AN ABSTRACT OF THE THESIS OF

DONALD LEROY EVANS (Name) for the MASTER OF SCIENCE (Degree)

in Electrical and Electronics Engineering (Major) presented on April 26, 1972 (Date)

Title: DIRECT DIGITAL SPEED CONTROL SYSTEM FOR A DC MOTOR USING A MINICOMPUTER

Abstract approved: Redacted for Privacy

Louis N. Stone

Direct digital control is becoming quite common in the control of industrial processes. However, due to the relatively high accuracy of both analog and conventional digital systems for controlling the speed of a dc motor, there has been a reluctance to try DDC for this purpose. This thesis describes a direct digital control system for speed control of a dc motor, and shows the significant advantages it has over the more conventional methods.

The system consists of a dc tachometer, an A/D converter, a clock, a time shared PDP-8 minicomputer, interface logic, and a resistor switching circuit. Computer simulation is used to optimize various system parameters and check the feasibility of the control scheme. The final system is capable of regulating speed to an accuracy of 0.2% and can also control the acceleration of the motor. The control program only uses 2% of the core storage and 0.3% of the

computer's time, thus enabling the computer to be time-shared with many other control loops. Overall, this thesis shows the feasibility of a direct digital speed control system for a dc motor.

Direct Digital Speed Control System for a DC Motor Using a Minicomputer

by

Donald LeRoy Evans

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

June 1972

APPROVED:

Redacted for Privacy

Professor of Electrical and Electronics Engineering
in charge of major

Redacted for Privacy

Head of Department of Electrical and Electronics Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented April 26, 1972

Typed by Muriel Davis for Donald LeRoy Evans

ACKNOWLEDGMENT

I would like to express my sincere appreciation for the guidance of my major professor, Professor Louis N. Stone. Also I would like to thank Professor Donald A. Amort and Professor John F. Engle for their excellent technical guidance. And finally, I would like to give a special thanks to my wife, Louise, without whose constant encouragement and understanding this project could never have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

DIRECT DIGITAL SPEED CONTROL SYSTEM FOR A DC MOTOR USING A MINICOMPUTER

I. INTRODUCTION

Industry frequently requires highly accurate control of dc motor speed. When a dc motor is used as a positioning device, a fairly precise speed control is needed. However, when more than one motor is involved and it is necessary to keep their speeds at some exact ratio, the accuracy of speed becomes even more important. This type of accuracy can only be reached by using some form of closed-loop automatic control.

The typical speed control system consists of some measurement of speed from a tachometer being compared with a reference signal, and the resulting error is fed back to either an amplifier or an auxiliary generator. This error signal then varies the voltage on the armature of the motor. The type of feedback can be analog, digital, or hybrid. The earliest and most common method was analog control; however, the techniques used in analog systems have improved greatly over the years. Smith (19) presented a typical analog system using thyristor control of the generator in 1969. As a result of reference drifts and analog summing inaccuracies, speed regulation using analog feedback is limited to long-term accuracies of about ± .2% of top speed.

To improve upon this accuracy, digital techniques were combined with the analog techniques to give a hybrid system. One of the first articles about such a system was written in 1962 by Schmidt and Potts (17). Their system consisted of a digital measurement of speed compared with a digital reference, and the result was stored in a digital error register. This error was then converted to analog form and was used as the actuating signal for the amplifier or auxiliary generator. Similar systems were described by Burman, Blachford and Campling in 1967 (3) and by Millar in 1968 (14). Thompson and Wavre in 1967 carried this one step further to include acceleration control as well as speed control in their hybrid system (20).

The logical extension of digital techniques to control motor speed is to utilize the concepts of direct digital control, or simply DDC. In a DDC system many individual control loops time-share the use of a digital computer. The computer takes the place of the digital comparator and digital error registers in the above-mentioned systems. Also the resulting error can be fed into an equation to give the required correction needed, therefore leading to the use of more advanced control schemes for acceleration as well as speed control.

The purpose of this study is to analyze the requirements for direct ditigal control systems in industry today and to develop a direct digital control system for speed control of a dc motor.

II. DIRECT DIGITAL CONTROL

Conventional Control

Control of a system can be thought of as the exerting of an influence on the system by some intelligent manipulation of its inputs. The objective of control is to attain accurate, stable regulation of the system variables over a wide range of operating conditions. The method of determining how to vary the inputs is what determines the method of control. It could be manual control or automatic control. Automatic control can be broken down to conventional automatic control or computer control, which can be further broken down to the categories of optimizing computer control, supervisory computer control, and direct digital control.

Control not involving the use of a computer will be referred to as conventional control. Conventional control has followed a very definite evolution. The first method of control was manual control. An operator would manually adjust the system inputs to control the output. His only aids were his own experience and judgment. This led to the addition of indicating and recording instruments. With these aids, the operator was able to tell which inputs should be changed and by how much. In both of these methods, the operator was the element which closed the feedback loop.

From that point, control evolved into automatic control; first localized automatic control and then centralized automatic control. Automatic control uses regulators and instrument controllers operating in individual control loops to automatically carry out the control function. The control consists of measuring the output, determining the error, determining the necessary correction, and applying it to the final element. No operator is needed to close the loops as was the case with manual control.

Computer Control

Computer process control, or the utilization of a computer for closed-loop control of an overall process, was first used in 1958 and its first users were in the chemical and petroleum industries (9). Since that time, computer process control, in all of its forms, has become accepted and widespread in industry.

If a computer is used to calculate the optimum setting of the controlled variables, then it is referred to as optimizing control. Usually, a mathematical model of the process is programmed into the computer. For a specified system output, the computer will calculate the needed system inputs. Optimizing control can be used with the conventional instruments of automatic control or it can be combined with a supervisory control system or direct digital control system.

Supervisory control performs what is known as the supervisory functions. These include: scanning and alarming, data logging, and startup, shutdown, and emergency actions (12). These are not always controlling the process in a closed-loop manner, but they are an aid to simplify other control functions. As with optimizing control, supervisory control can be combined with any of the other types of control methods.

Considerations of Direct Digital Control

Since its first public mention in The Economist in 1961 (11), direct digital control (DDC) has become the most common form of closed-loop computer control. The concept of DDC came from the idea that a digital computer can rapidly attend to many individual control loops if it is used in a time-shared manner. That is, the computer takes the place on the controller in conventional automatic control. Each control loop now consists of a detecting transducer, final control element, and the time-shared digital computer. Figure 2-1 shows a typical DDC system. This type of system has several advantages over conventional control systems, including better control, flexibility of design, and lower cost.

The control advantages of a DDC system are rather clear cut. More precise control can be reached due to the accuracy of the control constants. These constants do not drift with time, as do

constants in analog systems; they are always calibrated; and they can be adjusted over a virtually limitless range. The correction term which is outputted to the process can be compensated to allow for adverse characteristics in the feedback loop. More advanced control algorithms can be implemented in the system, including "feed-forward" and "interacting" control, without the need for additional hardware. Therefore, a DDC system is capable of giving improved system performance over other types of control systems.

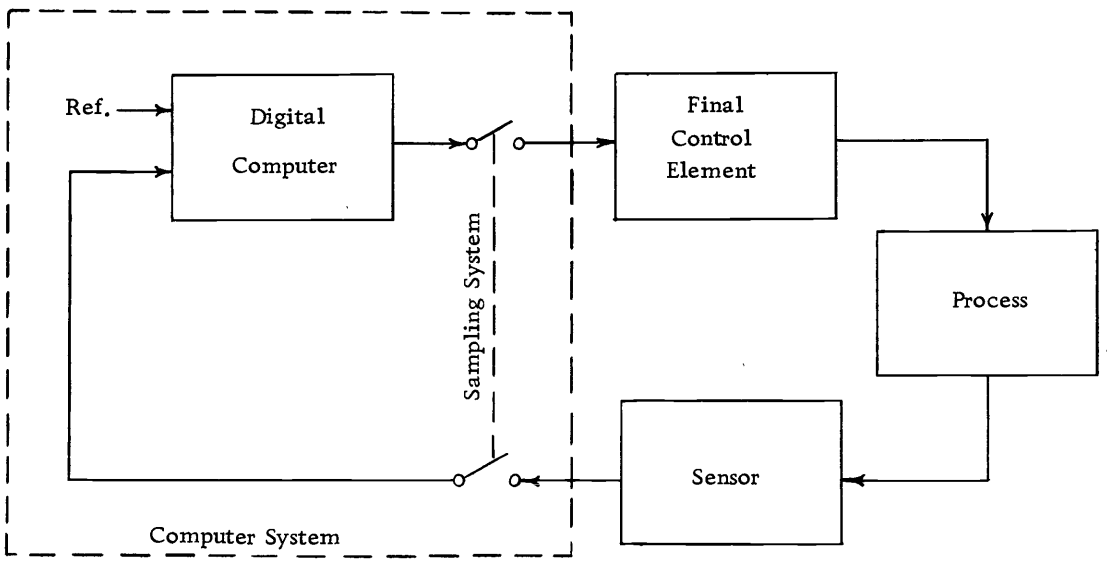


Figure 2-1. Typical DDC System

Flexibility is another key factor to the wide acceptance of DDC. Not only is a DDC system easier to design than a conventional automatic control system, but its design is not limited to standard available devices as with conventional systems. Also, the redesign of a

system is easier. As knowledge of actual process performance is gained, the control algorithm can be updated and improved by simple reprogramming of the computer. Such changes in a conventional system would result in expensive hardware modifications.

As is true with any system, a DDC system must be justified economically. This is one of the reasons for the rather slow acceptance of DDC. Conventional control loops are relatively inexpensive to implement. On the other hand, DDC can be fairly expensive for a system with only a few control loops, but additional loops can be added for little additional expense. There is, therefore, a break-even point where the two types of control cost the same. For systems larger than the break-even point, DDC can be economically justified. As the cost of control computers goes down, which is the current trend, it is becoming easier to justify DDC for process-control applications.

However, it still may be difficult to justify the total cost of a DDC system to do the same tasks as a conventional control system and nothing more. What also needs to be considered is the capability of the digital computer control system to perform many other tasks besides the basic control task.

It has already been mentioned that the computer has the capability of using more advanced control laws than a conventional controller. But this can be carried one step further by combining optimizing

control functions with the DDC system. Also, supervisory functions such as alarm printouts, clock-initiated data logs, and operator-demanded data logs can be combined with the DDC system. Variables not in the control loops can also be scanned and logged. Certain operating guides can be calculated, including efficiency and yield calculations, and can be printed out for operator convenience. Maintenance scheduling can also be handled by the computer, as well as a virtually limitless list of other tasks.

One should be cautioned against the combining of too many tasks in the system. Reliability and simplicity of organization might be better served by using another computer to handle some of the functions. A good example of this is to use a separate plant-optimization computer, which can continuously perform economic optimization without human intervention (22). This keeps the organization distinct while providing a backup computer which can immediately begin performing the DDC functions upon failure of the DDC computer. This also points out another use of the DDC computer: the preparation of data for communication to higher levels of computers.

The overall reliability of a DDC system must also be considered. In a conventional system, failure of some component only causes failure in that particular control loop. In a control system using a computer for set-point control, a computer failure leaves the plant still under analog control. However, in DDC a computer system

failure can cause complete loss of control for the entire plant.

