ADAPTIVE CONTROL OF CONVERTER FED D.C. DRIVE

A DISSERTATION

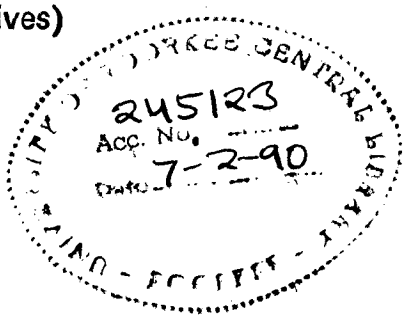
submitted in partial fulfilment of the requirements for the award of the degree of

MASTER OF ENGINEERING

in

ELECTRICAL ENGINEERING

(Power Apparatus and Electric Drives)



By

P. S. PUTTASWAMY

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

SEPTEMBER, 1989

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the dissertation entitled, ADAPTIVE CONTROL OF CONVERTER FED D.C. DRIVE, 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, submitted in the Department of Electrical Engineering, University of Roorkee, Roorkee, is an authentic record of my own work carried out for a period of about seven months from August 1988 to Feb 1989 under the supervision of Dr.BHIM SINGH and Dr.AMBRISH CHANDRA, Readers in Electrical Engineering, University of Roorkee, Roorkee.

The matter embodied in this dissertation has not been submitted by me for the award of any other degree or diploma.

Date: 16.9.89 (CANDIDATE'S SIGNATURE)

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

Dr.AMBRISH CHANDRA
Reader
Department of Electrical Engg.
University of Roorkee
Roorkee-247667

Dr.BHIM SINGH
Reader
Department of Electrical Engg.
University of Roorkee
Roorkee-247667

ACKNOWLEDGEMENTS

I wish to express my deep sense of gratitude and indebtedness to my guides, Dr.BHIM SINGH and Dr.AMBRISH CHANDRA, Readers in Electrical Engineering for invaluable assistance, excellent guidence and sincere advice given by them during this thesis work.

I am highly thankful to Dr.R.B.SAXENA, Professor and Head of the Department of Electrical Engineering, University of Roorkee, Roorkee for providing laboratory facilities.

I am very much grateful to our faculty member Dr.V.K.VERMA for his invaluable suggestions and kind co-operation during the work. I am also grateful to Mr.M.K.VASANTHA, Mr.Y.P.SINGH and Mr.PRAMOD AGARWAL for their kind suggestions.

I am highly thankful to my wife N.NALINI for her constant encouragement and co-operation during this work.

I am also thankful to P.E.S.COLLEGE OF ENGINEERING, Mandya, for sponsoring to this M.E. Course.

I am deeply grateful to all the laboratory staff for their co-operation during the fabrication of this work.

Thanks are also due to those who helped me directly or indirectly in preparing this dissertation.

(P.S.PUTTASWAMY)

SYNOPSIS

This dissertation deals with the experimental studies on self tuning control of a closed loop variable speed d.c. motor drive system.

A self tuning controller for speed control is developed and realised through software on IBM PC. And thyristorized three phase fully controlled converter, microprocessor controlled firing circuit, current controller (PI) and digital speed measurement circuit have been designed, fabricated and tested.

Chapter-I consists of literature review and brings out the advantages of self tuning control over other control schemes. Chapter-II gives the complete details of self tuning controller design. Chapter-III deals with the study of motor speed control and system hardware. Chapter-IV describes the system software and their flow charts. Chapter-V deals with the experimental results. In the last chapter-VI conclusions and scope of further work have been described.

CONTENTS

Contd..

CHAPTER PAGE NO.

Contd..

CHAPTER-1

INTRODUCTION

1.1 GENERAL

Many modern variable speed drives require a precise and smooth control of speed with longterm stability and good transient performance. From the very beginning, the conventional d.c. motors have been used as variable speed drives in many industrial applications due to the following advantages.

(i) D.C. motors can provide high starting torque, which are required for several applications such as traction drives.

(ii) D.C. motor's speed can be controlled easily over a wide range both below and above rated speed.

(iii) The methods of control are similar and less expensive than those of a.c. motors.

In modern d.c. drives, the classical motor-generator set has been replaced by a solidstate controlled d.c. drive, which has numerous advantages such as minimum maintenance, less bulk and weight, higher efficiency, faster time response and above all low cost[1].

For a reversible drive with continuous speed control over a wide range, the d.c. motor drive is the most popular

solution. The speed of the d.c. motor can be controlled by controlling either armature voltage or the field current. Former is for speed control below the rated value and the latter is for speed control above the rated value of speed of the motor. D.C. motors can closely match the needs of the application as the speed-torque characteristics can be varied to any useful form, for both motoring and regenerating applications. Dynamic or regenerative braking is easily obtainable with the d.c. motors for application, requiring quick stopping or speed reversals.

As mentioned earlier, d.c. motors are easily adaptable to variable speed drives. The speed characteristics of a seperately excited d.c. motor is given by

$$N = (V_a - I_a R_a) \mid KI_f$$

The seperately excited d.c. motor is rapidly and efficiently controlled by the variation of armature voltage and field current.

Solid state controllers enable in developing more accurate, efficient and compact control scheme for d c. motors [1,2]. In addition they help in obtaining variable characteristics of these motors. The conventional methods of speed control are gradually being replaced by thyristor control. The use of thyristors for power control affords considerable advantages over conventional methods owing to the possibilities of obtaining better economy, efficiency,

reliability, fastness facilities and their compatibility to adopt for closed loop system of speed control.

Now a days, the speed control of a d.c. drives is also tried with adaptive controllers. Adaptive control has been a challenge for control engineers for a long time [3]. It has been their dream for more than a quarter of a century. It arose from a desire and need for improved performance of increasingly complex engineering systems with large uncertainties. This is specially important in system with many unknown parameters that change with time. An adaptive system is one that continually monitors the changes and adjusts the control parameters automatically to maintain good performance. Hence, adaptive systems can be considered as a class of control systems that have been made deliberately non-linear.

The tuning problem is one reason for using adaptive control. It is a well known fact that many processes can be regulated satisfactorily with proportional integral (PI) or proportional integral derivative (PID) regulators. It is fairly easy to tune a PI regulator which only has two parameters to adjust. However, for an installation which has several hundred regulators, it is a substantial task to keep all regulators well tuned. A PID regulator which has three or four parameters is not always easy to tune, particularly if the process dynamics is slow. The derivative action is

therefore frequently switched off in industrial controllers. Since many control loops are not critical, three term controllers will be extensively used in future too.

Conventional control theory deals with the control of dynamical systems whose mathematical representations are completely known. In contrast to this adaptive control it refers to the control of partially known systems. For many years, there has been an increasing interest in adaptive control which can be attributed to the fact that there is invariably some uncertainity in the dynamic characteristics of most practical systems. The tools of conventional control theory, even when used efficiently in the design of controllers for such systems, are inadequate to achieve satisfactory performance in the entire range over which the characteristics of the system vary.

While feedback, in general, improves the performance of a system it could also make the system unstable [3]. Since adaptive control systems are non-linear feedback systems, there is the distinct possibility that such system can become unstable.

Research on adaptive control was very active in the fifties. It was motivated by the need of autopilots for high performance aircraft. The work was characterized by a lot of enthusiasm, bad hardware, and nonexisting theory. Interest in the area diminished due to lack of well developed stability theory and a fight disaster. Due to some recent developments

in the stability theory of such systems, the application of adaptive control to practical systems is being attempted on a large scale [3,4].

1.2 LITERATURE SURVEY REVIEW

1.2.1 Adaptive Control Schemes

The adaptive control of systems has been a dream of control system theorists and practitioners for atleast a quarter of a century. In the past decade the progress of adaptive control has been slow in coming to fruition. The early schemes for adaptive control, though ingeneous, did not gaurantee the stability of the system. Globally stable adaptive control algorithm have been developed very recently. With recent advances in microprocessor technology, it has become feasible to implement adaptive control algorithms efficiently in real time at reasonable cost.

To understand the recent developments in theory and the keen interest shown by industries in adaptive control promoted to organise the workshop on applications to the adaptive cotrol. Many workshops have been held to bringout the fruitful application [3].

Clarke and Gawtrop [5] have discussed the closed loop properties of various classes of selftuner, convergence concepts and results. They have suggested a new technique to design a selftuning controller which does not appear to

suffer excessively from the disadvantages of previous adaptive methods. Some of the technical problems involved with implementing selftuners on small computers or microprocessors are also discussed.

Allidina and Hughes [6] have discussed a selftuning controler that minimises cost function incorporating system input, output and setpoint variations, and adapts in such a way that the closed loop poles are located on prespecified locations. The link between classical control strategy of pole assignment and the suboptimal strategy of the selftuning controller is demonstrated and made use to produce generalised controller which possesses the advantages of both. The proposed controller combines the robustness of a pole assignment regulator with the ease of reference tracking. Simulation methods for first and second order systems are also discussed.

Gawtrop [7] has discussed the design of selftuning controller including rational transfer function terms in the associated cost function. Two interpretations of the selftuning controller, a model reference adaptive control and a selftuning least square predictor in conjunction with conventional compensation are examined in detail. Using standard form of techniques, the discrete time systems with step responses are determined. The proposed controller shows the advantages over Smith Control law.

Brickwedde [8] has discussed the application of microprocessor based adaptive control for electric drives. For its implementation as an on-line adaptive control system, the recursive least squares algorithm for the on line parameter identification combined with a pole assignment procedure is suggested. The proposed controller is implemented on a 16 bit microprocessor to control the speed of a 40 k.w d.c. drive. Experimental results of a drive under load conditions are also discussed.

A robust selftuning controller that can track a constant reference and rejects load disturbance developed by Balasubramanian and Kong Hung Wong [9], proposed controller is implemented on a microprocessor to control the speed of a d.c. motor under varying load and operating conditions. Experimental results are also discussed, which confirm the controller excellent adaption capability as well as transient recovery under load changing conditions.

Stephan et.al. [10] have discussed a cascade speed control strategy for a thyristor fed d.c. motor subjected to parameter variations. This is called as dual mode used with inner current loop and outer speed loop. The scheme is implemented on a 16 bit microprocessor using a floating point coprocessor to control the speed of a 1.1 k.w d.c. motor, and experimental results are also given. The proposed controller makes the behaviour of armature circuit practically

independent of the current conduction mode. Response is extremely fast and it gaurantee the stability and predefined performance under all operating conditions.

Liu et.al. [11] have discussed a controller design based on the placement of poles and zeros of the state tracking error transfer function. The problem formulation, parameter and state estimation, considering noisy and noise free systems are discussed. The proposed self tuning control is found very effective for both regulation and reference tracking signal.

A common problem in the d.c. motor drives is torque saturation. In order to overcome this, a generalised predictive control is suggested by Tsang and Clarke [12], which shows improvements in performance without major increase in computations.

Cluett et.al. [13] have discussed the consolidation of stability and robustness results based on input-output theory for discrete adaptive control systems. In this, an algorithm for a generic direct discrete time adaptive controller is presented. To obtain the stability of the system normalisation, and weighting of the polynomials are suggested.

1.2.2 Firing and Control Schemes

Ilango et.al. [14] have discussed the compact firing

scheme for three phase bridge rectifiers using minimum integrated circuit components. The proposed scheme had shown fast response for triggering angle and gives a fullrange of voltage, because the response of the firing angle to control voltage is almost instantaneous.

Alimirbod and Ahmed el-amawy [15] discussed a general purpose microprocessor based control circuit for a three phase controlled rectifier. For the proper operation of controlled rectifiers, the gating patterns are synchronized with the zero crossing of line to line voltage by properly compensating the effect of source impedance by compensation circuit. The proposed controller exihibited superior performance to previously presented circuits. The hardware and software implementation on 8086 microprocessor are discussed.

A firing scheme based on microprocessor to control an antiparallel connected 3 phase thyristor dual converter is discussed by Tang et.al. [16]. Look up algorithm is used to speed up the response, which gives full range control of the firing angle between 0-180^{o} for both positive and negative current control. The proposed scheme is reported to give good dynamic response and high system reliability.

Independently of the method of algorithm used to control thyristor firing of a three phase static converter, the signals that synchronize them with a.c. supply are always

necessary. A microprocessor based scheme is proposed [17] for this purpose. This scheme is tested as a part of the complete digital current control system feeding a 5 k.w d.c. motor also in a rectifier system fed by a 13 KVA synchronous generator. The results obtained are very good, and for a self controlled synchronous motor system, this scheme is more precise than a set of selective filters.

Remy Simard and Rajagopalan [18] have discussed a simple equidistant pulse firing scheme. The proposed scheme is economical to use with individual phase control circuits for industrial applications. Experimental results obtained on a laboratory d.c. drive are also discussed.

El-Bolok [19] has described a novel microprocessor based scheme for constant angle triggering operating under a variable anode frequency, in which the triggering angle is controllable by software. Implementation and testing of this scheme had shown a fast response to changes in anode supply frequency. This scheme is not affected by the number of phases of the supply voltage and/or its phase sequence.

Mario Benedetti and Carlos F. Christiansen [20] have described the design of a trigger system for full or half controlled thyristor converter upto 6 pulses using 8 bit microprocessor. This scheme provides more flexibility in which the triggering angle may take any value from 0 to 180^{o} between pulses, thus achieving the maximum rate of change of

the output voltage. The necessary hardware is also reduced to minimum. The design procedure and experimental setup are discussed.

Thadiappan Krishnan and Bellamkonda Ramaswami [21] proposed the design, construction and testing of thyristorized speed control unit for a seperately excited d.c. motor. The motor is fed from a three phase six pulse fully controlled thyristor bridge. The speed and current loops are used to maintain the desired speed and to provide fast response, overcoming the effect of disturbances. But this scheme is useful for only one direction of rotation.

Soon Chan Hong and Min ho Park [22] have described a method to improve the efficiency of d.c. motors based upon the field control. The power circuits and the control loops are suggested to realise the principle, and the real time control is implemented with a microprocessor. Experimental results had shown the remarkable improvement in efficiency as well as stability.

Sule et.al. [23] have proposed a digital speed control system for a d.c. motor with a pulse tachogenerator for speed sensing. The scheme provides a fast and accurate digital speed measurement and control by means of a microprocessor system. The measurement accuracy obtained is $\pm$ 0.0125 percent with reference to maximum speed and the time constant is approximately 8 m.s.

Plant et.al. [24] have proposed a suitable method for microprocessor implementation of controlling the speed of an S.C.R. controlled d.c. motor drive which uses a halfwave single phase supply, and realisation by a microprocessor. The determination of the necessary control requires only table look up operations and computations are kept minimum. The control algorithm, using the table look up, becomes simple, meets the real time requirement, and its implementation is well within the capability of the present day microprocessors. Experimental results of a 3HP d.c. motor control are also discussed.

Guyolvier et.al. [25] have discussed the operation of the modified thyristor bridge converter, the control algorithm and microprocessor implementation. The proposed scheme is shown to have faster response, precise and equidistant control of the thyristor triggerings.

Verma and Agarwal [26] have described the design and experimental studies of variable d.c. drive, fed from a single phase dual converter. The drive incorporates an inner current loop (PI) and outer speed loop (PI). The analytical and experimental studies had shown good transient and steady state drive performance.

Tadashi Egame and Takeshi Tsuchiya [27] have discussed a partial state and an output feedback control system synthesis method with feedforward compensation. It

doesn't make use of the state observer procedure. The proposed scheme can be applied to all control systems and its effectiveness tested experimentally.

Nandam and Sen [28] have discussed the comparative study of proportional and integral proportional schemes, for the speed control of a d.c. drive using both analog and microprocessor based digital system. It is confirmed that IP control has distinct advantages over PI control. Experimental and simulation results are discussed.

1.3 SCOPE OF THE PRESENT WORK

The exhaustive literature survey reveals that little research work has been reported on the performance of converter fed d.c. drive using adaptive controller. Therefore, a need is felt to investigate the variable speed operation and performance of a adaptive control of the d.c. drive.

In the proposed scheme, a three phase a.c. supply of fixed frequency and voltage is converted into a variable voltage d.c. supply which is being fed to a d.c. motor. The commutation of thyristor converter is achieved by line commutation. Self tuning controller suggested by Clarke and Gawtrop [5] is used to control the speed of a d.c. drive.

The objectives of the proposed work are the following:

(i) To design a self tuning controller for the speed control of a d.c. drive.

(ii) To design and develop a suitable firing scheme and control schemes for operating the converter.

(iii) To design proportional plus integral control for current loop processing.

(iv) To study the feasibility of the whole system as a variable speed drive and to obtain experimentally its no load and load characteristics.

The whole scheme has been fabricated and tested in a laboratory and the performance of the motor is obtained experimentally.

1.4 OUT LINE OF CHAPTERS

In Chapter-II, the various aspects of different types of adaptive controllers are described. The algorithm involved in the design of self tuning controller is also discussed.

In Chapter-III, the complete hardware of the present scheme is described in detail. The design of power circuit, microprocessor based firing control circuit using reduced hardware components and pulse amplifier circuit are also given in this chapter.

The implementation of software and flow charts of the

seperately excited d.c. motor speed control is illustrated in Chapter-IV. The various subroutines used in conjunction with the main program such as ADC subroutine, MODE selection subroutine, MULT subroutine, SMULT subroutine, PI processing subroutine V_{dc}^* CONSTRAINED subroutine, IR_0 interrupt subroutine, IR_1 interrupt subroutine are also developed in this chapter.

The behaviour of a drive on no load as well as on load condition is described in detail along with the performance characteristics in Chapter-V.

The suggestions are proposed to improve the drive performance is given in Chapter-VI.

The pin detail connections and function of various chips, the design of PI controller for current loop, basic language program for the selftuning controller and machine level language program for microprocessor system are given in appendices.

CHAPTER- 2

ADAPTIVE CONTROLLER

2.1 GENERAL

Different types of controllers are suggested in literature to control the speed of a d.c. motor. In this work, adaptive selftuning controller (STC) is used for this purpose. An adaptive controller is one that continuously monitors the change in system parameters and adjusts its control parameters automatically to give good performance. In this chapter, selftuning controller algorithm is discussed in detail.

2.2 SYSTEM MODEL

A physical object, in whose behaviour we are interested in studying, affecting or controlling is known as system. The knowledge of the properties of a system is generally called a model. The concept of a mathematical model is necessary when complex problems are treated. A mathematical model of a given system can build up by two approaches. In the first approach, by looking directly into the mechanisms of the system that generate signals and variables inside the system and construct a mathematical model. In second approach, knowledge of the systems

mechanism is incomplete or system properties are inpredictable, in those cases, signals produced by the system can be measured and be used to construct a mathematical model.

A practical system can be either single input/single output or multi input/multi output. But as for as d.c. motor is concerned, it can be regarded as single input/single output. Therefore, in this investigation, only the single input single output system is discussed. The mathematic model considered, in this work, is of the discrete form (difference equation) due to digital computer application. Many types of discrete mathematical model such as ARMA (Auto Regressive Moving Average), ARMAX (Auto Regressive Exogeneous), and LR (Linear Regression) are used in adaptive control techniques. ARMA model is not suitable for the present problem because it has no control over input. ARMAX model which uses the principal property of Recursive least square identification is not holds good because the prediction error is uncorrelated. Therefore to overcome this difficulty, 'Recursive extended Least squares' (RELS) identification is used with this type of model. RELS is not robust as that of RLS and the convergence can not be proved for all cases, it also increases the computational burden and hence it may create problems in real time implementation. That is why linear regression model is considered here due to its suitability and simplicity.

The system model is assumed to be of the form

$$A(Z^{-1})\, y(t) = Z^{-k} B(Z^{-1})\, u(t) + \xi(t) \qquad .. (2.1)$$

Where A and B are polynomials in Z^{-1} (backward shift operator) defined as

$$A(Z^{-1}) = a_o + a_1 Z^{-1} + ... + a_{n_a} Z^{-n_a}$$

$$B(Z^{-1}) = b_o + b_1 Z^{-1} + ... + b_{n_b} Z^{-n_b}$$

Where n_a and n_b, are the orders of polynomials A and B respectively. K represents the system time delay in sample intervals. y and u are the system output and input, with $\xi(t)$ an uncorrelated random sequence of zero mean which disturbs the system.

2.3 TEST SIGNALS

To obtain the information about the system and to identify it, the system should be excited by some input signal. Several researchers have treated the design of input signal for enhancing the parameter estimation for discrete time processes used in adaptive control techniques. The input signal should excite all models of the system to guarantee, the estimated parameters converge to the reasonably correct values. The signal having this quality is called 'rich enough' (having sufficiently rich frequency content) or persistently existing. While doing literature

survey it is found that most of researchers have considered white noise as a suitable signal. Hence, in this work, white noise is chosen for excitation purpose [5].

2.4 SYSTEM IDENTIFICATION

The type of a model mainly depends upon the method of identification. The identification of a system model provides the essential aspects of an existing system and represents the knowledge of that system in useful form. Identification may be either 'on-line' (recursive) or 'off-line' algorithm. On-line identification has two main disadvantages in comparison to 'off-line' identification. Firstly, a decision must be made regarding the structure of the model before starting the identification. Secondly, since 'off-line' methods can reprocess data many times, they are more accurate than recursive methods. Recursive least square identification algorithm is the most popular technique which has been used successfully in many applications. Therefore, in the thesis, this algorithm has been used and is briefly discussed below.

The RLS identification technique is mainly used to treat the LR model (eqn.2.1) defining the parameter vector:

$$\hat{\theta}^T(t) = [a_1, \ldots a_{n_a}, b_o, b_1, \ldots b_{n_b}] \quad .. (2.2)$$

and

$$x^T(t) = [-y(t-1), \ldots -y(t-n_a), u(t-k), \ldots u(t-k-n_b)] \quad .. (2.3)$$

Vector containing measured data.

Vector containing measured data.

Therefore, the equation (2.1) can be written as:

$$y(t) = \hat{\theta}^T(t)\, x(t) + \xi(t) \qquad .. (2.4)$$

Let $\hat{y}(t)$ be an estimate of $y(t)$ based on $\hat{\theta}(t)$ and $x(t)$ and it is as:

$$\hat{y}(t) = \hat{\theta}^T(t)\, x(t) \qquad .. (2.5)$$

There may be an error in the estimation because the data may be complicated by random measurement of noise, error in the model or combination of both. It is given by:

$$\epsilon(t) = y(t) - \hat{y}(t) = \xi(t) \qquad .. (2.6)$$

Now $\hat{\theta}$ is chosen in such a way that the criterion J is minimized.

$$J = \sum_{t=1}^{m} \epsilon^2(t) \qquad .. (2.7)$$

Expressing $J = (y - \hat{\theta}^T x)^T (y - \hat{\theta}^T x)$

$$= y^T y - y^T \hat{\theta}^T x - x^T \hat{\theta} y + x^T \hat{\theta}\hat{\theta}^T x$$

where

$$X = \begin{bmatrix} x^T(t) \\ x^T(t+1) \\ \vdots \\ \vdots \\ x^T(t+m) \end{bmatrix}$$

$$
= \begin{bmatrix}
-y(t-1) \ldots -y(t-n_a),\ u(t-k) \ldots u(t-k-n_b) \\
-y(t) \ldots -y(t-n_a+1), u(t-k+1) \ldots u(t-k-n_b+1) \\
\vdots \\
-y(t+m-1) \ldots -y(t-n_a+m-1), u(t-k+m-1) \ldots u(t-k-n_b+m-1)
\end{bmatrix}
$$

and

$$Y = [y(t),\ y(t+1), \ldots y(t+m)]^T$$

$$m = \text{number of samples}$$

$$\frac{\delta J}{\delta \hat{\theta}} = -2\, X^T y + 2\, X^T x\, \hat{\theta} = 0$$

$$X^T x\, \hat{\theta} = X^T y$$

$$\hat{\theta} = (X^T x)^{-1} X^T y \qquad .. (2.8)$$

The above result represents the least square estimator of θ, and the least square estimates can be computed recursively as:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + k(t)\,[y(t) - \hat{\theta}^T(t-1)\, x(t)] \qquad .. (2.9)$$

$$k(t) = \frac{P(t-1)\, x(t)}{[1 + x^T(t)\, P(t-1)\, x(t)]} \qquad .. (2.10)$$

$$P(t) = [1 - k^T x(t)]\, P(t-1) \qquad .. (2.11)$$

where $k(t)$ is considered as the gain matrix and $P(t)$ is proportional to the error covariance matrix.

From equation (2.9) it is evident that the estimated parameter $\hat{\theta}(t)$ is the weighted sum of the last estimation and prediction error $\epsilon(t)$. As time increases, in case of

time in-variant system $\hat{\theta}(t)$. converges to its true value along with prediction error $\epsilon(t)$, gain vector $k(t)$ and covariance matrix $P(t)$ tend to zero. But this new information is not making any contribution to the parameter estimation. In time variant system, if the estimated parameter reach its true value once the above problem is eliminated, but affects the parameter tracking.

The algorithm which is derived above based on criterion J considering equal weightage for every error. For time varying system the above criterion gives an estimate of the average behaviour during the period $1 \leq t \leq m$. As the time increases the number of measurements becomes very large and the contribution of each individual measurements to the parameter estimation tends to zero.

In order to increase the tracking capability, it is necessary to consider a criterion in which older values are discounted by an exponential weighting scheme which places heavy emphasis on more recent data.

Consider a criterion

$$J = \sum_{i=1}^{m} \beta^{m-i} \, t^2(i), \quad 0 < \beta < 1$$

In this criterion the later errors are given more weight than earlier ones.

Therefore, the algorithm explained above can be modified by changing equation (2.11) to

$$P(t) = \frac{[1-K^T(t)\, x(t)]\, P(t-1)}{\beta} \qquad .. (2.12)$$

If $\beta < 1$ then P(t) will not tend to zero and the algorithm is more capable of tracking the parameter variation.

2.5 CONTROL STRATAGIES

Adaptive Control can be classified into two categories

1) Model reference adaptive control (MRAC)

2) Self tuning control (STC)

The choice of reference model is the first step in case of MRAC. The controller parameters are adjusted so that the output co-ordinate of the system agrees with that of the reference model. To perform parameter adjustemnt the value of mismatch between the controlled co-ordinates of the system and the model is to be used. The main problem associated with MRAC is that, to obtain a closed loop response, the system zeros must be cancelled by controller poles. Hence it is not useful for the systems outside the stability region [6]. It is also found that the discrete time cancellation of zeros by poles that correspond to oscillatory continuous time functions can lead to unacceptable intersample behaviour. Therefore this technique is not very much suitable for d.c. motor systems.

The STC consisting with two loops an inner loop or the control loop and an outer loop that adjusts the control

parameters in the inner loop. STC deals with discrete systems and is applicable to a wider class of problems though the stability of such schemes may be harder to prove. Self tuning controller gained more popularity because of some advantages over MRAC. In this thesis self tuning controller [5] is used and here it is discussed in detail.

2.6 SELF TUNING CONTROL

Kalman probably was the first to propose selftuning under the title 'self optimizing'. It had little immediate consequence at the time due to lack of suitable technology. Moreover, MRAC could be implemented using analog hardware encouraging more work on its development. With the development of microprocessor technology there have been stimulating developments in self tuning control, although no guarantee that self tuning would be successful in practical applications, because the theory is proven for small cases only and even under the favourable conditions there is no guarantee of convergence.

Even with the incompleteness of established theoritical results, there is a development work on selftuning controller. The popularity of this control can be gauged from the number of paper published and the succesful applications reported in the literature. Now a days selftuning controller is mostly preferred because of the following reasons:

i) The algorithms used are very simple.

ii) It is derived in discrete form which can be readily implemented on microcomputers.

iii) It has been found possible to establish some theoretical results which, although incomplete indicate that satisfactory stable performance can be expected.

The controlled system parameters are identified in these techniques, on line and the control is calculated using a preselected strategy as shown in Fig. 2.1.

2.7 ALGORITHM

The algorithm for the generalised selftuning controller [5], used in this work can be summarized as follows:

i) Form a generalised cost function (k=1)

$$\phi(t) = Py(t) + Qu(t-1) - Rw(t-1)$$

ii) Estimate the control parameters, $\Theta(t)$ by a standard recursive least square algorithm

$$\hat{\theta}(t) = \hat{\theta}(t-1) + k(t)\,[\phi(t) - \hat{\theta}^T(t)\,x(t)\,]$$

$$k(t) = \frac{P(t-1)\,x(t)}{[1 + x^T(t)\,P(t-1)\,x(t)]}$$

$$P(t) = \frac{[1 - k^T(t)\,x(t)]\,P(t-1)]}{\beta}$$

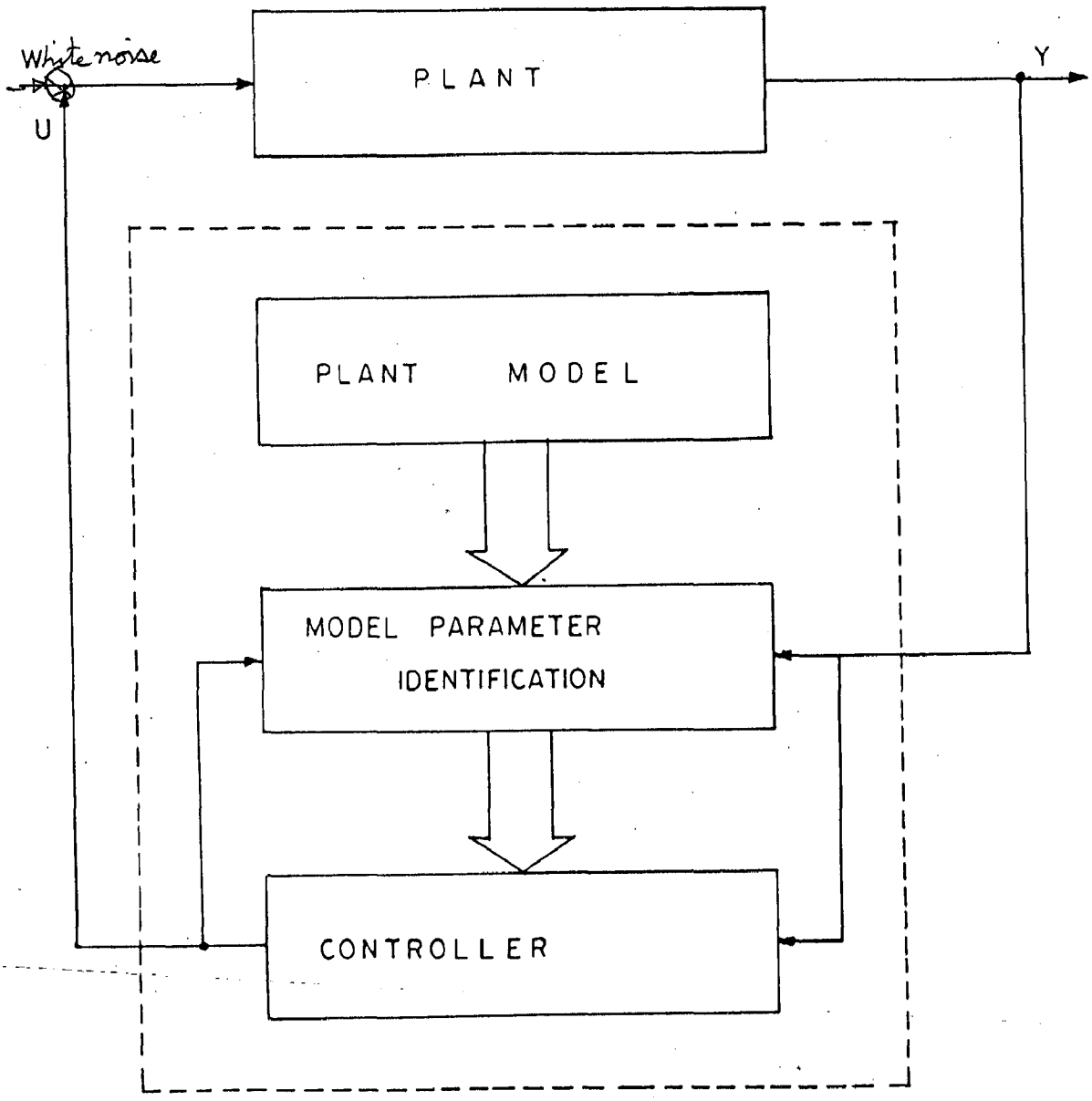


FIG.2.1 BLOCK DIAGRAM OF A SELF TUNING CONTROLLER.

where

$$\theta(t) = [h_o, h_1 \cdots h_{n_H}, g_o, g_1 \cdots g_{n_G}, e_o, e_1 \cdots e_{n_E}]$$

h, g and e are the co-efficients of the general polynomials $H(z^{-1})$, $G(z^{-1})$ and $E(z^{-1})$ of order n_H, n_G and n_E respectively.

The data vector is,

$$x(t) = [u(t), u(t-1)\ldots, y(t), y(t-1), \ldots, w(t), w(t-1), \ldots\ldots]$$

P(t) is a matrix proportional to the covariance of estimated parameters.

k(t) is the updating gain matrix, and is forgetting factor for exponential weighting of last data values.

iii) Compute Control Form

$$u(t) = -[g_o y(t) + g_1 y(t) + \ldots + e_o w(t) + e_1 w(t-1) + \ldots \ldots + h_1 u(t-1) + \ldots] / h_o$$

iv) Repeat from (i) for the incremented value of 't'.

The algorithm may be started with P_o of polynomial P set equal to unity and the rest of the co-efficients of R and Q polynomials set to any arbitrary values. Generally 'R' is equal to 'P'.

2.8 CONCLUSIONS

In this chapter, the selection of system model, test

signal, system parameter identification techniques and briefly about MRAC & STC strategies have been described. The reasons to use STC instead of MRAC are also discussed. The complete algorithm of selftuning control, used in this work, is also explained.

CHAPTER- 3

HARDWARE IMPLEMENTATION

3.1 GENERAL

The complete scheme for adaptive speed cotrol of the seperately excited d.c. motor is discussed in this chapter. The power converter used is a three phase fully controlled bridge converter. A adaptive controller based on selftuning control is realised on IBM personal computer. Microprocessor based scheme for its firing angle control is discussed in detail, together with the hardware requirements. An 8085 microprocessor based system is used for the current error processing by a software proportional plus integral controller (PI) and personal computer (PC) is used to process the software of self tuning controller (STC) which helps in controlling the speed of a d.c. motor.

