1-1-1976

An adaptive time-optimal digital positioning system.

David E. Bickford

AN ADAPTIVE TIME-OPTIMAL DIGITAL POSITIONING SYSTEM

by

David E. Bickford

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Electrical Engineering

Lehigh University

1976

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Masters of Science

December 7, 1976
(date)

Professor in Charge

Chairman of Department

ACKNOWLEDGMENT

In acknowledgment of his advice and encouragement, I wish to thank my advisor, Professor Arthur I. Larky. I also wish to express my appreciation to the personnel at AMP, Inc. for their support in sponsoring this project.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The use of servo motors in conjunction with various feedback elements have sucessfully been used to accurately control position. This thesis presents an analysis of a digital positioning system which utilizes a minicomputer as the feedback and control element. The system accurately controls and optimizes the positioning time. A fast response servo motor is used in conjunction with velocity and position feedback to control the velocity profile of any given move and to obtain an optimum response.

In addition to optimizing the response for a given load condition, the system has an adaptive capability. As the system load parameters change, the computer is utilized to analyze past performance, and thus, modify the system response to conform to the time-optimal profile.

If for any reason the system has not generated a time-optimal profile, it must modify its response and correct any resulting position error. In this instance, another mode of operation is entered which steps the system into position. The final position mode of operation is also adaptive. As unit distance moves are made, the system is continually monitoring the result of each

move, and if a move is not considered to be unit distance in length, the response is modified.

A detailed analysis of the system predicts the system deceleration and estimates system accuracy. An evaluation of the actual system performance confirmed these predicted system responses.

INTRODUCTION

The objective of this thesis will be to design, analyze, and evaluate an adaptive time-optimal digital positioning system. The system will be time-optimal in the sense that for the given system components, the positioning time will be minimized. The system will be adaptive in that it will adjust its responses based on changes in the system parameters.

The system is configured around a Data General Nova 1220 minicomputer as the feedback and control element. Velocity feedback is obtained by using an analog tachometer combined with an A/D converter. Position feedback is obtained by using a discrete point shaft encoder. The word point refers to each discrete position increment within the encoder. In this particular encoder there are one hundred such discrete points per revolution.

The operation of the system can be broken down into two modes. The first mode can be considered to be a main positioning move and the second a final positioning move.

During the main positioning move, the system is attempting to position itself from one discrete point to another in a time-optimal fashion. The positioning scheme was chosen to be incremental rather than absolute because a particular application requiring such performance initiated this project. However, it should be noted that it would be a very simple matter to incorporate absolute positioning into the system.

During any particular move, the system uses an up counter to accumulate encoder pulses as a measure of displacement. The value stored in this counter is continually compared with a preset value representing the desired length of the move. An offset value representing the difference between the preset value and the contents of the counter is also continually being calculated. This offset value is compared with a table entry proportional to the instantaneous velocity. Of course, these velocity table entries must be in units of position and as a consequence of the comparison, the computer decides whether or not to start slowing down. In this way the velocity profile of the main positioning move can be controlled to confrom to a time-optimal response.

The second mode of operation to be considered is the final positioning mode. In the event that the main positioning move failed to produce an optimized profile and a position error resulted, some correction must be invoked to adjust the position to the desired location. In this instance, a series of unit distance moves will be generated to "step" the system into position. Although these unit distance moves will not result in an optimized profile, it is anticipated that they will not be required frequently.

The adaptive nature of the system can also be considered as two discrete modes. The first adaptive mode monitors the main positioning move and modifies system responses based on errors in that move. The second adaptive mode controls the profile of the unit distance move in order to insure that they are indeed unit distance displacements.

System performance was experimentally evaluated with results in excellent agreement with the predicted performance. As a result of the system evaluation, the accuracy of this particular system is set at

$$-2 \leq d_e \leq 2 \quad \text{points}$$

where d_e is the absolute positioning error.

I. SYSTEM RATIONALE

Consisder the rationale for the development of an adaptive time-optimal digital positioning system. The need for time optimization is evident from the simple fact that time is money. If a given positioning task can be accomplished in a shorter period of time, the throughput of any machine utilizing such a positioning system likewise can be increased.

The need to use a minicomputer to control such a positioning system may not be as self-evident as the need for the system itself. Consider then the rationale for using a minicomputer.

Perhaps the first and most obvious reason for using a minicomputer is the flexibility which it offers the system. As a development tool, the minicomputer allows for major changes in feedback paths and controls algorithms with little or no change in the system hardware. This flexibility thus allows the investigation of alternative approaches to the system configuration at a minimal cost.

Another reason for utilizing a minicomputer is the fact that machines which would require such a time-optimal system already often have a minicomputer as an integral part of the machine control. Thus, the added cost of obtaining a time-optimal positioning profile can be minimal.

II. THE CRITERIA FOR TIME OPTIMIZATION

Consider now the question: What constitutes a time-optimal positioning system? Basically, in order to obtain a time-optimal positioning response, the velocity profile must be such that maximum acceleration and maximum deceleration occur and that the velocity is equal to zero when the final position is reached. A typical example of the profiles of motion for a time-optimal move are shown in Figure 1.

In a practical system, a limit on the maximum allowable velocity may be imposed by the system parameters. Thus, Figure 2 illustrates a set of typical velocity profiles for a real system.

Assuming then, that a system exists which can provide a constant acceleration and deceleration, the key to time optimization will be in the determination of the slow-down point. A slow-down point must be established which will indicate when to start deceleration such that when the velocity equals zero, the position offset equals zero.

The relationship defining this slow-down point can be shown to be:

$$d_s = v_0^2/2\alpha \qquad (1)$$

where

d = position offset

v_0 = instantaneous velocity

α = deceleration constant

The derivation of this relationship appears in Appendix A.

