHNOLOGY & ENGINEERING
THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA
VADODARA – 390 001
GUJARAT, INDIA

October 2013

Design and Comparative Analysis of MAC Architecture Using Soft Computational Paradigms



By

Rakesh B. Patel

सत्यं शिवं सुन्दरम्

ELECTRICAL ENGINEERING DEPARTMENT
FACULTY OF TECHNOLOGY & ENGINEERING
THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA
VADODARA – 390 001
GUJARAT, INDIA

Dedicated

To

My Family and Teachers

Certificate

This is to certify that the thesis entitled, **'Design and comparative analysis of MAC acrhitecutre using soft compuational paradigms '** submitted by **Rakesh B. Patel** in fulfillment of the degree of **DOCTOR OF PHILOSOPHY** in Electrical Engineering Department, Faculty of Technology & Engineering, The M. S. University of Baroda, Vadodara is a bonafide record of investigations carried out by his in the Department of Electrical Engineering, Faculty of Technology & Engineering, M. S. University of Baroda, Vadodara under my guidance and supervision. In my opinion the standards fulfilling the requirements of the Ph.D. Degree as prescribed in the regulations of the University has been attained.

October
2013

Prof. Satish K. Shah
Department of Electrical
Engineering,
Faculty of Technology & Engineering,
The Maharaja Sayajirao University of
Baroda, Vadodara – 390001

Head
Department of Electrical Engineering,
Faculty of Technology & Engineering,
The Maharaja Sayajirao University of
Baroda, Vadodara – 390001

Dean
Faculty of Technology & Engineering,
The Maharaja Sayajirao University of
Baroda, Vadodara – 390001

Declaration

I, **Rakesh B. Patel** hereby declare that the work reported in this thesis entitled **'Design and comparative analysis of MAC architecture using soft compuational paradigms'** submitted for the award of the degree of **DOCTOR OF PHILOSOPHY** in Electrical Engineering Department, Faculty of Technology & Engineering, The M. S. University of Baroda, Vadodara is original and has been carried out in the Department of Electrical Engineering, Faculty of Technology & Engineering, M. S. University of Baroda, Vadodara. I further declare that this thesis is not substantially the same as one, which has already been submitted in part or in full for the award of any degree or academic qualification of this University or any other Institution or examining body in India or abroad.

October 2013 **Rakesh B. Patel**

Acknowledgement

I express my thanks to Almighty for providing me inspirations, strength, energy and patience to start and accomplish my work with the support of all concerned a few of them I am trying to acknowledge.

I, with great privilege and deep sense of gratitude, heartily thank my research guide, **Prof. S.K.Shah**, who has constantly guided me throughout the tenure of research work. His positive and painstaking efforts, constructive suggestions and his readiness to help at any time even after office hours has resulted in research work. Need has less to say that without his assistance it would never have been possible to carry out this work. I am also thankful to **Dr. S. K. Joshi**, Head of Electrical Engineering Department, Faculty of Technology & Engineering, for his inspiration and support.

I am deeply indebted to entire **SVIT family, Chairman, Principal, Management** and all **my colleagues** who have supported me in research work. I heartily thank to my friend **Prof. Sandip R. Panchal** for their moral and technical support.

My deepest gratitude goes to my family for their unflagging love and support throughout my life. This dissertation is simply impossible without them. I express my heartiest gratitude to **my parents, my wife SUMI, my son OM** and **my friends** for their unconditional support in every aspect of my life. Without their patience and inspiration it would have not been possible for me to start and completed this research work.

Finally, I thank all those who helped me directly or indirectly and whose names would have been inadvertently missed above.

RAKESH B. PATEL

Abstract

A large number of control systems operate in the real-world and are expected to deal with the range of changing conditions of the physical process that is being controlled. It may be a laborious and difficult task to design one controller that fit all operating conditions. So, the multi-agent system is used for solving this type of control problem in this thesis. It is based on the concept of an autonomous agent; the agent will be active in specific operating regime. This concept is recently introduced in the artificial intelligence. The method enables the use of heterogeneous control algorithms and integrating techniques. Furthermore, it allows for an incremental design of the control system, which is attractive from a designer's point of view.

Integrating of the concepts of local controller and an agent has resulted in a so called controgent. A controgent is local controller which has knowledge about its operating regime. The result of controlgent can be combined into overall result using coordination objects which solves dependencies of partial control problem. The implementation framework has been developed to realize a controller using the multi-agent concepts which is known as Multi-Agent Control (MAC) architecture. This architecture contains the agent, controgents and coordination object.

Now a day, the systems are used for more and more complex problems requiring processing of huge amounts of data and long computational time. So the agent can be designed using soft computational paradigms to achieve its goal with minimum efforts. Three architectures: Basic MAC_SC, Advance MAC_SC and Intelligent MAC_SC are developed using soft computational paradigms during this research.

Three real world control problems are described to illustrate the proposed MAC_SC architectures. These are: 1) Basic MAC_SC for helicopter control, 2) Advance MAC_SC for boiler turbine plant, 3) Intelligent MAC_SC for water tank control. These applications demonstrate the utility of the framework and the power of the design method.

Table of Contents

List of Figures

List of Tables

List of Photographs

List of Abbreviations

MAS	Multi-Agent System
AI	Artificial Intelligence
BDI	Belief-Desire-Intention
JADE	Java Agent Development Environment
STD	State Transition Diagram
SCP	Soft Computable Paradigms
ANN	Artificial Neural Network
FIS	Fuzzy Inference System
ADLINE	ADaptive LINear Element
NARX	Nonlinear AutoRegressive network with Exogenous inputs
CPA	Control Performance assessment
GUI	Graphical User Interface
PMA	Performance Measure Agent
CA	Critic Agent
IA	Identification Agent
SPA	Signal Processing Agent
CO	Coordination Object

Chapter 1
Introduction

The chapter provides an overview and the context for the remainder of the thesis. It also introduces the objective of the research work and scope of the improvement in the existing methods.

In the recent trends, there has been a strong interest in control of complex system which has computational complexity, nonlinearity, significant uncertainty and heterogeneity. These types of control problems cannot be solved easily and completely by a single (traditional) control algorithm, i.e. a control system with only one controller such as a conventional PID controller. Some of practical solutions have been reported in survey that complex control problems can be solved by using adaptive control [1], model based control [2] and intelligent control [3]. However, there are several restrictions for applying these methods in industrial applications, as summarized below:

- A single algorithm based controller has a relatively narrow field of application due to diversity of real–life problems. Therefore, more flexible methods or a toolbox of methods are required in industry.
- The demand of deterministic real-time behavior is very high; the control system must be capable to build multi-threaded controllers with thread safety to react in synchronous and asynchronous events.
- Many nonlinear processes can be controlled using the well-known and industrially proven PID controller. A considerable direct performance increase (financial gain) is demanded when replacing a conventional control system with an advanced one.
- Due to the complexity of implementation and computational power, the hardware requirements are very high.
- New approaches are not easily available in a ready-to use as industrial demand. Custom design requires considerable effort, time and money.
- The complexity of tuning and maintenance, these methods are for non-specialized engineers.
- The modeling of non-linear real-time system is often in question in industrial environment.

1.1 Complex control system

Controlling a large scale complex system is not only expensive, but also cannot reach certain satisfactory level, because there are limits on a human as a decision maker and a problem solver (especially limits in cognitive processing), which is called bounded rationality [4]. Human beings can be part of the control system, but cannot be the only part. More importantly, complex dynamic systems are subject to component failures that tend to present major challenges to the control system.

In the control context, a control system controls a system to behave in a certain desired way to make the system accomplish one or more tasks, while satisfying constraints existing in the physical system states and control actions. The classical control strategy is feedback control as shown in figure 1.1.

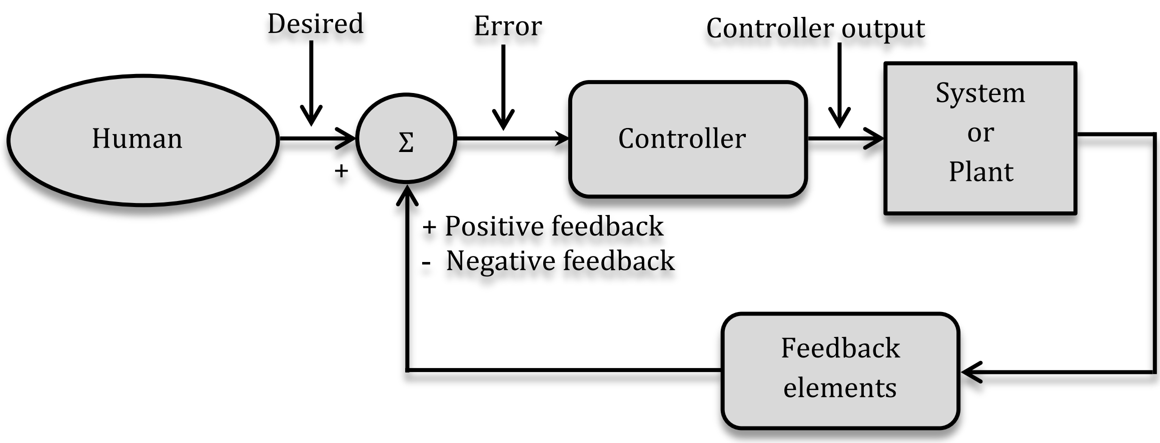


Figure 1.1 Classical control system

Two types of feedback control system are used in various control engineering application, Positive feedback (Desired and actual output are added) and Negative feedback control (Desired and actual output are subtracted). The negative feedback system is more stable compare than positive feedback system. It also makes systems more immune to random variations in component values and inputs.

The overall control system has four parts: sensing, assessing, planning and action. First, the controller senses its environment and collects the information for assessment. In second step, controller analyses the collected information and reasons about system current states. In third step, it takes decisions which are objective-oriented based as assessment of current situation without violating any system constraints. There are two type of goal for controller to achieve its objective: Short term goals such as bringing the system to a desired state and long-term goals, such as making the system achieve maximum rewards in a certain time period. In fourth step, the action will be taken on the system. As per these discussions, the general control sketch with reference of various activities is illustrated in figure 1.2.

Control of a complex system has to address the basic control issues mentioned above as well as a few more challenges:

- **Robustness:** automatic and optimal reconfiguration of the whole system, due to system dynamic changes/some failure modes without using predefined recipes.
- **Uncertainty inference:** relatively accurate and quick estimations of system states from noisy and incomplete information.
- **Flexibility:** easy adaptation when parts of the complex system are modified or the system is extended.
- **Global consistency:** actions taken by local system and information gathered and processed locally need to be globally consistent.
- **Time delay:** quick responses to local components' failures to avoid failure propagation.

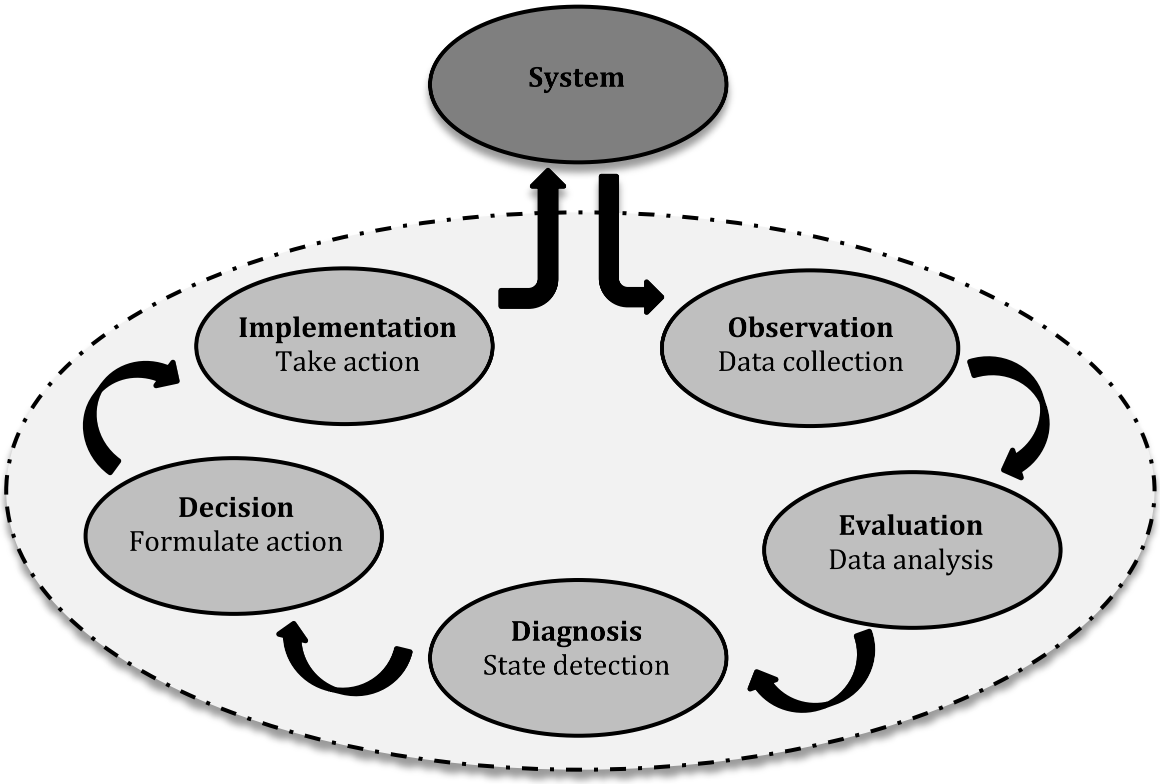


Figure 1.2 General Control Strucutre

1.2 Aim and research scope:

New controller is required to design for solving the industrial demand and sort out the delimitation of available controllers. Agent based control is an interdisciplinary field based on the combination of Artificial Intelligence, Automation Control and Intelligent Methods [5]. Ideas and research results of all sects of artificial intelligence have been applied to control domain [6] which give rise to emergence of Expert System Control [7-8], Fuzzy Control [9-10], Neural Network Control [11-12] and Simulating Human Intelligent Control [13], etc.

Agent technology has been extensively applied to many fields [14], from simple personal e-mail filter to complex control system such as traffic control system [15], process control [6, 16] and various automation applications [17-18]. Thus, a multi-agent approach would be sensible for problems that are inherently (physically or geographically) distributed where independent processes can be clearly distinguished. Multi-Agent System (MAS) employ various agents who are characterized by properties such as autonomy, social ability, reactivity, pro-activeness etc. This means that an agent can decide for itself whether it should undertake some action to achieve the target goal [19].

A pool of agents may be used to solve the complex control problem in which each one is responsible for solving a part of the whole problem, it is known as a Multi-Agent Control (MAC) [20-21]. Because multiple agents are acting on their own particular problems to solve the overall problem, conflicts between individual agents may arise, as these partial problems are interdependent. Conflicts between individual agents may also arise when outputs of these agents are combined. These conflicts are solved by coordination mechanisms between agents [6]. These coordination mechanisms determine when and how actions of the controller are applied to the plant. Main coordination mechanisms used in multi-agent system: Fixed priority, Sequential, Cyclic, Master-Slave and Parallel.

The above multi-agent architecture is very suitable to solve all kind of problems in various engineering including complex control problems. Literature Survey indicates that MAS technology is being used for a wide range of control applications including condition monitoring [22,23], distributed control [24], hybrid control [25], congestion control [26], and automation [27].

Survey also reveals the growing interest of the research community on the relatively new field of soft computing for engineering applications. Soft computational paradigms, is a consortium of tools for natural intelligence stemming from approximate reasoning (fuzzy logic), learning (Artificial neural networks), optimization techniques (genetic algorithms, genetic programming) etc. Soft computing is based on concepts viz: human-like reasoning, learning in uncertain environment, probabilistic reasoning, to give low cost robust solution to control problems.

Fuzzy control has proven to be a successful control approach to many complex nonlinear systems or even nonanalytic ones [28]. It has been suggested as an alternative approach to conventional techniques in many situations.

Artificial neural networks (ANN) [29] are technologies for optimization of performances. ANN is built with a systematic step-by step procedure to optimize a performance criterion or to follow some implicit internal constraint, which is commonly referred to as the learning rule. ANN is an adaptive nonlinear system that learns to perform I/O mapping from data.

Genetic algorithms are members of a collection of methodologies known as evolutionary computation (EC). These techniques are based on the selection and evolution processes that are met in nature and imitate these principles in many scientific domains. A few years later Goldberg, in his book [30], studied several aspects of the implementation of genetic algorithms and examined their potential in the context of optimization and learning for large-scale complex systems. The developments in the research field of genetic programming have been utilized in various application domains.

Such systems are used today for more and more complex problems requiring processing of huge amounts of data and long computational time. So the agent can be designed using soft computational paradigms to achieve its goal with minimum efforts [31]. The new architecture design using conceptual relation of agent and soft computing is very attractive area for researchers. The implementation of new proposed control strategy can be easily possible using software development support tools such as MATLAB, SIMULINK and related Toolboxes.

THE ABOVE MENTIONED RESEARCH WORK HAS INSPIRED THE AUTHOR TO WORK IN THIS DIRECTION AND TO THINK OF BETTER ALTERNATIVES FOR CONTROL ENGINEERING WITH MULTI-AGENT SYSTEM AND SOFT COMPUTATIONAL PARADIGMS.

Aim of proposed research work is to develop various MAC architectures to solve the control problem with use of soft computational paradigms. The features and discussion of research carried out in the thesis includes:

- Derive the mathematical modeling of non-linear system.
- Development of multi-agent control architecture. [32, 33, 34]
- Designing & simulation of PID controller for plant using classical control approaches, and intelligent controller as fuzzy logic controller and neural network controller [35].
- Design the new control performance assessment method using various control parameters as overshoot, steady state error, variance in output, rise time, settling time, and control effort [35].
- Design the performance measure using intelligent method fuzzy logic to identify the control algorithm response on the system and apply the best algorithm of operating regime.

- Designing the methods to identify the plant using neural network with back propagation tuning as well as genetic algorithm tuning. Also, cover the comparative analysis of the two tuning methods for neural network [36].
- Designing of multi-agent controller using multi-agent architecture and analysis with traditional approaches [32, 33, 34].
- Implement classical and Intelligent MAC_SC for standard benchmark problems such as coupled tank, boiler turbine plant, helicopter control system etc... and test on hardware prototype models.

1.3 Outcome of research work

During research work, the lots of literatures survey has been done for multi-agent architecture, multi-agent in control applications, model based control, multiple model control, intelligent methods and its role in control engineering, etc... After, various components of multi-agent control architecture are defined and discussed. Three MAC architectures are developed for SISO and MIMO systems as Basic MAC_SC, Advance MAC_SC and Intelligent MAC_SC. Basic MAC_SC is designed especially for helicopter system. Advance MAC_SC is developed for power engineering application in especially for boiler-turbine plant. Intelligent MAC_SC is designed for the water tank system in this thesis. But, it is generalized MAC architecture after some modification; it can be applied for highly changing operating conditions as well as nonlinear plants or systems [32, 33, 34, 35, 36].

1.4 Organization of thesis

The thesis organised in form of the ten chapters as follows:

Chapter: 1 ***Introduction***

The chapter provides an overview and the context for the remainder of the thesis. It also introduces the objective of the research work and scope of the improvement in the existing methods.

Chapter: 2 ***System Solution: Agent Technology***

Chapter gives the survey of different prevailing techniques of system solution is the main part of the literature survey. The exhaustive search is done to find out the basic techniques and modification suggested by various researchers. Survey of agent based control is covered in this chapter. Detail discussion on structuring of complex control problem is given with various examples.

Chapter: 3 ***Multi-Agent Control (MAC) Architecture***

This chapter gives a general overview of agents and multi-agent systems, which is required to develop the multi-agent control architecture. Various multi-agent architectures are discussed in this section. The key issues of multi-agent system (MAS) are also covered in this chapter. A new MAC architecture is designed using multi-agent concepts for control engineering applications.

Chapter: 4 ***Soft Computational Paradigms***

Chapter gives a brief overview and describes theoretical background of the computing techniques such as Fuzzy logic, neural network and genetic algorithm. The most popular tools used by the researchers for development and simulation study of the system under test such as MATLAB, SIMULINK and associated tool boxes for software development and testing are also described. Toolboxes available for deploying soft computing techniques in MATLAB and used in our research work for the design and testing of proposed techniques are described in detail.

Chapter: 5 ***Smart System Identification***

Chapter expresses the identification of nonlinear system with different methods. The intelligent approach is used to identify the model of nonlinear system. The neural network is tuned with back propagation and genetic algorithm for comparative analysis of two approaches.

Chapter: 6 ***Control Performance Assessment***

The performance of control is continuously assessed online with different parameters as steady state error, overshoot, settling time, rise time, variance and reference change. The fuzzy logic is used to identify the better control strategy for control problem with help of performance assessment. The detail of membership functions design, rule base and its implementation are discussed in chapter.

Chapter: 7 ***Classical MAC_SC***

The multi-agent control framework is discussed in chapter 3. In this chapter, two multi-agent controllers are developed using MAC framework. Basic MAC_SC is designed for helicopter control and Advance MAC_SC is designed for boiler-turbine control. Structuring of controls and its result analysis for both problems are given in this chapter.

Chapter: 8 ***Intelligent MAC_SC***

Intelligent MAC_SC are designed using soft computational paradigms. Control system design of a simplified water tank System as an application of the proposed methodology and process will be implemented. Detailed introduction of simulation and actual plant will be given in chapter. A step by step procedure for designing MAC controller using the proposed methodology and process will be described in detail.

Chapter: 9 ***Conclusion and suggestion for extension***

The chapter presents discussion on results and summarizing the contribution of research work carried out by author. The limitations and assumptions are also described in this chapter. Directions for future research work are suggested.

Chapter: 10 ***Bibliography***

Thesis ends with bibliography which includes the list of reference use in each chapter and list of publications and presentations done based on this work.

The organizing structure of thesis is also given in figure 1.3.

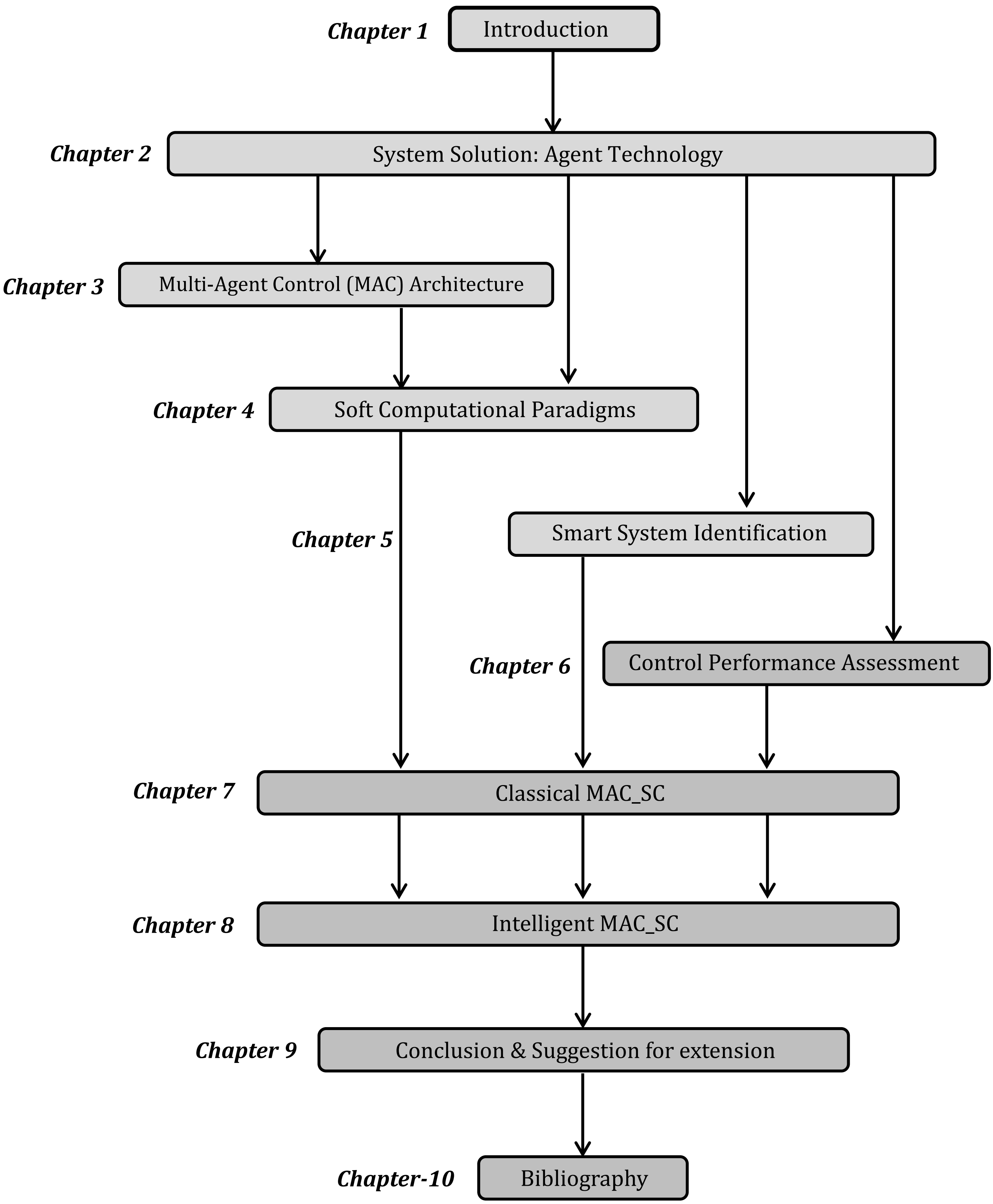


Figure 1.3 Organized structure of thesis

Chapter 2
System Solution: Agent Technology

Chapter gives the survey of different prevailing techniques of system solution is the main part of the literature survey. The exhaustive search is done to find out the basic techniques and modification suggested by various researchers. Survey of agent based control is covered in the chapter. Detail discussion on structuring of complex control problem is given with various examples.

The characteristic of complex control system or plant is focused in the chapter. There are possibilities and limitations of available control theory and tools in tracking the desired the response of the systems. The various control systems are given with its limitations. The motivate selection of multi-agent controller (MAC) as a research tool is discussed.

2.1 Review of various complex control methods

The actual control system is designed after definition of the control goal and modeling the plant. First specify all design specifications as objectives and constraints of problem and develop the required models of plant, disturbances and their uncertainties.

It is assumed that the uncontrolled system does not satisfy the specifications and that the task is to satisfy these through manipulating the system dynamics through control. The basic steps of design the controller are given in figure 2.1. But, the actual control system design includes selecting the control structure, selecting the control algorithm and tuning the controller.

In the idealized case of perfect knowledge of system and disturbances, the control system can be designed using an open-loop control approach. In an analytic design procedure, a possible approach to solve the design problem would be to apply dynamic optimization to search for satisfactory system performance, employing trajectories of manipulated variables as decision variables. While such an open-loop approach provides useful insight in achievable performance and system behavior, it is unsuccessful however to actually control systems due to uncertainty with respect to plant and disturbances. Instead of open-loop control, feedback control is usually applied to reduce sensitivity to uncertainty and to disturbances which is discussed in section 1.1.

Stage	Action
1	Definition of goal
2	Modeling of plant
3	Input-output selection
4	Controller configuration
5	Controller design
6	Controller evaluation
7	Implementation, testing and commissioning

Figure 2.1 Control system design steps

Many methods and approaches to control have been developed during the last few decades. An overview of methods is given in table 2.1, in relation to the characteristics of the activated with various control applications. Digital control was developed with the emergence of digital computing [1]. Optimal control was developed as a rational, mathematically well-based design method, both for deterministic systems and for stochastic systems [2,3], adaptive control for systems with varying parameters [4, 5], nonlinear control [6,7], robust control theory to better understand and account for model uncertainty [8], expert system control for crisp rule-based control and fuzzy control for fuzzy rules.

It was recognized in 1973 by Foss [9] that the control problem could not be coped with by one single all-embracing theory, while he indicated that it would be the theoretician to close the gap between actual problems and available theory. Still, no such single theory is available. With all the theoretical developments described above, several methods have well advanced, but it is still much of an art requiring significant application specific knowledge to select those pertinent for the application.

2.1.1 Classical control

Various control methods developed until the late 50's such as PID control, cascade control, ratio control, etc… are called classical control. PID controllers are very popular in industrial applications among all other classical control. For many applications, PID controllers are the optimum choice and will simply outperform almost any other control option.

Table 2.1 System characteristics Vs. control methods

System characteristics	Control methods
Interactions	• Static/dynamic decoupling • Multivariable modern control LQG, IMC, MPC, H_∞
Stiffness	• Hierarchical control • Singular perturbation
Uncertainty	• Robust control method, • H_∞, Internal model control, Adaptive control
Time varying parameters	Adaptive control, Robust control
Constraints	MPC
Nonlinearity	Several methods • Linearization • Piecewise linearization • Feedback linearization • Optimal control theory • Rigorous nonlinear optimization(MPC)
Lack of mathematical models	Rule based control: Expert systems, Fuzzy control, Multi-agent control

This is why they are currently used in over 95% of closed-loop processes worldwide [10], governing everything from temperatures, flow rates, mixing rates, chemical compositions and pressures in a limitless number of applications.

PID controllers can also be tuned by operators who do not possess a strong background in differential equations, electrical engineering or modern control theory; this grants PID controllers a very powerful ability to drastically change a given process (called a "plant model") with a system that is very simple and robust. The PID algorithm can be implemented in several ways. The parallel form is the typical form where the P, I and D components are as per shown in figure 2.1.

The output of the controller is given by:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \qquad 2.1$$

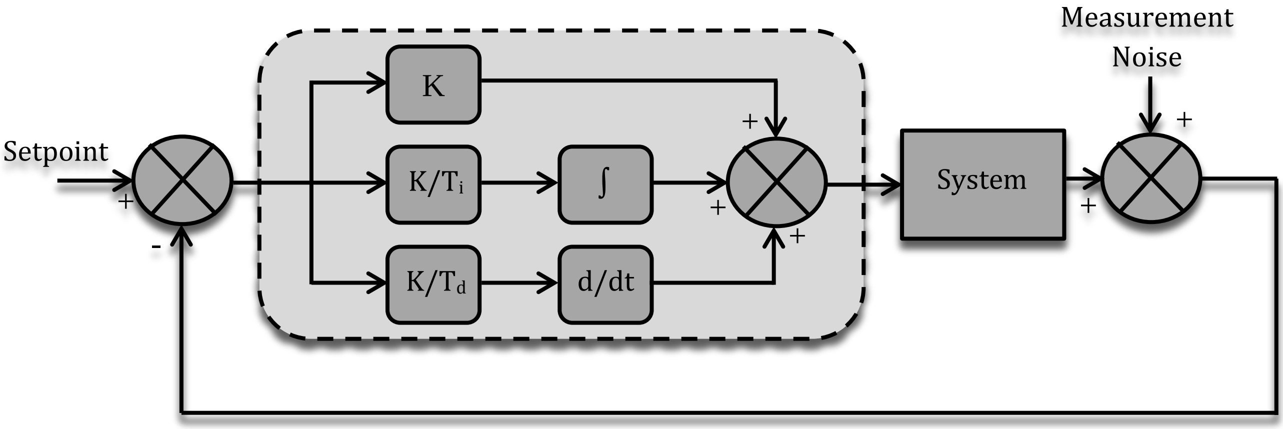


Figure.2.2 Parallel PID control system

Bhattacharyya and colleagues used a mathematical generalization of the Hermite-Biehler theorem to find all stabilizing PID controllers for systems with time-delay [11, 12, 13]. In [14, 15] an innovative controller design method, which did not require complex mathematical derivation, was presented. The authors of extended their research by obtaining the entire region of PID controllers that met certain gain and phase margin requirements. In [16, 17, 18, 19], techniques for finding all achievable PID controllers that stabilized an arbitrary order system and satisfied weighted sensitivity, complementary sensitivity, robust stability and robust performance constraints were introduced.

Although a PID controller provides an optimum solution to many processes, it is not a panacea for all control problems that may be encountered. This is especially true for processes with ramp style changes in set-point values or slow disturbances [20]. PID controllers can also perform poorly when the gain values must be greatly reduced in order to prevent a constant oscillation or hunting about the set-point value. Furthermore, PID controllers are linear and so care must be taken when using them with inherently nonlinear systems i.e., systems that do not satisfy the superposition principle or systems with an output that is not proportional to its input, such as air handling and mixing applications. For nonlinear systems, gain scheduling where utilizing a family of linear controllers that are independently activated based upon the values of scheduling variables determines current operating region of the system is most often used. Sometime the single PID controller is not suitable for various operating regime of plant.

2.1.2 Adaptive control

The severity of the nonlinearities in processes influences the selection of control algorithms for successful control of a process. Control strategies based on a linearized model may not yield satisfactory performance, if the process is subject to large disturbances or significant set point changes from, say, an on-line optimizer. In addition, the wide range of operating conditions encountered in start-up or shut-down of continuous processes and trajectory tracking of batch processes, in which there is the absence of a steady state, also pose an important challenge. These cases are best handled with adaptive control strategy. Only limited studies have been reported concerning the development of adaptive control strategies for and Seborg [21] presented an adaptive nonlinear control strategy for a bench-scale pH neutralization system, which is obtained by augmenting a nonlinear controller with a special indirect parameter estimation scheme. Clarke-Pringle and Mac-Gregor [22] studied adaptive temperature control of multi-product, semi-batch polymerization reactors. Their nonlinear adaptive controller consisted of a nonlinear controller with an extended kalman filter. Number of research papers [23, 24, 25, 26, 27] has been written, the number of application appears to be steadily increasing for this methods.

Adaptive control systems have been investigated for over four decades. Since the beginning, for the sake of mathematical tractability, adaptive control theorists confined their attention to time invariant systems with unknown parameters [28, 29]. The accepted philosophy was that is an adaptive system was fast and accurate when the plant parameters were constant but unknown, they would also prove satisfactory when the parameter varied with time, provided the latter occurred on a relatively slower time scale. Based on these general principles, adaptive control was extensively studied and numerous globally stable and robust adaptive control algorithms were derived [29]. Extensive computer simulations of adaptive control algorithms have revealed that when there are large error in the initial parameter estimated, the tracking error is quite often oscillatory with unacceptably large amplitudes during the transient phase. In the highly, competitive industrial world, new classes of problem are arising where such variations in parameters are quite common. It was to cope with such situations that the new approach of control is required [30].

2.1.3 Intelligent control

The concept of intelligent control was first introduced nearly two decades ago by Saridis [31]. Despite its significance and applicability to various processes, the control community has not paid substantial attention to such an approach. In recent years, intelligent control has emerged as one of the most active and fruitful areas of research and development (R&D) within the spectrum of engineering disciplines with a variety of industrial applications.

The ability of research to impact products hinges not so much on finding the best solution to a problem, but on finding the right problem and then solving it in a marketable way [32]. The study of intelligent control systems requires both defining some important expressions that clarify these systems, and understanding the desired application goals. It has been one of the fastest growing areas in the field of control systems over the last 15 years. Numbers of industrial applications have been developed using intelligent control. It has different tools for emulating the biological behavior that could solve problems as human beings do. The main tools for intelligent control are presented below:

- **Fuzzy logic systems** are based on the experience of a human operator, expressed in a linguistic form (normally *IF–THEN* rules).
- **Artificial neural networks** emulate the learning process of biologic neural networks, so that the network can learn different patterns using a training method, supervised or unsupervised.
- **Evolutionary methods** are based on evolutionary processes such as natural evolution. These are essentially optimization procedures.

The basic block diagram of intelligent control is given in figure 2.3.

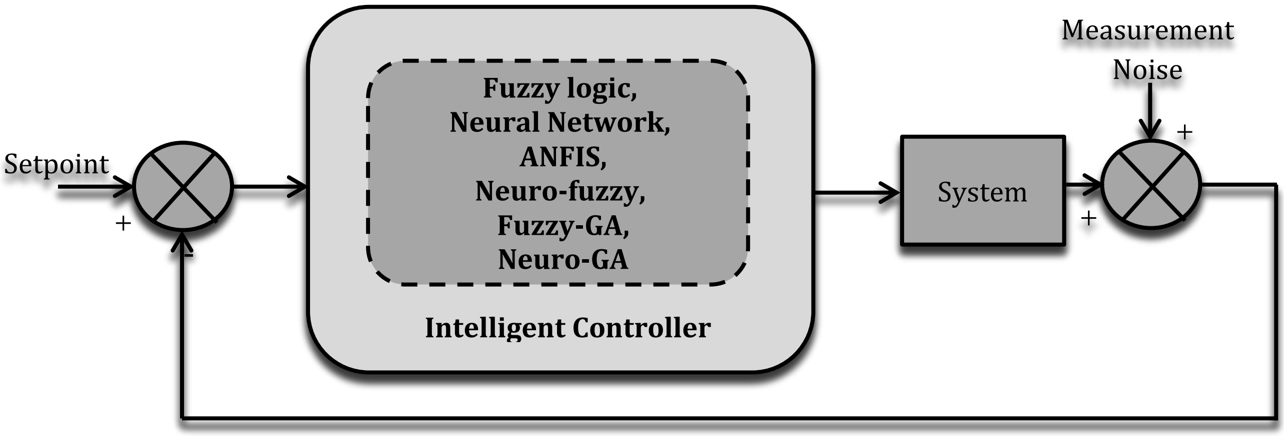


Figure.2.3 Intelligent control system

Each one has merits and demerits, but some of the demerits can be decreased by combining two or more methods to produce one system as hybrid systems. The fuzzy system is combined with neural networks to obtain a neuro-fuzzy system. For instance, the adaptive neural based fuzzy inference system (ANFIS) was proposed in order to utilize the best part of fuzzy logic inference using an adaptive neural network topology [33]. Different authors have presented many hybrid systems, but the most important and useful combinations are [34]:

- Neural networks combined with genetic algorithms [35].
- Fuzzy systems combined with genetic algorithms [36].

- Fuzzy systems combined with neural networks [37].
- Various other combinations have been implemented [38, 39].

Since fuzzy logic was first presented by Prof. Lotfi A. Zadeh, the number of fuzzy logic control applications has increased dramatically. For example, in a conventional proportional, integral, and differential (PID) controller, what is modeled is the system or process being controlled, whereas in a fuzzy logic controller (FLC), the focus is the human operator's behavior. In the PID, the system is modeled analytically by a set of differential equations, and their solution tells the PID controller how to adjust the system's control parameters for each type of behavior required. In the fuzzy controller, these adjustments are handled by a fuzzy rule-based expert system, a logical model of the thinking processes a person might go through in the course of manipulating the system. This shift in focus from the process to the person involved changing the entire approach to automatic control problems [40].

Intelligent control is a good solution for processes where the mathematical model that describes the system is known only partially. In fact, the PID controller is one of the most functional solutions used nowadays, because it requires a very short time for implementation and the tuning techniques are well known. The fuzzy systems can be used to tune direct and adaptive fuzzy controllers, as well as, how these systems can be used in supervisory control. Although the Intelligent control is more complex in structure than the PID controller, it gives a better response if the system changes to a different operation point [41]. So, the intelligent system is very popular to solve the control problem in various forms.

2.2 Agent in control: State of art

2.2.1 Review of agent in control

Nowadays agent technology is being used for various control problems. The group of agent are call multi-agent system which is widely used in various application as scheduling and planning [42, 43], diagnostics [44], condition monitoring [45-47], distributed control [46, 48], hybrid control [49], congestion control [50, 51], system restoration [52], market simulation [53, 54], network control [54, 55], and automation [56]. Moreover, the technology is growing to the point where the first multi-agent systems are now being immigrated from the laboratory to the utility, allowing industry to gain experience in the use of MAS and also to evaluate their effectiveness [42]. Some primary questions often arise when discussing multi-agent systems and their role in control engineering.

- What benefits are offered by multi-agent systems?
- What differentiates them from the existing systems and approaches?

- Which kind of problems can they be applied?
- How multi-agent systems should is designed?
- Are there any special considerations for the application of MAS in control engineering?
- How should multi-agent systems be implemented?

As multi-agent system (MAS) is a new technology, a number of technical challenges need to be overcome if they are to be used. It is essential concepts that how multi-agent system (MAS) can be explored for control engineering. The multi-agent basic concepts and its architectures are discussed in chapter: 3 in detail. Most important technical aspects in multi-agent system (MAS) implementation and development in engineering environment are also explained. Design methodologies, standards, tools, and various architectures to provide effective MAS-based control designs are addressed and a discussion on important related standards. Finally, some comments and new perspectives for design and implementation of agent-based control systems are discussed with various applications.

2.2.2 How agent can be used in control engineering?

Agent concept is continuous used in various engineering problem. Here, how the agent can be used to solve the control problem? The concept of controller and agent are to be re assembling together for control application [57]. Several similarities of agent and controller are as below:

- Both a controller and an agent have objectives that determine their behavior. In agent terms, this is called the goal of the agent. Sometimes the agent has an explicit representation of the objective such that it can reason about it (In deliberative architecture), but there are also agents that do not have an explicit representation of their objective (In reactive architecture). In control engineering terms, the "goal" of the controller is the design objective posed implicitly by the designer and is given by the specification of the control objectives. In this sense, a controller resembles an agent.
- Both controller and agent communicate with some surrounding using sensing and acting. An agent acts on its environment to achieve its goal, and a controller influences the plant it is controlling by generating control signals.
- From control engineering, the systems are analyzed base on dynamic behavior of plant. Analyzing the robustness, stability and performance properties of feedback control systems are major concerns. The field of multi-agent system is concerned with analyzing and designing systems in terms of agents. The field of control offers well-developed insights into the influence of feedback loops upon the dynamics of signal processing, so, the design of MAS can be developed to achieve this objective.

- A controller is usually designed to operate continuously during the whole operating time of a control system. An agent, however, decides whether it wants to get operational and produce actions. It might not produce actions during the whole operating time of the system. A controller that only operates in some restricted part of the operating regime of the controlled system comes close to the concept of an agent.
- Matters related to the thread of control are a major concern in multi-agent system (MAS), whereas this is generally not addressed during the design of controllers. On the other hand, control applications face hard real-time constraints and there is a significant body of knowledge in the control engineering community on how to deal with that. Agent based systems are typically structured and implemented in ways that make it impossible to guarantee that hard real-time bounds will be met.

Therefore, bringing the best of both fields together in the form of a multi-agent controller results in systems that:

- rely on organization principles as known from multi-agent system (MAS);
- have been designed with concern for dynamic behavior due to feedback loops;
- are structured and implemented in such a way that hard real-time bound will be met surely.

Two different ways can be implemented with combination of controller and agent. The first way is to design a controller for the sense-think-act mapping of a particular agent. The controller becomes the architecture of the agent. Another way is to use agents for execution of control algorithms. A controller would consist of several agents, each becoming active and producing control signals under particular operating conditions of the controlled plant. Both kinds of combinations have been considered in control engineering, as is discussed in the next section.

2.3 Discovering of control problem

2.3.1 Complex control problem

Control theory is concerned with solving control problems. The general feedback control system is illustrated by the block diagram in figure 1.1. The controller is designed in such way so that the control algorithm can be generated the control signal to achieve the control objective [58]. Selection of control algorithm is the essence of most control problem. The definition of a complex control problem starts with the observation that there are basically two kinds of control problems. The first kind of control problems are those that can be solved as a whole. For instance, if a control problem has been solved before (e.g. by using control theory), then any new problem that is similar in nature to this problem can be solved by an identical solution. Solving the control problem requires some tuning or re-calculation of control system parameters; the structure of the control problem needs not to be determined. The second kind of control problems are those that need to be divided into a set of partial control problems in order to solve them. A partial control problem can either be an elementary or a compound control problem. Depending on the way a control problem is solved, two different classes of problems can be defined.

- **Elementary control problem:** An elementary control problem is a control problem that is solved as a whole.
- **Compound control problem***:* A compound control problem is a control problem that is solved by dividing it into a set of two or more partial control problems.

The relations between a (partial) control problem, an elementary and a compound control problem are pictured in figure 2.4.

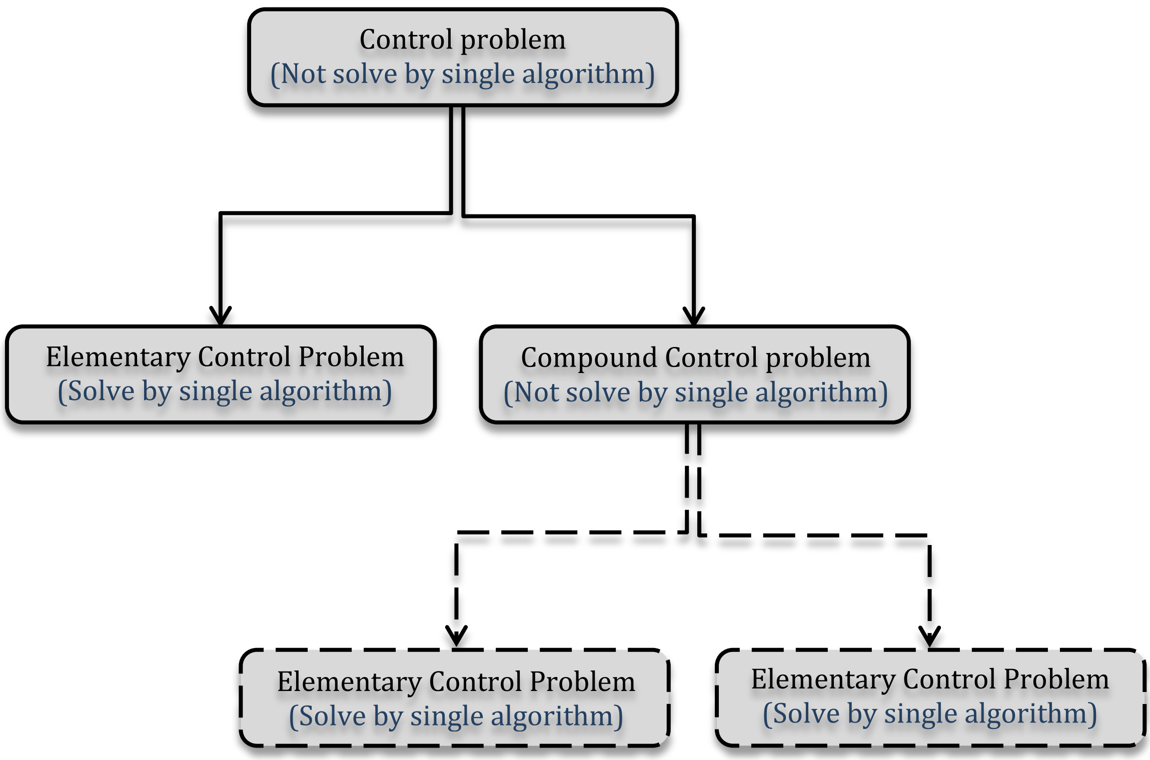


Figure 2.4 Classification of control problem

Whether a particular control problem is classified as an elementary or compound control problem not only depends on the nature of the problem, but also on the knowledge of the designer. A particular problem might be classified as an elementary control problem by an expert, while a novice would divide the problem into a set of partial problems and classify it as a compound problem.

2.3.2 Coupling relations between in control problems

The partial control problems that constitute a compound control problem can be solved with some degree of independence, since each partial problem will affect the others largely through its function, independent of the details of the mechanisms or algorithms that accomplish the function [59]. However, partial control problems are inherently coupled because they share the same parent problem. This has consequences for executing the solutions, i.e. control algorithms, of the partial problems. The type of coupling between the partial control problems determines the particular way in which the control algorithms have to be executed in order to have a well-behaved overall solution. Therefore, understanding how partial problems are coupled helps in making a decision about the way to decompose a problem and how to deal with the coupling effects. The following coupling relationships can be identified as on figure 2.5.

2.3.3 Structuring of control problem

The structure of a complex control problem is determined by the hierarchical organization of its elementary and compound control problems, as well as the coupling relationships. The process of discovering the structure of a problem is called structuring. Structuring is a modeling issue that is much related to the question of how to formulate the control problem. Therefore, the structure of a control problem is affected by design choices of the designer. The structuring problem has been given less attention than the design problem in the field of control theory. Skogestad and Postlethwaite [60] mention that one of the reasons for this lack of attention is that most control theories assume that the structure of the control problem is given at the outset of a problem. By taking into account the definition of a complex control problem, structuring a control problem consists of the following aspects:

- **Decomposing the overall problem:** The overall control problem is decomposed into a hierarchical structure of elementary and compound control problems. For each elementary or compound control problem a clear specification of the control objective in terms of the to-be-controlled variables is given. Such a specification indicates, for example, how the controller should respond to disturbances and commands.
- **Defining well-defined control problems:** For each elementary control problem a model of the plant should be derived that is used, together with the objective specification, to design a control algorithm. Because the to-be-controlled variables cannot not always be measured or directly controlled, candidate measured and controlled variables should be chosen that have a close relationship to the to-be-controlled variables.
- **Specifying coupling relationships:** Coupling relationships between the partial control problems are specified in order to take them into account when integrating the individual controllers.