The main concern is over the reliability of equipment which is common to all of the control loops. There seems to be no problem making equipment used in just one loop at least as reliable as analog controllers (2). However, the common equipment, including the central processor, needs to be much more reliable. Thanks to advancements in integrated circuits, computers can now be expected to have no more than one failure a year with a down time of only about four hours (9). Naturally, the larger the computer, the more components it has and the shorter the mean time between failures. To increase reliability, the computer should be kept in a controlled environment and should be isolated from large voltage spikes on the inputs. On-line diagnostics can prove to be very useful in providing for DDC system reliability.

Still the problem remains that the economic penalty of a computer failure may be too great. If so, a backup system will be needed. Backup systems range from an automatic switchover to manual control in the event of computer failure on up to the use of a separate backup computer which can automatically take over all the critical functions of the main DDC computer upon failure of that computer. Since it must be justified economically, the usual backup system consists of using automatic analog controllers on critical loops and placing all other loops under manual control, thus assuring 100% system reliability.

As well as system reliability, system security needs to be considered. A DDC system, being new and possibly confusing, is especially susceptible to manual interference. Careless knob-turning by unauthorized personnel or by inexperienced operators, can give the system more down time than all the component failures together. Systems, both hardware and software, should be designed to eliminate as much as possible of the accidental damage due to negligence.

System performance, economics, and reliability must be considered in all control system design. DDC is not always the best way. But today, more and more industries are making the choice to put their processes under DDC. DDC has proven its usefulness in the chemical industry, petroleum industry, steel industry, rolling mills, electric utilities, and many more. The possibility of using direct digital control should always be considered when a new control system is being designed.

III. SYSTEM DESIGN

System Requirements

The first step in the design of the DDC system to control the speed of a dc motor is to define the system requirements. Most importantly, the system must be able to accurately regulate the speed of the motor. Let us assume that the motor will be started manually and be brought to some base speed. The control system must take over at that point and bring the motor to the desired speed. Therefore, some form of acceleration control must also be provided by the system. An arbitrary decision will be made that the desired speed will be near or above the rated speed of the motor instead of at low speeds. The need for this decision will be shown later.

The cost of the system must not be prohibitive. The control program in the computer must take as little time and core memory as possible. It will be assumed that other control loops will utilize the rest of the computer's time and memory, thus allowing the cost of the computer to be shared by all of the control loops and economically justifying the system.

Method of Speed Control

The steady-state speed of a dc motor is governed by the basic speed equation. This equation is as follows:

$$\omega = \frac{V_A - I_A R_A}{K_g \phi} \text{ radian/sec.}$$

This suggests that the speed of a dc motor can be altered by varying either the armature voltage, the resistance in the armature, or the flux. Speed control of dc motors can therefore be broken down to the areas of armature-controlled motors and field-controlled motors.

In armature control, the field current is kept constant; and consequently, the magnetic field, ϕ, is also constant. This means that speed is directly proportional to $V_A - I_A R_A$ and that speed can be lowered from the basic speed by simply adding resistance to the armature.

In the field-control method of motor speed control, the armature current is kept constant. This results in the speed being inversely proportional to the flux, ϕ. Since the field current is directly proportional to the flux in the linear portion of the magnetization curve, speed becomes directly proportional to the resistance of the field circuit. Field control of a dc motor is a good method for speeds greater than the rated speed of the machine. High speeds can be reached by weakening the field (decreasing the field current), but care needs to be taken to insure that the motor does not become unstable and "run away." This method should not be used unless low torque presents little serious difficulty (6). Due to the difficulty and expense of obtaining a constant current source, this basic system is

often changed to use a constant voltage source instead of a constant current source for the armature.

Either method of speed control could be used in the DDC system. However, due to the fact that the field current is typically five to ten times smaller than the armature current, field control is more convenient to work with, and is therefore safer and more reliable. The final decision rests simply on whether one wishes to control high speeds or low speeds. Adding armature resistance lowers the speed below the basic speed, while adding resistance to the field raises the speed above the basic speed. The decision must be made knowing the use of the particular motor. An arbitrary decision is therefore made to control higher speeds and, as a result, the field-control method of speed control will be used in this DDC system.

System Components

One is now in a position to specify the various system components. The center of the DDC is, naturally, the computer. Large computers are capable of handling many control loops and long control programs. However, the cost of these computers is high and the reliability is not as good as with smaller computers. The "minicomputer" is especially well suited to DDC applications. It is large enough to handle most control jobs, yet the cost of the computer is quite low. One such computer is the PDP-8. This computer is fast,

1.6 μsec. cycle time, has 4K of core memory with a 12 bit word size and is well suited to use with many input/output devices. This is the computer that will be used in this control system.

In order to control the speed of the motor, the computer must have some means of actually monitoring the speed. This can be accomplished with either a digital tachometer or a dc tachometer. The digital tachometer, often called a "pulse tachometer," is a transducer which converts shaft speed into a series of pulses. The frequency of these pulses is directly proportional to the speed of the shaft. A logic circuit is then used to count and store the pulses (15). The other method is to use a dc tachometer which converts shaft speed into a dc voltage. This voltage is used as an input to an analog-to-digital (A/D) converter which changes the voltage into a binary representation.

Both methods of monitoring speed are susceptible to inaccuracies due to noise. However, the digital tachometer is inherently less accurate at high sampling rates. Assume that the digital tachometer gives 10 pulses per revolution. At a sampling rate of 10 hz., each pulse represents 60 r.p.m. (60 samples/min. ÷ 10 pulses/rev. = 60 rev./min./pulse). If a dc tachometer is adjusted such that 10 volts is equal to 2500 r.p.m. and this is fed into a 10 volt, 10 bit A/D converter, then each bit represents 2.44 r.p.m. (2500 r.p.m. / 2^{10} = 2.44). Also, the accuracy of the dc tachometer is not affected

by the sampling rate. Since the sampling rate necessary to control the motor is about 10 hz (shown later), a dc tachometer and A/D converter will be used to monitor the motor speed.

A measure of acceleration is also needed. This can most easily be done by dividing the difference in two consecutive measured speeds by the time between the measurements. Obviously, one needs to have a definite and known sampling rate. An electronic clock can be used to tell the computer when to initiate a time interval; or the computer can time itself by knowing the length of time needed to perform each instruction. The latter method can become quite complicated, especially if the program has any branches in it; that is, if it doesn't always follow the exact same sequence of instructions. So to ease the programming problem an electronic clock will be used. If one knew that a certain sampling rate would always be used, he could design a simple circuit to act as a clock. However, if one anticipates trying several different control schemes utilizing various sampling rates, he can visualize the problem of having to change the sampling rate in this simple clock. Therefore, at least to begin with, a clock with an adjustable sampling rate designed for laboratory instrumentation will be used.

The final necessary component of the system is the means by which the motor speed will actually be changed. As was shown before in field control of a dc motor, speed is increased by increasing the

resistance in the field of the motor. In fact, according to actual measurements taken, this is a fairly linear relationship, as illustrated in Figure 3-1 below. One way to have the computer change the amount of resistance in the field is to put several resistors in series with the field and have the computer short out a certain combination of these resistors. The maximum number of possible values of resistance for this resistor network is 2^n, where n is the number of

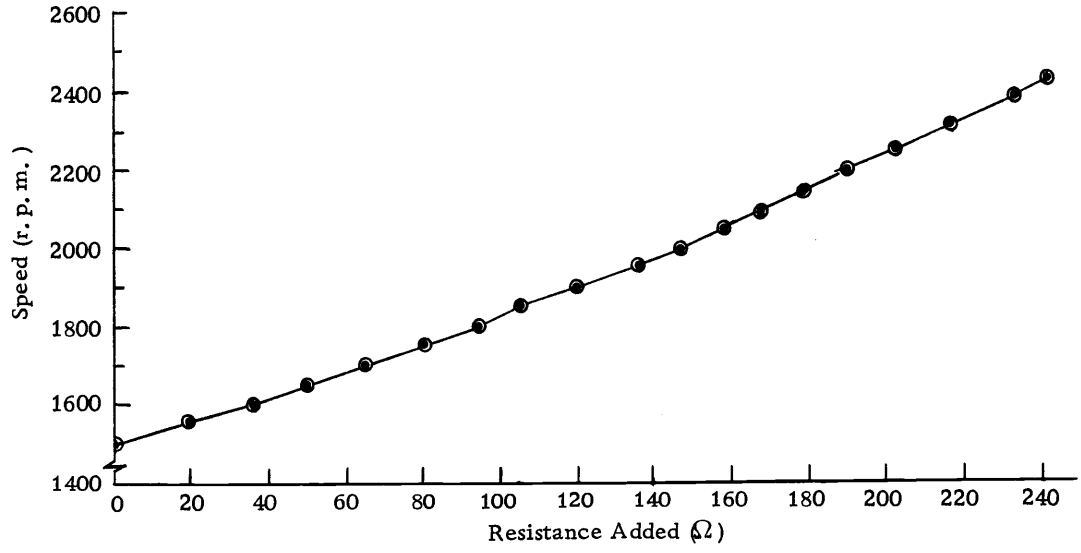


Figure 3-1. Speed Versus Resistance Added

resistors in series with the field. By choosing each resistor to be twice as large as the previous resistor, the possible values of total resistance are at evenly spaced intervals. That is, if a 2Ω, 4Ω, and 8Ω resistor are used, then there are 2^3, or 8 possible evenly

spaced values: 0Ω, 2Ω, 4Ω, 6Ω, 8Ω, 10Ω, 12Ω, and 14Ω. Due to the linear relationship illustrated in Figure 3-1, the evenly spaced steps of resistance enable the computer to control the motor to evenly spaced steps of speed. A block diagram of the system which has just been described is shown in Figure 3-2.

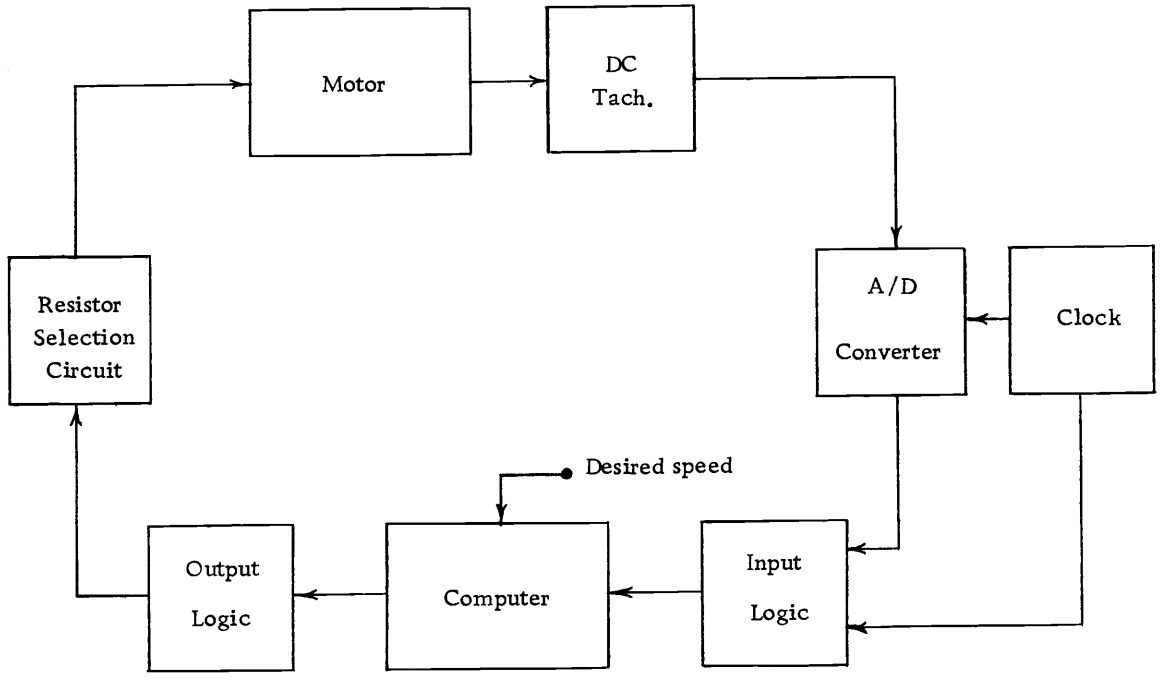


Figure 3-2. Direct Digital Speed Control System

System Software

The software for the system is just as important as the hardware. The progam must bring the motor up to a predetermined speed as rapidly as possible but with little overshoot. It has to keep the motor speed within certain limits around the desired speed and must

keep the speed from changing excessively in case of sudden change in load. The software is the most important factor in system performance.