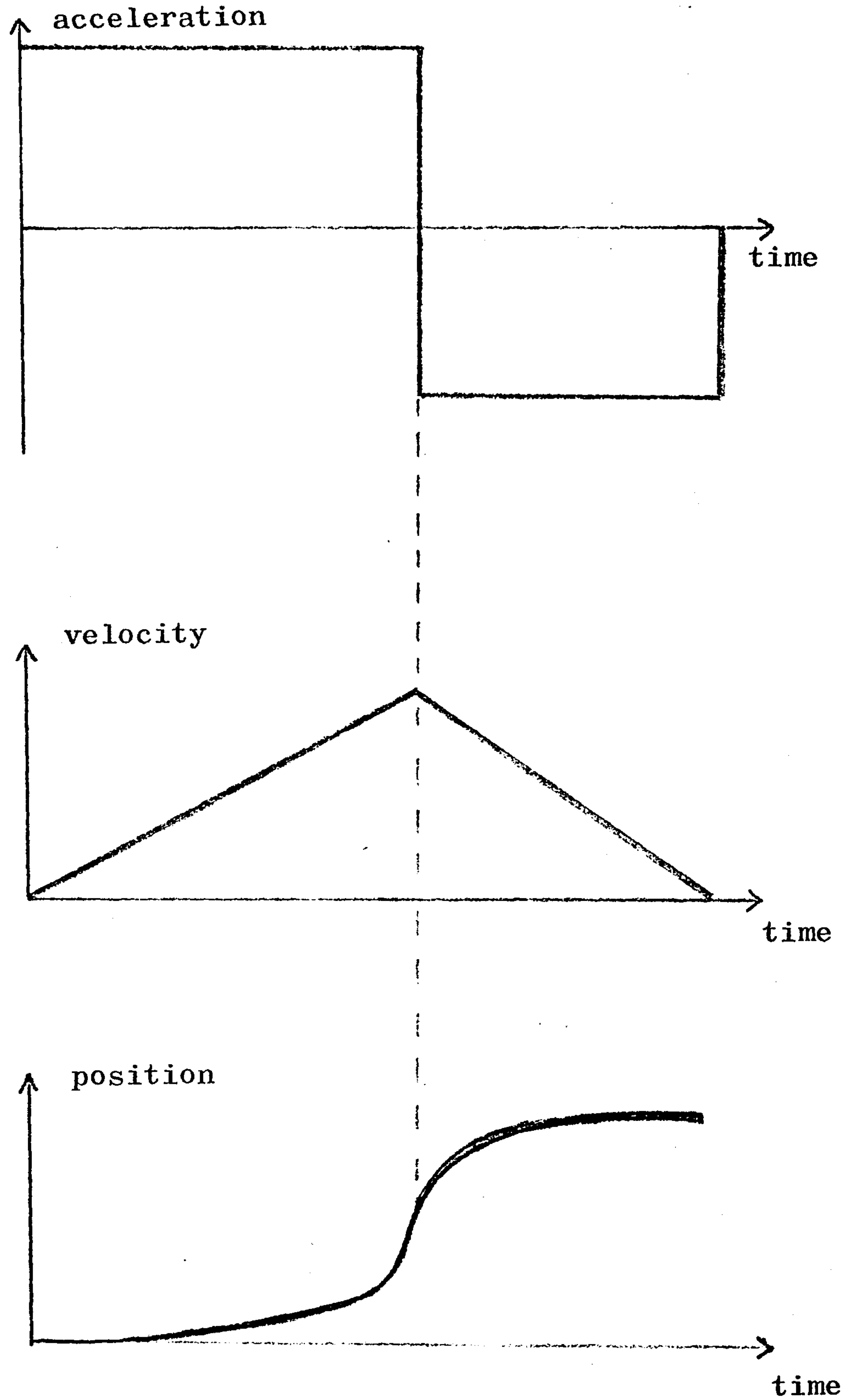


Fig. 1 Profiles of Motion - Time Optimal Move

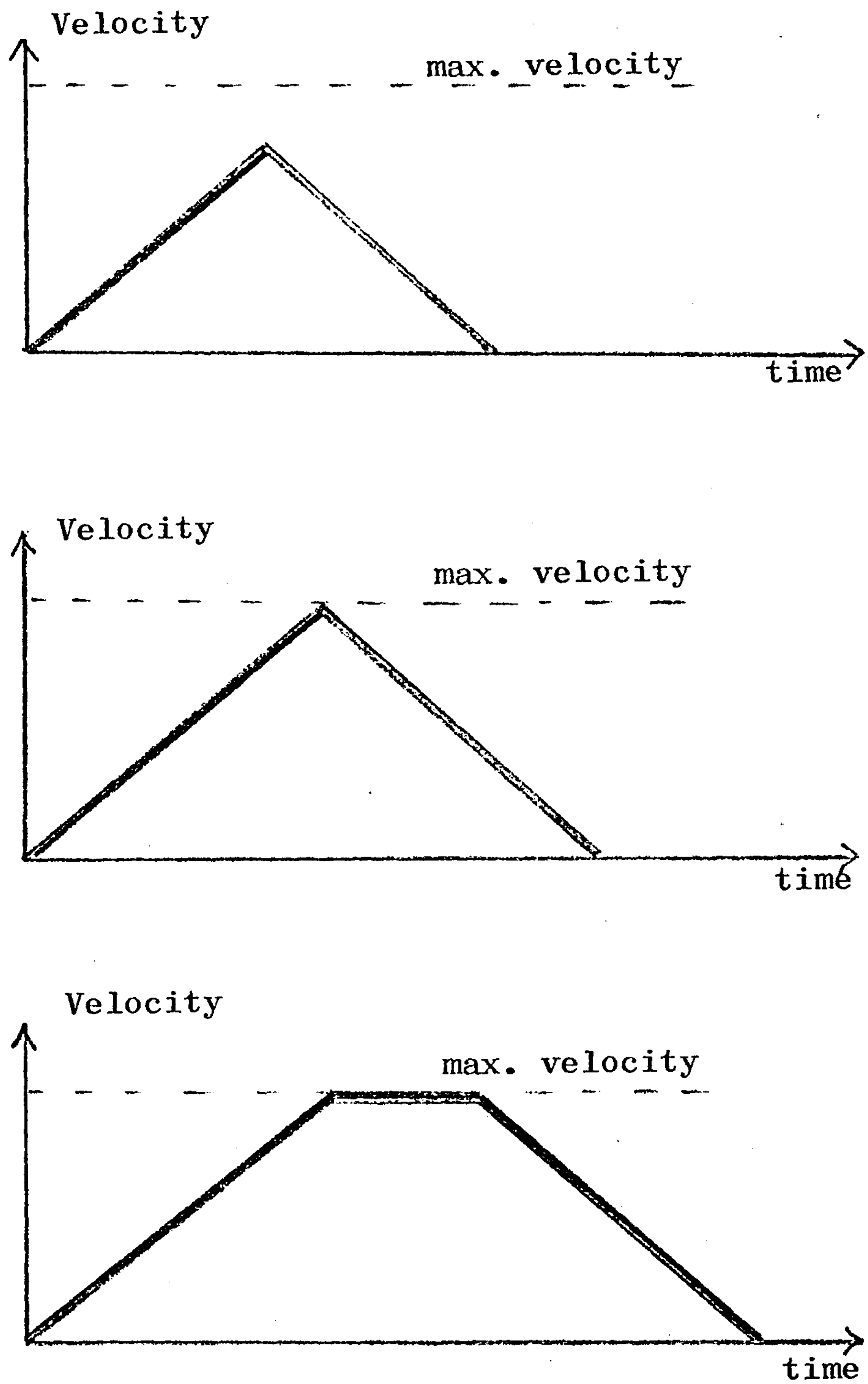


Fig. 2 Velocity profiles for a Real System

III. DESCRIPTION OF THE SYSTEM

The general requirements for the system are: (1) a central processor to handle signal routing and perform any calculations which may be necessary; (2) velocity and position transducers to provide feedback information on the velocity and position; and (3) a motor, drive amplifier, and Input/Output interfaces for the computer.

A basic block diagram of a system incorporating all of the previously mentioned components appears in Figure 3.

It is evident from Figure 3 that the computer is the feedback and control element of the system. The computer reads the length of the move from the position preset; and, upon receiving a start signal, drive is applied to the motor through the D/A converter and amplifier. By monitoring the position and velocity through the position counter and tachometer interface, the motor speed is adjusted to conform to a time-optimal profile.

Some alternatives to this design were also considered. One alternative was to eliminate the real time tachometer and its A/D converter interface and substitute a circuit which would determine the time between

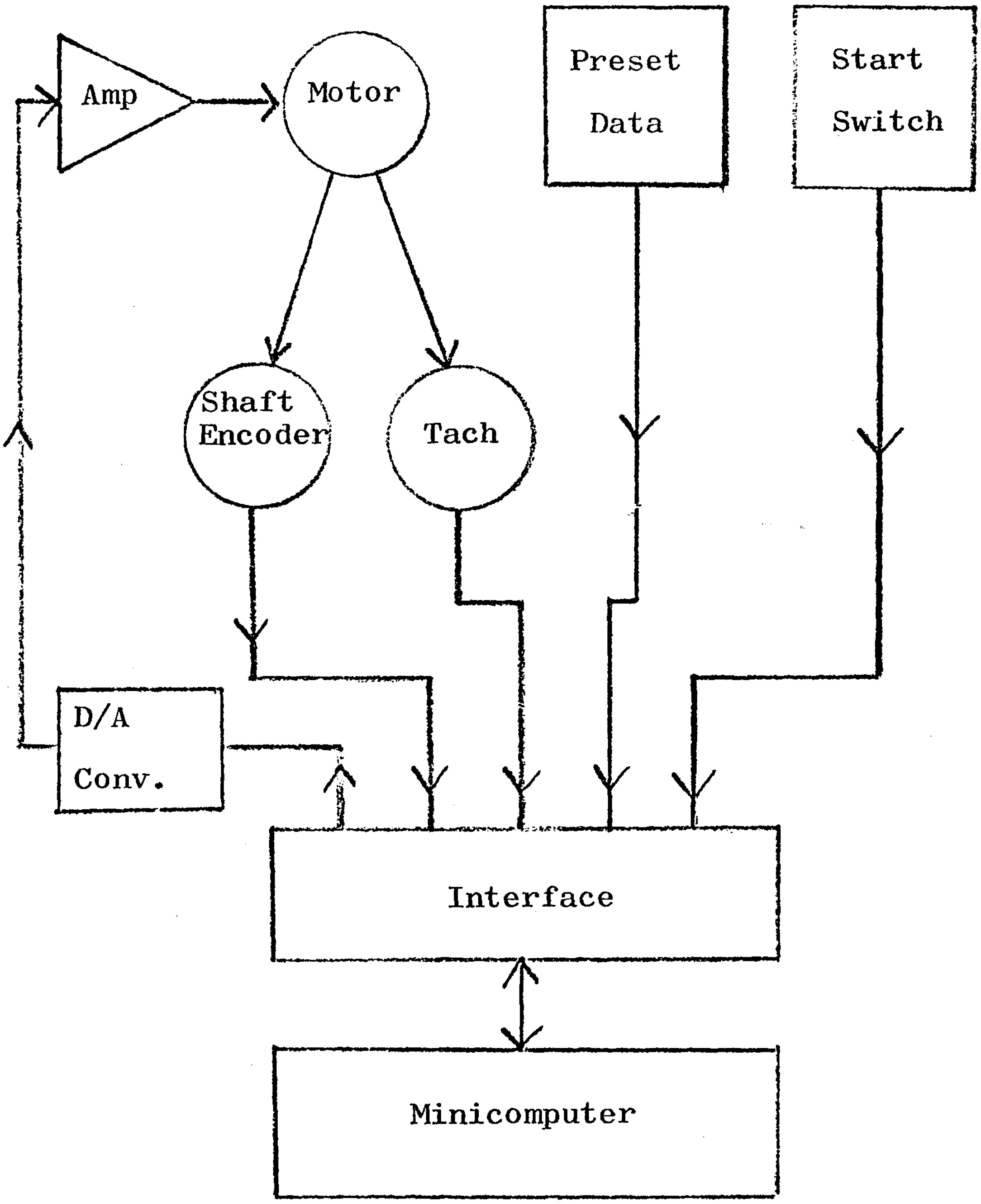


Fig. 3 Block Diagram

encoder points. This approach would work at the expense of conversion time. The time between encoder points can be several orders of magnitude greater than the conversion time of a moderate speed A/D converter; therefore, this approach was set aside in favor of the tachometer scheme.

A detailed block diagram of the system configuration finally chosen appears in Figure 4. All the interfacing is routed through a commercially available general purpose I/O system designed to be used with the Nova 1220 computer. This method of interfacing, although expensive, was chosen for the prototype system for two reasons. First, because of the flexibility it offered and second, because it was a piece of equipment already available in the laboratory. This approach eliminated the detailed design of the special purpose I/O equipment which would otherwise have been required. Since this system was conceived as an evaluation tool, the detailed design of specific I/O hardware can be deferred to a later phase of the project.

