

A DIGITAL SPEED CONTROL
FOR A CHOPPER-FED DC MOTOR
BY USING THE INTEL 8080 MICROCOMPUTER

by

ANDREW KOU-CHU LIN

Diploma, Taipei Institute of Technology, 1971

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

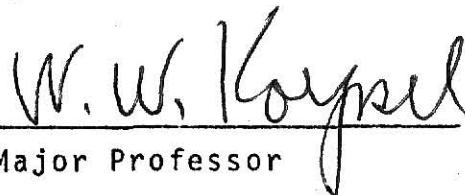
MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1976

Approved by:


Major Professor

LD
2668
T4
1976
L53
C.2
Document

i

TABLE OF CONTENTS

124

	PAGE
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 OVERVIEW OF SYSTEM CONCEPT	7
CHAPTER 3 CHOPPER CIRCUIT AND MOTOR	
3.1 Introduction	9
3.2 SCR chopper	9
3.3 Motor	11
CHAPTER 4 PULSE GENERATOR WITH CONTROLLABLE DELAY	
4.1 Introduction	13
4.2 Johnson counters	16
4.3 Operation details	18
4.4 Zero-order hold and transfer function	20
4.5 Conclusion	21
CHAPTER 5 INTEL 8080 MICROCOMPUTER	
5.1 General description	22
5.2 State-of-art in microprocessors	23
5.3 Software programming	24
5.4 Application	26
CHAPTER 6 DIGITAL MEASUREMENT OF THE SPEED	
6.1 Introduction	28
6.2 Shaft encoder and waveform shaping	28
6.3 Digital phase-locked loop and frequency multiplier	29
6.4 Software counter	37
6.5 Test result of measurement	39
6.6 Conclusion	39

PAGE

CHAPTER	7	SPEED CONTROL PROGRAM	
	7.1	Introduction	45
	7.2	Hardware configuration	45
	7.3	Data input	46
	7.4	Software configuration	46
	7.5	Conclusion	50
CHAPTER	8	SYSTEM ANALYSIS AND DESIGN	52
CHAPTER	9	FUTURE EXTENSIONS AND IMPROVEMENTS	59
CHAPTER	10	CONCLUSIONS	62
APPENDIX	A	MAIN PROGRAM FOR THE SPEED CONTROL	64
	B	BCD TRANSLATION SUBROUTINE	67
	C	ANALYZABLE MODEL FOR THE PROTOTYPE SYSTEM	70
BIBLIOGRAPHY			71
ACKNOWLEDGEMENTS			72

LIST OF ILLUSTRATIONS

	page
Fig. 1. A TACHOMETER FEEDBACK CONTROL SYSTEM.	3
Fig. 2. A PHASE-LOCKED LOOP SPEED CONTROL SYSTEM.	3
Fig. 3. THE FUNCTIONAL BLOCK DIAGRAM OF THE MICROCOMPUTER SPEED CONTROL SYSTEM.	6
Fig. 4. A SCR CHOPPER.	10
Fig. 5. AN ASTABLE OSCILLATOR FOR GENERATING PULSES.	12
Fig. 6. INPUT AND OUTPUT WAVEFORMS OF THE SCR CHOPPER.	12
Fig. 7. THE BLOCK DIAGRAM OF THE FIRING CIRCUIT.	14
Fig. 8. A 3-STAGE PROGRAMMABLE FREQUENCY DIVIDER.	17
Table 1. MICROPROCESSOR CHARACTERIZATION CHECK LIST.	24
Fig. 9. THE SCHEMATIC DIAGRAM OF THE SHAFT ENCODER.	30
Fig. 10. A BASIC PHASE-LOCKED LOOP SYSTEM.	32
Table 2. A SUMMARY OF DESIGN FORMULAS WHEN USING PHASE COMPARATOR II.	35
Fig. 11. THE FREQUENCY MULTIPLIER.	36
Fig. 12. THE FLOW CHART OF FREQUENCY COUNTING PROGRAM.	38
Fig. 13. THE DIGITAL TACHOMETER.	40
Table 3. EXPERIMENT RESULTS OF THE DIGITAL TACHOMETER.	41
Fig. 14. THE SIMPLIFIED FLOW CHART OF THE PROGRAMMING FOR THE MICROPROCESSOR SPEED CONTROL.	48
Fig. 15. THE TRANSFER CHARACTERISTICS OF MULTIPLICATION SUBROUTINES.	49
Fig. 16. THE TRANSFER CHARACTERISTIC OF THE COMPENSATOR.	51
Fig. 17. THE BLOCK DIAGRAM OF THE DISCRETE-TIME SYSTEM.	53
Fig. 18. BLOCK DIAGRAM REDUCTION OF THE SYSTEM IN Fig. 17.	56

CHAPTER 1

INTRODUCTION

The result of the tremendous progress in semiconductor technology paced by planar processing and LSI, makes digital electronic devices much smaller, faster, and cheaper. The dramatic advances in semiconductor memory such as ROMs, PROMs and EPROMs has broadened the application of the microcomputer in industrial control. One such application is the speed control of a DC motor. In industrial process control, it is at times necessary to adjust a motor's speed over a wide range with good speed resolution and reproducibility.

This paper discusses a digital system which uses C-MOS random logic and an Intel 8080 microcomputer to control the speed of a DC motor which is fed by a SCR chopper. For this application, the Intel 8080 microcomputer is used to implement the sampled-data feedback control. Both hardware and software configuration offer overall advantages in performance, price, flexibility, reliability, and power requirements. Furthermore, the micro-programming can be designed to be upgraded in a straightforward manner when fast access memory becomes available. The software can easily be added with some additional features, such as emergency stop, feedback gain self-adjusting only at the expense of response speed of the system.

Different methods are used to control the motor speed. Most employ DC amplifiers to drive motors. Conventional tachometer feedback (Fig. 1) and phase-locked loop (PLL) method (Fig.2) are currently being used. The digital type of DC motor control has certain advantages over conventional

analog regulators. Among these is the ability to obtain speed accuracies better than 0.1% and possibly approaching 0.002%.

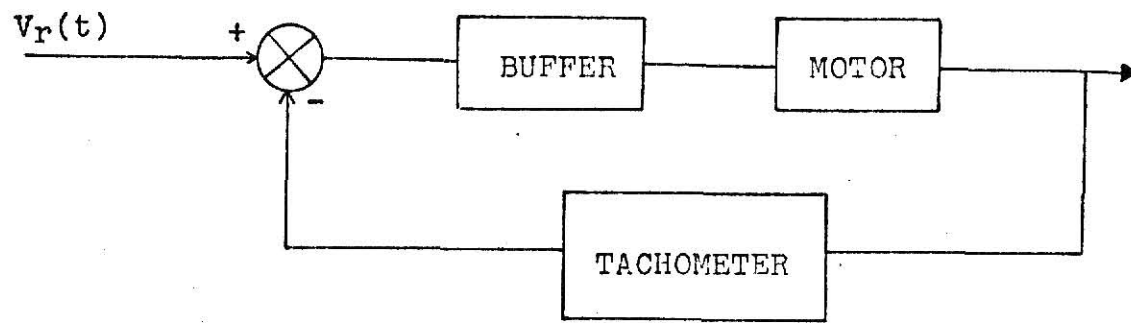
Conventional analog control methods suffer on several accounts, including nonlinearity in the analog speed transducer and difficulty in accurately transmitting the analog signal after it has been obtained from the transducer (tachometer generator). While manipulating the signal to effect control action on the motor, errors are incurred which are related to temperature, component aging, and other extraneous disturbances. In order to maintain the required quality in the output, the recalibration of reference input is necessary from time to time. In the application of phase locked loop speed control, the range of motor speed is limited by the frequency lock range of the PLL. A digital speed control system is superior in that there is no nonlinearity in the speed transducer, the digital signal representing speed can be transmitted long distances with no degradation of the original accuracy, and the digitally developed control signal is not subject to temperature variations, component changes, and noise. (8)

The advantage secured on the power circuit is that the driver amplifier, in this case a chopper, is digitally operated and the motor is the only analog device in the system. Therefore, no standard D/A converter is needed. The fact that the chopper can also be scaled-up very easily and adapted to a large DC motor control is another advantage.

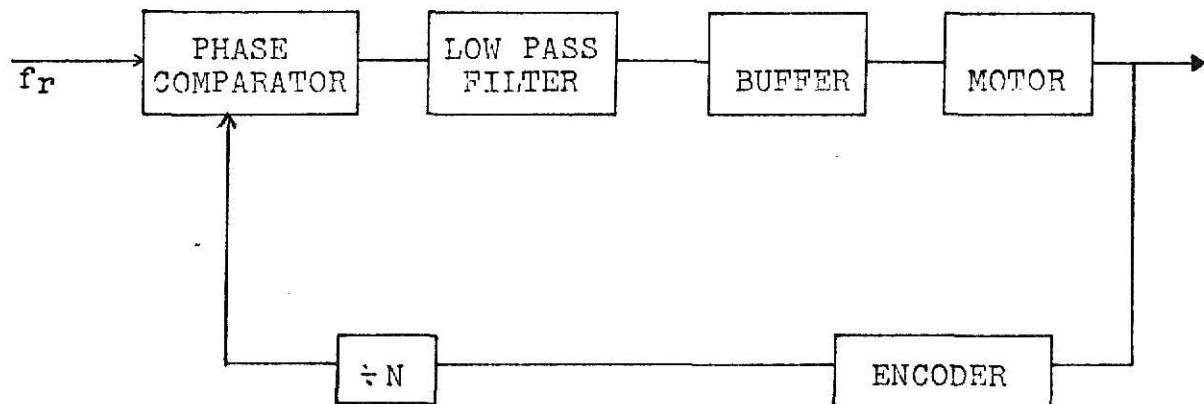
Two programmable divide-by-N 12-bit binary counters are used in the existing hardware controller with variable width pulses fed to the motor thereby varying the speed. The chopper-fed DC motor constitutes a high noise environment with four KHz noise from the motor excitation pulses and static from the motor brushes. The hardware system was implemented in C-MOS logic because of its high noise immunity. Other advantages in using

**THIS BOOK
CONTAINS
NUMEROUS PAGES
WITH DIAGRAMS
THAT ARE CROOKED
COMPARED TO THE
REST OF THE
INFORMATION ON
THE PAGE.**

**THIS IS AS
RECEIVED FROM
CUSTOMER.**



(FIG.1) TACHOMETER FEEDBACK CONTROL



(FIG.2) PHASE LOCKED LOOP SPEED CONTROL

C-MOS logic are: (a) low power consumption, i.e., when this system is in standby the current demand from the single five volt supply only 200 μ A; (b) fast operation, i.e., programmable counters can operate at 10 MHz, and (c) only one power supply in the range from 5 volts to 15 volts is required.

The microprocessor speed control system is shown in Fig. 3. The actual speed is obtained by measuring the frequency of the optical encoder output. The output pulses from the encoder are sampled for 200 msec. The measuring rate is approximately five samples per second. After measurement, the feedback signal (actual speed in rpm) is compared to the reference input (setting of desired speed) and the error is then used to adjust the phase shift between two pulse trains which control the "ON-time" of the SCR chopper such that the motor speed is maintained at its desired value. The control is effected by the firing circuit (pulse generator with variable delay) which is activated by the output of microcomputer, i.e., the amount of pulse delay to the DC chopper is adjusted according to the difference between the desired and the actual motor speed.

The prototype of speed measurement system operates on a 200msec. sampling interval and is scaled so that one binary bit in the frequency counter corresponds to 1 rpm, this gives the system a steady-state speed error of 1 rpm over the entire control range. The digital tachometer used introduces an error of less than 0.2% over much of the speed range. The high inertia of the load (a generator in the prototype system) and the synchronous drive maintain a constant speed within 0.02%. The total processing time after sampling is completed in 110 microseconds but can be shortened if the microprocessor and memory are speeded up. Transient settling takes no more than five seconds after a drastic change in either motor load or speed setpoint. The system response can be altered by adjusting certain system

parameters, such as the sampling interval, the gain of the compensator (Fig. 3), and the speed value of one binary bit, i.e., the resolution of the speed measurement system.

Since the motor speed is proportional to the output signal frequency in the digital tachometer, the speed is controlled as accurately as the frequency being measured. The digital measurement of the speed will be described in detail along with experimental results for a prototype system and suggestions for further improvement.

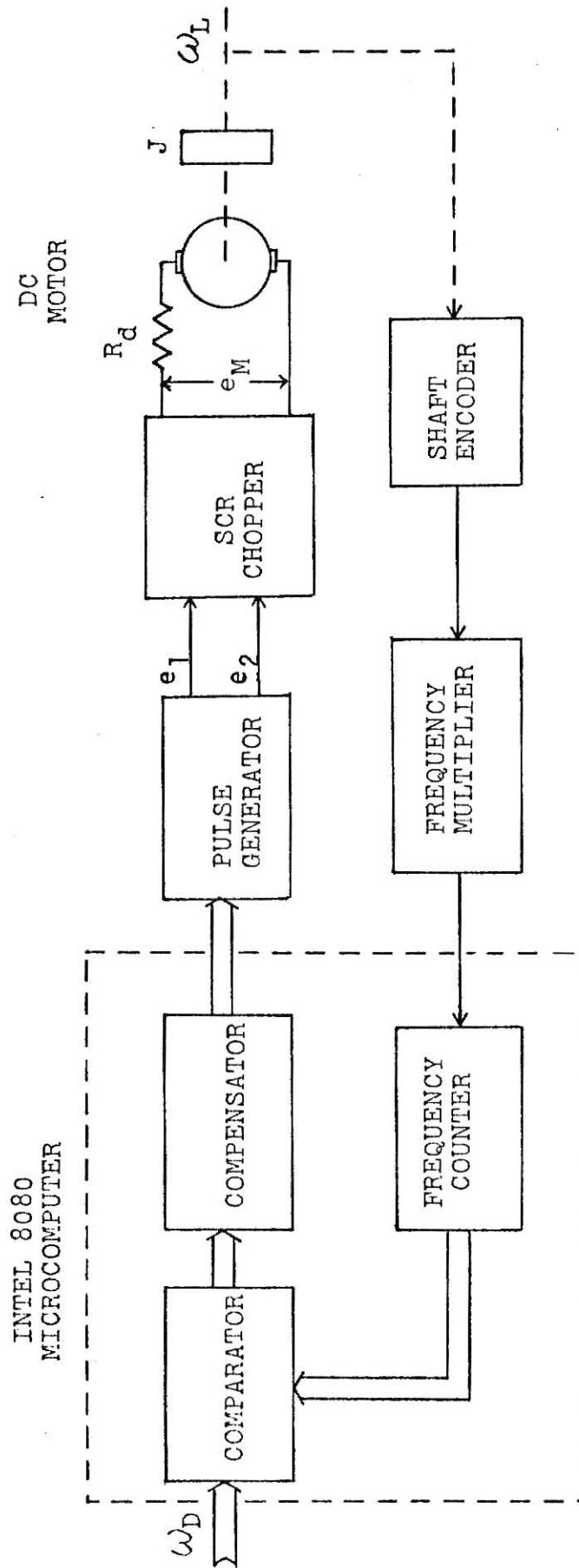


FIG. 3 THE FUNCTIONAL BLOCK DIAGRAM OF THE MICROCOMPUTER SPEED CONTROL SYSTEM.

CHAPTER 2

OVERVIEW OF SYSTEM CONCEPT

The functional block diagram in Fig. 3 illustrates the configuration of the speed control system. The system is not continuous, but measures and corrects motor speed at discrete intervals. The order of events in a complete control cycle is as follows:

- (1) Measure the actual motor speed.
- (2) Compare the actual speed to the desired speed.
- (3) Find the error.
- (4) Change the contents of the compensator an amount proportional to the error. The direction of the change depends on the result of the comparison made in step 2.
- (5) Transmit the content of the compensator to pulse generator in order to varying the duty cycle of the chopper output.
- (6) Reset the appropriate counters and logic devices in preparation for the next control cycle.

The motor speed is optically encoded using a disk with 50 holes mounted on the motor shaft. The pulse train output of this encoder is amplified, low-pass filtered, squared up by a Schmitt trigger and sent to a frequency multiplier circuit consisting of a phase locked loop with a divide-by-six counter in the feedback path. Pulses from the frequency multiplier are gated into a 12-bit software counter during a time period of 200 milliseconds, so the accumulated number of pulses stored in the counter at the end of the sampling time represents the actual motor speed in rpm. Multiplying the frequency by six is necessary to shorten the

sampling time required by extending the encoder output to 300 pulses per revolution. The 12-bit binary number of actual speed is compared to the binary number of the desired speed which is obtained via twelve toggle switches on the front panel of microcomputer or the contents in Random Access Memory and an error speed developed. This error is used to adjust the content of the control counter inside the compensator (Fig. 14) which is controlling the phase shift between two pulse trains shown in Fig. 6. The upper signal e_1 of fixed frequency fires the primary SCR (SCR_1 in Fig. 4), the signal e_2 fires the auxiliary SCR (SCR_2 in Fig. 4) which turns off the primary SCR. The voltage applied to the motor is determined by the duty cycle (δ in Fig. 6) which is a function of the relative displacement in phase of the two pulse trains. The digital content of control counter is transmitted to the pulse generator and adjusts the phase shift ranging from zero to 360 degrees. The amount of phase shift being adjusted is proportional to that speed difference. This output signal with adjusting duty cycle from chopper is then applied to the armature of the motor. Based on the capability of the microprocessor and the Random Access memory, the upper speed of the system is limited to approximately 2500 rpm.

