Chapter 1

Overview

1. Overview

Since the era of Industrial revolution the usage of real time control system is well known but in the last 3 to 4 decades it has gain much more importance and the field of control system doubled every now and then. With the invention of High Speed Computational facilities the work in the area has gone to manifolds.

Conventional control has provided numerous methods for constructing controllers for dynamic systems. Some of these are:

- *Proportional-integral-derivative (PID) control:* Over 90% of the controllers in operation today are PID controllers (or at least some form of PID controller like a P or PI controller). This approach is often viewed as simple, reliable, and easy to understand. Often, like fuzzy controllers, heuristics are used to tune PID controllers,
- *Classical control:* Lead-lag compensation, Bode and Nyquist methods, root-locus design, and etc.
- *State-space methods:* State feedback, observers, etc.,
- *Optimal control:* Linear quadratic regulator, use of Pontryagin's minimum principle or dynamic programming, and etc.
- *Robust control:* H_2 or H_∞ methods, quantitative feedback theory, loop shaping, etc.,
- *Nonlinear methods:* Feedback linearization, Lyapunov redesign, sliding mode control, back stepping, and so on,
- *Adaptive control:* Model reference adaptive control, self-tuning regulators, nonlinear adaptive control, and so on,
- *Stochastic control:* Minimum variance control, linear quadratic Gaussian (LQG) control, stochastic adaptive control etc.,
- *Discrete event systems:* Petri nets, supervisory control, infinitesimal perturbation analysis etc.

Basically, these conventional approaches to control system design offer a variety of ways to utilize information from mathematical models on how to do good control. Sometimes they do not take into account certain heuristic information early in the design process, but use heuristics when the controller is implemented to tune it (tuning is invariably needed since the model used for the controller development is not perfectly accurate). Unfortunately, when using such approaches to conventional control, many of the engineers become somewhat removed from the control problem (e.g., when they do not fully understand the plant and just take the mathematical model as given), and unfortunately this leads to the development of unrealistic control laws. Sometimes in conventional control, useful heuristics are ignored because they do not fit into the proper mathematical framework, and this can cause problems.

As ~~the~~ discussed above lots of work has taken place in the area of Linear PID controllers, which also used as reference to newer developments in the field. The main disadvantage of the PID controllers is that they require manual adjustments to compensate for external perturbations. These adjustments are time consuming and tedious and are usually not optimal for the real time system in question. Furthermore, the PID controller can cause difficulties when the real time system makes large changes involving nonlinear dynamic behavior. To avoid these problems of fixed structure PID controllers, adaptive controllers were introduced in the 1970s and have remained a major area of research until recently.[1-4]. Looking into the past and recent work published in the National & International Journals, Seminars One can say that the modern cognitive methods can do comparatively well in developing the Real- Time Control System with robustness. The Methods preferred are Neural Network Controllers, Fuzzy Logic Controllers, Genetic Algorithm, Simulated Annealing and Hybrid Approaches, which includes combination of more ~~than~~ methods from above.

Realistic models of physical systems are nonlinear and usually contain parameters, which are either poorly known or depend on changing environment. If the parameters vary in a broad range, it is common to employ adaptation: a parameter estimator – identifier, which continuously acquires knowledge about the plant and uses it to tune the controller "on-line"[5].

Adaptive control is a natural strategy to enhance performance of uncertain systems with minimal sacrifice in performance. Early state feedback approaches in adaptive control were developed for linear systems [6-9], and for nonlinear systems with linearly parameterized uncertainty [7, 10]. Recent progresses in state feedback adaptive control include approaches that do not require any parameterization of state-dependent uncertainties [11, 12]. Design approaches for adaptive output feedback control are more limited in their applicability due to the fact that a state observer must be employed in the feedback architecture. Ref. [13] highlighted a destabilizing effect of the high-gain observer, a so-called peaking phenomenon, and suggested the employment of saturation in control input to prevent it. The peaking phenomenon has proven to be a fundamental obstacle in extending global results in linear systems to nonlinear systems [14]. In the case of uncertain nonlinear systems, results on adaptive output feedback control have been restricted to the class of nonlinear systems either affine in uncertain parameters [15] or with uncertainties dependent only upon available measurements.