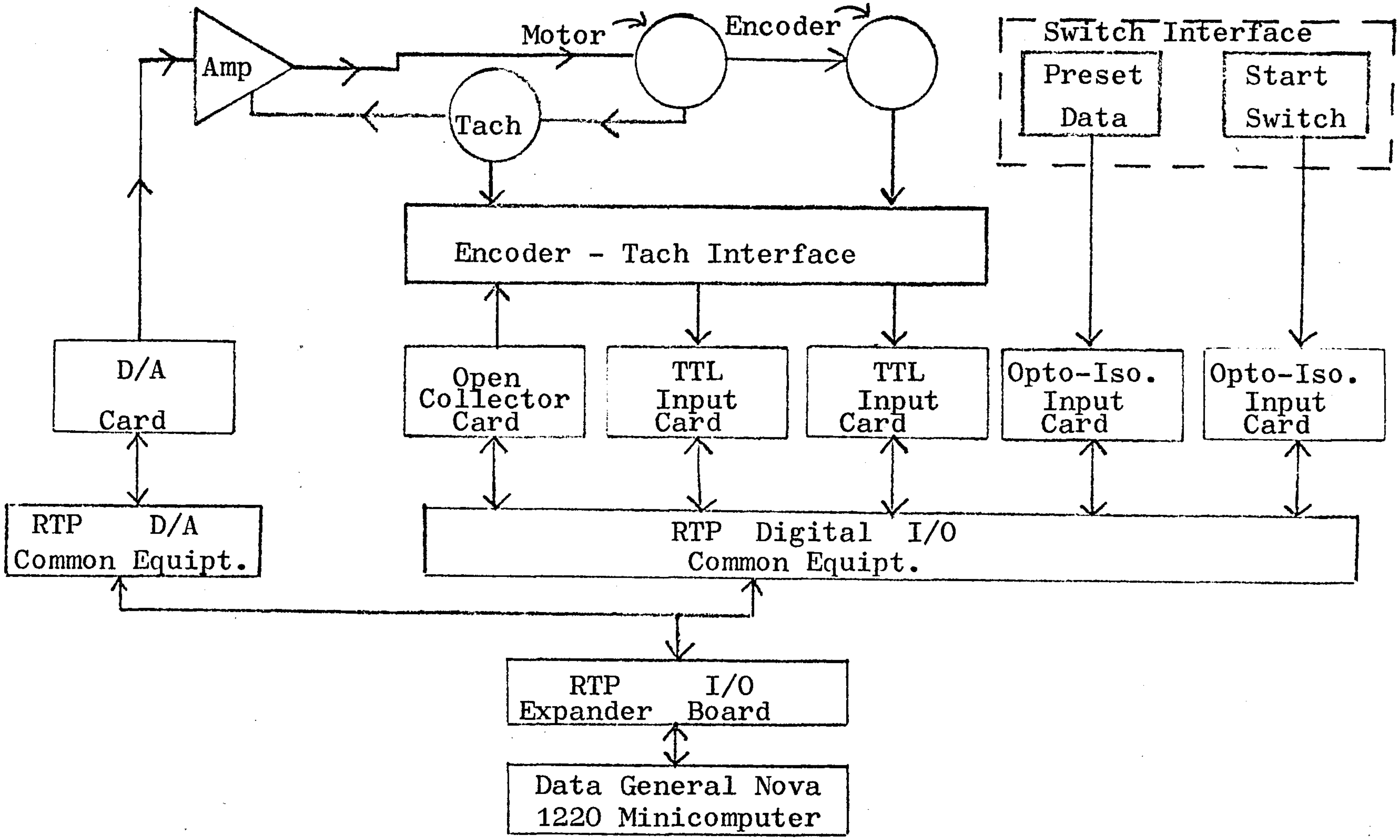


Fig. 4 Detailed System Block Diagram

IV. NON-ADAPTIVE OPERATION

Referring again to Figure 4, consider the non-adaptive operation of the system in more detail. The adaptive operation of the system will not be discussed at this time but will be considered later.

The objective of the system will be to take input data representing a particular displacement and cause the motor to move that exact displacement as quickly as possible; or, in other words, to minimize the time for that particular move. The first step is for the desired displacement to be entered into the preset data switches. Upon receipt of a start signal, a maximum signal is applied to the amplifier which in turn causes the motor to accelerate at its maximum rate (Appendix B). The position data and velocity data are monitored continuously. When the condition for slow-down, as given by equation 1, is met, a maximum negative voltage is applied to the amplifier.

This causes the motor to decelerate at its maximum rate (Appendix B). The velocity is monitored until a zero value is realized. At this point in the cycle, the position should equal that of the preset data and the optimized move should have been completed. If for any reason the system has failed to optimize the move, either because of external influence on the load or inaccurate information about system parameters, the error must be corrected. This is done by another routine which repositions the motor to the desired location. This "final positioning" routine need not be time-optimum since it is intended that this final positioning routine not be required very often.

In addition to correcting for errors resulting from the main positioning move, the final positioning routine must also maintain a given position once the move is complete. In the event that the system is forced out of position by some external influence, the final positioning routine must be able to "step" it back into position. In any event, this final positioning routine must correct for any error in a move as well as maintain the position once it is established. The detailed approach to this final positioning routine will be considered later. After final positioning, the move is complete and ready for another cycle.

The analysis of the system operation starts with a calculation of the inherent position error which exists. Once this is known, a determination of the position accuracy can be made. Inherent position error refers to those errors in position which can not be compensated for or altered by adaptive methods. Any errors outside the inherent error can, of course, be corrected.

Starting with the inherent error of the main position move, the other errors that exist are evaluated. A "dead-band" is established to define an "in position location. It is shown in Appendix C that the position error of the main position move is

$$-3 \text{ points} \leq d_e \leq 4 \text{ points}$$

where d_e is the position error. Thus, any move which ends up within this range is considered complete.

If, when the main position move has been completed, the position error still is outside the dead-band, the final positioning routine must then step the motor into position with a series of unit distance moves. Each unit distance moves is generated by a bipolar pulse to the amplifier. The profile and magnitude of this pulse is illustrated and calculated in Appendix D.

The function of the system during the unit distance move routine will be to read the velocity data; then, if the velocity is zero, and the position does not lie within the dead-band, another unit distance pulse will be issued. If the final position is still not correct, more pulses will be issued until it is. Note that the criterion used to determine the final position will not necessarily be the same as for the main position move. With the unit distance moves, a different dead-band must be determined to signify an in-position condition. This dead-band is relatively easy to determine and is shown in Appendix E to be $\pm$ 2 points. Therefore, if the final position is within two points of the preset value, it will be considered to be "in position". Recalling that the dead-band of the main move was four points, it would make sense to extend the dead-band of the final position routine to be equal that of the main move since the four point dead-band of the main move is acceptable. This will further serve to decrease the time it takes to perform a final positioning, if one is required.

The interface for the encoder consists of a sixteen bit up-down counter and a data latch. At the start of a move, the counter is reset to zero. As the move progresses, encoder pulses increment the counter until the counter data is equal to the preset data.

The tachometer interface is a sample-hold module coupled to an A/D converter. A signal issued by the computer starts a conversion. When the conversion is complete, the data for the A/D converter is read into the computer for processing.

The amplifier-motor combination is depicted in Figure 4. The amplifier chosen is of the constant current variety, which, when coupled with the high-response low-inductance motor, effectively yields a constant acceleration and deceleration as shown in Appendix B. Thus, to obtain maximum acceleration or deceleration, the maximum current of +24 amps or -24 amps must be maintained. Given the gain of the amplifier, it is then a simple matter to obtain the required input signal which guarantees saturation of the amplifier.

As noted earlier, a limit on the maximum velocity of the motor has been imposed by system parameters. In this particular system, one determining parameter is the limits of the A/D converter used for velocity feedback. Since there is a ten volt limit on the A/D converter, and a three thousand rpm limit on the speed of the motor (uncooled), the tachometer feedback signal is adjusted to indicate ten volts for a velocity of three thousand rpm.

V. ADAPTIVE OPERATION

The adaptive nature of the system will now be examined. For a system to be adaptive, it must alter its response to various inputs based upon decisions it has made. The decisions made must, in turn, be based upon past performance information as well as desired future performance.

The advantages to a system incorporating adaptive techniques are somewhat obvious. The first benefit is that the system will be self-correcting in the event that system parameters change. In a non-adaptive system, a change in the friction of the load would result in a different value for acceleration and deceleration. If the acceleration and deceleration values change from what is expected, the system's positioning characteristics will no longer be optimized. Such load changes are not uncommon in a real system in that wear of parts as well as dirt build-up causes change in the friction and inertia of the load. To be able to automatically account for such changes is an obvious advantage.

In order to make the system adaptive, a determination must be made of what to adapt to. In this system two main areas for adaptation will be considered. The first area will be that of adapting to error in position resulting from the main position move and the second area will be that of adapting to the position error resulting from a unit distance move. These two areas will be treated separately starting with the main position move.

In adapting for errors in positioning for the main move, the first consideration will be how much error must exist before adaptive actions will be taken. Obviously this limit on the allowable error will simply be the dead-band previously defined in Appendix C. The next consideration will then be whether or not to adapt the system every cycle or to average the error over several cycles. In other words, it must be determined whether the system should have a fast response or a slow response to errors.

In considering this question, some thought must be given to the construction of the drive system, as well as to the type of load which will be seen. For example, if the load is susceptible to sudden and random interference by some external load such as an operator holding back the drive motor, the fast response solution to adaptability is probably not desirable. Otherwise, the system would continually be adapting to random and inconsistent parameter changes on a cycle to cycle basis. This not only can waste time but can also lead to erratic operation of the system.

A better approach for this system would be to average out the errors over several cycles. Once an average error is defined, it can be used to correct the constants which describe the system operation.

As each cycle is completed, a determination of whether or not the move has positioned within the specified dead-band must be made. If it has, the error is considered to be zero. If not, the percentage error is calculated and that error is placed into a stack. This percentage error is the position error relative to current velocity table entry. As it turns out, this percentage error can be very small; in fact, it may be less than one bit of single precision data; therefore, double precision arithmetic will be used both in the velocity table and in the error calculations.

Once in the stack, the average error can be computed. Certain limits are placed on the size of the stack; namely, that the stack can not have more than fifty entries and that not less than ten entries are required before an average will be computed. If there are less than ten entries, the average is not computed, since there are too few data points to obtain a meaninful average. If there are ten to fifty entries, the mean is computed and then compared with an error limit of $\pm$ 0.03%.

This value, $\pm$ 0.03%, is chosen because it represents a change of one unit of position for the largest possible value for $V_0^2/2\alpha$. Assume now that an average percent error has been computed and it is outside the allowable limit for that error. This indicates that the system must adapt to the error by changing its parameters. The adaptive process will use this percentage error derived from past history to alter its parameters by that percent. After the calculated average percent error is used to change parameters, the stack is cleared and restarted. This insures that a given set of errors will not effect more than one correction. Once corrected, the system should be optimum. It should be recognized that the adaptive routine is used very seldom and therefore is not wasting much computing and positioning time.

Now that the adaptive nature of the main positioning move has been discussed, consider the adaptive nature of the final positioning move. It should be recalled that this final positioning move should not be required very often. In this respect, time wasted in positioning, as well as adapting, is not of primary importance.

Unlike the main positioning move, the unit distance move will have a fast response. Although time is not of prime importance, it would still be advantageous to position as fast as possible. Thus, a cycle to cycle correction, if required, will be invoked.

Also, unlike the main position move, more than one variable will be adaptive. Examine again the bipolar pulse required for the unit distance move. This appears in Appendix D. As can be seen:

$$t_2/t_1 = \alpha_1/\alpha_2$$

Therefore, if the ratio of t_2 to t_1 is incorrect, the velocity of the motor will not be zero at the end of the bipolar pulse, thereby resulting in a move larger or smaller than required. This indicates that one of the adaptive variables will be the ratio t_2/t_1. By sensing the velocity for a zero, or near zero, value, it can be determined how the ratio t_2/t_1 should be altered. It is now obvious that, after each unit distance move pulse, the ratio of t_2/t_1 must be adjusted if a non-zero velocity has resulted.

If the "unit distance" pulse pair does not produce a unit move, as determined from the position information, then the values of t_1 and t_2 must be altered to correct this error before the next unit move. Of course, if t_1 and t_2 are altered, the ratio t_2/t_1 must still be maintained.

The method for making these corrections is straight forward. In the case of the ratio correction, rather than trying to determine a percentage correction based upon final velocity, each correction will reflect the smallest possible change that can be made. This simply involves adding a 1 to t_1 (or t_2) and subtracting 1 from t_2 (or t_1). This then results in a ratio change. Note also that this adds an averaging effect to the final position correction in that it can only adapt a small amount each cycle.

