COMPUTER ADAPTIVE CONTROL IN MILLING

COMPUTER ADAPTIVE CONTROL

IN

MILLING

By

PETER R. MACNEIL, B.ENG.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Engineering

McMaster University

April, 1975

MASTER OF ENGINEERING (1975)
(Mechanical Engineering)

McMaster University
Hamilton, Ontario

TITLE: Computer Adaptive Control in Milling

AUTHOR: Peter R. MacNeil, B.Eng. (McMaster University)

SUPERVISOR: Professor J. Tlusty

NUMBER OF PAGES: viii, 182

ABSTRACT:

The purpose of this project was to develop a CNC software base and an Adaptive Control system for a CNC milling machine. The CNC software base consists mainly of Assembly language routines written for the HP-2100 minicomputer for the editing and debugging of N/C data programs. The Adaptive Control system consists of a cutting force transducer, and an Analog to Digital Processor which supply input to an A/C routine resident in the HP-2100 minicomputer. The Adaptive Control system and the cutting forces encountered in milling were simulated in computer programs. The system was stabilized by using the simulation to predict the response of the actual system.

ACKNOWLEDGEMENTS

I would like to thank Dr. J. Tlusty for his guidance throughout the course of the project as well as Dr. Y. Koren and Dr. D. Bonham for their assistance.

I would also like to thank Mrs. Janet MacNeil for typing the thesis.

Table of Contents

List of Figures

Chapter 1

Introduction

Numerical Control[1] is the term used to describe the flexible automatic machining process which uses numerical input, usually from paper tape, to precisely control the cutter path of a machine tool. The numerical input or N/C part program controls not only the cutter path, but also the feedrate and usually the spindle speed during the cutting process. All of the required numerical information is preselected by the part programmer and stored in a permanent form such as a paper tape, to allow the machining of several identical workpieces. Control is generally affected by hardwired circuitry which electronically interprets the coded information on the paper tape in order to control the motions of the machine tool. N/C allows simultaneous control of several axes, accurate machining of complex shapes, decreased set-up times and more management control of shop operations, as well as a general increase in productivity over manual methods.

Computer Numerical Control (CNC)[2] represents a somewhat higher level of N/C wherein much of the conventional controller hardware is replaced by software routines in a dedicated minicomputer. This allows greater flexibility in the controller. Certainly, not all of the hardware may be dispensed with, but tasks such as interpolation, feedrate control and the decoding and buffering of N/C data can be performed by the minicomputer. The availability of the dedicated minicomputer also provides the facility for on-line editing of N/C data tapes, as is discussed in chapter two. "CNC is also a logical building block for Computer Aided Manufacturing (CAM) applications. N/C data management, work station monitoring and reporting, and communication with a control computer are easily accomplished if the controller is a CNC."[1] The addition of Adaptive Control to an N/C machine to improve the cutting process is likewise facilitated if the machine is initially computer controlled. Then, the process improvement may be performed by software algorithms.

Beyond N/C, there are two areas of possible improvement in machining. The first involves decreasing the non-machining time by improving loading, unloading, scheduling and shop routing. The second involves decreasing the actual machining time. It is in this second area that Adaptive Control lies.

Adaptive Control is a logical extension from N/C in the further automation of metalworking machinery. Under Adaptive Control, the cutting process is continuously monitored by observing certain important process variables. The A/C controller, using these process variables as input, alters selected control variables to improve the cutting process. The term "improving the cutting process" may mean increasing the metal removal rate, decreasing the number of out-of-tolerance workpieces, or decreasing the total cost of machining, depending upon the application.

An A/C optimizing system[3] "seeks the best combination of machining rates, at every instant of time, in order to satisfy some predetermined optimum value for metal removal rate and cost."[2] Such an optimizing system would require an on-line tool wear sensor to obtain the economic tool life. Although many researchers have attempted on-line tool wear determination[4,5] by using cutting forces, cutting sound, light

[1]N. Akgerman, "BCLCNC - A Computer Numerical Control System", Manufacturing Technology Dept. Report (Columbus, Ohio, 1974), p. 2.

[2]E. J. Pilafidis, "Adaptive Control Systems and the Manufacturing Process", Process Research and Development Report (Cincinnati, May 18, 1970), p. 3.

reflection and pneumatic methods, a reliable on-line sensor has yet to be developed. Therefore, the alternative to full optimization is then an A/C constraint system wherein machining rates may be changed within predetermined bounds only.

"The degree to which 'optimum' performance can be maintained with adaptive control is dependent largely upon the availability of accurate performance sensing equipment."[3] Therefore, much of the work in the field of A/C milling has dealt with the selection and measurement of process variables. Those parts of the milling process which are monitored must sufficiently describe the process so as to enable a reasonable degree of control over the process. Some of the process variables which various researchers have measured include machine-spindle deflection, spindle torque, spindle horsepower consumption, temperature at the tool-workpiece interface, tool force, motor current, motor temperature, spindle speed, table feedrates, differential hydraulic pressure across hydraulic drive motors, and spindle vibration.[6,7,8,9,10,11,12,13] The controlled variables are generally feedrate and spindle speed. While the effectiveness of the A/C system may increase with the number of process variables measured, cost and complexity definitely increase. For the same reasons of cost and complexity, the latest trend in A/C has been toward simpler "feed-only" systems which measure fewer variables and control only the feedrate. The result is a simple system with considerably lower hardware costs which requires no new machinability knowledge about

[3] Robert J. Valek, "How to Go Adaptive Control", Tool and Manufacturing Engineer (July, 1967), p. 18.

the tool and workpiece to program. Therefore, it is in this spirit of simplicity that the Adaptive Control system described here was constructed.

The adaptive control system described in this thesis, based upon a TOS-FA4V knee type milling machine retrofitted for CNC[14,15] and controlled via a HP-2100 minicomputer with 16K of memory was designed for the control of die sinking. Die sinking done by N/C machines is, at best, inefficient. The load on an end mill when roughing out a complex die shape varies greatly and rather unpredictably. The part programmer is thus forced to specify a feedrate which will be safe for the heaviest cuts, or else spend considerable time trying to determine what feedrate should be required in each part of the die. In the former case machining time is wasted by milling at excessively slow feedrates, while in the latter case, programming times become excessive. Also, the cutters used in die sinking are often slender, "long-series" end mills which are more fragile and more expensive than "standard-series" end mills, thus accentuating the feedrate problem. Therefore, adaptive control is suggested as a solution.

The A/C system suited to die sinking must attempt to maximize the metal removal rate while maintaining a reasonable tool life. Included in the concept of tool life are both tool wear and catastrophic failure (breakage). The equation for metal removal rate in milling is:

$$MRR = S_t \cdot Z \cdot n \cdot b \cdot a \quad \text{(cu. in./minute)}$$

where: S_t = feed per tooth

Z = number of teeth

n = spindle speed (rpm)

b = width of cut

a = depth of cut

In a geometry as complex as that of a die cavity, it would be extremely difficult to vary the depth and width of cut so as to improve the cutting process without deviating from the desired geometry. Therefore, the parameters which could be controlled so as to maximize the metal removal rate in this situation are spindle speed n and feed per tooth S_t.

A simple and general equation describing tool wear is the Taylor tool life equation:

$$T = C \cdot Vb_{LIM} \cdot v^{-n} \cdot s^{-m}$$

where: T = tool life

Vb_{LIM} = limit value of flank wear on tool indicating end of useful tool life

v = surface speed of tool

s = feedrate

Typical values for the indices n and m are 3 and 1.5 respectively. It is therefore obvious that the spindle speed has a much more deleterious effect on tool life than does feedrate. For example, doubling the spindle speed (and therefore, v) decreases the tool life by a factor of 8 while doubling the feed will reduce tool life by a factor of about 2.8. Unfortunately, a reliable, on-line sensor for tool wear does not exist. Blindly controlling spindle speed in an adaptive control mode may thus make tool life unacceptably short. This would not only increase operating costs but may also cause a sub-optimal metal removal rate because of the excessive tool-change downtime. It therefore seems wise to set the spindle speed at the outset according to a reasonable value, perhaps as suggested by standard metal cutting data.