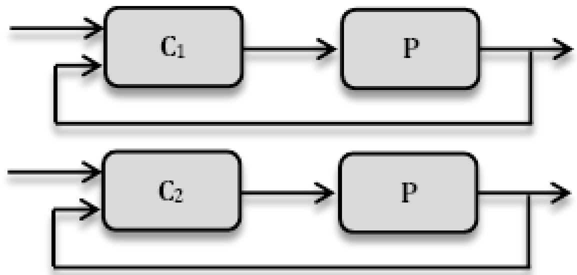
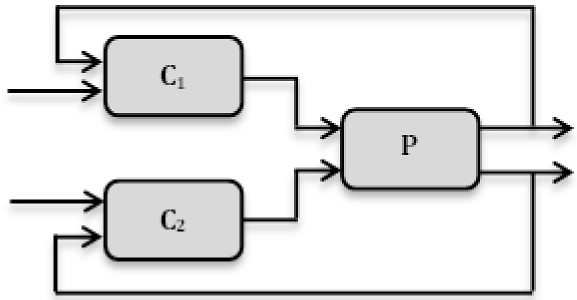
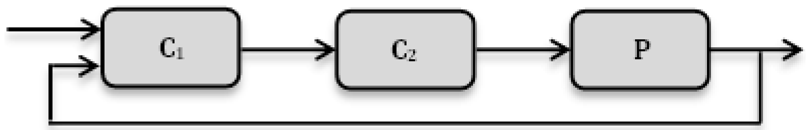
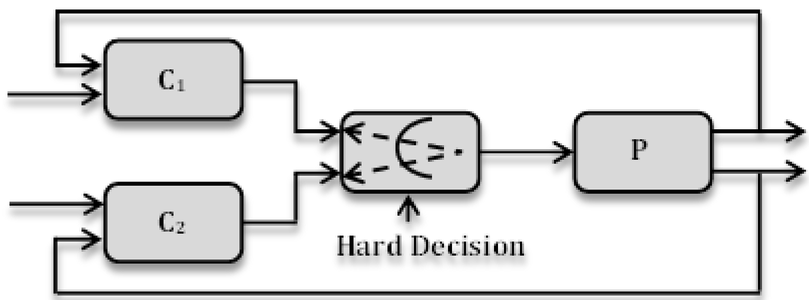
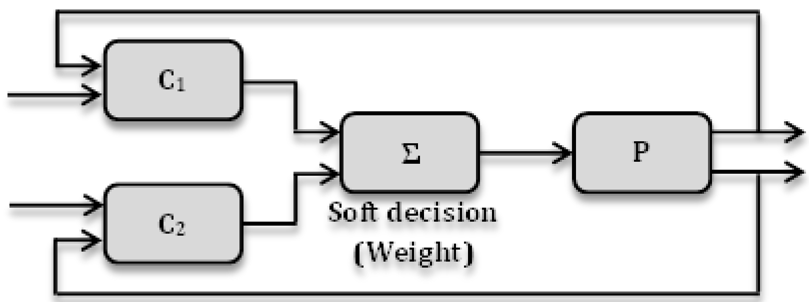
Independent Executions of C_1 are C_2 done independently.	
Indirect Dependent Executions of C_1 are C_2 done independently. The performances are influenced through control signals.	
Sub ordinal dependency Second controller is executed based on first controller output.	
Hard conditionally dependency At any moment, decision is made whether C_1 is executed or C_2.	
Soft conditionally dependency At any moment, degree of importance (weights) of control signals is determined.	

Figure.2.5 Coupling relation between control problems

2.3.4 Structuring: water tank control problem

Figure 2.6 shows a diagram of water level control in tank. The height of the vessel is limited to max_level. A corollary of this is that an overflow of the water may occur. There is a hole at the If the vessel is filled with water, the level H will drop naturally due to water leaking away through this hole. By using a pump, water can be pumped into the vessel. The pump cannot pump water out of the vessel and has an upper flow limit. A measurement of the water level H is available for feedback. The control problem consists of designing a control system that produces a control signal u in form of voltage that steers the pump, such that the water level is equal to some given reference signal R. An overflow may not occur.

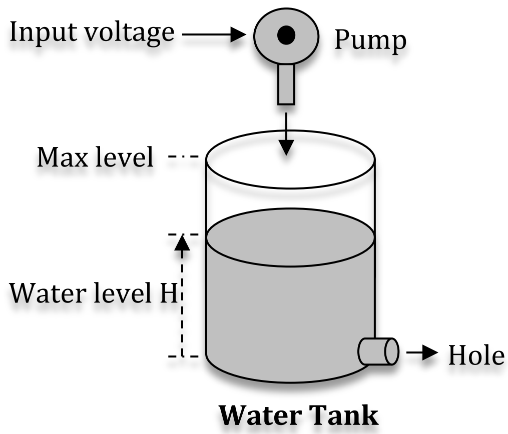


Figure 2.6 Water tank control problem

The non-linear model is:

$$\frac{d}{dt}Vol = A\frac{dH}{dt} = bV - a\sqrt{H} \qquad 2.2$$

Where Vol = Volume of water in the tank,

A = Cross-sectional area of the tank,

b = Constant related to the flow rate into the tank, and

a = Constant related to the flow rate out of the tank.

V = Input voltage of pump drive

H = Height of tank water level

The equation describes the height of water, H, as a function of time, due to the difference between flow rates into and out of the tank. It is nonlinear due to its dependence on the square-root of H. The control problem is stated as "design a controller that keeps the water level at some predefined level *r* and always prevents an overflow". This problem is either an elementary control problem, or a compound control problem. For this example, the control problem is considered as being a compound control problem, which consists of two partial control problems: "design a tracking controller that keeps the water level at the predefined level *r*" and "design a safeguard that prevents overflows". This decomposition is illustrated in Figure 2.7.

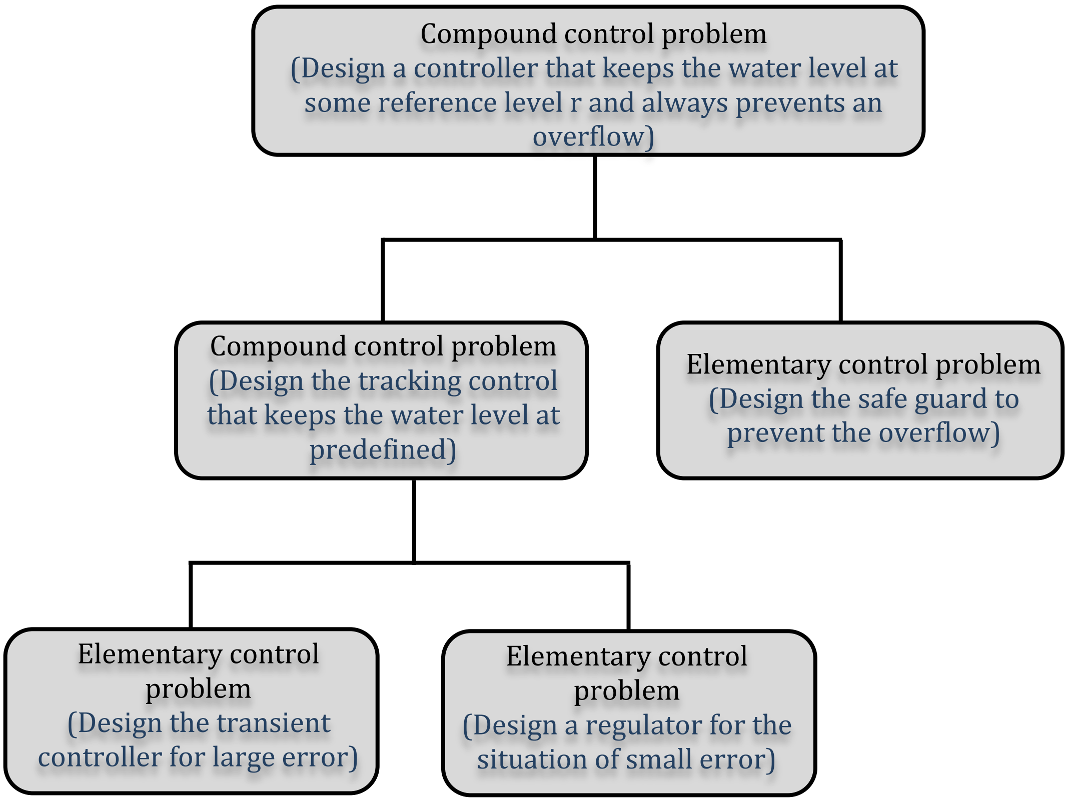


Figure 2.7 Structuring of water tank control system

The tracking control problem is also seen as a compound control problem. It consists of the partial control problems: "design a transient controller for the situation that there is a large difference between the water level *y* and the reference level *r*" and "design a regulator for the situation that the water level *y* is near the reference level *r* that means small error". Such decomposition is common in process control and may increase the performance of the control system. These two control problems are considered as elementary control problems. Because it is not clear when to switch between these two problems and because a smooth transition between them is desirable, the problems are coupled by a soft conditional dependency. By dividing the control problem in this manner, a hierarchically structured model of the overall control problem is obtained[#]. This model provides the basis to develop a solution, as will be shown in the remainder of this thesis.

[#] A paper entitled "Development and implementation multi-agent controller for SISO system" is published in International Conference on System Dynamic and Control, ICSDC-2010,MIT, Manipal, India, ISBN: 978-93-80578-58-3, August 19th-22th 2010. (Publisher: I. K. Publication, New-Delhi)

2.3.5 Structuring: Boiler turbine plant

The boiler-turbine model developed by Bell and Astrom [61] which is given in figure 2.8. The model is a 3rd order, non-linear MIMO system with hard constraints and rate limits imposed on the actuators.

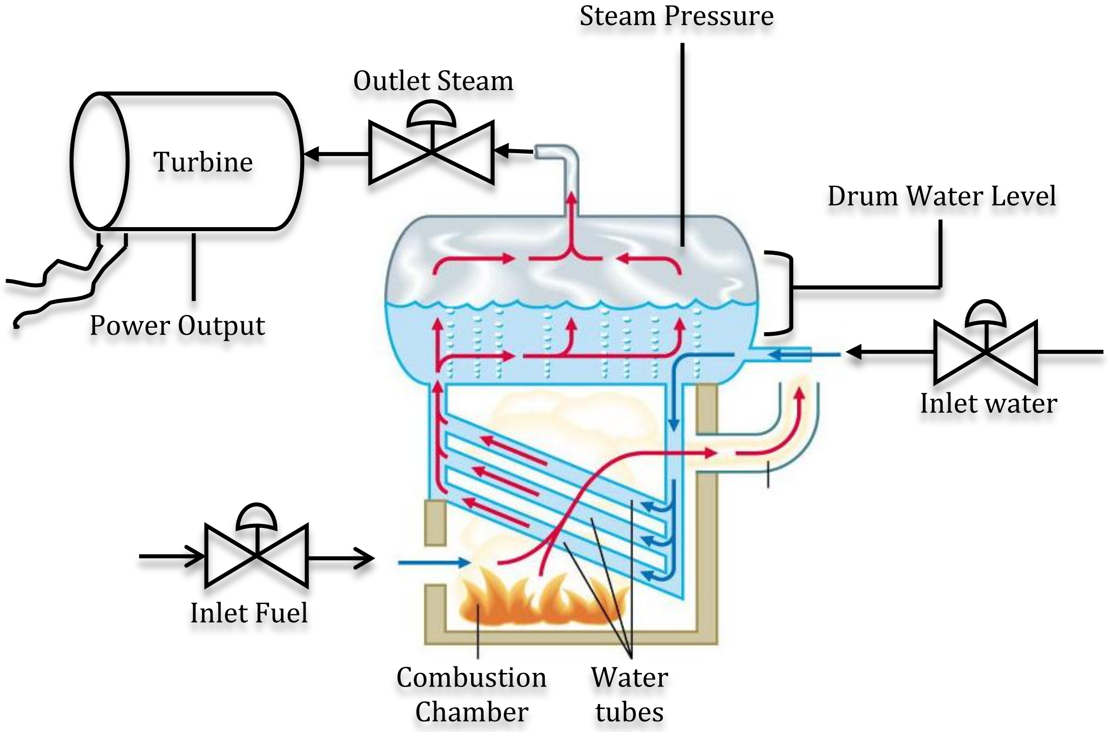


Figure: 2.8 Boiler turbine plant control system

The model is based on the boiler-turbine plant P16/G16 at the Sydvenska Kraft AB plant in Malmo, Sweden. The boiler is oil-fired and the rated power is 160 *MW.* Data acquired during a series of experiments in 1969 form the basis for the system identification. Although the model is of low order, it is capable of illustrating some of the complex dynamics system is:

$$\frac{dp}{dt} = -0.0018u_2 p^{9/8} + 0.9u_1 - 0.15u_3 \qquad 2.3$$

$$\frac{dp_0}{dt} = (0.073u_2 - 0.016)p^{9/8} - 0.1p_0 \qquad 2.4$$

$$\frac{dp_f}{dt} = \frac{(141u_3 - (1.1u_2 - 0.19)p)}{85} \qquad 2.5$$

Where p=drum pressure (Kg/cm^2), p_o=power output (MW), and p_f= fluid density (Kg/cm^3).

The normalized inputs to the system are u_1=fuel flow valve position, u_2=steam control valve position, and u_3=feed water flow valve position. The following limitations are imposed on the valves:

$$\left|\frac{du_1}{dt}\right| \le 0.007/\sec$$

$$-2/\sec \le \frac{du_2}{dt} \le 0.02/\sec$$

$$\left|\frac{du_3}{dt}\right| \le 0.05/\sec$$

And all valve position variables are constrained to lie in the interval [0, 1]. The outputs to the system are p (drum pressure), p_o (Output power) and X_w (Drum water level deviation). Pressure and power output are just the first two state variables whereas water level deviation is found through the auxiliary relationships.

$$X_w = 0.05(0.13073 p_f + 100\alpha_{cs} + \frac{q_e}{9} - 67.975) \qquad 2.6$$

$$\alpha_{cs} = \frac{(1-0.001538 p_f)(0.8p-25.6)}{p_f(1.0394-0.0012304)} \qquad 2.7$$

$$q_e = (0.854u_2 - 0.147)p + 45.59u_1 - 2.514u_3 - 2.096 \qquad 2.8$$

Where q_e=is the evaporation rate (Kg/s) and α_{cs}= is the steam quality.

From the nominal values, a lineralized model is obtained from a truncated Taylor series expansion of the non-linear equations using equations: 2.3-2.8 (Appendix A). The linear system matrices are found using $x^0 = [108, 66.65, 428]^T$ and $u^0 = [0.34, 0.69, 0.436]^T$ operating points:

$$A = \begin{bmatrix} -2.509\times10^{-3} & 0 & 0 \\ 6.94\times10^{-2} & -0.1 & 0 \\ -6.69\times10^{-3} & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0.9 & -0.349 & -0.15 \\ 0 & 14.155 & 0 \\ 0 & -1.389 & 1.659 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 6.34\times10^{-3} & 0 & 4.71\times10^{-3} \end{bmatrix}, D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.253 & 0.512 & -0.014 \end{bmatrix},$$

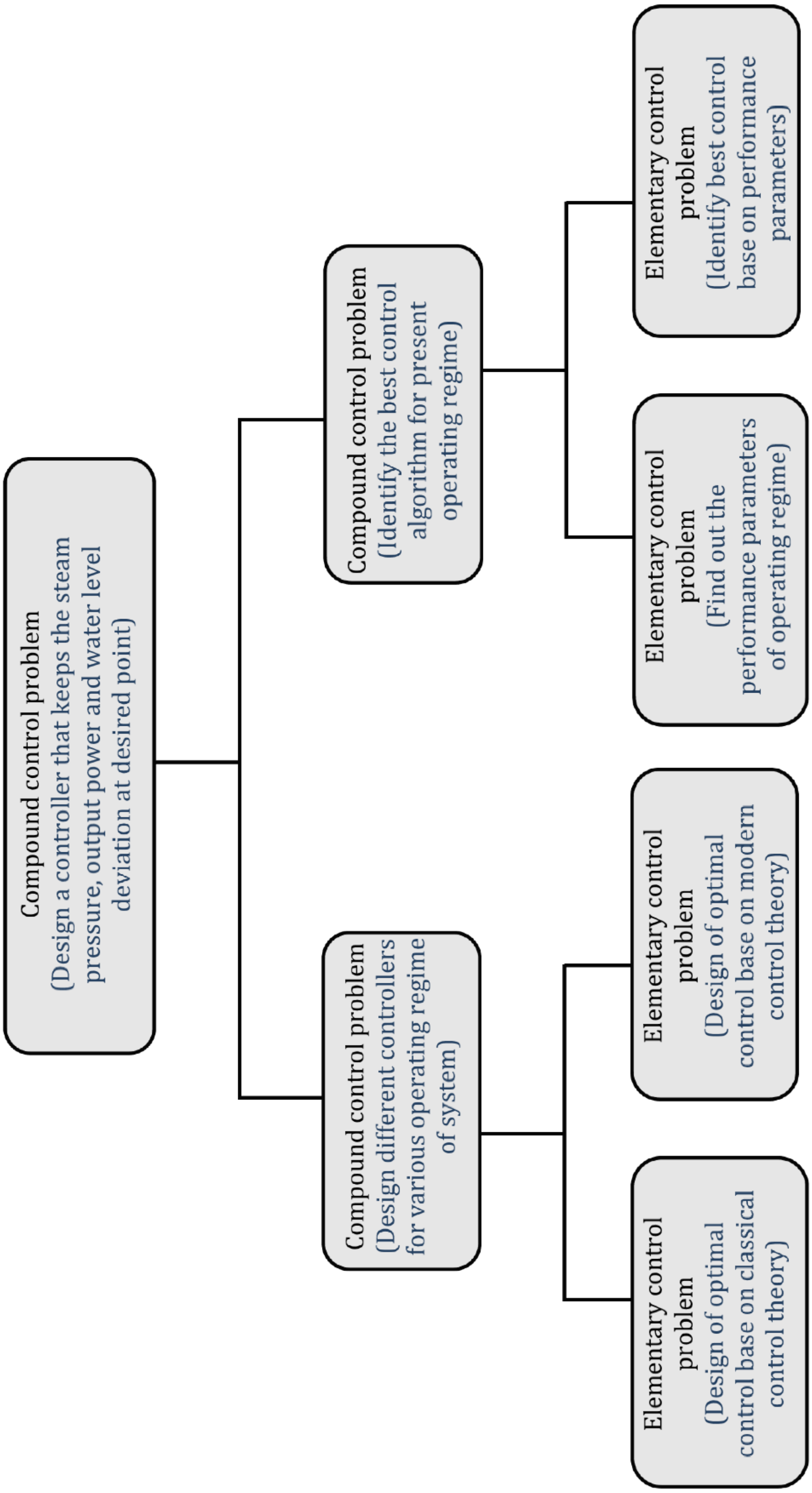


Figure 2.9 Structuring of boiler control problem

The control problem is stated as “design a controller that keeps the stream pressure, output power and water level deviation at some predefined reference points”#. This problem is either an elementary control problem, or a compound control problem. For this example, the control problem is considered as being a compound control problem, which consists of two partial control problems: first is “Design the various optimal controllers for various operating regime” and second is “Active one of best controller for particular plant model”. These both are again the compound problem. Single control algorithm is not suitable for all operating regime. First compound problem is decomposed in two parts as: first is “design an optimal controller that keeps all controlling parameter at reference point based on classical control theory”. Second is “design an optimal controller that keeps all controlling parameters at reference point based on modern control theory”. Second compound control problem is distributed in two partial problems: first is “Find out the performance parameters of plant model”. Second is “Decision making using fuzzy logic based on performance parameters”. This structure of decomposition is illustrated in figure 2.9.

2.4 Concluding remark

Because elementary control problems can generally be solved using design methods known from control theory (among others), this part is given no special attention hereafter. Rather, a method has to be found that deals with questions such as “how to combine individual solutions” and “how to deal with the coupling relationships”. Considering the nature of the iterative design process, this method also should allow for the addition, removal and modification of parts of the overall controller without affecting the remaining part. In the next chapter, the concepts of an agent and multi-agent system are discussed, which are suited for this.

 # A paper entitled “Multi-agent based controller for boiler turbine plant” is accepted to published in journal of control and intelligent after review process, ACTA press, March 2013.

Chapter 3

Multi-Agent

Chapter gives a general overview of agents and multi-agent systems, which is required to develop the multi-agent control architecture. Various multi-agent architectures are discussed in this section. The key issues of multi-agent system (MAS) are also covered in this chapter. A new MAC architecture is designed using multi-agent concepts for control engineering applications.

Agents and multi-agent systems are becoming a new way to analyze, design and implement complex systems. Specifically, when the design problem is distributed in nature, the development of a solution may benefit from an agent-based approach. The focus of an agent-based approach is on goals, tasks, communication and coordination that is concerned with structural aspects of a problem. This chapter introduces and reviews general concepts of agents and multi-agent systems, such that an agent based solution of control problems can be discussed in the last section.

3.1 Agents

3.1.1 Notion of an agent

The goal of this section is to explain what an agent is, such that this concept can be used as a modeling tool in various control engineering applications. However, introducing the concept of an agent in a precise and technical manner is a difficult thing to do, as there is no generally accepted definition of it.

Nwana [1] mentions two reasons why it is difficult to define precisely what an agent is:

- The term agent is not unique and is already widely used in everyday parlance as in travel agent, estate agent, etc.; and
- The term agent is an umbrella term for a heterogeneous body of research and development.

Despite the lack of a technical definition, researchers do have a notion of what an agent is, and is not, and are able to discuss their work with others based on this notion. A definition of this notion that captures the essential aspects of being an agent at least, what is agreed on by most researchers is given by Franklin and Graesser [2]:

Agent: An autonomous agent is a system situated within and part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.

Before further describing an agent, it must be mentioned that not having a technical definition may result in confusing situations. The term agent gets easily confused with terms such as object, actor or module. Sometimes the confusion even gets worse when terms from one particular context are compared with terms from another context. For example, the sentence an agent is an object can be both false and true, depending on what is meant with agent and object. The concept of an agent is not equal to the concept of an object. However, the concept of an agent can be realized by an object in some object-oriented programming language just as the concept of an object can be realized by procedures in a procedural programming language.

The collection of circumstances, objects, agents and conditions by which an agent is surrounded is called the environment of the agent. To realize its goal, the agent performs particular tasks in this environment. It has sensors and actuators to carry out these tasks. Sensors are used to sense the environment in order to gain information about it, and actuators are used to perform actions on the environment in order to change it. There is a continuous sense-think-act interaction between the agent and its environment. In figure 3.1, this interaction is illustrated.

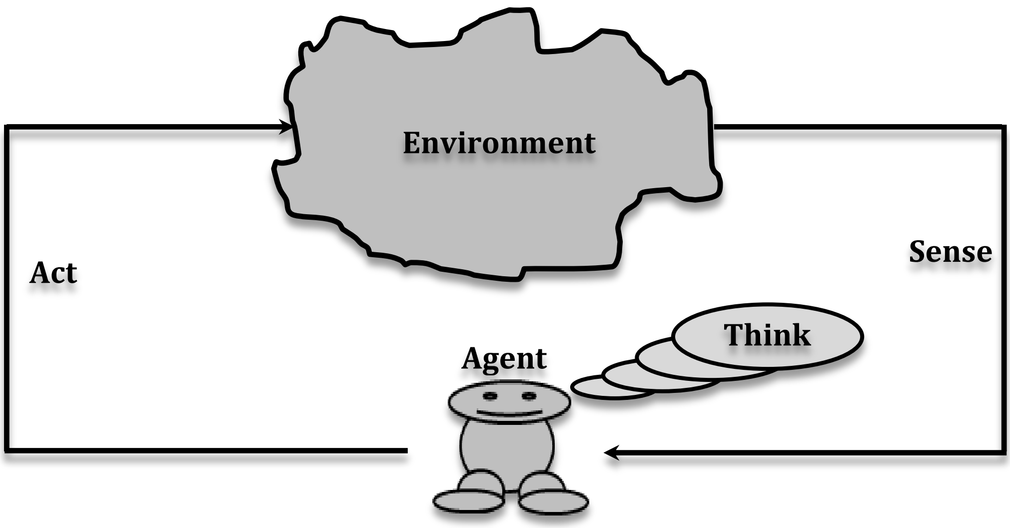


Figure 3.1 Agent entity that sense, think and act on environment

In general, an agent cannot sense all information of the environment. It is only capable of sensing a limited amount of information and therefore it has a local view of its environment. The same holds for its actions. In general, an agent is only capable of manipulating a restricted part of its environment. This makes that the term rational gets a more specific meaning; an agent acts rational if it selects the best action to achieve its goal, given its limited amount of resources and its limited view of the environment.

While realizing its goal, an agent must decide for itself at each moment which actions to perform. This decision is based on the sensed information and its internal state at that moment. The internal state of an agent captures information about the events in the environment. In addition, this decision is influenced by the goal of the agent, the skills required to accomplish its tasks and the resources that are needed. So, agents make rational decisions.

3.1.2 Agent attributes

The pervious given sketch of an agent is not sufficient to charaterise the agent in practical manner. So, Jennings, Sycara & Wooldridge [3] describe a minimum list of attributes which an entity must have in order to be denoted as agent. This list contains the following attributes:

- ***Situatedness:*** An agent is situated in some environment in which it can receive sensory input and can perform actions that change the environment in some way.
- ***Autonomous:*** Agents have control over their actions and internal state, and they are able to act without direct human intervention.
- ***Responsive:*** Agents should perceive their environment and respond in a timely fashion to changes that occur in it.
- ***Pro-active***: Agents do not simply act in response to their environment; they should be able to exhibit opportunistic, goal-directed behavior and take the initiative where appropriate.
- ***Social ability:*** Agents should be able to interact with other artificial agents or humans in order to complete their own problem solving and to help others with their activity.

3.1.3 Key motivations to use the concept of an agent

Agents are used to model and explain the behavior of complex systems, such as particular kinds of software systems [3]. In general, the functioning of these systems could also be explained without the concept of an agent. This raises the question why to use the agent concept. Somehow, there must be a clear benefit to use it. In literature several motivations are given:

- In many applications decisions have to be made whether to perform a particular action. An agent takes decisions for itself: it is a model for a decision making entity [4].
- An agent can be given a task which it realizes by taking actions and by communicating with other agents. By what means (hardware, software, communication patterns) this task is realized, and how the decisions that underly the actions are made, often is not the

first concern when developing or describing an application. Therefore, agents are used to hide implementation details [5].

- It is an intuitive and natural metaphor that can be used to describe the behavior of complex systems. The term intentional stance is used to describe the point of view from which the behavior of systems is explained based on the assumption of rationality [1].
- Using agents is a practical technique that is aimed at the creation of complex computing systems based on the concepts of agents, communication, cooperation and coordination of actions [6].

3.2 Multi-agent system

The group of agents is called as multi agent system. It may be considered as intelligent tool for the solution of such problems as planning, scheduling, decision-making and control in the framework of production processes. The MAS approach seems to be the most feasible. It respects the complicated characteristics of the goal that we aim to achieve. There are some significant reasons that motivate us to choose the MAS approach to the solution of decision-making, such as:

- **Modularity:** Each agent is an autonomous module and can work without interventions of the external world. Each agent can have different capabilities or functionalities and through cooperation the agents are able to achieve a variety of goal [4].
- **Parallelism:** The MAS approach enables to work in parallel. A complicated problem could be solved in an acceptable time by using a number of agents, e.g., gathering information from various resources allocated in different places [7].
- **Flexibility:** The MAS approach is able to react in a flexible manner to each change in the environment. Through cooperation the agents can assist each other to compensate the lack of capability or knowledge. They can share information or own capacity to resolve a newly appeared situation, if one agent is not able to do so. Beside that, each intelligent agent can do reasoning about whom and when it has to cooperate with, in order to achieve the effective performance [4].
- **Scalability:** Because agents are modular, it should be easy to add and remove agents from the MAS [7].

A MAS is an attractive concept to model particular types of distributed problems. It can be as complex as a system consisting of heterogeneous, communicating agents that should understand each other and negotiate, or as simple as a system consisting of homogeneous, non-communicating agents.

3.2.1 Design aspect

Four general design aspects of a MAS can be listed [6]:

- **Distribution of tasks:** The design aspect is concerned with defining the individual tasks of the agents. Naturally, these tasks are derived from the overall problem definition of the MAS. The activity of decomposing the overall problem in subproblems such that for each subproblem a task can be defined for an individual agent is called task decomposition.
- **Distribution of information:** The design aspect is concerned with distribution of information that mean which agent should receive which information, and which agent should create which information. The agent will take the decision based on received information. So, receiving and sending the right information is important for a proper design of the MAS.
- **Distribution of resources:** This aspect is regarding the practical limitation. Some of important limitations are the amount of computation time, communication bandwidth, computer memory, use of actuators, (environmental) space and services from other agents.
- **Distribution of coordination**: The detail discussion covered in next section.

These are illustrated in figure 3.2.

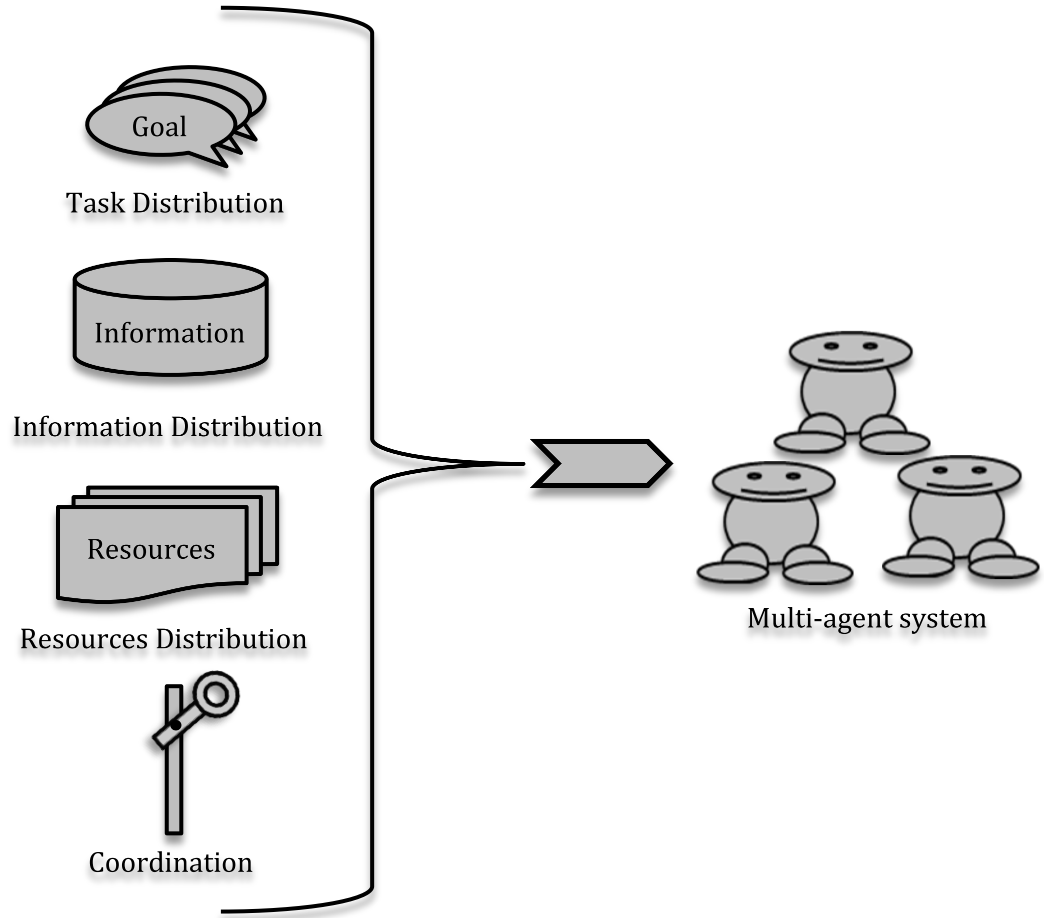


Figure 3.2 MAS with task, information, resource and coordination distribution

3.2.2 Coordination

The MAS would be a group of independently operating agents. The MAS have all the information, skills and resources to carry out its task. Agents have interdependencies because they need information or skills from each other, or because they share some common resource. In order for the society to operate coherently, the activities of the agents therefore need to be coordinated. Choosing the coordination mechanism given the interdependencies between agents in a MAS is of major importance, as the coordination mechanism may have a considerable impact upon the performance of the system [8]. Many types of coordination mechanisms are used in MAS design. They can be classified depending on the point of view [9]. The coordination mechanism can be

- **Centralized:** There is a single locus of coordination-related data / knowledge and decision-making / authory.
- **Decentralized:** Data, knowledge and decision making with respect to coordination are distributed.
- **Hybrid:** Some intermediate form of centralized and decentralized coordination.

3.2.3 MAS architecture

Several architectural styles have been used in the development of multi-agent architecture [10] are discussed as below:

- Hierarchical multi-agent systems, in which agents communicate according to a hierarchical structure, such as a tree. A system such as the open agent architecture, which uses brokers, is a hierarchical system, as each agent communicates only with a broker or facilitator agent. Shehory [11] gives the disadvantage of such a system as the reduction in autonomy of the individual agents, as lower levels of the hierarchy depend on and may be controlled by higher levels. However, hierarchical architectures can greatly reduce the amount of communications required, and also the complexity and reasoning capabilities needed in the individual agents.
- Flat multi-agent systems, in which any agent may contact any of the others. These provide the greatest agent autonomy, but result in more communications between agents. Also, agents in a flat structure must either know the locations of their communications partners, or be provided with agent location mechanisms such as yellow pages services. Many smaller multi-agent systems appear to be developed using a flat organization.
- A subsumption multi-agent system is a system in which agents are themselves made up of other agents. In this system, the subsumed agents are completely controlled by the containing agents. This is similar to the subsumption architecture for an individual

agent. According to Shehory, the fixed structure of a subsumption multi-agent system provides efficiency but restricts the flexibility of the system.

- A modular multi-agent system is comprised of number of modules. Each module normally employs a flat structure, while intermodule communications is relatively limited. A modular multi-agent system might be useful in a situation such as power system automation, in which each substation could be categorized as a single module. Most communications within a power system are either within a substation or between a substation and the control centre, and so this might be an appropriate multi-agent system structure.

3.3 Key issue in MAS technology

Because MAS comprises many different aspects, several different technologies are needed for the development and realization of MAS.

3.3.1 Agent architecture

- **Deliberative architecture:** There are several different architectures for intelligent agent implementation. A well-known cognitive architecture is the Belief-Desire-Intention (BDI) architecture [12, 13], in which the agent's knowledge base is described by a set of beliefs (those facts which an agent considers to be true), desires (those conditions which the agent wishes to bring about), and intentions (actions which the agent has committed to perform). On each execution cycle of the interpreter, the agent retrieves new events from the environment. It then generates a set of options, which are plans or procedures that the agent is capable of carrying out, both in response to events and in order to achieve its goals. The agent will then execute, or partially execute, one or more of the selected options. This process is repeated for the agent's lifetime.
- **Reactive Architecture:** The subsumption architecture [14] is an example of a reactive architecture which does not employ an explicit knowledge representation. A subsumption agent consists of a number of concurrently-executing behaviors [15]. These are arranged in a number of layers, with lower layers representing simpler behaviors, which have a high priority, and higher layers representing more abstract behaviors, and having lower priority. Low-level behaviors are unaware of the presence of the high-level behaviors. It is therefore possible to construct an agent using the subsumption architecture starting with the lowest-level behavior and working upwards, with the agent being functional, at least to a certain extent, after each layer is constructed.

- **Learning-based Architecture:** Learning-based architectures may be used to enhance the performance of an agent through various techniques such as reinforcement learning, genetic programming, neural networks or fuzzy logic programming [16]. While it is possible to use learning to improve the capabilities of an agent using architecture such as BDI, it is also common to incorporate learning into much simpler agent architecture. Learning agents have been applied in a number of domains. It is very popular agent architecture for control engineering application.
- **Layered Architecture:** According to Ferber [6], the advantage of a layered architecture is that a layered agent, by having different levels of behavior operating concurrently, is capable of reacting to changing circumstances while planning its future actions and reasoning about the behavior of other agents. Both of the architectures mentioned have three layers: TouringMachines has a reactive layer, modeling layer and planning layer, while INTERRAP has a behavior-based layer, local planning layer and cooperative planning layer. In TouringMachines, all three layers are connected to the agent's sensors and effectors. The three layers operate concurrently and are unaware of each other, while a control mechanism is used to filter the inputs and outputs and prevent conflicts. In INTERRAP, the sensors and effectors are connected only to the lowest layer (the behavior-based layer). Activation requests are passed upward through the layers, and commitments are passed downwards. Unlike the subsumption architecture (which is a form of layered architecture), both TouringMachines and INTERRAP are based on explicit knowledge representation. Feber has discussed various nine architectures, but the important architectures are discussed here.

3.3.2 Programming language

The development of an agent-based system would benefit from a programming language that allows one to program a system using agent and multi-agent system based concepts. Up to now a various and wide range of environments have been developed for design, modelling and simulation of agents#. Some examples are as follows:

- **Agent0:** an old agent language and interpreter built to run on top of Common Lisp [17].
- **April**: a process oriented language which act the processes communicate with each other in a multi-agent system [18].

 # Paper entitled "Multi-agent system: A recent trends in control engineering". Control, Microprocessor, Electronics & Communication CMEC-2011, Institution of electrical and electronics engineers, Vadodara, February 20th 2011.

- **JADE (Java Agent Development Environment):** a suitable environment for modelling agents using Java as programming Language and supporting FlPA standards for message passing [19].
- **Xraptor**: a simulator for continuous multi-agent systems that is written in C++ [20].
- **Agent Building Environment**: a toolkit for software developers that makes it easy to build an application based on agents or to add them to an existing application based on C++ and JAVA.
- **Oz Programming System**: a concurrent constraint programming language designed for applications that require complex symbolic computations, organization into multiple agents, and soft real-time control.
- **Agents Kernel Language (AKL)**: concurrent constraint programming language developed at the Swedish Institute of Computer Science (SICS).
- **Fresco:** CORBA-based language implemented in C++ which provides support for object distribution, a standard high-level notation for object definition, Tcl-based scripting and multi-threading.
- **Matlab:** particularly important for research purposes, because it provides almost unlimited flexibility for agent programming and utilization of direct commands [21].

Some special purpose environments are also developed for agent-based design, modelling and simulation in various fields. In the domain of control engineering, modelling and simulation packages such as Simulink and 2Osim are available. The recent year object oriented technology are strongly preferred in agent based environments because of their popularity of UTMAC, an object oriented C^{++} library for implementation and simulation of multi agent controller systems which provides several classes for implementation controller-agents with a simple.

3.4 Multi-agent control

3.4.1 Multi-agent control framework

The field of MAS presented previous, provides concepts to discuss the problem of integrating locally operating entities. It therefore provides a basis for developing a framework to handle the integration problem of local controllers. In order to develop this agent-based framework it will appear that three questions need to be addressed:

1. Which agent ***architecture*** is needed to implement solutions of partial control problems?

   It is concerned to design of agent so; it can be achieved the controller objective. The task of agent is same kind of controller so, it is known as **contro**ller plus a**gent**, **Controgent**. It is discussed in section 3.4.2.

2. How can ***coordination*** be used to deal with the dependencies between partial control problems?

   It is concerned to design of **coordination object** with respect of coupling relations between partial control problems. Detail discuss is covered in section 3.4.3.

3. How can a coordinated group of agents be used as one unit in another group, such that a ***hierarchically organized*** solution is obtained?

   It is concerned to design of **Multi-Agent Controller (MAC)** architecture. It is given in section 3.4.4.

3.4.2 Design of controgent

A controller is designed with agent concept for local operating regime, it is known as **Contro**ller + A**gent** =Controgent. It is a largely autonomous, locally operating controller that consists of a control algorithm, specific operating regime and an interface to coordinate its behavior in order to handle dependencies among other controgents.

The interface of a controgent is made up of its inputs and outputs, and its activation signals. The inputs and outputs are used to receive data from sensors, send data to actuators and to communicate with other controgents. The activation request is used to coordinate the behavior of the controgent with other controgents of the overall controller.

Figure 3.3 shows the symbol of a controgent together with its inputs, outputs, and activation signal. A controgent is labeled by a *name*, which usually reflects its objective. Inputs are pictured by a white and outputs by a black box. The activation signals are pictured by using a dashed arrow, which points to the object receiving the signal.

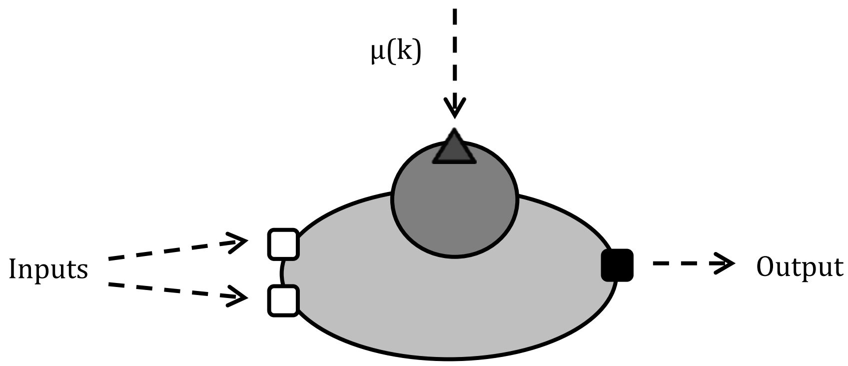


Figure 3.3 Symbol representation of controgent

3.4.3 Coordination of controgent

A controgent was proposed to implement the local controller algorithm. The mechanism is required to activate the controgent based on their intentions. This mechanism will be known as coordination object.

Coordination object: A coordination object solves dependencies between controgents based on their intentions. It calculates acknowledge signals by using a so called decide function. If necessary, it can blend the outputs of the controgents into an overall output, by using a combine function.

In the field of multi-agent systems, the design of a coordination mechanism is handled as a separate issue. Malone and Crowston [22] consider coordination as the process of managing dependencies between activities. They suggest to select a coordination mechanism based on a characterization of the dependencies between the activities. In Section 2.3.3, several coupling relationships were identified that may exist between partial control problems. Because controgents solve partial control problems, these dependencies also exist between the controgents. Therefore, the selection of a coordination mechanism to coordinate controgents should be based on these coupling relationships. In table 3.1, several coordination mechanisms are suggested for particular types of dependencies. These mechanisms are discussed in more detail hereafter.

Table 3.1 Coupling relation and coordination mechanics

Coupling relations	Coordination mechanics
Independent	Parallel
Indirect dependency	Parallel
Subordination dependency	Master-slave
Hard conditional dependency	Competitive
Soft conditional dependency	Cooperative

Figure 3.4 shows three different realizations of a coordination object. The first setup uses only the intentions of the controgents. Based on these intentions, the coordination object sends activation signals to the controgents to indicate whether they may become or stay active or inactive. The second setup is an extension of the first setup, as now the outputs of the controgents are blended into an overall output. Finally, in the third setup the coordination object has inputs to receive information, which is used to decide which controgent to activate or inactivate. This setup is similar to the supervisory architecture with a subtle difference.

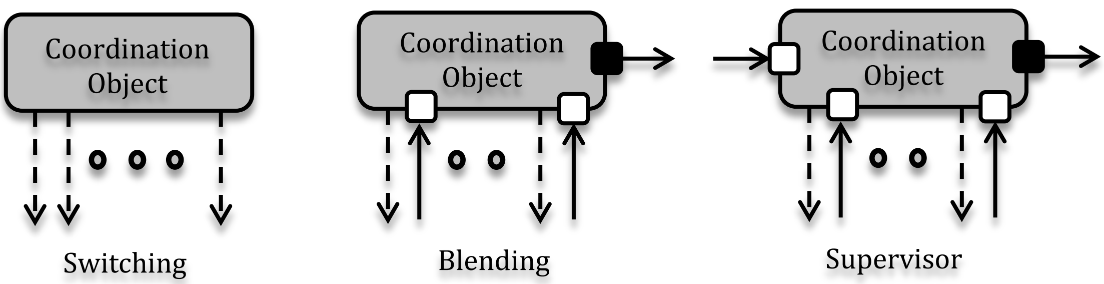


Figure 3.4 Three different setups of coordination mechanisms

These three realizations form the basis for several coordination mechanisms that can handle dependencies between partial control problems. They are discussed below.

Parallel: There may be situations in which no special coordination is required. That is, the partial control problems the controgents solve are independent or indirect dependent. However, for the sake of generality, the proposed organization is maintained and a "Parallel" coordination mechanism is provided. This coordination mechanism implements a switching coordination object.

Master-Slave: Master-Slave coordination mechanism is used in situations in which a subordination dependency exists between two partial control problems. .E.g., one controgent implements the solution of calculating reference signals needed by another controgent. The first one is called the master controgent, the latter the slave controgent. The slave controgent depends on the master; because it can only determine its intentions after the master controgent has calculated the reference signals. The Master-Slave coordination mechanism implements a switching coordination mechanism.

Competitive: Competitive coordination mechanisms are used in situations in which a hard decision has to be made between controgents. This means that given a set of controgents that wants to get active simultaneously, there is only one controgent that will get a positive acknowledge. There are several competitive coordination mechanisms:

- *Priority based:* Each controgent has a priority that is used to decide whether a controgent may get active. Of the controgents that want to get active, the one with the highest priority gets selected. The priority can be determined in several ways;
  - *Fixed Priority:* Each controgent is given a priority that is fixed during the operation of the overall controller.
  - *Dynamic Priority:* Each controgent determines its own priority based on the current operating conditions of the controlled system. The priority level can be encoded in the activation signal by using a distance-based activation request function.
- *Temporal order based:* Based on the order in time in which the controgents want to become active, it is decided which controgent gets activated. Some strategies are: **–**
  - *First Stays Active:* The first controgent that wants to become active becomes active. It stays active until it decides by itself to get inactive. No other controgent can become active until the first becomes inactive.

- *Last Stays Active:* The last controgent in time which wants to become active becomes active. If there is a previously active controgent, this controgent is inactivated.
- *Sequential:* The controgents within a group may only become active in a particular order. That is, first the first controgent may become active. Only after the first controgent has been activated, the second controgent may become active. This pattern repeats until the last controgent in the group has been activated. Then none of the controgent may become active.
- *Cyclic:* This is the same as sequential, but after the last controgent has been active, the first controgent may get active again.

Cooperative: Cooperative coordination mechanisms are used in situations where soft decisions must be made between controgents. This means that control actions of different controgents have to be blended somehow. There are several cooperative coordination mechanisms:

- *Addition:* Each controgent may become active independently from the other controgents. The control-actions of the active controgents are added. The control action of the group of controgents becomes:

$$o(t) = \sum_{i \in A} o_i(t) \qquad 3.1$$

  Where *A* is the index set of active controgents at a particular moment *t*, *n* is the total number of controgents, and the output of the *ith* controgent

- *Weighted Addition:* This coordination mechanism is the same as the addition coordination mechanism, but the control actions of the controgents are multiplied by the activation request signal, that is:

$$o(t) = \sum_{i \in A} \mu_i(t) o_i(t) \qquad 3.2$$

  Where *A* is again the index set of active controgents and μ_i the activation signal.

- *Normalized Weighted Addition*: This coordination mechanism is the same as the Weighted Addition coordination mechanism, but the overall weighted sum of the individual control actions is divided by the sum of the activation signals, that is:

$$o(t) = \frac{1}{\sum_{i \in A} \mu_i(t)} \sum_{i \in A} \mu_i(t) o_i(t) \qquad 3.3$$

- *Fuzzy Logic Addition:* This coordination mechanism is the same as the Normalized Weighted Addition coordination mechanism, but there is a restriction on the activation function of the controgents: it must be a fuzzy set.

Supervisor: the previously described coordination mechanisms all decide which controgent(s) to activate or inactivate based purely on the intentions of the controgent(s). Sometimes, it may be desirable to have a coordination mechanism that imposes its own "intentions" onto the controgents. For instance, this might be the case for situations in which the operating regimes of the controgents can only be determined in a centralized way. This will in general require that the coordination object has knowledge about particular variables in the overall controller (see figure 3.6). Such a coordination object acts like a supervisor, with the restriction that it has to take into account the intentions of the controgents. There are several supervisor-like coordination mechanisms.

- *Switch:* A Switch is a coordination object that can be set by some index signal (for instance, provided by a master controgent) and it determines the index of the (set of) controgent(s) that may become active.
- *Timing Diagram:* A Timing Diagram contains a list of starts and stop instances of the controgents.
- *State Transition Diagram (STD)*: A STD contains a set of states in which particular combinations of controgents may be active, and the allowed transitions between these states.
- *Flowchart:* A Flowchart contains a decision tree which determines the order in which the controgents may become active.
- *Scheduler / Planner:* A Scheduler determines the order in which the controgents should be active during operating time of the overall controller. This might be needed if a particular order cannot be determined a priori.