Similarly, for the correction to t_1 and t_2 based upon position error, t_1 and t_2 are both either incremented or decremented by one. As can be seen, this does not truly maintain the ratio of t_2/t_1 except for the case where both t_2 and t_1 are much greater than one. In this system, both t_1 and t_2 are sufficiently larger than one.

The basic unit of time represented by one bit in this system is approximately 4μs. This implies a minimum value for t_1 or t_2 to be in the area of 750 to 800 bits. This is seen to meet the condition for t_1 or t_2 being much greater than one.

A final consideration for the adaptability of the unit distance move is the determination of whether or not the move was exactly one unit of position. Since the allowable error in the final position was established to be $\pm$ 4 points, as long as the unit distance move obtains this accuracy, it is acceptable. Therefore, this implies an allowable range for the unit distance move (P) of

$$1 < P < 8 \text{ points}$$

This is certainly very conservative and since a much better accuracy on the move is available, the limit will be chosen to be

$$1 < P < 4 \text{ points}$$

This guarantees that the final position will always be within $\pm$ 4 points of the preset value.

VI. SOFTWARE CONSIDERATIONS

The entire control program for the system is written in Nova 1220 Assembly Language and uses 2554_8 memory locations. Approximately 1300_8 of these are used for program instructions and the remainder are used for various tables and storage locations. The program is written in modular form, utilizing subroutines whenever possible.

The first and most prominent software trade-off used was a table look-up technique for the values of $V_0^2/2\alpha$. The alternative approach would have been to perform a calculation of $V_0^2/2\alpha$ each time it was required. Since only six bits of the A/D converter are utilized, a table entry for every possible value of $V_0^2/2\alpha$ was calculated in advance and entered into a velocity table. This represents only 2^6 table entries. The table size must be double this because there are both positive and negative values of velocity which can address a table entry. This results in a total table size of 128 entries; not an unusually large number.

This technique results in a considerable time savings since each entry is pre-calculated; therefore, the calculation time is eliminated, with no concessions made to accuracy.

The method of accessing a table entry used is very simple and it promotes further time savings. A base address corresponding to the location of the zero velocity entry is established. This entry is then the middle table element; thus, to access any value corresponding to any velocity, the actual value of the velocity is simply added to the base address. The resulting address is then the address of the desired table value.

Another factor to consider is that each table entry must be a quadruple precision word in order to accomodate the data format of the multiply and divide subroutines as well as to maintain the required precision. The net result is that the size of the table is quadrupled. This still is not an unusually large or unacceptable table size. The only modification required because of this data format will be that the velocity data must be multiplied by four before it is used as an access address for the table.

The two left shifts required to quadruple use very little additional time. This is of prime importance since the system is operating in a real time application.

A final topic relative to the velocity table concerns the initialization of the table. Since the system is adaptive, loading the table with arbitrary numbers proportional to V_0^2 will be adequate. This was tried and proven to work as predicted. After ten cycles, the table adapted to proper values.

Another software feature worth considering concerns the stack used to maintain the history of percentage errors related to past moves. As noted earlier, this stack has a maximum limit of fifty entries. It is accessed by a software stack pointer which is used to determine the location of the next available stack address. Once it is full,the next entry overflows the stack causing the oldest entry to be lost.

In general, the entries in the error table do not correspond to a single entry in the velocity table, but represent the errors due to various moves with various slow-down points. The error entries should be normalized with respect to the table entry to which they pertain. This results in an error table which, when averaged, yields an average percent error which may be applied to correct all entries in the velocity table. This scheme indeed proves to work very well as long as all of the original table entries are proportional to

$V_0^2/2\alpha$.

If, however, some of the table entries are not proportional to $V_0^2/2\alpha$, these will contribute a false average error to the system which will cause the rest of the table entries to be improperly modified. The degree to which these entries affect the rest of the table is dependent first, on how much they are in error and second, how frequently they are used. This problem can be overcome by carefully initializing the $V_0^2/2\alpha$ table to the proper values.

Unfortunately, this correction technique is a linear correction and will fail if the system develops errors which, unlike the velocity table entries, are not proportional to V_0^2. However, a self correcting solution to the problem does exist and can be incorporated into the software. This new approach involves the addition of another table which counts the number of times each velocity table entry resulted in an error outside the dead-band. This structure is depicted in Figure 5.

This histogram table is scanned after each completed cycle of operation. If any count exceeds a predetermined limit (N), the average percent error is exam-

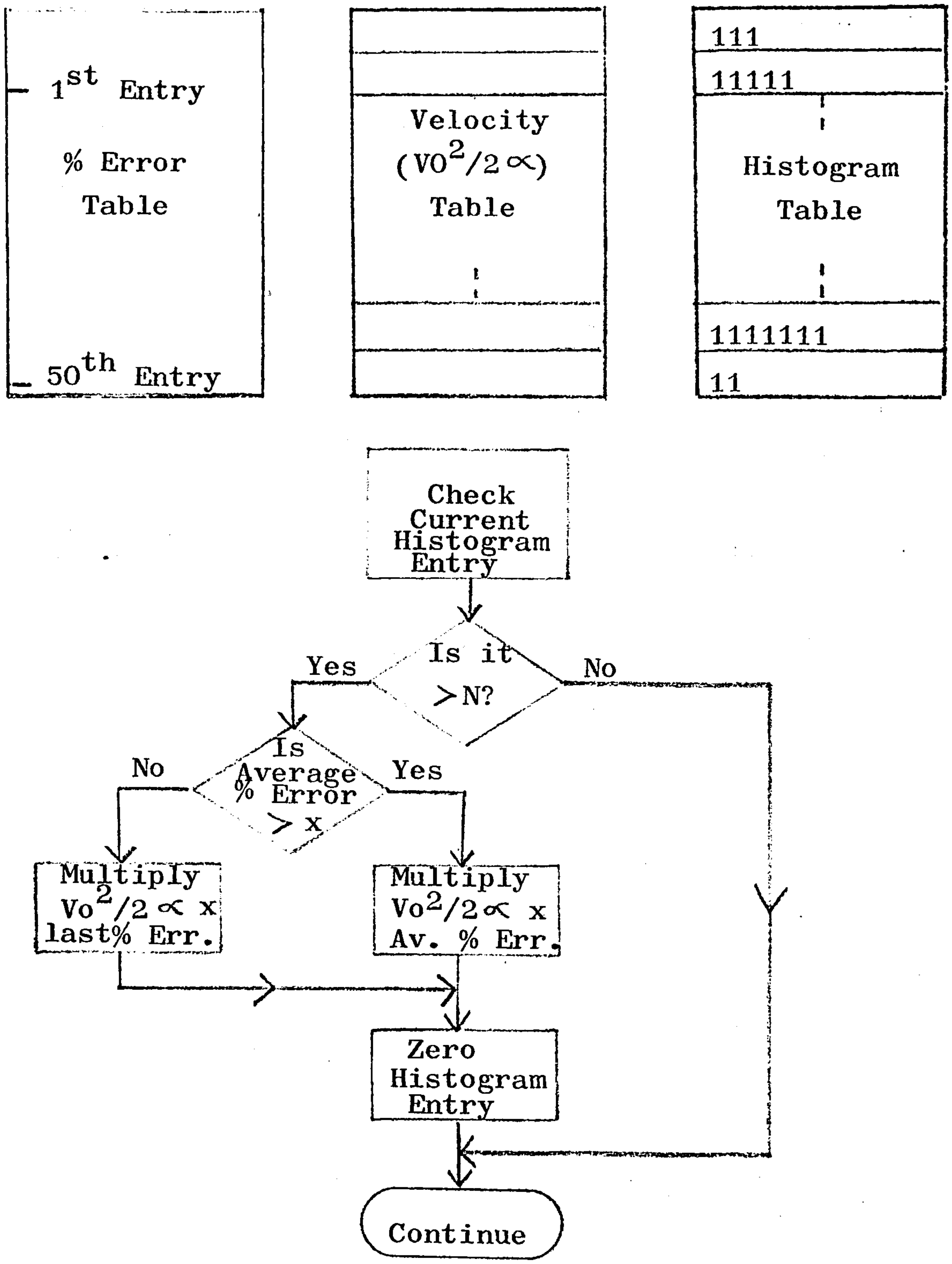


Fig. 5 Software Table Structure

ined to determine if it is greater than the specified limit, as defined earlier. If it is, the corresponding velocity table entry is corrected by the average percent error. Note that only this one entry is corrected. If the average percent error is within its defined limits, then the velocity table is corrected by the percentage error of the last move.

Unlike before, if a correction is made as a result of the averaging table, that table is not cleared and restarted. The histogram table entry corresponding to the corrected velocity table entry is cleared so that it must accumulate N more errors before correction.

In this way, only table entries which are used are corrected; thus saving some correction time at the expense of the histogram generation and scanning time. In order to insure that all the velocity table entries are initially correct, a rigorous test procedure must be run which involves at least N cycles for each entry in the velocity table. In this way, if any velocity table entry is incorrect, it will be modified to reflect the current system parameters. This procedure is advantageous since it means that the machine will be self-calibrating when initially installed, and self-correcting thereafter.

Just as in the velocity value table, some minimum number for N must be chosen. N must be large enough to prevent the unecessary correction of random errors on a cycle to cycle basis, yet not so large that drastic errors go uncorrected for too many cycles. The value of N = 10 was chosen since it is believed that it meets both of these requirements.

VII. PERFORMANCE EVALUATION

A good benchmark for evaluation might be the position error generated at the end of the main positioning move, without regard to the adaptive operation. The evaluation of the adaptive response of the system will be considered later. In order to have some indication of the absolute error at the end of each cycle, a display register consisting of sixteen indicator lamps was devised. At the conclusion of each main positioning move, the absolute error was determined and subsequently displayed. In this way the accuracy calculated for the system could be verified.

As indicated in Appendix C, the worst case position error resulting from the main move should be:

$$-3 \leq d_e \leq 4 \quad \text{points}$$

The system was cycled several hundred times at varying displacements which caused different slow-down points to be generated. The absolute error resulting from these moves was observed each time and the maximum range which resulted was:

$$-2 \leq d_e \leq 2 \text{ points}$$

The most frequent errors were in the range of:

$$-1 \leq d_e \leq 1 \text{ points}$$

This indicates excellent agreement with the predicted performance. The actual performance was, in fact, somewhat better than predicted because the predicted error range was a worst case estimate.

Now that the position accuracy of the main move has been verified, consider the unit distance move. A test program was written which allowed any displacement to be programmed into the system and positioned using only the unit distance move. In this way it was possible to evaluate the accuracy of the unit distance pulse. As predicted, the unit distance pulse was able to position the motor to within $\pm$ 4 points.

