IDENTIFICATION AND CONTROL OF DYNAMICAL SYSTEM USING NEURAL NETWORKS

By

D.CHITRA
Reg. No. 71203415002

of

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE – 641 006

A PROJECT REPORT

Submitted to the
FACULTY OF ELECTRICAL & ELECTRONICS ENGINEERING

In partial fulfillment of the requirements for the award of the degree

of

MASTER OF ENGINEERING

in

POWER ELECTRONICS AND DRIVES

ANNA UNIVERSITY
CHENNAI 600 025

June 2005

BONAFIDE CERTIFICATE

Certified that this project report titled **IDENTIFICATION AND CONTROL OF DYNAMICAL SYSTEM USING NEURAL NETWORK** is the bonafide work of Ms.**D.Chitra** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROJECT GUIDE

HEAD OF THE DEPARTMENT

The Candidate with University Register No: 11203415002 was examined by us in the project Viva-Voce examination held on 23-6-2005

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

The project takes the area of Artificial Neural Networks (ANN) and applies it to the inverted pendulum control problem. Control of an inverted pendulum has been considered as a fascinating, but difficult problem to solve since the system has very challenging characteristics such as non-linearity and a single-input multi-output structure. Neural networks have unique characteristics, which enable them to control non-linear systems. Feed forward Neural Networks (FNN) is used to model the inverted pendulum. A neuro-controller for the inverted pendulum was developed. Traditional control methods were utilized to develop a control law to stabilize the inverted pendulum. A feed-forward network was trained to mimic the control law. The neuro-control shows that if disturbance occurs in the system, the neural network learns to counteract this disturbance and produce the desired output.

ACKNOWLEDGEMENT

As I ponder in circumspect, there comes in number of persons who have contributed to my success in this work. Here, I want to thank the few among them.

I am thankful to Principal, **Dr.K.K.Padmanabhan B.Sc (Engg), M.Tech, Ph.D, FIE, FIMech.E, FI Prod.E, FIV, MISTE, MIIE.,** Kumaraguru College of Technology, for providing the facilities necessary to complete this project work.

I feel highly elevated in extending my gratitude to my course tutor, **Prof.K.Regupathy Subramaniam B.E(Hons), M.Sc, MIEEE, IES.,,** Professor and Head, Department of Electrical and Electronics Engineering, Kumaraguru College of Technology, for his constant support and encouragement throughout the project.

No word would be either superlative or hyperbolic to praise my guide **Prof.N.Kalaiarasi M.E, MISTE, AMIE, MSSI**, Assistant Professor, Department of Electrical and Electronics Engineering, whose adept advice and Valuable Suggestions made me stiff at the time of difficulty. Worthy to mention about her intransigent Character, which made me to work harder. I, Sincerely, owe my deepest sense of gratitude to her.

I take this opportunity to express my sincere thanks to **Dr.T.M.Kameswaran B.E, M.Sc (Engg), Ph.D, MISTE, Sr MIEEE, FIE.,** Former Head of Department, Electrical and Electronics, Kumaraguru College of Technology for his encouragement and moral support during the entire course.

Last, but not the least, I thank my Parents and Friends with felicitation who gave their moral encouragement and doughtiness to accomplish the project.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The inverted pendulum is a highly nonlinear and open loop unstable system. This means that standard linear techniques cannot model the nonlinear dynamics of the system. When the system is simulated the inverted pendulum falls over quickly. The characteristics of the inverted pendulum make identification and control more challenging. The aim of this project is to stabilize the angle of inverted pendulum and control the position of the cart.

Neural networks have shown great progress in identification of nonlinear systems. There are certain characteristics in Artificial Neural Network, which assist them in identifying complex nonlinear systems. Artificial Neural Network is made up of many nonlinear elements and this gives them an advantage over linear techniques in modeling nonlinear systems.

The main task of this project is to design a neural network controller, which keeps the pendulum system stabilized. There are 3 main types of neural control – supervised, direct inverse and unsupervised.

Supervised learning uses an existing controller or human feedback in training the neural network. In order to train the neural network to imitate an existing controller a vector of inputs and control targets from the controller must be collected. With supervised control, a neural network could be trained to imitate a robust controller. The robust controller can operate correctly, if the process operates around a certain point. The neuro-controller operates similarly to the robust controller but can also adapt if any disturbance occurs in the system. Direct inverse control does not require an existing controller in training.

A neural network is trained to model the inverse of the process. The neural network is cascaded with the process. Inverse control utilizes the inverse of the system model. A neural network is trained to model the inverse of the process. When the inverse controller is cascaded with the process the output of the combined system will be equal to the set point. Theoretically if the inverse model is very accurate, the nonlinearities in the ANN will cancel out the nonlinearities in the process.

1.1. OUTLINE OF THE PROJECT REPORT

Chapter 2 details the research on the inverted pendulum system. The dynamic system equations (nonlinear) are discussed. The simulink models of the nonlinear systems are developed. The development of the feedback controllers to stabilize the system is also discussed. Chapter 3 covers the theory, structure and operation of artificial neural networks. Chapter 4 details the development of the neuro-controller. Chapter 5 discusses the real time identification and control using the inverted pendulum rig. Chapter 6 details about the hardware implementation of the project. Finally Chapter 7 provides a summary of the work discussion of the results.

CHAPTER 2

INVERTED PENDULUM

2.1INTRODUCTION

An inverted pendulum is a physical device consisting in a cylindrical bar (usually of aluminum) free to oscillate around a fixed pivot. The pivot is mounted on a carriage, which in its turn can move in horizontal direction. The carriage is driven by a motor, which can exert on it a variable force. The bar would naturally tend to fall down from the top vertical position, which is a position of unsteady equilibrium.

The inverted pendulum is a traditional example (neither difficult nor trivial) of a controlled system. Thus it is used in simulations and experiments to show the performance of different controllers (e.g. PID controllers, state space controllers, fuzzy controllers....). An inverted pendulum application may suddenly be tasked with balancing a different mass, or length, or be placed on an incline, or it may encounter sustained gusts of wind, or friction on the ground or in the hinge, or obstacles in its way. Any number of new situations might arise, including wear and tear in the servos and actuators of the control machinery. An intelligent controller should be able to cope with change.

Inverted pendulum-like problems are found in applications as diverse as robotics, offshore drilling, Seaway Human Transporters, and even rocket science. For example, when a rocket launches into the sky, it has to be kept precisely balanced on top of its thrust vector or it will tumble out of control. Some of the solutions are very sophisticated: the space shuttle's vector control system monitors the ship's position and, if necessary, fires corrective bursts from a set of gimbaled thrusters, once every twenty milliseconds during the entire ascent.

2.2 INVERTED PENDULUM

The inverted pendulum system is a classic control problem. The process is non linear and unstable with one input signal and several output signals. The aim is to balance an inverted pendulum vertically on a motor driven wagon. It is possible to steer the wagon to different positions with a position reference signal. The problem resembles the control systems that exist in rockets. It is a suitable process to test prototype controllers due to its high non-linearities and lack of stability. The system consists of an inverted pole hinged on a cart which is free to move in the x direction. The dynamical equations of the system are derived, and the model is developed in simulink. The aim of developing an inverted pendulum in simulink is that the developed model will have the same characteristics as the actual process. The inverted pendulum model is developed in simulink, and the system dynamical equations are derived using 'Lagrange Equations'. The Lagrange equations are one of many methods of determining the system dynamical equations. Fig. 2.1 is a free-body diagram of the inverted pendulum system.

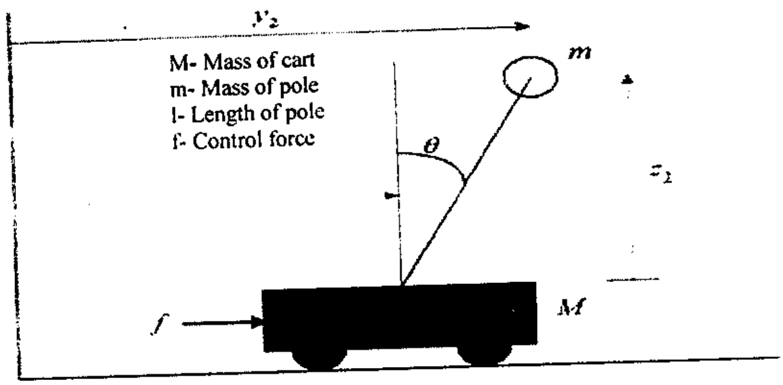


Fig.2.1.Free Body Diagram of Inverted Pendulum

The Lagrange equations use the kinetic and potential energy in the system to determine the dynamical equations of the cart-pole system.

The kinetic energy of the system is the sum of the kinetic energies of each mass. The kinetic energy, T_1 of the cart is

$$T_1 = \frac{1}{2} M \dot{y}^2 \quad (2.1)$$

The pole can move in both the horizontal and vertical directions so the pole kinetic energy is

$$T_2 = \frac{1}{2} m(\dot{y}_2^2 + \dot{z}_2^2) \quad (2.2)$$

From the free body diagram y_2 and z_2 are equal to

$$y_2 = y + l\sin\theta \quad (2.3)$$

$$z_2 = l\cos\theta \tag{2.4}$$

The total kinetic energy, T of the system is equal to

$$T = T_1 + T_2 = \frac{1}{2}\left[M\,y^2 + m\left(\dot{y}^2 + \dot{z}^2\right)\right] \tag{2.5}$$

Therefore,

$$T = \frac{1}{2}M\,\dot{y}^2 + \frac{1}{2}m\left[\dot{y}^2 + 2\,\dot{y}\,\dot{\theta}\,l\cos\theta + l^2\theta^2\right] \tag{2.6}$$

The potential energy, V of the system is stored in the inverted pendulum so

$$V = mgl\cos\theta \tag{2.7}$$

The Lagrangian function is

$$L = T - V = \frac{1}{2}(M+m)\dot{y}^2 + ml\cos\theta\,\dot{y}\dot{\theta} + \frac{1}{2}ml^2\theta^2 - mgl\cos\theta \tag{2.8}$$

The state variables of the system are y and θ, so the Lagrange equations are

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) - \frac{\partial L}{\partial y} = 0 \tag{2.9}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0 \tag{2.10}$$

But,

$$\frac{\partial L}{\partial \dot{y}} = (M+m)\dot{y} + ml\cos\theta\,\dot{\theta} \tag{2.11}$$

$$\frac{\partial L}{\partial y} = 0 \tag{2.12}$$

$$\frac{\partial L}{\partial \dot{\theta}} = ml\cos\theta\,\dot{y} + ml^2\,\dot{\theta} \qquad (2.13)$$

$$\frac{\partial L}{\partial \theta} = mgl\sin\theta \qquad (2.14)$$

Thus the non-linear dynamical equations for the inverted pendulum system are

$$(M+m)\ddot{y} + ml\cos\theta\,\ddot{\theta} - ml\dot{\theta}^2\sin\theta = f \qquad (2.15)$$

$$ml\cos\theta\,\ddot{y} - ml\sin\theta.\dot{y}\dot{\theta} + ml^2\,\ddot{\theta} - mgl\sin\theta = 0 \qquad (2.16)$$

Since the equations must be linearised we take cosθ=1 and sinθ=0 and assume θ to be small. The quadratic terms are also negligible. Therefore the two linear system equations are

$$\ddot{y} = \frac{f}{M} - \frac{mg}{M}\theta \qquad (2.17)$$

$$\ddot{\theta} = -\frac{f}{Ml} + \left(\frac{M+m}{Ml}\right)g\theta \qquad (2.18)$$

A set of equations describing the inverted pendulum has been developed. Then a simulnk model for the inverted pendulum system in Fig.2.2 is developed.