3.4.4 Multi-agent controller

A group of controgents coordinated by an extended coordination object will be called a multi-agent controller. Figure 3.7 shows the internal architecture of a multi-agent controller. It consists of a *pool* of controgents and a coordination object. The general definition of multi-agent controller is as below:

Multi-agent controller: A multi-agent controller is a controller that is implemented as a group of coordinated controgents. The coordination object solves the dependencies between the controgents, and realizes an interface to coordinate the group with controgents outside the group.

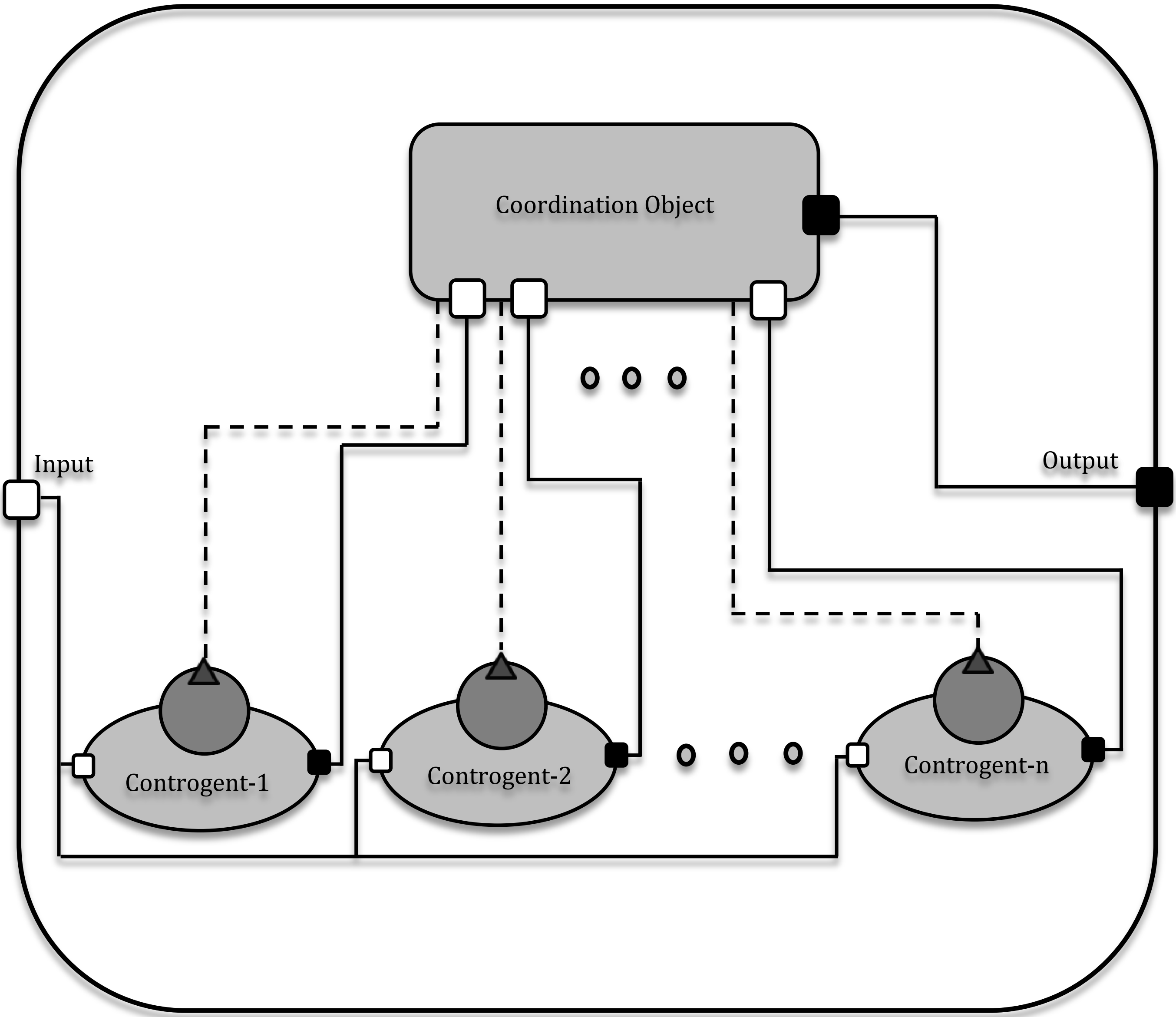


Figure 3.5 Multi-agent control architecture

3.5 Concluding remarks

The purpose of this chapter was to introduce concepts from the field of agent and multi-agent systems, such that it provides a language to discuss and model distributed problems. In particular, these concepts should be used in the context of control engineering problems. Each partial problem is represented by a separate agent. In general, there are dependencies between the partial problems. Therefore, there will be dependencies between the agents. Coordination was described as the mechanism to solve these dependencies in multi-agent systems. So, the controgent structure is designed and discussed with help of agent concept. The multi-agent control architecture is designed with multi-agent system for control engineering. Various multi-agent controller architecture based on applications are discussed in next chapters.

Chapter 4

Soft Computational Paradigms

Chapter gives a brief overview and describes theoretical background of the computing techniques such as Fuzzy logic, neural network and genetic algorithm. The most popular tools used by the researchers for development and simulation study of the system under test such as MATLAB, SIMULINK and associated tool boxes for software development and testing are also described. Toolboxes available for deploying soft computing techniques in MATLAB and used in our research work for the design and testing of proposed techniques are described in detail.

4.1 Introduction

Soft Computational Paradigms (SCP) is a successor of artificial intelligence. SCP relies on heuristic algorithms such as in fuzzy systems, neural networks, and evolutionary computation. They are combination of learning, adaptation, and evolution used to intelligent and innovative applications. It is closely associated with soft computing a combination of artificial neural networks, fuzzy logic and genetic algorithms, connectionist systems such as artificial intelligence, and cybernetics.

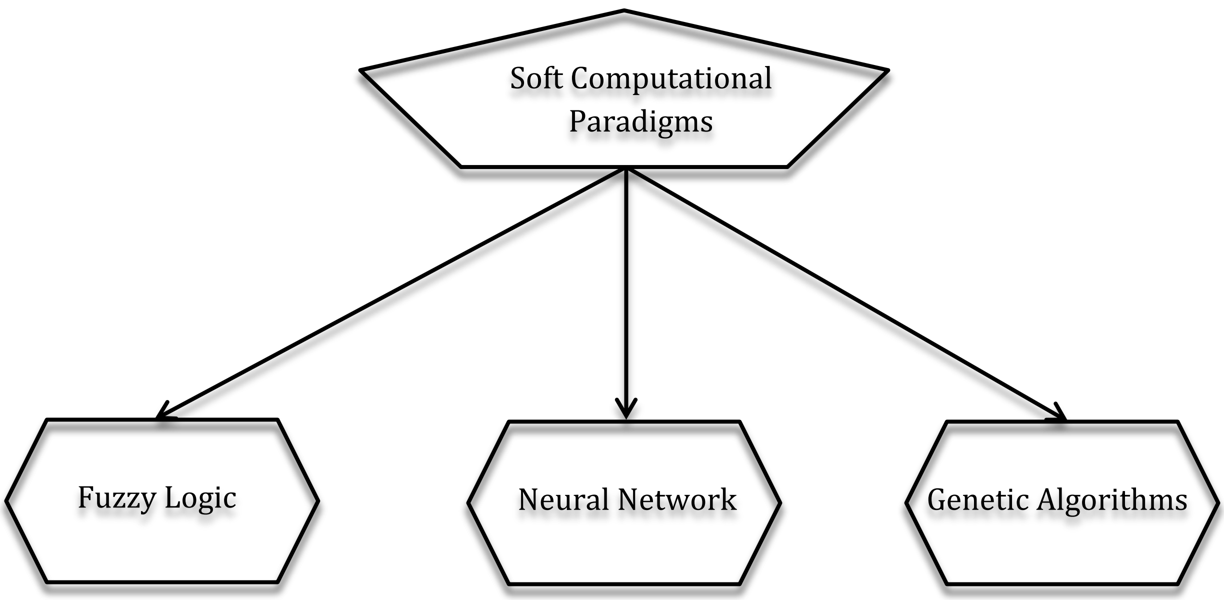


Figure 4.1 Soft computational paradigms

Arguably, SCP comprises of those paradigms in Artificial Intelligence (AI) that relate to some kind of biological or naturally occurring system. General consensus suggests that these paradigms are neural networks, genetic algorithms and fuzzy systems. Neural networks are based on their biological counterparts in the human nervous system. Similarly, genetic algorithm draws heavily on the principles of Darwinian evolution observed in nature. Finally, human reasoning using imprecise, or fuzzy, linguistic terms is approximated by fuzzy systems.

Figure 4.1 shows these three primary branches of SCP and illustrates that hybrids between the various paradigms are possible. More precisely, SCP is described as the study of adaptive mechanisms to enable or facilitate intelligent behavior in complex and changing environments. There are also other AI approaches, that satisfy both this definition as well as the requirement of modeling some naturally occurring phenomenon, that do not fall neatly into one of the paradigms mentioned thus far. This chapter describes these paradigms of SCP briefly. Also, the development tools of methods are covered in detail.

4.2 Fuzzy logic

The idea of fuzzy logic was invented by Professor L. A. Zadeh of the University of California at Berkeley in 1965 [1]. This invention was not well recognized until Dr. E. H. Mamdani, who is a professor at London University, applied the fuzzy logic in a practical application to control an automatic steam engine in 1974 [2], which is almost ten years after the fuzzy theory was invented. Then, in 1976, Blue Circle Cement and SIRA in Denmark developed an industrial application to control cement kilns [3]. That system began to operation in 1982. More and more fuzzy implementations have been reported since the 1980s, including those applications in industrial manufacturing, automatic control, automobile production, banks, hospitals, libraries and academic education. Fuzzy logic techniques have been widely applied in all aspects in today's society.

Fuzzy logic idea is similar to the human being's feeling and inference process. Unlike classical control strategy, which is a point-to-point control, fuzzy logic control is a range-to-point or range-to-range control [4]. The terminology is given below.

4.2.1 Keywords terminology

- **Fuzzy logic:** Fuzzy logic can be conceptualized as a generalization of classical logic which is also sometimes called diffuse logic; there are not just two alternatives but a whole continuum of truth values for logical propositions.
- **Fuzzy sets:** Fuzzy set is a mathematical set with the property that an object can be a member of the set, not a member of the set, or any of a continuum of states of being a partial member of the set.

- **Linguistic Variables:** Linguistic variables are the input or output variables of the system whose values are words or sentences from a natural language, instead of numerical values. A linguistic variable is generally decomposed into a set of linguistic terms.
- **Fuzzy Rule:** Fuzzy rules have been advocated as a key tool for expressing pieces of knowledge in "fuzzy logic". Fuzzy if/then rules are rules whose antecedents, consequences or both are fuzzy rather than crisp.
- **Conclusion:** Fuzzy output in terms of linguistic variable is called conclusion.
- **Universe of Discourse:** The Universe of Discourse is the range of all possible values for an input to a fuzzy system.
- **Crisp value:** A crisp variable is the kind of variable that is used in most computer programs: an absolute value. It is called crisp value.
- **Membership function:** The membership function describes the membership of the elements of the base set in the fuzzy set, whereby a large class of functions can be taken. Reasonable functions are often piecewise linear functions, such as triangular or trapezoidal functions. Various types of membership functions are as below:
  1. Triangular Function
  2. T –Function
  3. S-Function
  4. Trapezoidal Function
  5. Gaussian Function
  6. Exponential Function
- **Fuzzification:** Fuzzification is the first step in the fuzzy inference process which involves a domain transformation where crisp inputs are transformed into fuzzy inputs.
- **Fuzzy Inference:** The fuzzy inference process consists of two phases:
  1. The composition of rules applied.
  2. Obtaining crisp output from fuzzy inference.

  Task of meaningful combination of control actions prescribed by different rules activation or firing is called as composition.
- **Defuzzification:** Defuzzification is the process by which a fuzzy consequent is reduced to a singleton or crisp scalar value in order to provide an interface to a typically scalar ‘action’. That is to say, actions are typically associated with a scalar defining a specific measurable property.

4.2.2 Fuzzy Inference System (FIS)

A Fuzzy Inference System (FIS) consists of four main parts: fuzzification, rules, inference engine, and defuzzification. These components and the general architecture of a FIS is shown in figure 4.2. The knowledge base contains information about the boundaries of the linguistic variables data base. Fuzzification interface receives the crisp I/P value, maps it to a suitable domain i.e. it converts the value to a linguistic term/fuzzy set. Inference determines the O/P value from the measured I/P according to the knowledge base. Defuzzification interface has the task of determining the crisp O/P value. The fuzzy logic is designed using following algorithm.

- Define the linguistic variables and terms (initialization)
- Construct the membership functions (initialization)
- Construct the rule base (initialization)
- Convert crisp input data to fuzzy values using the membership functions (fuzzification)
- Evaluate the rules in the rule base (inference)
- Combine the results of each rule (inference)
- Convert the output data to non-fuzzy values (defuzzification)

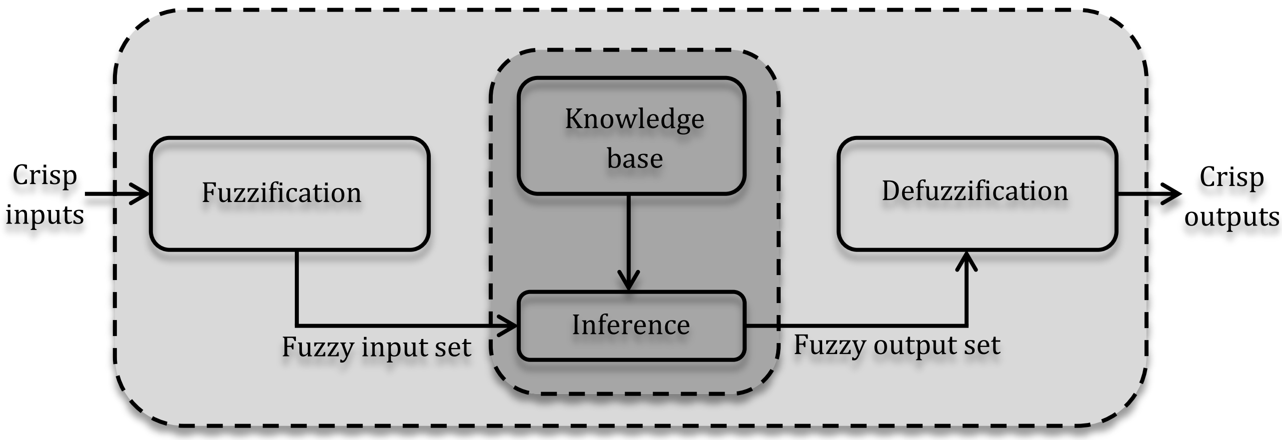


Figure: 4.2 Fuzzy Inference System (FIS)

4.3 Artificial Neural Network

The human brain is comprised of many millions of interconnected units, known individually as biological neurons. Each neuron consists of a cell to which is attached several dendrites and a single axon. The axon connects to many other neurons via connection points called synapses.

Artificial Neural Networks (ANNs) attempt to emulate their biological counterparts. First time, Mcculloch and Pitts [5] proposed a simple model of neuron. After, day by day, number of various designs developed by researchers in this technique. One of the first applications of this technology for control purposes was by Widrow and Smith [6]. They developed an Adaptive LINear Element (ADLINE) that was used to control an inverted pendulum application. The back propagation training was investigated by Werbos [7] and further improved by Rumelhart [8] and others, leading to concept of the Multi-Layer Perceptron.

Some important merits of artificial neural networks in intelligent control are:

- Learn from experience rather than by programming.
- Fast and can be implemented in real-time.
- Ability to generalize from given training data to unseen data.
- Fail 'gracefully' rather than 'catastrophically'.

4.3.1 Basic artificial model

The basic model of a single artificial neuron consists of a weighted summer and an activation function as shown in figure 4.3. The neuron in the jth layer, where

$x_{1,}\, x_{2\dots}\, x_i$ are inputs

$w_{j1,}\, x_{j2\dots}\, x_{ji}$ are weights

b_i is a bias

f_i is the activation function

b_i is the output function

The weighted sum S_j is therefore

$$S_j(t) = \sum_{i=1}^{N} w_{ji} x_i(t) + b_j \qquad 4.1$$

The metrics from is

$$S_j(t) = w_j x + b_j \qquad 4.2$$

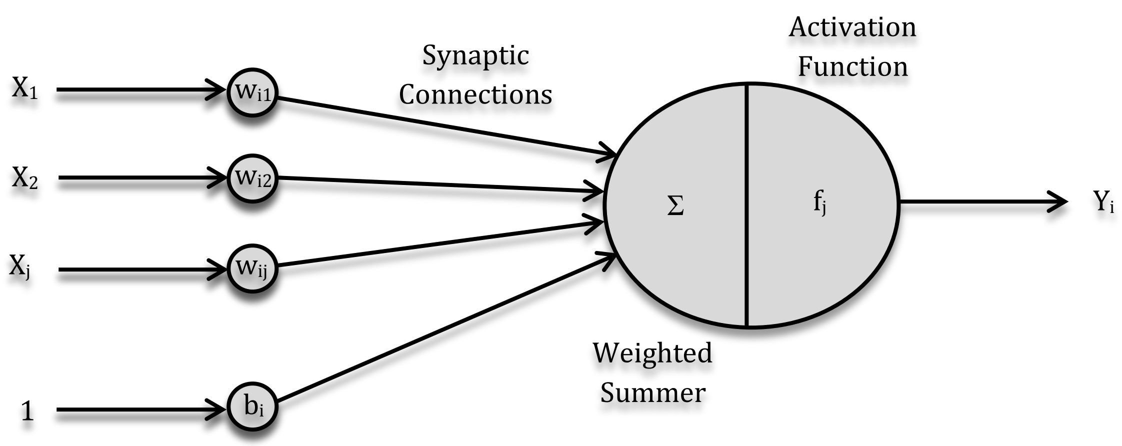


Figure 4.3 Basic neuron model

The activation function f(s) can take many forms as unit step, linear, hyperbolic tangent, sigmoid etc…The sigmoid activation function is popular for neural network applications since it is differentiable and monotonic, both of which are a requirement for the back propagation algorithm. The equation for a sigmoid function is

$$f(s) = \frac{1}{1+e^{-sj}} \qquad 4.3$$

4.3.2 Network architectures

- **Feed forward neural network**

An ANN is a network of single neurons jointed together by synaptic connections. In single layer neural network, the output signals of the neurons in the first layer are the output signals of the network. Multi-layer neural network has more layers of nodes between the input and output nodes.

The three-layer feed forward network shown in figure 4.4 consists of a three neuron input layer, a two neuron output layer and three neuron intermediate layers, called a hidden layer. This is generally called a fuzzy connected multilayer network, and there is not restriction on the number of neurons in each layer, and no restriction on the number of hidden layers.

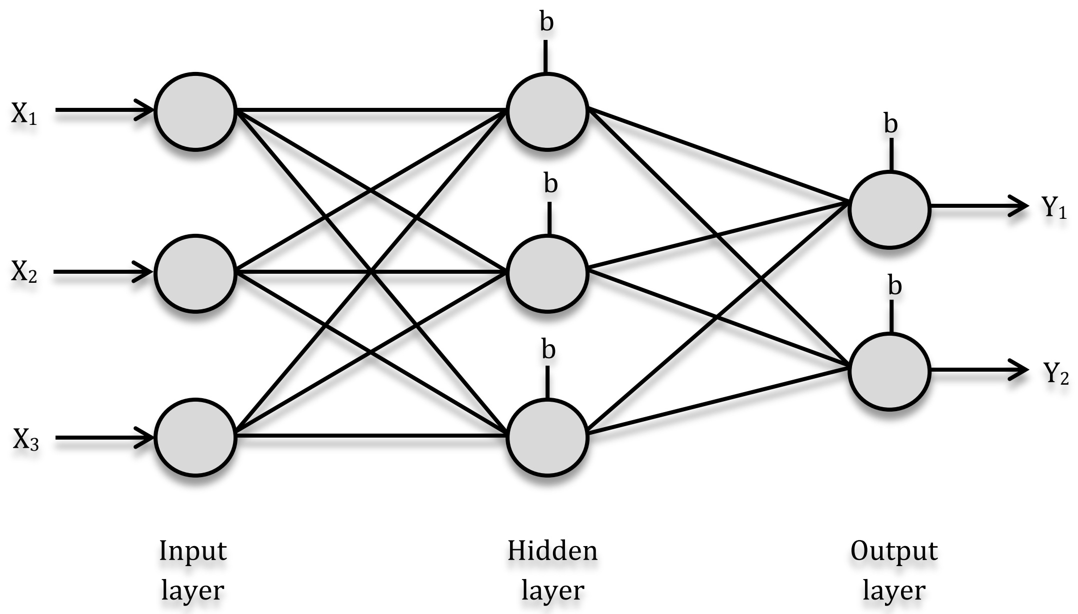


Figure: 4.4 three layer feed-forward network

- **Feedback (recurrent) networks**

Recurrent networks are based on the work of Hopfield and contain feedback paths. The recurrent network with delay is shown in figure 4.5. If the inputs occur at time (kT) and the outputs are predicted at time(K+1)T, then the network can be represented in matrix form by

$$y((k+1)T) = w_1 y(kT) + w_2 x(kT) \qquad 4.4$$

4.3.3 Learning algorithm

Learning in the context of a neural network is the process of adjusting the weights and biases in such a manner that for given inputs, the correct responses or outputs are achieved. Learning algorithms include:

- **Supervised Learning**

In supervised learning, the actual output of a neural network is compared with the desired output. Weights, which are randomly set to begin with, are then adjusted by the network so that the next iteration, or cycle, will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until acceptable network accuracy is reached. With supervised learning, the artificial neural network must be trained before it becomes useful. Training consists of presenting input and output data to the network. This collection of data is often referred to as the training set. A desired output or a target output is provided for every input set of the system.

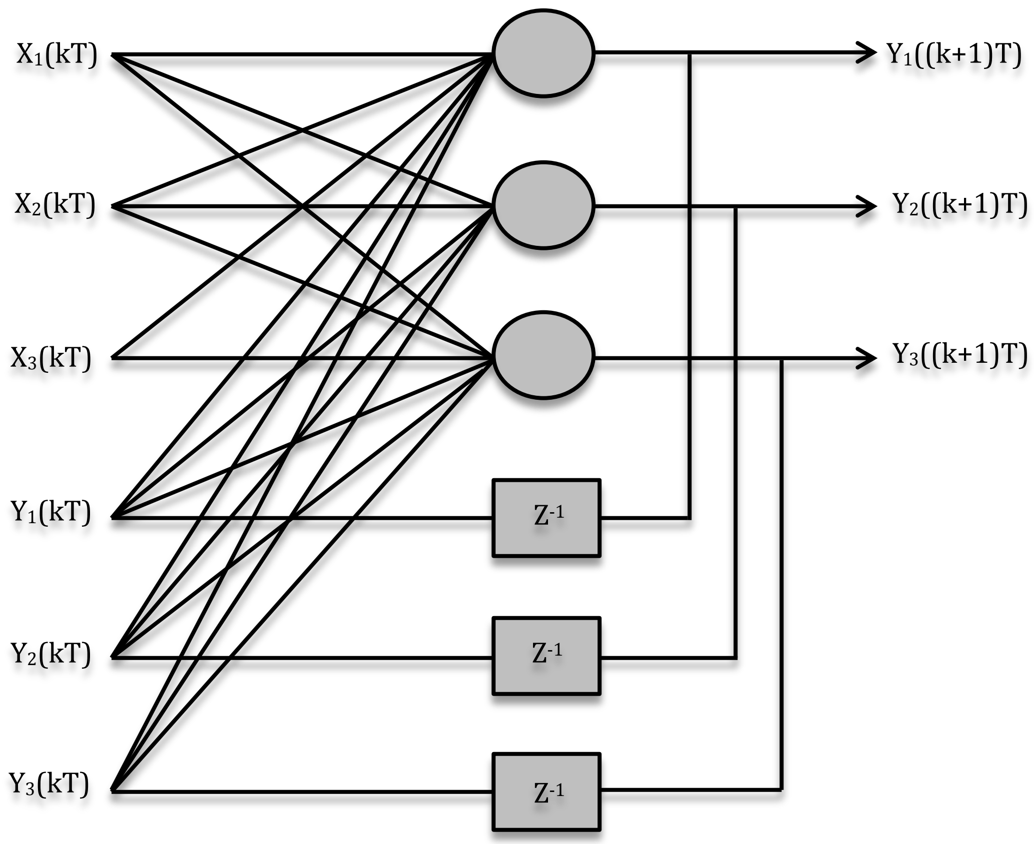


Figure 4.5 Single layer feedback network

- **Unsupervised Learning**

Unsupervised learning also known as clustering is the great promise of the future since it forms natural groups or clusters of patterns. It proves that computers could someday learn on their own in a true robotic sense. Presently, this learning technique is limited to networks known as self-organizing maps. Though these networks are not commonly applied, they have proved that they can provide a solution in a few cases, proving that their promise is not groundless. They have been proven to be more effective than many algorithmic techniques for numerical calculations. They are also being used in the lab where they are split into a front-end network that recognizes short, phoneme-like fragments of speech that are then passed on to a back-end network. The second artificial network recognizes these strings of fragments as words. This promising field of unsupervised learning is sometimes called self-supervised learning. These networks do not use any external tempts to adjust their weights, instead, they internally monitor their performance. These networks look for regular patterns in the input signals, and makes adaptations according to the function of the network. Even without being told whether it's right or wrong, the network still must have some information about how to organize itself. This information is built into the network topology and learning rules. At the present state of the art, unsupervised learning is not well understood and is still the subject of research. This research is currently of interest to the government because military situations often do not have a data set available to train a network until a conflict arises.

Table 4.1 has described the number of networks that follows the supervised and unsupervised learning.

Table 4.1 Networks following supervised and unsupervised learning

Supervised Learning	Unsupervised Learning
ADLINE	Hamming networks
MADALINE	Kohonen's self –organizing maps
Perceptron	Adaptive Resonance Theory(ART)
Multi-Layer Perceptron (MLP)	Counter Propagations Networks(CPN)
Radial Basic Neural Network(RBFN)	Neo-cognitions
Probabilistic Neural Network(PNN)	Adaptive Bidirectional Associative Memory
General Regression Neural Network (GRNN)	

4.3.4 Back propagation

Back propagation is a systematic method for training multi–layer artificial networks [9]. It has a mathematical foundation that is strong if not highly practical. It is a multi-layer forward network using extend gradient descent based delta learning rule, commonly known as back propagation (of errors) rule. Back propagation provides a computationally efficient method for changing the weights in a feed forward network, with differentiable activation function units, to learn training a set of input-output examples. Being a gradient descent method it minimizes the total error of the output computed by the net. The network is trained by supervised learning method. The aim of this network is to be train the net to achieve a balance between the ability to respond correctly to the input patterns that are used for training and the ability to provide good response to the input that are similar.

The various parameters used in back propagation are as follows.

x = Input training vector x = (x_1....x_i.....x_n)

t = Output target vector t = (t_1....t_i.....t_n)

δ_k = Error at output unit y_k δ_j = Error at hidden unit z_j

α = Learning rate V_{oj} = Bias on hidden unit j

z_j = Hidden unit j y_k = Output unit k

w_{ok} =Bias on output unit k

The training algorithm of back propagation involves four stages as below:

- ***Initialization of weights***

Step: 1 Initialize weight to small random values

Step: 2 while stopping condition is false, do steps 3-10

Step: 3 for each training pair do steps 4-9

- ***Feed forward***

Step: 4 Each input unit receives the input signal x_i and transmits this signals to all units in the layer above i.e. hidden units

Step: 5 Each hidden unit (z_j, j=1,...,p) sums its weights input signals

$$z_{-inj} = v_{oj} + \sum_{i=1}^{n} x_i v_{ij} \qquad 4.5$$

applying activation function

$$Z_j = f(z_{-inj})$$

and sends this signal to all units in the layer above i.e. output units.

Step: 6 Each output unit (y_k, k=1,...,m) sums its weighted input signals

$$y_{-inj} = w_{ok} + \sum_{i=1}^{p} x_j v_{jk} \qquad 4.6$$

and applied its activation function to calculate the output signals.

$$Y_j = f(y_{-inj})$$

- ***Back propagation of errors***

Step: 7 Each output unit (y_k, k=1,...,m) receives a target pattern corresponding to an input pattern error term is calculated as

$$\delta_k = (t_k - y_k) f(y_{-ink}) \qquad 4.7$$

Step: 8 Each hidden unit (z_j, j=1,...,p) sums its delta inputs from units in the layer above

$$\delta_{-inj} = \sum_{k=1}^{m} \delta_j w_{ik} \tag{4.8}$$

The error information term is calculated as

$$\delta_j = \delta_{-inj} f(z_{-inj}) \tag{4.9}$$

⌗ ***Updation of weight and biases***

Step: 9 Each output unit (y_k, k=1, . . . , m) updates its bias and weights (j=0, . . . ,p). The weight correction term is given by

$$\Delta W_{jk} = \alpha \delta_k z_j \tag{4.10}$$

and the bias correction term is given by

$$\Delta W_{ok} = \alpha \delta_k \tag{4.11}$$

Therefore,

$$W_{jk}(new) = W_{jk}(old) + \Delta W_{jk},\ W_{ok}(new) = W_{ok}(old) + \Delta W_{ok} \tag{4.12}$$

Each hidden unit (z_j, j= 1, . . . , p) updates its bias and weights (i= 0, . . . , n). The weight correction term is given by

$$\Delta V_{ij} = \alpha \delta_j x_i \tag{4.13}$$

and the bias correction term is given by

$$\Delta V_{oj} = \alpha \delta_j \tag{4.14}$$

Therefore,

$$V_{ij}(new) = V_{ij}(old) + \Delta V_{ij},\ V_{oj}(new) = V_{oj}(old) + \Delta V_{oj} \tag{4.15}$$

Step: 10 Test the stopping condition.

The stopping condition may be the minimization of the errors, number of epochs etc…

4.4 Genetic algorithm

In the 1950s and 1960s, several computer scientists independently studied evolutionary systems with the idea that evolution could be used as an optimization tool for engineering problems. A genetic algorithm (GA) uses the principles of evolution, natural selection and genetics from natural biological systems in a computer algorithm to simulate evolution. Essentially, the genetic algorithm is an optimization technique that performs a parallel, stochastic, but directed search to evolve the fitness population. In this section, the basic of genetic algorithm is covered which is used in various application.

The basic idea of a genetic algorithm is very simple. The term chromosomes typically refers to a candidate solution to a problem, typically stored as strings of binary digits (1s and 0s) in the computer's memory. The 'genes' are short blocks of adjacent bits that encode a particular element of the candidate solution. An 'allele' in a bit string is either 0 or 1. Crossover typically consists of exchanging genetic material between two single-chromosome haploid parents. Mutation consists of flipping the bit at a randomly chosen locus. Most applications of genetic algorithms employ haploid individuals, particularly, single-chromosome individuals. The genotype of an individual in a genetic algorithm using bit strings is simply the configuration of bits in that individual's chromosome.

4.4.1 Basic of genetic algorithm

- **Encoding**

Simple genetic algorithms require the natural parameter set of the problem to be coded as a finite-length string of binary bit 0 and 1. A bit string is a combination of 0s and 1s, which represents the value of a number in binary form. An n-bit string can accommodate all integers up to the value 2^n-1. For problem that is solved by the genetic algorithm, it is usually known that the parameters that are manipulated by the algorithms will lie in a certain fixed range say {θ_{min}, θ_{max}}. A bit string may then be mapped to the value of a parameter, say θ_i, by the mapping

$$\theta_i = \theta_{\min i} + \frac{b}{2^L - 1}(\theta_{\max i} - \theta_{\min i}) \qquad 4.16$$

Where “b” is the number in decimal form that is being represented in binary form, L is the length of the bit string.

Example: Consider a string X (a concatenation of two strings, X_1 and X_2)

| 33bit |
000001010100101001101111011111110
| 18bit | | 15bit |
X = X_1 + X_2

	Bit Length	Low Range	High Range	Binary Number	Decimal
X_1	18	-3.0	12.1	000001010100101001	5417
X_2	15	4.1	5.8	101111011111110	24318

The mapping (4.1) gives the value:

$$x_1 = -3.0 + 5417 \times \frac{12.1-(-3.0)}{2^{18}-1} = -2.687960$$

$$x_2 = 4.1 + 24318 \times \frac{5.8-4.1}{2^{15}-1} = 5.361653$$

⌗ Fitness function

A fitness function takes a chromosome (binary string) as an input and returns a number that is a measure of the chromosome's performance on the problem to be solved. Fitness function plays the same role in GAs as the environment plays in natural evolution. The interaction of an individual with its environment provides a measure of fitness to reproduction. Similarly, the interaction of a chromosome with a fitness function provides measure of fitness that the GA uses when carrying out reproduction. Genetic algorithm is a maximization routine; the fitness function must be a non-negative figure of merit. It is often necessary to map the underlying natural objective function to a fitness function form through one or more mappings. If the optimization problem is to minimize cost function $J'(\theta)$, where θ denotes the parameter set, and then the following cost-to-fitness transformation may be used:

$$J(\theta) = \frac{1}{J'(\theta)+\varepsilon} \qquad 4.17$$

Where ε is a small number, Maximization of *J* can be achieved by minimization of *J'*; so the desired effect is achieved.

⌗ Initialization of population:

The basic element processed by a GA is the string formed by concatenating substrings each of which is a binary coding of a parameter of the search space. If there are *N* decision variables in an optimization problem, and each decision variable is encoded as an n-digit binary number, then a chromosome is a string of *n* x N binary digits. Initial population is selected a randomly selected initial population of such chromosomes; each chromosome in the population represents a point in the search space, and hence a possible solution to the problem. Each string is then decoded to obtain its fitness value which determines the probability of the chromosome being acted on genetic operators. The population then evolves and a new generation is created through the application of genetic operators. The new generation is expected to perform better than the previous generation (better fitness values). The new set of strings is again decoded and evaluated, and another generation is created using the basic genetic operators. This process is continued until convergence is achieved within a population. The bit length is calculated as:

Example: suppose the two decision variable [X1, X_2],

X1= [-3, 12.1], X1= [4.1, 5.8]

The length of chromosome is calculated by

$$\left.\begin{aligned}
&(x_1^{max} - x_1^{min}) \times 10^4 = (12.1 - (-3.0)) \times 10^4 = 151000 \\
&2^{17} < 151000 \le 2^{18}, \qquad l_{x_1} = 18 \\
&(x_2^{max} - x_2^{min}) \times 10^4 = (5.8 - 4.1) \times 10^4 = 17000 \\
&2^{14} < 17000 \le 2^{15}, \qquad l_{x_2} = 15 \\
&l = \sum_{i=1}^{d} l_{x_i} = l_{x_1} + l_{x_2} = 18 + 15 = 33
\end{aligned}\right\} \quad 4.18$$

⌗ **Genetic Operators**:

Genetic operations are selection, crossover, and mutation are used to produce a new generation.

1. **Selection:**

The selection process in the genetic inheritance is the best chromosome gets more copies, the average stay even, and the worst die off. In Genetic algorithms the selection of a new population is with respect to the probability distribution based on the fitness values. In Information retrieval many researchers used the roulette wheel reproduction process.

2. **Crossover:**

The intuition of crossover in the Genetic Algorithm is to produce new solutions from the existing one. There is maybe one point crossover or multiple points' crossover. The suggested crossover is multiple point crossovers. High fitness chromosomes are more likely to be chosen in the crossover process. For example, two chromosomes are crossover between position 5 and 11.

1 0 1 1 1 1 1 1 0 0 1 1 1 0 1

1 0 0 1 1 0 0 1 1 1 1 0 0 0 0

The resulting crossover yields two new chromosomes.

1 0 1 1 1 0 0 1 1 1 1 1 1 0 1

1 0 0 1 1 1 1 1 0 0 1 0 0 0 0

3. **Mutation:**

It can help the search find solutions that crossover alone might not encounter. Chromosomes may be better or poorer than old chromosomes. If they are poorer than old chromosomes, they are eliminated in selection step. The objective of mutation is restoring lost and exploring variety of data.

Example:

1 0 1 1 1 1 1 1 0 ***0*** 1 1 1 0 1

Result 1 0 1 1 1 1 1 1 0 ***1*** 1 1 1 0 1

⌗ **Stopping criterion:**

There are many ways to terminate a genetic algorithm, many of them similar to termination conditions used for conventional optimization algorithm.

4.4.2 General produce for GA

The following steps are behind GAs.

- **[Start]** Generate random population of *n* chromosomes (suitable solutions for the problem).
- **[Fitness]** Evaluate the fitness *f(x)* of each chromosome *x* in the population.
- **[New population]** Create a new population by repeating following steps until the new population is complete.
  - **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected).
  - **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
  - **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).
- **[Accepting]** Place new offspring in a new population.
- **[Replace]** Use new generated population for a further run of algorithm.
- **[Test]** If the end condition is satisfied, stop, and return the best solution in current population.
- **[Loop]** Go to step 2.

4.5 Matlab development tools

4.5.1 Matlab

Development tools as MATLAB, Simulink, various tools are covered in this section. MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, technical computing problems can be solved faster than with traditional programming languages, such as C, C++, and Fortran. The following are main parts of MATLAB.

- **Desktop Tools and Development Environment**

This part of MATLAB is the set of tools and facilities that help to use and become more productive with MATLAB functions and files. Many of these tools are graphical user interfaces. It includes: the MATLAB desktop and Command Window, an editor and debugger, a code analyzer, and browsers for viewing help, the workspace, and folders.

⌗ Mathematical Function Library

This library is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

⌗ The Language

The MATLAB language is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick programs you do not intend to reuse. It can also do "programming in the large" to create complex application programs intended for reuse.

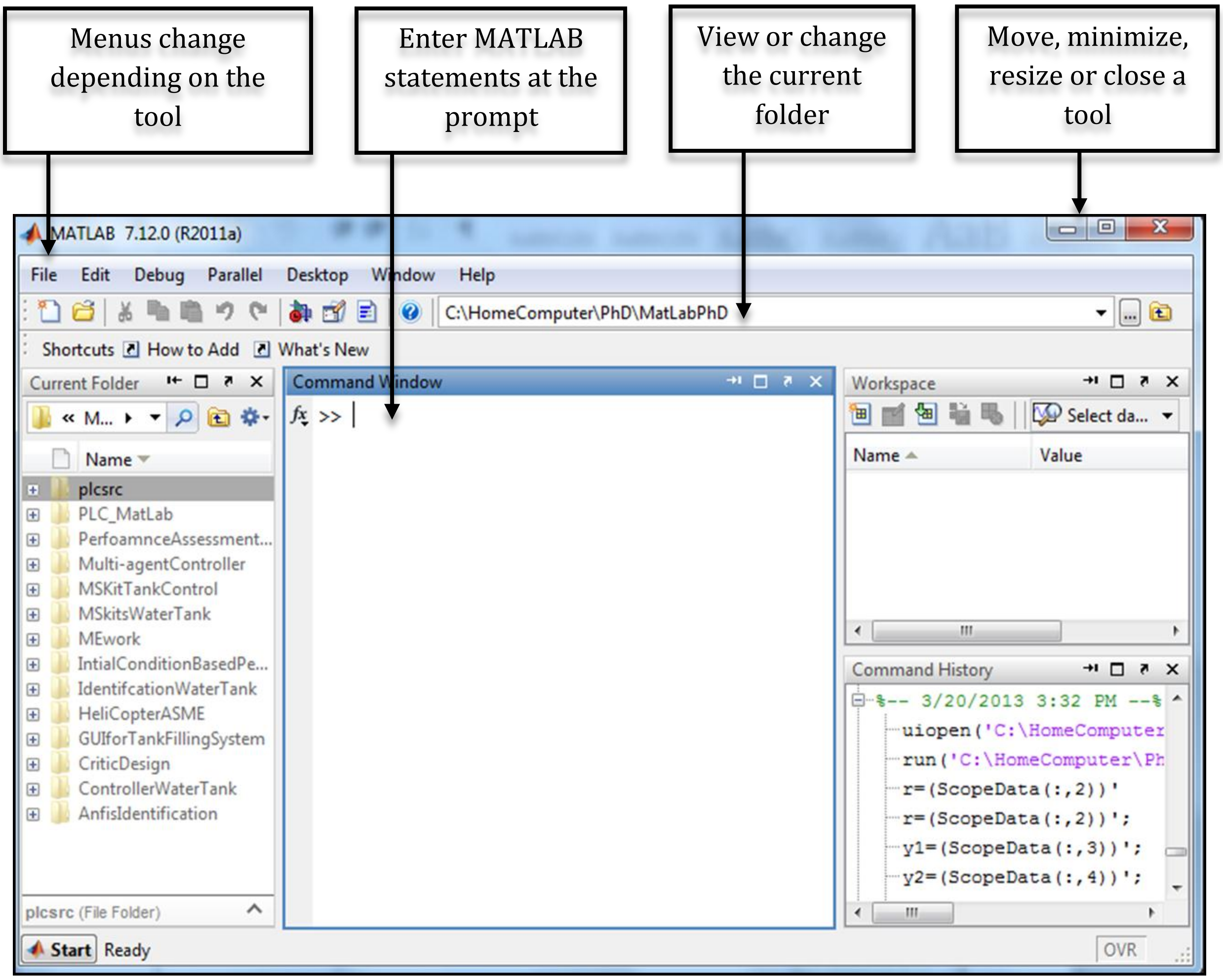


Figure: 4.6 Matlab view

⌗ Graphics

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **External Interfaces**

The external interfaces library allows to write C/C++ and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), for calling MATLAB as a computational engine, and for reading and writing MAT-files.

The MATLAB default view is given in figure 4.6.

4.5.2 Simulink

Simulink software models, simulates, and analyzes dynamic systems. It is tightly integrated with the MATLAB environment. It requires MATLAB to run, depending on it to define and evaluate model and block parameters. Simulink can also utilize many MATLAB features.

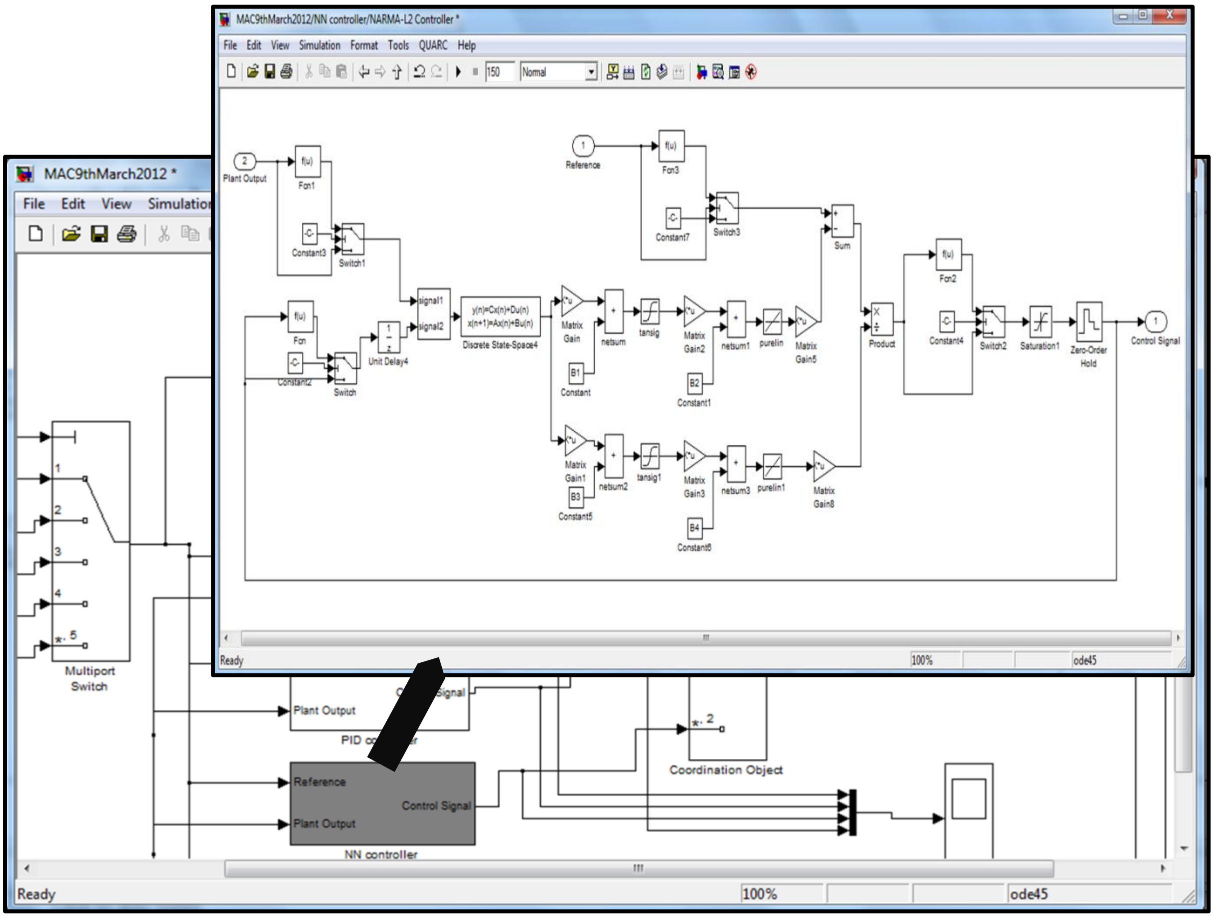


Figure: 4.7 Simulink view

Simulink can use the MATLAB environment to:

- Define model inputs.
- Store model outputs for analysis and visualization.

- Perform functions within a model, through integrated calls to MATLAB operators and functions

Simulink provides a graphical user interface (GUI) for building models as block diagrams, allowing to draw models. Simulink also includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. The interactive graphical environment simplifies the modeling process, eliminating the need to formulate differential and difference equations in a language or program. Models are hierarchical, so it can build using both top-down and bottom-up approaches. The default view is given in figure 4.7.

4.5.3. Toolboxes

- **Fuzzy logic toolbox**

Fuzzy Logic Toolbox™ provides MATLAB functions, graphical tools, and a Simulink block for analyzing, designing, and simulating systems based on fuzzy logic. The product guides through the steps of designing fuzzy inference systems. Functions are provided for many common methods, including fuzzy clustering and adaptive neuro-fuzzy learning. The stand - alone fuzzy inference system is designed for complex system.

- **Neural network toolbox**

There are four ways are way to use the Neural Network Toolbox software.

- The first way is through the four graphical user interfaces.
- The second way to use the toolbox is through basic command-line operations. The command-line operations offer more flexibility than the GUIs, but with some added complexity.
- The third way to use the toolbox is through customization. This advanced capability is allowed to create your own custom neural networks, while still having access to the full functionality of the toolbox.
- The fourth way to use the toolbox is through the ability to modify any of the functions contained in the toolbox. Every computational component is written in MATLAB code and is fully accessible.

- **Global optimization toolbox**

Global optimization toolbox provides methods that search for global solutions to problems that contain multiple maxima or minima. It includes

- Global Search
- Multistart
- Pattern search
- Simulated annealing
- Genetic algorithm

It can be used in both way command line and graphical user interface. The genetic algorithm is applied in our application.

4.6 Concluding remarks

The theoretical background of soft computing techniques such as fuzzy logic, ANN and GA is summarized and described. Toolboxes available for deploying soft computing techniques in MATLAB and used in our research work for the design and testing of proposed techniques are described in detail. Procedural steps to be followed in each trait are discussed in briefly discussed.

Chapter 5

Smart System Identification

Chapter expresses the identification of nonlinear system with different methods. The intelligent approach is used to identify the model of nonlinear system. The neural network is tuned with back propagation and genetic algorithm for comparative analysis of two approaches.

Non-linear physical models are very complex and usually required considerable development effort to arrive at an adequate form. In addition, they require intimate process knowledge and understanding. Their primary use has been in process design and simulation. It is usually the steady-state version of these models that is used to size new equipment or to simulate the process at different conditions such as feed rate, temperature and pressure, etc. These steady state models are also useful for supervisory control where they can be used in an online or offline mode to optimize the process conditions at various time intervals and readjust the set-points of the controllers in a basic control structure. Dynamic forms of these mechanistic models may be useful in simulation, prediction or real time control. They can, among others, be used to develop start-up procedures for continuous processes. System identification is crucial role for finding the model using input output data in various applications.

5.1 Introduction

System identification is the process of developing a mathematical model of a dynamic system based on the input and output data from the actual process [1]. This means it is possible to sample the input and output signals of a system and using this data generate a mathematical model. An important stage in control system design is the development of a mathematical model of the system to be controlled. In order to develop a controller, it must be possible to analyze the system to be controlled and this is done using a mathematical model. Another advantage of system identification is evident if the process is changed or modified. System identification allows the real system to be altered without having to calculate the dynamical equations and model the parameters again. System identification is concerned with developing models. The diagram of figure 5.1 shows the inputs and output of a system.