Figure 6 illustrates a typical sequence of unit distance pulses during a final position move. Several points of interest relating to Figure 6 should be discussed. Note the point on the graph denoted by Point A. As discussed earlier, this is the end of the first unit distance move. As can be seen, no appreciable delay is required for the software to determine that the final

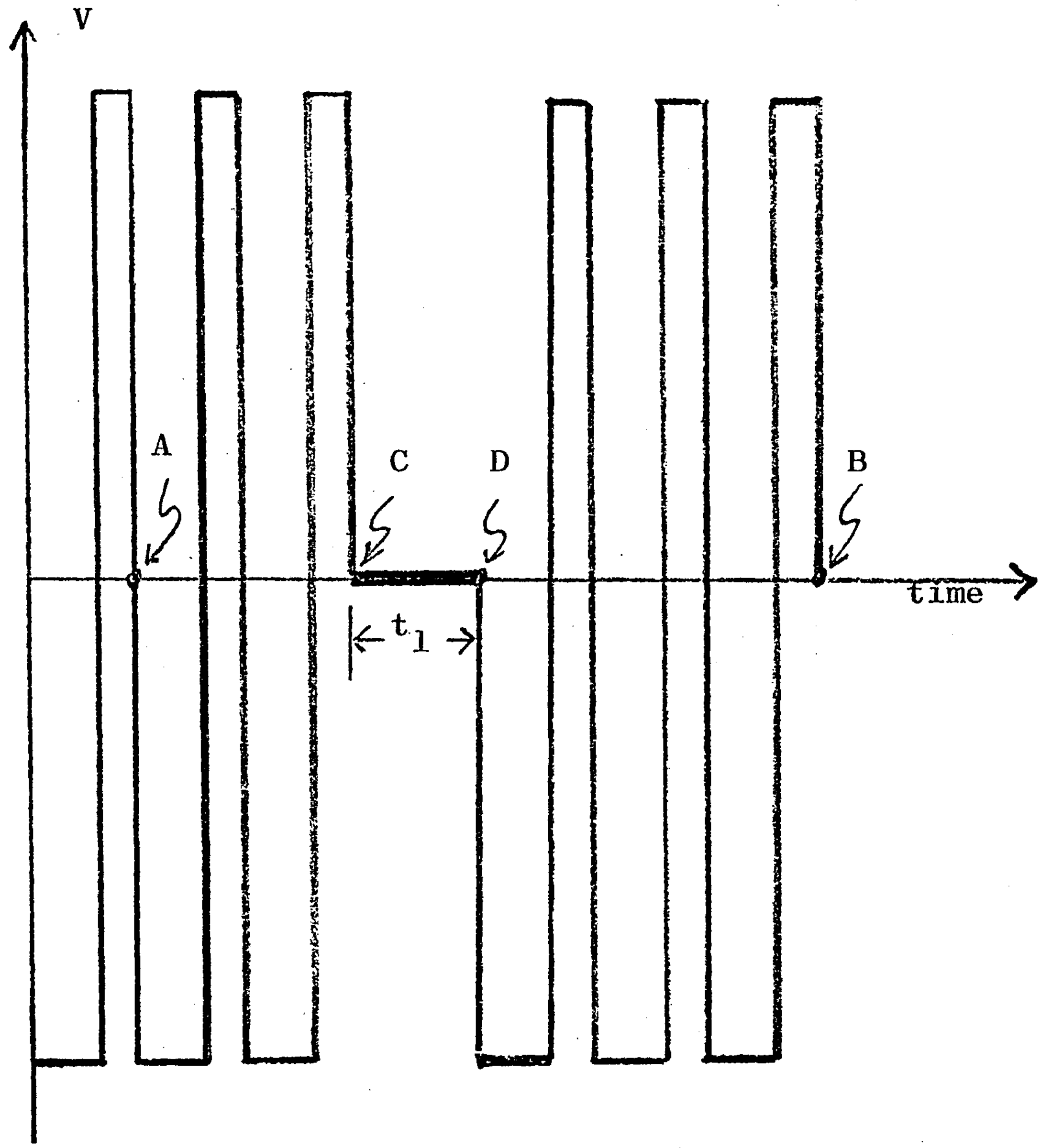


Fig. 6 Typical Sequence of Unit Move Pulses

position has not been reached and start the next pulse. Point B indicates the end position condition.

Still referring to Figure 6, the delay time, t_1, is the time taken by the adaptive routine operating on the unit distance move. At point C it has been determined that the unit distance move has exceeded its limits as previously defined. This routine must now alter the ratio of the positive to negative pulse and/or the width of the pulses. At point D, the unit move positioning resumes.

Now that the main positioning move and the final positioning move have been evaluated separately,consider the evaluation of a complete cycle move; that is, a main move which has resulted in an error. This is shown in Figure 7. Note that the velocity accelerates to a maximum under maximum motor current; then maintains that maximum at reduced motor current; and finally decelerates to zero under maximum reverse motor current. This is followed by a sequence of unit distance moves. This profile is typical of the system performance when a position error has resulted.

The performance of the adaptive routine will now be considered. A velocity table was constructed with values known to be erroneous and, as predicted, after ten identical moves the entry corresponding to that move

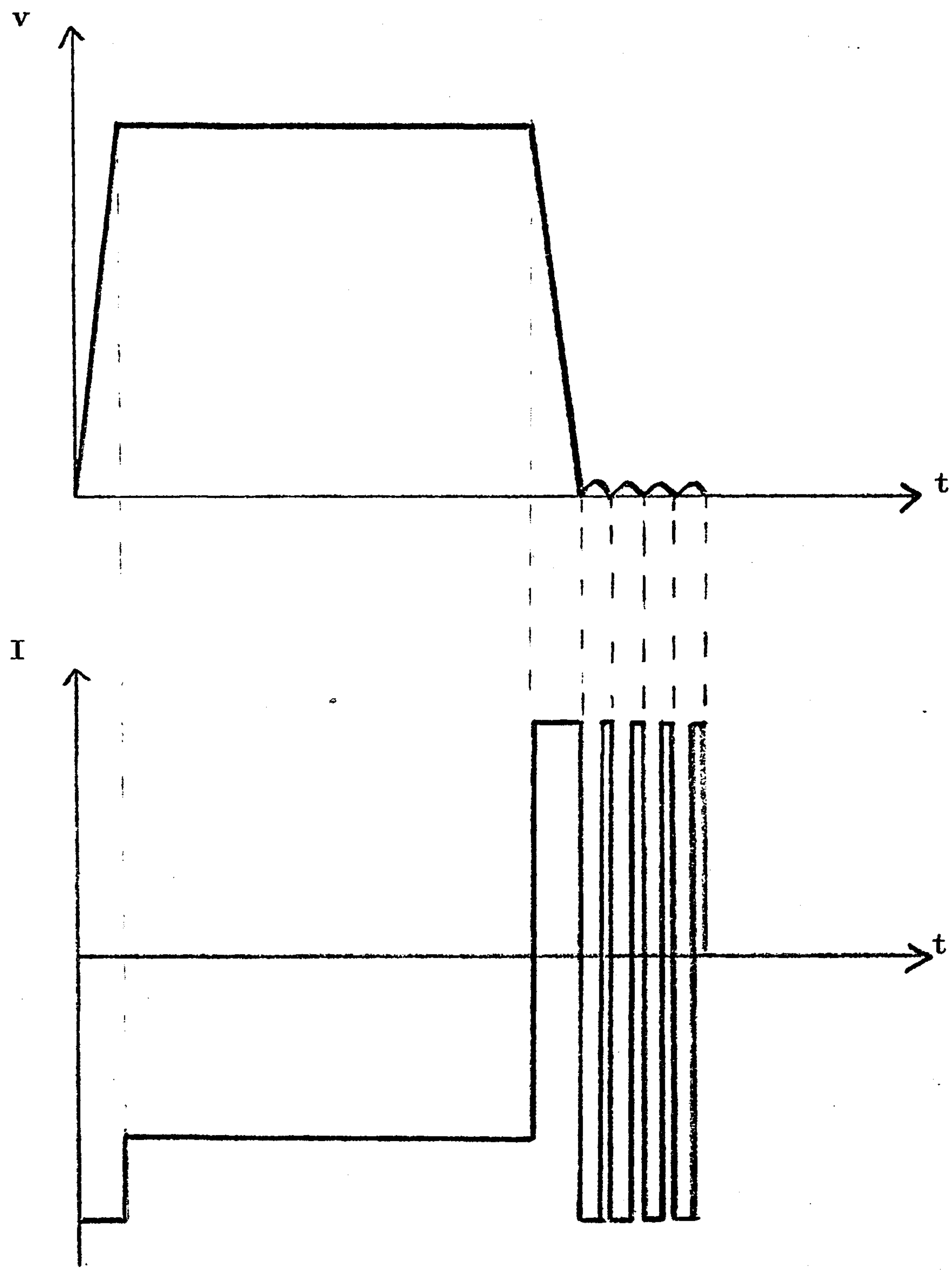


Fig. 7 Actual Move Resulting in Position Error

was corrected to an error free value. The typical profiles of motion for this sequence appear in Figure 8. The velocity profile of Figure 8A must occur N=10 times before the correction is made. Once this correction is made, the move profile of Figure 8B results and position accuracy is maintained at

$$-2 \leq d_e \leq 2 \text{ points}$$

Another test made on the averaging effect of the error stack was the application of an external load to the system for one cycle. The resulting move was greatly in error. This error was entered into the stack which, up to this time, had recorded only zero errors. The external load was removed and fifty more cycles involving the same velocity table value were made. Although the first error was quite large, when averaged with nine to forty-nine zero errors, the error limit was not exceeded. Therefore, no correction was made because of the transient nature of the error.