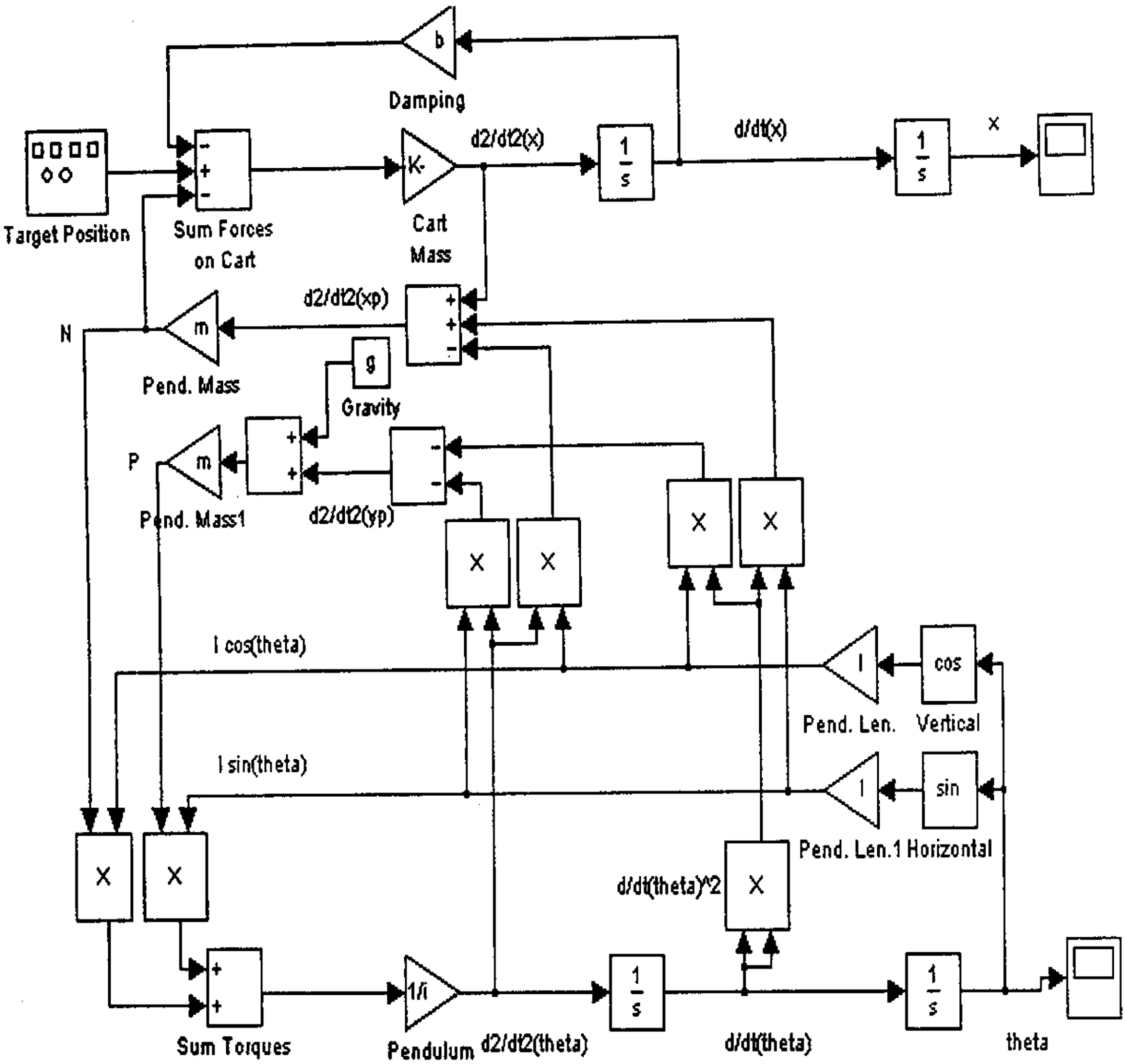


Fig.2.2. Simulink model of inverted pendulum

The following values are set for the system M = .5; m = 0.2; b = 0.1; i = 0.006; g = 9.8; l= 0.3. Thus the model is simulated and it is observed that the inverted pendulum angle falls over too quickly and becomes unstable as shown in Fig.2.3. In order to adequately model the inverted pendulum it is necessary to stabilize it using a feedback controller.

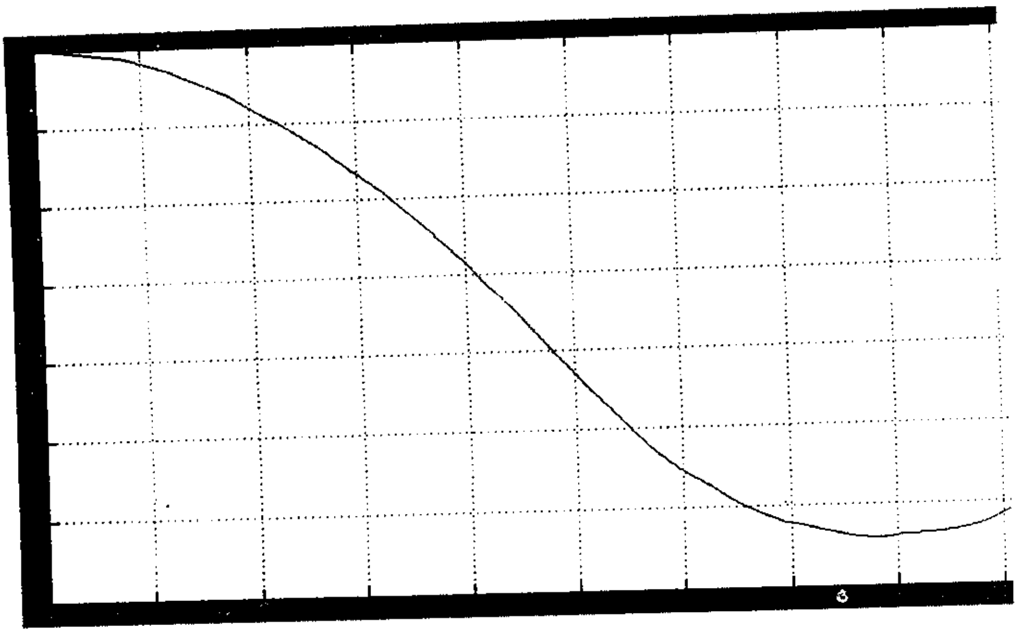


Fig.2.3. Response of Inverted Pendulum

CHAPTER 3

ARTIFICIAL NEURAL NETWORK

3.1 INTRODUCTION

Artificial Neural Networks are relatively crude electronic models based on the neural structure of the brain. The brain basically learns from experience. It is natural proof that some problems that are beyond the scope of current computers are indeed solvable by small energy efficient packages. This brain modeling also promises a less technical way to develop machine solutions. This new approach to computing also provides a more graceful degradation during system overload than its more traditional counterparts.

These biologically inspired methods of computing are thought to be the next major advancement in the computing industry. Even simple animal brains are capable of functions that are currently impossible for computers. Computers do rote things well, like keeping ledgers or performing complex math. But computers have trouble recognizing even simple patterns much less generalizing those patterns of the past into actions of the future.

Now, advances in biological research promise an initial understanding of the natural thinking mechanism. This research shows that brains store information as patterns. Some of these patterns are very complicated and allow us the ability to recognize individual faces from many different angles. This process of storing information as patterns, utilizing those patterns, and then solving problems encompasses a new field in computing. This field, as mentioned before, does not utilize traditional programming but involves the creation of massively parallel networks and the training of those networks to solve specific problems. This field also utilizes words very different from traditional computing, words like behave, react, self-organize, learn, generalize, and forget.

The fundamental processing element of a neural network is a neuron. This building block of human awareness encompasses a few general capabilities. Basically, a biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then outputs the final result. Fig.3.1 shows the relationship of these four parts.

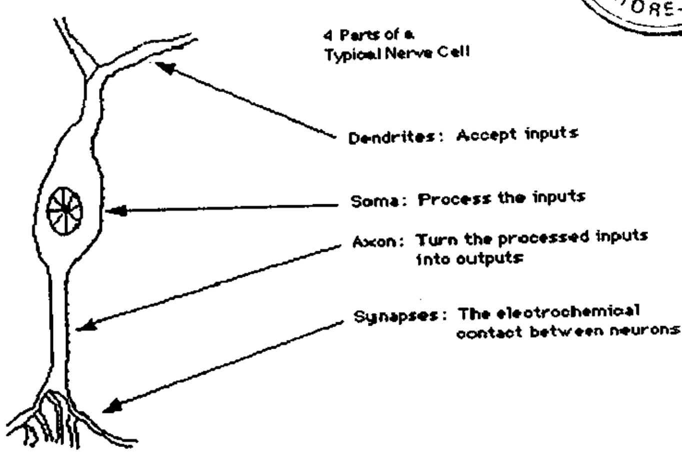


Fig.3.1.A Simple Neuron

The basic unit of neural networks, the artificial neurons, simulates the four basic functions of natural neurons. Fig.3.2. shows a fundamental representation of an artificial neuron.

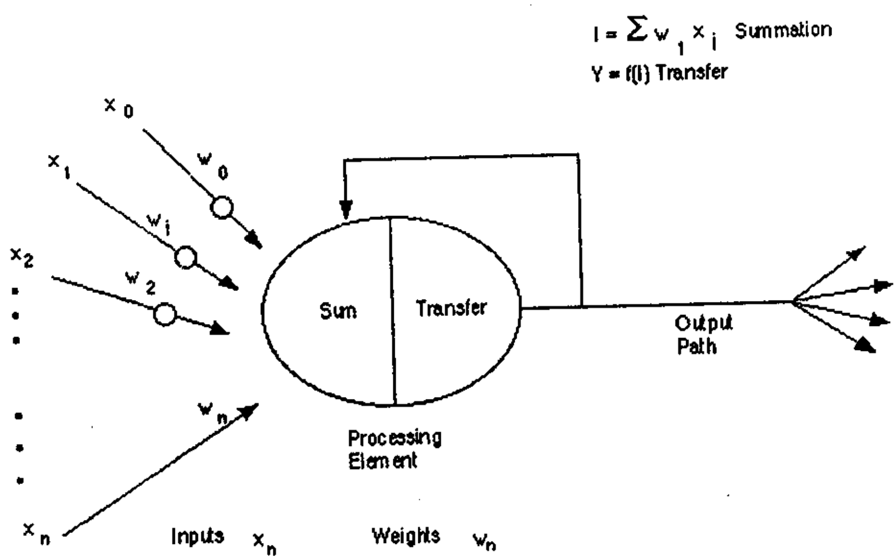


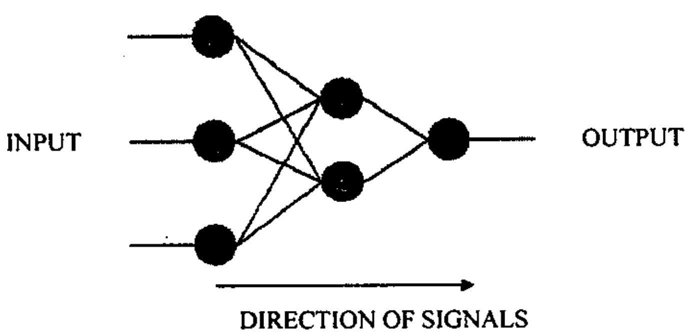
Fig.3.2.A Basic Artificial Neuron

In Fig3.2, various inputs to the network are represented by the mathematical symbol, x(n). Each of these inputs is multiplied by a connection weight. These weights are represented by w(n). In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then output. This process lends itself to physical implementation on a large scale in a small package. This electronic implementation is still possible with other network structures, which utilize different summing functions as well as different transfer functions.

3.2ADVANTAGES OF NEURAL NETWORK

1. The main advantage of neural networks is that it is possible to train a neural network to perform a particular function by adjusting the values of connections (weights) between elements. For example, if we wanted to train a neuron model to approximate a specific function, the weights that multiply each input signal will be updated until the output from the neuron is similar to the function.

2. Neural networks are composed of elements operating in parallel. Parallel processing allows increased speed of calculation compared to slower sequential processing.



3. Artificial neural networks (ANN) have memory. The memory in neural networks corresponds to the weights in the neurons. Neural networks can be trained offline and then transferred into a process where adaptive learning takes place. In our case, a neural network controller could be trained to control an inverted pendulum system offline say in the simulink environment. After training, the network weights are set. The ANN is placed in a feedback loop with the actual process. The network will adapt the weights to improve performance as it controls the pendulum system.

The main disadvantage of ANN is they operate as black boxes. The rules of operation in neural networks are completely unknown. It is not possible to convert the neural structure into known model structures such as ARMAX, etc. Another disadvantage is the amount of time taken to train networks. It can take considerable time to train an ANN for certain functions.

3.3TYPES OF LEARNING

A neural network is a massively parallel-distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network through a learning process.
2. Interconnection strengths known as synaptic weights are used to store the knowledge.

Basically, learning is a process by which the free parameters (i.e., synaptic weights and bias levels) of a neural network are adapted through a continuing process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place. In a general sense, the learning process may be classified as follows:

- Learning with a teacher, also referred to as supervised learning
- Learning without a teacher, also referred to as unsupervised learning

In supervised learning the output from the neural network is compared with a set of targets, the error signal is used to update the weights in the neural network. Reinforced learning is similar to supervised learning however there are no targets given, the algorithm is given a grade of the ANN performance. Unsupervised learning updates the weights based on the input data only. The ANN learns to cluster different input patterns into different classes.

