Chapter 6

Soft computing: Miscellaneous Control Applications

Chapter 6
Soft computing: Miscellaneous Control Applications

Chapter provides a comprehensive study of the applications of soft computing techniques in control system. Applications in various process controls and other area where soft computing is applied are described. Design, Implementation, Animation and Simulation of inverted robotic arm is also described

6.1 An Adaptive H^{∞} Controller Using Ridge Gaussian Neural Networks

The autopilot design for bank-to-turn (BTT) missiles has received considerable attention according to BTT missiles has higher maneuverability and aerodynamic acceleration compared with skid-to-turn missiles. However, the requirement of high roll rate for BTT missiles to change the orientation of the acceleration will induce undesirable cross coupling between pitch and yaw motions [1]. Furthermore, the highly nonlinear aerodynamics and missile dynamics of non-minimum phase make the autopilot design more difficult.

A wide variety of approaches have been used successfully to address the autopilot design for missiles. Adaptive robust control based on well-known input/output (I/O) feedback linearization technique to achieve the satisfactory tracking performance has been presented in [2-3]. In [4], the gain-scheduling approach based on H^{∞} control theory was proposed. In the past decade, H^{∞} optimal control has been widely discussed for robustness and its capability of disturbance rejection in linear and nonlinear control systems [5-6], however, for partly unknown dynamics; the gain-scheduling for autopilot was not satisfactory. Exploiting neural networks for BTT missiles control has been studied recently years [7-10]. [1] although the hybrid radial basis function (RBF) network autopilot with localized learning capability has demonstrated better performance than gain scheduled autopilot, the adjustable parameters of RBFs are only the hidden-to-output weights.

As to H^{∞} control theory combined with neural networks, not only optimal tracking can be achieved while perturbations are absent, but also the worst case effect on the tracking errors due to the parameter uncertainties and external disturbances can be reduced to be less than or equal a desired level [11-12].

[13] integrates a proposed ridge Gaussian neural network, which is just a three-layer neural network with Gaussian activation functions, and control H^{∞} theory to enhance the BTT missiles autopilot design in handling the tracking control problem with unmodeled uncertainties. It can be shown that the ridge Gaussian neural network is equivalent to the radial Gaussian neural network with matrices of scales and rotations of input vectors for each node. The advantages of ridge Gaussian neural network are fewer parameters to be tuned than traditional radial Gaussian neural network and both input-to-hidden and hidden-to-output weights can be on-line tuned.

6.2 ANN for real-valued GA in knowledge acquisition

Rule-based expert systems are rather practical development in the artificial intelligent (AI) field. They are based on the premise that expertise can be encapsulated in a set of If-Then statements. Expert systems have already proven useful in many applications such as decision making, pattern recognition, speech understanding, fuzzy control, and so on. An area where expert systems find exciting applications is in medical diagnosis because there are many diagnostic processes which are guided by precompiled If-Then rules. In many cases, we have to deal with data which are incomplete, imprecise, uncertain, or vague. In order to handle such kinds of data the probability-based method is often adopted.

An important and well-known example is the MYCIN expert system [14] which introduces the concept of \certainty factors" to deal with uncertainty. However, humans do not think in terms of probability values but in terms of such expressions as \large", \very hot", \slow", and so on. This motivates the methods of incorporating fuzzy sets and/or fuzzy logic into expert systems and forms so-called fuzzy systems. Fuzzy approach is in a sense matched to human reasoning or decision-making.

The performance of the rule-based systems (either conventional expert systems or fuzzy systems) is highly related to the completeness of the knowledge base. The construction of a complete knowledge base involves the process of knowledge acquisition. Traditionally, the design of rule-based expert systems involves a process of interaction between a domain expert and a knowledge engineer who formalizes the expert's knowledge as inference rules and encodes it in a computer.

There are several difficulties in obtaining an adequate set of rules from human experts. Expert may not know, or may be unable to articulate, what knowledge they actually used in solving their problems. Often, the development of an expert system is time-consuming. Thus, the process of building an expert system requires much effort. Another important problem is that it is difficult to determine whether the knowledge base is correct, consistent and=or incomplete. One way to alleviate these problems is to use machine learning to automate the process of knowledge acquisition [15].

Neural networks are attracting a lot of interest in the scientific community because of their dynamical nature, robustness, capability of generalization and fault tolerance. A very appealing aspect of neural networks is that they can inductively acquire concepts from examples; therefore, neural networks have long been considered a suitable framework for machine learning. Basically, a neural network is a massively parallel distributed processor and can improve its performance by adjusting its synaptic weights.

Although neural networks have many appealing properties, there are three main disadvantages in neural net- works. The first one is that there is no systematic way to set up the topology of a neural network. The second is that it usually takes lengthy time to train a neural network. The third and the most apparent disadvantage is that a trained neural network is unable to explain its response because the knowledge is encoded in the values of synaptic weights. In other words, a neural network cannot justify its response on the basis of explicit rules or logical reasoning process. There have been several attempts to overcome the problem. One approach is to interpret or extract rules from a trained backpropagation network [16]. The algorithm proposed by Bochereau and Bourgine can only extract some sets of the Boolean logic rules [17]

Gallant developed a neural network expert system MACIE (MAtrix Controlled Inference Engine), which possesses features that are usually associated with conventional expert systems [18-19]. Basically, the above mentioned methods extract crisp rules. A new approach which have been attracting the growing interest of researchers is to integrate neural networks and fuzzy systems into an intelligent system. T

This neuro-fuzzy synergistic integration reaps the benefits of both neural networks and fuzzy systems [20-25]. The integrated systems possess the advantages of neural networks (e.g. learning abilities, capability of generalization, optimization abilities, and connectionist structures) and fuzzy systems (e.g. high-level If-Then fuzzy rule thinking and reasoning). Each has its own advantages and disadvantages.

[26] proposes to train fuzzy degraded hyperellipsoidal composite neural networks (FDHECNNs) so as to provide an appealing solution to the problem of knowledge acquisition. The values of the network parameters, after sufficient training, can be then utilized to generate fuzzy If-Then rules on the basis of preselected meaningful features.

