Chapter 2

Background

2. Background

Modern Control theory, which has contributed so significantly to the exploration and conquest of space, has not had similar success in solving the control problems under various categories. Even though its progress, the chasm between theory and practice has been widening and many needs of the industry remain unsolved. Industry has had little choice, therefore, but to rely heavily on conventional control techniques that are based on PID controllers. Unfortunately these simple and ubiquitous devices can't always cope with the demands and complexity of modern systems.

The chasm between theory and practice has led to a search for new and unconventional techniques that are not subject to the constraints and limitation of modern control theory to solve the control problems faced by present day real time system.

As we move from linear to non-linear systems, we are faced with more difficult situation. The basic principles of control theory derived in classical methods do not hold any longer and analysis tools involve more complex mathematics.

Chapter provides a comprehensive study of the work done by the researchers using conventional as well as cognitive techniques for the design and development of real time control systems. The survey is classified as per the application of techniques based on Conventional methods, intelligent methods & Hybrid methods.

2.1 Conventional Methods

These methods depend on empirical knowledge of the dynamic behavior of the controlled plant, derived from the measurement of control and manipulated variables of the system. Traditionally we have relied heavily on three term (PID) controllers and Programmable logic controllers. Over the period researchers have used following methods to explore the various control issues.

2.1.1 Gain Scheduling

A technique for transforming original system models into equivalent models of a simpler form, also known as linearization about an equilibrium point. In this case it can be said that the linearization may not be a good approximation to the system for arbitrary configurations. Since the system is linearized about a single point, trajectory tracking can only be guaranteed in a sufficiently small ball of states about that point. There are several methods for circumventing this problem; One of the most common is gain scheduling as analyzed by Shamma and Athans[58]. The idea of gain scheduling is to select a number of operating points which cover the range of system operation. Then at each of points, the designer makes a linear time invariant approximations to the plant dynamics and designs a linear controller for each linearized plant. Between the operating points, the parameters of the compensators are interpolated, or scheduled, thus resulting in a global compensator. To use gain scheduling, tracking controllers are designed for many different equilibrium points and gains are chosen based on the equilibrium points to which the system is nearest.

The main problem is that it has only limited theoretical guarantees of stability in nonlinear operations, but it uses some loose practical guidelines such as "the scheduling variables should change slowly" and "the scheduling variables should capture the plant's nonlinearities". Another problem is the computational load in a gain scheduling design, due to the necessity of computing many linear controllers.

2.1.2 Feedback Linearization

An alternative technique is Feedback linearization, also known as non-linear dynamic inversion. Feedback linearization deals with techniques for transforming original system models into equivalent models of simpler form. Feedback linearization can be used as a non-linear design methodology. The main idea is to algebraically transform a nonlinear system into a linear form using state feedback like in [59,60] and then to use the well known linear design techniques to complete the control design. The purpose of dynamic inversion is to develop feedback control law that linearizes the plant response to commands, then a non linear control law is designed which globally reduces the dynamics of the selected controlled variables to integrators. A closed loop system is then designed to make the controlled variable exhibit the specified command response and robustness requirement of the overall system.

The approach can be used for both stability and tracking control problems and has been applied to a number of practical nonlinear control problems, including the control of helicopter, high performance aircrafts and industrial robots, electro servo hydro actuator [61,62,63].

Feedback linearization techniques requires full state measurement and desired tracking performance is only valid for exact knowledge of model parameters, however can be useful as model simplifying device for robust nonlinear control for real time systems, which are capable to provide robustness of the closed loop systems.

2.1.3 Variable Structure Control techniques

Variable structure control is a viable high speed switching feedback control. This variable structure control law provides an effective and robust means of controlling nonlinear plants. Essentially it utilizes a high speed switching control law to derive the nonlinear plant's state trajectory onto a specified and user chosen surface in the switching surface and to maintain the plant's state trajectory on this surface for all subsequent time [64]. The plant dynamics restricted to switching surface represent the controlled system's behavior. With proper design of switching surface, variable structure control attains the conventional goals of control such as stabilization, tracking, regulation etc.

Concepts of variable structure control systems have been utilized in the design of robust regulators, model reference systems, adaptive schemes, tracking systems, state observers, fault detection systems etc.[65,66].

This method is attractive in the design of controls for non linear uncertain dynamics systems with uncertainties and nonlinearities of unknown structure as long as they are bounded and occurring within a subspace of the state space, that's the reason that when this method is applied to a particular problem a very high quality control system results. Another problem is the need for complete state information.

2.1.4 Sliding Mode Control

The aim of the sliding mode controller is to design a non linear feedback controller for a class of nonlinear systems given the extent of parametric uncertainty, disturbances and range of unmodelled dynamics. The one of the most intriguing aspects of sliding mode control is the discontinuing nature of the control action, whose primary function of each of the feedback channels is to switch between two distinctively different system structures such that a new type of system motion, called sliding mode, exists in a manifold.[67,68] This peculiar system characteristic is claimed to result in superb system performance which includes insensitivity to parameter variation and complete rejection of disturbances.

Advantage of this method is only single design is required over the entire operating range of the plant so there is no need for a series of linear controllers. Stability is maintained in Lynapunov sense. This method has excellent robustness properties against parametric uncertainties when matching conditions are satisfied. In practice the switching, chattering control law should be replaced by a smooth approximation, which can be very inconvenient. Another drawback can be pointed as the need of complete state information, which may not be always available.

2.1.5 Back-stepping Approach

Another technique defined as a different version of variable structure control is the back-stepping approach. This technique has been approached by researchers for output tracking problem of a class of observable minimum phase uncertain nonlinear systems, an adaptive nonlinear control design etc. [69,70,71]. This approach can be applied to a large class of non-linear systems, including those that are not transferable into the parametric-pure and parametric – strict feedback forms.

Back stepping approach is a very promising technique for an autopilot design of missiles, which are highly nonlinear in aerodynamics with unknown parameters. This approach is very robust to parametric uncertainties. By properly chosen Lyapunov function a global asymptotic stability can be proved, conversely to sliding mode control no chattering effect is involved. However, there is need of an observer for the estimation procedure which is definitely not very appreciated by real time problems, when a fast response is required from the design.

2.2 Intelligent Methods

During past two decades or so, a major effort has been under way to develop new and conventional control techniques that can often augment or replace conventional control techniques. A number of unconventional control techniques have evolved, offering solutions to many difficult control problems.

An intelligent system should be able to cope with a variety of unexpected changes and environments, which requires learning and adaptation ability. Such a system can be referred to as an intelligent control system where technology plays major role in modern control design and implementation. Goal of the intelligent control approach is to make advanced control systems easier to design and also to make them less vulnerable to uncertainties in system parameters and to unknown environment.

This is the domain of soft computing, which focuses on stochastic, vague, empirical and associative situations, typical to most applications. Intelligent controllers are derivatives of soft computing being characterized by their ability to establish the functional relationship between their inputs and outputs from empirical data, without recourse to explicit models of the controlled process. This is radical departure from conventional controllers, which are based on explicit functional relations.

Unlike their conventional counterpart, intelligent controllers can learn, remember and make decision. Intelligent controllers use a qualitative description on how a process operates instead of an explicit quantitative description of the physical principles that relate the causes to the effects of the process. An intelligent controller is therefore based on the knowledge, stated linguistically in the form of production rules, which are elicited from human experts. Appropriate inference mechanism must then be used to process this knowledge in order to arrive at suitable control decisions. One or more of the following techniques of computational intelligence are used to this end. [72]

1. Expert Systems
2. Fuzzy Control
3. Neural Control
4. Neuro – Fuzzy Control
5. Evolutionary Computation

2.2.1 Expert Systems

The objective of an expert system is to permit a non-expert user to exploit the accumulated knowledge and experience of an expert in a specific field of expertise. Knowledge based expert systems use rules, data, events, simple logic and other form of knowledge in conjunction with search techniques to arrive at decisions. Where an element of knowledge is missing then the expert system cannot but return a don't know response, implying an inability to arrive at a decision. An expert system cannot extrapolate data and infer from similar or adjacent knowledge.

Expert system operate either in online when decisions are required in real time, as in case of fault detection, energy management and supervisory control, or offline, as in the cases of interactive, dialog based systems for fault diagnosis and production management.

In the case where a complete decision tree that can account for every possible situation that may arise in practice is available, then such knowledge based systems offer a simple and effective solution to the unconventional control problems. Its obvious that situations involving uncertainty and vagueness cannot be treated effectively using conventional decision tree based expert system. In contrast, when insufficient or incomplete knowledge about a process is available and when uncertainty and vagueness characterize the plant and its measurements, then such knowledge based expert systems are not always able to arrive at decision and are consequently unacceptable for real time control purposes. In such situations of uncertainty and vagueness, more effective mechanisms, capable of inferring decisions from incomplete data, are necessary. Fuzzy logic, Artificial Neural Network, Genetic Algorithm and their hybrids are the primary examples of techniques that possess appropriate mechanisms to deal with uncertainty and vagueness.

2.2.2 Fuzzy Control

Control system should have the capability to gain increasing knowledge of the system through operational experience, without interference of human operators. The knowledge based control techniques use reasoning mechanisms to determine the control action from the knowledge stored in the system and from the available measurements. These systems can improve the robustness of control systems by incorporating knowledge that cannot be accommodated in analytical models upon which conventional control algorithms are based. A common type of knowledge based control is rule based control, for which the control actions are described in terms of *if –then* rules. The principle of designing a fuzzy controller is to integrate an empirical knowledge and operator experience into the controllers by using fuzzy sets and fuzzy rules. The theory was developed by Zadeh [73] and then Lee [74] explored its applications in control. Much of the expert's knowledge contains linguistic terms such as small, negative, positive etc. which can be represented by fuzzy sets. Using fuzzy sets and fuzzy operations it is possible to design a fuzzy reasoning system, which can act as a controller. The control strategies are stored in the form of if-then rules in a rule base structure. The rules present an approximate static mapping from inputs to outputs i.e. control actions and are determined by using expert knowledge of the process.

Fuzzy logic controllers have been useful when applied to control uncertain nonlinear systems. Fuzzy reasoning builds the understanding of imprecision into the process which could be either parametric uncertainty, unmodelled dynamics or imprecise measurement values hence can provide the ability to control a system in uncertainty or unknown environments, which is the most important requirement of an intelligent controller. Fuzzy logic control is knowledge based system that derives control actions based on input-output relationships; therefore, estimation of the system parameters is not required. Fuzzy control can model complex nonlinear functions and derive smooth control action for uncertain system behavior.

However, fuzzy control requires some qualitative description of the rules with which human operator can control a process. For many complex processes a high level of precision is not possible or even necessary in order to provide acceptable control. If the initially chosen control parameters such as membership functions and rule base structures are not satisfactory in terms of closed loop performance, then it is necessary to use "trial and error" philosophy, which may not always convenient. Although fuzzy control strategies suffer from some limitations, they can produce robust control design for real time system in the presence of parametric uncertainties.