3.4STRUCTURE OF NEURAL NETWORK

There are 3 main types of ANN structures -single layer feedforward network, multi-layer feedforward network and recurrent networks. The most common type of single layer feedforward network is the perceptron. Other types of single layer networks are based on the perceptron model. The details of the perceptron are shown below

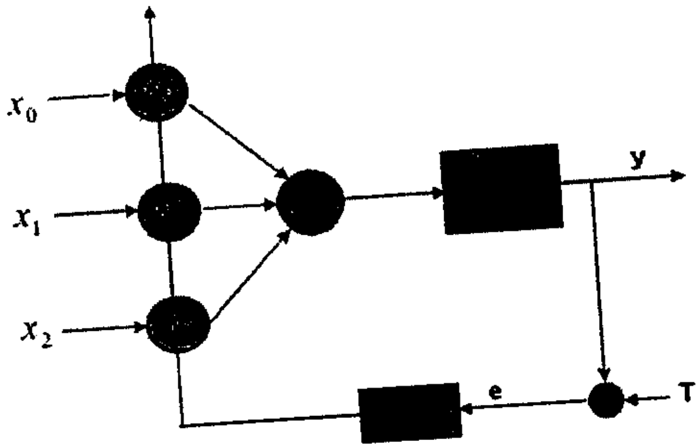
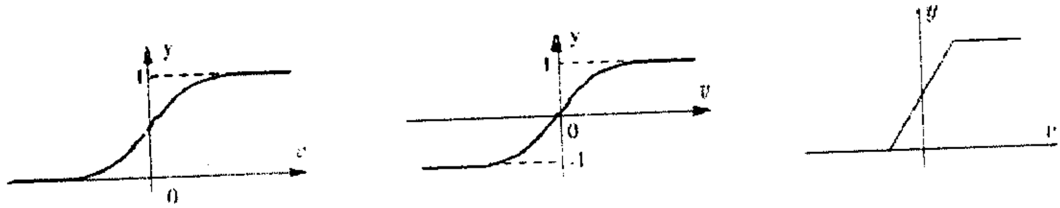


Fig.3.3.Details of perceptron

Inputs to the perceptron are individually weighted and then summed. The perceptron computes the output as a function F of the sum. The activation function, F is needed to introduce non-linearities into the network. This makes multi-layer networks powerful in representing nonlinear functions.

The 3 main types of activation function are -tan-sigmoid, log-sigmoid and linear. Different activation functions affect the performance of an ANN.



Log-sigmoid function **Tan-sigmoid function** **Linear function**

Fig.3.4.Activation Functions

The output from the perceptron is

$$y[k] = f\left(W^T[k]\right) \cdot x[k] \qquad (3.1)$$

The weights are dynamically updated using the feed forward algorithm. The difference between the target output and the actual output (error) is calculated.

$$e[k] = T[k] - y[k] \qquad (3.2)$$

The errors are back propagated through the layers and the weight changes are made. The formula for adjusting the weights is

$$W[k+1] = W[k] + \mu.e[k].x[k] \qquad (3.3)$$

Once the weights are adjusted, the feed-forward process is repeated. The weights are adapted until the error between the target and actual output is low. The approximation of the function improves as the error decreases. Single-layer feedforward networks are useful when the data to be trained is linearly separable. If the data we are trying to model is not linearly separable or the function has complex mappings, the simple perceptron will have trouble trying to model the function adequately.

3.4.1Multilayer perceptron

Neural networks can have several layers. There are 2 main types of multi-layer networks-Feed forward and recurrent.

The feed-forward neural network is a network of perceptrons with a differentiable squashing function, usually the sigmoid function. The back propagation algorithm adjusts the weights based on the idea of minimizing the error squared. The differentiable squashing function allows the back propagation algorithm to adjust the weights across multiple hidden layers. By having multiple nodes on each layer, n-separable problems can be solved, like the Exclusive-OR, or the XOR problem, which could not be solved with only the perceptron. Fig.3.5 shows a fully connected feed-forward neural network; from input to output, each node is connected to every node on the adjacent layers.

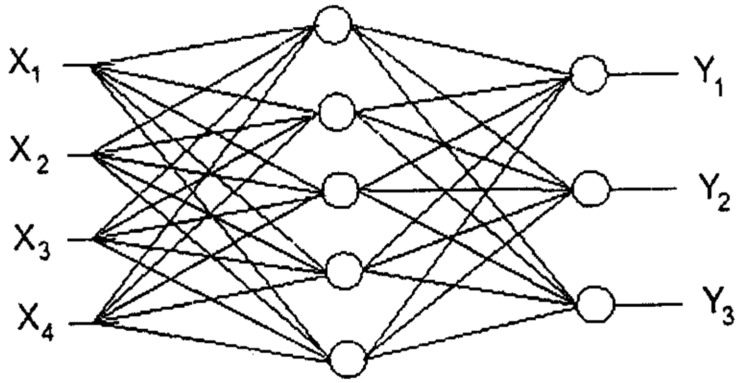


Fig.3.5 Feed-Forward Network

In feedforward networks the direction of signals is from input to output, there is no feedback in the layers. The diagram 3.6 shows a 3-layered feedforward network. Increasing the number of neurons in the hidden layer or adding more hidden layers to the network allows the network to deal with more complex functions. Cybenko's theorem states that, "A feedforward neural network with a sufficiently large number of hidden neurons

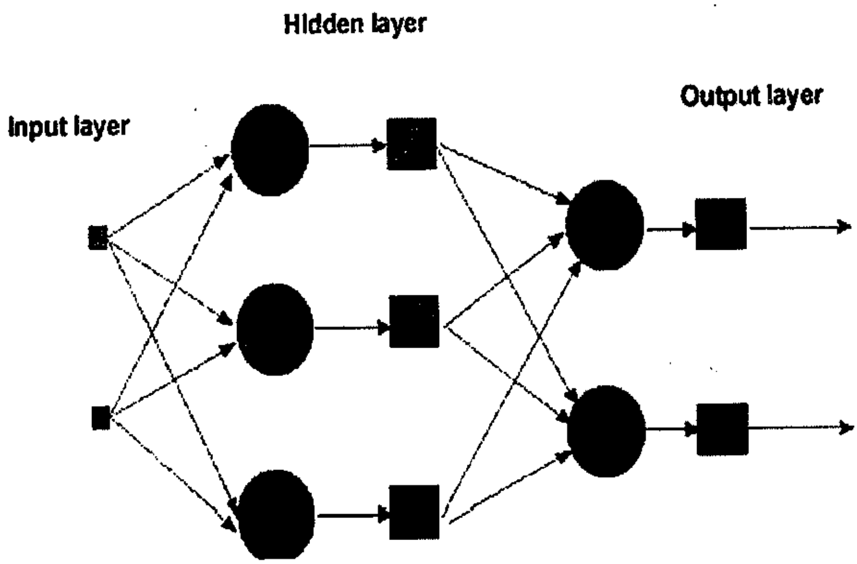


Fig.3.6.Multi layered perceptron

with continuous and differentiable transfer functions can approximate any continuous function over a closed interval." The weights in multi-layer network are updated using the back propagation learning. There are two passes before the weights are updated.

In the first pass (forward pass) the outputs of all neurons are calculated by multiplying the input vector by the weights. The error is calculated for each of the output layer neurons. In the backward pass, the error is passed back through the network layer by layer. The weights are adjusted according to the gradient decent rule, so that the actual output of the multi-layer networks moves closer to the desired output. A momentum term could be added which increases the learning rate with stability.

Multilayer perceptron networks are general-purpose, flexible, nonlinear models consisting of a number of units organized into multiple layers. The complexity of the multilayer perceptron network can be changed by varying the number of layers and the number of units in each layer. Given enough hidden units and enough data, it has been shown that multilayer perceptron can approximate virtually any function to any desired accuracy. In other words, multilayer perceptron are universal approximations. Multilayer perceptron are valuable tools in problems when one has little or no knowledge about the form of the relationship between input vectors and their corresponding outputs.

The training set is presented iteratively to the network until a stable set of weights is achieved and the error function is reduced to an acceptable level. Fig.3.7 summarizes the training procedure of the multi-layer feed-forward neural network. To measure the generalization ability of the multi-layer feed-forward neural network it is common to have a set of data to train the network and a separate set to assess the performance of the network during or after the training is complete. Once the neural network has been trained, the weights are saved to be used in the classification phase. During classification, image data are fed into the network, which performs classification by assigning a class number to a pixel or segment using the numerical values computed at the output layer. Typically the output node giving the highest value is taken as the best class to assign.

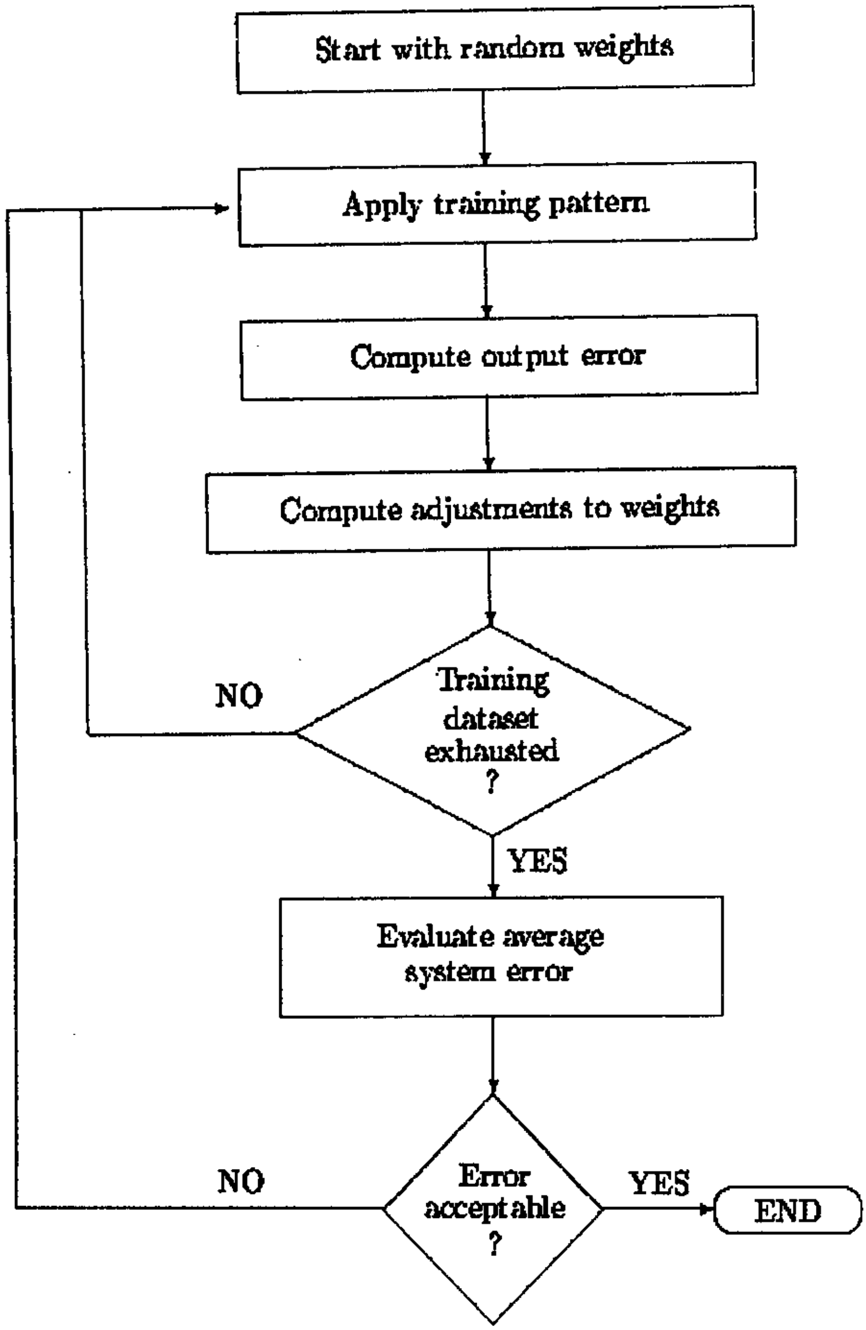


Fig.3.7.Flowchart of training procedure

The second type of multi-layer networks is recurrent Fig.3.8 Recurrent networks have at least one feedback loop. This means an output of a layer feeds back to any proceeding layer.