3.2 SYSTEM DESCRIPTION

The complete block diagram of the system is shown in Figure 3.1. The system implementation is started with the reference speed signal and the speed of the machine has to follow this signal. The controller (STC) parameters are computed in real time on line with IBM personal computer which are essential to calculate the controller output,

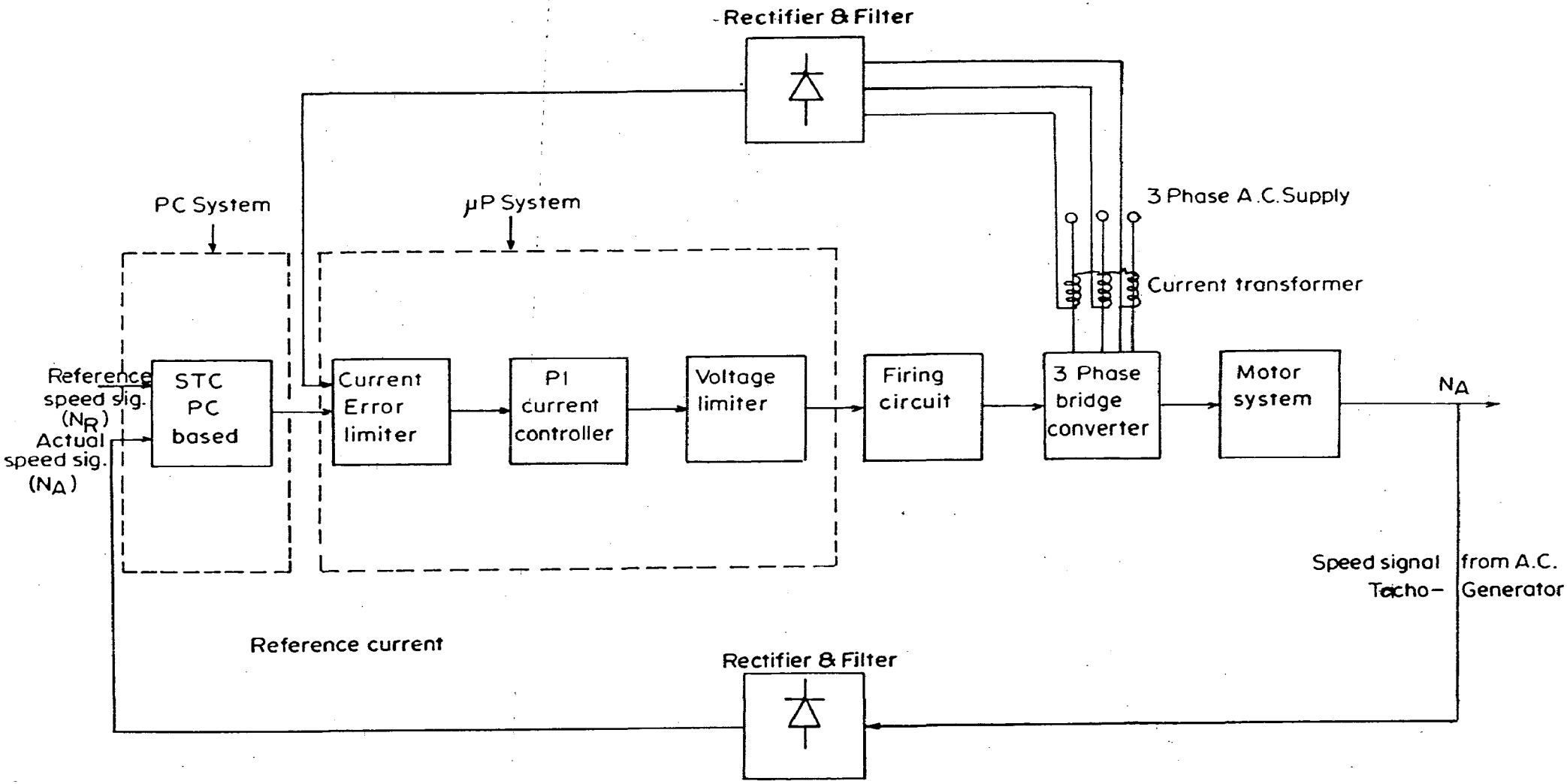


FIG.3.1 SYSTEM BLOCK DIAGRAM.

obtained from personal computer (P.C.) in analog form through DAC is considered as current reference signal (I^*_{dc}) and it is converted into digital signal through 8 bit ADC 0809 which is being fed to the microcomputer 8085. The actual current signal corresponding to motor current is also fed to another channel of ADC 0809. The current error is generated by comparing the reference and actual current signals. This error may be either positive or negative and it is limited to a corresponding to about $\pm$25% of full load current of the motor.

The controlling signal is generated by a microprocessor according to the current error using proportional plus integral (PI) control algorithm. The PI controller processing is carried out through microprocessor software. The control algorithm implemented in the microprocessor is a simple computation rule so that the response of the system can be fast. The PI control algorithm is easy to implement and does not need too much time to compute the controlling signals. Depending on this signal, constrained value is obtained from which corresponding firing angle is selected from the lookup algorithm. Using this firing angle first pair of the thyristors are fired and the remaining five pairs are fired at 60^o interval. The converter output voltage is fed to a d.c. motor system. The speed of the machine being adjusted through self tuning controller in such a way that irrespective of the load

condition the actual speed of the machine has to follow the reference speed signal. The design of PI controller is given in Appendix C.

3.3 PC BASED SELF TUNING CONTROLLER

Clarke and Gawthrop algorithm is used to develop the selftuning controller and it is realised through software on IBM personal computer (PC). Reference speed signal is set through a potential divider and is being fed to PC via one channel of 12 bit ADC. Actual motor speed signal is fed to PC via second channel of ADC. Depending on the difference between these two values, the controller parameters are generated which are essential in obtaining the controller output (u). This controller output is outputted through a channel of DAC which has been considered as the reference current signal to the microprocessor system for the processing of PI current controller

3.4 POWER CIRCUIT AND ITS WORKING [1,30,31]

For the experimental purpose, a 1 H.P., 220 V, 1500 RPM d.c. machine is used. The fullwave thyristor bridge rectifier is employed to feed the proposed machine. The ratio of PIV and line to neutral voltage is given by $E_{PIV}/E_{LN} = 2.45$. Therefore

$$E_{PIV} = 2.45 \times \frac{400}{\sqrt{3}} = 566 \text{ V}$$

Taking a factor of safety 2, a 1200 PIV Thyristors are used for the rectifier.

The output average d.c. voltage of a full wave rectifier is given by $V_{dc} = 3\sqrt{6}/\pi \cos\alpha$. The output voltage is taken equal to the rated voltage of the machine corresponding to minimum firing angle i e. $V_{dc} = 220$ V. Assuming the efficiency of 70% of d.c motor the input power (P/n) is as

$$P_I = V_{dc} I_{dc} = P/n = 746/0.7 = 1065 \text{ Watts}$$

$$I_{dc} = \frac{1065}{220} = 4.85 \approx 5 \text{ Amps.}$$

Taking a factor of safety 2 the thyristor current rating $I_T = 2x5 = 10$ Amps

So the thyristor of 12 Amp, 1200 V can be selected to meet the requirements. However the available thyristors of 16 A and 1200 PIV are used.

Figure 3·2 shows a three-phase fully controlled converter. It consists of six thyristors connected in the form of bridge configuration. These thyristors are connected to phases RYB will be denoted by (T_1, T_4) (T_3, T_6) and (T_5, T_2). Each thyristor conducts for 1/3 of a cycle i.e. for 120°

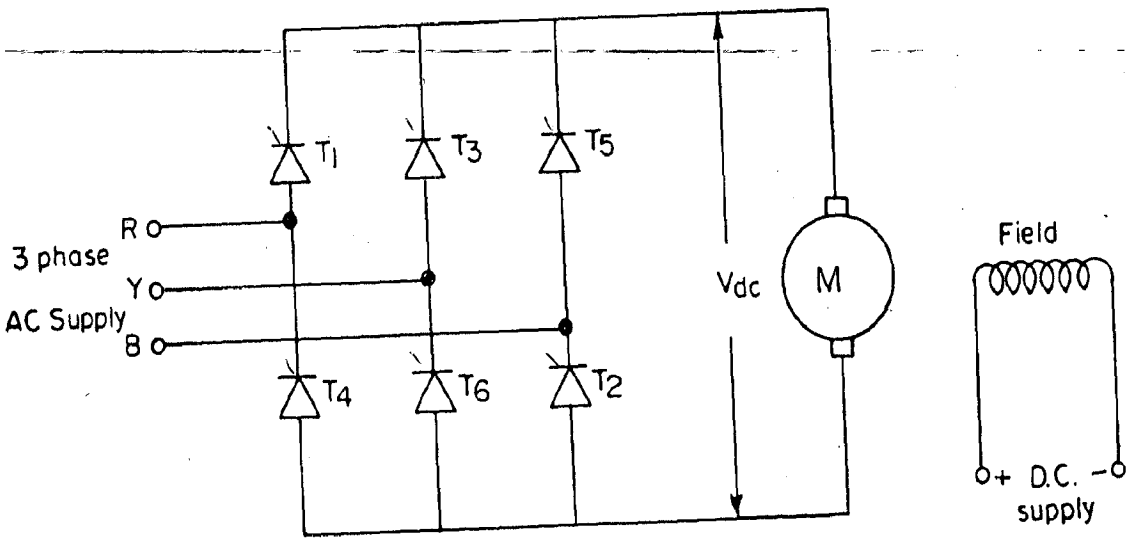


FIG.3.2 POWER CIRCUIT CONFIGURATION.

duration, and at any instant of time two thyristors are conducting. With respect to phase sequence RYB, T_1 commutates with T_3, T_3 with T_5 and T_5 with T_1. On the other side T_6 commutates with T_2, T_2 with T_4 and T_4 with T_6. Commutation from one thyristor to other occurs when two anodes pass through the same potential, one going up takes over from the other going down. This is called natural commutation. Thyristors are switched faster in three phase converters and therefore the time available for any current delay is less compared to the single phase case. This tends to make the motor current continuous.

Figure 3.2(a) shows the input supply line voltages and 3.2 (b) shows at which instants, the particular pair of thyristors are conducting. In this work six sequence of operation is assumed which can be understand from the following table.

Sequence	0	1	2	3	4	5
Conducting SCR's	(6,1)	(1,2)	(2,3)	(3,4)	(4,5)	(5,6)
SCR's to be fired	1	2	3	4	5	6
Outgoing SCR	6	1	2	3	4	5

The above firing sequence is used when the firing angle is less than 60°. If the firing angle becomes greater than 60° then the following sequence is used.

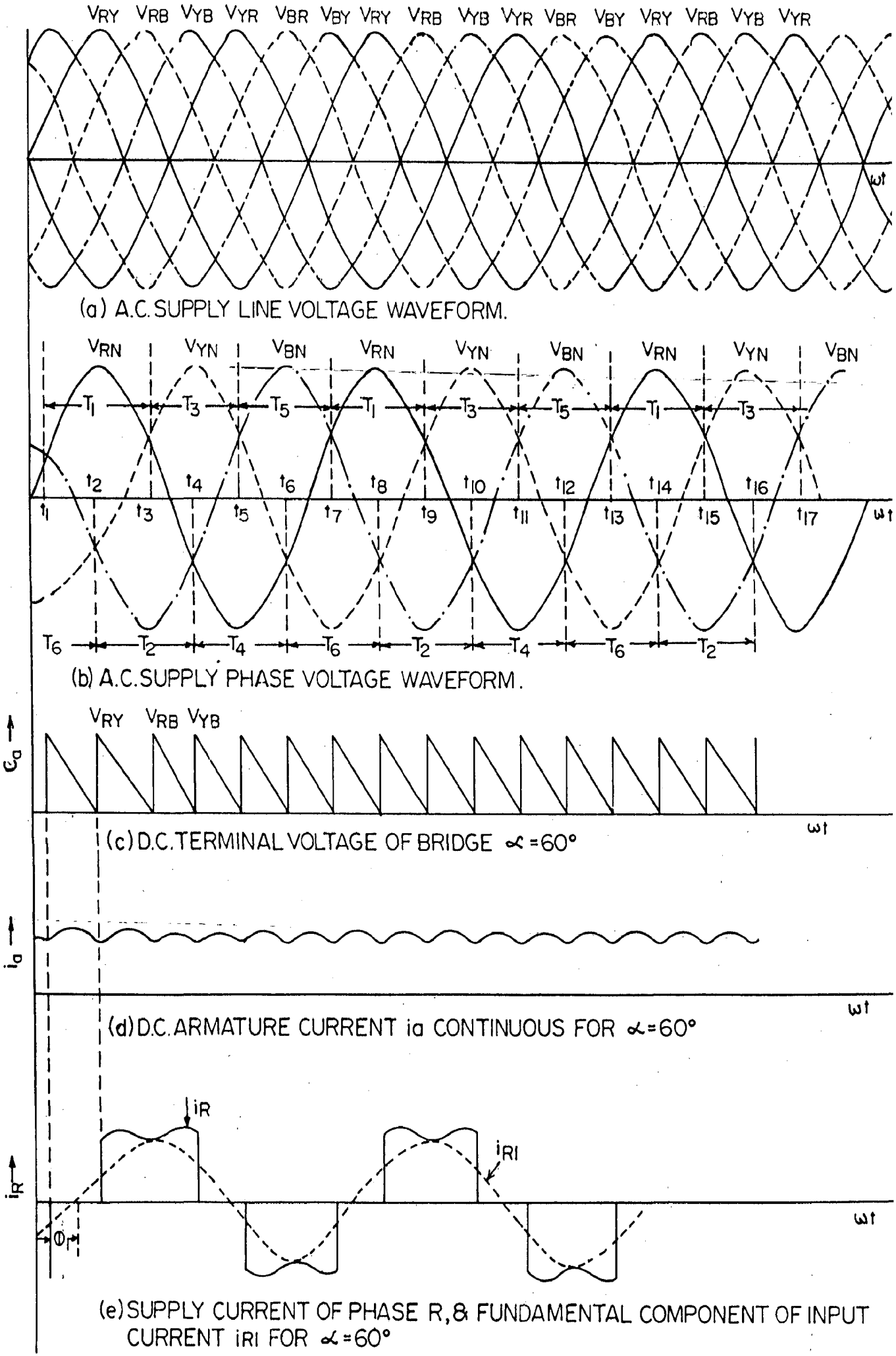

VRY VRB VYB VYR VBR VBY VRY VRB VYB VYR VBR VBY VRY VRB VYB VYR
ωt
(a) A.C. SUPPLY LINE VOLTAGE WAVEFORM.
VRN VYN VBN VRN VYN VBN VRN VYN VBN
T1 T3 T5 T1 T3 T5 T1 T3
T6 T2 T4 T6 T2 T4 T6 T2
t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12 t13 t14 t15 t16 t17
ωt
(b) A.C. SUPPLY PHASE VOLTAGE WAVEFORM.
ea
VRY VRB VYB
ωt
(c) D.C. TERMINAL VOLTAGE OF BRIDGE α=60°
ia
ωt
(d) D.C. ARMATURE CURRENT ia CONTINUOUS FOR α=60°
iR
iR1
ωt
Φ1
(e) SUPPLY CURRENT OF PHASE R, & FUNDAMENTAL COMPONENT OF INPUT CURRENT iR1 FOR α=60°

FIG. 3.3

Sequence	0	1	2	3	4	5
Conducting SCR's	(5,6)	(6,1)	(1,2)	(2,3)	(3,4)	(4,5)
SCR's to be fired	6	1	2	3	4	5
Outgoing SCR	5	6	1	2	3	4

If the firing angle becomes greater than 120° the following sequence is used

Sequence	0	1	2	3	4	5
Conducting SCR's	(4,5)	(5,6)	(6,1)	(1,2)	(2,3)	(3,4)
SCR's to be fired	5	6	1	2	3	4
Outgoing SCR	4	5	6	1	2	3

To maintain a symmetrical waveform of the input current the firing sequences are necessary. The frequency of firing will be six times the supply frequency.

The average output voltage of a fully controlled converter is given by

$$V_{dc} = \frac{3\sqrt{6}}{\pi} V \cos\alpha \qquad .. (3.1)$$

If $\alpha = 0^{\circ}$ then $V_{dc} = V_{max}$ and if $\alpha = 90^{\circ}$ then $V_{dc} = 0$.

Therefore equation (3.1) shows the output voltage depends upon the firing angle information.

For a seperately excited d.c. motor the back e.m.f.

$E_g = K_a \phi N$

In terms of average voltages

$$V_{dc} = I_a R_a + E_g = I_a R_a + K_a \phi N \; ; \quad N = \frac{V_{dc} - I_a R_a}{K_a \phi} \qquad .. (3.2)$$

Equation (3.2) represents the average speed.

In a seperately excited d.c. motor $T = K_a \phi I_a$ and motor speed (N) is

$$N = \frac{V_{dc}}{K_a \phi} - \frac{R_a}{(K_a \phi)^2} T \qquad .. (3.3)$$

The first term in equation (3.3) represents the ideal no load speed ($T \approx 0$), which, therefore, depends on V_{dc}. If the motor current is assumed continuous, motor terminal voltage V_{dc} depends only on the firing angle α. The variation of V_{dc} with α for continuous motor current is shown in Fig.3.3(c). These curves also represent the theoritical no load speed as a function of firing angle. The second term in equation (3.3) represents the decrease in speed as the motor torque increases. Since the armature resistance is small, the decrease in speed is small. In large motors, the motor current at no load (i.e. motor idling) is not small, and if a three phase converter is used the motor current is likely to be continuous even at no load condition. Therefore, three phase converter fed dc drives provide better speed regulation and improved performance as compared to single phase converter fed dc drives.

The motor terminal voltage can become negative for the triggering angle $=120^{o}$. This is the inversion mode of operation of the converter. If the motor voltage is reversed

with a reversing contactor or by reversing the field current, power can be transferred from the motor to the a.c. supply. That is what commonly known as regeneration. The motor will slow dcwn due to power feeding back, and thus the motor voltage decreases. Therefore, as the motor slows down, the firing angle of the converter is to be adjusted to keep the current up and to regenerate power.

3.5 MICROPROCESSOR BASED CONTROL SCHEME

Microprocessor based firing scheme is shown in Fig.3.4 and it is used for generating the firing pulses for converter thyristor in proper sequence decided by the control circuit and software. This scheme has several advantages over the analog firing control schemes. This scheme requires less number of hardware components and simple synchronizing technique. For this, one programmable peripheral interface 8255, one programmable interval timer and one programmable interrupt controller 8259. The pin diagram and operational details of these ICs are given in Appendices A and B . The synchronizing signal which is obtained from synchronizing circuit is inputted to IR_o interrupt. Gate of timer TM_1 is always made high by connecting it to +5V. Counter $TM_1(1)$ get the clock of 1.535 MHZ from µP. ADC 0809 is interfaced to microprocessor through port B and port C of 8255. ADC provides the information about the reference current and actual current. The firing angle information is obtained

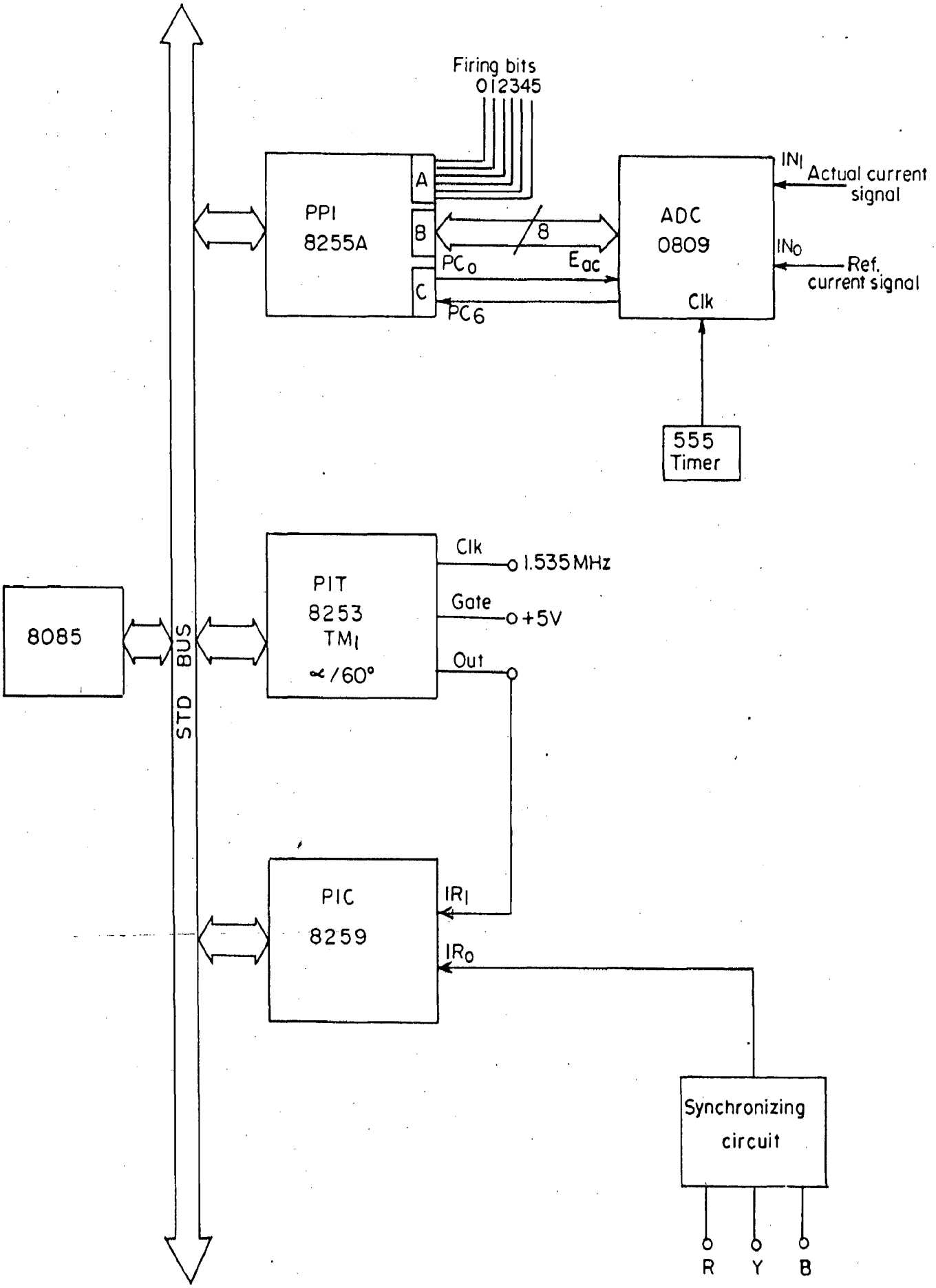


FIG. 3.4 MICROPROCESSOR BASED SYSTEM CONFIGURATION.

from the lookup table, and the pulses are outputted through port A of 8255 as per firing command. Different wave shapes for interrupt signals are given in Figure 3.5.

3.5.1 Synchronizing Technique

Independently of the method or algorithm used to control thyristors firing of a three phase static power converter, the signals that synchronize them with the alternating supply voltages are always necessary. For a three phase converter, there are six such logical signals, each one associated with a thyristor and defined from the natural commutation instant i.e. at the intersection of two phase to phase voltages, which coincides with zero crossing of the other one. The zero crossing voltage detector fulfills two different functions, the synchronization of the thyristor circuit to the input line frequency and the phase sequence identification.

Synchronizing signal is generated by sensing line to line voltages V_{RB} and V_{BY} of the supply mains. This signal has pulses of 60° duration corresponding to supply voltage frequency at every 360° interval. This signal provides the information about frequency of supply voltage and the instant of loading a counter of PIT 8253 with firing angle delay count. Synchronizing signal is required for generating firing pulses for converter thyristor in proper sequence.

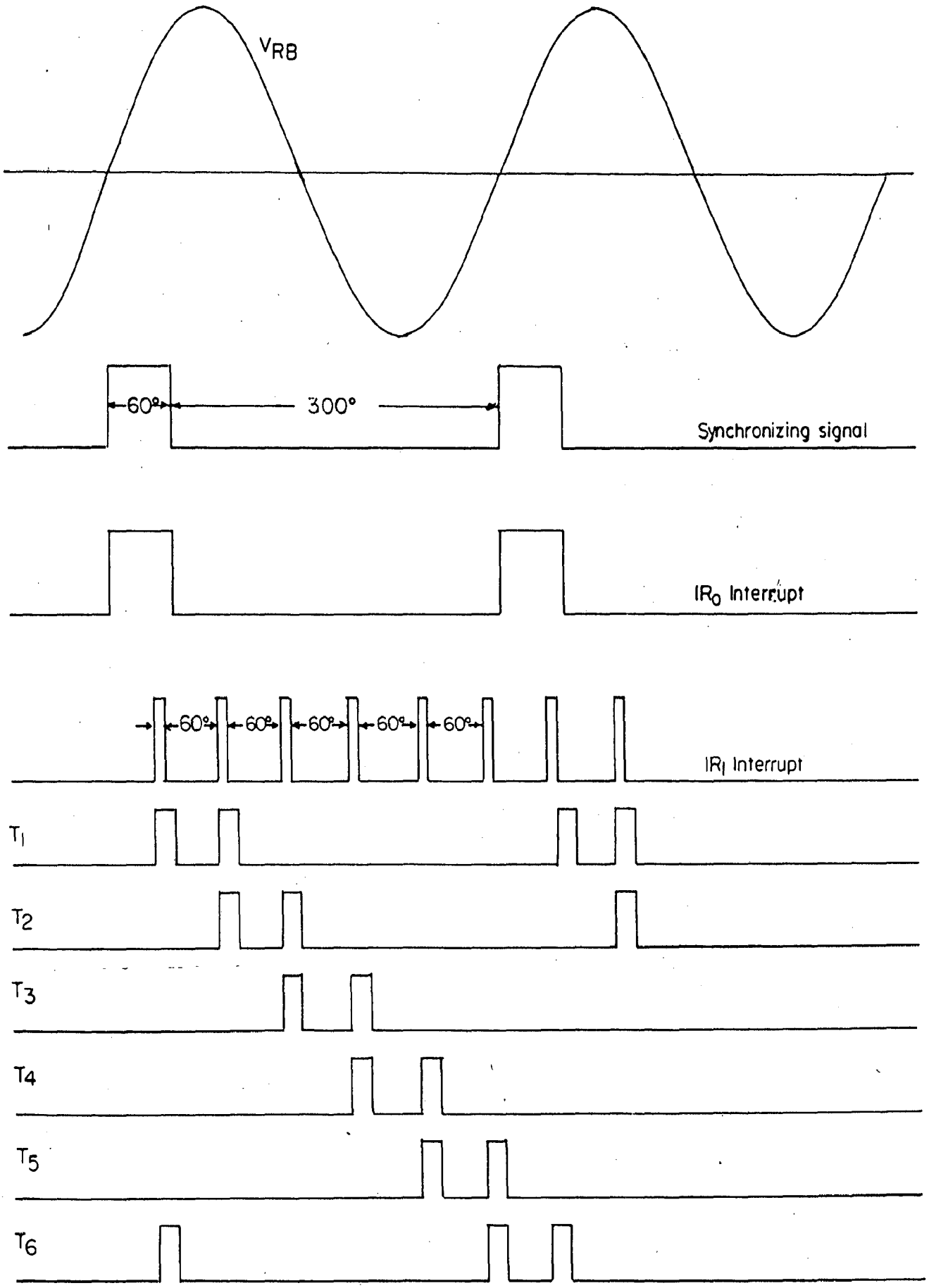


FIG.3.5 SYNCHRONIZING SIGNAL, IR_0 & IR_1 INTERRUPT SIGNALS & FIRING PULSES IN PROPER SEQUENCE.

Synchronizing circuit uses less number of components and is shown in Figure 3.6. This circuit consists of the following components.

i) Step down transformer

ii) Comparator

iii) Transistor drive

iv) AND gate

Stepdown transformer is a single phase 440 V/6 V transformer to realise voltage sensor for a wide range of frequency. The primary and secondary connections are made in such a way that secondary voltage is 180^{o} out of phase with respect to the primary voltage. Comparator circuit is realised with the help of operational amplifier IC 741. The inverting terminal of IC 741 is connected to common ground. The output of the comparator ideally swings between +12V to -12V at zero crossing of supply voltages and frequency is same as supply frequency. The output of comparator is reduced to a lower voltage by a series connected resistor and applied to the base of the transistor. Transistor stage is very important and required to interface the analog system to a digital system. Transistor is used to control the state of the TTL input. TTL outputs of transistors are shown in Figure 3.7.

The diode in the base circuit protects the base against excessive reverse voltage. Since the negative output

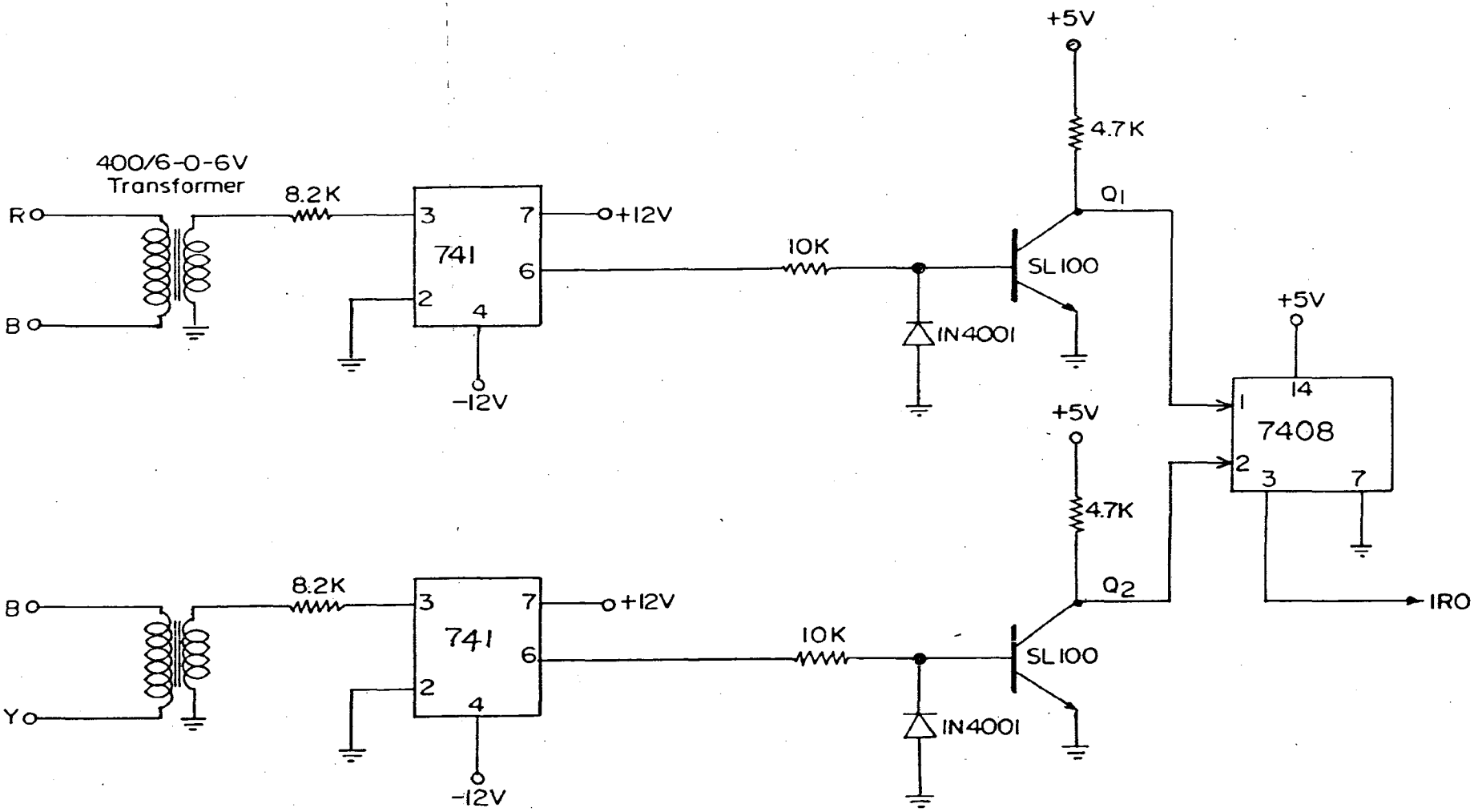


FIG.3.6 SYNCHRONIZING CIRCUIT.