2.2.3 Neural Control

Neural networks have shown great promise in solving nonlinear control problems because of their universal approximation capability. This powerful property has inspired the development of many neural network based controllers without significant prior knowledge of the system dynamics. Artificial Neural networks are based on the attempt to mimic the brains operations in a particular way with a move away from hard, exact mathematical calculations towards generalizing fuzzy computations[75]. The brain's powerful thinking, remembering and problem solving capabilities have inspired many scientists to attempt computers modeling of its operations. Artificial neural networks, as models of specific biological structures, have the advantages of distributed information processing and the inherent potential for parallel computations. An Artificial neural network consists of many interconnected identical simple processing units, called neurons, called neurons or nodes, which form the layered configurations. An individual neuron aggregates its weighted inputs and yields an output through a nonlinear activation function with a threshold. There are three types of synaptic interconnections, intra-layer, inter-layer and recurrent connections, in artificial neural networks. The recurrent connections provide self feedback links to the neurons.

Few important advantages of using neural control for controlling nonlinear control systems are: the dynamics of the controlled system does not need to be completely known for the design of the controllers or for the modeling of the system, the potential of online learning is very powerful feature of controlling any process in real time, in addition neural networks have the ability for adaptation and interpolation as well as ability of parallel computation and an universal approximation capability, which altogether makes them an attractive and useful technique for solving various nonlinear control problems. They can be trained to approximate any function sufficiently well. Conversely to such an attractive characteristics, the applications of neural control as element of real time control systems could be very limited due to large number of iterations over the desired mapping are required before the network adequately reproduces the required responses.

2.2.4 Neuro-Fuzzy Control

Fuzzy logic controllers have several important benefits in that they do not require a complete analytical model of a dynamic system. They provide knowledge based heuristic controllers for complex systems, and they can be analytically validated. However they are not well suited to learning. This means that fuzzy control can not meet the goals of adaptation to changes in system dynamics or to unmodelled dynamic characteristics; they can't gain increased performance through learning. On the other hand artificial neural networks have been successfully used to model and approximate various nonlinear relationships and systems. Neural networks can be trained to learn the mapping between the input and the output domains based on observations without requiring knowledge of the structure of the underlying systems. They can exploit the inherent parallelism associated with fuzzy algorithms because of the lack of dependencies on control rules. Once the network is trained it can process the rules in parallel. They have shown to posses the ability to adapt to dynamic environment changes through continuous training. The application of knowledge based control techniques for flight control [76,77] has indicated that techniques like neuro-fuzzy control can provide appropriate tools for adaptive nonlinear identification and control.

In the conventional fuzzy design, membership functions of the fuzzy sets are tuned, as defined in the input and output universe of discourse by trial and error. This drawback can be eliminated with neuro-fuzzy networks. Due to supervised learning methods it is possible to optimize the antecedent and consequent parts of the linguistic rule base fuzzy systems. The neuro-fuzzy systems are universal approximators of any nonlinear functions. There is no need of trial and error procedure to tune the control parameters of the fuzzy controller, as self learning inherently exists. This becomes an obvious advantage when neuro-fuzzy controls are used for real time systems, as number of iterations can be reduced by large. These systems can have high learning speed and be able to process the rules in parallel. By combining fuzzy logic and neural network the controller becomes more robust to imprecise information and external disturbances and an improvement in performance can be guaranteed. However a major drawback is the design complexity.

2.2.5 Evolutionary Computation

Evolutionary computation is a generic term for computational methods that use models of biological evolutionary processes for the solution of complex engineering problems. The techniques of evolutionary computation have in common the emulation of the natural evolution of individual structures through process inspired from the natural selection and reproduction. These processes depend on the fitness of the individuals to survive and reproduce in a hostile environment. Evolution can be viewed as optimization process that can be emulated by a computer. Evolutionary computation is essentially a stochastic search technique with remarkable abilities for searching for global solutions.

By analogy, in evolutionary computation, solutions that maximize some measure of fitness will have high probability of participating in the reproduction process for new solutions and it is likely that these solutions are better than the previous ones. This is a fundamental premise in evolutionary computation. Solution of an optimization problem evolve by following the well know Darwinian principles of "survival of fittest". Stochastic methods of optimization are computer intensive but in recent past impressive progress has been observed in computational technology. Ready availability of extremely fast and powerful computes has made these techniques very attractive. One of the ascending techniques of intelligent control is the fusion of fuzzy and neural control with evolutionary computation.

The greatest advantage of the evolutionary computation comes from the ability to address problems for which there are no human experts. Although human expertise should be used when it is available, it often proves less than adequate for automating problem solving routines. It can be also of great help while adapting the solutions for changing circumstances. The ability to adapt on the fly to changing circumstance is of critical importance to practical real time control problems. Most classic optimization techniques require appropriate settings of exogenous variable, which is true for evolutionary methods too but it is possible that evolutionary process itself optimize to these parameters as a part of the search for optimal solutions[78,79].

Evolutionary algorithms are a subset of evolutionary computation and belong to the generic fields of the simulated evolution and artificial life. The search for an optimum solution is based on the natural processes of biological evolution and is accomplished in parallel manner in the parameter search space. The terminology used in evolutionary computation is familiar, candidate solution of an optimization problem are termed as *individuals*. The *population* of solutions evolves in accordance with the laws of natural selection. After initialization, the population undergoes *selection, recombination and mutation* repeatedly until some termination condition is satisfied. Each iteration is termed as generation, while individual that undergo recombination and mutation are named *parents* that yield *offsprings*. The most common types of the evolutionary algorithms are Genetic Algorithm, Evolutionary strategies, Evolutionary Programming, simulated annealing etc.

The difference between Genetic Algorithms, Evolutionary strategies and Evolutionary programming lays in the operations that candidate solutions are subject to, during the evolutionary procedure. Due to significant advantages of evolutionary methods in terms of achieving robust adaptive performance for the real time system, I have used genetic algorithm with fuzzy logic to carry our present research work.

These methods are discussed in detail in the later chapters along with their applications to real time control problems.

2.3 Design of Fuzzy Systems

This section deals with the brief description of the foundation and techniques developed and used by the researchers for design, development, simulation and development of fuzzy logic based real time control of non linear systems.

The difficult task of modeling and simulating complex real-world systems for control systems development, especially when implementation issues are considered, is well documented. Even if a relatively accurate model of a dynamic system can be developed, it is often too complex to use in controller development, especially for many conventional control design procedures that require restrictive assumptions for the plant (e.g., linearity). It is for this reason that in practice conventional controllers are often developed via simple models of the plant behavior that satisfy the necessary assumptions, and via the ad hoc tuning of relatively simple linear or nonlinear controllers. Regardless, it is well understood that heuristics enter the conventional control design process as long as we are concerned with the actual implementation of the control system. It must be acknowledged that conventional control engineering approaches that use appropriate heuristics to tune the design have been relatively successful. We can have questions like how much of the success can be attributed to the use of the mathematical model and conventional control design approach, and how much should be attributed to the clever heuristic tuning that the control engineer uses upon implementation? And if we exploit the use of heuristic information throughout the entire design process, can we obtain higher performance of control systems?

Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system. In this section, a philosophy of how to approach the design of fuzzy controllers is discussed.

The fuzzy controller block diagram is given in Figure 2.1, where we show a fuzzy controller embedded in a closed-loop control system. The plant outputs are denoted by $y(t)$, its inputs are denoted by $u(t)$, and the reference input to the fuzzy controller is denoted by $r(t)$.

The fuzzy controller has four main components:

(1) The "rule-base" that holds the knowledge, in the form of a set of rules, of how best to control the system,

(2) The inference mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be,

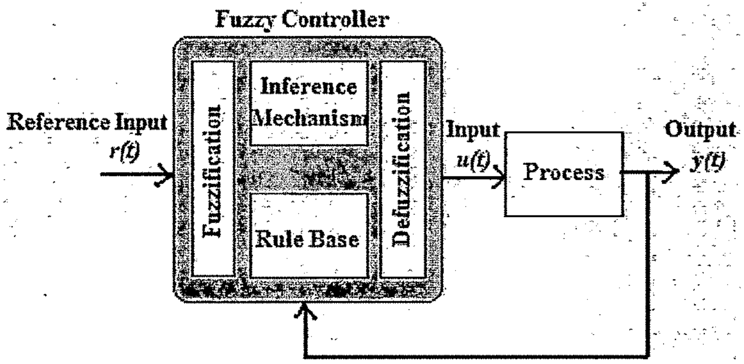

Figure 2.1 : Fuzzy Controller

(3) The fuzzification interface simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base, and

(4) The defuzzification interface converts the conclusions reached by the inference mechanism into the inputs to the plant.

Basically, the fuzzy controller is an artificial decision maker that operates in a closed-loop system in real time. It gathers plant output data $y(t)$, compares it to the reference input $r(t)$, and then decides on what the plant input $u(t)$ should be, to ensure that the performance objectives are met. To design the fuzzy controller, the control engineer must gather information on how the artificial decision maker should act in the closed-loop system. Sometimes this information can come from a human decision maker who performs the control task, while at other times the control engineer has to understand the plant dynamics and write down a set of rules about how to control the system without outside help. These "rules" basically say, "If the plant output and reference input are behaving in a certain manner, then the plant input should have some value." A whole set of such "If-Then" rules is loaded into the rule-base, and an inference strategy is chosen, then the system is ready to be tested to see if the closed-loop specifications are met.

2.3.1 Fuzzy Controller Design

Fuzzy control system design essentially amounts to (1) choosing the fuzzy controller inputs and outputs, (2) choosing the preprocessing that is needed for the controller inputs and possibly post processing that is needed for the outputs, and (3) designing each of the four components of the fuzzy controller shown in Figure 2.1. There are standard choices for the fuzzification and defuzzification interfaces. Moreover, most often the designer settles on an inference mechanism and may use this for many different processes; hence, the major area of focus is the design of the rule-base. The rule-base is constructed so that it represents a human expert "in-the-loop." Hence, the information, which is loaded into the rule-base normally come from an actual human expert, who has spent long time learning how best to control the given process. In situations, where there is no such human expert, the control engineer has to study the plant dynamics and write down the set of control rules that makes sense. As an example, in the cruise control problem it is clear that anyone who has experience driving a car can practice regulating the speed about a desired set-point and load this information into a rule-base. For instance, one rule that a human driver may use is "If the speed is lower than the set-point, then press down further on the accelerator pedal." A rule that would represent even more detailed information about how to regulate the speed would be "If the speed is lower than the set-point AND the speed is approaching the set-point very fast, then release the accelerator pedal by a small amount." This second rule characterizes our knowledge about how to make sure that we do not overshoot our desired goal (the set-point speed). Generally speaking, if control engineer is able to load very detailed expertise into the rule-base, then it enhances the chances of obtaining better performance.