CHAPTER 3

CHOPPER CIRCUIT AND MOTOR

3.1 INTRODUCTION:

The thyristor chopper is now very widely used for the control of DC motors. The advantages of the thyristor chopper over the conventional resistance controller are high efficiency, flexibility in control, regeneration down to very low speed, light-weight and small size. The chopper output voltage can be controlled either by using a current-limit control scheme or by pulsewidth control. The present work makes use of the pulsewidth control scheme in which chopper "ON" to "OFF" time ratio is controlled.

3.2 SCR CHOPPER:

The power circuit used to drive the motor is shown in Fig. 4. The chopper circuit operates on the principle of anode pulse commutation. In order to increase turn off reliability, an activated commutation technique is used (10). Assuming that the capacitor C_1 is charged with its left-hand plate positive and the main SCR_1 is conducting. If the auxiliary SCR_2 is fired, the main SCR_1 is reverse-biased and turned off. The commutation capacitor C_1 is charged through the path consisting of the SCR_1 , transformer T_1 , and diode D_1 with its left-hand plate positive. E_s is an auxiliary source generating square waves to charge capacitor C_1 and to turn off SCR_2 when its gate voltage drops to zero. The presence of an auxiliary source

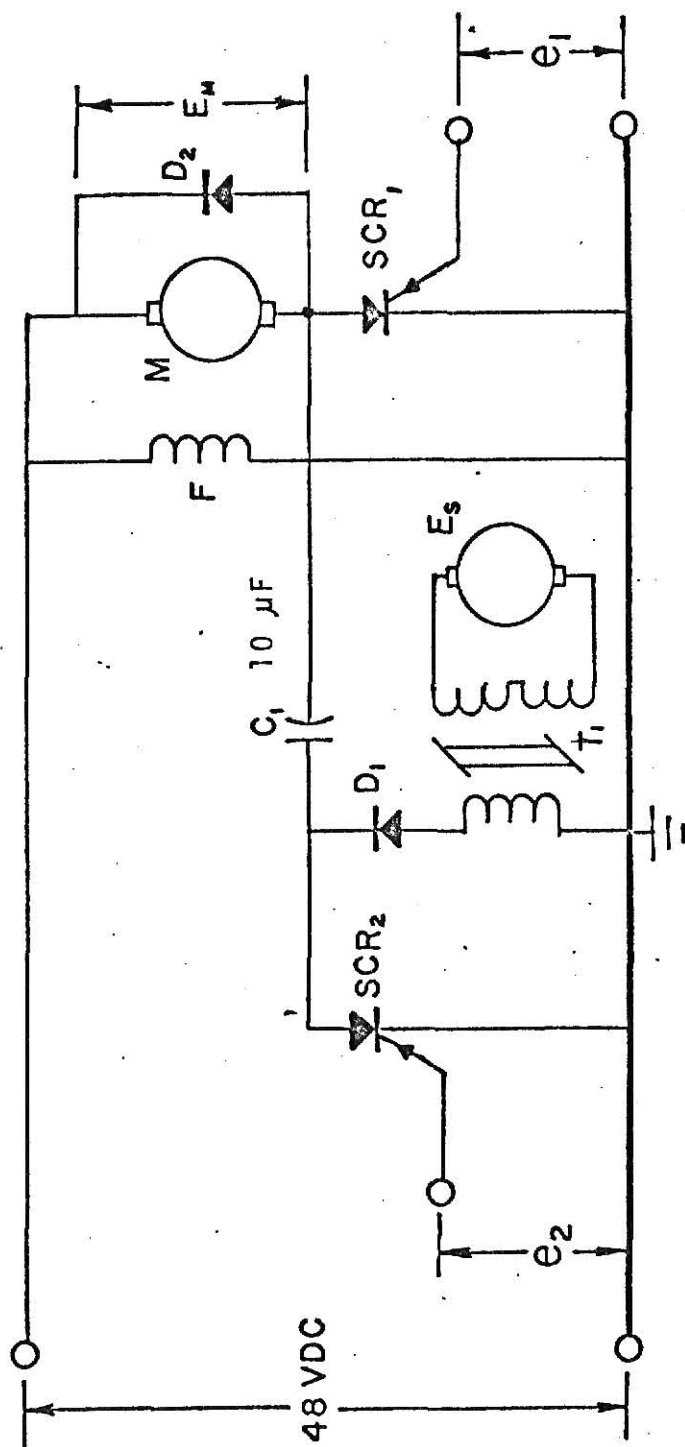


FIG. 4 SCR CHOPPER

ensures commutation at any speed. The use of an auxiliary source also permits the use of a smaller commutation capacitor. The diode D_2 is a free wheeling diode to provide a path for motor current when SCR_1 is turned off. The chopper is operated at a frequency of 150 Hz while the auxiliary source is generating pulses with frequency of 1 KHz. The power applied to the motor has an average value determined by the relative displacement of the pulse trains controlling the chopper. The gain constant for the chopper is determined by its duty cycle. The voltage applied to the motor is given by $E_M(s) = \frac{48}{2\pi} \phi(s)$ where $\phi/2\pi$ is the proportionate time during which power is applied to the motor. ϕ is determined by the relative phase displacement in time between e_1 and e_2 (αT in Fig. 6). Thus we can write

$$\frac{E_M(s)}{\phi(s)} = \frac{48}{2\pi}$$

3.3 MOTOR:

The motor controlled by the chopper is a Robbins and Meyer separately excited 48 volt DC motor rated 1/20 hp at 2000 rpm. It requires 1.3 amperes full load current. The transfer function of the motor is

$$G_M(s) = \frac{1/K_e}{(T_M s + 1)}$$

where $K_e = 0.229$ volt-second

$$T_M = 0.04 \text{ second}$$

This transfer function for the motor is a ratio of angular velocity transform of the motor shaft to input voltage transform. In order to provide a reasonable load on the motor, a similar motor connected as a generator is coupled to the shaft of the controlled motor.

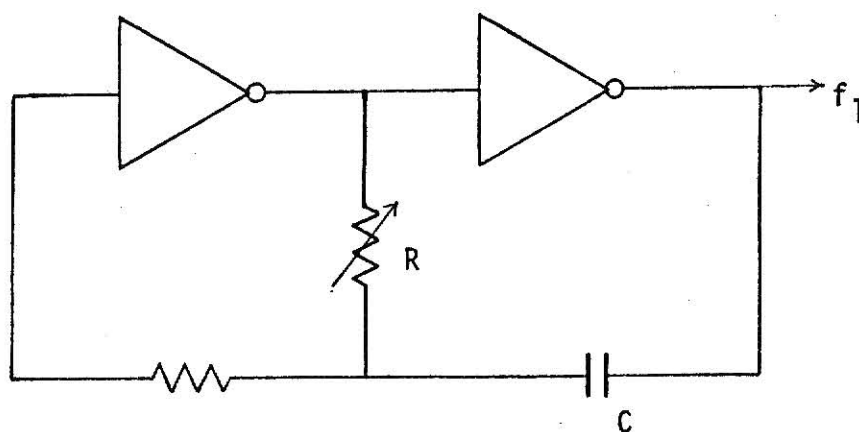


FIG. 5 AN ASTABLE OSCILLATOR FOR GENERATING PULSES

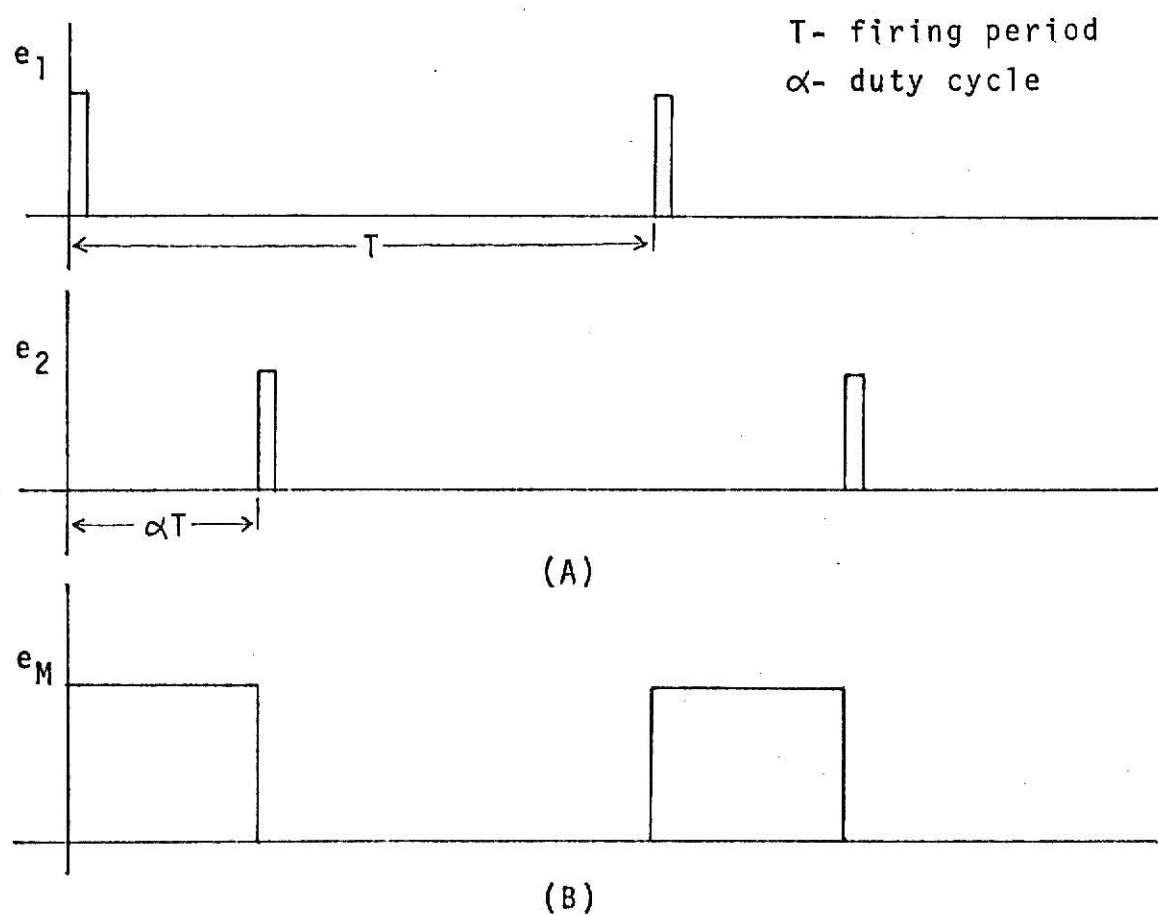


FIG. 6 (A) PULSE TRAIN OUTPUT OF THE PULSE GENERATOR.
 (B) SCR CHOPPER OUTPUT UNDER IDEAL CONDITIONS.

CHAPTER 4

PULSE GENERATOR WITH CONTROLLABLE DELAY

4.1 INTRODUCTION:

The firing circuit is the most effective controller in the entire feedback control system. A large number of circuits to obtain time delay for trigger control of SCRs are available. Monostable multivibrator and UJT time delay circuits are the most commonly used circuits. However these monostable circuits have the following drawbacks:

- (1) They are sensitive for stray pulses which normally result in false triggering.
- (2) The relationship between control voltage and time delay is not linear.
- (3) There is a finite recovery time and it increases for large time delay.

The UJT time delay circuit is quite stable in its operation. However, the shape of the output pulse is dependent on the load in which the timing capacitor discharges through the emitter of a UJT. For automatic control of the time delay the charging current of the capacitor C is controlled. In such cases the delay could not be reduced below a fixed minimum which depends on the maximum allowable charging current. To maintain linearity between control voltage and the charging current, particularly at low current (for large time delay), an improved current generator is required in the UJT time delay circuit.

The simple digital scheme for triggering SCR's using programmable counters is discussed. The block diagram of the proposed scheme is shown in Fig. 7. The astable oscillator, clock 1, is the basic pulse source from

which two other pulse trains with time delay are derived. One pulse train periodically turns on a primary SCR which applies power to the controlled motor. The second pulse train is used to fire a second SCR to commutate the primary SCR. The amount of phase shift between two pulse e_1 and e_2 is transmitted to this pulse generator from two output ports of the microcomputer. The digitally controlled pulse generator can vary the phase difference between two pulse trains in binary increments up to 360 degrees. Referring to Fig. 7, the principle of operation in triggering chopper circuits can be outlined as follows:

- (1) Block (8) produces the trigger pulse for the main SCR.
- (2) Block (7) gives a delayed trigger pulse to auxiliary SCR; the delay is controlled by the microcomputer.
- (3) Block (9) controls block (7) and gives an "off" time for chopper circuits before producing the next trigger pulse for the main SCR.

The firing period is determined by the frequency of clock 1 and the count of block (8), i.e., $T = 4095/f_1$.

The pulses e_1 and e_2 repeat for each complete cycle of the 12-bit binary counter, i.e., every 2^{12} input pulses, and at the termination of each cycle the output "0" of the Johnson counter goes high and produce a narrow pulse. Thus, for an 12-bit pulse generator the amount of phase shift $\Delta\theta$ over any interval T which is a multiple of $1/f_1$ is given by

$$\Delta\theta = N \times \frac{360^\circ}{4095} = N \times 0.088^\circ \quad (0 \leq N < 4096)$$

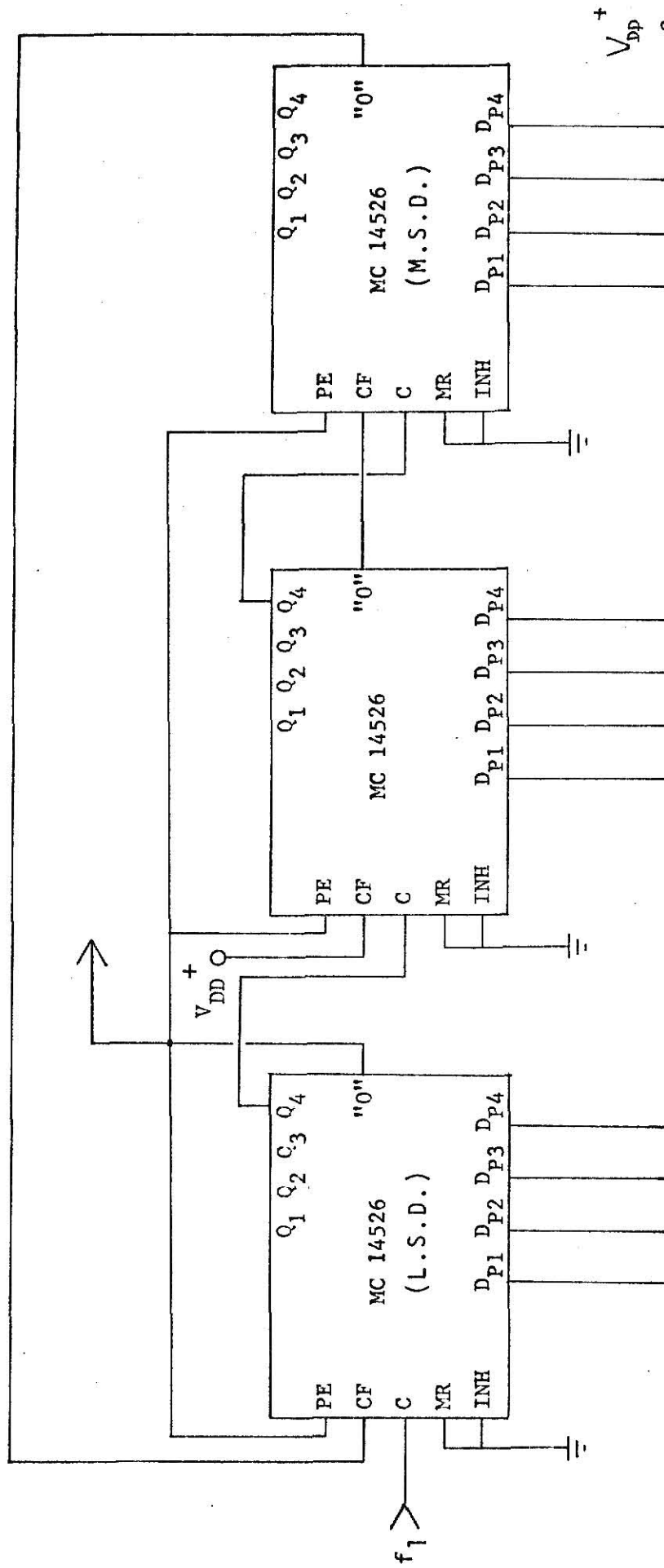
The quantity N is the binary number transmitted from the Intel 8080 microcomputer. The phase shift has been quantized. The resolution, i.e., the accuracy of controller can be greatly improved by adding just one more "bit".

Incremental accuracies of better than ± 0.088 degree/bit is accessible.

4.2 JOHNSON COUNTERS:

High speed operation and spike-free decoded outputs are the principal features of the Johnson counters. The programmable divide-by-N 4-bit binary counter MC 14526 is a down Johnson counter and is composed of 4 toggle enable flip-flops, connected in cascade, with associated additional gating to provide the down counting mode and possibility of cascading several counters without external gates. Normally, the outputs are low, with the exception of the one decoded state, which is high (4). The programming is achieved by applying the preset variable to the D_p 's inputs (D_{p1} - D_{p4} in Fig. 8). This is transferred to the counter cells by the preset enable (PE) input control. The clock inhibit allows the counter function to be disabled. The Master Reset (MR) provides synchronous initiation of the divide-by-N cycle. Cascade feedback (CF) input and "0" output are provided for cascading purposes. In operation, the "0" output goes high when the counter content reaches "zero" on all Q outputs provided that CF is high. If CF is low, the output "0" remains zero even if the counter content is zero. This feature ensures that with the "0" output connected to the PE input, the preset variable is entered only when all the counters are zero.