The mathematical model in this case is the black box, it describes the relationship between the input and output signals. Here, the water tank system is a nonlinear system. To adequately model it, non-linear methods using neural networks must be used. The neural networks are successfully used for modeling of nonlinear systems. Before neural networks are investigated for identification, linear techniques such as auto regressive with exogenous input (ARX) and auto regressive moving average with exogenous input (ARMAX) is discussed.

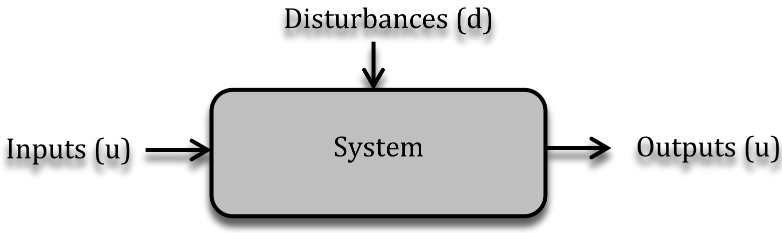


Figure.5.1 System with inputs, outputs, disturbances

Recent trends in control theory are designing of model based controllers as model predictive control, model reference control, various adaptive controllers, etc… It is not possible to derive the mathematical model of every system. Neural network being capable of describing nonlinear phenomena can be used to simulate such systems for its identification. Researchers have developed the various identification techniques using neural network for nonlinear systems. Nikoliae [2] has covered the feed forward and digital recurrent network for identification of non-linear system. Xiaoli [3] used the neural network for water tank plant identification. Bogdanov [4] was reported back propagation neural network for unstable system.

Water tank control is benchmark problem for process industry and it is nonlinear system whose time constant and gain vary considerably throughout the operating range. The present work is to model the water tank system through chosen framework with input-output data. The data measured at discrete instants of time t and collects as an array for finite duration. The feedfoward neural network (MLP) has been used for water tank system to fit the data in-place of mathematical questions. The back propagation and genetic algorithm have applied to tune the weight of neural network. The comparative studies of different algorithms have been discussed.

5.2 System identification

In identification process, the priori model structure has to be chosen and subsequently the model parameters are identified by suitable techniques. The method employed could for example be a least squares method.

Consider a process with an input *u(k)* and an output *y(k)*, where *k* is the discrete time value. Assuming that the signals can be related by a linear process,

$$\mathrm{A}(\mathrm{z}^{-1})y(k) = \frac{B(\mathrm{z}^{-1})}{F(\mathrm{z}^{-1})}u(k-p) + \frac{\mathrm{C}(\mathrm{z}^{-1})}{\mathrm{D}(\mathrm{z}^{-1})}e(k) \qquad 5.1$$

Where z^{-1} is the shift operator, defined by $y(k-1) = z^{-1}y(k)$ and the polynomials *A, B, C, D* and *F* are given by:

$$\begin{aligned}
A(z^{-1}) &= 1 + a_1 z^{-1} + \ldots + a_{na} z^{-na} \\
B(z^{-1}) &= b_0 + b_1 z^{-1} + \ldots + b_{nb} z^{-nb} \\
C(z^{-1}) &= c_0 + c_1 z^{-1} + \ldots + c_{nc} z^{-nc} \\
D(z^{-1}) &= d_0 + d_1 z^{-1} + \ldots + d_{nd} z^{-nd} \\
F(z^{-1}) &= f_0 + f_1 z^{-1} + \ldots + f_{nf} z^{-nf}
\end{aligned}$$

Where p is the time delay (sampling intervals) or dead time between the process input and output and e the modeling error. There are two ways of looking at this error, when the polynomials $C/D = 1$, e can be considered as the modeling error. However, in all other cases when $C/D \neq 1$, e is usually assumed to be white noise and the polynomials C and D are fit such that the remaining modeling error is as small as possible. A white noise signal is an uncorrelated signal with average zero and variance equal to one. In the literature [5], several special cases of model types are presented:

- ARX model structure:

$$A(z^{-1})y(k) = B(z^{-1})u(k-p) + e(k) \tag{5.2}$$

Which implies that $nc = nd = nf = 0$.

- ARMAX model structure:

$$A(z^{-1})y(k) = B(z^{-1})u(k-p) + C(z^{-1})e(k) \tag{5.3}$$

Which implies that $nf = nd = 0$.

- Output error model structure:

$$y(k) = \frac{B(z^{-1})}{F(z^{-1})}u(k-p) + e(k) \tag{5.4}$$

Thus $na = nc = nd = 0$.

One of the major drawbacks of above linear models are the limited range of applicability. Especially extrapolation capabilities of the models beyond the region for which they were developed can be poor owing to process non-linearities. One way to avoid this is to adapt the neural network based identification [6].

5.3 Neural network based identification

The nonlinear autoregressive network with exogenous inputs (NARX) is dynamic network which has several layers including feedback connections. The NARX model is very popular for time series applications and it is derived from linear ARX model. The mathematical representation for the NARX model is

$$\hat{y} = F(y(k),\ldots, y(k-n_s-1), u(k),\ldots, u(k-n_u-1)) \quad 5.5$$

Where *k* is the discrete time, ny and nu are integers related to the order of the system. The output signal of the model is regressed from past output signal and past values of exogenous input signal. The output function is approximated for NARX model using a feed forward neural network MLP as per figure 5.2 [7]. This implementation is also useful for vector ARX where the input and output can be multiple dimensional. During training both the process and ANN receive the same input, the outputs from the ANN and process are compared, this error signal is used to update the weights in the ANN. This is an example of supervised learning the teacher provides target values for the learner.

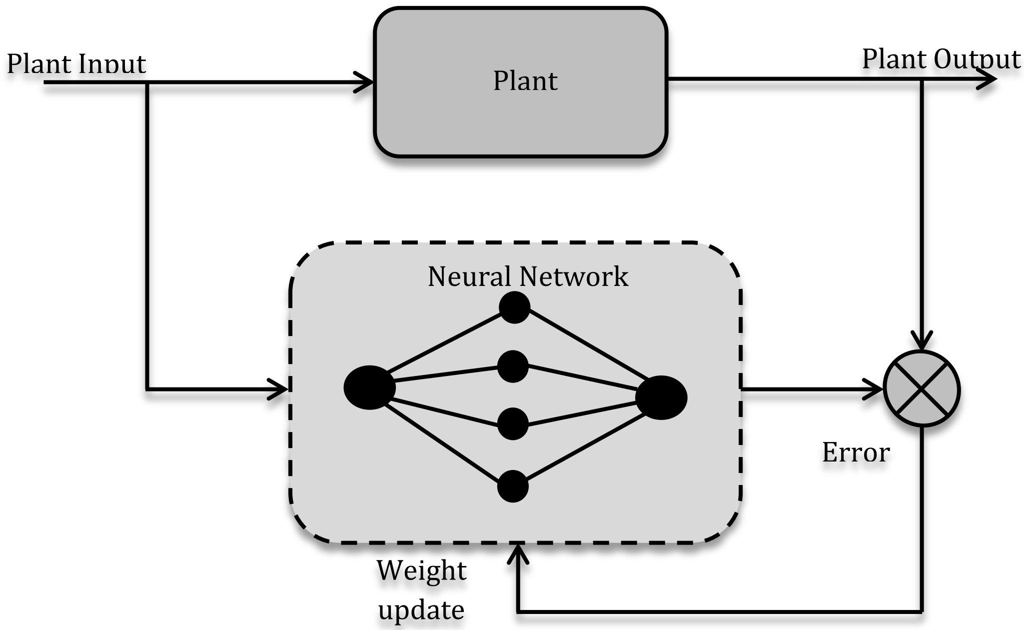


Figure 5.2 Neural network forward modeling

The function f is approximated for NARX model using a feed forward neural network MLP. A two layer MLP is used to approximation and its resulting diagram is shown in figure 5.3. This implementation is also useful for vector ARX where the input and output can be multiple dimensional. NARX network using various tuning algorithms is demonstrated for modeling of nonlinear dynamics system applications [8].

5.4 Serial-parallel mode of networks

If strong observability conditions are known (or assumed) to be satisfied in the system's region of operation, then the identification procedure using a feed forward neural network is quite straightforward. At each instant of time, the inputs to the network (not to be confused with the inputs to the system) consisting of the system's past *n* input values and past *n* output values (altogether 2*n*), are fed into the neural network [9].

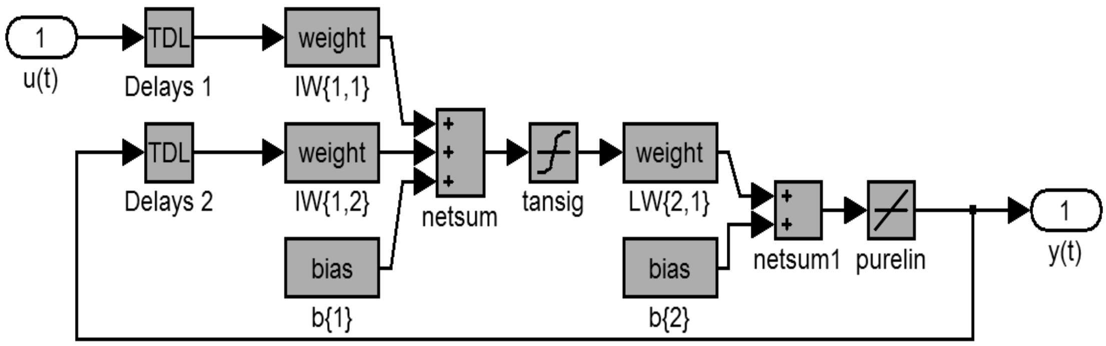


Figure 5.3 NARX Neural network structure modeling method

The network's output is compared with the next observation of the system's output, to yield the error

$$e(k+1) = y(k+1) - NN[Y_n(k-n+1), U_n(k-n+1) \qquad 5.5$$

The weights of the network are then adjusted using static back propagation to minimize the sum of the squared error. Once identification is achieved, two modes of operation are possible:

- **Series Parallel mode:** In this mode, the outputs of the actual system are used as inputs to the model. This scheme can be used only in conjunction with the system and it can generate only one step ahead prediction. The architecture is identical to the one used for identification as per figure 5.4.

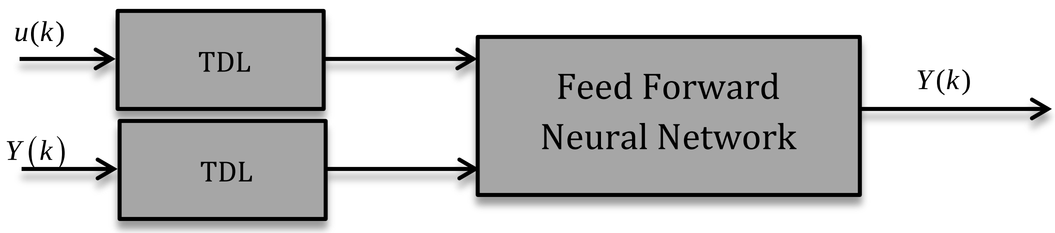


Figure 5.4 Series parallel mode neural network

- **Parallel Mode:** If more than one-step-ahead prediction is required, the independent mode must be used. In this scheme, the output of the network is fed back into the network as per shown in figure 5.5, the outputs of the network itself are used to generate future predictions. While one cannot expect the identification model to be perfect, this mode of operation provides a viable way to make short term prediction (> 1). Further, in many cases the objective is not to make specific predictions concerning a system but rather to train the network to generate complex temporal trajectories. In this case, if identification is accurate, the model will exhibit the same type of behavior as the original system.

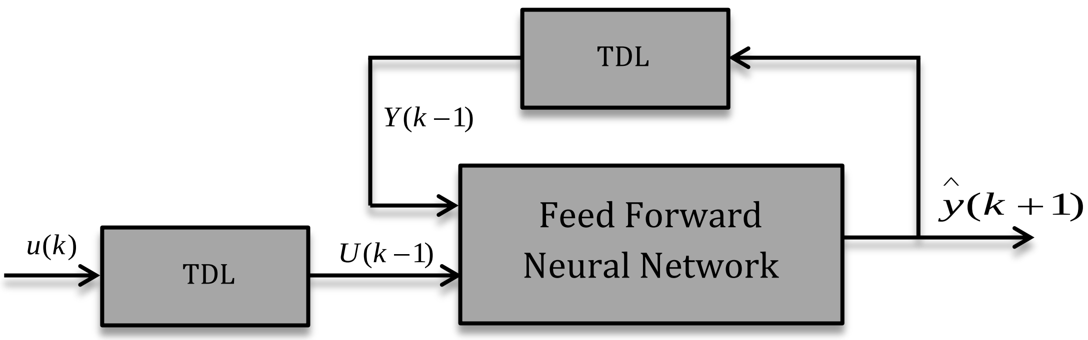


Figure 5.5 Parallel mode neural network

For standard NARX architecture, the feed forward neural network is connected with feedback, as figure 5.4. The true output is not available so, the architecture figure 5.5 is used for training. There are some merits that the network is very accurate and the architecture is purely feedfoward MLP.

5.4 Training algorithm: back propagation

The standard back-propagation learning rule is used to update the weights. The error function is:

$$J(W_I,t) = \frac{1}{2}(y(t)-\hat{y}(t))^2 = \frac{1}{2}(\hat{e}(t))^2 \tag{5.6}$$

Where $\hat{e}(t) = y(t) - \hat{y}(t)$; y(k) and y (k) are the output of the system and NN output, respectively. The parameters W_I represent the weights of neural network. Using algorithm, the error function of equation 5.6 is used to optimize the weight parameter W_I of the network and the general update function is given as

$$W_I(t+1) = W_I(t) - \eta_I \hat{e}(t) \frac{\partial \hat{y}(t)}{\partial W_I} \tag{5.7}$$

Where η_I is learning rate. The 'train' function adjusts the weights of the network so the output of the network is similar to the non-linear system. The back propagation training with 600 epochs is shown in figure 5.6. At the cut-off of 600 epochs, the algorithm was still converging on an error, but extremely slowly.

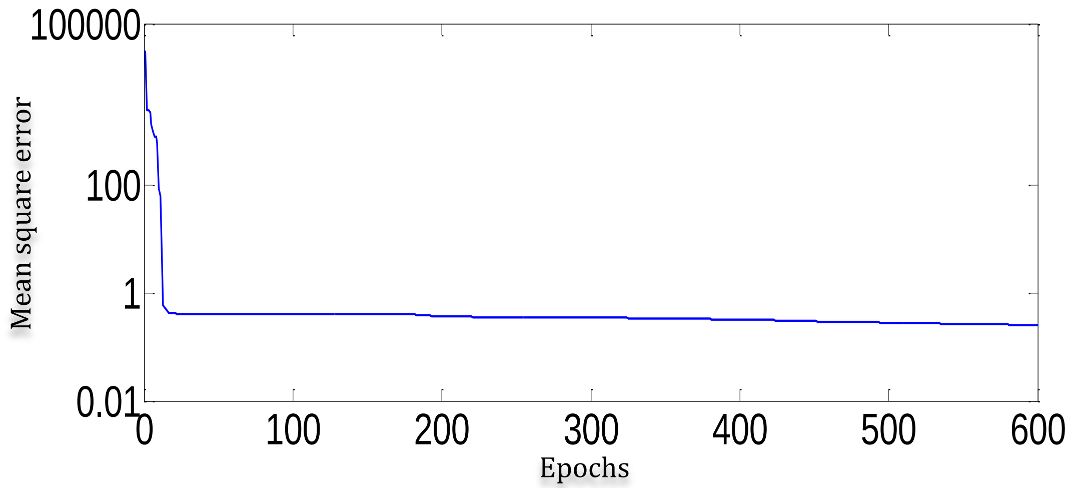


Figure 5.6 Performance of backpropagation training

This is not suitable for online training of neural network in controller design. So, the genetic algorithm is applied to train the neural network which is discussed in next section.

5.5 Training algorithm: Genetic Algorithm

The genetic algorithm is used to tune the weights of neural network which is discussed in section 4.4. The error function equation 5.6 is selected as fitness function where the weights are used to create the chromosome [10], [11] as per below.

Bits of chromosome	$B_1, B_2, \ldots, B_n$	$B_1, B_2, \ldots, B_n$	....	$B_1, B_2, \ldots, B_n$
Weights of neural network	W_1	W_2	...	W_n

The algorithm is driven towards maximizing this fitness measure. The following GA parameters are selected for the training cycle:

Table 5.1 Tuning parameters of genetic algorithm

Population size	100
Crossover rate	0.9
Mutation	0.01
Terminating generation	150
Stall time limit	50
Selection	Roulette Wheel
Fitness function	$10^7/J$

The performance of this algorithm is given in figure 5.7.

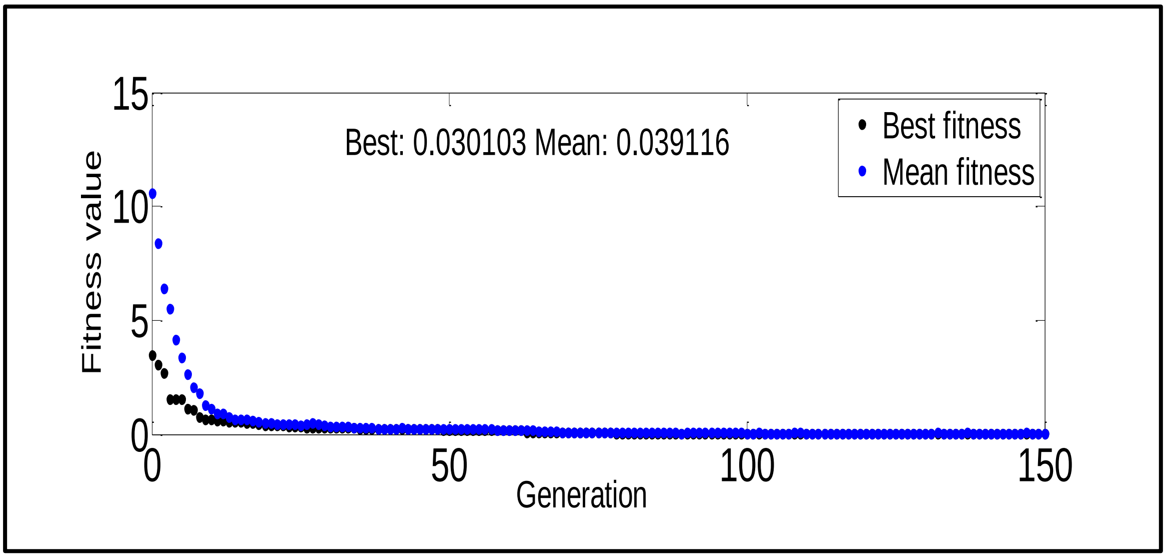


Figure 5.7 Performance of genetic algorithm training

5.7 Comparative analysis

First, the two layers MLP are used to identify the system. The weights of neural network are tuned with two algorithms: Back propagation and Genetic algorithm. The simulation is developed for both techniques with nonlinear water tank control plant. The plant applied input data is given in figure: 5.8.

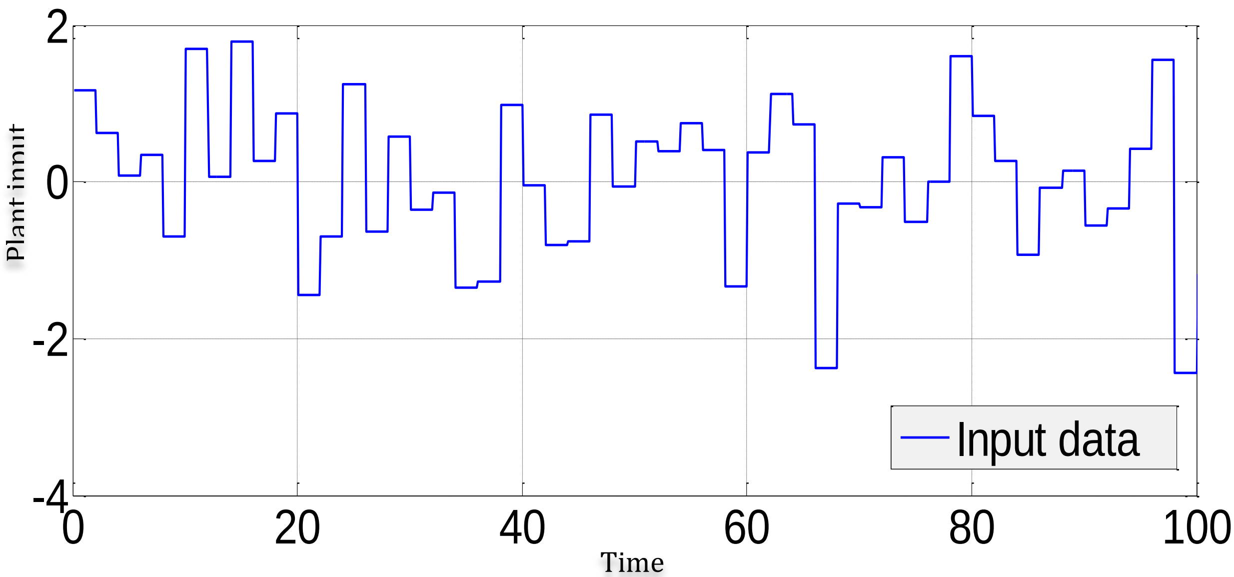


Figure 5.8 Training data of water tank system

The outputs of neural network tuned using back propagation and genetic algorithm are displayed in figure: 5.9. The errors for both techniques are shown in figure 5.10.

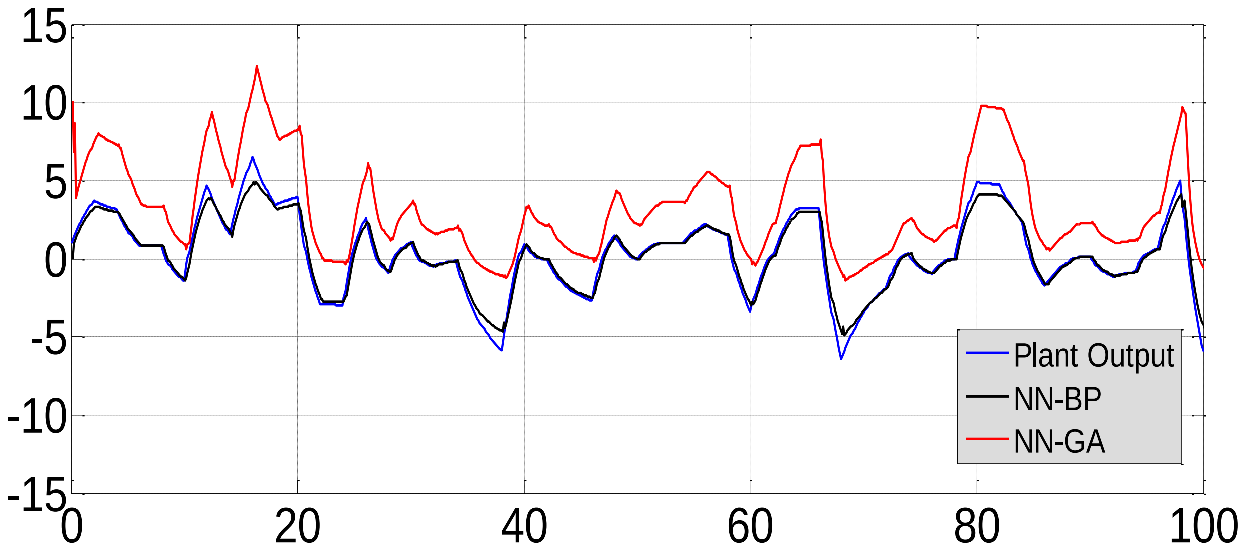


Figure 5.9 Plant model output, NN-BP output, NN-GA Output

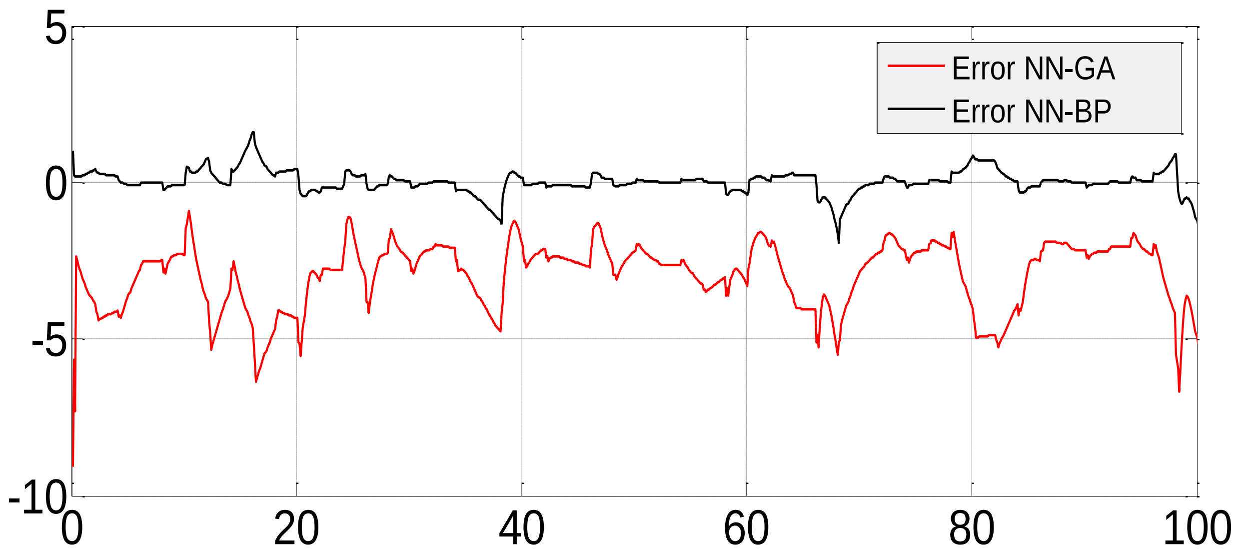


Figure 5.10 Error NN-GA and NN-BP

The mean square errors for given input data are 0.135 and 9.948 for NN-BP and NN-GA, respectively. So, it is clear from graphical anlysis, the back propagation weight tuning is better compared then genetic algorithm based tuning.

5.8 Concluding remarks

Two layers MLP are used to develop an identification of nonlinear system. The neural network is used in form of nonlinear auto regressive external inputs (NARX). The back propagation and genetic algorithm are used to tune the weights of neural network. The performance of back propagation is provided effective solution compared the genetic algorithm.

So, the back propagation is used to tune the neural network. The identification agent is design on this theory and used in multi-agent architecture in chapter 8.

Chapter 6

Controller Performance Assessment

The performance of control is continuously assessed online with different parameters as steady state error, overshoot, settling time, rise time, variance and reference change. The fuzzy logic is used to identify the better control strategy for control problem with help of performance assessment. The detail of membership functions design, rule base and its implementation are discussed in chapter.

A Control Performance Assessment (CPA) as a software agent is designed for assessment the SISO control system performance. A dedicated system for data pre-processing is applied in order to guaranty robust properties of the extracted process features. The CPA provides early warning information when an inadmissible performance is detected. Some results of experimental verification of the CPA are reported in the chapter.

6.1 Introduction

Number of methods have been reports in literatures for control performance both for SISO [1,2,3,4,5,6] and MIMO [7,8,9] systems in the last decade. The control performance assessment is key operation for process monitoring aiming to meet the technological challenges to cope with the changing requirements and process specifications, varying raw materials properties, fluctuation of operating conditions due to the equipment degradation and/or nonlinearity. The approaches are designed in three directions:

- Data centered approach using transient response, random output value deviations or oscillations [3,5,6,10], applying a variety of techniques to detect the main classical indicators as overshoot, raising time, settling time, decay, steady state error oscillation, Variance in output, etc…
- By comparing the output from the loop to the output of an ideal minimum variance controller (MVC) [1,3] with a number of modifications [4,2,9].
- Benchmarking [7, 8].

Various performance parameters will be covered in detail. Performance evaluation is designed using fuzzy logic to generate activation signal for various controgents.

6.2 Control performance assessment

The survey papers by Jelali [11] and Qin and Yu [12] provide a very good data centered approach for performance monitoring area for SISO and MIMO control problems. The performance assessment criterion has been defined in order to find the best control algorithm among multiple algorithms. 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, measure about which control algorithm will give the best performance as compared with the others [11, 13, 14].

The variance of the system output is also another important criterion considering for controller performance assessment because of its direct relationship to process performance, product quality, and profit [11, 15, 16]. The signal variance is directly affected to control actions. 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 [11]. Figure 6.1 depicts the performance criteria which are specified in terms of control action, output variance and meeting user specifications [17].

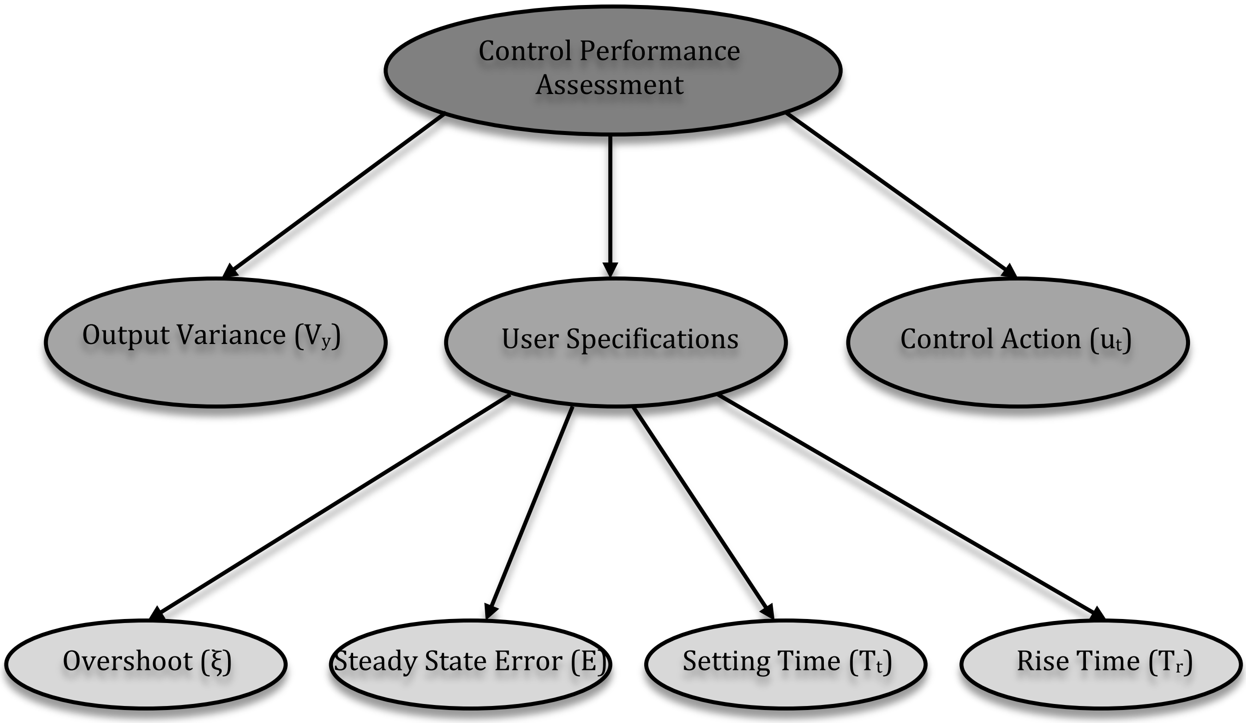


Figure 6.1 Control performance assessment criteria

The behavior of the system is characterized through the online measurements of parameters listed below. The average of the current and last two values of the measurements is used as output parameters to reduce the shattering problem (Zeno behavior) [18].

[1] Overshoot (ξ_y)

$$\zeta_y(t) = \frac{y_{max} - y_\infty}{y_\infty} \times 100 \tag{6.1}$$

Where y_{max} is amplitude maximum value at the output and y_∞ is the steady state value of the output.

[2] Steady State Error (e_∞)

$$e_\infty(t) = \lim_{t \to \infty}(w(t) - y(t)) \cong 0 \tag{6.2}$$

Where w (t) is desired output and y (t) is actual output.

[3] Settling time (T_s)

This is the required for the measured process variable y(t) to first enter and then remain within a band Δy whose width is computed as ±5% of the total change in y(t) [19].

[4] Rise time (T_r)

The output signal rise time represents the time required to response from 10% to 90 % of the output signal.

[5] Variance of output signal (V_y)

$$V_y(t) \text{ or } V_u(t) = \frac{\sum_{i=1}^{N}(y_i - y_\infty)^2}{N-1} \tag{6.3}$$

Where the variance of sampled population of the output signal y (t) is mean squared deviation of the individual values y_i of y (t) from population mean [20] and N denotes the size of the sampled population of the output signal.

[6] Changes in reference signal state (Π_w)

$$\prod_w = w(t) - w(t-1) \tag{6.4}$$

Where w (t) is present sate and w (t-1) is last state. By comparing the current set-point with the previous one, it will check the state of the reference signal whether it is increasing, decreasing or remaining as it was in the last state.

The control performance index is calculated using various control performance parameters. It is accepted in two forms. The first form is calculated as follows:

$$J = \frac{1}{\sum_{i=1}^{n} \beta_i q_i}, \sum_{i=1}^{n} \beta_i = 1 \quad 6.5$$

Where $q_1, q_2 \ldots q_n$, are the various control performance parameters as overshoot, steady state error, reference change etc. The weights β_i are determined in expert way. Weights could be corrected by iterative process of data analysis and discussions in order to use the experience gathered in years by operators of running the particularly process [21]. The second form of the Performance Index calculation is fuzzy logic based which is discussed in next section.

6.3 Fuzzy logic based performance evaluation

Fuzzy logic is used to identify the best control algorithm performance for current operating regime with help of control performance assessment parameters. In other word, it is used to generate a switch signal which determines at instant the controlgent that is to be activated [22, 23]. Different six variables are discussed in pervious section, will represent the input parameters of the fuzzy logic which can be expressed as:

$$\Lambda(t) = [\xi(t), e(t), \tau_r(t), \tau_s(t), V_y(t), \prod_w(t)] \quad 6.6$$

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

$$C_\eta = [C_1, C_2, C_3] \\ where\ C_1 = [0,10],\ C_2 = [11,20],\ C_3 = [21,30] \quad 6.7$$

The switching parameter C_η can be set between 0 to 30, for controgent-1 is between 0 to 10 as same controgent-2 and controgent-3 are 11 to 20 and 21 to 30 respectively. The controlgents are various agents with different control algorithms #.

6.3.1 Fuzzy sets

To build the fuzzy system, first each variable must be decomposed into a set of regions and output or solution variable then redefined into a set of fuzzy regions. There are six inputs and one output for the fuzzy performance evolution system. The inputs and the output are defined as fuzzy regions (sets) in a fuzzy logic system as shown in the table 6.1.

The membership functions (MFs) used in the fuzzy logic play a crucial role in the final performance evaluation.

 # A paper entitled " Multi-agent controller suing soft computational paradigms" is published in Indian Journal of Applied Research , ISSN 2249-555X, Vol.3, Issue 3, Pages 138-141, February 2013.

Therefore, selection of the appropriate functions is an important design problem. So, in order to design an optimal fuzzy system the proper membership functions are searched by using several simulation experiments. The fuzzy input variables and output variable are characterized by three fuzzy membership functions. From the point of view of simplicity and computational complexity [22], 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 [23].

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

$$f(x,a,b,c) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & \mathrm{b} \le x \le c \\ 0, & \mathrm{c} \le x \end{cases} \qquad 6.8$$

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

$$f(x,a,b,c,d) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \dfrac{d-x}{d-c}, & \mathrm{c} \le x \le d \\ 0, & \mathrm{d} \le x \end{cases} \qquad 6.9$$

6.3.2 Fuzzy rule

The fuzzy logic in the proposed section is used as performance evaluation algorithm for multi-agent control system. The rule base is determined by the experimental consideration of the influence of each of the switching logic input parameters and output parameters as below:

Table 6.1 Fuzzy sets

Parameter	Actual Range	Range (%)	Name of Mf	Type of membership function	Mf range
Overshoot	0 to 500	0 to 100	Low	Triangle	[-5 0 100]
			High	Trapezoidal	[0 20 20 100]
Steady state error	-5 to 5	-100 to 100	N-error	Trapezoidal	[-100 -100 25 16]
			M-error	Triangle	[-21 0 21]
			P-error	Trapezoidal	[16 25 100 100]
Variance in output	0 to 40	0 to 100	Low	Trapezoidal	[0 0 5 100]
			High	Trapezoidal	[0 10 100 100]
Rise time	0.1 to 500	0 to 100	Fast	Triangle	[0 0 10]
			Average	Trapezoidal	[5 10 20 25]
			Slow	Trapezoidal	[20 40 100 100]
Settling time	20 to 500	0 to 100	Fast	Trapezoidal	[0 0 20 30]
			Optimal	Triangle	[20 40 60]
			Slow	Trapezoidal	[50 60 100 100]
Reference change	-15 to 15	-100 to 100	Dec	Trapezoidal	[-100 -1000 -5 0]
			Norm	Triangle	[-5 0 5]
			Inc	Trapezoidal	[0 5 100 100]

Table 6.2 Rule base

Rule-1	Overshoot is High and steady state error is N-error	NN
Rule-2	Overshoot is High and steady state error is M-error	PID
Rule-3	Overshoot is High and steady state error is P-error	Fuzzy
Rule-4	Rise time is fast and settling time is fast	NN
Rule-5	Rise time is fast and settling time is slow	PID
Rule-6	Rise time is slow and settling time is fast	NN
Rule-7	Rise time is slow and settling time is slow	PID
Rule-8	Variance is low	PID
Rule-9	Variance is high	NN
Rule-10	Reference change is norm	PID
Rule-11	Reference change is Inc or Dec	NN

The above fuzzy logic based switching logic is applied to SISO water tank control system in chapter 8.

6.3.3 Fuzzy inference procedure

At each sampling time the switching logic input parameters are compared to their desired values using their fuzzy sets. The graphical user interface (GUI) is developed for experimental study of whole design as per shown in figure 6.2. So assume that the output signal overshoot ξ_y is 37%, the variance of the output V_y is 30, the reference signal (Π_w=0) is not changing, settling time T_s is 45 sec, Rise time T_r is 44 sec and the steady state error e_∞ is 5 as shown in figure 6.2. The combination cause rules 2 and 10 to fire. The three rules have somehow to be combined to form a single switching output ($C_\eta(t)$). Therefore, from the combined region, one of the several techniques of defuzzification can be applied to produce final evaluation C_η. In the case, Middle-of-the-Max defizzification approach is used as shown in figure 6.3 where the controlgent-1(PID) is selected. The middle-of-the-Max defuzzification approach grantee the selection of one controgent each time. Figure 6.4 next illustrates the case when the controlgent-2 (Fuzzy) is selected. Same, figure 6.5 is the case for selection of controlgent-3(NN).

6.4 Structure and function of CPA

The scheme of CPA is presented in figure 6.7. It consists of three section as preprocessing of data, situation classification and preformation evaluation.

- **Preprocessing:** In the preprocessing activity, it scans the buffer of recent real-time signals for recognizable events. It is preprocessed for other performance measure.
- **Performance measure:** It checks if the process is in a steady-state; if so, it terminates processing. If an event that may be evaluated is detected and the conditions for feature estimation are fulfilled so the performance measure is excited. It may extract the following features of detected events: overshoot, steady state error, variance in output, rise time, settling time and reference change as per discussed in section 6.2. The figure: 6.7 is given flow chart of working.
    - Oscillation check: calculate the overshoot.
    - Variation check: calculate the variance.
    - Settlement check: calculate the rise and settling time.
    - Initial steady state check: calculate the steady state error.
    - Reference check: calculate the change in reference.

  The consulate array of performance measure is passed to performance evaluation using fuzzy logic for further analysis
- **Performance evaluation using fuzzy logic:** The parameters of performance are copied from buffer of performance measure. It is designed using fuzzy logic to generate the activation signal as per discussed in section 6.3.

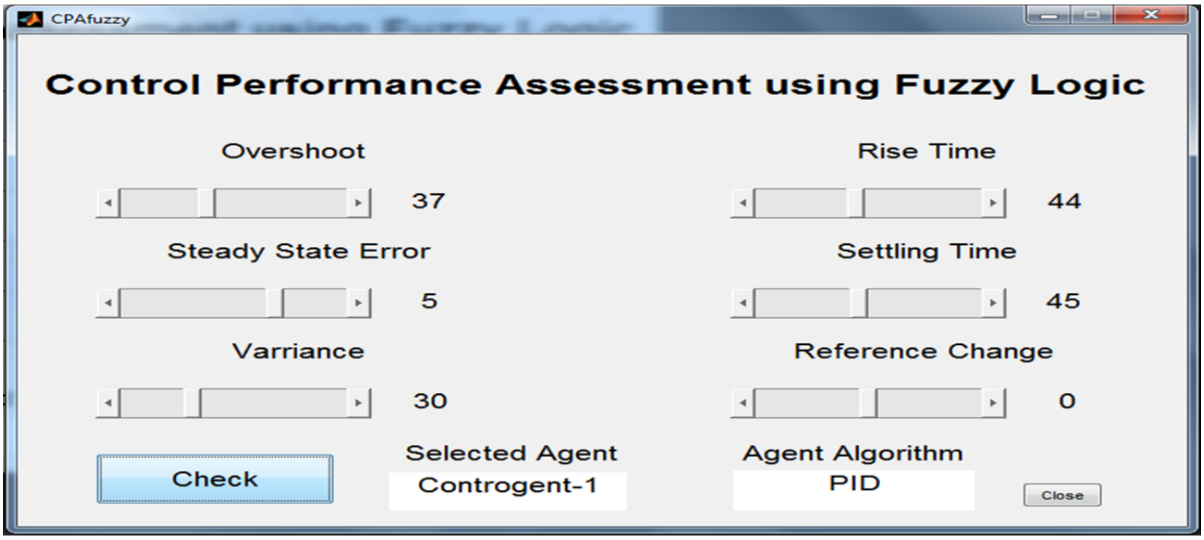


Figure 6.2(a) Graphical user interface: PID controgent activation

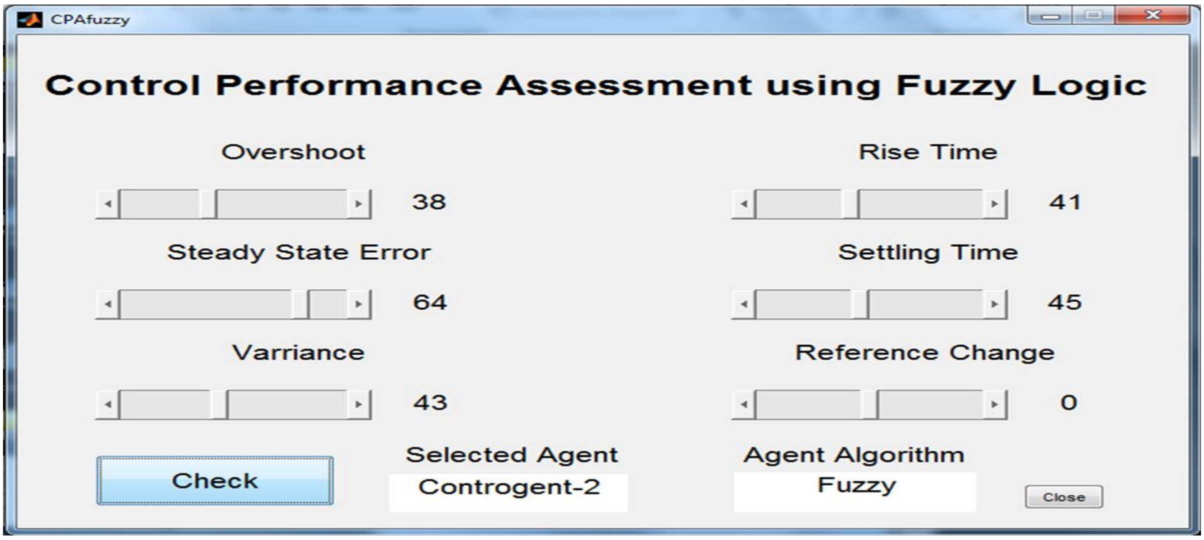


Figure 6.2(b) Graphical user interface: Fuzzy controgent activation

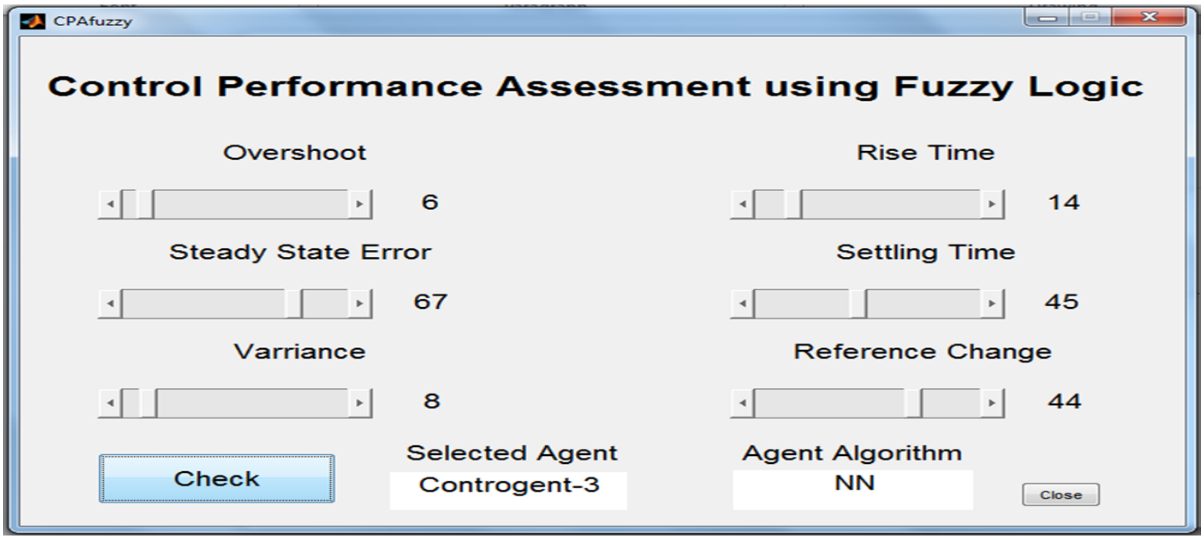


Figure 6.2(C) Graphical user interface: NN controgent activation

Figure 6.2 GUIs for CPA response using Fuzzy Logic

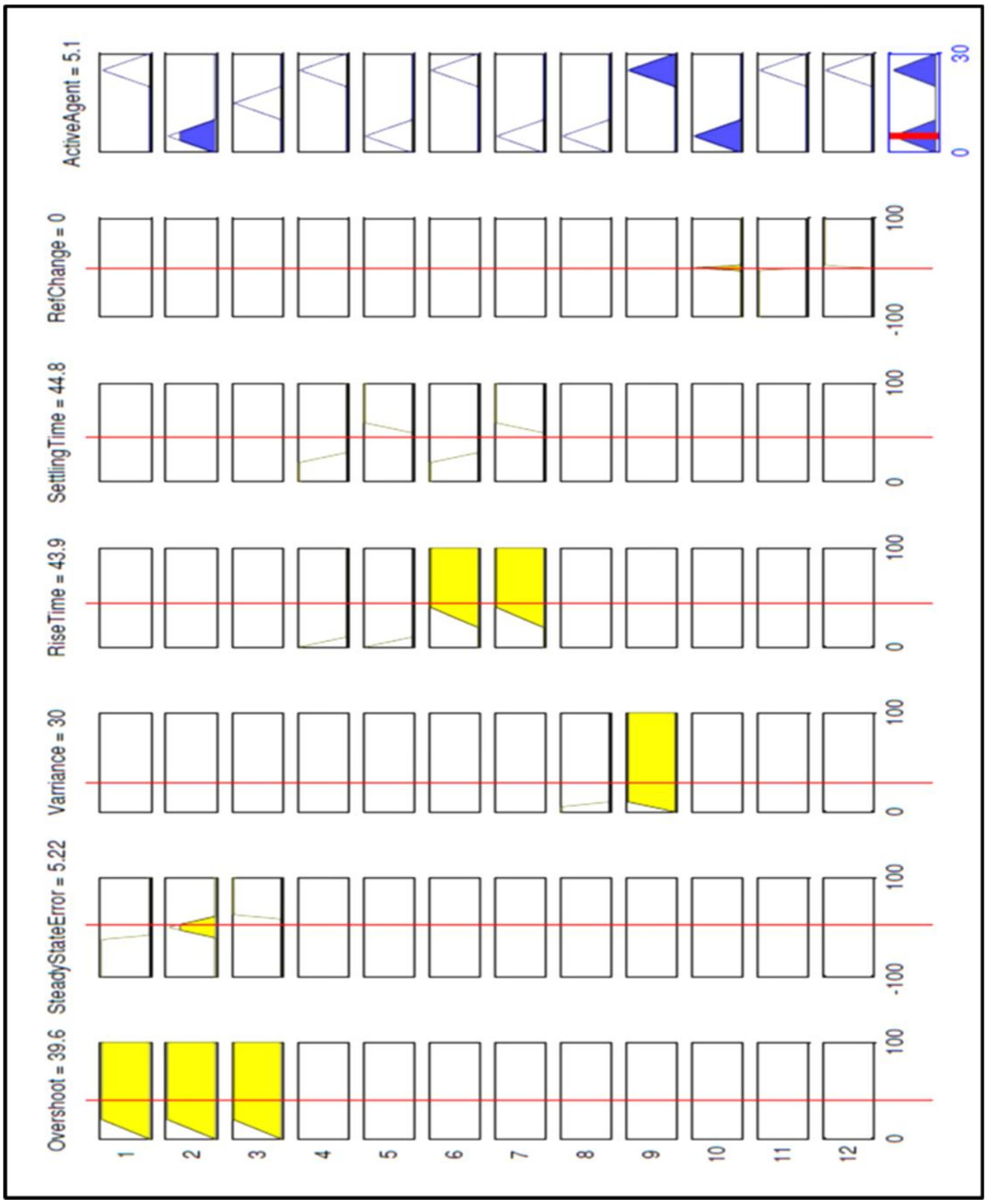


Figure 6.3 Controgent-1 (PID) is selected as fuzzy logic output based in various input parameters

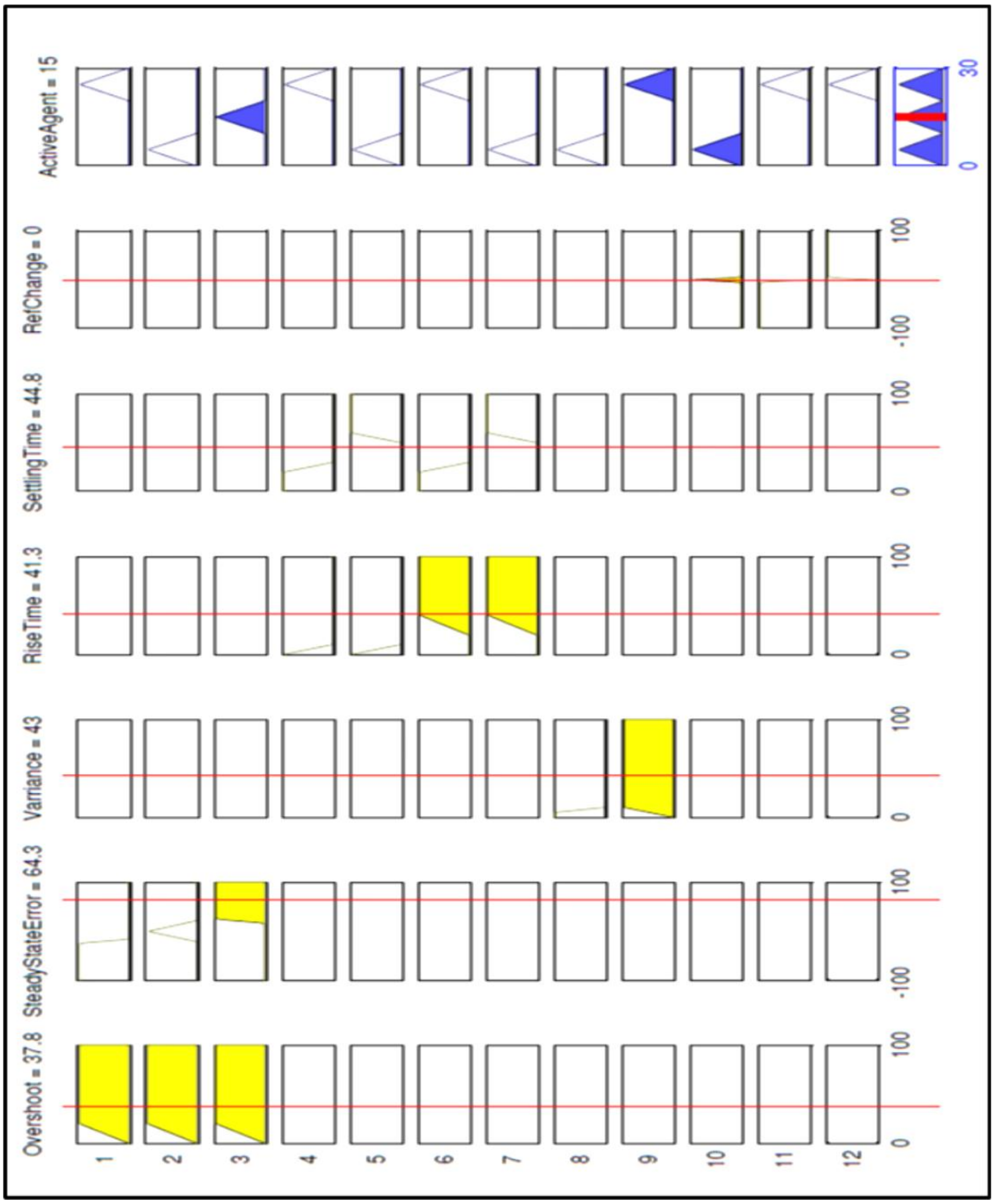


Figure 6.4 Controgent-2 (Fuzzy) is selected as fuzzy logic output based in various input parameters

.