2.3.2 Adaptive Robust Fuzzy Control of Nonlinear Systems

Uncertainties are inevitable in dynamical systems, and they may arise from errors in system modeling, parameter variations, unknown physical phenomena and working environments. In addition to the classical feedback control theory, adaptive control and robust control are effective techniques to treat these uncertainties. Adaptive control, by online tuning the parameters (of either the plant or the controller—corresponding to indirect, or direct adaptive control), can deal with large uncertainties, but generally, suffers from the disadvantage of being able to achieve only *asymptotical* convergence of the tracking error to zero. The online computation burden to update the parameters is also very high in case of real time systems. In robust control designs, on the other hand, a fixed control law based on *a priori* information on the uncertainties is designed to compensate for their effects, and *exponential* convergence of the tracking error to a small ball centered at the origin is obtained. But if the uncertainties are larger than the assumed bounds, no stability or performance is guaranteed.

In recent past analytical studies of nonlinear control, using universal approximators such as fuzzy logic system (FLS) [9],[59],[73],[74] is used to approximate the unknown functions involved in the control design. In the class of approximators, which are linear in parameters, FLS is much closer in spirit to human thinking and natural language, and preferred by control engineering practitioners. The problem of controlling nonlinear systems expressed in the canonical form to follow a reference trajectory in the presence of uncertainties. Fuzzy logic systems are used to approximate the unknown dynamics of the system. This problem has been extensively investigated; however, most results reported in the literature suffer from at least one of the following drawbacks:

- lack of robustness to unmodelled dynamics and/or external perturbations due to only *asymptotical* convergence of the tracking error to a residual set of the origin is achieved
- requirement of the knowledge on the nonlinear systems for controller implementation, such as bounding functions on f(x) and b(x);
- requirement of the bound on the norm of the optimal parameter vector of the universal approximators, or a compact set to which the optimal parameter vector of the universal approximators belongs;
- Heavy online computation burden due to updating the parameters of the universal approximators.

By combining advantages of FLSs, adaptive, and robust control techniques, an adaptive robust fuzzy control capable of achieving *exponential* convergence of the tracking error to a small ball of the origin, whose radius can be made arbitrarily small by properly choosing some design parameters. From a practical point of view, exponential tracking is more desirable for its robustness against unmodelled dynamics and/or external perturbations. To implement the controller only the knowledge of b[a constant lower bound on b(x)], and a nominal parameter vector of the FLS are required. The nominal parameter vector may be obtained either from *a priori* knowledge of the plant or through offline training, and may be set zero if no *a priori* knowledge of the plant is available nor offline training is done. The online computation burden is also reduced since only uncertainty bounds are adaptively tuned online.

2.3.3 Integrating membership functions and fuzzy rule sets from multiple knowledge sources

Expert systems have been successfully applied to many fields and have shown excellent performance. Knowledge-base construction remains, however, one of the major costs in building an expert system even though many tools have been developed to help with knowledge acquisition. Building a knowledge-based system usually entails constructing new knowledge bases from scratch. The cost of the effort is high and will become prohibitive as we attempt to build larger and larger systems. Reusing and integrating available knowledge from a variety of sources, such as domain experts, historical documentary evidence, current records, or existing knowledge bases, thus plays an important role in building effective knowledge-based systems.

Especially for complex application problems, related domain knowledge is usually distributed among multiple sites, and no single site may have complete domain knowledge. The use of knowledge integrated from multiple knowledge sources is thus especially important to ensure comprehensive coverage.

Most knowledge sources or actual instances in real-world applications contain fuzzy or ambiguous information. Especially in domains such as medical or control domains, the boundaries of a piece of information used may not be clearly defined. Expressions of the domain knowledge using fuzzy descriptions are thus seen more and more frequently. Several researchers have recently investigated automatic generation of fuzzy classification rules and fuzzy membership functions using evolutionary algorithms [80,81]. These methods can be categorized into the following four types:

- learning fuzzy membership functions with fixed fuzzy rules;
- learning fuzzy rules with fixed fuzzy membership functions [81] ;
- learning fuzzy rules and membership functions in stages (i.e., first evolving good fuzzy rule sets using fixed membership functions, then tuning membership functions using the derived fuzzy rule sets);
- learning fuzzy rules and membership functions simultaneously [80].

2.3.4 Analysis and Design of Fuzzy Controller Based on Observer

Fuzzy logic control techniques suffer from following limitations...

- the design of the fuzzy logic controller is difficult because no theoretical basis is available
- the performance of the fuzzy logic controller can be inconsistent because the fuzzy logic control depends mainly on the individual operators' experience.

Therefore, despite the fact that much progress has been made in successfully applying fuzzy logic control to industrial control systems, it has become evident that many basic issues remain to be further addressed.

Stability analysis and systematic design are certainly among the most important issues to fuzzy control systems. Recently, the issue of stability of fuzzy control systems has been considered extensively in nonlinear stability frameworks. The stability analysis and robust fuzzy controllers design methods for a class of uncertain nonlinear systems was discussed in [82], which only considered the uncertainty of fuzzy model without considering the unobservable problem of states of systems. The stabilization of a feedback system containing a fuzzy controller and a fuzzy observer for Fuzzy systems for multi-input and multi-output linear systems was addressed in paper [83, 84], which only took into account of the state unobservable problem of the system without considering the uncertainty of fuzzy model. It is well known that the observer design and robust control are very important problems in control systems; the fuzzy observer design is hardly addressed. A very key problem is that the stability of the whole system must be guaranteed.

2.3.5 Fuzzy Observer-Based Control of Nonlinear Systems

In recent past there had been rapid growth in using Takagi-Sugeno [85] fuzzy models to approximate nonlinear systems. These models consist of fuzzy *If...Then* rules with linguistic terms in antecedents, and analytic dynamical equations in the consequents. There has been a great deal of effort in trying to find conditions for stability of these types of control systems. The approach proposed by Sugeno and Tanaka [86,87], uses a common quadratic Lyapunov function. The most important draw back of this method was that finding a common matrix that satisfies all

Lyapunov inequalities is not easy. With the emergence of new optimization methods [88] that can solve Linear Matrix Inequalities (LMI's) in polynomial time, this problem has been solved.

Among many results in fuzzy control, Takagi and Sugeno introduced the now so-called T-S Fuzzy model. The T-S fuzzy systems have certain relations to conventional linear models, and therefore many classical control theory methods can be used for the analysis and synthesis of fuzzy systems. Many control strategies are achieved via state feedback control like ...

- Thau-Luenberger Observers for TS Fuzzy systems
- Sugeno-type Fuzzy Observers etc.

2.3.6 Design and analysis of a fuzzy logic controller

People very often make decisions in their daily lives based on qualitative information. Zadeh's fuzzy sets theory was thus proposed to enable people to describe d formulate the linguistic mental 'models apparent in daily life behaviour. Mamdani and his coworkers [89,90,91] were pioneers in applying fuzzy techniques to process control. Their results, as well as those of many other researchers, have demonstrated the potential value of the fuzzy logic control system on simple process dynamics. Practical fuzzy logic control applications for real time applications have also been reported in turn. A comprehensive review of the classical design and implementation of the fuzzy logic controller can be found in [92], more advanced design techniques have also been reported in the literature, such as the adaptive fuzzy logic control [93,94,95,96].

The superior performance of fuzzy logic controllers reported in the literature usually conjectured to have its origin in their switching nature, where the magnitude of the rate of change in controller output is greater for larger error and small for plant output close to the set point. The nonlinearity and complexity of fuzzy control responses, however, causes the analysis and systematic tuning of fuzzy logic controllers to remain a difficult research problem.

The relations between fuzzy and conventional PID controllers have been studied by various researchers and found that some specialized fuzzy logic controllers [97], have been proven to be equivalent to a nonlinear two-mode PI controller with state-dependent variable controller gains. A uniformly distributed triangular family of membership functions is applied for linguistic members of each fuzzy variable, and all control laws are expressed in a simple form in such a controller. Such a simplified design makes the number of undetermined fuzzy members, the sole design variable, and the input/output scalars along the most relevant tuning variables in the controller. Such an FLC, no matter the number of fuzzy members it uses, have proved to be functionally equivalent to a non-fuzzy nonlinear PI or PD controller with state-dependent controller gains and a value of integral (derivative) time dependent on input scalars only. Several extensions of the basic FLC, including the FLC with dual control laws (PI and PD forms) by including one switching factor, and the FLC with varying gains according to the discrepancy between process variables and set point, are thus proposed to enhance control performance. The superior results obtained employing various extensions to the FLC are illustrated by several numerical examples. A neutralization process is also employed to demonstrate the potential applicability of the fuzzy logic control method on real time control problems. To understand the various steps involved in proper design of fuzzy controller, let us discuss the design of simple fuzzy controller with reference to Inverted Pendulum.

2.3.6.1 Design of Simple Fuzzy Controller

Let us discuss each of the components of the fuzzy controller for a simple problem of balancing an inverted pendulum on a cart, as shown in Figure 2.2. Here, y denotes the angle that the pendulum makes with the vertical place, in radians, l is the half-pendulum length, in meters, and u is the force input that moves the cart, in Newtons. r is used to denote the desired angular position of the pendulum.

The goal is to balance the pendulum in the upright position (i.e., $r = 0$), when it initially starts with some nonzero angle off the vertical (i.e., $y \neq 0$). This is a very simple and academic nonlinear control problem, and many good techniques already exist for its solution. Indeed, for this standard configuration, a simple PID controller works well even in implementation. Later, I am using the same problem to discuss much more general issues in fuzzy control system design for more challenging applications also.

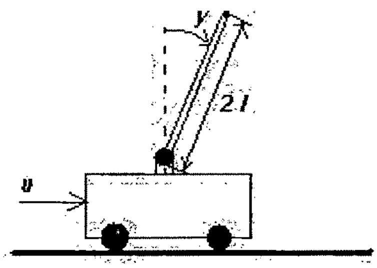

Figure 2.2 Inverted Pendulum on a Cart.

2.3.6.2 *Choosing Fuzzy Controller Inputs and Outputs*

Consider a human-in-the-loop whose responsibility is to control the pendulum, as shown in Figure 2.3. The fuzzy controller is to be designed to automate how a human expert, who is successful at this task, would control the system. First, the expert tells us (the designers of the fuzzy controller) what information she or he will use as inputs to the decision-making process. Suppose that for the inverted pendulum, the expert (this could be you!) says that she or he will use $e(t) = r(t) - y(t)$ *and* $d/dt\,(e(t))$ as the variables on which to base decisions. Certainly, there are many other choices (e.g., the integral of the error e could also be used) but this choice makes good intuitive sense. Next, we must identify the controlled variable. For the inverted pendulum, we are allowed to control only the force that moves the cart, so the choice here is simple.