A FDHECNN is a two-layer feedforward neural network. As can be seen, a multilayer feedforward neural network incorporated with a gradient-based training algorithm is apt to getting stuck in local minima during the training procedure. In order to increase the chance of arriving at the global minimum, genetic algorithms (GAs) provide a feasible alternative since GAs have proved to be robust, domain independent mechanisms for optimization. While GAs have the advantage of not getting stuck in local optima, conventional GAs have shortcomings.

Conventional GAs require the natural parameter set of an optimization problem to be coded as a finite-length string over some _nite alphabets, therefore, it results in the problem of imprecision and computation load resulting from the coding and decoding processes. Besides, when the search space is large, GAs usually take lengthy time to get into the region of global optimum and then arrive at it. These problems motivated us to propose a real-valued genetic algorithm which use real-valued genes. This special real-valued GA is then utilized to train FDHECNNs.

6.3 GA-Sugeno Integral for Set function Determination

The weighted average method, a classical linear model, is often used as an aggregation means in information fusion. It is, essentially, the Lebesgue integral [4] with respect to a classical measure additively determined by the weights on a discrete space consisting of all information sources. The use of this model is based on an assumption that the functions of diverse information sources are independent to each other and, therefore, the joint effect of several information sources to the target is just the simple weighted sum of effects of the individual information sources. Cases where weighted method fails, weights should be replaced by a nonnegative monotone set function, so-called importance measure that describes the importance of each individual information source as well as the interaction among them due to its non-additivity, and some type of nonlinear integral should be used as an aggregation means.

The crux of using the above mentioned new model in a real problem is how to determine the values of the importance measure. It is much more difficult than determining weights in the weighted average method since the weights are defined on the set of all information sources while the importance measure is defined on the power set of this set and a monotonicity restriction on the set function should be satisfied. Various strategies to construct nonnegative monotone set functions in systems have been proposed recently and are being developed now [27-32]. Among them, using statistics from data is the most practical one.

Recently, [33] shows a genetic algorithm used to determine nonnegative monotone set functions in systems where the Choquet integral is adopted as the aggregation means. In the current paper, we adopt another often used type of nonlinear integral, the Sugeno integral (fuzzy integral), as the aggregation means and a relevant algorithm is developed to determine importance measures.

[34] Introduces the concept of importance measure. It focuses on a discussion of Sugeno integrals (fuzzy integrals), a type of nonlinear integrals, with respect to an importance measure and demonstrates the fuzzy linearity of the Sugeno integral when the importance measure is fuzzy additive. A model of information fusion is developed where the Sugeno integral with respect to the importance measure is used as a means of aggregation.

To determine the values of the importance measure, a statistical method and, therefore, an optimization problem The Sugeno integral is regarded as a multi-input single-output system and its input{output data are collected. To solve the optimization problem, some soft computing methods can be adopted. A special genetic algorithm that is rather effective for our purpose. The algorithm has been run for a number of examples successfully.

6.4 Fuzzy Observer Design for Mobile Robot

The mobile robot can be applied to home appliance machine for assistance and security. It has to navigate autonomously and intelligently. When it moves on the floor, its dynamics depends on the floor materials and conditions. We have to design controller with considering them although we do not have any information on them.

Fuzzy logic algorithm is based on human intelligence and an expert's knowledge and performs well for the system that has nonlinear characteristics. And fuzzy logic .is robust on the change of system parameters and copes with the disturbance that has bad effects on the system [35] [36]. Because of these merits, fuzzy logic control is used to control a mobile robot nowadays.

In order to get better performance on the controller of mobile robot, we propose an observer which estimates state variables as well as disturbance. We usually use the sonar sensor to get the motion information of mobile robot. After that, it carries out path tracking. In the process, the path consists of straight routines and curves.

However, due to contour error and direction error, the robot stays easily out of the way. Those errors are caused by load disturbance include slip between wheel and floor and some unknown disturbance like nonlinear friction. In mobile robot, nonlinear friction is the factor which makes stick-slip induced by stiction and Stribeck effect [37].

To reduce these errors, [38] add it to controller so that the observer can estimate load disturbance as well as state variables to controller. It is necessary to control the robot with high performance and accuracy by proper modeling and friction compensation.

In designing an observer, [39] separates friction torque characteristic into two parts as one part due to mechanical characteristics of motor which has highly nonlinear characteristics and the other part due to slip between wheel and floor which has time varying characteristic. According to analysis on the observer, we can show the convergence of estimate states to true value. Simulation results show that fuzzy controller using the observer has good performance ever if the robot has different slip condition between two wheels.

6.5 Sliding Mode Observer

In the last couple of years the issue of observers in fuzzy systems was discussed from different points of view like observability and the separation of the controller and observer.

In [40] a fuzzy Thau-Luenberger observer is presented which eliminates uncertainties coming form different initial conditions of the observer and the plant. In some cases, however, local linear observers fail to eliminate modeling uncertainties which might even lead to instability. Sliding mode techniques used for observers have the advantage to cope with matched and even unmatched uncertainties very effectively. An extensive review of these techniques can be found in [41-42].

In [43] fuzzy sliding mode observers were firstly introduced and conditions have been given under which matched uncertainties can be eliminated. In [44] a sliding mode observer for a very general case was designed. In [45] a controller/ observer scheme for unmatched uncertainties is proposed. The fuzzy observer discussed in [47] uses a transformation of the locally linearized system into a specific canonical form. [47] approach utilizes results from [41,46]. The main idea of this approach is that one assumes the nonlinear system to be linear dominant within a certain operating region. The fuzzy approximation of a nonlinear system is briefly described. The design of a fuzzy sliding mode observer for a linear dominant system according to [46] is presented in [47] with its application to pendulum on a cart.

6.6 ANN Observers for Synchronous Generators

In recent years, there has been a considerable interest in the on-line estimation of synchronous generator parameters [48-53]. Online methods are particularly attractive since the machine's service need not be interrupted and parameter estimation is performed by processing measurements obtained during the normal operation of the machine.