The control algorithms for all DDC systems are similar. They take the system output, in this case speed, and compare it with the desired system output, thus getting an error quantity. Steps are then taken to reduce this error to zero, therefore controlling the system to the desired output. One possible implementation of this is: if the error shows that the motor speed is below the desired speed, then the resistance is increased by one step and the new error is found. Similarly, if the speed is too high, then the resistance is lowered by one step. However, since the motor does not respond instantly to a change in resistance, then the full effect of the change may not be completely realized for several time increments. The resulting response may tend to overshoot the desired speed excessively and may take a long time to narrow in on that speed.

In order to improve upon the response of the system, two factors need to be taken into account besides the algebraic sign of the error. These factors are the magnitude and the rate of change of error, or acceleration. One method which considers all three of these factors is the phase plane method. A phase plane is a plot of rate of change of error versus error. The basic concept of using a phase plane for speed control is that the acceleration should be such that when the

desired speed is reached ($\epsilon = 0$) then the acceleration is also zero ($\dot{\epsilon} = 0$). This allows the speed to remain at the desired value.

If the computer were able to predetermine the value of resistance which should be added to the field so that the desired speed would be reached at steady-state, then the system response would look somewhat like what is illustrated in Figure 3-3. By adding a dead-band

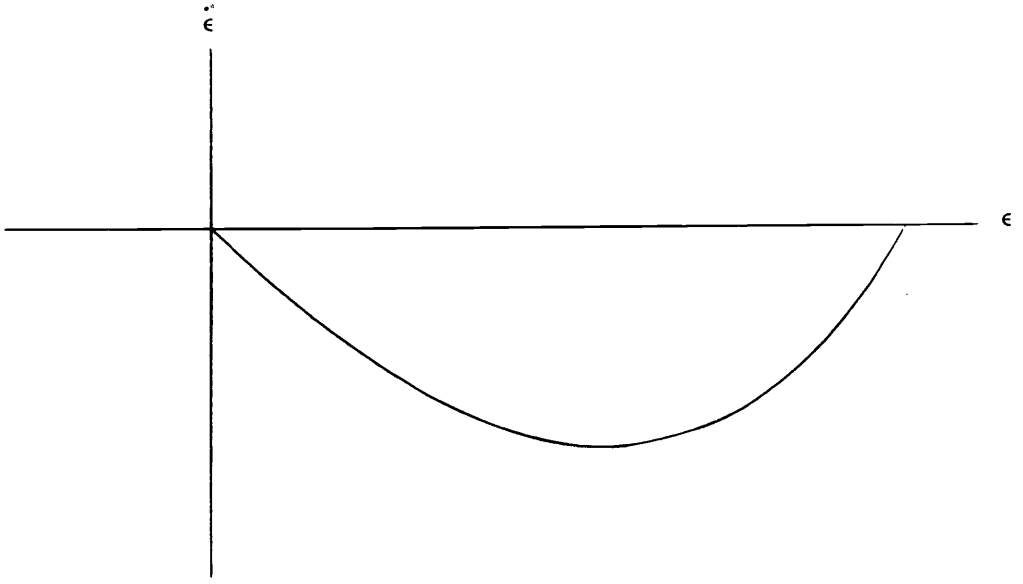


Figure 3-3. Response on a Phase Plane.

to the phase plane, as illustrated in Figure 3-4, no predetermination of resistance is required. The dead-band is a desired system response with which the system is made to coincide. This control scheme can easily be implemented by measuring the speed and calculating the acceleration. If these quantities show that the system response is to the right of the desired response, then the resistance is increased by one step. If the response is to the left, then the

resistance is decreased. And if the response is inside of the dead-band, then the resistance remains unchanged.

The slope of the dead-band is an important factor in the dynamic response of the system. If the slope is too steep, then the initial overshoot of the dead-band will cause the speed of the motor to overshoot the desired mark before the resistance can be lowered enough to bring the speed back inside of the dead-band. This could be dangerous if the desired speed is quite high and near the speed where the motor becomes unstable. On the other hand, if the slope is too flat, the motor will not get maximum acceleration and will not get to the desired speed in the least amount of time possible. Due to the

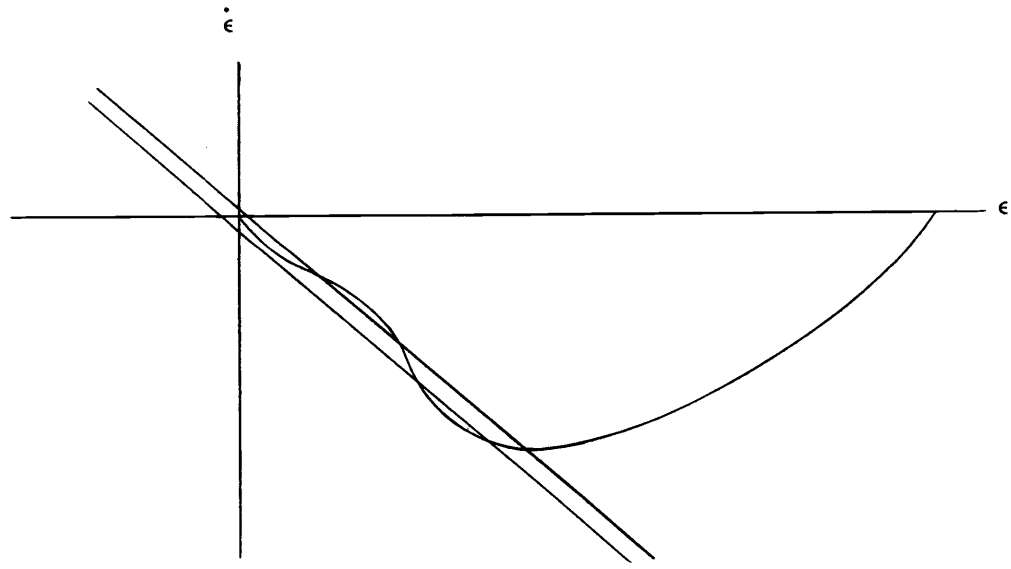


Figure 3-4. Typical System Response under DDC

difficulty in making performance calculations to determine the optimum slope, it was decided to develop a computer simulation of the system. This would enable the width of the dead-band and the sampling rate as well as the slope to be optimized. Also it would provide a feasibility check for the actual control scheme.

Simulation of System

The first step in the simulation is to develop a model of the field-controlled dc motor. If one can assume a constant current source for the armature, then the model is quite simple (7). However, the actual system has a constant armature voltage instead. One simplifying assumption made is that the inductance of the armature is negligible. The model of the motor along with the governing equations are shown in Appendix II. From this, a block diagram of all the transfer functions for the system can be developed, which is also shown in Appendix II. One will note that a dynamic model was used. This is because the output quantity of the system, speed, is always in a state of change, never settling down to any one value.

The values of the various parameters in the model need to be determined. Since this data is not already given for the motor, they had to be measured directly in the laboratory. This was done and the resulting measured and calculated values for these parameters are shown in Appendix II. Also included is another block diagram of the

motor, this one including the values for the motor's parameters in the transfer functions.

The computer used for the simulation was an EAI 680 digital computer and the program was written in HOI (Hybrid Operations Interpreter). To perform the actual simulation, the computer essentially controls the model the same way as the controlling computer would control the actual motor. As is illustrated in the flow chart in Appendix III, the model of the motor is used to calculate the output of the system at the end of each time increment. To solve the differential equations the simple Euler numerical integration routine was considered sufficiently accurate to not limit the overall accuracy of the simulation. Depending on what the output is, the resistance in the model was changed as was described earlier.

The first step in the utilization of the simulation was to assume values for the width of the dead-band and for the sampling rate. Then various values of slope were tried and the system response was observed. The quantity which was referred to as slope was the ratio of ϵ to $\dot{\epsilon}$, or actually the inverse of the true slope. The first values tried for this quantity were so large that the slightest acceleration exceeded the slope of the dead-band and the motor consequently accelerated very slowly. It was found that decreasing the magnitude of the inverse of the slope would result in the desired speed being reached earlier. However, at values below a certain point, the

desired speed was not reached much sooner and would be overshot, resulting in increased oscillation around the desired speed. Finally, a satisfactory compromise of overshoot and fast response was found.

Next the sampling rate was varied. Up to a certain point, the faster the sampling rate, the faster the machine would reach the desired speed. But, if the sampling rate was faster than the motor could respond to the previous change, then little additional acceleration was achieved and, again, the oscillation around the desired speed increased in magnitude.

Finally the width of the dead-band was varied. A small value was found to be important for maximum regulation. However, if the width were made extremely small, the control scheme seemed to have more difficulty in finding the desired resistance and consequently the oscillation increased.

All three quantities were again optimized, one at a time, using the optimum value already determined for the other quantities. The following conclusions were drawn from the simulation: 1) A value of -0.15 for the inverse of the slope provides for rapid acceleration with limited overshoot and oscillation; 2) A sampling rate of 10 hz. is sufficient to provide maximum acceleration; 3) A width of 4 r.p.m. is the optimal width of the dead-band; and 4) The control scheme used is a feasible method of providing field control of the dc motor.

It must be remembered that these conclusions are for the

model, not the actual system. For instance, the simulation showed that there was a slight oscillation around the desired speed. No attempt was made to change the control scheme to eliminate the oscillation because the magnitude of the oscillation, 5 r.p.m., was only 0.3% of the actual speed; and quite obviously, the simulation was not that accurate. Just because the simulation showed an oscillation, there was no guarantee that this oscillation would also be present in the real system. Similarly, the system could have an oscillation in it even if there was not one in the simulation.

One must realize the purpose and limitations of simulation. Simulation is a design tool, used to help in the design of a system. It has served this purpose. It has shown that -0.15 is a reasonable slope for the dead-band, that 10 hz. is a reasonable sampling rate, and that the control algorithm used is a feasible way of controlling the speed of a dc motor. But the simulation must not be used beyond the limits of accuracy of the model. This model is simulating a real process and therefore cannot be perfect. Actual system response can only be found accurately from the performance of the system itself.

