DESIGN AND IMPLEMENTATION OF A NEURAL CONTROLLER FOR PHASE-CONTROLLED DC MOTOR DRIVE

A DISSERTATION

submitted in partial fulfilment of the
requirements for the award of the degree
of
MASTER OF ENGINEERING
in
ELECTRICAL ENGINEERING
(With Specialization in Power Apparatus and Electric Drives)

By

KANCHAN SINGH

DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE-247 667 (INDIA)

MARCH, 1999

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this dissertation entitled **"DESIGN AND IMPLEMENTATION OF A NEURAL CONTROLLER FOR PHASE - CONTROLLED DC MOTOR DRIVE"** submitted in partial fulfilment of the requirement for the award of Degree of **MASTER OF ENGINEERING** with specialization in **POWER APPARATUS AND ELECTRIC DRIVES**, in the Department of Electrical Engg., University of Roorkee, Roorkee is an authentic record of my own work carried out from July 1998 to March 1999 under the guidance of **Dr. Pramod Agrawal**, Asstt. Professor, Electrical Engg. Deptt., University of Roorkee, Roorkee.

Date : 11th March, 1999

Place : Roorkee

Kanchan

(KANCHAN SINGH)

CERTIFICATE

This is to certify that the above statement made by the candidate is true to the best of my knowledge and belief.

Dr. Pramod Agrawal

Asstt. Professor
Deptt. of Electrical Engineering
University of Roorkee
Roorkee, U.P. (India)

ACKNOWLEDGEMENT

It is my privilege to express my profound sense of gratitude and indebtedness to my guide **Dr. Pramod Agarwal**, *Assistant Professor, Department of Electrical Engineering, University of Roorkee, Roorkee* for his meticulous guidance, keen interest, sincere advices, continuous encouragement and kind support throughout my work.

Thanks are also due to Dr. Indra Gupta, Asstt. Professor and Shri M. Vesantha, Professor, Electrical Engg. Deptt., U.O.R., Roorkee for their moral support.

Finally, I am thankful to those who have supported me directly or indirectly during the work.

Kanchan

KANCHAN SINGH

M.E. IInd Year

Power Apparatus & Electric Drives

ABSTRACT

The dc motors have occupied a wide spectrum of applications for variables speed drive. The present work deals with the design and development of concverter fed dc motor drive using fuzzy neural network control. A 3-phase thyristorized phase controlled converter is designed and devleoped. The firing pulses for the thyristors are generated using 8031 microcontroller based card. The firing angle at which the converter is to be operated is transmitted serially from personal computer at 2400 baud rate to microcontroller card. For precise control over wide speed range and to get improved performance of dc motor drive, closed loop operation of system is desired. For closed loop operation speed measurement and error processing at much faster rate is required. This necessitates an accurate and fast acting control of speed. A pulse tachogenerator is used for accurate measurement of speed. The motor current is sensed using Hall effect current sensor. The conventional control (PID) is slow and lack of efficiency in handling system non-linearities. To overcome these limitations fuzzy neural controller has designed and tested. The basic concept behind hybrid controller is that to get the advantages of both fuzzy logic and neural network.

Since the machine system is ill defined fuzzy logic is used for pattern generation and then the ability of data processing in a non-linear system of neural network is used.

For generating patterns from fuzzy logic triangular membership function is used. The method of height is used for defuzzification. Then neural network is trained off-line using these patterns. Error back propagation algorithm is used for training and adjusting the weights of controller. The performance of the converter fed dc motor drive is investigated in open loop and closed loop mode.

NOMENCLATURE

X_i	Inputs to the neural network
Y_i	Output of neural network
α	Coefficient of gain
W_{ij}	Weight matrix
μ	Learning rate
E_p	Energy function
P	Number of training pattern
T_i	Desired target
$Y_i(H)$	Activation corresponding to ith neuron
ν	Momentum gain
R	Fuzzy rule
E	Error
CE	Change in error
ME	Medium
ωm^*	Reference speed
K_c	Controller gain
$\varepsilon_I(\alpha)$	Desired steady state error
A	Converter gain
K_r	Current feedback gain
R_a	Armature resistance
L_a	Armature inductance
K_b	Back emf constant
B	Viscous friction coefficient
J	Moment of inertia

CONTENTS

INTRODUCTION

Direct current (DC) motors have been used in variable speed drives for a long time. The versatile control characteristics of dc motors have contributed to their extensive use in industry. DC motors can provide high starting torque, which are required for traction drives. Control over a large speed range both below and above the rated speed can be easily achieved. The methods of control are simpler and less expensive than those of alternating current (AC) motors. Although commutators prohibit their use in certain applications, such as high speed drives and operation in hazardous atmosphere, dc motors play a significant role in many industrial drives [1].

The dc motors have occupied a wide spectrum of applications for variable speed drive. The speed of DC motor can be controlled by :

(1) Using field control

(2) Using armature control

Certain limitations of field control method are :

(1) Speed lower than the rated speed cannot be obtained because the field cannot be made any stronger.

(2) This control method is not suited to applications needing speed reversal.

Armature control method is superior to the field control scheme in three respects :

(a) It provides constant-torque drive.

(b) Since the main field ampere turns are maintained at a large value, flux density distortion caused by armature reaction is limited

(c) Unlike field control scheme, speed reversal can be easily implemented.

The conventional Ward-Leonard method of controlling speed of a separately excited motor using armature voltage control method is being used extensively in

industrial applications. This method, however has disadvantages namely high initial cost, low efficiency, the requirement of large space as the system is bulky and requirement of frequent maintenance.

Normally closed loop operation, with PI controllers in the inner current loop and outer speed loop is employed for speed control. These controllers are designed and implemented using the values of motor parameters these are armature resistance, armature inductance, viscous friction coefficient and moment of inertia supplied by the manufacturers or the values of parameters obtained through measurements. With these controllers, the response of the system may become unsatisfactory due to changes in parameters caused by ageing or errors in measurements. Adaptive controllers are used to cope up with the parameter variations. They are not advisable when fast response is required as it takes much computation time in calculating the control input.

Using thyristors and electronic control, dc drive system can be controlled efficiently. This system provides improved response and better accuracy.

Thyristor converter provides variable armature voltage for motor drive. The basic methods for obtaining a variable DC output voltage are :

(a) Phase control

(b) Integral cycle control, and

(c) Chopper control

In all these methods, thyristor connect and disconnect the supply to motor terminals. The frequency of switching is rapid, therefore, the motor responds to the average output voltage level and not to the individual voltage pulses.

The author has used a three-phase converter for speed control of dc motor in the present work.

1.1 MICROCOMPUTER CONTROL OF DRIVES

The control of the static power converters can be realised by analog components or by digital devices. A digital control system is free from drift and offset errors and is immune to transients and distortion of line voltages.

The prospects of digital control are further improved by the latest low cost microprocessor and microcomputer systems to improve the performance of drive systems with applicable sophisticated control methods.

Microcomputer based intelligent motion control systems are playing a vital role in today's industrial automation. Today's motion control is an area of technology that embraces many diverse disciplines such as electrical machines, power semiconductor devices, converter circuits, dedicated hardware, signal electronics, control theory and microcomputer [2].

Microcomputers provide significant cost reduction in control electronics, improve reliability and eliminate drift and EMI problems. They also permit design of universal hardware and flexible software control. Software can be updated or altered as system performance demands change.

"Micro" has the powerful capability of complex computation and decision making.

In short, the advantages of digital control can be summarized as below :

- There is significant reduction in the controller hardware.
- Reliability of system improves, as software is more reliable as compared to hardware.
- There is no problem of drift with change in temperature age, wear and tear.
- Electromagnetic interference problem can easily be taken care of.
- Software control provide flexibility, i.e., control philosophy can be changed easily.
- Sophisticated control is possible, i.e., advanced power circuit topologies which are very difficult to implement otherwise can be used.

Micro-control has the disadvantage of signal quantization and sampling delay. It is sluggish as compared to dedicated hardware. One of the main difficulties with conventional tracking controllers for electric drives is their inability to capture the unknown load characteristics over a widely ranging operating point. This makes the tuning of the respective controller parameters difficult.

1.2 CONTROL TECHNIQUES

1.2.1 Self Tuning Adaptive Control

The applications of adaptive control theories, still just beginning, are growing at rapid pace. A conventional PI (proportional-integral) or PID (proportional-integral-derivative) controller with fixed parameters cannot generate optimal response in a plant parameter varying system. In self-tuning adaptive control, the controller parameters are tuned to adapt the plant parameter variation. Such a general control scheme is indicated in Fig. 1.1.

The plant parameter estimation algorithm solves the plant model in real-time and updates the plant parameters on the basis of recursive least square identification techniques. A tuning algorithm then adjusts the regulator parameter based on plant parameters. The tuned system may have pole-assignment controls but dead-beat, state-space, or design of time-series control can also be used. The regular parameters may be updated at a slower rate than the main control loop sampling rate, if the plant parameters vary slowly. For successful operation of the global system, stability is essential [3].

1.2.2 Model Reference Adaptive Control

In a *model reference adaptive control* [MRAC], the plant response is forced to track the response of a reference model irrespective of plant parameter variation. The reference model with fixed parameters is stored in microcomputer memory, and therefore the response of plant becomes insensitive to parameter variation. The speed command generated by the position control loop is applied in parallel to the reference model and plant controller. The reference model output is compared with the measured plant speed, and the resulting error signal actuates the adaptation algorithm.

The feedforward and feedback gains of the plant controller are iterated by the adaptation algorithm so as to dynamically reduce error to zero. The plant can track the reference model without saturation provided the parameters in the reference model are defined on a worst case basis. Therefore, the desired

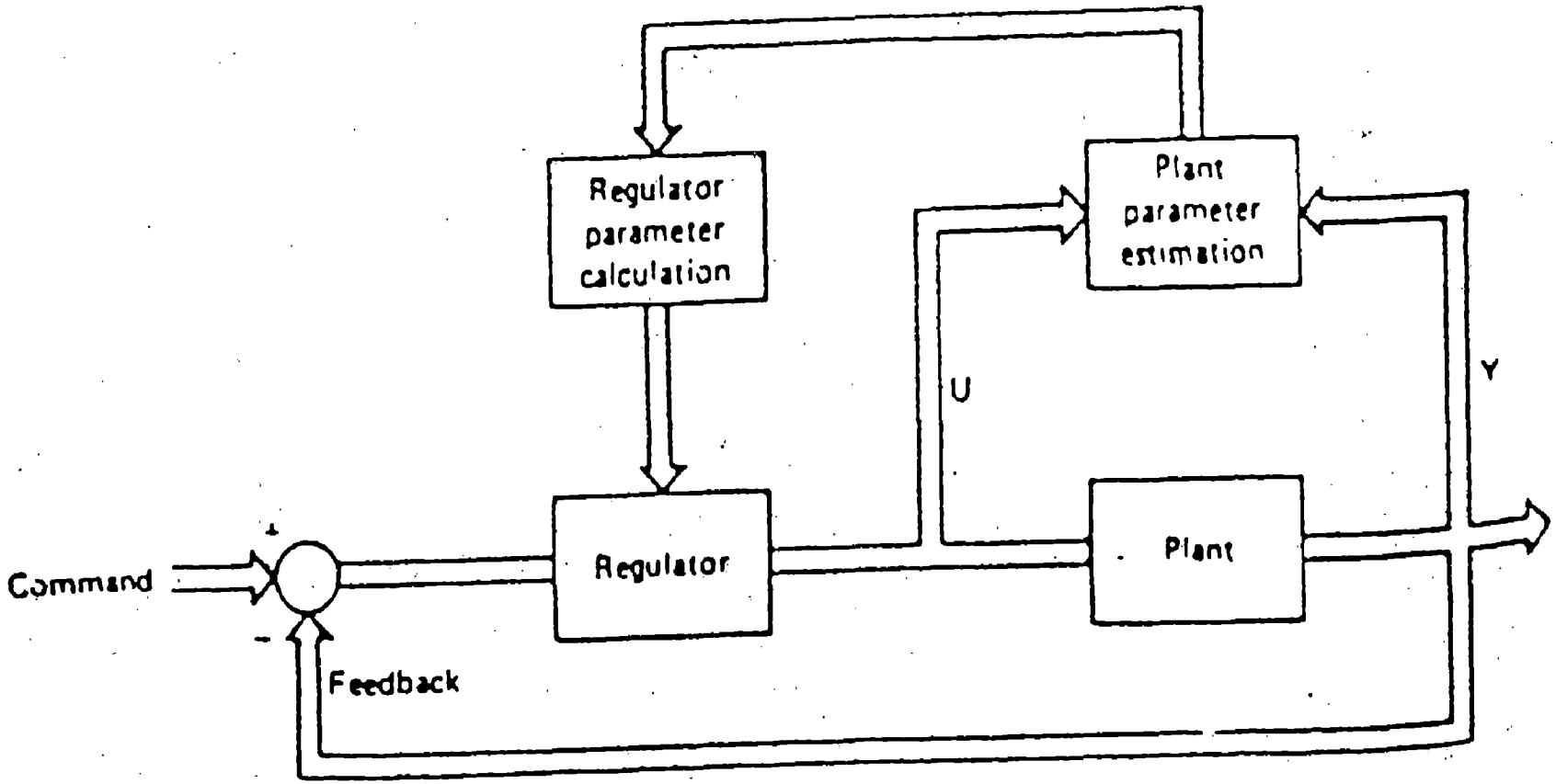


Fig. 1.1 Self tuning regulation of drive system

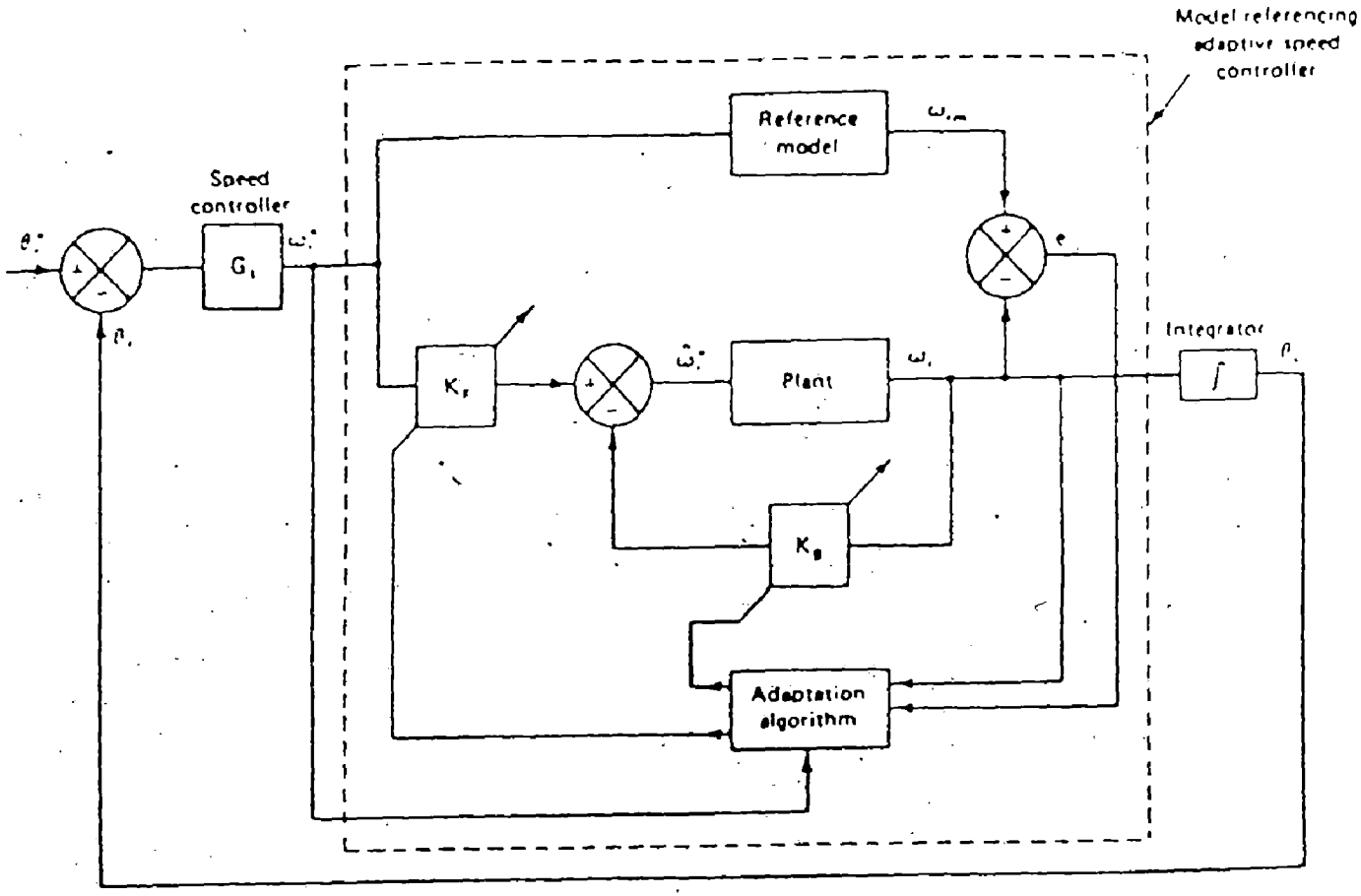


Fig. 1.2 Model referencing adaptive control (MRAC) system

robustness of the control system is obtained at the sacrifice of optimum response speed. In general, the structure of reference model and the plant should be the same, and the parameters should be compatible for satisfactory adaptation. The global stability of the system can be analyzed by Popov's hyperstability theorem (Fig. 1.2) [2].

1.2.3 Sliding Mode Control

A sliding mode or variable-structure control technique has been applied successfully to both dc and ac drive systems. Basically, it is an adaptive model-referencing control [MRAC], but is easier to implement by microcomputer than the conventional MRAC systems. The sliding mode control is ideally suitable for position servo, such as robot and machine tool drives, where problem related to mechanical inertia variation and load disturbance effect can be eliminated. The control can be extended to multiple drives where close speed or position tracking is desired. In sliding mode control, the reference model or a predefined trajectory in the phase is stored in a microcomputer, and the drive system is forced to follow or slide along the trajectory by a switching control algorithm, irrespective of plant parameter variation and load torque disturbance. The microcomputer detects the deviation of the actual trajectory and corresponding changes in the switching topology to restore tracking [4].

Using sliding mode control, the problem of slow response, non-zero errors, sensitivity to variation in controller gains and load changes etc. In the drive can be solved. The sliding mode control improves the dynamic performance of the controlled system (Fig. 1.3).

Artificial Neural Networks (ANNs) have shown great potential for adaptive control applications due to their capability to learn the system non-linear characteristics through non-linear mappings. ANNs provide a distinctive computational paradigm by exploiting the massively parallel processing performed in their elementary processing elements called neurons. Relevant features of the neural networks in the control context include among others their ability to model

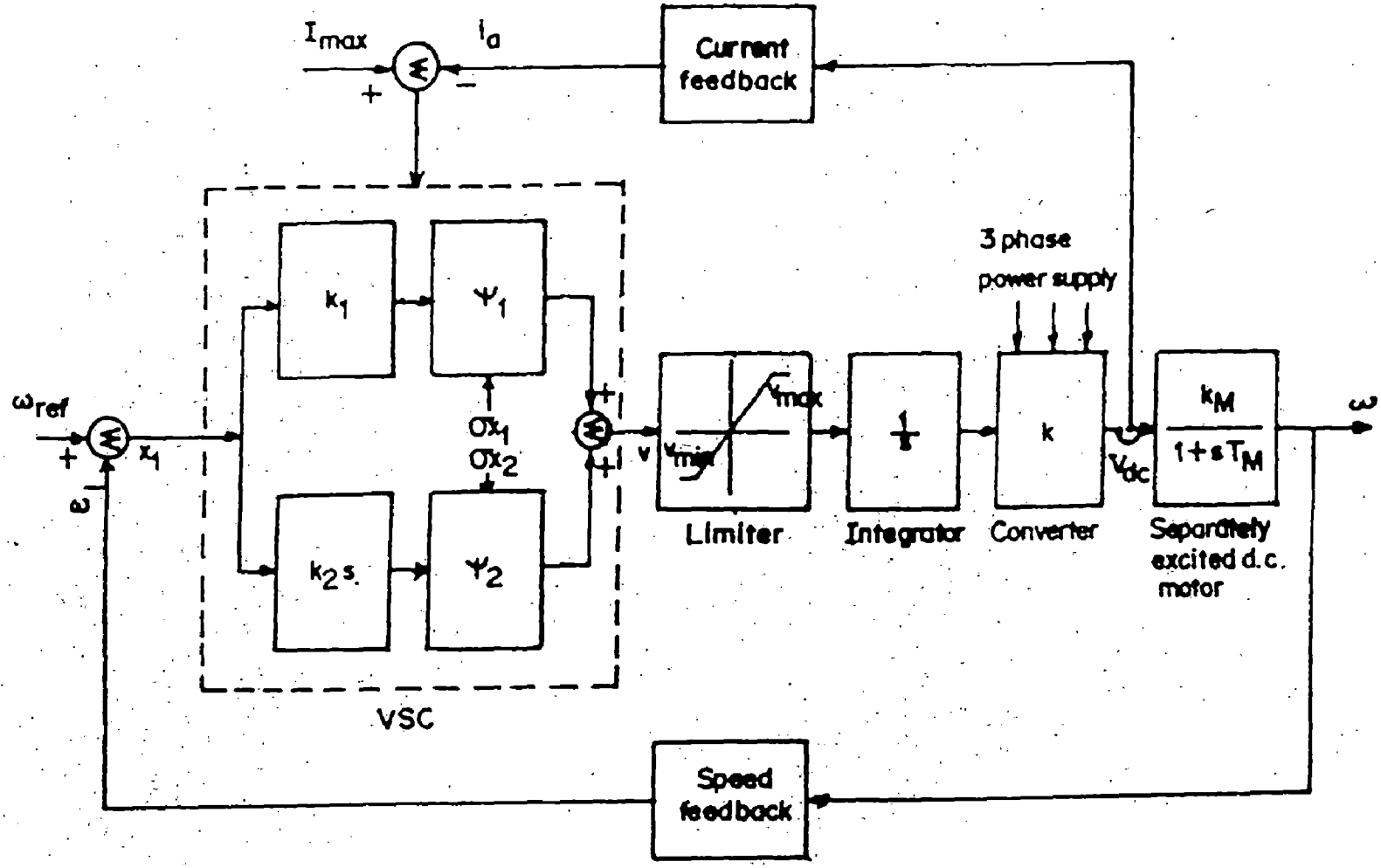


Fig. 1.3 Control model of the dc drive in sliding mode

arbitrary differential functions, and their intrinsic on-line adaptation and learning capabilities [5].

The ability of ANN to learn large classes of non-linear functions is well known. It can be trained. Once system dynamics have been identified using an ANN, many conventional control techniques can be applied to achieve the desired objective. Among these techniques model reference adaptive control is specifically used in trajectory control applications.

In recent years the fuzzy logic has gained much popularity in many control applications. Fuzzy logic is an attractive technique for complex or ill defined plant model. In fuzzy control system, translation of linguistic control rule into corresponding control actions is very tedious task. If fuzzy logic and neural network are combined then much powerful control system can be achieved [6].

The dissertation deals with the design and implementation of an ANN based controller for the speed control of a dc motor.

NEURAL NETWORKS

2.1 NEURAL NETWORKS

2.1.1 Principle

Neural network or Artificial Neural Network (ANN), as the name indicated, is the interconnection of artificial neurons that tend to simulate the nervous system of a human brain.

The model of an artificial neuron that closely matches a biological neuron is given by an op-amp summer-like configuration shown in Fig. 2.1. The artificial neuron is also called a processing element (PE), a neurode, a node or a cell. The input signals X_1, X_2....., X_m are normally continuous variables instead of discrete pulses that occur in a natural neuron. The weights can be positive (excitory) or negative (inhibitory) corresponding to acceleration of inhibition, respectively of the flow of electrical signals. The summing node accumulates all the input weighted signals and then passes to line output through the transfer function which is usually non-linear. The transfer function can be step-on threshold type (that passes logical 1 if the input exceeds a threshold; or else 0), signum type (output is +1 if the input exceeds a threshold; or else −1) or linear type with the output clamped to +1. The transfer function can also be non-linear continuously varying type, such as sigmoid (shown in Fig. 2.1), inversetan, hyperbolic or Gaussian type. The sigmoidal transfer function is most commonly used, and it is given by [7]

$$Y = \frac{1}{1+e^{-\alpha x}} \qquad (2.1)$$

Where a is the coefficient or gain which adjusts the slope of the function that changes between the two asymptotic values (0 and +1). Note that with high gain, it approaches a step function. The sigmoidal function is non-linear, monotonic,

differentiable, and has the largest incremental gain a zero signal, and these properties are of particular interest. All the above transfer functions are characterized as *squashing function*, because they squash or limit the output values between the two asymptotes. It should be mentioned here that the linear transfer function removes non-linearity from the neuron and eliminates the capability of neural network to emulate non-linear phenomena.

In general neural networks can be classified as feed forward and feed back types depending on interconnection of the neurons. At present majority of problems (roughly 90%) use feed forward architecture, and it is of direct relevance to power electronics and motion applications. Fig. 2.2 shows the structure of feed forward multilayer network with three input and two output signals. The topology is based on perceptron which was proposed by **Rosenblatt** (*1958*) and was used to emulate the biological vision system. The circles represent neurons and the data in the connection represent the weights. The network has three layers, defined as input layer (a), hidden layer (b), and output layer (c). The hidden layer functions as a connection between the input and output layers. The input and output layers (defined as buffers) have neurons equal to the respective number of signals. The input layer neurons do not have transfer function, but there are scale factors, as shown, to normalize the input signals. There may be more than one hidden layer. The number of hidden layers and the number of neurons in each hidden layer depend on the network design considerations. The input layer transmits the signals to the hidden layer and the hidden layer, in turn, transmits the signal to the output layer, as shown. There is no self, lateral or feedback connection of neurons. The network is fully connected when each of neuron in given layer is connected with each of the neurons in the next layer, as shown in Fig. 2.2 or can be "partially connected" when some of these connections are deleted. A neural network and input and output signals may be logical (0, 1), discrete bidirectional (± 1) or continuous variables. Often, continuous-variable signals, such as sigmoid functions at the output are clamped to convert to logical variables.

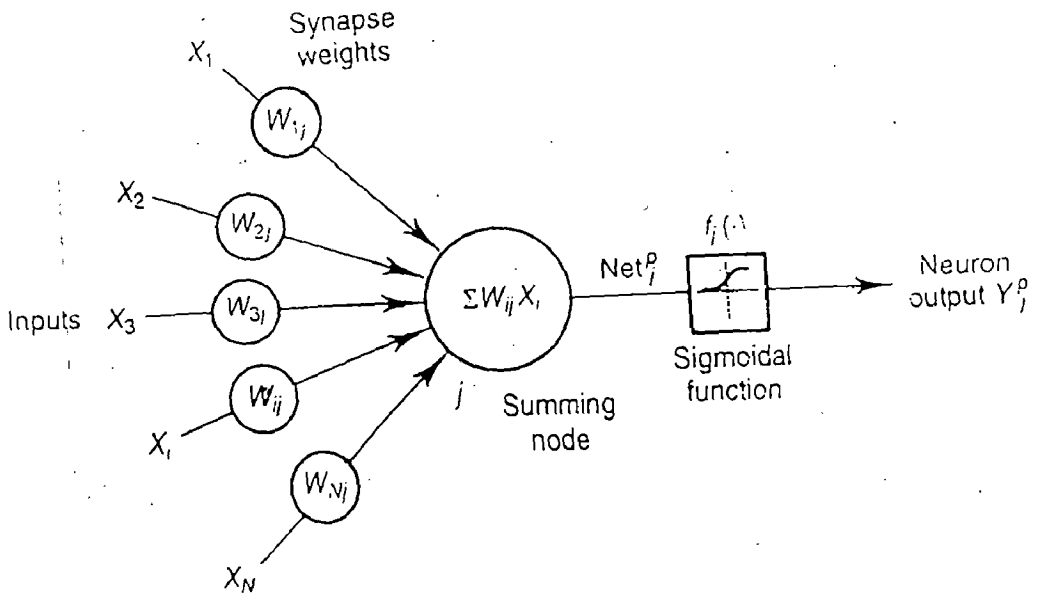


Fig. 2.1 **Structure of an artificial neuron**

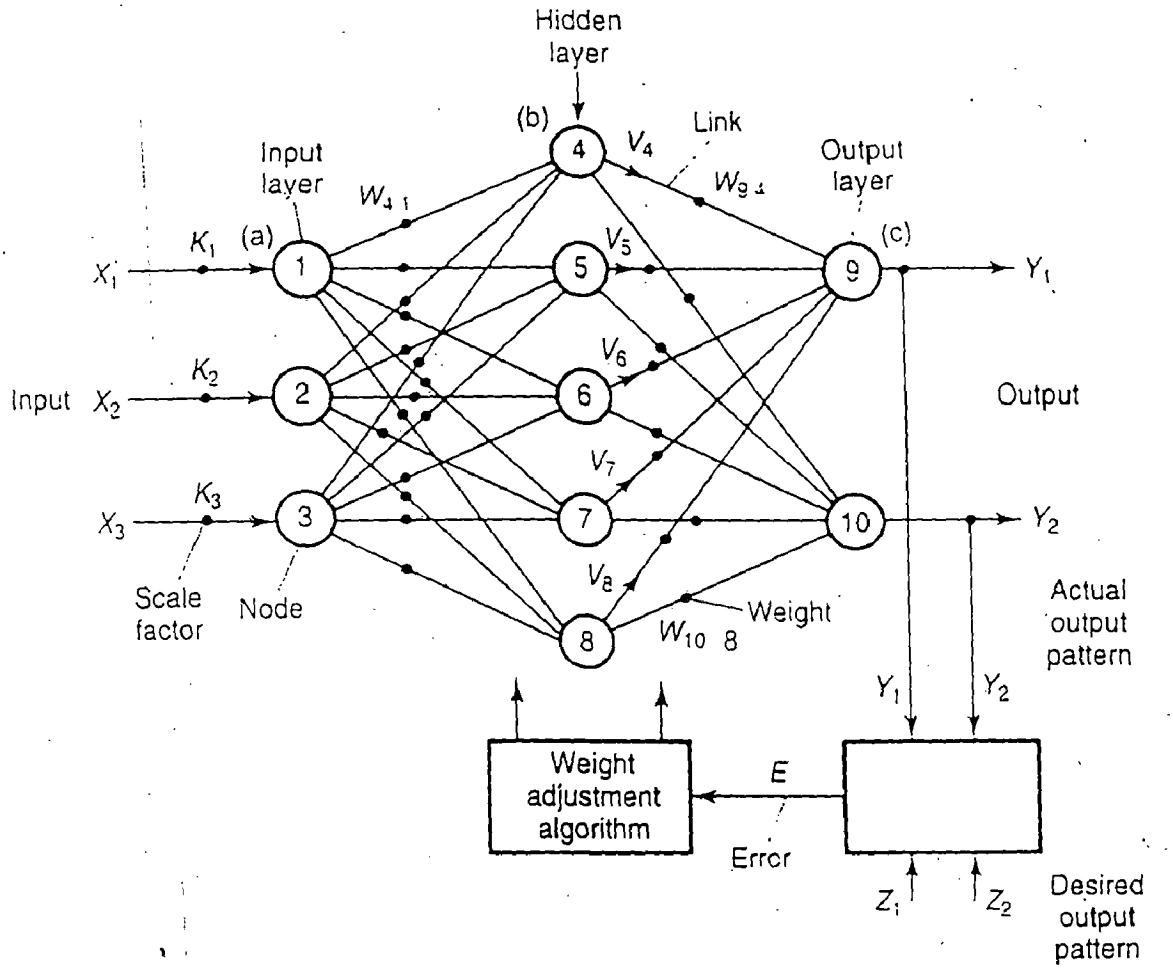


Fig. 2.2 **Structure of feed forward neural network showing back-propagation training**

The vector and matrix notation is often convenient in dealing with the inputs, outputs and weight. In Fig. 2.2, assume that the hidden layer neuron outputs are V_4, V_5, V_6, V_7 and V_8, as indicated. If the transfer functions are assumed to be linear with unity gain, the output of the hidden layer in matrix can be given as

$$\begin{bmatrix} V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \end{bmatrix} = \begin{bmatrix} W_{41} & W_{42} & W_{43} \\ W_{51} & W_{52} & W_{53} \\ W_{61} & W_{62} & W_{63} \\ W_{71} & W_{72} & W_{73} \\ W_{81} & W_{82} & W_{83} \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \tag{2.2}$$

or,

$$\overline{V}_b = \overline{W}_{ba}\, \overline{X}_a$$

where $\overline{V}_b$ is the output vector of layer b which is given as the dot product of the weight or connectivity matrix $\overline{W}_{ba}$ and the input layer signal vector $\overline{X}_a$. Similarly, the network output signals can be given in the matrix form as

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} W_{94} & W_{95} & W_{96} & W_{97} & W_{98} \\ W_{10,4} & W_{10,5} & W_{10,6} & W_{10,7} & W_{10,8} \end{bmatrix} \cdot \begin{bmatrix} V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \end{bmatrix} \tag{2.3}$$

or,

$$\overline{Y}_c = \overline{W}_{cb} \cdot \overline{V}_b$$

Combining (2.2) and (2.3)

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} W_{94} & W_{95} & W_{96} & W_{97} & W_{98} \\ W_{10,4} & W_{10,5} & W_{10,6} & W_{10,7} & W_{10,8} \end{bmatrix} \begin{bmatrix} W_{41} & W_{42} & W_{43} \\ W_{51} & W_{52} & W_{53} \\ W_{61} & W_{62} & W_{63} \\ W_{71} & W_{72} & W_{73} \\ W_{81} & W_{82} & W_{83} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \tag{2.4}$$

or,

$$\overline{Y}_c = \overline{W}_{cb} \cdot \overline{W}_{ba}\, \overline{X}_a = \overline{W}_{cba} \cdot \overline{X}_a$$

which indicates that the output vector $\overline{Y}_c$ is the dot product of combined weight matrix $\overline{W}_{cba}$ and the input vector $\overline{X}_a$. The above calculations are strictly invalid with the non-linear transfer function [7].

2.1.2 Training of Neural Network

The neural network computes very fast in parallel and distributed manner compared to the sequential computation in a conventional computer that requires the help of centralized CPU and storage memory. It is more like analog computation.

Neural network performs the function of *nonlinear mapping* or *pattern recognition*. This means that if an input set of data corresponds to a definite signal pattern, the network can be "trained" to give correspondingly a designed pattern at the output. The network has the capability to "learn" because of the distributed intelligence contributed by the weights. The input-output pattern matching is possible if appropriate weights of selected.

The network can be learn using supervised and unsupervised training is that, in the former case, external prototypes are used as target outputs for specific inputs, and the network is given algorithm to follow and calculate new connection weights that bring the output closer to target output. Unsupervised learning is the sort of learning that takes place without a teacher [8].

For neural networks, in the unsupervised case, a learning algorithm may be given but target outputs are not given. In such a case, data input to the networks gets clustered together; similar input stimuli cause similar responses.

There are two main rules for learning Hebbian learning, used with unsupervised learning and the delta rule, used with supervised learning.

(a) Hebb's Rule

Learning algorithms are usually referred to as learning rules. The foremost such rule is due to Donald Hebb. Hebb's rule is a statement about how the firing of

one neuron, which has a role in the determination of the activation of another neuron, affects the first neuron's influence on the activation of another neuron, affects the first neuron's influence on the activation of the second neuron, especially if it is done in a repetitive manner. As a learning rule, Hebb's observation translates into a formula for the difference in a connection weight between two neurons from one iteration to the next, as a constant μ times the product of activations of two neurons. How a connection weight is to be modified is what the learning rule suggests. In the case of Hebb's rule, it is adding the quantity $\mu\, a_i\, a_j$, where a_i is the activation of the i^{th} neuron, and a_j is the activation of the j^{th} neuron to the connection weight between the i^{th} and j^{th} neurons. The constant μ itself is referred to as the learning rate. The following equation using the notation just described, states it succinctly.

$$\Delta w_{ij} = \mu\, a_i \,.\, a_j \qquad (2.5)$$

The learning rule derived from Hebb's rule is quite simple and is used in both simple and more involved networks. Some modify this rule by replacing the quantity a_i with its deviation from the average of all a's and, similarly, replacing a_j by a corresponding quantity. Such rule variations can yield rules better suited to different situations [8].

(b) Delta Rule

The delta rule is also known as the *least mean squared error rule* (LMS). First calculate the square of the errors between the target or desired values and computed values, and then take the average to get the mean squared error. This quantity is to be minimized. For this, realize that it is a function of weights themselves, since the computation of output uses them. The set of values of weights that minimizes the mean squared error is what is needed for the next cycle of operation of neural network. Having worked this out mathematically, and having compared the weights thus found with the weights actually used, one determines

their difference and gives if in the delta rule, each time weights are to be updated. So the delta rule, which is also the rule first by **Widraw** and **Hoff**, in the content of learning in neural networks, is stated as an equation defining the change in the weights to be affected [9].