The final remaining parameter in the metal removal rate equation is the feed per tooth S_t. This may be altered by varying spindle speed at constant feedrate or by varying feedrate at a given spindle speed. Not wishing to vary the spindle speed, the feedrate becomes the controlled parameter. The commanded feedrate in the CNC system exists as a single computer word fetched from the required section of core at the beginning of each N/C block. Thus, the adaptive controller need only alter this word according to a preset policy.

The policy must, of course, depend upon some monitored process variables which should vary with the controlled variable and give some information about tool life. Unable to measure tool wear, only catastrophic failure can be considered. Thus, both force and torque acting on the milling cutter are measured. Beforehand, values for the maximum force and torque may be calculated and these are then compared with measured values on-line to determine what action the adaptive controller should take. The maximum force represents a limit determined by the bending strength of the end mill while the maximum torque should determine the limit for failure of the tool in torsion or failure by breakage of one tooth. A very elementary analysis of these limits is given in reference 15. The initialization routine described in chapter 2 makes use of this simple calculation.

Thus, in this A/C system, the following procedure is performed:

1. Taking into account the workpiece material and the tool material, a cutting speed is selected, based on available tables, formulae or experience. The selected cutting speed which assures a reasonable tool wear rate is not altered during milling.

2. Using the initialization routine, the maximum allowable force and torque are calculated. The "optimal" values of force and torque which are taken to be 80% of the maximum allowable values are automatically stored as constants in the computer.
3. During milling, the feedrate is adaptively controlled so as to be at its highest value without exceeding the constraints set in step 2. The actual force and torque are continuously monitored on-line and used to improve the feedrate.

The work discussed in this thesis follows that of reference 16 wherein an A/C algorithm was suggested, the design of the Analog to Digital Processor was discussed, and a simple method of obtaining force and torque criteria was discussed. The transducer discussed here is a re-designed version of that discussed in reference 16.

Chapter 2

CNC Software System

Introduction

One of the big advantages of a CNC (Computer Numerical Control) system as compared with a conventional N/C system is the capability of correcting and editing the N/C data program. In a CNC system, the N/C data tape is read only once for each manufacturing series and is stored in computer memory. This allows the part programmer to examine and modify the data via the peripheral devices of the computer. N/C data is entered via a high speed reader and then may be examined with the aid of an X-Y plotter capable of showing three orthogonal planes of the cutter path. The teletype may be used to examine and alter the data program and a high speed punch may be used to produce a corrected copy of the N/C data program. This chapter deals with the software developed to perform these many functions.

The software was developed in a modular form, so as to enable the operation of some programs independently of the rest of the software system, or without the use of all the peripheral devices. These software routines may be divided into three separate groups:

a) control routines,

b) N/C data service routines,

c) A/C auxiliary routines.

Supervising all of these routines is a COMMAND routine which allows the user to access any of these routines easily from the teletype.

a) Control Routines

Control of either the milling machine or the X-Y plotter is done in a foreground-background mode; the computer continuously loops in a main program (which, in the adaptive control application, is the A/C program) until an interrupt occurs causing a transfer of control to the CNC routine which executes the N/C data program. Interrupts occur at regular intervals, caused by an external clock. The routines in this group, therefore, are:

1) CNC routine: This is discussed in detail in Reference 16. Its purpose is to duplicate the operation of a conventional N/C controller.

2) MAIN program or A/C program: The former is simply a dummy program used in the computer numerical control application. The latter is the program used to control the feedrate in the adaptive control application. This is discussed in more detail in Chapter 6.

b) N/C Data Service Routines

These routines manipulate the N/C data program so as to exploit the inherent capabilities of the CNC system.

1) READ routine: This routine enables the computer to read an N/C data tape via the high speed tape reader. The tape may be in one of two formats: tab-sequential format with incremental dimensions or word address format with incremental dimensions. (A different routine is required for each format.)

2) PRINT AND PUNCH routine: This routine enables the user to obtain a listing of the N/C data program on the teletype or

a paper tape of the N/C data program from the high speed punch. The format produced is compatible with the format used in reading the tape.

3) DATA EDITOR routine: This routine is used to correct and edit the N/C data stored in the computer. These corrections, entered via the teletype include changing, shifting, adding, deleting and repeating any blocks or groups of data blocks. A listing of a selected series of data blocks may be obtained and a new data tape may be produced.

4) DATA DIAGNOSTIC routine: This routine is helpful in debugging N/C data programs. It has three main functions:

    a) checking for illegal codes of G (preparatory function), M (miscellaneous function) and F (feedrate code);

    b) comparing the initial and final radii of every circular segment.

    c) determining the incremental distance sum in each axis of motion.

5) PDP-8 LINK routine: This routine enables data transmission at a rate of 110 baud in both directions between the PDP-8 and HP-2100 minicomputers. Data sent to the HP-2100 may be printed on the teletype or punched on paper tape.

6) CUTTER COMPENSATION routine: This routine performs a cutter radius compensation algorithm on a stored N/C data program according to input parameters entered via the teletype.

These N/C data service routines are generally used as follows. A data tape is read through the high speed tape reader using the READ routine and is stored in computer memory. The programmer then carries out any required corrections using the DATA EDITOR routine. He is able to check the new data using the DATA DIAGNOSTIC routine and then run the N/C data program on the X-Y plotter using the control routines. If the result is satisfactory, he may produce a part. If the part is satisfactory, he may continue producing parts and make a data tape for future use. If at any point the result is not satisfactory the procedure may be repeated. The CUTTER COMPENSATION routine may be used on correct N/C data at any time when the cutter to be used differs slightly in diameter from the programmed one.

c) A/C Auxiliary Routines

These routines service only the adaptive control function.

1) INITIALIZATION routine: This routine calculates an optimum force and torque (for use in the A/C routine) according to requested inputs.

2) TRANSDUCER CALIBRATION routine: This routine prints out values of the three channels of the adaptive controller. It is useful both in transducer calibration and in balancing of strain gauge bridges.

Command Routine

This is the supervisory routine which allows the user to access all of these routines easily from the teletype. When started at the starting address of this routine (6000_8), the computer requests an input, by prompting with:

COMMAND

The required input is a single letter followed by CR (CARRIAGE RETURN) and LF (LINE FEED). The possible inputs are as follows:

1) N CR LF [4]

This input causes the execution of the N/C data program currently in memory. The background routine is either MAIN or A/C, whichever is stored at the time. The interrupt system is enabled so that an interrupt caused by the teletype will transfer control to the DATA EDITOR routine. Upon completion of the N/C data program, the computer halts. This is done so as to facilitate production of several identical parts; the N/C data program may be executed again by simply pressing the computer's RUN pushbutton.

2) A CR LF

This input begins execution of the A/C routine (or the MAIN routine if it is present). This is helpful when debugging a new A/C policy routine. The computer loops continuously in this routine; the interrupt system is disabled.

3) E CR LF

This input transfers control to either the A/C or MAIN routine and establishes a teletype interrupt linkage. Thus, when an interrupt is caused on the teletype (this is discussed in the DATA EDITOR routine), control is transferred to the DATA EDITOR routine. Entering the character ↑ while in the DATA EDITOR routine returns control to the COMMAND routine. Entering the

[4] All user inputs will hereafter be underlined for clarity although they are not underlined in practice.

character E will return control to the MAIN or A/C routine.

4) R CR LF

This input reads an N/C data tape into the computer via the high speed reader. Upon completion, the reader halts and control is returned to the COMMAND routine.

5) P CR LF

This input will print on the teletype a listing of the N/C data program currently in memory. Of course, if the teletype's tape punch is ON, a tape will also be punched. If the user desires a tape only, he may enter the number 15_8 in the computer's switch register before entering the above input. The tape will then be punched on the high speed punch. Upon completion, control is returned to the COMMAND routine.

6) D CR LF

This input starts the DATA DIAGNOSTIC routine's examination of the N/C data program currently in memory. Results are printed on the teletype. Upon completion, the COMMAND routine regains control.

7) L CR LF

This input transfers control to the PDP-8 LINK routine where the user is asked to give an input describing the direction of data transmission. If, at this point, the character ↑ is entered, control returns to the COMMAND routine. During data transmission, if the character # is encountered, control returns to the COMMAND routine.