In Fig. 8, a 3-stage cascading counters connection is illustrated. The preset variable D_{p1} - D_{p4} is determined by the output of the Intel 8080 microcomputer with which the cycle length can be changed. The counting capacity can easily be increased by using the cascading features provided internally in the counters. Fig. 8 illustrates a programmable frequency divider which divides each firing period into 4095 units. Note that the CF of the second stage which is right after L.S.D. is always "1".



(FIG. 8) 3-STAGE PROGRAMMABLE FREQUENCY DIVIDER

4.3 OPERATION DETAILS:

The operating details of the system shown in Fig. 7 are given as follows. Clock 1 is the free running oscillator generating sharp pulses. Although the UJT relaxation oscillator is simple, the transient effect (spike) on power source and unstable output with temperature change are the deficiencies. The present system uses an astable free running oscillator shown in Fig. 5, whose frequency can be set by an external resistor and capacitor. The generator producing e_1 and e_2 shown in Fig. 6 can best be thought of as consisting of "units" based on the oscillator frequency. Assume that one "unit" equals the period of the oscillator frequency f_1 . By multiplying the number of "units" that are contained in a pulse width or delay times the period of the oscillator frequency, the approximate time delay in seconds can be derived as:

$$\alpha T = \frac{N}{4095} \times T = \frac{N}{4095} \times \frac{4095}{f_1} = \frac{N}{f_1} \quad (0 \leq N < 4096)$$

In this programmable time delay firing circuit, because of the 12-bit Johnson binary counter, the time period between each "turn on" pulse is divided into 4095 units. A major advantage of this pulse generator is the phase shift resolution can be very high with only a small increase in I.C. chip usage. For example, a low resolution containing 256 units in one period using only 8 bits of divider would require two I.C. chips. A high resolution containing 4096 units may use 12 bits counter which would require three I.C. chips with only a 50% increase in cost. The speed control system performance, i.e., its accuracy, static error and transient response can easily be improved by simply increasing the resolution. It appears that a resolution of 4096 unit is already good enough for the prototype system.

The amount of phase shift is in 12-bit binary form. Each unit stands for 0.001534 radians or 0.088 degrees. The phase shift can be programmed by changing the data $D_{p1}-D_{p12}$ slightly each time the content of the control counter is sent out of the microcomputer over the ribbon cable. Another advantage is the versatility of this kind of pulse generator which can be the interface of a microcomputer and any other kind of the SCR gating network.

The frequency of master clock f_1 is divided by a 12-bit binary counter, i.e., the frequency output of the counter is $f_1/4095$. This narrow pulse train will be stretched out for 100 μ sec. by the monostable multivibrator and used to turn on SCR_1 . The programmable divider can divide the f_1 by any number from 1 to 4095 according the input variable $D_{p1}-D_{p12}$. The number preset at the input $D_{p1}-D_{p12}$ is introduced into the counter chain when all counters are "zero", i.e., MR of block (7) being set "1" while "0" output of block (8) goes high. Then the block (7) is decreased at each positive clock transition of f_1 . When the zero state of programmable divide-by-N counter is reached, the output "0" of block (7) will be high. This positive transition will trigger block (9) which will set MR on block (7) to "high" and disable block (7) from counting until the next base timing pulse from block (8) initiates block (7) by way of block (9) and preset the predetermined number again into the counter chain (the "0" output of the L.S.D. counter is connected to all preset enable inputs as shown in Fig. 8) and the firing cycle repeats. The output port of the Intel 8080 sets N on block (7) which is enabled by "0" output from block (8) through the reset and inhibit logic control (9). The N clock pulses are counted by block (7) and the number of these pulses determines the length of time SCR_1 is on. The output "0" of the 12-bit binary counter in block (8) determines the actual

firing rate of SCR_1 . This signal goes to SCR_1 through the monostable multivibrator and pulse stretcher (10), buffer (12), and electro-optical coupler (14). The "0" output of block (7) is delayed by N clock pulses and fires SCR_2 through the pulse stretcher (11), buffer (13) and electro-optical coupler (15), turning off SCR_1 .

4.4 ZERO-ORDER HOLD AND TRANSFER FUNCTION:

The transfer function relating the error data from the microcomputer with the displaced angle between e_1 and e_2 is K_p . It is given by $K_p = 1/2^n \times 360^\circ = (2\pi)/4096$ when $n=12$ so that a resolution of 1 in 4096 is produced. n can be increased but the gain is reduced. For accomplishing better smoothing of the ripples, a holding device is often employed in sampled-data control system to hold or to clamp the sampled values. The holding device converts the sampled control signal into continuous form which is an approximate reproduction of the desired actuating error signal before it is fed into the controlled system. One of the simplest holding devices is the zero-order holding circuit in which the value of an input pulse (i.e., the strength of its equivalent impulse) is held constant until the next sampling pulse. The output of the pulse generator (i.e., phase difference) will change for every 0.2 second and maintain a constant value in between samplings. Therefore, this pulse generator can be considered as a perfect zero-order hold circuit. The transfer function should be modified as follows:

$$K_p = \frac{2\pi (1 - e^{-Ts})}{4096 \times s}$$

$$T = 0.2 \text{ second}$$

4.5 CONCLUSION:

The above delay control circuit is highly flexible and has wide application in solid state power control. The variation of the delay with input signal is smooth and undisturbed by stray pulses. The firing period and the ratio of ON time to the period can be controlled by two independent digital processor. Also, the circuits overcomes many of the drawbacks of the conventional circuits mentioned above. The followings are the advantages of this delay control circuit compared to the conventional delay circuits.

- (1) The design of the delay circuit is extremely simple as it involves the choice of R and C only to determine the frequency of clock 1 (Fig. 5).
- (2) The output of the pulse generator is firmly held at $+V_{CC}$ or ground and is not disturbed by stray signals. This ensures reliable operation.
- (3) The shape of the output pulse of the generator is independent of the load.
- (4) In the application of chopper control circuit, the range of the ratio of ON time to the firing period is fully controlled, whereas there is some limitation for the conventional circuit.
- (5) The relationship between control signal (digital data) and time delay is linear. The delay could be changed in a full range.
- (6) The wiring of the circuit was normal and shielded wires were not used in the circuit. In spite of open access to the stray pulses, the operation of the delay circuit was found to be stable and reliable. The operation of delay control is being quantized rather than continuously. By increasing the resolution of the firing period properly, it will not present any problem.

CHAPTER 5

INTEL 8080 MICROCOMPUTER

5.1 GENERAL DESCRIPTION:

A new technology which has emerged in this decade is that of microcomputers. It may be said that microcomputer technology is the product of a marriage of minicomputer architecture with large scale integrated circuit techniques (13). Microprocessors have been made possible by the advances in semiconductor processing technology. They are only five years old but are mature in the sense that there are over 25 designs on the market (15).

The microprocessor hierarchy can be divided into three levels: the microprocessor, the microcomputer, and the microcomputer system. A microprocessor is only a LSI chip which is capable of processing logical and arithmetic data under program control. A microcomputer consists of the elemental LSI components required to realize a digital computer. The components, mounted on a printed circuit board, are the microprocessor, minimum memory, and some limited I/O circuitry. A microcomputer system contains a more developed microcomputer plus the supporting power supplies, interface boards, control panels, etc., which present the user with a capability nearly equal to that of a minicomputer.

The Intel 8080 is a second generation design of the most popular microcomputer in the field. It is a general purpose byte-oriented machine (8-bit word). It can support up to 64K of directly addressable memory and can address 256 separate input and output devices. The Intel 8080's powerful instruction set has 78 basic machine language instructions (7). In the prototype system, an Intellec 8 microcomputer system is actually used to

control the motor speed. In its basic configuration, the Intellec 8 microcomputer system contains the cpu board with Intel 8080A microprocessor and Intel 8224 clock generator and 8216 bus driver, 512 8-bit bytes of PROM, 2048 8-bit bytes of RAM, one port serial interface, multi-port parallel interfaces for connection to external devices, front panel interface and an rack connector for memory expansion. Since the Intellec 8 computer is 8080A based, its memory capability can be expanded to 65536 bytes. Real time clock of master timing circuit in which clock pulse widths and phasing as well as frequency are crystal controlled (Ref. 3). The complete schematic diagram for the Intellec 8 microcomputer system is much too large to be reproduced in this paper. For a complete schematic, readers may refer to operation manual of Intellec 8 (Ref. 3).

In order to perceive the manner in which the computer functions, a description on the basic circuits is necessary. The cpu which is the heart of the computer controls the interpretation and execution of software instruction program within the memories. The I/O provides a communication link between the cpu and external device. The front panel allows the operator to manually perform various operations with the Intellec 8. All communication between the computer and the operator is through a Teletype ASR-33 teletypewriter and the switches on the front panel of Intellec 8 computer. The paper tape reader may be used for data and command entry.

5.2 STATE-OF-ART IN MICROPROCESSORS:

The key to microcomputer's versatility is its integral microprocessor (μ p) chip. The μ p operates on instructions stored in the programmable read only memories (PROM's) or the random access memories (RAM's). A microprocessor characterizes a microcomputer system. Its architecture determines

programming techniques, system speed, system cost, and gives guidance as to which model would be best for a given application. Important parameters are listed in Table 1 (Ref. 13).

TABLE 1
MICROPROCESSOR CHARACTERIZATION CHECKLIST

-
-
1. speed
 2. power of instruction set
 3. support hardware required
 4. hardware system availability
 5. program development aids
 6. applications assistance
 7. price and availability
-

Speed can only be determined by running a benchmark program. Clock rate, power of instruction set, number of accumulators and working registers, and word size determine speed. The instruction set determines basic abilities, memory utilization, and the ease of programming. A simple instruction set can often do complex functions by coding subroutines which sacrifice speed and memory utilization. Program development aids include compilers, assemblers, debugging, and editing programs. These may be run on a time-share computer or a microprogrammer's console (Ref. 3).

5.3 SOFTWARE PROGRAMMING:

A microcomputer, like any other computer, is able to do a given task because it first contains hardware to do a specific task or instruction (one at a time), and second it contains a list of these instructions (

called a program), performed in the proper order to accomplish the desired result. Although the particular instruction set available is dependent on a particular microprocessor, most microcomputers can be programmed to do any task any other one can do, but it may take more instructions to do it. The major steps involved in programming a given task are as follows (Ref. 13).

- (1) Flow charting-Charting a logical sequence of events to accomplish the desired task.
- (2) Coding-Conversion of each logical step into one or more instructions.
(Normally it is assembly language which allows the programming to be done using word symbols rather than numbers.)
- (3) Key punching-The handwritten coding is punched into an input media such as paper tape or cards.
- (4) Assembly-The source code is used as the input to a computer which converts it to machine language coding using a program called the assembler.
- (5) Loading-The object code is now loaded into a PROM or RAM. After completing the programming of the microcomputer, the program is executed and debugged if necessary to correct logic or programming mistakes.

The software is an essential component of an microcomputer. The microcomputer hardware is useless without a software to run it. The ability to alter the stored program quickly without the need for corresponding wiring changes is the most persuasive advantage of microcomputers. The amount of software needed for an microcomputer, and the cost of that software, are basically functions of the complexity of the units. When an microcomputer is designed to support a big system the necessary software and adapter hardware may cost very much.

The program used for the speed control system is rather straight forward. The most crucial portion of the stored program will then be examined in more detail, with reference to simplified flow charts (Fig. 14) and program listing (Appendix A&B). The program are entered from the teletypewriter, either through the keyboard or the paper tape reader.

5.4 APPLICATIONS:

Because the digital approach will usually be more economical for high accuracy, computer control has become increasingly popular in recent years. Flexibility, reliability and low power requirement open up a large range of potential application areas for microcomputers. The microprocessors are becoming more widely used as the cost of digital circuitry is reduced and the access time of semiconductor memory is speeded up. Advantages of digital controllers (microprocessors) over analog controllers have been concluded as follows:

- (1) Digital controllers are capable of performing complex computations with constant accuracy at high speed. Digital computers can have almost any desired degree of accuracy in computations at relatively little increase in cost.

- (2) Microprocessors are extremely versatile. By merely issuing a new program, one can completely change the operations being performed.

Because of the step-by-step operation in the microprocessor, the execution of a big program will take a relatively long period of time. Using microprocessors on the control application will pose some problems if the system response speed is much concerned. Therefore, some portion of the software program must be replaced by some digital hardwired circuitry under the consideration of tradeoffs between the speed and the cost.

Interfacing, which deals with the adapter hardwares, will be the major consideration while connecting the microcomputer to the real world. The adapter hardware configuration will be varied considerably due to the different requirements or the different plant control applications. Another interface concern is connecting the microcomputer to a system where the data accuracy is highly in demand. In other words, the word size for the data must be large enough to achieve the required accuracy. Since a microprocessor typically has a 8-bit word, the input/output lines may have to be multiplexed.

CHAPTER 6

DIGITAL MEASUREMENT OF SPEED

6.1 INTRODUCTION:

Accurate frequency measurement in short period of time is of fundamental importance. The accuracy may be increased by counting for a longer period of time. However, the measurement is required within an interval of time that should be very small compared to the mechanical time constant of the machine. A method commonly used for accurate and fast measurements is to multiply the signal frequency and then count it.

This chapter presents details of a digital method of frequency measurement with hardware interfacing and Intel 8080 software programming to measure the angular speed of a motor. Measurement of high accuracy are made in a very short time. The output signals are available in a digital format suitable for direct use in an on-line digital control device.

A block diagram for the measurement system is shown in Fig. 3. The unknown frequency from the shaft encoder is multiplied by a factor of 6. The output frequency which is only 6 times the input frequency can be measured in 0.2 seconds instead of 1.2 seconds to achieve the desired accuracy. The multiplication is accomplished with standard inexpensive integrated circuits.

6.2 SHAFT ENCODER AND WAVEFORM SHAPING:

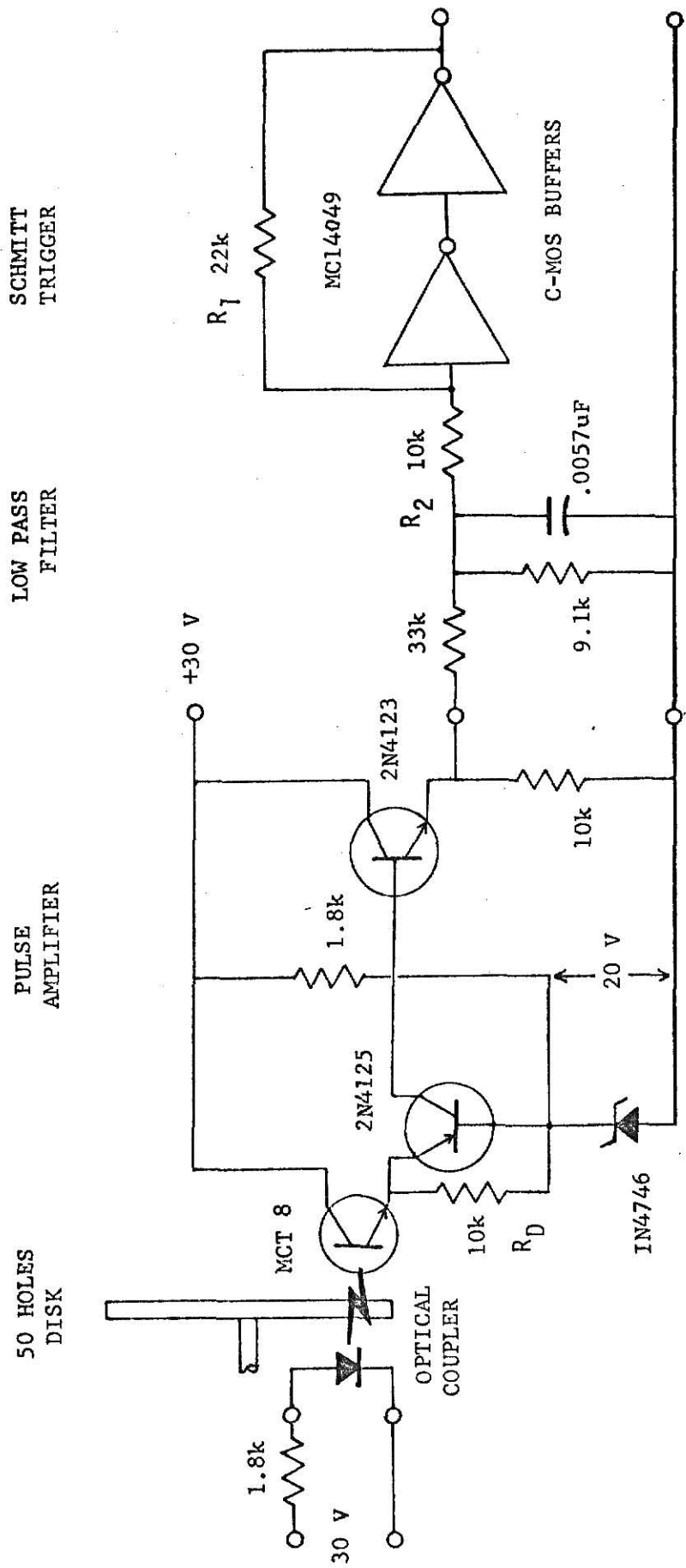
This section is describing an optical transducer which provides a train of pulses with a frequency proportional to the angular velocity. The low inertia and frictionless disk with 50 evenly spaced slots, is tightly coupled

to the shaft of motor. It should be noted that the number of slot does not have to be very large for good resolution. In fact, the resolution can be improved by multiplying the output of optical transducer times a factor, within certain limits imposed by the speed of the microprocessor which performs the counting. A schematic arrangement of the optical transducer is shown in Fig. 9. The motor shaft rotation is converted to pulses with the use of an opto-coupler. This consists of an infrared LED exciting a photo-transistor. The light path is broken by the slotted disk attached to the motor shaft. The weak pulses are amplified, low-pass filtered, and sent to a Schmitt trigger. It was found necessary to add a low-pass filter and Schmitt trigger to the output of the pulse amplifier to eliminate stray spikes and noise and shape the pulse which is then applied to the input of the frequency multiplier. The Schmitt triggers are commonly found interface between the analog and digital systems. Conventional Schmitt triggers are analog amplifiers with positive feedback. In the system under discussion, to obtain positive feedback, two C-MOS inverters is connected in cascade and feedback applied around. The degree of hysteresis (Ref. 4) is dependent on the ratio of the resistors R_1/R_2 as shown in Fig. 9.