(c) Generalized Delta Rule

While the delta rule uses local information on error, the generalized delta rule uses error information that is not local. It is designed to minimize the total of the squared errors of the output neurons. In trying to achieve this minimum, the *steepest descent method*, which uses the gradient of the weight surface is used (also used in the delta rule). For the next error calculation, the algorithm looks at the gradient of the error surface, which gives the direction of the largest slope on the error surface. This is used to determine the direction to go to try to minimize the error. The algorithm chooses the negative of this gradient, which is the direction of steepest descent during starting or change in the reference speed setting [9].

(d) Error Back Propagation

An ANN is trained to emulate a function by presenting it with a representative set of input/output functional patterns. This technique adjusts the weights in all connecting link thresholds in the nodes so that the difference between the actual output and the target output are minimized for all given training patterns. For pth training pattern (p = 1,....., P), this is done by minimizing the energy function,

$$Ep = \left(\frac{1}{2}\right)\sum_i \left(T_i - Y_i(H)\right)^2 \qquad (2.6)$$

w.r.t. all the weights and thresholds. Yi(H) corresponds to the activation of the ith neuron in the output layer H, Ti denotes the desired target. The corresponding updates for the weights are calculated using the iterative gradient descent technique, where,

$$w_{ij}^{new}(h) = w_{ij}^{old}(h) + \eta \frac{\partial E_p}{\partial w_{ij}(h)} + \nu \Delta w_{ij}(h) \qquad (2.7)$$

The above algorithm is commonly known as error back propagation. The constant η is the learning step while the constant ν is the momentum gain, Δwij(h) indicates the weight change in the previous iteration. Weights are iteratively updated for all P training patterns. Sufficient learning is achieved when the total error function

$$E_{total} = \sum_p E_p$$

summed over the set of all P trajectory patterns goes below a preselected threshold value [10].

2.2 FUZZY LOGIC PRINCIPLE [7]

Fuzzy logic deals with problems that have vagueness, uncertainty, or imprecision and used membership function with values varying between 0 and 1. In fuzzy set theory based on fuzzy logic, a particular object has a degree of membership in a given set that may be anywhere in the range of 0 and 1.

A fuzzy variable has values, which are expressed by natural English language. For example, the speed of a machine can be defined by linguistic variables Low, Medium and High, where each is defined by a gradually varying bell shaped membership function. The shape can also be triangular or trapezoidal and can be symmetric or assymetric.

2.2.1 Properties

The basic properties of fuzzy sets are as follows :

Union – Given two fuzzy subsets A and B of a universe of discourse X, the union $A \cup B$ is also a fuzzy set of X with membership function given by

$$\mu_{A \cup B(X)} = \min [\mu_A(X), \mu_B(X)]$$

Intersection – The intersection of two fuzzy sets A & B of the universe of discourse X, denoted by A $\cap$ B has the membership function given by

$$\mu_{A\cap B(X)} = \min\left[\mu_A(X), \mu_B(X)\right]$$

Complement or Negation – The complement of a given set A of the university of discourse X, defined by TA, has the membership function

$$\mu_A(X) = 1 - \mu_T\ A(X)$$

Fuzzy control is described by a set of IF....THEN rules, where the rule has the following general structure.

IF x is A AND y is B THEN z is C

Where x, y and z are the fuzzy variables and A, B and C are the fuzzy subsets in the universe of discourses X, Y and Z respectively.

In general, a fuzzy rule base is first constructed by the designer and then all the fuzzy sets of each variable are described by appropriate membership function. The individual rules are combined to give an overall rule R which is computed by the union operator as follows :

$$R = R1 \cup R2 \cup R3 \ldots\ldots \cup R_n$$

For the given rule base of a control system, the fuzzy controller determines the rules to be fired for the specific input signal condition and then computes the effective control action. The commonly used SUP_MIN method is given as

$$U = x \,.\, R$$

or

$$\mu_u\ (\mu) = \text{Sup}\ X\left[\min(\mu_X(x) \cdot \mu_R(x\,u)\right]$$

The "fuzzification" operation can be performed by considering the crispy input values as "singletons" (fuzzy sets that have membership value of 1 for a given input value and 0 at other prints) and taking the values of the sets membership function at the respective data value.

The common methods of "Defuzzification" are center of gravity (or control) and height method.

2.2.2 Centroid Method

The centroid defuzzification method, determines the output crisp value from center of gravity of the output membership function and is given by the expression

$$U_o = \frac{\int U.\mu(U).dU}{\int \mu(U).dU}$$

2.2.3 Height Method

In the height method, the centroid of each output membership function for each rule is first evaluated. The final output is then calculated as the average of the individual centroids weighted by their heights (degree of membership) as follows :

$$U_o = \frac{\sum_{i=1}^{n} U_i \, \mu(U_i)}{\sum_{i=1}^{n} \mu(U_i)}.$$

2.3 NEURO FUZZY THEORY

The ability to recognize a pattern is the first requirement for any intelligent machine. Pattern recognition is a must component of the so called "Intelligent control systems" which involves processing and fusion of data from different sensors and transducers. It is also a necessary function providing "failure detection", "verification" and "diagnosis task". Machine recognition of patterns can be viewed as two-fold task, consisting of learning like in variant and common properties of a set of samples characterizing a class and of deciding that a new sample is a possible member of the class by noting that it has properties common to those of the set of samples [11].

In a pattern recognition or vision system, uncertainties can arise at any phase of the aforementional tasks resulting from incomplete or impressive input information, ambiguity or vagueness in input images, ill defined and/or overlapping boundaries among the classes or regions, and indefiniteness in defining/extracting features and relations among them. Any decision taken at a particular level will

and uncertainty management) to handle real life recognition problems. A large number of researchers have now concentrated on exploiting these modern concepts during the past decades to solve complex problems in various fields under a new branch called neuro-fuzzy computing.

Major areas in which neural networks have been applied in order to exploit the computational power, and to make robust decisions are :

- Feature selection and pattern classification.
- Image preprocessing and scene analysis.
- Text processing.
- Expert system design/rule generation.
- Controller design.
- Natural language processing
- Approximate reasoning and speech recognition.

2.3.1 Neural Fuzzy Control

This type of control has advantages that it permits automatic identification of fuzzy rules and tunes the membership functions. The FNN topology can be either on rule based approach or relational approach. The FNN topology for closed loop adaptive control is shown in Fig... The network has two inputs error (E) and change in error (CE) and one output control signal (U). Each premise has three membership function (SMALL, MEDIUM and BIG) which are synthesized with the help of sigmoids (f) [7].

The weights w_c and w_g give spacing and slope, respectively, for the membership functions. The weights are determined by back propagation method. The premises are identical to both rule-based and relational topologies. The nine outputs of the premises after product give nine rules. The inferred value of the FNN is obtained as sum of the products of the equations in the consequences, as shown in Fig. 2.3. A typical rule can be read as

$$\text{IF E is SM AND CE is ME THEN } U = \frac{U_1\mu_1 + U_2\mu_2}{\mu_1 + \mu_2}$$

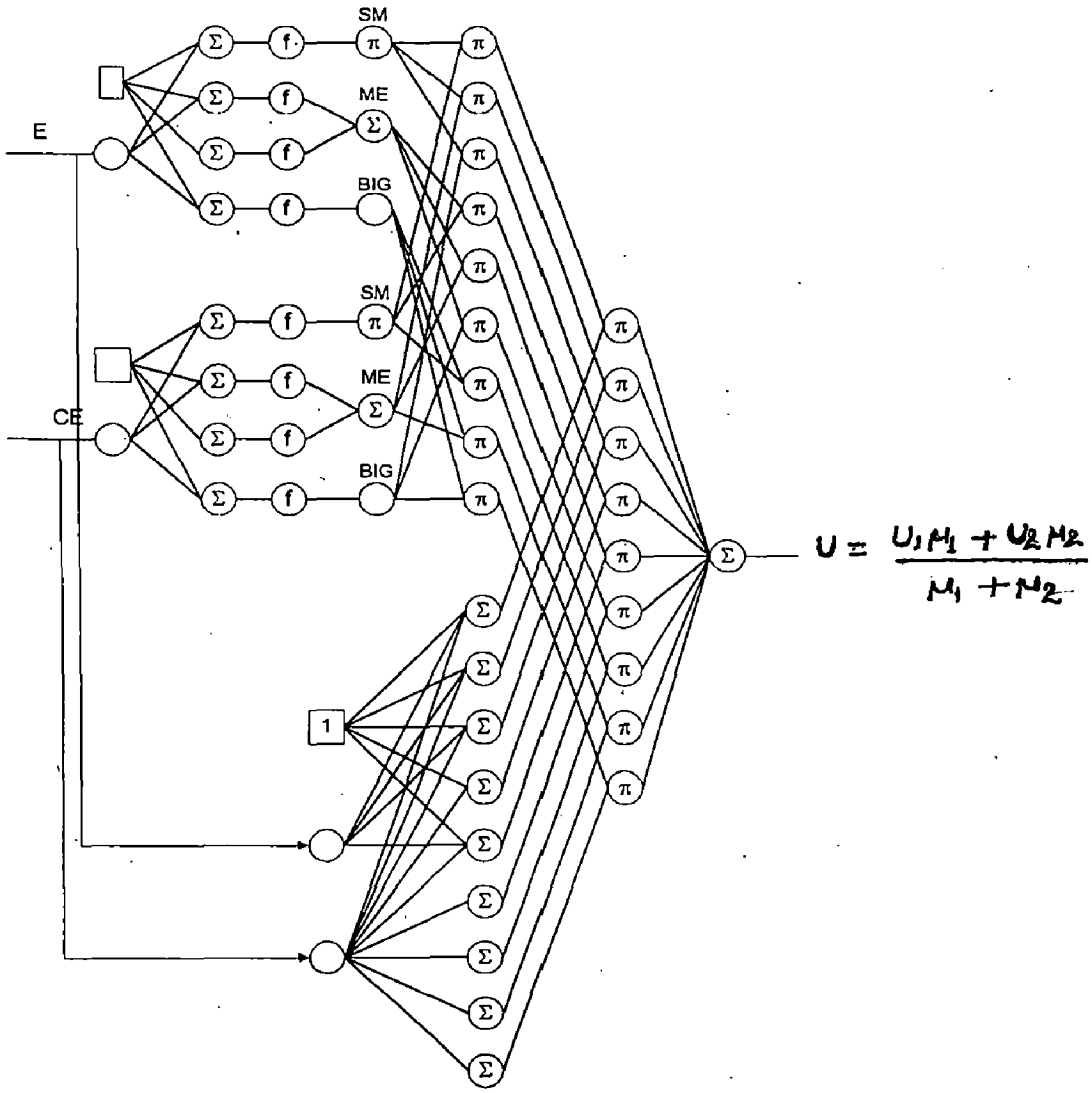


Fig. 2.3 Structure of Fuzzy Neural Control

Where,

$$U_1 = w_{a01} + w_{a11}.E + w_{a21} \cdot CE$$

$$U_1 = w_{a01} + w_{a11}.E + w_{a21} . CE$$

$$\mu_1 = SM . IM$$

and

$$\mu_2 = SM . M.E$$

2.4 LITERATURE REVIEW

The concept of neurocontroller is not very old. A lot of research is going on using neural network for the control of dynamic system around the world in recent years.

Alongwith the progress in neuroanatomy and neurophysiology, the efforts for making model of human brain were always done. In 1949, D. O. Hebb [12] proposed one such model with a learning law that became starting point for artificial neural network training algorithms.

In 1950s and 1960s, a group of researchers combined these biological and physiological insight to produce the first Artificial Neural Network. Z. M. Zurada [13] presented a very detailed discussion on artificial neural network in his book. He explained different architecture for neural network configuration and number of training algorithms.

B. K. Bose [7] discussed applications of Expert System, Fuzzy Logic and Neural Network in power electronics in his paper. The paper gives a brief review of the three branches of artificial intelligence. The theoretical principles of each of that are relevant to power electronics and motion control applications are described. Then, several applications in each topic are described to supplement the concept.

Chu et al. [14] discussed two different approaches for utilization of neural networks in identification of dynamical systems. In the first approach, a Hopfield network is used to implement a least square estimation for time varying and time-

invariant systems. The second approach which is in the frequency domain utilizes a set of orthogonal functions and Fourier analysis to construct a dynamic system in terms of its Fourier coefficients.

Nguyen and Widrow [15] presented the "truck backer-upper", a neural controller steering a trailer truck while backing up to a loading dock, in their paper. The controller is able to guide the truck to the dock from almost any initial position. The controller developed by them should be applicable to a wide variety of non-linear control problems.

Low, Lee and Lim [9], in their paper presented a closed-loop methodology for neural network training for control of drives with non-linearities. In the paper, problems associated with more common training scheme, and how these are addresses by the proposed closed-loop method, are discussed. An inverse non-linear control using NN for control of non-linear systems is discussed.

In the paper "Neural Network Control for DC Motor Micromaneuering", the application of an ANN controller for compensating the effects induced by the friction in a dc motor micromaneuering system is discussed. A back-propagation NN operating in the specialized learning mode, using sign gradient descent algorithm is employed. The on-line training of the neural network is performed in the region of interest of output domain. The output of ANN resembles that of a PWM controller. The effect of number of neurons in the input and hidden layers on the transient system response is explored and experimental studies are presented [5].

Min-Huei Kim et al [16] explores the application of neural networks for estimation of power electronic waveforms, in their paper. The distorted line current waveforms in a single-phase thyristor ac controller and a three phase diode rectifier that feeds an inverter-machine load have been taken into consideration, and NN have been trained to estimate the total rms current, fundamental rms current, displacement factor and power factor. The performance of the neural network based estimation has been compared with the actual values, and excellent performance is indicated.

Artificial neural network based high performance speed control system for a dc motor introduced by Weerasooriya and El-Sharkawi in their paper. The unknown non-linear dynamics of the motor and the load are captured by an artificial neural network. The trained neural network identifier is combined with a desired reference model to achieve trajectory control of speed. Performance of the identification and control algorithms are evaluated by simulating them on a dc motor model [10].

Chen [17] developed a neural network based self-tuning controller with a traditional model structure. Identification and control of dynamic system using neural network were expanded by Narendra and Parthasarthy [18], who outlined their applications in Model Reference Adaptive Control Systems.

L. A. Zadeh [19] first introduced the fuzzy logic in 1965, combined the multivalued logic, probability theory, artificial intelligence and neural networks to develop this digital control methodology, that stimulates human thinking by incorporating the imprecision inherent in all physical systems.

Mamdani and Assilian [20] first reported the application of fuzzy logic to control a model laboratory steam engine. The purpose was to control engine speed and boiler steam pressure by using heat applied to the boiler and the throttle setting on the engine.

Most of the supervised and reinforcement learning methods of neural network process only numerical data. For supervised learning problems, some approaches have been proposed to process linguistic information with fuzzy inputs. Fuzzy output or fuzzy weights. Ishibuchi and his co-workers [21, 22] have proposed a series of approaches and applications with the capacity of processing linguistic input or/and linguistic output. In their methods, the weights, inputs and outputs of the neural network are fuzzified using fuzzy numbers represented by α-level sets. They derived learning algorithms from a cost function defined by the α-level sets of actual fuzzy outputs and target outputs.

Hayashi et al. [23] also proposed a similar method with fuzzy signals and fuzzy weights by using triangular fuzzy members.

Janardanan and Gajendran [6] proposed a new scheme for the speed control of DC drive using ANN. The training patterns are generated using fuzzy logic principles. The signal corresponding to the motor speed error and change in error are used as the inputs. The ANN is simulated using back-propagation algorithm with adaptive learning and momentum.

Author's Contribution :

A three-phase fully controlled bridge converter has been designed and developed for speed control of dc motor. The firing pulses are generated using 8031 microcontroller based dedicated card which is serially linked with personal computer. For closed loop control, speed and current are measured using ADD-ON card. The speed of the motor is controlled using ANN based speed controller which is designed and trained off-line. Patterns for neural network are generated using fuzzy logic. The performance of the system is investigated in open loop and closed loop mode.

SIMULATION OF DC DRIVE

The open loop operation may not be satisfactory in many applications. The satisfactory operation can be achieved in closed loop control system.

3.1 CLOSED LOOP CONTROL OF DC DRIVES

A basic scheme of the closed loop speed control system employing current limit control is shown in Fig. 3.1. ω_m^* sets the speed reference. A signal proportional to the motor speed is obtained from the speed sensor. The speed error output is filtered to remove the ac ripple and compared with the speed reference. The speed error is processed through a speed controller. Output of speed controller V_c, adjusts the converter firing angle α to make the actual speed close to the reference speed.

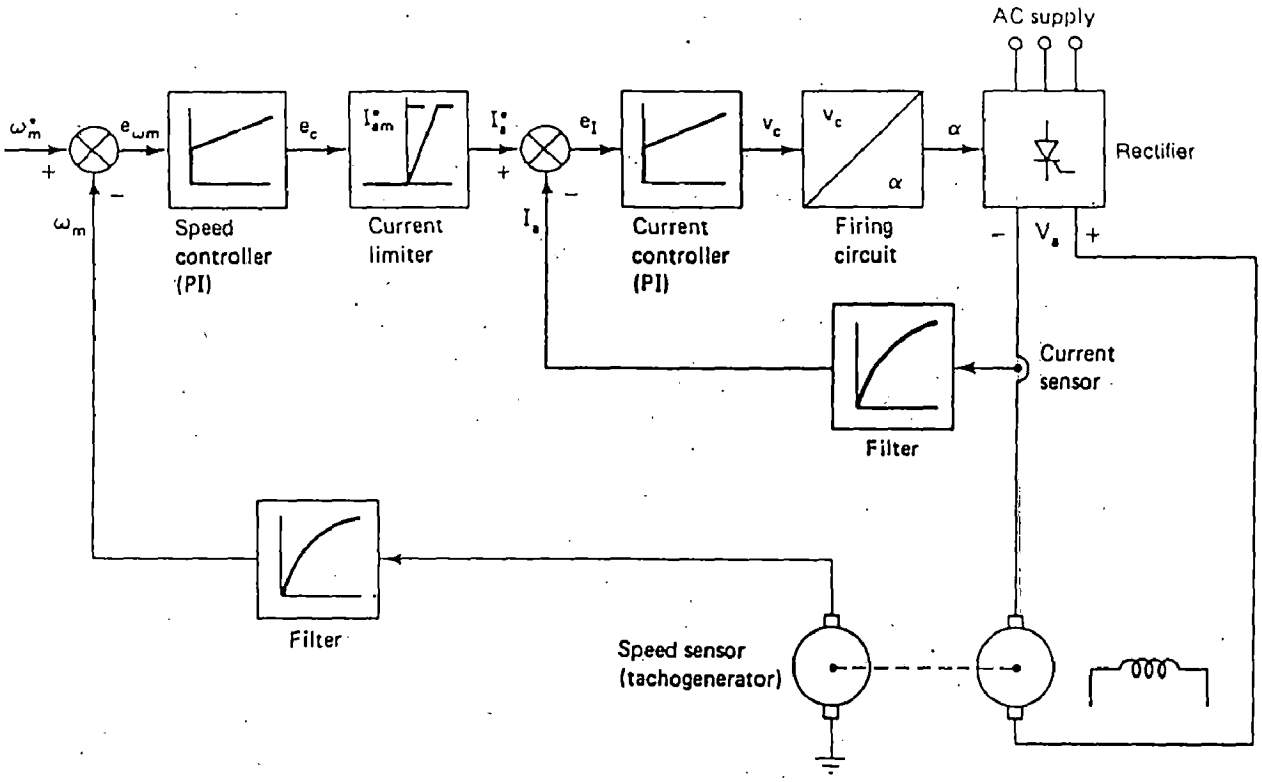
The drive employs current limit control. The error is processed through a controller. The output of controller is fed to a limiter, which sets a current reference for the closed loop current control.

The purpose of current control is to prevent the current from exceeding safe values. Sometimes, the purpose of current control is to prevent the current from exceeding safe values. Sometimes, the purpose of current control is to intentionally force the current to the maximum permissible value during the transient operations. This allows full use of drive torque capability and consequently gives very fast response [24].

3.2 DESIGN OF CURRENT CONTROLLER

A current feed back loop is used to limit the armature current. As the function of the controller is only to limit the current, it is analog proportional controller of constant gain Kc. The value of Kc can be calculated using machine parameters. The method is as follows [1].

The gain of current controller can be chosen on the basis of steady-state error consideration where



(b) Drive with inner current control loop

Fig. 3.1 **One quadrant closed-loop speed control**

$$\varepsilon I(\alpha) = \left.\frac{1}{1+G(s)*H(s)}\right|_{s=0} \quad (3.1)$$

From the current control loop

$$G(s)|_{s=0} = K_c A\, K_{m1} \quad (3.2)$$

and $H(s)|_{s=0} = K_r$

where, A = converter gain

$$K_{m1} = \frac{B}{K_b^{\,2} + RaB}$$

K_b = Back emf constant

Ra = Armature resistance including external resistance

B = Viscous friction coefficient

K_r = Current feed back gain.

$$\varepsilon_I(\alpha) = \frac{1}{1+AK_cK_{m1}K_r} \quad (3.3)$$

$$K_c = \frac{\frac{1}{\varepsilon_I(\alpha)} - 1}{A\,K_{m1}K_r} \quad (3.4)$$

where $\varepsilon_I(\alpha)$ is the desired steady-state error. While the current error is not so critical, it must not be too large. A practical value might be 10% for $\varepsilon_I(\alpha) = 0.1$.

The motor parameters are given in Appendix A-1. The current feedback gain K_r is taken as 1.

Cosine firing technique is used to control the output of the converter. Therefore, the gain of the converter V_a/V_c becomes a constant (A). A 3-phase 220 V, 50 Hz supply is applied to the 3-phase bridge converter. Assuming the control voltage varies from +10 V to – 10V for the entire firing angle control range (0-180°), the gain of the power converter (A) is given by

$$A = \frac{\frac{3\sqrt{2}}{\pi} \times 220\cos 0^{\circ}}{10} \approx 30 \quad (3.5)$$

On substituting various values the gain of the current controller comes out to be 27.

3.3 DESIGN OF SPEED CONTROLLER USING ANN

In a closed loop control scheme, the reference speed and actual speed are compared and the speed error is processed in the speed controller to generate the current reference. The speed error and change in speed error are calculated after every 10 msec and error is processed to calculate the change in current reference. In the present work, the speed controller is designed using a ANN. A feed forward network is shown in Fig. 3.2 is used to represent speed controller. The network has two inputs, one output and one hidden layer.

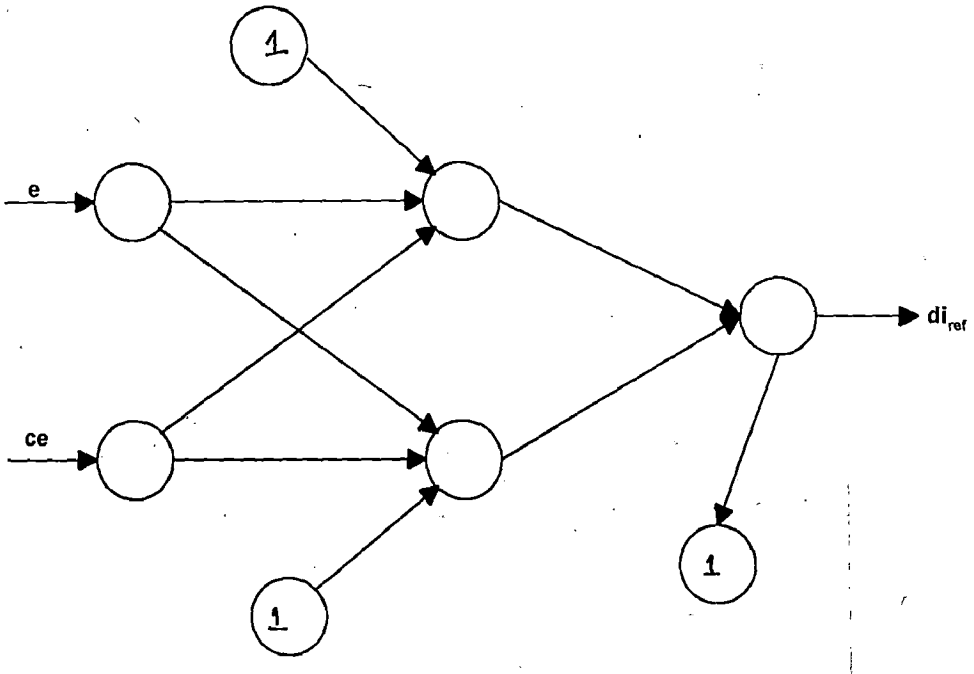


Fig. 3.2 : Neural Network for speed controller

The inputs to the network are speed error and change in speed error and output is change in current reference. The network is trained off-line using error-back propagation training algorithm. The settling time and overshoot are calculated for step change in reference speed setting.

The flowchart for obtaining the response of the system for step change in reference speed is shown in Fig. 3.3 which consists of main program, speed error processing subroutine, current processing error subroutine and machine dynamics subroutine.

3.2.1 Main Program

Flow chart for main program is shown in Fig. 3.3. Various initializations are done. Fuzzy rule base table also initialized. Values of rated current, reference speed and rated speed have been read. Fuzzy partitions for error and charge in error are calculated using rated speed. Index km is initialized to zero. This is used to run current for ten times after each speed loop indices i and t are initialized to zero. Percentage overshoot and settling time is also calculated in main program for which cont is initialized to zero and set to one after calculating settling time.

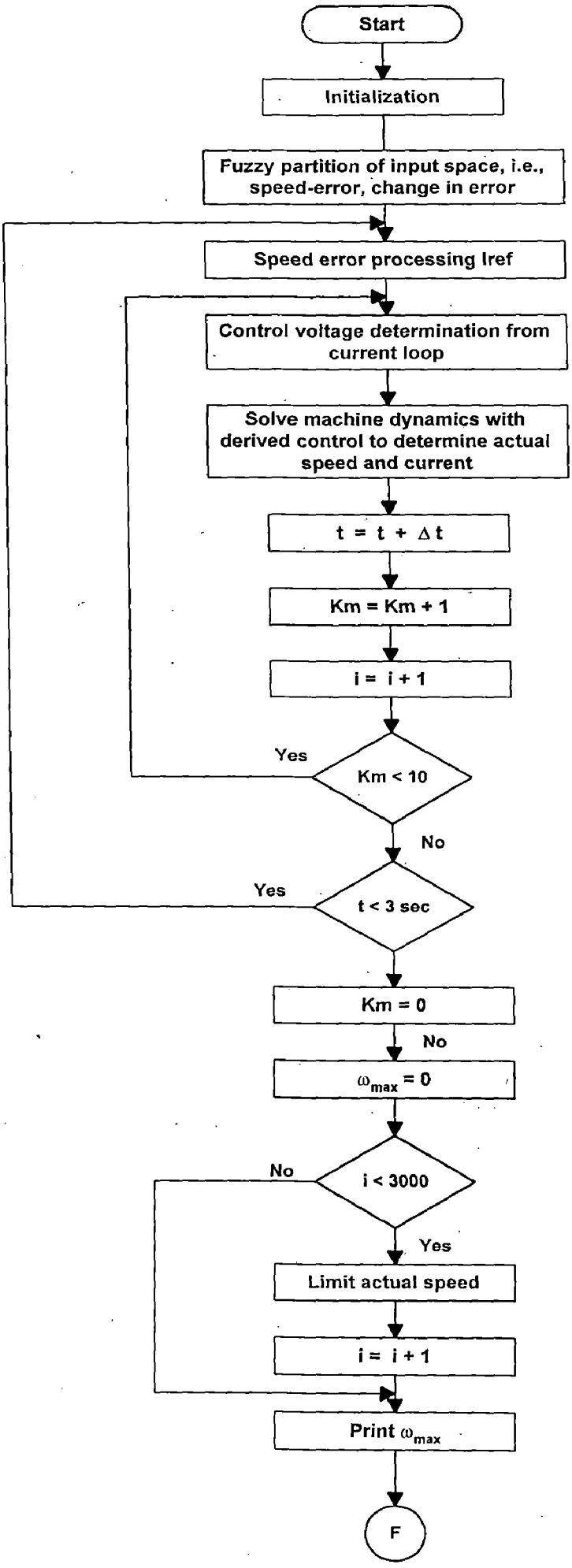
3.2.2 Speed Error Processing Loop

This is called by main once after ten consecutive iterations of current loop. The flow chart for this subroutine is shown in Fig. 3.4. The first step after entering in this subroutine is to initialized local variable. Error at 0th iteration and index num is initialized to zero. Then error is calculated at i th iteration, using reference speed and actual speed at previous iteration. A fuzzy partition of speed error and change in speed error processing is shown in Fig. Membership function is triangular and 50% overlap is taken. Four rules for each value of speed error and change in error can be generated. From these rules membership function for output is calculated using SUP_MIN method.

For a particular value of error and change in error, output membership function depending on values of I_s, I_s', J_s and J_s', region in which they are lie has determined. These regions are determined using comparison method. This output membership function is then defuzzified using height method to get reference current. This is added algebraically to previous value. This value is limited within Imax and 0. Values of error, change in error and reference current is stored and used as training pattern for neural network.

3.2.3 Current Error Processing Loop

This loop is used for calculating control voltage and voltage at the terminal of armature. The gain of proportional controller is multiplied with current error. This control value is then limited with in Vmax and Vmin. Armature voltage is


Start
Initialization
Fuzzy partition of input space, i.e., speed-error, change in error
Speed error processing Iref
Control voltage determination from current loop
Solve machine dynamics with derived control to determine actual speed and current
t = t + Δt
Km = Km + 1
i = i + 1
Yes
Km < 10
No
Yes
t < 3 sec
Km = 0
No
ωmax = 0
No
i < 3000
Yes
Limit actual speed
i = i + 1
Print ωmax
F

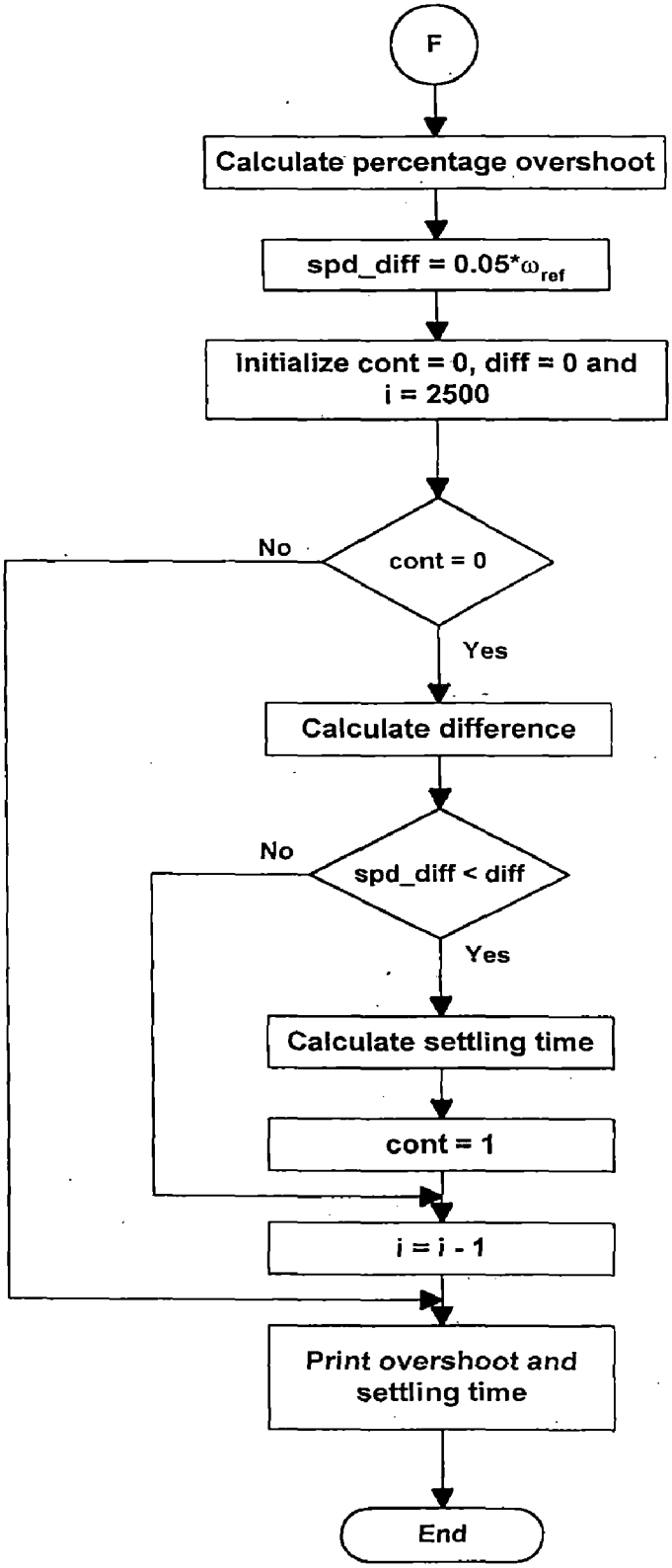


Fig. 3.3 : Simulation Main Flow Chart

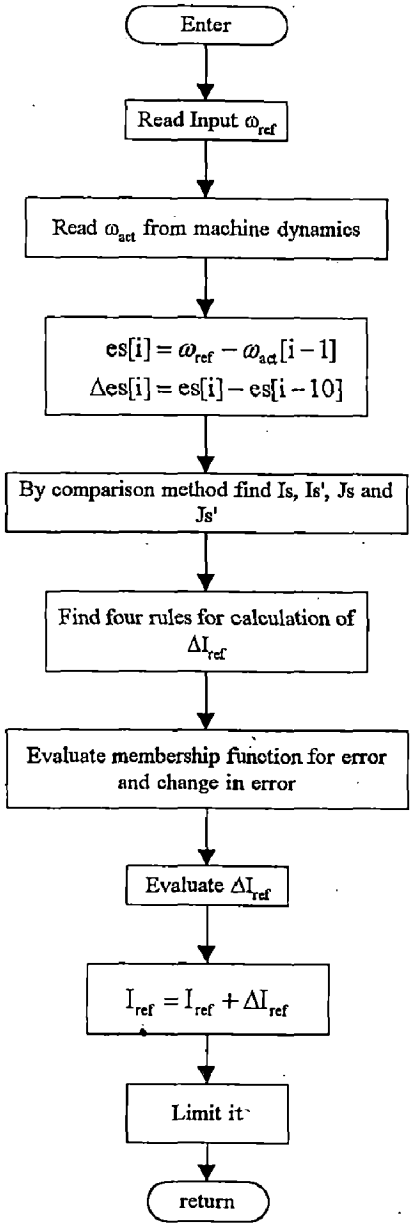


Fig. 3.4 Flow Chart for Speed Error Processing

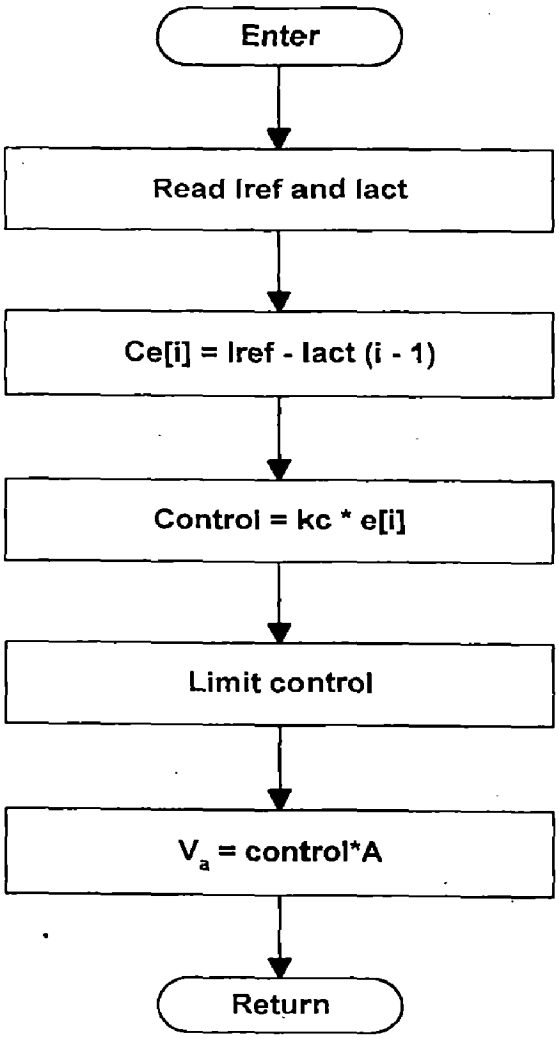


Fig. 3.5 : Current error processing

calculated after multiplying the gain of converter in this control voltage. Flow chart for this loop is shown in Fig. 3.5. This allows full use of drive torque capability and consequently gives very fast response.

3.2.4 Machine Dynamics

In order to determine the actual speed and current for error processing, machine dynamics is solved by numerical method. The parameters of motor, which are measured by various methods are required to solve the motor dynamics.