8) C CR LF

This input causes the CUTTER COMPENSATION routine to be executed on stored N/C data. Again, entering the character ↓ at the beginning of the routine returns control to the COMMAND routine. Upon completion, control automatically returns to the COMMAND routine.

9) I CR LF

This input starts the INITIALIZATION routine which returns control to the COMMAND routine upon completion.

10) T CR LF

This input begins the TRANSDUCER CALIBRATION routine. Upon completion of the routine, the computer halts. This allows multiple execution of the routine. No return to the COMMAND routine occurs.

N/C Data Service Routines

1) READ Routine

This routine is used to read an N/C data tape into computer memory via the high speed reader. The routine also calculates and stores the appropriate value of the PATH or RADIUS[5] for each block of data while reading the tape.

N/C data tapes, in general, can be in one of four formats:

a) tab-sequential format with incremental dimension data;

b) tab-sequential format with absolute dimension data;

c) word address format with incremental dimension data;

d) word address format with absolute dimension data.

[5]For an explanation of these parameters, see Y. Koren and J. Tlusty, "CNC/AC System for a Milling Machine - Part I", Metalworking Research Group Report No. 43 (Hamilton, March 1974).

At present, READ routines are available for either format (a) or (c). (Actually, the tab-sequential format used may be more correctly termed an "equivalent tab-sequential" format since the SPACE(S) character replaces the TAB character).

The READ routine may be used in two modes for more flexible operation. The first mode involves accessing the READ routine from the COMMAND routine by entering via the teletype:

R CR LF

After reading the tape, control returns to the COMMAND routine.

The second mode of operation permits the user to use the CNC system as a standard N/C machine. In such a case, the teletype is not required, and the READ routine may be accessed by entering a starting address of 6400_8. In this mode, the computer stops with a halt code of 102077_8 when the routine is complete. Both modes of operation are possible with either tape format.

In reading the tape, data is converted into binary form and stored in a number of preselected memory locations. The dimension words are always stored in incremental form since this is the form in which the CNC routine requires the data. At present, the region of memory reserved for N/C data is limited to 255 (decimal) blocks. All of the N words are stored successively starting at address 30000_8. Similarly, the G words begin at 30400_8, X words at 31000_8, Y at 31400_8, Z at 32000_8, I at 32400_8, J at 33000_8, K at 33400_8, F at 34000_8 and M at 34400_8. (The appropriate value of PATH or RADIUS begins at 35000_8). The computer stops with a halt code of 102071_8 if a particular block exceeds the allowable length (i.e., vector length or radius = 3.2767 inches); pushing the RUN pushbutton causes an error message to be printed on the teletype. The computer then continues to read the tape. A halt code of 102070_8 will occur if the user attempts to read more than 255 blocks of N/C data. Upon encountering a / (SLASH) on the data tape, the reader stops.

While reading data. and calculating the appropriate value of PATH or RADIUS, use is made of the SQRT routine. (This routine is also used by the DATA EDITOR and DATA DIAGNOSTIC routines.) This routine calculates the square root of the sum of the squares of two numbers which enter the routine via the A and B registers. For accuracy, the subroutine chooses between two strategies, both of which use eight terms of the Taylor series expansion. The accuracy acheived limits the absolute error to unity.

The SQRT routine uses two parameters, U and V which enter via the A and B registers respectively. They are sorted in order to maintain U>V. Then the choice of strategy is made so as to insure good accuracy.

Strategy #1 (for $U \geq 5/3 V$)

$$\text{SQRT} = \sqrt{U^2+V^2} = U\sqrt{1+V^2/U^2} \approx U + \frac{V^2}{2U} - \frac{V^4}{8U^3} + \frac{V^6}{16U^5} - \frac{5}{128}\frac{V^8}{U^7} + \frac{7}{256}\frac{V^{10}}{U^9} - \frac{21}{1024}\frac{V^{12}}{U^{11}} + \frac{33}{2048}\frac{V^{14}}{U^{13}}$$

Strategy #2 (for $U < 5/3 V$)

$$\text{SQRT} = \sqrt{U^2+V^2} = \sqrt{(U-V)^2+2UV}$$

$$= \sqrt{2}\sqrt{\frac{(U-V)^2}{2} - U(U-V) + U^2}$$

$$= \sqrt{2U}\sqrt{U-(U-V)+\frac{(U-V)^2}{2U}} = \sqrt{2U}\sqrt{U-D+\frac{D^2}{2U}}$$

where $D = U - V$

$$\text{SQRT} = \sqrt{2}\,U\sqrt{1-\frac{H}{U}} \quad \text{where } H = D - \frac{D^2}{2U}$$

$$= \sqrt{2}\left[U - \frac{H}{2} - \frac{H^2}{8U} - \frac{H^3}{16U^2} - \frac{5}{128}\frac{H^4}{U^3} - \frac{7}{256}\frac{H^5}{U^4} - \frac{21}{1024}\frac{H^6}{U^5} - \frac{33}{2048}\frac{H^7}{U^6}\right]$$

The accuracy of the truncated expansion of $(1 + X)^{1/2}$ is dependent upon the assumption that $X \ll 1$. The use of two strategies simply maintains that this is a reasonable assumption.

2) PRINT AND PUNCH Routine

This routine produces a complete listing of the N/C data program currently in computer memory either on the teletype or on paper tape. The format of the resulting output, which may be either tab-sequential or word address, is compatible with the format used by the READ routine which is currently in memory. The PRINT AND PUNCH routine may be accessed in four distinct ways, and in each, the output may be either in the format of a teletype listing, or a paper tape produced by the high speed punch. To obtain the latter, in all four cases, the user must first enter 15_8 in the computer's switch register before accessing the routine. (When the tape is finished, the switch register should be cleared.)

In the first mode, access may be gained from the COMMAND routine by typing:

<u>P CR LF</u>

Upon completion, control returns to the COMMAND routine. The second mode of access is that of starting the computer at address 7000_8. Using this method, the computer halts upon completion with halt code 102077_8 in the display register. The third and fourth modes are the PRINT (P) and LIST (L) options respectively, which are available in the DATA EDITOR routine which is discussed in detail later. In using these two options, control returns to the DATA EDITOR routine upon completion.

The output of the routine in all cases other than the LIST option is all of the N/C data up to the first M code of 30. At the end of the data, a / (SLASH) is printed or punched. In addition, a leader and trailer are output.

The routine uses two parameters p and q which are in the A and B registers respectively at the beginning of the routine. The routine will print all N/C data from line p to line q inclusive, except when p = q = 0; this occurs in the first three modes of operation wherein all the N/C data is printed out. In the fourth mode (the LIST option), p and q are line numbers entered by the user under the supervision of DATA EDITOR. In this mode, the leader and trailer are omitted and output ceases with line q (or with an M code of 30, whichever occurs first).

3) DATA EDITOR Routine

This routine permits modifications and editing of the N/C data stored in core. DATA EDITOR provides communication between the part-programmer and the computer through the teletype which serves as an input/output device. The part-programmer can perform the following editing operations on the stored data:

a) Change the numerical data in any block.

b) Shift forward data blocks. This option is useful for inserting new blocks of data with aid of options (a) or (c).

c) Repeat a group of successive blocks of data.

d) Delete a series of successive blocks of data.

e) Place the correct sequential number (N) in all data blocks.

f) Multiply all feed rate codes in the N/C data by a factor.

g) Print out a selected group of data blocks.

h) Print a full listing of the N/C data; a new data tape can be punched simultaneously.

All of these modifications can be done either while cutting a part (or plotting the shape) or in a stand-alone mode (without cutting). While cutting, in addition to the data editing, the operator can change the actual cutting feedrate via the teletype (this data is not stored in memory).

The routine can be used in two modes:

1) in an interrupt mode in conjunction with the MAIN or A/C routine. It may be accessed from the COMMAND routine by entering N CR LF (allowing simultaneous execution of N/C program) or E CR LF .

2) as a stand-alone routine to correct N/C data. In this mode, the starting address is 4010_8.