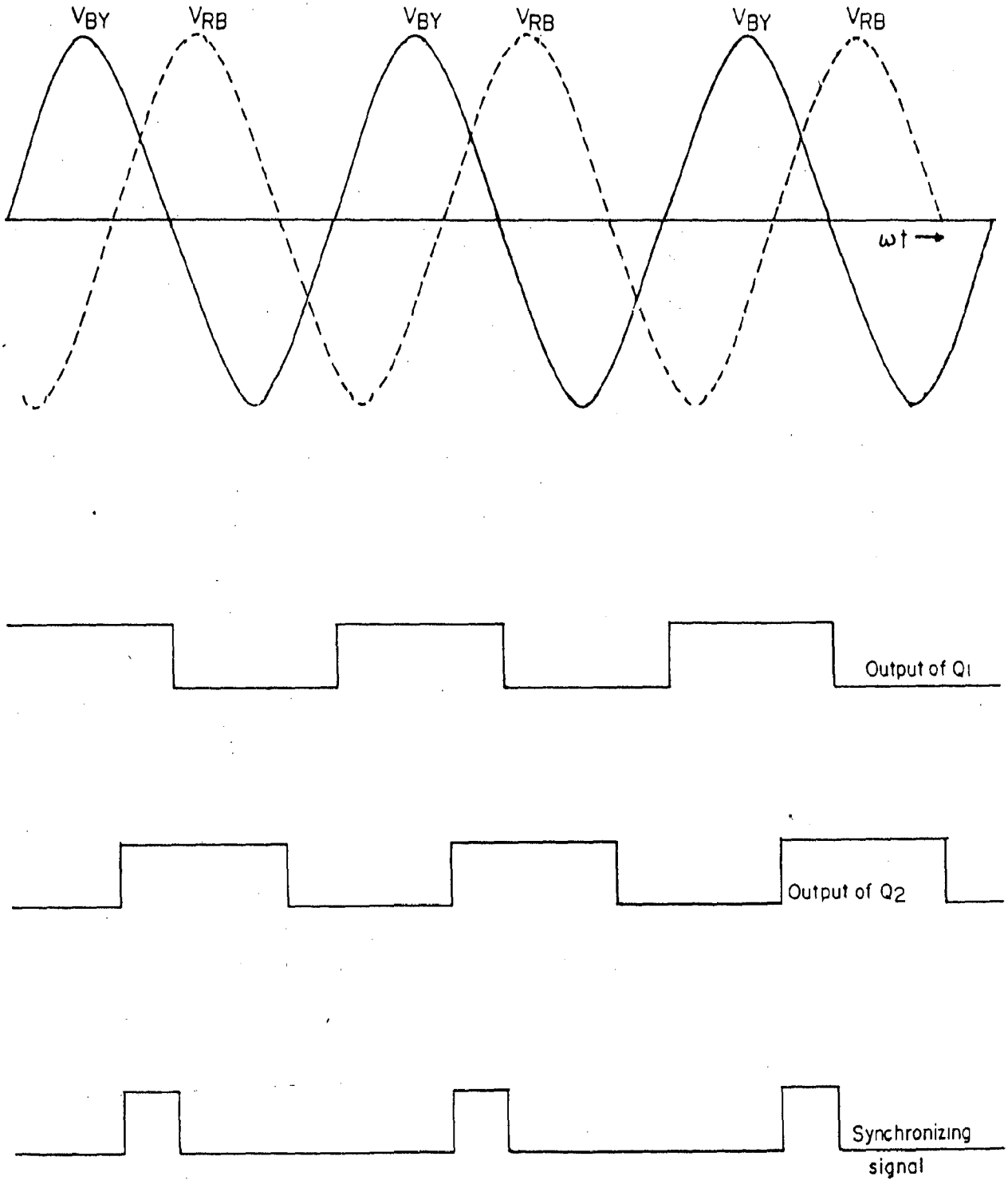


FIG.3.7 VOLTAGE WAVEFORM AT DIFFERENT POINTS OF SYNCHRONIZING SIGNAL.

of the comparator approaches -12V, we need to use a protective diode between the base and ground. This diode clamps the base voltage at approximately -0.7V on the negative swing. AND gate is used for necessary AND operation. The synchronizing signal is obtained after ANDING two output signals of the transistors. The microcomputer makes use of this signal for generating firing pulses in proper sequence and firing angle control.

3.5.2 Pulse Amplifier Circuit

The duration of firing pulses generated by the microprocessor via port A is not sufficient to trigger converter thyristors. The required pulse width is obtained using monostable circuit. The output of the monostable is applied to the pulse amplifier circuit as shown in Figure 3.8.

A pulse transformer is used for electrical isolation between control circuit and power circuit. The pulse at the transistor collector is the amplified inverted form of the input signal. But the isolating transformer connection are made in such a way that the final firing pulses has no phase shift with respect to the input pulse. A diode is connected across the transformer primary to avoid saturation of the pulse transformer. Another diode is connected in series with the secondary of the pulse transformer to block negative pulses. The diode across the primary also serves to protect

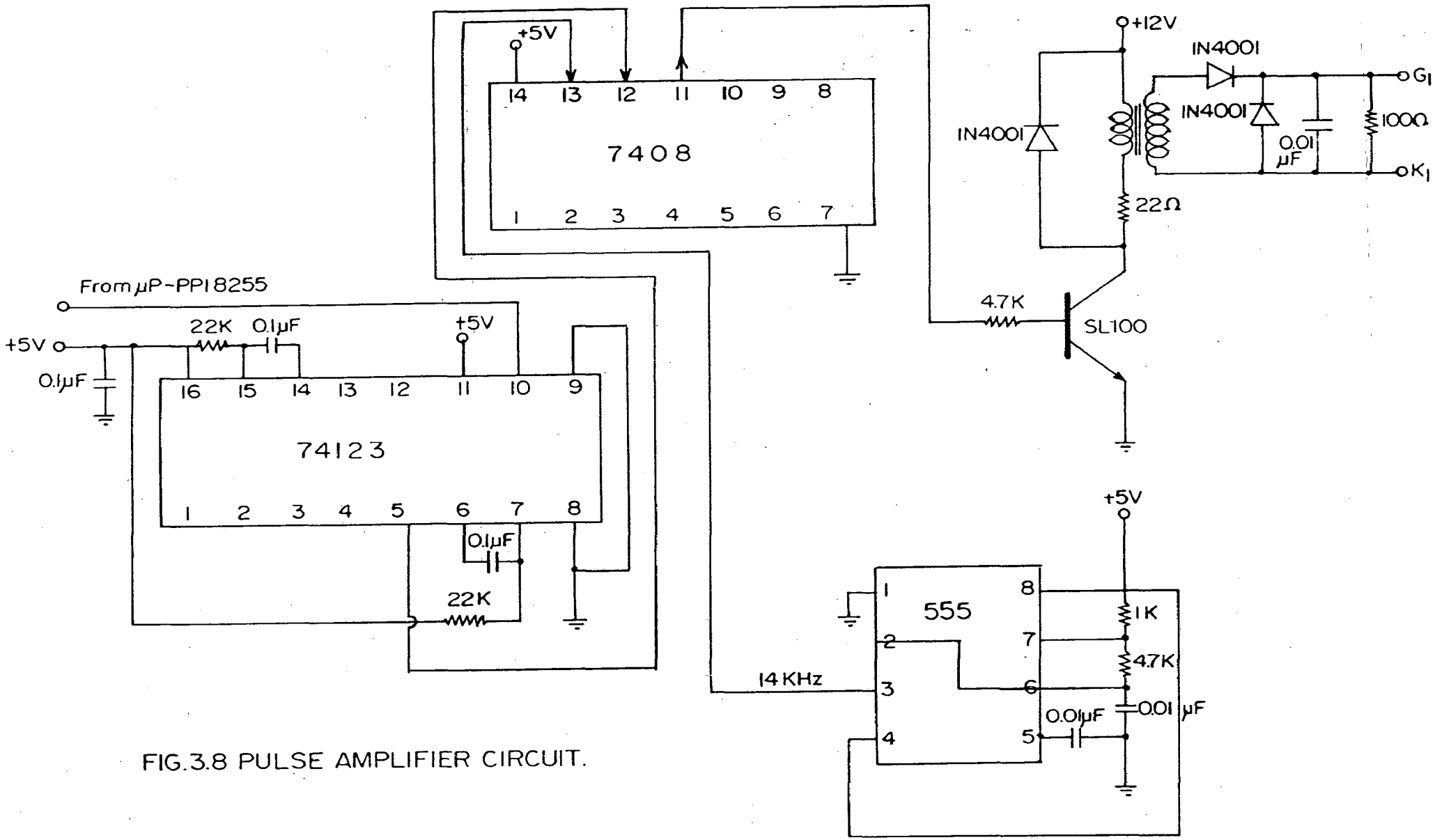


FIG.3.8 PULSE AMPLIFIER CIRCUIT.

the transistor. The capacitor at the secondary prevents any spurious high frequency from triggering the thyristor. The gate of the thyristor can be protected from long reverse voltage by connecting a diode across the gate as shown in Figure 3.8.

3.6 CONCLUSIONS

In this chapter, the complete hardware philosophy of the speed control of a d.c. motor has been described using self tuning controller. The STC adaptive controller for speed control loop has been realised on IBM PC system and current control loop using PI controller has been realised through 8085 microprocessor based system. The design of power circuit and synchronizing circuit has been developed. The details of microcomputer based firing control scheme has been described in detail.

CHAPTER - 4

SYSTEM SOFTWARE IMPLEMENTATION AND FLOW CHARTS

4.1 GENERAL

In this chapter, the complete software for self tuning controller (STC) based closed loop speed control of thyristor converter fed d.c. motor is described. The self tuning controller is realised through IBM personal computer (PC) in which speed feedback and reference speed signals are inputted through 12 bit 16 channel ADC Card and output of controller (STC) is obtained through a DAC channel of same card from personal computer. The analog output of self tuning controller is considered as current reference signal and inputted to 8085 based microcomputer system via 8 bit 8 channel ADC 0809 along with motor current feedback signal. A current proportional integral (PI) controller is realized through software on 8085 based system which also provides various firing commands for thyristors of converter in proper synchronization with mains through various hardware interrupts. The complete software for microprocessor 8085 based system is explained through main programme and associated various subroutines.

4.2 SOFTWARE IMPLEMENTATION OF SELF TUNING CONTROLLER

In the present work, the speed control of a d.c. motor is achieved by self tuning controller. The flowchart of

implementation of STC algorithm is given in Fig.4.1. The program starts with the initialization of variables which are necessary to carryout the processing in order to obtain controller parameters and controller output. Proper weightage of polynomials P,Q and R is considered. Reference speed is set through the potential divider, and actual speed signal of a d.c. motor is measured through a.c. tachogenerator after rectifying and filtering. These two signals are taken as input via AD converter channels (I_o and I_1) of ADC card of IBM (PC). Depending on these signals the controller parameters are estimated which adjust the controller output (U). The value of 'U' is computed and limited in the manner as demanded by controller to maintain constant speed of a d.c. motor irrespective of the load conditions. The developed basic language program for self tuning controller is given in appendix 'D'.

4.3 SOFTWARE FOR 8085 MICROPROCESSOR SYSTEM

4.3.1 Main Program

The flow chart of the main program is shown in Fig.4.2. The program is started with initialization of STACK pointer and input-output ports. In the 8085 system, two timers of PIT 8253, six input-output ports with two PPI 8255 and one interrupt controller PIC 8259 are available to the user at J_1,J_2 and J_3 space. In the present work, one timer TM_1 is used for loading firing angle α as well as 60^o count. Out of

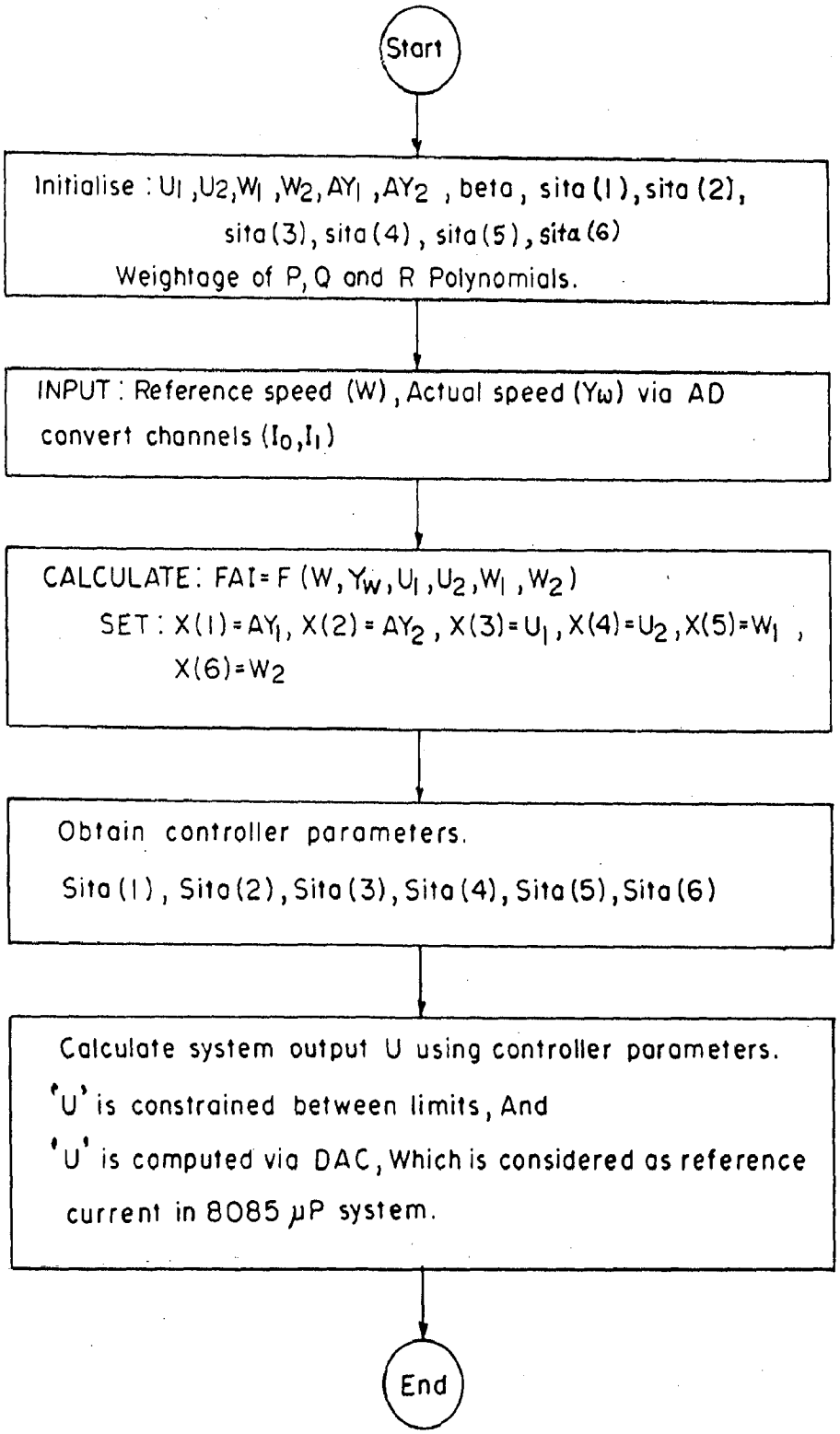


FIG.4.1 FLOW CHART OF SELF TUNING CONTROLLER.

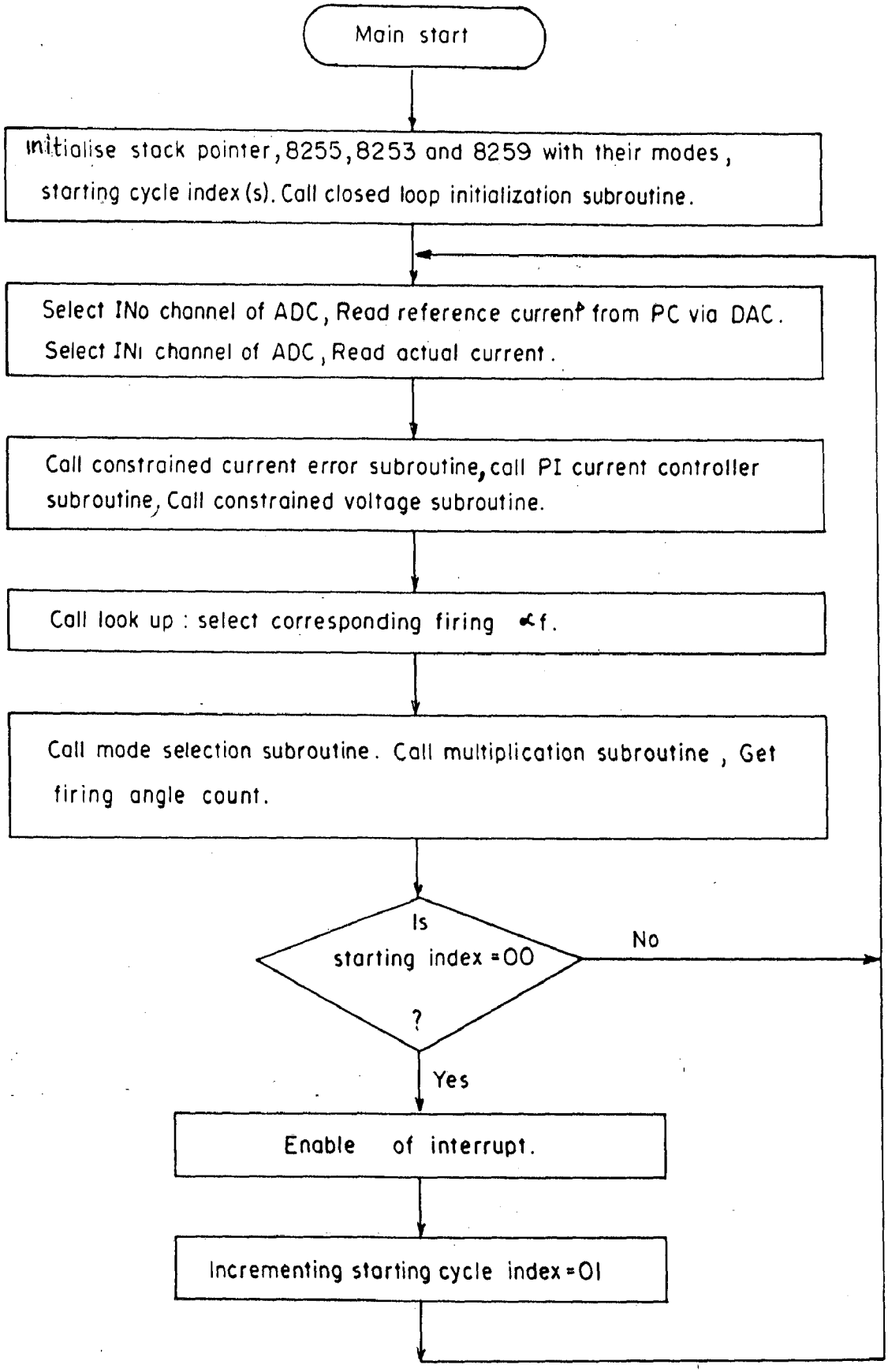


FIG.4.2 FLOW CHART OF MAIN PROGRAM.

six input-output ports, three input-output ports of PPI 8255(1) are used in the present work. It is initialized as follows:

8255(1) (Control word registor = 03H)

PORT A	PORT B	PORT C_U	PORT C_L	
OUT	IN	IN	OUT	10001010 = 8A

Figure 3.4 given in the previous chapter shows the configuration of the microprocessor based control scheme. Port A is used for firing the converter thyristors (Bits PA_0 to PA_5), port B is used for inputting digital output of ADC. Port C is used to send start of convertion (SOC) and end of conversion (EOC) signals. The timer TM_1 is used for loading both firing angle count as well as 60^o count for triggering converter thyristors. The timer is initialised as follows.

TM_1 (11H) in mode '0'

TM_1	R/L	MODE 0	CODE
01	11	000	0 = 70H.

After initialization of all the necessary chips, all firing bits are made low and the status of output ports are stored. Starting cycle index 'S' is stored with initial value as OOH and the 60^o count value is also stored. The PIC is initialized for the vector addresses of the interrupts IR_0, IR_1. All interrupts are considered edge triggered.

After initialization process, two ADC channels are inputted one by one by selecting channels through port C_L. Reference current is inputted by selecting IN_0 channel of ADC 0809 from personal computer (PC) via DAC, actual current is inputted by selecting IN_1 channel of ADC 0809.

After the measurement of reference and actual currents, the current error is constrained and stored at the end of constrained current error subroutine. Now, it enters PI subroutine.

In the PI control part, the change in error is calculated from the constrained current error and previous constrained current error. This change in error is used to obtain the new output unconstrained voltage of 2 byte, and then program enters constrained voltage subroutine.

In the constrained voltage part, previously obtained unconstrained voltage of 2 byte is considered to one byte only, because the voltage values stored are of one byte. This value of constrained voltage is limited between minimum (00H) to maximum (FFH) output voltage (Vdn), then it enters lookup algorithm to select the corresponding firing angle.

The firing angle which has been selected through look up subroutine is used to select the particular sequence of firing the thyristors of converter. This is being done in 'MODE SELECTION' subroutine. The above obtained firing angle

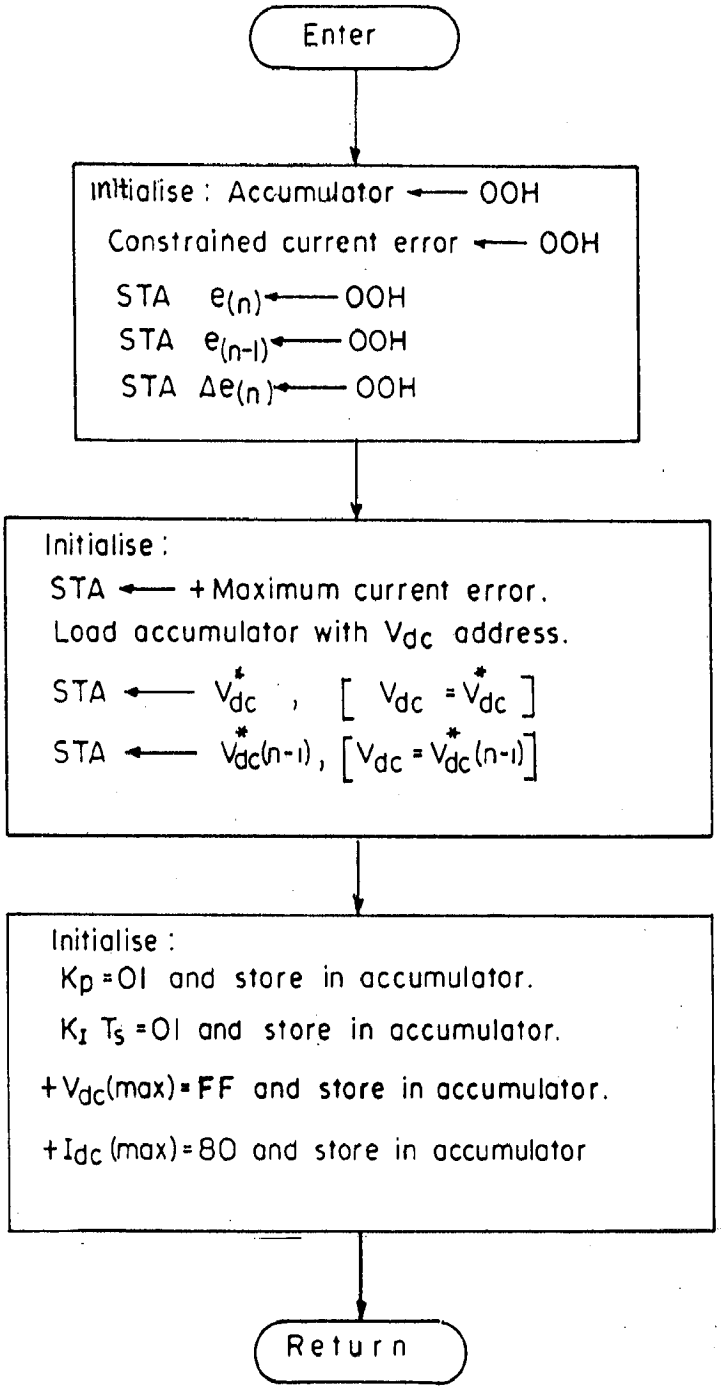


FIG.4.3 FLOW CHART OF CLOSED LOOP INITIALIZATION.

is converted into its equivalent counts by multiplying it with 'K' counts using 'MULT' subroutine. This firing angle count is used to load the timer (TM_1).

After this starting cycle index 'S' is checked. If it is found as zero then the interrupt are enabled and increment index 'S' to one, otherwise jump back to 'CALL ADC' and carryout other processings.

The IR_0 interrupt service susbroutine is used to load the timer with firing angle count once in every cycle and IR_1 interrupt service subroutine is used to load the timer with 60° count five times in every cycle, and to issue all six firing commands.

The machine level language programs are given in Appendix E.

4.3.2 Close Loop Initialization

In order to carryout the PI processing, it is necessary to initialise some of the variables such as error, change in error, maximum voltage, maximum current, controller constant etc. This is done using closed loop initialization subroutine. The self explainatory of which is shown in Figure 4.3.

4.3.3 ADC Subroutine

Reference current is measured and stored through this

subroutine. The self tuning controller output from PC via DAC converter is considered the value of reference current. An ADC converts this analog signal into digital signal. This ADC conversion is done once in every cycle of the main programme.

ADC clock in is given from a timer of 380 KHz frequency. ADC starts converting the analog signal into digital signal at the instant, when the start conversion (SOC) bit of ADC goes from low to high to low. ADC sends a signal to I/O port through end of conversion (EOC) bit going high, indicating that the conversion process is over. The ADC count is measured and stored in the memory location alloted for reference current. The flow diagram of ADC subroutine is shown in Fig. 4.4.

NAME OF SUBROUTINE	: ADC SUBROUTINE
INPUTS	: Input to DAC through AD/DA converter from PC via channel IN_o which is selected.
OUTPUTS	: Reference current I_{dc}
CALLS	: None
DESTROYS	: A, F
DESCRIPTION	: This subroutine measures the reference current I^*_{dc}. The result is stored in the reference current I^*_{dc} at the end of subroutine, and it is displayed in the address field.

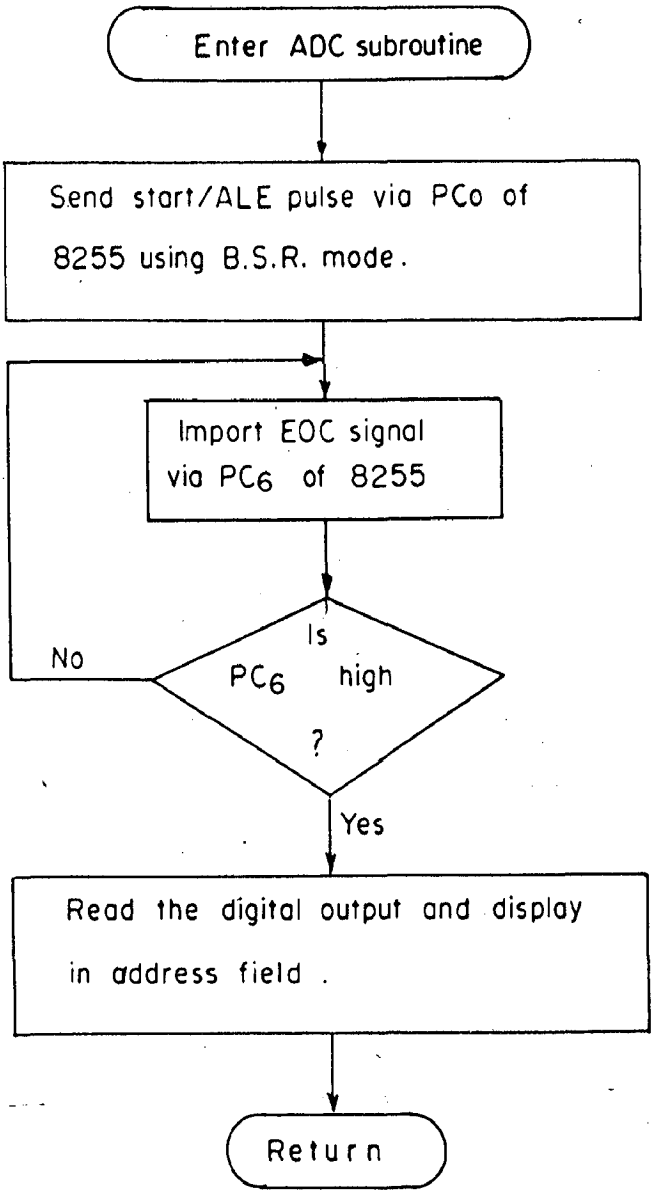


FIG.4.4 FLOW CHART FOR ADC SUBROUTINE.

4.3.4 Constrained Current Error Subroutine

This subroutine is used to limit the magnitude of current error corresponding to actual current and reference current. The maximum magnitude of current error is decided in the CLI initialization routine. Difference between reference current and actual current results in error, this error is being limited not to cross the specified limits. The error may be positive or negative, if the magnitude of the current error is greater than the maximum current error, then the error is set equal to maximum current error, if the magnitude of -ve current error is greater than the maximum current error then error is set equal to maximum value and 2's complement of it. This value is the constrained value of current error. The flow chart of this subroutine is given in Fig. 4.5.

NAME OF SUBROUTINE	: Constrained Current Error Subroutine.	
INPUTS	: Reference Current I^{*}_{dc} Actual Current I_{dc}	1 byte each
OUTPUT	: Constrained Current Error (1 byte 2's Complement)	
DESTROYS	: A, F, H, L.	
DESCRIPTION	: This subroutine limits the current error between minimum and maximum error.	

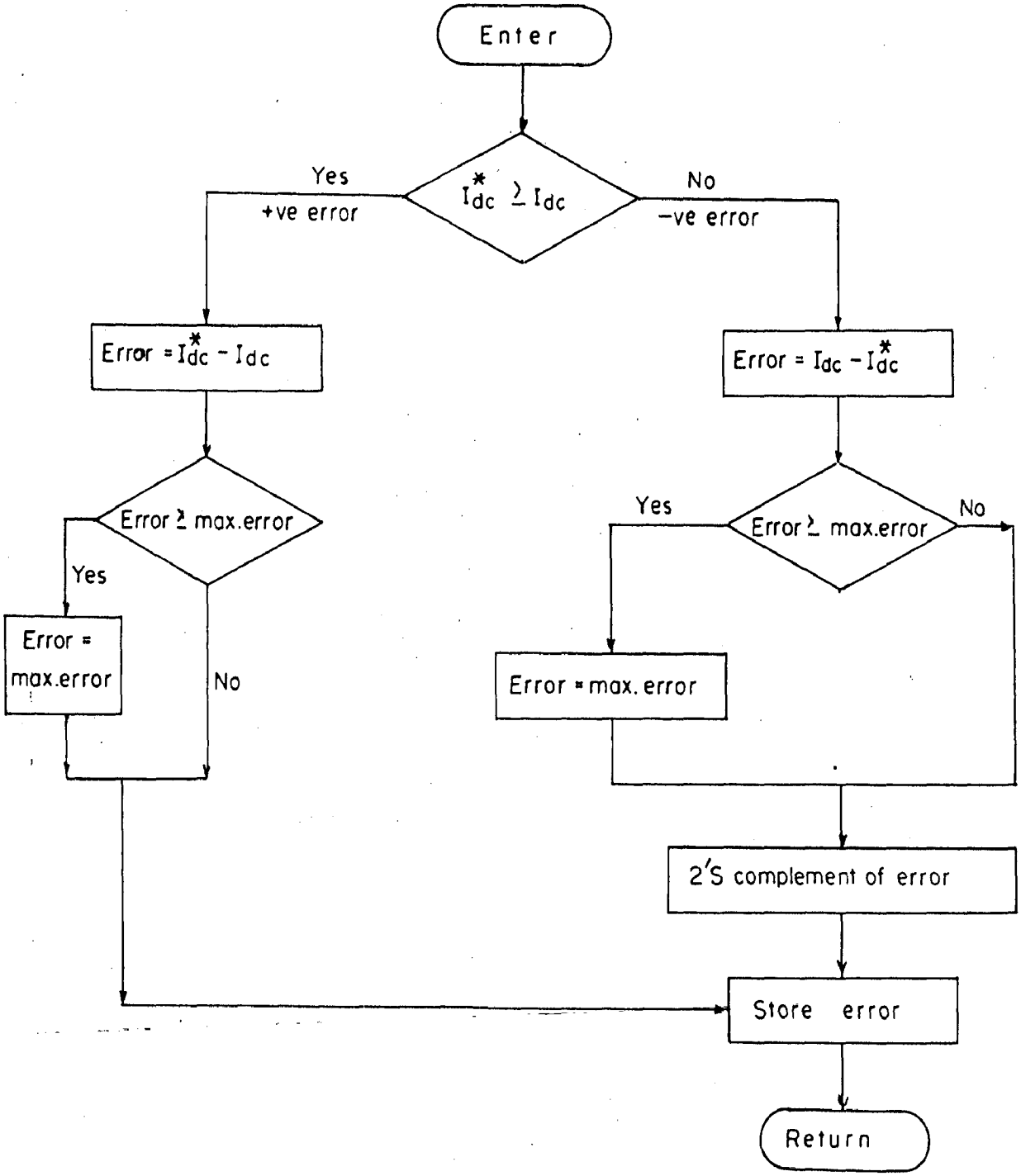


FIG.4.5 FLOW CHART OF CONSTRAINED CURRENT ERROR SUBROUTINE.

4.3.5. PI Current Controller Subroutine

This subroutine computes and stores the unconstrained voltage V^*_{dn} . During the execution of this subroutine, voltage and current error is updated. The constrained current error is considered as present status of error, which is used along with integral gain to obtain $K_I e_n$, the change in error is used with proportional gain $K_p \Delta e_n$. In both these case 'SMULT' subroutine is used, resulting in partial product. The product being added to get change in voltage. This change in voltage is added with previous voltage to obtain present status of voltage which is unconstrained 2-byte in 2's complement and is stored in the alloted memory location. The flow chart of it is given in Fig.4.6.