The transfer function is $K_2 = 50/2\pi$, since the encoder produces 50 pulses for each revolution.

6.3 THE PHASE LOCKED LOOP (PLL) AND FREQUENCY MULTIPLIER:

A PLL frequency multiplier is added to the measurement circuit in order to retain good resolution and to reduce the time needed for measurement. The reduction of sampling time will be dependent upon the number of quantization intervals per revolution. The frequency multiplier for times six associated with 50 slot encoder is equivalent to an optical encoder with

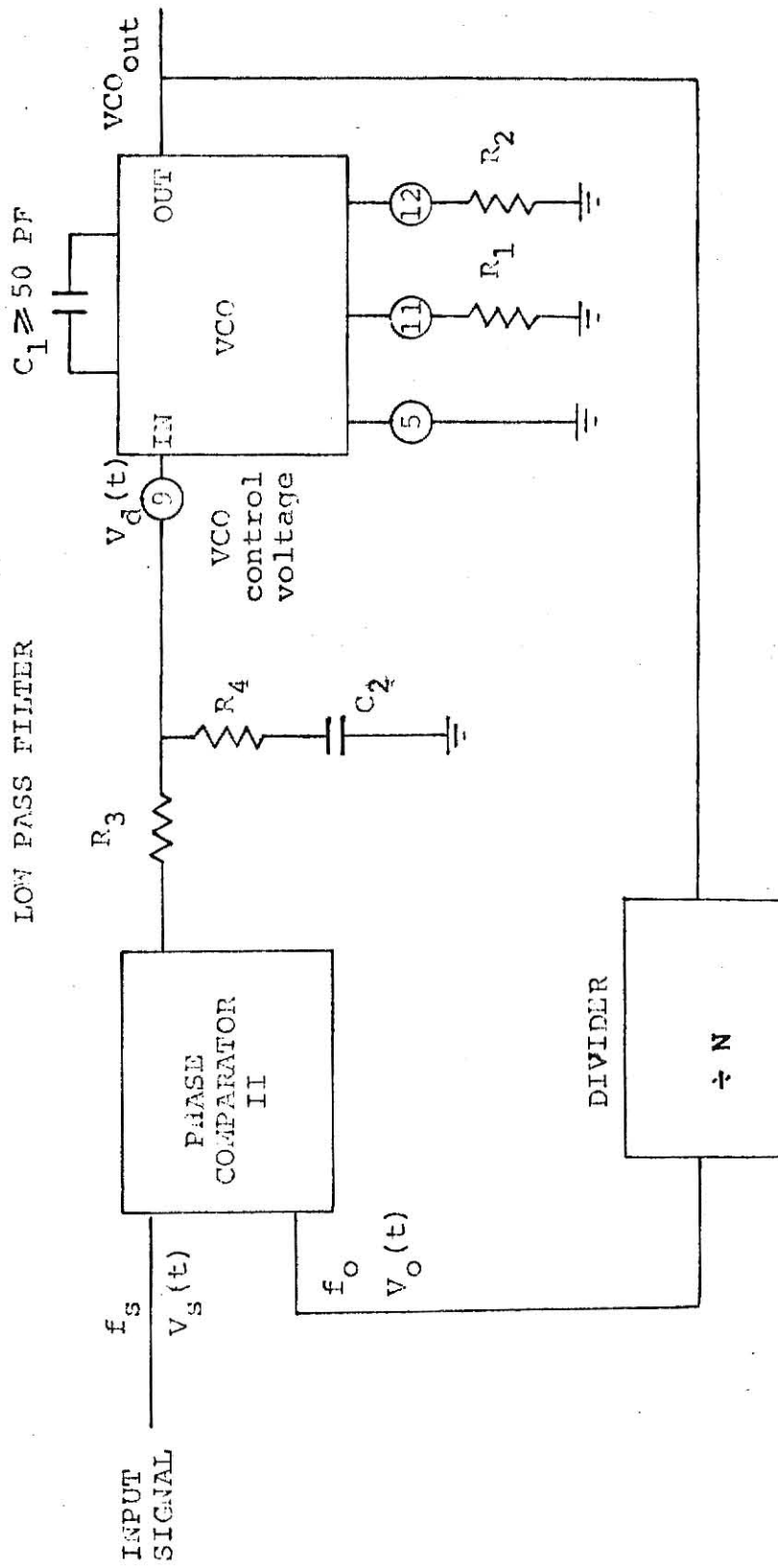


(FIG. 9) THE SCHEMATIC DIAGRAM OF THE SHAFT ENCODER.

300 slots per revolution. If there is an encoder of 300 holes available at the market with lower price, this multiplier will not be necessary.

For the sake of completeness, the basic theory of the phase-locked system will be presented first, followed by the details of practical implementations and the results. Phase-locked loops (PLL's) are finding significantly increased usage in signal-processing and digital systems. Frequency multiplication is one of the many applications of a PLL. The basic phase-locked loop system is shown in Fig. 10. It consists of three parts; digital phase comparator, low pass filter and voltage-controlled oscillator (VCO). All are connected to form a closed-loop frequency-feedback system with zero error voltage at the output of the phase comparator. The VCO will operate at a set frequency, f_0 called the center frequency (Ref. 12). $V_d(t)$ varies in a direction that reduces the frequency difference between the VCO and signal-input frequency, when the input frequency is sufficiently close to the VCO frequency, the closed-loop nature of the PLL forces the VCO to lock in frequency with the signal input; i.e., when the PLL is in lock, the VCO frequency is identical to the signal input except for a finite phase difference. The range of frequencies over which the PLL can maintain this locked condition is defined as the lock range of the system. The lock range is always larger than the band of frequencies over which the PLL can acquire a locked condition with the signal input. This latter band of frequencies is defined as the capture range of the PLL system.

There are two types of digital phase-comparator in one I.C. chip (MC 14046). Phase-comparator I is an exclusive-OR network. When using this phase comparator, the signal- and comparator-input frequencies must have 50 percent duty cycle. Phase-comparator II is an edge-controlled digital memory network. This type of phase comparator acts on the positive edges



$R_1 > 10 \text{ K}$

R_2 defines the current which offsets the VCO operating frequency for VCO input signal of "0" volts

(FIG. 10) A BASIC PHASE-LOCKED LOOP SYSTEM

of the signal- and comparator-input signals. The duty cycles of the signal and comparator inputs are not important since positive transitions control the PLL system. Utilizing this type of comparator, thus, for phase-comparator II, no phase difference exists between signal and comparator input over the full VCO frequency range. It should be noted that the PLL lock range for this type of phase comparator is equal to the full VCO range. The PLL system can function as a frequency-selective frequency multiplier by inserting a frequency divider into the feedback loop between the VCO output and the comparator input (see Fig. 10). N , the frequency-divider modulus, can vary from 1 to 15 in steps of 1.

Multiplication is achieved by dividing the VCO output by the desired multiplication factor and then comparing this signal to the input signal with a phase detector. When the PLL system is in lock, the signal and comparator-inputs are at the same frequency. Because the output frequency is divided by N before being compared to the input frequency, it is N times the input frequency, i.e., $f = N \times f_{in}$. The output of the VCO is fed back through a MOTOROLA MC 14526 binary counter with proper wire connections to give a divide-by six square wave output. A buffered digital output drives the conventional TTL input port of Intel 8080 microcomputer. The output frequency is ready to be counted.

The speed measurement has two major design features. The first is the dynamic response of PLL. The other is the time interval of measurement. The design concerning time scaling is limited by the angular resolution of the disk and the speed of microcomputer. The frequency of the pulse train from multiplier provides angular velocity by means of correct scaling and integration by the Intel 8080 microcomputer. More details on choosing multiplication factor and time scaling will be presented in next sections.

It is evident that the performance of the Phase-Locked Loop will depend considerably on the transfer function of the filter (Ref. 5&6). The low-pass filter for this application is a two-pole, lag-lead filter which enables faster locking for input changes in frequency. The important loop parameters, such as acquisition time, capture range, lock-in range, etc., are discussed extensively in the literature on phase-locked loops (Ref. 11).

The VCO is designed so that its output frequency range is equal to or greater than the required output frequency range of the system. The ratio of change in output frequency to input control voltage is called "gain constant", K_o . Phase-comparator II is used for this application because it will not lock on harmonics of the signal-input reference frequency (phase-comparator I does lock on harmonics). Since the duty cycle of the output of the Divide-by-N frequency divider is not 50 percent, phase-comparator II lends itself directly to this application. Using the formulus for phase-comparator II shown in Table II (Ref. 12), the VCO is set up to cover a range of 0 to 12.5 Kiloherztzes(0rpm - 2500rpm), i.e., the highest frequency that can fed into software counter is

$$2500 \text{ rpm} \times \frac{50 \text{ holes}}{60 \text{ seconds}} \times 6 = 12.5 \text{ KHz}$$

Based on the information from Table II, the component value turns out to be $C_1 = 873 \text{ pF}$ and $R_1 = 91.6 \text{ KiloOhms}$. System dynamics when in lock are determined by the amplifier and the filter block. Its gain determines how much phase error exists between f_{in} and f_{out} , and filter characteristics shape the capture range and transient performance. In a well defined system controlling factors such as ω_n and ζ (undamped natural frequency and damping ratio) may be chosen either from a transient basis (time domain response)

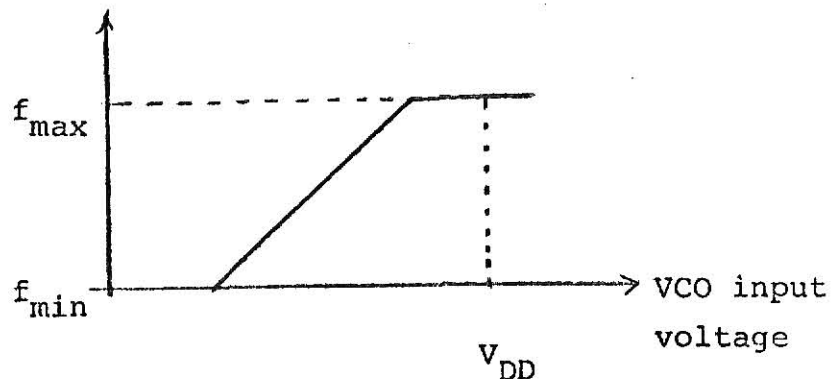
or steady state frequency plot. Once these two design goals are defined, synthesis of the filter is relatively straightforward (Ref. 2&11). The damping ratio γ has been defined as 0.7 as it usually does. Because good filtering and fast acquisition are conflicting requirements, the design must usually be a compromise. Trial-and-error is one of the easiest way to end up with good component values on low-pass filter, i.e., C_2 , R_3 and R_4 shown in Fig. 11.

TABLE II

SUMMARY OF DESIGN FORMULAS WHEN USING COMPARATOR II

1) VCO without offset when $R_2 = \infty$

2) Transfer characteristic



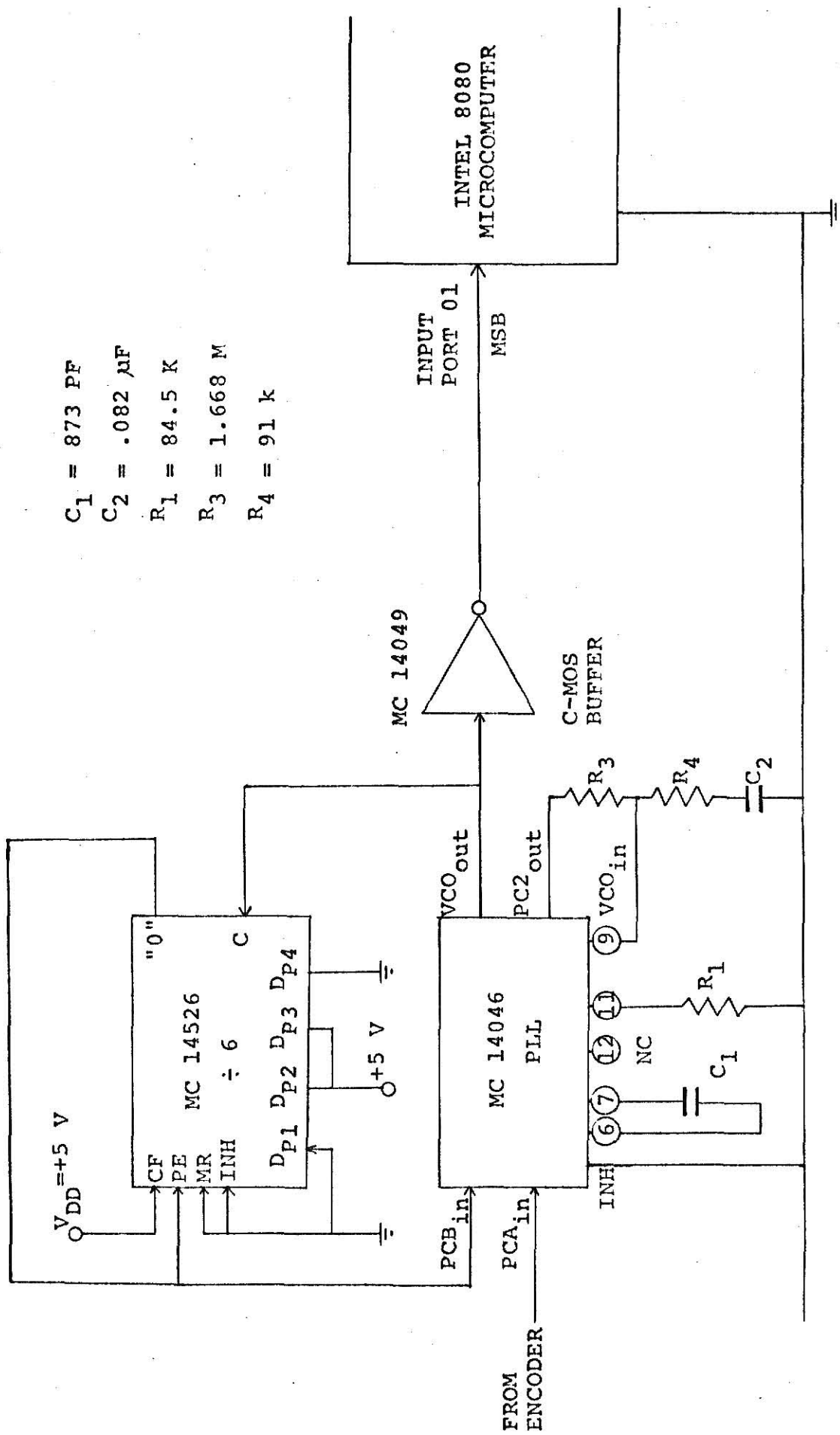
3)

$$R_1 = \frac{2 K_0}{f_{\max} \cdot C_1}$$

4) $K_0 = 0.5$ while supply voltage $V_{DD} = 5$ volts

5) Frequency lock range f_L = full VCO frequency range.