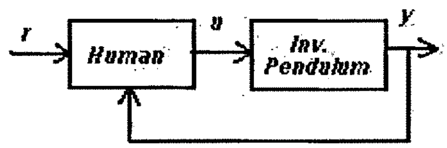

Figure 2.3 Human Controlled Inverted Pendulum on a Cart

For more complex applications, the choice of the inputs to the controller and outputs of the controller (inputs to the plant) can be more difficult. Essentially, we have to make sure that the controller will have the proper information available to be able to make good decisions and have proper control inputs to be able to steer the system in the directions needed to be able to achieve high-performance operation. Practically speaking, access to information and the ability to effectively control the system often cost money. If the designer believes that proper information is not available for making control decisions, he or she may have to invest in another sensor that can provide a measurement of another system variable. Alternatively, the designer may implement some filtering or other processing of the plant outputs. In addition, if the designer determines that the current actuators will not allow for the precise control of the process, he or she may need to invest in designing and implementing an actuator that can properly affect the process. Hence, while in some academic problems the plant inputs and outputs may be readily available; in many practical situations we may have some flexibility in their choice. These choices affect what information is available for making on-line decisions about the control of a process and hence affect how we design a fuzzy controller. Once the fuzzy controller inputs and outputs are chosen, we must determine what the reference inputs are. For the inverted pendulum, the choice of the reference input $r = 0$ is clear. In some situations, however, we may want to choose r as some nonzero constant to balance the pendulum in the off-vertical position. To do this, the controller must maintain the cart at a constant acceleration so that the pendulum will not fall.

After all the inputs and outputs are defined for the fuzzy controller, we can specify the fuzzy control system. The fuzzy control system for the inverted pendulum, with our choice of inputs and outputs, is shown in Figure 2.4.

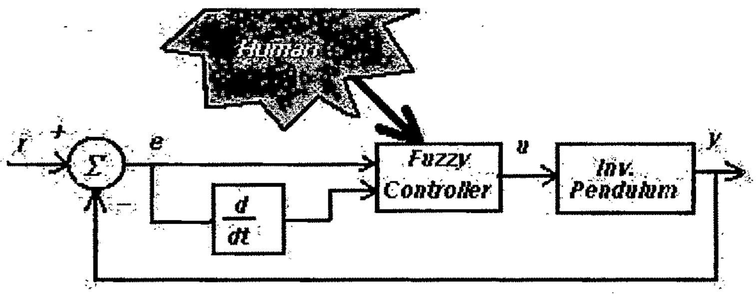

Figure 2.4 Fuzzy Controller for Inverted Pendulum on a Cart

Now, *within this framework* we have to obtain a description of how to control the process. We see then that the choice of the inputs and outputs of the controller places certain constraints on the remainder of the fuzzy control design process. If the proper information is not provided to the fuzzy controller, there will be little hope for being able to design a good rule-base or inference mechanism. Moreover, even if the proper information is available to make control decisions, this will be of little use if the controller is not able to properly affect the process variables via the process inputs. It must be understood that the choice of the controller inputs and outputs is a fundamentally important part of the control design process.

2.3.6.3 Putting Control Knowledge into Rule-Bases

Suppose that the human expert shown in Figure 2.3 provides a description of how best to control the plant in some natural language. We have to interpret this "linguistic" description and load them into the fuzzy controller, as indicated by the arrow in Figure 2.4.

Linguistic Descriptions

The linguistic descriptions provided by the expert are generally broken into several parts. There will be "linguistic variables" that describe each of the time varying fuzzy controller inputs and outputs. For the inverted pendulum, under consideration

"error" describes $e(t)$

"change-in-error" describes $d/dt(e(t))$

"force" describes $u(t)$

As $e(t)$ takes on a value of, for example, 0.1 at $t = 2$ ($e(2) = 0.1$), linguistic variables assume "linguistic values." That is, the values that linguistic variables take on over time change dynamically. Suppose for the pendulum example that "error," "change-in-error," and "force" take on the following values:

"neglarge"

"negsmall"

"zero"

"possmall"

"poslarge"

Where, "negsmall" is used as an abbreviation for "negative small in size" and so on for the other variables. Such abbreviations help keep the linguistic descriptions short yet precise. For an even shorter description we can use integers:

"-2" to represent "neglarge"

"-1" to represent "negsmall"

"0" to represent "zero"

"1" to represent "possmall"

"2" to represent "poslarge"

This is normally preferred choice for the linguistic values since the descriptions are short and nicely represent that the variable we are concerned with has a numeric quality. Here, associating "-1" has nothing to do with any particular number of radians of error; the use of the numbers for linguistic descriptions simply quantifies the sign of the error in the usual way and indicates the size in relation to the other linguistic values. Use of this type of linguistic value quite convenient and hence its given the special name, "linguistic-numeric value." The linguistic variables and values provide a language for the expert to express their ideas about the control decision-making process in the context of the framework established by our choice of fuzzy controller inputs and outputs. Recall that for the inverted pendulum $r = 0$ *and* $e = r - y$ and hence,

$$e = -y \quad and \quad {}^{d}\!/_{dt}(e) = -{}^{d}\!/_{dt}(y) \quad since \quad {}^{d}\!/_{dt}(r) = 0 \ .$$

Let us first, quantify certain dynamic behaviors with linguistics and later we will see how to quantify knowledge about how to control the pendulum using linguistic descriptions.
For the inverted pendulum shown in Figure 2.2 each of the following statements quantifies a different configuration of the pendulum

- The statement "error is poslarge" can represent the situation where the pendulum is at a significant angle to the *left* of the vertical.
- The statement "error is negsmall" can represent the situation where the pendulum is just slightly to the right of the vertical, but not too close to the vertical to justify quantifying it as "zero" and not too far away to justify quantifying it as "neglarge."
- The statement "error is zero" can represent the situation where the pendulum is very near the vertical position (a linguistic quantification is not precise, hence we are willing to

accept any value of the error around $e(t) = 0$ as being quantified linguistically by "zero" since this can be considered a better quantification than "possmall" or "negsmall").

- The statement "error is poslarge **and** change-in-error is possmall" can represent the situation where the pendulum is to the left of the vertical and, since $d/_{dt}(y) \prec 0$, the pendulum is moving *away* from the upright position (note that in this case the pendulum is moving counterclockwise).
- The statement "error is negsmall **and** change-in-error is possmall" can represent the situation where the pendulum is slightly to the right of the vertical and, since $d/_{dt}(y) \prec 0$, the pendulum is moving *toward* the upright position (note that in this case the pendulum is also moving counterclockwise).

In order to quantify the dynamics of the process we need to have a good understanding of the physics of the underlying process we are trying to control. While for the pendulum problem, the task of coming to a good understanding of the dynamics is relatively easy; this is not the case for many real time physical processes. Quantifying the process dynamics with linguistics is not always easy, and certainly a better understanding of the process dynamics generally leads to a better linguistic quantification. Often, this leads to a better fuzzy controller *provided* that you can adequately measure the system dynamics so that the fuzzy controller can make the right decisions at the proper time.

Rules

Next, the above linguistic quantification is used to specify a set of rules (a rule-base) that captures the expert's knowledge about how to control the plant. In particular, for the inverted pendulum in the three positions shown in Figure 2.5, following rules

1. *If error is neglarge **and** change-in-error is neglarge **Then** force is poslarge* - This rule quantifies the situation in Figure 2.5(a) where the pendulum has a large positive angle and is moving clockwise; hence it is clear that we should apply a strong positive force (to the right) so that we can try to start the pendulum moving in the proper direction.

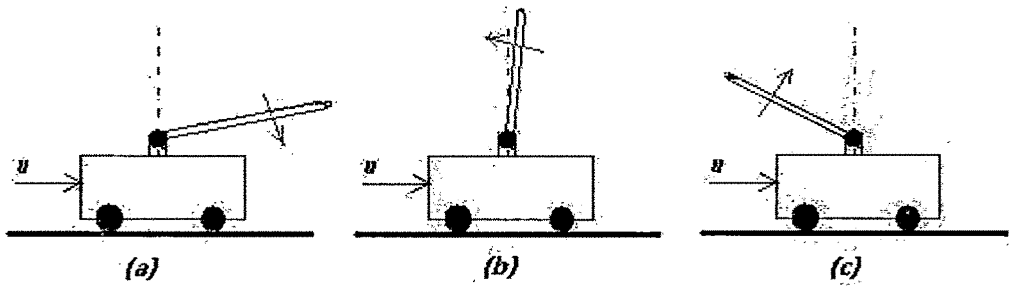

Figure 2.5 : Inverted Pendulum in different positions

2. ***If** error is zero **and** change-in-error is possmall **Then** force is negsmall* - This rule quantifies the situation in Figure 2.5(b) where the pendulum has nearly a zero angle with the vertical (a linguistic quantification of zero does not imply that $e(t) = 0$ exactly) and is moving counterclockwise; hence we should apply a small negative force (to the left) to counteract the movement so that it moves toward zero (a positive force could result in the pendulum overshooting the desired position).
3. ***If** error is poslarge **and** change-in-error is negsmall **Then** force is negsmall* - This rule quantifies the situation in Figure 2.5(c) where the pendulum is far to the left of the vertical and is moving clockwise; hence we should apply a small negative force (to the left) to assist the movement, but not a big one since the pendulum is already moving in the proper direction.

Each of the above rules is a "linguistic rule" since it is formed solely from linguistic variables and values. Since linguistic values are not precise representations of the underlying quantities that they describe, linguistic rules are not precise either. They are simply abstract ideas about how to achieve good control that could mean somewhat different things to different people. They are, however, at a level of abstraction that humans are often comfortable with in terms of specifying how to control a process. The general form of the linguistic rules listed above is ***If** premise **Then** consequent.*

The number of fuzzy controller inputs and outputs places an upper limit on the number of elements in the premises and consequents. Here, there does not need to be a premise (consequent) term for each input (output) in each rule, although often there is.

Rule-Bases

Using the same approach, as discussed earlier one can continue to write down rules for the pendulum problem for all possible cases. But since we specify a finite number of linguistic variables and linguistic values, there is only a finite number of possible rules.

For the problem under consideration, having two inputs and five linguistic values, there are at the most $5^2 = 25$ possible rules (all possible combinations of premise linguistic values for two inputs). A tabular representation of one possible set of rules for the inverted pendulum is shown in Table 2.1. The table lists the linguistic-numeric consequents of the rules, and the left column and top row of the table contain the linguistic-numeric premise terms. Then, for instance, the $(2,-1)$ position (where the "2" represents the row having "2" for a numeric-linguistic value and the "-1" represents the column having "-1" for a numeric-linguistic value) has a -1 ("negsmall") in the body of the table and represents the rule.

Table 2.1 : Rule Table for Inverted Pendulum

force (u)		*Change in error – d/dt (e)*				
		-2	-1	0	1	2
error (e)	-2	2	2	2	1	0
	-1	2	2	1	0	-1
	0	2	1	0	-1	-2
	1	1	0	-1	-2	-2
	2	0	-1	-2	-2	-2

***If** error is poslarge **and** change-in-error is negsmall **Then** force is negsmall,* which is rule 3 above. Table 2.1 represents abstract knowledge that the expert has about how to control the pendulum given the error and its derivative as inputs.

Its not that these are the rules only valid for given inverted pendulum, this rule bas can differ with different individuals. Here one can notice the diagonal of zeros and viewing the body of the table as a matrix we see that it has certain symmetry to it. This symmetry that emerges, when the rules are tabulated, is no an accident but is actually a representation of abstract knowledge about how to control the pendulum; it arises due to a symmetry in the system's dynamics. This type of similar patterns also found when constructing rule-bases for more challenging applications, and we also need to exploit this symmetry in implementing fuzzy controllers.