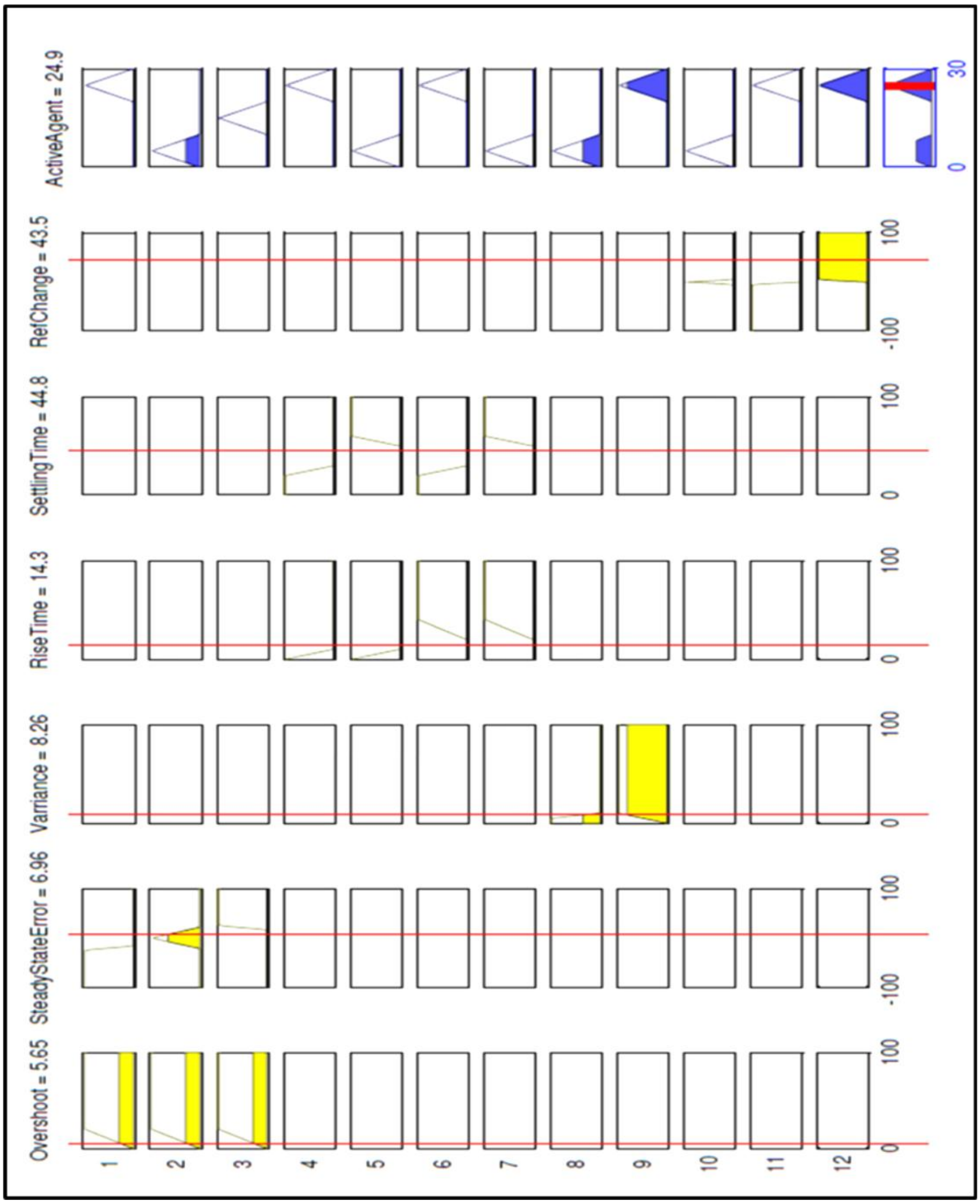


Figure 6.5 Controgent-3 (NN) is selected as fuzzy logic output based in various input parameters

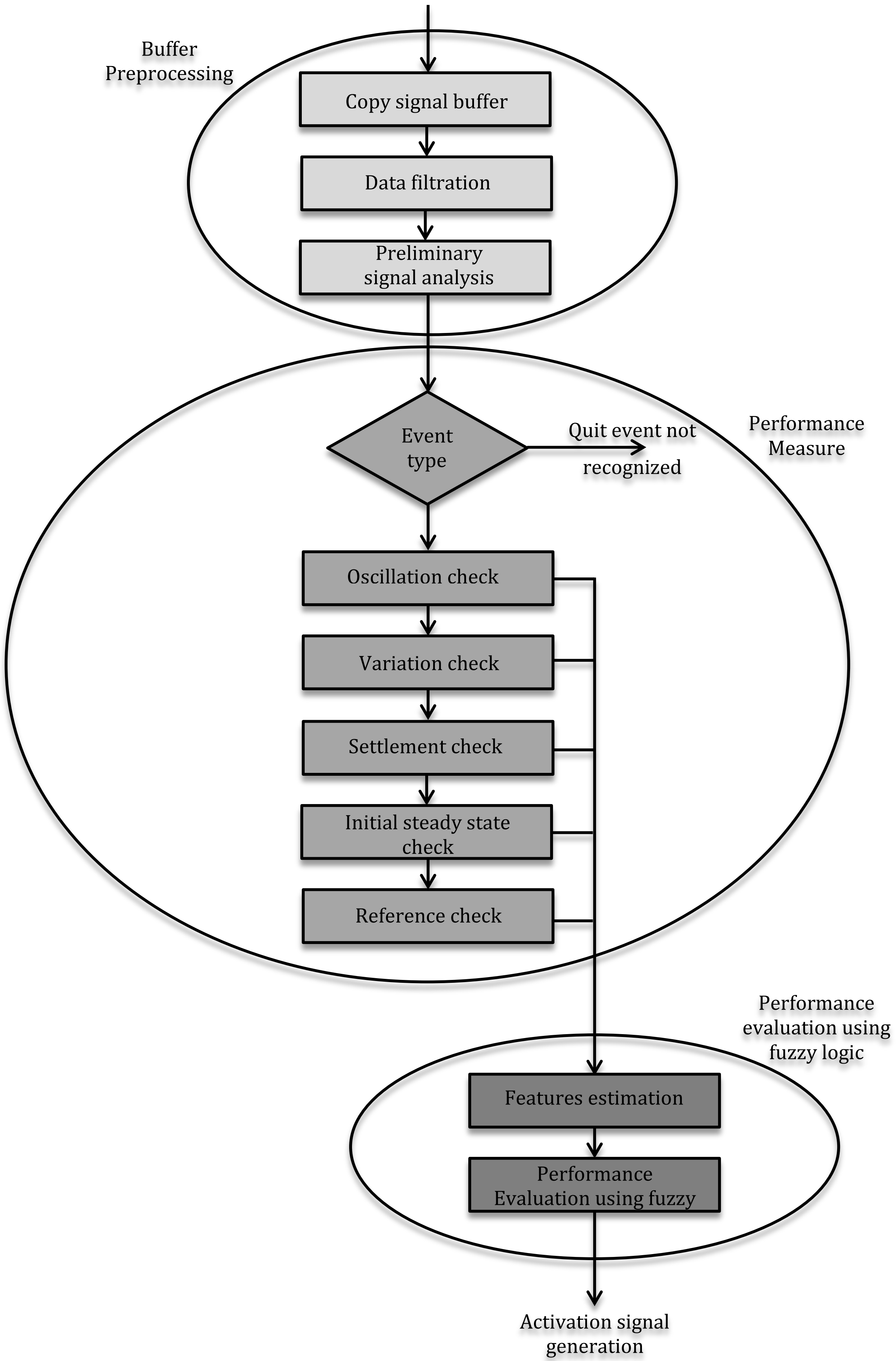


Figure 6.6 CPA overview

6.5 Concluding remarks

In this chapter, new technique for control performance assessment is developed for multi-agent control system. Various parameters for control performance are calculated online using performance measure. The fuzzy logic is used to identify the best control algorithm for complex system. From experimental study, the fuzzy set, fuzzy rules and fuzzy inference are developed for CPA. The overview of whole control performance assessment is discussed in detail. The next chapter presents the basic multi-agent controller design for SISO and MIMO systems.

Chapter 7
Classical MAC_SC

The multi-agent control framework is discussed in chapter 3. In this chapter, two multi-agent controllers are developed using MAC framework. Basic MAC is designed for helicopter control and Advance MAC_SC is designed for boiler-turbine control. Structuring of controls and its result analysis for both problems are given in this chapter.

7.1 Introduction

The previous chapters are discussed the basic multi-agent control architectures. Various agents are designed using different technique for partial control problem. The integrating solution of various agent output are generated using coordination object. For optimized design of agent, here the agents are designed using soft computational paradigms as fuzzy logic, neural network and genetic algorithm etc… Design of basic multi-agent controller using soft computational paradigms is called classical MAC_SC. Two classical MAC_SCs are designed for basic MAC_SC and advance MAC_SC for helicopter and boiler-turbine plant respectively. The detail discussions are given in next sections.

7.2 Basic MAC_SC

Control system design is very different in some situations when the problem can be thought of being composed of an interconnection of a set of sub problems. But, some complex plants can be modeled as physical processes that are operating in a limited set of operating regimes [1]. Here, the dynamic behavior of the plant is changed due to changes in plant operating regime. Each operating regime is considered as sub problem. There are some conventional control techniques to control such plants, but each has its own drawback and constraint. The robust controller can be used to control the plant in all operating regimes, but this often results into a loss of performance. One more approach for this type of control problem is an adaptive controller. But it may respond too slowly to the instantaneous change in the plant dynamics [2]. When the plant's operating regime (parameters) changes, the corresponding control module is switched for particular control system. Multi-agent control architecture is suitable for this type of application.

7.2.1 System dynamics

The twin rotor CE 150 helicopter is shown in figure 7.1. Generally, processes dynamical behavior may change slowly or instantaneously [3], [4]. Dynamics of helicopter can be varied with two different ways: (1) Moving small weight (2) Changing elevation angle. But the effect of angle changed is very important compare than small weight changed. The movement in vertical direction is nonlinear in helicopter model and also dynamics change with elevation angle.

The plant's dynamic equation is obtained with writing torques equilibrations and combining them with DC motor dynamic equations. Torques equilibrations in vertical plane can be expressed as follows [3]:

$$I_m \ddot{\theta}_1 = T_u - T_f - T_m \tag{7.1}$$

Where

$$T_u = K\omega^2, T_m = T_g \sin(\theta_1 + \alpha), T_f = (\beta_{11}|\omega| + \beta_{21})\dot{\theta}_1$$

The Dynamical behavior of motor and propellers is:

$$I\dot{\omega} = \frac{k_i}{R}(u - k_b\omega) - k_1\omega^2 - k_2\omega \tag{7.2}$$

Where I_m denotes the moment of inertia around the horizontal axis, θ_1 is the elevation angle, T_u is the torque of the propulsion force of the main propeller, T_f is the friction torque, T_m is the torque of the gravitation force of the helicopter body, ω is the angular velocity of DC motor and I, R, α, β_{11}, β_{21}, k_i, k_b, k_2, k_1 are constants. Suppose that

$$b_1 = \frac{k_b k_i + Rk_2}{k_i}, b_2 = \frac{k_i R}{k_i}, I_r = \frac{IR}{k_i} \tag{7.3}$$

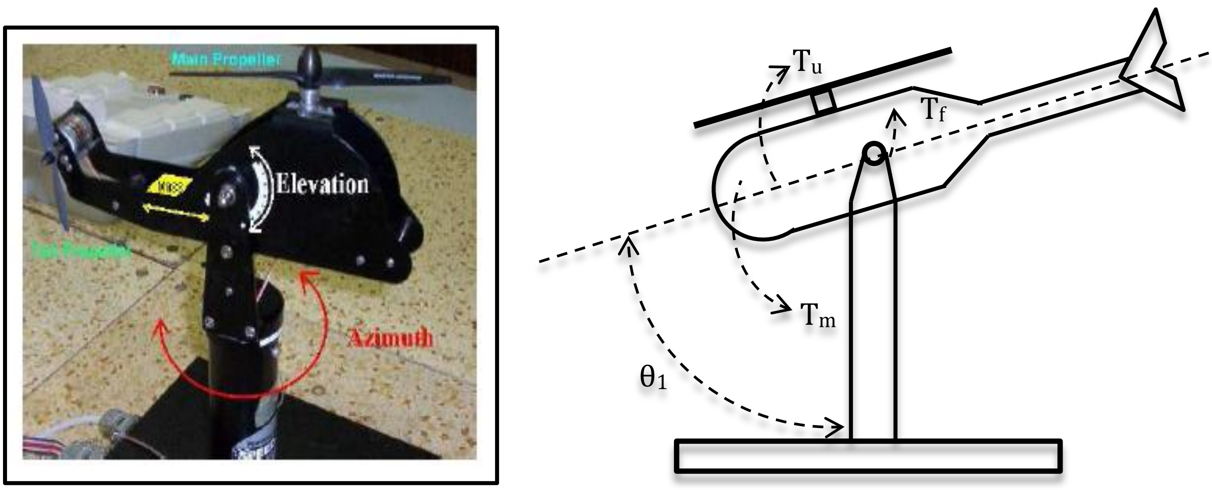


Figure 7.1 Helicopter CE 150 (Courtesy of Humusoft)

Table 7.1 Parameters for helicopter

Tg	α	b_1	b_2	k_1	I_m	I_{r1}	β_{11}	β_{21}
9.55* 10^-2	6.87* 10^-2	3.21* 10^-2	5.10* 10^-2	2.97* 10^-2	4.30* 10^-2	5.55* 10^-2	0.0043* 10^-4	1.4178* 10^-4

By defining the state variables $x_1 = \theta_1,\ x_2 = \dot{x}_1 = \dot{\theta}_1$ and $x_3 = \omega$, the state space description of elevation direction of the laboratory helicopter is obtained [5] using 90° elevation angle and the 0.6 control input as operating parameters:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \qquad 7.4$$

Where
$$A = \begin{bmatrix} 0 & 0 & 1.000 \\ 0 & -6.091 & 0 \\ -22.167 & 2.361 & -0.033 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 18.018 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, D = [0]$$

7.2.2 Structuring of control problem

Before applying the agent base control approach, the structuring of control problem is required to identify for agent design and its coordination mechanism. The structuring of control problem is given in figure 7.2 with reference section 2.3. The proportional-integral-derivative controller is well proven for various industry applications. But the control algorithm with fixed parameters is not suitable for the nonlinear system with different operating system. As per operating condition, the parameters of control algorithm must be tuned. So, the control problem is decomposed in five compound control problems for various five operating regimes. Each compound problem is divided in two elementary problems. Various agents are designed in next section for each elementary control problem.

7.2.3 Basic MAC_SC design

All partial sections of control problem are used an object called a controgent as building block to build the whole controller. A controgent implements a locally operating control algorithm and contains several elements that are needed to carry out its tasks. One is local control algorithm and the tuning agent is for tuning the parameters of controller. The operating parameter of system is changes; the respective controgent is activated for respective operating regime. The whole operating regime is divided in different five local operating regimes. Different controgent is designed for respective regime with best overall performance. Competitive coordination is used as per section 2.4 and the architecture is developed as shown in figure 7.3.

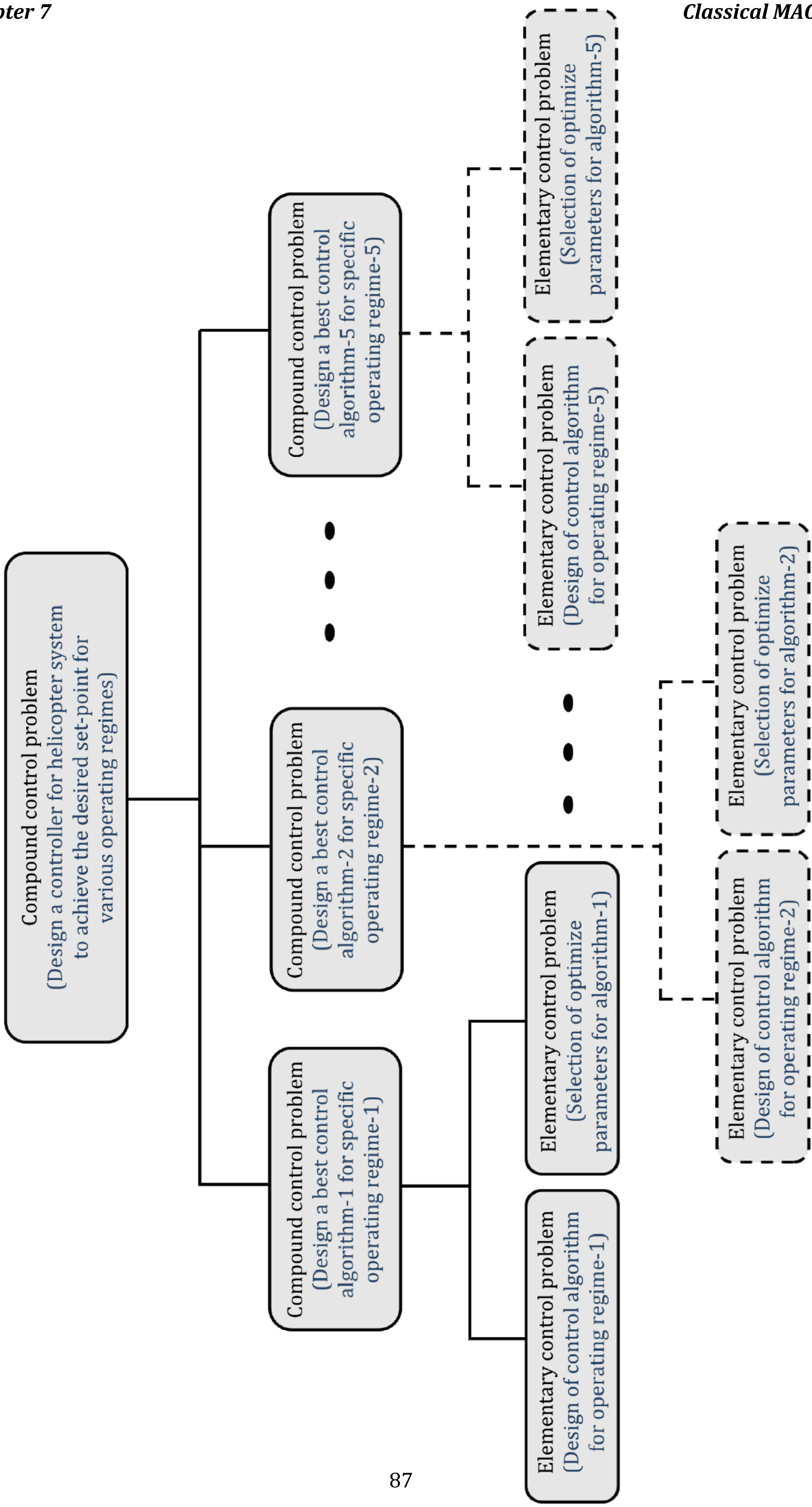


Figure 7.2 Structuring of helicopter control problem

PID controller is a feedback controller that makes a plant less sensitive to change in the surrounding environment as well as to small changes in the plant itself so it is used as control algorithm for local operating regime [6]. It is known as PID agent. The continuous form of a PID controller, with input e and output u, is given in equation 2.1.

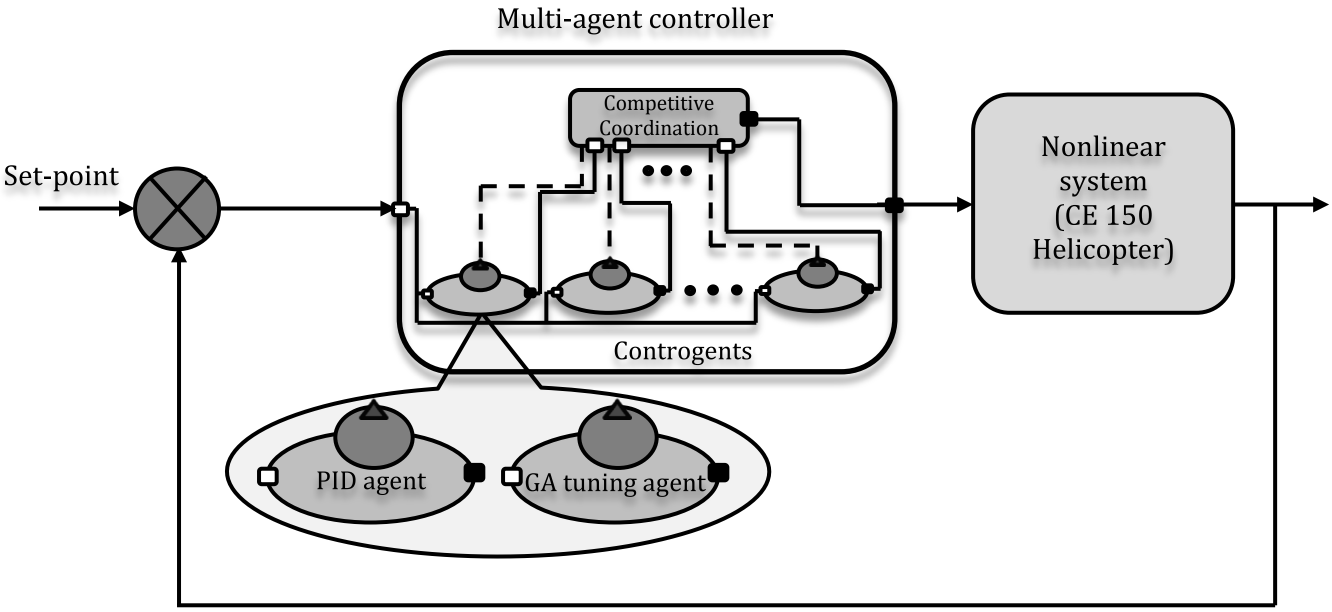


Figure 7.3 Basic MAC_SC for helicopter

Genetic Algorithm (GA) is implemented as tuning agent in multi-agent controller to optimize the each PID parameters for specified objectives. The GA used in this is very similar to the algorithm that can be found in the standard literature on the topic [7]. The fitness of each candidate is evaluated through some appropriate measure [8]. In the control theory, the objective is to minimize the cost function, defined as the integral absolute error (IAE) which determines the performance of the controller for whole control problem. Thus, the cost function and fitness function is written as:

$$J(t) = \int_0^\infty e(t)dt, \ \& \ Fitness = \frac{1}{J(t)} \tag{7.5}$$

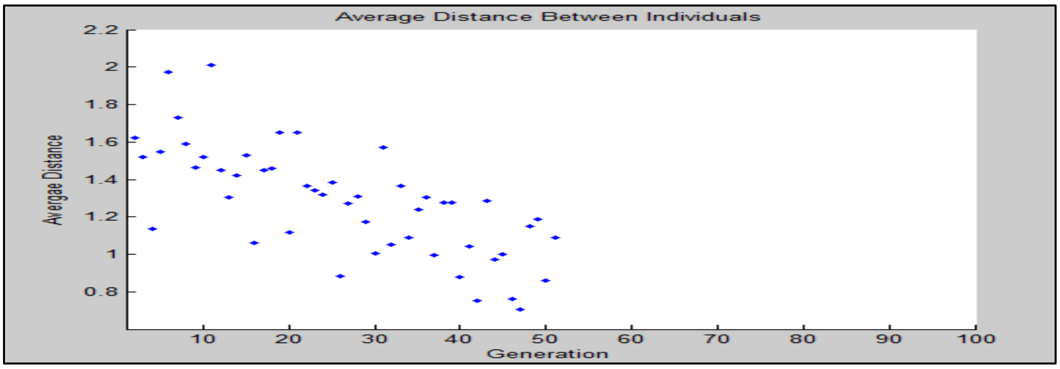
GA parameters are selected for the training cycle as per table 7.2.

Table 7.2 Parameters for genetic algorithm

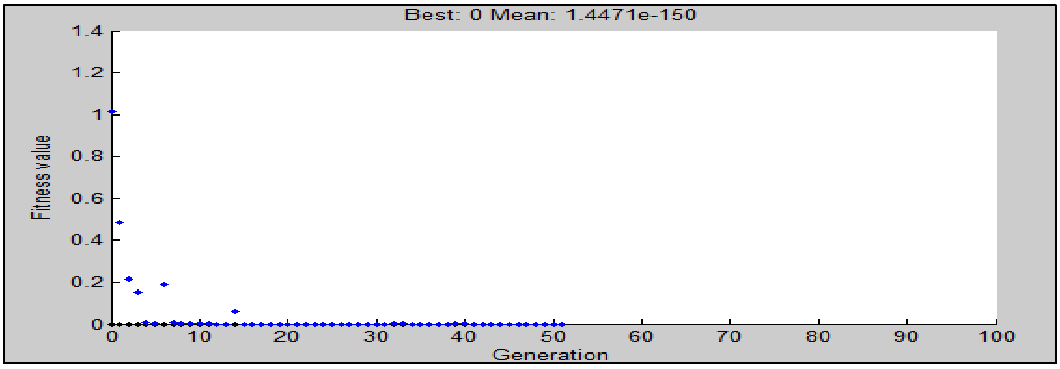
Population Size	Crossover Rate	Mutation	Terminating Generation	Stall Time Limit	Selection
100	0.9	0.01	51	100	Roulette Wheel

The various plots for training are given figure 7.4.

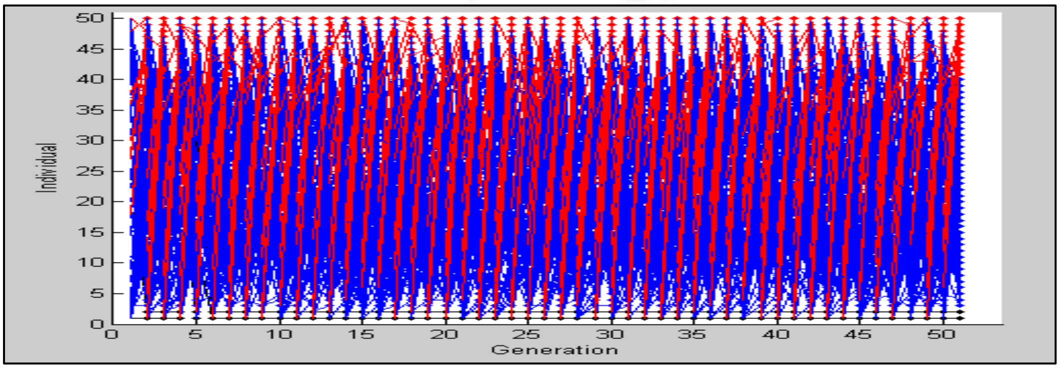
7.2.4 Result analysis



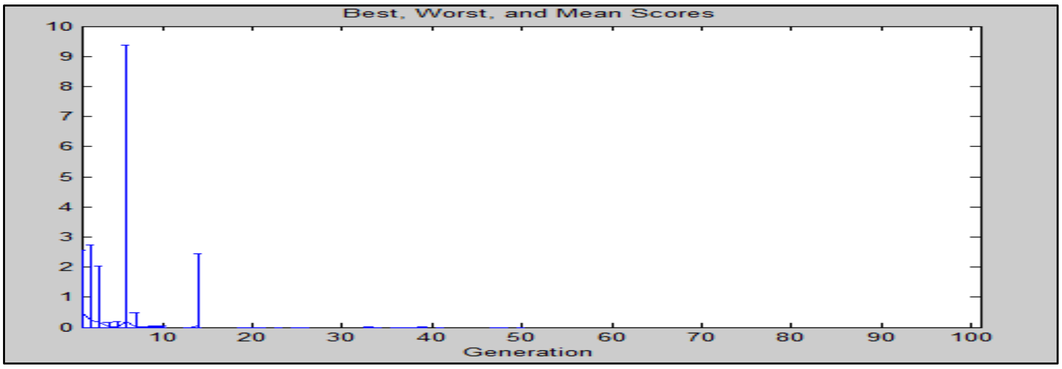
(a. Fitness function)



(b. Genealogy)



(c. Individual)



(d. Generations)

Figure 7.4 Genetic algorithm tuning plots

The single loop PID controller tuned with genetic algorithm is applied on the helicopter model and the different operating parameters are used to see the overall performance. The result is given in figure 7.5. The following two design criteria are selected to determine the performance:

- Overshoot 0 %
- Settling time T_s <4 sec.

Here, the performance of tuned with single operating parameters PID controller is different in each operating regime [9].

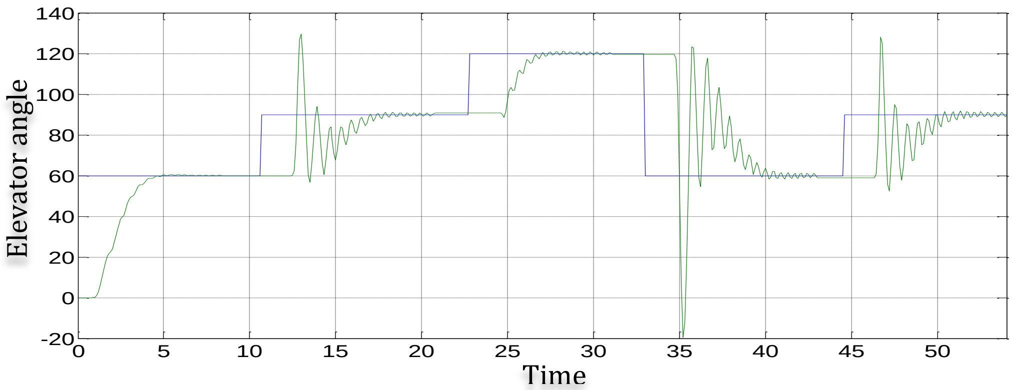


Figure 7.5 Performance of PID controller

In this the multi-agent controller is very suitable compare than traditional controller in this case#. The separate PID control agent is tuned with genetic algorithm for different operating range. It will be activated on respective operating regime. The multi-agent controller result is given in figure 7.6. The performance of this controller is improved compare than tradition controller.

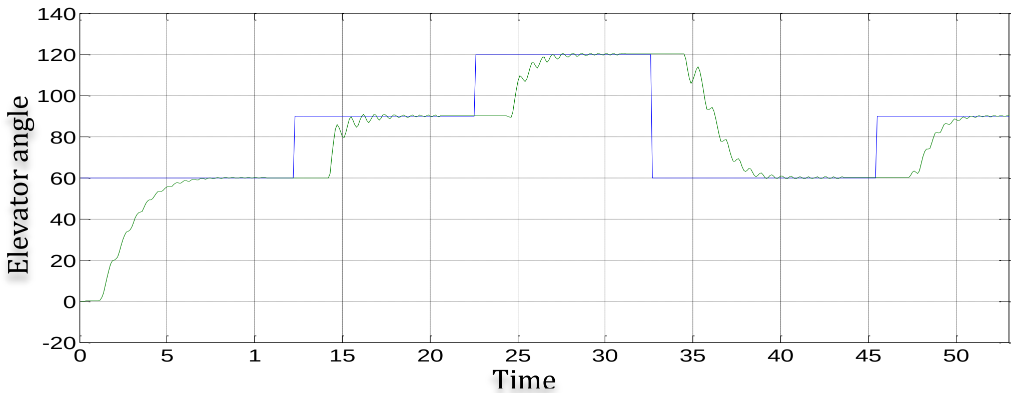


Figure 7.6 Performance of basic MAC_SC

 # paper entitled "GA based competitive multi-agent controller for non-linear system" is published in 3rd International Conference on Mechanical and Electrical Technology, ICMET-2011, Dalian, China, ISBN: 978-0-7918-5981-0, August 26th-27th 2011. (Publisher: ASME press, New York, USA, DOI: http://dx.doi.org/10.1115/1.859810.paper186)

The basic MAC is developed using multi-agent control architecture. It is implemented on CE 150 helicopter. The superiority of basic MAC for rapidly changing the operating parameters systems is investigated by comparison with tradition methods. This architecture is suitable for single input and single output (SISO) system only. The new advance MAC architecture is developed for multi-input multi output (MIMO) system and it is discussed in next section.

7.3 Advance MAC_SC

Control engineers are sometimes deal with real time control system in which they have to design the controller with multiple control algorithms and integration technique in place of single algorithm control [10]. These situations occur when the system is covered with complexity and nonlinearity components. It can be composed the set of interconnected partial control problem. Since in real time systems, initial condition of control variables or operating condition is changed continuously. So, the derived the control algorithm based on particular initial and operating parameters may or may not be valid for all operating conditions. Hence control scheme needs to be modified to accommodate change in environment. So, the advance MAC_SC is developed for multi input multi output (MIMO) of system.

7.3.1 Control problem description

Conventional methods help to design one robust controller that controls the plant in all operating regimes, but sometime it does not provide optimal operation for the current operating regime [9]. The conventional coupled PI controller is used to obtain satisfactory closed-loop system performance due to ease implementation, simple structure and robustness. The modern controller is also useful with more stability in some of operating regime of plant compare then PI controller. So, the signal control algorithm based controller is not suitable for all operating regime of boiler turbine plant. For this type of case, the new controller based on multi-agent concept is developed. The plant model and structuring is already given in section 2.3.5 for boiler turbine plant. The problem was structured, elementary control problems were defined and individual control algorithms were proposed, the partial solutions were organized into an overall solution and finally the controller was specified. First, the control problems are identified in partial problem in the structure diagram of figure 2.9. There are two elementary control problems and one compound control problem:

- Designing a *supervisor that* generate activation signal for different controller. It is compound control problem;
    - → Designing a *performance measure* to identified plant parameters based on step response.

→ Designing a *critic* for decision making based on plant parameters.

- Designing an optimal *Coupled PI controller* based on classical control theory.
- Designing an optimal *Linear Quadratic Regulator* (LQR) Controller based on modern control theory.

7.3.2 Advance MAC_SC design

The solutions of these control problems will be implemented by controgents. They are labeled respectively: Supervisor Agent, Critic agent, Performance Measure Agent, Classical Agent, and Modern Agent. Figure 7.7 shows the overall organization of the solution. Notice the preservation of the structure of the original control problem (see figure 2.7).

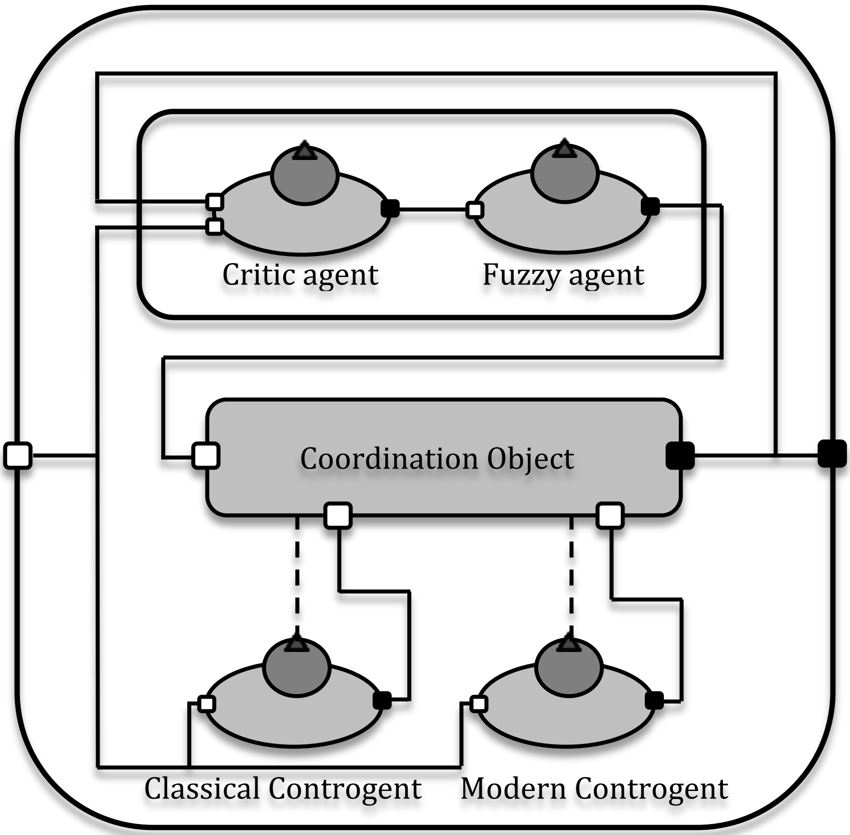


Figure 7.7 Advance MAC_SC

The supervisor agent is designed to active the suitable agent based on performance parameters of close loop system [11,12]. The work of supervisor agent is structured in two sections: performance measure agent who identifies the performance parameters and critic agent who is identify the suitable agent with help of soft computable paradigms and its pre-defined desired response.

- **Performance measure agent:** The behavior of plant model which is derived from operating points and initial condition is characterized through the online measurements of parameters of overshoot, variance of output, steady state error and change of reference as per discussed on section 6.2.
- **Critic agent**: This agent is designed using fuzzy logic and four input parameters from output of critic agent are given to identify the performance of control agents in respective regime as per covered in section 6.3. The membership functions are selected as per table 7.3 and the rule base is selected as per table 7.4.
- **Classical agent:** The boiler-turbine plant is multi input multi output model. The PI algorithm as classical agent is implemented on boiler turbine plant with six coupling gain to reduce interaction between loops. It is too much difficult to tune all 12 Proportional, integral, and cross coupling gains using simple methods illustrated in figure 7.8.

Table 7.3 fuzzy sets for input variables

Parameter	Actual range	Range in percentage (%)	Name of Mf	Type of membership	Mf range
Overshoot	0 to 500	0 to 100	Low	Triangle	[-5 0 100]
			High	Trapezoidal	[0 20 20 100]
Steady state error	-5 to 5	-100 to 100	N-error	Trapezoidal	[-100 -100 25 16]
			M-error	Triangle	[-21 0 21]
			P-error	Trapezoidal	[16 25 100 100]
Variance in output	0 to 40	0 to 100	Low	Trapezoidal	[0 0 5 100]
			High	Trapezoidal	[0 10 100 100]
Reference change	-15 to 15	-100 to 100	Dec	Trapezoidal	[-100 -100 -5 0]
			Norm	Triangle	[-10 0 10]
			Inc	Trapezoidal	[0 5 100 100]

Table 7.4 Rule base for fuzzy logic

Function related	Selected controgent
High undershoot & overshoot with high variance	Classical controgent
low undershoot & overshoot with minimum variance	Modern controgent
output overshooting due to set point change	Classical controgent
low steady state error	Modern controgent

It is difficult task to tune these all gains using tradition tuning method. Genetic algorithm discussed in section 4.3 is used to tune the various gains. The cost function is selected as per below:

$$J = \sum_{i=1}^{3} e_i(t_f) \quad \text{Where} \quad e_i = y_i(t) - y_{iref}(t) \qquad 7.7$$

- **Modern agent:** The modern agent works on modern control theory mean it is a Linear Quadratic Regulator (LQR) [13,14]. The optimal control algorithm is designed using Riccati Equation as per below:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$
$$K = R^{-1}B^T P \qquad 7.8$$

Here, the controller is developed using the following infinite time horizon quadratic performance index and input weighting matrices Q and R:

$$J = \int_0^{\infty} (X^T QX + U^T RU)dt \qquad 7.8$$

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 724 & 0 & 0 \\ 0 & 141 & 0 \\ 0 & 0 & 5 \end{bmatrix} \quad K = \begin{bmatrix} 0.0359 & 0.0011 & 0.0031 \\ -0.0057 & 0.0770 & -0.0064 \\ -0.0413 & 0.0232 & 0.4444 \end{bmatrix}$$

A feed forward gain matrix (F) is added to the system as shown in figure 7.8 to ensure state tracking. This is obtained by genetic algorithm which is described in section 4.3, so that the system tracks step demands in the steady state. To achieve this following performance index is selected to be minimized:

$$J = \int_0^{\infty} ((y - y_{ref})^T Q(y - y_{ref}) + U^T RU)dt \qquad 7.9$$

- **Coordination Object:** The competitive type coordination object is used in this MAC_SC design as per discussed in section 3.6.

7.3.3 Result analysis

The response of advance MAC_SC is analyzed using graphical user interface and Simulink in matlab 7.10 [15] #. The graphical user interface is developed to check the response of controller and its comparative result analysis as per figure 7.5-7.11 Figure 7.8 depicts the main menu of advance MAC_SC.

#A paper entitled "Multi-agent based controller for boiler turbine plant" is accepted to published in journal of control and intelligent after review process, ACTA press, March 2013.

the best controgent for particular operating regime as per given in figure 7.10. The structure of classical agent using coupled PI control and tuned gains using genetic algorithm are given in figure 7.11. Similarly, the modern agent is shown in figure 7.12. In last, the overall controller response is given in figure 7.13.