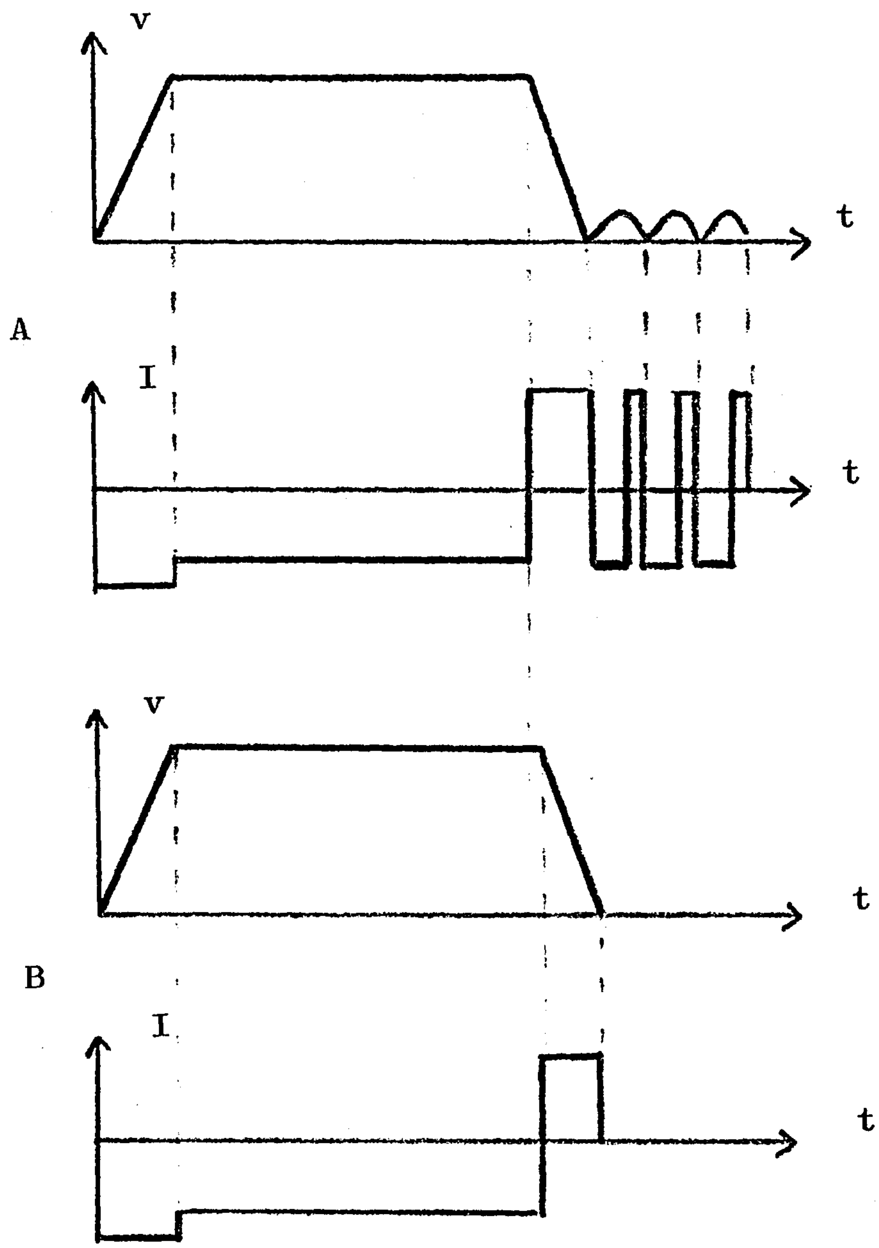


Fig. 8 Adaptive Operation

VIII. CONCLUSION

The system's actual performance was in agreement with the calculated performance. The calculated deceleration was

$$-1.582 \times 10^3 \ \text{rev/sec}^2$$

The measured value for the deceleration was

$$-1.44 \times 10^3 \ \text{rev/sec}^2$$

The calculated value for the in position dead-band was

$$-3 \leq d_e \leq 4 \ \text{points}$$

The actual dead-band observed was

$$-2 \leq d_e \leq 2 \ \text{points}$$

Again, excellent agreement was noted. Finally, the unit distance move pulse was measured to be approximately 16 ms. The calculated value was 5.2 ms.,however, since the unit distance move was extended to a 4 point move, this value of 16 ms. is in reasonable agreement.

The following modifications may be proposed for future improvement of this system: In order to gain more speed, the motor size could be increased to a motor size having a higher maximum velocity. In addition, a motor amplifier combination which can yield a higher acceleration and deceleration rate will act to furthur decrease the positioning time. This is straight forward and the only effect on the system accuracy would be to increase the residual error in the main move.

Recalling that the residual error was dependant on the instantaneous velocity at the slow-down point, as shown in Appendix C; then an increase in the maximum velocity will increase the residual position error. Therefore, accuracy is sacrificed as maximum velocity is increased. This error may be reduced if the resolution of the velocity reading can be increased. This can be accomplished by using more than six bits in the velocity A/D converter.

The position resolution may be improved by using a more finely divided shaft encoder. Then the encoder and counter error will equate to a small position error.

Another area exists for future performance expansion as can be seen by considering a machine which is characterized by predictable load changes on a cycle to cycle basis. In such a case, the simple fixed adaptive velocity table is no longer sufficient since α changes (predictably) from cycle to cycle. As long as the value of α is known before the cycle starts, the proper value of $V_o^2/2\alpha$ can be calculated.

A separate set of tables must now be generated for each possible value of α . This obviously involves a great deal of memory; however, it may be worthwhile since the alternative is to compute $V_o^2/2\alpha$ each time the velocity is read throughout the entire move cycle.

The particular system configuration used was chosen in order to facilitate evaluation and allow for easy modification during design. Now consider how the system might be modified to generate a production model. The first point to examine might be the rationale for use of a minicomputer. As can be seen, a definite need for relatively high speed computing power does exist. Instead of using a separate minicomputer, or part of one which is being used as a machine controller, a separate dedicated microprocessor might be used. As long as the speed of the microprocessor was fast enough to maintain the real time nature of the system, a drastic cost reduction in the overall system could be realized. Along with the microprocessor, dedicated special purpose I/O interfaces could be designed to replace the general purpose I/O structure now utilized. The result would be a unique digital positioning system which would be cost competitive with non-optimal, non-adaptive systems. The usefulness of such a system has indeed been recognized and will be pursued as a future project using this system configuration as a foundation.

BIBLIOGRAPHY

Barron, R. L., "Learning Networks Improve Computer-Aided Prediction and Control", Computer Design, (1975), pp. 65-70

Baumeister, T. Marks Standard Handbook for Mechanical Engineers. 7th ed. revised. New York: McGraw Hill, 1967

Control Systems Research, Inc. Instruction Manual for Installation, Operation, and Maintenance of the System 500 High Response Digital Positioning System. A Report Prepared by Control Systems Research, Inc., Pittsburgh, Pa.

Control Systems Research, Inc. TM - 1014 Technical Manual Model NC 102 Solid State DC Servo Controller. A Report Prepared by Control Systems Research, Inc., Pittsburgh, Pa.

Datel Systems, Inc. Engineering Product Handbook, 1976-77. Canton, Mass., Datel Systems, Inc., 1976

Electro-Craft Corp., DC Motors, Speed Controls, Servo Systems. 2nd ed. Hopkins, Minn: Electro-Craft Corp., 1973

Lajoie, P. E. The Dual Channel Encoder. An Application Report Prepared by Trump-Ross Industrial Controls, Inc., North Billeria, Mass.

PMI. Incremental Motion Servos. An Application Report Prepared by PMI, Glen Cove, N.Y.

Waagen, H. O., "Selecting Motors for Incremental Applications", Instruments and Control Systems, (1972)

APPENDIX A

The Derivation of the Slow-down Point Equation

For a time-optimal velocity profile, a slow-down point must be calculated such that the velocity is equal to zero when the position error is equal to zero. To determine this relationship, assume that the maximum deceleration (α) for the given system is known and constant.

Now using the three basic equations of motion,

$$a = -\alpha$$

$$v = v_0 + at$$

$$d = v_0 t + \frac{1}{2}at^2$$

where

a = acceleration

v = velocity

v_0 = instantaneous velocity

d = position offset

t = time

The time at which the velocity equals zero can be derived:

$$v = v_0 + at$$

$$v = v_0 - \alpha t$$

Since $v = 0$

$$0 = v_0 - \alpha t$$

$$t = v_0/\alpha$$

Now solving for the distance at which to start slowing down (i.e. position offset):

$$d = vt + \tfrac{1}{2}at^2$$

$$d = v_0 t - \tfrac{1}{2}\alpha t^2$$

$$d = v_0(v_0/\alpha) - \tfrac{1}{2}(v_0/\alpha)^2$$

$$d = v_0^2/\alpha - \tfrac{1}{2}v_0^2/\alpha$$

$$d = v_0^2/2\alpha$$

Therefore, the time to start slowing down is when the position offset is equal to $v_0^2/2\alpha$.

APPENDIX B

Calculation of the System Acceleration and Deceleration

To determine the system acceleration and deceleration, the friction and inertia of the load must be computed.

The load of the prototype system consists of seven components:

1. Inertia of the motor
2. Inertia of the tachometer
3. Inertia of the encoder
4. Friction torque of the motor
5. Friction torque of the tachometer
6. Friction torque of the encoder
7. Inertia of the coupling

These items are defined by manufacturers' specifications as follows:

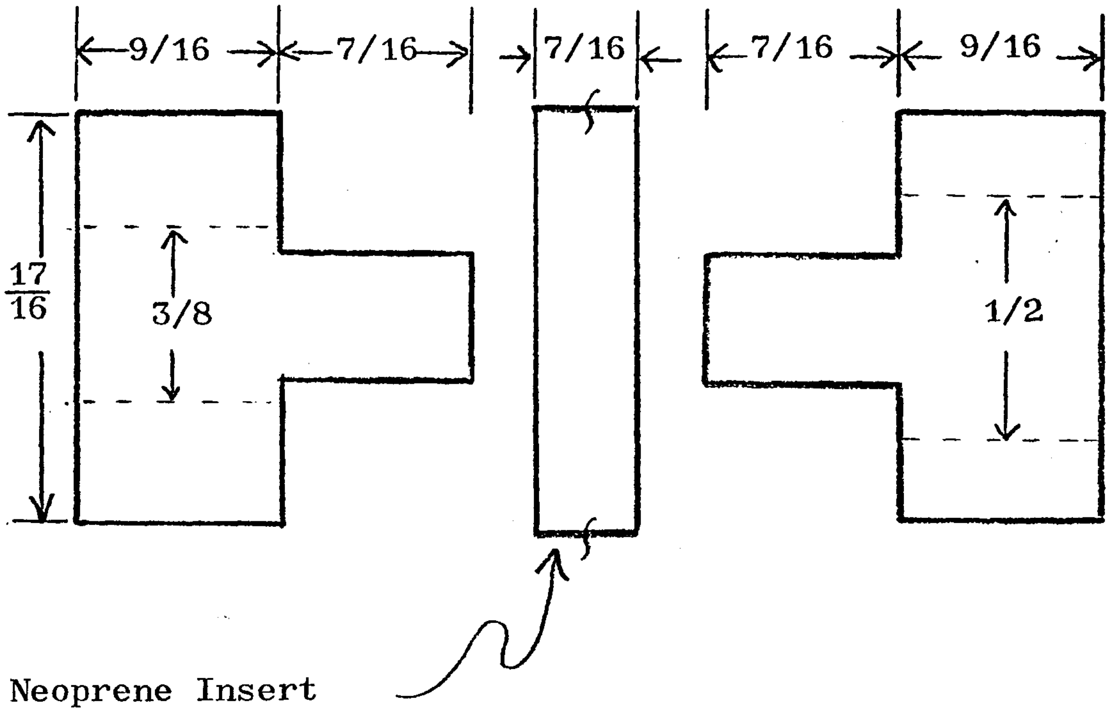
Motor + Tachometer Inertia = 3.3×10^{-2} oz-in-sec^2