NAME OF SUBROUTINE	: PI Subroutine
INPUTS	: V^*_{dn}] one byte each : V^*_{dn-1}]
	: e_n [1 byte 2's complement form]
	: e_{n-1} [1 byte 2's complement form]
	: e_n [1 byte 2's complement form]
	: Constrained current error [1 byte 2's complement form]
OUTPUT	: Unconstrained V^*_{dn}
CALLS	: SMULT
DESTROYS	: ALL
DESCRIPTION	: This subroutine measures and stores the unconstrained V^*_{dn} value

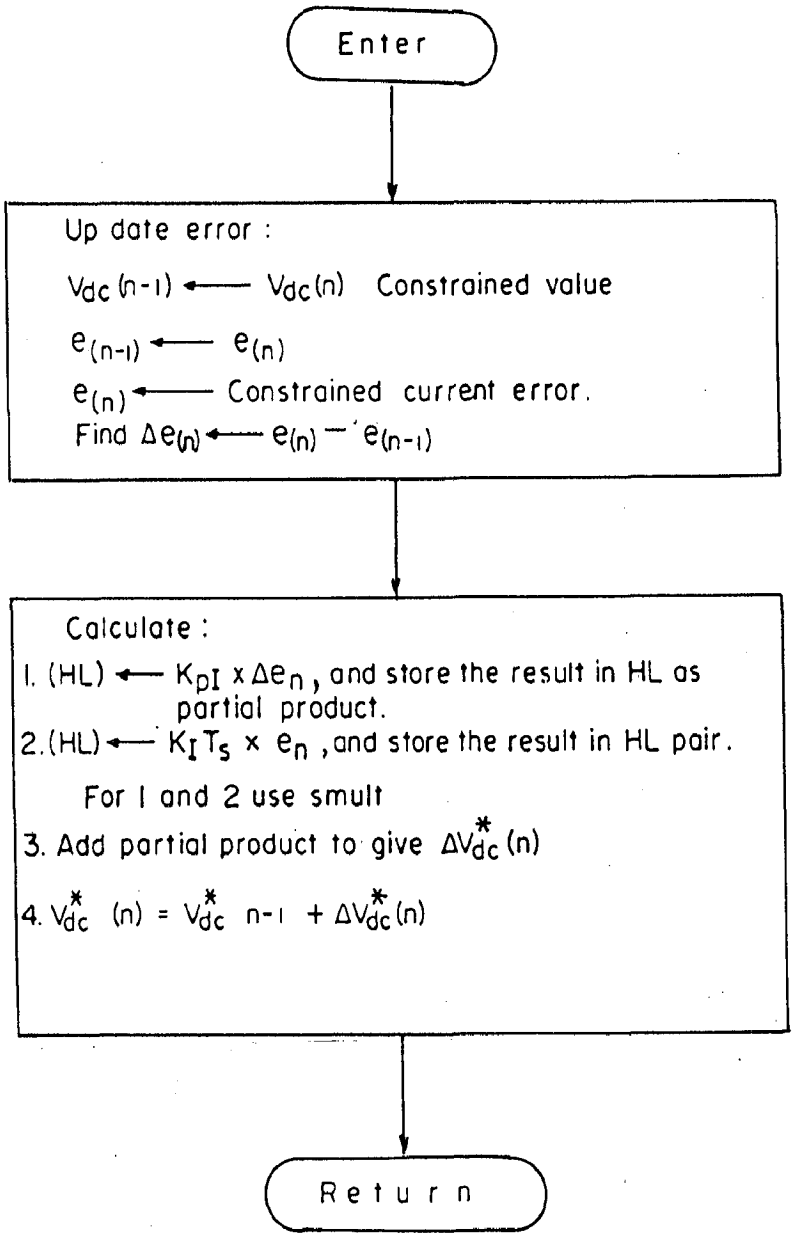


FIG.4.6 FLOW CHART OF PI CONTROLLER OF CURRENT ERROR.

[2 byte 2's complement] in V^*_{dn} address at the end of Subroutine.

4.3.6 Subroutine SMULT

Two's complement numbers can not be multiplied with the algorithm for unsigned binary numbers, because if either or both the operands are negative, the result is incorrect. In one of the methods for multiplying two's complement number, first determine and save the signs of operands. Negative operands are made positive, and the operands are multiplied using the add and shift algorithm. If the operand signs are different, then the two's complement of the product is taken.

Booth's algorithm is a mere direct method of multiplying two's complement numbers. Booth's algorithm is based on the fact that a string of zeros in the multiplier requires no addition, just shifting and string of 1's running from 2^p to 2^q is treated as $2^{q+1} - 2^p$. In Booth's algorithm, the multiplication requires only two operations in addition to the shifts, one subtraction and one addition. This results in a faster multiplication because fewer operations are required.

Booth's algorithm is used which is as follows. Let X_i be the i^{th} bit of an n-bit multiplier. But X_{n-1} is the most significant bit and X_o the least significant bit. A bit $X_{-1}=0$

is assumed. The multiplicant is Y. Starting with i=0, X_i and X_{i-1} are compared. Depending on the comparison, one of the following action occurs.

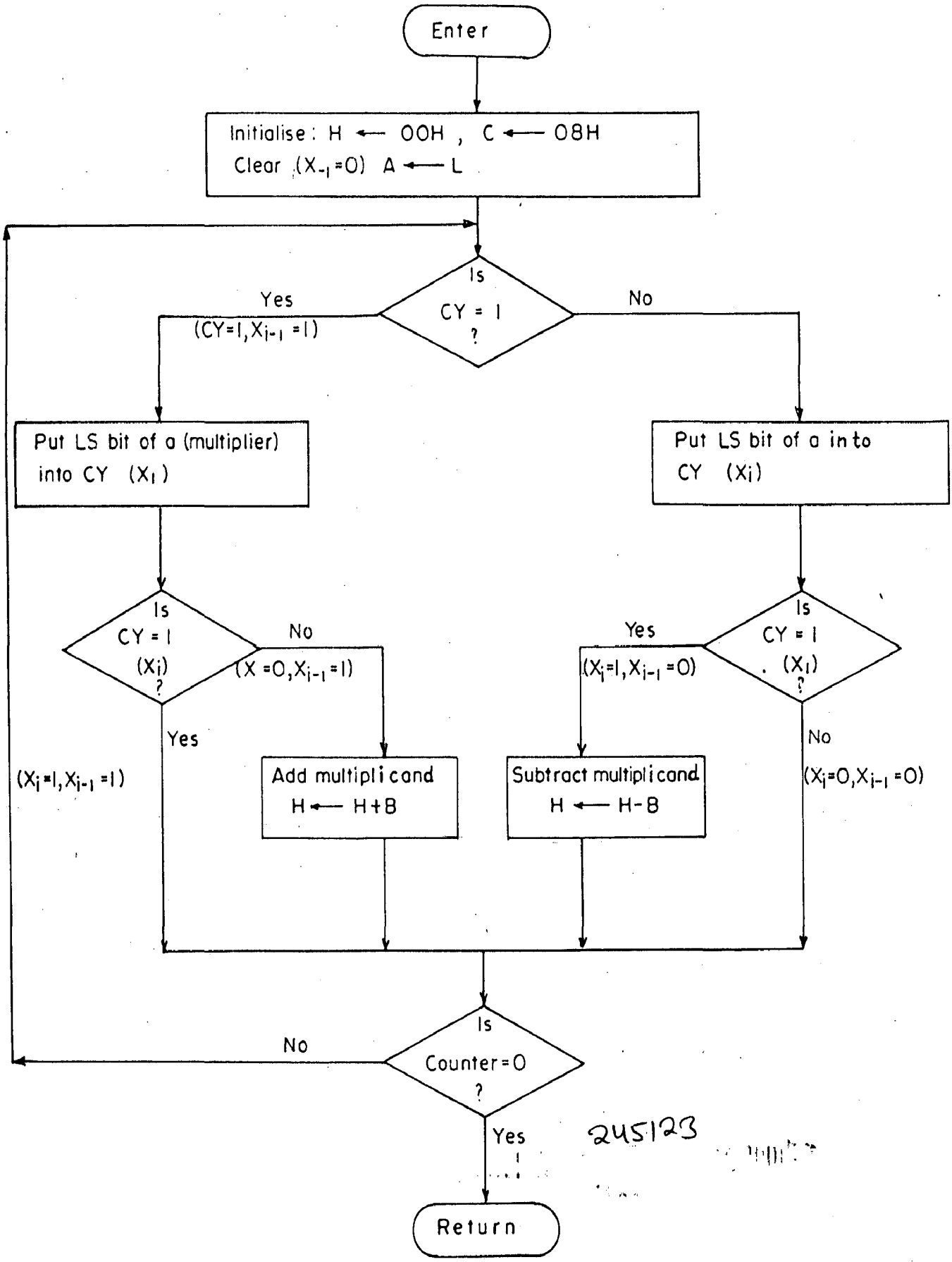
X_i	X_{i-1}	operation
0	0	Shift Y (left with respect to partial product)
0	1	Add Y to partial product and Shift Y.
1	0	Subtract Y from partial product and Shift Y.
1	1	Shift Y

This process is repeated unitl n comparisons are made completing the multiplication.

The flow chart of subroutine is given in Fig.4.7 which implements Booth's algorithm for an 8-bit multiplicand and 8-bit multiplier. The subroutine is entered with the multiplicand in register D and the multiplier in register C. The 16 bit product is retained in register pair HL. This susbroutine is called in the PI subroutine for software implementation of PI controller.

NAME OF SUBROUTINE : SMULT

INPUTS : Register B has multiplicand,
Regist L has multiplier,
Register C is used as counter.



245123

FIG.4.7 FLOW CHART FOR SMULT SUBROUTINE.

OUTPUT	:	Register pair HL has the 16 bit product of the multiplication.
CALLS	:	None
DESTROYS	:	A,F,B,C,H,L Registers.
DESCRIPTION	:	This susbroutine multiplies two numbers (1 byte each), which are in 2's complement form. It stores the result of multiplication in register pair HL. This subroutine can be used for both unsigned numbers multiplication and for 2's complement numbers multiplication.

4.3.7 Constrained Voltage Subroutine

As mentioned in Appendix-C, in the design of PI controller, the maximum value of output voltage is to be limited between maximum and minimum specified limits. In the case of fully controlled converter, the normalised output voltage 00H corresponds to the maximum firing angle i.e. 90°, and FFH corresponds to the minimum firing angle i.e. 0°. The maximum and minimum value of output voltage is limited between FFH and 00H.

This subroutine limits the V_{dn} value to 00H if V_{dn} unconstrained becomes negative, and to FFH if V_{dn} unconstrained is positive also greater than maximum voltage. The flow diagram of it is given in Fig.4.8.

NAME OF SUBROUTINE : Constrained Voltage Subroutine.

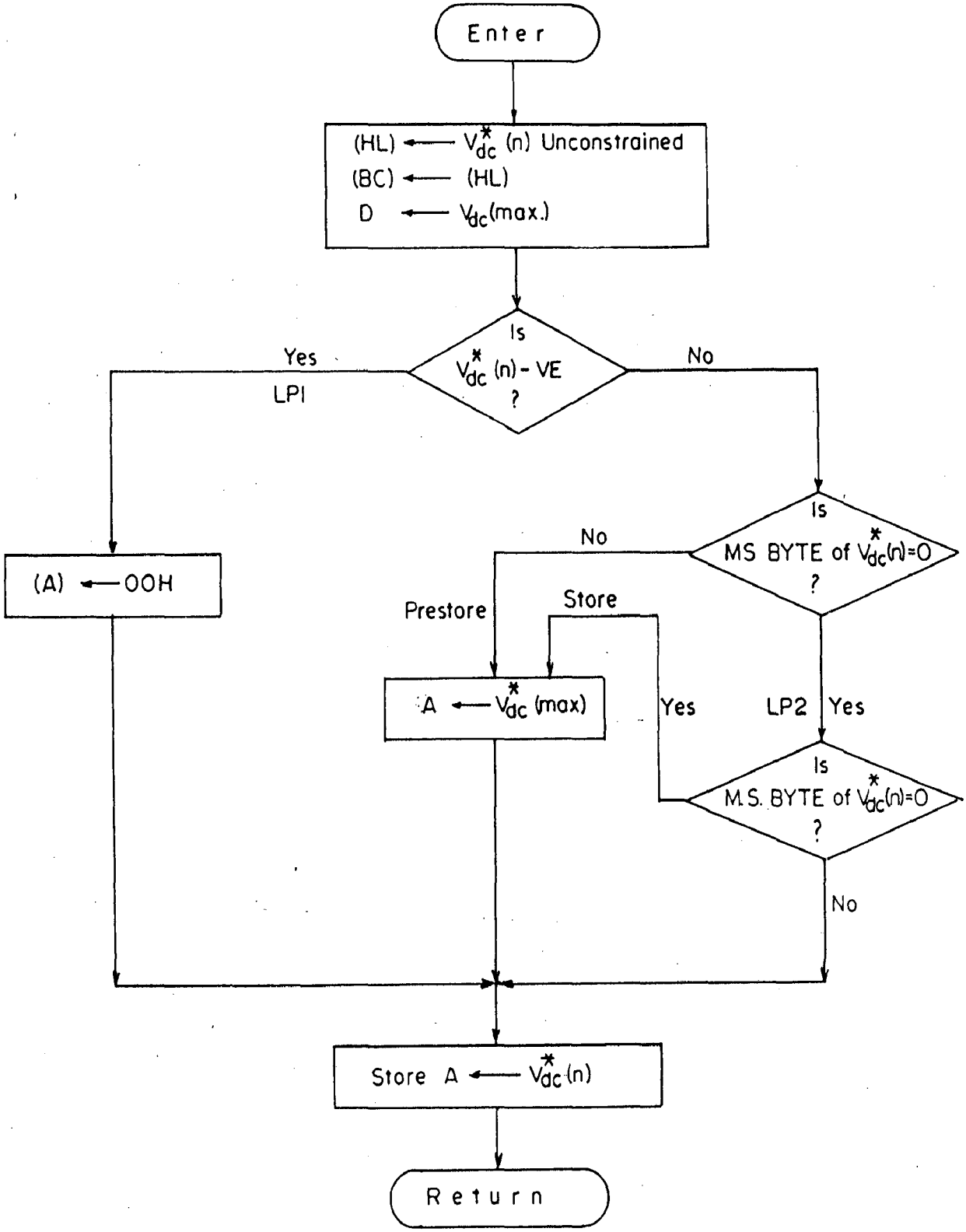


FIG.4.8 FLOW CHART OF CONSTRAINED VOLTAGE SUBROUTINE.

INPUTS	:	V^*_{dn} unconstrained [2 byte data in 2's complement form]
	:	V_{dn} maximum voltage [1 byte]
OUTPUT	:	Constrained V^*_{dn} @[1 byte]
CALLS	:	None
DESTROYS	:	All
DESCRIPTION	:	This subroutine limits V_{dn} value between 00H and FFH, i.e. if V_{dn} is negative, V_{dn} is set as 00H, and if V_{dn} is positive and greater than FFH, then it is set as FFH.

4.3.8 Lookup algorithm Subroutine

While microprocessors potentially offer more flexibility and economy than the analog circuits, their relatively limited real time capabilities restrict their functions to table lookups and simple arithmatic operations. The determination of the necessary control requires only table lookup operations and computations are kept to a minimum. The control algorithm using the table lookup, becomes simple, meets the real time requirements, and its implementation is well within the capability of the present day microprocessors.

Determination of lookup table values:

Assuming (1) continuous conduction mode of thyristor bridge converter.

(2) Ignoring overlap period.

(3) Fixed sequence R-Y-B of R.M.S. input voltage with constant frequency.

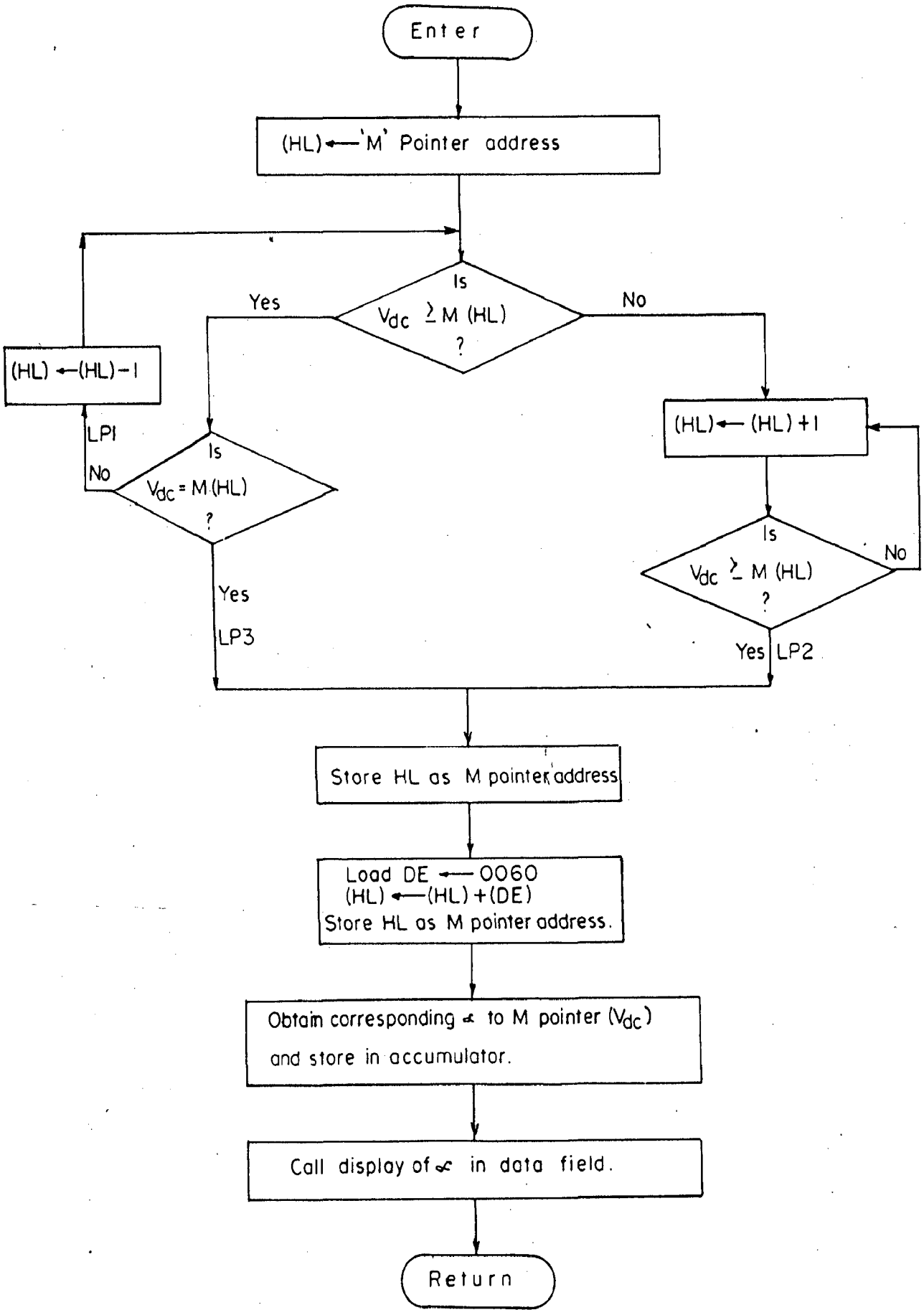


FIG.4.9 FLOW CHART FOR LOOK UP TABLE.

For a three phase fully controlled converter, D.C. output voltage

$$V_{dc} = \frac{3\sqrt{6}}{\pi} V \cos\alpha$$

where V = RMS phase voltage

α = Firing angle

Maximum value of V_{dc} occurs at $\alpha = 0^{o}$

$$V_{dc}\ (max) = \frac{3\sqrt{6}}{\pi} V$$

Therefore $$\frac{V_{dc}}{V_{dc}(max)} = \frac{\frac{3\sqrt{6}}{\pi} V \cos\alpha}{\frac{3\sqrt{6}}{\pi} V} = \cos\alpha$$

V_{dc} (normalised or V_{dn}) varies from '1' to '0' as firing angle 'α' is varied from 0^{o} to 90^{o}. The V_{dn} is represented by a one byte number and have V_{dn} (max) equal to 256 states. Therefore one state is of 1/256 equal to 0.003911 units.

For 'α' count;

360^{o} corresponds to 20 m sec, Hence

1^{o} corresponds to 20/360 m sec = $20/360 \times 10^{-3}$ sec.

Clock frequency of timer = 1.535 MHz

$= 1.535 \times 10^{6}$ Hz

Time for one 'T' State $= \frac{1}{1.535 \times 10^{6}}$ Secs

$\therefore$ 1 Sec corresponds to 1.535×10^{6} 'T' States.

$\therefore \frac{20}{360} \times 10^{-3}$ Sec corresponds to $1.535 \times 10^{6} \times \frac{20}{360} \times 10^{-3}$ 'T' State

$= 82.57$ 'T' States

1° count corresponds to 82.57 'T' States (Decimal)

1° count corresponds to 55H 'T' States (Hexa)

$\therefore \alpha_c = \alpha^{\circ} \times 55H$

The value of α_c and V_{dn} are formulated from the above equations for every 1° variation of firing angle in the range of 0° to 90°. The calculated value of αo and V_{dc} are converted into Hexadecimal form and stored in the memory locations.

V_{dn} is of one byte and α° is also one byte. So in the look up table V_{dn} takes 90 memory locations and corresponding α° takes 90 memory locations for storing.

The look up table consider here in which the V_{dn} values in decreasing order are stored at higher memory locations. The flowchart of it is shown in Fig.4.9.

NAME OF SUBROUTINE : Lookup

INPUTS	:	Constrained V_{dn} which is of 1 byte stored at CVDC address. Its 'm' pointer is loaded with the address having the starting address of V_{dn}.
OUTPUTS	:	It stores 'm' pointer (lookup) table address of V_{dn} (new) at

: 'm' address and also stores α^{o} corresponding to this voltage.

CALLS : None

DESTROYS : A, F, H, L

DESCRIPTION : This subroutine stores lookup table address of V_{dn}(new) at 'm' address, and also stores α^{o} corresponds to V_{dn}(new) which is displayed in data field.

4.3.9 Mode Selection Subroutine

Depending upon the firing angle, it is necessary to fix up the sequence of thyristor firing. This is being done using Mode Selection Subroutine. If the firing angle is between 0^{o} to 60^{o} then Mode '0' is selected which specifies thyristor pair (6,1) has to be fired first and then the remaining pairs. If the firing angle is between 60^{o} to 120^{o}, then 60^{o} is subtracted from it and Mode '5' being selected which specifies Thyristors pair (5,6) has to be fired first and then the remaining pairs. If the firing angle is between 120^{o} to 180^{o}, then 120^{o} is subtracted from it and Mode '4' being selected which specifies thyristor pair (4,5) has to be fired first and the remaining pairs. Therefore, the mode selection helps us in fixing up the particular sequence of firing of thyristor.

NAME OF SUBROUTINE : MODE

INPUTS : Register B has 60^{o} count.
Register C has 120^{o} count.

OUTPUTS : Accumulator is having an 8 bit value of selected mode index.

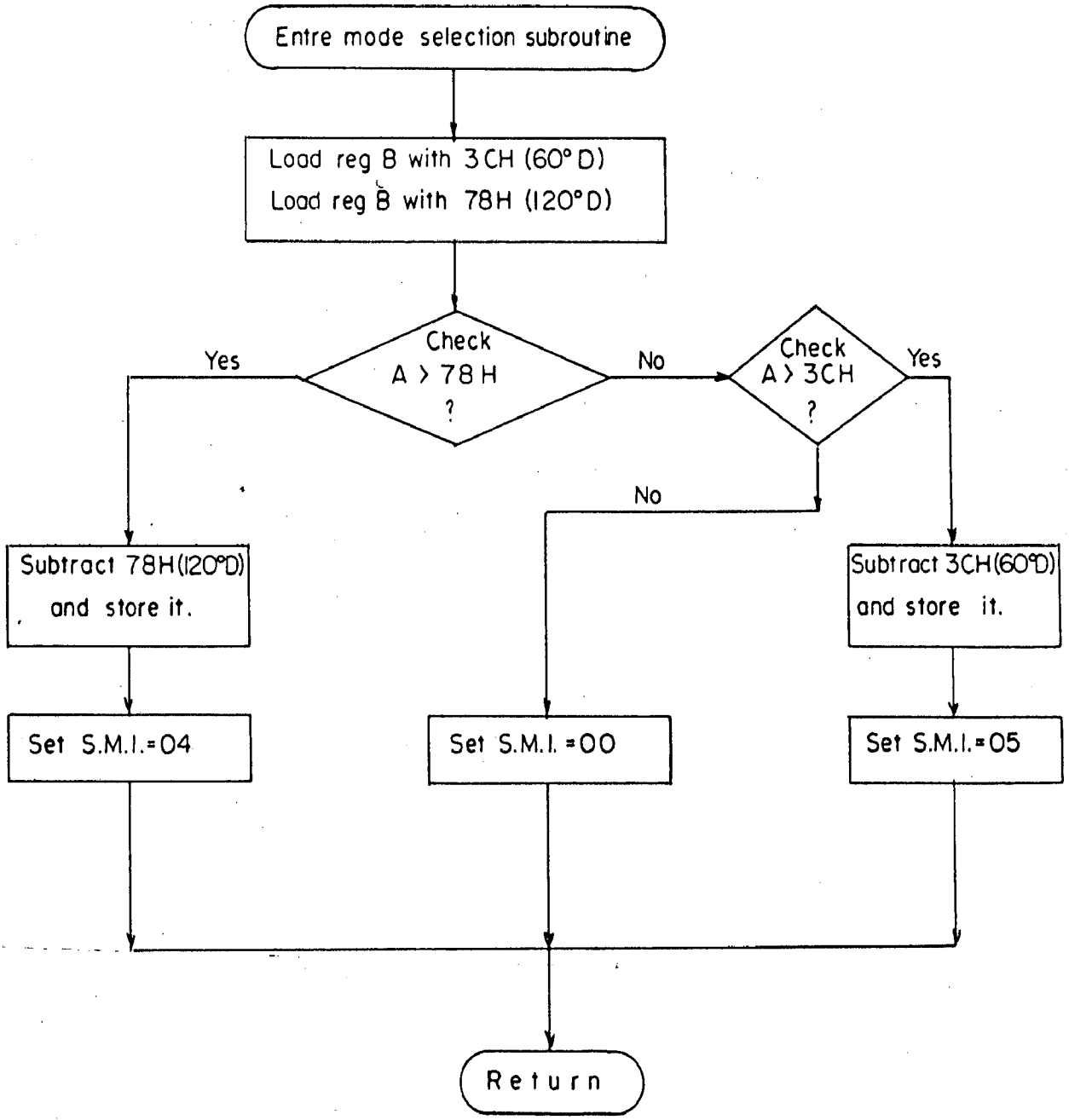


FIG.4.10 FLOW CHART FOR MODE SELECTION SUBROUTINE.

CALLS	:	None
DESTROYS	:	A,F,B,C Registers.
DESCRIPTION	:	This subroutine gives us the particular sequence of Thyristor firing depending upon the firing angle.

4.3.10 Multiplication Subroutine

This subroutine is used to multiply two unsigned numbers. During the execution of this subroutine the firing angle which has been selected from look up table is converted into its equivalent counts by multiplying with 55H counts. This is the firing angle count which is to be loaded to the timer while serving IR_o interrupt Service Subroutine. The flow chart of it is shown in Fig. 4.11.

NAME OF SUBROUTINE	:	MULT
INPUTS	:	Firing Angle
	:	One degree equivalent counts.
OUTPUTS	:	Firing angle count
CALLS	:	None
DESCRIPTION	:	This subroutine multiply the firing angle obtained with 1^o equivalent count and is stored in the memory space allocated for it.

4.3.11 IR_o Interrupt Subroutine

This subroutine is needed to load the

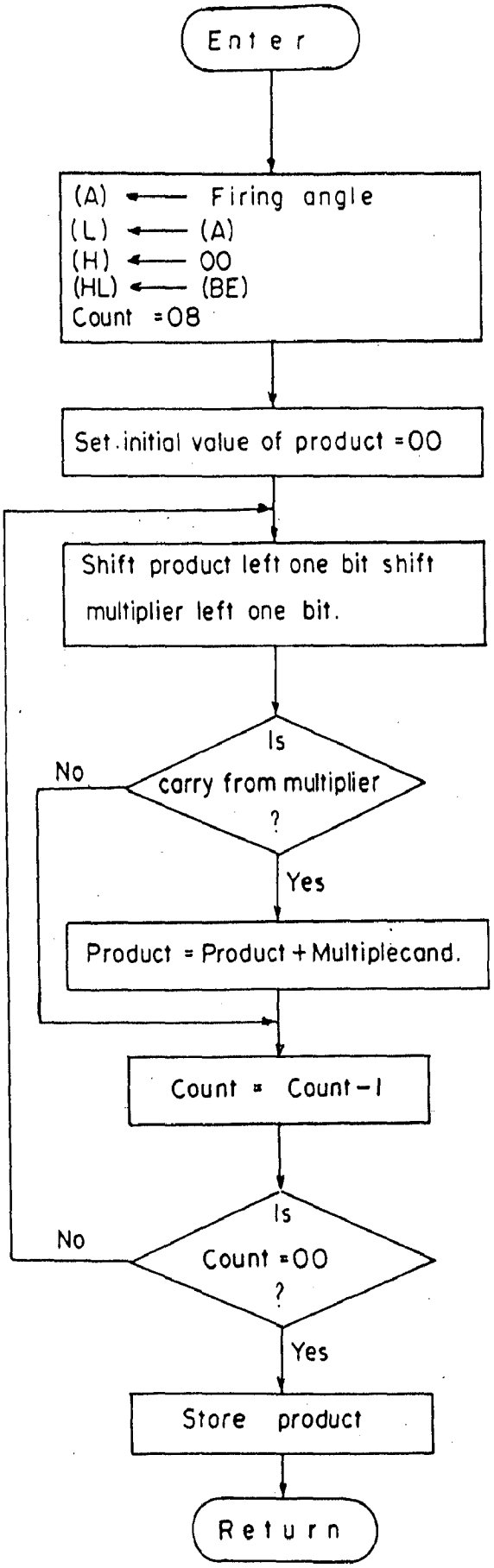


FIG.4.11 FLOW CHART OF MULTIPLECATION SUBROUTINE.

timer or counter TM_1 with firing angle count and also fix up the sequence of firing. The subroutine is entered with saving of all registers, which will be used in the processing. Timer is given with the microprocessor system clock frequency of 1.535 MHz, and gate is made always high by connecting it to +5V. Synchronizing signal is connected to IR_o pin. The IR_o interrupt is served according to the status of the synchronizing signal. When synchronizing signal becomes high it enters IR_o subroutine, then the counter starts down counting till it becomes equal to firing angle count, and when the synchronizing signal becomes low it is disabled. This subroutine is served once in every cycle. The flow chart of IR_o subroutine is shown in Fig. 4.12.

NAME OF SUBROUTINE	:	IR_o
INPUTS	:	Firing angle count
	:	Starting mode index.
OUTPUTS	:	IR_o interrupt output signal.
CALLS	:	None
DESTROYS	:	A,F,H,L
DESCRIPTION	:	This subroutine load the timer TM_1 with firing angle count and triggered it. The output of this subroutine is outputted to IR_1 interrupt subroutine.

4.3.12 IR_1 Interrupt Subroutine

This subroutine is used to issue all six firing commands. The subroutine is entered with saving of all registers.

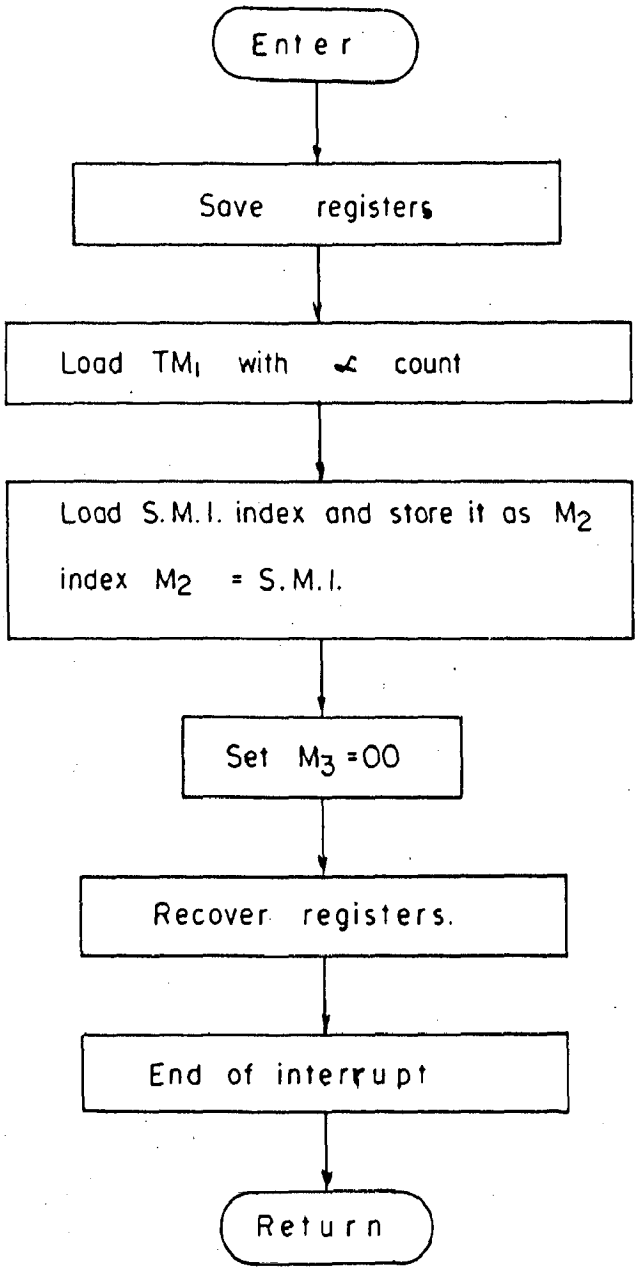


FIG.4.12 FLOW CHART FOR IR_0 INTERRUPT SUBROUTINE.