The need for on-line parameter estimation arises because generator parameters tend to deviate substantially from nominal values obtained from off-line testing. These deviations are usually due to magnetic saturation [54-57], machine aging, internal temperature, the effect of centrifugal forces on winding contacts, and incipient faults within the machine. It is to be noted that although incipient faults can typically be detected by continuous or periodic monitoring of characteristic quantities [58-60] (such as efficiency, fuel and oil consumption, impurities in the cooling stream, radio frequency noise level, temperature etc.), not all faults may manifest themselves in change of these quantities. However, parameter estimates obtained by processing online measurements are useful for generator condition monitoring [61].

6.6.1 On-line Tracking of Data

Using an extensive database, online parameter estimates may be used to monitor generator condition and take preventative maintenance measures before complete breakdown occurs. Measurements acquired during synchronous generator testing are often a small subset of the machine's state vector.

The remaining unmeasurable components of the state vector are typically composed of currents in the rotor body circuits [62-64] which encapsulate high frequency sensitivity due to the flow of eddy currents in solid parts. In the absence of full-state measurement, the parameter estimation algorithm may require heavy computation to achieve convergence or may even fail to converge on a set of parameters, especially when poor initial estimates are used. When the state vector is completely known, parameter convergence is guaranteed and recursive estimation algorithms may be used to estimate machine parameters. Observers are frequently used to estimate immeasurable state vector components based on operating data.

It must be recognized that the problem of estimating a system's un-measurable states by processing its measurable states is essentially a system identification task and neural networks offer a promising means of achieving this [65])

An ANN observer is developed in [66] to map sequences of measured machine outputs to un-measurable rotor body currents by processing data acquired during transient disturbances. Data for developing the neural network model are obtained through off-line simulations of the synchronous machine model connected to an infinite bus system. It is assumed that the structure and order of the machine model used in generating data for developing the observer is accurately known. Nominal parameter values are used in the machine model. These assumptions are reasonable because, over the years, significant advances have been made to accurately model synchronous generators based on well-established modeling and parameter estimation techniques.

During training, all state variables of the machine model are assumed measurable. This would correspond to the stage when simulations are carried out to obtain a sufficiently accurate observer model. After developing the neural network observer, it can be used to estimate rotor body currents by processing measurements acquired on-line in an actual operating environment.

6.6.2 State Vector Estimation from On-line Operating Data

The measured responses are a small subset of the synchronous generator's state vector. In the absence of information pertaining to the generator's unmeasurable states (such as currents in the rotor-body circuits), estimation algorithms based on non-linear minimization techniques have to be used to estimate machine model parameters. If complete state information is available, recursive algorithms may be used to estimate the machine parameter vector.

Observers have often been used to estimate state information by processing available measurements. Investigators have developed various observers for estimating the state vector of a synchronous generator. Neural networks, with their parallel processing abilities provide a viable means for reconstructing, in real time, the synchronous generator's state vector from a set of measurements.

The implementation of a linear neural network in an actual operating environment will result in incorrect estimates of rotor body currents. This is because machine model parameters estimated on-line can deviate substantially from corresponding nominal estimates obtained off-line. On-line machine model parameter estimates are nonlinear in nature and may be influenced by generator operating condition.

ANN [67] observers account for model parameter non-linearities and provide accurate estimates of rotor body currents irrespective of generator operating condition. Instead of using nominal machine parameters, data for training the observers are generated through off-line simulations of a machine model whose parameters are varied according to on-line parameter estimates.

During training, all state variables of the machine model are assumed to be known. This corresponds to a stage when simulations are carried out to determine the order and parameters (weights and biases) of the ANN observers. After enhancing observer robustness to simulated parameter variations and noise, the trained ANN observers are tested with experimentally obtained on-line measurements to provide estimates of un-measurable rotor body currents. These estimates are then used along with experimental measurements to estimate machine parameters.

6.7 ANN Adaptive Controller for Robotic Manipulator

Robotic manipulators are complicated nonlinear dynamical systems with inherent unmodeled dynamics and unstructured uncertainties. These dynamical uncertainties make the controller design for manipulators a difficult task in the framework of classical adaptive and nonadaptive control. Design of ideal controllers for such systems is one of the most challenging tasks in control theory today, especially when manipulators are asked to move very quickly while maintaining good dynamic performance.

Conventional control methods such as proportional, integration, and derivative (PID) scheme, the computed torque scheme (CTM) and the adaptive control scheme (ACM) etc. are used.

The traditional PID control with a simple structure and implementation has been the predominant method used for industrial manipulator controllers. Though the static precision is good if the gravitational torques are compensated, the dynamic performance of PID controllers leave much to be de-sired.

CTM and ACM give very good performance, if manipulator dynamics are exactly known or the linearity in parameters of the robot dynamics holds. However, they suffer from three difficulties. First, they require explicit *a priori* knowledge of individual manipulators, which is very difficult to acquire in most practical applications. Second, uncertainties existing in real manipulators seriously devalue the performance of both methods. Although ACM has the ability to cope with structured uncertainties, it does not solve the problem of unstructured uncertainties. Third, the computational load of both methods is high. Since the control-sampling period must be at the millisecond level, this high computational load requires very powerful computing platforms that result in a high implementation cost.

A class of computational model known as neural networks (NNs) has been applied to robot control, which provides robotic manipulators with just such enhanced adaptive capability. Justification for using NNs for robot control lies in their excellent capability in learning any complicated mapping from training examples and generalizing what it has learned such that the robot controller is able to respond to an unexpected situation.

The parallel processing capability, when NNs have been implemented in hardware using very large scale integration (VLSI) technology, enables NNs to respond quickly in generating timely control actions. Much research effort has been put into the design of NN applications for robot control. The early applications of NNs in the control of robotic manipulators include Albus and Miller's CMAC Controller [67-68], Iiguni's linear optimal control techniques with backpropagation NNs [69], Kawato and Ozaki's feed-forward compensators using backpropagation NNs [70,71] for improving the control performance, etc.

NN-based control approaches described could give good simulations or even experimental results but lack of theoretical analysis and stability security makes industrialists wary of using the results in real industrial environments. To cope with these problems, stable NN-based adaptive control both in continuous and discrete time for robots has been recently investigated by many researchers.