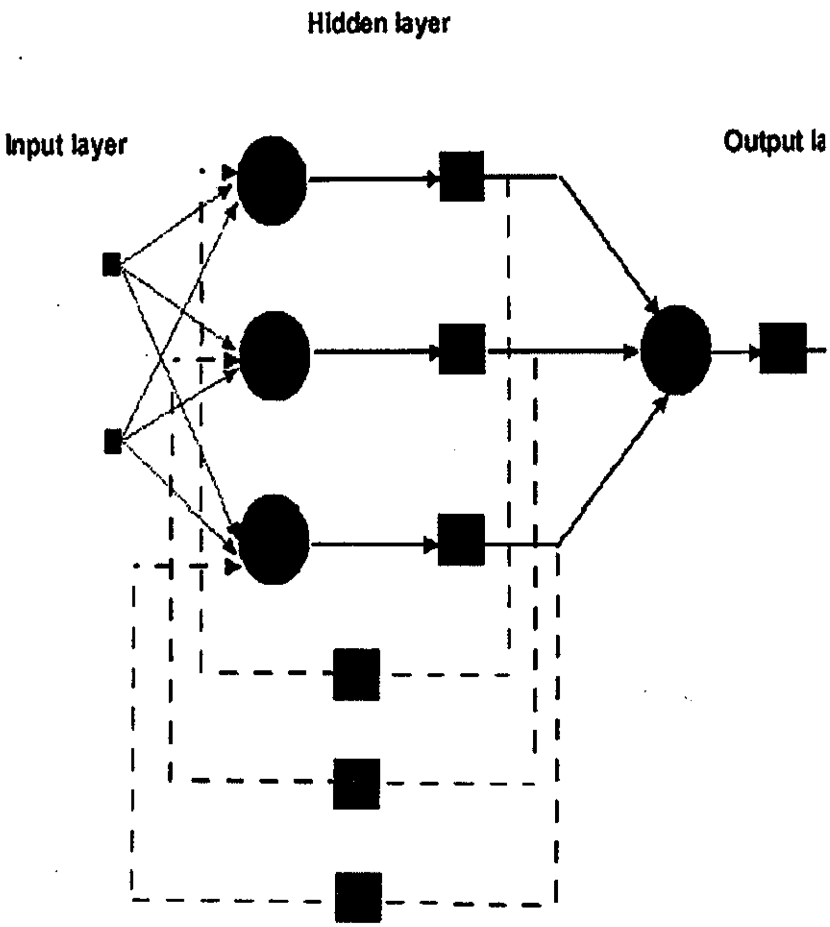


Fig.3.8.Recurrent neural network

This gives the network partial memory due to the fact that the hidden layer receives data at time t but also at time t-1. This makes recurrent networks powerful in approximating functions depending on time. Feedback networks can have signals traveling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Recurrent networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Recurrent architectures are also referred to as interactive or feedback, although the latter term is often used to denote feedback connections in single-layer organizations.

The simulink model for the nonlinear inverted pendulum shows that there are no feedback loops. It is expected that to accurately model this type of dynamic system, a feedforward neural network will perform better. Thus a static feedforward neural network is used to control the angle of the dynamical system and in the corresponding chapters the model is described.

CHAPTER 4

NEURAL CONTROL OF INVERTED PENDULUM

The main task of this project is to design a controller which keeps the pendulum system inverted. There are a few important points to remember when designing a controller for the inverted pendulum. The inverted pendulum is open-loop unstable, non-linear and a multi-output system. To show the advantages of using neural-control in this project a comparison between a standard PID control and neuro-control is made.

Nonlinear system: Standard linear PID controllers cannot be used for this system because they cannot map the complex nonlinearities in the pendulum process. ANN's have shown that they are capable of identifying complex nonlinear systems. They should be well suited for generating the complex internal mapping from inputs to control actions.

Multi-output system: The inverted pendulum has four outputs, in order to have full state feedback control four PID controllers would have to be used. Neural networks have a big advantage here due to their parallel nature. One ANN could be used instead of four PID's.

Open-loop unstable: The inverted pendulum is open-loop unstable. As soon as the system is simulated the pendulum falls over. Neural networks take time to train so the pendulum system will have to be stabilized somehow before a neural network can be trained. Before the actual neuro-controller is developed in matlab, the main types of neuro-control are discussed.

The five types of neural network control methods that have been researched are supervised, model reference control, direct inverse, internal model control and unsupervised.

4.1SUPERVISED CONTROL

Learning with a teacher, is referred to as supervised learning. Most tradional controllers (feedback linearization, rule-based control) are based around an operating point. This means that the controller can operate correctly if the plant/process operates around a certain point. These controllers will fail if there is any sort of uncertainty or change in the unknown plant. The advantage of neuro-control is if an uncertainty in the plant occurs the ANN will be able to adapt its parameters and maintain controlling the plant when other robust controllers would fail. In supervised control, a teacher provides correct actions for the neural network to learn. In offline training the targets are provided by an existing controller, the neural network adjusts its weights until the output from the ANN is similar to the controller as shown in Fig.4.1.

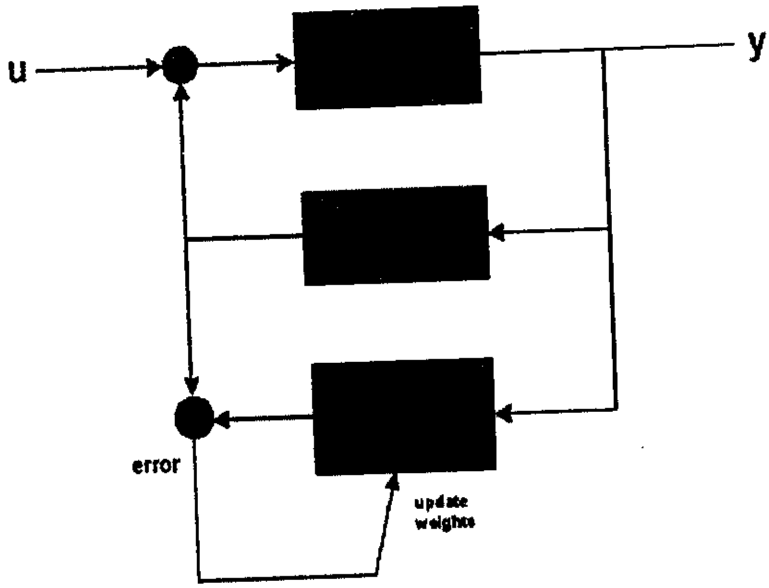


Fig.4.1.Supervised learning using an existing controller

Supervised learning algorithm, utilize the information on the class membership of each training instance. The information allows Supervised learning algorithm to detect pattern misclassification as a feedback to them. Error information contributes to the learning process by rewarding accurate classifications – a process known as credit and blame assignment. It also helps to eliminate implausible hypothesis. Supervised learning which incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be. During the learning process global information may be required. Paradigms of supervised learning include error-correction learning, reinforcement learning and stochastic learning. An important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of error between the desired and computed unit values. The aim is to determine a set of weights, which minimizes the error.

4.2MODEL REFERENCE CONTROL

Model reference control or, Neural Adaptive Control uses two networks: a controller network and model network. The model network can be trained off-line using historical plant measurements. The controller is adaptively trained to force the plant output to track a reference model out-put. The model network is used to predict the effect of controller changes on plant output, which allows the updating of controller parameters.

Model reference control is regarded as an adaptive servo system in which the desired performance is expressed in terms of a reference model, i.e. desired response to a command signal. It consists of two loops: an ordinary feedback loop composed of process and controller and another loop that changes the controller parameter. The parameters are changed on the basis of feedback from error and output of reference model. Thus the ordinary feedback loop is known as inner loop and parameter adjustment loop is known as outer loop. The mechanism for adjusting the parameter in model reference control is by two ways: using gradient method and by applying stability theory.

In model reference control the desired closed loop response is specified through a stable reference model. The control system attempts to make the process output similar to the reference model output as shown in Fig.4.2.

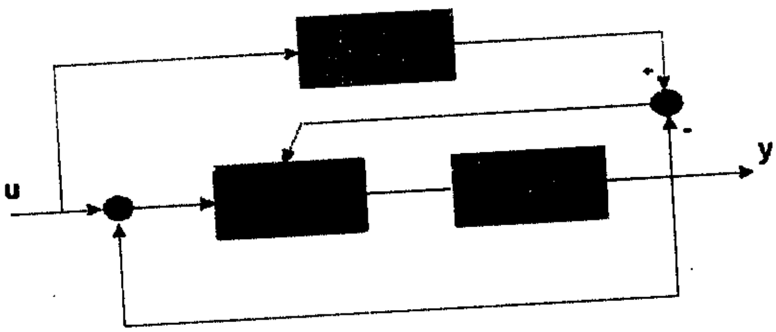
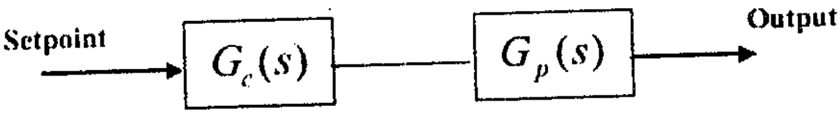


Fig.4.2.Model reference control

4.3DIRECT INVERSE CONTROL

The advantage of using inverse control over supervised control is that inverse control does not require an existing controller in training. Inverse control utilizes the inverse of the system model. The diagram 4.3 is a simple example of direct inverse control. A neural network is trained to model the inverse of the process. When the inverse controller is cascaded with the process the output of the combined system will be equal to the set point. The inverse nonlinearities in the controller cancel out the nonlinearities in the process. For the nonlinearities to be effectively cancelled, the inverse model must be very accurate.



$$G_c(s) = [G_{model}(s)]^{-1}$$

Fig.4.3.Direct inverse control

To generate the inverse of the process Inverse modeling can be used. The system output is used as an input to the network as shown in Fig.4.4. The ANN output is compared with the training signal (the system input) and this error signal is used to train the network. This training method will force the neural network to represent the inverse of the system.

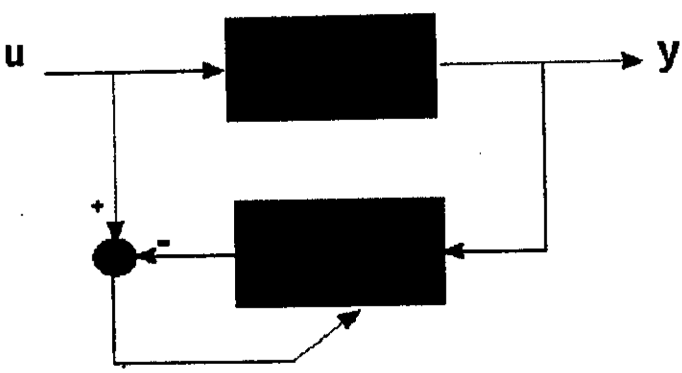


Fig.4.4.Inverse modeling of a process

There are certain problems associated with direct inverse control. In the case of the inverted pendulum, the process may not be invertible. The inverted pendulum is open-loop unstable, the training data would not show the dynamics of the system as the pendulum falls over quickly. There also can be process-model mismatches. The training of the ANN for an inverse model might lead to the model being not strictly proper. This will lead to unknown disturbances in the system.

4.4INTERNAL MODEL CONTROL

Internal model control is based on direct inverse control. A control device consisting of the controller and model of the process characterizes internal Model control systems. The internal model loop computes the difference between the outputs of the process and of the internal model control. This difference represents the effect of disturbances and of a mismatch of the model. Internal model control devices have shown to have good robustness properties against disturbances and model mismatch in the case of a linear model of the process. Developments of internal model control in the case of nonlinear models of the process have proposed, mainly for continuous-time models, but also for discrete-time control systems

Internal model control characteristics are the consequence of the following properties:

(a) If the process and the controller are (input-output) stable, and if the IM is perfect, then the control system is stable.

(b) If the process and the controller are stable, if the IM is perfect, if the controller is the inverse of the IM, and if there is no disturbance, then perfect control is achieved.

(c) If the controller steady-state gain is equal to the inverse of the IM steady-state gain, and if the control system is stable with this controller, then offset-free control is obtained for constant set points and output disturbances.

The problems associated with direct inverse control such as process-model mismatch are reduced using internal model control. The diagram 4.5 shows the set-up of internal model control.

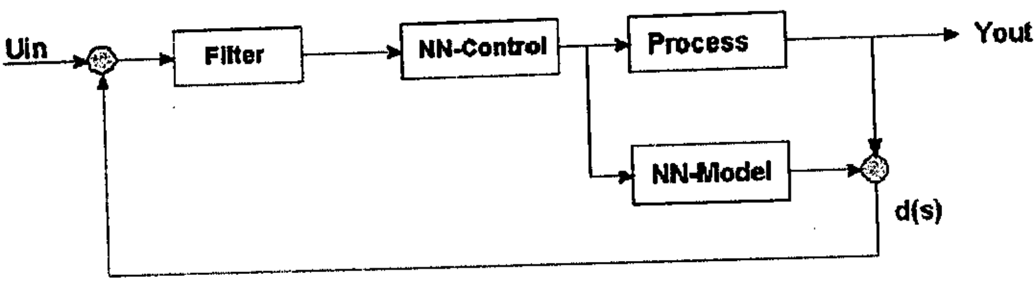


Fig.4.5.Internal model control

A neural network model is placed in parallel with the real system. The controller is an inverse model of the process. The filter makes the system robust to process-model mismatch. With the internal model control scheme, the aim is to eliminate the unknown disturbance affecting the system. The difference between the process and the model d(s) is determined. If the ANN model is a good approximate of the process then the d(s) is equal to the unknown disturbance. The signal d(s) is the information that is missing from the NN-model and can be used to improve the control. The d(s) signal is subtracted from the input set point Uin. In theory, using this method it is possible to achieve perfect control.

4.5UNSUPERVISED CONTROL

Unsupervised learning uses no external teacher and is based upon only local information. It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their emergent collective properties. Paradigms of unsupervised learning are Hebbian learning and competitive learning. From Human Neurons to Artificial Neuron Esther aspect of learning concerns the distinction or not of a separate phase, during which the network is trained, and a subsequent operation phase. We say that a neural network learns off-line if the learning phase and the operation phase are distinct. A neural network learns on-line if it learns and operates at the same time. Usually, supervised learning is performed off-line, whereas unsupervised learning is performed on-line.