IV. SYSTEM IMPLEMENTATION

Resistor Selection Circuit

The resistor selecting circuit is a circuit which takes a binary number from the computer output interface logic and shorts out the appropriate combination of resistors in series with the field of the motor. As was mentioned in Chapter III, by properly choosing the values for the resistors, equal steps in resistance can be realized. It is arbitrarily decided to control the motor up to 2200 r.p.m. (no load); and according to Figure 3-3, this requires an addition of about 200 Ω to the field resistance. If only five resistors are used, then the appropriate values are: 100Ω, 50 Ω, 25Ω, 12.5Ω, and 6.25Ω. This enables one to reach values of resistance from zero to 193.75Ω by steps of 6.25Ω. The schematic for the resistor network and the selection circuit is shown in Figure 4-1.

The first part of the circuit which must be decided upon is the means by which the resistors will be shorted. This can be accomplished by the use of either electromechanical or solid-state devices. Electromechanical devices (relays) have a relatively low initial cost but do not have the long-term reliability found in solid-state devices. Also the relays cannot be actuated from a logic level and would therefore require an amplifier or driver circuit of some type. The

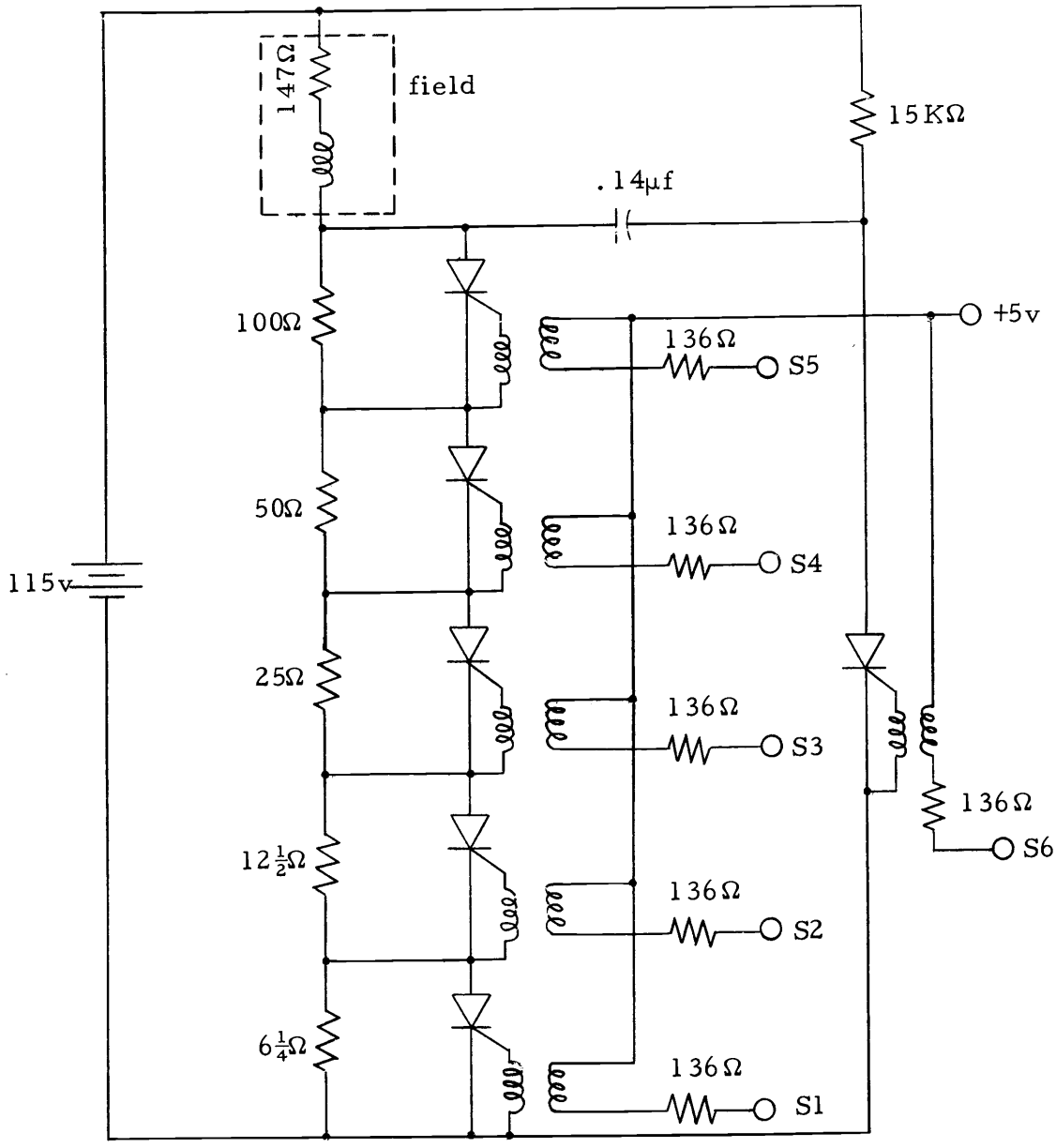


Figure 4-1. Resistor Selection Circuit

most likely solid-state device to short out the resistors is a silicon-controlled rectifier (SCR). However, SCR's in a dc system have the handicap of not being easily turned off. But due to the reliability of the SCR's and the ease of interfacing them with logic levels, they will be used to short out the resistors.

SCR's are relatively easy to turn on; all they require is a momentary current to flow between the gate and the cathode. This can easily be done from a logic level. However, due to the high voltages on the SCR, the logic must be isolated from it. The best way to do this is with a transformer. But transformers are expensive. One inexpensive device which can be used is a pulse transformer, which is a transformer that gives a pulse output on its secondary when a dc level is applied to its primary. This pulse has long enough duration to turn on the SCR, and therefore permits the logic to actuate the SCR yet be electrically isolated from it. By connecting one side of the primary to five volts, all the logic has to do is ground the other terminal and the SCR will short out the resistor.

The only way to turn off an SCR is to bring its current below the "holding current," the current below which the SCR will stop conducting. One easy way of doing this is to short the anode and cathode together. This can be done by turning on another SCR which is in parallel with the others. To turn this shorting SCR off, a commutating circuit is used. A capacitor is charged high enough so that

it will sustain current flow through the shorting SCR long enough to turn off the other SCR's. But when the charge drops off on the capacitor, there is not enough current flow to keep the shorting SCR turned on. One can then turn on the next desired combination of SCR's.

The SCR's used are Motorola MCR 2305-4. These are 200 volts SCR's with a turn-on time of 1 μsec., turn-off time of 16 μsec., and a holding current of 10 ma. From this, one can calculate the needed values for the capacitor and resistor in the commutating circuit.

$$R = \frac{115v}{10ma} = 11.5\ K\Omega \quad (15K\Omega \text{ will be used})$$

$$C = \frac{T}{R} = \frac{17\ \mu sec}{147\Omega} = .116\mu f\ (.14\mu f \text{ will be used})$$

Computer Interface Logic

The PDP-8 used in the DDC system already has an interface unit attached to it. The diagram of the interface card is shown in Appendix IV. The basic purpose of this interface unit is to make various I/O functions easily available externally. Otherwise, the user would have to make all I/O connections inside of the computer, requiring a knowledge of the circuits in the computer. The interface also serves as a buffer between the computer and the external logic. This lessens the chance that the components inside the computer will be damaged by some external voltage spike or short circuit. Since

the interface is made exclusively of Texas Instruments Series 7400 TTL logic, this makes the computer compatible with all external TTL logic.

The outputs available from the computer interface are: the contents of the accumulator, I/O pulses and device codes. The device codes enable selection of one I/O device and the I/O pulse causes this device to perform some particular function. In this system, device code BMB4 will refer to the input device and BMB3 will be used for the output device. All outputs from the interface unit are positive logic.

Input to the accumulator, the clear accumulator instruction, and the skip instruction are the possible inputs to the interface unit. Putting a number on the input to the accumulator terminals cause that number to be entered into the accumulator. The inputs to the computer interface need to be in negative logic. Also, only gates with open-collector outputs should be used to input to the interface unit.

The basic function of the clock input logic is to cause the computer to jump out of the clock-monitoring loop when the output of the clock changes from a logical zero to a logical one. This is accomplished by three NAND gates (see Appendix IV). Actually, one three-input open-collector NAND gate would be the ideal choice, but no such device is available in TTL. Only two-input open-collector NAND

gates can be acquired. One of the NAND gates serves simply as an inverter, but this is the only inverter needed in the logic (except for one inverter-driver), so the purchase of one chip with six inverters on it is not practical.

The A/D converter input logic gates the 10 bits of the converter into the accumulator upon receipt of the proper device code and I/O pulse. This logic is made up of 12 NAND gates. Once again, this is not the minimum number needed, but one has to use only two-input open-collector gates. A 4 input NAND driver is used for the inverter-driver needed to fan-out to 10 gates.

The requirements of the output logic is that upon receipt of the correct I/O pulse and device code, the contents of the lower six bits of the accumulator will turn on the proper combination of the six SCR's. As was discussed before, this involves putting a logical zero on the inputs to the pulse transformers corresponding to the SCR's which should be on. The logical way of doing this is to use six latches which will change their state to correspond to the accumulator whenever the proper device code and I/O pulse is given. However, latches cannot be used to drive external loads, so their output must be fed into a driver which will be used to put the dc level on the pulse transformer. As one can see, many gates are required to perform this logic. It was experimentally determined that the length of the I/O pulse, 400 nsec., was long enough to turn on the SCR's; therefore,

making the requirement of maintaining a dc level unnecessary. The entire output logic can therefore be realized by using six four-input NAND drivers (three-input would have been sufficient). This, along with the other logic diagrams, is shown in Appendix IV.

Control Program

The control program needs to carry out the necessary control scheme in an efficient manner; the amount of time and core storage the control program requires should be kept to a minimum. One way to do this is to write the program in assembly language instead of FORTRAN. In fact, the assembly language program will be translated into machine code and entered manually from the computer console.

A flow chart for the program is shown in Appendix V. One can see from this flow chart that basically the program does the following: waits in a loop for the clock pulse, delays 100 μ sec. (A/D conversion time), reads the speed, finds if the motor is still being accelerated manually, calculates to see if R needs to be changed, turns off the SCR's, and finally turns on the next combination of SCR's. The only part of this which would be time consuming would be the calculation to determine if the speed and acceleration are within the dead-band. This is because both multiplication and division are needed and the PDP-8 does not have machine instructions to do these calculations.

An interactive subroutine would have to be used. So to make the program consume as little time as possible, the slope of the dead-band and the sampling rate are picked such that the ratio, "SLOPE/DT" is some power of 2, in this case -2. Therefore, a slope of -.16 is used with a sampling rate of 12.5 hz. (DT = .08 sec.).

The program itself (see Appendix V) only takes up 78 words of core. This is less than 2% of the available core. Also the maximum length of time from the time a cycle begins until the computer starts waiting for another clock pulse is about 233 μsec., only 0.15% of the time between clock pulses. Thus, 98% of the computer's core and 99.85% of its time is left over to control other processes. Also, 100 μsec. of the program is simply a delay loop, so this time could also be used for other purposes. The program has therefore succeeded in the requirement of making the most efficient use of the computer time and core storage.

V. RESULTS AND CONCLUSIONS

Results

The performance evaluation of the DDC system should be based on two factors. These are: the ability to rapidly bring the motor from the base speed to the desired speed and the ability to accurately regulate the speed of the motor. A plot of the system response is the best way of evaluating the first factor. As can be seen in Figure 5-1, the control system was able to bring the motor up to the desired speed of 2000 r.p.m. from a base speed of 1500 r.p.m. in 5.1 seconds. But a more important observation was made--there was no observable overshoot of the desired speed.