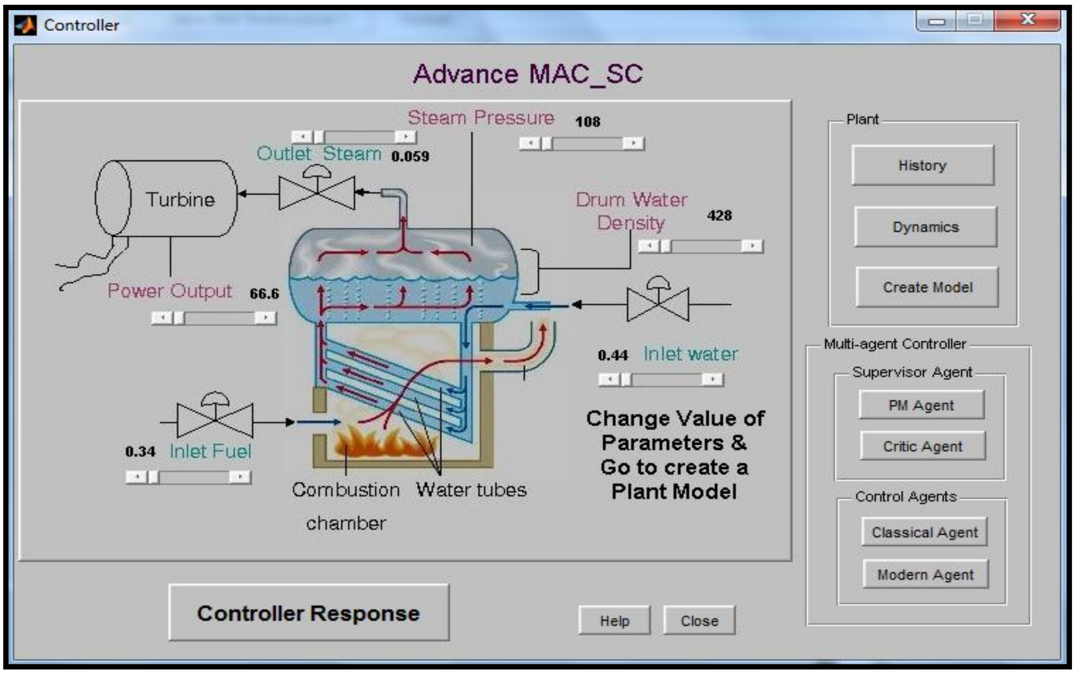


Figure 7.8 GUI for Advance MAC_SC

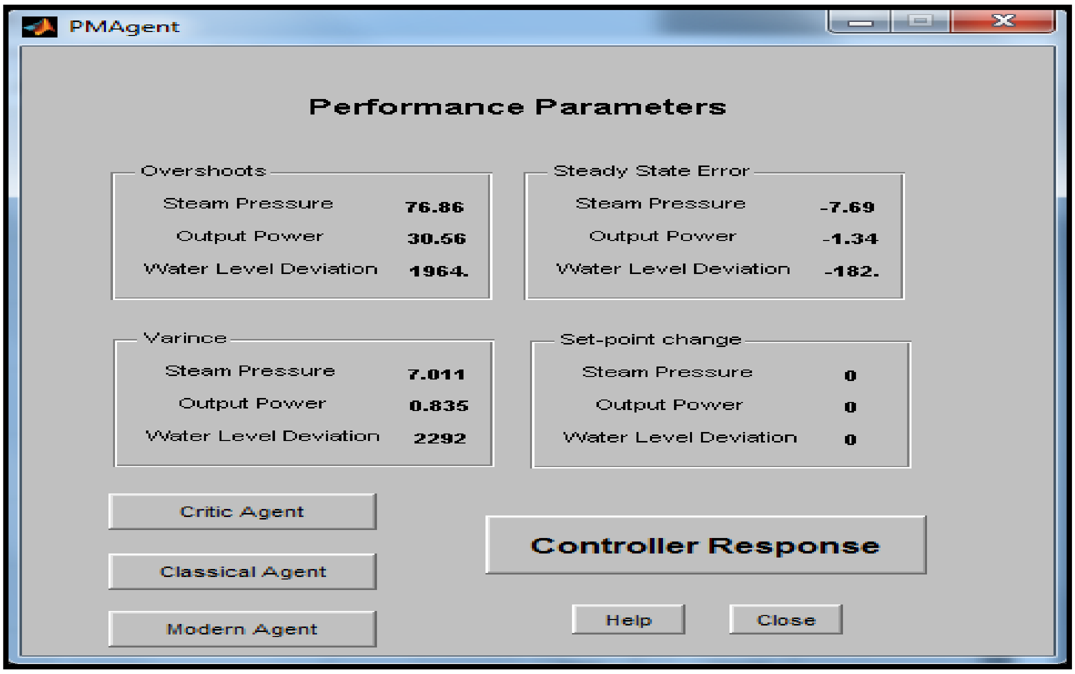


Figure 7.9 Performance parameters of plant

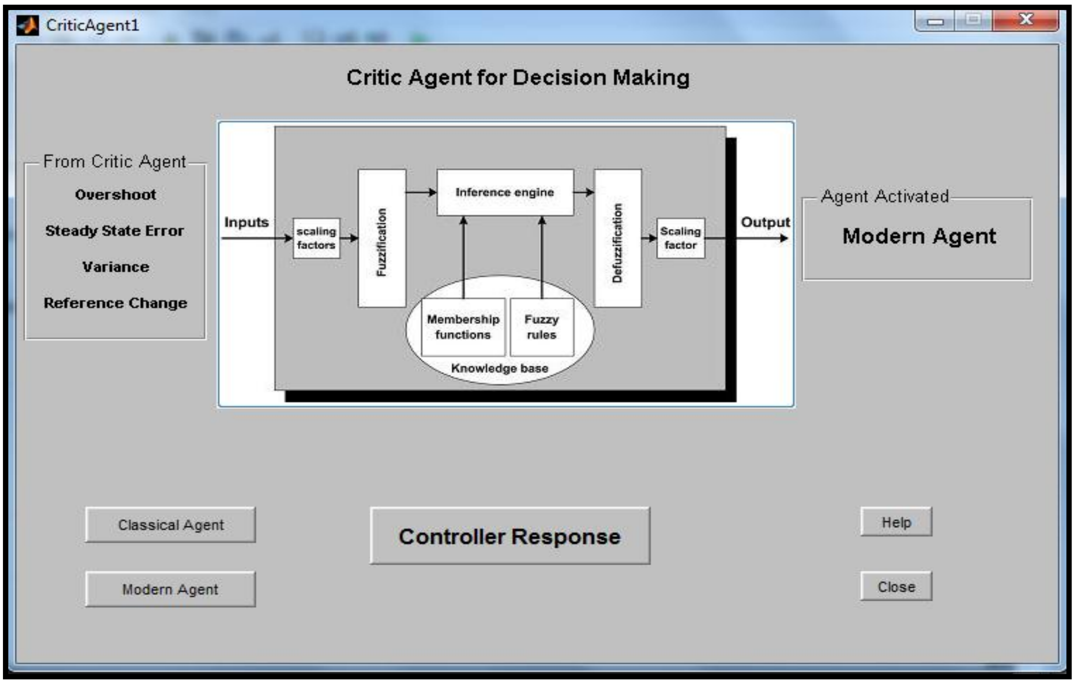


Figure 7.10 Critic agent response

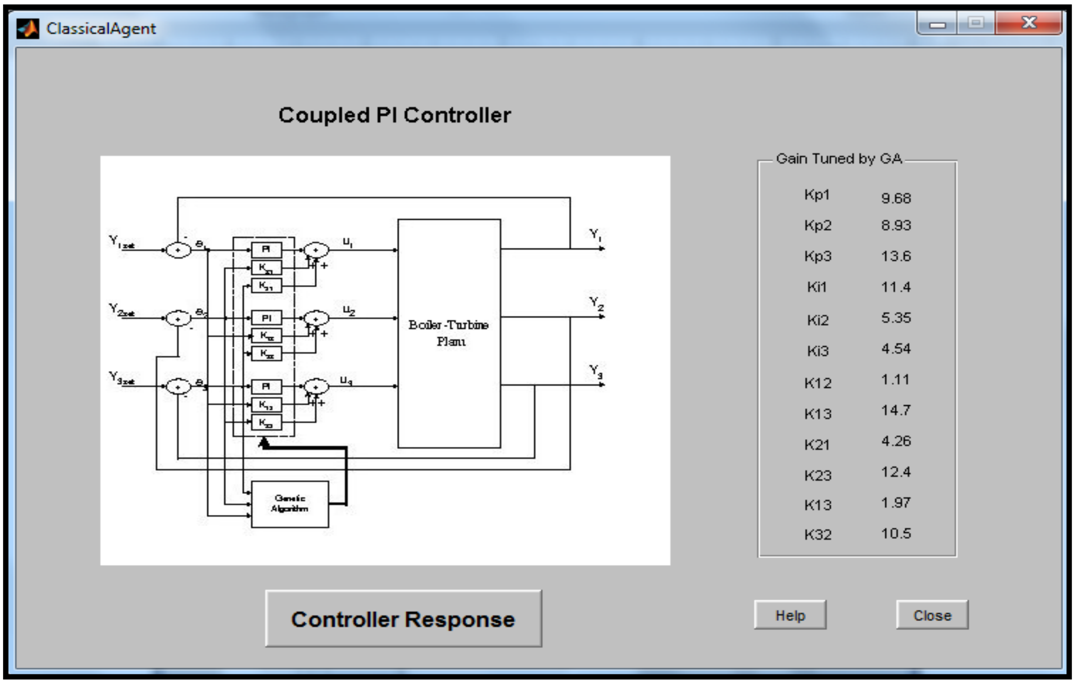


Figure 7.11 Classical agent

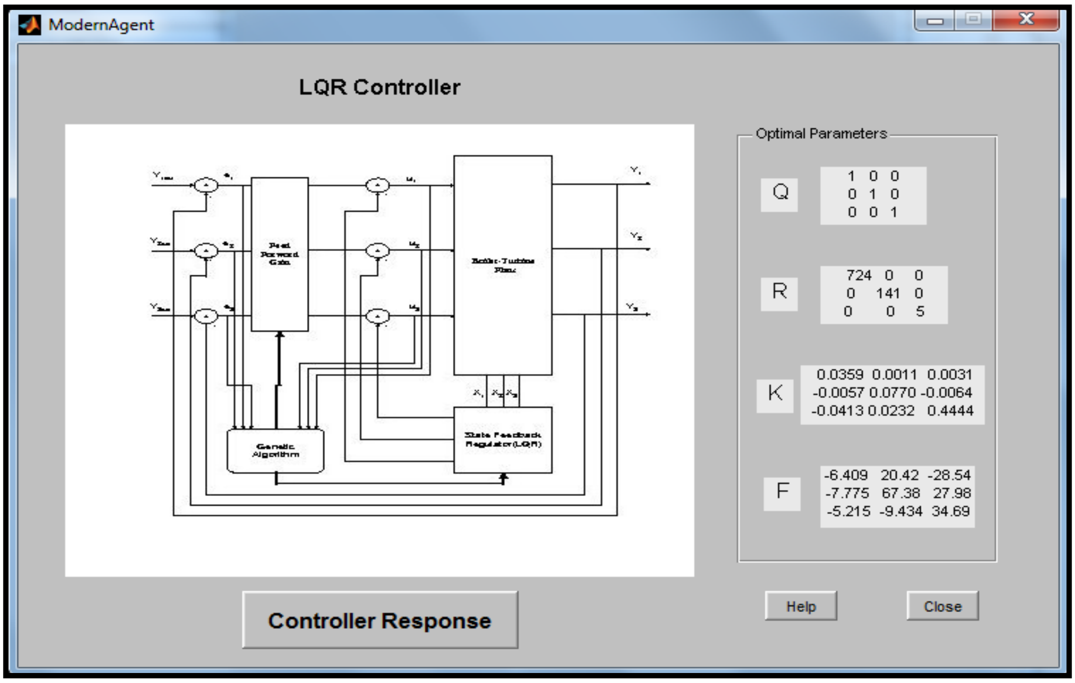


Figure 7.12 Modern agent

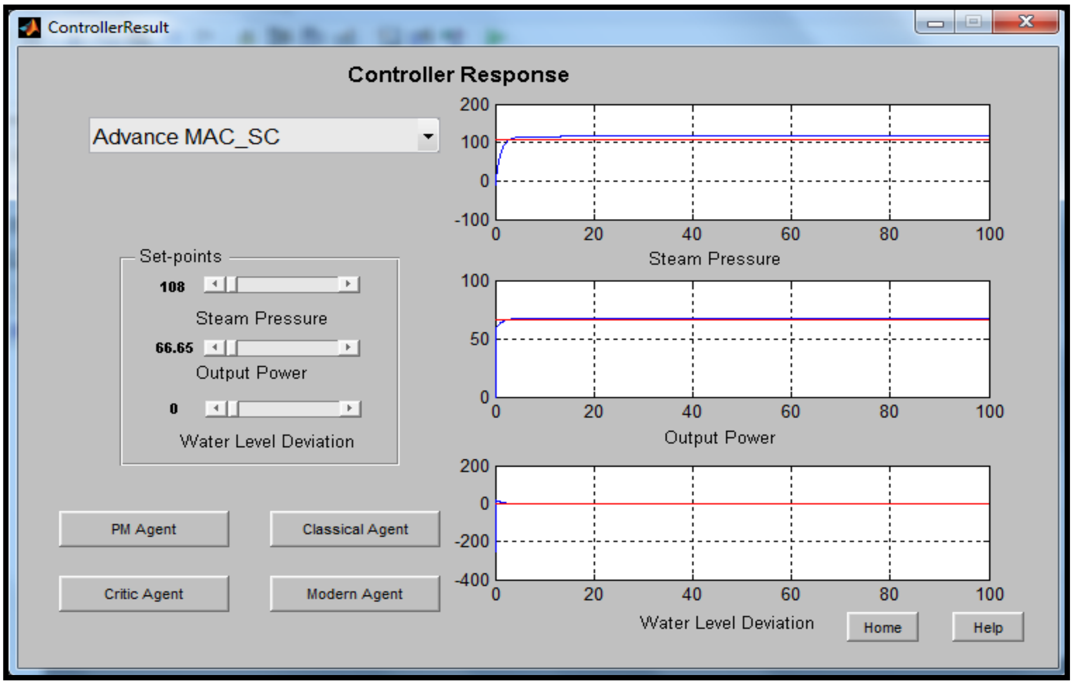


Figure 7.13 Controller result analysis

The multi-agent architecture based controller was applied on boiler turbine plant with various operating condition from case a-g as table 7.5. It shows controller result, performance parameters and selection of control agent for various operating points with changed of 70% to 130 % from normal operating points and some random value of process parameters. The overshoot of power output and water level deviation are less so the classical controgent is activated for case a. Similarly, other cases from b-f are given in table 7.6 with performance parameters. The operating condition is remained same but the set point is changed, so, the controller action is changed which is indicated in table 7.7 with various result. Finally, the best controgent is activated by proposed advance MAC_SC based on performance of process performance parameters for all cases [#].

Table 7.5 Performance parameters and controller response for various operating condition without change set-points of any process parameters.

	Different Operating Points						
Case	**a**	**b**	**c**	**d**	**e**	**f**	**g**
Change parameter form normal operating	**70%**	**85%**	**100%**	**115%**	**130%**	**R_1***	**R_2***
X_1	75.6	91.8	108	124.2	140.4	130.9	101.4
X_2	46.65	56.65	66.65	76.64	86.64	64.8	86.04
X_3	299.6	363.8	428	492.2	556.4	549.2	422.5
U_1	0.238	0.289	0.034	0.391	0.442	0.760	0.550
U_2	0.041	0.050	0.059	0.670	0.760	0.060	0.770
U_3	0.304	0.369	0.435	0.500	0.565	0.530	0.490
	Steam Pressure						
Overshoot	100%	98.6%	100%	54.20%	48%	130%	63.7%
Variance	95.6%	100%	99.85%	27.12%	21%	110%	38.7%
Error	-10%	-13.4%	13%	-5.42%	-4.85%	13.2%	-6.3%
	Power Output						
Overshoot	73%	89.5%	100%	76.17%	85.0%	110%	61.3%
Variance	18.36%	26.4%	35%	39.43%	58.0%	67.0%	18.6%
Error	-10%	-13.4%	-13%	5.42%	-4.85%	-10.1%	-6.3%
	Water Level Deviation						
Overshoot	42%	46.51%	51%	68.76%	72%	58.5%	66.6%
Variance	15.3%	17.69%	20%	35.53%	39%	26.2%	33.5%
Error	-4.6%	-5.03%	-5%	-7.10%	-7.44%	-6.1%	-6.9%
	Advance MAC_SC						
Activated Agent	Classical	Classical	Classical	Modern	Modern	Classical	Modern

Table 7.6 Controller response for various set points changes in process parameter with same operating condition

	Different Operating Points					
X_1	120	120	120	120	120	120
X_2	66.65	66.65	90	90	58	58
X_3	500	500	475	475	450	450
U_1	0.31	0.31	0.69	0.69	0.44	0.44
U_2	0	0	0	0	0.25	0.25
U_3	0.304	0.304	0.43	0.43	0.56	0.56
	Set Points Change					
Steam Pressure	120	80	120	80	80	80
Power Output	100	50	100	50	90	100
Level Deviation	0	0	0	0	0	0
	Advance MAC_SC					
Activated Agent	Classical	Modern	Classical	Modern	Classical	Modern

Table 7.7 Process parameters performance and activated agent for cases a-g.

Cases	Classical Agent	Modern Agent	Activated Agent by controller
a	Small Overshoot	Large Steady State Error	Classical Controgent
b	Small Overshoot	Large Steady State Error	Classical Controgent
c	Normal	Low Variance	Classical Controgent
d	High Variance	Low Overshoot, Low Variance	Modern Controgent
e	High Overshoot	Low Variance Low Overshoot	Modern Controgent
f	Low Variance	Large Steady State Error	Classical Controgent
g	High Overshoot & Variance	Low Overshoot & Variance	Modern Controgent

7.4 Concluding remarks

Two type of MAC_SC designs are covered basic MAC_SC and Advance MAC_SC. In first design, the control algorithm is remained fix and the suitable parameters are tuned according the operating regime. This problem is solved using the basic MAC_SC design. But, it is not suitable for different control problem. In second design, the different control algorithms are designed for different operating regime. The best control algorithm is selected for specific operating regime. The problem is solved using advance MAC_SC. But, in this design, the control performance assessment is done using initial condition of plant model parameters. So, the new design of MAC_SC is required for online identification of actual model than the control performance is given in better way. The Intelligent MAC_SC will be discussed in next chapter.

Chapter 8

Intelligent MAC_SC

Intelligent MAC_SC are designed using soft computational paradigms. Control system design of a simplified water tank System as an application of the proposed methodology and process will be implemented. Detailed introduction of simulation and actual plant will be given in chapter. A step by step procedure for designing MAC controller using the proposed methodology and process will be described in detail.

8.1 Introduction

In the previous chapter, Basic MAC_SC and advance MAC_SC were discussed for helicopter and boiler-turbine plant respectively. Basic MAC_SC was developed for plant using fix control algorithm and different tuning parameter for different operating regimes. In second case, Advance MAC_SC was developed using control performance assessment and various control algorithms for different operating regimes. But, the system identification based multi-agent architecture is required for non-linear system. So, the Intelligent MAC_SC is designed using soft computational paradigms and the detail discussion is in next sections.

8.2 Structuring of control problem

In the previous chapter, two MAC architectures are discussed but they are application specific. That mean they are not generalized design for all control problems. New Intelligent Multi-Agent Controller with Soft-Computational paradigms (Intelligent MAC_SC) is designed and discussed in detail which is useful for nonlinear and continuous variable system. This algorithm can be applied for any control problem after some modification. It is applicable for SISO or MIMO both types of systems. The structuring of control problem is given as per figure 8.1. There are two compound control problems:

- Designing a *supervisor that* generate activation signal for selection of control algorithm. It is distributed in three elementary control problem;
  - Designing an *Identifier* to identify plant model.
  - Designing a *performance measure* to calculate the performance parameters.
  - Designing a *critic* for decision making based on performance parameters.

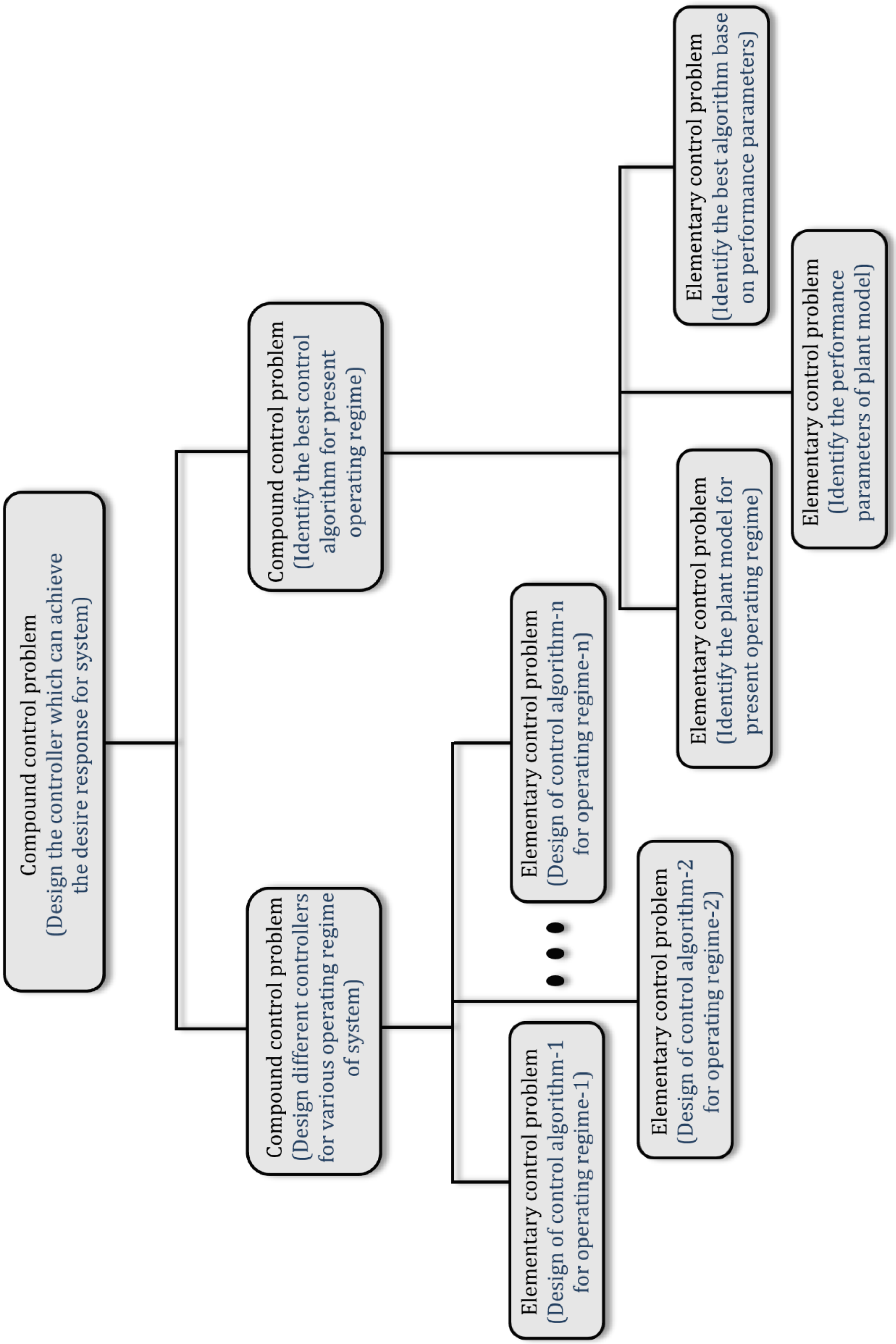


Figure 8.1 Structuring of generalized control problem

- Designing a best controller for particular operating regime. It is distributed in various elementary control problems;
    - Designing a *Controller* for best performance on respective operating regime.

8.3 Intelligent MAC_SC

Various agents are designed to solve the elementary control problem with specific goal and predefined active region. This is a layered structure of MAS in which different layers have different objective. With help of structuring, the multi-agent architecture is designed as per figure 8.2. The overall distribution is in four sections: Identification, Performance Analysis, Control algorithm, Coordination. Figure 8.2 shows the architecture of multi-agent and its components: Signal pre-processing agent (SPA), Identification agent (IA), Performance measure agent (PMA), Citric agent (CA), Controgent agent (CGA) and Coordination object (CO).

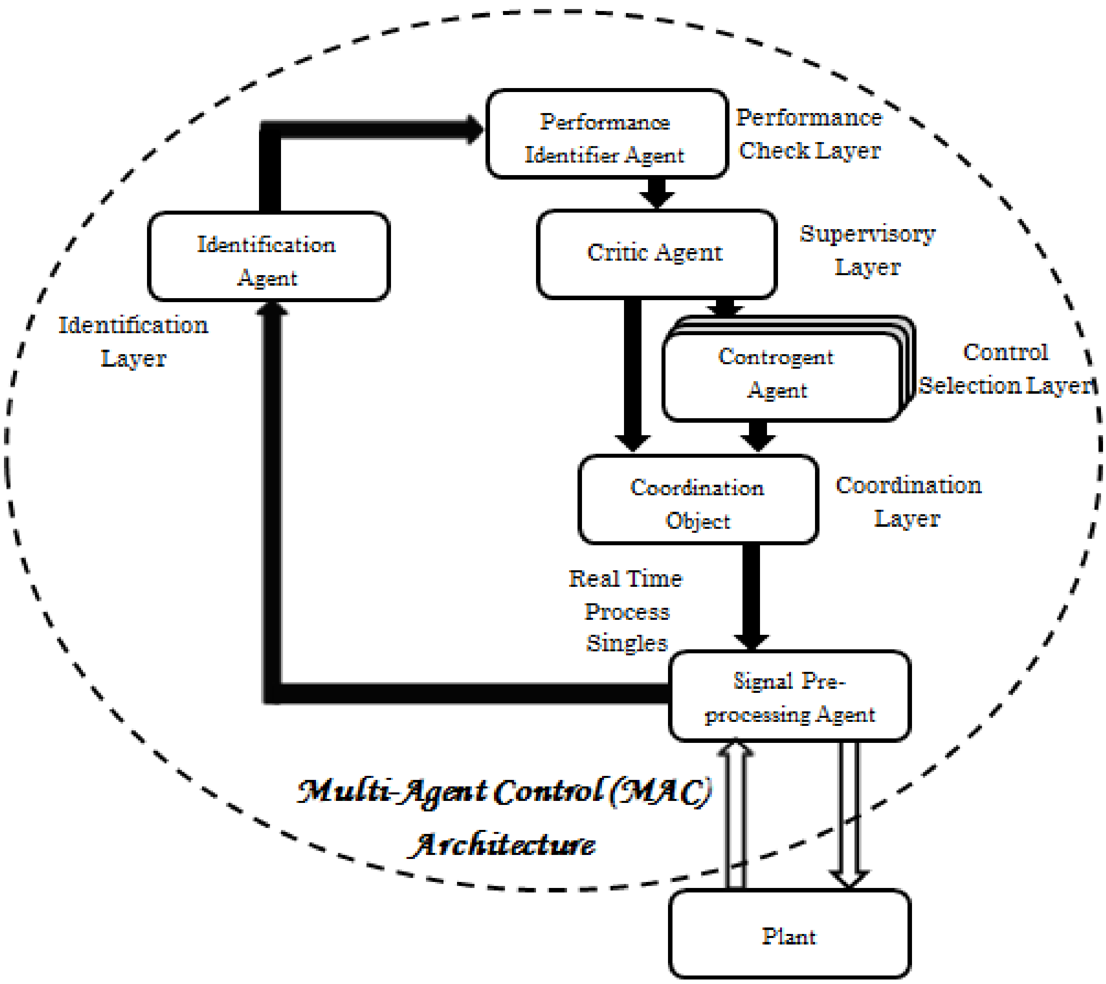


Figure 8.2 Block diagram of MAC architecture

8.3.1 Supervisor Unit

Supervisor is distributed in three separate objectives as identifier, performance measure and critic. Three agents are designed to archive these goals as below:

- **Identification Agent (IA):** The IA scans the buffer of recent real-time signals, prepared by the signal processing agent (SPA), and estimates the parameters of the local models that are excited by the signals. The local model is identified using nonlinear autoregressive network with exogenous inputs (NARX) approach base on present condition of system as per discussed in chapter 5. The new estimated model is passed to PMA.
- **Performance Measure Agent (PMA):** The performance measure scans the buffer of recent real-time signals for recognizable events. When events are detected, it estimates the features of closed-loop control response and an overall performance index as per discussed in section 6.2.
- **Critic Agent (CA):** The parameters of performance are copied from buffer of PMA. It is difficult to identify the suitable control algorithm directly for particular operating regime of available data. The critic agent (CA) is designed using fuzzy logic to generate the activation signal for coordination object as per discussed in section 6.3.

8.3.2 Control Unit

The control unit comprises an industrial nonlinear control algorithm with automatic tuning of its parameters. Normally, it is known as controgent. Several controgents may be used in the control unit and may be interchanged in the initial configuration phase. The following modes of operation are supported:

- **Manual mode:** open-loop operation (actuator constraints are enforced)
- **Intelligent mode:** a nonlinear controller or several modes with different intelligent methods for different operating regime.

Three controgents have been developed and discussed[#] as below:

- **PID Controgent:**

  It is designed as per discussed in section 7.2

[#]A paper entitled "Controller Performance Assessment (CPA) of intelligent control for non-linear system" is published in Journal of Control & Instrumentation, ISSN: 2229-6972, Vol.2, Issue 2-3, Pages 17-23, 2011. (Online:http://www.stmjournals.com/index.php?journal=JoCI&page=article&op=view&path%5B%5D=1378)

⌗ Fuzzy Controgent:

Fuzzy controgent is designed using fuzzy logic. Two variables are selected as input variables and one output variable [1,2]. The membership functions for input and output variable are selected as per table: 8.1. after number of experiments. The rule base is designed as per table: 8.2.

Table 8.1 Fuzzy set for variable

Parameter	Range	Name of membership	Types of Mf	Mf Range
Error	-10 to 10	HN	Trapmf	[-10 -10 -4 -1]
		N	Trimf	[-3 -1 0.5]
		S	Trimf	[-2 0 2]
		P	Trimf	[-0.5 1 3]
		HP	Trapmf	[1 4 10 10]
Rate of Error	-4 to 4	N	Trapmf	[-4 -4 -1 1]
		S	Trimf	[-1 01]
		P	Trapmf	[-1 1 4 4]
Output Voltage	-15 to 15	O	Trimf	[-8 -7 -6]
		OB	Trimf	[-3 -2 -1]
		M	Trimf	[-0.05 0 0.05]
		CM	Trimf	[1 2 3]
		C	Trimf	[6 7 8]

Table 8.2 Rule base

Rule Base				
Error (Input Variable)	Rate of Error(Input Variable)			
	Name of Mfs	**N**	**S**	**P**
	HN	**O**	**O**	**O**
	N	**OB**	**OB**	**M**
	S	**OB**	**M**	**CB**
	P	**M**	**CB**	**CB**
	HP	**C**	**C**	**C**

⌗ NN Controgent:

The NN controgent is designed using neural network. It is created directly based on the neural network NARMA-l2 model [3, 4]. The block diagram is shown in figure 8.3. The neural network controller [5] is obtained as below:

$$u(k+1) = \frac{y_r(k+d) - f \cdot [y(k), y(k-1), \ldots, y(k-n+1), u(k-1), \ldots, u(k-m+1)]}{g \cdot [y(k), y(k-1), \ldots, y(k-n+1), u(k-1), \ldots, u(k-m+1)] \cdot u(k)} \quad 8.1$$

Where is the reliable for d >= 0. The neural network is designed using the parameters given in table 8.4.

Table 8.3 Parameters for NARMA-l_2 controller

Number of hidden layers	9
Number of delay plant inputs	3
Number of delay plant outputs	2
Sampling rate	0.01
Number of epochs	100
Training algorithm	trainlm

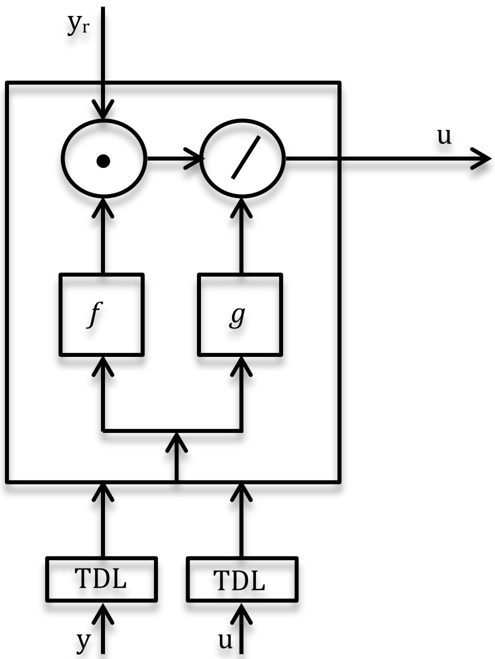


Figure 8.3 NARMA-l_2 Controller

8.3.3 Coordination & Preprocessing

- **Coordination Object (CO):** Hard conditional dependency between the elementary control problems, so, the competitive type is selected as per discussed in section3.7.
- **Signal Preprocessing Agent (SPA):** It will receive the data from the plant and send the controller signal to plant.

The final multi-agent controller (MAC_SC) is designed as per figure 8.4.

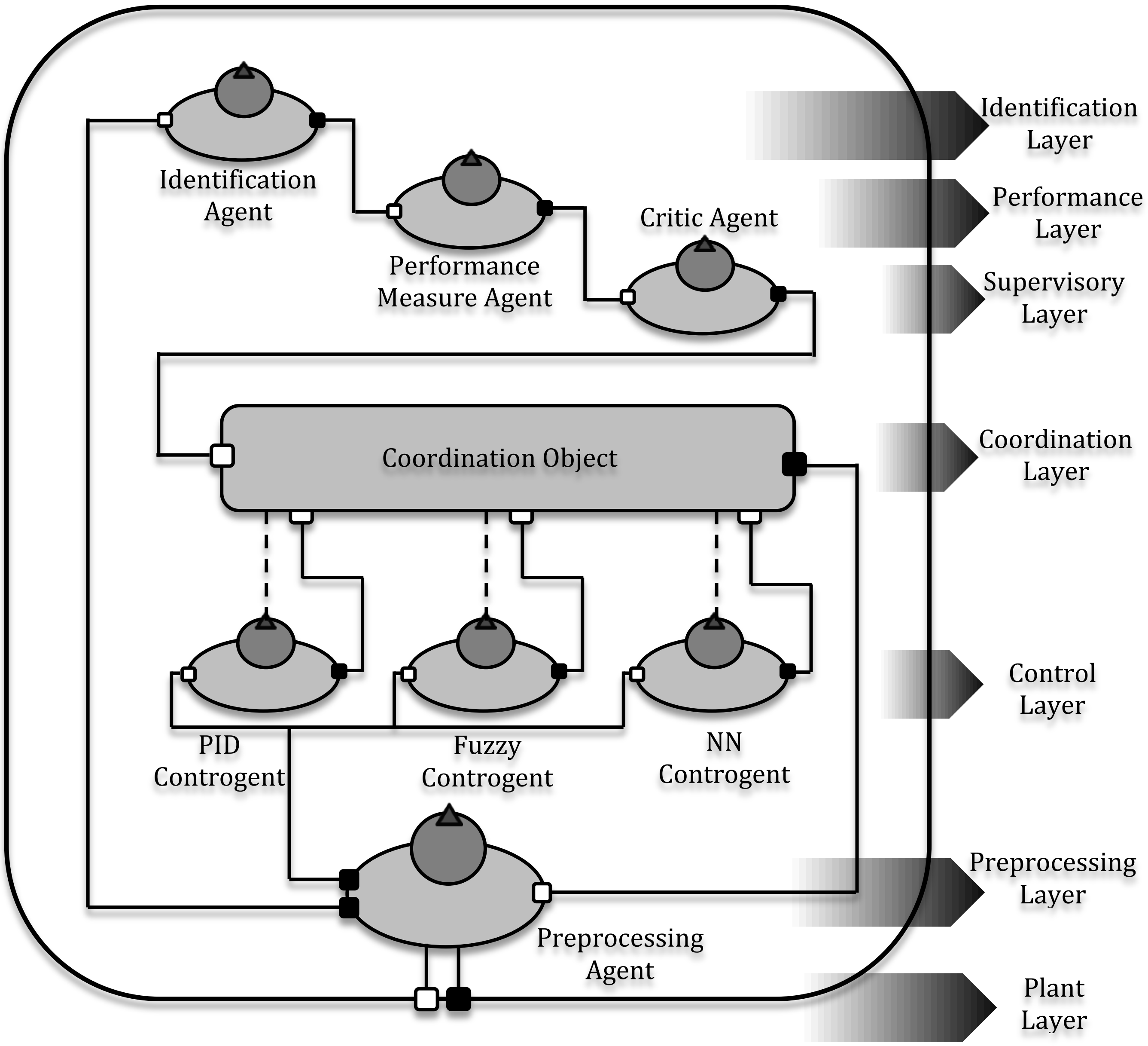


Figure 8.4 Intelligent MAC_SC architecture

8.4 Real time system

The multi-agent controller performance is analyzed using water tank plant. It is designed and developed by Quanser Inc, Canada for control algorithm analysis [6]. The typical water tank plant is depicted in figure 8.4. The tank specialty module is a bench "Two-Tank" plant consisting of a pump with a water basin and two tanks of uniform cross sections. The two tanks, mounted on the front plate are configured such that flow from the first tank can flow into the section tank. Flow from the second tank flows into the main water reservoir. The single system is configured for three types of experiment study, Single input single output, State coupled-SISO, and both. In this analysis, the first configuration Single Input Single Ouput (SISO) is used and mathematical modeling is given herewith. The details about the setups is discussed in Appendix: B.

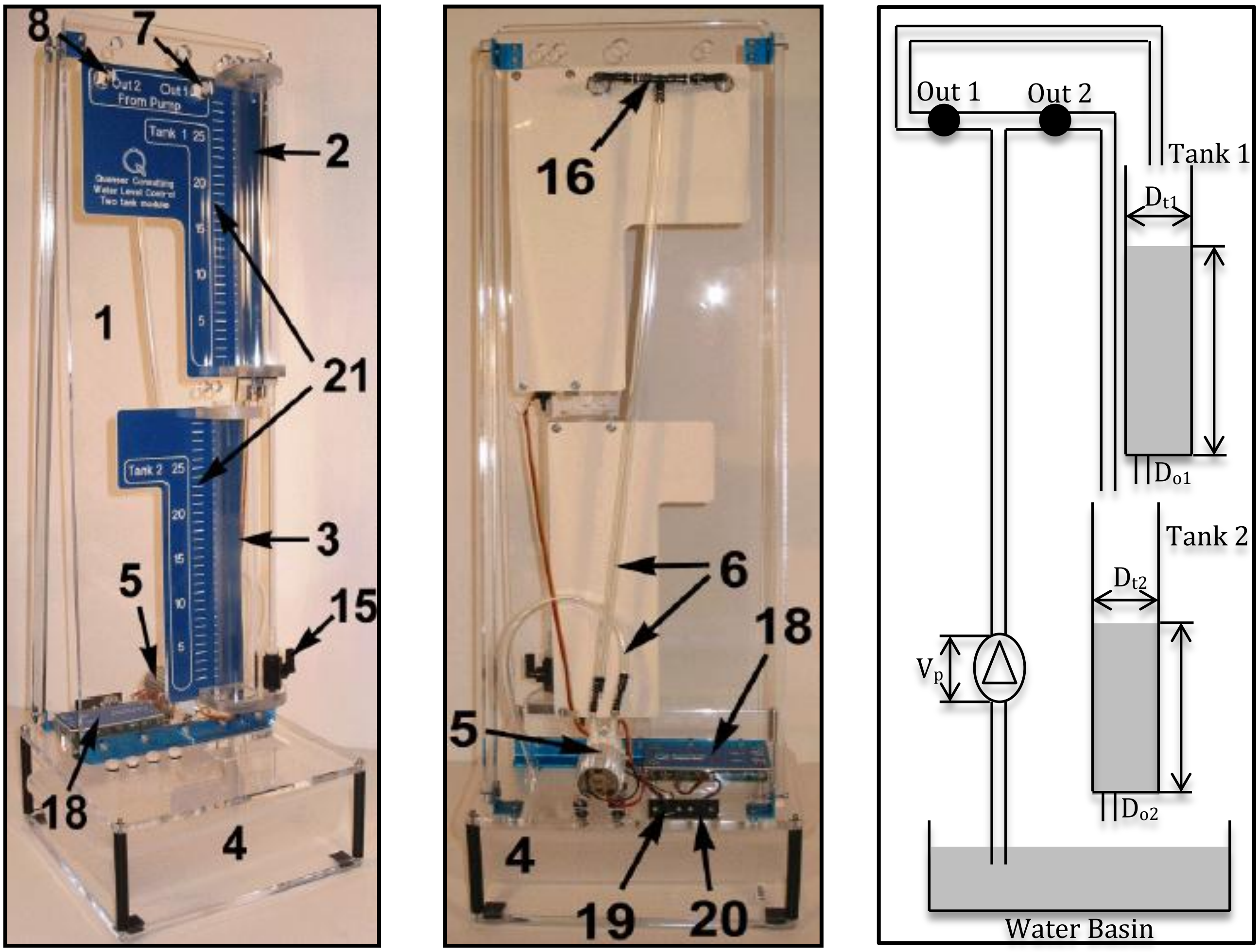


Figure 8.5 Coupled tank plant (Quanser Inc., Canada)

The outflow rate from tank 1, Fo_1, can be expressed by:

$$F_{o1} = A_{o1}\sqrt{2}\sqrt{gL_1} \qquad 8.2$$

The inlet volumetric flow rate of tank-1 is:

$$F_{i1} = K_p V_p \qquad 8.3$$

Using mass balance principle for tank 1, the first order differential equation is:

$$A_{t1}\frac{d}{dt}L_1 = F_{i1} - F_{o1} \qquad 8.4$$

The description of system parameters are given in table: 8.4. Substituting equation 8.2 and 8.3 into 8.4 and rearranging results in the following equation of motion for the tank-1 system:

$$\frac{d}{dt}L_1 = \frac{K_p V_p - A_{o1}\sqrt{2}\sqrt{gL_1}}{A_{t1}} \qquad 8.5$$

With pump constraints:

$$V_p = \begin{cases} 0 & \upsilon(\mathrm{t}) < 0 \\ \upsilon(\mathrm{t}) & 0 \le \upsilon(\mathrm{t}) \le \upsilon_{\max} \\ \upsilon_{\max} & \upsilon(\mathrm{t}) > \upsilon_{\max} \end{cases}$$

Due to the square root function applied to L_1, the first-order differential equation expressed by equation 8.4 is non-linear.

Table 8.4 Description of parameters

Symbol	Description	Unit
F_{i1}	Volumetric inlet flow rate to tank-1	cm^3/s
F_{o1}	Volumetric outlet flow rate to tank-1	cm^3/s
A_{t1}	Tank-1 inside cross-section area	cm^2
A_{o1}	Tank-1 outside cross-section area	cm^2
L_1	Tank-1 water level	cm
K_p	Pump volumetric flow rate	$cm^3/s/V$
V_p	Applied pump voltage	V
D_{o1}	Tank-1 inside diameter	cm
Dt_1	Tank-1 outlet diameter	cm

8.5 Results analysis

Intelligent MAC_SC is applied on the water tank control system. The various operating regimes are used to evaluate the response of controller for various initial conditions and changing in parameters. If the system is having continuous changing in operating condition and its plant operating parameters so the simple classical control is not suitable with good efficiency. As per discussed the general multi-agent controller architecture (Intelligent MAC_SC) in previous sections is suitable. The controller response with different operating regime is given in figure 8.6. The control effort of intelligent MAC_SC is given in figure 8.7 to achieve the targeted goal. The control efforts of various controgents (PID, Fuzzy and NN) are given in figure 8.9.

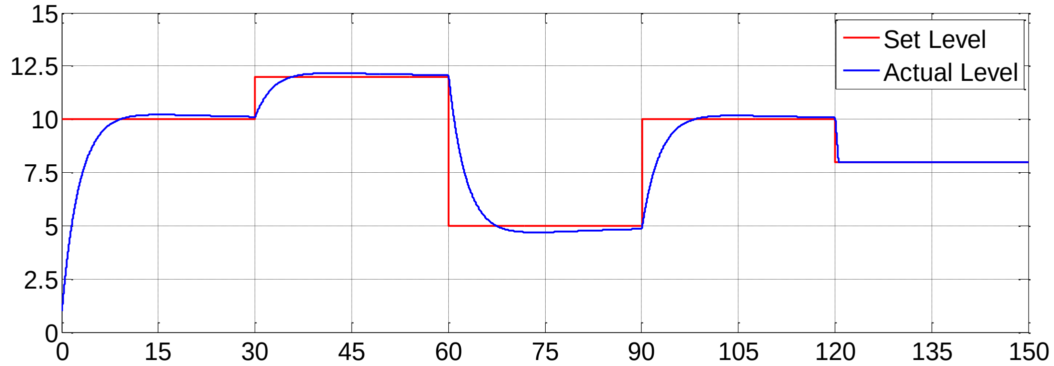


Figure 8.6 Intelligent MAC_SC controller response with desired output

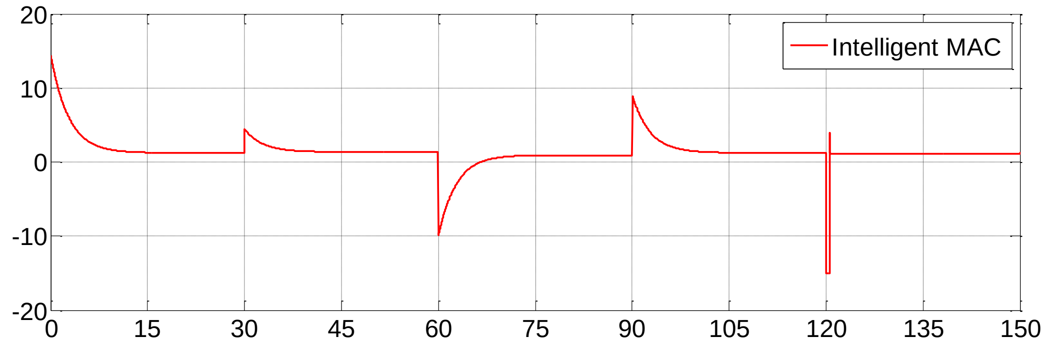


Figure 8.7 Control effort of MAC

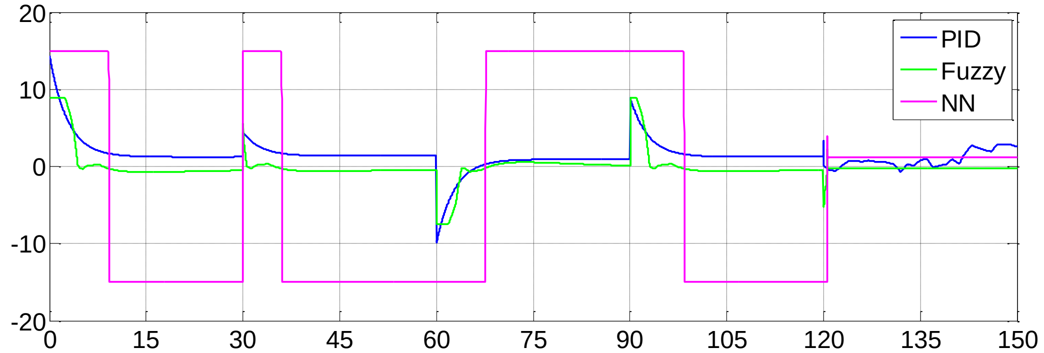


Figure 8.8 Control efforts of PID, Fuzzy and NN controgents

The controller is activated best contrôgent base on response of plant after some prefixed interval for respective regime. From figure 8.7 and 8.8, the PID contrôgent selected for first four operating regime and the NN contrôgent is selected in last operating regime. The response of conventional PID controller for last operating regime is given in figure 8.9 and figure 8.10. From graphical demonstration, the conventional PID controller response is not suitable for last operating regime. The point is that it is continuously oscillatory in the system output. NN contrôgent is activated for this regime by MAC controller for desired output response. Numerical analysis of MAC controller with various control performance parameters are discussed in next session.

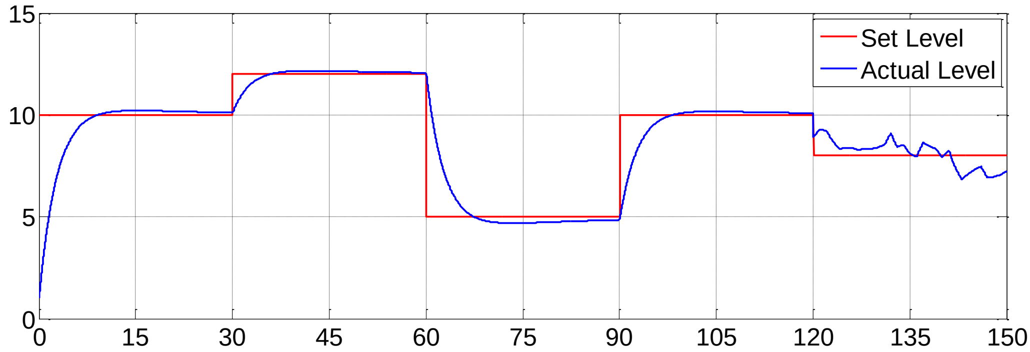


Figure 8.9 PID controller response with desired output

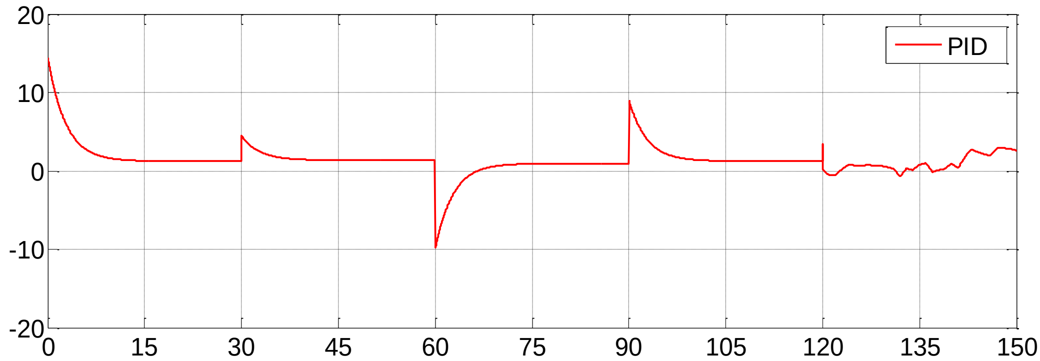


Figure 8.10 Control effort of PID

Performance measure agent is identified the various control performance parameters for various operating regime as given in figure 8.11. The measured parameters are not in proper range, they are converted in suitable range with percentage as per discussed in chapter 6 and given in table 8.5.

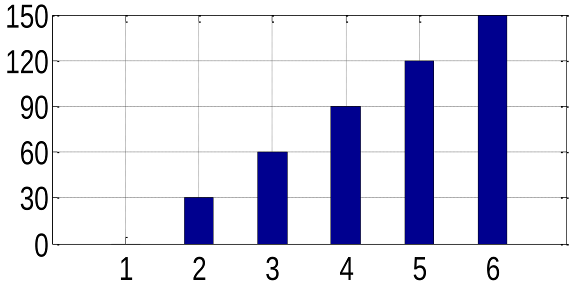
The critic agent activates optimized and efficient contrôgent for particular environment using different close loop parameters of plant. The critic agent output is in range of 0 to 30 as per discussed in chapter 6. So, the output is in 0 to 10, critic agent generate the 0 for PID contrôgent activation. The output is in range of 11 to 20 then it generates 1 for fuzzy contrôgent.