In the first mode, the computer will be running on the main program until interrupted by the teletype. Depressing any key will cause an interrupt; but all inputs other than the character ! (EXCLAMATION MARK) are ignored. When ! is entered, the computer will disable the entire interrupt system and print START on the teletype. If the milling machine is cutting (or plotting), it will halt when START is printed.

In the second mode, starting the program causes the computer to print ! (EXCLAMATION MARK) on the teletype and then wait for input. Again, the first input must be !. All other inputs will cause the program to halt with 102077_8 in the display register. When ! is entered, the computer prints START on the teletype.

After printing START in either mode, the program then waits for input from the teletype. The options available are as follows:

a) N/C data change

N/C data change in any block by first entering N and the line number[6] followed by a COMMA. Then, any word in that data block may be altered by entering the appropriate address letter (G, X, Y, Z, I, J, K, F or M) followed by the new numerical data and a COMMA. The order in which the changes are made can be arbitrary. The option is terminated by CR and LF.

e.g. N128, G02, X-10965, M3, Y+50, CR LF

Should the changes entered exceed the allowable segment length, the computer will stop with a halt code of 102071_8. Pressing the RUN button will then provide an error message.

b) Shift

Shift forward the N/C data (by any number of blocks) by entering S followed by the line number to be shifted (p), a COMMA and the number of shifts (q) followed by a COMMA.

[6]The "line number" corresponds to the logical order of blocks in the computer memory (beginning with line number zero) whereas the "block number" is the N word found on the data tape. Since the programmer may use any arbitrary system of "block numbers", the "line numbers" represent the simplest method of manipulating the N/C data.

S p, q,

This causes line number p to be shifted to (p + q); line (p +1) to (p + 1 + q); etc. All the numerical data in lines p to (p + q - 1) inclusive, is erased. When the process is completed, the computer prints a PERIOD and terminates the option. If the last available memory location for N/C data contains data, an error message is printed out:

N256 ERASED

and that block is erased. For example, if q = 4, the message may appear four times as the last four blocks are erased.

c) Repeat

Repeat a series of data blocks after a stated line number. This is done by entering R followed by the first line number of the series to be repeated (p), a COMMA, the last line number of the series (q), a COMMA, the line number after which the series is to be repeated (r), and a final COMMA.

R p, q, r

where $p \leqslant q$

When the process is completed, a PERIOD is printed and the option is terminated, If the restriction $p \leqslant q$ is violated, then ? (QUESTION MARK) is printed by the computer and the option is terminated.

e.g. R 100, 99, ?

d) Delete

Delete a series of blocks of N/C data by entering D followed by the first line number to be deleted (p), a COMMA, the last line number of the series (q) and a COMMA.

D p, q,

When the process is completed, the computer prints a PERIOD and the option is terminated.

e) Arrange

Arrange the N numbers in sequential order from the beginning of the N/C data to the first block containing an M code of 30. This is done by entering A. When the process is completed, the computer prints a PERIOD.

A

f) Feedrate

Feedrate change in computer memory is accomplished by entering F then a factor (p) by which all feedrate codes are to be multiplied followed by a COMMA. The factor may be any decimal number between 9.9 and 0.1. When the process is completed, the computer prints a PERIOD.

F p,

All subsequent feedrate codes are checked to ensure that they fall within the limit $30 < f < 3000$. If too small, the code is automatically set to the minimum limit and if too large, the code is set to the maximum limit.

g) List

List a series of data blocks by entering L then the first line number of the series (p), a COMMA, the last line number of the series (q) and a COMMA.

L p, q,

Once the listing is printed, the option is terminated. Again, if $q < p$, then ? (QUESTION MARK) is printed out by the computer and the option is terminated. Note that the following instruction will result in the entire N/C data program being printed out:

L 0, 0,

h) Print or Punch

Print or Punch out all of the N/C data. To print out a listing of the N/C data, the user must enter P. To punch a data tape from the high speed punch, the user must first enter 15_8 in the switch register before typing P. (The switch register should be cleared when the tape is finished.) A leader and trailer will be produced in both printing and punching.

P

i) End

End the DATA EDITOR routine by entering E. If in mode 1, this command returns computer control to the main routine and enables the interrupt system. If the milling machine was previously cutting, it will begin to cut again from the point at which it stopped. If in mode 2, the computer halts with code 102077_8 in the display register.

E

j) Feedrate correction

Feedrate correction in the current operation can be performed while the milling machine is cutting (or while plotting). This is done by entering " (QUOTATION MARK) and the feedrate code (f) followed by a COMMA, where f is a positive integer less than 3000. After entering the COMMA, the computer checks that the code is less than 3000. If not, the computer prints ? (QUESTION MARK) and the option is terminated. If the feedrate code is acceptable, the computer prints E and either returns to the main routine in mode 1 or halts with 102077_8 in the display register in mode 2. (Of course, this option is useful only in mode 1 while cutting or plotting.)

" f, E

k) Return to COMMAND routine

Return to COMMAND routine can be performed whenever the computer is not already executing another option by entering ↑.

↑ COMMAND

Notes:

1) the program allows the use of any number of options successively until option (i), (j) or (k) are used.

2) Inserting a block or group of blocks of N/C data is done by using two successive options. First the SHIFT option (b) (with as many shifts as blocks to be inserted) is used, followed by either option (a) or (c) to place the desired data in the correct blocks.

e.g. S 180, 2

R 77, 78, 180, .

or S 180, 1 .

N 180, G 1, X +2000 CR LF

3) Any deviations from the described format will result in ? (QUESTION MARK) being printed. This terminates the option. This is useful if an error is made in entering a command.

e.g. N 321 Q ?

Here, 321 is too large, and the error was noticed by the user. Thus, by entering any non-digit (except a comma), the user violates the required format, causing the computer to print ? (QUESTION MARK) and terminate the option. The characters SPACE, CARRIAGE RETURN, and LINE FEED are always ignored except in option (a) wherein LINE FEED terminates the option (and initiates the calculation of a new value of PATH or RADIUS).

e.g. N 128 X ? - COMMA after 128 is absent.

N 128 , Q ? - Q is not an appropriate address letter.

R 121 . ? - The number must be terminated by a COMMA.

Z ? - Z can be entered as an address letter only when already in option (a).

4) DATA EDITOR can be used to write a complete N/C data program by successively using option (a) to enter data. It is advisable to first clear the region of memory reserved for N/C data by

reading in a simple N/C tape containing only one block of information. (The READ routine first clears the entire N/C data region.)

4) DATA DIAGNOSTIC Routine

This routine is used to debug N/C data in computer memory. It checks for illegal G or M codes and for feedrate codes either above or below set limits. The routine also computes the final radius of each circular segment and compares it with the initial radius. Both the final radius and the difference between the two radii are printed on the teletype. This helps the part programmer to discover a very common programming error (i.e., that the initial and final points do not lie on the same circle). Finally the sums of the X, Y and Z motions are printed out on the teletype. This shows the part programmer what motions are required to return the tool to its initial point in preparation for machining another part, having just completed one.

The DATA DIAGNOSTIC routine is accessed from the COMMAND routine by entering:

D CR LF

No further input is required of the user to initiate the routine which checks the stored N/C data program block by block. In each block, the data is checked as follows:

1) The feedrate code

The feedrate code is checked to ensure that the restriction:

$$30 < F < 3000$$

is not violated. (This is compatible with feedrate limitations of 0.3 and 30 ipm.) If it is violated, the computer will print out the block number (p) and the feed word in that block (q):

N p, F q,

The computer then waits for input, allowing the part programmer to change the feed word. A new value is inserted by entering a number followed by CR and LF. The computer stores the new value and continues to check the data. If no change is desired, the user needs to enter only CR and LF. The feed word is tnen unaltered.

2) The G word

The G word may be one of the following: 0, 1, 2, 3, 4, 9, 17, 18, or 19. Any other G word will result in the block number (p) and the G word (q) being printed on tne teletype:

N p, G q,

Once again, the user may insert a new G word or leave it unaltered as in the feedrate code check. CR and LF let tne computer continue checking the data.

3) The M code

The M code may be one of the following: 3, 5, 6, or 30. Any other M code is illegal and will result in the block number (p) and the illegal M code (q) being printed on the teletype:

N p, M q,

As in the previous two checks, the part programmer may alter the M code, or leave it unaltered. Cr and LF cause the computer to continue with the checking procedure.