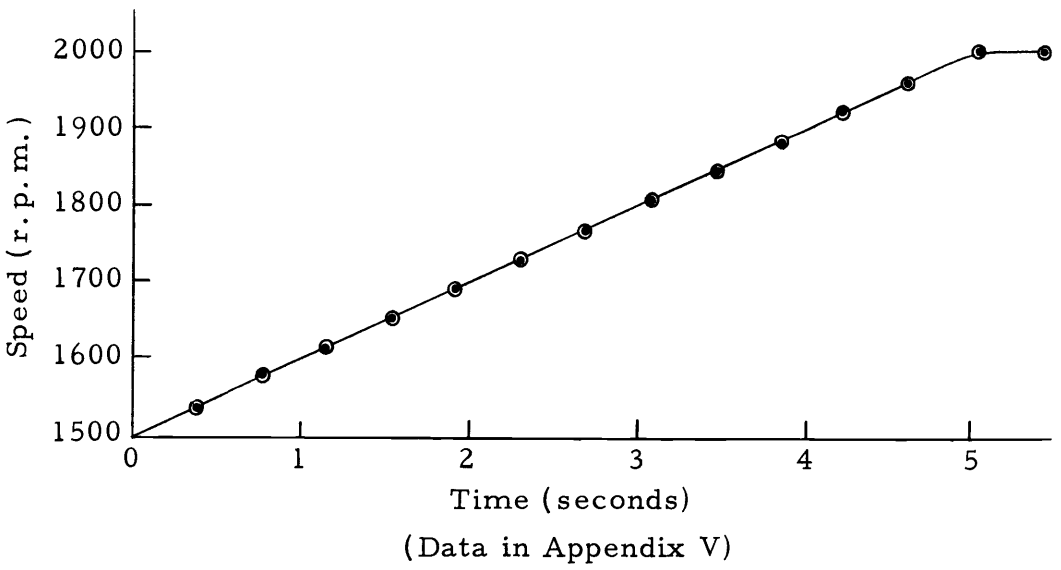


Figure 5-1. System Response

It was quite difficult to get some measurement of the accuracy of the regulation. By use of an oscilloscope measuring the output of the dc tachometer, one could verify that there was no oscillation of speed at a magnitude of .5% or more, but that there was occasional noise at this level. This would result in the computer detecting other than the actual speed and try to change the speed. But on the next time increment, it would again find the speed to be satisfactory. The only other way of approaching the regulation accuracy is to periodically monitor what the computer sees the speed to be. This was done by periodically storing in memory several consecutive speed readings. The result of this was that the speed never varied more than $\pm$ 1 bit ($\pm$2.44 r.p.m.) from the desired speed except for an occasional erroneous reading which most likely was caused by noise on the wires connecting the tachometer to the A/D converter. Also there appeared to be no specific length of time between the slight variations in speed, therefore indicating that the motor speed was not oscillating. The accuracy of regulation of the DDC system can therefore be considered to better than 0.2%

Conclusions

A direct digital speed control system for a dc motor has been described in this study. This system is capable of regulating the speed to an accuracy of 0.2%, which is approximately the same

accuracy which can be achieved by use of an analog control system or a conventional digital control system. One may ask why he should bother with a DDC system when it does not improve accuracy any over other methods. The answer lies in cost, simplicity of implementation, and potential system capabilities.

Many processes require the use of direct digital control to reach the desired system performance. It would not be uncommon for these processes not to be fully utilizing all of the computer. Therefore, one could easily put a dc motor under DDC along with the rest of the control loops in the process. This results in the computer itself being economically justified. The cost of the rest of the system is small compared to comparable analog or conventional digital control system.

The system performance is potentially better than has been shown. If one were to use a 12 bit A/D converter, the speed monitoring would be four times more accurate, bringing up the possibility of regulation with an accuracy of .05%. To accomplish this, steps would have to be taken to eliminate all noise in the tachometer circuit. Also, it would be quite likely that the control scheme would have to be improved. One possible improvement would be to have the computer constantly changing between two resistor values instead of just setting one. The length of time spent on each value would determine the equivalent resistance the field circuit would see.

All that would be needed to accomplish this is a change in the control program. He should weigh the cost, flexibility and ease of implementation, control performance, and reliability. Such considerations are leading industry to use DDC more and more. Therefore, the future of direct digital control cannot be anything but good.

BIBLIOGRAPHY

1. Audeh, A. S. and A. N. Deodhar. Speed control: four methods compared. Control Engineering 17:53-57.

2. Bernard, John W. and Joseph F. Cashen. Direct digital control. Instruments and Control Systems 38:151-158. Sept. 1965.

3. Burman, H. W., C. W. Blachford and C. H. R. Campling. Digital controller for a closed-loop speed control system. Engineering Journal 50:26-30. April 1967.

4. Cundall, C. M. Backup methods for DDC. Control Engineering 12:98-103. April 1965.

5. Donohoe, Dennis E. Electromechanical versus solid-state control systems in packaging and material handling: case studies. IEEE Transactions on Industry and General Applications IGA-3:551-553. 1967.

6. Gertler, Janos. Position versus velocity algorithms in d. d. c. Instrument Practice 22:1014-1015. 1968.

7. Gourishankar, Vembu. Electromechanical energy conversion. Scranton, Pennsylvania, International Textbook Company, 1965. 564 p.

8. Harper, S. D. Next steps in computer control. Chemical Engineering 72:171-176. June 7, 1965.

9. Hix, A. H. Status of process control computers in the chemical industry. Proceedings of the IEEE 58:4-10. Jan. 1970.

10. Hornor, Joseph F. Direct digital control. In: Computer control of industrial processes, ed. by Emanuel S. Savas. New York, McGraw-Hill, 1965. p. 287-302.

11. Instruments in evolution. The Economist 198:1219-1220. 1961.

12. Laspe, Charles. A look at computer control in the process industries. IEEE Transactions on Industry and General Applications IGA-3:11-14. 1967.

13. Lee, T. H., G. E. Adams and W. M. Gaines. Computer process control: modeling and optimization. New York, John Wiley and Sons, 1968. 386 p.

14. Millar, R. A. Digital control of shaft speed and position. IEEE Spectrum 5:90-95. Jan. 1968.

15. Ponstingl, John C. Digital tachometer generator systems. IEEE Transactions on Industry and General Applications IGA-2:65-76. 1966.

16. Reed, Richard E. Guidelines for DNC. Automation 18:61-64. May 1971.

17. Schmidt, W. C. and R. R. Potts. Speed regulation by digital methods. Transactions of the AIEE, Part II, 81:221-5. 1962.

18. Scott, M. J. Pneumatic controllers can backup d. d. c. Instrument Practice 22:1011-1013. 1968.

19. Smith, J. W. Automatic closed loop variable speed drives. Electrical Review 85:679-683. 1969.

20. Thompson, Francis T. and Andre Wavre. Speed, acceleration, and load control solid-state system. IEEE Transactions on Industry and General Applications IGA-3:538-542. 1967.

21. Ware, William E. Direct digital control. Instruments and Control Systems 38:79-83. June 1965.

22. Williams, Theodore J. What to expect from direct-digital control. Chemical Engineering 71:97-104. March 2, 1964.

APPENDICES

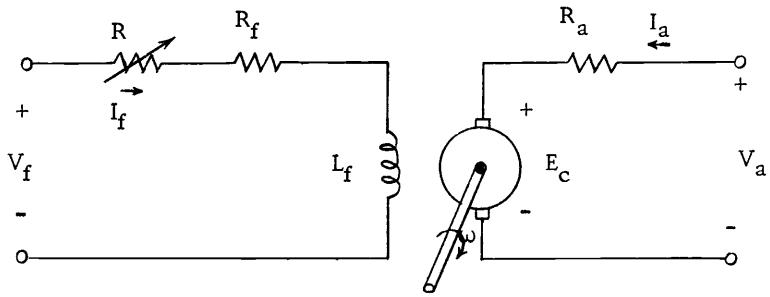
APPENDIX I

Equipment

Clock	General Radio, Model 1340 Pulse Generator OSU #149032
Computer	Digital Equipment Corporation, PDP8/L OSU #158216F
Tachometer	Reeves Instrument Corporation, DC tachometer. 2 poles, 6.5 volts
A/D Converter	Zeltex Inc., model ZD461 μverter
Power Supply	Power/Mate Corporation (for A/D converter) Regulated Power Supply, Model BP 118 OSU #156569
DC Motor	General Electric Model 5B225B478, 115 volt, 1800, $1\frac{1}{2}$ KW, 13 Amp, OSU #51940
DC Generator	General Electric Model 5B224B541 125 volt, 1800 r.p.m., $1\frac{1}{2}$ KW, 12 Amp, OSU #51940

APPENDIX II

Model of Motor



Governing Equations:

$$V_f = (R_f + R) I_f + L_f \frac{dI_f}{dt}$$

$$V_a = E_c + I_a R_a$$

$$E_c = K_a \phi \omega$$

$$T = k_a \phi I_a$$

$$T = J \frac{d\omega}{dt} + f\omega + T_L$$

Values for Constants:

Motor

$V_f = 115v$

$R_f = 147\Omega$

$L_f = 27.9H$

$R_a = 2.06\Omega$

$V_a = 115v$

$J = f\omega/2.1$

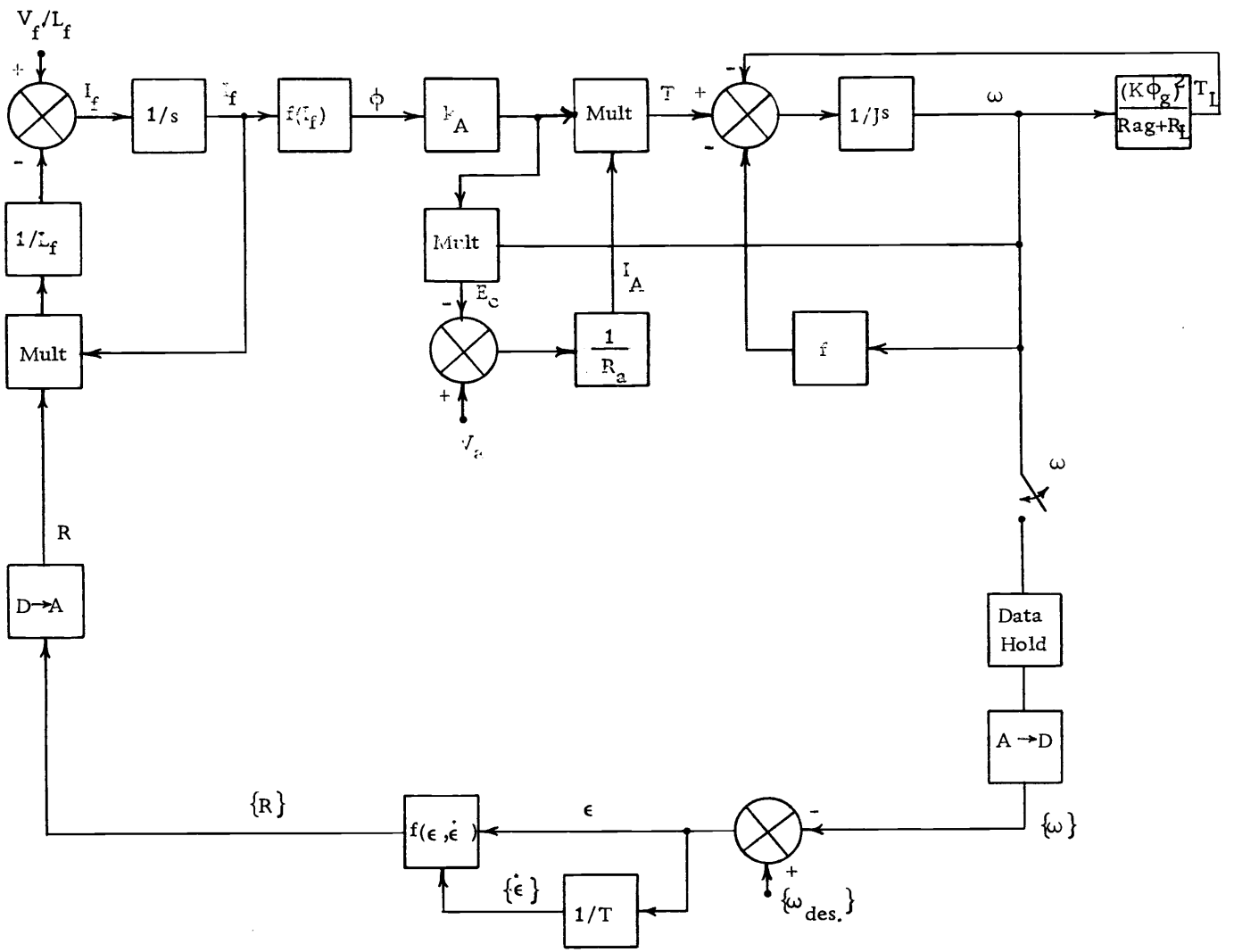
$f = 1/(9.45\omega + 10^4)$

Gen.