Unsupervised learning has less computational complexity and less accuracy than supervised learning algorithm. Unsupervised learning can be designed to learn rapidly and this make it practical in many high speed, real time environment where we may not have enough time and information to apply supervised techniques. Unsupervised learning is used for scientific discovery. Unsupervised learning involves more parameter tuning than supervised learning.

In unsupervised learning set-up, no existing controllers can be imitated and the ANN doesn't have a target to compare to its output. The ANN must try different states and determine which state produces a good output. Learning from experience during periods of no performance feedback is difficult.

Anderson et al developed an unsupervised controller for an inverted pendulum system. Modifications to this controller are based on a failure signal. The failure signal occurs when the pole falls past a certain angle or when the cart reaches the end of the track. A long period of the pole being inverted can occur before the failure signal occurs and the controller must decide which actions in the sequence contributed to the failure. The graph below 4.6 shows the results of the unsupervised controller.

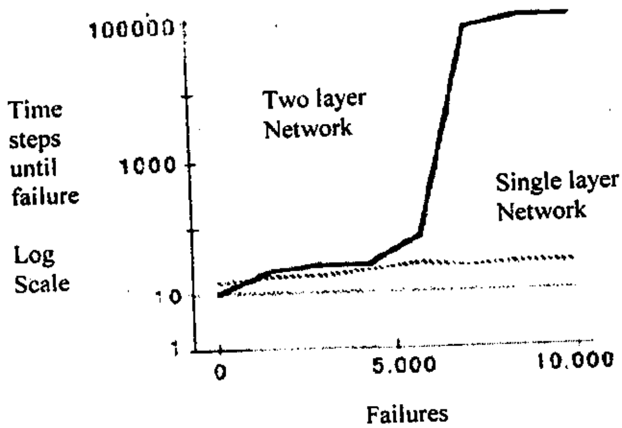


Fig.4.6.Graph of Unsupervised controller

Thus the learning time in unsupervised control is very high but the unsupervised ANN can deal with uncertainty and the complexities of nonlinear control.

To determine the type of neural control to be used, the pros and cons of each of the control methods was determined specifically relating to the pendulum system. To develop a supervised neural controller for the inverted pendulum an existing controller is required. A nonlinear controller has already been developed using feedback linearization for the inverted pendulum. This controller could be used as a teacher. The main disadvantage with this method is the neural controller is based on a control law. The control law can only effectively control the inverted pendulum model developed earlier. If this controller were applied to a more complex pendulum model the controller would fail to keep the pendulum stable.

Inverse control and internal model control are both based on developing an accurate inverse model of the inverted pendulum system. The problem with this method is the pendulum is open loop unstable. To develop an inverse model, the pendulum must be stabilized using a controller. When a feedback controller is used, the inverse model contains some of the dynamics of the controller. The inverse model developed using a feedback controller would never be accurate enough to be used in direct inverse control. This type of neural control is suited to robotic applications and control of stable open-loop systems. The unsupervised control of the inverted pendulum developed by Anderson is the only neural control method that does not require some sort of existing controller for training. The results using unsupervised learning are promising and show that it's possible for the controller to 'learn' to keep the pendulum upright.

CHAPTER 5

REAL TIME IDENTIFICATION AND CONTROL

The inverted pendulum rig consists of a simple cart, which runs along a track. The cart is restricted to traveling in the track axis. The position of the cart is controlled by a DC motor and drive belt. A pole with mass on the end is pivoted on the cart and is free to swing in the same axis. The outputs from this system are the position of the cart along the track and the angle of the inverted pendulum. These are both measured using optical encoder sensors. The two output signals are sent to a control algorithm in MATLAB via a data acquisition card. The control algorithm determines a control action to keep the pendulum inverted.

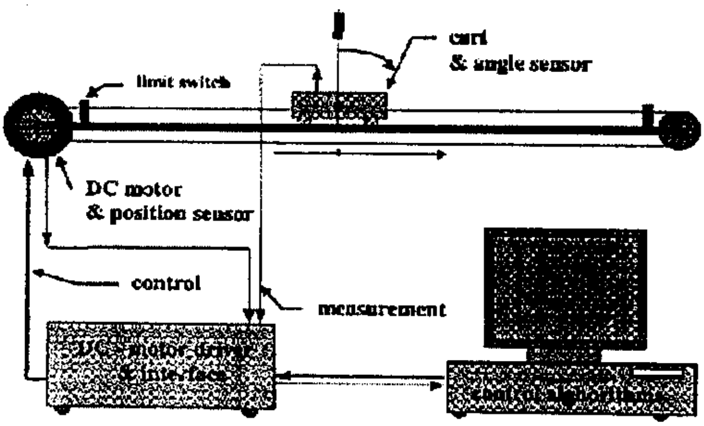


Fig.5.1.Inverted Pendulum Controlled by DC motor

The PID controllers are used to stabilize the inverted pendulum angle and also to control the movement of the cart.

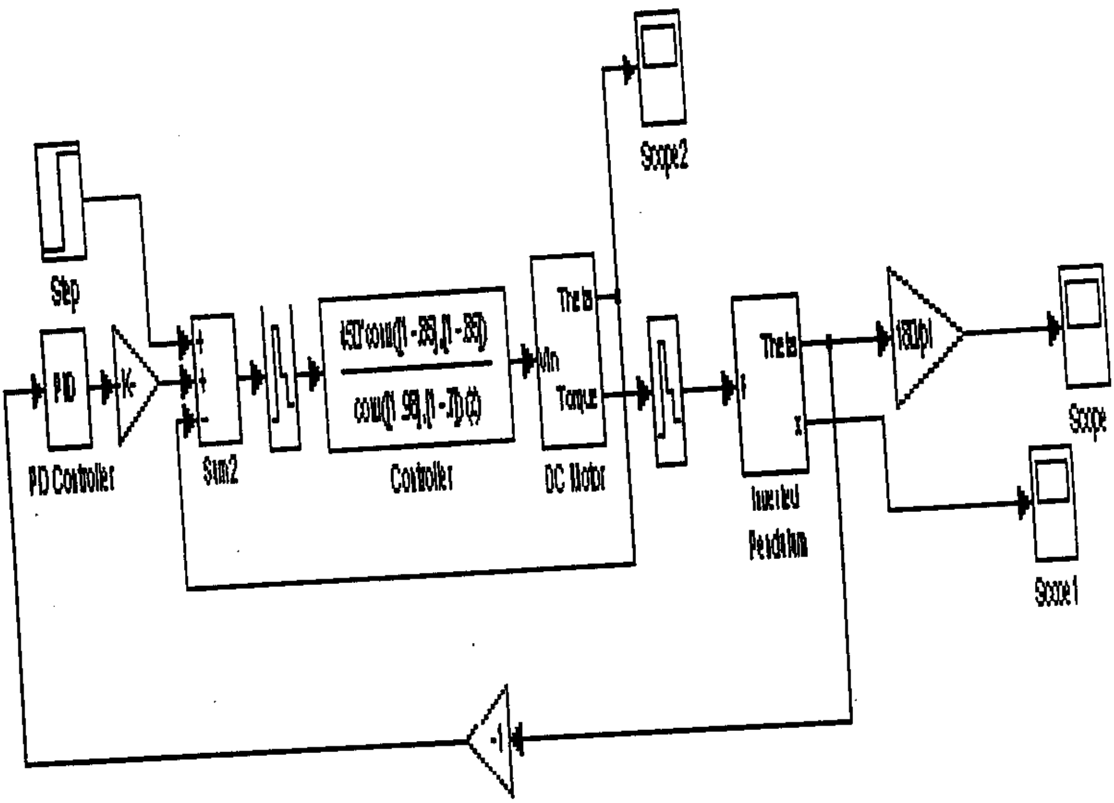


Fig.5.2.PID Controller

Thus the simulink model of an inverted pendulum controlled by DC motor is modeled and simulated with PID is shown in Fig.5.2 It is observed that if there is a fall in the angle of inverted pendulum it has to be tuned manually in Fig.5.3

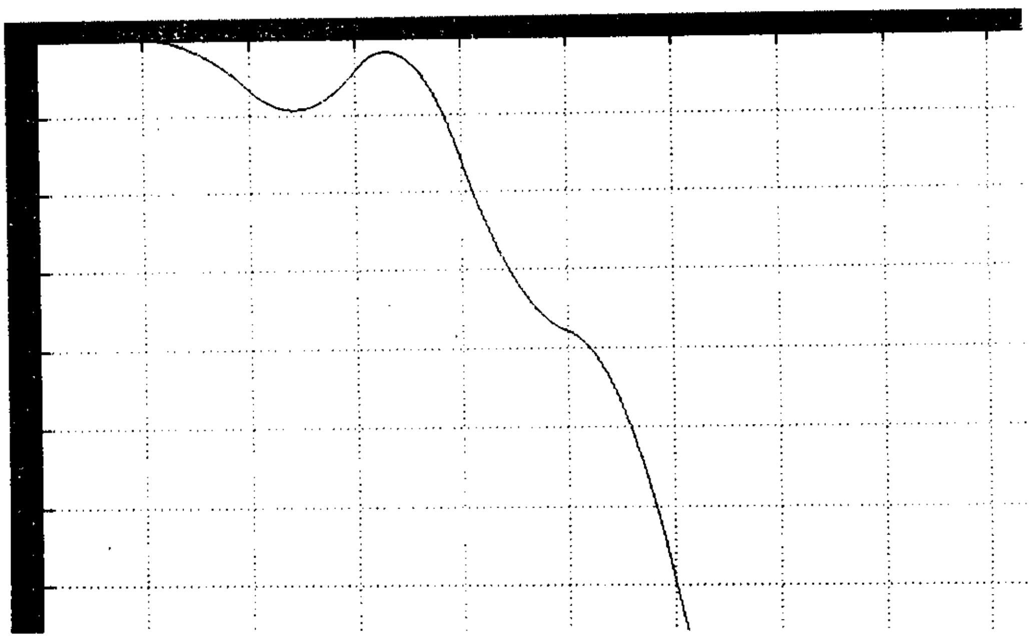


Fig.5.3. Response of Plant Using PID

In order to improve the performance of the system Feed forward Neural Network (FNN) is used. Since the neural network is a good candidate for nonlinear system control.

The back-propagation algorithm has emerged as the workhorse for the design of a special class of layered feed forward networks known as multilayer perceptrons (MLP).A multilayer perceptron has an input layer of source nodes and an output layer of neurons (i.e., computation nodes); these two layers connect the network to the outside world. In addition to these two layers, the multilayer perceptron usually has one or more layers of hidden neurons, which are so called because these neurons are not directly accessible. The hidden neurons extract important features contained in the input data. Thus the structure used is shown in Fig.5.4

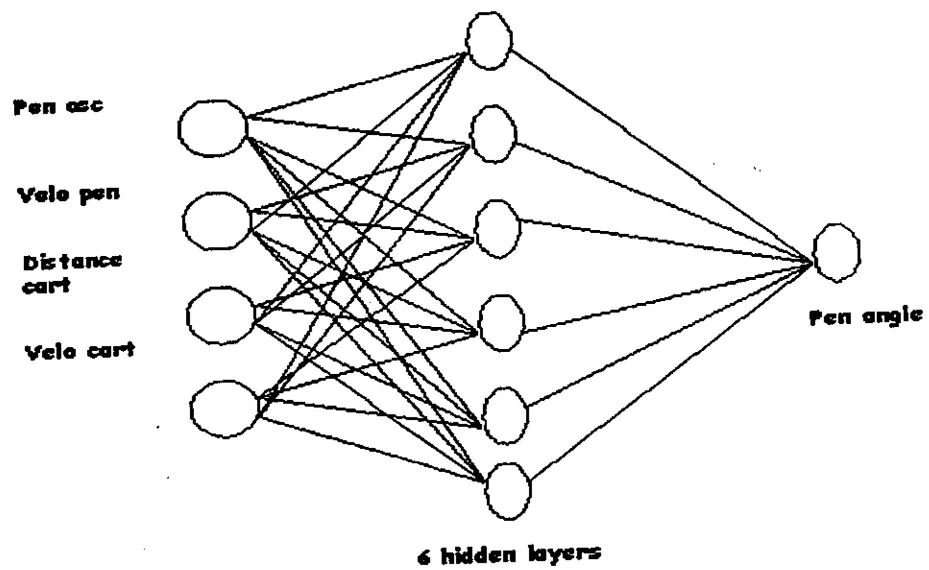


Fig.5.4. Structure of the network

The main advantage of neural networks is that it is possible to train a neural network to perform a particular function by adjusting the values of connections (weights) between elements. For example, if we want to train a neuron model to approximate a specific function, the weights that multiply each input signal will be updated until the output from the neuron is similar to the function.