Adaptive control is easier to achieve if one uses a reference model for the process [16]. The problem is how to obtain the good model for the process, considering that mathematical models are inefficient. Moreover there is some concern about the potential instabilities associated with adaptive systems behavior.[17]

Instabilities in nonlinear systems can be more explosive than in linear systems. During the parameter estimation transients, the state can escape to infinity in finite time. Hence adaptive nonlinear controller can't simply be the adaptive version of standard nonlinear controllers.

Recent advances in the computing technology are driving the development of the field of control by providing alternative strategies for the functionality and implementation of controllers for dynamical systems. In fact there is a trend in the field of control to integrate the functions of intelligent systems with conventional control systems to form highly autonomous systems that have capability to perform the complex control tasks independently with a high degree of success.

Intelligent control, the discipline in which control algorithms are developed by emulating certain characteristics of intelligent biological systems [5-7]. Intelligent control achieves automation via the emulation of biological intelligence. It either seeks to replace a human who performs a control task or it borrows ideas from how biological systems solve problems and applies them to the solution of control problems.[18].

Does the Intelligent controller exist? And the answer is yes. If you are to control something, you can call yourself an intelligent controller. Lets' take an example of control of child's behavior, an uncontrollable, stochastic, nonlinear and sometimes unstable system, don't we control them? There are intelligent controllers exists in technological problems too e.g. process operator at nuclear power plant, Missile guidance etc. But these are biological intelligent systems. Can we replace the biological intelligent systems with computers? Yes, we can but limiting to certain applications. If successful we can say that such artificial system as intelligent controller. In this sense conventional controllers can be called "intelligent" but there are degrees of intelligence. But if you look at completely replacing biological intelligent systems replacement or emulation using computers it is quite difficult as still the human brain as well as intelligence is not well understood. It is best to view "intelligent control" as a goal, rather than a reality.

Design and implementation of motion control applications includes the transition from control design to real time system implementation. To make this transition smooth, the specification model for the real time system should allow also for temporal requirements other than deadlines e.g. deviation from nominal period time of an activity, end-to-end timing constraints etc. [19-20]. To successfully design and implement real time control applications there is a need to make a smooth and predictable transition from the design of a control system to its implementation in the system. One important prerequisite to accomplish this for a real time system is to appropriately derive and model, application timing requirements. The typical timing constraints are tolerances on sampling periods, end-to-end timing constraints, and temporal correlation between sampling tasks.

For a real time system to function correctly, the controlling subsystem must be logically correct and free from timing faults. A system in which minor timing errors can be tolerated is called a soft real time system; a hard real time system fails catastrophically if even a single operation is performed at wrong time.

Neural Network Controllers, which are based on non-linear matrix-vector multiplication, are generally used as generic function approximators to map the desired input-output characteristics of the controller. The interesting advantage of such controllers is that they are trained from examples therefore they do not need any algorithmic model of the problem but again they require reference controller, which limits the performance.

An ARTIFICIAL NEURAL NETWORK (ANN) [21-31] is an attempt, to mimic the action of the brain using simple structure. The ANN is built up using a class of adaptive machine that perform computation through process of learning. The large number of inter-connected artificial neurons forms the network. Thus neural network consists of massively parallel distributed processors which have a neural propensity for storing experienced knowledge and making it available for use.

Input output relations (mapping) in the form of traditional mathematical modeling is replaced by ANN learning the synaptic weights by undergoing a training process. ANN has built in adaptability or can be trained to modify the weights with the change in environment. The ANN can deal naturally with contextual information. Since knowledge is represented by the regular structure and activation state of network. Every neuron is potentially affected by the global activity of all other neurons. ANN can be trained to make decisions and they are also fault tolerant in the sense that if a neuron or connecting link is damaged, recalling a pattern will be impaired in quality but due to distribution of information in the network damage has to be extensive for overall degradation. Since neurons are the common ingredients for all ANN, it is possible to Share the algorithm and structures in different applications. So it is possible to have a seamless integration of modules.

The ANN is suitable in the following situation….

Correct model of process may not be available or mode may be, complex with to many unacceptable assumptions The classical modeling algorithm may not respond well to the measurement noise in sensors or performance through classical algorithms may not be adequate. The FUZZY LOGIC based systems [32- 35] may be developed to overcome classical algorithm problems. There are many - similarities between ANN and FUZZY CONTROLLERS. The fuzzy logic frees us from the true/false reasoning of logical system of type that are used in symbolic languages.