$K_a \phi = 7.72 \times 10^{-2}$

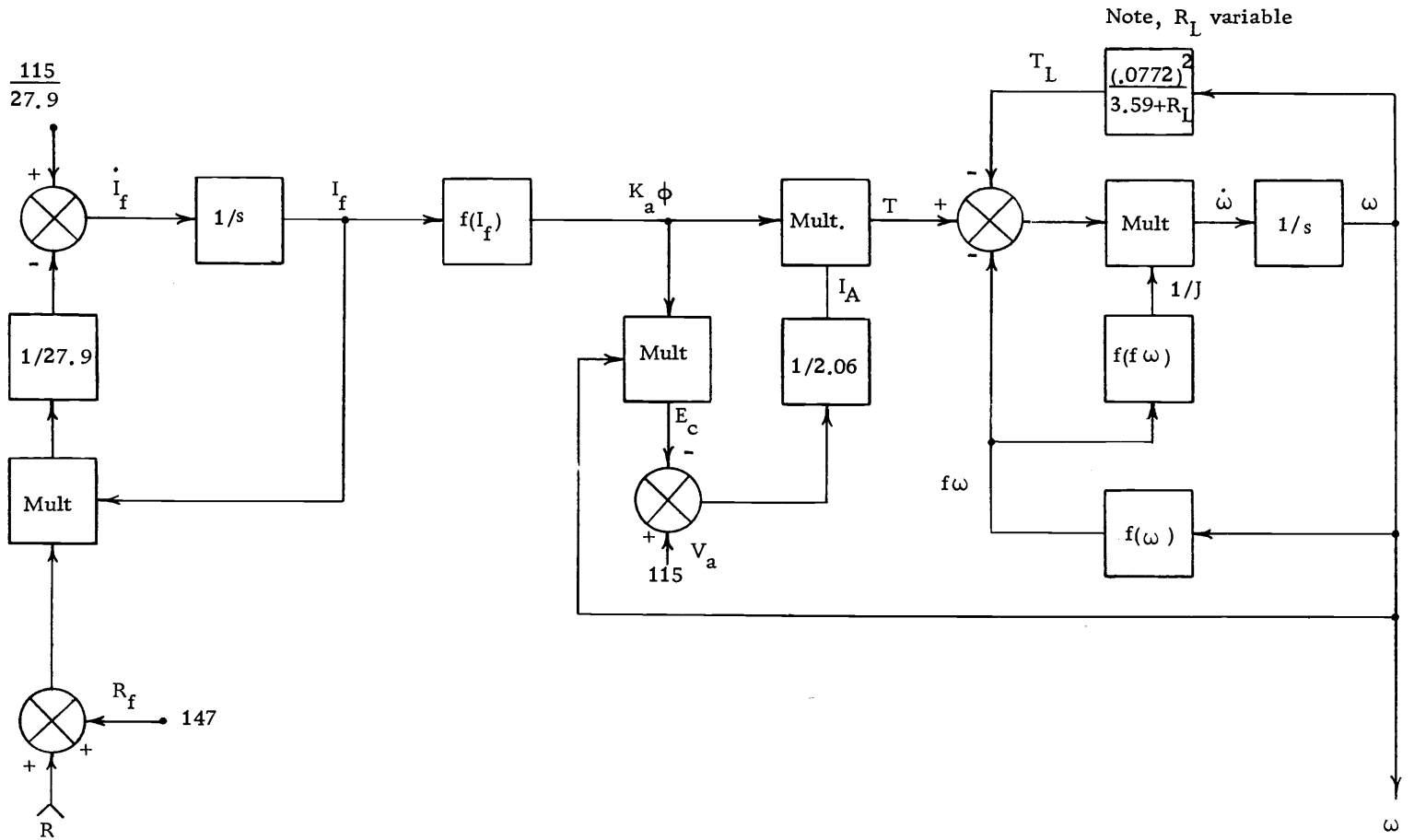
$R_a = 3.59\Omega$

If	$K_a\phi$
.28A	4.35×10^{-2}
.32	4.77×10^{-2}
.36	5.20×10^{-2}
.40	5.63×10^{-2}
.44	5.94×10^{-2}
.48	6.20×10^{-2}
.52	6.43×10^{-2}
.56	6.62×10^{-2}
.60	6.80×10^{-2}
.64	6.97×10^{-2}
.68	7.11×10^{-2}
.72	7.24×10^{-2}
.76	7.34×10^{-2}
.80	7.44×10^{-2}



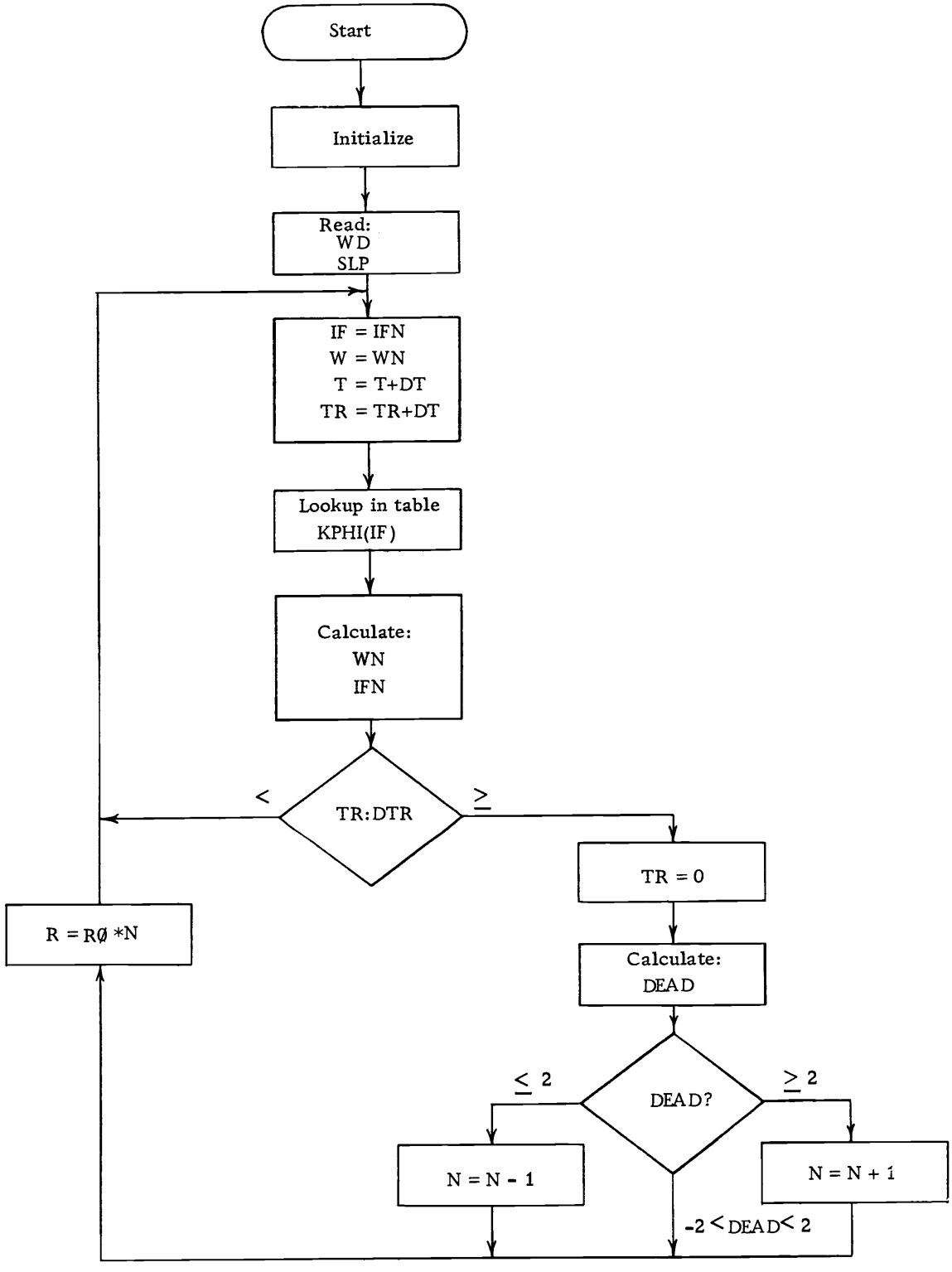
Block Diagram of Entire System

Block Diagram of Motor with Values



APPENDIX III

Simulation Flow Chart



Simulation Program

```
1.001   @L,0;
1.010   VAG=115
1.020   RAG=3.59
1.030   KPHIG=.0772
1.040   VF=115
1.050   RF=147
1.060   LF=27.9
1.070   RA=2.06
1.080   VA=115
1.090   IFN=VF/RF
1.091   WN=1400
1.092   DT=.02
1.093   DTR=.1
1.094   RO=3.125
1.095   RL=0
1.100   TBL(1)=.0435
1.110   TBL(2)=.0477
1.120   TBL(3)=.052
1.130   TBL(4)=.0563
1.140   TBL(5)=.0594
1.150   TBL(6)=.062
1.160   TBL(7)=.0643
1.170   TBL(8)=.0662
1.180   TBL(9)=.068
1.190   TBL(10)=.0697
1.200   TBL(11)=.0711
1.210   TBL(12)=.0724
1.220   TBL(13)=.0734
1.230   TBL(14)=.0744
1.300   N=0
1.310   T=0
1.320   TR=0
1.325   R=0
1.330   WD, SLP ←
1.340   E=WD-WN
1.500   IF=IFN
1.510   W=WN
1.520   T=T+DT
1.530   TR=TR+DT
1.540   2;
1.550   TR<DTR-.0001? 1.50.
1.560   TR=0
1.570   3;
1.580   DEAD<=-2 N=N-1
1.590   DEAD>=2? N=N+1
1.600   R=RO*N
1.610   1.50.
2.010   I=0
2.020   I=I+1
2.030   IF>.24+I*.04? 2.02.
2.040   KPHI=TBL(I-1)+(25*IF-5-1)*(TBL(I)-TBL(I-1))
2.050   EC=KPHI*W
2.060   IA=(VA-EC)/RA
2.070   TQ=KPHI*IA
2.080   TL=KPHIG*KPHIG*W/(RAG+RL)
2.090   F=1/(9.45*W+10000)
2.100   J=F*W/125
2.110   WDOT=(TQ-F*W-TL)/J
2.120   IFDOT=(VF-(RF+R)*IF/LF
2.130   WN=W+WDOT*DT
3.010   EO=E
3.020   E=WD-WN
3.030   DE=E-EO
3.040   EDOT=DE/DTR
3.050   DEAD=E-SLP*EDOT
3.060   E, EDOT, DEAD:
3.070   T,R,IFN,WN:
```