6) Capture range = f_L



$C_1 = 873\text{ PF}$
 $C_2 = .082\text{ }\mu\text{F}$
 $R_1 = 84.5\text{ K}$
 $R_3 = 1.668\text{ M}$
 $R_4 = 91\text{ k}$

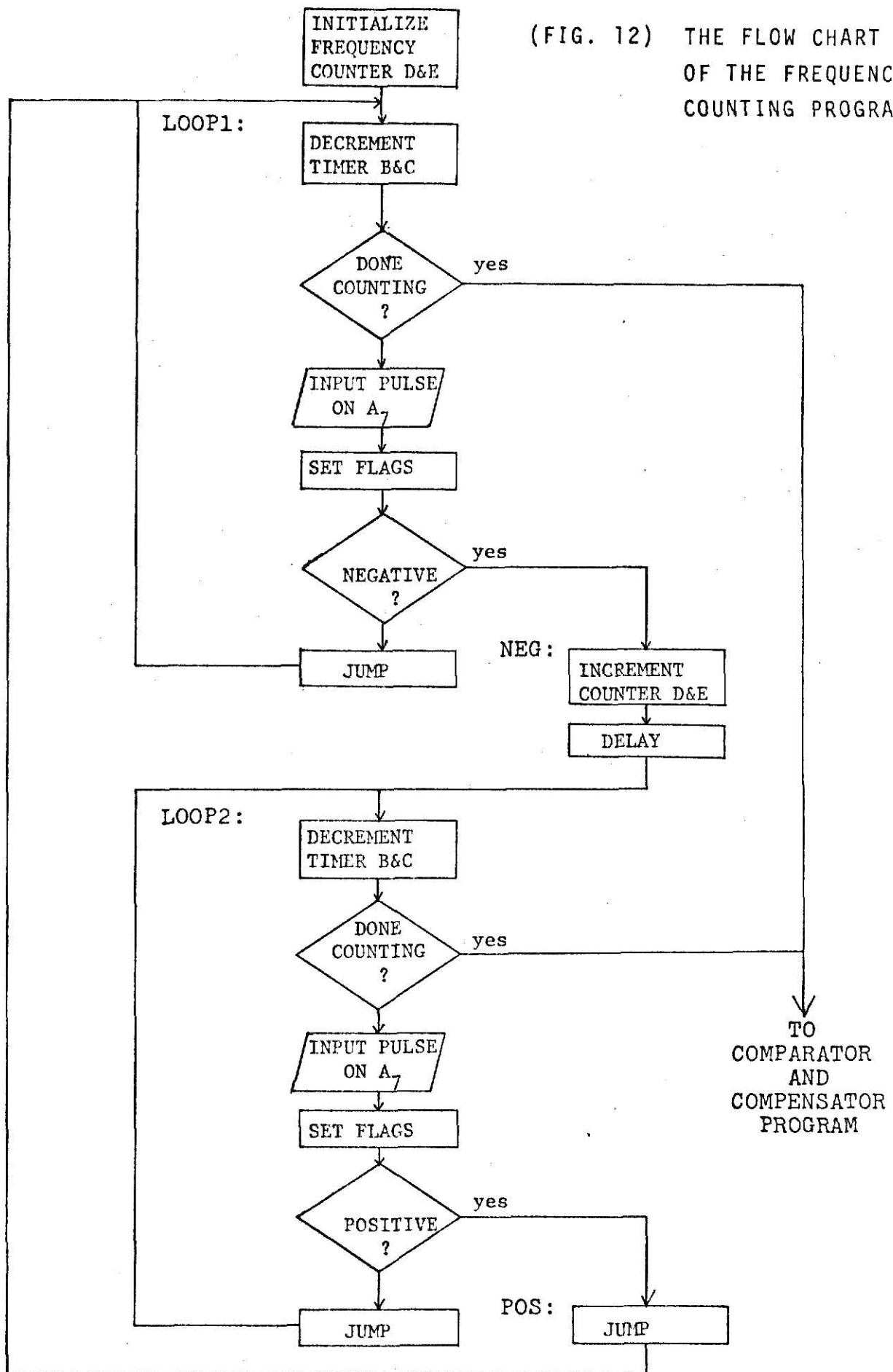
(FIG. 11) THE FREQUENCY MULTIPLIER.

6.4 SOFTWARE COUNTER:

The digital design for the measurement of angular velocity involves the measurement of angular distance during a given time by counting the number of uniformly spaced signals on a disk. The process of counting the pulses is equivalent to an integration over a given interval of time.

The frequency counting routine was designed to sample and measure the frequency from the encoder system. The flow chart of this counting routine is shown in Fig. 12. The MSB of input port 01 was tied to the output of the phase-locked loop. During the sampling time, this bit is checked every 37 microseconds for a change in voltage level. The author assumed a clock cycle of 0.5 microseconds in the Intel 8080. The sampling time could easily be modified to accommodate other microprocessor clock frequencies. The counting routine is separated into two loops (Fig. 12). The microcomputer executes the first loop if the input is high and continues looping and checking the input. If the input goes low, the next check will cause a jump to the next part of the routine. The frequency counter is now incremented and the second loop is initiated. Loop 2 is exactly the same as loop 1 except it checks for a change to a high level at the input. At the beginning of each loop, the timer counter (register pair B & C) is decremented and checked for completion of the measurement period. All four possible loop paths must have equal execution times for the sampling time to be accurate. One of the objectives was to minimize the time between input checks by reducing instruction set and using fast access memory. This "check frequency" must be greater than twice the input frequency which is from the encoder system to ensure that no level transitions are lost. At high frequencies, the periods of some of the VCO pulses could be smaller than twice the "check period" and consequently some level transitions could

(FIG. 12) THE FLOW CHART
OF THE FREQUENCY
COUNTING PROGRAM



go unnoticed. The final count giving the value of the speed in digital form is stored in two 8-bit registers D & E. The counter for timer which keeps track of sampling will be "zero" content that indicates a count of 5426.

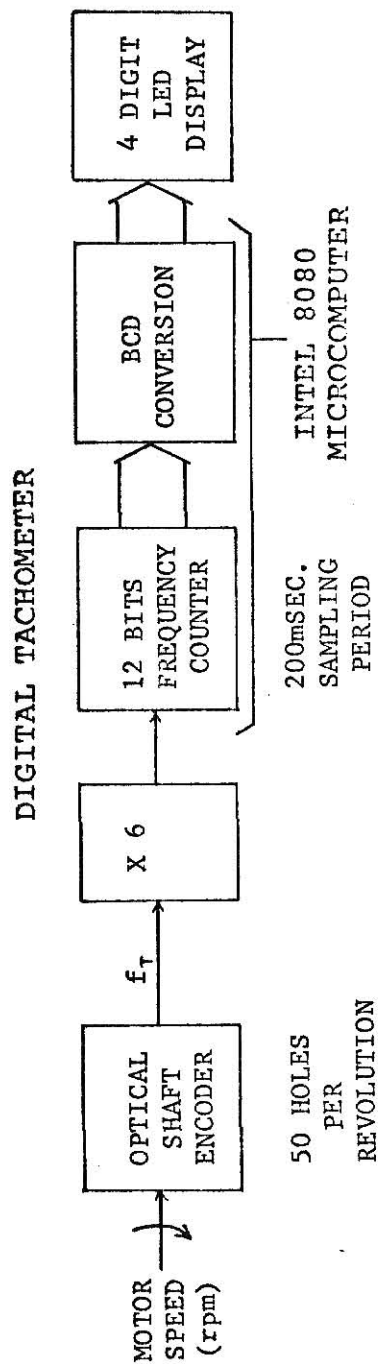
6.5 TEST RESULT OF MEASUREMENT:

In order to verify the accuracy of this digital tachometer which consists of the shaft-encoder, frequency multiplier and software frequency counter, a subroutine is needed to display the speed in rpm on seven segment LED displays. After sampling, the frequency count is in registers D and E. This twelve digit binary number is then converted to a four digit BCD number. This routine (Appendix B) is called BCD translation because it is a specialized division routine. After the translations are complete, the BCD digits are in four memory locations. They are then outputted to ports 02 and 03 and shown on the seven-segment display modules.

A block diagram of this experimental measurement system is shown in Fig. 13. The experimental system described has not been optimized but it has allowed the author to test the concept. The transducer was built with $N_m = 50$ slots and a frequency scaler for times six was selected. Sampling interval was approximately 200 milliseconds for each measurement. A measurable range of angular velocity is from 200 to 2500 rpm. Initial tests using fixed frequency (f_T in Fig. 13) pulses showed that the counter was very stable except in the least significant bit which tended to fluctuate over a period of time. Experimental results with this digital tachometer are shown in Table 3.

6.6 CONCLUSION:

The interval (check period) between successive readings of the input



(FIG. 13) THE DIGITAL TACHOMETER

TABLE 3

EXPERIMENT RESULTS OF THE DIGITAL TACHOMETER

f_T (cps)	ACTUAL SPEED (rpm)	READOUT	ERROR (%)
250	300	301	+0.333
313	375.6	377	+0.373
439	526.8	528	+0.228
549	658.8	660	+0.182
657	788.4	789	+0.076
733	879.6	880	+0.045
828	993.6	994	+0.040
1043	1251.6	1252	+0.032
1158	1389.6	1390	+0.029
1200	1440	1440	0.0
1261	1513.2	1513	-0.013
1364	1636.8	1636	-0.049
1438	1725.6	1724	-0.093
1535	1842	1840	-0.109
1652	1982.4	1980	-0.121
1739	2086.8	2083	-0.182
1841	2209.2	2205	-0.190
1958	2349.6	2344	-0.238
2005	2406	2400	-0.249
2101	2521.2	2514	-0.286

$$f_T = \frac{\text{SPEED (rpm)}}{60} \times 50 = \frac{\text{SPEED (rpm)}}{1.2}$$

frequency is 37 microseconds, so according to the sampling theorem (Ref. 9), the highest input frequency must have a period larger than 2×37 microseconds, i.e., with a frequency of 13.5 Kiloherztzes. Because the checking (reading data from input ports) rate is limited by the speed of the microprocessor, this digital tachometer will not be able to directly measure angular velocity higher than 2700 rpm. Two factors, the "checking period" of the microcomputer and the highest motor speed to be measured, affect the trade-off between the sampling interval and the multiplication factor contributed by the phase-locked loop.

The present digital measurement system has a resolution of 1 rpm (0.105 radians/second) over a speed range of 0-2700 rpm, and it can be extended with a faster microcomputer. The attainable resolution can be derived as follows:

- V - the motor speed in rpm
- H - the number of pulses per disc revolution
- M - the multiplication factor
- P - the measurement interval in second

The resolution of the shaft encoder is equivalent to $H \times M$ pulses per revolution.

$$V \frac{\text{revolution}}{\text{minute}} \times \frac{1 \text{ minute}}{60 \text{ seconds}} \times H \frac{\text{pulses}}{\text{revolution}} \times M \times P \text{ second}$$

$$= \frac{V \times H \times M \times P}{60} \text{ (pulses)}$$

The resolution of this digital measurement system becomes

$$\frac{60}{H \times M \times P} \frac{\text{RPM}}{\text{PULSE}}$$

In the prototype system, H was chosen to be 50, M=6, P=0.2 to yield
 1 rpm = 1 pulse = 1 binary bit, since one pulse will be stored in the
 frequency counter as one binary bit.

The reader should be aware of the fact that the use of the frequency multiplier which was used to improve the resolution of the speed measurement system is acceptable only if the following assumption is true. If the response of the controlled motor is not too fast, then the motor speed can be assumed to be constant during the time duration between two adjacent slots on the shaft encoder. If for example, the disk with 50 slots runs at a speed of 200 rpm, then the time duration between each slot can be derived as follows:

$$200 \frac{\text{revolution}}{\text{minute}} \times \frac{1 \text{ minute}}{60 \text{ seconds}} \times \frac{50 \text{ slots}}{\text{revolution}}$$

$$= 166.67 \frac{\text{slot}}{\text{second}}$$

The time duration between each slot will be $1/166.67 = 0.006$ seconds.

In this case, the time duration is one-seventh of the motor time constant 0.04, used in the prototype system. Therefore, the assumption is valid.

Intellec 8 uses slow-access RAM of 1 microsecond access time, therefore, different instruction may end up with different numbers of waiting states.

By adding some dummy instruction could make each checking loop (Fig. 12) having exactly same execution time. But this approach will extend the checking period and shorten the measurable range of the motor speed.

Because the pulses are gated into the counter during a time period which is not precisely controlled, the number stored in the counter at the end of the sampling time represents the approximate motor speed. This explains why readings had errors of about $\pm 0.3\%$. In general, by using fast microcomputers can make out a system with high accuracies and large spectrum of measurable

motor speed. Test result shows that this digital speed measurement system appears to be practical. Followings are some design considerations of the digital tachometer.

(1) checking period = 37 microseconds = T_r = the execution time for each checking loop shown in Fig. 12.

(2) maximum input frequency = $1/2T_r = 13.5$ KiloHertz

(3) motor speed (rpm) $\times \frac{50 \text{ pulses/revolution}}{60 \text{ second/minute}} = \frac{\text{MOTOR SPEED (pulses)}}{1.2 \text{ seconds}}$
 $= f_T$ (frequency output from shaft encoder)

(4) USING PLL ($\times 6$)

f_m (input frequency into the frequency counter) = $6 \times \frac{\text{MOTOR SPEED}}{1.2}$

= $5 \times \text{MOTOR SPEED} = 10 \text{ KHz @ } 2000 \text{ rpm}$

(5) SCALING

$\frac{2000 \text{ (counted pulses)}}{10 \text{ KHz}} = 200 \text{ milliseconds.}$

$\frac{200 \text{ (millisecond sampling interval)}}{37 \text{ microseconds}} = 5405.4 \text{ LOOPS}$

In the prototype system, there are totally 5426 checking loops during the 200 milliseconds interval.

(6) MEASURABLE MAXIMUM SPEED

$(13.5 \text{ KHz}) \times (200 \text{ milliseconds}) = 2700 \text{ pulses (rpm)}$

(7) TRANSFER FUNCTION OF THE SPEED MEASUREMENT SYSTEM

gain constant: velocity $\frac{\text{radian}}{\text{second}} \times \frac{50 \text{ pulses}}{2\pi \text{ radians}} \times 6 \times 0.2 \text{ second}$

= velocity $\frac{\text{radian}}{\text{second}} \times 9.55 \frac{\text{pulses}}{\text{radian/second}}$

frequency counter: functioning as an integrator $1/S$

The overall transfer function is $H_s(s) = \frac{9.55 \text{ pulses}}{s \text{ radian/second}}$.

CHAPTER 7

SPEED CONTROL PROGRAM

7.1 INTRODUCTION:

The basic principle of computer speed control for a DC motor is illustrated in the block diagram of Fig. 3. The amount of phase shift between two pulse train e_1 and e_2 is adjusted according to the difference between the desired and the actual motor speed. The sequence of actions may be stated as follows: The reference input (12 bit binary settings) is set according to the speed desired. If the actual speed drops below the desired value, then the increase in the control-counter of the compensator in the Intel 8080 causes the pulse e_2 to move backward (to the right in Fig. 6), supplying more current from the SCR chopper, and the speed of the motor increases until the desired value is reached. On the other hand, if the speed of the motor increases above the desired value, then the decrease in the control counter of the compensator in the Intel 8080 causes the pulse e_2 to move forward (to the left in Fig. 6). This decrease the supply of current from the chopper and the speed of the motor decreases until the desired value is reached.

7.2 HARDWARE CONFIGURATION:

The pulse generator with programmable delay is connected to two parallel output ports on the computer. Data sent to one pair of ports may cause the "2nd pulse" to change from where it was last to the new position specified by the data output. The same program action using another pair of ports just sent the high order byte of data to the pulse generator.

ordinary output instructions are used to send data to the pulse generator. A complete data (12-bit) set is sent to the pulse generator only when the data on port 03 is sent out of the microcomputer.

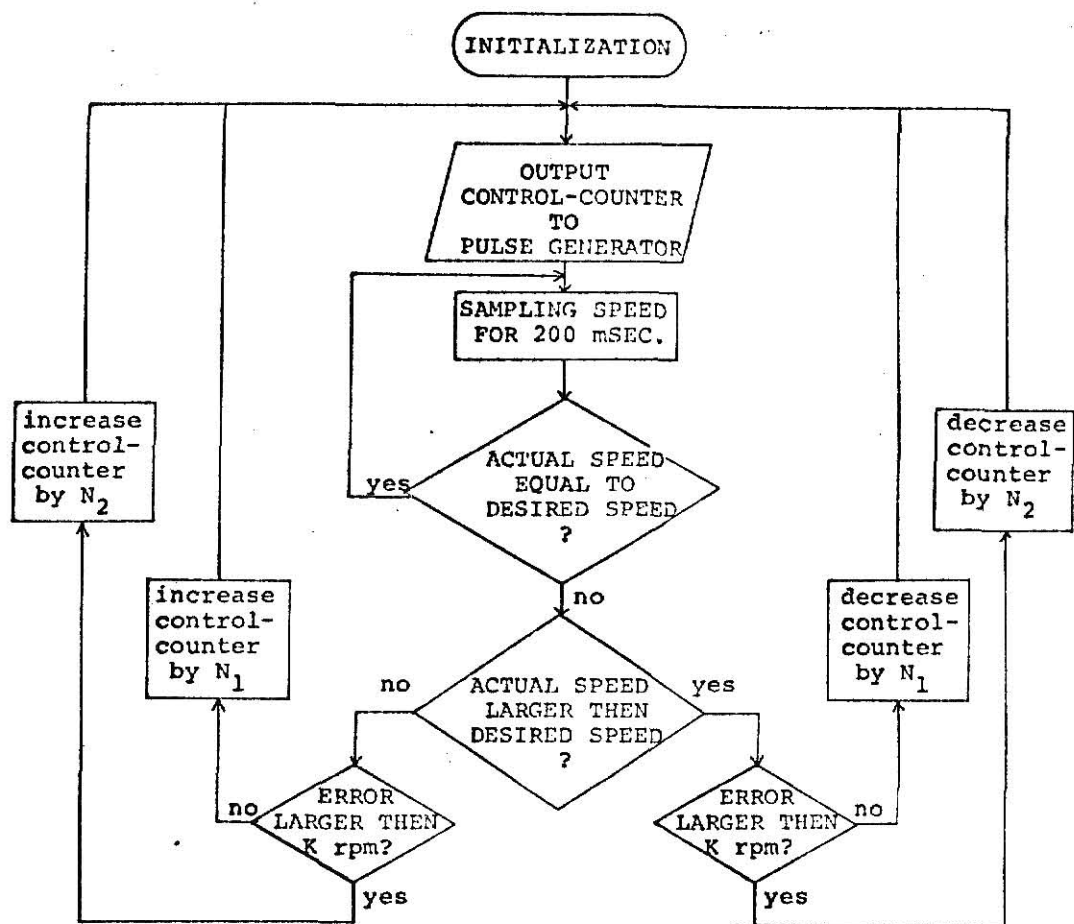
7.3 DATA INPUT:

Input data for a desired speed can be loaded in computer in two ways. The first one is to have the desired speed stored in some specific memory location. A very simple program loop is used to work through the list and send the data of desired speed to the accumulator. This technique can save a lot of toggle switches in the input port and control the speed in a defined sequence. The second method might be the simplest one. In a system with input devices (such as, wheel switch), data can be read from input ports and sent directly to the accumulator without storing it in memory. In most speed control application, the combination of these two methods might be used. The prototype system is using memory access method. By using those switches on front panel of microcomputer, desired speed is stored into reserve memory location 2000H and 2001H in which is going to compare with the feedback signal (actual speed). While loading the input data, the microcomputer must be in the memory access mode instead of running mode (Ref. 3).