If the voltage at armature terminate is $V_a(t)$, back emf is E_g armature circuit inductance and resistance are L_a and R_a respectively then $V_a(t)$ at any instant t is given by

$$V_a(t) = E_g + R_a\, i_a(t) + \frac{L_a\, di_a(t)}{dt} \tag{3.6}$$

$$T_e = J\frac{d\omega}{dt} + B\omega + T_L \tag{3.7}$$

$$T_e = K_t\, i_a \tag{3.8}$$

from equation (3.6) and (3.7)

$$\frac{di_a(t)}{dt} = \frac{v_a(t) - E_g - R_a\, i_a(t)}{L_a} \tag{3.9}$$

$$\frac{d\omega}{dt} = (T_e - T_L - B\omega)/J \tag{3.10}$$

instantaneous values of current and speed can be determine from differential equations (3.9) and (3.10) using Runga-Kutta fourth order method.

$$f_1 = \left(v_a(t) - K_b\,\omega(t-1) - R_a\, i_a(t-1)\right)/L_a \tag{3.11}$$

$$f_2 = (K_t\, i_a\,(t-1) - T_L - B\omega(t-1)/J \tag{3.12}$$

$$K_{11} = h f_1 \tag{3.13}$$

$$K_{21} = h f_2 \tag{3.14}$$

here h is step size.

$$f_3 = \left[v_a(t) - K_b(\omega(t-1) + K_{21}) - R_a(i_a(t-1) + K_{11})\right]/L_a \tag{3.15}$$

$$f_4 = \left[K_t(i_a(t-1)+K_{11}) - T_L - B\omega(i-1)+K_{21}\right]/J \quad (3.16)$$

$$K_{12} = hf_3 \quad (3.17)$$

$$K_{22} = hf_4 \quad (3.18)$$

$$i_a(t) = i_a(t-1) + 0.5(K_{11}+K_{12}) \quad (3.19)$$

$$\omega(t) = \omega(t-1) + 0.5(K_{21}+K_{22}) \quad (3.20)$$

$i_a(t)$ and $\omega(t)$ are actual values of current and speed.

Machine dynamics is solved in each iteration along with current error processing. The whole drive is simulated for time of 3.0 secs, which is sufficient for the machine to be settle down to steady state. The response of the drive for finally selected rule based table is shown in Fig. 3.6 for reference speed of 100 rad/sec. The settling time is 0.868 sec and overshoot is 1.81 %. The fuzzy rule based table is used to generate the patterns for training neural network. These patterns are given in Table 3.1.

3.2.5 Program for Training of Neural Network

Flow chart for this is shown in Fig. 3.7. Firstly weights and maximum number of iteration IT is initialized. Input data and number of training pattern IP with number of input, output and hidden nodes have been read. Set IT equal to zero. Check if number of iteration is less than ITmax, if it is then intialize error level and take first pattern from training set. Output of hidden layer is calculated using sigmoid function. For which product of weights and input are summed up. Output of hidden layer is used to calculate the final output. Weights of output layer are used for this purpose.

Error is calculated between target and output of ANN. Change in weight matrix for hidden layer is also calculated. These weight matrices are used for updating the weights of corresponding layer.

Take the next pattern from training set till IP = IPmax. After adjusting the weights for all the error level stop adjusting the weights, otherwise increment IT. Repeat process till IT = ITmax. The trained network is shown in Fig. 3.8.

Table 3.1

e	ce	du	e	ce	du	e	ce	du
0.1433	0.1433	0.1302	0.2962	0.6019	0.4973	0.6529	0.8567	0.8035
0.1433	0.1943	0.1373	0.2962	0.6529	0.4983	0.7038	0.1433	0.4985
0.1433	0.2452	0.1444	0.2962	0.7038	0.5000	0.7038	0.1943	0.4990
0.1433	0.2962	0.1560	0.2962	0.7548	0.5005	0.7038	0.2452	0.4995
0.1433	0.3471	0.1965	0.2962	0.8057	0.5010	0.7038	0.2962	0.5000
0.1433	0.3981	0.2269	0.2962	0.8567	0.5015	0.7038	0.3471	0.5017
0.1433	0.4490	0.2760	0.3471	0.1433	0.1965	0.7038	0.3981	0.5027
0.1433	0.5000	0.3033	0.3471	0.1943	0.2079	0.7038	0.4490	0.5250
0.1433	0.5510	0.3973	0.3471	0.2452	0.2180	0.7038	0.5000	0.5438
0.1433	0.6019	0.4694	0.3471	0.2962	0.2367	0.7038	0.5510	0.6463
0.1433	0.6529	0.4858	0.3471	0.3471	0.2976	0.7038	0.6019	0.7437
0.1433	0.7038	0.4985	0.3471	0.3981	0.3501	0.7038	0.6529	0.7633
0.1433	0.7548	0.4993	0.3471	0.4490	0.4168	0.7038	0.7038	0.7852
0.1433	0.8057	0.4996	0.3471	0.5000	0.4779	0.7038	0.7548	0.8048
0.1433	0.8567	0.5000	0.3471	0.5510	0.4889	0.7038	0.8057	0.8244
0.1943	0.1433	0.1373	0.3471	0.6019	0.4990	0.7038	0.8567	0.8440
0.1943	0.1943	0.1485	0.3471	0.6529	0.5000	0.7548	0.1433	0.4993
0.1943	0.2452	0.1573	0.3471	0.7038	0.5017	0.7548	0.1943	0.4997
0.1943	0.2962	0.1756	0.3471	0.7548	0.5080	0.7548	0.2452	0.5000
0.1943	0.3471	0.2079	0.3471	0.8057	0.5107	0.7548	0.2962	0.5005
0.1943	0.3981	0.2367	0.3471	0.8567	0.5142	0.7548	0.3471	0.5080
0.1943	0.4490	0.2980	0.3981	0.1433	0.2269	0.7548	0.3981	0.5120
0.1943	0.5000	0.3543	0.3981	0.1943	0.2367	0.7548	0.4490	0.5629
0.1943	0.5510	0.4187	0.3981	0.2452	0.2465	0.7548	0.5000	0.5948
0.1943	0.6019	0.4787	0.3981	0.2962	0.2563	0.7548	0.5510	0.6837
0.1943	0.6529	0.4893	0.3981	0.3471	0.3501	0.7548	0.6019	0.7535
0.1943	0.7038	0.4990	0.3981	0.3981	0.4482	0.7548	0.6529	0.7820
0.1943	0.7548	0.4997	0.3981	0.4490	0.4743	0.7548	0.7038	0.8048
0.1943	0.8057	0.5000	0.3981	0.5000	0.4973	0.7548	0.7548	0.8295
0.1943	0.8567	0.5004	0.3981	0.5510	0.4983	0.7548	0.8057	0.8427
0.2452	0.1433	0.1444	0.3981	0.6019	0.5000	0.7548	0.8567	0.8556
0.2452	0.1943	0.1573	0.3981	0.6529	0.5010	0.8057	0.1433	0.4996
0.2452	0.2452	0.1705	0.3981	0.7038	0.5027	0.8057	0.1943	0.5000
0.2452	0.2962	0.1952	0.3981	0.7548	0.5120	0.8057	0.2452	0.5003
0.2452	0.3471	0.2180	0.3981	0.8057	0.5213	0.8057	0.2962	0.5010
0.2452	0.3981	0.2465	0.3981	0.8567	0.5306	0.8057	0.3471	0.5107
0.2452	0.4490	0.3163	0.4490	0.1433	0.2760	0.8057	0.3981	0.5213
0.2452	0.5000	0.4052	0.4490	0.1943	0.2980	0.8057	0.4490	0.5813
0.2452	0.5510	0.4371	0.4490	0.2452	0.3163	0.8057	0.5000	0.6457
0.2452	0.6019	0.4880	0.4490	0.2962	0.3537	0.8057	0.5510	0.7020
0.2452	0.6529	0.4920	0.4490	0.3471	0.4168	0.8057	0.6019	0.7633
0.2452	0.7038	0.4995	0.4490	0.3981	0.4743	0.8057	0.6529	0.7921
0.2452	0.7548	0.5000	0.4490	0.4490	0.4887	0.8057	0.7038	0.8244
0.2452	0.8057	0.5003	0.4490	0.5000	0.4990	0.8057	0.7548	0.8427
0.2452	0.8567	0.5007	0.4490	0.5510	0.5000	0.8057	0.8057	0.8515
0.2962	0.1433	0.1560	0.4490	0.6019	0.5017	0.8057	0.8567	0.8627
0.2962	0.1943	0.1756	0.4490	0.6529	0.5111	0.8567	0.1433	0.5000
0.2962	0.2452	0.1952	0.4490	0.7038	0.5250	0.8567	0.1943	0.5004
0.2962	0.2962	0.2148	0.4490	0.7548	0.5629	0.8567	0.2452	0.5007
0.2962	0.3471	0.2367	0.4490	0.8057	0.5813	0.8567	0.2962	0.5015
0.2962	0.3981	0.2563	0.4490	0.8567	0.6027	0.8567	0.3471	0.5142
0.2962	0.4490	0.3537	0.5000	0.1433	0.3033	0.8567	0.3981	0.5306
0.2962	0.5000	0.4562	0.5000	0.1943	0.3543	0.8567	0.4490	0.6027
0.2962	0.5510	0.4750	0.5000	0.2452	0.4052	0.8567	0.5000	0.6967
			0.5000	0.2962	0.4562	0.8567	0.5510	0.7240

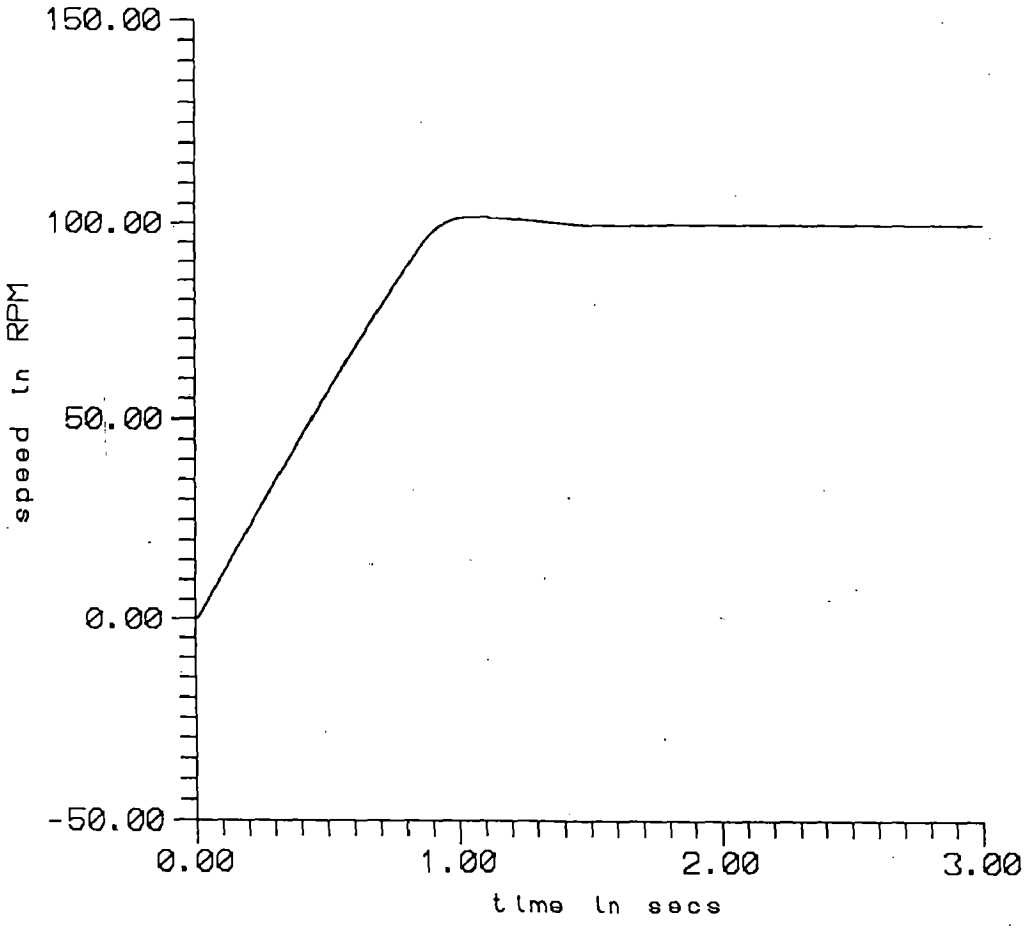


Fig. 3.6 Speed response at 100 rad/sec as reference speed

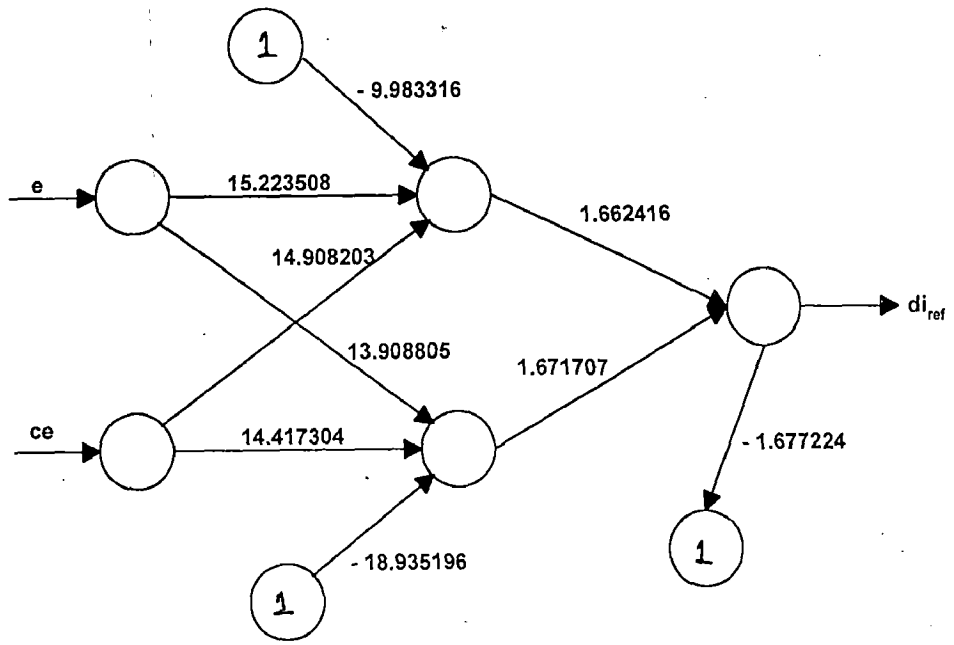


Fig. 3.8 : Trained Network for speed controller

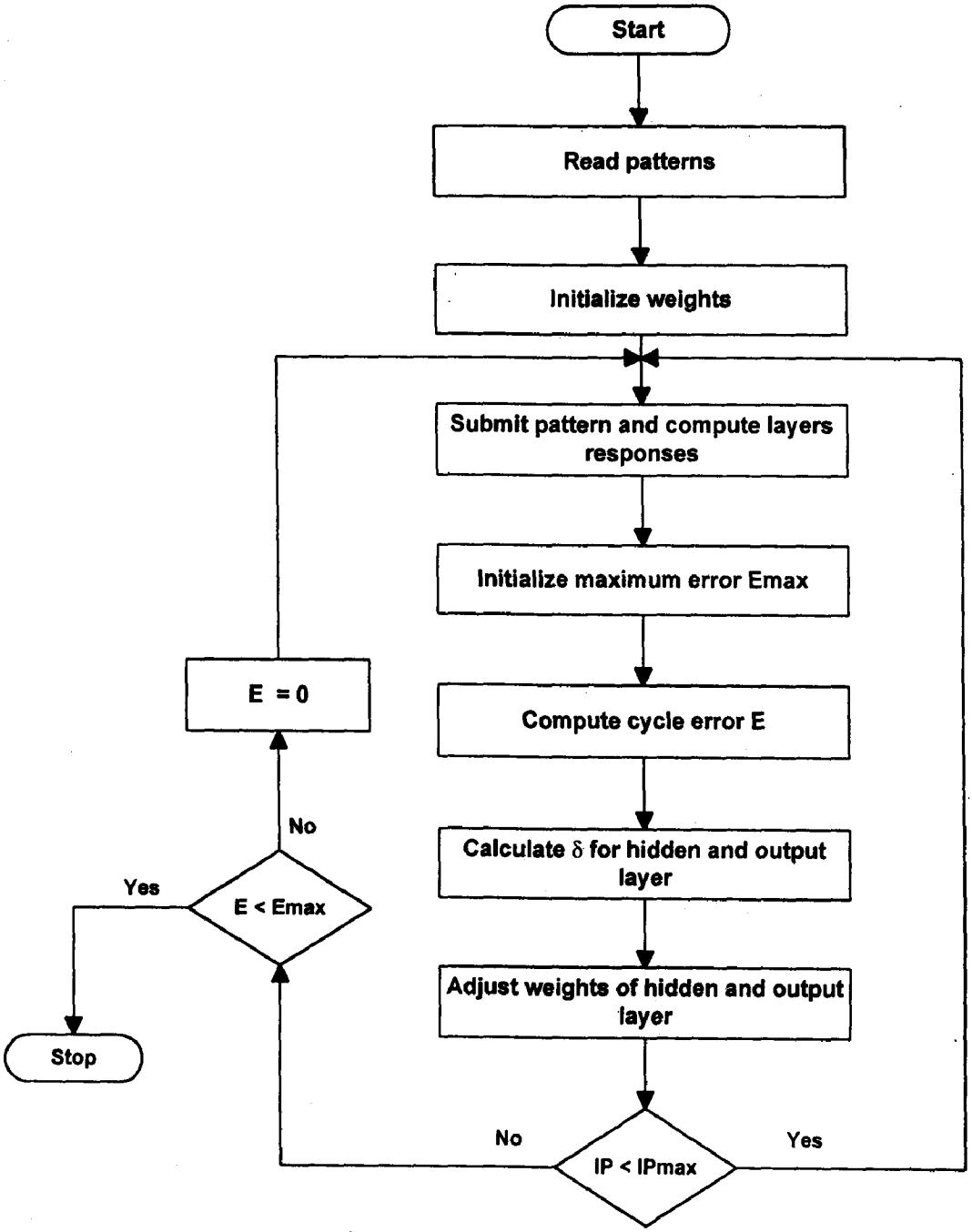


Fig. 3.7 : Flow chart for training of neural network

3.3 CONCLUSION

The desired response, i.e., settling time and overshoot of the dc motor drive system can be achieved using closed loop control scheme. The speed controller is implemented using feed forward neural network. The network is trained using error-back propagation technique. The patterns for training the neural network are obtained using the fuzzy logic control.

SYSTEM HARDWARE AND DESIGN ASPECTS

The speed of the dc motor is controlled using 3-phase fully controlled bridge converter. The firing pulses are generated using 8031 microcontroller based system. The complete system is described in this chapter in detail. It consists of following blocks.

1. Power circuit
2. Snubber circuit
3. Pulse amplifier and firing circuit
4. Quantizer and zero crossing generating circuit
5. Power supply
6. Speed measurement circuit
7. Current measurement circuit
8. Dedicated 8031 micro controller circuit

4.1 POWER CIRCUIT CONFIGURATION AND ITS WORKING

Fig. 4.1 shows a three phase fully controlled bridge converter circuit. Thyristors are connected to phases R,Y,B denoted by T_1, T_2, T_3, T_4, T_5 and T_6. Each thyristor conducts for 120^0 and at a time two of the converters are conducting one from upper circuit and other form lower circuit. If phase sequence is R, Y, B, T_1 commutates with T_3, T_3 with T_5 with T_1. On the other half T_6 commutates with T_2, T_2 with T_4 and T_4 with T_6. All thyristors are line commutated, i.e., whenever a thyristor is triggered, line-to-line voltage appear across the outgoing thyristor and reverse biases it. The thyristors are triggered in a sequence 1,2,3,4,5,6. The conducting pairs are (6,1), (1,2), (2,3), (3,4), (4,5), (5,6). Each thyristor is triggered for full 120° period. Firing pulses at any firing angle α are shown in Fig. 4.2.

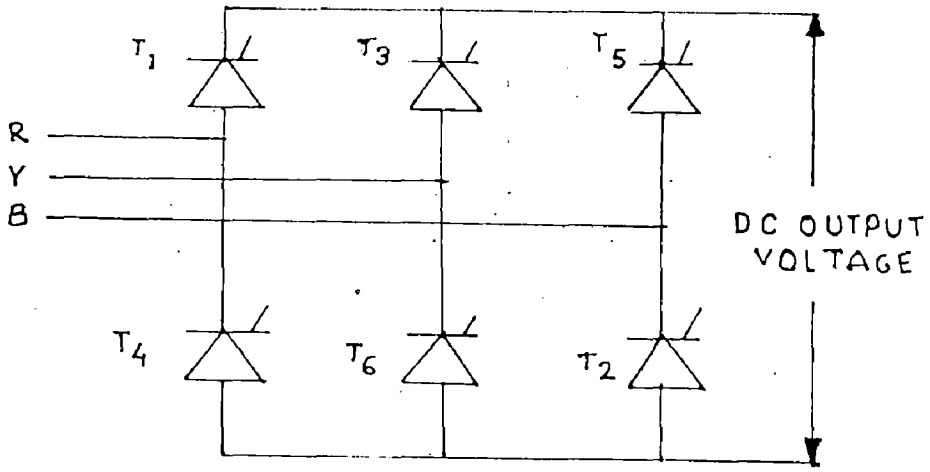


Fig. 4.1 Power circuit

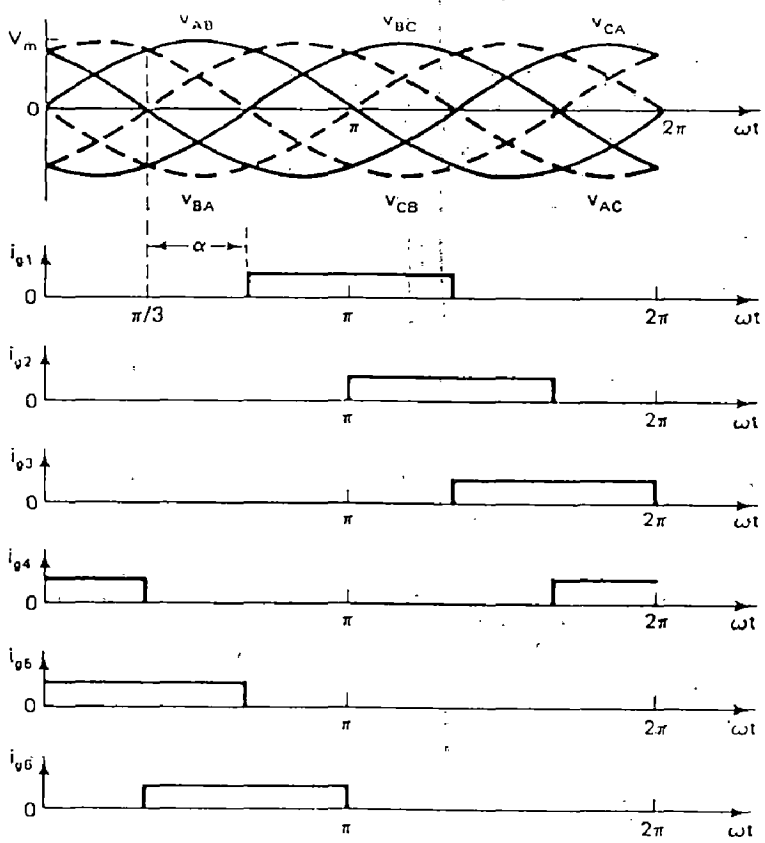


Fig. 4.2 Firing pulses for thyristor at firing angle α

Commutated i.e., whenever a thyristor is triggered line-to-line voltage appear across the outgoing thyristor and reverse biases it.

4.2 SNUBBER CIRCUIT

Fig. 4.3 shows a thyristor with snubber circuit. The series inductance L is added to limit di/at during turn-on of the device.

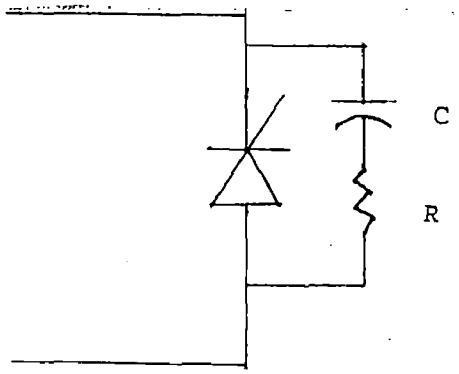


Fig. 4.3 Thyristor with snubber circuit

The capacitor c charges from zero to full voltage V with a slow rate show that dv/at is less than the specified maximum permissible dv/dt rating of the device. When the device is turned on, the capacitor discharges through the device and forces a current equal to v/(resistance in local path formed by c, device and R); thus the resistance R limits the capacitor discharge current at the turn-on of the device. In the actual design of the snubber circuit, the value of R,C and L should be such that dv/at across the capacitor c during its charging is less than the specified dv/at rating of the device within reasonable limits.

The typical values chosen for the snubber circuit are 50 ohms, 5W and 0.1 μf, 600 V respectively. The thyristor chosen have current rating of 15 amps rms current, and voltage rating of 200 volts.

4.3 PULSE AMPLIFIER AND FIRING CIRCUIT

The isolation of power circuit to control circuit is an essential requirement especially in micro controller or microcomputer controlled systems in which sophisticated devices are used which do not have transient with standing

capability. For this isolation purpose pulse transformer is used. The pulses obtained from I/O part of 8031 is amplified and ANDed with high frequency pulses obtained from 555 timer. Then these are amplified and fed to pulse transformer. To avoid saturation of pulse transformer a diode is connected in series with this.

Gate protection is required for overvoltages and over current. The gate can be protected against overvoltage by connecting a diode across to gate and cathode and against overcurrent by connecting a resistance with the input of thyristors. A capacitor is connected across the gate to cathode to bypass the noise pulses. The pulse amplifier circuit with firing circuit is shown in Fig. 4.4.

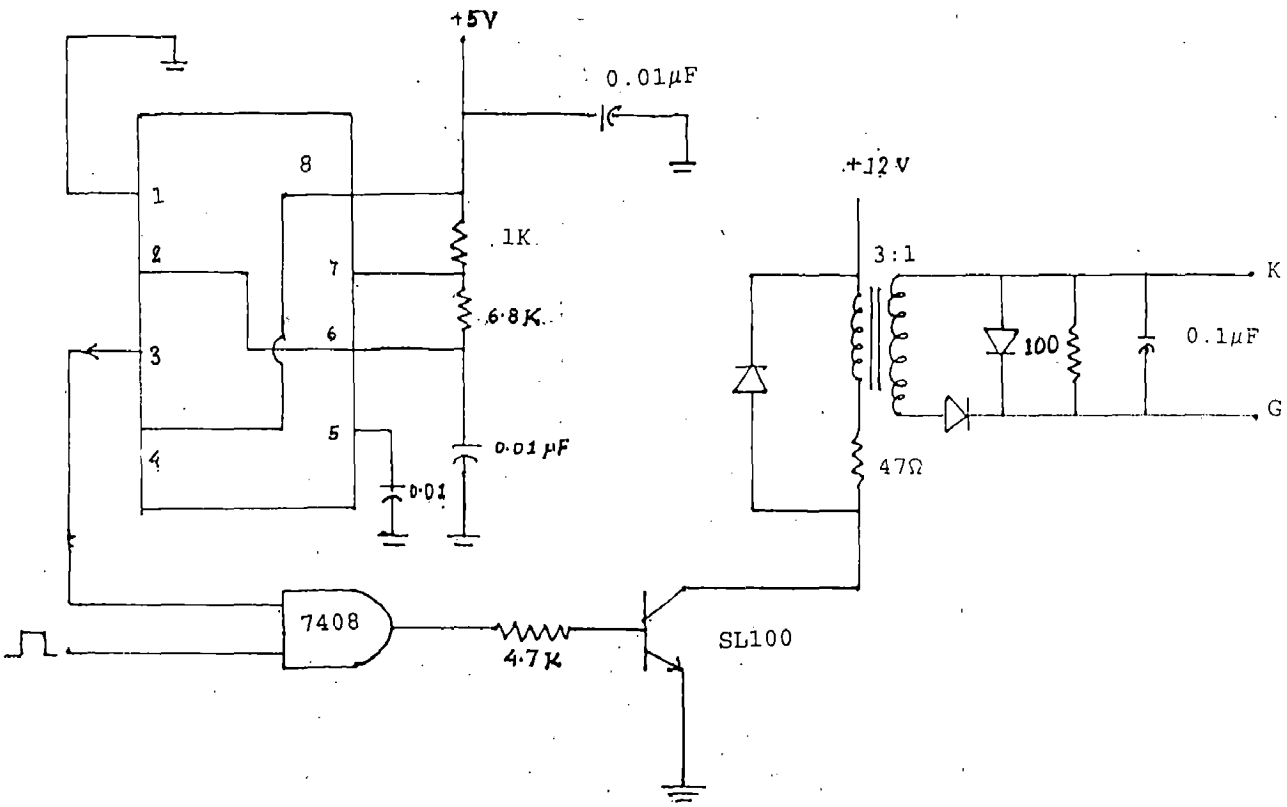


Fig. 4.4 Pulse amplifier and firing circuit

4.4 QUANTIZER AND ZERO CROSSING GENERATING CIRCUIT

To control the firing angle of thyristors, signals that synchronize them with the alternating supply voltages are always necessary. For a three-phase fully controlled signals there are three quantizers or logical signals, each one associated with a thyristor and defined from the natural commutation instant, i.e. at the intersection of two phase voltages, which coincides with the zero crossing of another one. The zero voltage detection fulfils two different functions the synchronization of the thyristor firing circuit to the input line frequency and the phase sequence identification.

To obtain quantizers ϕR, ϕY and ϕB three two winding transformer are used to step down line-to-line voltage at low level, as shown in Fig. 4.5. These signals are amplified and saturated. These quantizers are cophasor to the supply line voltages V_{RY}, V_{YB} and V_{BR} respectively and displaced by 120^0 from each other. The waveforms of these signals with line voltages are shown in Fig. 4.5.

Using these signals, for full range of ac cycle, a firing command data table is prepared. The firing commands are outputted via port 1 of 8031 (P1.0-P1.5) for thyristors T_1 to T_6. Hence the control word to fire thyristor 3 and 4 will be P1.0-P1.5 = XX001100 = OCH

Table 4.1 shows the control words for different pairs of thyristors to be forced.

TABLE 4.1 : FIRING COMMAND DATA

Range of firing angle $\alpha = 0^0$ to 60^0

ϕR	*ϕY*	*ϕB*	Quantizer	*On SCRs*	*Firing command*
1	1	0	6	1,2	00000011=03H
1	0	1	5	5,6	00110000=30H
1	0	0	4	6,1	00100001=21H
0	1	1	3	3,4	00001100=0CH
0	1	0	2	2,3	00000110=06H
0	0	1	1	4,5	00011000=18H

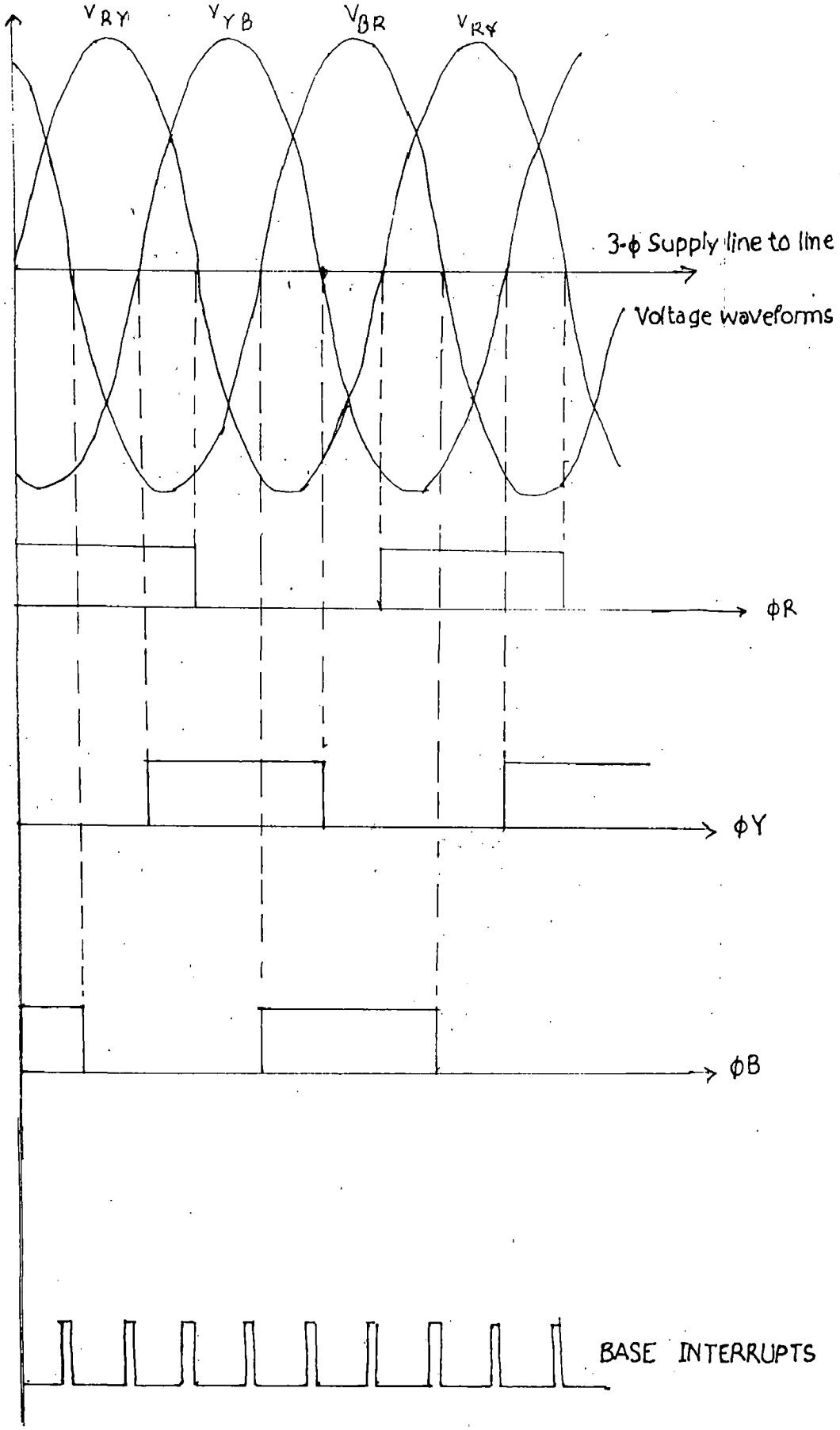


Fig. 4.5 Theoretical waveforms for base interrupt generation circuit

Range of firing angle $\alpha = 60^0$ to 120^0

ϕR	ϕY	ϕB	Quantizer	*On SCRs*	*Firing command*
1	1	0	6	6,1	00100001=21H
1	0	1	5	4,5	00011000=18H
1	0	0	4	5,6	00110000=30H
0	1	1	3	2,3	00000110=06H
0	1	0	2	1,2	00000011=03H
0	0	1	1	3,4	00001100=0CH

Range of firing angle $\alpha = 120^0$ to 180^0

ϕR	ϕY	ϕB	Quantizer	*On SCRs*	*Firing command*
1	1	0	6	5,6	00110000=30H
1	0	1	5	3,4	00001100=0CH
1	0	0	4	4,5	00011000=18H
0	1	1	3	1,2	00000011=03H
0	1	0	2	6,1	00100001=21H
0	0	1	1	2,3	00000110=06H

The quantizers ϕR, ϕY and ϕB will be read via P3.5, P3.6 and P3.7 of port 3.

74121 non retriggerable monoshot is used to produce two pulses, one at the rising edge other at the falling edge of each digitized signal, for each phase. Combining these signals of each phase, a base interrupt signal is obtained. The base interrupt signal has frequency six times of the ac source. A new firing cycle is started at the falling edge of the base interrupt signal. The waveform for base interrupts are shown in Fig. 4.6.

4.5 POWER SUPPLY

The system requires dc power supply of +12v, -12 +5V. To generate power supply of different voltage levels a multiwinding transformer is used. +5V supply, bridge rectifier circuit is used and for +12V and −12V supply centre tap configuration is used as shown in Fig. 4.7.