Artificial neural networks (ANN) have memory. The memory in neural networks corresponds to the weights in the neurons. Neural networks can be trained offline and then transferred into a process where adaptive learning takes place. In our case, a neural network controller could be trained to control an inverted pendulum system offline say in the simulink environment. After training, the network weights are set. The ANN is placed in a feedback loop with the actual process. The network will adapt the weights to improve performance as it controls the inverted pendulum system.

Neural networks can have several layers. In our system we consider two layers. The first layers (hidden) consist of six neurons and the second layer (output) consists of one neuron.

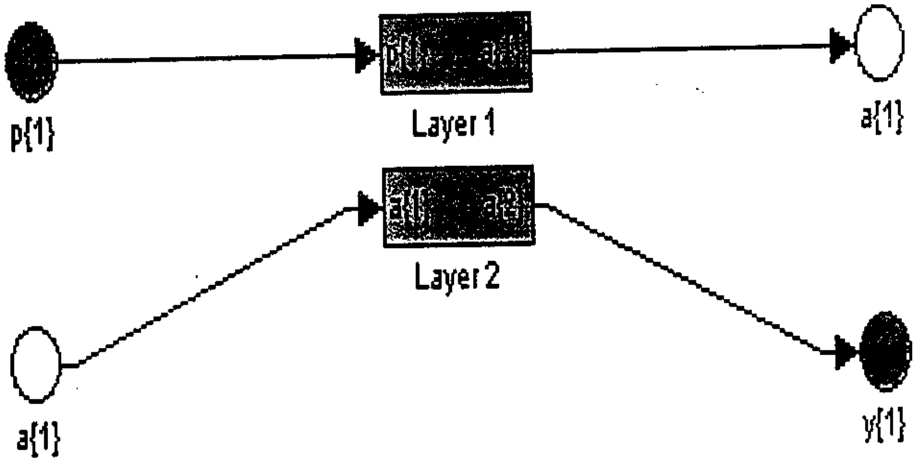


Fig.5.5 Layers of the Network

The four inputs (x, xdot, tetha, tethadot) signals are stored in matlab. The target for the neural network is the output from the controller. The four input signals and the target output are exported to the matlab workspace. The following matlab code trains the neural network. The first section of code generates the 'cell array'. The cell array combines the 4 different inputs into 1 input vector. The FF network has 6 neurons in the hidden layer. The activation functions in the hidden layer are tan-sigmoid and the output layer is a linear function.

[Input1]

Name = 'in1'

Range = [-0.3 0.3]

NumMFs = 2

MF1='in1mf1':'gbellmf',[0.3 2 -0.3 0]

MF2='in1mf2':'gbellmf',[0.3 2 0.3 0]

Similarly for all the inputs the codes are written in MATLAB. The error signal, between the process output and the ANN output is adjusted to the weights of the MLP.

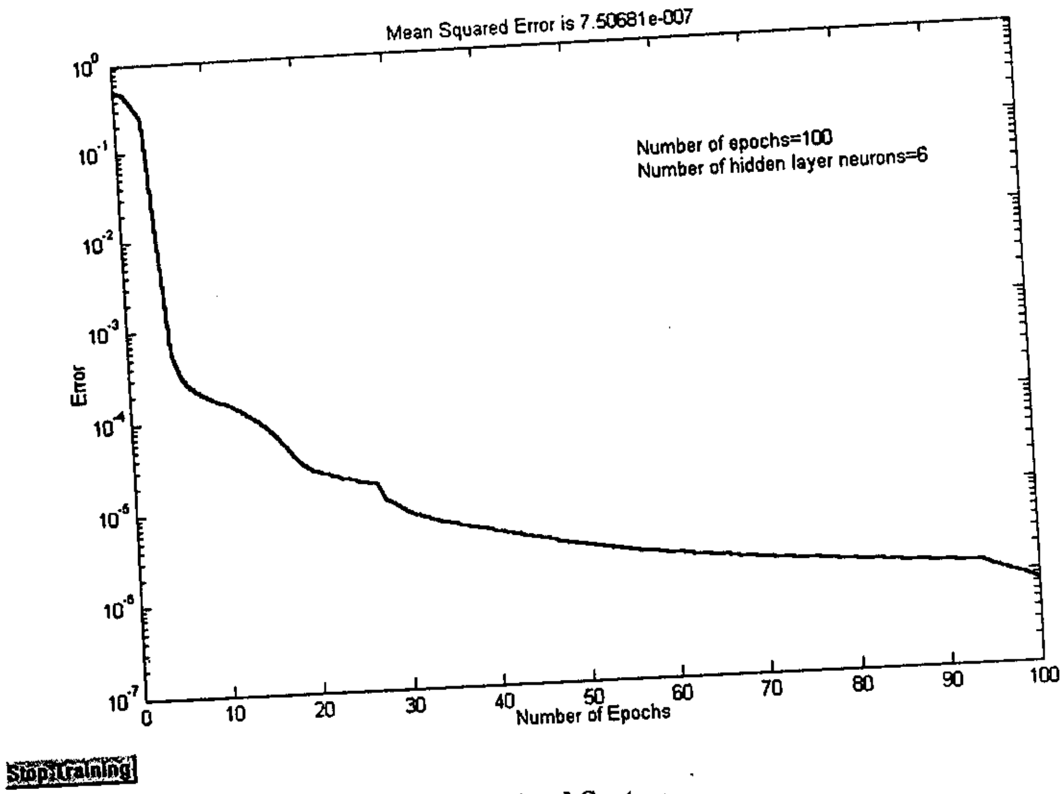


Fig.5.6 Trained System

When the training is finished, the weights are set and a simulink ANN is generated.

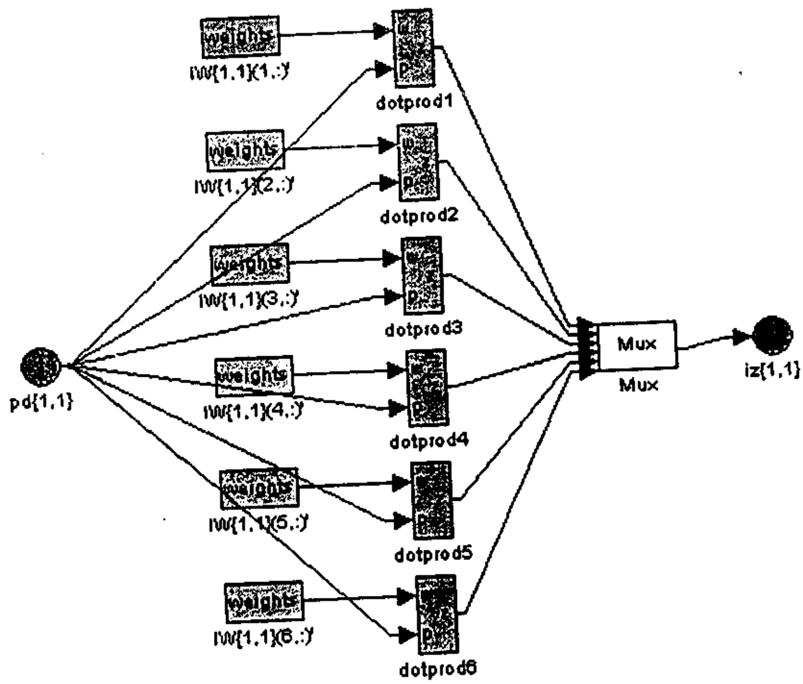


Fig.5.7 Training of the system with the weight set for a single input

Similar to the PID controller, neural network is also simulated in MATLAB. The fig.5.8., shows the simulated block as,

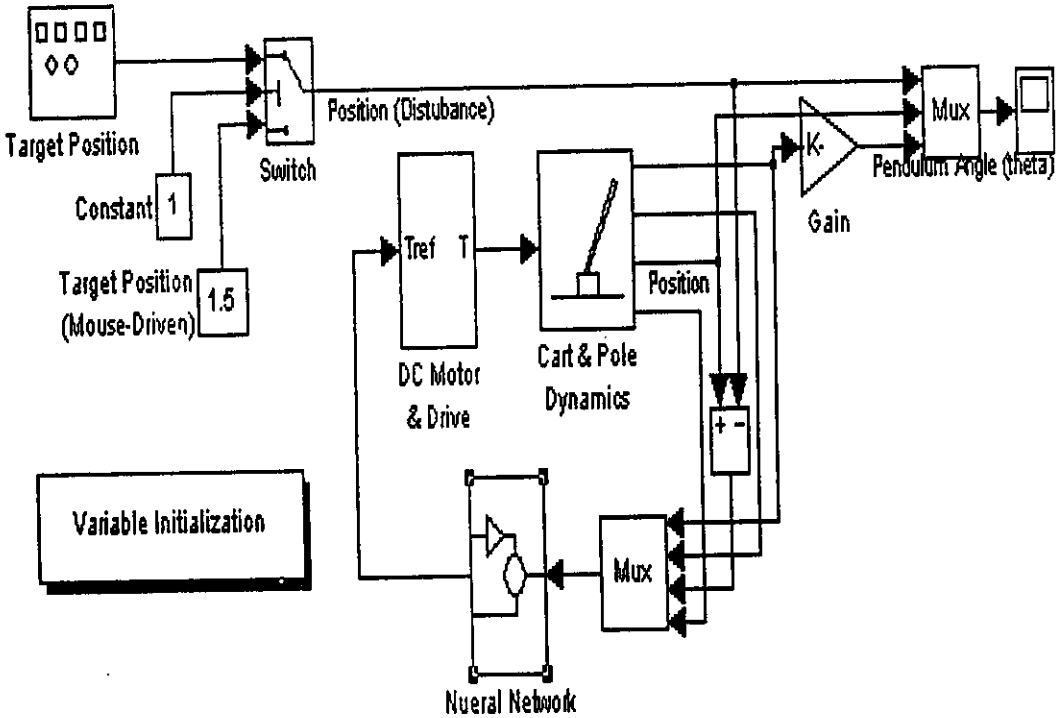


Fig.5.8. Simulink Model Using ANN

After simulating the above block its observed that the inverted pendulum angle does not fall and its maintained within ±15° as the position of the cart is also maintained with help of DC motor and the motor run nearer to zero speed so that the cart position and pendulum angle are stabilized. The output waveforms are as shown in Fig.5.9.

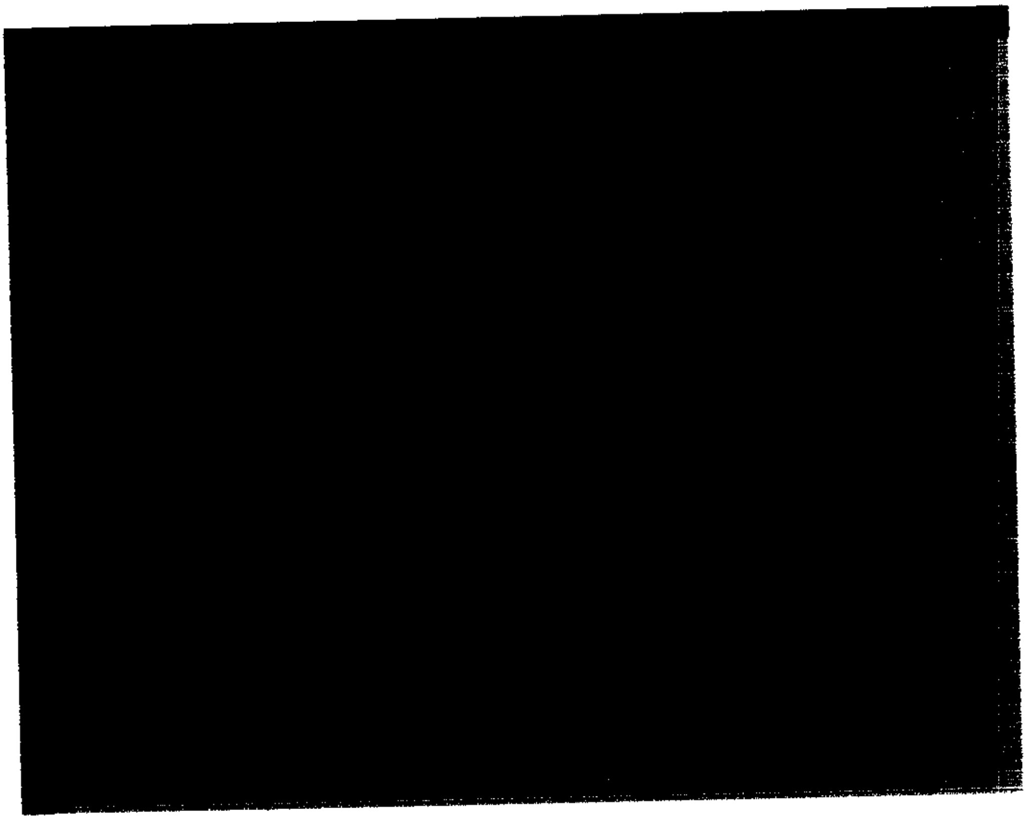


Fig.5.9. Response of the System using ANN

CHAPTER 6

HARDWARE DESCRIPTION

6.1. STRUCTURE

The base platform is made of metal and this is fabricated mainly for the stability purpose of the system. In the base platform two support clamps of the conveyor are attached and this helps the conveyor to run. The motor and the conveyor are connected through a 1:2 gear ratio.