2.3.6.4 Fuzzy Quantification of Knowledge

Up till now, the knowledge of the human expert about how to control the plant is quantified in an abstract way, now; we have to use fuzzy logic to fully quantify the meaning of linguistic descriptions so that we may automate the control rules specified by the expert, in the fuzzy controller.

Membership Functions

First, we need to quantify the meaning of the linguistic values using "membership functions." Consider, for example, Figure 2.6. This is a plot of a function μ versus $e(t)$ that takes on special meaning. The function μ quantifies the *certainty* that $e(t)$ can be classified linguistically as "possmall." To understand the way that a membership function works, it is best to perform a case analysis where we show how to interpret it for various values of $e(t)$:

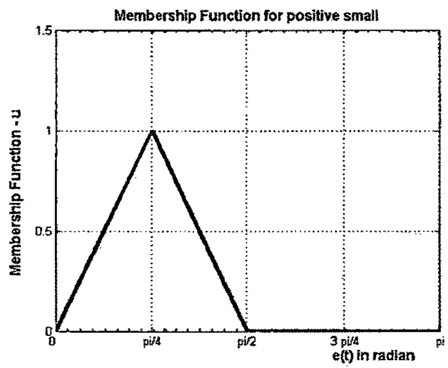

Figure 2.6 : Membership Function for possmall

- If $e(t) = -\pi/2$ then $\mu(-\pi/2) = 0$, indicating that $e(t) = -\pi/2$ is *not* "possmall."
- If $e(t) = \pi/8$ then $\mu(\pi/8) = 0.5$, indicating that we are halfway certain that $e(t) = \pi/8$ is "possmall"
- If $e(t) = \pi/4$ then $\mu(\pi/4) = 1.0$, indicating that $e(t) = \pi/4$ is what meant as "possmall."
- If $e(t) = \pi$ then $\mu(\pi) = 0$, indicating that $e(t) = \pi$ is not "possmall", actually, it is "poslarge".

The membership function quantifies, in a continuous manner, whether values of $e(t)$ belong to the set of values that are "possmall," and hence it quantifies the meaning of the linguistic statement "error is possmall". The membership function shown in Figure 2.6 is not the only one possible definition of the meaning of "error is possmall"; it can also be a bell-shaped function, a trapezoid, or many others. Few of possible shapes of memberships functions are shown in Figure 2.7.

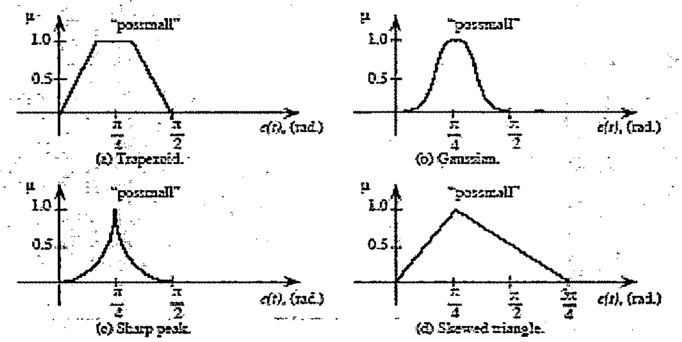

Figure 2.7: Different possible shapes of Membership Functions

For some application someone may be able to argue that we are absolutely certain that any value of $e(t)$ near $\pi/4$ is still "possmall" and only when you get sufficiently far from $\pi/4$ we lose our confidence that it is "possmall." One way to characterize this understanding of the meaning of "possmall" is via the trapezoid-shaped membership function in Figure 2.7(a). For other applications one may think of membership in the set of "possmall" values as being dictated by the Gaussian-shaped membership function as shown in Figure 2.7(b). For still other applications one may not readily accept values far away from $\pi/4$ as being "possmall," in that case the membership function in Figure 2.7(c) can be used to represent them. Figure 2.7 (a) to (d) show membership functions, where symmetric characterizations of the meaning of linguistic values is considered but selection of membership function is not restricted to these. Figure 2.7(d) represents belief that as $e(t)$ mover to the left of $\pi/4$, one is very quick to reduce their confidence that it is "possmall", but if one move to right of $\pi/4$, confidence that $e(t)$ is "possmall" diminishes at slower rate. In summary, depending on the application and the designer (expert), many different choices of membership functions are possible.

Now, on the same line we need to specify the membership functions for all 15 linguistic values – five for *e(t)*, five for *d/dt(e)* and five for *u(t)* for inverted pendulum problem under consideration. Figure 2.8 displays the same for one choice of membership functions, which can be any function, as we have discussed earlier.

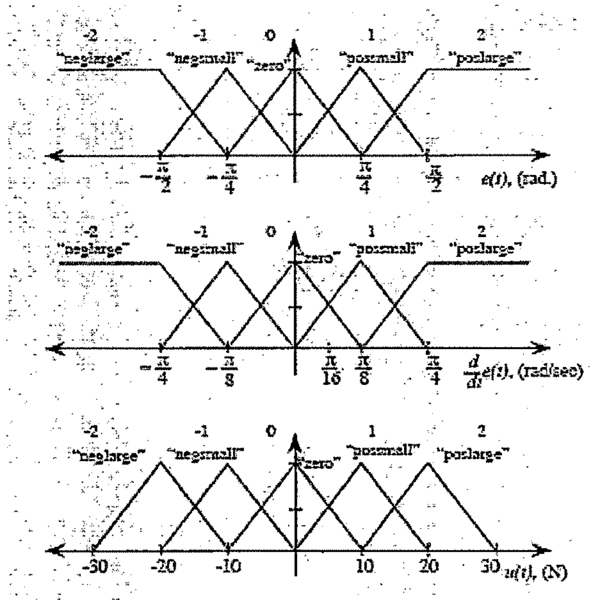

Figure 2.8: Membership Functions for Inverted Pendulum on a Cart.

The membership functions for the inputs *e*(*t*) and *d/dt*(*e*(*t*)), the outermost membership functions "saturate" at a value of one. This is to mark sense as at some point the human expert would just group all large values together in a linguistic description such as "poslarge." The membership functions at the outermost edges appropriately characterize this phenomenon since they characterize "greater than" (for the right side) and "less than" (for the left side).

For the output *u*, the membership functions at the outermost edges cannot be saturated for the properly defined fuzzy system. The basic reason for this is that in decision-making processes, we seek to take actions that specify an exact value for the process input. We can not indicate to a process actuator that "any value bigger than, say, 10, is acceptable."

The rule-base of the fuzzy controller holds the linguistic variables, linguistic values, their associated membership functions, and the set of all linguistic rules as shown in Table 2.1, this was about the description of the simple inverted pendulum. Now, lets move to the fuzzification process.

Fuzzification

It is actually the case that for most fuzzy controllers the fuzzification block in Figure 2.1 can be ignored since this process is so simple. The exact operations of the fuzzification process can be referred in [32,33,34,35]. Lets us discuss the simplest of the fuzzification process as the act of obtaining a value of an input variable (e.g., $e(t)$) and finding the numeric values of the membership function(s) that are defined for that variable.

For example, if $e(t) = \pi/4$ and $d/dt(e(t)) = \pi/16$, the fuzzification process amounts to finding the values of the input membership functions for these. In this case

$$\mu_{possmall}(e(t)) = 1$$

with all others zero and

$$\mu_{zero}\left(\frac{d}{dt}(e(t))\right) = \mu_{possmall}\left(\frac{d}{dt}(e(t))\right) = 0.5$$

So, the membership function values are as an "encoding" of the fuzzy controller numeric input values. The encoded information is then used in the fuzzy inference process that starts with "matching."

2.3.6.5 Matching: Determining Which Rules to Use

Now, let us discuss how the inference mechanism in Figure 2.1 operates. The inference process generally involves two steps:

1. The premises of all the rules are compared to the controller inputs to determine which rules apply to the current situation. This "matching" process involves determining the certainty that each rule applies, and typically rules which are more certain to apply to the current situation are recommended strongly.
2. The conclusions (what control actions to take) are determined using the rules that have been determined to apply at the current time. The conclusions are characterized with a

fuzzy set(s) that represents the certainty that the input to the plant should take on various values.

Premise Quantification via Fuzzy Logic

To perform inference, quantifying each of the rules with fuzzy logic is necessary. To do this we have to first quantify the meaning of the premises of the rules that are composed of several terms, each of which involves a fuzzy controller input. Consider Figure 2.9, where we list two terms from the premise of the rule

*If error is zero **and** change-in-error is possmall **Then** force is negsmall*

The meaning of the linguistic terms "error is zero" and "change-in-error is possmall" is already quantified via the membership functions shown in Figure 2.9. Let us now quantify the linguistic premise "error is zero **and** change-in-error is possmall." Here, focus is on how to quantify the logical "and" operation that combines the meaning of two linguistic terms.

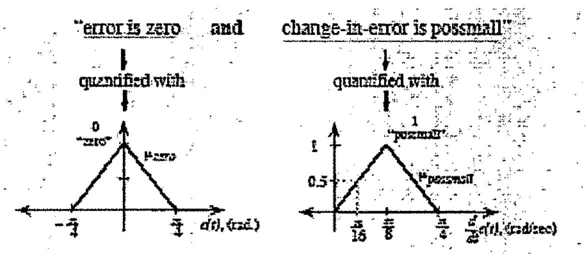

Figure 2.9: Membership Functions of e(t) and d/dt (e(t))

To see how to quantify the "and" operation, begin by assuming that

$$e(t) = \pi/8 \quad and \; \frac{d}{dt}(e(t)) = \pi/32$$

referring Figure 2.8 or Figure 2.9, we can get

$$\mu_{zero}(e(t)) = 0.5 \text{ and}$$

$$\mu_{possmall}\left(\frac{d}{dt}(e(t))\right) = 0.25$$

these values of *e*(*t*) and *d/dt*(*e*(*t*)), is the certainty of the statement

*"error is zero **and** change-in-error is possmall"*

Let us denote this certainty by $\mu_{premise}$.

There are actually several ways to define it:

- *Minimum:* Define $\mu_{premise} = \min\{0.5, 0.25\} = 0.25$, that is, using the minimum of the two membership values.
- *Product:* Define $\mu_{premise} = (0.5)(0.25) = 0.125$, that is, using the product of the two membership values.

Similarly, there will be a different premise membership function for each of the rules defined in the rule-base, and each of these will be a function of $e(t)$ and $d/dt(e(t))$ so that for a given specific values of $e(t)$ and $d/dt(e(t))$, a quantification of the certainty that each rule in the rule-base applies to the current situation is derived. It is very much important that designer defines in their mind about the situation where $e(t)$ and $d/dt(e(t))$ change dynamically over time. When this occurs the values of $\mu_{premise}\left(e(t), \frac{d}{dt}(e(t))\right)$ for each rule change, and hence the applicability of each rule in the rule-base for specifying the force input to the pendulum, changes with time.