4) Each circular segment

For each circular segment, the final radius is checked. There are eight possible circular segments which emanate from a point, in any given plane. But from the point of view of interpolation as well as final radius, the eight possibilities are reduced to two only. Using Boolean algebra, the decision logic for each one is easily seen. Define a clockwise rotation (G = 2) as A and a counter-clockwise rotation (G = 3) as A'. Similarly, define equal signs in the distance dimension words as B and unequal signs as B'. The Boolean conditions for the two cases are as follows:

$$\text{CASE } 1 = A \cdot B + A' \cdot B'$$

$$\text{CASE } 2 = A \cdot B' + A' \cdot B$$

In the following text only X, Y, I and J are used. However, since any two coordinate motions can be programmed simultaneously, Z may be substituted for X or Y and, similiarly, K for I or J. It can be proven that the equation for the final radius of a circular segment of CASE 1 is as follows:

EQUATION 1:

$$\text{FINAL RADIUS (FR)} = \sqrt{(I - X)^2 + (J + Y)^2}$$

Similarly, for CASE 2, the equation is:

EQUATION 2:

$$\text{FINAL RADIUS (FR)} = \sqrt{(I + X)^2 + (J - Y)^2}$$

In both equations, the square root is calculated by the SQRT routine which was discussed in the READ routine. After this final radius is calculated, it is compared with the initial radius (calculated and stored during reading of the data tape):

$$\text{INITIAL RADIUS (IR)} = \sqrt{I^2 + J^2}$$

Then, the block number (p), the final radius (q) and the amount by which the final radius exceeds the initial radius (r) are printed:

N p, RADIUS = q, DIFF = r

The computer does not stop for input, but continues to check the data. (A difference of 1 unit may occur, even for correct data, because of truncation errors within the computer.)

5. Difference between initial and final points

The difference between the initial and final points in the data (actually the sum of all incremental X, Y and Z motions) is printed out when an M code of 30 is encountered.

SUM OF X = p

SUM OF Y = q

SUM OF Z = r

Control is then returned to the COMMAND routine.

5) PDP-8 LINK Routine

This routine enables data transmission between the HP-2100 and PDP-8 minicomputers at a rate of 110 baud. This allows the use of the APT language with this CNC installation. The APT compiler and the CNC postprocessor require memory far beyond the capabilities of a minicomputer; they are run on the university's CDC-6400 computer. The PDP-8 has the capability of communicating with the CDC-6400 (at a rate of 110 baud at present). Thus, the post-processed result of an APT job performed on the CDC-6400, may be written on to cassette or paper tape at the PDP-8 site. Once this task has been completed, the user may disconnect the CDC-6400 link, and, using the same hardware, establish a link with the HP-2100 minicomputer. The same software- the EDIT operating system- is used by the PDP-8 whether communicating with the CDC-6400 or HP-2100. Then, the user must access the PDP-8 LINK routine (on the HP-2100) from the COMMAND routine by typing:

L CR LF

The routine then requests an input from the user:

ENTER R S OR ↑

This allows the user to select the desired option where R will initiate the receive mode (to receive data from the PDP-8), S will initiate the send mode, and ↑ will return control to the COMMAND routine. In receiving or sending data, recognition of the character # (POUND) will return control to the COMMAND routine, and recognition of * (ASTERISK) will switch to the opposite direction of transmission. At present, all data sent to the PDP-8 must originate at the teletype (or the low speed reader). Data received from the PDP-8 may be directed to punched tape or the teletype by the use of character recognition:

% (PERCENT) initiates data output on high speed punch,

and & (AMPERSAND) terminates punching.

Otherwise, all data received is printed on the teletype.

Thus, N/C data written on a cassette at the PDP-8 site may be transferred directly to paper tape via the HP-2100 high speed punch. Then, all of the N/C data service routines are available for use with the APT-generated tape.

6) CUTTER COMPENSATION Routine

The CUTTER COMPENSATION routine calculates a new cutter path in order to accomodate a cutter of slightly different diameter than was originally intended. It operates on N/C data stored in computer memory, using two operator-supplied inputs. The inputs are responses to two requests:

ENTER CUTTER COMPENSATION IN INCHES

and

ENTER +1 FOR LEFT AND -1 FOR RIGHT

The response to the first command must be the signed difference between the radii of the desired cutter and the original cutter. The response to the second command indicates to the algorithm whether the part will be to the left or right of the cutter, when looking in the direction of the cutter motion.

The CUTTER COMPENSATION routine is accessible from the COMMAND routine by entering:

C CR LF

Upon completion, control is returned to the COMMAND routine. If, due to the compensation calculations, a block exceeds the allowable limit for length or radius, the computer stops with a halt code of 102071_8. Pressing the RUN pushbutton causes printing of an error message, and the calculations are then resumed.

The algorithm uses trigonometry at the intersection points of all blocks in order to calculate the changes required in each block to accomodate the new cutter size. Slightly different equations must be used to determine the compensation required in the four possible cases:

a) Line to Line

b) Line to Circle

c) Circle to Line

d) Circle to Circle

In each of the cases, the algorithm uses the N/C data for the first block and the second block of each intersection pair; in the equations that follow, the data from the first block will have subscript 1 and the second, 2. (Note that, in general, X_1, Y_1 and J_1 may have already been altered by the algorithm at the previous intersection point.)

a) Line to Line

$$Q = \frac{Y_2 X_1 - Y_1 X_2}{P_1 P_2 + X_1 X_2 + Y_1 Y_2}$$

- where P_1 and P_2 are the values of PATH for blocks 1 and 2.

$$\Delta X_1 = C\,Q\,X_1 / P_1$$
$$\Delta Y_1 = C\,Q\,Y_1 / P_1$$
$$\Delta X_2 = C\,Q\,X_2 / P_2$$
$$\Delta Y_2 = C\,Q\,Y_2 / P_2$$

- where ΔX_1, ΔY_1, ΔX_2 and ΔY_2 are the changes in the X and Y words of blocks 1 and 2, and C is the value of the compensation.

b) Line to Circle

$$Q = \frac{I_2 X_1 + J_2 Y_1}{R_2 P_1 - J_2 X_1 + I_2 Y_1}$$

- where R_2 is the radius of the circular segment of block 2.

$$\Delta X_1 = C\,Q\,X_1 / P_1$$

$$\Delta Y_1 = C\,Q\,Y_1 / P_1$$

$$\Delta X_2 = C\,(Q\,J_2 + I_2) / R_2$$

$$\Delta Y_2 = C\,(Q\,I_2 - J_2) / R_2$$

$$\Delta I_2 = C\,(+Q\,J_2 + I_2) / R_2$$

$$\Delta J_2 = C\,(-Q\,I_2 + J_2) / R_2$$

c) Circle to Line

$$Q = \frac{-V_1 Y_2 - X_2 U_1}{R_1 P_1 - V_1 X_2 + U_1 Y_2}$$

– where $V_1 = Y_1 + J_1$,

$U_1 = X_1 + I_1$

and R_1 is radius of circular block 1.

$$\Delta X_1 = C\,(-Q\,V_1 + U_1) / R_1$$

$$\Delta Y_1 = C\,(+Q\,U_1 + V_1) / R_1$$

$$\Delta I_1 = 0$$

$$\Delta J_1 = 0$$

$$\Delta X_2 = C\,Q\,X_2 / P_2$$

$$\Delta Y_2 = C\,Q\,Y_2 / P_2$$

d) Circle to Circle

$$Q = \frac{U_1 J_2 - I_2 V_1}{R_1 R_2 + V_1 J_2 + U_1 I_2}$$

$$\Delta X_1 = C\,(-Q\,V_1 + U_1) / R_1$$

$$\Delta Y_1 = C\,(Q\,U_1 + V_1) / R_1$$

$$\Delta I_1 = 0$$

$$\Delta J_1 = 0$$
$$\Delta X_2 = C (-Q J_2 - I_2) / R_2$$
$$\Delta Y_2 = C (+Q I_2 - J_2) / R_2$$
$$\Delta I_2 = C (+Q J_2 + I_2) / R_2$$
$$\Delta J_2 = C (-Q I_2 + J_2) / R_2$$

The part programmer must program an empty block (plus a feed word) as the zero block. This enables the algorithm to store an initial compensation value, allowing the operator to begin cutting from the same initial point, regardless of cutter diameter. The algorithm also creates a new final block, to allow the cutter to return to the initial point. The M code of 30 in the previous block is erased and is inserted in the new block. Therefore, the part programmer must ensure that his program does not completely fill up available memory; sufficient space must remain for the algorithm to create a last block.