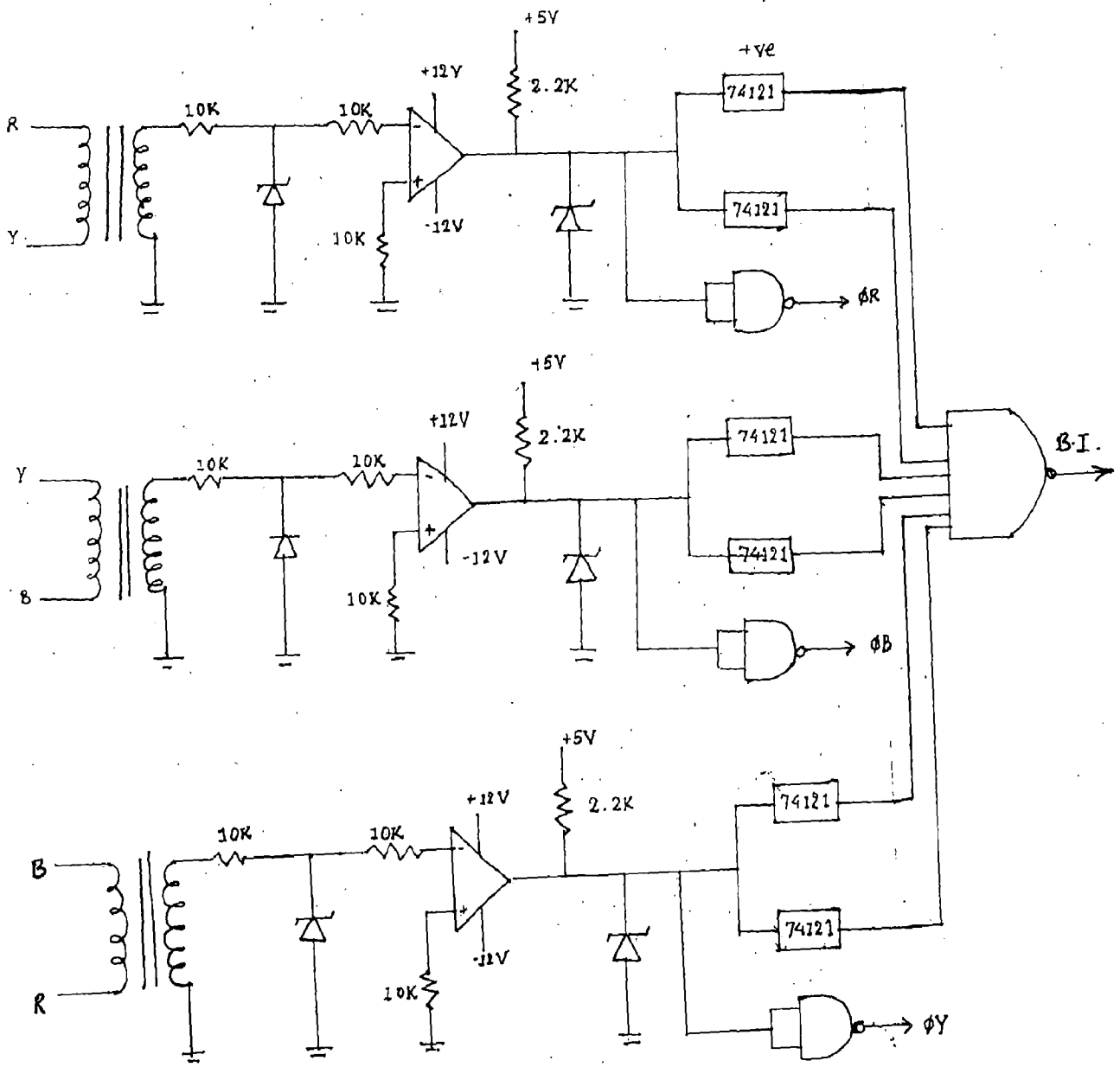
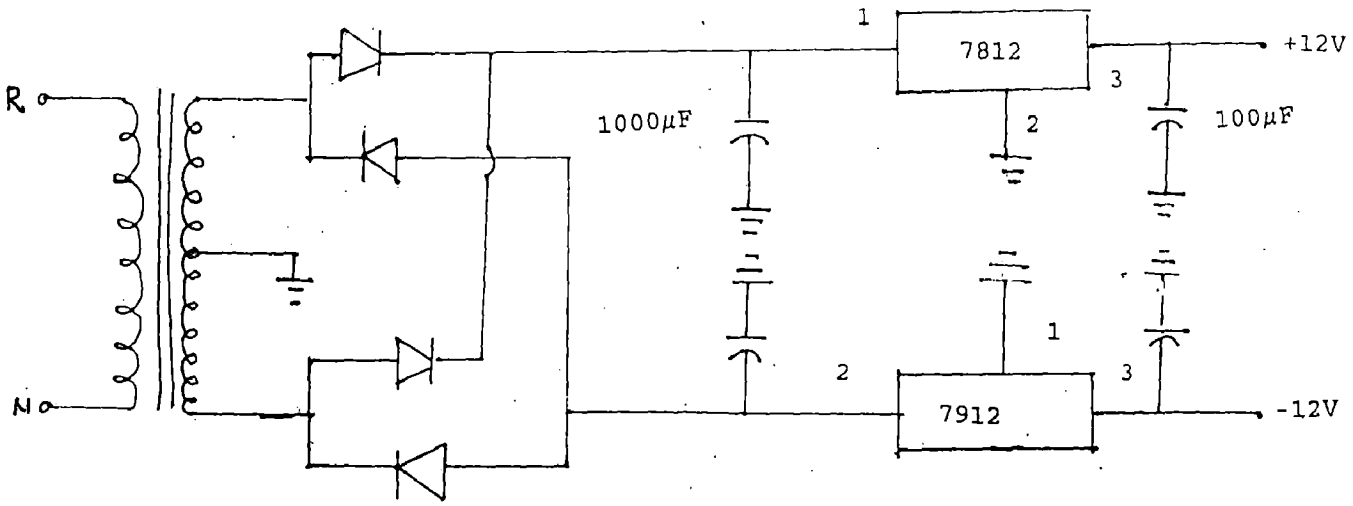


Fig. 4.6 Quantizer and zero crossing generating circuit



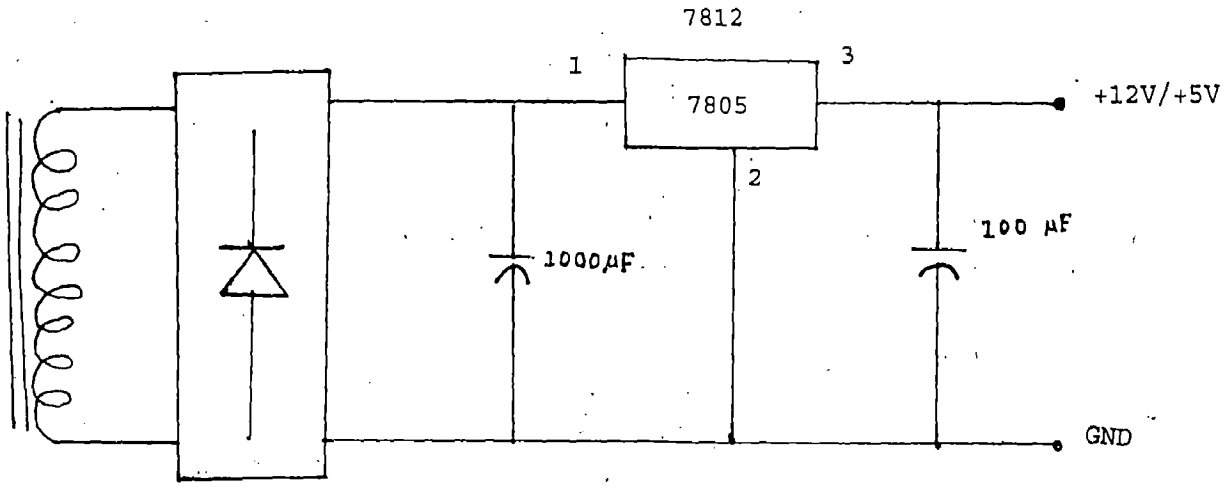


Fig 4.7 Power Supply Circuit

4.6 SPEED MEASUREMENT

The speed of the dc motor is measured using pulse tachogenerator which is coupled to the shaft of motor. Tachogenerator is supplied with +5V. It generates square wave pulses. Frequency of pulses generated depend on the speed of motor. The actual speed of motor is measured with the help of two timers one of which is PC timer and other is timer 0 of ACL-8112 card. PC timer is programmed in mode 3 (square wave mode) while other one is in mode 0 (interrupt on terminal count mode). The speed is measured at every 10msec. But to reduce the time of calculation pulses are counted for a period of 12 msec and then speed is calculated. PC timer generates interrupt at every 1msec instants. Other timer is loaded with all F's when the first interrupt comes it starts down counting and stop downcounting when interrupt comes and calculates the value of speed in bits.

The tachogenerator generates pulses at the rate of 5000 pulses/ revolution at N rpm it will generate 5000 * N pulses/min.

If the rated speed of motor is 1500 rpm then number of pulses generated will be 125000 p/sec. A program is developed in C language to measure the speed. The actual speed and measured speed are given in Table 4.2.

Table 4.2 : Speed measurement

S.No.	Actual speed by tachometer in rpm	Measured speed rpm
1	367	370
2	474	477
3	534	530
4	603	604
5	715	712
6	783	780
7	843	844
8	895	895
9	907	904
10	975	973
11	1036	1033
12	1146	1146

13	1242	1248
14	1345	1344
15	1390	1383
16	1428	1422
17	1458	1459
18	1522	1524

Fig. 4.8 shows the graph between the actual speed and the measured speed through software. The graph is linear one which shows the satisfactory accuracy.

4.7 CURRENT MEASUREMENT CIRCUIT

The scheme to measure current is shown in Fig. 4.9. Hall effect current sensor is used for current measurement. The main advantage of such a transducer is that it is non contact device with high resolution and very small size.

The output of the sensor is current which depends on the number of turns wound on sensor itself and current flowing in the primary circuit. The output current is given by

$I_O = I_L * (N_P / N_S)$

$N_S = 1000$, $N_b = 2$, I_O = Output current, I_L = Load current

$I_O = 2 \times 10^{-3} I_L$

The output current is converted into voltage signal by connecting resistance. After this a low pass filter is used. When current signal is positive, the output of this circuit will be positive, otherwise it will be 0V. Presets are set at 0.5V for 1 Amp.

To convert voltage signal into digital signal ADC of ACDL-811 2HG cand has been used. ADC is B.B. ADS774 successive approximation type, having resolution of 12-bit. This ADC can be programmed in both bipolar and unipolar mode. Since the current obtained from 3-ϕ fully controlled bridge converter is unidirectional, so it is programmed in unipolar mode for present work. The conversion time is 8 μsec. The control word for unipolar mode is given by

	7	6	5	4	3	2	1	0
BASE T09	X	X	X	X	G_3	G_2	G_1	G_0
	0	0	0	0	0	1	0	0
						=	04 H	

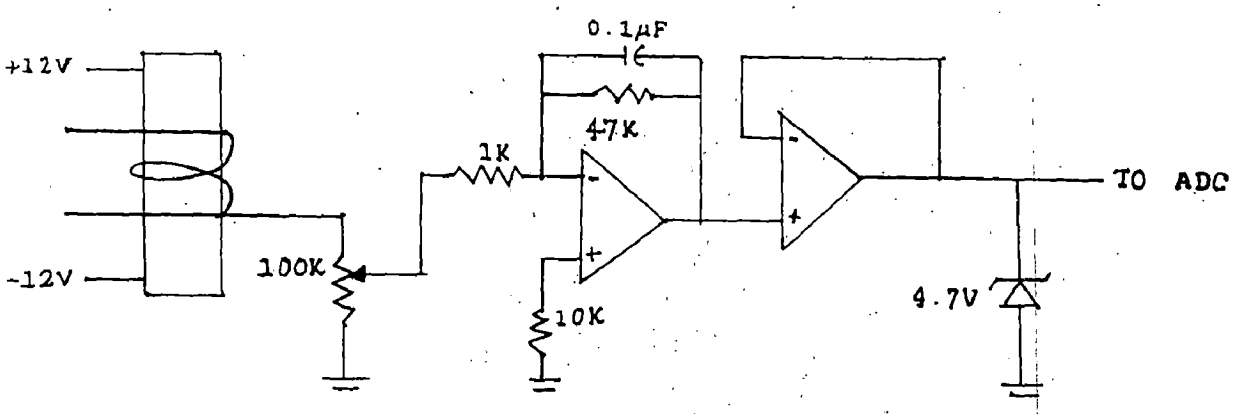


Fig. 4.9 Current measurement circuit

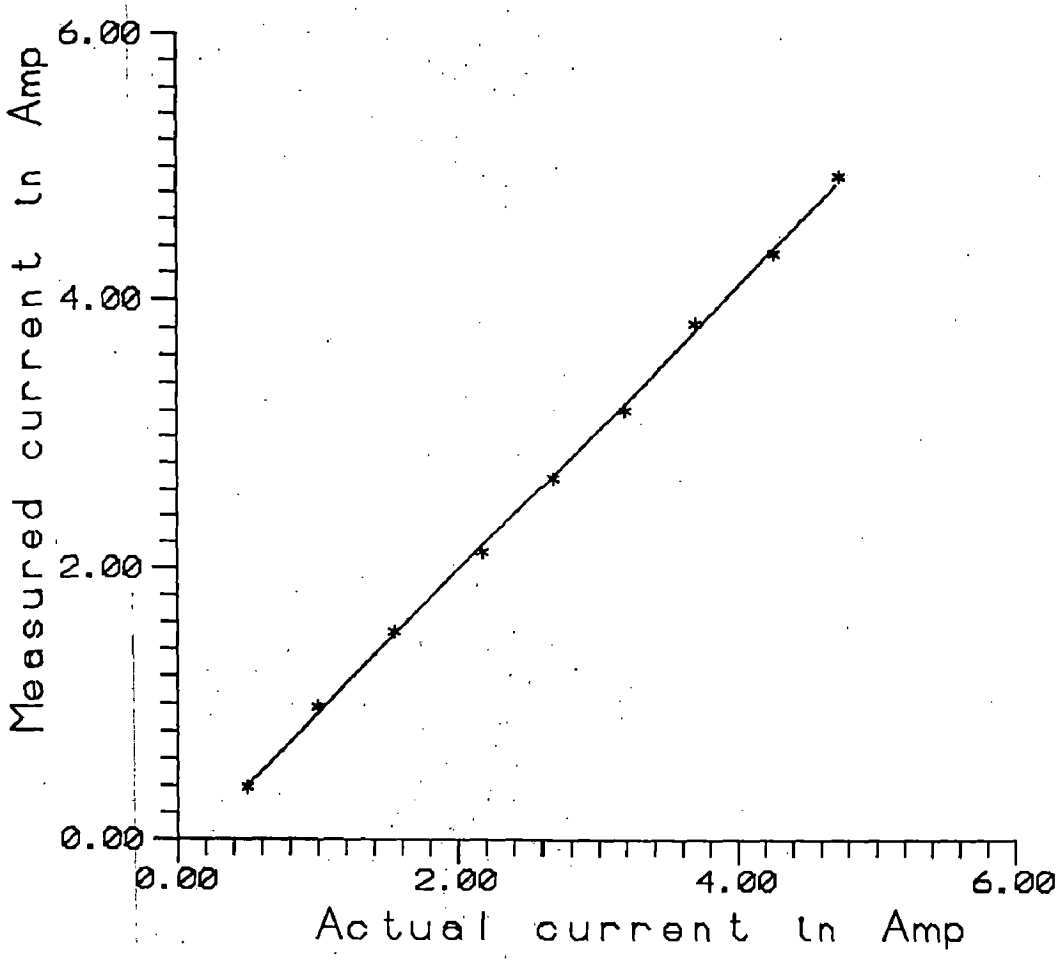


Fig. 4.10 Actual current vs measured current

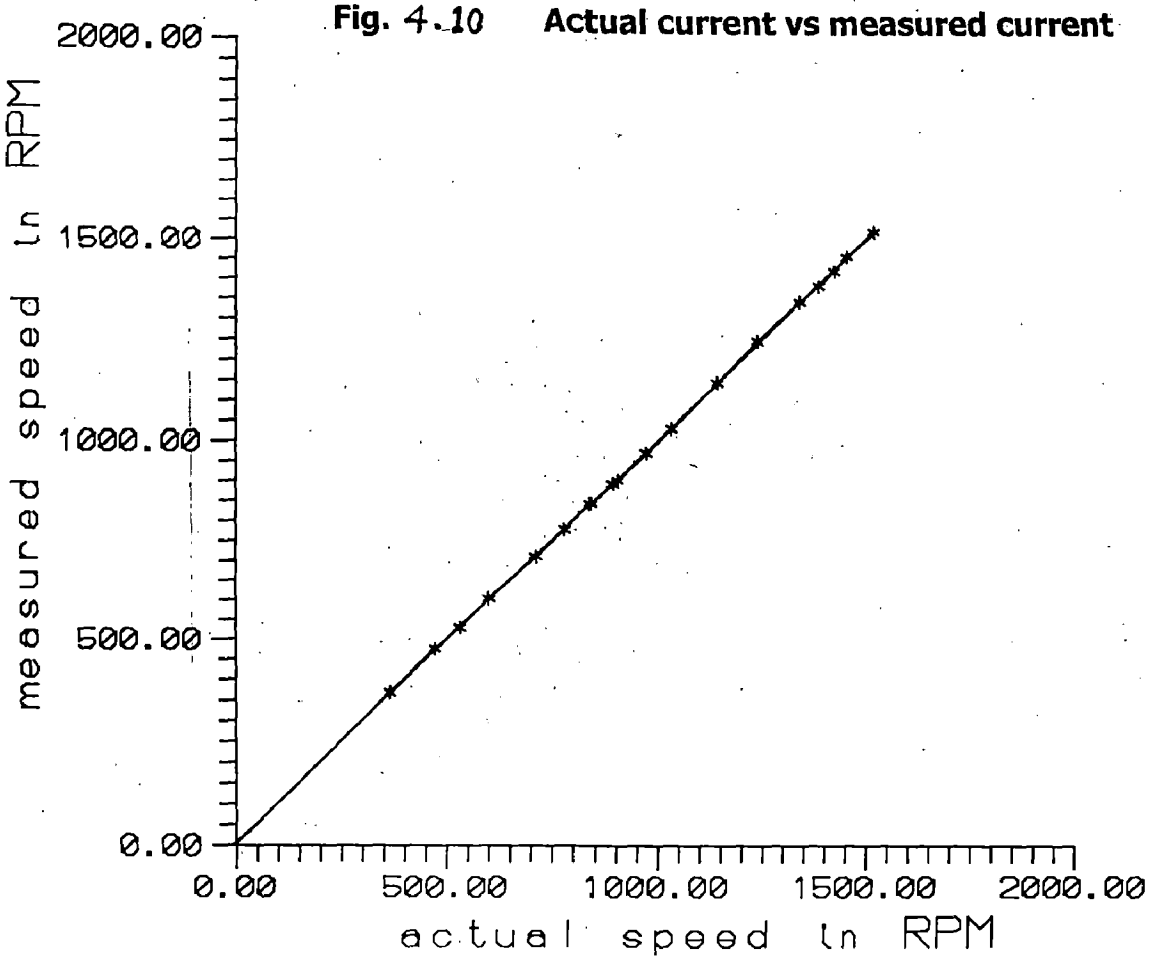


Fig. 4.8 Actual speed vs measured speed

The initial range is 0-10V and gain is 1 for the particular control word.

The key features of ACL-8112 HG are :

- AT HGs
- 16 single ended or 8 differential analog input channels.
- Bipolar or unipolar input signals.
- Programmable high gain
- On-chip sample and hold
- Two 12-bit monolithic multiplying analog output channels
- 16 digital input and output channels
- 3 independent programmable 16-bit down counter compact size.
- Programmable sampling rate of upto 100 kHz.

The experimental results of actual current and measured current are given in Table 4.3. Fig. 4.10 shows graph of actual current versus measured current. The graph shows a high degree of linearity between actual and measured current values.

Table 4.3 : Current measurement

Actual current in Amp.	Measured Current in Hex	Measured current in Amp.
0.50	80	0.390
1.00	200	0.976
1.55	313	1.528
2.18	434	2.120
2.68	550	2.685
3.20	656	3.200
3.70	785	3.830
4.28	890	4.345
4.75	1010	4.930

4.8 MICRO CONTROLLER BASED CARD CIRCUIT

The firing pulses, to turn on thyristors of power circuit are generated using 8031 based microcontroller based card circuit. The interfacing diagram of this circuit is shown in Fig. 4.11. In this circuit, 8031 microcontroller is interfaced with 2764 EPROM, 74LS245 bidirectional buffer, 74LS373 D type latch. Output of buffer is used for firing command of each thyristors. The firing angle, start and stop command are sent serially through PC using RxD. To convert RS-232 level signal into TTL level or vice versa MAX-232 chip is used.

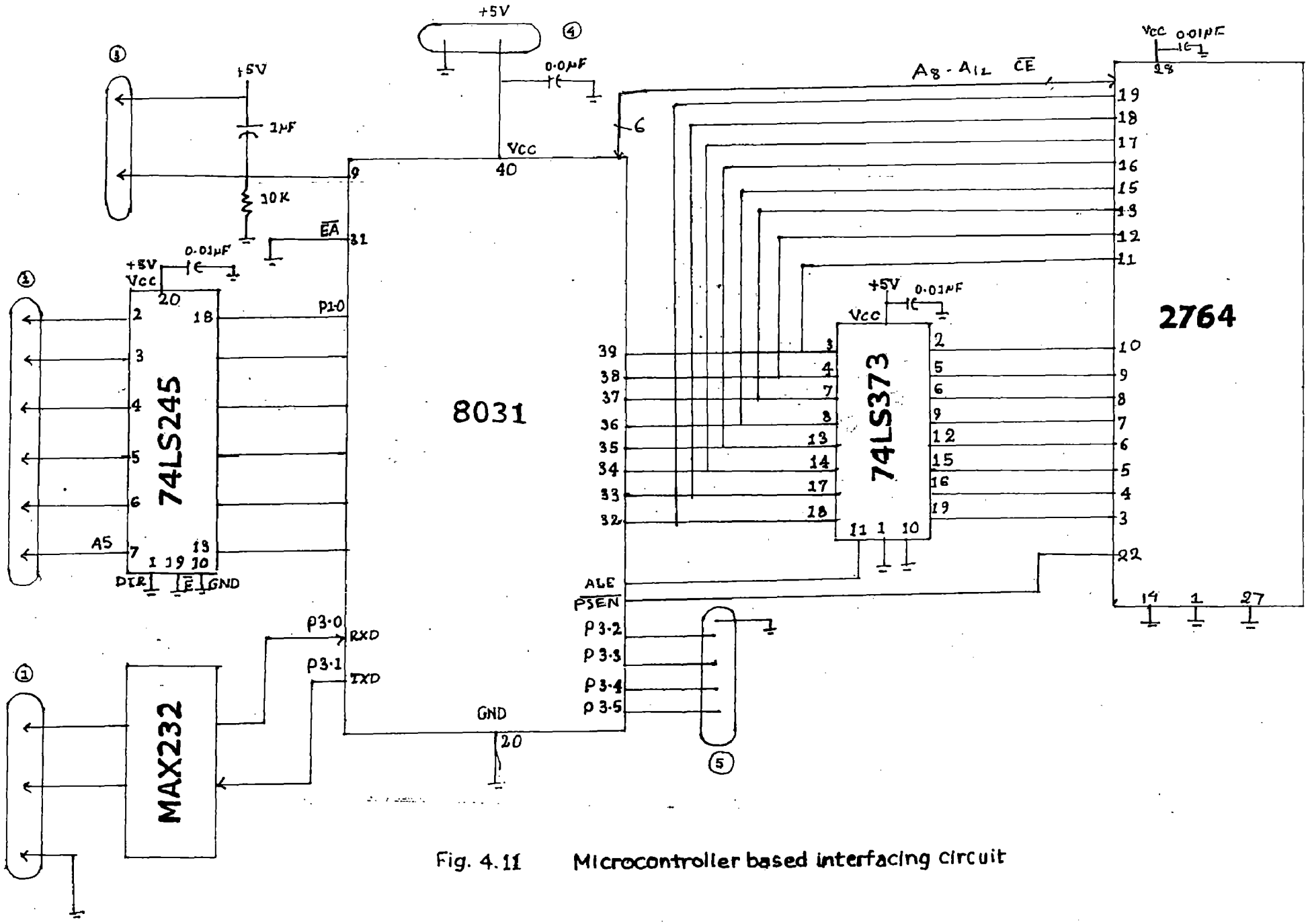


Fig. 4.11 Microcontroller based interfacing circuit

MAX-232 IC contains four sections dual charge pump DC-DC voltage converter, RS-232 drivers, RS-232 receivers and receiver and transmitter enable control inputs.

The two internal charge-pumps convert +5V to +10V on C3 at the V+ output. The second converter uses capacitor C2 to invert +10V to –10V on C4 at the V-output.

2764 is 64 KB erasable and electrically programmable ROM. The important feature of 2764 is separate output control, output enable (OE) from the chip enable ($\overline{E}$) control. The $\overline{OE}$ control eliminates bus contention in multiple bus microprocessor systems.

The 2764 has a stand by mode which reduces the power dissipation without increasing access time. The stand by mode is achieved by applying a TTL high signal to the $\overline{CE}$ input.

The 74LS245 is a bidirectional buffer also octal bus transceiver. This is commonly used as a driver for data bus. The direction of data flow is controlled by the pin DIR.

The 74LS373 is a transparent D-type latch. In this type of latch, when the clock signal is high the output changes according to the input, when the clock goes low, the output will latch the last value of the input.

Port P1 pins P1.0 to P1.5 are used to issue firing commands to six thyristors T1 to T6. Port P3 pins P3.5, P3.4 and P3.3 are used to sense three quantizers ϕR, ϕY and ϕB. INT0 at pin3.2 is used for base interrupt.

4.9 CONCLUSIONS

The design of a 3-ϕ fully controlled bridge converter along with snubber circuit is describes in this chapter. The speed and current measurement circuits are designed and implemented for close loop control. The firing pulses are generated using 8031 microcontroller based card which consists of EPROM 2762 and MAX 232 chip for level conversion. To generate the firing pulses for six thyristors, zero-crossing signals (base interrupts) and three quantizer signals are generated using digital analog circuit.

SYSTEM SOFTWARE

This chapter includes discussion of system software for both open loop mode and closed loop mode. To generate firing commands for 3-ϕ bridge converter assembly language program is developed for 8031 microcontroller. The firing angle obtained in open loop and close loop mode using C programs and transmitted serially to 8031 microcontroller. The description of main program and all subroutines are as given below.

5.1 OPEN LOOP OPERATION

The system software consists of main program, serial subroutine, zero crossing Iss, compare subroutine, zone subroutine and firing Iss.

5.1.1 MAIN PROGRAM

Flow charts for main program is shown in Fig. 5.1. Program starts with initialization of various ports of I/O ports and timer, serial port etc. After initializations all interrupts have been disabled and only serial interrupt has been enabled. Index I, J and CYCLE are also initialized.

The microcontroller continuously monitors RXD line if any character or data is received serially, index I is made 1 in serial link subroutine. If the character received is 'G', then program sets the indices to receive firing angles and lower byte and higher byte one by one serially. If the character received is 'R', then processor enables all the interrupts used to generate the firing pulses. If the received character is 'E', the processor disables all the interrupts and stop using the firing pulses.

Initially the firing angle is kept maximum and it is gradually decreased to increase the output voltage. This controls the starting of DC motor, if used as load

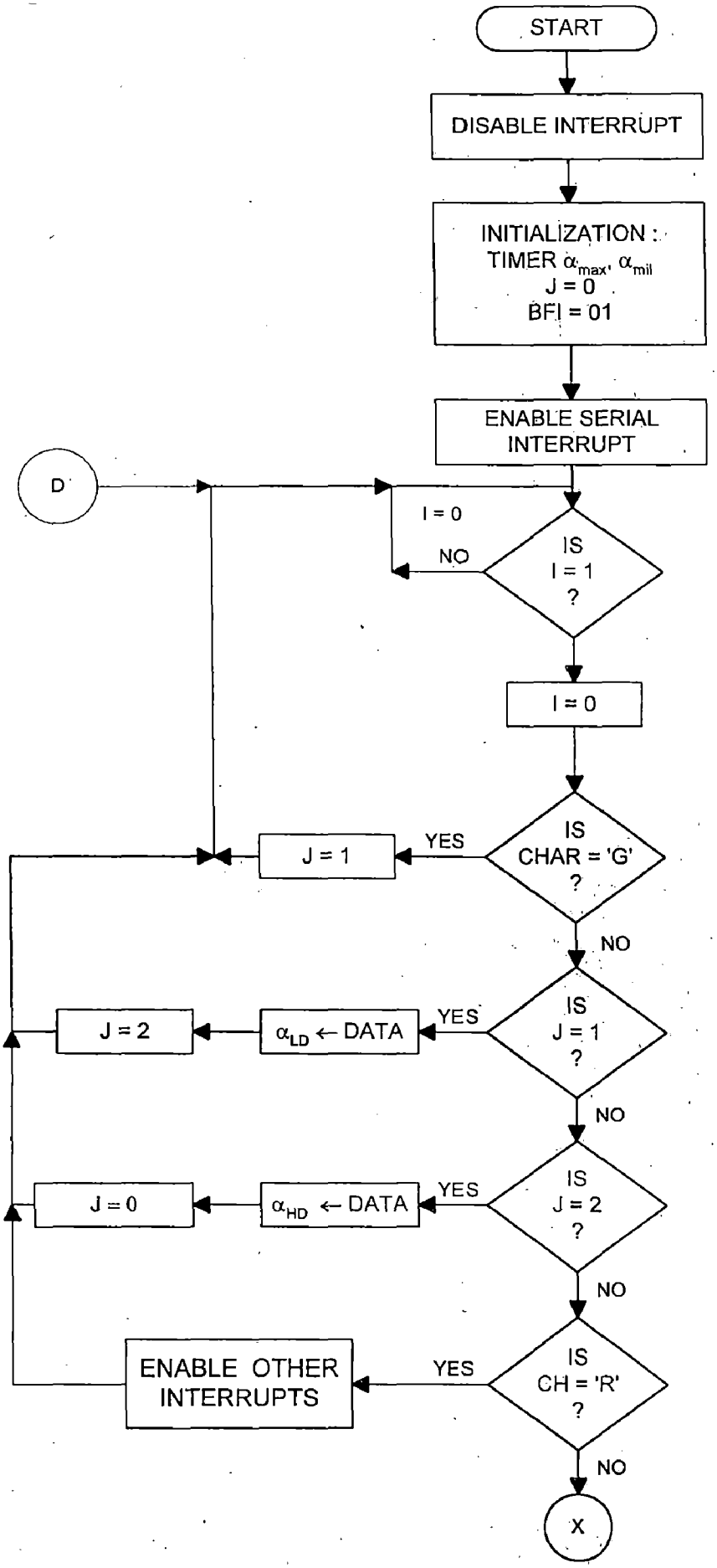
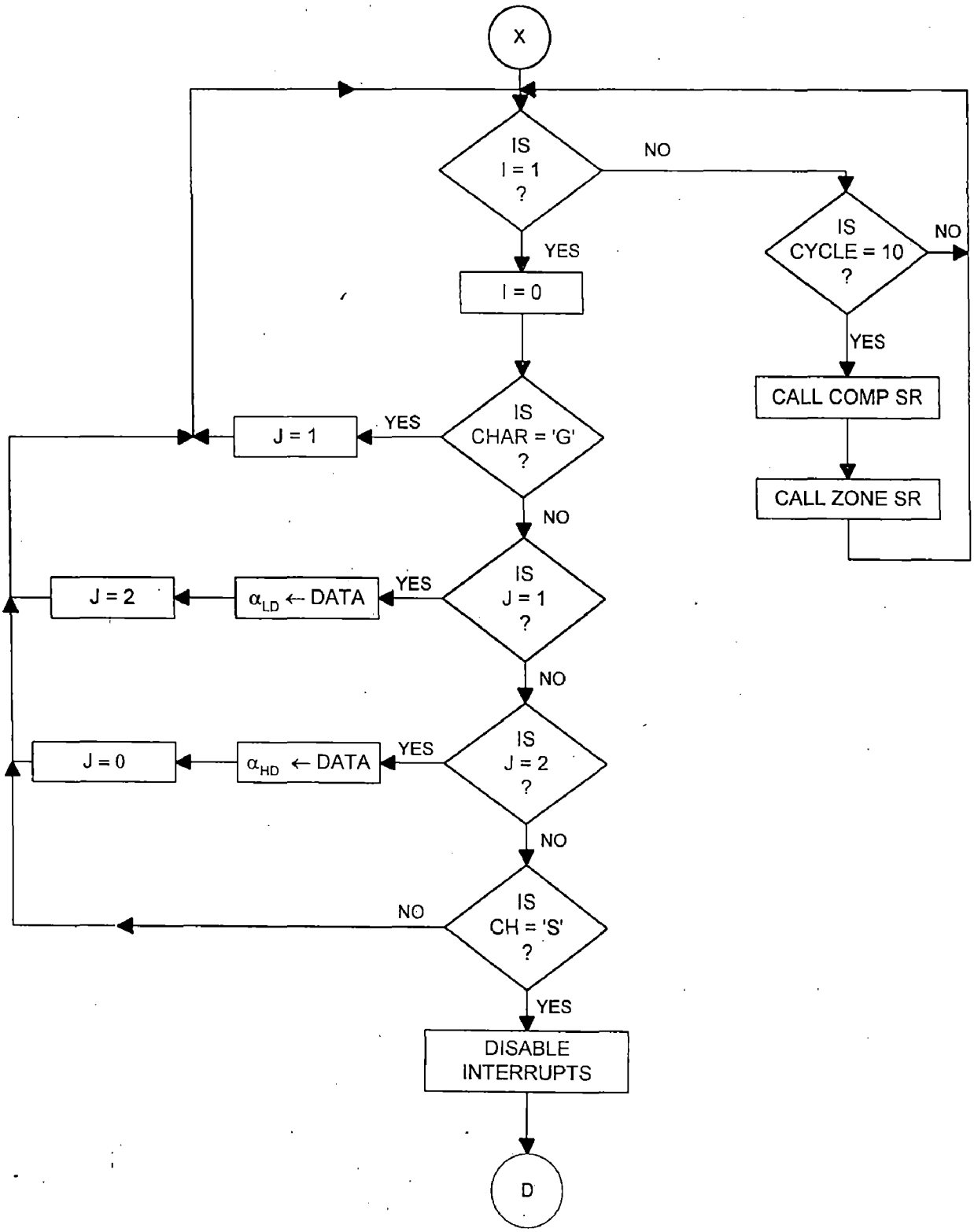


Fig. 5.1 : Main Program for Calculating Firing Angle and Zone in Open Loop Operation (Contd....)


X
IS I = 1 ?
NO
YES
I = 0
IS CYCLE = 10 ?
NO
YES
CALL COMP SR
CALL ZONE SR
IS CHAR = 'G' ?
YES
J = 1
NO
IS J = 1 ?
YES
α_{LD} ← DATA
J = 2
NO
IS J = 2 ?
YES
α_{HD} ← DATA
J = 0
NO
IS CH = 'S' ?
NO
YES
DISABLE INTERRUPTS
D

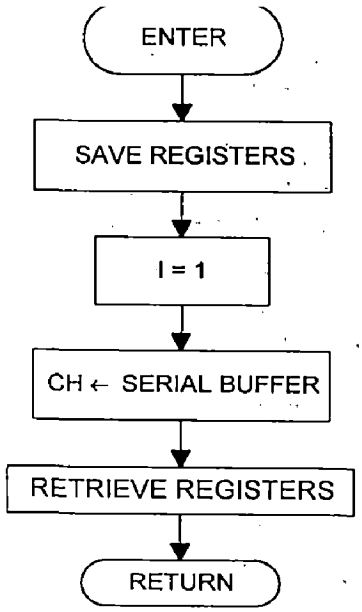


FIG. 5.2 : Serial Subroutine

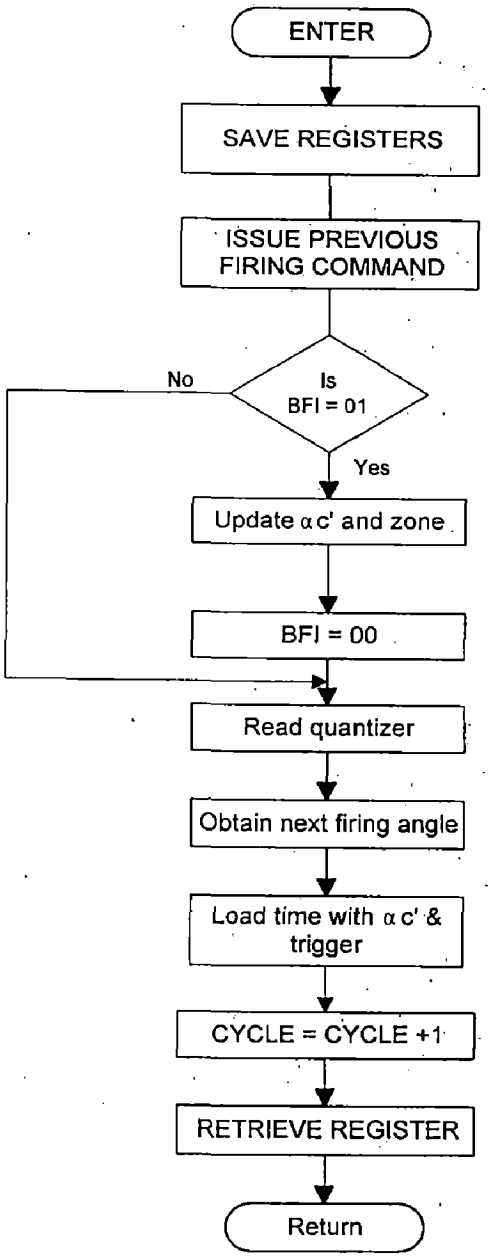


Fig. 5.3 : Zero Crossing Interrupt Subroutine

under running conditions also. The firing angle is changed gradually to control the current in open loop operation of DC motor.

5.1.2 SERIAL SUBROUTINE

Flow chart for this subroutine is shown in Fig. 5.2. After entering in this SR registers are saved. A character/data is received from serial buffer and returned to main program. Before returning to main registers have been retrieved.

5.1.3 ZERO CROSSING INTERRUPT SR

The flow chart for this interrupt service subroutine is shown in Fig. 5.3. The subroutine begins with previous command outputted to port P1. This is done to ensure that if due to some reason, timer interrupt has not come earlier, then the firing pulses will be issued at zero-crossing poin. This situation may come when the α is around 60° or 120°. The timer is loaded with the corrected firing angle (< 60°) and triggered. The quantizer signals are sensed using port P3 bits and used to find out the firing command depending on the zone. Cycle index is incremented to control the acceleration and deceleration.