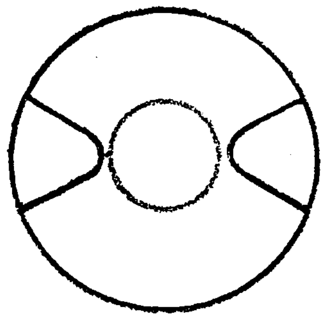
Encoder Inertia = 9.0×10^{-4} oz-in-sec^2

Friction Torque of Motor + Tachometer = 6.0 oz-in

Friction Torque of Encoder = 5.0 oz-in

Item 7, the inertia of the coupling, must be calculated. The physical dimensions of the coupling appear in Figure B1. As can be seen, the coupling consists of





View of Jaws

Fig. B1 Connecting Coupling

three pieces, two similar steel components coupled with a neoprene insert. The inertia of each piece must be calculated, then the sum of these inertias will be the total inertia of the coupling.

Consider first, the half coupling with the 3/8 inch bore. This piece can be further broken down to simplify the calculation. The first segment will be a hollow cylinder representing the portion of the coupling up to the jaws. The jaws themselves can be further simplified by realizing that the jaws will have ¼ the inertia of a hollow cylinder having the same width as the jaws.

The inertia of the first segment is then:

$$I = \tfrac{1}{2}M(R_o^2 + R_I^2)$$

where

M = mass

R_o = outside radius

R_I = inside radius

The material is steel which has an approximate density of 4.51 oz/in^3.

Therefore

$$I_1 = \frac{1}{2} \left((1/384 \text{ in/sec}^2)(4.51 \text{ oz/in}^3) \pi \right.$$

$$\left. (9/16 \text{ in}) \; (R_o^{\,2} - R_I^{\,2})\text{in}^2) \; (R_o^{\,2} + R_I^{\,2})\text{in}^2 \right.$$

$$I_1 = 1.038 \times 10^{-2} \, (R_o^{\,4} - R_I^{\,4})$$

$$I_1 = 1.038 \times 10^{-2} \, (\, (17/32)^4 - (3/16)^4)$$

$$I_1 = 1.038 \times 10^{-2} \, (7.84 \times 10^{-2})$$

$$I_1 = 8.14 \times 10^{-4} \text{ oz-in-sec}^2$$

Now consider the jaws of this half coupling:

$$I_2 = (\tfrac{1}{4})(\tfrac{1}{2}) \left((1/384)(4.51)(\pi)(7/16) \right)$$

$$(R_o^{\,4} - R_I^{\,4})$$

$$I_2 = \tfrac{1}{4}(8.071 \times 10^{-3})(7.84 \times 10^{-2})$$

$$I_2 = 1.582 \times 10^{-4} \text{ oz-in-sec}^2$$

The total inertia for the 3/8 inch bore section is then:

$I_1 + I_2$

Similarly, the inertia for the ½ inch bore section can be derived:

$$I_3 = ½((1/384)(4.51)(\pi)(9/16))(R_o{}^4 - R_I{}^4)$$

$$I_3 = 1.038 \times 10^{-2} ((17/32)^4 - (1/4)^4)$$

$$I_3 = 1.038 \times 10^{-2} (7.57 \times 10^{-2})$$

$$I_3 = 7.859 \times 10^{-4} \text{ oz-in-sec}^2$$

For the jaws of the ½ inch bore section:

$$I_4 = (¼)(½)((1/384)(4.51)(\pi)(7/16))(R_o{}^4 - R_I{}^4)$$

$$I_4 = (¼)(8.071 \times 10^{-3})(7.57 \times 10^{-2})$$

$$I_4 = 1.528 \times 10^{-4} \text{ oz-in-sec}^2$$

The total inertia for the ½ inch bore section is then:

$I_3 + I_4$

Now finally consider the neoprene insert. The approximate density of the neoprene is 0.5 oz/in.^{3}. As a first order approximation and since the density of the neoprene is so much less than the steel, consider a solid cylinder of neoprene 7/16 inches wide.

$$I_5 = \tfrac{1}{2}(1/384)(0.5)(\pi)(7/16)R^4$$

$$I_5 = 8.948 \times 10^{-4}\ (17/32)^4$$

$$I_5 = 7.13 \times 10^{-5}\ \text{oz-in-sec}^2$$

Now the total inertia of the coupling is:

$$I_T = I_1 + I_2 + I_3 + I_4 + I_5$$

$$I_T = 1.98 \times 10^{-3}\ \text{oz-in-sec}^2$$

The total system inertia is then:

$$I_s = I_{coupling} + I_{motor} + I_{encoder}$$

$$I_s = 3.588 \times 10^{-2}\ \text{oz-in-sec}^2$$

The total friction load is:

$$F_s = F_{motor} + F_{encoder}$$

$$F_s = 11.0 \text{ oz-in}$$

The Torque constant of the motor (K_T) is given as:

$$K_T = 14.4 \text{ oz-in/amp}$$

Therefore:

$$K_T \text{ (I amps)} = 11.0 \text{ oz-in} + 3.58 \times 10^{-2} (\alpha)$$

$$\text{oz-in-sec}^2$$

Where α is the acceleration.

Solving for α:

$$\alpha = ((K_T)(I) - 11.0)/3.588 \times 10^{-2} \text{ rad/sec}^2$$

From this equation it can be seen that during acceleration, (I is positive) the friction hampers the acceleration rate and during deceleration (I is negative), it helps.

The amplifier used saturates to plus or minus 24 amps during acceleration and deceleration.

Therefore, during acceleration:

$$\alpha = (\ (14.4)(24)-11.0)/(3.588 \times 10^{-2})\text{rad/sec}^2$$

$$\alpha = 9.326 \times 10^3 \text{ rad/sec}^2$$

$$\alpha = 1.48 \times 10^3 \text{ rev/sec}^2$$

During deceleration:

$$\alpha = (\ (14.4)(-24)-11.0)/(3.588 \times 10^{-2})$$

$$\text{rad/sec}^2$$

$$\alpha = -9.938 \times 10^3 \text{ rad/sec}^2$$

$$\alpha = -1.582 \times 10^3 \text{ rev/sec}^2$$

The actual value of α was measured and agreed within ten percent of those calculated. This is perfectly acceptable since the values given for friction and inertia were only within ten percent.

APPENDIX C

Inherent Position Error Analysis for a Main Positioning Move

The inherent position error of the main positioning move is a combination of several errors. The first of these errors results from the determination of the value of velocity at the slow-down point.

The A/D converter used has nine bits plus a sign bit. Of these nine bits only six bits will be utilized. As will shortly be seen, six bits will allow sufficient resolution as well as minimize the effort expended in calculating the slow-down point. Given six bits, a quantizing error for the velocity can be established:

$$Q = \pm V_{max}/2^6$$

where

Q = Quantizing error

V_{max} = Max velocity limit of the system.

Since V_{max} = 5000 points/second

$Q = \pm 5000/2^6 = \pm 78.125$ points/second.

Therefore the known velocity error (V_e) is

$$V_e = \pm 78.125 \text{ points/second}$$

Now from Appendix A, the predicted distance moved during deceleration is:

$$d = V_0^2/2\alpha$$

The position error due to this velocity error is then:

$$d_e = ((V_0 \pm V_e)^2/2\alpha) - (V_0^2/2\alpha)$$

$$d_e = ((V_e^2/2\alpha) \pm V_0 V_e/\alpha)$$

Now from Appendix B, the deceleration, α, is given as:

$$\alpha = -1.582 \times 10^3 \text{ rev/sec}^2$$

Since a 100 point/revolution encoder is being used

$$\alpha = -1.582 \times 10^5 \text{ points/sec}^2$$

As can be seen, the position error (d_e) is a function of the instantaneous velocity, V_0. With the given system, it can be seen that the position error attributable to the velocity error is given as:

$$0 \leq |d_e| \leq 2.5 \text{ points}$$

for

$$0 \leq V_0 \leq V_{max}$$

Another source of position error is due to a round-off error when V_0 is used as the key for the velocity table look-up. This round-off error results directly in a $\pm$ ½ point error.

A third inherent error is in the position read-in process. When the position is read in, it is known to within one point, as long as the sample time is less than the period between points. This is given by:

$$\text{Period} = 1/V_{max} \quad \text{sec/point}$$

For

$$V_{max} = 5000 \text{ points/sec.},$$

$$\text{Period} = 200 \mu\text{s/point}$$

The software used to realize this function does indeed take less than 200μs; therefore, the inherent error attributable to the uncertainty of position at the slow-down point is:

$$0 \leq e \leq 1 \text{ point}$$

This of course must be rounded off to a +1 point error.

Another way of looking at this error would be to say that a 1 point error exists due to the fact that the data is not used the instant it is read in. Again, as long as the processing time is less than the period of the points, the error is no more than 1 point.

Now the total inherent position error can be seen to be the algebraic sum of the previously discussed errors:

$$d_{e_{max}} = \pm 2.5 \pm ½ +1 \text{ points}$$

$$-3 \leq d_e \leq 4 \text{ points}$$

This then defines the "dead-band" for an in position signal after a main move.

APPENDIX D

Calculation of a

"Unit Move Pulse"

During the final position routine it is necessary to generate a move of unit distance. Since a 100 point/rev encoder is being used, this displacement corresponds to a move of 0.01 revolution. Figure D1 illustrates the profiles of motion for this unit position move.

Referring to Figure D1 it can be seen that maximum acceleration and maximum deceleration will be required to generate this one position move as quickly as possible.