Fuzzy linguistic models hold the promise of providing a finite qualitative partition of a quantitative dynamic system while being applicable to any system that can be described in linguistic terms. Fuzzy models provide a succinct and robust representation of systems that lack a complete quantitative model or have uncertain system perturbations. Consistency in reasoning, however, has not yet been proven for a fuzzy linguistic representation of a quantitative system.

Fuzzy linguistic models use fuzzy sets to create a finite number of partitions MBF of the inputs, outputs and states of a quantitative system. Currently most fuzzy models are implemented as a set of **if-then** rules, where the system input is used to evaluate the rules' antecedents and the model's output is the combined output of all the rules evaluated in parallel. This simple logical system, a Fuzzy Inference System (**FIS**), does not implement inference chaining and can only evaluate a simplified qualitative model of a plant. Recent work has expanded the usefulness of this structure by providing machine learning methodologies to adapt and **tune** fuzzy linguistic models and to automatically generate new models through **self-organization**.

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 (although sometimes forgotten) that heuristics enter the conventional control design process as long as you are concerned with the actual implementation of the control system. It must be acknowledged, moreover, that conventional control engineering approaches that use appropriate heuristics to tune the design have been relatively successful. You may ask the following questions: 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 control systems? Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system. Learning or tuning allows the initial linguistic fuzzy model developed from heuristic domain knowledge to be optimized.

To design the fuzzy controller,[36 – 40] 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 can come 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 be 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.

However, there are certain problems that encounter for practical control problems, including the following:

- The design of fuzzy controllers is performed in an ad hoc manner so it is often difficult to choose at least some of the controller parameters. For example, it is sometimes difficult to know how to pick the membership functions and rule-base to meet a specific desired level of performance.
- The fuzzy controller constructed for the nominal plant may later perform inadequately if significant and unpredictable plant parameter variations occur, or if there is noise or some type of disturbance or some other environmental effect.

Hence, it may be difficult to perform the initial synthesis of the fuzzy controller, and if the plant changes while the closed-loop system is operating we may not be able to maintain adequate performance levels meant the performance of the fuzzy controller is degraded. This suggests that we need a way to automatically tune the fuzzy controller so that it can adapt to different plant conditions. Indeed, it would be nice if we have the design method that could automatically perform the whole design task for us initially so that it would also synthesize the fuzzy controller for the nominal condition. This leads to adaptive fuzzy controllers, which can be either direct using reference model or using fuzzy inverse rules.

Evolutionary computing is a family of stochastic search techniques that mimic the natural evolution proposed by Darwin. If we consider intelligence as a kind of capability of an entity to adapt it self to ever changing environment, we could consider evolutionary algorithms as a subdivision of soft computing. These algorithms are made of the several iterations of basic evolution cycle.[41]. Evolutionary algorithms include various optimization methods such as Genetic Algorithm, Genetic Programming, Evolutionary Programming, Evolutionary Strategies etc.

The control of nonlinear systems is a formidable problem even if the process model is known. Normally, nonlinear models are linearized around a certain operating pointed. linear control theory is used to design controllers for the linearized models. A natural way is to search over the whole input space to find the optimal control sequence by direct evaluation of performance index in the whole input search space. However, the computations become prohibitive for even simple nonlinear plants.[42]. Dynamic programming using Bellman Principle of optimality found that search space grows very rapidly with the state dimension of the system, an undesirable inherent problem of dynamic programming. Gradient based optimization techniques are also available for the nonlinear controller design. However only local optimal solution is obtained and convergence of these algorithms is not guaranteed. Neural network and fuzzy logic have been also applied widely in the control of nonlinear systems.

Genetic Algorithms are robust search and optimization techniques in many practical problems.[43-50] GA are powerful optimization methods which can be used within real time control system in conjunction with other techniques either to find the best control parameters of the best network topology. They are substantially the trial and error methods based on the way biological evolution works. Genetic Algorithms are used in the most popular search tools, to get the best solutions of the optimization problems, in learning tasks, and in resolving designing problems. The main idea behind these algorithms is to choose a population of solutions for a given problem (selection) and to apply different genetic operators such as mutation and crossover over this population of solution in order to get the best one. The selection's criterion of individuals is a fitness function that's closely related to design problem on hand. Such function serves to evaluate the individuals of the population, in order to authorize the cycle of generation. Genetic algorithms are theoretically and empirically proven to provide robust search in complex spaces, giving the valid approach for the problems requiring efficient and effective searching.