Determining Which Rules Are On

Determining the applicability of each rule is called "matching." A rule is said to be "on at time t", if its premise membership function $\mu_{premise}\left(e(t), \frac{d}{dt}(e(t))\right) \succ 0$. Hence, the inference mechanism seeks to determine which rules are on to find out which rules are relevant to the current situation. And later the inference mechanism will seek to combine the recommendations of all the rules to come up with a single conclusion.

2.3.6.6 Inference Step: Determining Conclusions

It is very much important that how to determine which conclusions should be reached, when the rules that are on are applied to decide, what the input force should be, to the cart carrying the inverted pendulum. To find out this, first need to find the recommendations of each rule independently and later need to combine all the recommendations from all the rules to determine the force input to the cart.

Recommendation from One Rule

Consider the conclusion reached by the rule

*If error is zero **and** change-in-error is zero **Then** force is zero,*

Let us refer the same as "rule (1)." Using the minimum to represent the premise,

$$\mu_{premise}(1) = \min\{0.25, 1\} = 0.25$$

where $\mu_{premise}(1)$ represents membership of premise for rule (1), meaning that we are 0.25 certain that this rule applies to the current situation. The rule indicates that if its premise is true then the action indicated by its consequent should be taken. For rule (1) the consequent is "force is zero", here the pendulum is balanced, so no need to apply any force since this would tend to move the pendulum away from the vertical.. The membership function for the conclusion reached by rule (1), which is denoted as $\mu(1)$, is given by $\mu(1)(u) = \min\{0.25, \mu_{zero}(u)\}$.

This membership function defines the "implied fuzzy set" for rule (1) (i.e., it is the conclusion that is implied by rule (1)). The justification for the use of the minimum operator to represent the implication is that *we can be no more certain about our consequent than our premise.* The membership function $\mu(1)(u)$ is a *function* of u and that the minimum operation will generally "chop off the top" of the $\mu_{zero}(u)$ membership function to produce $\mu(1)(u)$. For different values of $e(t)$ and $d/dt(e(t))$ there will be different values of the premise certainty $\mu_{premise}\left(e(t), \frac{d}{dt}(e(t))\right)$ for rule (1) and hence different *functions* $\mu(1)(u)$ obtained.

Recommendation from another Rule

Now, consider the conclusion reached by the other rule that is on,

*If error is zero **and** change-in-error is possmall **Then** force is negsmall*

Let us refer the same as "rule (2)." Using the minimum to represent the premise,

$$\mu_{premise}(2) = \min\{0.75, 1\} = 0.75$$

It means that we are 0.75 certain that this rule applies to the current situation. It's also clear here that application of rule (2) is certain then rule (1), in current situation. For rule (2) the consequent is "force is negsmall", because for here the pendulum is perfectly balanced but is moving in the counterclockwise direction with a small velocity. The membership function for the conclusion reached by rule (2), which we denote by $\mu(2)$, is given by $\mu(2)(u) = \min\{0.75, \mu_{negsmall}(u)\}$. This membership function defines the implied fuzzy set for rule (2) (i.e., it is the conclusion that is reached by rule (2)). Once again, for different values of $e(t)$ and $d/dt(e(t))$ there will be different values of $\mu_{premise}\left(e(t), \frac{d}{dt}(e(t))\right)$ for rule (2) and hence different *functions* $\mu(2)(u)$ obtained. Rule (2) is quite certain that the control output or process input should be a small negative value. This makes sense since if the pendulum has some counterclockwise velocity then we need to apply a negative force (i.e., one to the left). As rule (2) has a premise membership function that has higher certainty than for rule (1), we see that we are more certain of the conclusion reached by rule (2).

2.3.6.7 Converting Decisions into Actions

Next, we need to carry out the defuzzification operation, which is the final component of the fuzzy controller shown in Figure 2.1. Defuzzification operates on the implied fuzzy sets produced by the inference mechanism and combines their effects to provide the "most certain" controller output. One can think of defuzzification as "decoding" the fuzzy set information produced by the inference process (i.e., the implied fuzzy sets) into numeric fuzzy controller outputs. We need one output, which we denote by "u^{crisp}," that best represents the conclusions of the fuzzy controller that are represented with the implied fuzzy sets. There are actually many approaches to defuzzification.

2.3.6.8 MATLAB Simulation of Fuzzy Controller for Inverted Pendulum

As there is no general systematic procedure for the design of fuzzy controllers that will definitely produce a high-performance fuzzy control system for a wide variety of applications, it is always better to learn about fuzzy controller design via examples. Let us continue with the inverted pendulum example to understand the various typical procedures used in the design of a fuzzy controller.

To simulate the fuzzy control system shown in Figure 2.4 it is necessary to specify a mathematical model of the inverted pendulum. Here we are not using the model for the initial design of the fuzzy controller but to accurately assess the quality of a design, we need either a model for mathematical analysis or simulation-based studies, or an experimental test bed in which to evaluate the design. Here, we will study simulation-based evaluations for design. One model for the inverted pendulum shown in Figure 2.2 is given by

$$\ddot{y} = \frac{9.8 \cdot \sin(y) + \cos(y)\left[\dfrac{-\bar{u} - 0.25 \cdot \dot{y}^2 \cdot \sin(y)}{1.5}\right]}{0.5 \cdot \left[\dfrac{4}{3} - \dfrac{1}{3} \cdot \cos^2(y)\right]} \qquad \ldots(2.1)$$

$$\dot{\bar{u}} = -100 \cdot \bar{u} + 100 \cdot u$$

The first order filter on u to produce $\bar{u}$ represents an actuator. Given this and the fuzzy controller discussed in previously, we can simulate the fuzzy control system shown in Figure 2.4. Let us define the initial condition be $y(0) = 0.1\, radians$, $\dot{y}(0) = 0$, and the initial condition for the actuator state is zero. The simulink model of the same is shown in Figure 2.10, and response of the same is shown in Figure 2.11.

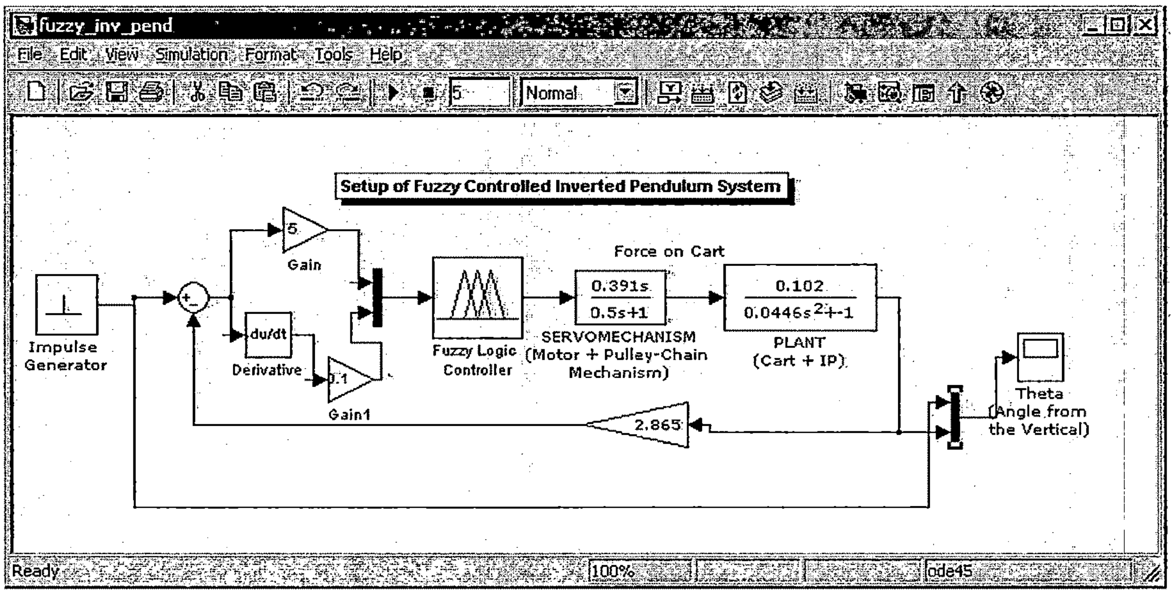

Figure 2.10 : Simulink Model of the Inverted Pendulum

2.3.7 Sliding Mode Observers for Takagi–Sugeno Fuzzy Systems

A dynamic TS [98] fuzzy system is composed of multiple local affine dynamic linear models. These local models are related to local linearization, via Taylor series expansion, of the original nonlinear system at off-equilibrium points. Thus, the local models have no equilibrium point within their regions of validity, i.e., they are off-equilibrium local models

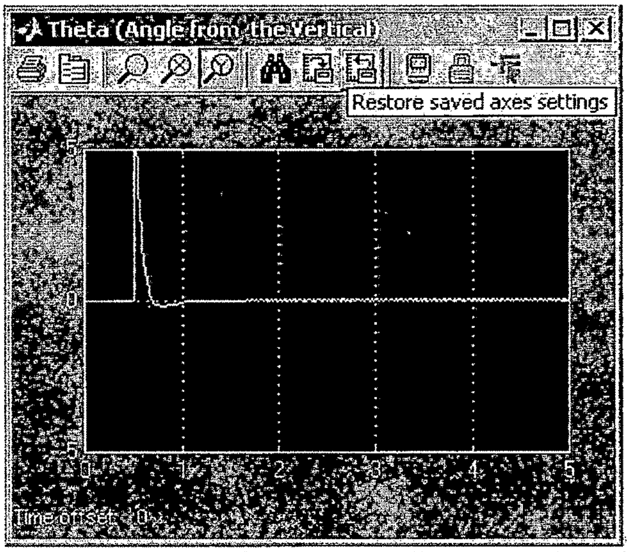

Figure 2.11 : Response of Fuzzy controlled Inverted Pendulum

In [83] it is proved that a TS fuzzy system where the local affine dynamic models are off-equilibrium local linearization leads to an arbitrary close approximation of the linear time varying (LTV) dynamic system resulting from dynamic linearization of the original nonlinear system about an arbitrary trajectory. Thus, the results concerning observers for TS fuzzy systems are also relevant to systems such as linear parameter varying (LPV) systems, piecewise linear systems, and conventional gain-scheduled systems.

TS fuzzy system is subject to observation and defines a Luenberger type of observer, which may be realized in terms of a parallel distributed compensation scheme incorporating an interpolation between local Luenberger observers. This is the type of nonlinear observer that has received most attention in the fuzzy control literature, but all results reported assume no matched/unmatched uncertainties.

A sliding mode fuzzy observer that is related to the so called min–max observer described by Zak and Walkot [99] utilizes interpolation between local observer gains which has one major negative effect: large number of local models will give a large number of linear matrix inequalities (LMI) in the stability analysis and design, which may prohibit the use of existing LMI tools.

TS fuzzy system may be reduced to a dominant linear one, that is one local model is chosen and the effect of the rest are incorporated in it in terms of known deviations interpreted as uncertainties. This avoids the use of LMIs for analysis and design and instead a direct sliding mode observer design is possible. The model mismatches are taken into account as known upper bounds of matched and unmatched uncertainties.