5.1.4 ZONE SUBROUTINE

In this subroutine, zone and firing angle are calculated. The flow chart for this subroutine is shown in Fig. 5.4. If firing angle is less than 60°, then it is in Zone I. If α_c is greater than 60° but less than 120°, it is in Zone II, for this zone firing angle is calculated by subtracting 60° count from the current firing angle. And if firing angle is less than 180° and greater than 120° it is in Zone III. For this zone, the value of firing angle will be current firing angle subtracting with count for 120°.

5.1.5 FIRING Iss

The flow chart for this interrupt service subroutine is shown in Fig. 5.5. In this subroutine firing commands are issued to thyristor.

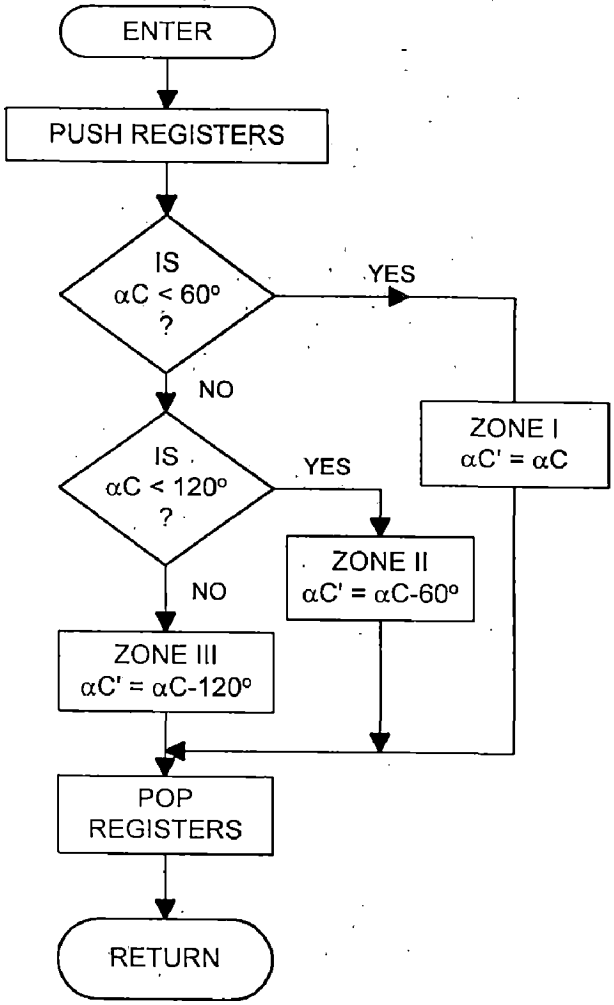


FIG. 5.4 : Zone Subroutine

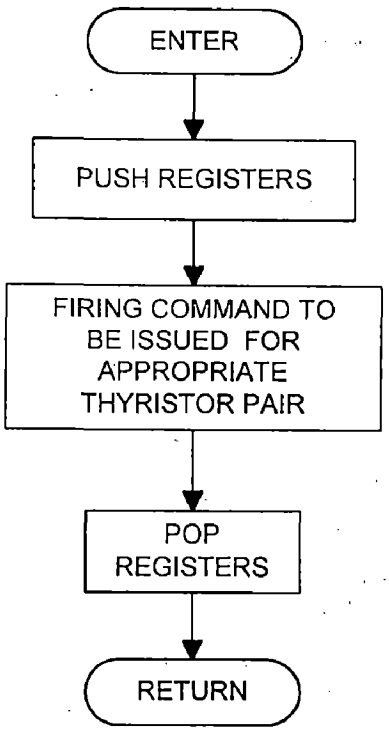


FIG. 5.5 : Firing Interrupt Service Subroutine

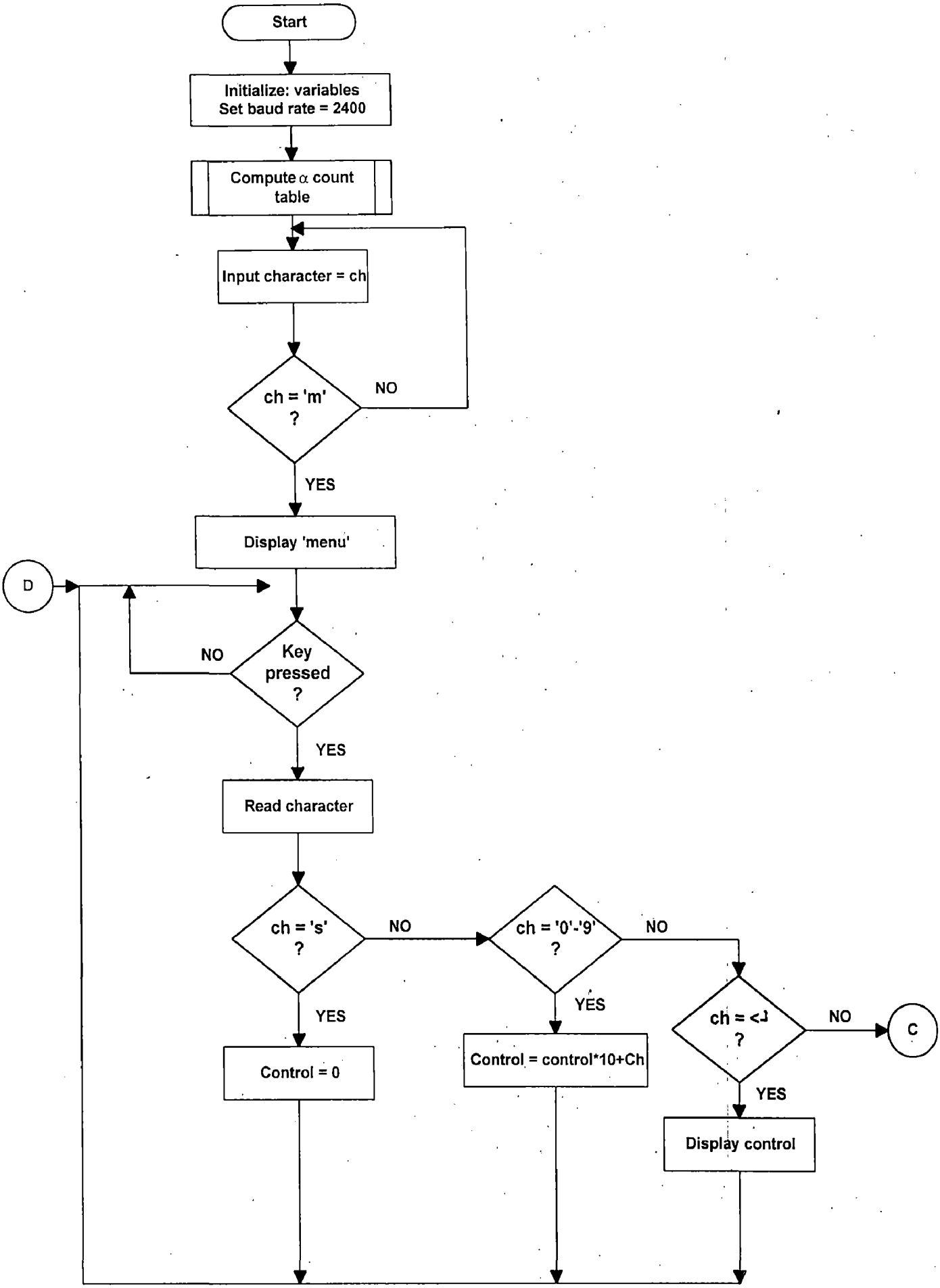
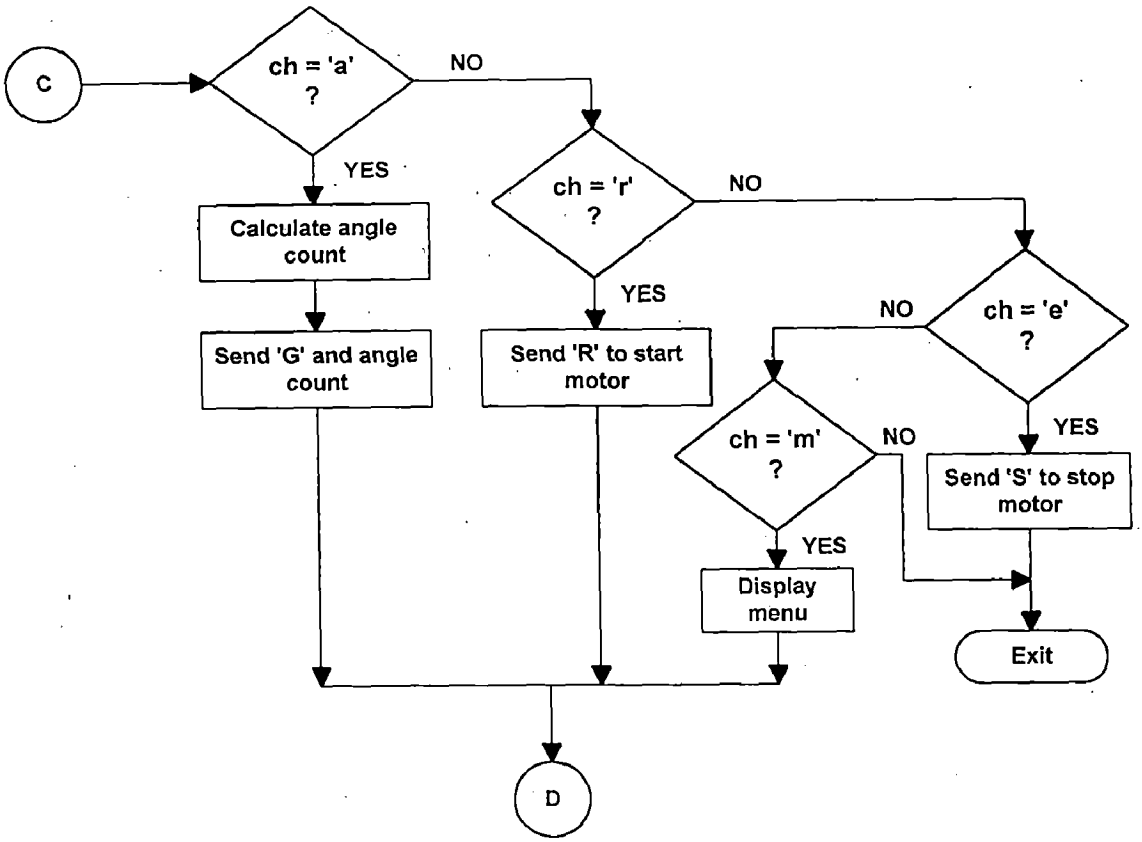


FIG. 5.7 : Flowchart of Main for Open Loop Operation



5.1.8 TABLE

Flowchart for this program is shown in Fig. 5.8. This program is used for calculating firing angle in degrees and also the count value of firing angle.

Firing pulses are generated using cosine firing angle technique. In this technique, the output voltage across the converter is given as

$$V_o = V_{max} \cos\alpha$$

from here, firing angle is calculated as

$$\cos\alpha = \left[\frac{V_o}{V_{max}} \right] = V_{norm}$$

or α (in degrees) $= \cos^{-1}(V_{norm})$

Here, V_{norm} is normalized voltage.

Theoretically, α can be varies from 0° to 180° but practically, only α's value from 10° to 170° can be used.

To get α from 0° to 180°, V_{norm} should be vary from +1 to −1.

To calculate α, for a given control input, it is assumed that control voltage V_c varies from 0 to 2000. The normalized voltage can be calculated as

$$V_{norm} = (V_c / 1000) - 1$$

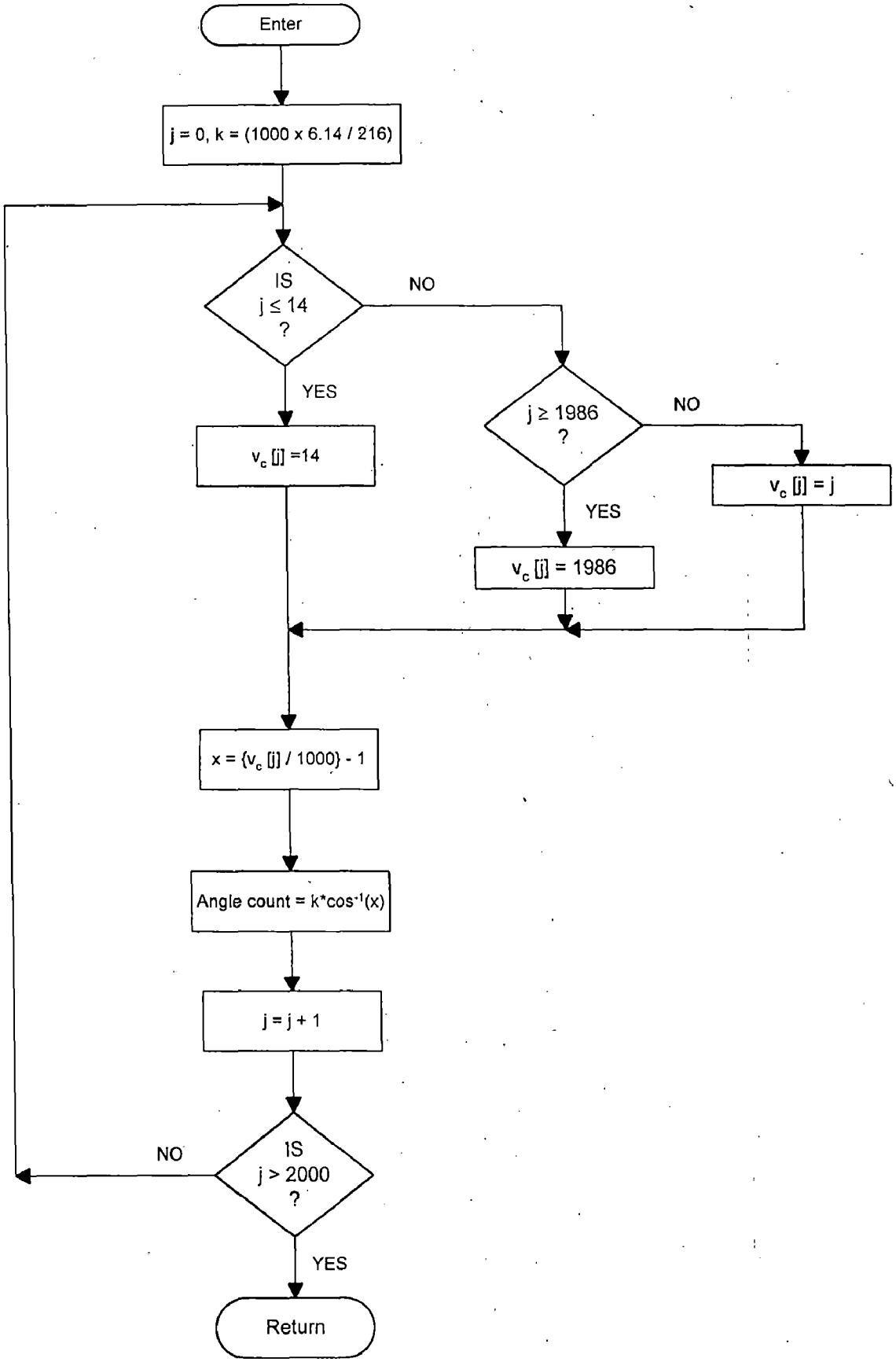


FIG. 5.8 : Flowchart for Calculating α_count table

The count value for firing angle α, can be calculated depending on the clock frequency of microcontroller.

$$\alpha_{count} = (\alpha/180) \times 10 \times 10^{-3} \times (f_{clk}/12) \quad = \alpha \times k$$

where 'k' is a scale factor which is initialized at the starting of the program. The clock frequency of microcontroller is 6.14 MHz. So the value of k is

$$k = (100 \times 6.14 / 216)$$

5.2 CLOSED LOOP OPERATION

In closed loop operation inner current and outer speed loop are used. Speed-controller are designed using artificial neural network. Current controller is a simple proportional controller. These loops are realized through software in C language and a control signal is generated. The control signal decides the new firing angle as in open loop and transmits it to the microcontroller bases system.

The system software for 3-ϕ fully controlled bridge converter in closed loop mode is discussed below which consists of main program, serial subroutine, zone subroutine, zero crossing ISS and firing ISS.

5.2.1 PROGRAM FOR FIRING ANGLE GENERATION IN CLOSED LOOP MODE

The flowchart for this is shown in Fig. 5.9. This is almost same as in the case of open loop. The difference is that in this case due to closed loop operation the firing angle at which the converter is to be operated at a particular instant is send directly from the PC. Therefore, there is no need to gradually increase or decrease firing angle. All other ISS are same as of open loop mode.

The system software is developed in C language. This includes main program with different subroutines for measurement of speed current, calculating the firing angle, reading the weights, calculating the actual speed and current. The main program and subroutines are discussed below.

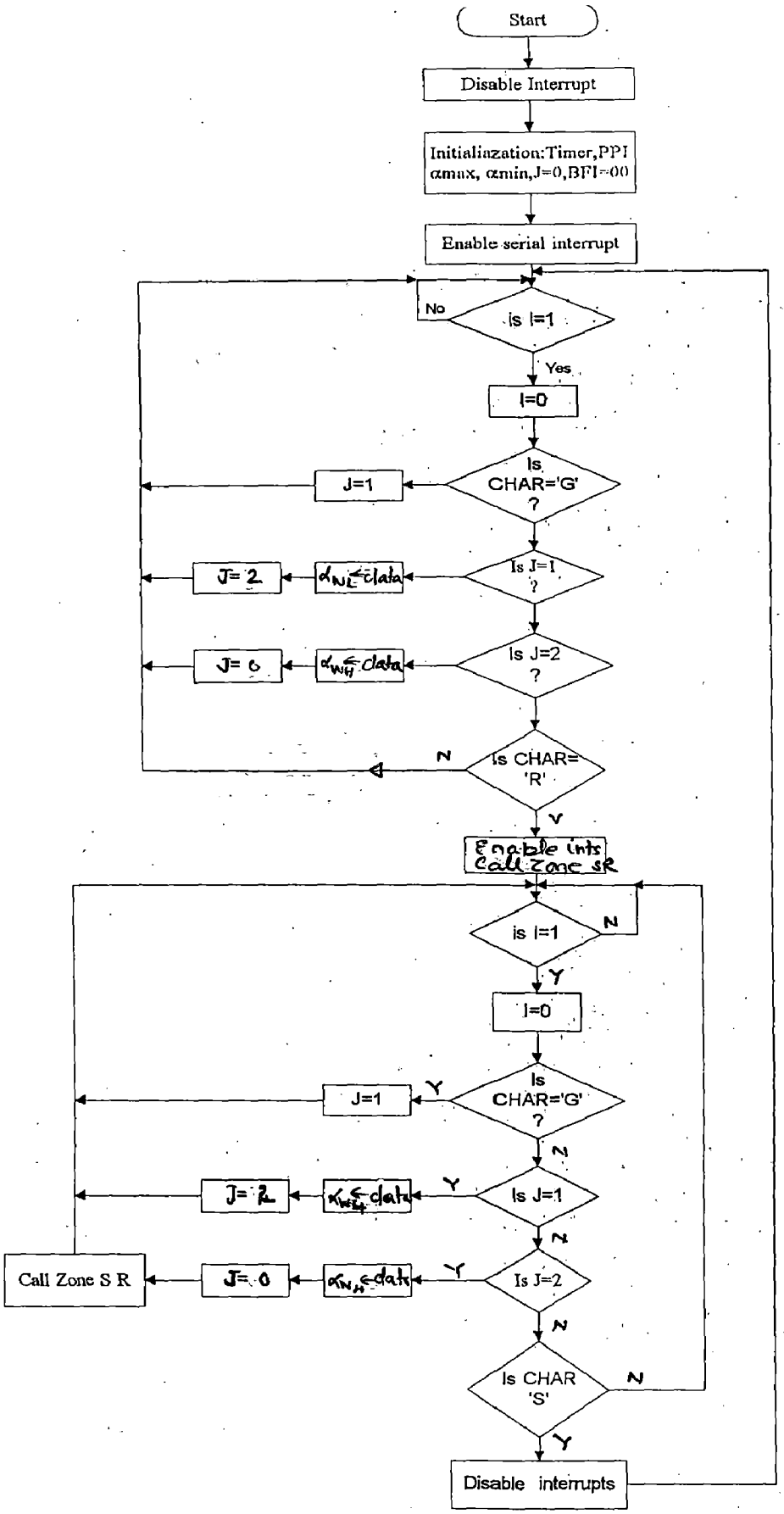


Fig. 5.9 Flow chart for calculating firing angle in closed loop mode

5.2.2 MAIN PROGRAM

The flow chart for main program is shown in Fig. 5.10. The first step is to initialize all variables and baud rate is set as 2400. All interrupts are disabled. PC timer is initialize in mode 3 and timer 0 in mode 0. α-count table has been calculated using α-count table subroutine. Weights for neural controller of speed error processing have been read, which have been already adjusted by off-line training of neural network.

Interrupts are enabled. Program checks for pressing of key. If any key pressed then it reads input character and if character is `m' then displays menu. For any other character it goes back and again search for key pressed and any character.

After displaying menu it again check for key pressed and read input character if character is 's' it sets speed equals to zero and then search for the next instant of key pressing. If character is 'a' instead of 's' then it updates speed and goes back. On pressing the enter key reference speed has been displayed.

In the presence of 'a' character speed loop and current loop are called. Firing angle has been calculated and transmitted. If character 'r' is pressed then `R' is send to start the motor.

After sending the command to start the motor PC timer is initialize to generate 1msec interrupts for measurement of speed. Program checks for the value of KK if it is 0 then goes back, if it is 2 then new speed is measured and speed loop processing is done to generate new current reference. Then, the current is measured and current loop processing is done to obtain new control signal and hence firing angle. Firing angle in terms of angle count is calculated and transmitted. If KK = 1, only current loop processing is done.

5.2.3 SUBROUTINE TO READ WEIGHTS FOR NEURAL CONTROLLER

The flow chart for this subroutine is shown in Fig. 5.11. After initialization weights for input hidden layer and hidden output layer has been read. The weights read are adjusted weighted obtained from the off-line training of neural network.

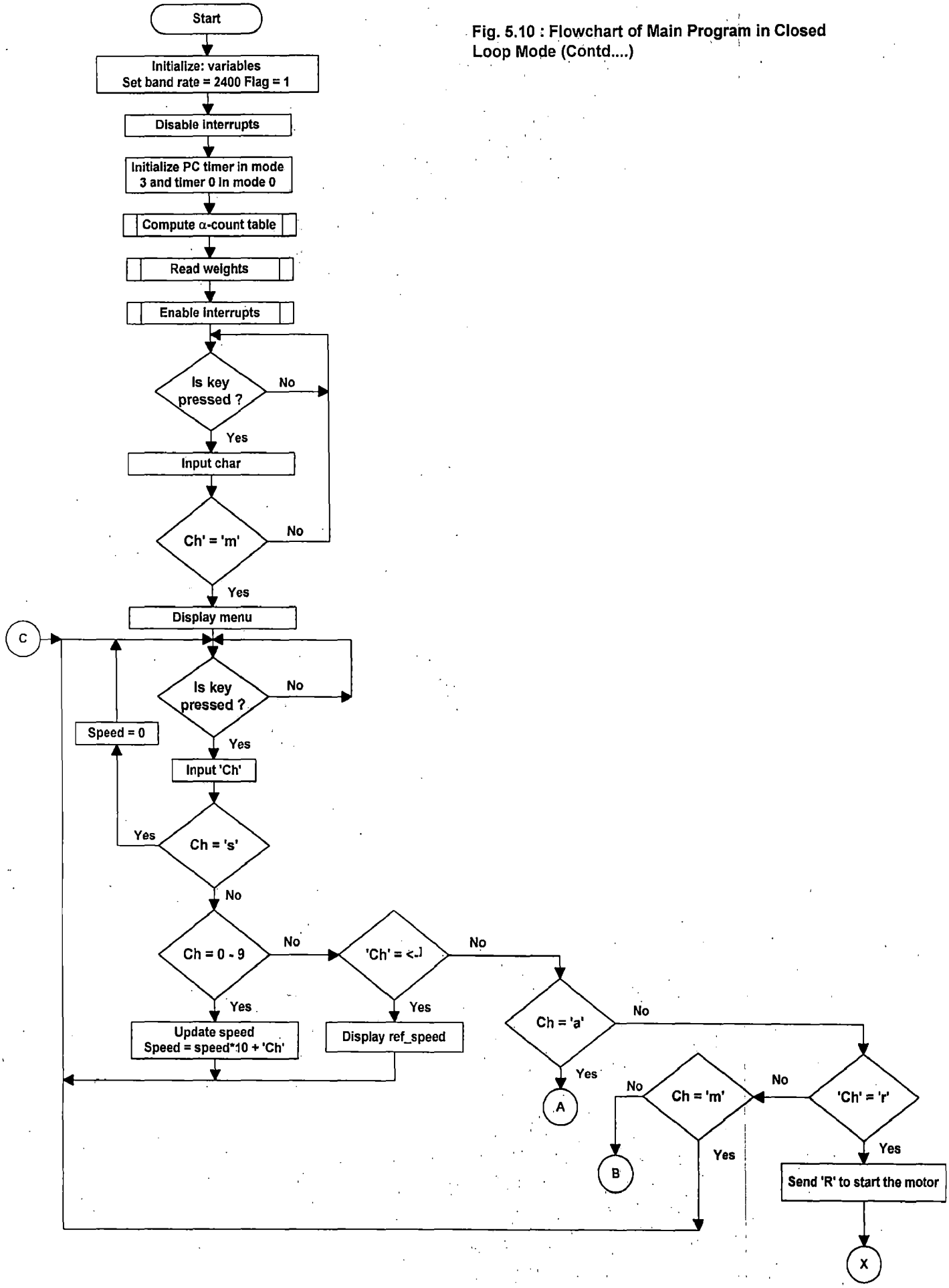
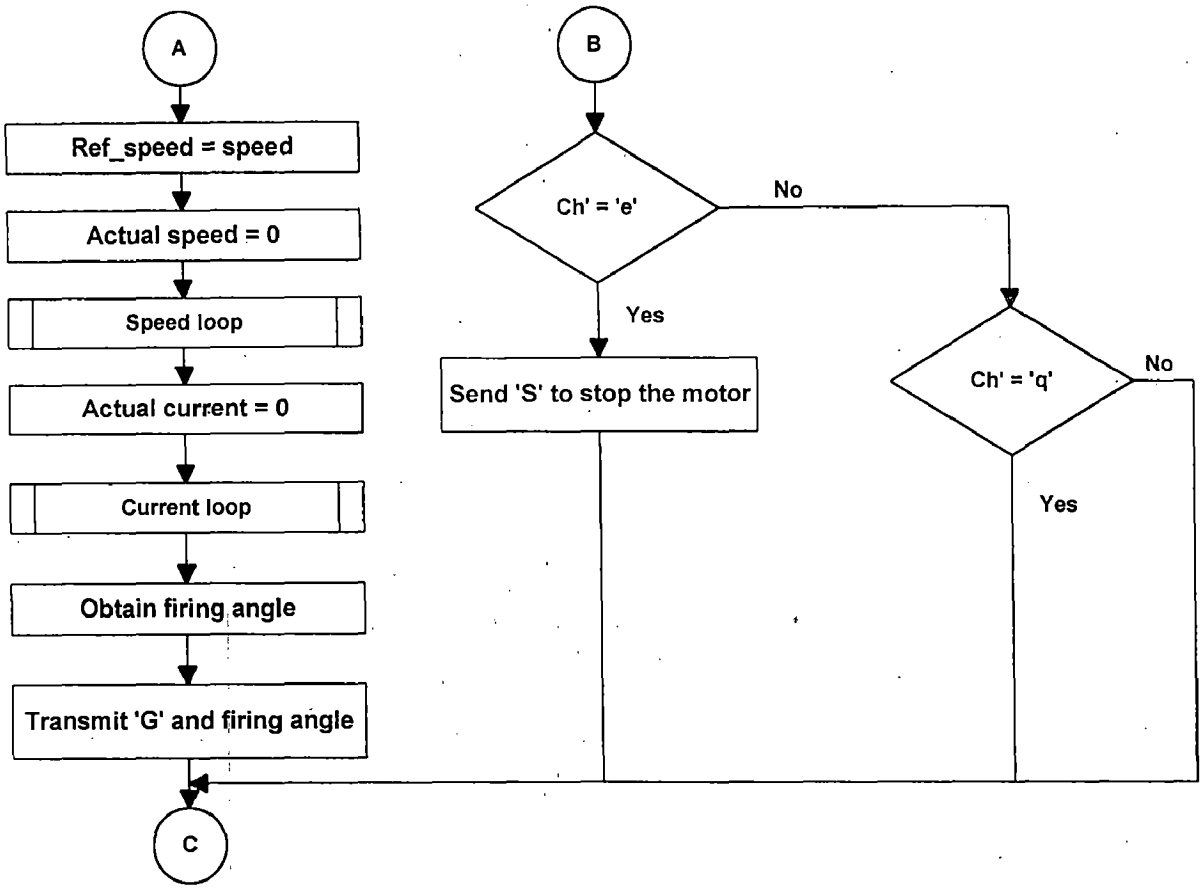


Fig. 5.10 : Flowchart of Main Program in Closed Loop Mode (Contd....)



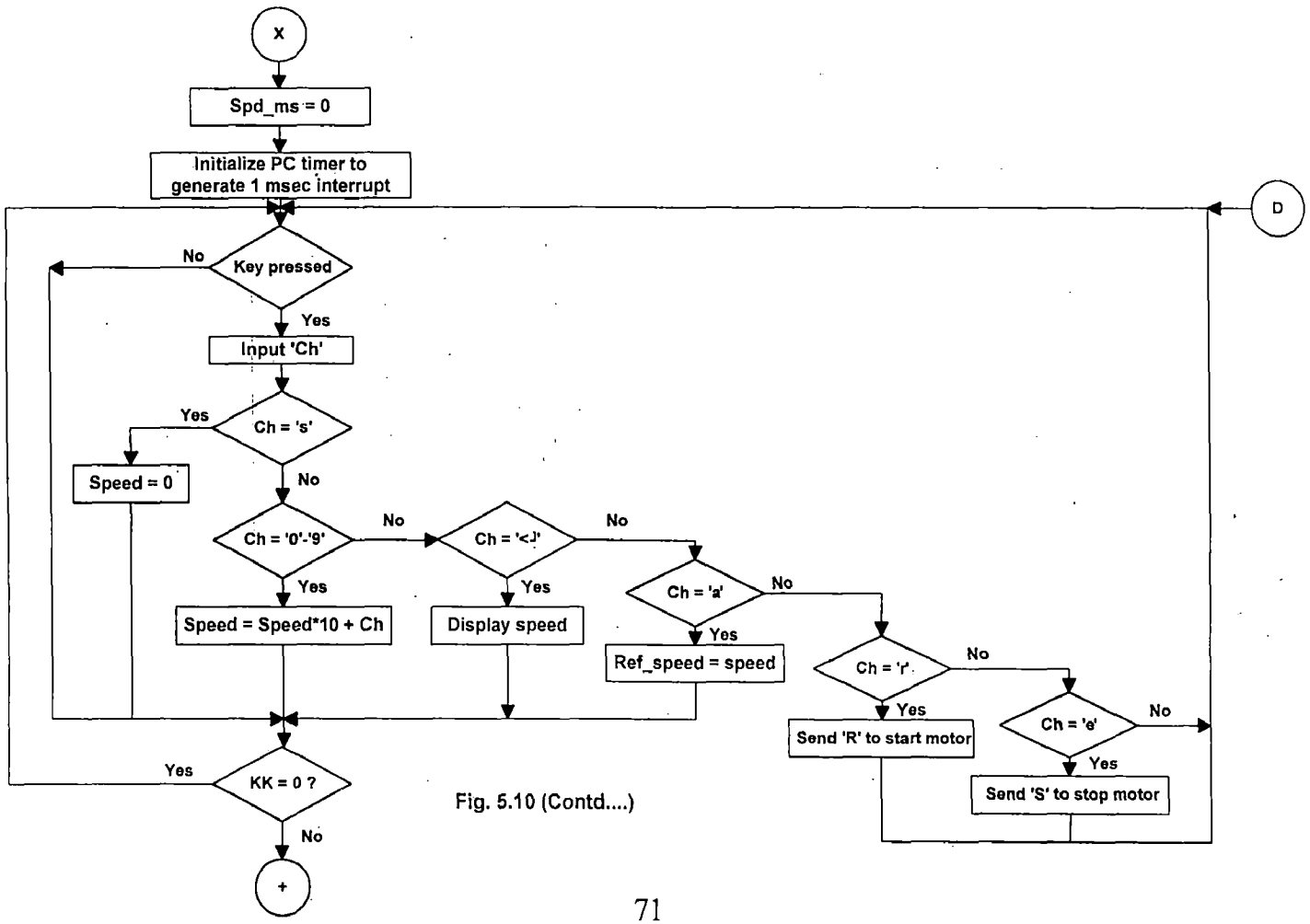


Fig. 5.10 (Contd....)

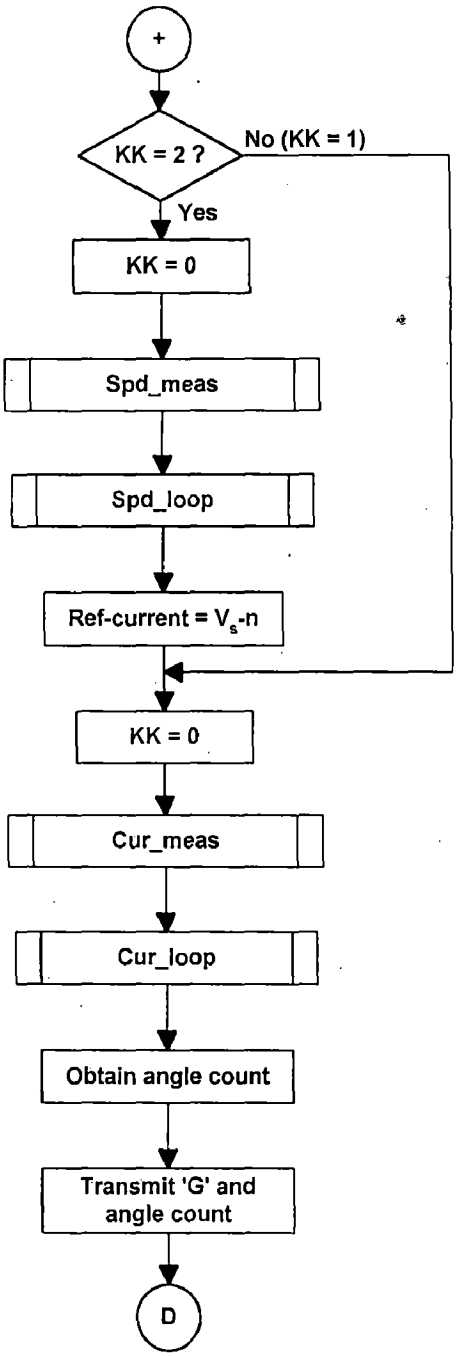


Fig. 5.10 (Contd.....)

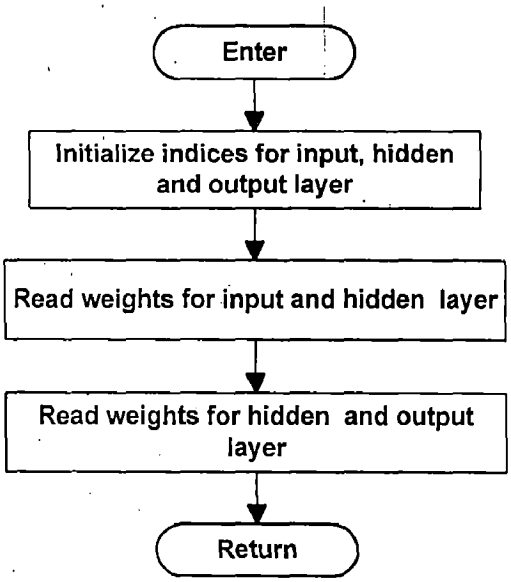


Fig. 5.11 : Program to read weights for neural controller

5.2.4 TABLE SUBROUTINE

This subroutine is called in main. In this routine α-count value at different control voltages has been calculated. This is the same as in the case of open loop configuration.

5.2.5 TIMER INTERRUPT SUBROUTINE

The flow chart for this subroutine is shown is 5.12. PC timer '0' is used to generate 1msec interrupt. These interrupts are used to identify the instant when to do speed error processing and current error processing. An index KK is initialized to one. Speed measurement are done at every 10 msec, while current measurement is done at every timer interrupt, i.e., at every msec interval. If spd-ms index is 0 then timer 0 is loaded with all F's and this index is incremented. When first interrupt of PC timer comes ADD-ON card timer 0 starts down counting and at 8^{th} interrupt from PC timer it stops down counting, and returns.

5.2.6 SPEED_MEAS SUBROUTINE

Lower and higher byte of speed obtained from timer interrupt subroutine is used to calculate actual speed in RPM. The scale of conversion is 1:1 i.e., 1 bit corresponds to 1 RPM. SPD_MS_O index is set to 0 when speed measurement get over. The flow chart for this is shown in Fig. 5.13.