The motor used is permanent magnet DC motor. The specification of the motor as follows:

Operating Voltage - 12V
Operating Current - 2A
Watts - 25W
Revolution per Minute - 90RPM
Gear Box Type - Worm Gear

6.2. MICROCONTROLLER

The microcontroller is the core of the project. The microcontroller has internal memory to store the code lock number. It reads the keyboard value and compares the number stored in the memory; if both the keyboard input and the memory of the microcontroller are equal the micro controller gives the output signal.

The keyboard is input device to the microcontroller to enter the code lock number. It consists of 10 push buttons. i.e. push to on type. If the switch is closed a +5V signal is given to the microcontroller.

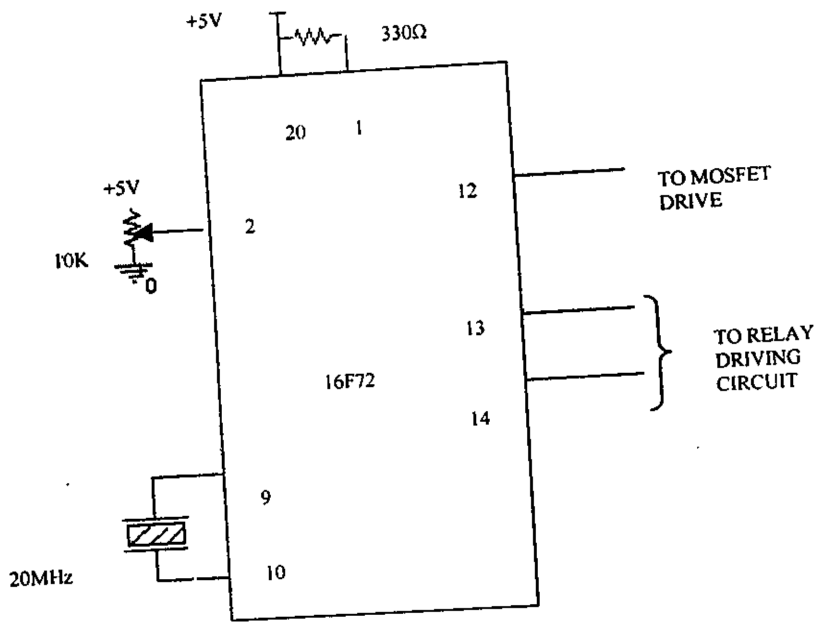


Fig.6.1.Microcontroller

The oscillator is formed with a crystal and capacitors to generate the resonant frequency. The microcontroller program counter increments for each 12 clock cycles, because the c51 core set takes 12 clock cycles to execute a command.

6.3. CONTROL CIRCUIT

6.3.1. Regulated Power Supply

The power supply circuit consists of:

1. Transformer:

The transformer is used to step-down the voltage from 230vac to 12vac and it is designed to give a supply current of 500mA.

2. Bridge rectifier:

The bridge rectifier is used to convert the 12vac to 12vdc supply. The full wave bridge rectifier is used to convert the ac supply into pulsating dc supply.

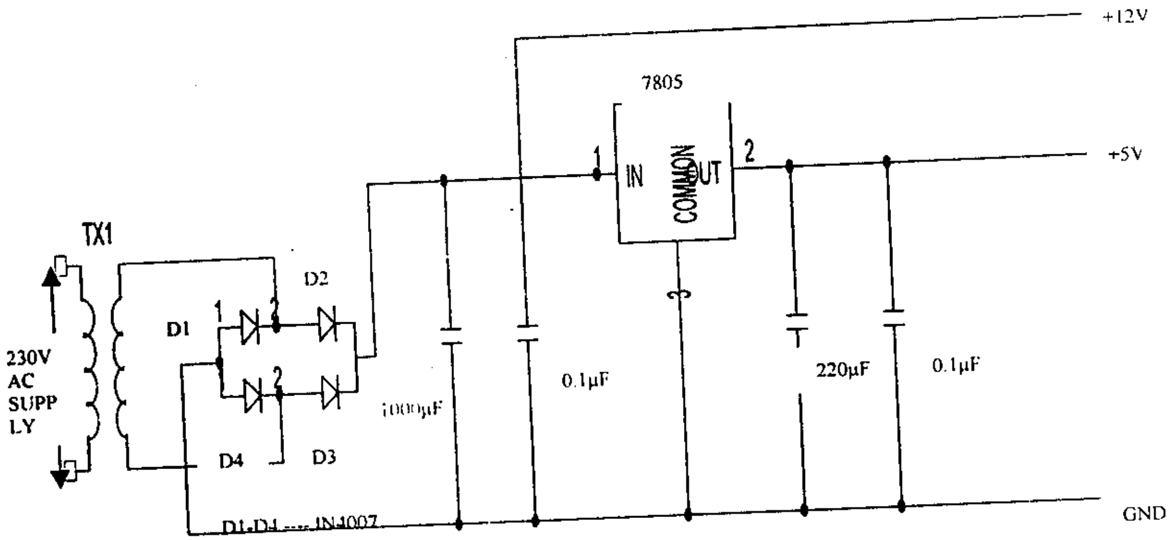


Fig.6.2.Regulated Power Supply

3. Smoothing circuit:

The output from the bridge rectifier is of pulsating ac voltage, which is smoothed by the smoothing circuit. The smoothing circuit consists of capacitors to remove ripples from the dc supply line.

4. Regulator:

The regulator is a 3 terminal solid device. The 12vdc input is given to the regulator and the regulator output is fixed to 5v which will not exceed even tough there is change in the input supply voltage. The output of the regulator is given to the circuits.

An unregulated input voltage Vi is filtered by capacitor C1 and connected to the IC's IN terminal. The IC's OUT terminal provides a regulated + 12V which is filtered by capacitor C2 (mostly for any high-frequency noise). The third IC terminal is connected to ground (GND). While the input voltage may vary over some permissible voltage range, and the output load may vary over some acceptable range, the output voltage remains constant within specified voltage variation limits. These limitations are spelled out in the manufacturer's specification sheets. A table of positive voltage regulated ICs is provided in table.6.1

IC Part	Output Voltage (V)	Minimum Vi (V)
7805	+5	7.3
7806	+6	8.3
7808	+8	10.5
7810	+10	12.5
7812	+12	14.6
7815	+15	17.7
7818	+18	21.0
7824	+24	27.1

TABLE 6.1 Positive Voltage Regulators in 7800 series

6.3.2. MOSFET Drive

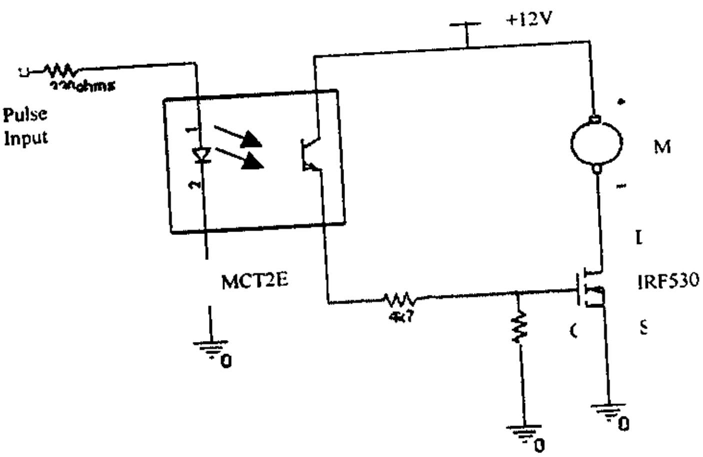


Fig.6.3.MOSFET Driver Circuit

The proper isolation between input and output becomes very important in several of the digital and analog applications. The traditional methods isolation involves the use of such devices as capacitor, relays, transformers and optocouplers of these the Optocoupler provide an ideal combination of speed, OC response, high common mode nose for both analog and digital applications in industrial, medical and military products. Examples are logic isolations, line receivers, sensing circuits, power supply feed back, high voltage current monitoring, telephone, lines patient monitoring equipments, adaptive control, audio and video amplifiers, triggering of thyristors, and so on.

Optocoupler consists of an LED emitter and a photo sensor of transistor or diode type. The one that utilizes a phototransistor is limited in its improves the speed up to a hundred times that of a phototransistor coupler by reducing the base-collector capacitance. Al high speed Optocoupler employs this as principle of operation.

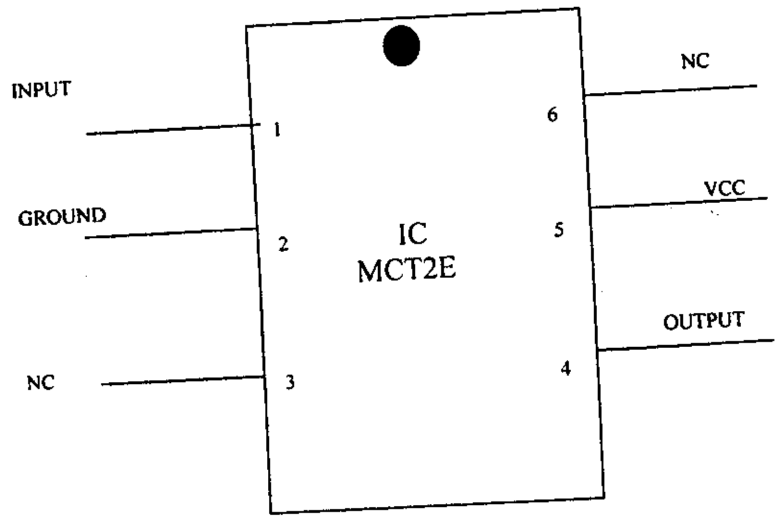


Fig.6.4. Pin Diagram of IC MCT2E

3.3. Relay Driving Circuit with Optocoupler

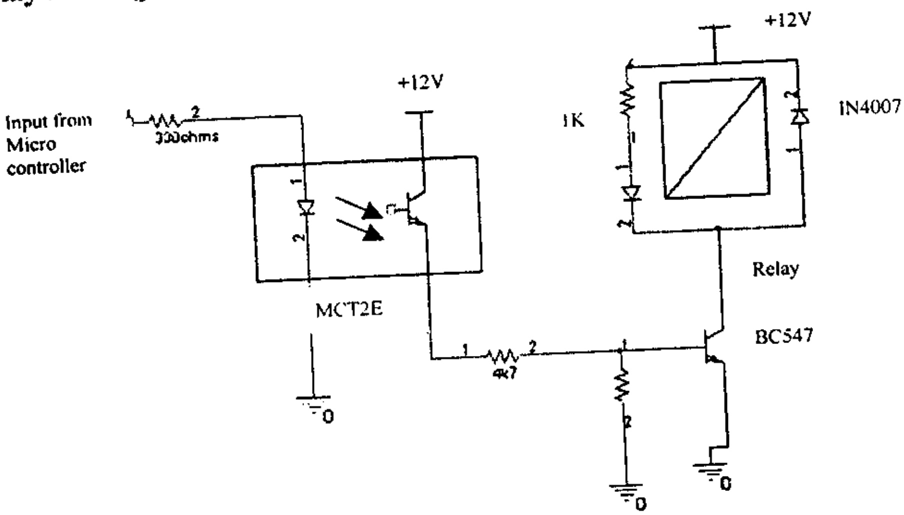


Fig.6.5.Relay Driving Circuit with Optocoupler

Relay Driver:

The relay driver is used to drive the relay. It is connected to the circuit. The relay is used to switch the output load on/off. The relay consumes higher current around 100mA, and the microcontroller gives an output of only 20mA maximum. So a transistor driver is connected in between the microcontroller and the relay to turn the relay on/off.

Relay:

The relay is an electro mechanical device, which is used to isolate control circuits and the power circuit and also to switch higher power loads such as door control motors, solenoid coils etc. The freewheeling diode is provided to protect the relay from reverse current flow.

BC547:

The BC 547 is a switching transistor. The transistor is connected to the relay, when the transistor saturates, the relay is grounded. A pull up transistor is connected to the transistor to prevent the external noises from triggering the transistor.

CHAPTER 7

CONCLUSION

The technique of artificial neural network to the identification and control of the inverted pendulum controlled by DC motor is applied. Before identification techniques could be tested, a model representing the inverted pendulum was developed in simulink. One of the requirements for accurate identification is experimental input-output data that shows the dynamics of the system.

Initially single-input single-output networks were developed, the input being the control force and the output being the inverted pendulum angle. The first types of neural network to be developed are feedforward. Feedforward networks with a range of hidden layer neurons were tested. In open-loop identification, increasing the number of hidden layer neurons will have a direct influence on the accuracy of the model. The feedforward network was modeled for the inverted pendulum controlled by DC motor. Thus the simulation results are verified and the pendulum angle and cart positions are stabilized.