2.3.8 LMI-Based Design of T-S Fuzzy Estimator based Controllers

Tanaka and Sugeno [100] proposed a theorem on the stability analysis of T-S fuzzy model. Later Wang et al. [101] proposed the so-called PDC as a design framework and also modified the Tanaka's stability theorem to include the effect of control. An important observation in the paper is that the stability problem is a standard feasibility problem with several LMIs when the feedback gains are pre-determined and can be solved numerically using an algorithm named interior-point method.

They are, however, NMIs (Nonlinear Matrix Inequalities) when the feedback gains are treated as unknowns. Later, Joh et al [102] converted the NMIs to LMIs for both of continuous and discrete T-S fuzzy controllers by applying the Schur complements to the Wang et al.'s stability criterion and named it as stability LMIs. And they proposed a systematic design method based on the stability LMIs for T-S fuzzy controllers which guarantees global asymptotic stability and satisfies desired performance of the closed-loop system.

Joh et al.'s assumed, however, that all the state variables are accessible. The fuzzy state estimator is proposed as a T-S type fuzzy rules using Wang et al.'s PDC structure to estimate the inaccessible states. In particular, a systematic design method for PDC - Fuzzy controllers with inaccessible states is obtained by combining LMIs for fuzzy state estimator and LMIs for control.

2.3.9 Fuzzy Regulators and Fuzzy Observers: Relaxed Stability Conditions and LMI-Based Designs

The issue of stability of fuzzy control systems has been considered extensively in nonlinear stability frameworks [103,104,105,106,107]. Specially, the stabilization of a feedback system containing a fuzzy regulator and a fuzzy observer for discrete fuzzy systems and, more importantly, new relaxed stability conditions and LMI- (linear matrix inequality) based design procedures are obtained for both continuous and discrete fuzzy systems. The stability analysis and design procedures proposed here are straightforward and natural, although the nonlinear regulator and observer design is difficult in general. Linear regulators and linear observers play an important role in modern control theory and practice. We envision that a systematic design method of fuzzy regulators and fuzzy observers would be important for fuzzy control as well. To begin with, Takagi–Sugeno (T–S) fuzzy models and previous stability results are recalled. To design fuzzy regulators and fuzzy observers, nonlinear systems are represented by T–S fuzzy models. The concept of parallel distributed compensation (PDC) [103,104,105] is used to design fuzzy regulators and fuzzy observers from the T–S fuzzy models. LMI-based design procedures for fuzzy regulators and fuzzy observers are constructed using the PDC and the relaxed stability conditions. Other LMI's with respect to decay rate and constraints on control input and output are also derived and utilized in the design procedures.

2.3.10 Separation principle: for the analysis & design of fuzzy controller & observer

The objective of this section is to develop a concept of separation property for the design of Fuzzy logic controller and observer, which helps in designing both of them separately as in normal control system. Analysis & Design of the Fuzzy Controller and Observer based on Takagi-Sugeno (T-S) fuzzy model is also discussed. A numerical simulation is also carried out to illustrate performance of the fuzzy controller & the fuzzy observer.

Since the last decade fuzzy logic based systems are gaining more attention from the scientific and industrial community. The fuzzy control departs significantly from the traditional control theory, which is essentially based on the mathematical models of the controlled process. Instead of deriving controller via modeling the controlled process quantitatively and mathematically, the fuzzy control methodology tries to establish a controller directly from the domain experts or operators, who are controlling the process manually and successfully. Other way it can be said that it's a expert system where primary attention is paid to the human's behavior and experience rather than to the process to be controlled. It is this distinctive feature that makes fuzzy control applicable and attractive for dealing with those problems where the process is so complex and ill-defined that is either impossible or too expensive to derive a mathematical model, which is accurate and simple enough to be used by the traditional control methods but the process may be controlled satisfactorily by the operators. This in turn leads to the fact that is has lack of theoretical basis and also that its performance is inconsistent.

Based on the above concept some fuzzy models based on the fuzzy control system design methods have appeared in fuzzy control field [108,109,110]. Linear feedback control methods can be utilized as in the case of feedback stabilization.

The procedure for that is as follows:

First, the non-linear plant is represented by its T-S type fuzzy model. In this type of fuzzy model local dynamics in different state space regions are represented by linear models. The overall model of the system is obtained by combining these linear models using non-linear fuzzy membership functions. The controller design is carried out using the parallel distributed compensation scheme. The resulting overall controller is non-linear in general and is again a fuzzy combination of each individual linear controller. The same procedure is used to design a fuzzy observer. The important point here is the separation property, which helps in designing the fuzzy controller and the fuzzy observer independently.

- **Plant Definition:** Obtaining mathematical models of the complex physical systems can be difficult or sometimes impossible too. But many of these systems can be expressed in some form of the mathematical model locally. Takagi and Sugeno have proposed a fuzzy model to describe the complex system [111]. In [112] we have consider a dynamic model to represent a complex MIMO system, which has both local analytic linear model and fuzzy membership functions. T-S dynamic model is described by the fuzzy IF-THEN rules, which locally represents linear input-output relations of nonlinear systems.

The i^{th} rule of the fuzzy model is:

Plant Rule i :

IF $m_1(t)$ is $\mathbf{F}_{i1}$ and … and $m_g(t)$ is $\mathbf{F}_{ig}$

$$\text{THEN} \quad \begin{aligned} \dot{x}(t) &= A_i x(t) + B_i u(t) \\ \dot{y}(t) &= C_i x(t) \end{aligned} \qquad (2.1)$$

Where, $\mathbf{F}_{ij}$ (j = 1, 2, …, g) are fuzzy sets, $\mathbf{x}(t) \in R^n$ is the state vector, $\mathbf{u}(t) \in R^m$ is the input vector and $\mathbf{y}_i(t) \in R^p$ is the output vector. ($A_i \in R^{nxn}$, $B_i \in R^{nxm}$, $Ci \in R^{pxn}$) is the matrix triplet, r is the number of IF-THEN rules and $m_1 \sim m_g$ are some measurable system variables.

Considering the T-S model and using a standard fuzzy inference method the final state of the fuzzy system is inferred as …

$$\dot{x}(t) = \sum_{i=1}^{r} \mu_i[m(t)][A_i x(t) + B_i u(t)] \tag{2.2}$$

Where, $\mu_i[m(t)]$ (= μ_i) is weighted average of fuzzy membership function at each local level and $m(t) = [m_1(t)\ m_2(t)\ ...\ m_g(t)]$. It is assumed in this paper that fuzzy membership at each level is positive and hence

$$\mu_i[m(t)] \geq 0,\ i = 1,2,...,r;$$
$$\sum_{i=1}^{r} \mu_i[m(t)] = 1 \quad \forall t$$

The final state of the fuzzy system can be represented as

$$\dot{x}(t) = \sum_{i=1}^{r} \mu_i A_i x(t) + \sum_{i=1}^{r} \mu_i B_i u(t) \tag{2.3}$$

The final output of the fuzzy system is inferred as follows…

$$y(t) = \sum_{i=1}^{r} \mu_i C_i x(t) \tag{2.4}$$

- **Design of Fuzzy Controller**

From the plant definition given earlier one can say that if the pairs (A_i, B_i), i = 1, 2,…, r are controllable, the fuzzy system is called locally controllable. For the design of fuzzy controller it is assumed that the fuzzy system (2.1) is locally controllable. First, the local state feedback controllers are designed based on the controller rules for each pairs (Ai, Bi):

Controller Rule i:

IF $m_1(t)$ is $\mathbf{F}_{i1}$ and …and $m_g(t)$ is $\mathbf{F}_{ig}$ THEN $u(t) = -K_i x(t)$, i =1, 2, …,r (2.5)

Then, the final output of the fuzzy controller is given as

$$u(t) = -\sum_{i=1}^{r} \mu_i K_i x(t) \qquad (2.6)$$

Where, μ_i is the same weight as i^{th} rule of the fuzzy system. The parameters of the controller are K_i in each rule. Substituting (2.6) into (2.3) we get

$$\dot{x}(t) = \sum_{i=1}^{r}\sum_{j=1}^{r} \mu_i \mu_j (A_i - B_i \cdot K_j)\, x(t) \qquad (2.7)$$

A sufficient condition that guarantees the stability of the fuzzy system is obtained in terms on Lyapunanov's direct method. The equation (2.7) is said to be asymptotically stable if there exists a positive semi definite matrix $\mathbf{P}_1$ such that

$$(A_i - B_i K_i)^T P_1 + P_1(A_i - B_i \cdot K_i) < 0$$

- **Design of Fuzzy Observer**

As we know in practice all the states of the systems are not fully measurable. Hence it is necessary to design the fuzzy observer in order to implement the fuzzy controller of (2.6).

It can be said that if pairs (A*i*, C*i*), $i = 1, 2, \ldots, r$ are observable, the fuzzy system is locally observable. First, the local state observers are designed based on the triplets (A*i*, B*i*, C*i*):

<u>Observer Rule *i*:</u>

IF $m_1(t)$ is $\mathbf{F}_{i1}$ and … and $m_g(t)$ is $\mathbf{F}_{ig}$ THEN

$$\dot{\hat{x}}(t) = A_i \hat{x}(t) + B_i u(t) + G_i[y(t) - \hat{y}(t)]$$
$$\hat{y}(t) = C_i \hat{x}(t), \qquad i = 1, 2, \ldots, r \qquad (2.8)$$

Where, G*i* (i=1, 2,…, r) are observation error matrices. $y(t)$ and $\hat{y}(t)$ are the final output of the fuzzy system and the fuzzy observer, respectively. Then the final estimated state of the fuzzy observer is

$$\dot{\hat{x}}(t) = \sum_{i=1}^{r} \mu_i A_i \hat{x}(t) + \sum_{i=1}^{r} \mu_i B_i u(t) + \sum_{i=1}^{r} \mu_i G_i[y(t) - \hat{y}(t)] \qquad (2.9)$$

the final output of the fuzzy observer is

$$\hat{y}(t) = \sum_{i=1}^{r} \mu_i C_i \hat{x}(t) \tag{2.10}$$

where, we use the same weight μ_i as the weight of i^{th} rule of the fuzzy system. The observer parameter for each rule is G_i. Substituting (2.4) & (2.10) into (2.9) we get

$$\begin{aligned}\dot{\hat{x}}(t) &= \sum_{i=1}^{r} \mu_i A_i \hat{x}(t) + \sum_{i=1}^{r} \mu_i B_i u(t) \\ &+ \sum_{i=1}^{r}\sum_{j=1}^{r} \mu_i \mu_j G_i C_j [x(t) - \hat{x}(t)]\end{aligned} \tag{2.11}$$

Using the final estimated state $\hat{x}(t)$, in (2.5) and (2.6), we get the following fuzzy controller:

Controller Rule *i*:

IF $m_1(t)$ is $\mathbf{F}_{i1}$ and ...and $m_g(t)$ is $\mathbf{F}_{ig}$ THEN $u(t) = -K_i \hat{x}(t)$ $i = 1,2,\ldots,r$ (2.12)