5.2.7 SPD-LOOP

This loop is for processing of error between actual and reference speed. Patterns obtained from fuzzy logic is used for error processing. This loop is basically a neural controller to control speed. The neural network is trained off-line for the patterns derived from fuzzy logic in the form of error, change in error and control output. The output of output layer is the value of change in reference current (V_s-n). The new reference current is calculated and its value is limited between maximum and minimum limit of current. Fig. 5.14 shows flow chart for this loop.

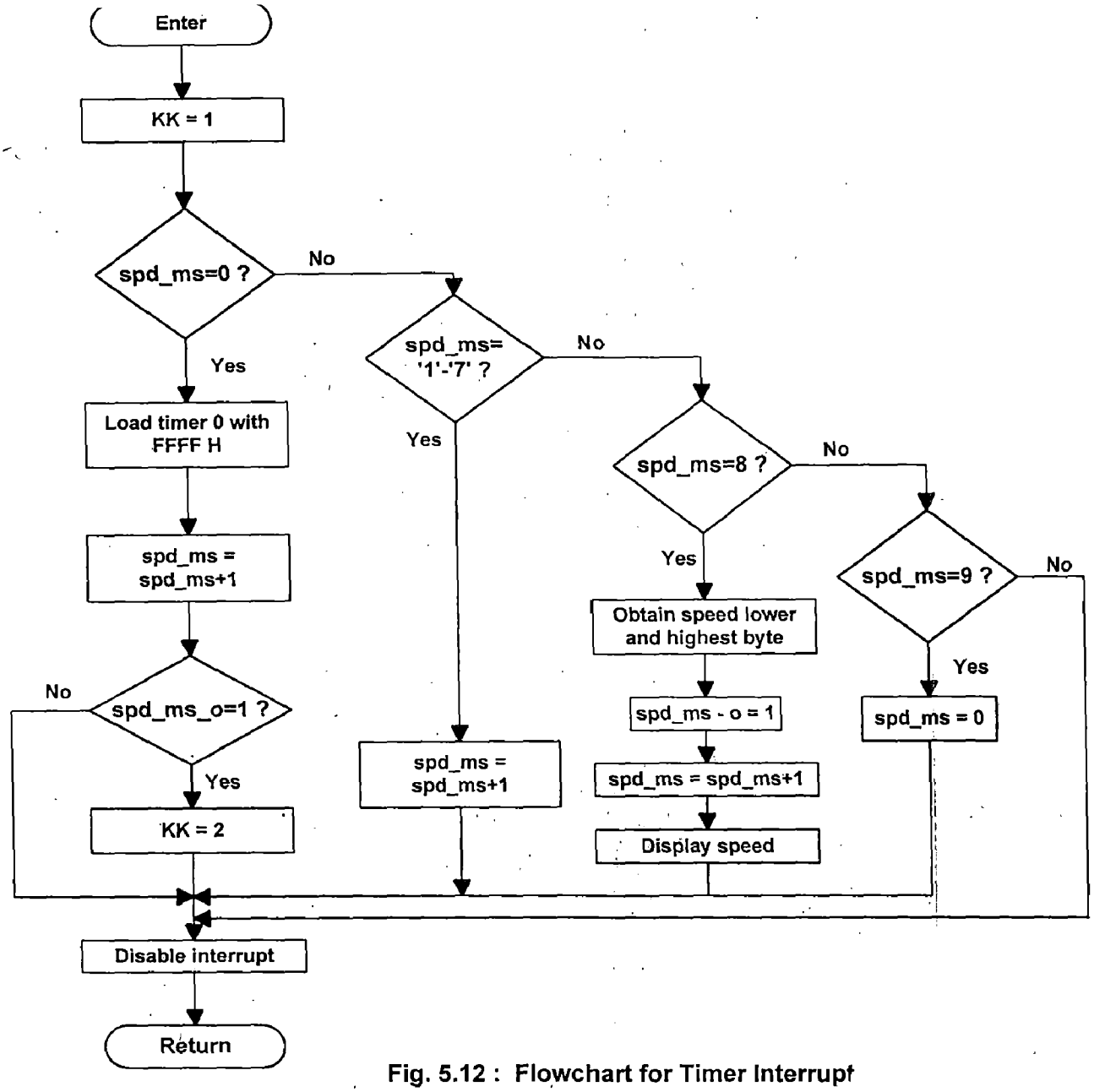


Fig. 5.12 : Flowchart for Timer Interrupt

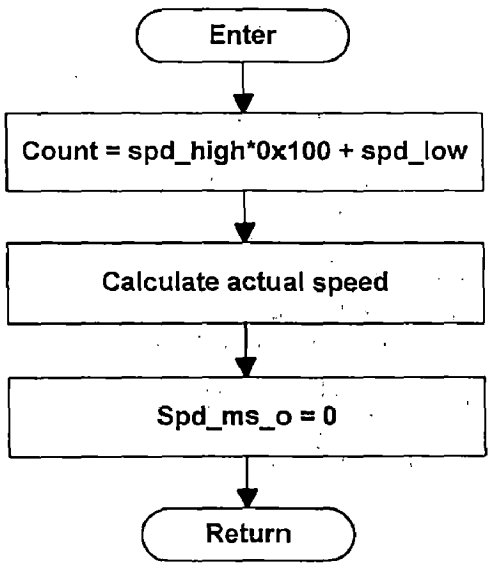


Fig. 5.13 : Flowchart for Speed Measurement

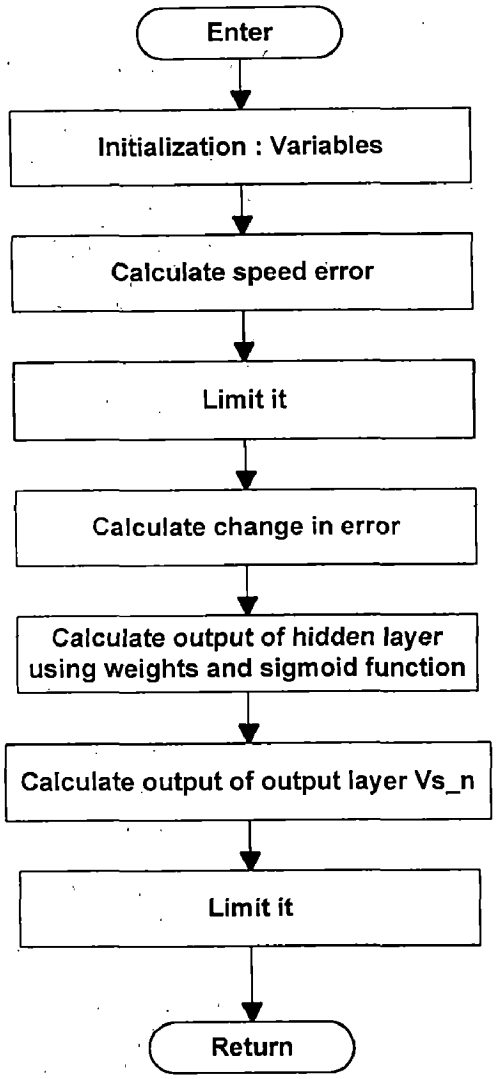


Fig. 5.14 : Flowchart for Speed Error Processing

5.2.8 CURRENT-MEAS SUBROUTINE

This subroutine is called in main program. To measure the actual current of system ADC of ACL-8112 HG is used. It is programmed for unipolar current measurement. Using this ADC analog voltage has been directly converted into digital value. Digitized value of current can be obtained.

5.2.9 CUR-LOOP

Reference current obtained from spd-loop and actual current from current_meas subroutine is compared in this subroutine. Control voltage is generated proportional to this error. The control voltage is limited between 0 to 2000. The

firing angle is obtained from look up table and transmitted serially to the microcontroller based system.

5.3 CONCLUSIONS

To control the speed of dc motor in open and closed loop, system software is developed in C language. The assembly language program for 8031 μC is developed. The various subroutines are discussed in detail.

RESULTS AND DISCUSSION

The performance of 3-ϕ fully controlled bridge converter fed DC drive is investigated in both open loop and closed loop mode of operation. This chapter deals with hte experimentation, performance investigation and discussion of results obtained. The performance of 3-ϕ fully controlled bridge converters is investigated with R load, R-L load and motor load and various waveforms are recorded under various operating conditions. After successful testing of converter it used to run the dc motor in open loop mode and closed loop mode and performance of the system is experimentally obtained.

6.1 TESTING OF CONVERTER

To investigate the performance of converter, the waveforms for zero crossing signal quantizers and firing pulses are recorded. Fig. 6.1 shows the waveforms of zero crossing with phase voltages. Zero crossing or base interrupts are coming at the interval of every 60°. These zero crossing interrupts are given to INT0 of 8031 μC.

In Fig. 6.2 waveforms of quantizers alongwith base interrupt is shown. These waveforms are used to generate firing pulses for thyristors. Six monoshots are triggered at rising and falling edge of quantizers and pulses are combined to generate zero crossing interrupt. The firing pulses of thyristors Th_1 and Th_2 are shown in Fig. 6.3. For proper commutation of thyristors firing pulses are extended for 120° period. Figure 6.3 shows the overlap of 60° of two firing pulses.

Figs. 6.4 to 6.11 show the waveforms of output current, output voltage, input current at various firing angles. These waveforms are obtained with R load. Output current follows output voltage at different firing angles. Waveforms of input

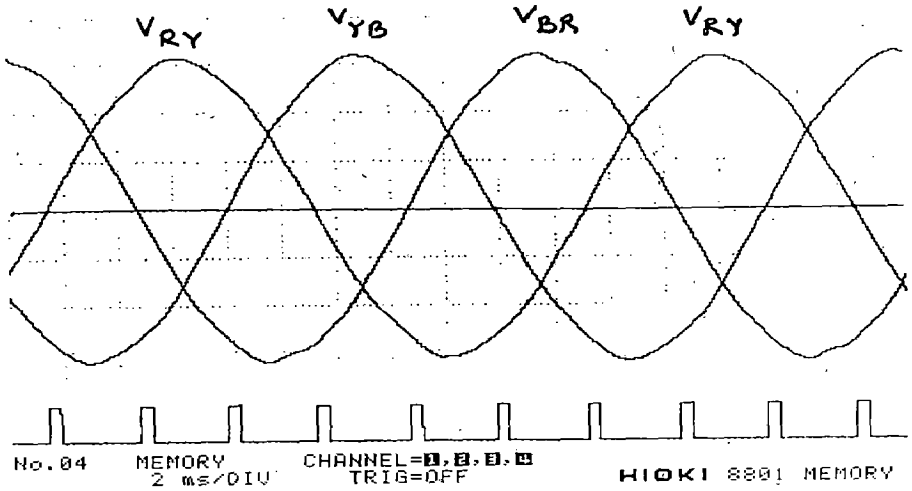


Fig. 6.1 Waveform of zero crossing and phase voltages

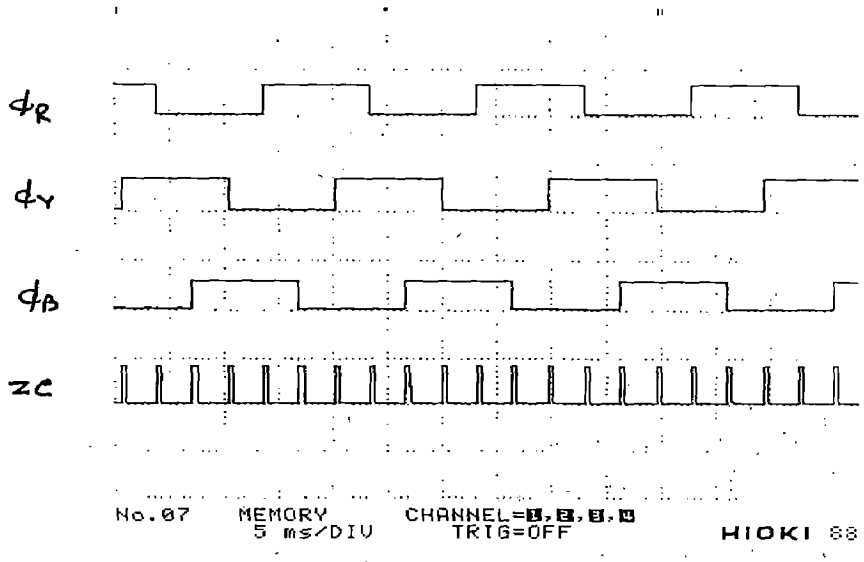


Fig. 6.2 Waveform of quantizer and zero crossing

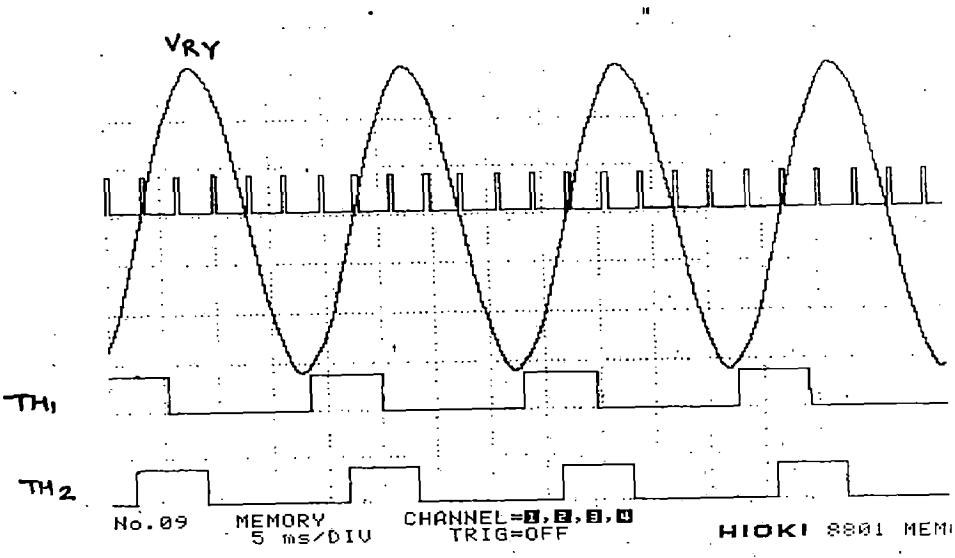


Fig. 6.3 Waveform of firing pulses for Th1 and Th2

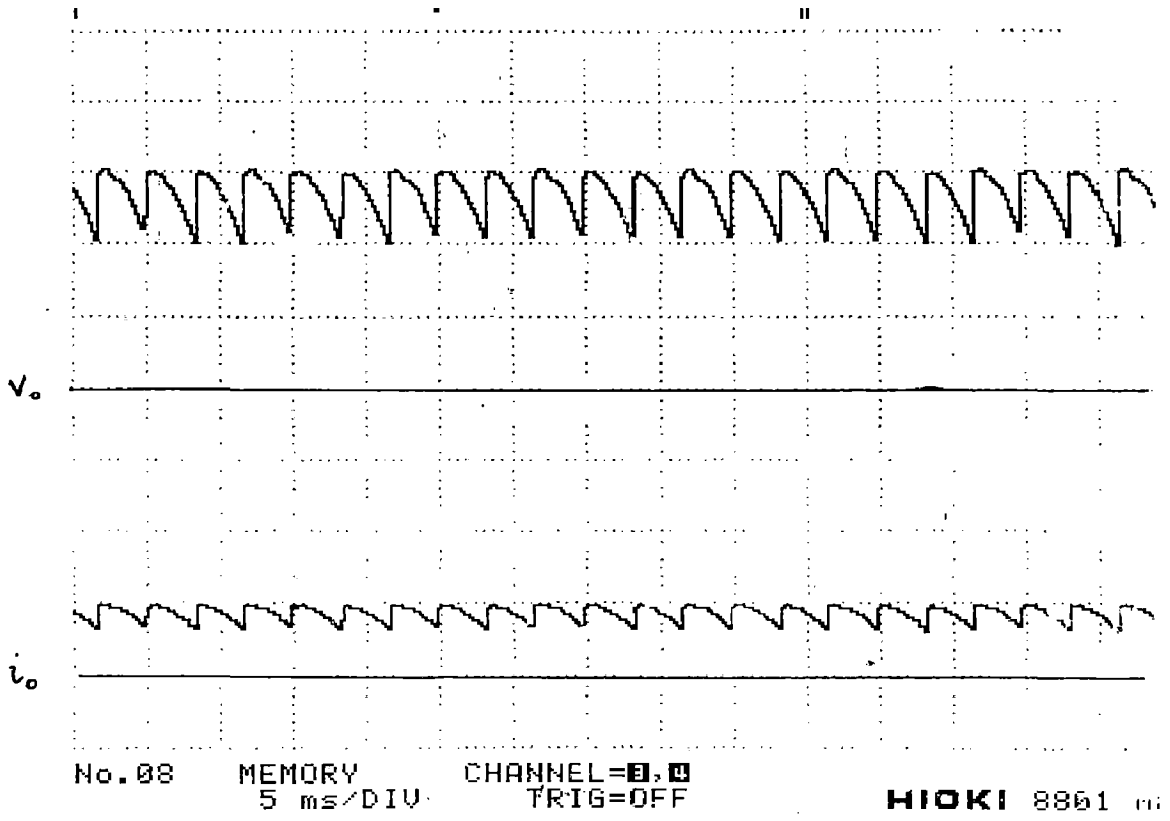


Fig. 6.4 Waveform of output voltage and output current at firing angle 9.53°

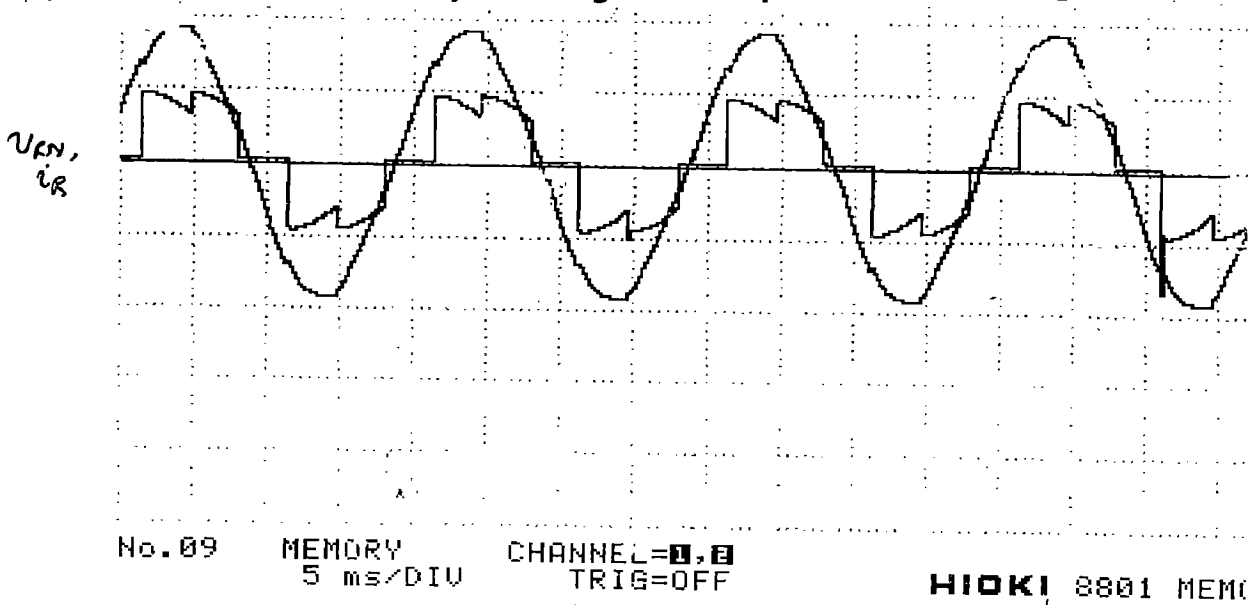


Fig. 6.5 Waveform of input current at firing angle 9.53°

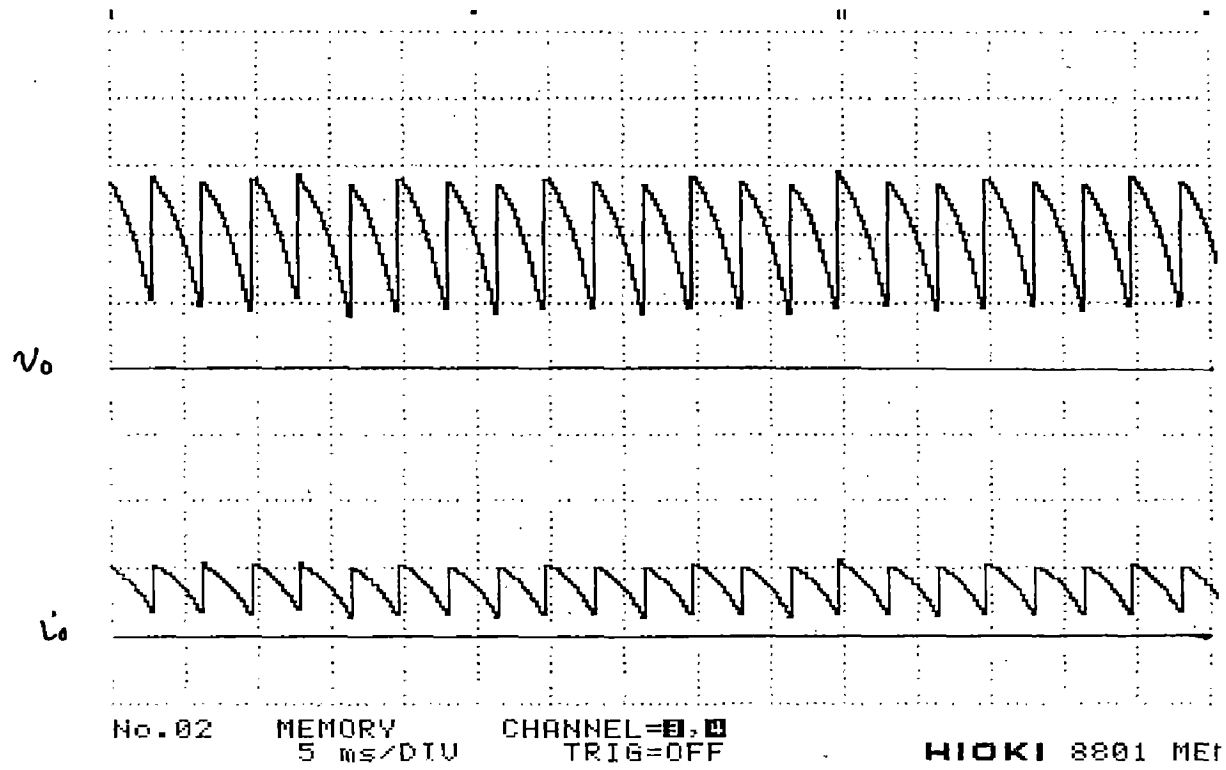


Fig. 6. 6 Waveform of output voltage and output current at firing angle 36.8°

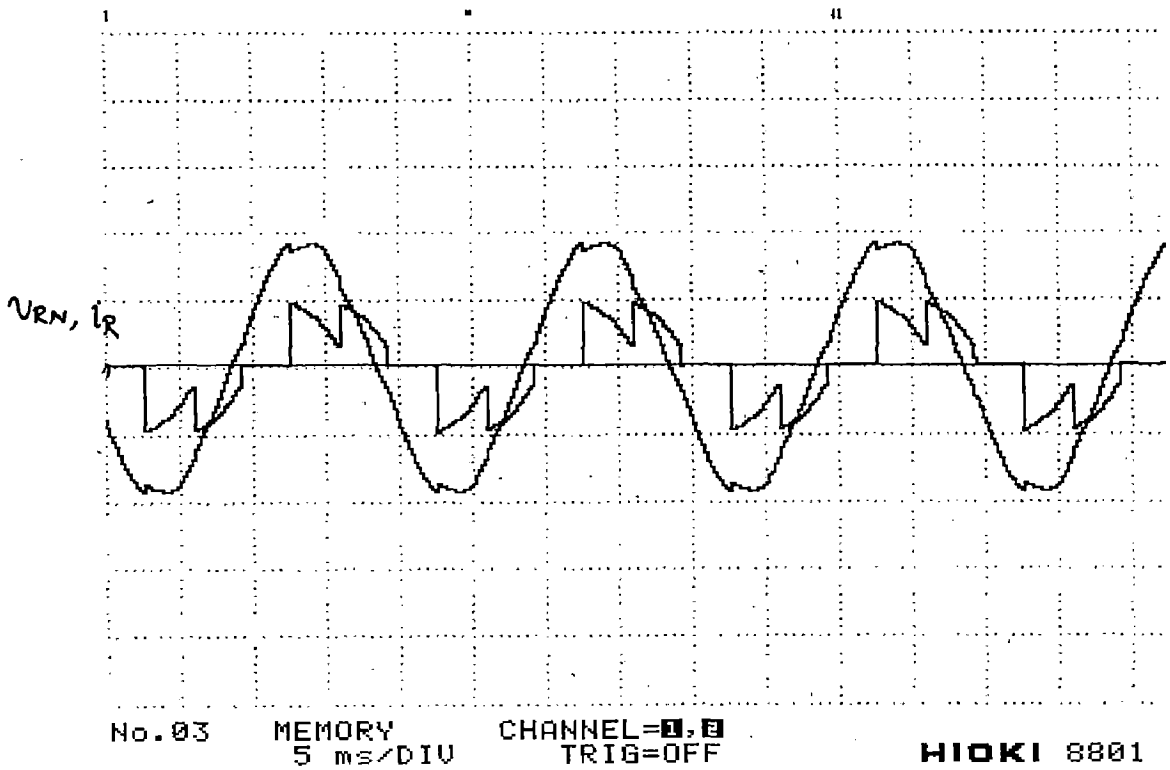


Fig. 6.7 Waveform of input current at firing angle 36.8°

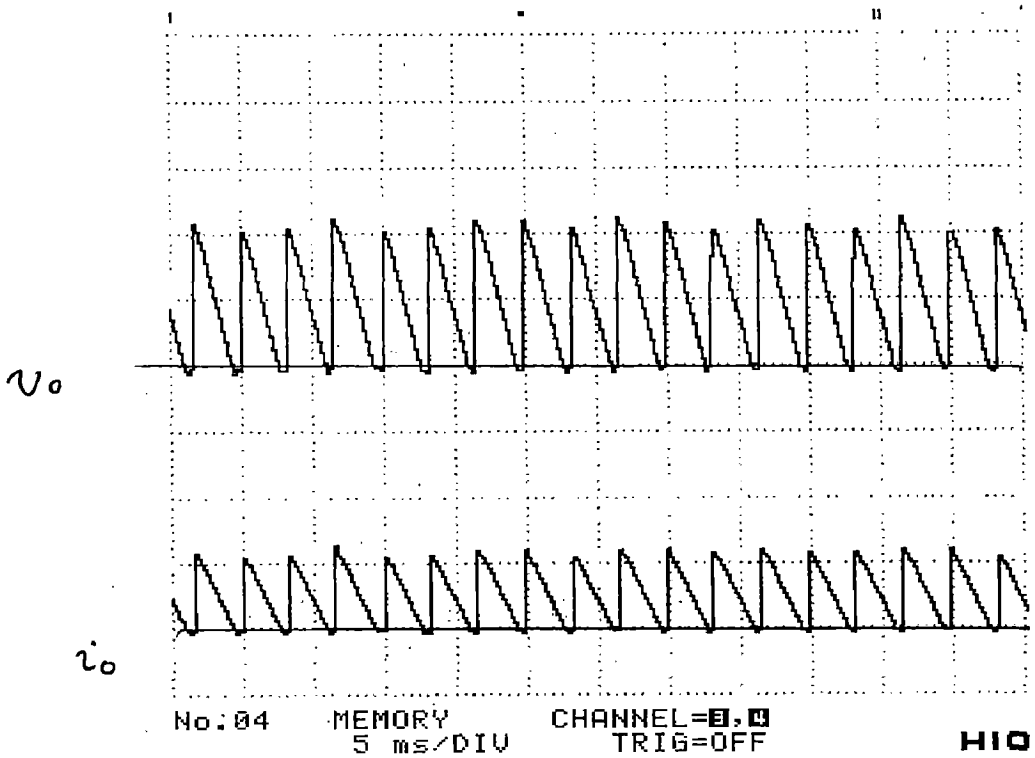


Fig. 6.8 Waveform of output voltage and output current at firing angle 60.5°

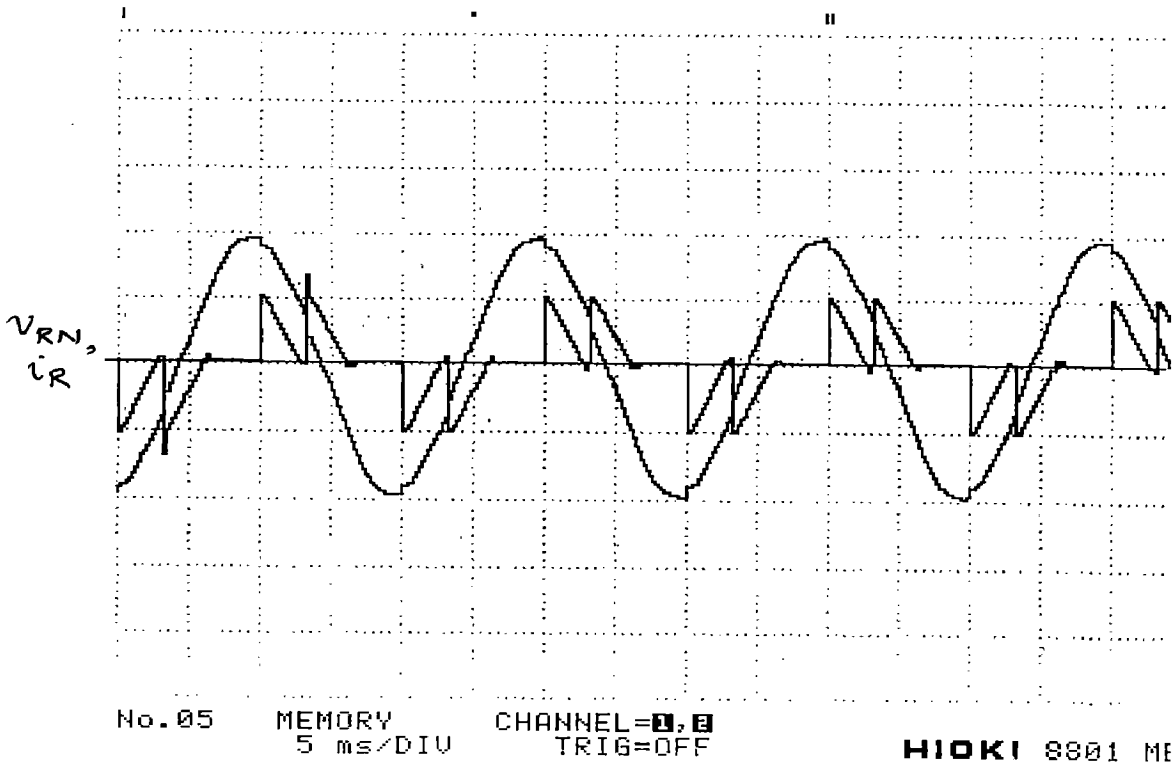


Fig. 6.9 Waveform of input current at firing angle 60.5°

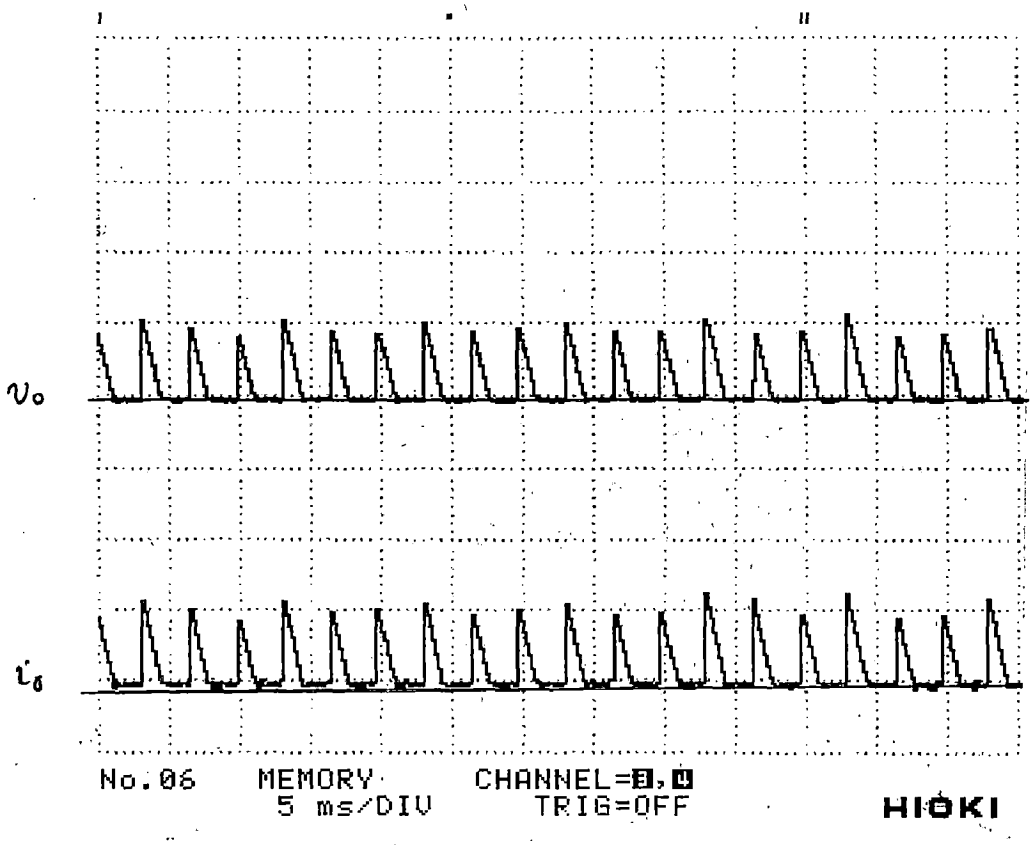


Fig. 6.10 Waveform of output voltage and output current at firing angle 90°

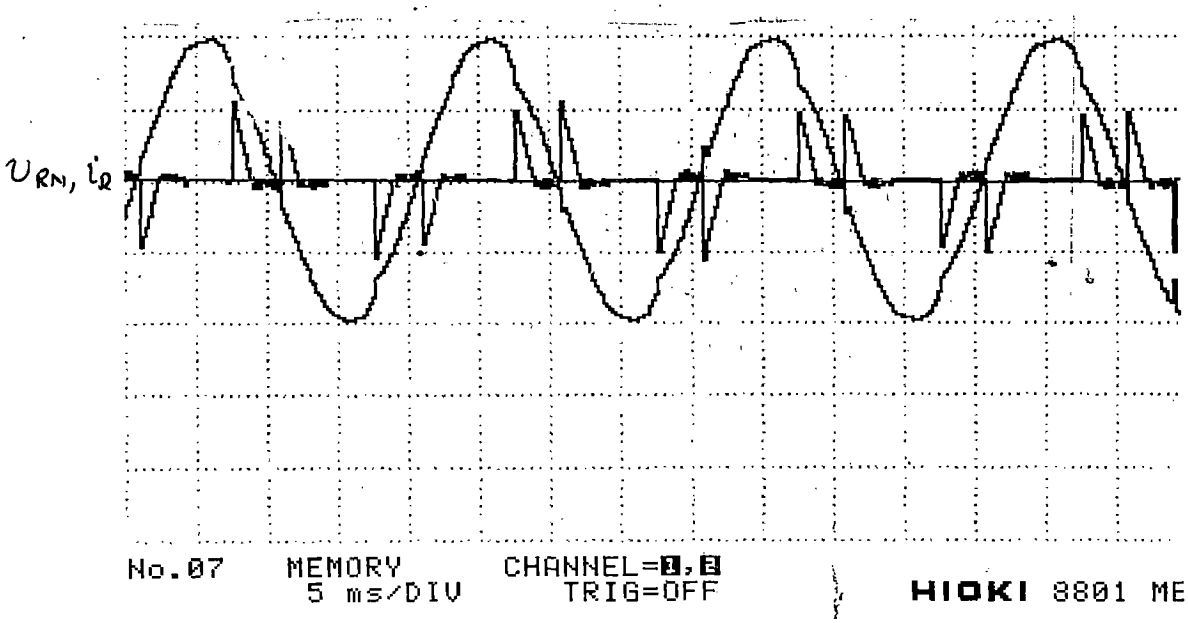


Fig. 6.11 Waveform of input current at firing angle 90°

current for one phase shown in these figures also confirm the basic characteristics of converter at resistive load.

Converter is further tested for R-L load to decide the inductance for continuous conduction and ripple free current. Waveforms of output current and output voltage are recorded at different firing angle keeping high value of inductance. The inductance value selected is 400 mH.

Fig. 6.12 is recorded with less inductance in circuit. Due to less inductance the current is not continuous, which can be seen from the Fig. 6.12.

Fig. 6.13 is obtained with high inductance at same firing. In this case current is continuous unlike to previous case. This shows the effect of inductance on performance of converter. By increasing the firing angle output of converter can be decreased.

Figs. 6.17, 6.18, 6.19 . show the waveforms of output current and output current at various firing angle with motor load. In these waveforms peaks are coming in output voltage due to load inductance of motor. Current is almost continuous with different values of firing angle less than 90° as shown in figures. All these waveforms confirm the satisfactory operation of the converter. It is clear that output of converter increases as firing angle decreased. The software for open loop has been used in testing of converter.