7.4 SOFTWARE CONFIGURATION:

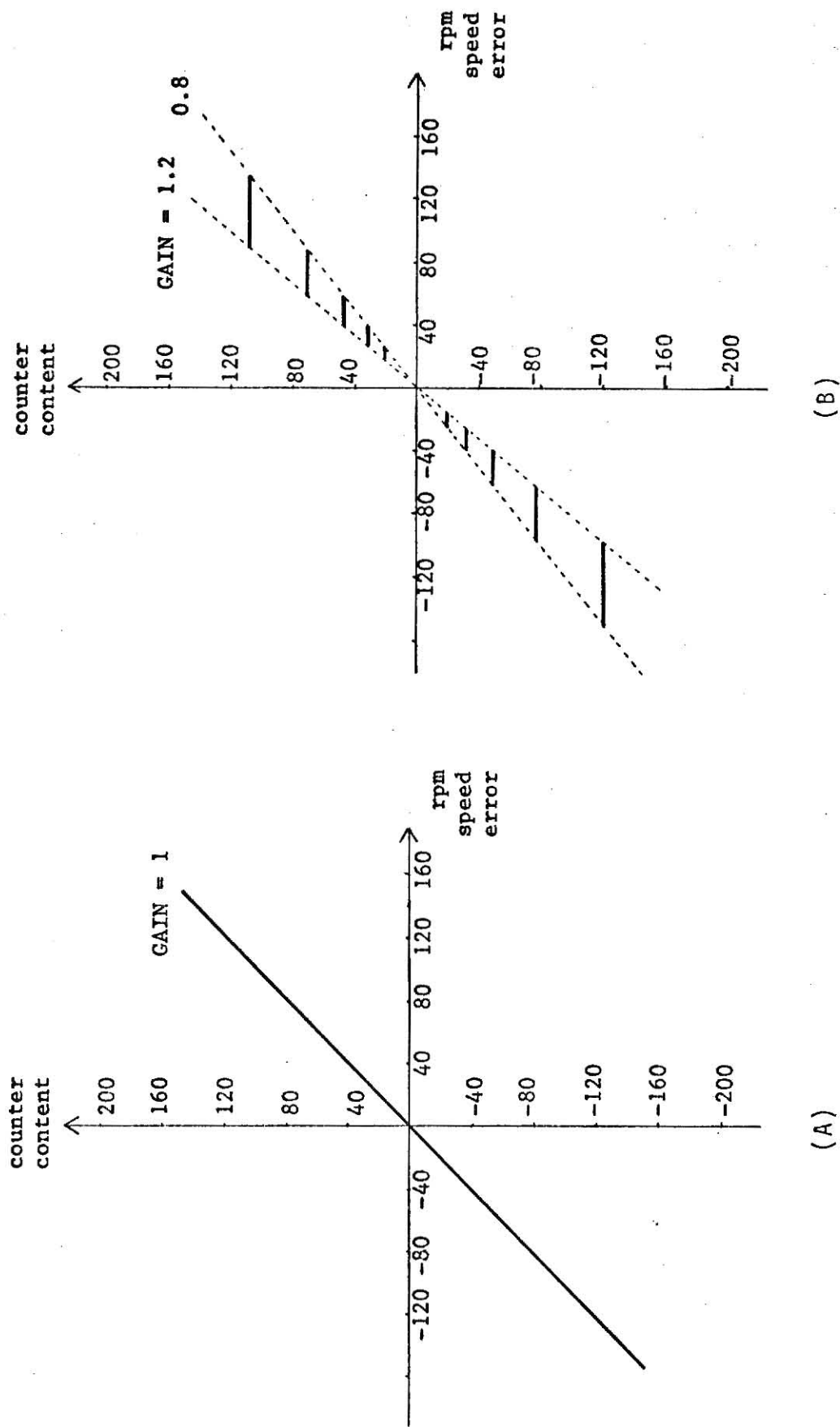
A detailed flow chart of the system's software would be too complicated to present or discuss here. However, an understanding of some of the considerations involved may be obtained from a simplified flow chart of the most basic portion of the program. The software in microprogramming is generating the data (content of control counter) to be sent to the pulse

generator. One of the decisive operation is to increase or decrease the control counter according the difference of speed. The output of the comparison circuit depends on which of three conditions prevails; either the set point is larger than actual speed, the set point is smaller than actual speed, or the set point and actual speed are exactly equal. The comparator output is used in two ways. First, it decides whether the compensator (control counter) is to be driven in the up direction or the down direction. Driving the control counter in the up direction increases the motor armature current and increases motor speed, whereas driving it in the down direction decreases the current and motor speed. If the motor speed and desired speed are equal, the comparator inhibits the control counter from changing state. Secondly, it decides how much the content (N_1 or N_2 shown in Fig. 14) of the compensator needs to be adjusted. After the error has been developed by comparison, multiplication was usually used to acquire the proper amount of compensation. The multiplication factor was considered as the gain of that compensator. The multiplication algorithm itself is straight forward, but double register testing and shifting makes it difficult to implement. Note that any computer can only fetch, add, store, and compare. An estimation has been made that a multiplication subroutine dealing with 12-bit binary variables will be approximately 160 bytes long. An alternative is used and its simplified flow chart is shown in Fig. 14. Because computers operate only on numbers, adding and comparing are important functions in the programming. The Intel 8080's powerful instruction set has an DAD instruction (add register pair to H & L) (Ref. 7) which functions to add two register pairs (two eight bit words per register pair) together. This instruction DAD can make a comparison of two 16-bit variables quite easy. The transfer characteristic shown in Fig. 15(B) is



* $N_2 > N_1$

(FIG. 14) THE SIMPLIFIED FLOW CHART OF THE PROGRAMMING FOR THE MICROPROCESSOR SPEED CONTROL.

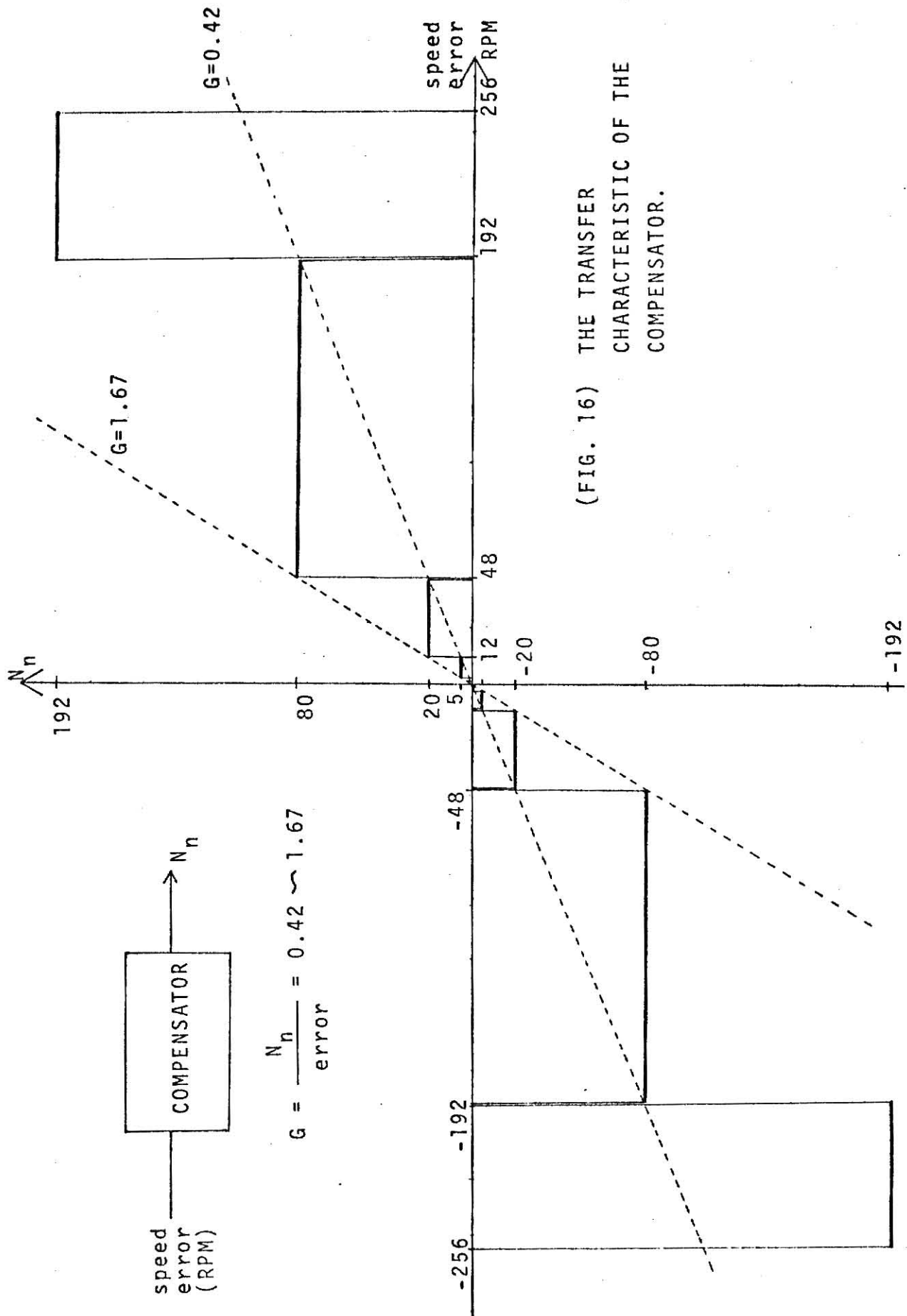


(FIG. 15) THE TRANSFER CHARACTERISTIC OF (A) IDEAL MULTIPLICATION
(B) QUANTIZED-MULTIPLICATION.

the quantized-multiplication subroutine that simply has addition involved. If the quantization size (parameter K in Fig. 14) is decreased in the transfer characteristic, that quantized-multiplication subroutine could become very similar to multiplication. The gain of the compensator, i.e., the slope of the dotted line in Fig. 15(B) could easily be modified just by changing the parameters (N_1 , N_2 , and K as shown in Fig. 14) in the software program. It is obvious that the gain of the compensator is varied according to the magnitude of the rpm error. From the simplified flow chart shown in Fig. 14, small errors usually are corrected quickly by adjustment of the phase shift. However, large errors will require more comparison steps before the pulse generator can properly be adjusted. This strategy not only stabilizes the speed control system but also prevents the current which is driving the motor from abruptly changing when correcting a large error in speed. These considerations become very important when the motor is applied to a big system.

7.5 CONCLUSION:

The transfer curve of the prototype system is shown in Fig. 16. It is a very rough approximation to the ideal multiplier transfer curve depicted in Fig. 15(A). The test results show the performance of the prototype system successfully meet the requirements. As information is updated, the computer initiates a transfer command to load the appropriate data from the control counter into the pulse generator. The program must continuously execute a loop to keep the speed constant. The transfer function of this comparison and compensation unit is the varying gain ranging from 0.42 to 1.67 shown in Fig. 16.



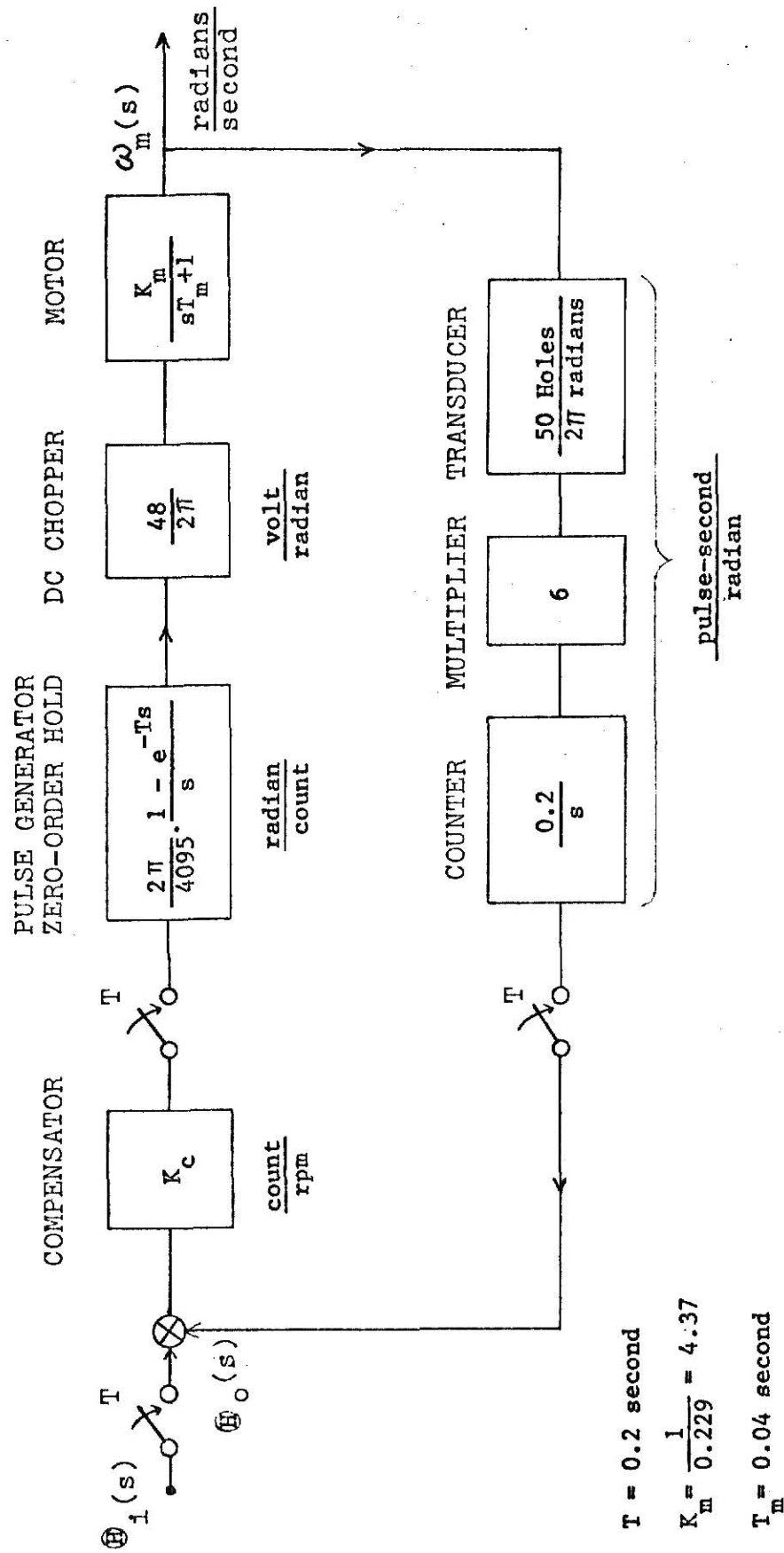
(FIG. 16) THE TRANSFER CHARACTERISTIC OF THE COMPENSATOR.

CHAPTER 8

SYSTEM ANALYSIS AND DESIGN

Any control system must be stable. This is primary requirement. In addition to absolute stability, a control system must have a reasonable relative stability. That is, the speed of response must be reasonably fast and this response must show reasonable damping. A control system must also be capable of reducing errors to zero or to some small tolerable value. The requirement of reasonable relative stability and that of steady state accuracy tend to be incompatible. In designing control system, we therefore find it necessary to make the most effective compromise between these two requirements. The basic approach to the design of any practical control system will necessarily involve trial-and-error procedures. The synthesis of linear control system is theoretically possible. In practice, however, the system may be subjected to many constraints or may be nonlinear. In addition, the characteristic of components may not be precisely known. Thus trial-and-error procedures are always necessary. The prototype has being constructed are using programmable counters and microcomputer. Those hardware and software configurations are very flexible. The prototype system can be modified very easily and repeated many times until a satisfactory system is obtained.