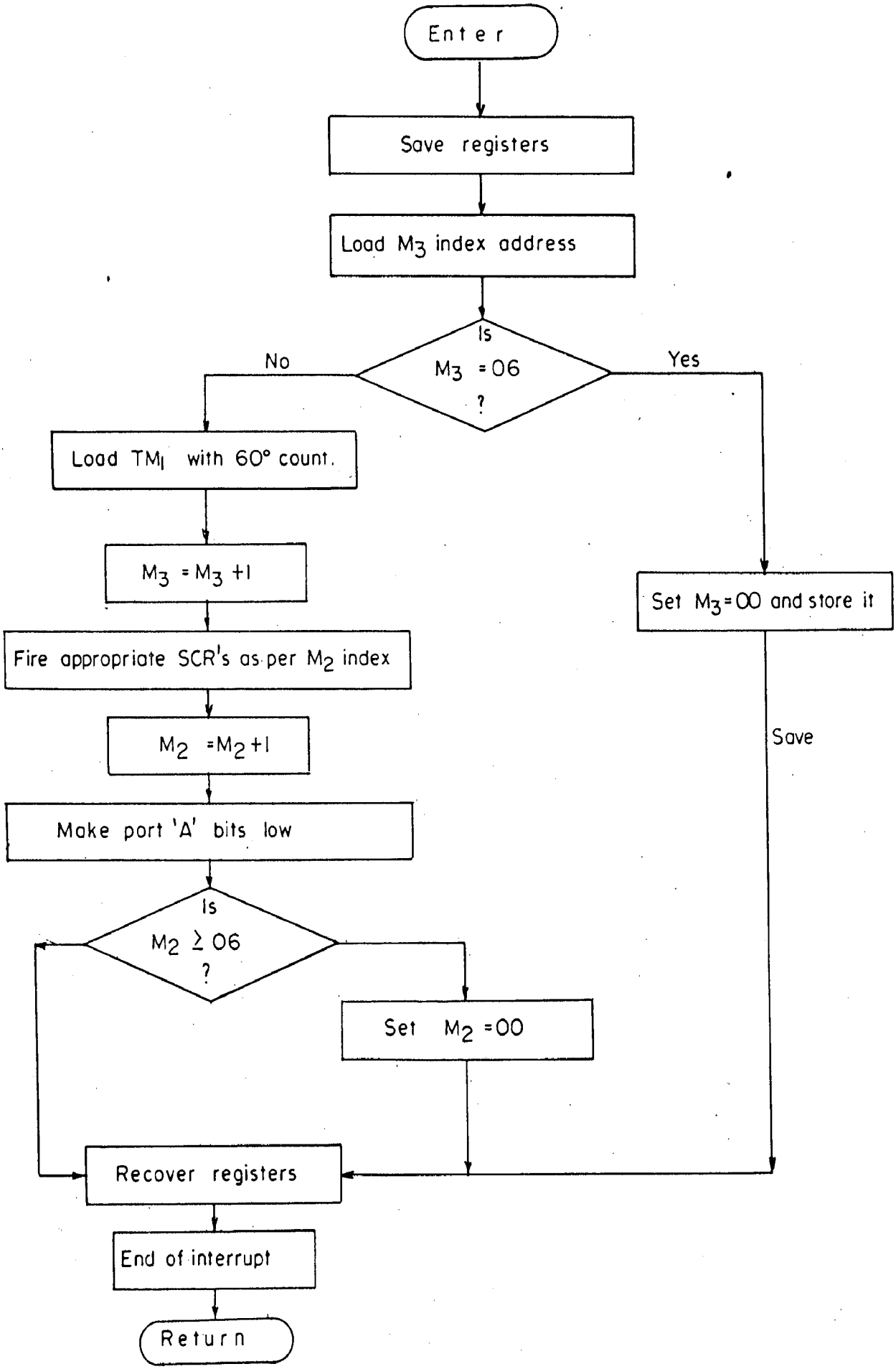


FIG.4.13 FLOW CHART FOR IR_1 INTERRUPT SUBROUTINE

During the execution of subroutine first M_3 index is checked whether timer TM_1 to be loaded with 60^o count or not. The timer is loaded again when the M_3 index is less than 06, after loading the timer with 60^o count each time M_3 index is incremented. When M_3 index becomes equal to 06 indicating that loading of timer is over. Firing command has been issued as per M_2 index which is decided from mode selection subroutine. After issuing each firing command M_2 index is incremented and is checked whether all six firing commands has been issued or not, if not once again IR_1 interrupt subroutine is served to issue next firing command. This is continued till all six firing commands are over. This subroutine is served six times in every cycle. The flow chart of it is given in Fig. 4.13.

NAME OF SUBROUTINE	:	IR_1
INPUTS	:	60^o count, M_3index, M_2 index.
OUTPUTS	:	six firing commands.
CALLS	:	None
DESTROYS	:	All
DESCRIPTION	:	This subroutine is used to issue six firing commands.

4.4 CONCLUSIONS

In this chapter, the complete software implementation is discussed. The self tuning controller program is developed in basic language and is realised through IBM personal computer (PC).

In the 8085 microprocessor system, current proportional plus integral controller is realised through software. The various subroutines used in conjunction with main routine are also discussed. The various described subroutines are ADC subroutine, constrained current error subroutine, PI subroutine, SMULT subroutine, constrained voltage subroutine, mode selection subroutine, IR_0 interrupt service subroutine, IR_1 interrupt service subroutine. The various subroutine software programs are developed and tested individually.

Speed control through selftuning controller on personal computer together with 8085 microprocessor based current loop PI controller scheme has worked satisfactorily with the developed software for the system.

CHAPTER-5

PERFORMANCE OF THE SYSTEM

5.1 GENERAL

This chapter deals with the simulation of self tuning controller for the known system as well as for the d.c. motor considered for the testing purpose. It is realised through software on PC. Recorded waveforms of the self tuning controller are presented here.

The microprocessor based system software is implemented to generate the firing pulses for thyristors triggering at different instants. The recorded waveforms of input voltage, IR_o interrupt, IR_1 interrupt, zero crossing signals are presented here. The current controller proportional plus integral (PI) is realised through microprocessor software system. Motor voltage and motor current at different points of the firing angle control are presented and discussed.

5.2 EXPERIMENTATION

The objectives of the experimentation is to investigate the steady state performance and transient performance of the system. The experimental investigations to obtain the steady state and transient response are done for both on no load and on load conditions. The steady state performance of the

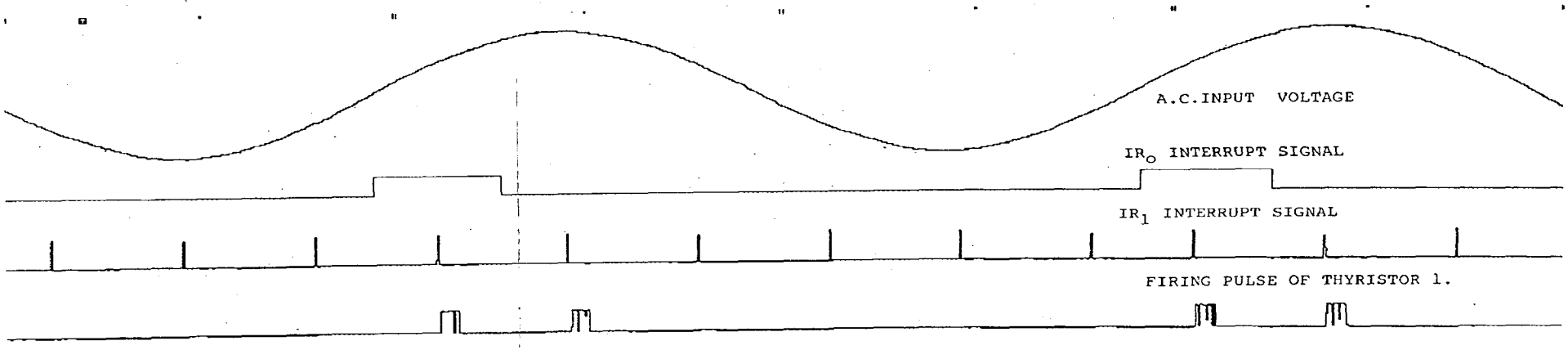


FIG.5.1 VOLTAGE, IR$_o$, IR$_1$ AND THYRISTOR FIRING PULSE (1)
FIRING ANGLE AROUND 40°. 1 ms/division

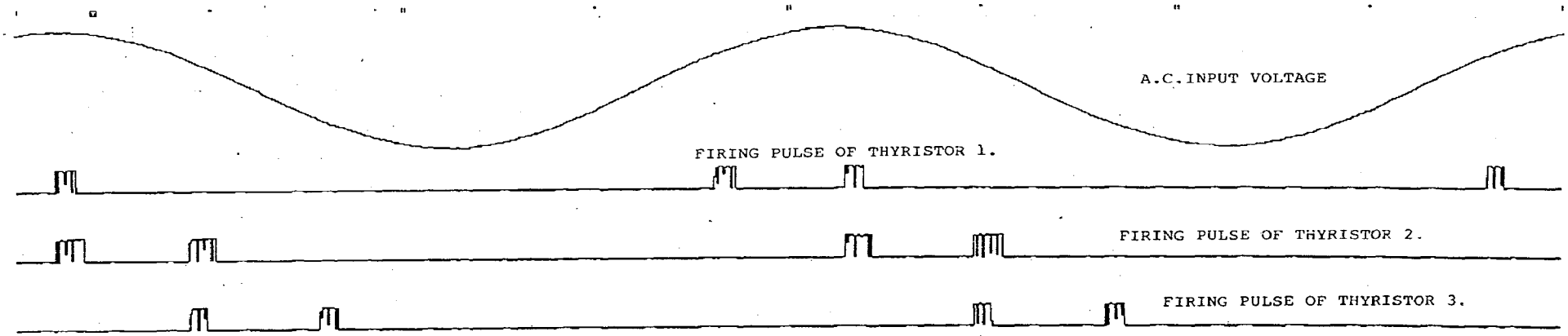


FIG.5.2(a) VOLTAGE, FIRING PULSE OF THYROSTORS 1,2,3
FIRING ANGLE AROUND 40°, 1 ms/division

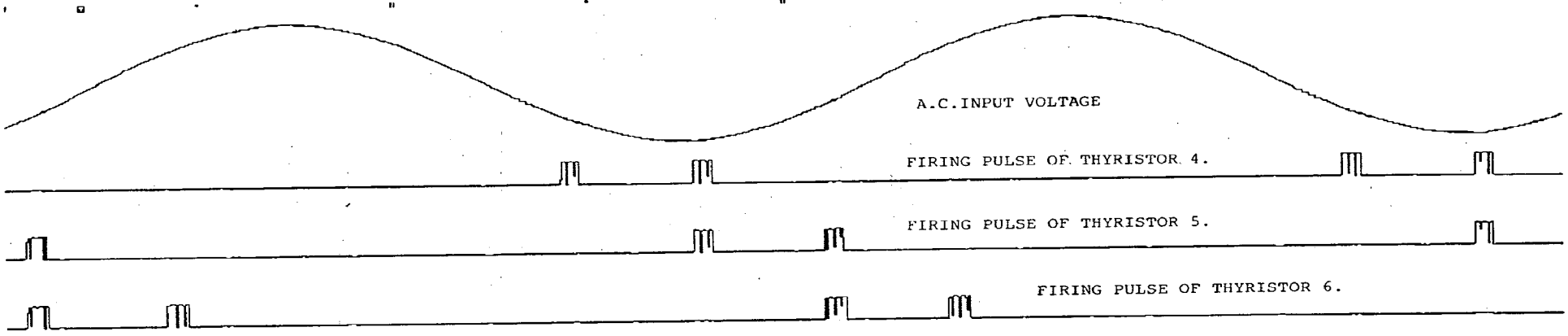


FIG.5.2(b) VOLTAGE,FIRING PULSES OF THYRISTORS 4,5,6
FIRING ANGLE AROUND 40°, 1 ms/division

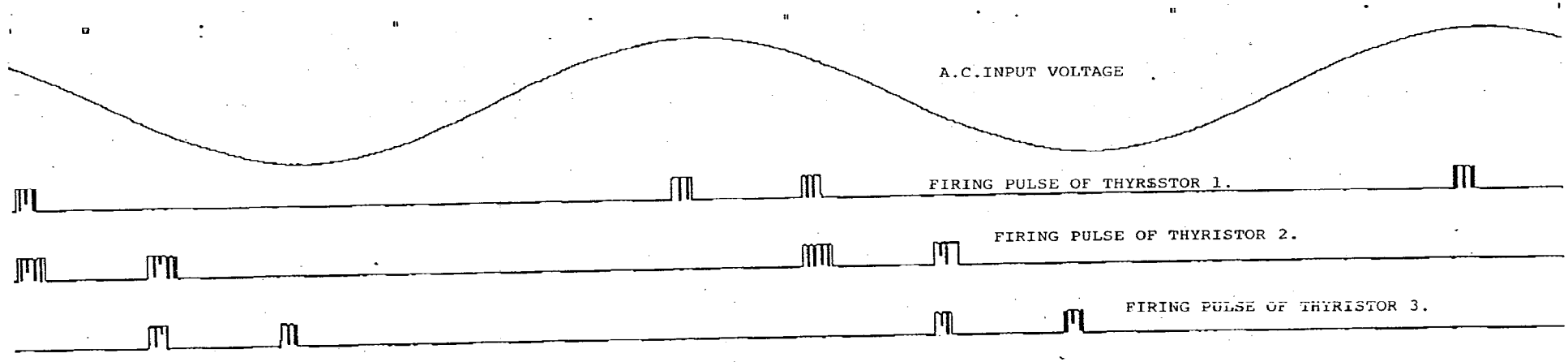


FIG.5.3(a) VOLTAGE,FIRING PULSES OF THYRISTORS 1,2,3
FIRING ANGLE AROUND 75°, 1 ms/division

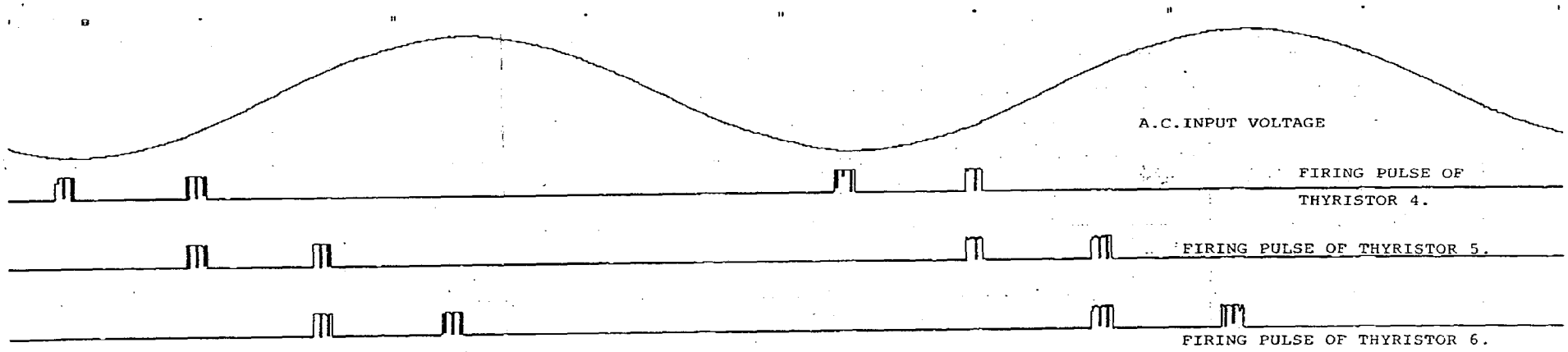


FIG..5.3(b) VOLTAGE,FIRING PULSES OF THYRISTORS 4,5,6
FIRING ANGLE AROUND 75^{o}, 1 ms/division

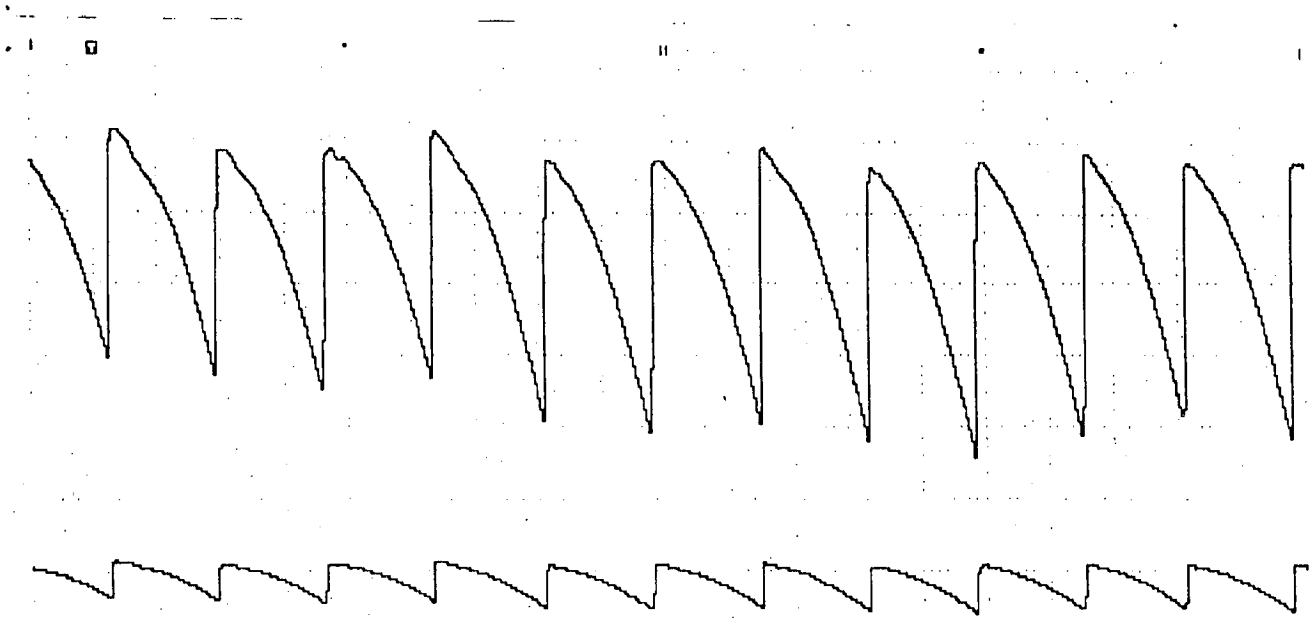


FIG.5.4 VOLTAGE,CURRENT WAVEFORMS OF RESISTIVE LOAD
FIRING ANGLE AROUND 40°, 2 ms/division

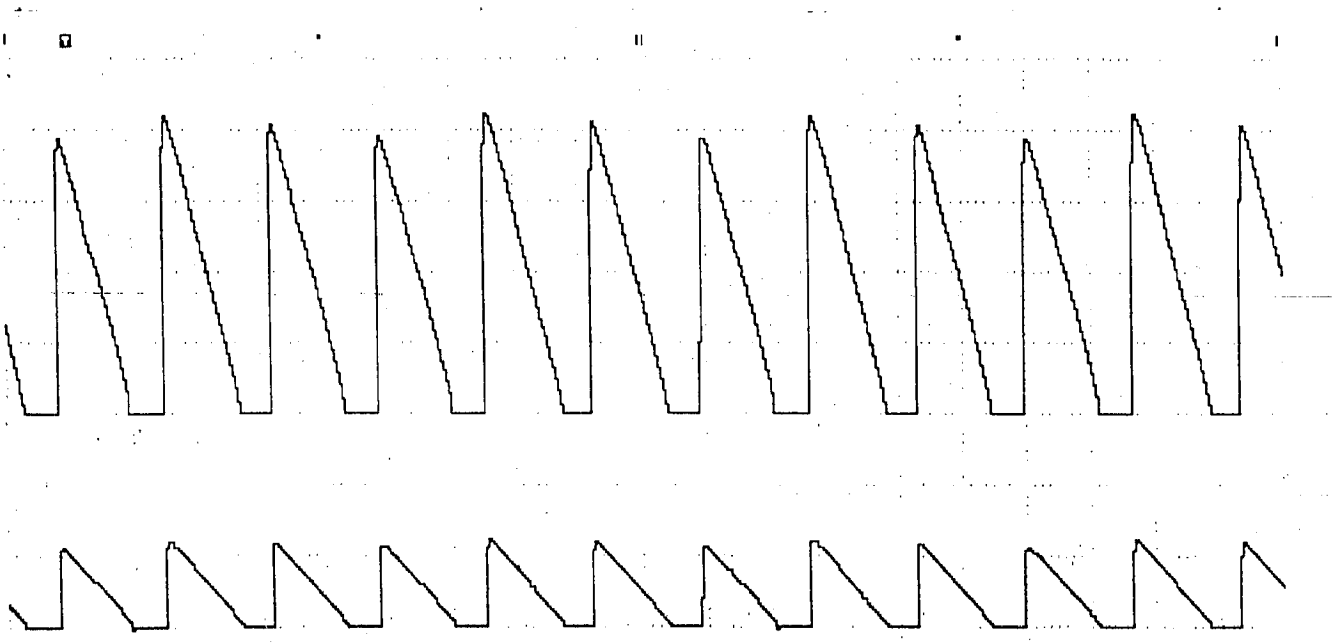


FIG.5.5VOLTAGE,CURRENT WAVEFORMS OF RESISTIVE LOAD
FIRING ANGLE AROUND 75°, 2 ms/division

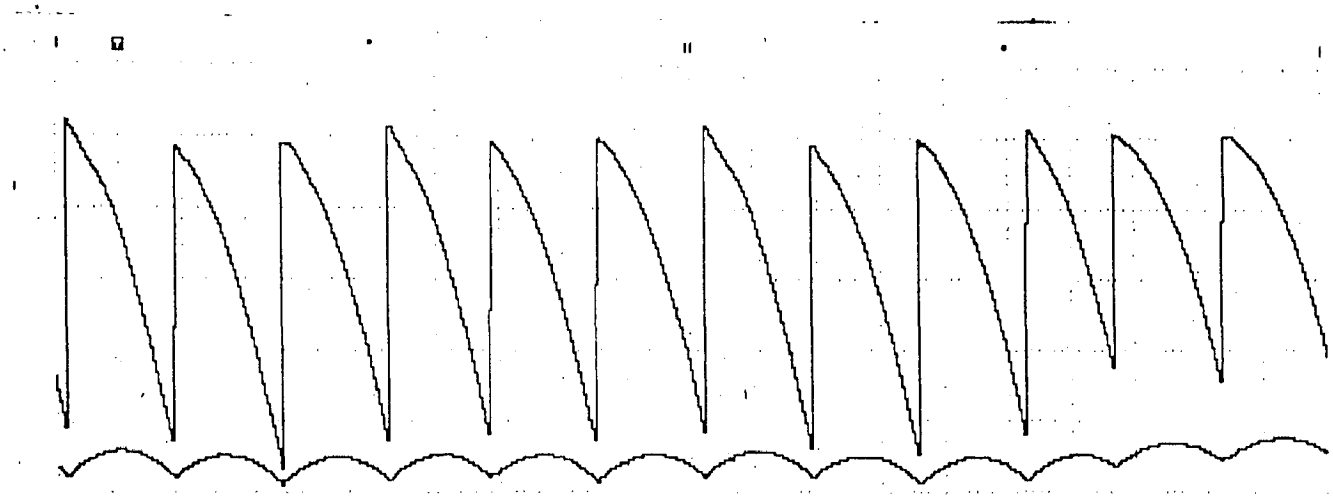


FIG.5.6 VOLTAGE,CUURENT WAVEFORMS FOR R-L LOAD
FIRING ANGLE AROUND 40°, 2ms/division

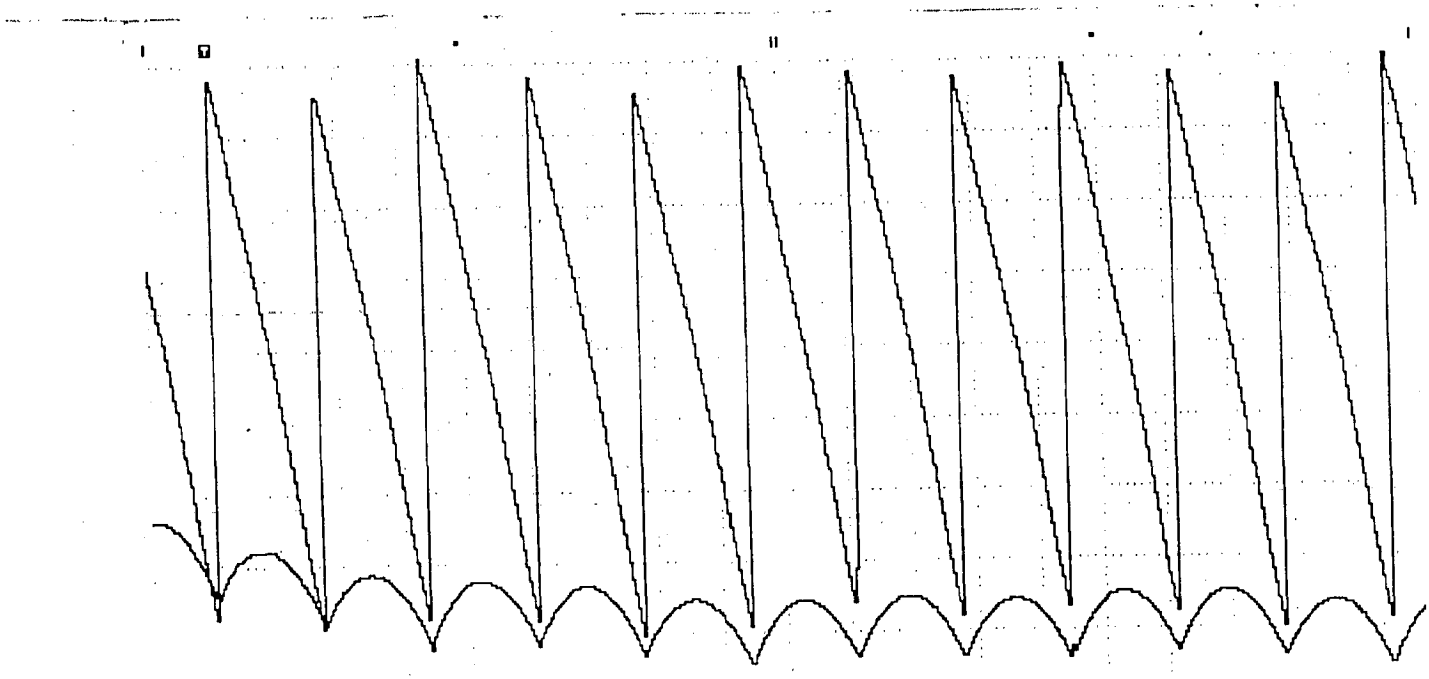


Fig.5.7 VOLTAGE,CURRENT WAVEFORMS FOR R-L- LOAD
FIRING ANGLE AROUND 75°,2ms/division.

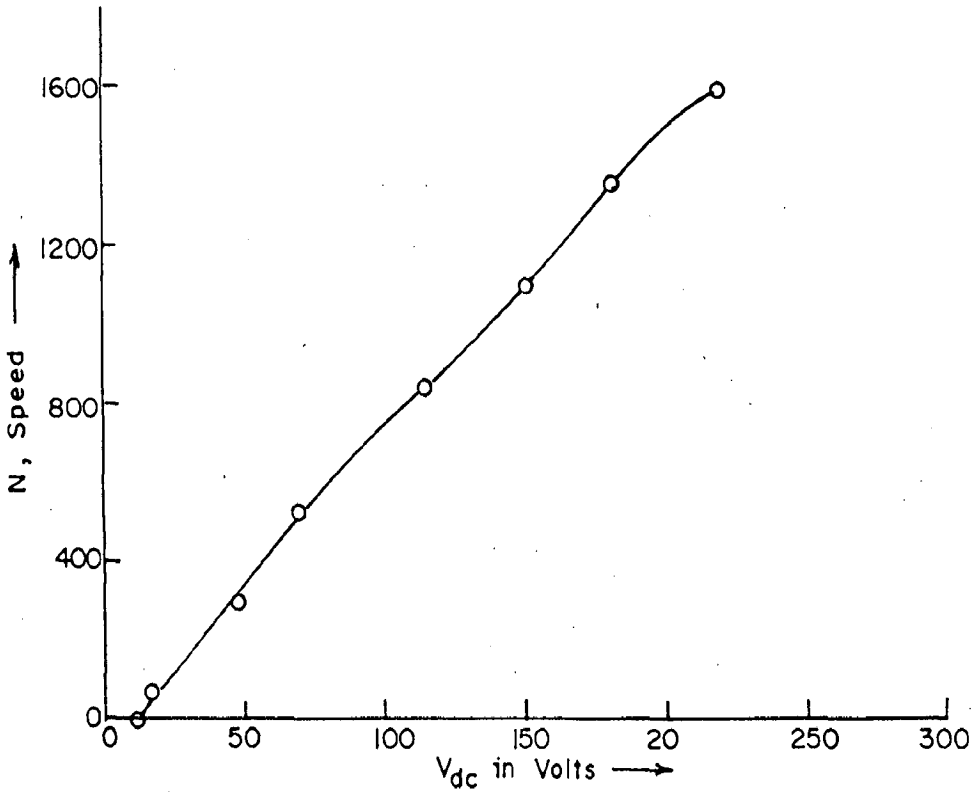


FIG.5.8 OUTPUT D.C.VOLTAGE Vs SPEED.

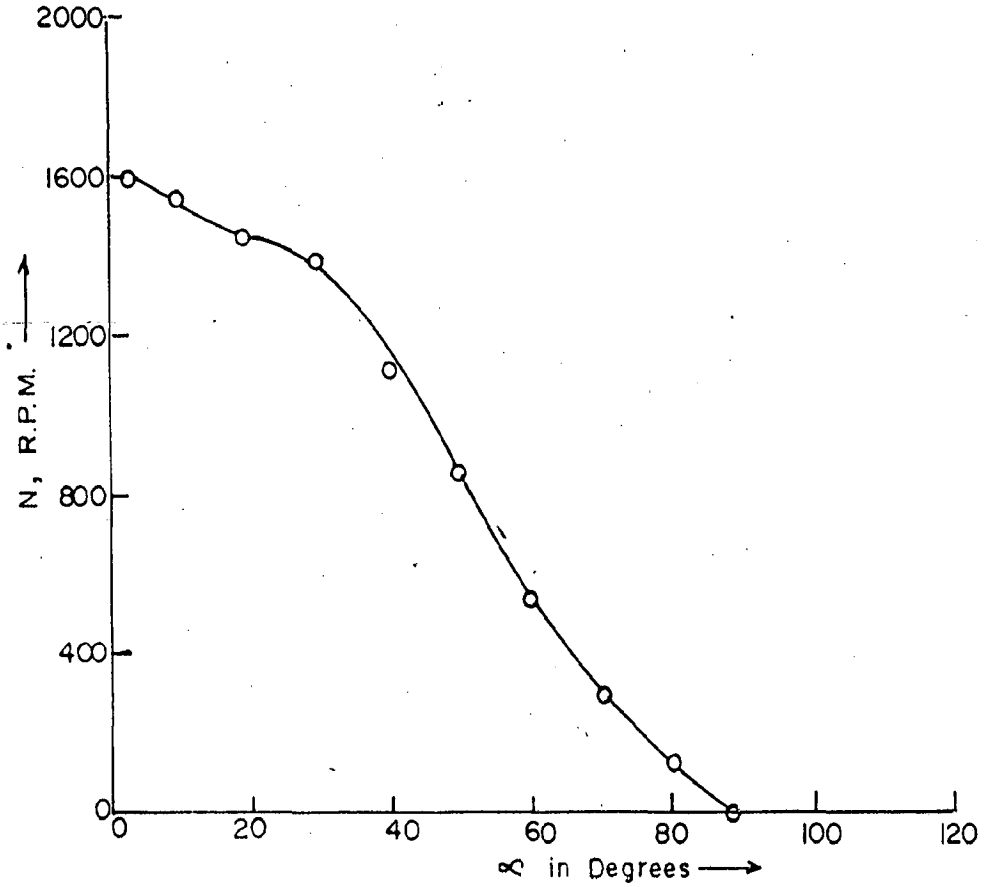


FIG.5.9 FIRING ANGLE Vs SPEED.

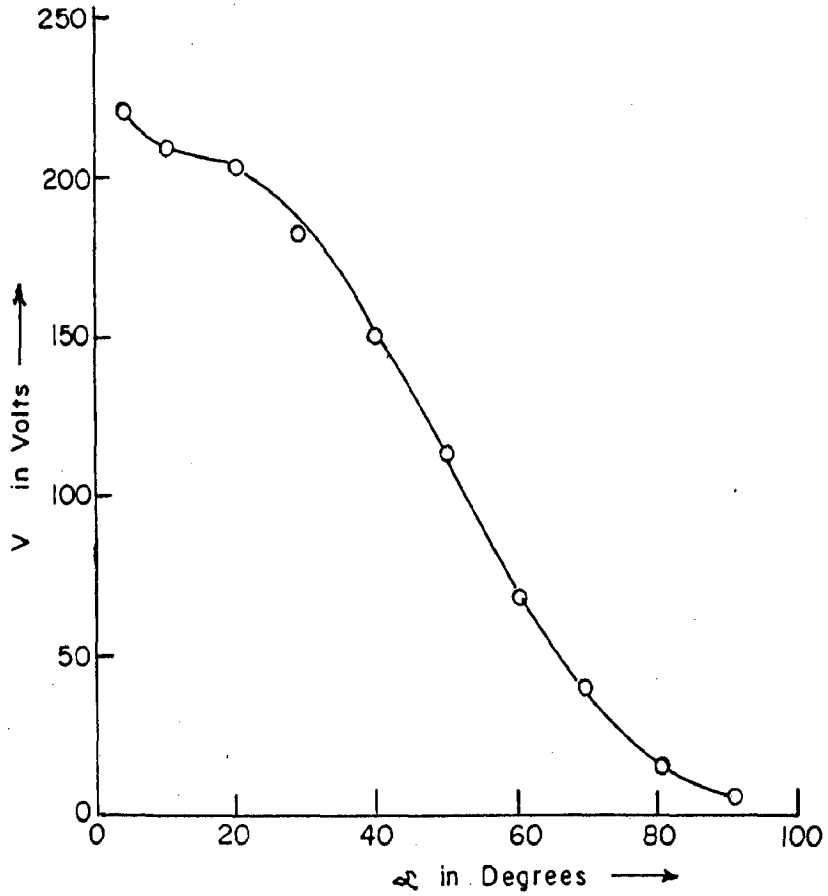


FIG.5.10 FIRING ANGLE Vs D.C. OUTPUT VOLTAGE.