Representatives of these researches are nonlinearly parameterized NN-based adaptive controllers [72-75] and linearly parameterized NN-based adaptive ones [76-79] for robotic manipulators.

In the proposed control schemes above, NNs are used to approximate the nonlinear components in the robot dynamic system, and Lyapunov stability theory or passive theory is employed to design a closed-loop control system with stability, convergence and improved robustness. As a result, the designed systems are stable, and online NN weight updating laws yield the function approximations. All these results have showed that stable NN-based control approaches do have the potential to overcome the difficulties in robot control experienced by conventional adaptive and nonadaptive controllers [80].

Most of the existing NN-based control approaches require the measurements of robot joint angle velocity, which may significantly deteriorate the control performance of these methods, because the velocity measurements are often contaminated by a considerable amount of noise. Furthermore, velocity sensors such as tachometers increase the weight and volume of the moving parts of the robot, thereby decreasing the robot's efficiency. Therefore, it is desired to achieve good control performance by using only joint position measurement [81].

In order to solve the NN-based adaptive tracking control problem for those manipulators using the position measurements only, an NN-based output feedback controller with an observer is proposed by Kim [82] for rigid robotic manipulators, which contains two NNs, one for the observer and the other for the controller. The controller design requires accurate knowledge of the robot inertia matrix, and the controller structure and the computing algorithms are very complicated.

In [83] hybrid control design is investigated by incorporating the merits of the NN-based adaptive control with the output feedback control of a robot. The output feedback control is used to stabilize the robot system with a linear observer, while the NN approach is employed to further improve the control performance of the controlled system by approximating the modified robot dynamics function The whole NN-based controller design, with a linear observer to estimate the velocity of the robot, only requires one NN. At the same time, the robot dynamics is assumed to be unknown. This paper gives the main results for designing such an observer-based adaptive controller for robots using multilayer NNs with sigmoidal activation functions. For performance comparison with the conventional adaptive algorithm as on-line approximator, the adaptive control algorithm proposed by Bayard and Wen [84] is expanded with an observer in the same control framework as the NN approach for robot trajectory tracking.

The effectiveness and efficiency of the proposed observer-based controller [84] using multilayer NNs are demonstrated in comparison studies with the conventional adaptive control algorithm by simulations of a two-link manipulator.

6.8 ANN based Process Estimation and Control

The common approach adopted to develop a realistic nonlinear model of a process system is based upon a first principles understanding. Since process systems may be complex this often necessitates the devotion of considerable time. Moreover, simplifying assumptions have to be made in many instances to enable a tractable solution to the modeling problem. A first principle model is very costly to construct and will be subject to inaccuracies due to the assumptions made during the development.

Although the concept of inferential measurement can improve control performance using conventional instrumentation there are certain situations where more advanced methodologies are required. In particular, the use of model based controllers have been shown to be useful when the process is nonlinear or large time delays exist. Significant attention has already been directed to the use of a nonlinear model directly within a control strategy (e.g. Lee and Sullivan. 1988 [84]

Unfortunately, the above techniques are primarily based upon mechanistic models and are thus dependent upon the accuracy of the nominal model used during control law synthesis. Initial studies by Willis et al. (1991) have shown that it is possible to develop a long range predictive controller where the nominal model is a neural network thus facilitating rapid and cheap development of a nonlinear control philosophy. This contribution builds upon these early studies to further demonstrate the utility of an ANN model based predictive control technique: Dynamic Network Control (DNC).

A desirable objective is the development and application of a technique which possesses generality of model structure (facilitating rapid and cheap development) but which could also be capable of learning and expressing the process nonlinearities and complexities. The ANN appears to offer this possibility.

In [85] ANNs are modified in order to model dynamic systems and utilized in process application. After introducing the basic concepts of ANNs, inferential estimation is discussed and a tentative exposition of an inferential controller comprising a neural network model (NNM) and a simple PI algorithm is presented.

The NNM is used to provide estimates of "difficult-to-measure" controlled variables by inference from other easily measured outputs. These estimates are then used for feedback control. The philosophy is used to provide more frequent measurements than could be achieved by hardware instrumentation. The particular advantage is that standard industrial controllers can then be employed.

6.9 Fuzzy Predictive Functional Control

In recent years, the predictive control has become a very important area of research. It is based on the prediction of the output signal at each sampling instant. The prediction is obtained implicitly or explicitly according to the model of the controlled process. Using the actual predictive control law, the control signal is calculated which forces the predicted process output signal to follow to the reference signal in way to minimize the difference between the reference and the output signal in the area between certain time horizons.

The fundamental methods that are essentially based on the principle of predictive control are Clarke's method, (generalized predictive control [86]), Richalet's method (model algorithmic control and predictive functional control [87]), Cutler's method (dynamic matrix control [88]), De Keyser's method (extended prediction self-adaptive control [88]), and Ydstie's method (extended horizon adaptive control [90]).

[91] combines a well-known method of predictive functional control together with fuzzy model of the process. The prediction is based on a global linear model, which is obtained by fuzzy model given in the form of Takagi–Sugeno (T–S) type.

The predictive control based on a fuzzy model [6] is capable to control also very difficult processes such as strongly nonlinear processes, processes with long time delay and nonminimum phase processes. The controllers based on prediction strategy also exhibit remarkable robustness with respect to model mismatch and unmodeled dynamics. The proposed fuzzy predictive control has been evaluated by implementation on heat-exchanger plant, which exhibits a strong nonlinear behavior

6.10 Optimal Fuzzy PID Controllers

Fuzzy PID-like controllers the design parameters within two groups: structural parameters and tuning parameters.

The structural parameters are determined during off-line design. Tuning parameters can be calculated during on-line adjustments of the controller to enhance the process performance, as well as to accommodate the adaptive capability to system uncertainty and process disturbance. Some parameters can be called either structural or tuning parameters depending on their usage.