Since any system is made up of components, analysis must ^{START} with a mathematical description of each component. The transfer function block diagram is shown in Fig. 17. The prototype system can be considered as a discrete time system. Data can be changed only at discrete instants of time. These instants, which shall be denoted by KT ($K=0,1,2,\dots$), specify the time at



(FIG. 17) THE BLOCK DIAGRAM OF THE DISCRETE-TIME SYSTEM

which some physical measurement is made or the time at which the memory of a digital computer is read-out. Since sampling intervals take 200 milliseconds while measuring the speed, speed error and data of phase angle are available in digital form 5 times per second. Thus, sampling period T equals to 0.2 seconds. Because the zero-order hold simply holds the magnitude of the input impulse at the sampling instant until the next input comes along, the output signal is a sequence of steps. The pulse generator output unit, which acts like a zero-order holding device, produces the proper phase shift, which is piecewise constant from the last input value until the next input is available.

Strictly speaking, the prototype speed control system (as shown in Appendix C) is a multirate sampled-data control system with nonsynchronized samplers and with transport lags. The system contains two synchronized samplers which near the frequency counter, one operating at a rate $1/T$, and the other being 5426 times as fast. In digital control systems, the time for computation or processing command reflects as a transport lag in the system. Two other samplers which simply transfer the data to the next stage operate with the same rate but are not synchronized in phase. In order to apply the commonly used Z-transform techniques, it is essential that the delayed sampler be represented by an equivalent sampler, which operates in synchronism with the basic sampler, preceded by an advance element, $e^{\Delta T_s}$. ΔT_1 is the time for comparison and ΔT_2 is the time for compensation. If comparing the delayed ($\Delta T = 32\text{-}80$ microseconds) and faster ($T/5426 = 37$ microseconds) samplers with the basic samplers ($T = 200$ milliseconds), the reader should note that the delayed effect and the faster sampler become insignificant. Some reasonable ignorances are necessary to simplify the analysis. The system shown in Fig. 17 is the single rate sampled-data

system. All the samplers are assumed to operate in synchronism with the same sampling rate ($T = 200$ milliseconds). $\Theta_1(s)$ is the number which stands for the desired speed setting in rpm. $\Theta_0(s)$ is the output data from the speed measurement system, which stands for the motor speed in rpm. The transfer function for the plant and controller in the forward path is derived as follows:

$$\begin{aligned} G(s) &= \frac{2\pi}{4095} \frac{1 - e^{-0.2s}}{s} \frac{48}{2\pi} \frac{4.37}{0.04s + 1} \\ &= \frac{0.0512}{s} \frac{1 - e^{-0.2s}}{1 + 0.04s} \\ &= \frac{1.28 (1 - e^{-0.2s})}{s (s + 25)} \end{aligned}$$

$$H(s) = \frac{0.2}{s} \cdot 6 \cdot \frac{50}{2\pi} = \frac{9.55}{s} \frac{\text{rpm} \cdot \text{second}}{\text{radians}}$$

$H(s)$ is the transfer function for the feedback path. By using the partial fraction expansion, $GH(s)$ becomes

$$\begin{aligned} GH(s) &= \frac{12.224 (1 - e^{-0.2s})}{s^2 (s + 25)} \\ &= (1 - e^{-0.2s}) \left[\frac{0.489}{s^2} - \frac{0.01956}{s} + \frac{0.01956}{s + 25} \right] \end{aligned}$$

From the simplified block diagram in Fig. 18, it is easy to find the pulse transfer function of the discrete-time closed-loop system as follows:

$$E(s) = R(s) - H(s)C(s)$$

$$C(s) = G(s)D^*(s)$$

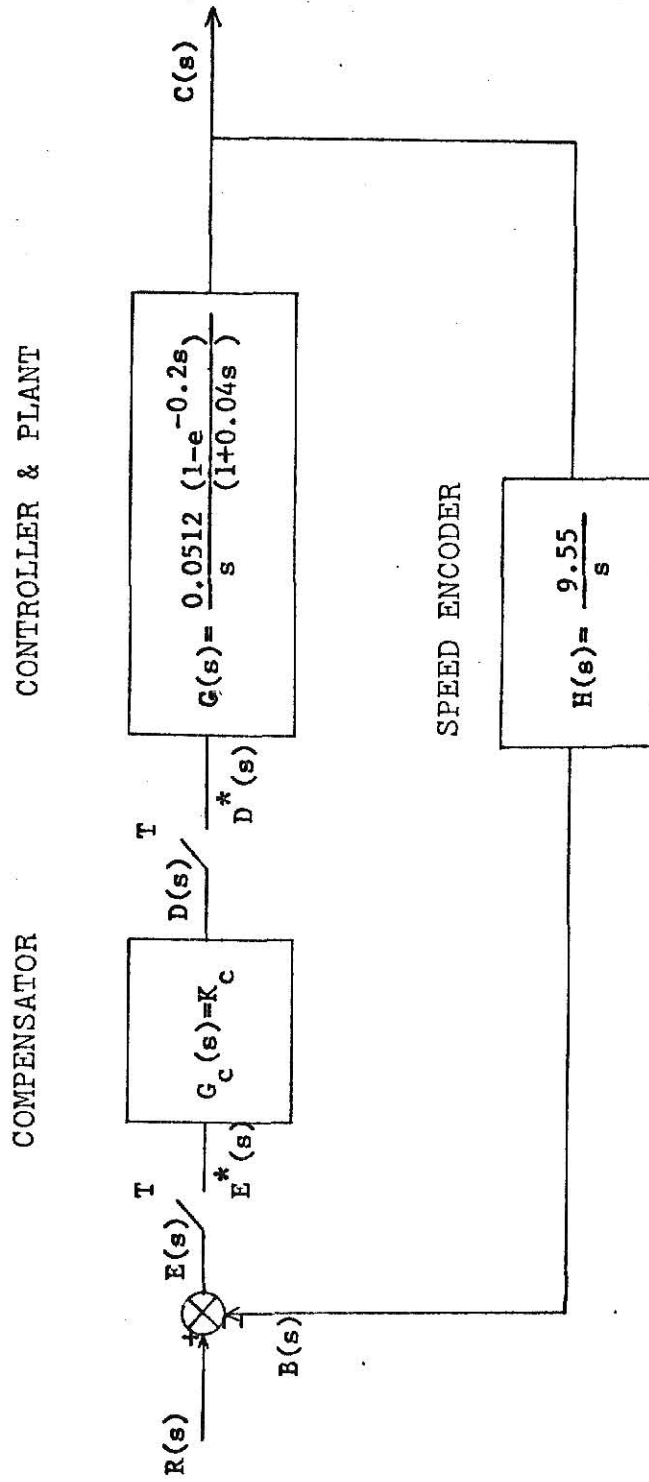
$$D(s) = G_C(s)E^*(s)$$

$$E(s) = R(s) - H(s)G(s)D^*(s)$$

$$D^*(s) = G_C^*(s)E^*(s)$$

$$E^*(s) = R^*(s) - HG^*(s)D^*(s)$$

$$D^*(s) = G_C^*(s)R^*(s) - G_C^*(s)HG^*(s)D^*(s)$$



(FIG. 18) BLOCK DIAGRAM REDUCTION OF THE SYSTEM IN FIG. 17.

$$D^*(s) = \frac{G_c^*(s) R^*(s)}{1 + G_c^*(s) H G^*(s)}$$

$$C^*(s) = G^*(s) D^*(s) = \frac{G_c^*(s) G^*(s) R^*(s)}{1 + G_c^*(s) H G^*(s)}$$

Upon taking the Z-transform corresponding to $C^*(s)$, it is

$$C(Z) = \frac{G_c(Z) G(Z) R(Z)}{1 + G_c(Z) H G(Z)}$$

The pulse transfer function of the entire feedback system is

$$\frac{C(Z)}{R(Z)} = \frac{G_c(Z) G(Z)}{1 + G_c(Z) H G(Z)}$$

Up taking the Z-transform corresponding to $GH(s)$, it becomes

$$\begin{aligned} GH(Z) &= \frac{0.0978 Z}{Z - 1} - \frac{2.8834 Z}{Z - e^{-5}} \\ &= \frac{0.0978 Z}{Z - 1} - \frac{2.8834 Z}{Z - 0.00674} \end{aligned}$$

$$G_c(Z) = \frac{K_c Z}{Z - 1}$$

$$G_c(Z) GH(Z) = \frac{K_c Z}{Z - 1} \left(\frac{0.0978 Z}{Z - 1} - \frac{2.8834 Z}{Z - 0.00674} \right)$$

The stability of the closed-loop system shown in Fig. 18 will now be discussed.

The stability of such a system can be determined from the location of the roots of the characteristic equation

$$1 + G_c(Z) GH(Z) = 0.$$

For stability all the roots Z_i of the characteristic equation must lie inside the unit circle, or $|Z_i| < 1$ (Ref. 14).

The characteristic equation becomes

$$Z^3(1 - 2.7856 K_c) + Z^2(-2.00674 + 2.88274 K_c) + 0.01348 Z - 0.00674 = 0$$

Noting that $K_c = 1.67$ we can simplify the characteristic equation to

$$3.6525 Z^3 - 2.8074 Z^2 - 0.01348 Z + 0.00674 = 0$$

from which the roots are found to be

$$Z_1 = 0.770 \quad Z_2 = -0.050 \quad Z_3 = 0.048$$

Thus, any root of the characteristic equation has a magnitude less than unity, and the system is stable. Because the poles lie on the real-axis of the Z-plane, the entire system response is overdamp. In the physical sense, it is in accordance with the nature of a nonsynchronized and multirate sampled-data control system with slight transport lags (Ref. 14).

It is important to remember that in the Z-transform approach, or pulse transfer function approach, to the analysis of discrete-time system, the sampled signal is assumed to be a train of impulses whose strengths, or areas, are equal to the continuous-time signal at the sampling instants. Such assumption is valid only if the sampling duration of the sampling is very short compared with the most significant time constant of the system. However, the programmable pulse generator will response the change for every 200 milliseconds. The "plant" (motor) response is much more faster than that of "controller" (pulse generator and microcomputer). Therefore, the response for the time between these sampling points can not be predicted. That is why only the stability analysis has been carried out in the Z-plane.

Modified Z-transform may be used to find out the exactly response of the system. However, the analysis using modified Z-transform will not be carried out here.

CHAPTER 9

FUTURE EXTENSIONS AND IMPROVEMENTS

The dynamic characteristics of most control systems are not constant because of the changes in parameters and environment (for example, changes in load). Although the effects of small changes on the dynamic characteristics are attenuated in a feedback control system, large changes in the system parameters and environment are significant; a preferred system would be adaptable. An adaptable system is one which has the ability to change it's dynamic characteristics to improve performance when perturbed by large changes in environment or structure.

Microcomputers provide many advantages in size and flexibility which of for many possibilities for adaptive control in the systems utilizing microprocessors. For an adaptive DC motor control scheme, information about the response, stability, and loop gain would be fed to the computer at certain time intervals. The complex computations for determining the optimal loop gain and settling time would be handled by the computer. From the comparison between the actual speed and the desired speed, the computer could calculate the magnitude of the load change and determine which control algorithm to use to restore the system to equilibrium..

The average acceleration during the speed measuring interval can be obtained by dividing the change in speed by the time interval. This can be done by using a division subroutine.

The activities of the Intel 8080 microprocessor are timed by a master clock oscillator. The clock period determines the timing of all processing activity. The speed of the processing cycle, however, is limited by the

memory's "Access Time". Once the processor has sent a read address to memory, it cannot proceed until the memory has had time to respond. Most memories are capable of responding much faster than the processing cycle requires. A few, however, cannot supply the addressed byte within the minimum time established by the processor's clock. Therefore, a processor should contain a synchronization provision which permits the memory to request a "Wait State". When the memory receives a read or write enable signal, it places a request signal on the processor's READY line, causing the CPU to idle temporarily. After the memory has had time to respond, it frees the processor's READY line, and the instruction cycle proceeds (Ref. 7). Actual, in the Intellec 8 microcomputer, the access time at the memory is longer than that at the processor. During instruction or data fetch, there is always one "Wait" state in the instruction cycle for each memory access. The invasion of the "Wait" state will cause the execution time of the instruction last longer. Fast access time memory will be advantageous in a microcomputer system. Not only the full utilization of the CPU will be achieved, but the response and accuracy of the control system will be improved. While the speed of response is not the important characteristic of a system, the microcomputer does provide a very accuracy control without increasing the cost and the complexity of the system configuration. If a high performance system is needed, further evaluation of the tradeoffs between the speed of response and accuracy is necessary.

Further improvements can be made by making the speed encoder more accurate and using a fast access memory and high performance CPU in the microcomputer. If the speed value of one pulse is reduced, the control error is reduced accordingly. This can be accomplished by simply generating more pulses per shaft revolution. For example, if each shaft revolution generated

100 pulses instead of 50 pulses, and if sampling time were held at 200 milliseconds, the value of a single pulse would be 0.5 rpm. Naturally, the control circuitry would have to be more extensive to accomplish more precise speed control. For a maximum speed of 3000 rpm, instead of control based on a 12-bit number, all counters, comparators, and registers would have a length of 13 bits. Such an improvement in resolutionⁿ would also place more stringent demands on the pulse generator and on the SCR chopper which drives the motor armature.

The digital speed control system can be implemented strictly on random logic circuits or only by microcomputer instead. The reason by using random logic to build the firing circuit is the consideration of tradeoff between the performance and cost. Hardware cost reductions and greater flexibility can be achieved if a microprocessor is used instead of using random logic only.

CHAPTER 10

CONCLUSION

A motor speed control system has been described that is entirely digital except for the output device itself. The concept being investigated should work well for small installations as well as large ones. Such a system eliminates the design of elaborate DC power amplifiers that have drift or off-set associated with their operation. The design of the chopper used in this system is much simpler than is the corresponding design of a DC power amplifier. C-MOS integrated circuits offering very low stand-by-power dissipation and high noise immunity are used throughout. In addition, electro-optical couplers are used for isolating the signal circuits from the power circuits.

The microprocessor motor control described in this paper was designed to achieve accurate speed regulation over a range of speeds from 200 rpm to 2500 rpm. By measuring the frequency of the pulse train from the shaft encoder, it was shown that the motor speed varied less than 0.2% at the low end of the range and less than 0.02% over the majority of the range.

The use of a microprocessor represents a very satisfactory solution in the application discussed here. The task could have been performed by a hard-wired controller, but many of the conveniences and the flexibility obtained with the microprocessor would have been sacrificed. If necessary, the software program for speed control can be upgraded in a simple manner. High-performance CPU and fast access memory could have a major impact on the system performance. Although the microprocessor solution to the problem

may not be economical, the convenience of operation of the system and the flexibility offered by possible stored program changes in the future justify the cost.

The present program is executed on an "Intellec 8" (Ref. 3) and occupies 329 bytes in RAMs. A special purpose microcomputer for this speed control application can be built using only inexpensive EPROMs or metal masked ROMs without any RAM.

APPENDIX A

MAIN PROGRAM FOR THE SPEED CONTROL

8080 MACRO ASSEMBLER, VER 2.3 ERRORS = 0 PAGE 1

```

; SPEED CONTROL PROGRAM
00F8      ORG      0F8H
00F8      LXI      SP, 8000
00FB      LXI      H, 0065H ; INITIALIZE OUTPUT COUNTER
00FE      CD3302   COUCN: CALL OUTCN
0101      7C       CHELO: MOV     A, H ; 'DOWN' FLOW
0102      87       ORA      A ; CHECK IF A=0
0103      C21501   JNZ      TIMER
0106      7D       MOV     A, L
0107      FE64     CPI      100
0109      D21501   JNC      TIMER
010C      3EAF     WARN0: MVI     A, 0AFH ; COMPLEMENT OF 50H
010E      0303     OUT     03 ; OUTPUT WARNING '1' AND DISABLE PULSE
0110      3EEF     MVI     A, 0EFH ; COMPLEMENT OF 10H
0112      0303     OUT     03 ; DISABLE PRESET
0114      76       HLT
0115      013215   TIMER: LXI     B, 5426 ; LOAD NUMBER OF TIMING LOOPS
0118      110000   LXI     D, 0000 ; ZERO FREQUENCY COUNTER
0118      0B       LOOP1: DCX     B ; DECREMENT TIMER
011C      78       MOV     A, B ; VALIDATE ZERO FLAG, CY CLEARED
011D      81       ORA      C
011E      CA3E01   JZ       DONE ; TIMING LOOP FINISHED?
0121      0B01     IN       1 ; INPUT FREQUENCY ON MS3 OF PORT 1
0123      A7       ANA      A ; MINUS FLAG IS VALIDATED
0124      FA2A01   JM       NEG ; IS MSB OF REG A HIGH?
0127      C31B01   JMP     LOOP1
012A      13       NEG: INX     D ; INCREMENT FREQUENCY COUNT
012B      7F       MOV     A, A ; DUMMY INSTR FOR DELAY
012C      0B       LOOP2: DCX     B
012D      78       MOV     A, B
012E      81       ORA      C
012F      CA3E01   JZ       DONE
0132      0B01     IN       1
0134      A7       ANA      A ; MINUS FLAG IS VALIDATED
0135      F23B01   JP       POS ; IS MSB OF REG A LOW?
0138      C32C01   JMP     LOOP2
0139      C31B01   POS: JMP     LOOP1
013E      3A1110   DONE: LDA     1011H ; LOAD HIGH ORDER BYTE OF DESIRE SPEED
0141      92       SUB     D ; COMPARE WITH HIGH ORDER BYTE OF ACTUAL
                                ; SPEED
0142      CA7C01   JZ       LOKLW
0145      D25901   JNC      CP256
0149      FEFF     CPI      -1
014A      CAF201   JZ       CHLOW
014D      FEFC     CPI      -4
014F      D29A01   JNC      DN192
0152      1100FE   LXI     D, -512
0155      19       ADDDN: DAD     D ; DECREASE RP. HCL
0156      C3FE00   JMP     COUCN
0159      FE01     CP256: CPI      1
015B      CA1202   JZ       CPL0W