And in last case, it generates 2 for the NN controgent. The coordination agent actives the respective controgents through output of critic agent. So the output of controgent for various operating regimes is given figure 8.9(b). In all operating regimes, the PID controlgent is selected excluding forth regime. In forth regime, NN controgent is activated.

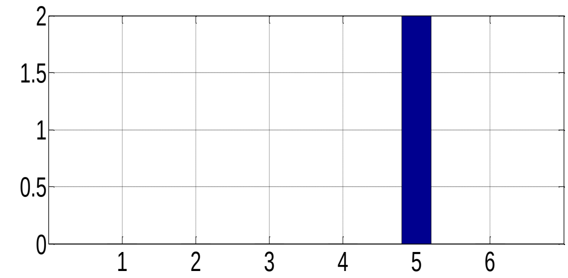
Table 8.5 Performance parameter measured and processed value

Parameters		1	2	3	4	5	6
Overshoot	Processing value	0.000	12.34	0.4829	2.041	11.53	0.53
	Measured value	0.000	61.71	2.4143	10.20	57.67	2.65
Steady state error	Processing value	-100.0	0.674	0.447	-88.98	-42.94	0.604
	Measured value	-5.220	0.033	0.022	-4.449	-2.147	0.030
Variance of output	Processing value	62.81	0.139	0.018	30.04	25.92	0.017
	Measured value	25.12	0.055	0.071	12.0164	10.37	0.007
Rise time	Processing value	4.230	11.74	17.98	15.31	13.25	18.36
	Measured value	6.480	17.71	27.07	23.07	19.98	27.64
Settling time	Processing value	100.0	100.0	20.20	100.0	100.0	22.35
	Measured value	999.0	999.0	50.00	999.0	999.0	53.00
Reference change	Processing value	0.000	20.00	-50.00	0.000	-20.00	-20.00
	Measured value	0.000	2.000	-5.000	0.000	-2.000	-2.000

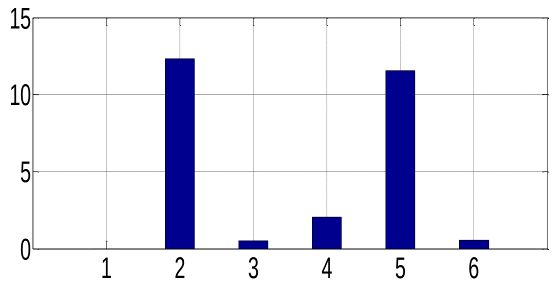
As evident from table 8.6 which summarize the operation of intelligent MAC_SC performed with various operating and performance parameter to achieve the targeted goals given above. The design of any controller is depend on initial parameter of plant or system. These parameters are not remaining fix for all operating regimes of plants. In real time system application, the controller performance should be compensated these disturbances also. The numerical analysis of controller with various initial conditions is given in table 8.7 to see the performance of intelligent MAC_SC for initial condition change. Therefore, it can be concluded with graphical and numerical analysis, the Intelligent MAC_SC controller is preserved the smooth and efficient response compare then conventional classical control. This architecture is flexible so that the more control algorithm can be included as per plant performance and its operating regimes. After, partly modification in Intelligent MAC_SC architecture, it can be used for wide range of nonlinear systems and continuously changing operating parameters of plants.



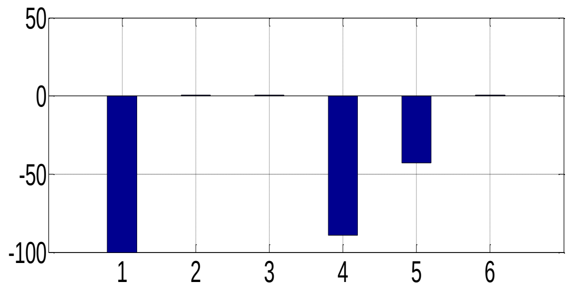
8.11(a) Sampling time



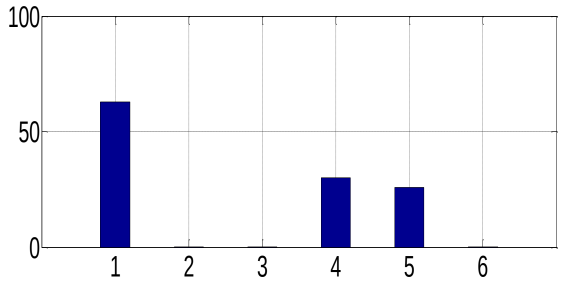
8.11(b) Activated controgent for various operating regime



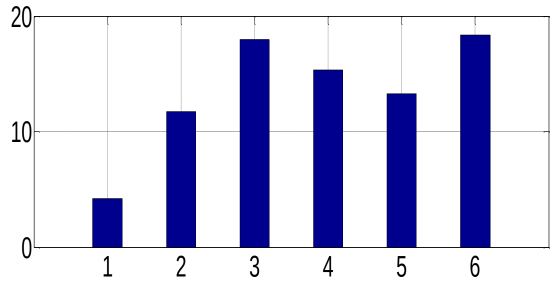
8.11(c) Overshoot parameter for various operating regime



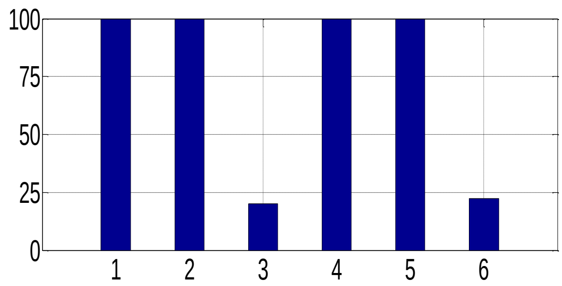
8.11(d) Steady state error for various operating regime



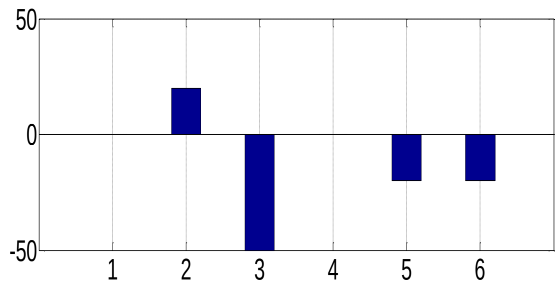
8.11(e) Variance of output parameter for various operating regime



8.11(f) Rise time parameter for various operating regime



8.11(g) Settling time parameter for various operating regime



8.11(h) Change in reference parameter

Figure 8.11 Various control performance assessment parameters

Table 8.6 Activated controgent for various performance parameter with all constant parameters

Overshoot	Steady state error	Variance of output	Rise time	Settling time	Change of reference	Activated controgent	Remark
27.4	-48.7	34.3	43.8	53.8	26.1	NN	Rule 1 and 9 are fired due to negative steady state error and high variance.
93.3	28.9	10.8	64.8	49.1	12.2	Fuzzy	Rule 3 is fired due to high overshoot.
53.1	-70.1	5.67	75.8	62.4	-24.7	PID	Rule 7 and 8 are fired due to low variance, rise time and settling time.
14.3	3.48	77.0	16.1	30.0	64.3	NN	Rule 9 and 11 are fired due to high variance and increase in reference change.
63.4	16.5	31.4	0.51	61.3	0.00	PID	Rule 5 and 10 are fired due to fast settling time and no reference change.
38.7	-33.0	18	2.58	68.6	68.6	NN	Rule 6 and 12 are fired due to slow rise time and increase reference.
30.9	-0.05	5.65	27.4	45.7	1.74	PID	Rule 2 and 8 are fired due to medium steady state error and low variance.
38.6	99.1	17.8	16.1	30.0	0.00	Fuzzy	Rule 3 is fired due to high steady state error.
93.3	-4.12	10.8	3.61	16.0	8.25	NN	Rule 4 and 9 are fired due to fast rise time and high variance.

Table 8.7 Performance of controller with various initial conditions and different operating points

Operating parameter	Overshoot	Steady state error	Variance of output	Rise time	Settling time	Change of reference	Activated contrcareent
Initial condition (A=16, a=2, b=5)							
10	0.00	-12.37	0.07	33.10	43.22	0.00	PID
12	0.35	-3.62	0.02	15.27	7.72	20.0	PID
5	0.00	-7.54	0.04	22.64	23.61	-50.0	PID
10	0.00	-84.17	38.66	5.75	100.0	0.00	NN
8	0.00	-91.16	41.97	6.97	100.0	-20.0	NN
Initial condition (A=18, a=2, b=5)							
10	0.75	1.70	0.01	17.41	24.74	0.00	PID
12	0.00	-9.69	0.05	27.50	31.52	20.0	PID
5	5.28	-87.05	67.45	4.30	100.0	-50.0	PID
10	0.06	-23.42	0.17	25.16	18.91	0.00	NN
8	0.70	1.08	0.02	16.57	19.54	-20.0	NN
Initial condition (A=18, a=2, b=5)							
10	0.23	-1.14	0.03	21.17	15.10	0.00	PID
12	0.00	-6.39	0.04	22.23	18.68	20.0	PID
5	0.00	100.0	43.39	4.02	100.0	-50.0	PID
10	0.00	83.00	48.97	5.48	100.0	0.00	NN
8	54.26	35.76	8.55	0.04	100.0	-20.0	Fuzzy
Initial condition (A=18, a=1.8, b=4)							
10	0.00	-70.20	33.71	8.35	100.0	0.00	NN
12	69.94	30.78	2.98	1.16	100.0	20.0	Fuzzy
5	0.00	-9.26	0.06	26.84	27.92	-50.0	PID
10	0.00	-4.60	0.03	19.44	15.74	0.00	PID
8	0.00	-4.57	0.03	19.19	15.02	-20.0	PID

8.6 Concluding remarks

In this chapter, the generalized multi-agent control algorithm Intelligent MAC_SC is developed for water tank control system. Its control problem structuring was discussed in detail. This is also covered with the real time control module prototype coupled tank system for analysis of MAC controller. All agents in MAC architecture were covered. The graphical user interface (GUI) was developed for analysis of this controller. Performance analysis of controller with reference of various performance parameters as overshoot, steady state error, variance of output, rise time, settling time and change of reference was discussed in detail. Intelligent MAC_SC architecture is flexible so that the more controgents can be included as per plant performance and its operating regime. If the plant is nonlinear with noise and continuously changes in parameters then this MAC architecture is more efficient compare than classical control. The final conclusion and future scope of this work is given in next chapter.

Chapter 9
Conclusion and suggestion for extension

The chapter presents discussion on results and summarizing the contribution of research work carried out. Directions for future research work are suggested.

9.1 Research contributions

The contributions of research work are listed below:

- Definition and interpretation of complex control problems which provides structure of underling the mechanisms based the divide-and-conquer approach to solve control problems.
- An agent-based design method for constructing modular multi-agent control architecture having as open character, such that parts can be added, modified or removed without disordering the operation of the remaining parts of the controller.
- Design and implementation of MAC architecture framework.
- Application of MAC to various benchmark control problems and comparative analysis with traditional methods.

These contributions are discussed in more detail hereafter.

9.1.1 Formulation of control problem

In general, it is concluded that control problems are not solved in a single step, because most control problems are complex. Complexity is a property of control problems and can be defined in terms of partial control problems and their interdependencies. Solving large-scale and real-world control problems should start with modeling the structure of the problem. Structuring is necessary in order to formulate a well-defined overall control problem. It should result into a structure diagram, which visualizes the partial control problems and their interdependencies. Five different types of dependencies, also called coupling relationships, were identified: independent, indirect dependent, sub ordinal, hard conditional and soft conditional dependency.

Once a control problem is structured, the elementary control problems can be solved, for instance by using available control theory. The integration problem of a set of several controllers is given less attention within the field of control theory than the problem of solving elementary control problems. Traditionally, control engineers use a supervisory architecture to integrate a set of locally operating controllers.

9.1.2 Multi-Agent Control (MAC) architecture

An agent and the concept of a locally operating controller are not much different; they can be integrated as controgent. The field of multi-agent system provided concepts to organize multi-agent control architecture while the field of control engineering provided the insight to design the architecture of controgent. To handle dependencies between controgents, different coordination mechanisms were discussed. With respect to the set of coupling relationships and coordination mechanisms, it can be concluded that they are suitable to handle general control problems. Due to required faster response in real time application, the optimized agents were designed using soft computational paradigms as fuzzy logic, neural network and genetic algorithm.

9.1.3 Design and Implementation of MAC

Many of the properties of the conceptual solution are preserved by the implementation. Conceptually defined individual controgents are also implemented as individual components. Also, like controgents in a conceptual solution, controgent components can be added, modified or removed without re-programming the remaining system. This greatly speeds up the development of controllers for multiple operating regime systems or plants. Furthermore, compositions of individual controllers can be tested, without having a complete realization of the overall control system.

It has been demonstrated that the design method and the developed high-level compiler work in practice. There multi-agent control architecture as Basic MAC_SC, Advance MAC_SC and Intelligent MAC_SC are designed for various applications. Three control problems –two simulations and one real-time application- were successfully solved by applying the agent based control architecture. The Matlab and various tool-boxes were helped to realize various MAC architecture.

9.1.4 Overall conclusion

The agent-based design method presented in this thesis helps the designer to solve complex control problems, by offering concepts to structure the problem and to organize the solution. The design method encourages to develop local solutions and to reason about their dependencies. It offers different coordination mechanisms to deal with these dependencies. The summary of work is presented in the form of multi-agent controller having a graphical user interface shown in figure 9.1 to 9.4. The synopsis and report are embedded in the software for on-line help. Figure 9.2 depicts the options menu of various MAC architectures. The introduction to the benchmark problems and application of MAC are depicted in figure 9.3. The intelligent MAC_SC home page is given in figure 9.4.

Figure 9.1 Main screen

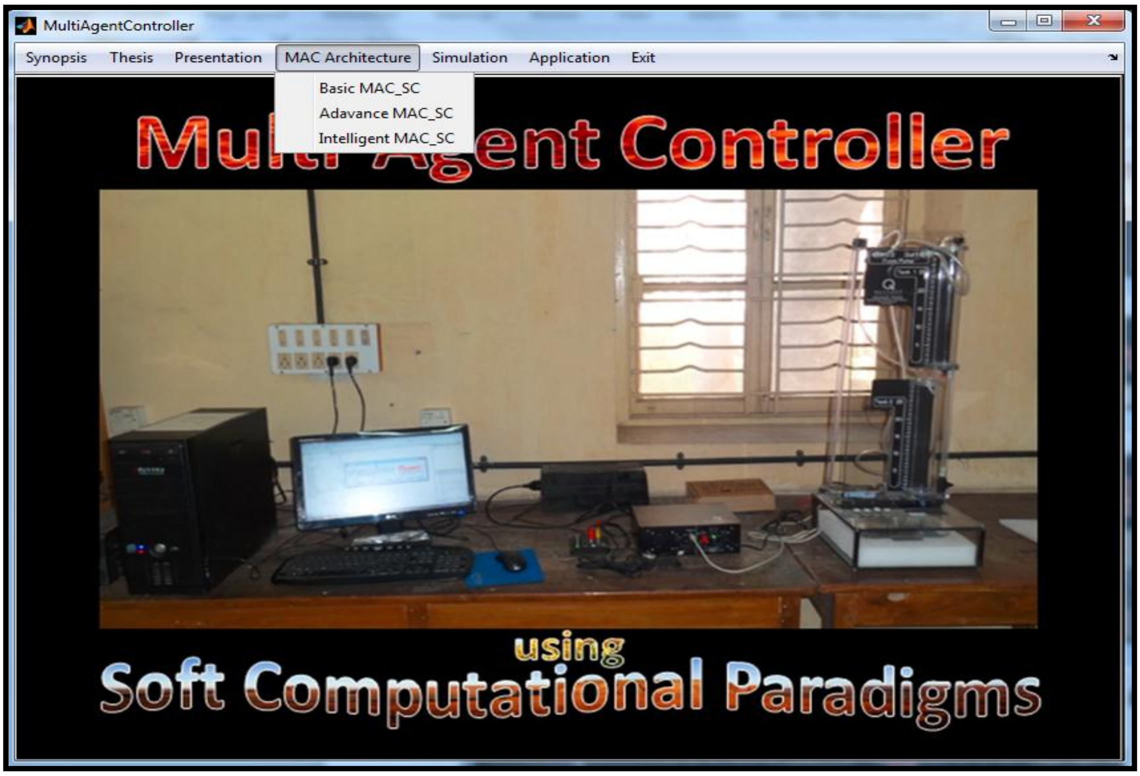


Figure 9.2 Options with various MAC architectures menu

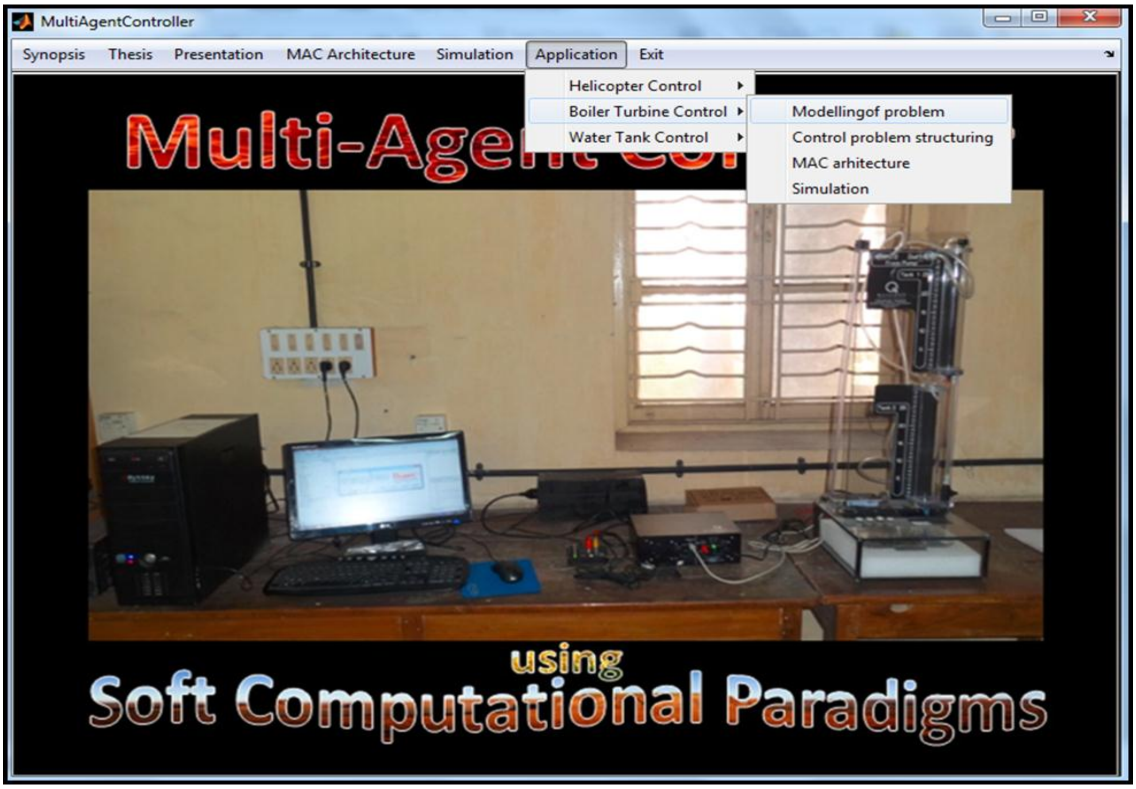


Figure 9.3 List of benchmark problem for MAC applications

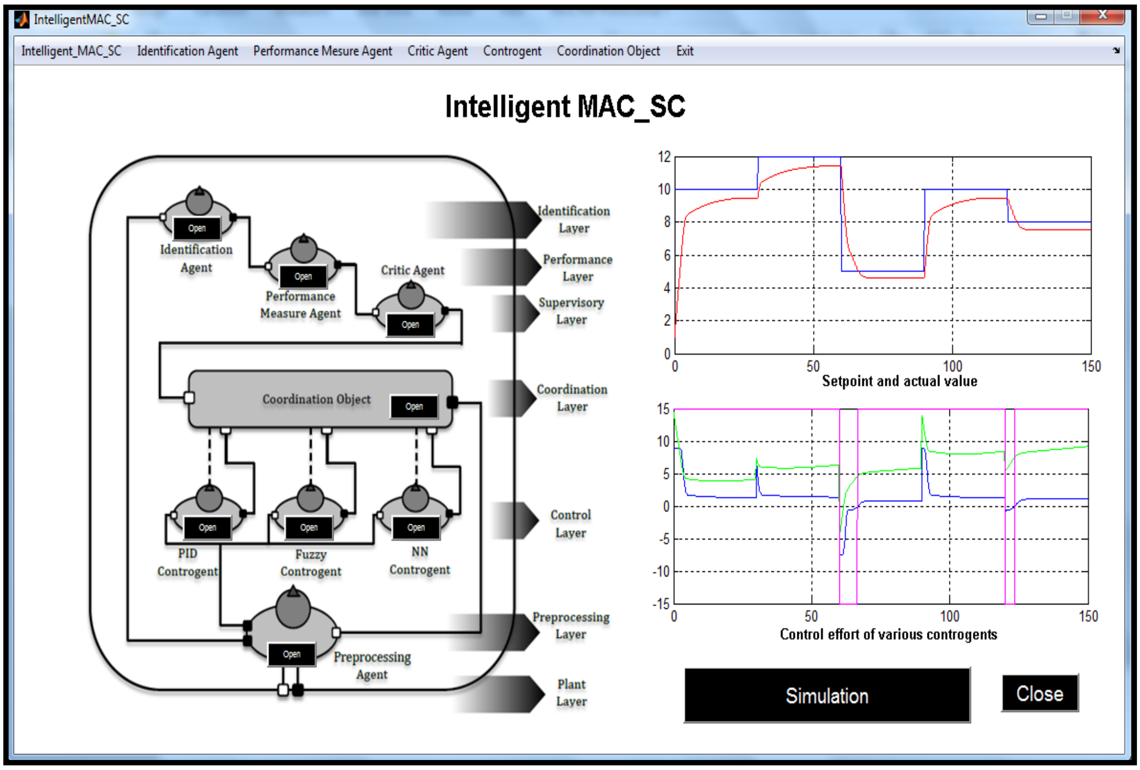


Figure 9.4 Intelligent MAC_SC

9.2 Direction for future extension

Now that it has been established that the proposed multi-agent controller can be functionally useful, further work is required to understand the extent of this usefulness. Some proposals in this regard are presented next.

- The architecture of multi-agent controllers is designed on the basis of generalized minimum variance control with PID. The PID control provides a possibility of adaptive changing the conventional PID control into PI, PD or PID according to the plant behavior and targeted performance. That means additional controgent would be including in the multi-agent control architecture which are most commonly applied in the control industry.
- In complex system or plant, nonlinearities large range of operating conditions. Also the systems are restricted due to mechanical, aerodynamic, thermal and flow limitations. The online prediction is delayed due to number nonlinearities and constraints. The verification of the identified model is also missing in this design. So, reducing these limitations, the identification agent should be designed using hybrid soft computational paradigms.
- During the research, multi-agent control architecture was realized using matlab code. Generalized MAC tool should be designed using programming software. It will be included with library for all agents and coordination objects with various intelligent tool supports. It is suggested to develop this tool in graphical based software environment for reducing the complexity and easily development.
- Analysis and debugging tool mean monitoring tool that plots operating status of controgent as well as value of their attributes. The application studied showed that this information helps in graphical and numerical analysis of overall behavior of the controller system and in local operating regime the controgent that give rise to unwanted behavior. Analysis and debugging tool for MAC architecture should be developed.

Chapter 10

Bibliography

Thesis ends with bibliography which includes the list of reference use in each chapter and list of publications and presentations done based on this work.

Chapter 1

[1] B. D. Anderson, A. Dehghani, "Challenges of adaptive control- Past, permanent and future", *Annual reviews in control*, Vol.32, Pages 123-135, 2008.

[2] B. Yao, "High performance adaptive robust control of nonlinear systems: a general framework and new schemes", *Proceeding of the IEEE Conference on Decision and Control*, Pages 2489-2494, 1997.

[3] M. Nasir Uddin, "Development implementation and performance analysis of intelligent controllers for IPM synchronous motor drive systems", *Journal of control and intelligent systems,* Vol.35, Issue.1, January 2007.

[4] H. A. Simon, "The Science of the Artificial", *MIT Press Science/ Philosphy,* 1996.

[5] A. Breemen, "Agent-based Multi-Controller Systems", *University of Twente*. Enschede, Netherlands: Twente University Press, 2001.

[6] V. Uraikul, C.W. Chan, P. Tontiwachwuthikul, "Artificial intelligence for monitoring and supervisory control of process system", *Engineering Applications of artificial intelligence*, Vol. 20, Pages 115-131, 2007.

[7] H. L. Chiang, L. E. Russell, D. R. Braatz, "Fault Detection and Diagnosis in Industrial Systems", *Springer*, London, Great Britain. 2001.

[8] Z. J. Viharos, L. Monostori, "Optimisation of process chains and production plant using hybrid, AI and simulation based general process models", *The Fourteen International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems*, Budapest, Hungary, June, 4–7, 2001.

[9] H. S. Kim, Y. K. Shin, "Expanded adaptive fuzzy sliding mode controller using expert knowledge and fuzzy basis function expansion for UFV depth control", *Ocean engineering*, Vol. 34, Pages 1080-1088, 2007.

[10] C. L. Jain, P. R. Johnson, Y. Takefuji, A. L. Zadeh, "Knowledge based intelligent techniques in industry", *CRC Press*, New York, 1999.

[11] H. Deng, Z. Xu, H. X. Li, "A novel neural internal model control for multi-input multi-output nonlinear discrete-time processes", *Journal of process control*, Vol. 19, Pages 1392-1400, 2009.

[12] P. A. Alexandridis, I. C. Siettos, K. H. Sarimveis, G. A. Boudouvis, V. G. Bafas, "Modelling of Nonlinear Process dynamics using kohonen's neural networks", fuzzy systems and chebyshev series, *Computers & Chemical Engineering,* Vol. 26, Pages 479–486, 2002.

[13] Z. Li, D. Liang, "Human simulated intelligent controller with fuzzy online self-tuning of parameters and its application to a cart-double pendulum", *Journal of computers*, Vol.3, Issue 9, 2008

[14] V. S. Lazarou, S. K. Gardikiotis, N. Malevris, "Agent systems in software engineering", *Tools in artificial intelligence*, Intech, 2000.

[15] B. Chen, H. H. Cheng, "A review of the applications of agent technology in traffic and transportation systems", *IEEE transactions on intelligent transportation systems*, Vol. 11, Issue 2, June 2010.

[16] S.K.Shah, R.B.Patel, S.R.Panchal, "Design and implementation of multi-agent controller for SISO system", *International conference on system dynamics and control ICSDC-2010*, ISBN: 978-93-80578-58-3, Manipal University, Karnataka, India, August 19th-20th 2010, Publisher: I. K. Publication, New-Delhi.

[17] C .K. Fan, T. N. Wong, "An agent-based intelligent FMS control system", *proceeding of the fifth asia pacific industrial engineering and management system conference*, 2004.

[18] Q. M. Doos, Z. Al-Daoud, S. M. Al-Thraa, "Agent based fuzzy ARTMAP neural network for classifying the power plant performance", *Jordan journal of mechanical and industrial engineering*, Vol. 2, Issue 3, Pages 123-129, 2008.

[19] M. Wooldridge, "Intelligent agents", In G. Weiss, editor, *Multiagent Systems*, The MIT Press, Cambridge, Massachusetts, pages 27–77, 1999.

[20] F. Daneshfar, H. Bevrani, "Multi-agent systems in control engineering: a survey", *Journal of control science and engineering*, DOI:10.1155/2009/531080, 2009.

[21] R.B.Patel, "Multi-agent system: A recent trends in control engineering", *Control, Microprocessor, Electronics & Communication CMEC-2011*, Institution of electrical and electronics engineers, Vadodara, February 20th 2011.

[22] S. D. J. McArthur, S. M. Strachan, and G. Jahn, "The design of a multi-agent transformer condition monitoring system", *IEEE Transactions on Power Systems*, vol. 19(4), Pages 1845–1852, 2004.

[23] J. Zhou, G. Chen, H. Zhang, W. Yan, and Q. Chen, "Multi-agent based distributed monitoring and control of hazard installations", in *Proceedings of the 2nd IEEE Conference on Industrial Electronics and Applications (ICIEA '07)*, Pages 2892– 2895, 2007.

[24] J. Galdun, L. Takac, J. Ligus, J. M. Thirie, and J. Sarnovsky, "Distributed control systems reliability: consideration of multi-agent behaviour", *Proceedings of the 6th International Symposium on Applied Machine Intelligence and Informatics (SAMI '08)*, Pages 157–162, 2008.

[25] K. Fregene, D. C. Kennedy, and D. W. L. Wang, "Toward a systems- and control-oriented agent framework", *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35(5), Pages 999–1012, 2005.

[26] A. L. Dimeas and N. D. Hatziargyriou, "Operation of a multiagent system for microgrid control," *IEEE Transactions on Power Systems*, vol. 20, Issue 3, Pages 1447–1455, 2005.

[27] D. P. Buse, P. Sun, Q. H. Wu, and J. Fitch, "Agent-based substation automation", *IEEE Power and Energy Magazine*, vol. 1, Issue 2, Pages 50–55, 2000.

[28] K.M. Saridakis, A.J. Dentsoras, "Soft computing in engineering design–A review", *Advance engineering informatics*, Vol. 22, Pages 202-221, 2008.

[29] B.Kosko, "Neural Networks and Fuzzy Systems", *Englewood Cliff,* NJ, Prentice-Hall, 1992.

[30] D.E. Goldberg, "Genetic Algorithms in Search: Optimization and Learning", *Addison-Wesley Publishing Co.,* Reading, 1989.

[31]. P. Bieganski, A. Byrski, M. K. Dorohinicki, "Multi-agent platform for distributed soft computing". *Journal of Artificial Intelligence*, Vol. 9(28), Pages 63-70, ISSN: 1137-3601, 2005.

[32] R.B.Patel, S.K.Shah, "GA based competitive multi-agent controller for non-linear system", *3rd international conference on mechanical and electrical technology ICMET-2011,* ISBN: 978-0-7918-5981-0, Dalian, China, August 26th-27th 2011, Publisher: ASME press, New York, USA. DOI: 10.1115/1.859810.paper186

[33] R.B.Patel, S.K.Shah, "Multi-agent based controller for boiler turbine plant", *International journal of control & intelligent system*, Accepted for publication after review (Publisher: ACTA press, Canada)

(Online: http://asmedl.org/ebooks/asme/asme_press/859810/859810_paper185)

[34] R.B.Patel, S.K.Shah, "Multi-agent controller using soft computational paradigms", *Indian journal on applied research*, ISSN: 2249-555X, Vol.3, Issue 2, Pages 138-142, 2013.

(Online: http://www.theglobaljournals.com/ijar/file.php?val=MTAwOQ==)

[35] R.B.Patel, S.K.Shah, "Controller performance assessment (CPA) of intelligent control for nonlinear system". *Journal of control & instrumentation*, ISSN: 2229-6972, Vol.2, Issue 2-3, Pages 17-23, 2011.

[36] R.B.Patel, S.K.Shah, "Identification of non-linear system using computational paradigms". *International conference on automatic control & artificial intelligence ACAI-2012*, ISBN: 978-1-

84919-5379, Pages 1156-1159, Publisher: IET, U.K. DOI: 10.1049/cp.2012.1183, Xiamen, China, March 24th-26th 2012.

Chapter 2

[1] Ragazzini, Franklin, "Sampled-Data Control Systems", *McGraw-Hill*, New York, 1958.
[2] H. Kwakernaak, R. Sivan, "Linear Optimum Systems Control", *Wiley Interscience*, New York, 1972.
[3] A. P. Sage, C.C. White, "Optimum Systems Control", *Prentice-Hall*, Englewood Cliffs, 1977.
[4] K.J. Åström, B. Wittenmark, "Adaptive Filtering, Prediction and Control", *Addison Wesley*, 1988.
[5] P.E. Wellstead, M.B. Zarrop, "Self-tuning Systems, Control and Signal Processing", *John Wiley & Sons*, Chichester, England, 1991.
[6] A. Isidori, "Nonlinear Control Systems: An Introduction", 2nd Edition, *Springer*, 1989.
[7] J.J Slotine, W. Li, "Applied Nonlinear Control", *Prentice-Hall*, 1991.
[8] K. Zhou, J.C. Doyle, K. Glover, "Robust and Optimal Control", *Prentice-Hall*, Englewood Cliffs, 1996.
[9] A.S. Foss, "Critique of chemical process control theory", *AIChE Journal*, vol.19, no.2, Pages 209-2014, 1973
[10] K. J. Astrom, T.H. Hagglund. "New Tuning Methods for PID Controllers," *Proceedings from the 3rd European Control Conference*, 1995.
[11] K.W. Ho, A. Datta, S.P. Bhattacharya, "Generalizations of the Hermite-Biehler theorem," *Linear Algebra and its Applications*, Vol. 302-303, 1999, pp. 135-153.
[12] K.W. Ho., A. Datta, S.P. Bhattacharya, "PID stabilization of LTI plants with time-delay," *Proceeding from the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, 2003.
[13] S. P. Bhattacharyya, L.H. Keel, "PID controller synthesis free of analytical methods," *Proceeding of IFAC 16th Triennial World Congress*, Prague, Czech Republic, 2005.
[14] S. Sujoldzic, J.M. Watkins, "Stabilization of an arbitrary order transfer function with time delay using PI and PD controllers," *Proceeding of American Control Conference*, Pages 2427-2432, 2006.
[15] S. Sujoldzic, J.M. Watkins, "Stabilization of an arbitrary order transfer function with time delay using PID controller," *Proceeding of IEEE Conference on Decision and Control,* Vol. 45, 2005.
[16] T. Emami, J.M. Watkins, "Weighted sensitivity design of PID controllers for arbitrary-order transfer functions with time-delay," *Proceeding of the IASTED International Conference on Intelligent Systems and Control*, Pages 20-25, 2008.

[17] T. Emami, J.M. Watkins, "Robust stability design of PID controllers for arbitrary-order transfer functions with uncertain time delay," *Southeastern Symposium on System Theory University of Tennessee Space Institute*, Pages 184-189, 2009.

[18] T. Emami, J.M. Watkins, "Complementary sensitivity design of PID controllers for arbitrary-order transfer functions with time delay," *Proceeding of ASME Dynamic Systems and Control Conference*, 2008.

[19] T. Emami, J.M. Watkins, "Robust performance characterization of PID controllers in the frequency domain," *WSEAS Transactions Journal of Systems and Control*, Vol. 4, No. 5, Pages 232-242, 2009.

[20] S.W. Sung, "Limitations and Counter measures of PID Controllers," *Department of Chemical Engineering, Pohang University of Science and Technology*, Pohang, Korea, 1996.

[21] M. A. Henson, D. E. Seborg, "Adaptive nonlinear control of a pH neutralization process", *IEEE Transactions on Control Systems Technology*, Vol.2, 1992.

[22] T. Clarke-Pringle, J. F. MacGregor, "Nonlinear adaptive temperature control of multi-product", *semi-batch polymerization reactors*, Computers in Chemical Engineering, Vol. 21, 1997.

[23] B. D. Anderson, "Adaptive systems, lack of persistency of excitation and bursting phenomena", *Automatica*, Vol. 21, No. 3, Pages 247–258, 1985.

[24] K. J. Astrom, "Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics'', *IEEE Transactions on Automatic Control*, Vol.30, No. 9, Pages 889–889, 1985.

[25] B. Guo, A. Jiang, X. Hua, A. Jutan, "Nonlinear adaptive control for multivariable chemical processes*", Chemical engineering science*, Vol. 56, Pages 6781-9791, 2001.

[26] A. Dehghani, "Robust adaptive control schemes", *Ph.D. thesis*, The Australian National University, Australia, 2007.

[27] C. Manuelli, S. G. Cheong, E. Mosca, M. G. Safonov, "Stability of unfalsified adaptive control with non-scli controllers and related performance under different prior knowledge", *Proceedings of the European control conference,* Pages 702-708, 2007.

[28] K. S. Narendra, A. M. Annaswamy, "Stable adaptive systems", *Englewood cliffs*, NJ, Prentice-hall, 1989.

[29] K. J. Astrom, B. Wittenmark, "Adaptive control", *Addison-Wesley publishing company*, 1995.

[30] B. D. Anderson, "Challenges of adaptive control- past, permanent and future", *Annual reviews in control*, Vol.32, Pages 123-135, 2008.

[31] G. N. Saridis, "Foundations of the Theory of Intelligent Controls," *Proceeding of the IEEE Workshop on Intelligent Control*, Eds. G. Saridis, A. Meystel, Troy, New York, Pages 23-28, 1985.

[32] S. Chiu, "Developing commercial applications of intelligent control*", IEEE Control System Magazine*, Vol. 17, No. 2, Pages 94-100, 1997.

[33] J. Jang, "ANFIS: Adaptive network-based fuzzy inference systems", *IEEE Trans System Man Cabernet*, Vol.23, Pages 665–685, 1993.

[34] C. L. Karr, "Control of a phosphate processing plant via a synergistic architecture", *Engineering application in artificial intelligence*, Vol.6, No.1, Pages 21-30, 2003.

[35] AJF Van Rooij, "Neural network training using genetic algorithms", *World Scientific*, River Edge, NJ, 1996.

[36] E. Sanchez, T. Shibata, L. A. Zadeh, "Genetic algorithms and fuzzy logic systems", *Advances in fuzzy systems: application and theory*, Vol. 7. World Scientific, River Edge, NJ, 1997.

[37] B. Kosko, "Neural networks and fuzzy systems: a dynamical systems approach to machine Intelligence", *Prentice Hall*, New York, 1991.

[38] S. Goonatilake, S. Khebbal, "Intelligent hybrid systems", *Wiley*, New York, 1996

[39] L. R. Medsker, "Hybrid intelligent systems", Kluwer, Dordrecht, 1995.

[40] D.G. Schwartz, G.J. Klir, "Fuzzy logic flowers in Japan", *IEEE Spectrum*, Vol.29, No.7, Pages 32–35, 1992.

[41] A. Zilouchian, M. Jamshidi, "Intelligent control systems using soft computing methodologies", *CRC*, Boca Raton, FL, 2001.

[42] A. W. Colombo, R. Schoop, R. Neubert, "An agent-based intelligent control platform for industrial holonic manufacturing systems," *IEEE Transactions on Industrial Electronics*, Vol. 53, No. 1, Pages 322–337, 2006.

[43] D. Z. Zhang, A. Anosike, M. K. Lim, "Dynamically integrated manufacturing systems (DIMS)—a multi-agent approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, Vol. 37, No. 5, Pages 824–850, 2007.

[44] E. M. Davidson, S. D. J. McArthur, J. R. McDonald, T. Cumming, I. Watt, "Applying multi-agent system technology in practice: automated management and analysis of SCADA and digital fault recorder data," *IEEE Transactions on Power Systems*, Vol. 21, No. 2, Pages 559–567, 2006.

[45] S. D. J. McArthur, S. M. Strachan, G. Jahn, "The design of a multi-agent transformer condition monitoring system," *IEEE Transactions on Power Systems*, Vol. 19, No. 4, Pages. 1845– 1852, 2004.

[46] D. P. Buse, Q. H. Wu, "Mobile agents for remote control of distributed systems," *IEEE Transactions on Industrial Electronics*, Vol. 51, No. 6, Pages 1142–1149, 2004.

[47] J. Zhou, G. Chen, H. Zhang, W. Yan, Q. Chen, "Multi-agent based distributed monitoring and control of hazard installations," in *Proceedings of the 2nd IEEE Conference on Industrial Electronics and Applications (ICIEA '07)*, Pages 2892–2895, 2007.

[48] J. Galdun, L. Takac, J. Ligus, J. M. Thirie, J. Sarnovsky, "Distributed control systems reliability: consideration of multi-agent behavior," in *Proceedings of the 6th International Symposium on Applied Machine Intelligence and Informatics (SAMI '08)*, Pages 157–162, 2008.

[49] K. Fregene, D. C. Kennedy, D. W. L. Wang, "Toward a systems- and control-oriented agent framework," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol. 35, No. 5, Pages 999–1012, 2005.

[50] K. S. Hwang, S.W. Tan, M.C. Hsiao, C.S. Wu, "Cooperative multi-agent congestion control for high-speed networks," *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, Vol. 35, No. 2, Pages 255–268, 2005.

[51] D. Srinivasan, M. C. Choy, "Cooperative multi-agent system for coordinated traffic signal control", *IEE Proceedings: Intelligent Transport Systems*, Vol. 153, No. 1, Pages 41–50, 2006.

[52] T. Nagata and H. Sasaki, "A multi-agent approach to power system restoration," IEEE Transactions on Power Systems, vol. 17, no. 2, pp. 457–462, 2002.

[53] S. E. Widergren, J. M. Roop, R. T. Guttromson, Z. Huang, "Simulating the dynamic coupling of market and physical system operations", *Proceedings of IEEE Power Engineering Society General Meeting*, Vol. 1, Pages 748–753, 2004.

[54] D. Koesrindartoto, J. Sun, L. Tesfatsion, "An agent-based computational laboratory for testing the economic reliability of wholesale power market designs," *Proceedings of IEEE Power Engineering Society General Meeting*, Vol. 3, Pages 2818– 2823, 2005.

[55] A. L. Dimeas, N. D. Hatziargyriou, "Operation of a multi-agent system for microgrid control," *IEEE Transactions on Power Systems*, Vol. 20, No. 3, Pages 1447–1455, 2005.

[56] D. P. Buse, P. Sun, Q. H. Wu, J. Fitch, "Agent-based substation automation," *IEEE Power and Energy Magazine*, Vol. 1, No. 2, Pages 50–55, 2003.

[57] F. Daneshfar, H. Bevrani, "Multi-agent systems in control engineering: A survey", *Journal of control science and engineering,* Vol.9, DOI:10.1155/2009/531080, 2009

[58] M. Gopal, "Control system: principle and design", Second edition, *Tata McGraw hill publication*, 2006.

[59] H.A. Simon, "The Sciences of the Artificial", *The MIT Press*, Cambridge, Massachusetts, 2nd printing edition, 1998.

[60] S. Skogestad, I. Posthlethwaite, "Multivariable Feedback Control", John Wiley & Sons, Baffins Lane, Chichester, England, 1996.

[61] R.D. Bell, K.J. Astrom, "Dynamic models for boiler-turbine alternator units: data logs and parameter estimation for a 160 MW unit", *Tech. Rep. Report LUTFD2/(TFRT-3192)/1-137*, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1987.

Chapter 3

[1] H. Nwana, M. Wooldridge, "Software agent technologies", *BT Technology Journal*, Vol.14, Issue 4, Pages 68–78, 1996. Available @ http://www.labs.bt.com/projects/agents.htm

[2] S. Franklin, A. Graesser, "Is it an agent, or just a program?", *Proceedings Third International Workshop on Agent Theories, Architectures and Languages*, Pages 193–206, 1996.

[3] N. R. Jennings, K. Sycara, M. Wooldridge, A roadmap of agent research and development, *Autonomous Agents and Multi-Agent Systems*, Vol.1, Pages 275–306, 1998.

[4] N. R. Jennings, M. Wooldridge, "Agent Technology", *Applications of Intelligent Agents*. Springer, 1998.

[5] M. M. Charles, J. N. Michael, "tutorial on agent based modeling and Simulation part 2: how to model with Agents?*", Winter* Simulation conference, IEEE, 2006.

[6] J. Ferber, "Multi-agent Systems - An Introduction to Distributed Artificial Intelligence", *Addison-Wesley*, Harlow, England, 1999.

[7] P. Stone, M. Velose, "Multi-agent systems: a survey from a machine learning perspective", *Autonomous Robotics*, Vol.8, Issue.3, 2000.

[8] V. R. Lesser. "Reflections on the nature of multi-agent coordination and its implecations for an agent architecture". *Autonomous Agents and Multi-Agent Systems*, Vol.1, Pages 89–111, 1998.

[9] K. Decker, "Coordinated problem solving", *Proceedings First European Agent Systems Summer School*, Vol. 1, July 26-30 1999.

[10] D. H. Buse, Q. H. Wu, "IP network based multi-agent systems for industrial automation", *Information management, condition monitoring and control of power systems*, ISBN-13: 9781846286469, British library publication, 2007.

[11] O. Shehory, "Architectural properties of multi-agent systems", *Research report*, Army research lab, December 1998.

[12] A.S. Rao, M.P. Georgeff, "BDI agents: From theory to practice". *Proceedings of the International Conference on Multi-Agent Systems (ICMAS)*, San Francisco, USA, Pages 312–319, 1995.

[13] F.F. Ingrand, M.P.Georgeff, A.S. Rao, "Architecture for real-time reasoning and system control", *IEEE Expert*, Vol. 7, Issue 6, Pages 34–44, 1992.

[14] R. A. Brooks, "A robust layered control system for a mobile robot"*. IEEE Journal of Robotics and Automation*, Vol.2, Issue 1, Pages 14–23, 1986.

[15] M. Wooldridge, "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence", *The MIT Press,* Cambridge, Pages 27–77, 1999.

[16] K.M. Saridakis, A.J. Dentsoras, "Soft computing in engineering design – A review", *Advance engineering informatics*, Vol. 22, Pages 202-221, 2008.

[17] Y. Shoham, "Agent0: A simple agent language and its interpreter", *In ninth national conference on artificial intelligence,* 1991.

[18] Jeinnings & Wooldridge, "April: agent process interaction language", *Intelligent Agents*, 1995.

[19] S. Nikolaos, P. Moraitis, “The gaia2JADE process for multi-agent systems development”, *Applied artificial intelligence journal*, Taylor & Francis, Vol. 20, No. 2-4, pp. 251-273, 2006.

[20] P. Mossinger, D. Polani, R. Spall, 1995, “Xraptor-a synthetic multi-agent environment for evaluation of adaptive control mechanisms”, *Proceeding of the EUROSIM Conference*, 1995.

[21] T. M. Hopkins, R. Thaens, “A multi agent system for MS windows using matlab-enabled agents”, *16th Belgian-Dutch Conference on Artificial Intelligence (BNAIC)*, 2004.

[22] T.W. Malone, K. Crowston, “The interdisciplinary study of coordination”, *ACM Computing Surveys*, Vol.26, Issue 1, Pages 87-119, 1994.

Chapter 4

[1] L.A. Zadeh, “Fuzzy sets”, *International journal of information control*, Vol.8, Pages 338-353, 1965.

[2] E.H. Mamdani, S. Assilion, “An experiment in linguistic synthesis with a fuzzy logic controller”, *International journal of man-machine studies*, Vol.7, Pages 1-13, 1974.

[3] L.P. Holmblad, J.J. Ostergaard, “Control of cement kiln by fuzzy logic”, Pages 389-399, 1982.

[4] D. Dubois, H. Prade, “What are fuzzy rules and how to use them”, *Fuzzy sets and systems*, Special issue in memory of Prof. A. Kaufmann, 1972.

[5] W. Mclloch, W. Pitts, “A logical calculus of the ideas immanent in nervous activity”, *Bulletin of mathematical biophysics,* Vol.7, Pages 115-133, 1943.

[6] B. Widrow, F.W.Smith, “Pattern recognizing control systems”, *Proceeding of computer information sciences symposium*, Pages 288-317, 1964.

[7] P.J. Werbos. “Back propagation through time: what it does and how to do it”, *Proceeding of the IEEE*, vol. 78, Issue 10, Pages 1550-1560, 1990.

[8] D.E. Rumelhart, G.E. Hinton, R.J. Williams, “Learning representations by back propagation”, *Nature*, Vol.323, Issue 9, Pages 533-536, 1986.

[9] S.N. Sivanandam, S. Sumathi, S.N. Deepa, “Introduction to neural networks using MATLAB 6.0”, *Tata McGraw-Hill Publishing Company Limited*, New Delhi, ISBN: 0-07-059112-1, 2006.

[10] M. Gopal, “Digital control state variable methods”, *Tata McGraw Hill Publication*, Addition-2005.

[11] Matlab software, www. mathworks.in, 2013

Chapter 5

[1] A. Sengupta, A. Sutradhar, V. R. Challa. “Identification of Servo-driven Inverted Pendulum System using Neural Network”, *Annual IEEE India Conference*, 2010.

Available:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5712589

[2] I. Z. Nikoliae, V. M. Rankoviae, "Identification of Nonlinear Models with Feed-forward Neural Network and Digital Recurrent Network", FME Transactions, Vol. 36, Issue 2, Pages 87-92, 2008.