A wide variety of fuzzy PID-like controllers have been developed Significant studies based on the closed-form analysis of fuzzy PID-like controllers started with the work of Ying, Siler, and Buckley [92-94], where they have used a simple four-rule controller similar to that of Murakami and Maeda [95]. More analytical work in this regard was subsequently reported for the four-rule controllers [96-98], and linear-like fuzzy controllers [99-100][3], [5]. Palm has analytically demonstrated the equivalence between the fuzzy controller and sliding-mode controllers [101]

It is possible to build a fuzzy controller which provides better performance than a conventional PID controller. In a study of optimal design for fuzzy controllers, two relationships must be established: 1) design parameters and control nonlinearity, and 2) control nonlinearity and process performance.

[102] uses an analytical approach to the optimal design of fuzzy controllers. A simple controller applying a single variable, three rules, and six design parameters is developed. The properties of the control action are discussed in terms of the design parameters. The nonlinear proportional gain is explicitly derived in an error domain.

In [102] The issues of nonlinear controller design are discussed and a conservative design strategy is suggested for a guaranteed-PID-performance fuzzy controller. Two indexes are proposed for the evaluation of nonlinear controller designs. For an optimal system design using genetic algorithms, an overall performance index is proposed including several individual performance indexes. Numerical studies are performed on several processes including nonlinearities due to time delay and saturation.

6.11 A Fuzzy RISC Processor

To provide high-performance fuzzy computation [103-104], several special purpose fuzzy circuit implementations and processors have been proposed. However, an approach based on specialized computing engines for performing fuzzy computation does not take into account that fuzzy processing is often not the only task required of the processing unit.

Fuzzy processing is often embedded into a complex system, requiring input/output (I/O) management, and other crisp operations. A few processors can support both general purpose computing and fuzzy computing, but they are mostly available at the low-performance microcontroller level. In this paper, we show how fuzzy processing can be implemented efficiently on general purpose processors and what functionality is required to achieve peak performance.

[105] has extended a general purpose reduced instruction set computing (RISC) processing unit with specialized fuzzy control operations to achieve high fuzzy processing performance with only minor changes in the processor, thereby preserving general purpose computing performance. This approach to designing application-specific processor variants based on a common reconfigurable processor core allows to build systems-on-a-chip with high fuzzy processing performance. Alternatively, the extensions can be integrated into a standard microprocessor part to create product differentiation for particular markets. To optimize tradeoffs between hardware resource utilization and fuzzy processing performance, a tech-Manuscript unique called hardware/software coevaluation is employed.[106] to evaluate different instruction sets and find the most promising in terms of performance and hardware complexity. The application specific instructions for fuzzy processing support is referred to as MIPS-F, for MIPS with fuzzy processing support. Amongst the new instructions, only one instruction is fuzzy specific, whereas all others are also useful for general purpose programs.

To achieve maximum performance, a technique called subword parallelism is used to pack multiple fuzzy data in a single processor word. This approach utilizes processor resources to the fullest, parallelizing multiple fuzzy inference steps and reducing memory traffic and reduces both memory access penalty and power consumption, which is of major concern in embedded applications where most fuzzy computation is used.

6.12 Adaptive Neural Observer-Estimator for Non-linear System

An Artificial Neural Network (ANN) based design of adaptive observer with error back propagation is considered for continuous time system. In this new approach, neural network is constructed and trained off-line and estimates of the plant output are generated through trained neural network.

The proposed scheme is adaptive and it is designed and simulated for control and stabilization of robot arm. Simulation of proposed scheme is done using MATLAB Simulink. Error and settling time are well below the acceptable limit. The capability of neural network is applied to estimate a part of nonlinear dynamics to control complex and unknown nonlinear control system. Application of neural network in such control will transform real time estimation of state (Observer) problem to simply adjustment of weights of neural network to find desired solution. The problem of Robot Arm Stabilization using Neural Network and Digital Signal Processor is addressed. It combines the approximation capabilities of neural network and computational power and speed of digital signal processor.

Many systems exist whose characteristics are difficult to mathematically model, making the design of an adequate controller a computationally intensive task. A controller that does not depend on exact characteristics but can adapt to differences in the system could eliminate the need for an exact model. An example of this type of system is the motor-driven inverted robot arm (Fig. 6.1): gear-induced time delay, wind and other such disturbances, and mass-distribution differences combine to produce a complex model, needing an even more complex controller. Since a human cannot always be on hand to balance the robot arm, it would be advantageous to have a controller which can mimic this ease of control.

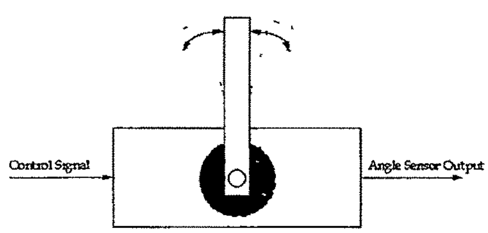


Fig 6.1: Inverted robot arm

The neural network is having linear variation in its output with variation in the input. The Network output is adaptive to the changes in the input.

6.12.1 MODES OF OPERATION

Fig. 6.2: Modes of operation: System block

The two modes of operation are described: learning and controlling.

1. In learning mode (Fig. 6.2, Top), a joystick is used to manually control the robot arm through a gain and protection circuit which keeps the current and arm-angles from exceeding safety limits and brings the control signal up to the power needed to drive the motor. The motor then adjusts the arm angle, which is detected using a sensor in the motor/arm assembly. The neural network to learn how to control the arm uses the angle-signal and the joystick signal. (Learning will be more fully described in the next section.)

2. In controlling mode (Fig. 2, Bottom), the joystick is removed and the neural network is connected to produce the control signal to manipulate the robot arm. It is necessary to generate the training data for the Learning Mode of controller using Simulink and to train the network and calculate the weights of Neurons using Least Mean Square (LMS) Algorithm.