From Figure D1:

$$d = d_1 + d_2 = 0.01 \text{ rev} \qquad \text{(D1)}$$

$$d_1 = \tfrac{1}{2}\alpha_1 t_1^2 \qquad \text{(D2)}$$

$$d_2 = (\alpha_1 t_1)t_2 + \tfrac{1}{2}\alpha_2 t_2^2 \qquad \text{(D3)}$$

$$\alpha_1 t_1 + \alpha_2 t_2 = 0 \qquad \text{(D4)}$$

Using the above relationships, solving for d_1 in equation D1:

$$d_1 = 0.01 - d_2$$

Substituting into equation D2:

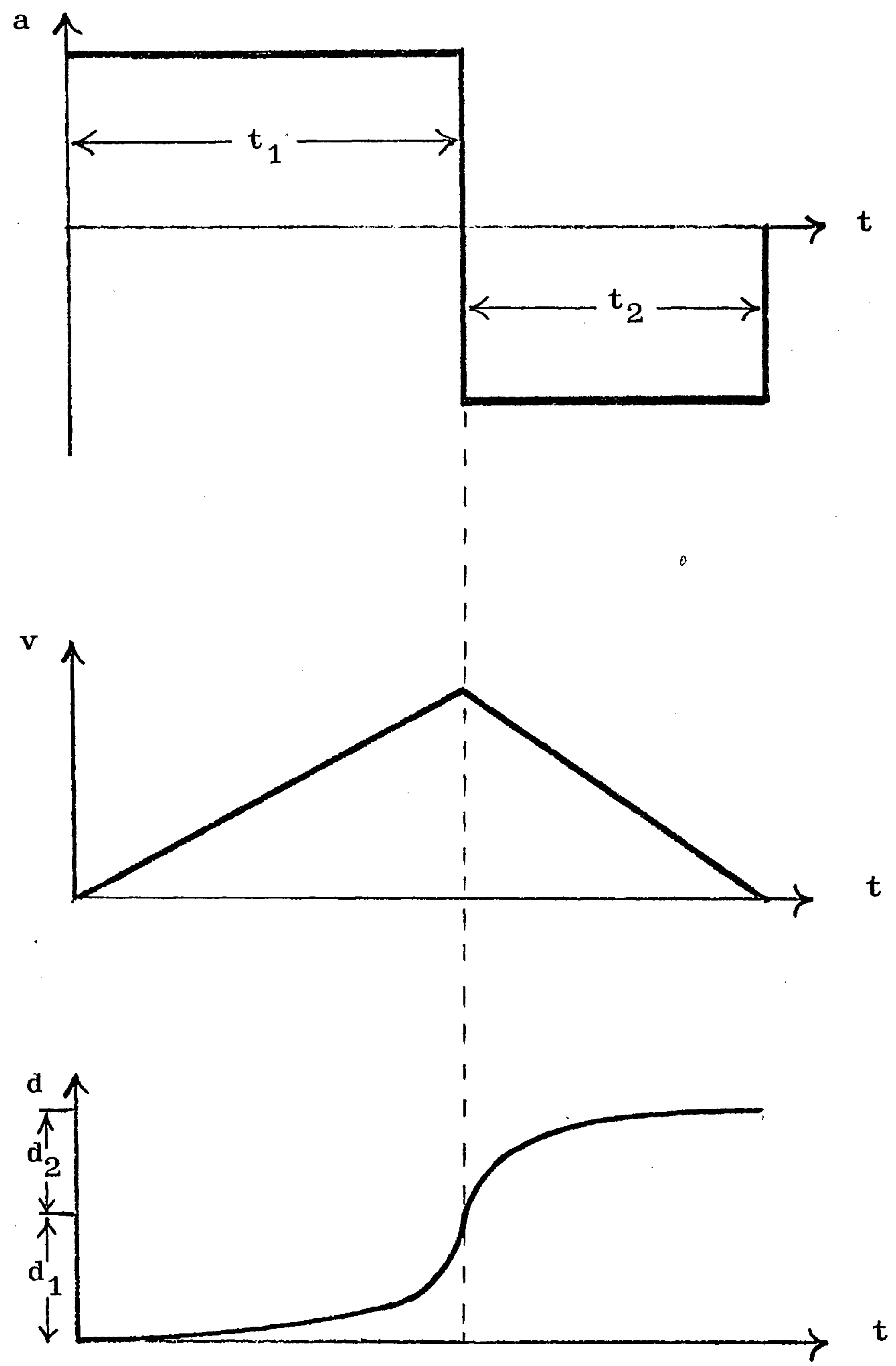


Fig.D1 Unit Move - Profiles of Motion

$$0.01 - d_2 = \frac{1}{2}\alpha_1 t_1^2$$

$$d_2 = 0.01 - \frac{1}{2}\alpha_1 t_1^2$$

Substituting this into equation D3:

$$0.01 - \frac{1}{2}\alpha_1 t_1^2 = (\alpha_1 t_1)t_2 + \frac{1}{2}\alpha_2 t_2^2$$

Now solving for t_2, from equation D4 and substituting:

$$t_2 = -(\alpha_1/\alpha_2)t1$$

$$-\alpha_1 t_1(\alpha_1/\alpha_2)t_1 + \frac{1}{2}\alpha 2(\alpha_1^2/\alpha_2^2)t_1^2 = .01 - \frac{1}{2}\alpha_1 t_1^2$$

Now solving for t_1^2:

$$t_1^2 = \frac{.01}{\frac{\alpha_1}{2} - \frac{\alpha_1^2}{2\alpha_2}}$$

Now from Appendix B:

$$\alpha_1 = 1.48 \times 10^3 \text{ rev/sec}$$

$$\alpha_2 = -1.58 \times 10^3 \text{rev/sec}$$

Substituting:

$$t_1^{\,2} = \frac{.02}{\alpha_1 - \alpha_1^{\,2}/\alpha_2}$$

$$t_1 = 2.64\text{ms}$$

Now substituting this into equation D4:

$$t_2 = -(\alpha_1/\alpha_2)t_1$$

$$t_2 = 2.474\text{ms}$$

Note that the polarity of the pulse will determine the direction of correction, thus Figure D2 illustrates the resulting pulses for a positive and negative position error.

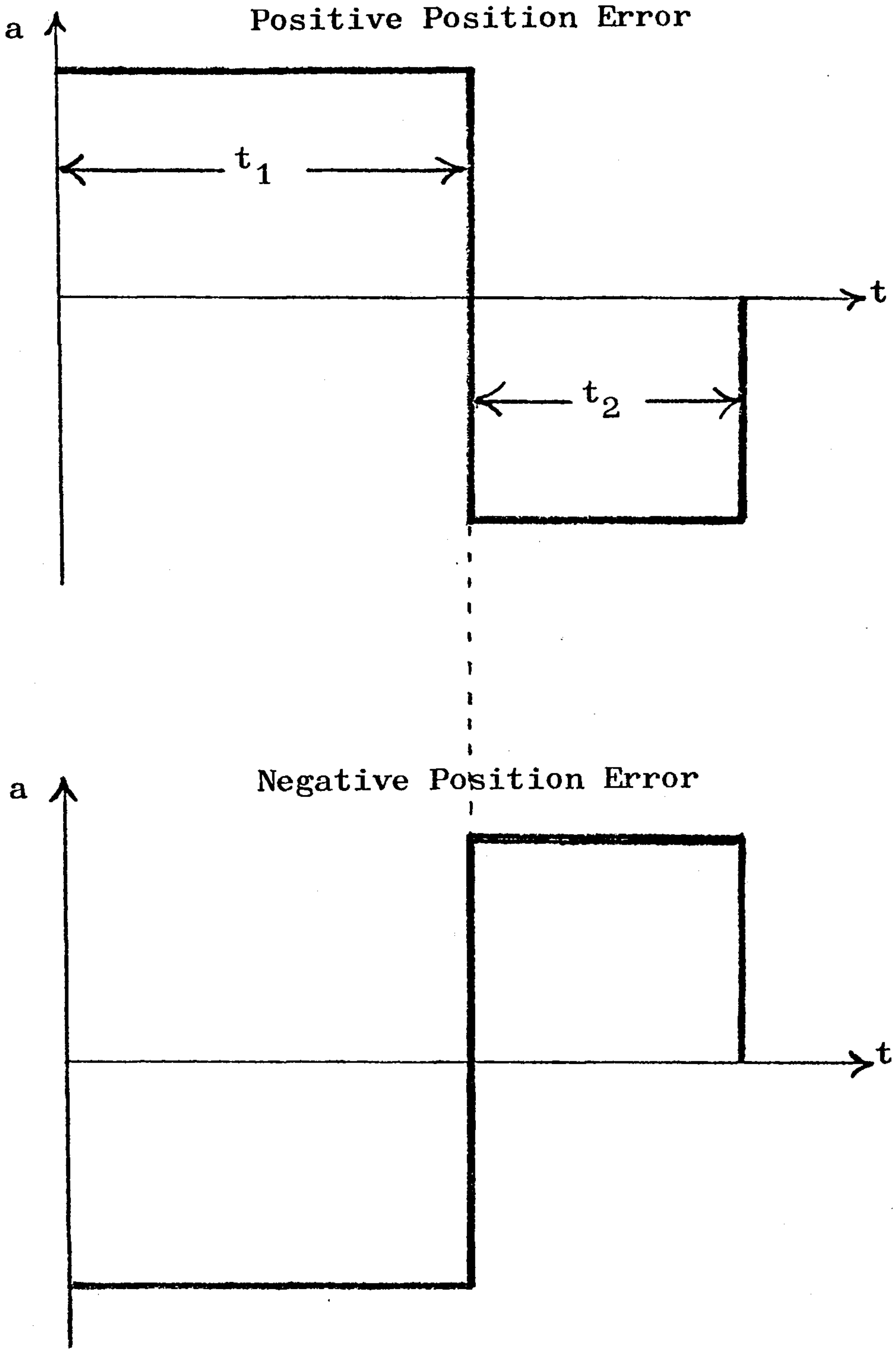


Fig. D2 Unit Move Pulse

APPENDIX E

Determination of the Dead-Band for a Final Position Move

To determine the dead-band for the final position move, first consider the resolution of the position encoder. There is an inherent position error of up to one point in the encoder. This can be expressed as:

$$0 \leq d_e < 1 \text{ point}$$

Since this one point error exists, the unit position pulse described in Appendix D may in fact be a move in the range:

$$0 < \text{Move} < 2 \text{ points}$$

and still be considered a unit move; therefore, the dead-band of the unit move is:

$$\text{Dead-Band} = \pm\, 2 \text{ points}$$

VITA

David E. Bickford was born in Bethlehem, Pa. on February 18, 1950. He received his B.S.E.E. degree from Lehigh University in 1972. He was married in 1973 and currently resides in Harrisburg, Pa. with his wife, Susan. He is employed by AMP, Inc. as an Electronic Development Engineer.