The CUTTER COMPENSATION routine has three main limitations:

1) Thus far, the algorithm has been developed for a two-dimensional part in the X-Y plane only. The extension to a three-dimensional part should not be difficult; this CNC system is such that only two axes move simultaneously. (Thus, even a complete cutter compensation algorithm would involve switchable planes of compensation.) It must also be remembered that the algorithm is one of radial cutter compensation, not of length compensation. This greatly simplifies compensation with Z-axis motions. For an end mill, therefore, compensation would be required only in circular interpolation in the X-Z or

Y-Z plane where a block of linear interpolation of length equal to the cutter diameter is required at the apex of the circle. (This is discussed in the programming manual.) A ball end mill, however, would involve a slightly more complex algorithm since its radius must also be considered in programming in the Z-direction.

2) The second limitation is that the algorithm assumes that extra blocks have been inserted wherever compensation would be impossible otherwise. An example is shown in Figure 1 The example indicates that positive cutter compensation (i.e., a larger diameter cutter) would require an extra block of data. If manually prepared, the N/C data would need a block of zero distance to allow the compensation routine to make the proper correction at this point. Data generated by the APT programming language would already include an extra block.

3) The third limitation is that of the inherent inaccuracy of the algorithm; in compensation involving circular segments, the tangent to the circle at the point of intersection is used in the calculation rather than the circle itself. This simplifies calculations considerably. If the compensation is considerably less than the radius of the part being cut (e.g. 1/50 of the part radius) then this assumption causes no errors. (Thus, for a part radius of 0.500 inches, a radial compensation of 0.01 inches is allowed. This corresponds to a

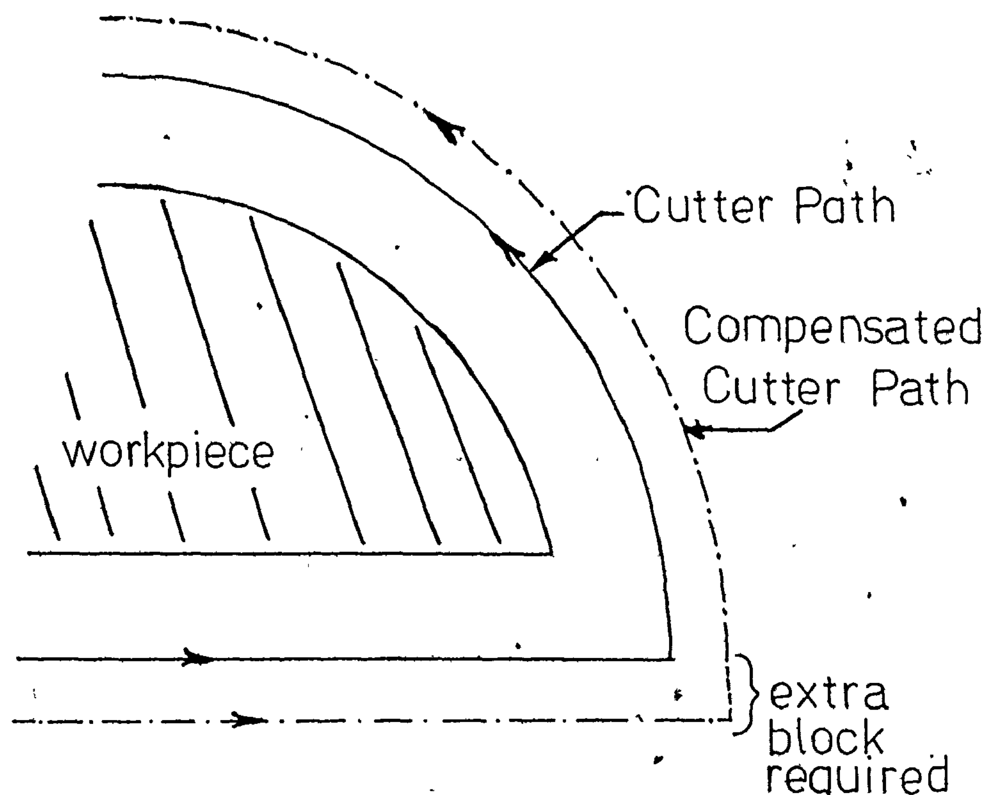


Figure 1

Extra Block for Cutter Compensation

tool 0.02 inches larger or smaller than the original cutter.) With decreasing part accuracy, larger compensations may be used.

A/C Auxiliary Routines

1) INITIALIZATION Routine

This routine is used to determine the "optimum" force and torque values to be used in the adaptive control mode of milling. This routine may be accessed from the COMMAND routine by typing:

I CR LF

The computer then requests user input from the teletype by printing:

INITIAL DATA :

CUTTER DIAM = X.XXXX CR LF

LENGTH = X.XXXX CR LF

SPEED = XXXX. CR LF

At each equal sign, the computer waits for input which is entered in the indicated format. Input units are inches and revolutions per minute. (A maximum of five digits is used in the calculations.) Input is terminated by CR (CARRIAGE RETURN) and LF (LINE FEED). A RUBOUT character entered during input will cause the repetition of that input sequence and the entry of a non-digit (other than PERIOD, CR and LF) will cause the computer to output ? (QUESTION MARK) and recycle to the beginning of the INITIALIZATION routine.

The formulae used for calculating the "optimal" force and torque are based on the assumption that a right circular cylinder may be a crude approximation to the shape of an end mill[16]. The equations used are:

$$F_{max} = \frac{\sigma_{ys} \cdot \pi \cdot D^3}{L \cdot 32} \quad \text{(lb)}$$

$$T_{max} = \frac{\pi \cdot \tau_{max} \cdot D^3}{16 \cdot 12} \quad \text{(ft-lb)}$$

where σ_{ys} = yield strength of high speed steel.

τ_{max} = shear strength of high speed steel.

D = diameter of cutter.

L = length of cutter.

Also, two constraint equations exist governing the maximum force and torque which the milling machine can apply:

$$F_{MACH} = 2560 \quad \text{(lb)}$$

$$T_{MACH} = \frac{P \cdot 3.3 \cdot 10^4}{SPEED \cdot 2\pi} \quad \text{(ft-lb)}$$

where P = power of spindle motor.

SPEED = spindle r.p.m.

Using the following values:

$$\sigma_{ys} = 1.2 \cdot 10^5 \quad \text{lb/in}^2$$

$$\tau_{max} = 8.0 \cdot 10^4 \quad \text{lb/in}^2$$

$$P = 6.5 \text{ hp.},$$

the four equations become:

$$F_{max} = 1.178 \cdot 10^4 \; \frac{D^3}{L}$$

$$T_{max} = 1.309 \cdot 10^3 \cdot D^3$$

$$F_{MACH} = 2560$$

$$T_{MACH} = \frac{3.41 \cdot 10^4}{SPEED}$$

The routine then calculates these four values and prints out the smaller of the two calculated forces as the "optimum" force, and similarly for the "optimum" torque. However, in the adaptive control mode, these two quantities may be exceeded for short periods of time during which the tool may be broken. Thus, for use in the A/C routine, a safety factor of 1.25 is used (i.e., 80% of the calculated "optimum"). This value is then multiplied by the A/C transducer calibration factor (0.35), and the final values of FOPT and TOPT are stored for the A/C routine's use, in memory locations 20001_8 and 20003_8 respectively.

Upon completion, the routine returns control to the COMMAND routine.