6.2 OPEN LOOP PERFORMANCE OF DC DRIVE

The basic diagram of the system in open loop mode is shown in Fig. 6.21.

The performance of dc drive in open loop mode is investigated by conducting load test on it. The load test is conducted at firing angles 36.84° and 66.42°. The various voltage, currents and power are measured under loaded condition. The various performance curves are shown in Fig. 6.22 to 6.25. The output power vs torque curves are shown in Fig. 6.22. From this graph it can be observed that output power is varying in linear manner with load torque as the speed variation is less. For higher values of firing angle, output power is less due to reduced speed.

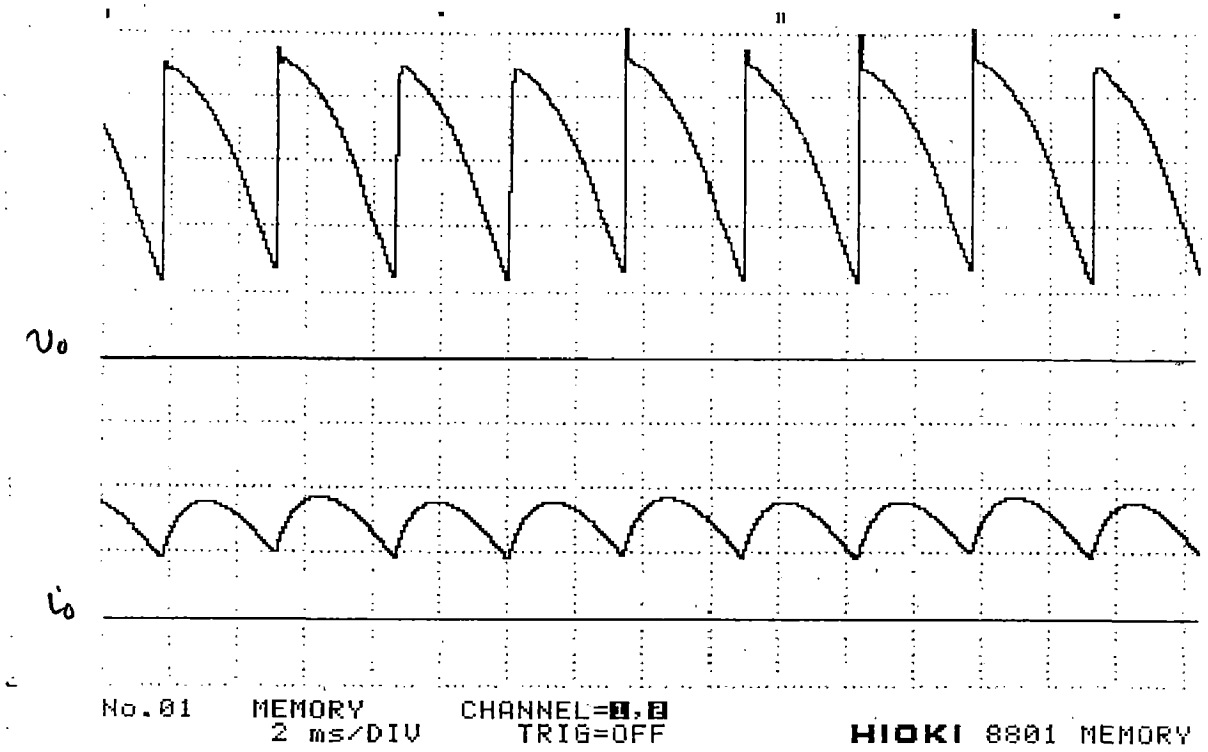


Fig. 6.12 **Waveforms of output current and output voltage at α = 36.84° (R-L load less inductance**

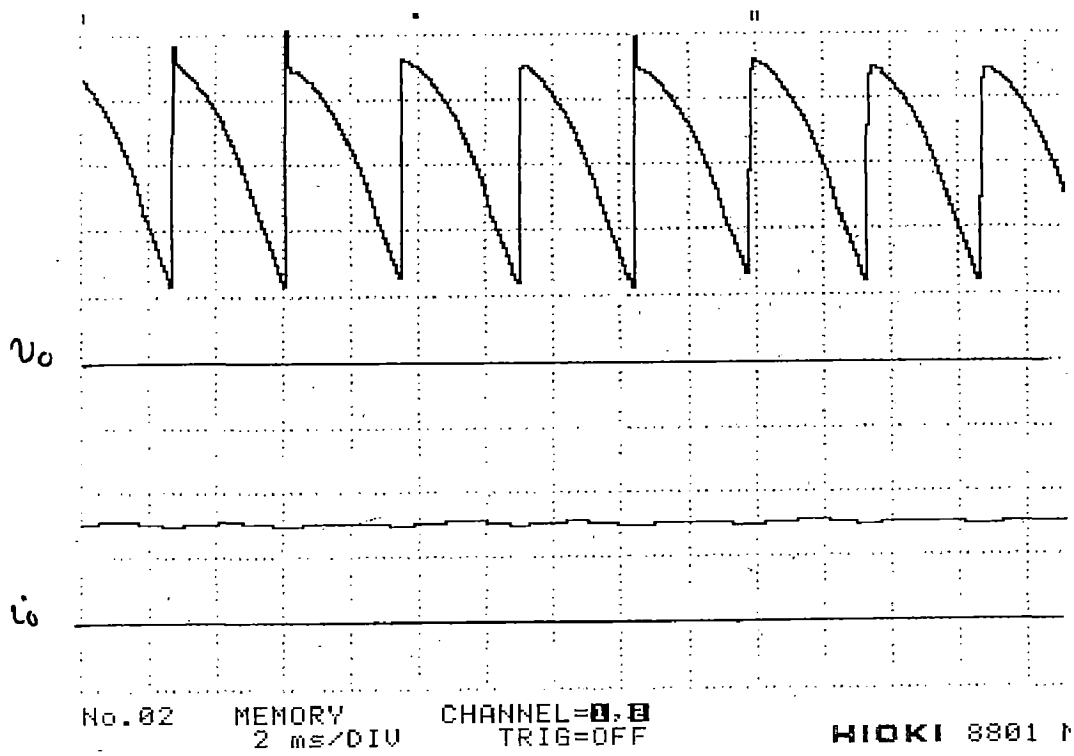


Fig. 6.13 **Waveform of output current and output voltage at α = 36.84° (R-L load high inductance**

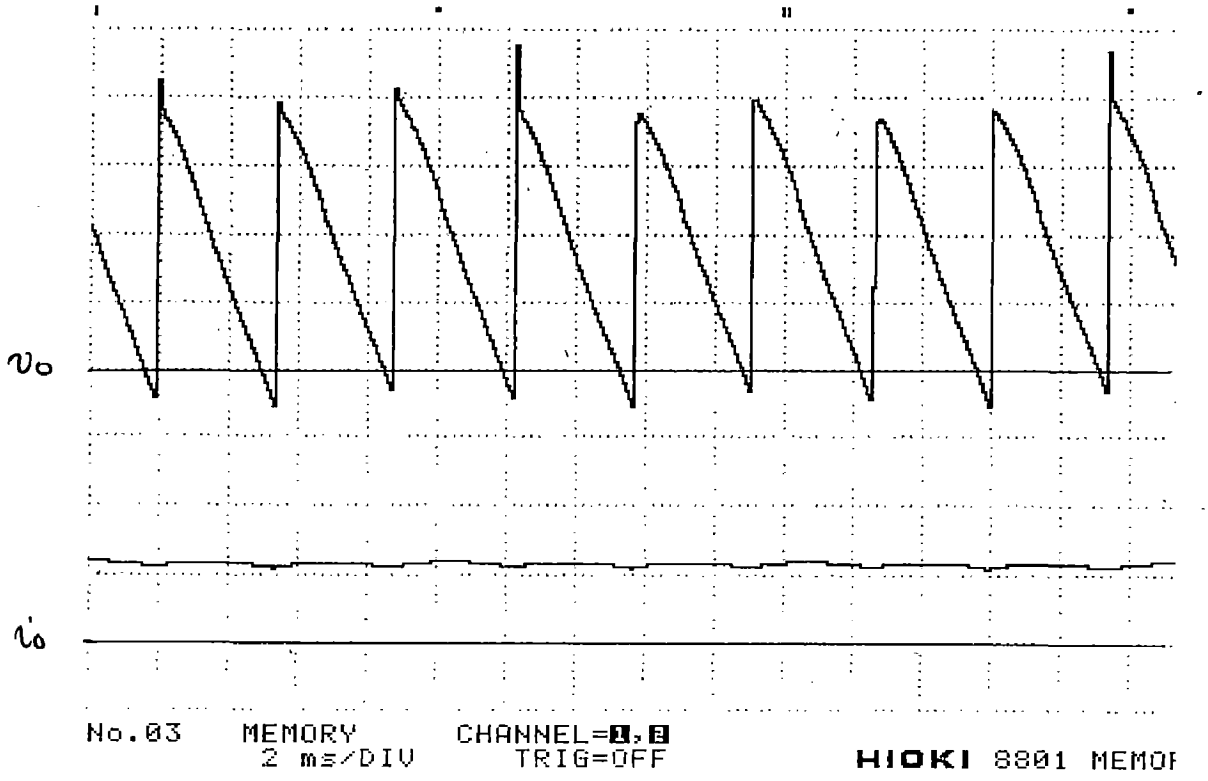


Fig. 6.14 **Waveform of output current and output voltage at α = 59.32° (R-L load high inductance**

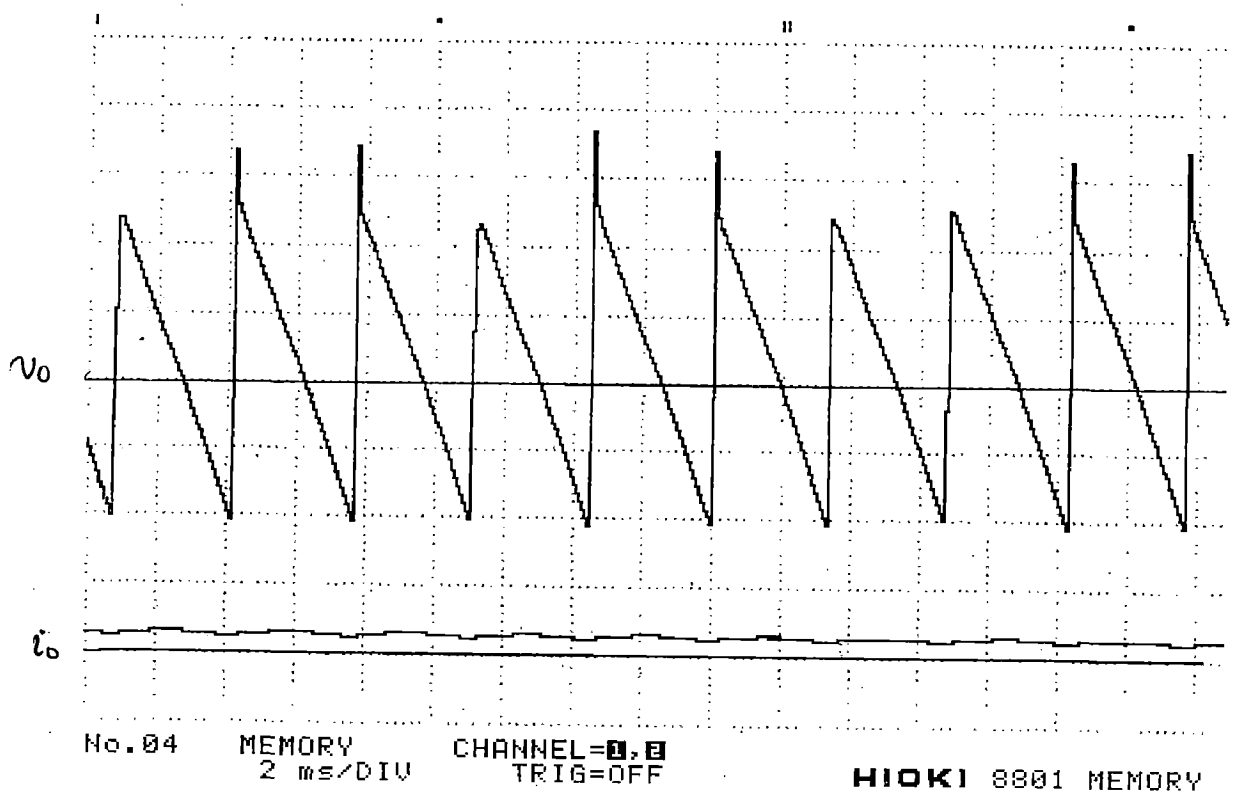


Fig. 6.15 **Waveform of output current and output voltage at α = 78.4° (R-L load high inductance**

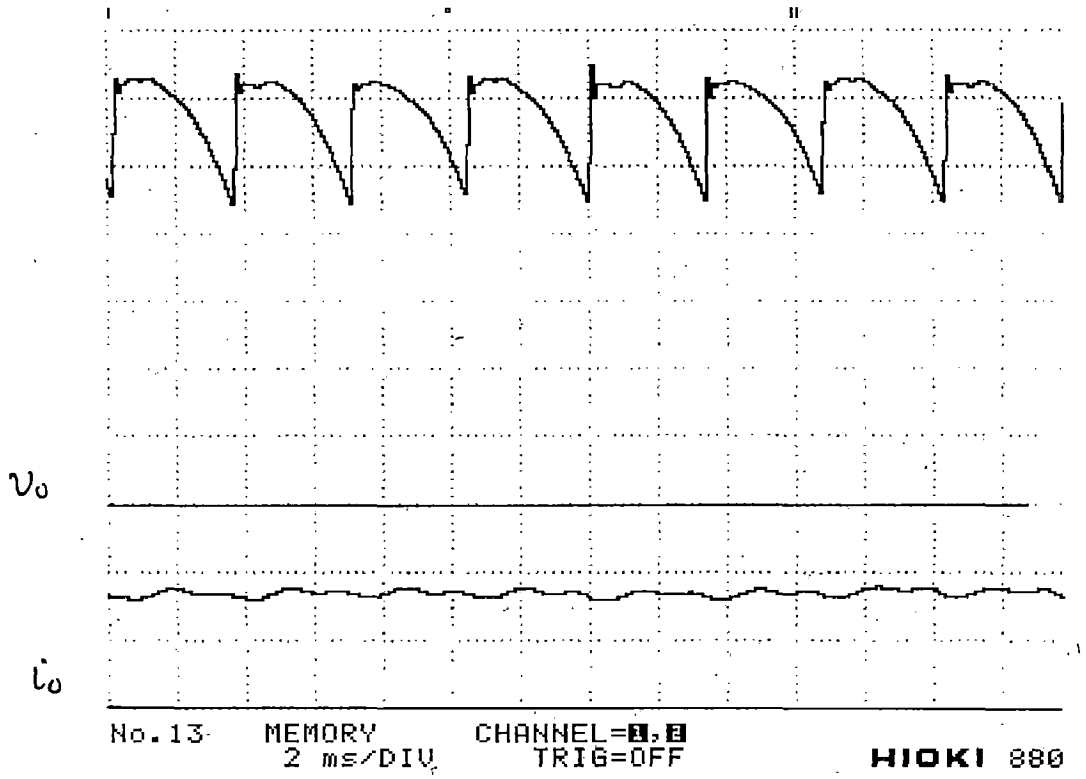


Fig. 6.16 **Waveform of output current and output voltage at α = 9.594° (motor load high inductance**

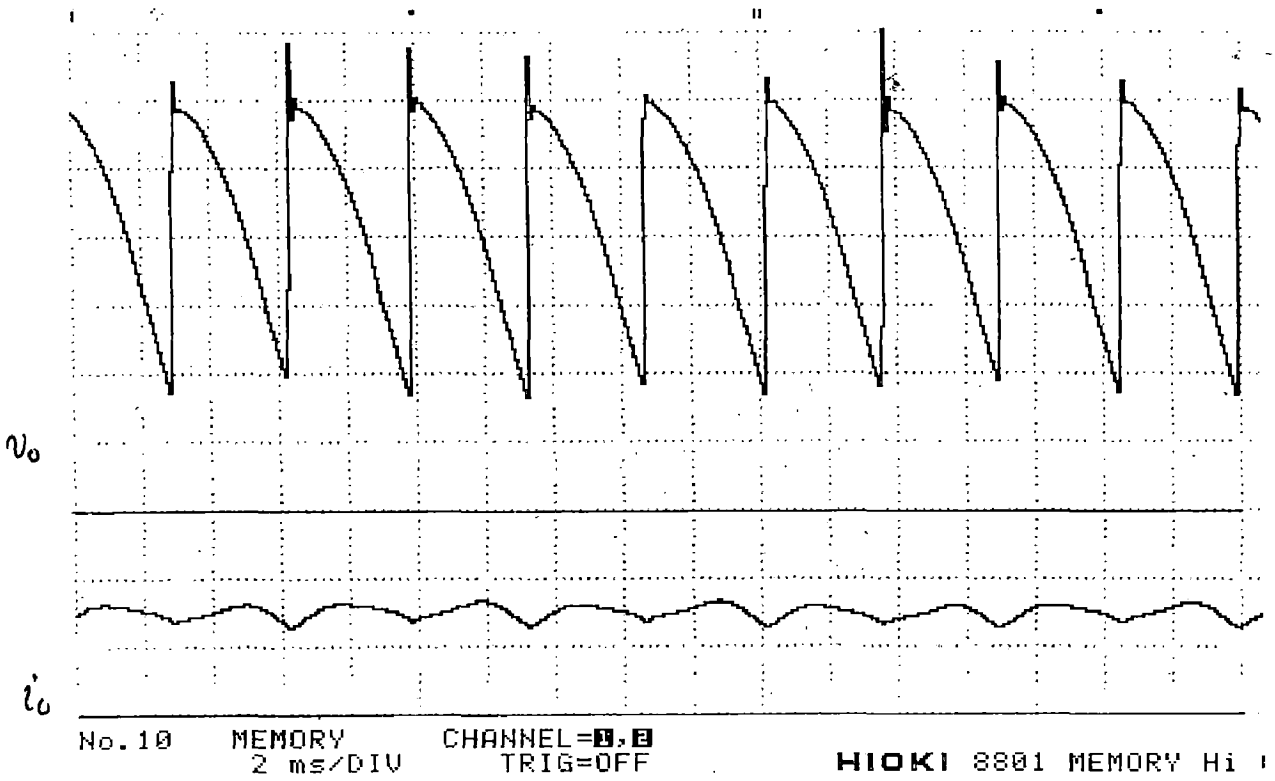


Fig. 6.17 **Waveform of output current and output voltage at α = 36.84° (motor load high inductance**

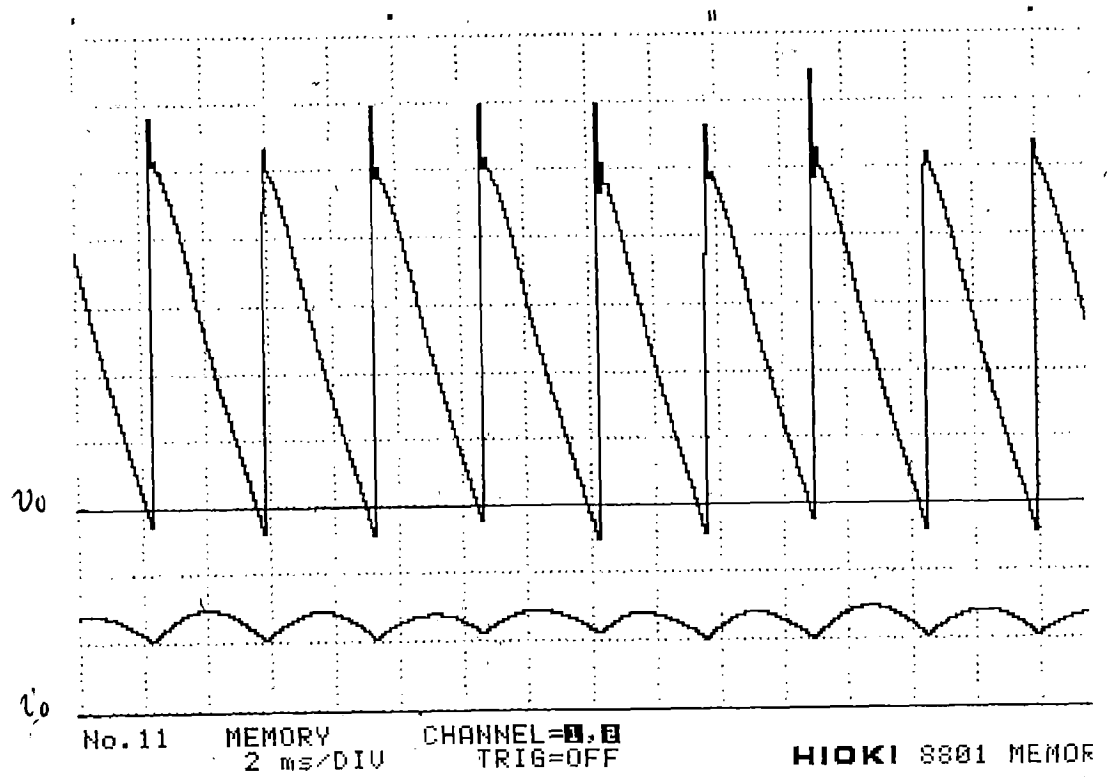


Fig. 6.18 **Waveform of output current and output voltage at α = 59.652° (motor load high inductance**

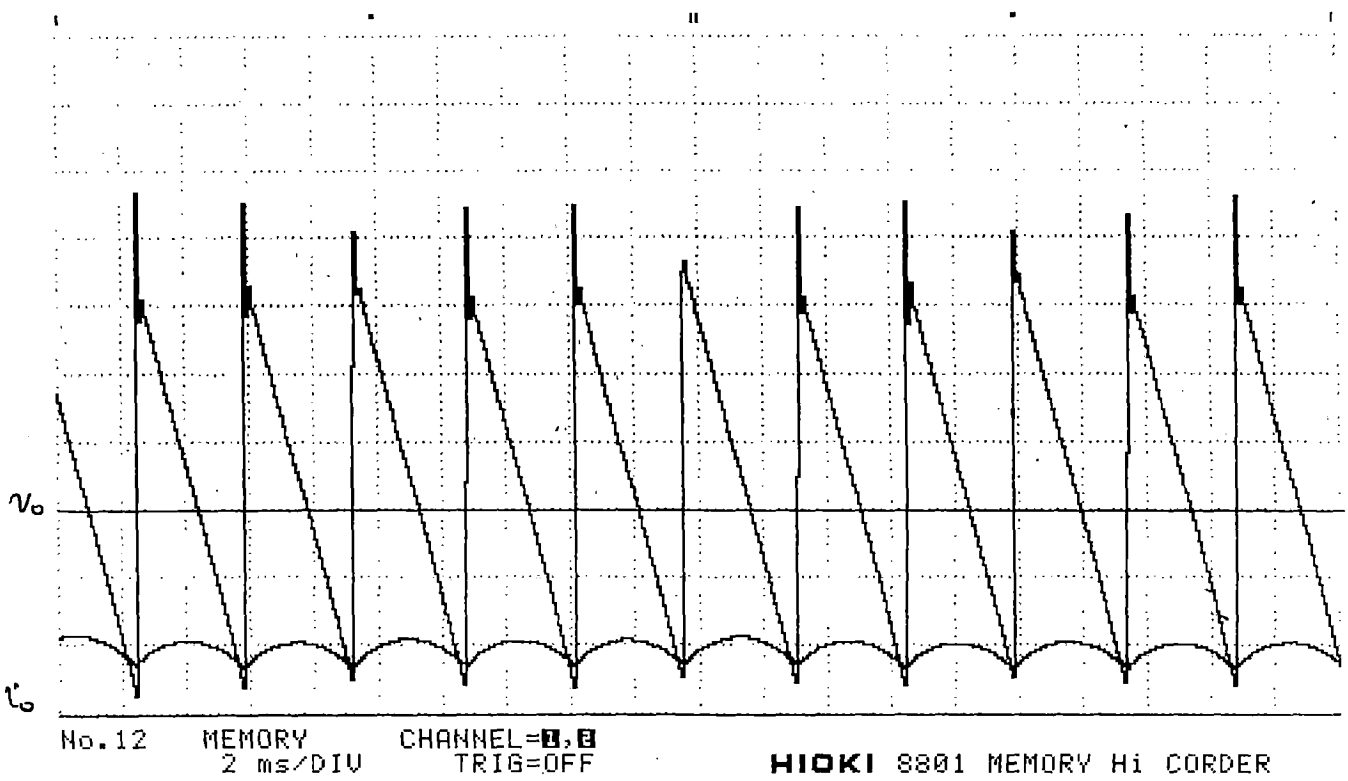


Fig. 6.19 **Waveform of output current and output voltage at α = 78.46° (motor load high inductance**

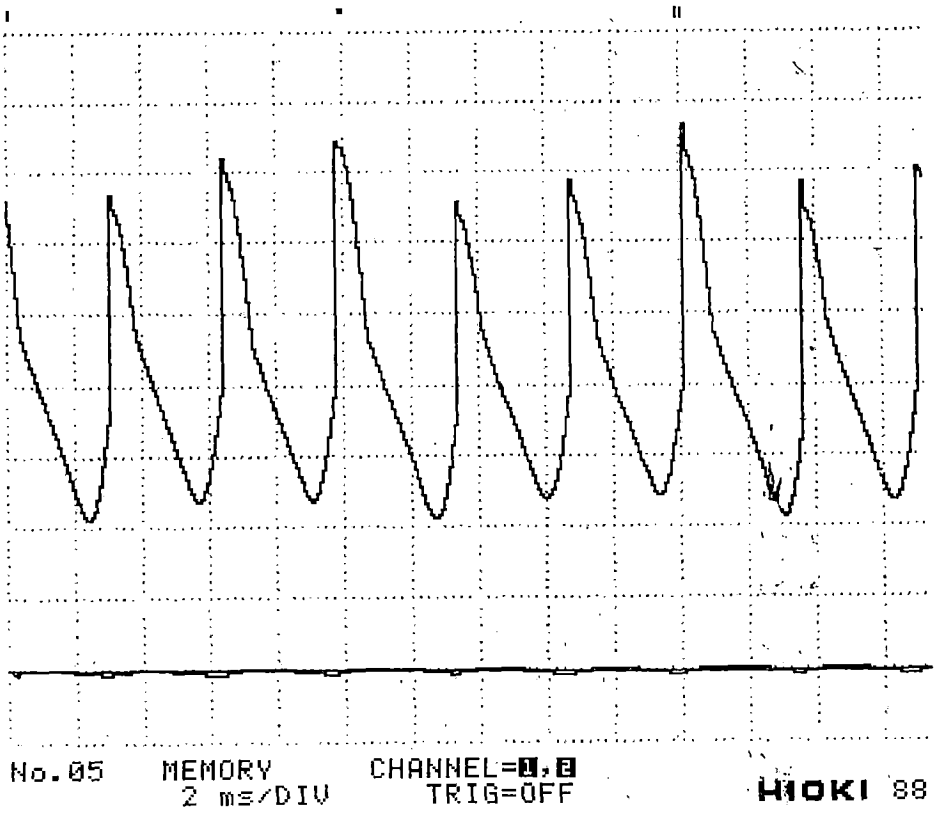


Fig. 6.20 **Waveform of output current and output voltage at $\alpha = 90^{\circ}$ (R-L load high inductance**

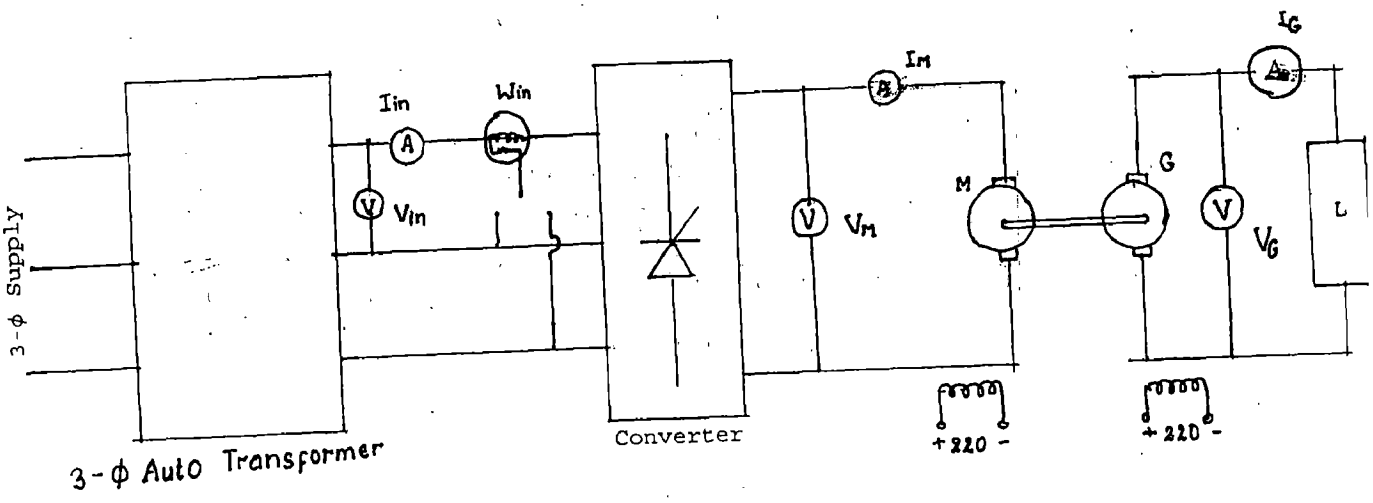


Fig. 6.21 **Block diagram of open loop system**

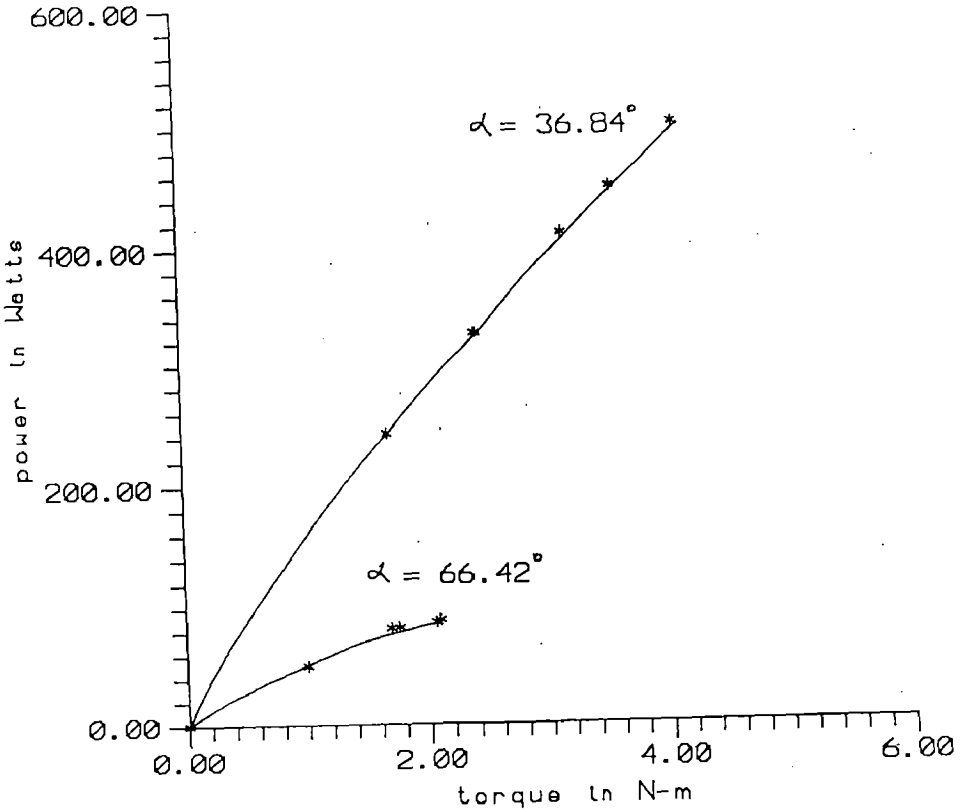


Fig. 6.22 Output power vs load torque

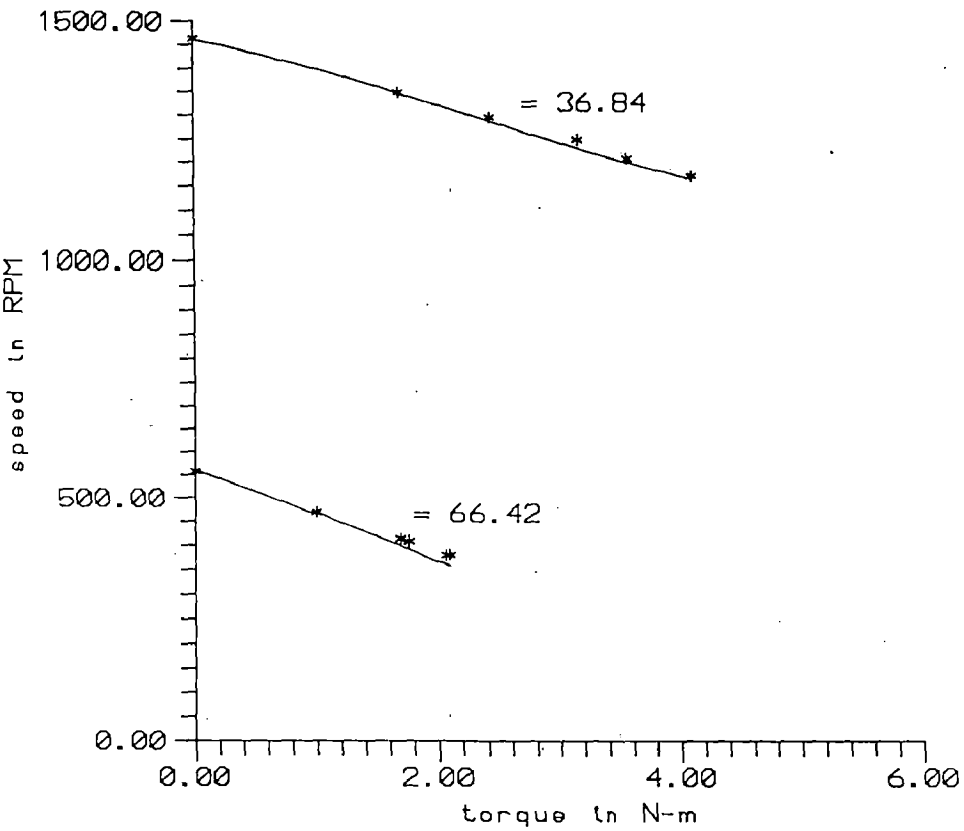


Fig. 6.23 Speed vs load torque at α = 36.84° and α = 66.42° in open loop mode

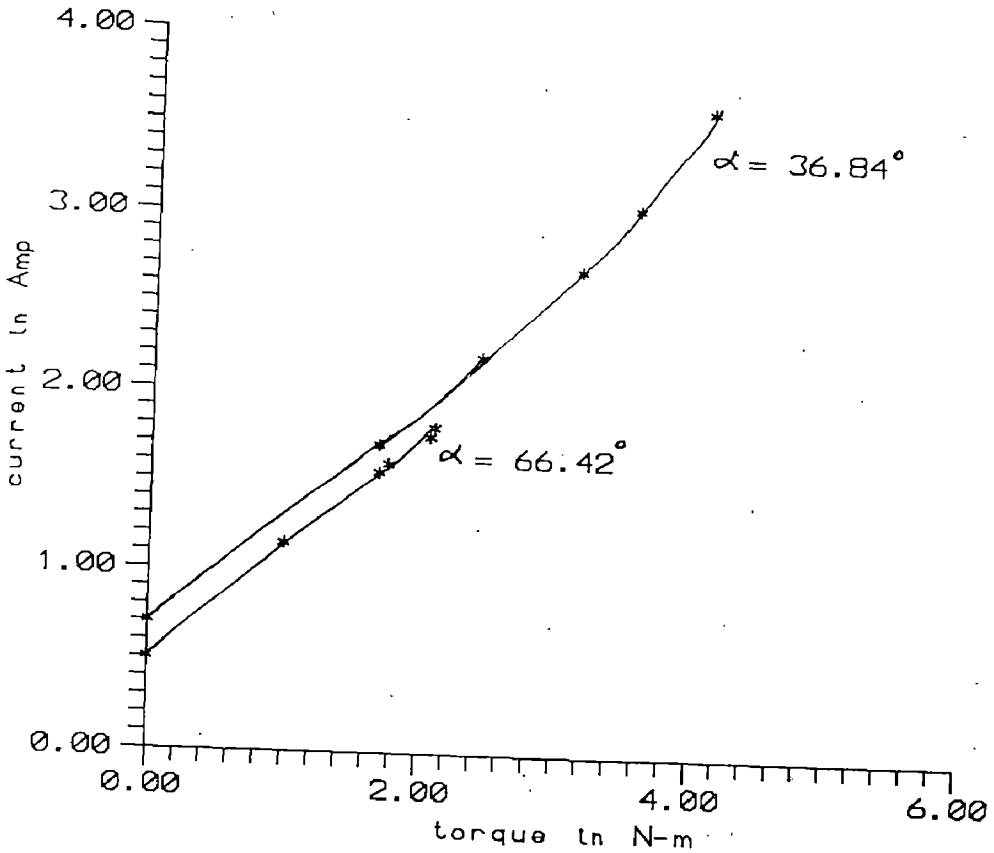


Fig. 6.24 **Input current vs load torque at α = 36.84° and α = 66.42° in open loop mode**

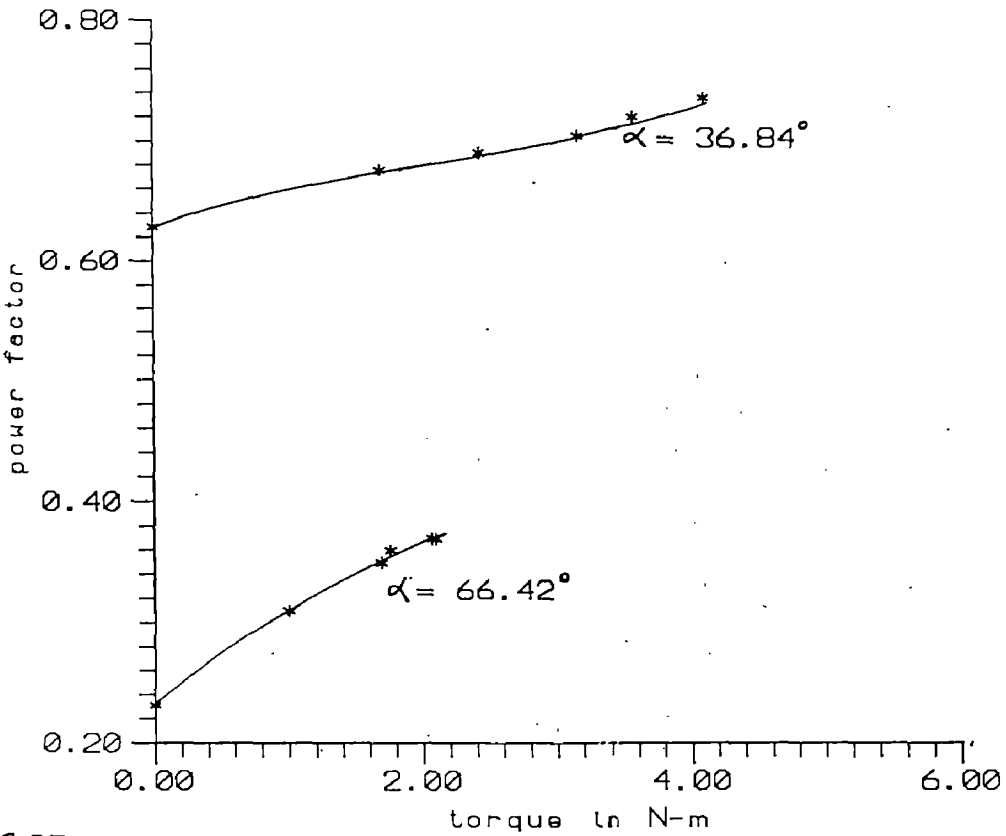


Fig. 6.25 **Power factor vs load torque at α = 36.84° and α = 66.42° in open loop mode**

Fig. 6.23 shows the drop in speed with increasing value of torque. The graph confirms the drooping characteristics of torque speed curve of separately excited dc motor.