6.12.2 PROGRAMMING

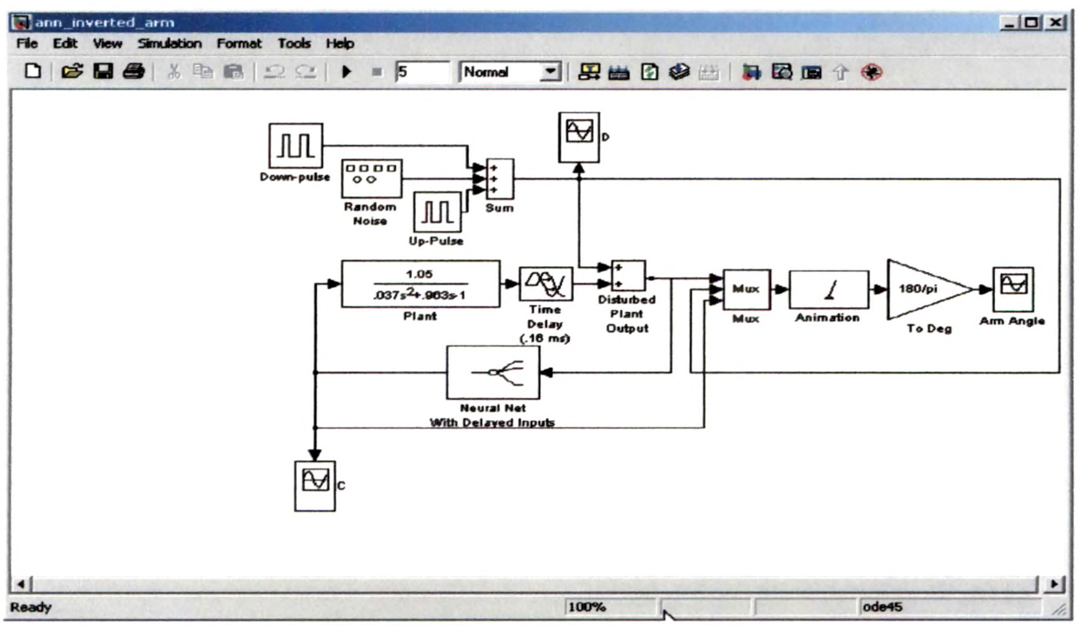


Fig 6.3: SIMULINK model for Inverted Robot Arm

The learning mode (Fig 6.4) generates the training data for the Neural Network (Fig 6.3). (Fig 6.5) depicts the Data set generation signals.

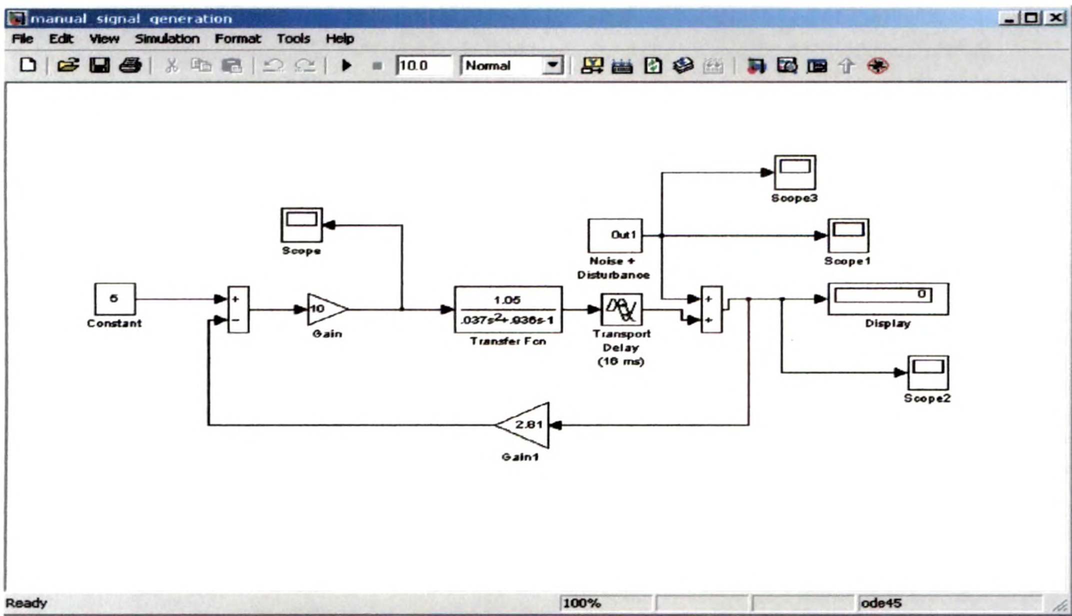
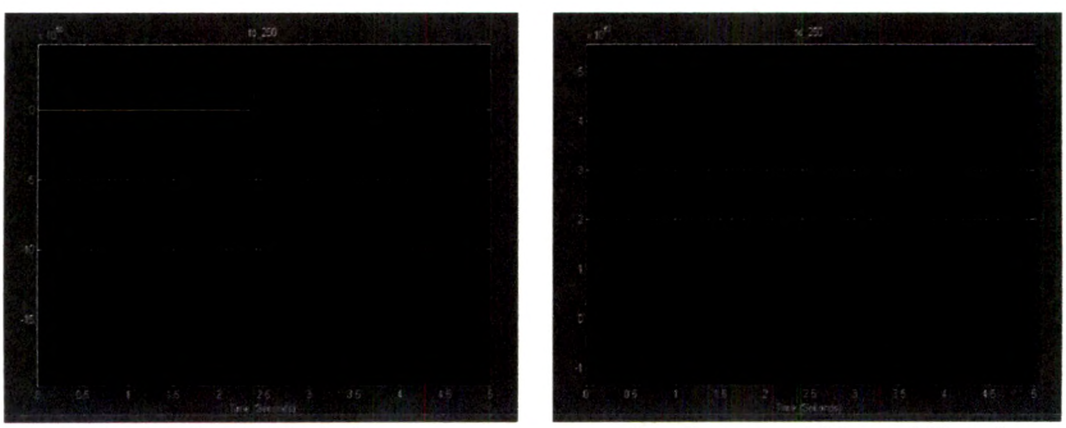


Fig 6.4 SIMULINK Setup for Manual Signal Generator

Least Mean Square (LMS) algorithm is used for training. ***MATLAB program*** calculates weights of Neural Network. These weights are used to determine response of Neural Network in the controlling mode. The accuracy of control is in the range of ± 2 degrees of variation in arm angle for worst case of noise and disturbance.