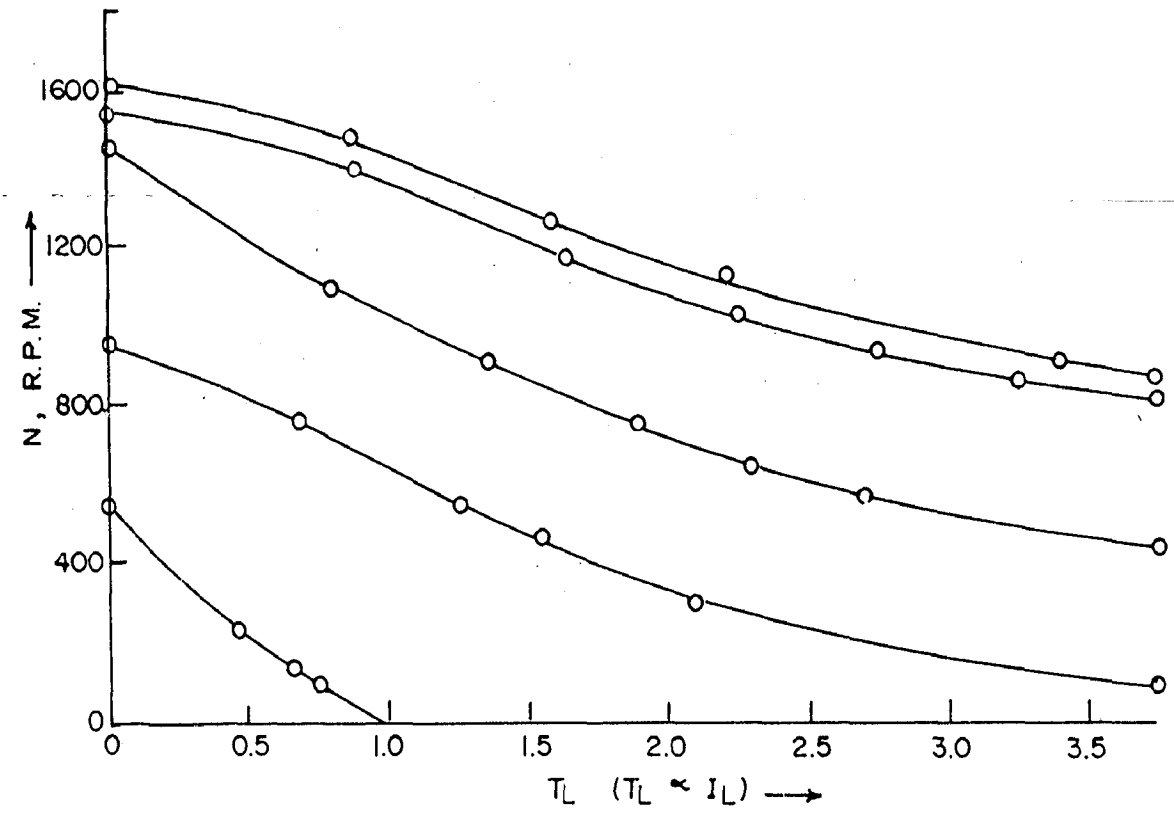


FIG.5.11 TORQUE ($T_L \propto I_L$) Vs SPEED.

drive remains same on both conditions, but changes during transient condition.

The thyristors of the converter are switched ON for 120° duration and at any instant of time two thyristors conduct together. The thyristors are switched ON in the proper sequence depending upon the firing angle information. Port 'A' bits from PA_0 to PA_5 are used to send the firing command from the microprocessor. Six firing commands are executed for completion of one cycle of converter operation based on interrupt signals. This cycle is repeated again and again. The converter develops a variable d.c. output voltage. Recording of a.c. input voltage, synchronizing signal, the interrupt signal, IR_1 interrupt signal and the firing pulses obtained for all six thyristors are shown in Figs.5.1 to 5.3. All these waveforms are taken at a firing angle about 40 degrees. Sufficient inductance is added to smoothen the armature current.

The converter performance is first tested for both resistive load and on RL load ; The recorded waveforms are shown in Figs.5.4 to 5.7. The converter output voltage is then fed to a d.c. motor to study the behaviour of the system on no load condition (open loop). While conducting these tests the input a.c. voltage is adjusted in such a way that for minimum firing angle the full load d.c. output voltage is obtained. The characteristic curves obtained are shown in Figs. 5.8 to 5.10.

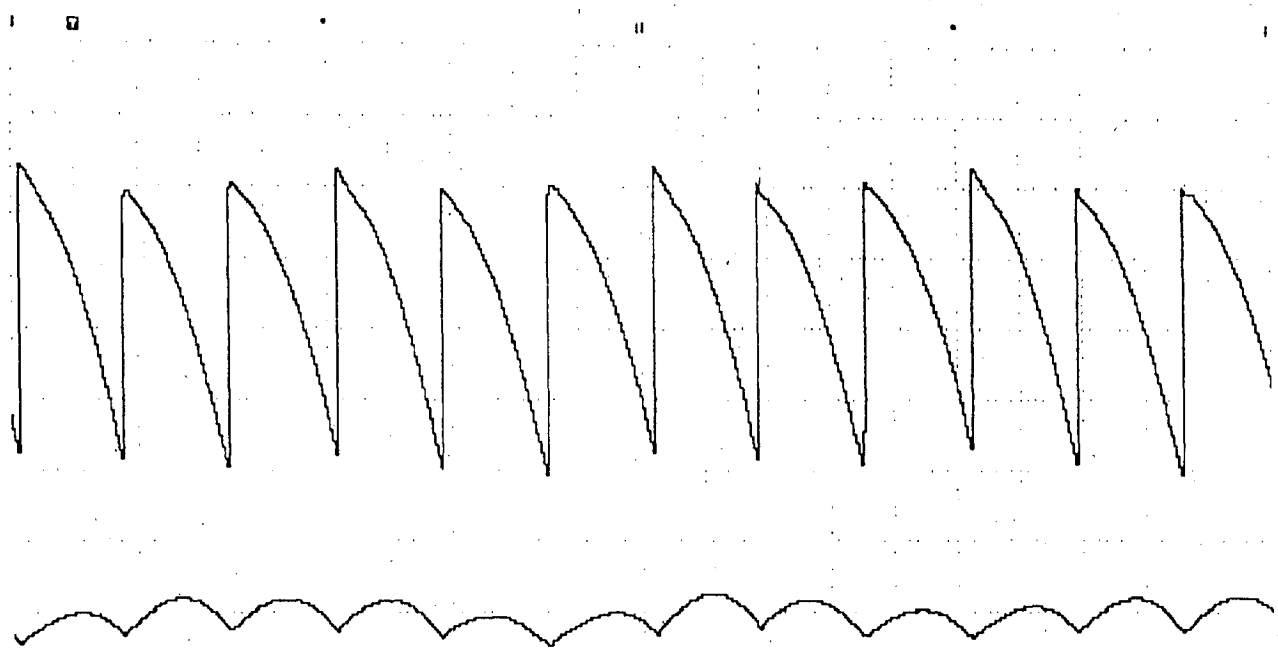


FIG.5.12 NO LOAD MOTOR VOLTAGE AND CURRENT WAVEFORMS
= 40°, N=200 rpm, V_m= 30V, I_m=0.45A,
V_{ac}= 50V, 2 ms/division

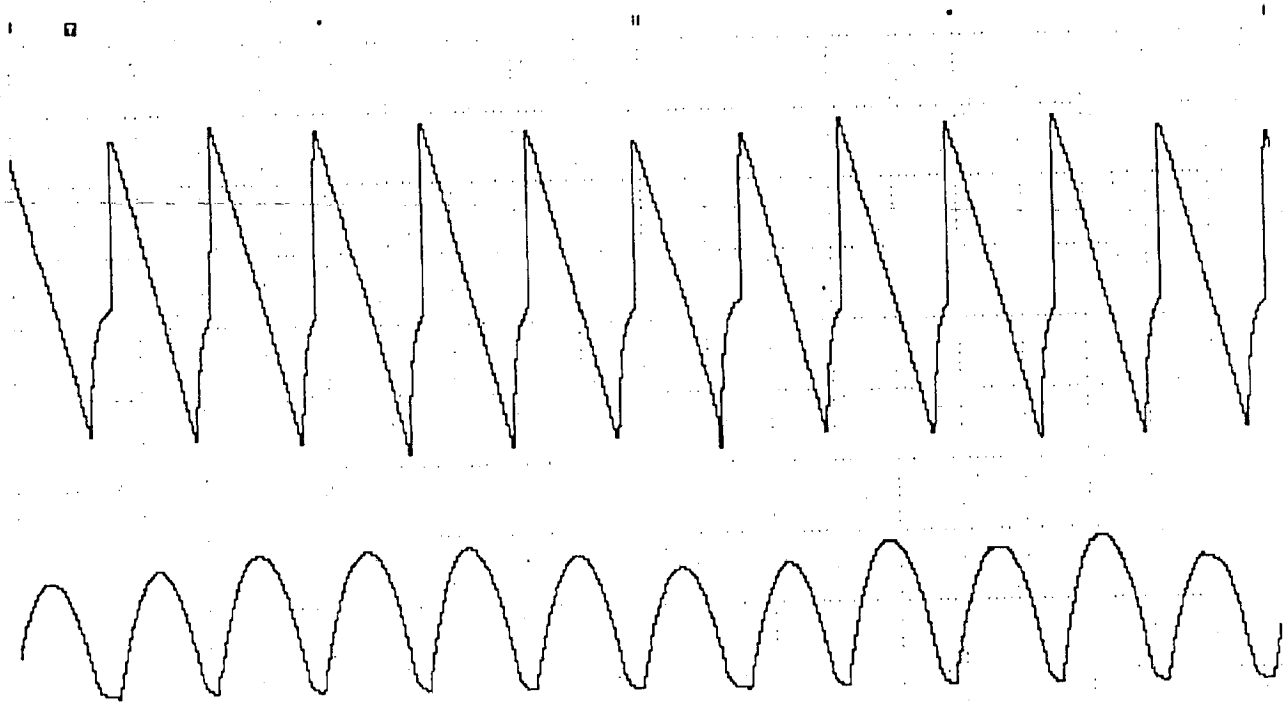


FIG.5.13 NO LOAD MOTOR VOLTAGE AND CURRENT WAVEFORMS
= 75°, N = 440 rpm, V_m = 65V, I_m=.5A,V_{ac}=185V,
2 ms/div.

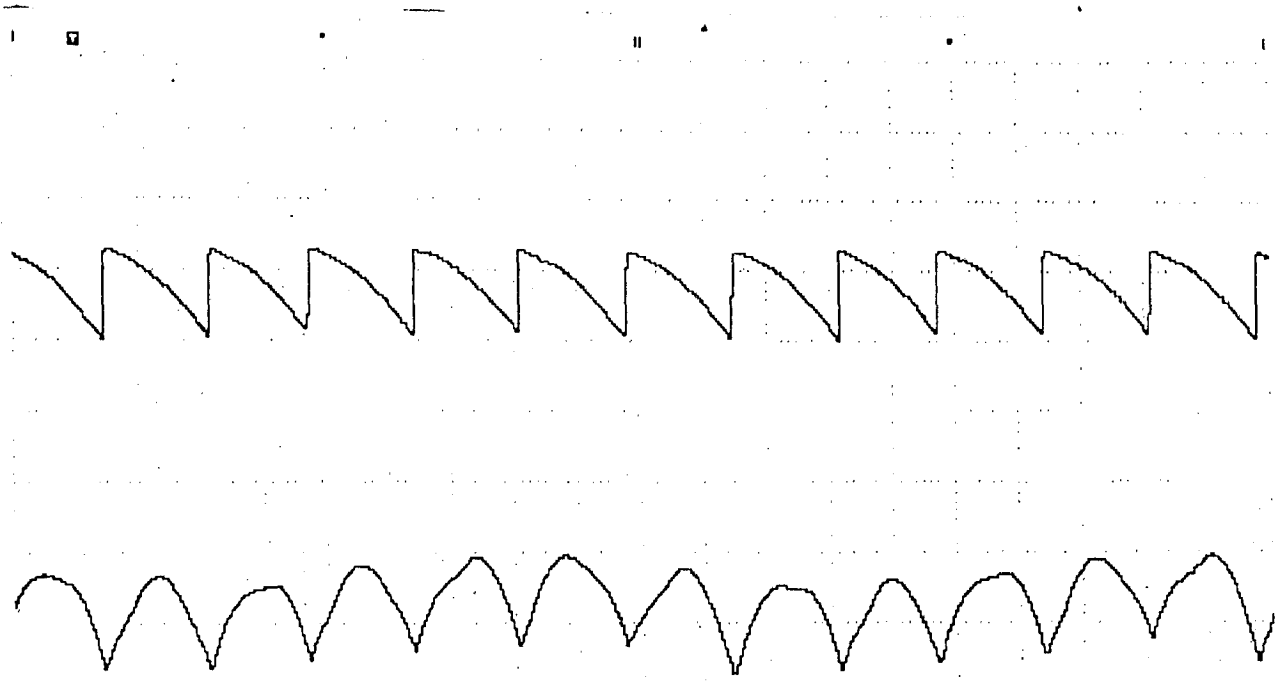


FIG.5.14 NO LOAD MOTOR VOLTAGE AND CURRENT WAVEFORMS
$\alpha = 40^{o}$, N=1530, V_m=210V, I_m=.7A, V_{ac}=185V, 2ms/division

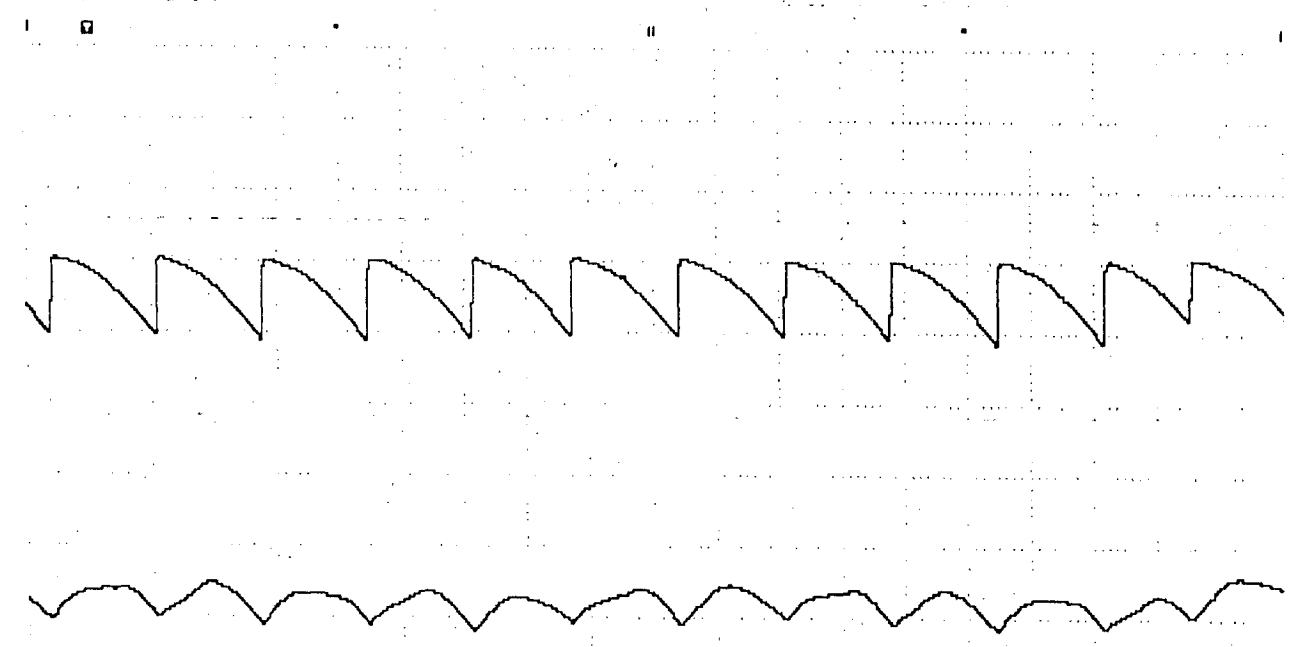


FIG.5.15 LOAD VOLTAGE AND CURRENT WAVEFORMS

V_m=200V, I_m=2.05A, N = 1400 rpm, V_{ac}=185, I_g=1.4A, V_g=200V
$\alpha = 40^{o}$, 2ms/division.

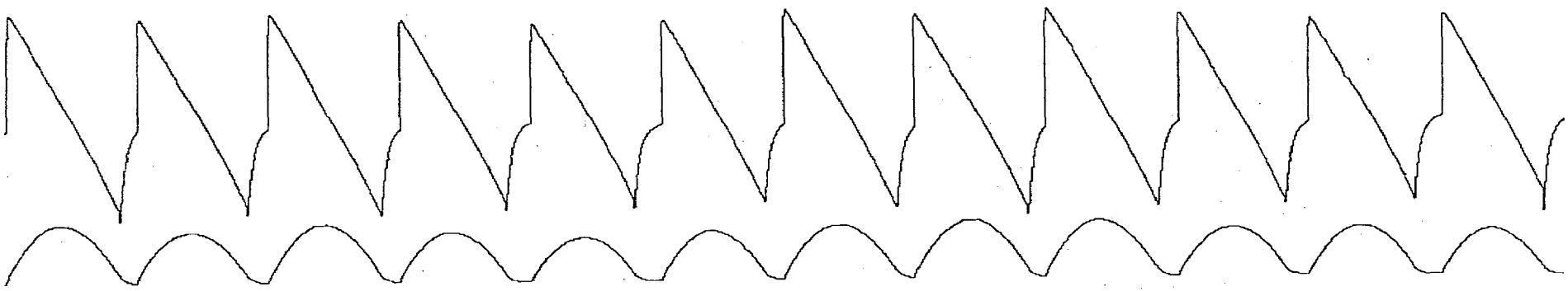


FIG 5.16 LOAD MOTOR VOLTAGE AND CURRENT WAVEFORMS
V_m=50V, I_m = .5A, N = 370 rpm, I_g = 4A, α = 75°, 1 ms/division

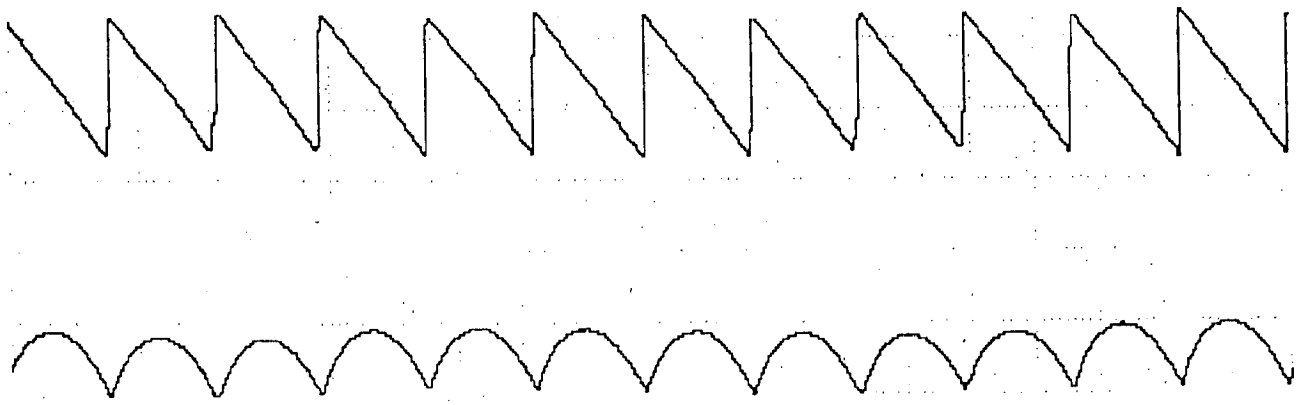


FIG.5.17 LOAD MOTOR VOLTAGE AND CURRENT WAVEFORMS
V_m=50V, I_m = 1.1A, N = 250rpm, I_g=.6A, V_g=40V
V_{ac} = 185 V, = 75°, 2 ms/division

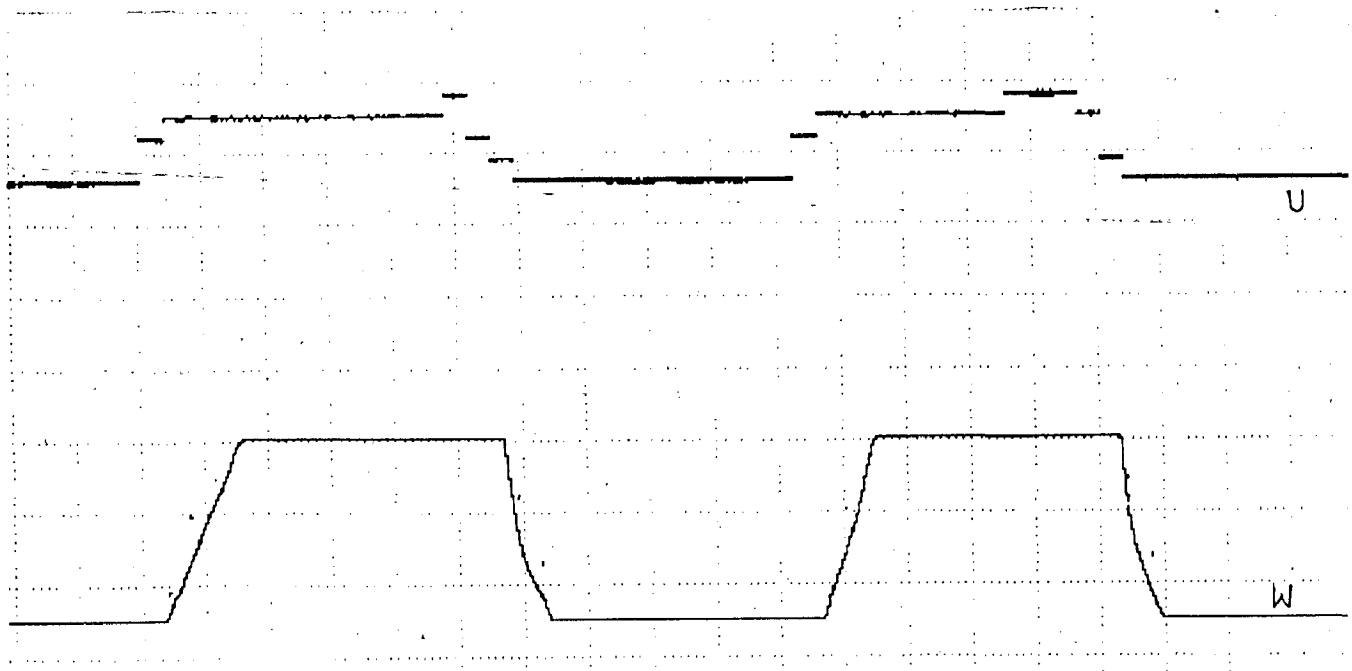


FIG.5.18 WAVE FORMS OF REFERENCE TRACKING SIGNAL & CONTROLLER OUTPUT.
2 ms/division.

The behaviour of the system on load condition (open loop) is studied. During these tests the firing angle is set to a particular value and then the machine is loaded by maintaining the field current as constant. The characteristic curves obtained for various firing angles are shown in Fig. 5.11.

The d.c. motor voltage and current on no load condition for various firing angles are recorded which are shown in Figs. 5.12 and 5.13. The d.c. motor voltage and current waveforms for about 20% and 80% of full load conditions are also recorded and shown in Figs. 5.14 to 5.17.

Self tuning controller which has been realised through the software on PC works satisfactorily for 'OFF' line studies. Recording of the reference tracking signal (W) and the controller output (U) while keeping actual speed (YW) constant as shown in Fig. 5.18.

5.3 DISCUSSION OF RESULTS

The following salient features on the performance of the system may be observed from the results.

The pulses generated to trigger the converter thyristors alongwith the synchronizing signal, IR_o and IR_1 interrupt signal are similar to the theoritical wave forms given earlier in Chapter-III.

The converter which has been tested with resistive load as well as RL load confirms that the current in case of RL load is continuous where as the current may be discontinuous in case of resistive load.

The motor output voltage and current waveforms obtained experimentally are exactly identical with the theoritical ones given in Chapter-III. It shows that the no load motor current is small for lower speed than the higher speed of the machine. For larger firing angle the drive operates in discontinuous current mode of operation which is evident from Fig. 5.13.

From Fig.5.8 it is observed that under no load condition speed increases with increase in voltage. The motor speed rises almost linearly with increase in d.c. voltage, which is one of the desirable feature of the drive.

Fig. 5.9 shows the variation of no load speed of the motor with variation in firing angle. The speed of the motor is large for minimum firing angle and then decreases as the firing angle increases.

The converter output d.c. voltage $V_{dc} \propto V_m \cos\alpha$, confirms the variation of output voltage as a function of cosine angle as shown in Fig. 5.10, which resembles the theoritical wave form.

It is observed from Fig. 5.11 that the motor exhibits the natural torque versus speed characteristics. It is

obvious that at lower torque the motor speed is high and speed is low at higher torques. It is also observed that for different firing angles there is a shift in maximum and minimum speed range which decreases as the firing angle increases along with the increase in torque.

The motor output voltage and current waveforms on loaded condition reveals that at lower firing angle the current is more continuous where as at higher firing angles current becomes discontinuous. There is a dip in the curve because of back emf of motor, at this instant current becomes zero as shown in Figs. 5.15 and 5.17.

The self tuning controller with constant actual speed of the machine had confirmed that the controller output is tracking according to the variation of reference signal, which is the essential feature of self tuning controller,

5.4 CONCLUSIONS

The 'OFF' line studies on self tuning controller for reference tracking shows the satisfactory performance for known system and for discretized model of a d.c. drive under consideration.

Even on the 'ON' line implementation the self tuning controller output is following with respect to reference tracking signal with little oscillations in the controller output because of slow response of DAC and larger execution time.

The firing control scheme results in satisfactory performance and is capable of controlling the firing angle of converter over a wide range. The recorded waveforms confirms that the wave forms of control scheme are identical to theoritical expected ones giving its reliable and satisfactory performance.

The current controller realised through microprocessor based system results in satisfactory performance and capable of maintaining the current constant with in a few cycles of a.c. supply.

The performance of a converter fed d.c. motor has been found satisfactory at different values of firing angles and load conditions.

The motor output voltage and current on load condition shows the continuous current mode and discontinuous current mode of operation of the drive. At larger firing angle the current wave form confirms the discontinuous current mode of operation than at lower firing angles.

CHAPTER - 6

CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

6.1 MAIN CONCLUSIONS

The work reported, here, covers the design, fabrication and testing of closed loop speed control of seperately excited d.c. drive. A three phase fully controlled thyristor bridge is used to feed the variable voltage d.c. supply to armature of d.c motor. The adaptive controller based on self tuning principle has been realised through IBM PC for speed loop and current loop based on proportional integral controller (PI) is achieved through eight bit microprocessor system. The firing pulses for thyristors of converter have been obtained from microprocessor and fabricated hardware in proper sequence at appropriate instants. The scope and main conclusions of the present work are given as follows.

(i) The proposed self tuning controller (STC) for reference tracking has been simulated for the known second order system. From the simulated results it is observed that the controller output and its parameters are setled very quickly within two-three iterations. Moreover, the simulated results of discrete model of d.c. motor under consideration also show the satisfactory response of controller. Based on this investigation it may be concluded that the proposed STC gives the satisfactory performance for known system of second order and discretized model of d.c. drive system under consideration.

(ii) The control scheme for generating the firing pulses of thyristors used in bridge converter has been developed using minimum hardware and software on eight bit 8085A microprocessor based system. The hardware includes only two hardware interrupts (IR_0 and IR_1 of PIC 8259) and only one counter (TM_1 of PIT 8253), synchronizing circuit, ADC interfacing (ADC-0809) and amplifier circuit to amplify the firing pulses issued by microprocessor through the bits of ports (PPI 8255). Apart from reduced hardware, the scheme has only small software due to involvement of only two interrupt routines. The firing scheme results in satisfactory operation and capable of controlling the firing angle of converter over the wide range. The recorded wave forms of control scheme are identical to theoritical expected ones giving its reliable and satisfactory performance.

(iii) The current loop of the drive is also realised through same microprocessor system in which the reference current signal is fed through ADC obtained from IBM PC via DAC channel and armature current feedback is fed through another channel of ADC. These current signals are processed using PI controller and resulted output is taken as d.c. voltage to find delay angle of converter from look-up table. The current PI controller results in satisfactory performance and capable of maintaining the current within few cycles of a.c. supply.

(iv) The self tuning controller for speed tracking is implemented through IBM PC. In which the reference speed (W) and motor speed (YW) signals are fed through the two channels of 12 bit ADC and the output of controller (U) is considered as current reference for

the inner current control loop of the drive. STC output responds very quickly and satisfactory manner to reference perturbations. Moreover, it also responds to load perturbations on the dirve but in slightly sluggish manner. The sluggish response of STC results in oscillations in the speed of drive system. Its main reasons are the delay and error in speed signal sensing, large computation time in high level language (BASIC) and low speed of IBMPC (around 5 MHz).

(v) The performance of converter fed d.c. motor has been obtained experimentally at different values of control parameters. The speed drop of drive for a fixed firing angle of converter is found more compared to case of ideal d.c. supply. The wide range of speed control of drive is observed by using the developed control scheme.

(vi) The motor current wave forms are found continuous over the wide range of speed even at no load and more undulations are observed at low speeds of the drive. The voltage wave forms observed are similar to theoritically expected ones.

Based on the investigations carred out it may be concluded that the STC type of adaptive controller is quite suitable for reference tracking for the slightly slow processes such as in chemical industries, voltage regulating system etc. However, the developed control scheme with reduced hardware is found quite suitable for speed control of d.c. drive. However by improving speed sensing and decreasing computation time of PC by operating it at higher

frequency and choosing suitable programming language it is possible to get the satisfactory speed control of d.c. drive.

6.2 SUGGESTIONS FOR FURTHER WORK

During the course of investigations, certain problems arose which require attention. These problems alongwith some additional aspects for further extension of work are listed as follows:

(i) Development of dedicated system using 8086 microprocessor with 8087 coprocessor may improve the system response.

(ii) Improvement in sped signal sensing using fast encoder will result in fast response and accuracy in control of speed of the drive.

(iii) For fast response and four quadrant operation of the drive, the dual converter or the diode bridge with chopper may be used with additional features of regeneration and bracking.

(iv) The proposed adaptive controller may also be used for higher rating of the drive to justify its cost consideration in various industrial applications.

(v) The reference tracking STC adaptive controllers may also be used for the control of a.c. drive with suitable converter systems.

(vi) The other types of adaptive controllers such as sliding mode (SM) may also be used for control of d.c. drive and performance may be compared with the proposed adaptive controller.

REFERNECES

1. P.C.Sen 'Thyristor d.c. drives', A Wiley-Inter science publications, John Wiley and Sons, Newyork 1985.

2. Y.Pradeep Kumar, 'A microprocessor controlled thyristorized d.c. drive system', M.E. Thesis, March 1985.

3. Kumpati S. Narendra and Richard V. Monopoli, 'Applications of adaptive control', Acedamic press, 1980.

4. Ambrish Chandra, 'Self tuning AVR cum stabilizer', Ph.D. Thesis, June 1987.

5. D.W. Clarke and P.J.Gawtrop, 'Self tuning control', IEE procedings, Vol-126, No.6, pp 633-639, June 1979.

6. A.Y.Allidina and Hughes, 'Generalised self tuning controller with pole assignment', IEE Procedings, Vol-127, No.1, pp 13-18, Jan 1980.

7. P.J.Gawtrop, 'Some interpretations of the self-tuning controller', IEE Procedings, Vol-124, No.10, pp 889-894, October 1977.

8. A.Brickwedde, 'Microprocessor based adaptive control for electrical drives', IFAC control in power electronics, Switzerland, pp 119-123, 1983.

9. Rama chandran Balasubramanian and Kong Hung Wong, 'A micro computer based self tuning IP controller for d.c.

Machines', IEEE Transactions on IA, Vol.IA-22, No.6, pp 989-998, Nov/Dec. 1986.

10. R.M.Stephan, V.Hahn, H.Unbehauen, 'Cascade adaptive speed control of a thyristor driven d.c. motor', IEE Procedings, Vol-135, No.1, pp 49-54, June 1988.

11. L.Liu and N.K.Sinha, 'Adaptive state tracking', IEE Procedings, Vol-135, No.6, pp 429-434, Nov. 1988.

12. T.T.C.Tsang and D.W.Clarke, 'Generalised predictive control with input constraints', IEE Procedings, Vol-135 No.6, pp 451-460, Nov. 1988.

13. W.R.Cluett, S.L.Shah and D.G.Fisher, 'Robustness analysis of discrete-time adaptive control systems using input-output stability theory: a tutorial', IEE Procedings, Vol-135, No.2, pp 133-141, March 1988.

14. B.Ilango, R.Krishnan, R.Subramanian and Sadasivam, 'Firing circuit for three phase thyristor bridge rectifier', IEEE Transactions on IECI, Vol.IECI-25, No.1 pp 45-49, Feb. 1978.

15. Alimirbod and Ahmed El-Amawy, 'A general purpose microprocessor based control circuit for a three phase controlled rectifier bridge', IEEE Transcations on IE, Vol.IE-33, No.3, pp 310-317, Aug. 1986.

16. Pei-Chong Tang, Shui-Shonglu and Yung-Chunwu, 'Microprocessor based design of a firing circuit for three phase fullwave thyristor converter', IEEE Transactions on IE, Vol.IE-29, No.1, pp 67-73, Feb 1982.

17. Gerson H. Pfitsher, 'A microprocessor based synchronization scheme for digitally controlled three phase thyristor power converters', IEEE Transactions on IE, Vol.IE-30, No.4, pp 330-333, Nov. 1983.

18. Remy Simard and V.Rajagopalan 'Economical equidistant pulse firing scheme for thyristorized d.c. drive', IEEE Transactions on IECI, Vol.IECI-22, No.3, pp 425-429, Aug. 1975.

19. H.M.El-Bolock, 'A microprocessor based novel scheme for constant angle triggering of thyristors under a variable frequency anode supply', IEEE Transactions on IA, Vol.IA-16, No.2, pp 165-171, March/April 1980.

20. Mario Benedetti and Carlos F. Christiansen, 'Universal microprocessor controller for thyristor phase control of multiphase converters', Int.J.Electronics, Vol-62, No.3, pp 385-392, 1987.