The main advantages of the GAs in optimizing process are: A. The solution space is explored in parallel by searching different regions. This allows a global search in the solution space. B. The search mechanism is not based on gradient information and therefore no requirements on continuity or convexity of the solution space.[44]

The circumstances when the GA could be applied online and in real time includes: a. A process model is available and the time interval between samples is long enough to allow processing of GA for convergence, b. A GA could also be used, with little or no knowledge about the process, to learn to control the process in real time and c. A GA can be used for online adaptation of a well designed controller.[50]

Looking into the above aspects it can be said that a GA can prove better for the control of real time applications. But again none of the methods (other than PID controllers) can be efficiently used alone, for real time control problems, while combinations of two or more of them may provide interesting advantages. The Scope of this work is to establish the feasibility of using above methods when controlling real time systems.

The recent work published in the national/ international journals, from the seminars on control system, Soft computing, Fuzzy controllers, Genetic Algorithms, Tuning of Fuzzy rule base, different learning mechanism for adaptive fuzzy systems, Stability analysis and domain of attraction for stable systems, various learning algorithm, various learning methods using computer programming are developed.

The above mentioned research work has inspired the author to work in this direction and to think of better alternatives for implementation of Intelligent control for real time uncertain nonlinear systems using current soft computation techniques such as Fuzzy logic, Fuzzy model reference learning, Genetic Algorithm, Fuzzy – GA approach. The research work aims at developing algorithms for various real time systems using these techniques. Since in many cases it may not be possible to have an exact mathematical model for given system due to unknown dynamics, changes in the system parameters due to external perturbations etc., an alternative methods in terms of fuzzy, GA may be designed and implemented on computational platforms. The use of technical computing development support tools such as MATLAB, SIMULINK and related Tool Boxes [51-57] makes simulation study as well design of graphical user interface simpler.

The work described in the thesis includes:

- Design of Stable Fuzzy Observer & controller using Separation Principle using mathematical foundation of fuzzy system design.
- Development of Neuro-fuzzy controller for uncertain nonlinear real time systems.
- Development and Implementation of Fuzzy Model Reference learning controller, using first order and second order reference model, for uncertain nonlinear systems. MATLAB/ SIMULINK/ Fuzzy/ GA are used for Implementation of intelligent controller.
- Optimum tuning of the controller gains for the PID controls of the Real time systems using Genetic Algorithm and thus improving the performance of the for real time systems.
- Development of a Genetic Algorithm based intelligent controller for uncertain nonlinear real time system. Implementation of the GA based learning mechanism for the fuzzy model reference control of the nonlinear systems.

The rest of the thesis is organized as follows:

Chapter: 2 Background: Describes the survey of current trend in design of real time robust adaptive systems using classical methods. Design of Fuzzy systems and its applications to real time controllers is described. MATLAB is used for testing the design with various set of parameters. Design & stability analysis of Fuzzy observer estimator using Separation Principal is discussed.

Chapter:3 Evolutionary Computation: An Overview: The general preview of the Evolutionary computation for real time output Feedback control system such as Fuzzy logic, GA etc. are provided in this chapter with reference to controller design for real time systems. It also describes the software tools available for development of referred models and to carryout their simulation study.

Chapter: 4 Evolutionary Computing techniques: It provides a comprehensive study of the work done by the researchers using evolutionary computing techniques for the design and development of real time control system and also Contains design methodology for uncertain nonlinear real time controller design using hybrid evolutionary techniques.

Chapter: 5 Application: It provides a comprehensive study of the applications of evolutionary computing techniques in real time control system.

Chapter: 6 Describes a comprehensive study of the work done by the researchers for the real time control of Ship Steering and Aircrafts Maneuvering & Control Applications.

Chapter: 7 Discussion on the mathematical models developed for the applications under consideration.

Chapter: 8 Discussion on results of the simulation carried out for the applications using Evolutionary computing methods.

Chapter: 9 It contains conclusions as well as scope of future work

Chapter:10 It contains Bibliography.