(a). Angle Variation **(b). Generated Manual Signal**

Fig 6.5: Data set Generation

- **Learning Mode** Learning mode of the controller can be described as learning phase of the neural network. In this mode the Neural Network is provided with the set of training data. The type of learning algorithm is decided. This training data is used as designated by the training rule of the Neural Network. The weights of Neurons are calculated as per the algorithm. The development of this phase basically consists of three considerations:
  - a) To generate the training data set
  - b) To decide the algorithm

c) To implement the chosen algorithm and calculate weights

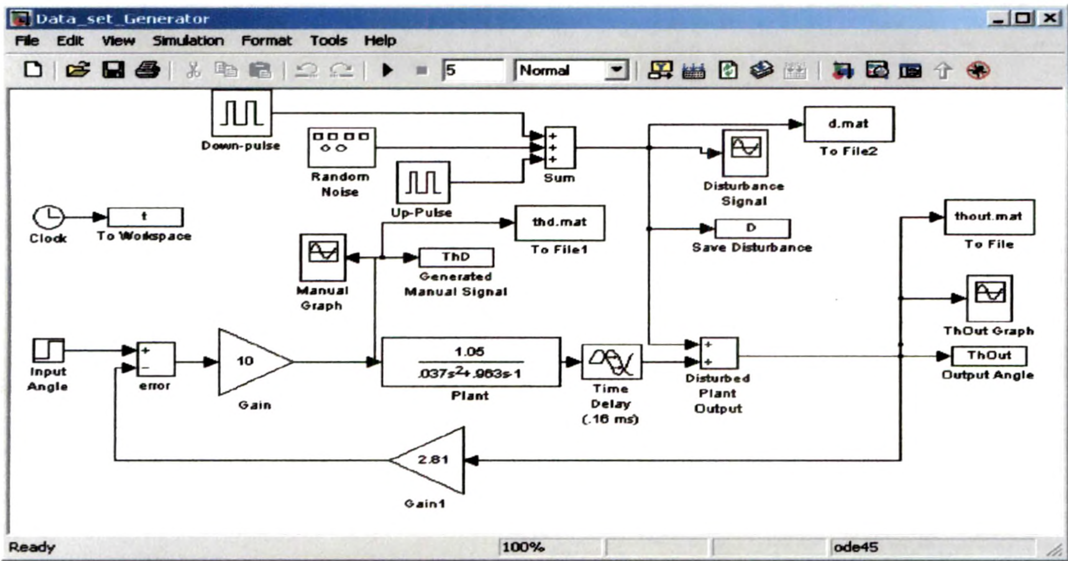


Fig 6.6 Classical SIMULATION setup

- **Controlling Mode:** Controlling the mode is the mode of normal operation of the ADALINE Controller. In this mode controller effectively controls robot arm angle based on its previous training experience. This mode can be implemented by following modifications in the model of learning mode. Artificial Neural Network governs the feedback loop. It takes output arm angle as input and considering the weights of neurons; it gives control signal, which is multiplexed with plant output and disturbance signal to generate the control signal. .

6.12.3 Simulation

The steps followed to simulate Neural Adaptive Observer and Estimator for robot arm areas follows:

- First to generate data for training of Neural network, Run td_250.m: this will create the disturbance and a couple other signals via Simulink.
- Save the results, which are in workspace to th250.mat file. So, the training data set is ready.
- Now, run lms2.m: this trains the neural network according to the training data found within th250.mat.
- You can clear out most of the variables from the workspace to save memory; just make sure you keep a copy of the weights (W).
- Create a new 30 x 1 matrix Xes by executing Xes = zeros(30,1);
- run ann_ctrl_cntr.m: to simulate the **controlling mode of ANN.**

After simulating the model following signals (Fig 6.7) are generated. The disturbance signal represents the noise and disturbance output.

(a). Disturbance signal.	**(b). Arm angle**	**(c)Generated Manual Signal**

Fig 6.7 Response after running Simulation

Arm angle and Manual Signal are the worth noting signals. By observing both signals we can say that arm angle and the Manual signals behave in complementary way. When Arm angle has a positive deviation due to noise and disturbances the Manual signal deviates in negative side to null the effect of the destabilization.

Animation Block: This block (Fig 6.8) is a MATLAB function block. It is used to pass the parameters to the M-function 'ann_robot.m', the arm angle measured from vertical position and direction vectors of robot arm. It draws the pendulum graphic to graphically show the robot arm during the simulation of the controlling mode

The simulation generates the disturbance signal representing the noise and disturbance output. The output of the animation block is a pendulum animation. The animation Fig 6.9 shows the effect of the noise and disturbance signals on the robot arm angle graphically.