21. Thadiappan Krishnan and Bellakonda Ramaswami, 'A fast response d.c. motor speed control system', IEEE Transactions on IA Vol.IA-10, No.5, pp 643-651, Sept./Oct. 1974.

22. Soon Chan Hong and Min ho Park, 'Microprocessor based high efficiency drive of a d.c. motor', IEEE Transactions on IE, Vol.IE-34, No.4, pp 433-440,Nov.1987.

23. R.R.Sule, Balakrishna J. Vasanth, P.C.Sen and Mudit Kumar, 'Microprocessor based speed control system for high accuracy drives', IEEE Transactions on IE, Vol.IE-32, No.3, pp 209-214, Aug. 1985.

24. J.B.Plant, S.J.Jorna and Y.T.Chan, 'Microprocessor control of position or speed of an S.C.R. d.c. motor drive', IEEE Transactions on IECI, Vol.IECI-27, No.3, pp 228-234, Aug. 1980.

25. G.Olvier, V.R.Stefanovic and G.E.April, 'Microprocessor controller for a thyristor converter with an improved powerfactor', IEEE Transactions on IECI, Vol.IECI-28, No.3, pp 188-194, Aug. 1981.

26. V.K.Verma and Pramod Agarwal, 'Parameter plane synthesis of a dual converter fed variable d.c. drive system', Electric machine and power system, No.12, pp 65-68,1987.

27. Tadeshi Egame, Jiaging Wang and Takeshi Tsuchiya, 'Efficiency-optimized speed control system synthesis method based on improved optimal regulator theory - application to seperately exited d.c. motor system', IEEE Transactions on IE., Vol.IE-32, No.4, pp 372-380, Nov. 1985.

28. Pradeep K.Nandam and P.C.Sen, 'Analog and digital speed control of d.c. drives using proportional integral and integral proportional control techniques', IEEE Transactions on IE, Vol.IE-34, No.2, pp 227-233, May.1987.

29. B.R.Pelly, 'Thyristor phase controlled converters and and cycloconverters', A Wiley-Inter science publications John Wiley and Sons, New York, 1971.

30. G.K.Dubey, S.R.Doralda, A.Joshi and P.M.K. Sinha, 'Thyristorized power controllers', A Wiley easter limited, 1986.

31. M.Ramamoorthy, 'Introduction to thyristors and applications', East West Press, New Delhi, Second Edition, 1977.

BIBLIOGRAPHY

32. K.P.Gokale and G.N.Revankar, 'Microprocessor controlled seperately exited d.c. motor drive system', IEE Procedings, Part B, Vol-129, No.6, pp 344-352, Nov.1982.

33. Richard Bonert, 'Digital tachometer with fast dynamic response implemented by a microprocessor', IEEE Transactions on IA, Vol.IA-19, No. 4 , pp 1052-1056, Nov/Dec 1983.

34. S.K.Tsu and F.W.Pu, 'Software realization of synchronization and firing control of thyristor converters', IEE Procedings, Vol-131, No.4, pp 141-148, July 1984.

35. T.J.Maloney and F.L.Alvarado, 'A digital method sfor d.c. motor speed control system', IEEE Trans. on IECI, Vol.IECI-23, No.1, pp 44-46, Feb 1976.

36. B.K.Bose and K.J.Jentzen, 'Digital speed control of a d.c. motor with phase locked loop regulations', IEEE Transactions on IECI, Vol.IECI-25, No.1, pp 10-13, Feb.1978.

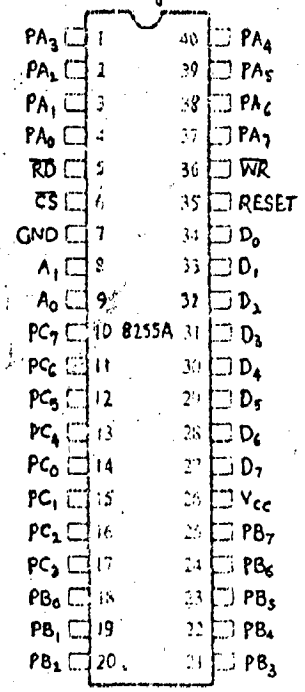
37. A.P.Malvino, 'Digital Computer Electronics', TMH, New Delhi, Second Edition, 1983.

38. Ramesh S. Goankar, 'Microprocessor Architecture, programming and applications with the 8085/8086A', Wiley Eastern limited, New Delhi.

APPENDIX—A

CHIP DETAILS.

Pin Configuration

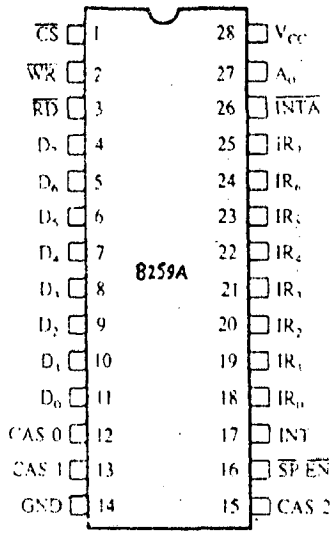


Pin Names

D_7-D_0	Data Bus (Bidirectional)
RESET	Reset Input
$\overline{CS}$	Chip Select
$\overline{RD}$	Read Input
$\overline{WR}$	Write Input
A_0, A_1	Port Address
PA_7-PA_0	Port A (Bit)
PB_7-PB_0	Port B (Bit)
PC_7-PC_0	Port C (Bit)
V_{CC}	+5 Volts
GND	0 Volts

FIGURE A1. 8255A Block Diagram

Pin Configuration

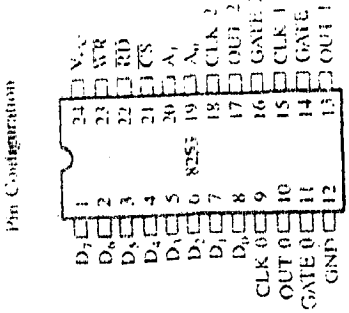


Pin Names

D_7-D_0	Data Bus (Bidirectional)
$\overline{RD}$	Read Input
$\overline{WR}$	Write Input
A_0	Command Select Address
$\overline{CS}$	Chip Select
CAS_2-CAS_0	Cascade Lines
$\overline{SP}$ $\overline{EN}$	Slave Program/Enable Buffer
INT	Interrupt Output
$\overline{INTA}$	Interrupt Acknowledge Input
IR_7-IR_0	Interrupt Request Inputs

FIGURE A3.
The 8259A Block Diagram

Pin Configuration



Pin Names

D_7-D_0	Data Bus (8 Bit)
CLK N	Counter Clock Inputs
GATE N	Counter Gate Inputs
OUT N	Counter Outputs
$\overline{RD}$	Read Counter
$\overline{WR}$	Write Command or Data
$\overline{CS}$	Chip Select
A_0-A_1	Counter Select
V_{CC}	+5 Volts
GND	Ground

FIGURE A2
8253 Block Diagram

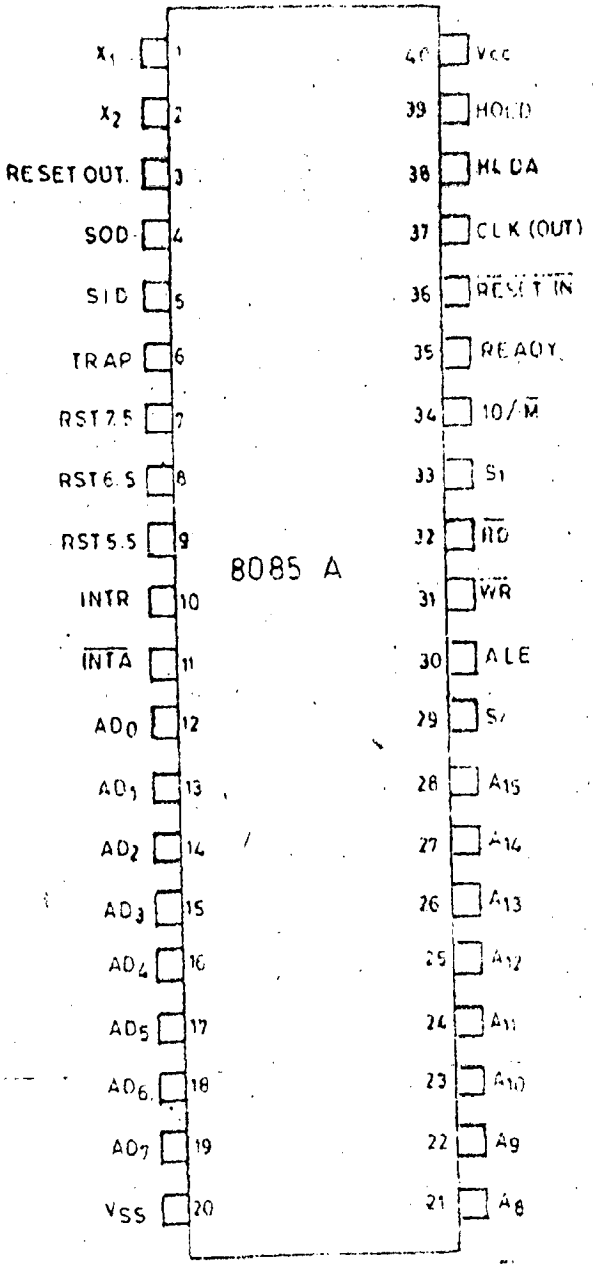
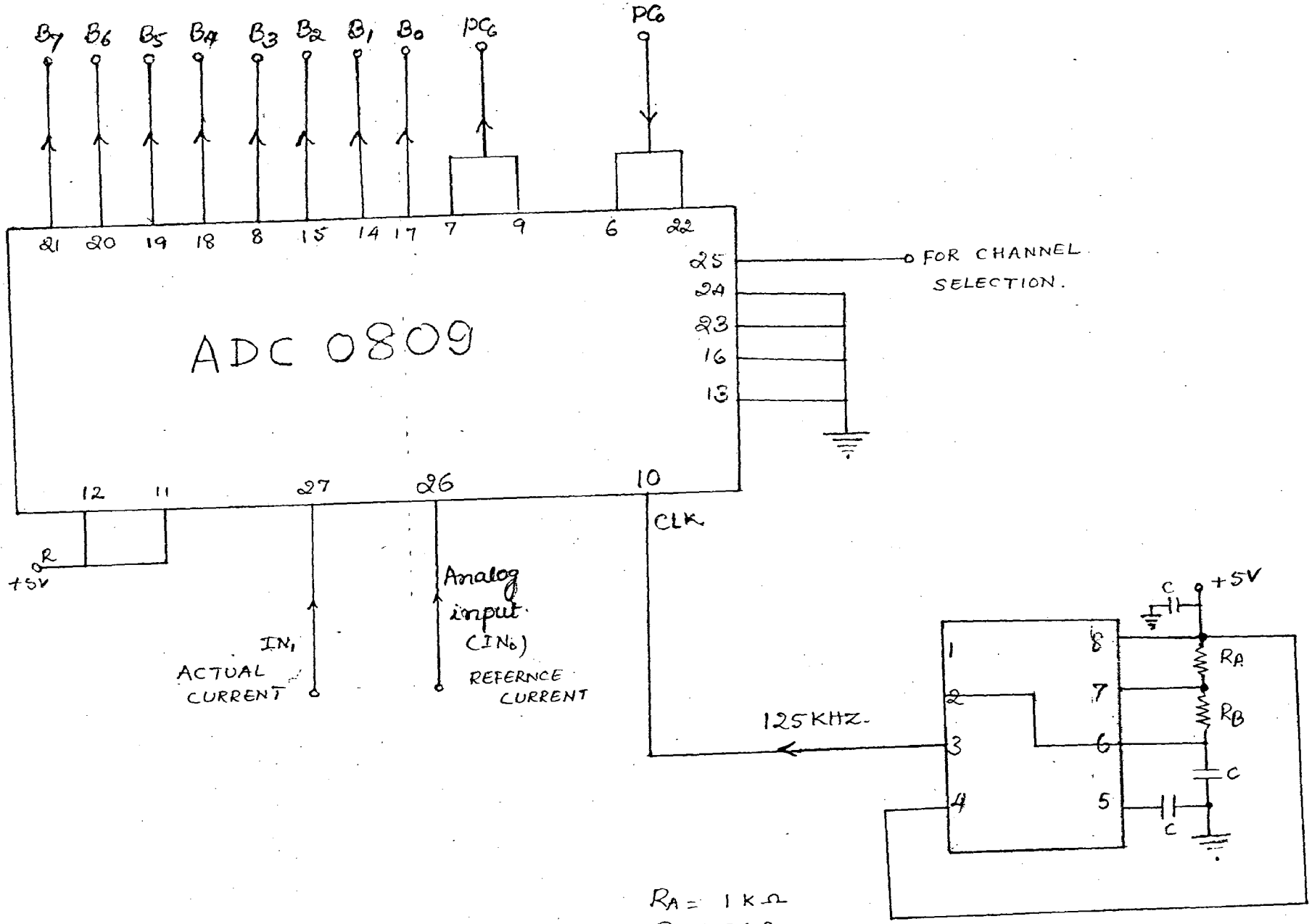


FIG. A4 8085A PINOUT DIAGRAM



$R_A = 1\,K\Omega$
$R_B = 76\Omega$
$C = 0.01\,\mu F.$

FIG. A5 ADC INTERFACING.

APPENDIX-B

PIN CONNECTION DIAGRAMS OF VARIOUS IC CHIPS USED BY THE AUTHOR IN THE PRESENT WORK ARE GIVEN AS FOLLOWS.

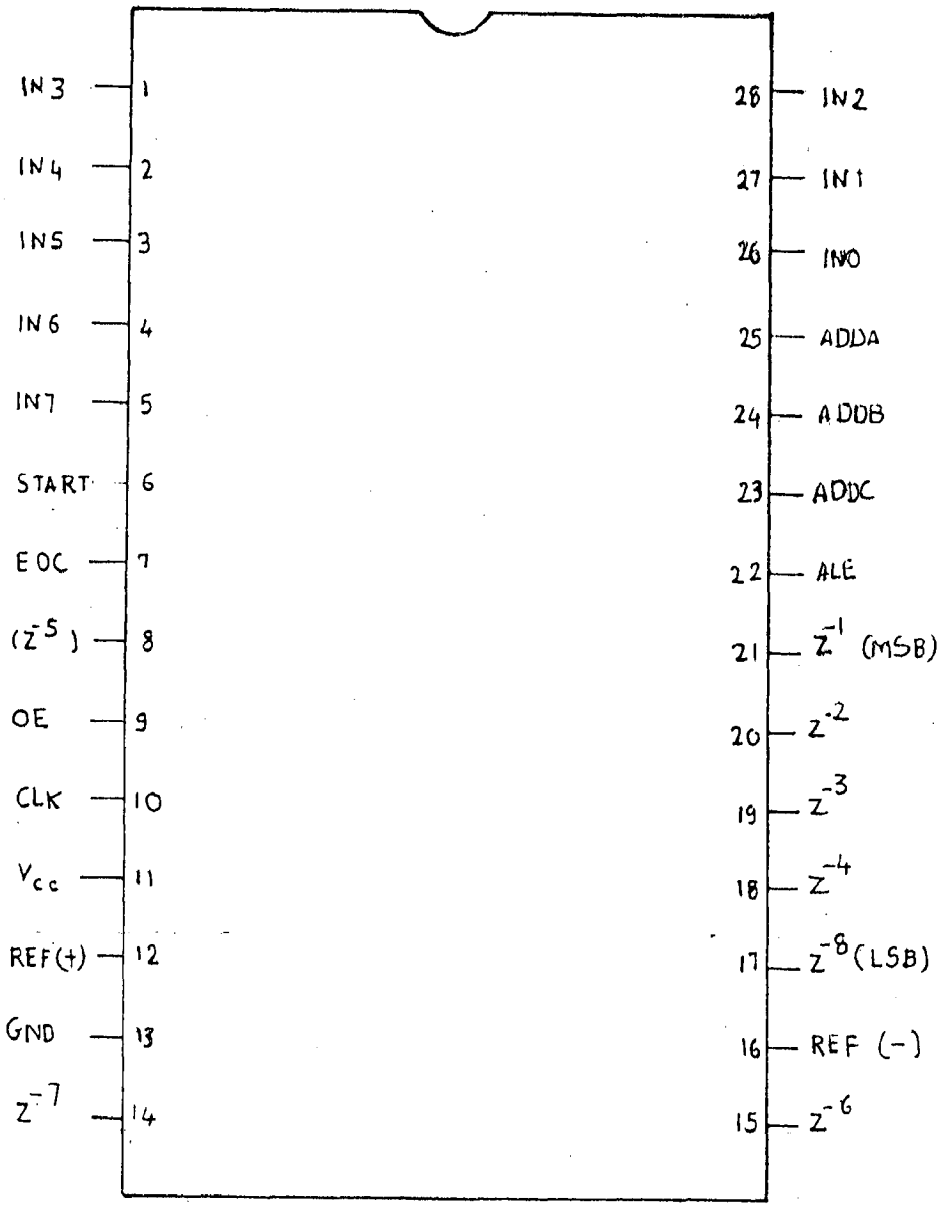


FIG.B1 0809 (ADC) PIN CONNECTION

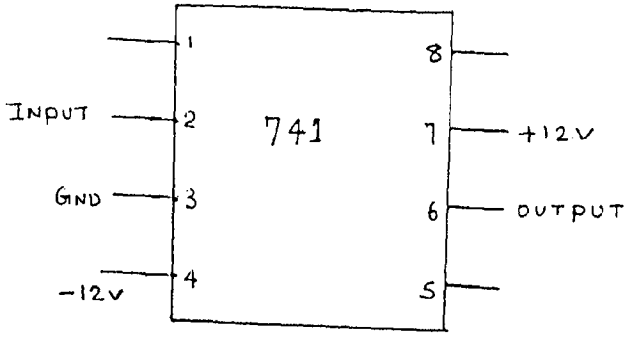


FIG B2 741 I.C (AS COMPARATOR)

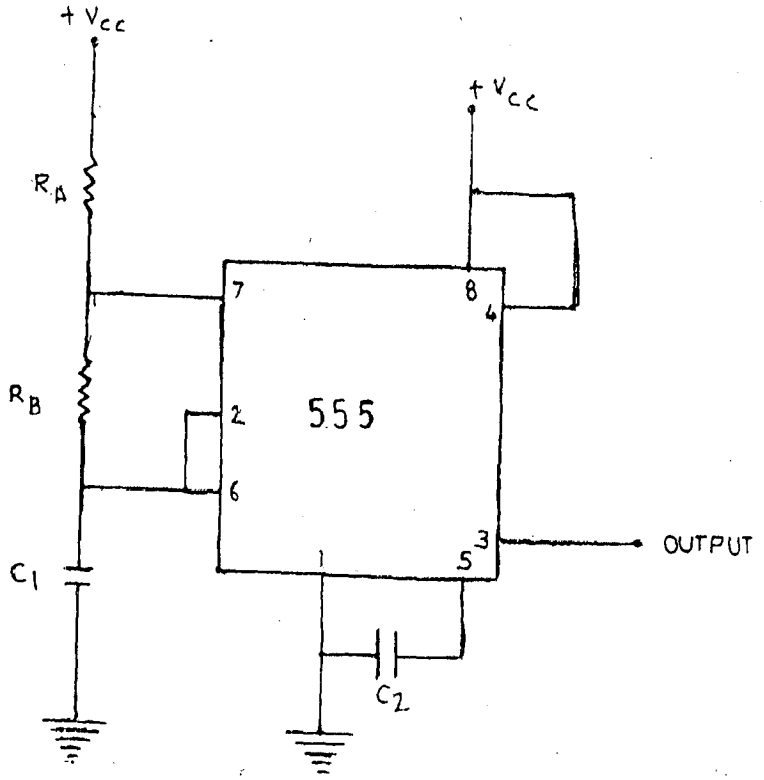


FIG. B3 555 TIMER (AS AN OSCILLATOR)
PIN CONNECTION DIAGRAM

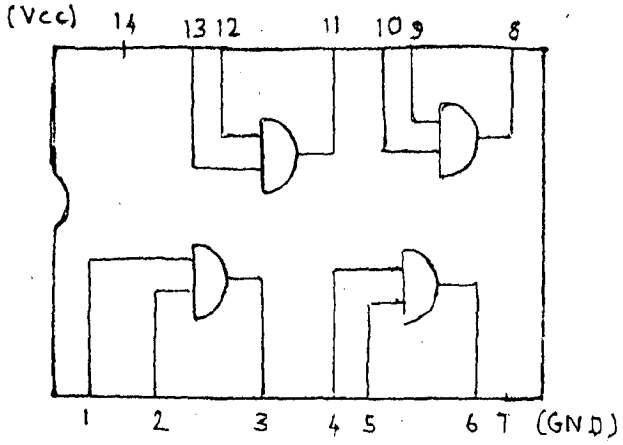


FIG. B4 7408 (QUAD 2-INPUT NAND GATE)
PIN CONFIGURATION

APPENDIX-C

DESIGN OF PI CONTROLLER

Considering the steady state operation only, integral control system seems preferable to proportional systems. When a controller generates only a signal proportional to error the steady state error or offset remains as a result of sustained disturbance. The controllers which generate integral control in addition are able to regulate without leaving a steady state error. It is possible to combine the basic features of proportional controller and an integral controller to form a proportional plus integral controller. The action of a proportional plus integral controller in response to change in the input of external disturbance is initially similar to that of proportional controller, but as the new equilibrium point is reached, the control action becomes as that of an integral controller. This general purpose nature of the control system makes it applicable to numerous motor drive systems. Although the proportional and integral control algorithm is easily implemented, the control co-efficients must be carefully selected to assure that the operating system is stable.

In the present work a proportional plus integral controller has been designed and implemented through software for the current controlling purpose. The equation for the PI controller is

$$\text{Output Voltage } V = \left(K_p + \frac{K_i}{s}\right)(I_{ref} - I) \qquad .. (C1)$$

Where K_p = Proportional controller constant,

K_i = Integral controller constant,

I_{ref} = Reference current in Amps,

I = Actual current in Amps,

$S = \frac{d}{dt}\ Sec^{-1}$.

$$V_n = K_p e_n + K_i \int_{-\infty}^{nT} e.dt \text{ ---- at the } n^{th} \text{ sampling instant } (t = nT) \quad .. (C2)$$

e_n = Error in reference current and actual current at the n^{th} instant.

$e_n = (I_{ref} - I)$ in amps.

$$V_{n-1} = K_p . e_{n-1} + K_i \int_{-\infty}^{(n-1)T} e.dt \text{ - at the } (n-1)^{th} \text{ sampling instant} \quad .. (C3)$$

$e_{n-1} = (I_{ref} - I_{n-1})$ in amps.

$$V_n - V_{n-1} = K_p (e_n - e_{n-1}) + K_i \int_{(n-1)T}^{nT} e.dt \quad .. (C4)$$

$$\Delta V_n \approx K_p \Delta e_n + K_i . T . e_n \quad .. (C5)$$

$K_i T$ can be expressed as K_i'

$$\Delta V_n = (K_p . \Delta e_n + K_i' e_n) \quad .. (C6)$$

$$\therefore V_n = V_{n-1} + K_p [e_n - e_{n-1}] + K_i' e_n \quad .. (C7)$$

$$V_n = V_{n-1} + \Delta V_n \quad .. (C8)$$

As the bit length of 8085 microprocessor is 8 bits, the maximum value of d.c. output voltage is digitized form

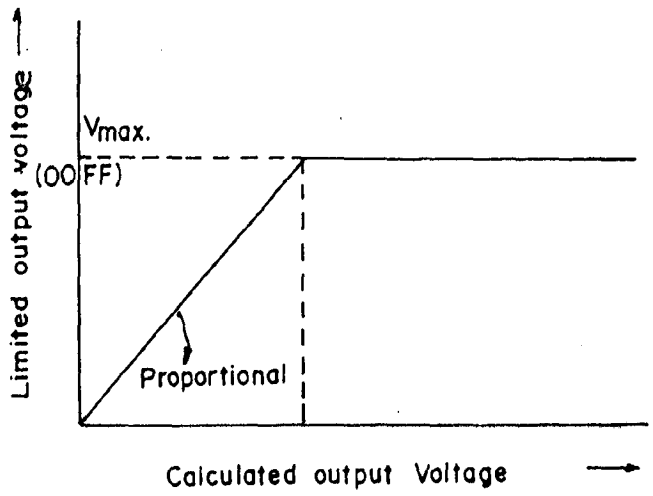


FIG. C-1 LIMITING THE OUTPUT VOLTAGE.

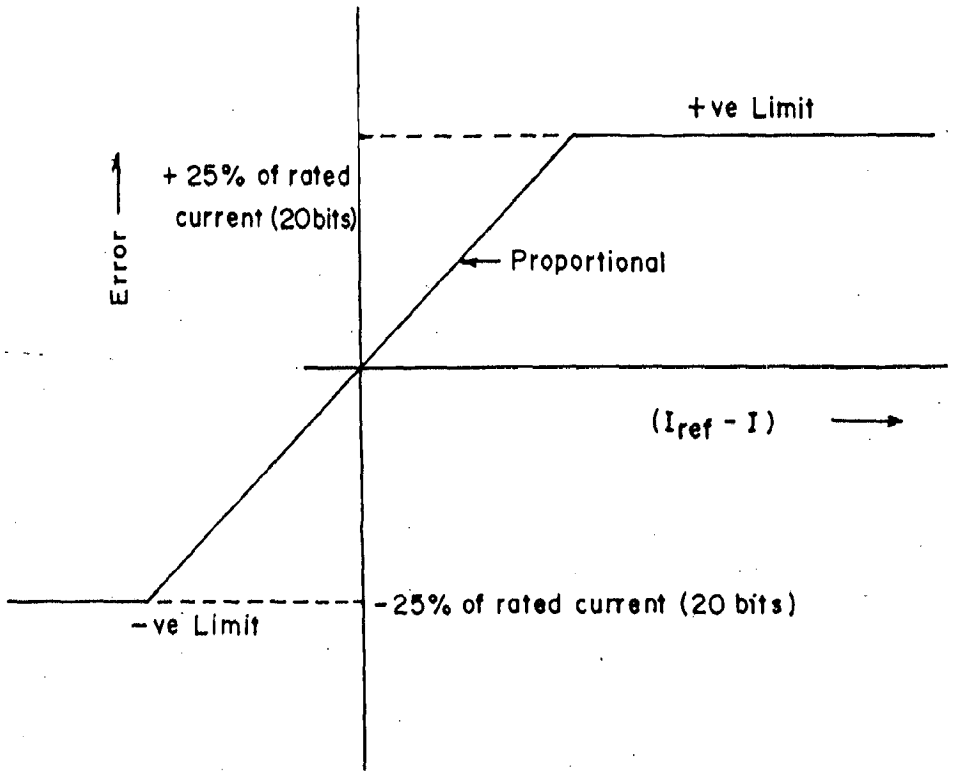


FIG. C-2 LIMITING THE ERROR.

i.e. V_{dn} is limited to 256 bits of digital scale. Expressing the value of ΔV_n in digital form.

$$\frac{V_{dn\ max}}{256}\ \Delta V_n\ (\text{Bits}) = [\ K_p.\Delta e_n + K_i'.e_n] \quad .. (C9)$$

Where e_n and Δe_n are in amps.

The current scale calculated in digital form 1 amps = 8 bits

$$\therefore\ \Delta V_n = [K_p.\Delta e_n + K_i'.e_n]\ \frac{8 \times 256}{V_{dn}max} \quad .. (C10)$$

where ΔV_n, e_n and Δe_n are in bits. If the maximum value of d.c. output voltage V_{max} is limited to 25% of current error as shown in Fig. (C-1).

Considering only the proportional controller

$$V_{max}\ \text{Volts} = K_p.\Delta e_n \quad .. (C11)$$

As the maximum speed of the drive is 1500 rpm and the rated current of 4.6 Amps.

25% of full load current error = [25/100] x 4.6 = 1.15 Amps.

Maximum value of D.C. output voltage is limited to 220V

$$220 = K_p\ [1.15]$$

$$\therefore\ K_p = \frac{220}{1.15} \approx 191$$

As an integral controller keeps the steady state error to minimum possible value, it is also incorporated in the controller. For accomadating the integral constant, the

proportional controller constant K_p must be kept low. From equation (C10), with ΔV_n, Δe_n and e_n in bits

$$\Delta V_n = [K_p . \Delta e_n + K_i' . e_n] \quad \frac{8x256}{220} \qquad \ldots (C12)$$

$$\Delta V_n = [K_p . \Delta e_n + K_i' . e_n] \qquad .. (C13)$$

$$\text{or} \quad \Delta V_n(\text{Hexa}) = C_p . \Delta e_n(\text{Hexa}) + C_i . e_n(\text{Hexa}) \qquad .. (C14)$$

Where C_p and C_i are in the order of 0-10 (Binary) or OAH. The equation (C14) is implemented through software as a PI controller algorithm.

Thus the error is limited between the limits before the PI control processing. If the reference current is greater than the actual current, then the error will be positive, and if the reference current is less than the actual current, then the error will be negative. For this the error is limited on either side as shown in Fig.(C2). The error is limited to +25% of rated current, when the error is positive, and the error is limited to -25% of rated current, when the error is negative.

$$25\% \text{ of maximum current} = \frac{25}{100} \times 4.6 = 1.15 \text{ Amps.} = 20 \text{ bits}$$

So the error is limited to 20 bits on either side. The complete PI control algorithm is processed in 2's complement form, since the error may have positive or negative values. The calculated value V is limited to V_{max} if it is greater than V_{max}, and to zero if its calculated value is negative.

The maximum d.c. voltage output is limited for a given 3 phase a.c. input with firing angle at zero degrees.

Thus the designed controller has excellent flexibility, and the optimum values for C_p and C_i can be chosen for a satisfactory system response experimentally.