2) TRANSDUCER CALIBRATION Routine

This routine prints out the three channels of the Analog to Digital Processor (ADP). Thus, it is used not only in calibration of the transducer, but also in checking to ensure that the strain guage bridges are balanced. It is accessed from the COMMAND routine by the code:

T CR LF

It immediately prints the values of the three channels and stops, displaying a halt code of 102077_8. Pressing the computer RUN button will repeat the cycle. While calibrating the transducer, the user may wish to print continuously. This may be done by changing the contents of memory location 11555_8 from 102077_8 to 006400_8. Upon completion of the routine, there is no return to the COMMAND routine.

Chapter 3

A/C Hardware

The adaptive control transducer is basically an instrumented tool holder which is bolted directly to the spindle of the milling machine. The milling cutter is placed in a collet which is then drawn into the transducer, thereby securing the cutter. The structure of the A/C transducer is seen in figure 2 while the individual parts are shown in figure 3.

Structurally, the transducer consists of two planes; the upper plane A contains a flexure plate which is radially rigid so as to withstand radial cutting forces, yet flexible in torsion about the cutter axis and in angular tilt in a plane perpendicular to this axis. The radial stiffness is obtained by four relatively thick spokes which are the only connection between the inner bolt circle (connected to the tool-holder) and the outer bolt circle (attached to the spindle). While the spokes are thick in a direction parallel to the cutter axis, they are quite thin at their extremities in the plane of the flexure plate; this is responsible for the torsional flexibility of the flexure plate. The angular flexibility is achieved by two slots parallel to plane A which are visible in figure 2. They create a simple universal joint between plane A and the cutter axis. The flexure plate was machined from A-2 tool steel and subsequently hardened.

The tool-holder which is bolted to the flexure plate holds the collet, and in it, the milling cutter. The inner surface of the tool-

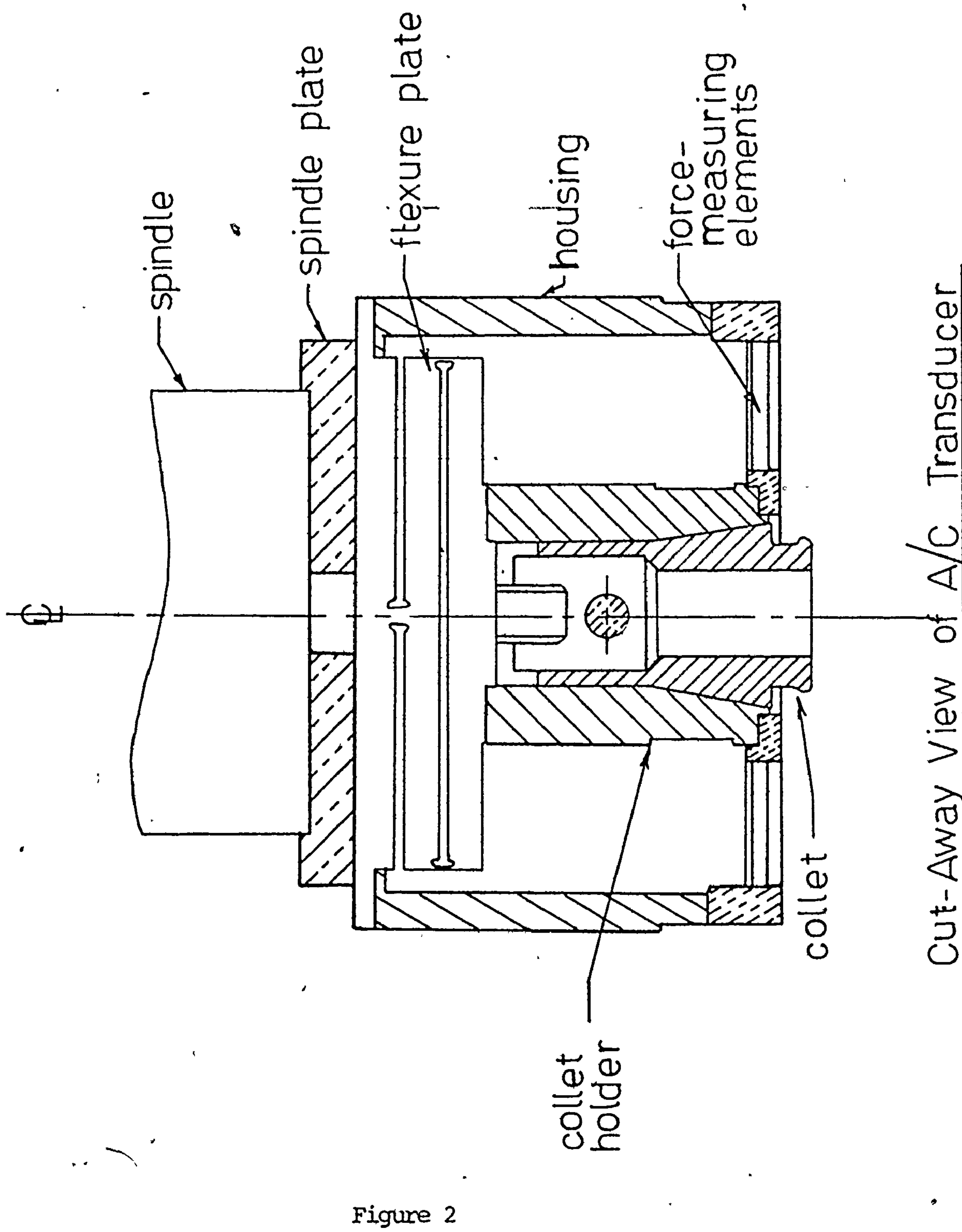

spindle
spindle plate
flexure plate
housing
force-measuring elements
collet
collet holder
Cut-Away View of A/C Transducer