The main task in the project was to design a controller, which keeps the pendulum system inverted. The four main types of neural control (supervised, unsupervised, direct inverse and internal model control) were studied to determine which control technique would be the most efficient to implement. The earliest application of neural networks to the inverted pendulum is by Widrow and Smith and Widrow . They used traditional control methods to derive a control law to stabilize the linearized system. They then trained a neural network to mimic the output of the control law. It was decided that supervised control would be the least complex to implement. It was not possible to develop direct inverse control because this control method requires that the process to be controlled is already open-loop stable. The unsupervised control technique developed by Anderson was just too complex for the project time frame. The first neuro-controller was developed by training a feedforward network to model the control law. Elman networks were also studied to model the control law but were not as accurate. When the training was finished the neural network was exported into simulink and the network was placed in the feedback loop instead of the existing control law. The neural network controlled the inverted pendulum similar to the control law. The ANN was trained offline. The advantage of using this type of network is if a disturbance occurs during operation, the error signal is fed back into the Adaline, which adjusts the weights of the network, and this counteracts the disturbance.

It was decided to test some of the identification and control techniques on the real time inverted pendulum rig. The real-time inverted pendulum is also open-loop unstable. It was not possible to develop a neural controller for the real time system but significant progress was made.

APPENDICES

```
#include <pic.h>
#include <stdio.h>

unsigned int pwm;
unsigned char lsb;

unsigned char adc_read(unsigned char channel);
void out_pwm1();
void Hardware_Setup( void );

main()
{
        Hardware_Setup();

        while(1)
        {
                pwm = adc_read(0) << 2;
                out_pwm1();
        }
}

void Hardware_Setup( void )
{
/* PORT DECLERATION */
        TRISC = 0x80;
        TRISA = 0XFF;

/* ADC INTILISATION */
        ADCON1 = 0X80;

/* PWM Intilisation */
        OPTION = 0x03;
        PR2 = 0xFF;
        T2CON = 0x06;
        CCP1CON = 0x0C;
}

unsigned char adc_read( unsigned char channel )
{
```

```
        unsigned char del;
        ADCON0 = ( channel << 3 ) + 0x81;
        for( del = 100; del--; );
        ADGO = 1;
        while( ADGO );
        return( ADRES );
}

void out_pwm1()
{
        lsb=*(((unsigned char*)&pwm));
        lsb = lsb & 0x03;
        lsb = lsb << 4;
        CCP1CON = CCP1CON & 0xCF;
        CCP1CON = CCP1CON | lsb;
        pwm = pwm >> 2;
        CCPR1L=*(((unsigned char*)&pwm));
}
```

PIC16F72

28-Pin, 8-Bit CMOS FLASH MCU with A/D Converter

Device Included:

- PIC16F72

High Performance RISC CPU:

- Only 35 single word instructions to learn
- All single cycle instructions except for program branches, which are two-cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- 2K x 14 words of Program Memory, 128 x 8 bytes of Data Memory (RAM)
- Pinout compatible to PIC16C72/72A and PIC16F872
- Interrupt capability
- Eight-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

Peripheral Features:

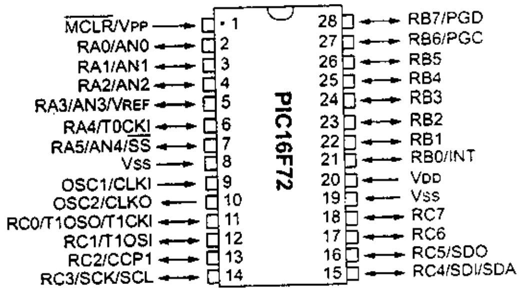
- High Sink/Source Current: 25 mA
- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Capture, Compare, PWM (CCP) module
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 8-bit, 5-channel analog-to-digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master/Slave) and I²C™ (Slave)
- Brown-out detection circuitry for Brown-out Reset (BOR)

CMOS Technology:

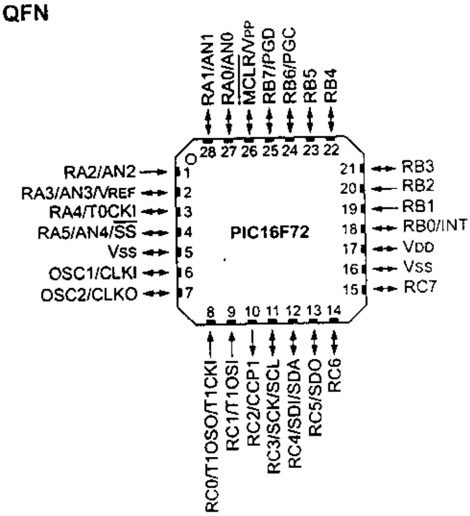
- Low power, high speed CMOS FLASH technology
- Fully static design
- Wide operating voltage range: 2.0V to 5.5V
- Industrial temperature range
- Low power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 μA typical @ 3V, 32 kHz
  - < 1 μA typical standby current

Pin Diagrams

PDIP, SOIC, SSOP



QFN



Special Microcontroller Features:

- 1,000 erase/write cycle FLASH program memory typical
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- In-Circuit Serial Programming™ (ICSP™) via 2 pins
- Processor read access to program memory

Key Reference Manual Features	PIC16F72
Operating Frequency	DC - 20 MHz
RESETS and (Delays)	POR, BOR, (PWRT, OST)
FLASH Program Memory - (14-bit words, 1000 E/W cycles)	2K
Data Memory - RAM (8-bit bytes)	128
Interrupts	8
I/O Ports	PORTA, PORTB, PORTC
Timers	Timer0, Timer1, Timer2
Capture/Compare/PWM Modules	1
Serial Communications	SSP
8-bit A/D Converter	5 channels
Instruction Set (No. of Instructions)	35

1.0 DEVICE OVERVIEW

This document contains device specific information for the operation of the PIC16F72 device. Additional information may be found in the PICmicro™ Mid-Range MCU Reference Manual (DS33023), which may be downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The PIC16F72 belongs to the Mid-Range family of the PICmicro devices. A block diagram of the device is shown in Figure 1-1.

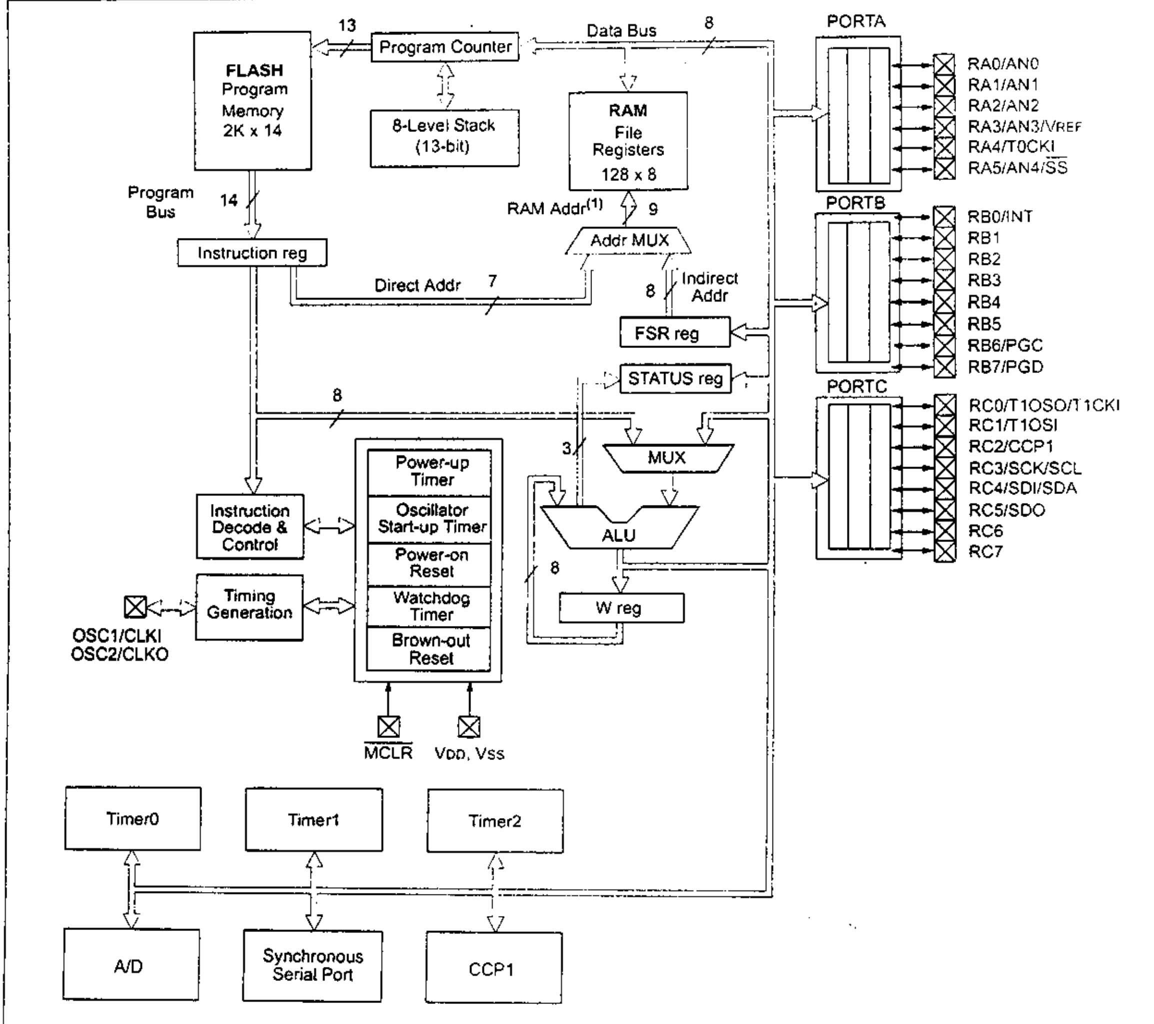
The program memory contains 2K words, which translate to 2048 instructions, since each 14-bit program memory word is the same width as each device instruction. The data memory (RAM) contains 128 bytes.

There are 22 I/O pins that are user configurable on a pin-to-pin basis. Some pins are multiplexed with other device functions. These functions include:

- External interrupt
- Change on PORTB interrupt
- Timer0 clock input
- Timer1 clock/oscillator
- Capture/Compare/PWM
- A/D converter
- SPI/I^{2}C

Table 1-1 details the pinout of the device with descriptions and details for each pin.

FIGURE 1-1: PIC16F72 BLOCK DIAGRAM



Note 1: Higher order bits are from the STATUS register.

NCEIS 2005

DIAMOND JUBILEE YEAR

Government College of Technology

Coimbatore 641 013

Tamil Nadu

~~Dr.~~/~~Prof.~~/~~Mr.~~/Ms. D. CHITRA

of Kumaraguru College of Technology, Coimbatore

has presented a paper titled Identification and control of dynamical systems using neural networks

in the **National Conference on Modern Trends in Electrical and Instrumentation Systems** *organised by the Electrical and Electronics & Electronics and Instrumentation Engineering Departments on*

9th & 10th March, 2005.

TECHNICAL CHAIR

ORGANISING CHAIR

PATRON

REFERENCES

1. Andreas Idebrant & Jan Brugård. (2004)' Inverted Pendulum', MathCore AB.

2. C.W. Anderson. (1992) 'Learning to control an inverted pendulum using neural networks', IEEE Controls Systems Magazine, 9:31-37.

3. B.L.walcott , M.J.Corless, and S.H.Zak (1987) 'Comparative Study of non linear State observation Techniques', Int. J. Contr., Vol 45, pp 2109-2132.

4. Hunt and Sbarbaro. (1992) 'Neural Networks for Control System - A Survey', Automatica, Vol. 28, pp1083-1112.

5. John G.Kuschewski, and stefen Hui. (1993) 'Application of Feedforward Neural Networks to Dynamical System Identification and Control', IEEE Trans on Control Systems, Vol.1,No.1, pp 37-49.

6. J.J.Hopfield. (2000) 'Neural Network and physical Sysems with emergent collective Computational Abilities', Proc.Nat.Acad.Sci, Vol.79, pp 2554-2558.

7. J.G. Kuschewski, S. Hui, and S.H. Zak. (2001) 'Application of feedforward neural networks to dynamical system identification and control', IEEE Trans. Contr. Syst.Technol. Vol. 1, No. 1, pp. 37-49.

8. M.Kawato, Y.Uno, M.Isobe and R.Suzuki (1999), 'Hierarichal Neural network Model for Voluntary Movement with Application to Robotics', IEEE Contr. Syst. Mag., Vol 8, pp 8-16.

9. Seul Jung and Hyun Taek Cho. (2004) 'Decoupled Neural Network Reference Compensation Technique for a PD Controlled Two Degrees-of-Freedom Inverted Pendulum', International Journal of Control, Automation, and Systems Vol. 2, No.1

10. B.Widrow and M.A.Lehr (1990), '30 years of adaptive neural networks: Perceptron, madline and backpropogation', Proc. IEEE, Vol 78, pp 1415-1442.