APPENDIX D

```
5 REM self-tunning controller for dc motor cotrol
20 DIM X(6),P(6,6),SITA(6),PXT(6),AK(6),AKT(6),AKXPX(6),AKXK(6,6),PK(6,6)
25 OPEN "WN200" FOR INPUT AS #1
30 BETA=.98:AY2=0!:AY1=0!:W2=0!:W1=0!:U1=0!:U2=0!:
40 FOR I=1 TO 6:P(I,I)=100!:SITA(I)=.001:NEXT I
50 A=10
60 FOR L=1 TO 200
70 T = L
80 AX = A
90 GOSUB 600
100 W=10*AW
110 AS=-1
120 AL=(200/A)+1
125 FOR JL=1 TO AL
130 AX=AX+A
140 GOSUB 600
150 IF AX<=200 THEN 170
160 IF AX>200 THEN 185
170 W=W+10*AS*AW
180 AS=-AS
185 NEXT JL
190 INPUT #1,UN
200 YW=1.5*AY1-.7*AY2+U1+.5*U2+UN
210 GOSUB 250
220 PRINT L,W,YW,U,SITA(1)
225 PRINT SITA(2),SITA(3),SITA(4),SITA(5),SITA(6)
230 NEXT L
240 IF EOF(1) THEN CLOSE #1
245 REM COTROLLER
250 FAI=YW-W1+.5*U1
260 X(1)=AY1:X(2)=AY2:X(3)=U1:X(4)=U2:X(5)=W1:X(6)=W2
270 FOR I= 1 TO 6:PXT(I)=0!:FOR K= 1 TO 6
280 PXT(I)=PXT(I)+P(I,K)*X(I):AKXK(I,K)=0!
290 NEXT K,I
300 XPXT=0!:FOR K= 1 TO 6
310 XPXT=XPXT+X(K)*PXT(K)
320 NEXT K
330 XPX=1!+XPXT:XPXIN=1!/XPX
340 FOR I= 1 TO 6:AK(I)=PXT(I)*XPXIN:NEXT I
350 FOR I= 1 TO 6:AKXPX(I)=AK(I)*XPX:NEXT I
360 FOR I= 1 TO 6:FOR J= 1 TO 6
370 AKXK(I,J)=AKXK(I,J)+AKXPX(I)*AK(J)
380 NEXT J,I
390 FOR I= 1 TO 6:FOR J= 1 TO 6
400 PK(I,J)=P(I,J)-AKXK(I,J)
410 P(I,J)=PK(I,J)/BETA
420 NEXT J,I
430 S=0!
440 FOR I=1 TO 6:S=S-X(I)*SITA(I):NEXT I
450 FXTH=FAI+S
460 FOR I= 1 TO 6:SITA(I)=SITA(I)+AK(I)*FXTH:NEXT I
470 U=-(SITA(1)*YW+SITA(2)*AY1+SITA(4)*U1+SITA(5)*W+SITA(6)*W1)/SITA(3)
500 AY2=AY1:AY1=YW:U2=U1:U1=U:W2=W1:W1=W
510 RETURN
600 IF T>AX THEN AW=1
610 IF T<=AX THEN AW=0
620 RETURN
```

1	0	.046312	-.046312	.001
.001	.001	.001	.001	.001
2	0	-.009133	-8.962501E-03	-.1051327
.001	.1071327	.001	.001	.001
3	0	-.0560984	-2.542441E-02	-6.577437E-02
-.19858	.1457563	.2005801	.001	.001
4	0	-.1713942	.2524076	.7002136
-6.486835E-02	.4929877	.3170308	.001	.001
5	0	6.362604E-02	-.134325	.4681342
-.1947735	.9676786	.2647103	.001	.001
6	0	.1480407	4.891717E-03	.4964408
-.3206402	.8952897	.427857	.001	.001
7	0	.0422181	1.250662E-02	.591459
-.2545559	.8990299	.3006209	.001	.001
8	0	-2.045189E-02	2.418479E-02	.5808638
-.3195583	.894776	.2983171	.001	.001
9	0	-5.990741E-02	3.011107E-02	.6227276
-.3482514	.8124193	.2365447	.001	.001
10	0	-.1365673	.173416	.9046934
-.3214852	.6039417	-1.464433E-02	.001	.001
11	10	.1096528	-.1232258	.5644613
-.361092	1.17333	.1347999	.001	.001
12	10	.2007205	8.297165	.5583221
-.3592802	1.178485	.1199615	-.9845974	.001
13	10	8.482436	1.719219	.560635
-.3589733	1.25163	.1178056	-.9874096	.2061321
14	10	18.39602	-7.002163	1.017934
-.3563188	1.261034	.8205232	-.9867544	.1583376
15	10	15.51429	-9.799779E-02	.9264608
-3.028819E-02	1.170775	.8168611	-.9871749	.1890075
16	10	6.805222	4.108249	.893577
-.1805827	1.172304	.8071902	-.9871266	.1854876
17	10	3.462917	.4837133	.8391091
.5525168	-2.38933	.7965697	-.9860614	.1078095
18	10	3.03232	-3.298171	.8877452
-.9397239	-2.028201	1.033686	-.9890582	.3263478
19	10	-.9766312	5.562347	.8357604
-.4421621	2.218232	1.002655	-.988714	.30125
20	10	.2868398	8.450682E-02	.8098664
-1.166393	1.111934	.8203325	-.9898909	.3870531
21	0	3.992002	-11.16665	.7753781
-1.28682	1.077986	-1.289644	-.9975032	.9420276
22	0	-5.380321	-2.957633	.6965627
-1.278659	1.862378	-1.286005	-.9975032	.7493881
23	0	-19.44428	-2.872797	.1581535
-1.76859	1.450788	.7683785	-.9975032	.7493881
24	0	-29.75796	5.408876	2.002086
-2.598889	1.861657	.3567991	-.9975032	.7493881
25	0	-27.01397	15.6796	.2690768
1.526621	3.030169	.9588192	-.9975032	.7493881
26	0	-1.252839	6.236061	.3861332
1.259407	3.148697	.9483588	-.9975032	.7493881
27	0	31.00856	-5.029572	.4160233
-4.786074E-02	3.493824	.7396007	-.9975032	.7493881

CONTROLLER PAPRAMETERS OBTAINED FOR DISCRETIZED MODEL OF A D.C. MOTOR

Sl.No.	SAMPLING PERIOD	SYSTEM OUTPUT EQUATION (YW)	CONTROLLER PARAMETERS
1	T = 0.1 Sec	$.9969\ ay_1 - 0.000094\ ay_2 + 0.1769\ u_1$	$\theta_1 = 0.5115$, $\theta_2 = 0.5609$ $\theta_3 = .8143$ $\theta_4 = .1132$ $\theta_5 = -.9546$, $\theta_6 = -.1162$
2	T = .05 Sec	$1.0082\ ay_1 - 0.0097\ ay_2 + 0.1755\ u_1$	$\theta_1 = 0.5114$, $\theta_2 = 0.5571$ $\theta_3 = 0.8142$, $\theta_4 = 0.1170$ $\theta_5 = -0.9543$, $\theta_6 = -0.1166$
3	T = .075 Sec	$.9985\ ay_1 - .00096\ ay_2 + 0.1768\ u_1$	$\theta_1 = 0.5126$,, $\theta_2 = 0.5808$ $\theta_3 = 0.6205$, $\theta_4 = 0.1835$ $\theta_5 = -.9548$, $\theta_6 = -0.1165$
4	T = .025 Sec	$1.0979\ ay_1 - 0.0986\ ay_2 + 0.1598\ u_1$	$\theta_1 = 0.4999$, $\theta_2 = 0.4186$ $\theta_3 = 0.6253$ $\theta_4 = 0.1000$ $\theta_5 = -.9472$, $\theta_6 = -0.1141$

APPENDIX-E

MAIN PROGRAM

2050	31,00,27	LXIH, 2700	Initialize Stack Pointer
2053	3E,8A	MVIA, 8A	
2055	D3,03	OUT 03	Initialize 8255(1) in mode zero
2057	3E,00	MVI A,00	
2059	D3,00	OUT 00	Set port A bits low
205B	3E,00	MVI A,00	
205D	D3,02	OUT 02	Set port C bits low and store the status 2007
205F	32,07,20	STA 2007	
2062	3E,00	MVI A,00	Set the index S = 0
2064	32,01,20	STA 2001	and store it at 2001
2067	3E,70	MVI A,70	Initialize TM_1 in
2069	D3,13	OUT 13	'mode' 0.
206B	3E,16	MVI A,16	
206D	D3,28	OUT 28	Initialization control word ICW_1
206F	3E,2B	MVI A,2B	Initialization control
2071	D3,29	OUT 29	word ICW_2
2073	3E,FC	MVI A,FC	
2075	D3,29	OUT 29	Operational control word OCW_1
2077	32,00,20	STA 2000	Store mask address at 2000.
207D	22,04,20	SHLD 2004	Load 60^{o} count in HL Pair.
2080	CD,00,29	CALL CLI	Call closed loop initialization.
2083	3A,07,20	LDA 2007	

2086	E6,F3		ANI F3	
2088	F6,00		ORI 00	
208A	D3,02		OUT 02	Select IN_o Channel
208C	32,07,20		STA 2007	
208F	CD,00,28		CALL ADC	
2092	32,85,23		STA $I^*_{dc}(n)$	$I_{dc}=I^*_{dc}$
2095	32,86,23		STA $I^*_{dc(n-1)}$	$I_{dc}=I^*_{dc(n-1)}$
2098	32,87,23		STA $I_{dc}(n)$	I_{dc}
2098	3A,07,20	BACK	LDA 2007	
209E	F6,00		ORI 00	
20A0	D3,02		OUT 02	
20A2	32,07,20		STA 2007	Select IN_o channel
20A5	CD,00,28		CALL ADC	and store actual current
20A8	32,87,23		STA 2387	I_{dc}.
20AB	32,F4,27		STA 27F4	
20AE	3A,07,20		LDA 2007	
20B1	E6,F3		ANI F3	
20B3	F6,06		ORI 06	
20B5	D3,02		OUT 02	
20B7	32,07,20		STA 2007	Select IN_1 channel and store reference current
20BA	CD,00,28		CALL ADC	
20BD	32,85,23		STA 2385	
20C0	32,F5,27		STA 27F5	
20C3	CD,70,2A		CALL CCI SR	Call constrained current Subroutine
20C6	CD,30,2B		CALL PI	Call PI Subroutine
20C9	CD,70,2B		CALL $V^*_{dc}(n)$SR	Call constrained voltage subroutine
20CC	3A,8A,23		LDA $V^*_{dc}(n)$	
20CF	CD,A0,2B		CALL 'LOOK'	

20D2	32,10,20	STA ∝add	
20D5	32,F6,27	STA 27F6	
20D8	CD,25,23	CALL MODE	Call mode selection subroutine
20DB	CD,00,23	CALL MULT	Call mult subroutine
20DE	CD,E3,06	CALL MOD AD	Call for display
20E1	CD,FA,06	CALL MODDT	
20E4	3A,01,20	LDA 2001	
20E7	FE,00	CPI 00	
20E9	C2,9B,20	JNZ BACK	
20EC	FB	EI	
20ED	3C	INR C	
20EE	32,01,20	STA 2001	
20F1	C3,9B,20	JMP BACK	

IR_0 INTERRUPT

2100	F5	PUSH PSW	SAVE REGISTERS
2101	E5	PUSH H	
2102	2A,02,20	LHLD $\propto_{count}$	Load HL with $\propto_{count}$ address and output to Timer TM_1
2105	7D	MOVA,L	
2106	D3,11	OUT 11	
2108	7C	MOV A,H	
2109	D3,11	OUT 11	
210B	3A,15,20	LDA S.M.I. add.	SMI is stored
210E	32,14,20	STA 2014	

2111	3E,00	MVIA,00	Set M_3 index = 00
2113	32,08,20	STA 2008	
2116	F3	DI	Disable of Interrupt
2117	3E,60	MVIA,60	Specific EOI
2119	D3,28	OUT 28	
211B	E1	POP H	
211C	F1	POP PSW	Recover Registers
211D	FB	EI	Enable of Interrupt
211E	C9	RETURN	

IR_1 SUBROUTINE

2200	F5	PUSH PSW	Save all registers
2201	E5	PUSH H	
2202	C5	PUSH B	
2203	3A,08,20	LDA M_3 Index	Load M_3 index address
2206	FE,06	CPI 06	
2208	CA,57,22	JZ LOOP1	(A) = M_3
220B	2A,04,20	LHLD 60°C add	HL 60° count address
220E	7D	MOV A,L	LSB of 60° count
220F	D3,11	OUT 11	
2211	7C	MOV A,H	MSB of 60° count
2212	D3,11	OUT 11	
2214	3A,08,20	LDA M_3 index	~~(A)←M_3 Index~~
2217	3C	INR A	$M_3 = M_3+1$ & Store it
2218	32,08,20	STA 2008	
221B	3A,14,20	LDA M_2 Index	(A)←M_2 index address

221E	FE,00		CPI 00	Compare immediate with 00
2220	CA,5F,22		JZ_o	(M_2) = 00,Jump to JZ_o
2223	FE,01		CPI 01	
2225	CA,6C,22		JZ_1	M_2=01,Jump to JZ_1
2228	FE,02		CPI 02	
222A	CA,79,22		JZ_2	M_2=02,jump to JZ_2
222D	FE,03		CPI 03	
222F	CA,86,22		JZ_3	M_2=03,Jump to JZ_3
2232	FE,04		CPI 04	M_2=04,Jump to JZ_4
2234	CA,93,22		JZ4	
2237	FE,05		CPI 05	
2239	CA,A0,22		JZ_5	M_2=05,Jump to JZ_5
223C	3E,00	BACK	MVIA,00	Set firing bits as zero
223E	D3,00		OUT 00	
2240	3A,14,20		LDA M_2 index	
2243	FE,06		CPI 06	
2245	DA,4D,22		JC SAVE	M_2<06,Jump to save
2248	3E,00		MVIA,00	
224A	32,14,20		STA 2014	
224D	F3	SAVE	DI	Disable of interrupt
224E	3E,61		MVIA,61	Specific EOI of IR_1
2250	D3,28		OUT 28	
2252	C1		POP B	Recover Registers
2253	E1		POP H	
2254	F1		POP PSW	
2255	FB		EI	
2256	C9		RETURN	

2257	3E,00	LOOP1	MVIA,00	Set M_3 index = 00 & store it
2259	32,08,20		STA 2008	
225C	C3,4D,22		JMP SAVE	
225F	3A,20,20	JZ_0	LDA 2020	FC=00 Ist firing command
2262	D3,00		OUT 00	
2264	3E,01		MVIA,01	Set M_2=01
2266	32,14,20		STA 2014	Store M_2 index at 2014
2269	C3,3C,22		JMP BACK	
226C	3A,21,20	JZ1	LDA 2021	FC=01,2nd Firing Command
226F	D3,00		OUT 00	
2271	3E,02		MVIA,02	M_2=02
2273	32,14,20		STA 2014	Store M_2 index at 2014
2276	C3,3C,22		JMP BACK	
2279	3A,22,20	JZ2	LDA 2022	FC=02,3rd firing command
227C	D3,00		OUT 00	
227E	3E,03		MVIA,03	M_2=03
2280	32,14,20		STA 2014	Store M_2 index at 2014
2283	C3,3C,22		JMP BACK	
2286	3A,23,20	JZ3	LDA 2023	
2289	D3,00		OUT,00	Issue 4th firing command
228B	3E,04		MVIA,04	M_2=04
228D	32,14,20		STA 2014	Store M_2 index at 2014.
2290	C3,3C,22		JMP BACK	
2293	3A,24,20	JZ4	LDA 2024	FC=04,issue 5th firing command
2296	D3,00		OUT 00	
2298	3E,05		MVIA,05	M_2=05

229A	32,14,20		STA 2014	Store M_2 index at 2014
229D	C3,3C,22		JMP BACK	
22A0	3A,25,20	JZ5	LDA 2025	FC=06,issue 6th firing command
22A3	D3,00		OUT 00	
22A5	3E,06		MVIA,06	M_2=06
22A7	32,14,20		STA 2014	Store M_2 index at 2014.
20AA	C3,3C,22		JMP BACK	

ADC SUBROUTINE

2800	3E,00		MVIA,00	Set PC_o low
2802	D3,03		OUT 03	
2804	3C		INR A	Set PC_o high
2805	D3,03		OUT 03	
2807	3D		DCR A	Set PC_o low
2808	D3,03		OUT 03	
280A	DB,02	LOOP	IN 02	In Port 'C' Status
280C	E6,40		ANI 40	Check SOC is over
280E	CA,0A,28		JZ LOOP	
2811	DB,01		IN 01	In port 'B' signal
2813	C9		RETURN	

MODE SELECTION SUBROUTINE

2325	3A,10,20		LDA 2010	(A) $\propto_f$ address
2328	06,3C		MVIB,3CH	(B) 60^o Value
232A	0E,78		MVIC,78H	(C) 120^o Value
232C	B9		CMP C	

232D	D2,3D,23		JMC LOOP1	$\alpha_f > 120^\circ$, jump to loop1
2330	FE,3D		CPI 3D	
2332	D2,49,23		JNC Loop2	$\alpha_f > 60^\circ$ Jump to loop2
2335	3E,00		MVIA,00	Set S.M.I.=00 & Store it
2337	32,15,20		STA 2015	
233A	C3,52,23		JMP RETURN	
233D	91	LOOP1	SUB C	(A)←(A)-120° & Store it.
233E	32,10,20		STA 2010	
2341	3E,04		MVIA, 04	Set SMI=04 & Store it.
2343	32,15,20		STA 2015	
2346	C3,52,23		JMP RETURN	
2349	90	LOOP2	SUB B	(A)←(A)-60° & Store it.
234A	32,10,20		STA 2010	
234D	3E,05		MVIA,05	Set SMI=05 & Store it.
234F	32,15,20		STA 2015	
2352	C9		RETURN	

MULTIPLICATION SUB ROUTINE

2300	3A,10,20		LDA 2010	(A) ← α_f
2303	6F		MOVL,A	(L) ← α_f
2304	3E,00		MVI A,00	(A) ← 00
2306	67		MOV H,A	(H) ← 00
2307	EB		XCHG	(HL) ← (DE)
2308	3A,55,25		LDA 2555	(A) ← equivalent 1°count
230B	21,00,00		LXIH,0000	Set product in (HL)=00
230E	0E,08		MVI C,08	Set Count in C=08
2310	29	LOOP	DAD H	(HL) ← (HL) + (HL)

2311	17	RAL	Rotate ACC. left
2312	D2,16,23	JNC AHEAD	Is multiplication is over go ahead
2315	19	DAD D	(HL) ← (HL) + (DE)
2316	0D AHEAD	DCR C	(C) ← (08) - 1
2317	C2,10,23	JNZ LOOP	Is C=00,No jump to loop
231A	22,02,20	SHLD 2002	Store $\propto_{count}$ at 2002 and 2003
231D	C9	RETURN	

CLOSED LOOP INITIALIZATION OF VARIABLES

2900	3E,00	MVI A,00	A ← 00
2902	32,88,23	STA CCE	(A) ← CCE address
2905	32,8C,23	STA e_n add	e_n ← 238C
2908	32,8D,23	STA e_{n-1}	e_{n-1} ← 238D
290B	32,8E,23	STA e_n add	e_n ← 238E
290E	32,9A,23	STA e_{n-1} add	e_{n-1} ← 239A
2911	3E,10	MVI A,I_{dc} max	Max.error = 10% of rated current
2913	32,89,23	STA 2389	
2916	3A,0F,20	LDA V^*_{dc} add	
2919	32,8A,23	STA V^*_{dc}(n) add	
291C	32,8B,23	STA V^*_{dc}(n-1) add	
291F	3E,01	MVI A,K_p	
2921	32,8F,23	STA 238F	Store proportional controller constant
2924	3E,01	MVI A,$K_I T_S$	
292C	32,90,23	STA 2390	
2929	3E,FF	MVI A + V_{dc}(max)	

292B	32,93,23		STA 2393	
292E	3E,78		MVI A,+ I_{dc}(max)	
2930	32,A6,23		STA 23A6	
2933	21,00,24		LXIH,2400	'm' pointer address
2936	22,11,20		SHLD,2011	
2939	3E,55		MVI A,55	1^{o} = 55 counts
293B	32,55,25		STA 2555	
293E	C9		RETURN	

CONSTRAINT CURRENT ERROR SR

2A70	3A,87,23		LDA I^{*}_{dc}(n)add	Load with reference current address.
2A73	21,85,23		LXIH,I_{dc}(n)add	HL has I_{dc} (n) address
2A76	BE		CMP M	
2A77	DA,86,2A		JC LP1	$I^{*}_{dc} < I_{dc}$ the jump LP1,
2A7A	96		SUB M	(A) ← (A)-(M)
2A7B 2A	21,89,23		LXIH,Max.add	I_{dc} max.add.at 2389
2A7E	BE		CMP M	
2A7F	DA,95,2A		JC STORE	I_{dc}max.<M,Jump to store
2A82	7E		MOV A,M	(A)←I_{dc}(max)
2A83	C3,95,2A		JMP STORE	
2A86	7E	LP1	MOV A,M	
2A87	21,87,23		LXIH,I^{*}_{dc}(n)add	
2A8A	96		SUB M	(A) ← (A)-(M)
2A8B	21,89,23		LXIH,Max.add	I_{dc}max.add.is in HL pair
2A8E	BE		CMP M	
2A8F	DA,93,2A		JC LP2	

2A92	7E		MOV A,M	(A) ← I_{dc}(max.)
2A93	2F	LP2	CMA	Complement accumulator
2A94	3C		INR A	Increment Acc.
2A95	32,88,23	STORE	STA C.C.E.	
2A98	C9		RETURN	

PI CONTROLLER SUBROUTINE

2B30	3A,8A,23	LDA V^*_{dc}(n)	A←V^*_{dc}(n)add.
2B33	32,8B,23	STA V^*_{dc}(n-1)	V^*_{dc}=V^*_{dc}(n-1)
2B36	3A,8C,23	LDA $e_n$add.	
2B39	32,8D,23	STA $e_n$add.	
2B3C	47	MOV B,A	
2B3D	3A,88,23	LDA CCE add.	Load CCE add.from CCE Subroutine
2B40	32,8C,23	STA $e_n$add.	
2B43	90	SUB B	$\Delta e_n = e_n - e_{n-1}$
2B44	32,8E,23	STA $e_n$add.	
2B47	47	MOV B,A	B ← Δe_n
2B48	3A,8F,23	LDA $K_p$add.	A ← K_p
2B4B	6F	MOV L,A	L ← K_p add.
2B4C	CD,A02A	CALL SMULT	
2B4F	22,91,23	SHLD PR RESULT	Store the partial result at 2391.
2B52	3A,8C,23	LDA e_n add.	
2B55	47	MOV B,A	
2B56	3A,90,23	LDA K_I T_S add.	A ← K_I T_S
2B59	6F	MOV L,A	L ← K_I T_S

2B5A	CD,A0,2A	CALL SMULT	
2B5D	EB	XCHG	(DE) ← (HL)
2B5E	2A,91,23	LHLD PR RESULT	Store PR Result at 2391
2B61	19	DAD D	$K_p\Delta e_n+K_IT_S= V^*_{dc}(n)$
2B62	EB	XCHG	(DE) ← (HL)
2B63	32,8B,23	LDA $V^*_{dc}(n-1)$add.	
2B66	6F	MOV L,A	(L) ← (A)
2B67	26,00	MVIH,00	(H) ← (00)
2B69	19	DAD D	HL ← (HL) + (DE)
2B6A	22,91,23	SHLD $V^*_{dc}(n)$add	
2B6D	C9	RETURN	

SMULT SUBROUTINE

2AA0	26,00	MVIH,00	Clear MS Byte of product
2AA2	0E,08	MVI C,08	Set Counter in register C = 08
2AA4	AF	XRA A	Clear accumulator set X_i=0
2AA5	70	MOV A,L	
2AA6	DA,B2,2A LP1	JC LOOP2	Check X_{i-1}
2AA9	0F	RRC	Rotate right 'A' with carry X_{i-1}=0.
2AAA	7C	MOV A,H	A ← H
2AAB	D2,B8,2A	JNC LP3	Check X_i
2AAE	90	SUB B	(A) ← (H)-(B)
2AAF	C3,B8,2A	JMP LP3	
2AB2	0F LP2	RRC	X_{i-1}=1 place X_i in carry
2AB3	7C	MOV A,H	Move partial product in ACC.

2AB4	DA,B8,2A		JC LP3	Check X_i
2AB7	80		ADD B	(A) ← (H)+(B)
2AB8	67	LP3	MOV H,A	(H) ← (A)
2AB9	17		RAL	
2ABA	7C		MOV A,H	Load MSB of PP
2ABB	1F		RAR	Arithmatic right shift
2ABC	67		MOV H,A	Save Shifted MSB
2ABD	7D		MOV A,L	Load LSB of PP
2ABE	1F		RAR	Arithmatic right shift
2ABF	6F		MOV L,A	Save shifted LSB
2AC0	0D		DCR C	Check for completion
2AC1	C2,A6,2A		JNZ LP1	
2AC4	C9		RETURN	

CONSTRAINED V^*_{dc} (n) SUBROUTINE

2B70	2A,91,23		LHLD V^*_{dc}(n) add	
2B73	44		MOV B,H	(HL) ← (BC)
2B74	4D		MOV C,L	
2B75	3A, 93,23		LDA V_{dc}(max.)	
2B78	57		MOV D,A	D ← V_{dc}(max)add.
2B79	29		DAD H	(HL) - (HL)+(HL)
2B7A	DA,92,2B		JC LP1	If V^*_{dc} is -ve jump to LP1.
2B7D	78	LP3	MOV A,B	
2B7E	B7		ORA A	Is MSB of V^*_{dc}=00
2B7F	CA,85,2A		JZ LP2	Yes,go to LP2.
2B82	C3,8A,2B	LP4	JMP Prestore	

2B85	79	MOV A,C	MSB of V^*_{dc}=00
2B86	BA	CMP D	Is LSB $\geq$ max.value
2B87	DA,8B,2B	JC Store	Yes,Jump to Store
2B8A	7A PRE STORE	MOV A,D	
2B8B	32,8A,23 STORE	STA V^*_{dc}(n)	
2B8E	CD,40,28	CALL V^*_{dc}add	
2B91	C9	RETURN	
2B92	3E,00	MVI A,00	Set V^*_{dc}=00 and
2B94	C3,8B,2B	JMP STORE	Jump to store it.

V^*_{dc}(n) SUBROUTINE

2840	3A,8B,23	LDA V_{dn-1}add.	
2843	57	MOV D,A	
2844	3A,8A,23	LDA	V_{dn}add
2847	BA	CMP D	
2848	DA,60,28	JC LP1	$V_{dn}<V_{dn-1}$, yes Jump to loop1.
284B	CA,60,28	JZ LP1	$V_{dn}=V_{dn-1}$ Jump to LP1.
284E	92	SUB D	$V_{dn}-V_{dn-1} = V_{dn}$
284F	FE,05	CPI 05	
2851	DA,78,28	JC LP2	V_{dn}<05,Jump to LP2
2854	CA,78,28	JZ LP2	V_{dn}=05 Jump to LP2
2857	3A,8B,23	LDA V_{dn-1}add.	
285A	C6,05	ADI 05	
285C	32,8A,23 STORE	STA V_{dn}add.	
285F	C9	RETURN	

2860	3A,8A,23	LP1	LDA V_{dn}add.	
2863	57		MOV D,A	
2864	3A,8B,23		LDA V_{dn-1}add.	
2867	92		SUB D	$V_{dn-1} - V_{dn} = \Delta V_{dn}$
2868	FE,05		CPI 05	
286A	DA,78,28		JC LP2	$\Delta V_{dn} < 05$, Jump to LP2
286D	CA,78,28		JZ LP2	$\Delta V_{dn} = 05$ Jump to LP2
2870	3A,8B,23		LDA V_{dn-1}add.	
2873	D6,05		SUI 05	
2875	C3,5C,28		JMP STORE	
2878	3A,8A,23	LP2	LDA V_{dn}add.	
287B	C3,5C,28		JMP STORE	

LOOK UP SUBROUTINE

2BA0	2A,11,20		LHLD M POINTER ADD.	
2BA3	BE	LP1	CMPM	
2BA4	DA,AE,2B		JC LP2	If $V_{dc} < V_{dc}(n-1)$ Jumpt to Loop2.
2BA7	CA,B3,2B		JZ LP3	A=M,Jumpt to LP3.
2BAA	2B		DCX H	(HL) ← (HL)-1
2BAB	C3,A3,2B		JMP LP1	
2BAE	23	LP2	INX H	(HL) ← (HL)+1
2BAF	BE		CMP M	
2BB0	DA,AE,2B		JC LP2	A<M Jump to LP2

```
2BB3    22,11,20  LP3     SHLD M POINTER ADD.
2BB6    11,60,00          LXID,0060       (DE) ← 0060
2BB9    19                DAD D           (HL) ← (HL)+(DE)
2BBA    7E                MOV A,M         (A) ← (M)
2BBB    C9                RETURN
```

#######

APPENDIX - F

MICROPROCESSOR AND ITS PERIPHERALS

The complete control scheme for the Adaptive Controller fed d.c. drive has been developed using personal computer (PC) and microprocessor 8085A system. An 8085A is 8 bit INTEL's most popular microprocessor system. In this system, the programmable interrupt controller (PIC) 8259A programmable interval timer (PPI) 8253, and the programmable peripheral interface (PPI) 8255Aare interfaced.

Peripherals are used to facilitate parallel data transfer between microprocessor and input output devices. These devices can act as input port which is tristated buffer to read data from inputting devices. They can act as output port to latch data sent by the microprocessor for output device. These devices can generate an interrupt signal and receive/transmit certain control signals for data communication between microprocessor and input output devices.

The 8255Ais a programmable peripheral interface (PPI). It can be programmed to transfer data under various conditions. It has 24 I/O pins, that can be grouped primarily into 8 bit parallel ports, port A, Port B, and remaining 8 bits as port C. Eight bits of port C can be used individually or be grouped in two 4 bit ports. Port C upper

(PC_u) and port C lower (PC_l). The functions of these ports are defined by writing a control word in control word register, format of which is shown in Fig. B-1. Functions of the 8255A are classified according to two modes: the Bit set/Reset (BSR) mode and the I/O mode. The BSR mode is to set or reset the bits of port C. The I/O mode is further divided into three modes: Mode 0, Mode 1, and Mode 2. In mode 0 all ports function as simple input output ports. In Mode 1 port A and/or port B use bits from port C as handshaking signals that is why this mode is named as hand-shake mode. In handshake mode, two types of input/output data transfer can be implemented: status check and interrupt. In Mode 2, port A can be set for bydirectional data transfer using hand-shake signals from port C, and port B can be set up either in Mode '0' or in Mode '1'.

The 8253 programmable interval timer (PIT) is a support chip to the system. The 8253 includes three identical 16 bit counters that can operate independently in any one of the six modes. Each counter has two input signals CLOCK and GATE and one output signal OUT. The clock input pin is connected to a clock of suitable frequency. To operate a counter, a 16 bit count to be loaded in its register and depending upon the mode of the timer, logic high voltage or low to high voltage transition at GATE initiates or enables counting process. The

different six modes of operation are Mode 0 - interrupt on terminal count, Mode 1 - programmable monoshot, Mode 2 - rate generator or devide by counter, Mode 3 - square wave generater, Mode 4 - software triggered strobe, Mode 5 - Hardware triggered strobe. The control word format of 8253 is shown in Fig. B-2.

The 8259A is a programmable interrupt controller (PIC) designed to work with INTEL microprocessor 8080A, 8085A, 8086 and 8088. The 8259A can manage eight interrupts according to the instructions written into its control word registers. This is equivalent to providing eight interrupt pins on the processor in place of INTER/INT Pin. It can vector an interrupt request anywhere in the memory map. However, all eight interrupts are spaced at the interval of either eight or four locations. It can resolve eight levels of interrupt priorities as a variety of modes, such as fully nested mode, automatic rotation mode and specific rotation mode. It can mask each interrupt request individually. The 8259A can read status of pending, in-service interrupts and masked interrupts. It can accept either the level-triggered or the edge-triggered interrupt request. It can be expanded to 64 priority levels by cascading additional 8259's. To implement interrupts, the interrupt enable flip flop in the microprocessor should be enabled by writing EI instruction and the 8259 should be initialized by writing control words in control word register. The 8259 requires the two types of

control words: Initialization command words (ICWS) and operation command words (OCWS). The 8259 can be initialized with four ICWS; the first two are essential, and the other two are optional. Based on the modes being used, these words must be issued in a given sequence. Once initialized, the 8259 can be setup to operate in various modes by using three different OCWS; however, they no longer need to be issued in a specific sequences.

different six modes of operation are Mode 0 - interrupt on terminal count, Mode 1 - programmable monoshot, Mode 2 - rate generator or devide by counter, Mode 3 - square wave generater, Mode 4 - software triggered strobe, Mode 5 - Hardware triggered strobe. The control word format of 8253 is shown in Fig. B-2.

The 8259A is a programmable interrupt controller (PIC) designed to work with INTEL microprocessor 8080A, 8085A, 8086 and 8088. The 8259A can manage eight interrupts according to the instructions written into its control word registers. This is equivalent to providing eight interrupt pins on the processor in place of INTER/INT Pin. It can vector an interrupt request anywhere in the memory map. However, all eight interrupts are spaced at the interval of either eight or four locations. It can resolve eight levels of interrupt priorities as a variety of modes, such as fully nested mode, automatic rotation mode and specific rotation mode. It can mask each interrupt request individually. The 8259A can read status of pending, in-service interrupts and masked interrupts. It can accept either the level-triggered or the edge-triggered interrupt request. It can be expanded to 64 priority levels by cascading additional 8259's. To implement interrupts, the interrupt enable flip flop in the microprocessor should be enabled by writing EI instruction and the 8259 should be initialized by writing control words in control word register. The 8259 requires the two types of

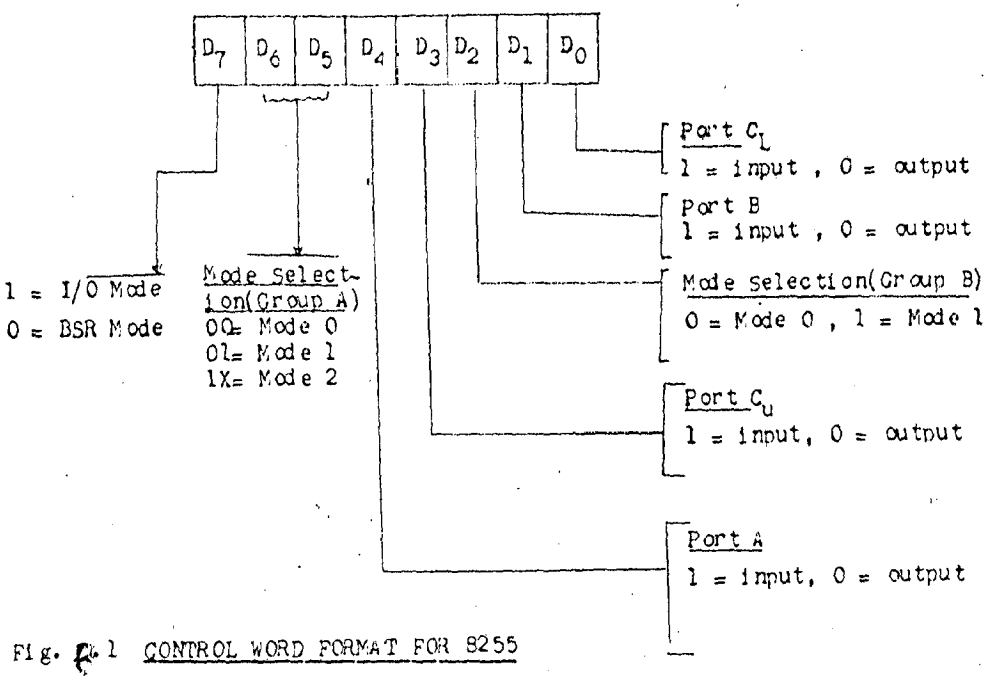


Fig. 5.1 CONTROL WORD FORMAT FOR 8255

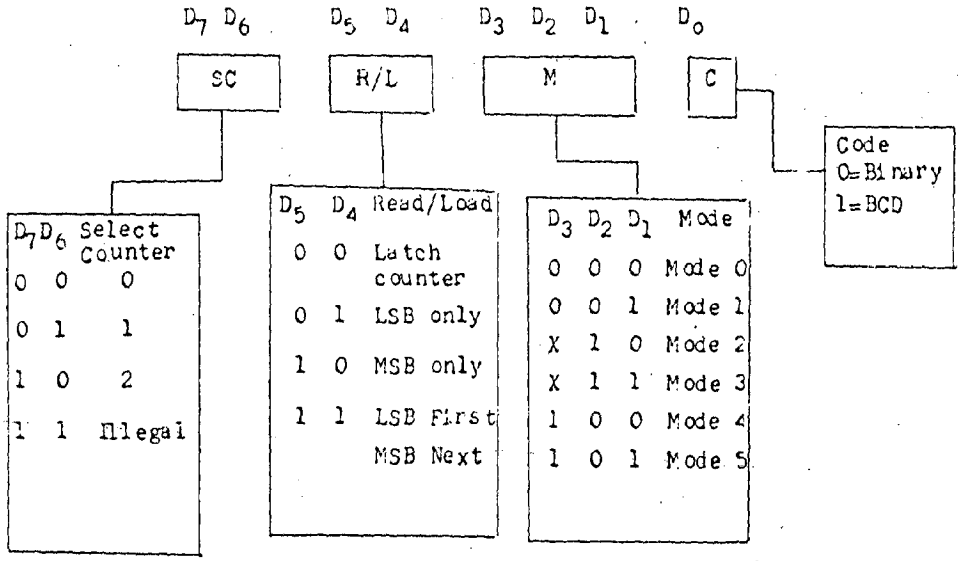


Fig. 5.2 : CONTROL WORD FORMAT OF 8253

control words: Initialization command words (ICWS) and operation command words (OCWS). The 8259 can be initialized with four ICWS; the first two are essential, and the other two are optional. Based on the modes being used, these words must be issued in a given sequence. Once initialized, the 8259 can be setup to operate in various modes by using three different OCWS; however, they no longer need to be issued in a specific sequences.