Simulation Results

D = 1600 r.p.m. slp = -.15 DT = .1 sec. $|\epsilon| < 2$

T	R	IFN	N	E	EDOT	DEAD
.10	.000	.7823	1219.6	380.35	-196.4	350.8
.20	3.125	.7753	1229.6	370.39	- 99.5	355.4
.30	6.250	.7646	1236.2	363.77	- 66.1	353.8
.40	9.375	.7518	1241.9	358.07	- 56.9	349.5
.50	12.500	.7381	1247.5	352.49	- 55.8	344.1
.60	15.625	.7241	1253.1	346.81	- 56.7	338.3
.70	18.750	.7102	1259.2	340.78	- 60.2	331.7
.80	21.875	.6966	1265.9	334.01	- 67.7	323.8
.90	25.000	.6834	1273.0	326.91	- 71.0	316.2
1.00	28.125	.6706	1280.3	319.62	- 72.8	308.6
1.10	31.250	.6583	1287.8	312.16	- 74.6	300.9
1.20	34.375	.6464	1295.3	304.68	- 74.7	293.4
1.30	37.500	.6349	1302.8	297.20	- 74.8	285.9
1.40	40.625	.6239	1310.7	289.25	- 79.4	277.3
1.50	43.750	.6132	1318.9	281.07	- 81.8	268.8
1.60	46.875	.6029	1327.1	272.86	- 82.1	260.5
1.70	50.000	.5929	1335.3	264.64	- 82.1	252.3
1.80	53.125	.5833	1343.6	256.37	- 82.7	243.9
1.90	56.250	.5740	1351.8	248.14	- 82.3	235.8
2.00	59.375	.5650	1359.9	240.01	- 81.3	227.8
2.10	62.500	.5563	1368.0	231.98	- 80.3	219.9
2.20	65.625	.5478	1376.0	223.93	- 80.4	211.8
2.30	68.750	.5396	1384.0	215.92	- 80.0	203.9
2.40	71.875	.5317	1392.0	208.00	- 79.2	196.1
2.50	75.000	.5239	1399.8	200.18	- 78.1	188.4
2.60	78.125	.5165	1407.6	192.40	- 77.8	180.7
2.70	81.250	.5092	1415.7	184.21	- 81.8	171.9

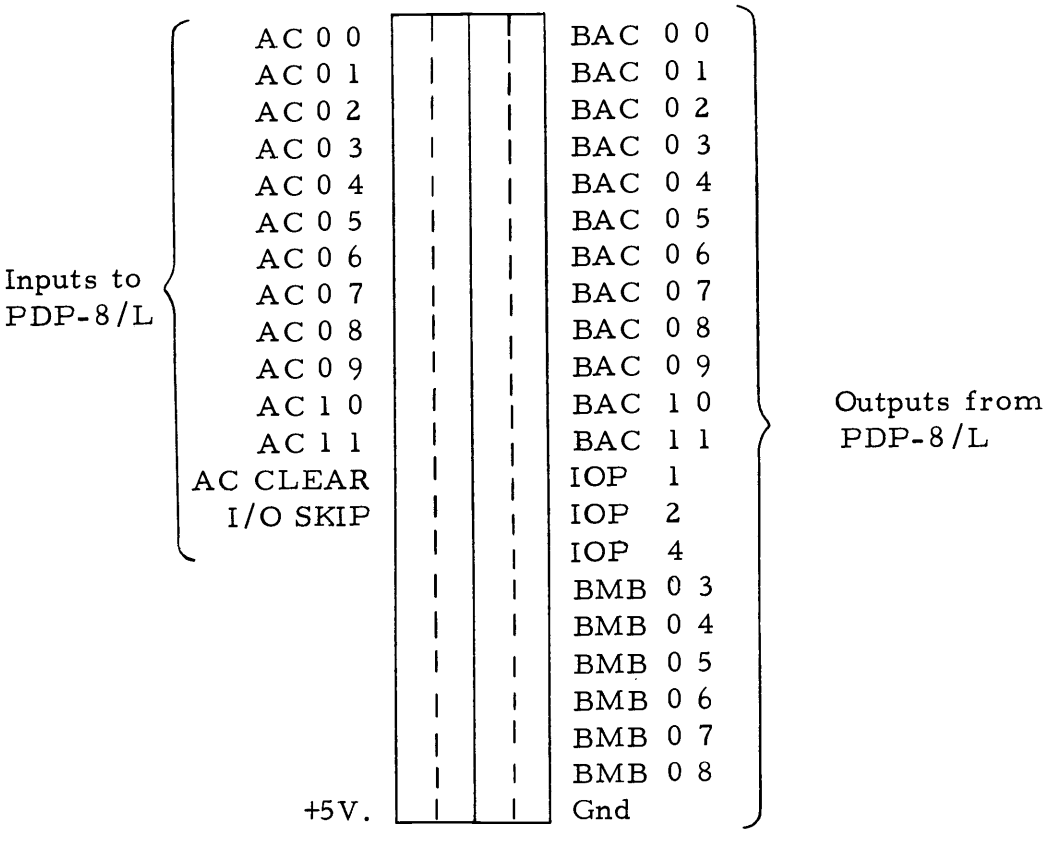
T	R	IFN	N	E	EDOT	DEAD
2.80	84.375	.5021	1424.2	175.78	- 84.3	163.1
2.90	87.500	.4952	1432.7	167.26	- 85.1	154.4
3.00	90.625	.4886	1441.2	158.75	- 85.0	145.9
3.10	93.750	.4821	1449.6	150.30	- 84.5	137.6
3.20	96.875	.4757	1458.1	141.82	- 84.8	129.1
3.30	100.000	.4696	1466.9	133.09	- 87.2	120.0
3.40	103.125	.4636	1475.7	124.25	- 88.4	110.9
3.50	106.250	.4577	1484.6	115.39	- 88.6	102.0
3.60	109.375	.4520	1493.4	106.55	- 88.3	93.3
3.70	112.500	.4465	1502.2	97.78	- 87.7	84.6
3.80	115.625	.4410	1510.9	89.09	- 86.8	76.0
3.90	118.750	.4358	1519.7	80.27	- 88.1	67.0
4.00	121.875	.4306	1528.9	71.08	- 91.9	57.2
4.10	125.000	.4256	1538.3	61.67	- 94.0	47.5
4.20	128.125	.4206	1547.8	52.16	- 95.0	37.9
4.30	131.250	.4158	1557.3	42.64	- 95.2	28.3
4.40	134.275	.4111	1566.8	33.13	- 95.0	18.8
4.50	137.500	.4065	1576.3	23.66	- 94.6	9.4
4.60	140.625	.4021	1585.7	14.26	- 93.9	0.1
4.70	140.625	.4005	1593.9	6.04	- 82.2	- 6.2
4.80	137.500	.4029	1599.0	.95	- 50.9	- 6.6
4.90	134.375	.4067	1600.3	- .34	- 12.9	- 2.2
5.00	131.250	.4110	1598.4	1.56	19.0	4.4
5.10	134.375	.4094	1596.4	3.53	19.6	6.4
5.20	137.500	.4060	1596.9	3.04	- 4.8	2.3
5.30	140.625	.4019	1599.9	.08	- 29.5	- 4.3
5.40	137.500	.4034	1602.5	- 2.52	- 26.1	- 6.4
5.50	134.375	.4068	1602.5	- 2.50	.2	- 2.4
5.60	131.250	.4110	1599.8	.17	26.7	4.1

T	R	IFN	N	E	EDOT	DEAD
5.70	134.375	.4095	1597.3	2.61	24.4	6.2
5.80	137.500	.4060	1597.5	2.43	- 1.8	2.1
5.90	140.625	.4019	1600.3	- .31	- 27.5	- 4.4
6.00	137.500	.4034	1602.8	- 2.80	- 24.8	- 6.5

APPENDIX IV

PDP-8 Interface Card

FRONT VIEW (Card plug-in side)



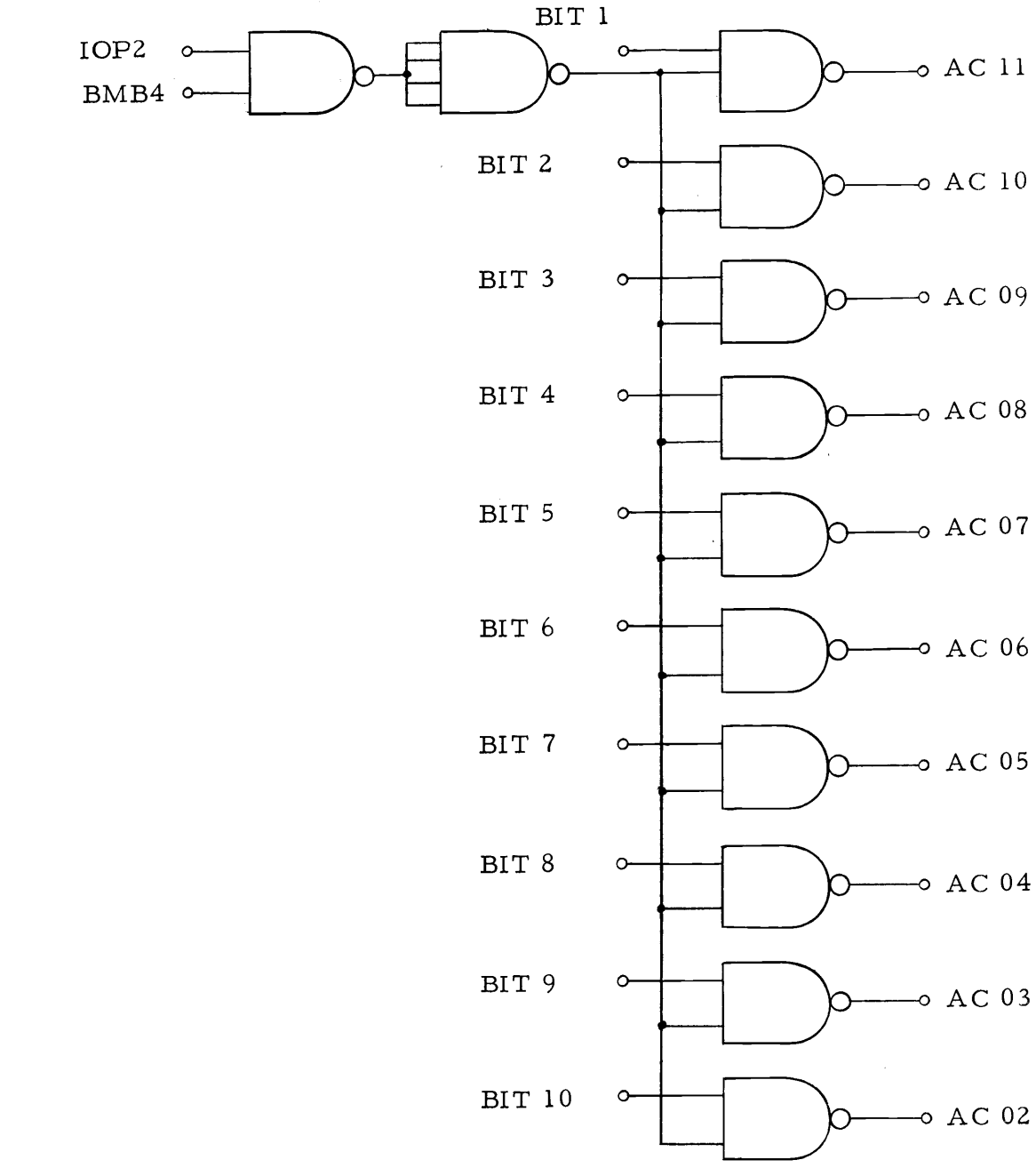
Note: Outputs from PDP-8/L are positive logic.