```

8080 MACRO ASSEMBLER, VER 2.3 ERRORS = 0 PAGE 2

015E	FE04		CPI	4	
0160	DA0001		JC	IN192	
0163	110002		LXI	D, 512	
0166	19	ADDUP:	DAO	D	:INCREASE RP. HCL
0167	CD3302	CPUCN:	CALL	OUTCN	
016A	7C	CHEOV:	MOV	A, H	:CHECK OVER FLOW
016B	D60F		SUI	OFH	
016D	FA1501		JM	TIMER	
0170	C20C01		JNZ	WARNO	
0173	7D		MOV	A, L	
0174	FEA0		CPI	0A0H	:SEE IF OUTPUT HCL OVER 4000
0176	DA1501		JC	TIMER	
0179	C30C01		JMP	WARNO	
017C	3A1010	LOKLW:	LDA	1010H	:LOAD LOW ORDER PART OF DESIRE SPEED
017F	93		SUB	E	:COMPARE WITH LOW ORDER PART OF ACTUAL :SPEED
0180	CA1501		JZ	TIMER	
0183	D28C01		JVC	PLUS	
0186	FEFE		CPI	-2	
0188	D2A001		JNC	DNR	
018B	FEF5		CPI	-11	
018D	D2A401		JNC	DN5	
0190	FED1		CPI	-47	
0192	D2AA01		JNC	DN20	
0195	FE41		CPI	-191	
0197	D2B001		JNC	DN80	
019A	1140FF	DN192:	LXI	D, -192	
019D	C35501		JMP	ADDON	
01A0	28	DNR:	OCX	H	
01A1	C3FE00		JMP	COUCN	
01A4	11FBFF	DN5:	LXI	D, -5	
01A7	C35501		JMP	ADDON	
01AA	11ECFF	DN20:	LXI	D, -20	
01AD	C35501		JMP	ADDON	
01B0	1180FF	DN80:	LXI	D, -80	
01B3	C35501		JMP	ADDON	
01B6	1180FF	DN128:	LXI	D, -128	
01B9	C35501		JMP	ADDON	
01BC	FE03	PLUS:	CPI	3	
01BE	DA0601		JC	IN1	
01C1	FE0C		CPI	12	
01C3	DADA01		JC	IN5	
01C6	FE30		CPI	48	
01C8	DAE001		JC	IN20	
01CB	FE00		CPI	192	
01CD	DAE601		JC	IN80	
01D0	11C000	IN192:	LXI	D, 192	
01D3	C36601		JMP	ADDUP	
01D6	23	IN1:	INX	H	
01D7	C36701		JMP	CPUCN	
01DA	110500	IN5:	LXI	D, 5	

8080 MACRO ASSEMBLER, VER 2.3 ERRORS = 0 PAGE 3

```

01DD C36601 JMP ADDUP
01E0 111400 IN20: LXI D, 20
01E3 C36601 JMP ADDUP
01E6 115000 IN80: LXI D, 80
01E9 C36601 JMP ADDUP
01EC 118000 IN128: LXI D, 128
01EF C36601 JMP ADDUP
01F2 3A1010 CMLOW: LDA 1010H
01F5 93 SUB E
01F6 FE01 CPI 1
01F8 DA8601 JC DN128
01FB FEFE CPI 254
01FD D2A001 JNC DNR
0200 FEF5 CPI 245
0202 D2A401 JNC DNS
0205 FED1 CPI 209
0207 D2AA01 JNC DY20
020A FE41 CPI 65
020C D2B001 JNC DN80
020F C39A01 JMP DN192
0212 3A1010 CPLLOW: LDA 1010H
0215 93 SUB E
0216 CAEC01 JZ IN128
0219 D2EC01 JNC IN128
021C FE03 CPI -253
021E DAD601 JC IN1
0221 FE0C CPI -244
0223 DADA01 JC IN5
0226 FE30 CPI -208
0228 DAE001 JC IN20
022B FE0C CPI -64
022D DAE601 JC IN80
0230 C30001 JMP IN192
0233 7D OUTCN: MOV A, L
0234 2F CMA
0235 D302 OUT 02 ;OUTPUT LOW ORDER BYTE
0237 7C MOV A, H
0238 F640 ORI 40H ;PRESET '1' AT BIT 6 ON PORT 03
023A 2F CMA
023B D303 OUT 03 ;OUTPUT HIGH ORDER BYTE AND PRESET '1'
023D F640 ORI 40H ;DISABLE PRESET
023F D303 OUT 03 ;OUTPUT 00000000 TO PORT 03
0241 C9 RET

```

END

NO PROGRAM ERRORS

APPENDIX B

BCD TRANSLATION SUBROUTINE

8080 MACRO ASSEMBLER, VER 2.3 ERRORS = 0 PAGE 1

```

; BCD TRANSLATION ROUTINE
0000 C00600 03NE: CALL BCDTD ;CALL BCD(4 DIGIT) TRANSLATION AND DISPLAY
U 0003 C30000 JMP TIMER
0006 F5 BCDTD: PUSH PSW
0007 C5 PUSH B
0008 D5 PUSH D ;SAVE STATUS AND CONTENTS OF REGISTER
0009 210010 LXI H, 1000H
000C 0E04 MVI C, 04H
000E 3600 LJOP8: MVI M, 00H ;CLEAR 4 BYTE MEMO. LOC. FOR REMAINDERS
0010 23 INX H
0011 00 DCR C
0012 C20E00 JNZ LJOP8
0015 210010 LXI H, 1000H ;INITIALIZE MEMO. LOC.
0018 7A MOV A,D
0019 07 RLC
001A 07 RLC
001B 07 RLC
001C 07 RLC
001D 16A0 MVI D, 10100000B ;LOAD DIVISOR BY 10
001F E6F0 LOOP9: ANI 11110000B
0021 4F MOV C,A
0022 7B MOV A,E
0023 07 RLC
0024 07 RLC
0025 07 RLC
0026 07 RLC
0027 5F MOV E,A
0028 E60F ANI 00001111B
002A 81 ORA C
002B 47 MOV B,A
002C 7B MOV A,E
002D E6F0 ANI 11110000B
002F 5F MOV E,A
0030 7B MOV A,B
0031 87 ORA A ;CHECK IF (A)=0,
0032 CA7100 JZ LOOP4 ;CONVERT LAST REMAINDER INTO BCD
0035 0E09 MVI C, 09H
0037 0A6800 LOOP7: JC LOOP5
003A 8A CMP D ;COMPARE DIVIDENT WITH DIVISOR
003B D26800 JNC LOOP5 ;IF (A)>(D) JUMP TO SUBTRACTION
003E 47 MOV B,A ;TEMPORARY STORE A INTO B
003F 3F CMC ;READY FOR ENTER '0'
0040 7B MOV A,E ;ROTATE REG. PAIR AGE TO LEFT BY 1 BIT
0041 17 RAL
0042 5F MOV E,A
0043 7B MOV A,B
0044 17 RAL
0045 00 OCR C
0046 C23700 JNZ LOOP7
0049 F5 PUSH PSW ;SAVE CONTENT OF A
004A 7D MOV A,L

```

8080 MACRO ASSEMBLER, VER 2.3 ERRORS = 1 PAGE 2

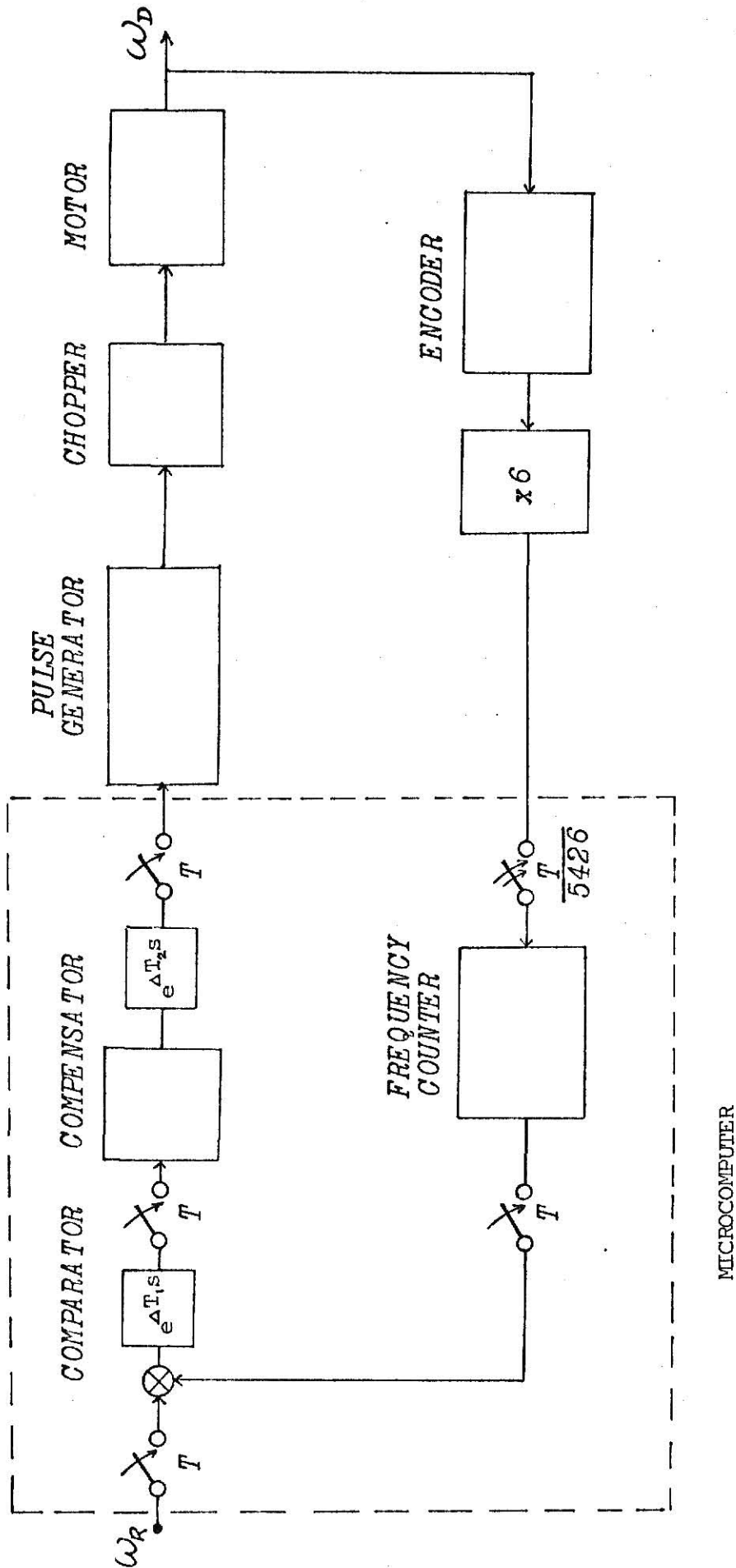
0048	1F		RAR		;SHIFT AO INTO CY
004C	0A5D00		JC	EVEN	
004F	F1		POP	PSW	
0050	17		RAL		
0051	17		RAL		
0052	17		RAL		
0053	17		RAL		
0054	47		MOV	B,A	
0055	E60F		ANI	00001111B	
0057	77		MOV	M,A	
0058	78		MOV	A,B	
0059	23	INMAD:	INX	H	
005A	C31F00		JMP	LJOP9	
005D	F1	EVEN:	POP	PSW	
005E	47		MOV	B,A	
005F	1F		RAR		
0060	E6F0		ANI	11110000B	
0062	77		MOV	M,A	
0063	78		MOV	A,B	
0064	17		RAL		
0065	17		RAL		
0066	17		RAL		
0067	17		RAL		
0068	C35900		JMP	INMAD	
0068	92	LJOP5:	SUB	D	
006C	47	LOOP6:	MOV	B,A	
006D	37		STC		;SET CY
006E	C34000		JMP	SHIFT	
0071	7D	LOOP4:	MOV	A,L	
0072	1F		RAR		;SHIFT AO INTO CY
0073	0A9400		JC	DEVEN	
0076	78		MOV	A,E	
0077	8A		CMP	D	
0078	02A100		JNC	SUBOU	
0078	0F		RRC		
007C	0F		RRC		
007D	0F		RRC		
007E	0F		RRC		
007F	77	BACK:	MOV	M,A	
0080	210010	DISPL:	LXI	H,1000H	
0083	7E		MOV	A,M	
0084	23		INX	H	
0085	86		ORA	M	
0086	2F		CMA		
0087	D302		OUT	02	
0089	23		INX	H	
008A	7E		MOV	A,M	
008B	23		INX	H	
008C	86		ORA	M	
008D	2F		CMA		
008E	D303		OUT	03	

8080 MACRO ASSEMBLER, VER 2.3 ERRORS = 1 PAGE 3

```

0090 01          POP      D
0091 C1          POP      B
0092 F1          POP      PSW
0093 C9          RET
0094 78          OEVEN:  MOV      A,E
0095 8A          CMP      D
0096 DA7F00      JC      BACK
0099 92          SUB      D
009A 77          MOV      M,A
0098 23          INX      H
009C 3601        MVI      M, 01H ;ENTER '1' INTO MOST SIGNIFICANT DIGIT
009E C38000      JMP      DISPL
00A1 92          SUB80U: SUB      D
00A2 37          STC
00A3 1F          RAR
00A4 1F          RAR
00A5 1F          RAR
00A6 1F          RAR
00A7 C37F00      JMP      BACK
                                ;READY FOR ENTER '1' TO NEXT NIBBLE
                                END
1 PROGRAM ERROR

```



BIBLIOGRAPHY

- [1] A.O. Williman and H.J. Jelinek, "Introduction to LSI Microprocessor Developments," IEEE, COMPUTER, vol. 9, n6, June 1976.
- [2] Floyd M. Gardner, "Phase Lock Techniques," John Wiley & Sons, Inc. New York, 1966.
- [3] "Intellec 8/MOD 80 Microcomputer System Operator's Manual" Intel Corporation.
- [4] "McMOS Handbook," 1st edition, Motorola Inc., 1974.
- [5] Alan B. Grebene, "Analog Integrated Circuit Design," VanNostrand Reinhold Co., 1972.
- [6] N.K. Sinha and N.H. Bailey, "Speed Control of A DC Servomotor Using Phase-Locked Loop: Some Test Results of a Practical Design," IEEE Trans. Ind. Electron. Contr. Instrum., vol. IECI-23, n1, pp. 22-26, Feb. 1976.
- [7] "8080 Microcomputer System User's Manual," Intel Corporation, Sept. 1975.
- [8] T.J. Maloney and F.L. Alvarado, "A Digital Method for DC Motors Speed Control," IEEE Trans. Ind. Electron. Contr. Instrum., vol. IECI-23, n1, pp. 44-46, Feb. 1976.
- [9] William D. Stanley, "Digital Signal Processing," Reston Publishing Company, Inc., 1975.
- [10] Patent 3,919,622 assigned to the Kansas State University.
- [11] G. Nash, "Phase-Locked Loop Design Fundamentals," Motorola Semiconductor Products Application Notes, AN 535, 1970.
- [12] "COS/MOS Digital Integrated Circuits," RCA Solid State Data Book Series, SSD-203A, Application Note ICAN-6101, 1973.
- [13] Donald E. Knoop and Mark C. Loessel, "Microcomputers: A Technique for Electronic Control," IEEE Trans. Ind. Applications, vol. IA-12, n4, pp. 354-358, July/Aug. 1976.
- [14] Julius T. Tou, "Digital and Sampled-Data Control Systems," McGraw-Hill Book Company, Inc., 1959.

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to Dr. Wellington W. Koepsel, my major professor, for his valuable assistance and advice in preparing this thesis.

I am thankful to Dr. Michael S. P. Lucas for providing the components and to the Department of Electrical Engineering for providing the facilities to make this project a reality.

Thanks are also due my friend, John L. Schmalzel (a graduate student in Electrical Engineering), who reviewed parts of the manuscript and suggested improvements.

I would also like to extend my thanks to Dr. Floyd W. Harris and Dr. Nasir Ahmed for serving on my committee.

A DIGITAL SPEED CONTROL
FOR CHOPPER-FED D.C. MOTOR
BY USING INTEL 8080 MICROCOMPUTER

by

ANDREW KOU-CHU LIN

Diploma, Taipei Institute of Technology, 1971

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1976

ABSTRACT

An approach using digital techniques to control the speed of a chopper-fed DC motor is discussed. This digital system consisting of random logic circuits and an Intel 8080 microcomputer offers overall advantages in performance, price, flexibility, reliability, and power requirements.

While keeping the speed constant, it is measured and compared with a reference setting (desired speed in binary form) and the error is then used to adjust the "ON-time" of an SCR chopper such that the motor speed is maintained at its desired value. For this application, an Intel 8080 microcomputer was used to implement the sampled data feedback control. The effective controller is a firing circuit unit with the changing delay between two pulse trains. The operation as well as control of this firing circuit unit which is superior to the conventional trigger circuit in many respects will also be emphasized.

In the prototype model, the range of speed control is from 200 rpm to 2500 rpm. This digital approach eliminates measurement nonlinearity and makes the speed setting reproducible. The prototype's steady-state error is less than 0.02% of maximum speed. The analysis of this discrete-time system is carried out in the Z-transform approach.