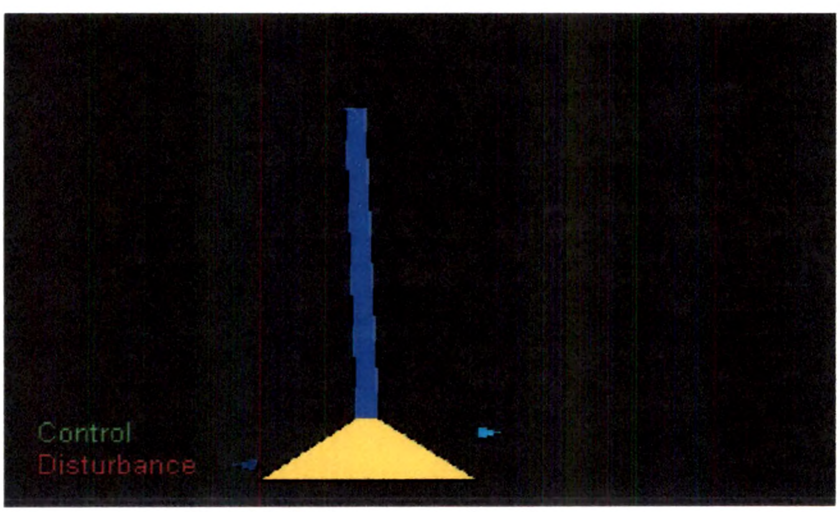
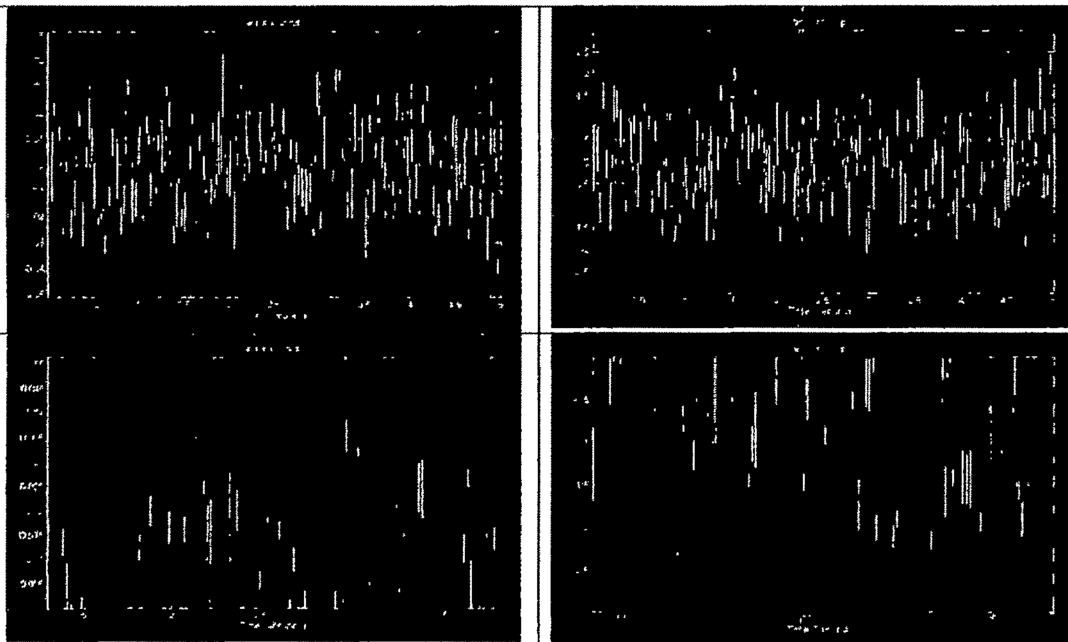


Fig 6.8. Robot arm animation

The output of M-function **anndelay** is the control signal, which takes into account the effect of noise and disturbance on the robot arm. This signal varies in the opposite direction of that of robot arm deviation in order to null the deviation.



Control signals	Arm angles
Fig 6.9 Response of signal generator	

The simulation response i.e Fig.6.9, makes it clear that when robot arm angle has positive deviation, control signal has negative deviation to null the effect of the deviation. The positive peak of the control signal and negative peaks of the arm angle deviation coincides The proposed Neural Adaptive Observer and Estimator for continuous time nonlinear system can be extended for discrete time. It may also be implemented using DSP

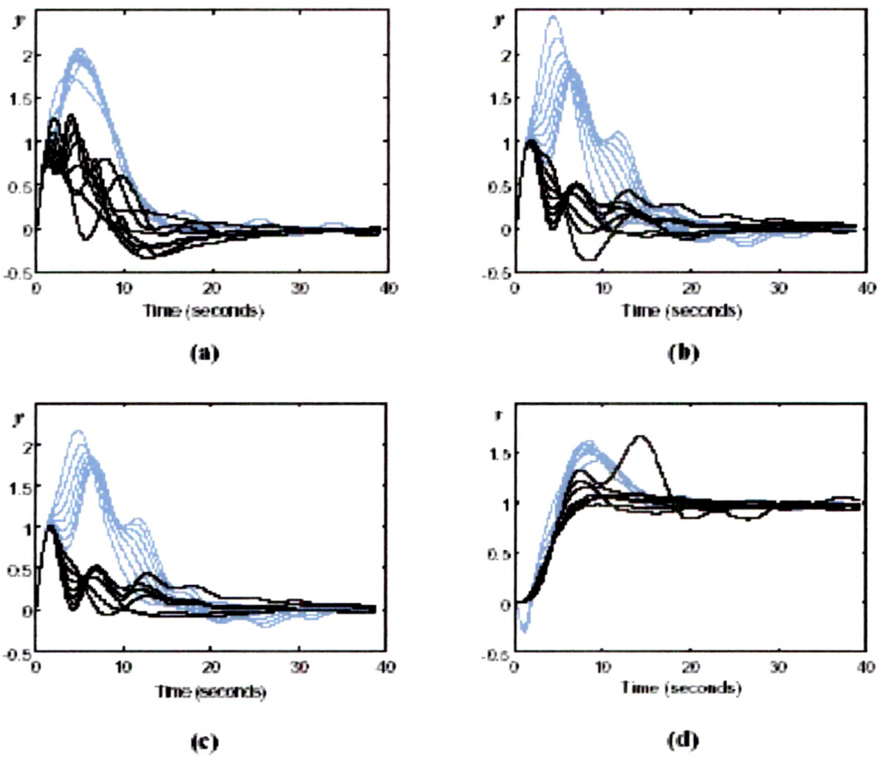


Fig 5.22: Performance comparison of fuzzy controller using state estimates from a PFI Kalman filter (black lines) and Comp1, a 5_{th} order H_2 compensator from Marrison and Stengel (gray lines).

(a) Output response to a unit impulse disturbance to m_2 for k = 0.6 to k = 2.0 in steps of 0.2

(b) for m_1 = 0.6 to m_1 = 2.0 in steps of 0.2, and

(c) for m_2 = 0.8 to m_2 = 2.0 in steps of 0.2.

(d) Tracking of a unit step command for k = 0.6 to k = 2.0 in steps of 0.2.

The following tabulates simulation results for an impulse disturbance on m_2:

	Comp1			**Fuzzy Controller A**		
k	0.5	1.0	2.0	0.5	1.0	2.0
$t_{\|y\| \le 0.1}$	26.0	14.4	14.3	25.3	15.0	24.2
$t_{\|L\| \le 0.05}$	>40	14.5	7.5	26.1	6.7	8.8
y_{max}	1.8	2.1	2.0	1.45	1.05	1.58
$\Sigma_{u15.0}$	2.6	2.8	2.7	3.1	2.2	3.9

	Comp3			**Fuzzy Controller B**		