[3] Longhui Shi, Xiaoli Li, Ji Li , "Modeling and Simulation of Water Level System", *Proceedings of the IEEE International Conference on Automation and Logistics*, 2008
Availabe:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4636663

[4] A. Bogdanov, 'Optimal Control of a Double Inverted Pendulum on a Cart*", OHSU Technical Report, Department of Computer Science & Electrical Engineering*, OGI School of Science & Engineering, CSE-04-006, 2004.
Availabe:http://speech.bme.ogi.edu/publications/ps/bogdanov04a.pdf

[5] L. Ljung, "System Identification – Theory for the User", Prentice Hall, Englewood Cliffs, N.J., 1987.

[6] B. Roffel, B. Betlem, "Process dynamics and control: Modeling for control and prediction", John Wiley & sons, the atrium, Southern gate, Chichester, West Sussex, England, 2006.

[7] Hunt and Sbarbaro, "Neural Networks for Control System - A Survey", *Automatica*, Vol. 28, Pages 1083-1112. 1992.

[8] A. Akbarimajd, S Kia," NARMA-L2 controller for 2-DoF under actuated planar manipulator" *International conference on control automation*, Pages 195-200, 2010.
Available:http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5707431

[9] O.M. Omidvar, D. L. Elliott, "Neural systems for control", *Academic press*, ISBN: 0-1252-6430-5, 1997.

[10] M.A.S. Kamal , M. I. Solihin, , A.Legowo, "Objective Function Selection of GA-Based PID Control Optimization for Automatic Gantry Crane*", Proceedings of the International Conference on Computer and Communication Engineering,* IEEE, pp. 883-887, 2008
Available:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4580732

[11] B. Nagaraj, Dr. N. Murugananth, "A Comparative Study of PID Controller Tuning Using GA, EP, PSO and ACO", *ICCCCT*, IEEE, pp. 301-313, 2010.
Available:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5670571

Chapter 6

[1] T. Harris, "Assessment of closed loop performance", *Journal of chemical engineering*, Vol. 67, 1989.

[2] A. Horch, "Condition monitoring of control Loops", *Ph.D. thesis*, Royal Institute of technology, Stokholm, Sweden, 2000.

[3] T. Harris, C. Seppala, L Desborough, "A review of performance monitoring and assessment techniques for univariate and multivariate control systems", *Journal of process control*, Vol. 9, 1999.

[4] A. Horch, A. Isaksson, "A modified index for control performance assessment", *Journal of process control,* Vol.9, Issue 6, 1999.

[5] A. Swanda, D. Seborg, "Controller performance assessment based on set-point Response Data", *Proceeding of the ACC*, San Diego, 1999.

[6] S.L Jamsa-Jounela, R.Pokonen, K.Halmevaara, "Evaluation of Level Control Performance", *Proceeding of the 15th IFAC World Congress*, Barcelona, 2002.

[7] M.J. Grimble, "PID Control of Multiple Model Systems and Benchmarking", *International Conference of Process Control and Instrumentation*, Glasgow, Scotland, UK, 2000.

[8] D. Uduehi, A.Ordis, M.J.Grimble, "A Generalized Predictive Control Benchmark Index for MIMO Systems", *Proceeding of the IEEE International Conference on Control Application*, Glasgow, Scotland, UK, 2002.

[9] B. Huang, S. Shah, "Performance Assessment of Control Loops", Springer, 1999.

[10] N. Thornhill, S. Shah, B .Huang, "Detection and Diagnosis of Unit-wide Oscillation", *Proceeding of the International Conference of Process Control and Instrumentation*, Glasgow, Scotland, UK, 2000.

[11] M. Jelali, "An overview of control performance assessment technology and industrial application", *Control Engineering Practice*, Vol. 14, Pages 441-466, 2006.

[12] S. J. Qin and J. Yu, "Recent developments in multivariable controller performance monitoring", *Journal of Process Control*, Vol. 17, Pages 221–227, 2007.

[13] K.M.Passino, S. Yurkovich, "Fuzzy control", Addison Wesley Longman, Inc., UAS, 1998.

[14] B. Huang, S. L. Shah, "Performance Assessment of Control Loops", Springer- Verlag, London, 1999.

[15] A. Horch, A. Isaksson, "A modified index for control performance assessment", *Journal of Process Control*, Vol. 9, Pages 475-483, 1999.

[16] J. Schafer, A. Cinar, "Multivariable MPC system performance assessment, monitoring and diagnosis", *Journal of Process Control*, Vol. 14, Pages 113–129, 2004.

[17] M. Farsi, K. Karam, H. Abdalla, "Intelligent multi-controller assessment using fuzzy logic", *Journal of Fuzzy Sets and Systems*, Vol. 79, Issue 1, Pages 25-41, 1996.

[18] A. Breemen, T. Vries, "Agent-Based Framework for Designing Multiple Controller Systems", *Proceeding Fifth International Conference on practical Applications of Intelligent Agents & Multi-Agents Technology*, Manchester, UK, Pages 219-235,11th and 12th April,2000.

[19] R. Jyringi, R. Rice, D. J. Cooper, "Opening the Black Box: Demystifying Performance Assessment Techniques", *Proceeding ISA Expo 2005*, Issue 495, Pages 25-27, 2005.

[20] O. L. Davies, P. L. Goldsmith, "Statistical Methods in Research and Production", Oliver and Boyd, Edinburgh-UK, 1972.

[21] M. Hadjiski, S. Strmcnik, K. Boshnakov, S. Gerksic, N. Christova, J. Kocijan, "On-line control performance with robust properties", *IEEE International conference on Industrial Technology*, Vol.2, Pages 918-923, 10th-12th December 2003.

[22] A. E. Ruano, "Intelligent Control Systems using Computational Intelligence Techniques", *The Institution of Electrical Engineering*, London, UK, 2005.

[23] A. Bagiş, "Determining Fuzzy Membership Functions with Tabu Search – An Application to Control", *Fuzzy Sets and Systems*, Vol.139, Pages 209-225, 2003.

Chapter 7

[1] R.A.Hilhorst, "Supervisory control of mode-switch processes", *Ph.D. Thesis*, University of Twente, the Netherlands, 1992.

[2] K.S. Narendra, J. Balakrishnan, "Adaptive control using multiple models: switching and tuning", *Proceeding of the 8th Yale Workshop on Adaptive and Learning Systems*, New Haven, CT, Pages 220-226, 1994.

[3] Humusoft, "CE150 Helicopter Model User's Guide", 2003.

[4] Michal Hoc, "Helicopter in virtual space", *Thesis*, Lulea University of Technology, Prague, 2008.

[5] V. Nazarzehi, "Identification of linear models for a laboratory Helicopter", *Second International Conference on Computer and Electrical Engineering*, 2009.

[6] K.J. Astrom, "Advanced PID control*", ISA-The Instrumentation, Systems and Automation Society,* Research Triangle Park, NC ISA, 2005.

[7] D.E. Goldberg, "Genetic algorithms in search, optimization & machine learning", Addison-Wesley, 1989.

[8] K. Krishnakumar, D.E. Goldberg, "Control system optimization using genetic algorithms", *Journal of Guidance, Control and Dynamics*, Vol. 15, Issue 3, Pages 735-740, 1992.

[9] R.Vinodha, S. Abraham Lincoln, J. Prakash, "Multiple model and neural based adaptive multi-loop PID controller for a CSTR process", *World Academy of Science, Engineering and Technology*, 2010.

[10] A. J. Breemen, T. J. Vries "Design and implementation of a room thermostat using an agent-based approach", *Control Engineering Practice*, Pages 233-248, 2001.

[11] J. Ferber, "Multi-agent systems" Harlow, England: Addison- Wesley.1999.

[12] R. J. Marks, L. J. Fox " Fuzzy-Logic Autonomous Agent Applied as a Supervisory Controller in a Simulated Environment*", IEEE transactions on fuzzy systems*, Vol. 12, Issue 1, February 2004.

[13] C. Wongsathan, C. Sirima, "Application of GA to design LQR controller for an Inverted Pendulum System", *IEEE transaction on robotics*, Pages 951-954, 2009.
[14] A. Fekih, "Improved LQR-based control approach for high performance induction motor drives", *International journal of control and intelligent systems*, ACTA press, Canada, 2009, DOI:10.2316/Journal.201.2009.1.201.2014
[15] Mathwork, MatLab & Simulink, www.mathworks.com, 2012

Chapter 8

[1] A. E. Ruano, "Intelligent Control Systems using Computational Intelligence Techniques", *The Institution of Electrical Engineering*, London, UK, 2005.
[2] A. Bagiş, "Determining Fuzzy Membership Functions with Tabu Search – An Application to Control", *Fuzzy Sets and Systems*, Vol.139, Pages 209-225, 2003.
[3] Hagan, M.T., and H.B. Demuth, "Neural Networks for Control," *Proceedings of the 1999 American Control Conference*, San Diego, CA, Pages 1642-1656, 1999.
[4] K.S. Narendra, S. Mukhopadhyay, "Adaptive Control Using Neural Networks and Approximate Models," *IEEE Transactions on Neural Networks*, Vol. 8, Pages 475-485, 1997.
[5] Narendra, K.S. and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE Transactions on Neural Network* ,Vol.1, No.1, March 1990.
[6] Coupled tank laboratory manual, *Quanser Inc.*, www.quanser.com

Appendix A

Linearization of boiler-turbine plant

The non-linear dynamics are of the form

$$\frac{dx}{dt} = f(x,u)$$
$$y = g(x,u)$$

The non-linear dynamics of boiler-turbine plant is

$$\frac{dp}{dt} = -0.0018u_2 p^{9/8} + 0.9u_1 - 0.15u_3$$

$$\frac{dp_0}{dt} = (0.073u_2 - 0.016)p^{9/8} - 0.1p_0$$

$$\frac{dp_f}{dt} = \frac{(141u_3 - (1.1u_2 - 0.19)p)}{85}$$

The output to the system

$$X_w = 0.05(0.13073p_f + 100\alpha_{cs} + \frac{q_e}{9} - 67.975)$$

$$\alpha_{cs} = \frac{(1 - 0.001538p_f)(0.8p - 25.6)}{p_f(1.0394 - 0.0012304)}$$

$$q_e = (0.854u_2 - 0.147)p + 45.59u_1 - 2.514u_3 - 2.096$$

The notation for boiler-turbine plant is given in table A. And linearization of the system about the nominal operating point (x0, u0), requires calculating the linear system matrices

$$A = \left[\frac{\partial f}{\partial x}\right]_{(x^0,u^0)}, B = \left[\frac{\partial f}{\partial u}\right]_{(x^0,u^0)}, C = \left[\frac{\partial g}{\partial x}\right]_{(x^0,u^0)}, D = \left[\frac{\partial g}{\partial u}\right]_{(x^0,u^0)}$$

Table A.1 Notation for boiler-turbine plant

Notation	Parameter	State variable	Output Variable	Control variable
P	Drum pressure	x_1	y_1	
P_o	Power output	x_2	y_2	
P_f	Fluid density	x_3		
u_1	Fuel flow valve position			u_1
u_2	Steam valve position			u_2
u_3	Feed water valve position			u_3
X_w	Drum water level deviation		y_3	
q_e	Evaporation rate			
α_{cs}	Steam quality			

The linear approximation to the system is

$$\frac{d\bar{x}}{dt} = A\bar{x} + B\bar{u}$$

$$\begin{bmatrix} \dot{p} \\ \dot{p}_0 \\ \dot{p}_0 \end{bmatrix} = \begin{bmatrix} a_1 & 0 & 0 \\ a_2 & a_3 & 0 \\ a_4 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ p_0 \\ p_f \end{bmatrix} + \begin{bmatrix} b_1 & b_2 & b_3 \\ 0 & b_4 & 0 \\ 0 & b_5 & b_6 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$\bar{y} = C\bar{x} + B\bar{u}$$

$$\begin{bmatrix} p \\ p_0 \\ X_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_1 & 0 & c_2 \end{bmatrix} \begin{bmatrix} p \\ p_0 \\ p_f \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ d_1 & d_2 & d_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

Where $\bar{x} = x - x^0, \bar{u} = u - u^0, \bar{y} = y - y^0$

And $a_1 = -0.002025 \times u_2 \times x_1^{1/8}$,

$a_2 = (0.002025 \times u_2 \times x_1^{1/8}) - (0.0821 \times x_1^{1/8})$,

$a_3 = -0.1$,

$a_4 = (-1.1 \times u_2 + 0.19)/85$,

$b_1 = 0.9$,

$b_2 = -0.0018 \times x_1^{9/8}$,

$b_3 = -0.15,$
$b_4 = 0.073 \times x_1^{9/8},$
$b_5 = (-1.1 \times u_2 + 0.19)/85,$
$b_6 = 1.6588,$
$z_0 = (x_3(1.0394 - 0.001204 \times x_1))^2$
$z_1 = (1 - 0.001538 \times x_3)[0.8x_3(1.0394 - 0.001204 \times x_1)] - [-0.001204x_3],$
$z_2 = (0.8x_1 - 25.6)(1.0394 - 0.001204 \times x_1)(-0.001538x_3 - (1 - 0.001538 \times x_3)),$
$c_1 = 5(z_0 / z_2) + 0.05(0.854u_2 - 0.147)/9,$
$c_2 = 5(z_0 / z_2) + (0.05 \times 0.13073),$
$d_1 = (0.05/9) \times 45.59,$
$d_2 = (0.05/9) \times (0.854 \times x_1),$
$d_1 = (0.05/9) \times (-2.514),$

This formulation is used to identify the linear plant model based on the initial condition of plant and operating parameters. The plant model is derived as per given in section 2.3.5 for given $x^0 = [108, 66.65, 428]^T$ and $u^0 = [0.34, 0.69, 0.436]^T$ operating regime.

Appendix B
Prototype system: coupled tank

B.1 Coupled tank system description

The typical coupled-tank is depicted in figure B.1. The coupled tank specialty module is a benchtop “Two-tank” plant consisting of a pump with water basic and two tanks of uniform cross sections. Such an apparatus forms an autonomous closed and recirculating system. The two tanks, mounted on the front plate, are configured such that flow from the first (upper) tank can flow into the second (lower) tank. Flow from the second tank flows into the main water reservoir. In each one of the two, tanks, liquid is withdrawn from the bottom through an outflow orifice (i.e. outlet).

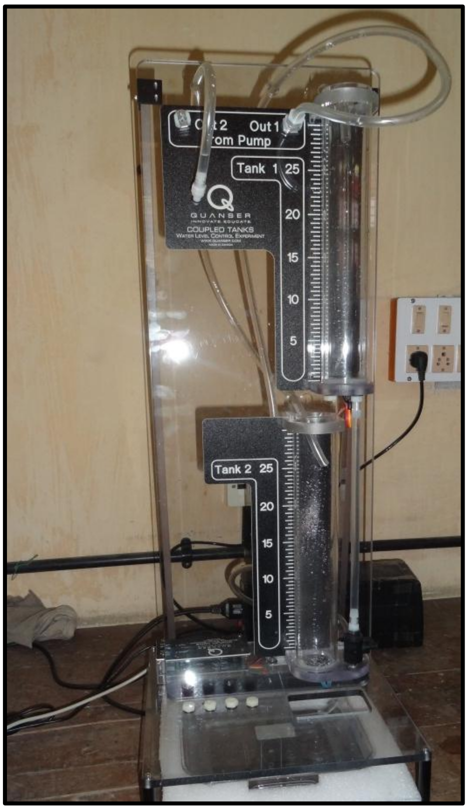


Figure P.1 Coupled Tank System

The outlet pressure is atmospheric. Both outlets inserts are configurable and can be set by changing inserts that screw into the tapped holes at the bottom of each tank. In order to introduce a disturbance flow, the first tank is also equipped with a drain tap so that, when opened, flow can be released directly into the water basin. The pump thrusts water vertically to two quick-connect orifices "out1" and "out2" which are normally closed. For configurability purpose, these two orifices, or inlets, have different diameters. Rubber tubing with appropriate coupling is supplied to enable the pump to feed water into one or both tanks. The selection of outputs from the pump controls the flow ratio between the two outlets "out1" and "out". The water level in each tank is measured using a pressure–sensitive sensor located at the bottom of the tank. The offset and gain potentiometers of each pressure sensor are readily available for proper calibration. Additionally, a vertical scale is also placed beside each for visual feedback regarding each tank's water level. The single system can be used in three configurations as given below.

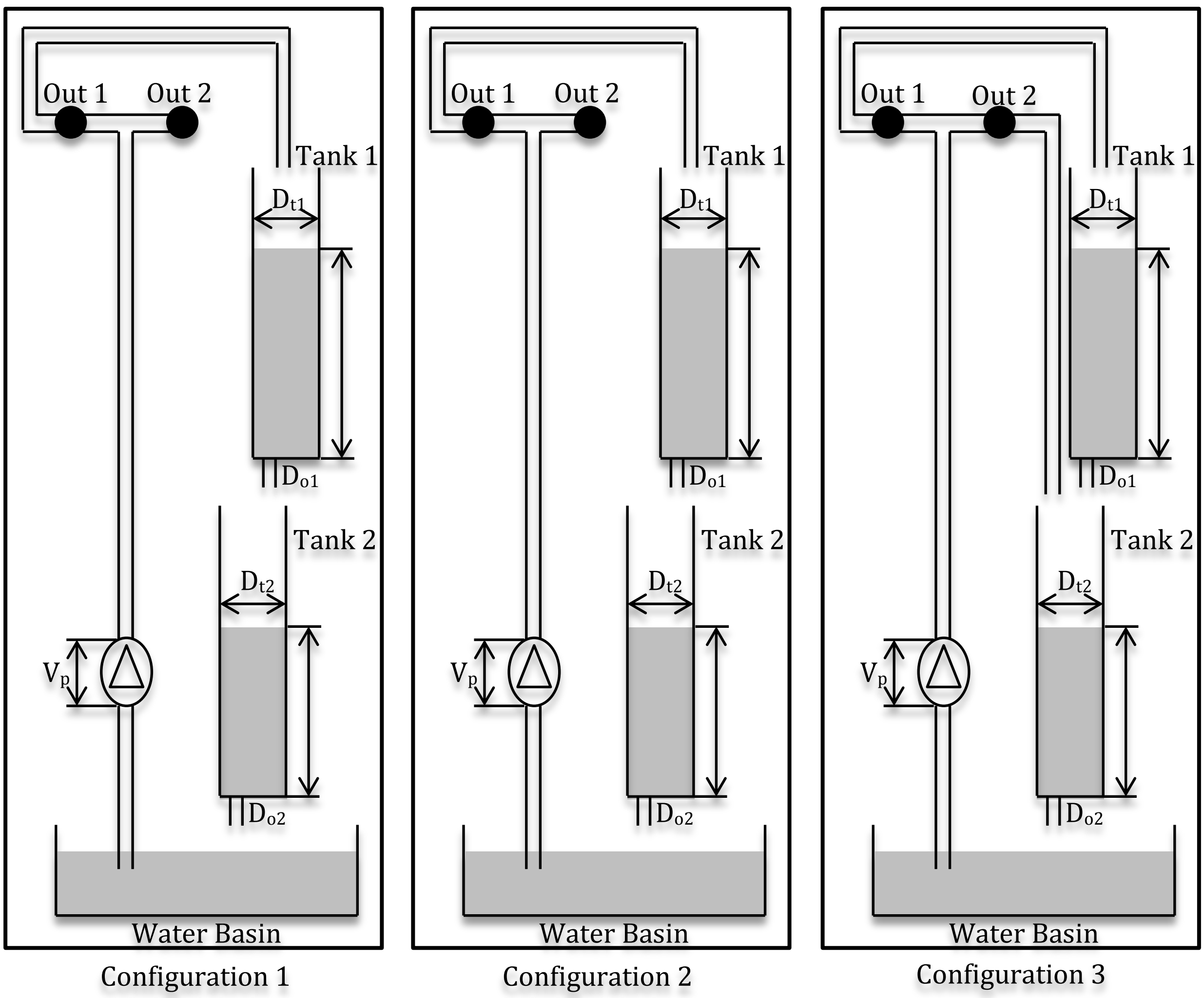


Figure B.1 Different configurations of coupled Tank System

Configuration 1

- Single input single output system
- Pump feeds the tank 1 and tank 2 is not used at all.
- Controller is designed to track the level of tank 1.
- Different inlet-outlet diameter in tank 1 can be set-up and tried.

Configuration 2

- State coupled Single input single output system
- Pump feeds the tank-1 which is turned to feed tank 2.
- Controller is designed to regulate or track the level of tank 2.
- Different inlet-outlet diameter in tank1 and tank 2 can be set-up and tried.

Configuration 3

- State coupled and input coupled SISO system
- Pump feeds the tank1 and tank 2 with split flow. Also, tank 1 feeds tank 2..
- Controller is designed to regulate or track the level of tank 2.
- Different inlet-outlet diameter in tank 1 and tank 2 can be set-up and tried.

For, this study the configuration 1 is used as single input single output to check the response of MAC architecture.

B.2 Component Nomenclature

As a quick nomenclature, Table B.1 provides a list of all the principal elements of the system. Every element is located and identified through a unique identification number in figure B.2.

Table B.1 component nomenclature

ID#	Description	ID#	Description
1	Coupled tank overall frame	2	Tank 1
3	Tank 2	4	Main water basin
5	Pump	6	Flexible tubing
7	Quick connect inlet orifice “Out 1”	8	Quick connect inlet orifice “Out 2”
9	“Out 1” coupling and hose	10	“Out 2” coupling and hose
11	Small outlet insert	12	Medium outlet insert
13	Large outlet insert	14	Plain outlet insert
15	Disturbance tap	16	Flow splitter
17	Pressure sensor	18	Signal conditioning circuit board
19	Pump Motor 4-pin connector	20	Pressure sensors 6-pin connector
21	Tank level scale		

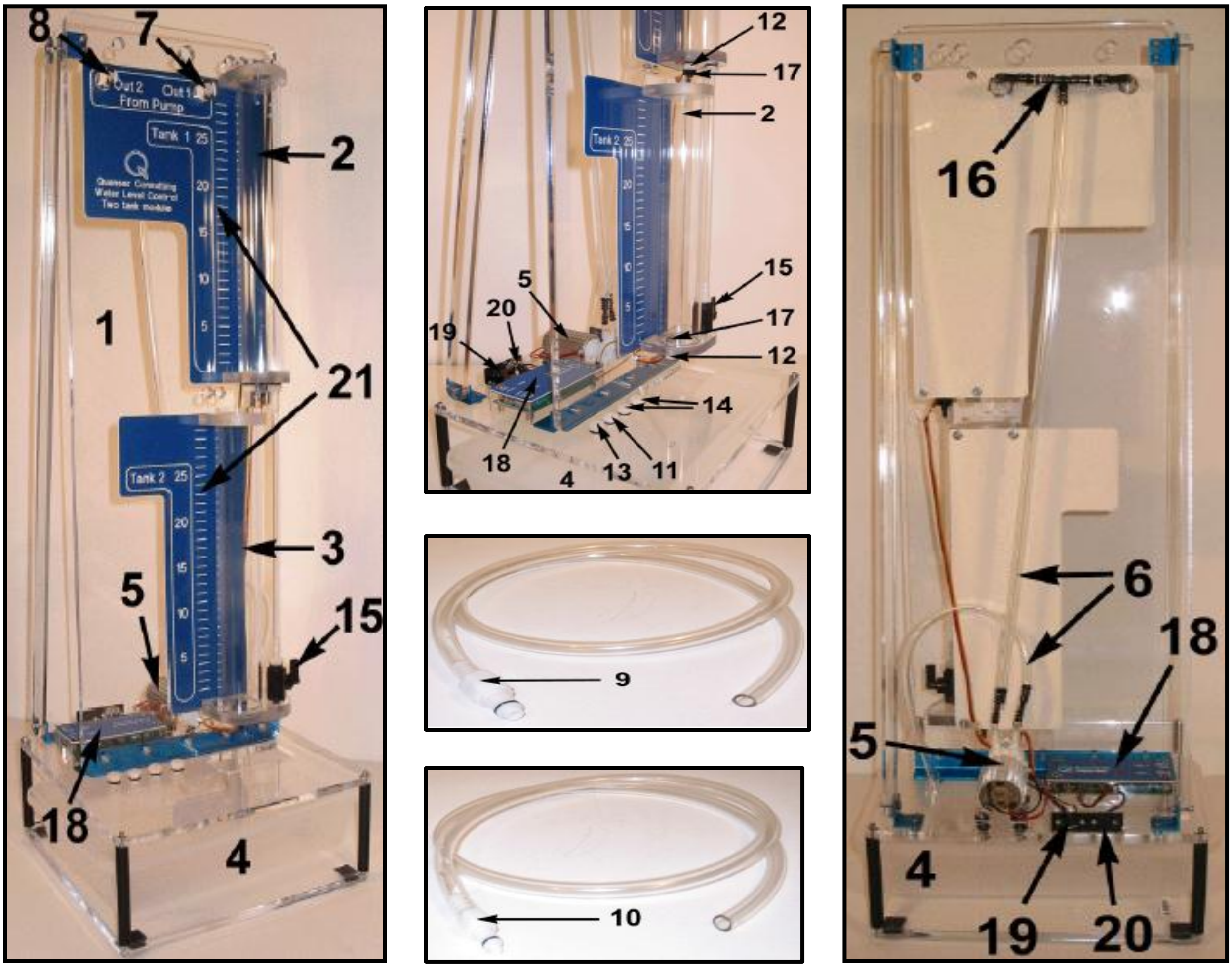


Figure B.2 Coupled Tank System: different views

B.3 Coupled-tank model parameters

Table B.2 is lists and characterize the main parameters (e.g. mechanical and electrical specifications, conversion factors, constants) associated with plant. Some of these parameters are used for mathematical modeling of the coupled-tank system as well as to obtain the water level equation.

B.4 Wiring procedure

This section describes the standard wiring procedure for the plant. The following hardware, accompanying the coupled tanks is assumed:

- Power amplifier Quanser VoltPAQ
- Data Acquisition Card Quanser Q2-USB

Table B.2 Value of parameters of coupled tank

Symbols	Description	Value	Unit
K_p	Pump flow constant	3.3	$Cm^3/s/V$
V_{p_max}	Pump maximum continuous voltage	12	V
V_{p_peak}	Pump peak voltage	22	V
D_{out1}	"Out 1" orifice diameter	0.635	cm
D_{out2}	"Out 1" orifice diameter	0.476	cm
L_{1_max}	Tank 1 height	30	cm
D_{t1}	Tank 1 inside diameter	4.445	cm
K_{L1}	Tank 1 water level sensor sensitivity	6.1	cm/V
L_{2_max}	Tank 2 height	30	cm
D_{t2}	Tank 2 inside diameter	4.445	cm
K_{L2}	Tank 2 water level sensor sensitivity	6.1	cm/V
V_{bias}	Tank 1 and Tank 2 pressure range	0-6.89	kPa
D_{so}	Small outflow office diameter	0.31750	cm
D_{mo}	Medium outflow office diameter	0.47625	Cm
D_{lo}	Large outflow office diameter	0.55563	cm
g	Gravitational constant	981	cm/s^2

Figure B.4 shows the DAQ terminal board, the back of the coupled-tank plant and the amplifier, all connected with the necessary cabling to interface to and use the coupled-tank plant.

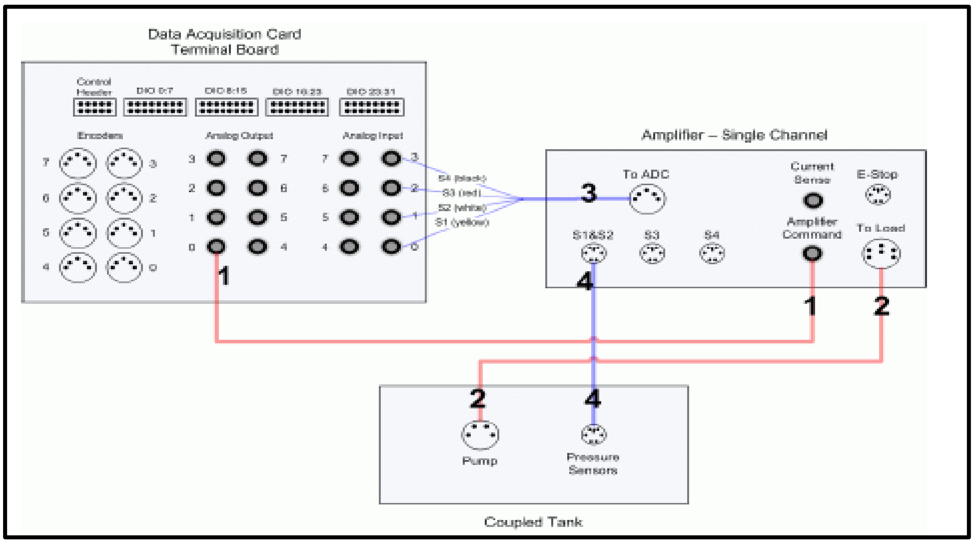


Figure B.3 Wiring diagram for coupled tank system for interfacing with PC

B.5 Nonlinear equation of motion

In configuration #1, a single tank system, consisting of the top tank (i.e. tank 1), is considered. The designed closed-loop system is to control the water level inside tank 1 via the commanded pump voltage. The outflow rate from tank 1, F_{o1}, can be expressed by:

$$F_{o1} = A_{o1} V_{o1}$$

The cross-section area of tank 1 outlet hole can be calculated by:

$$A_{o1} = \frac{1}{4} \pi D_{ol}{}^{2}$$

The outflow rate from tank 1 given in equation becomes:

$$F_{o1} = A_{o1} \sqrt{2} \sqrt{g L_1}$$

The inlet flow of tank 1 is as:

$$F_{i1} = K_p V_p$$

Moreover using the mass balance principle for tank 1, the following first order differential equation in L_1

$$A_{t1} \frac{d}{dt} L_1 = F_{i1} - F_{o1}$$

From the above equation and rearranging results in the following equation of motion for the tank 1 system:

$$\frac{d}{dt} L_1 = \frac{K_p V_p - A_{o1} \sqrt{2} \sqrt{g L_1}}{A_{t1}}$$

Due to the square root function applied to L_1, the first-order differential equation expressed by above equation is non-linear. At equilibrium $V_p = V_{po}$ and $L_1 = L_{1o}$, all time derivative terms equate zero and equation becomes:

$$K_p V_{po} - A_{o1} \sqrt{2} \sqrt{g L_{1o}} = 0$$

Solving above equation for V_{po} gives the pump voltage at equilibrium. V_{po} results to be a function of L_{1o} and K_p as expressed below:

$$V_{po} = \frac{A_{o1} \sqrt{2} \sqrt{g L_{1o}}}{K_p}$$

Using the system's specification given in table B.1 and the evaluation of equation results to be :

$$V_{po} = 9.26\ [V]$$

Applying the Taylor's series approximation about (L_{1o}, Vpo), the equation can be linearized as represented below:

$$\frac{d}{dt}L_1 = \frac{K_p V_{po} - A_{o1}\sqrt{2}\sqrt{gL_{1o}}}{A_{t1}} - \frac{A_{o1}\sqrt{2}gL_{11}}{2\sqrt{gL_{1o}}A_{t1}} + \frac{K_p V_{p1}}{A_{t1}}$$

Substituting V_{po} in equation results to the following linearized EOM for the tank 1 water level system:

$$\frac{d}{dt}L_{11} = -\frac{A_{o1}\sqrt{2}gL_{11}}{2\sqrt{gL_{1o}}A_{t1}} + \frac{K_p V_{p1}}{A_{t1}}$$

Applying the Laplace transform to equation and rearranging yields the desired open-loop transfer function for the coupled-tank's tank1 system such that:

$$G_1(s) = \frac{K_{dc\_1}}{\tau_1 s + 1}$$

With $$K_{dc\_1} = \frac{K_p\sqrt{2}\sqrt{gL_{1o}}}{A_{o1}g}$$ and $$\tau_1 = \frac{A_{t1}\sqrt{2}\sqrt{gL_{1o}}}{A_{o1}g}$$

Evaluating equation, accordingly to the system's parameters and desired design requirements gives:

$$K_{dc\_1} = 3.2\left[\frac{V}{cm}\right]$$ and $$\tau_1 = 15.2[\mathrm{s}]$$

This modeling is used for simulation and real time application analysis of MAC controller. The photographs of prototype are given in figure P.2 to P.6.

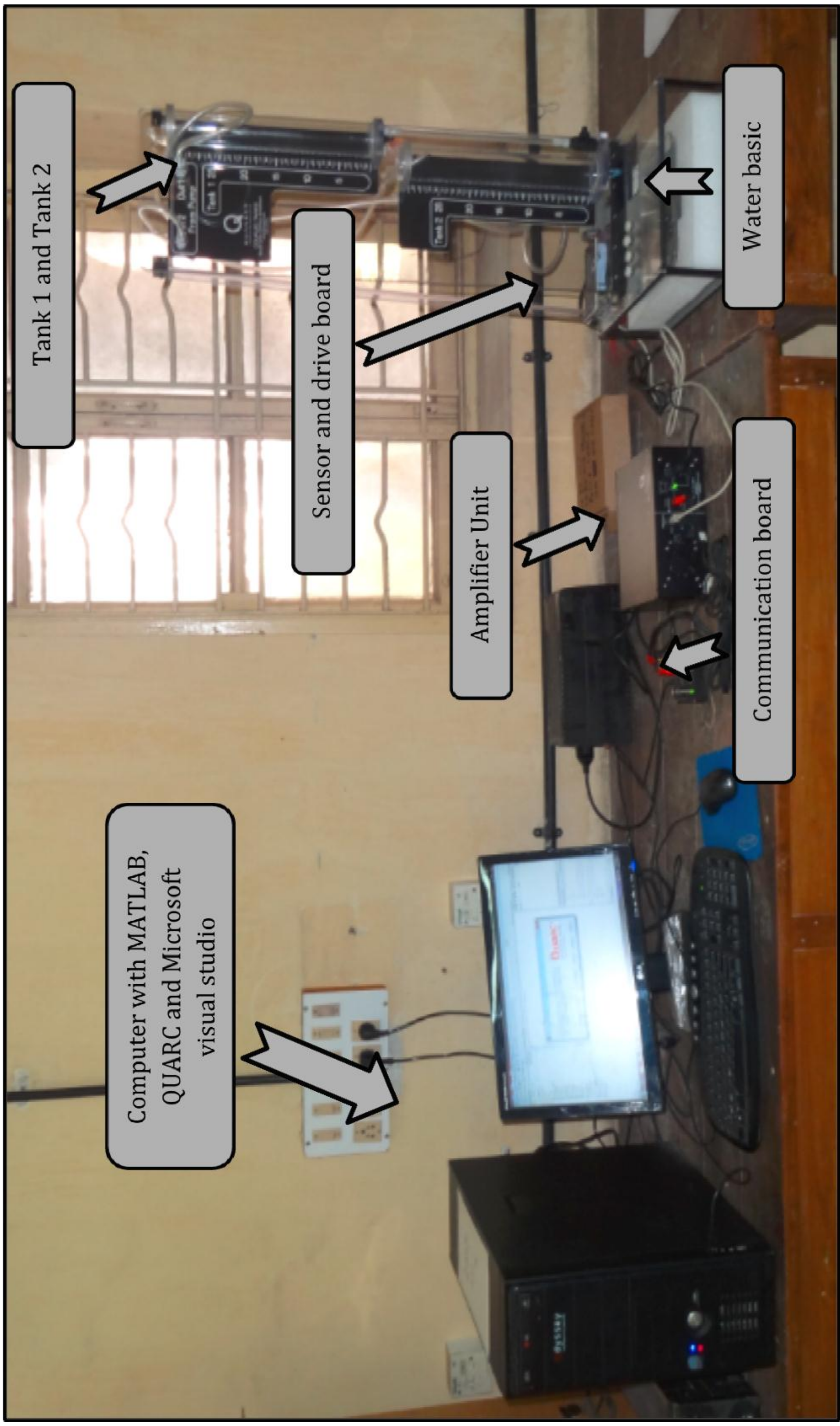


Figure P.2 Real time system with interfacing computer

Figure P.3 Communication board & Amplifier unit

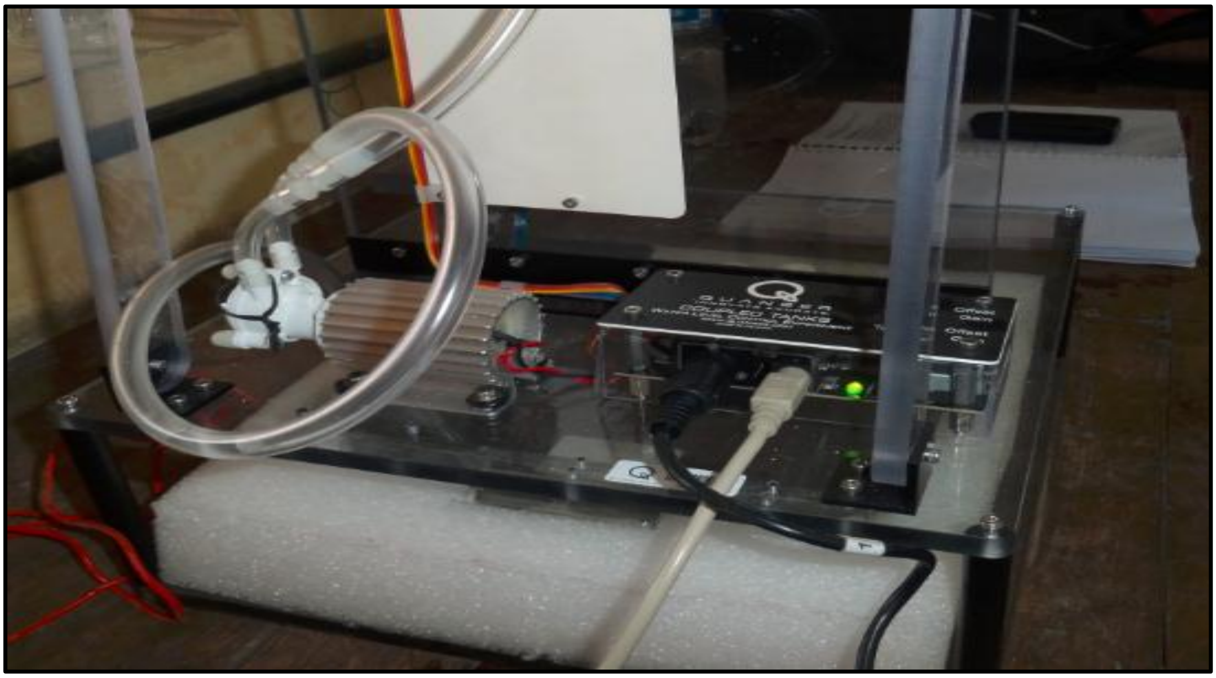


Figure P.5 Water Pump and Sensor & Drive unit

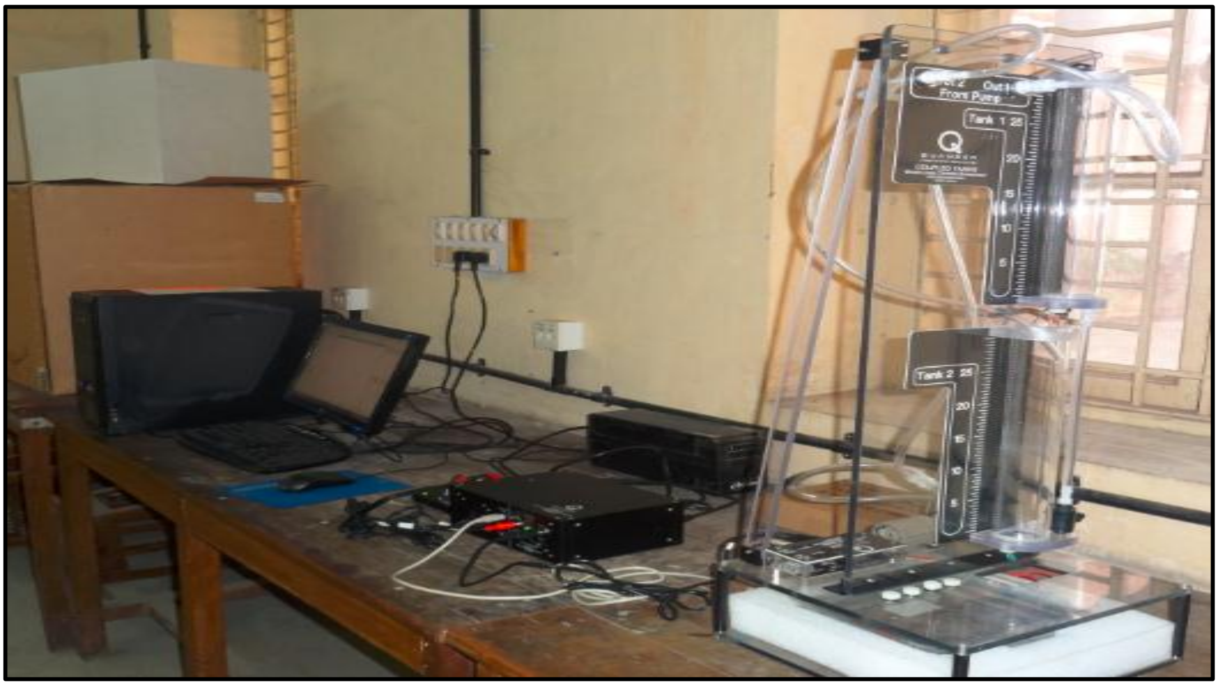


Figure P.5 View of real time system

Figure P.6 View of result

Appendix C
Development of software program

List of software and developed files for multi-agent controller architecture are as below.

List of software used to develop the algorithm:

Name of software	Description	Company
Matlab & Simulink (Ver 7.12, R2011a) (Ver 7.10, R2010a)	For simulation and overall of development of whole algorithm	Mathwork, USA www.mathwork.com
Quarc accelerate design (Ver 2.1)	For interface real-time system and matlab	Quanser, Canada www.quanser.com
Visual Studio 2010	For interface real-time system and matlab	Microsoft, USA

Matlab Files:

Name of file	**Description**
setp_info.m	To identify the parameter as pick overshoot, rise time and settling time
multi.m	To generate various set-points to check the response of controller
nn_backPP.m	To train the neural network using back propagation
fit_fun_ga.m	Fitness function for genetic algorithm
nn_ga_withfitness.m	To tune the neural network using genetic algorithm
CPAfuzzy.m	Matlab file for the control performance assessment
Heliga.m	Fitness function of mean square error for helicopter
ARXmodelTimeDelayNeural Netwrok.m	Develop the neural network for identification using arm neural network
num_val.m	Calculate the number of weight and bias of the neural network
tankNN.m	Neural network training
DataRangeCoversion.m	Convert the performance parameter to suitable fuzzy set range
PerfAssessDataGenIntial.m	Parameters for performance assessment based on neural network identification
IdenPara.m	Online parameter identification using neural network tuned through genetic algorithm
ModernAgent.m	Modern agent design using modern control theory

ClassicalAgent.m	Classical agent design using classical control theory
PlantModel.m	Boiler–turbine plant model
PerformanceAgent.m	Performance assessment agent for boiler turbine plant
ControllerResult.m	Comparative result analysis of Multi-agent controller and tradition controller
FuzzyAgent.m	Fuzzy based critic design for boiler turbine plant

Simulink files:

Name of file	Description
PIDcontroller.mdl	PID controller for water tank system
FuzzyController.mdl	Fuzzy controller for water tank system
NNController.mdl	Neural network controller for water tank system
Model.mdl	Model of water tank system for simulation
PIDcontrollerNoise.mdl	PID controller with noise
PIDcontrollerWithVariableIntialCon.mdl	Change in initial condition of plant and the response of PID controller
RecogniseAgent.mdl	To find the values of steady state error, overshoot and variance
HeliModel.mdl	Helicopter model
PIDmultiple.mdl	Multiple PID controllers for various operating regime of helicopter
NARMANNcontroller.mdl	NARMA-l2 controller for water tank system
CriticDesign_MAC.mdl	Critic agent design for mutli-agent controller
MAC.mdl	Multi-agent controller for water tank system
MACwithNoise.mdl	Multi-agent controller performance in noise environment

Fis files:

Name of file	Description
WaterVolatgeFuzzyChange.fis	Fuzzy sets and its structure for fuzzy controller of water tank system
CriticAgentFuzzyWithSixVaraible.fis	Fuzzy base the critic design

Graphical user interface:

Name of file	Description
CPAfuzzy.fig	Graphical user interface for the control performance assessment
Advance MAC_SC.fig	Graphical user interface for the Multi-agent controller for boiler turbine plant (Various file for it)
Multi-agent controller.fig	Graphical user interface for summary work

Appendix D
List of research publications

Following is the list of our publications, presentations, award and received grant relevant to the work included in this thesis.

Referred Journal

- **R.B.Patel**, S.K.Shah (2013).”Multi-agent controller using soft computational paradigms”, Indian journal on applied research. Vol.3, Issue 2, Pages 138-142, ISSN: 2249-555X. ***(Impact factor = 0.825)***
  (**Online:** http://www.theglobaljournals.com/ijar/file.php?val=MTAwOQ==)
- **R.B.Patel**, S.K.Shah (2012).”Multi-agent based controller for boiler turbine plant”, international journal of control & intelligent system. Accepted for publication after review (Publisher: ACTA press, Canada)
- **R.B.Patel**, S.K.Shah (2011).”Controller performance assessment (CPA) of intelligent control for nonlinear system”. Journal of control & instrumentation, ISSN: 2229-6972, Vol.2 (2-3), Pages 17-23.
  (**Online:**http://www.stmjournals.com/index.php?journal=JoCI&page=article&op=view&path%5B%5D=1378)

Peer-reviewed International Conferences

- **R.B.Patel**, S.K.Shah (2013). “Tuning of classical controller using evolutionary methods for real time system”. International conference on control system and power electronics, CSPE-2013, Dubai, UAE, July 11th-12th 2013. Accepted after review

- **R.B.Patel**, S.K.Shah (2012).”Identification of non-linear system using computational paradigms”. International conference on automatic control & artificial intelligence ACAI-2012, ISBN: 978-1-84919-5379, Pages 1156-1159, Xiamen, China, March 24th-26th 2012, Publisher: IET, U.K. DOI: 10.1049/cp.2012.1183
  (Online: http://digital-library.theiet.org/content/conferences/10.1049/cp.2012.1183)

(Online:http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6492790&searchWithin%3Dpatel%26sortType%3Dasc_p_Sequence%26filter%3DAND%28p_IS_Number%3A6492510%29)

- **R.B.Patel**, S.K.Shah (2011)."GA based competitive multi-agent controller for non-linear system". 3rd international conference on mechanical and electrical technology ICMET-2011, ISBN: 978-0-7918-5981-0, Dalian, China, August 26th-27th 2011, Publisher: ASME press, New York, USA. DOI: 10.1115/1.859810.paper186
(Online: http://asmedl.org/ebooks/asme/asme_press/859810/859810_paper185)

- S.K.Shah, **R.B.Patel**, S.R.Panchal (2010)."Design and implementation of multi-agent controller for SISO system", International conference on system dynamics and control ICSDC-2010, ISBN: 978-93-80578-58-3, Manipal University, Karnataka, India, August 19th-20th 2010, Publisher: I. K. Publication, New-Delhi.

National Conference and Technical Event

- **R.B.Patel** (2012). "Adaptive neuro-fuzzy controller for non-linear system". IT security using 4G communication ITS4G-2012, Institution of electrical and electronics engineers, Vadodara, April 22nd 2012.

- **R.B.Patel** (2011). "Multi-agent system: A recent trends in control engineering". Control, Microprocessor, Electronics & Communication CMEC-2011, Institution of electrical and electronics engineers, Vadodara, February 20th 2011.

International Presentation

- "Identification of non-linear system using computational paradigms". International conference on automatic control & artificial intelligence ACAI-2012, Xiamen, China, March 24th-26th 2012.

National Presentations

- "Adaptive neuro-fuzzy controller for non-linear system". IT security using 4G communication ITS4G-2012, Institution of electrical and electronics engineers, Vadodara, April 22nd 2012.

- "Multi-agent system: A recent trends in control engineering". Control, Microprocessor, Electronics & Communication CMEC-2011, Institution of electrical and electronics engineers, Vadodara, February 20^{th} 2011.

Publication in International Book

- S. Sumathi, P. Surekha (2011). "Computational Intelligence paradigms: Theory and application using matlab" CRC press, Taylor & Francis Publication, U.K., ISBN: 978-1-4398-0902-0, **R.B.Patel** contributed on "Fuzzy logic based washing machine: Matlab code".

Technical Award

- 2^{nd} prize in review category for presentation on "multi-agent system: A recent trends in control engineering"

Travel Grant

- Travel grant **Rs. 1,00,000/-** received from AICTE, New-Delhi to attend the international conference on "Automatic control & artificial intelligence ACAI-2012" @Xiamen, china, March 24^{th}-26^{th} 2012. ***(File no. 1-7/RID/TR/(110)/2011-12)***