Then, the final output of the fuzzy controller is given as.

$$u(t) = -\sum_{i=1}^{r} \mu_i K_i \hat{x}(t) \tag{2.13}$$

Now, substituting (2.13) into (2.3) and (2.11),

$$\dot{x}(t) = \sum_{i=1}^{r} \mu_i A_i \cdot x(t) - \sum_{i=1}^{r}\sum_{j=1}^{r} \mu_i \mu_j B_i \cdot K_j \hat{x}(t) \tag{2.14}$$

$$\begin{aligned}\dot{\hat{x}}(t) &= \sum_{i=1}^{r}\sum_{j=1}^{r} \mu_i \mu_j (\mathrm{Ai} - \mathrm{Bi} \cdot \mathrm{Kj}) \hat{x}(t) \\ &+ \sum_{i=1}^{r}\sum_{j=1}^{r} \mu_i \mu_j\, Gi \cdot Cj \left[x(t) - \hat{x}(t) \right]\end{aligned} \tag{2.15}$$

let, $\tilde{x}(t) = x(t) - \hat{x}(t)$, this gives

$$\dot{\tilde{x}}(t) = \sum_{i=1}^{r}\sum_{j=1}^{r} \mu_i \mu_j \left(Ai - G_i \cdot C_j\right) \tilde{x}(t) \tag{2.16}$$

The equation (2.16) is asymptotically stable if there exists a positive definite matrix $\mathbf{P_2}$ such that

$$(A_i - G_i \cdot C_i)^T P_2 + P_2(A_i - G_i \cdot C_i) < 0 \qquad (2.17)$$

- **Separation Property**

The Separation principle of Estimation and Control in traditional control theory is [113]

"*When the control law* $u = -K\hat{x}$ *is used in conjunction with either a full order or reduced order state observer for*

$$\dot{x}(t) = Ax + Bu$$
$$y(t) = Cx$$

the controller gain K does not influence the eigenvalues of the state observer and the choice of the observer gain G does not influence the remaining eigenvalues".

The state- space transformation allows us to look at the system from a different but possibly more informative way. Lets look at the fact that under such transformations, the matrix "A" becomes "$P^{-1}AP$". Representing the equations (2.14) and (2.16) in matrix form as below...

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \sum_{i=1}^{r}\sum_{j=1}^{r} \mu_i\mu_j \begin{bmatrix} A_i - B_iK_j & B_iK_j \\ 0 & A_i - G_iC_j \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} \qquad (2.18)$$

Note that in this new realization, the matrix is block- triangular. According to the matrix algebra [114] "*Eigenvalues of a block triangular matrix are equal to the eigenvalues of the matrices along the diagonal blocks*".

Using this fact and the knowledge that system eigenvalues remain invariant under state transformations, we can say that the closed loop poles of the fuzzy observer based control system are union of the fuzzy observer poles and the fuzzy controller poles.

As controllability allows us to place the eigenvalues of

$$\sum_{i=1}^{r}\sum_{j=1}^{r}(\mu_i A_i - \mu_i\mu_j B_i K_j)$$

arbitrarily and the observability does the same for the eigenvalues of

$$\sum_{i=1}^{r}\sum_{j=1}^{r}\mu_i\,\mu_j\,(Ai - G_i C_j)$$

From the above two conditions we can say that designer has complete freedom in fuzzy controller and fuzzy observer pole selection as in the case of traditional control theory.

Stability and real time implementation are the major issues that also required to be addressed before finally controller gets implemented. Let us address the stability issues first.

2.3.11 Design of Fuzzy Controller – Stability concerns

Design of a stable controller for non-linear systems is always a complex problem. The choice of the identification and controller model for non-linear plants is a formidable problem and successful identification and control has to depend upon several strong assumptions regarding the input and output behavior of the plant. Soft computing based approach helps such problems by learning the system behavior. In order to analyze the system stability TSK fuzzy plant model is proposed. Fuzzy controller is designed and stability conditions for the same are derived.

To investigate the stability of the system, the Takagi-Sugeno-Kang (TSK) fuzzy plant model [111,115] is proposed. There are two ways to obtain the fuzzy plant model. One way is through identification of the model using input-output data of the plant and other is by direct derivation using mathematical model of the plant. Stability of such fuzzy plant and controller is to be investigated. Different stability conditions based to the Lyapunov stability theory [100] and other related approaches were reported. Using these stability conditions, closed loop system stability can be tested after finding the fuzzy controller parameters, which are usually determined by trial and error. Further ways to solve the stability conditions are usually not considered. If the stability conditions can be formulated as some LMI, there are softwares available to find the solutions numerically. However, formulating the stability conditions into an LMI problem will limit the realm of the stability analysis. In order to have systematic method to obtain fuzzy controller with guaranteed system stability, a fuzzy controller can be derived from genetic algorithm, which will be discuss in later chapter is proposed. The stability conditions for fuzzy controllers are first derived and based on these conditions, the parameters of the fuzzy controller is to be obtained using GA in the later chapter.

2.3.11.1 TSK Fuzzy Plant Model, Fuzzy Controller and Fuzzy Control System

Lets' consider a fuzzy control system, where a Non-linear plant is connected with a fuzzy controller in closed loop. The TSK fuzzy plant model is employed to describe the dynamics of the model. Let there are p rules for describing non-linear plant and c rules for fuzzy controller. The i-th rule for the plant and model can be written as

$$Rule\ i: IF\ f_1(x(t))\ is\ M_1^i\ and\ ...\ and\ f_\psi(x(t))\ is\ M_\psi^i\ THEN\ \dot{x}(t) = A_i x(t) + B_i u(t)$$

Where M_α^i is a fuzzy term of rule i corresponding to the function $f_\alpha(x(t))$ containing the parameter uncertainties of the nonlinear plant. $\alpha = 1,2, ... \psi$, $i = 1,2, ...p$. ψ is positive integer. $A_i \in \Re^{nxn}$ and $B_i \in \Re^{nxm}$ are known constant system and input matrices, respectively. $x(t) \in \Re^{nx1}$ is the system state vector and $u(t) \in \Re^{mx1}$ is input vector. The inferred system is given by

$$\dot{x}(t) = \sum_1^p w_i(x(t))(A_i x(t) + B_i u(t)) \quad where \sum_1^p w_i(x(t)) = 1,\ w_i(x(t)) \in [0\ 1]\ for\ all\ i$$

$$w_i(x(t)) = \frac{\mu_{M_1^i}(f_1(x(t))) \times \mu_{M_2^i}(f_2(x(t))) \times \cdots \times \mu_{M_\psi^i}(f_\psi(x(t)))}{\sum_{k=1}^{p} \left(\mu_{M_1^k}(f_1(x(t))) \times \mu_{M_2^k}(f_2(x(t))) \times \cdots \times \mu_{M_\psi^k}(f_\psi(x(t))) \right)}$$

is a non-linear function of x(t) and $\mu_{M_\alpha^i}(f_\alpha(x(t)))$ is the membership function corresponding to M_α^i. The value of $\mu_{M_\alpha^i}(f_\alpha(x(t)))$ can be known or unknown. If it is unknown function $(f_\alpha(x(t)))$ reflects the parameters uncertainties of non-linear plant.

A fuzzy controller will be obtained based on the TSK model [116]. The j-th rule of the controller has the form of...

$$Rule\ j: IF\ g_1(x(t))\ is\ N_1^j\ and\ ...\ and\ g_\Omega(x(t))\ is\ N_\Omega^j\ THEN\ u(t) = G_j x(t)$$

Where N_β^j is a fuzzy term of rule j corresponding to the function $g_\beta(x(t))$. $\beta = 1,2, ...\Omega$, $j = 1,2, ...c$. Ω is positive integer. $G_j \in \Re^{mxn}$ is the feedback gain of the rule j to be designed. The inferred output of the fuzzy controller is given by

$$u(t)=\sum_{j=1}^{c} m_j(x(t))(G_j x(t)) \quad where \sum_{j=1}^{c} m_j(x(t))=1,\ m_j(x(t))\in[0\ 1]\ for\ all\ j$$

$$m_j(x(t))=\frac{\mu_{N_1^j}(g_1(x(t)))\times\mu_{N_2^j}(g_2(x(t)))\times\cdots\times\mu_{N_\Omega^j}(g_\Omega(x(t)))}{\sum_{k=1}^{c}\left(\mu_{N_1^k}(g_1(x(t)))\times\mu_{N_2^k}(g_2(x(t)))\times\cdots\times\mu_{N_\Omega^k}(g_\Omega(x(t)))\right)}$$

is the non-linear function of x(t) and $\mu_{N_\beta^j}(g_\beta(x(t)))$ is the membership function of N_β^j to be designed. In order to carry out the analysis, closed loop fuzzy systems is to be there, which can be given by

$$u(t)=\sum_{i=1}^{p}\sum_{j=1}^{c} w_i(x(t))m_j(x(t))H_{ij}x(t) \quad where\ H_{ij}=A_i+B_iG_j.$$

2.3.11.2 Stability Analysis

Stability Analysis of the fuzzy control system can be carried out by considering the Taylor Series $x(t+\Delta t)=x(t)+\dot{x}(t)\Delta t+o(\Delta t)$.

Where $o(\Delta t)=-x(t)-\dot{x}(t)\Delta t+x(t+\Delta t)$ is the error term and Δt > 0, and

$$\lim_{\Delta t\to 0}\left\|o(\Delta t)/\Delta t=0\right\|.$$

Multiplying a transformation matrix T, having rank n and T^TT being symmetric positive definite matrix and then taking l_2 norm of the same we can derive that fuzzy control system will be exponentially stable [100,117],

if $\mu[TH_{ij}T^{-1}]\le -\varepsilon$ for all i & j.

Where $\mu[TH_{ij}T^{-1}]$ is given as $\mu[TH_{ij}T^{-1}]=\mu[T(A_i+B_iG_j)T^{-1}]=\lim_{\Delta t\to 0}\frac{\|I+TH_{ij}T^{-1}\Delta t\|-1}{\Delta t}$.

Here, ε is a non zero positive constant scalar. Which can be summarized in the lemma: *"The fuzzy control system defined above, which may have parameter uncertainties, is exponentially stable if THijT-1 is designed such that μ[THijT-1] ≤-ε for all i & j"*. It should be noted that with the use of a suitable transformation matrix T we can have the system which is exponentially stable. So, now problem is left with finding such matrix T for the given system.

In the paper [117], refers the derivation of transformation matrix T by way of Fuzzy logic and then later implemented using GA, which is discussed in the later chapters.

2.3.12 Fuzzy Controller - Real Time Implementation Issues

When it comes to implementing a fuzzy controller, we often want to try to minimize the amount of memory used and the time that it takes to compute the fuzzy controller outputs given some inputs. The pseudo-code in the earlier section was not written to exploit certain characteristics of the fuzzy controller that I had developed for the inverted pendulum; if I have to actually implement this fuzzy controller and have severe implementation constraints, I need to try to optimize the code with respect to memory and computation time.