In Fig. 6.24, graph between torque and input current to the converter has been plotted. As the motor is loaded, the input current increases.

The power factor vs torque curves for different firing angles are shown in Fig. 6.25. As the motor is loaded, the input power factor increases.

6.3 CLOSED LOOP PERFORMANCE OF DC DRIVE

The basic block diagram for closed loop operation is shown in Fig. 6.26. To run drive in closed loop, current and speed feed back are required for this purpose accurate current and speed measurement at much faster rate is required. The necessary hardware is designed and fabricated for speed and current measurement as discussed in chapter 4. ACL-8112 card is used for measuring speed and current.

Software in C language is developed and tested. With the help of software actual speed and actual current is measured. To check the design of speed controller, trained neural network is implemented and response of the dc motor is recorded for step change in reference speed.

Fig. 6.27 shows the response of the drive in closed loop mode. The reference speed is taken as 1500 rpm in this case. These waveforms are recorded assuming that generator voltage is proportional to speed of the motor. The settling time of motor is approximately 1.4 sec.

Actually the speed does not settle perfectly at any instant but motor runs in a band of $\pm$ 50 rpm which shows the oscillations in the output voltage.

Fig. 6.28 has been recorded for step reference speed of 1000 rpm. The settling time is approximately 0.9 sec.

The closed loop performance of the system is investigated by measuring the actual speed of the dc motor for different reference speed setting under no load condition. Fig. 6.29 shows the actual speed vs reference speed which is almost linear. It confirms the closed loop operation. However, it was found that in close

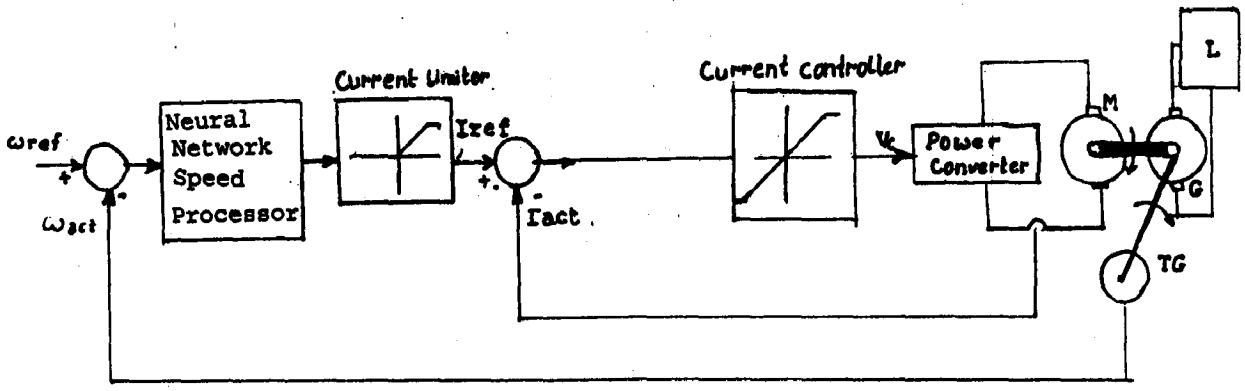


Fig. 6.26 Block diagram of closed loop operation

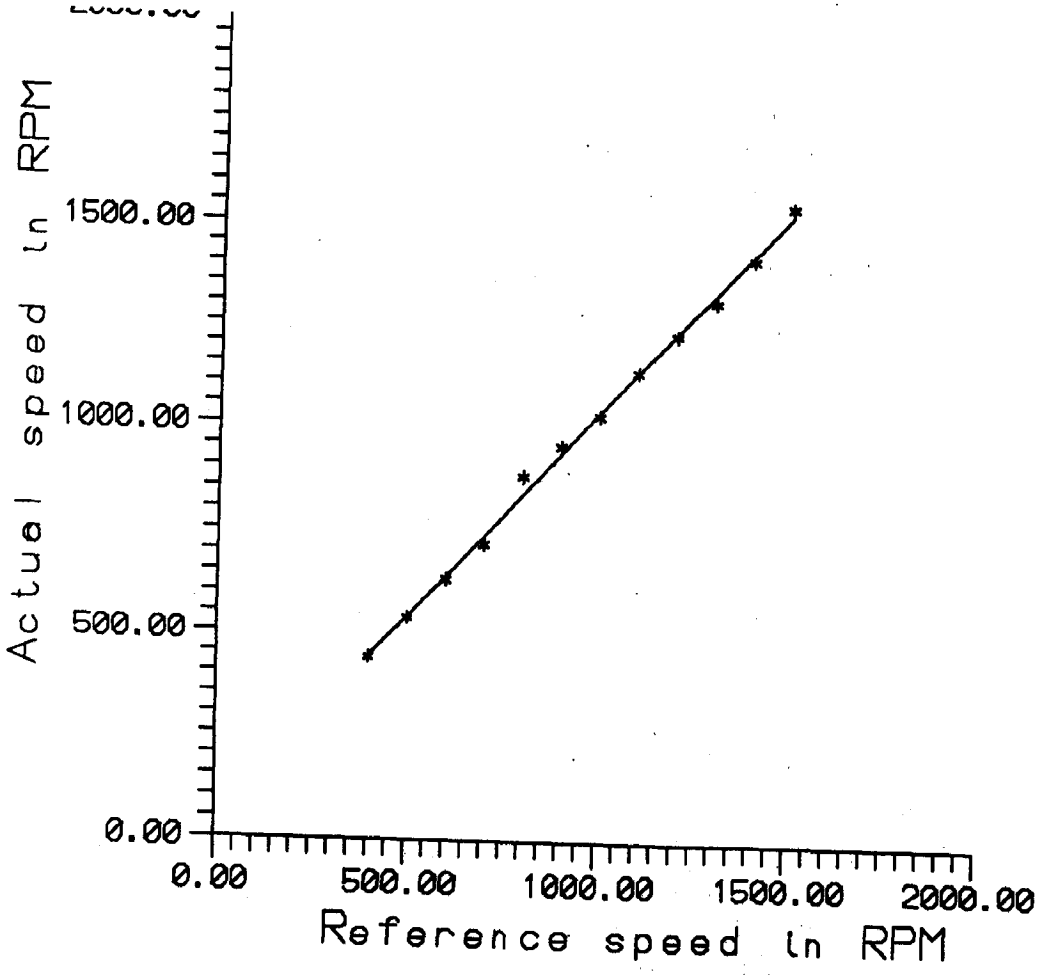


Fig. 6.29 Actual speed Vs Reference speed

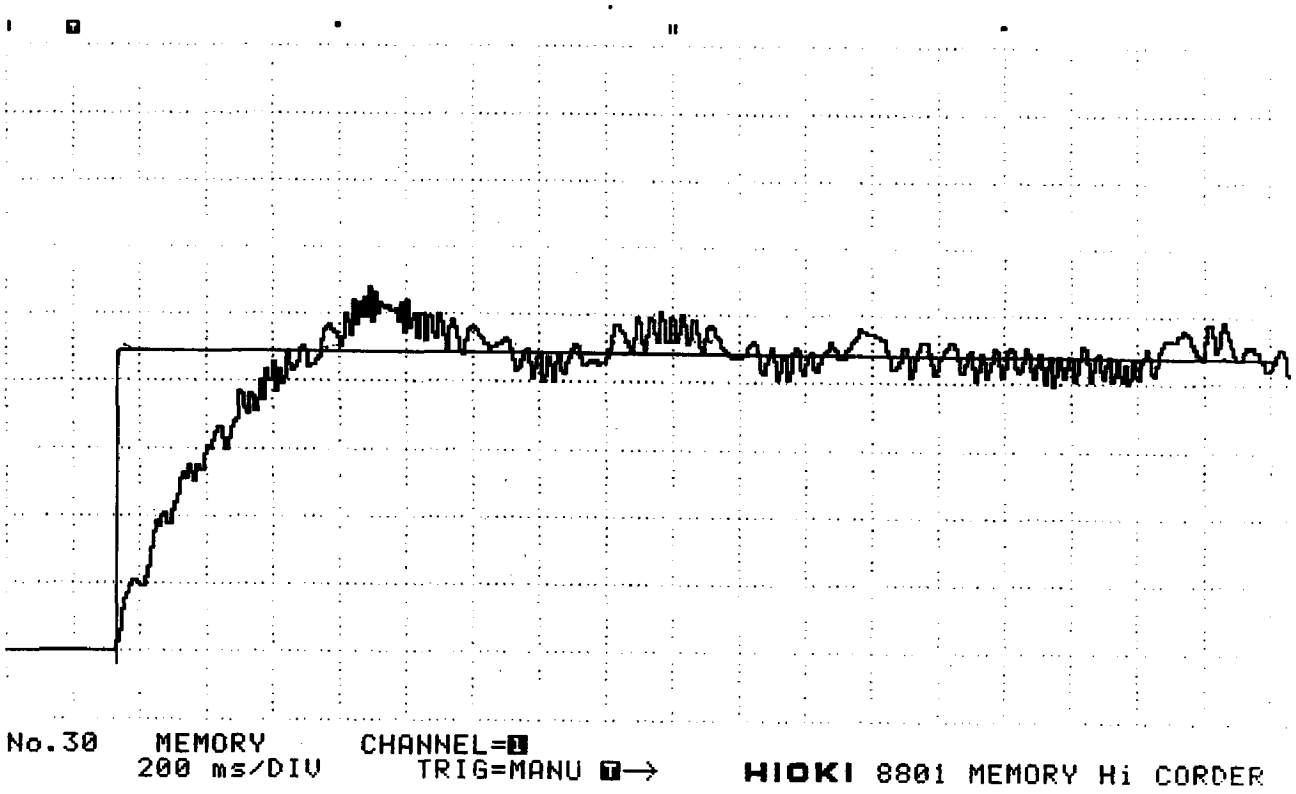


Fig. 6.27 Variation of motor speed in closed loop mode at 1500 rpm reference speed

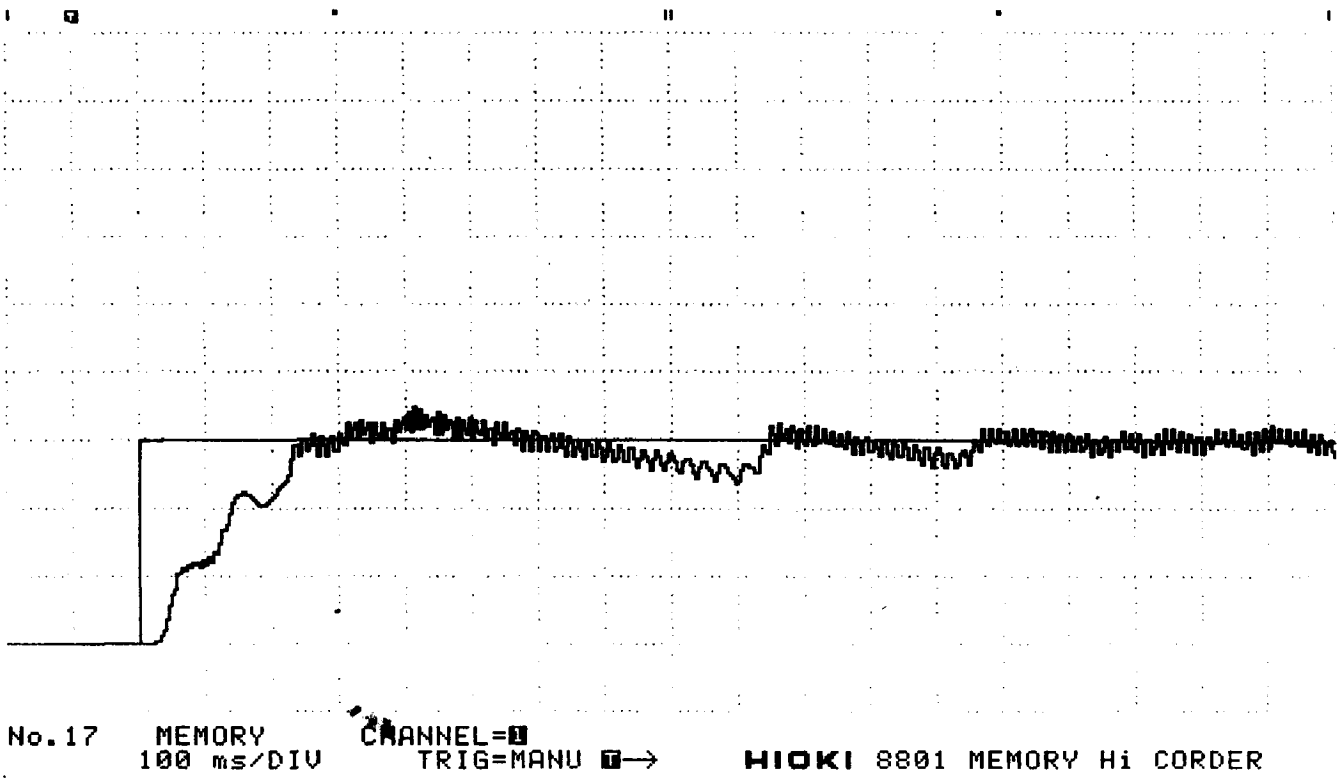


Fig. 6.28 Variation of motor speed in closed loop mode at 1000 rpm

loop, the speed fluctuates within a range of ± 50 rpm about the reference speed. This is due to selection of neural network to implement speed controller. Only one hidden layer and two neurons in the hidden layer are used. Due to variation in the speed, it was not possible to conduct load test in closed loop.

6.4 CONCLUSION

The speed of dc motor can be varied using phase controlled converter. The 3-ϕ bridge converter is designed and developed. The performance of the converter is investigated with R load, R-L load and motor load. Various voltage and current waveform are recorded. These waveforms confirm the theoretical one.

The open loop performance of dc drive is experimentally obtained. The load test is conducted and various performance curves are plotted. The drop in speed with increasing load torque shows the open loop performance is not satisfactory, where constant speed is desired.

To improve the performance of dc drive closed loop operation is required. A neural controller is used for processing the speed. Patterns for neural controller has been generated using fuzzy logic. Neural network is trained off-line for speed error processing.

Proportional controller has been used for current error processing. A simulation for closed loop dc drive system is carried out to design fuzzy based rule for minimum overshoot and settling time. These rules are used to generate input/output pattern for off-line training of the controller.

Actual speed and actual current are measured successfully through software for closed loop operation. Response of the dc drive is checked for step reference speed input and found satisfactory. The close loop performance of the drive is also investigated.

Future Scope

The present dissertation deals with the design and development of a phase controlled converter fed dc motor speed control. The speed controller is designed using ANN. The pattern for training the network are obtained using fuzzy logic.

Although the system operation is found satisfactory, some improvements are possible in the present work.

1. The neural network used to represent speed controller may be designed using more hidden layers and more neurons for better performance.
2. Model reference adaptive control method using ANN may be used for speed control of phase controlled converter fed dc motor drive.
3. Since the converter operates in single quadrant, therefore current becomes discontinuous at large firing angle. To avoid discontinuous operation of converter at large firing angles, instead of fully controlled bridge half controlled bridge converter should have been used.
4. Online training of neural network can be employed by altering the software which will give more accurate results.

REFERENCES

1. Sen, P.C., "Thyristor dc drive", Willy Interscience Publication 1981.
2. Bose, B.K., "Technology trend in microcomputer control of electrical machines", IEEE Trans., Ind. Ele. Vol. 35, No. 1, Feb. 1988.
3. El-Sharkawi, M.A. and Siri Weerasooriya, "Development and implementation of self-tuning tracking controller for dc motors", IEEE Trans., Energy Conversion, Vol. 5, No. 1, March 1990.
4. Prakasha and Verma, V.K. "Micro-computer based dc drive in sliding mode", IEE Ann. Paper meet., Vol. 72, Feb. 1992.
5. Tzes, Anthony, Peng, Pei-Yuan and Houng, Cheng-Chung, "Neural network control for dc motor micromaneuvering", IEEE Trans., Industrial Elect., Vol. 42, No. 5, Oct. 1994.
6. Janardanan, E.G. and Gajendra, F., "Artificial neural network based control scheme for a dc motor", IEE Jour. 1998.
7. Bose, B.K., "Expert system, fuzzy logic and neural network applications in power electronics and motion control", Proc. IEEE, Vol. 82, No. 8, August 1994.
8. Rao, Vallura B. & Rao, Hayagriva, V., "Neural network and fuzzy logic", BPB publication, second edition 1996.
9. Low, Teek-Seng, Lec, Tong-Heng and Lim, Hock-Koon, "A methodology for neural network training for control of drives with nonlinearities", IEEE Trans., Ind. Elec., Vol. 39, No. 2, April 1993.
10. Siri Weerasooriya and Sharkawi, M.A., "Identification and control of dc motor using back-propagation neural networks", IEEE Trans. Energy Conversion, Vol. 6, No. 4, Dec. 1991.
11. Pal, Sankar K., "Soft computing tools and pattern recognition", IETE Journal of Research, Vol. 44, No. 182, Jan-April 1998, pp. 61-87.
12. Hebb, D.O., "The organisation of behaviour", NY, Willy Interscience publication, 1949.

13. Zurada, J.M., "An introduction to artificial neural networks", Jaico publishing house, Mumbai, 1994, ISBN-81-7224-2662.

14. Chu, Reynold S., Shoureshi, Rahmat and Tenorio, Manoel, "Neural networks for system identification", IEEE, control systems magazine, April 1990.

15. Nguyen, Derrick H., and Widrow, Bernand, "Neural networks for self-learning control systems", IEEE control systems magazine, April 1990.

16. Kim, Min-Huei, Simoes, Godoy, M., and Bose, B.K., "Neural network based estimation of power electronic waveforms", IEEE Trans., Power electronics, Vol.11, No. 2, March 1996.

17. Chen, F.C., "BP neural network for non linear self tuning adaptive control", Proc. IEEE, Int. Sym. Intelligen control (Albaney NY), Sept. 25-26, 1989, pp. 274-279.

18. Narendra, K.S., and Parthsarthy K., "Identification and control of dynamical systems using neural networks", IEEE Trans., Neural networks, Vo. 1, pp. 4-27, March 1990.

19. Zadeh, L.A., "Fuzzy sets", Informat. Contr. Vol. 8, pp. 338-353, 1965.

20. Mamdani, E.H. and Assilian, S., "An experiment in linguistic synthesis with a fuzzy logic controller", Int. Jour., Man machine studies, Vol. 7, pp. 1-13, 1975.

21. Ishibuchi, H., Fujuoka R., and Tanaka, H., "Neural network that learns fuzzy if-then rules", IEEE Trans., Fuzzy systems, Vol. 1, No. 2, pp. 85-97, 1993.

22. Ishibuchi, H., Tanaka, H, & Okada, H., "Fuzzy neural network with fuzzy weights and fuzzy biases", Proc. IJ CNN San Fransisco, 1993, pp. 1650-1655.

23. Hayashi, Y., Buckley, J.J., & Lzogula, E., "System engineering application of fuzzy neural networks", Proc. IJCNN, Baltimore, 1992, pp. 413-418.

24. Dubey, G.K., "Power semiconductor controlled drives", Prentice Hall, New Jersey, 1989.

APPENDIX-I

Measurement of Motor Parameters

Measurement of machine parameters is necessary part of the present dissertation work. Method to measure various parameter is as follows :

To measure armature resistance, armature is connected with dc supply. Field is left unexcited. Readings of ammeter and voltmeter are recorded. Armature resistance can be calculated using these readings.

For measuring armature inductance above procedure is used except that armature is connected with ac supply.

Moment of Inertia

The basic principle of measuring moment of inertia is as follows : when the source of energy to a rotating system is cut off, it will continue to rotate due to the initial kinetic energy stored in the system. This energy is used up to supply the rotational losses of the system, it slows down and gradually stops. The power is given by

$$P = \frac{d}{dt}\left(\frac{1}{2} J \omega^2\right) \text{ where } \omega = 2\pi n$$

$$P = J 4\pi^2 n \frac{dn}{dt}$$

To determine the rate of change of speed, the machine is run to a speed, a little higher than the normal, and the power supply to it is cut off. The speed at different instants is recorded and a curve is plotted, shown in Fig. A1 and A2. The rate of change of speed is obtained by the slope of the tangent to the speed-time curve at the required speed. The speed-time curve is linear in small region around the desired speed. If the speed drops from n_1 to n_2 in time t_1, then

$$\frac{dn}{dt} = \frac{n_1 - n_2}{t}$$

To determine the moment of inertia of the armature, an electric retarding brake, in the form of a resistance across the terminal is applied. The K.E. of the

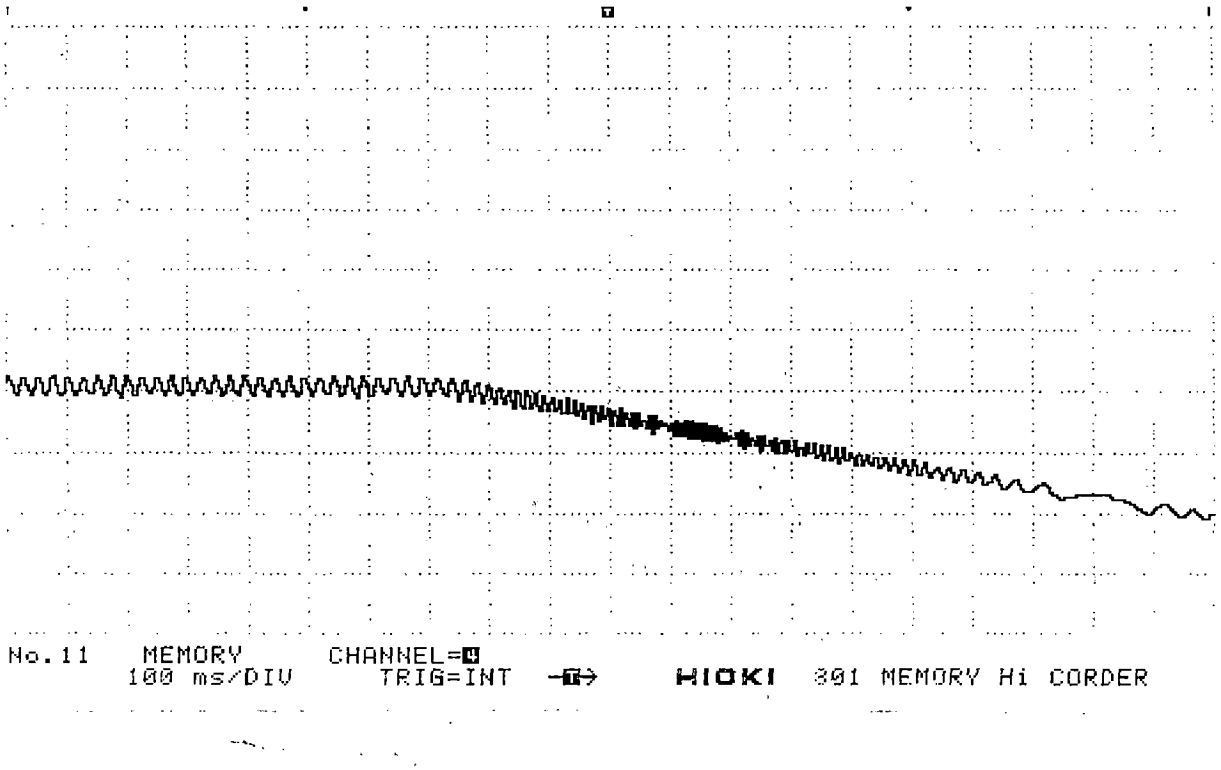


Fig. A-1 Variation of motor speed at no load

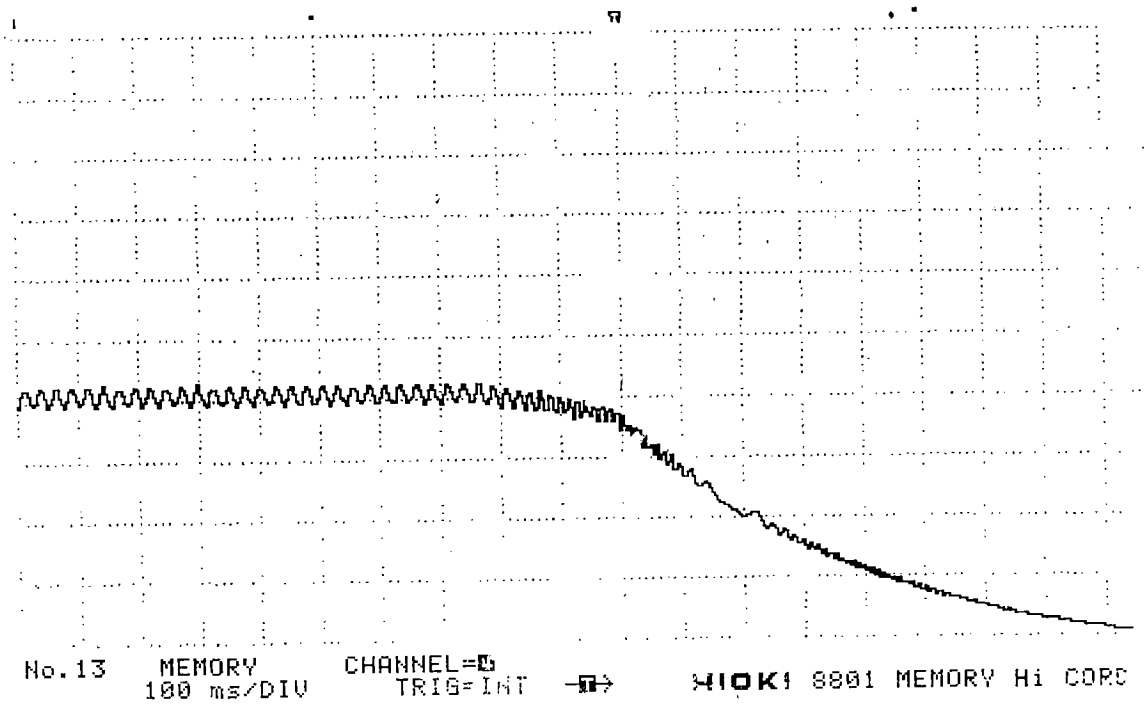


Fig. A-2 Variation of motor speed with electric brakes

armature has to supply the electric power to the resistor besides its own rotational losses. If, for the same speed fall from n_1 to n_2, the time taken is t_1 without electric brake and t_2 with electric brake.

$$P = J.4\pi^2 n \frac{n_1 - n_2}{t_1}$$

$$P + P' = J.4\pi^2 n \frac{n_1 - n_2}{t}$$

Here P' is the power dissipated in resistance

$$P = P' \frac{t_2}{t_1 - t_2}$$

The graph is shown in Figs. A-1 and A-2 for no load condition and loaded condition. From here t_1 and t_2 has been calculated as 1.50 msec and 37.5 msec respectively.

$I = 4.3$ Amp., $R_{ext} = 50\ \Omega$, $R_a = 5\ \Omega$

$$P' = I^2 (R_a + R_{ext}) = 1016.95 \text{ Watts.}$$

$$P = P' \frac{t_2}{t_1 - t_2} = 338.98 \text{ Watts.}$$

Speed drop is taken from 1359 rpm to 1265 rpm.

$$\omega_m = 1359 \times \frac{2\pi}{60} = 143.31 \text{ rad/sec.}$$

$$\frac{d\omega}{dt} = \frac{(1359 - 1265)}{150 \times 10^{-3}} \times \frac{2\pi}{60} = 65.624$$

$$J = \frac{P}{\omega_m \frac{d\omega}{dt}} = \frac{338.98}{143.31 \times 65.624}$$

$$= 0.0346 \text{ Kg m}^2$$

Viscous friction Coefficient

Viscous friction coefficient can be measured by loading the motor with generator. Field and armature are excited. Various readings for armature voltage, armature

current and field current have been recorded. Then a graph is plotted between power loss and field current. This graph is extrapolated so that power loss due to friction and windage loss can be obtained. Using this loss, B can be calculated as follows :

$$T_{FW} = B\omega$$

or $$B = \frac{P_{FNL}}{\omega}$$

It comes to be 0.00417 N-m sec/rad.

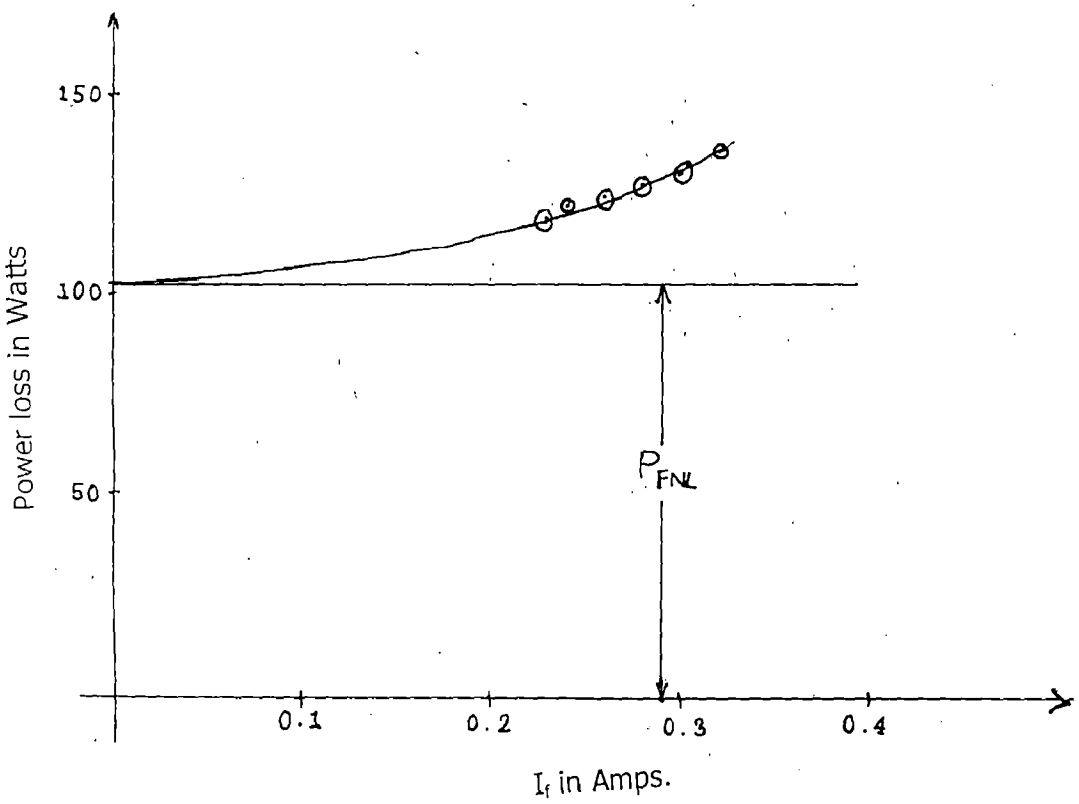


Fig. A-3 Power loss in watts vs field current in amps.

APPENDIX-II

Specifications

Thyristor :

Voltage rating	=	220 V
Current rating	=	12 A

Motor :

Voltage rating	=	230 V
Current rating	=	4 A
Rated Speed	=	1500 rpm
Power	=	1 HP
R_a	=	10.52 Ω including external resistance of 4 Ω
L_a	=	0.167 H
K_b	=	1.4252 V rad/s
J	=	0.0346 Kgm^2
B	=	0.00417 N-m sec/rad

Generator :

Voltage rating	=	230 V
Current rating	=	4.6 A
Rated Speed	=	1500 rpm
Power	=	0.75 kW

Load :

Lamp load

APPENDIX-III

8031 MICROCONTROLLER

PIN DESCRIPTION

V_{cc} : Supply voltage

V_{ss} : Circuit ground

Port 0 : It is an 8-bit open drain bi-directional I/O port. As an output port each pin can sink 8 LS TTL inputs. Port 0 is also the multiplexed low-order address and data but during accesses to external program and data memory.

Port 0 also receives the code bytes during programming of the EPROM parts, and outputs the code bytes during program verification of ROM and EPROM parts.

Port 1 : It is an 8-bit bi-directional I/O port with internal pull-ups.

Port 2 : It is an 8-bit bi-directional I/O port with internal pull-ups.

Port 2 emits the higher order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses.

Port 3 : Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. Port 3 also serves the functions of various special features, as listed below:

PORT PIN	ALTERNATE FUNCTION
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

The major features are :

- 8-bit CPU
- On-chip oscillator and clock circuitry
- 32 I/O lines
- 64 K address space for external data memory
- 64 K address space for external program memory
- Two 16-bit timer/counter
- A five-source interrupt structure with two priority levels.
- Full duplex serial port
- Boolean processor.

Memory Organization :

The 8031 has two separate address spaces for program memory and data memory. The program memory can be upto 64 kB long. The lower 4k may reside on-chip. The data memory can consist of upto 64 kB of off-chip RAM, in addition to which it includes 128 bytes of on-chip RAM, plus a number of special function registers (SFRs) as described below :

1. **Accumulator (ACC)** : It's address is OEOH. The mnemonics for accumulator–specific instructions, however, refer to the accumulator simply as A.

2. **B Register :** It's address is OFOH. It is used during multiply and divide operations. For other instructions, it can be treated as another scratch pad register.

3. **Program Status Word (PSW)** : It's address is ODOH. It contains program status information.

4. **Stack Pointer (SP)** : It's address is 81H. It is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM the stack pointer is

initialized to 07H after a reset. This causes the stack to begin at location 08H.

5. **Data Pointer (DPTR)** : The DPTR consists of high byte (DPH) and low byte (DPL). It's address is 83H and 82H. Its intended function is to hold a 16-bit address. It may be manipulated as 16-bit register or as two independent 8-bit registers.

6. **Ports 0 to 3** : P0, P1, P2 and P3 are the SFR lathes of ports 0, 1, 2 and 3 respectively.

7. **Serial Data Buffer** : Serial Data Buffer is actually two separate registers, a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer where it is held for serial transmission. When data is moved from SBUF, it comes from the receiver buffer.

8. **Timer Registers** : Register pairs (TH0, TL0), (TH1, TL1) and (TH2, TL2) are the 16-bit counting registers for timer/counters 0, 1 and 2 respectively.

9. **Control Regisers** : Special Function Registers IP, IE, TMOD, TCON, SCON, PCON contain control and status bits for the interrupt system, the timer/counters and the serial port.

10. **Oscillator and Clock Circuit** : XTAL1 and XTAL2 are the input and output of a single stage on-chip inverter, which can be configured with off-chip component as a Pierce Oscillator.