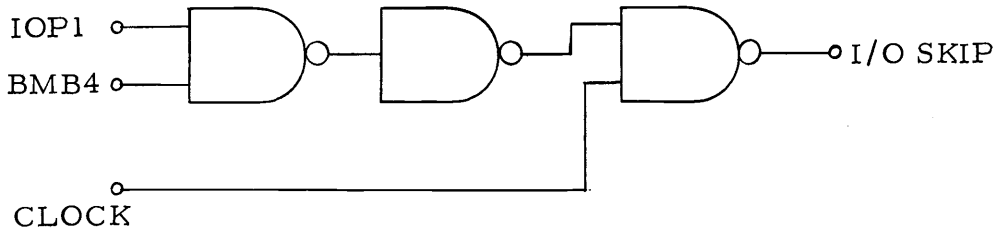
Inputs to PDP-8/L are negative logic.

All Gates are T. I. Series 7400 TTL

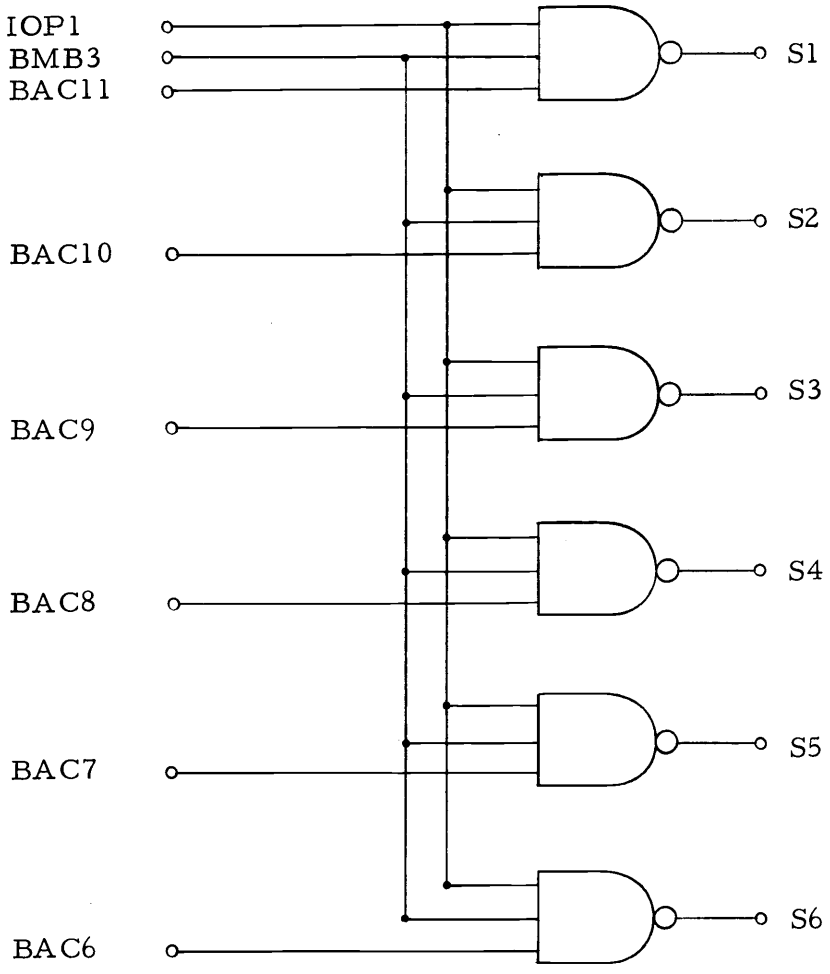
Logic "1" 2.4 volts
Logic "0" 0.4 volts

Input Logic


IOP2
BMB4
BIT 1
AC 11
BIT 2
AC 10
BIT 3
AC 09
BIT 4
AC 08
BIT 5
AC 07
BIT 6
AC 06
BIT 7
AC 05
BIT 8
AC 04
BIT 9
AC 03
BIT 10
AC 02

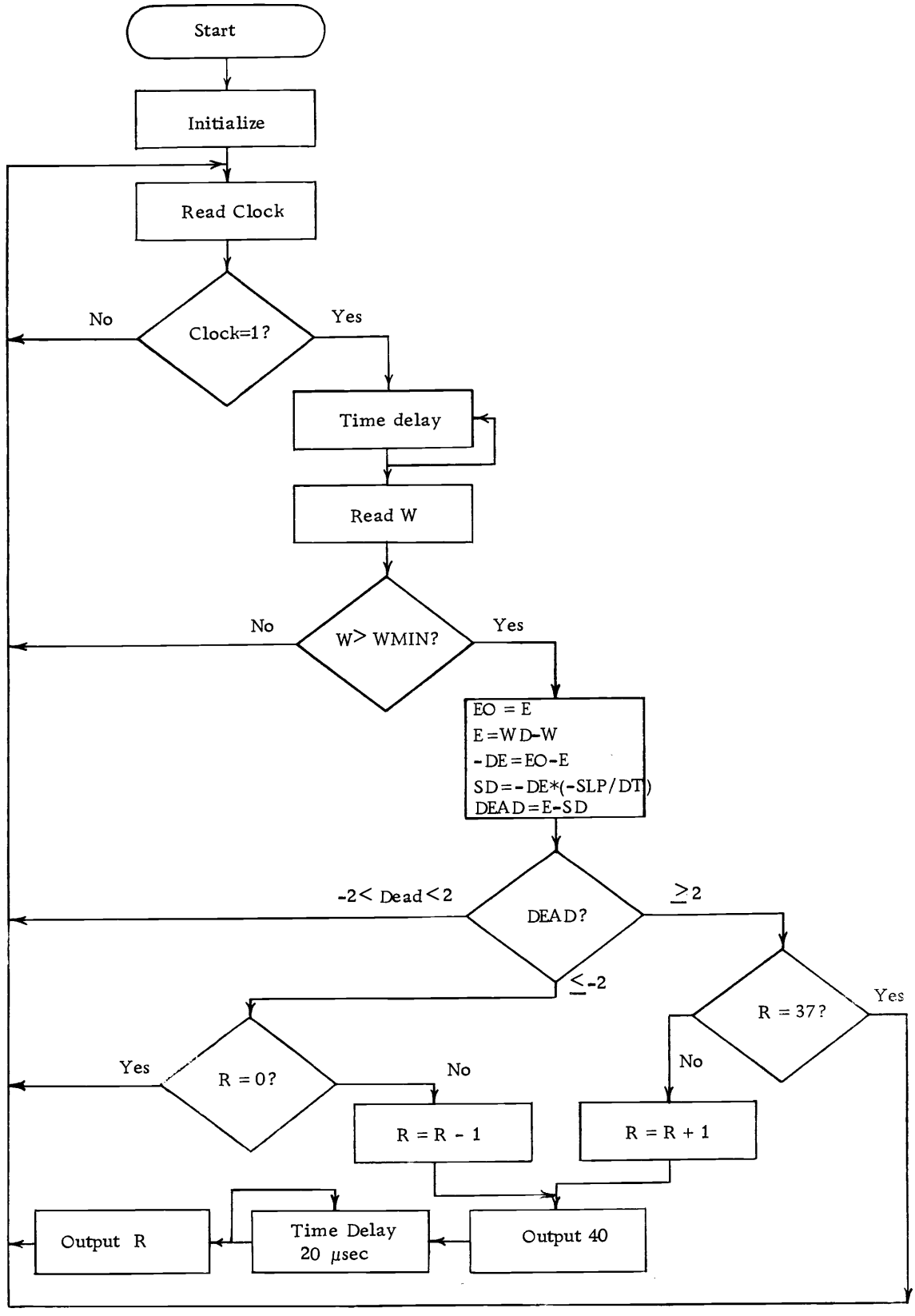

IOP1
BMB4
I/O SKIP
CLOCK

Output Logic



APPENDIX V

Control Program Flow Chart



Control Program

```
00      CLA CLL         7300
01      TAD WMIN        1320
02      CIA             7041
03      TAD W           1321
04      DCA E           3330
05      DCA R           3333
06    A READ T          6201
07      JMP .-1         5206
10      CLA CLL         7300
11      TAD C1          1322
12      DCA CD          3331
13      ISZ CD          2331
14      JMP .-1         5213
15      READ W          6202
16      DCA W           3334
17      TAD WMIN        1320
20      CIA             7041
21      TAD W           1334
22      SMA             7500
23      JMP .+2         5225
24      JMP A           5206
25      CLA             7300
26      TAD E           1330
27      DCA EO          3332
30      TAD WD          1321
31      CIA             7041
32      TAD W           1334
33      DCA E           3330
34      TAD E           1330
35      CIA             7041
36      TAD EO          1332
37      RAL             7004
40      NOP             7000
41      CIA             7041
42      TAD E           1330
43      TAD MTWO        1323
44      SPA             7510
45      JMP .+2         5247
46      JMP B           5261
47      TAD FOUR        1324
50      SMA             7500
51      JMP A           5206
52      CLA             7200
53      TAD R           1333
54      SNA             7450
55      JMP A           5206
56      TAD MONE        1325
57      DCA R           3333
60      JMP C           5267
61    B CLA CLL         7300
62      TAD R           1333
63      TAD M37         1326
64      SNA             7450
65      JMP A           5206
66      ISZ R           2333
67    C CLA CLL         7200
70      TAD C40         1327
71      OUTPUT          6402
72      CLA             7200
73      TAD C2          1335
74      DCA CD          3331
75      ISZ CD          2331
76      JMP .-1         5275
77      TAD R           1333
100     OUTPUT          6402
101     JMP A           5206

120     WMIN            1122
121     WD              1217
122     C1              7754
123     MTWO            7776
124     FOUR            0004
125     MONE            7777
126     M37             7741
127     C40             0040
130     E               0000
131     CD              0000
132     MEO             0000
133     R               0000
134     W               0000
135     C2              7775
```

System Response Data

Data taken from a storage oscilloscope of the response of the motor under direct digital control (see plot, Figure 5-1).

Time (seconds)	Speed (r.p.m.)
0	1500
0.4	1530
0.8	1560
1.2	1540
1.6	1620
2.0	1660
2.4	1700
2.8	1740
3.2	1780
3.6	1820
4.0	1860
4.4	1910
4.8	1960
5.2	2000
5.6	2000