Figure 2

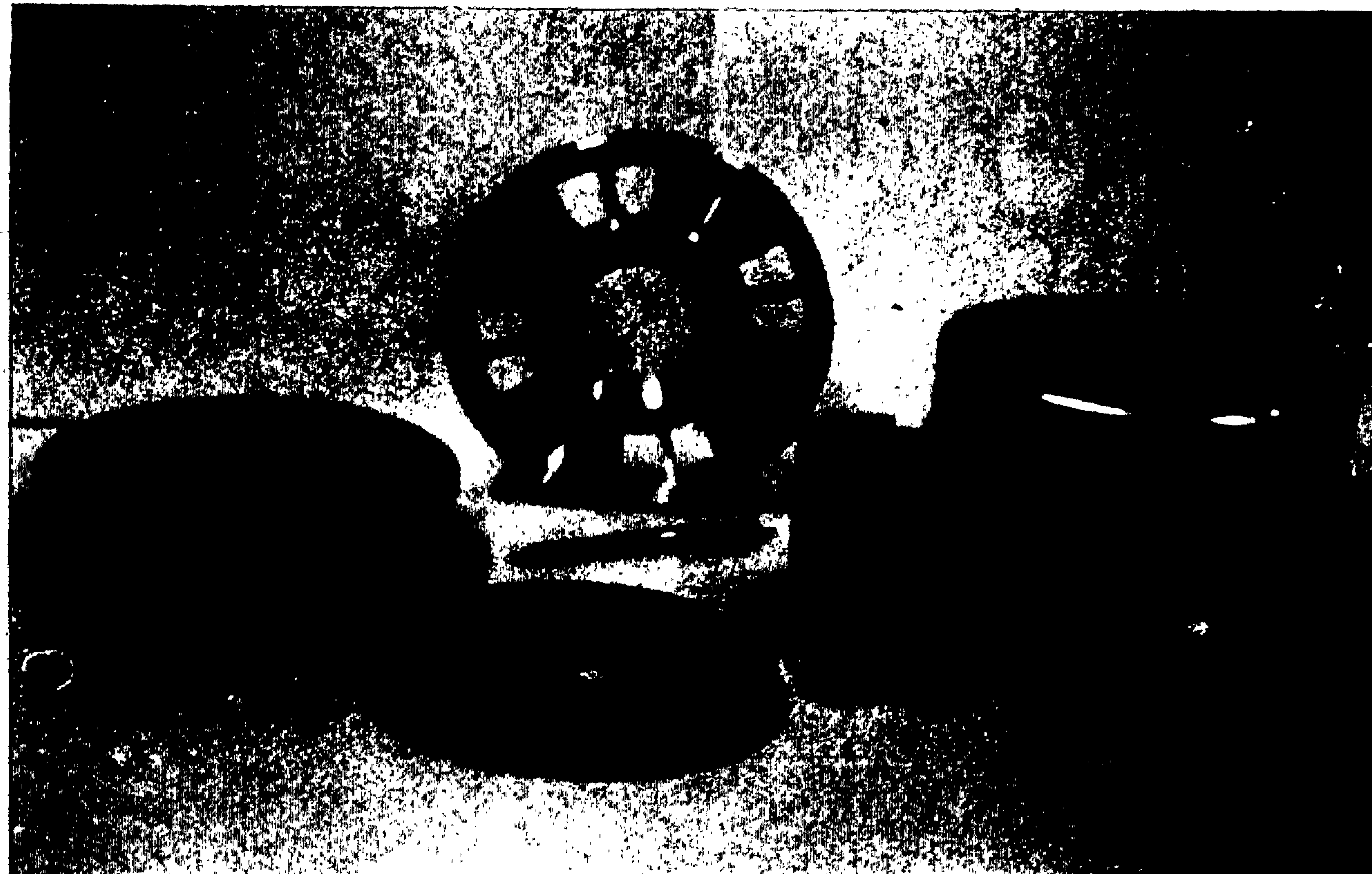


Figure 9A

[illegible] Transducer [illegible]

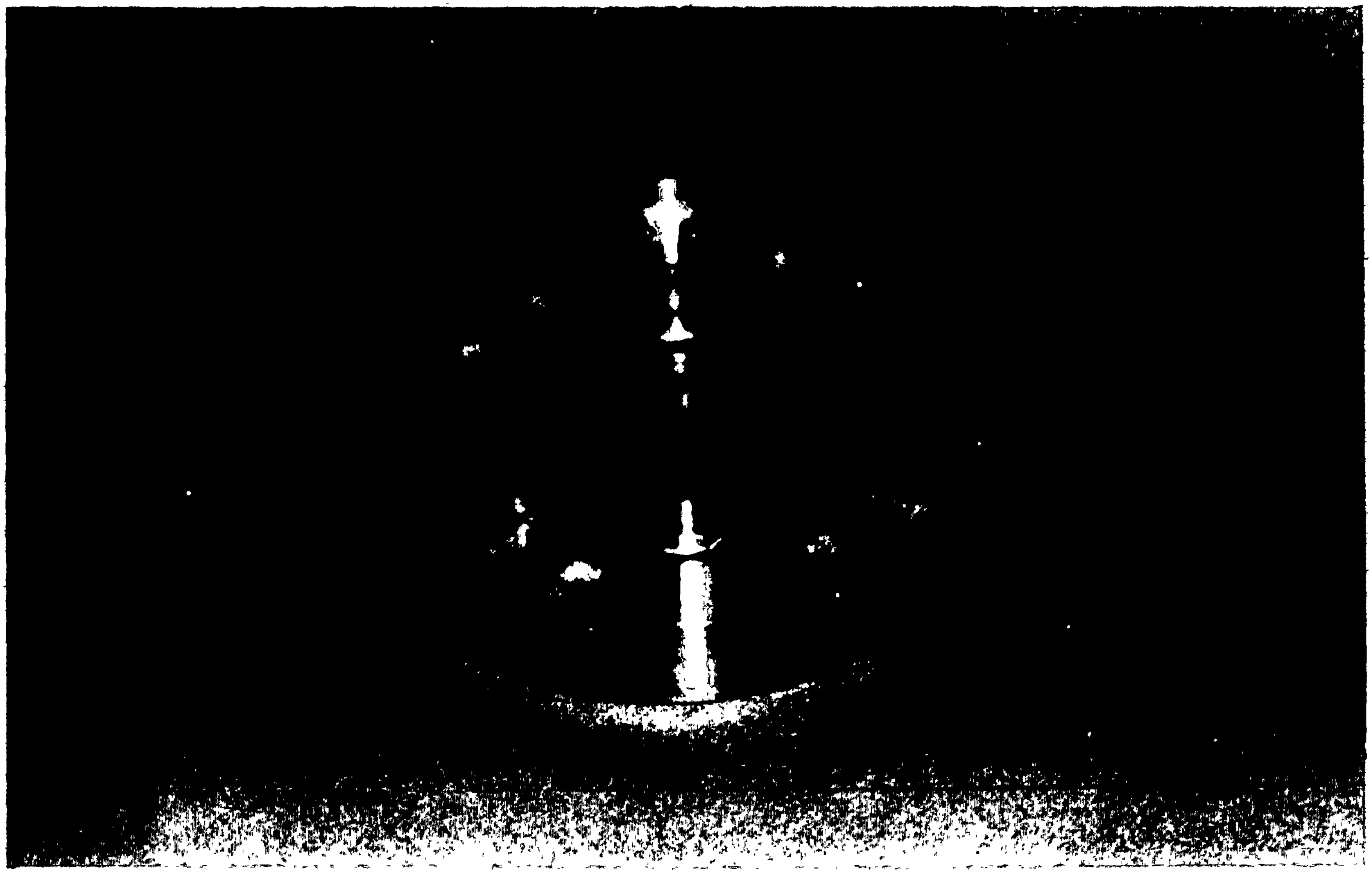


Figure 3b
A [illegible]

holder has been ground to match the outer surface of the collet. The tool is secured by drawing the collet up into the collet holder and thereby compressing the collet. The tool-holder is also of hardened A-2 tool steel.

The housing, made of aluminum, connects the outer rim of the flexure plate with outer rim of the baseplate in plane B. It also serves as a protective cover over the strain gauges within.

The measurement of force and torque occurs in plane B. The tool-holder is bolted to the inner ring of the baseplate (which may be seen in figure 3). The tool is therefore connected to the spindle only through the spokes of the flexure plate and the spokes of the baseplate which have been instrumented with thirty-two strain gauges. The strain gauges used were MICRO-MEASUREMENTS Ltd. type EA-13-250BG-120. The bond used was M-Bond AE-10. The baseplate was milled from 7075-T6 aluminum on the CNC machine according to an N/C data program produced with the help of the software discussed in chapter two. The baseplate, consisting of four spokes joining the inner and outer bolt circles, has been designed to withstand radial cutting forces of 5000 lb. and torques of 400 in-lb.

The baseplate houses three separate strain gauge bridges - one for torque containing sixteen gauges, and two for force containing eight gauges each. The two force bridges measure forces F_1 and F_2 in two perpendicular directions in plane B. Gauges on opposite sides of a spoke were placed in opposite ends of the force bridge so as to minimize the effect of torque in the bridge. The opposite strategy was employed in the torque bridge to minimize the effect of radial force.

Voltage is supplied to the bridges by a 10 volt D.C. power supply located in the Analog to Digital Processor (ADP). The strain gauges in the transducer are connected to a series of eight slip rings mounted on a nylon shaft supported by two ball bearings, at the top of the spindle. To minimize contact resistance and improve the signal to noise ratio, silver slip rings were chosen to mate with silver-graphite brushes (two brushes per ring). Connection between the slip ring shaft and the spindle is affected by a bellows coupling so as to tolerate some shaft misalignment.

In the ADP are potentiometers for balancing the three bridges and for adjusting the gain used in amplifying the force and torque signals. Figure 4 is a circuit diagram of this. The analog signals are digitized in the ADP before being supplied to the minicomputer. The ADP[17] has a 10-bit A/D converter for each of the three channels, a digital scanner, a code generator and a status generator. A/D conversion begins with a computer-supplied conversion command. Upon completion, the status generator sets a flag on a computer I/O board in slot 12_8 to inform the A/C routine that the digitized information is available. The digitized input is a 13-bit word composed of 10 bits from the A/D converter and 3 bits from the code generator identifying the channel which supplied the data. (The digital scanner, incremented by the conversion command, selects only one channel for input per software cycle.) The input word is then de-packed by software and the data is stored in the appropriate location according to the channel code.

In designing a transducer to measure force and torque in milling, two rather conflicting qualities were required; the transducer had to be sufficiently strong to withstand rather high cutting forces and stiff to avoid chatter while milling. But, at the same time, the transducer had to

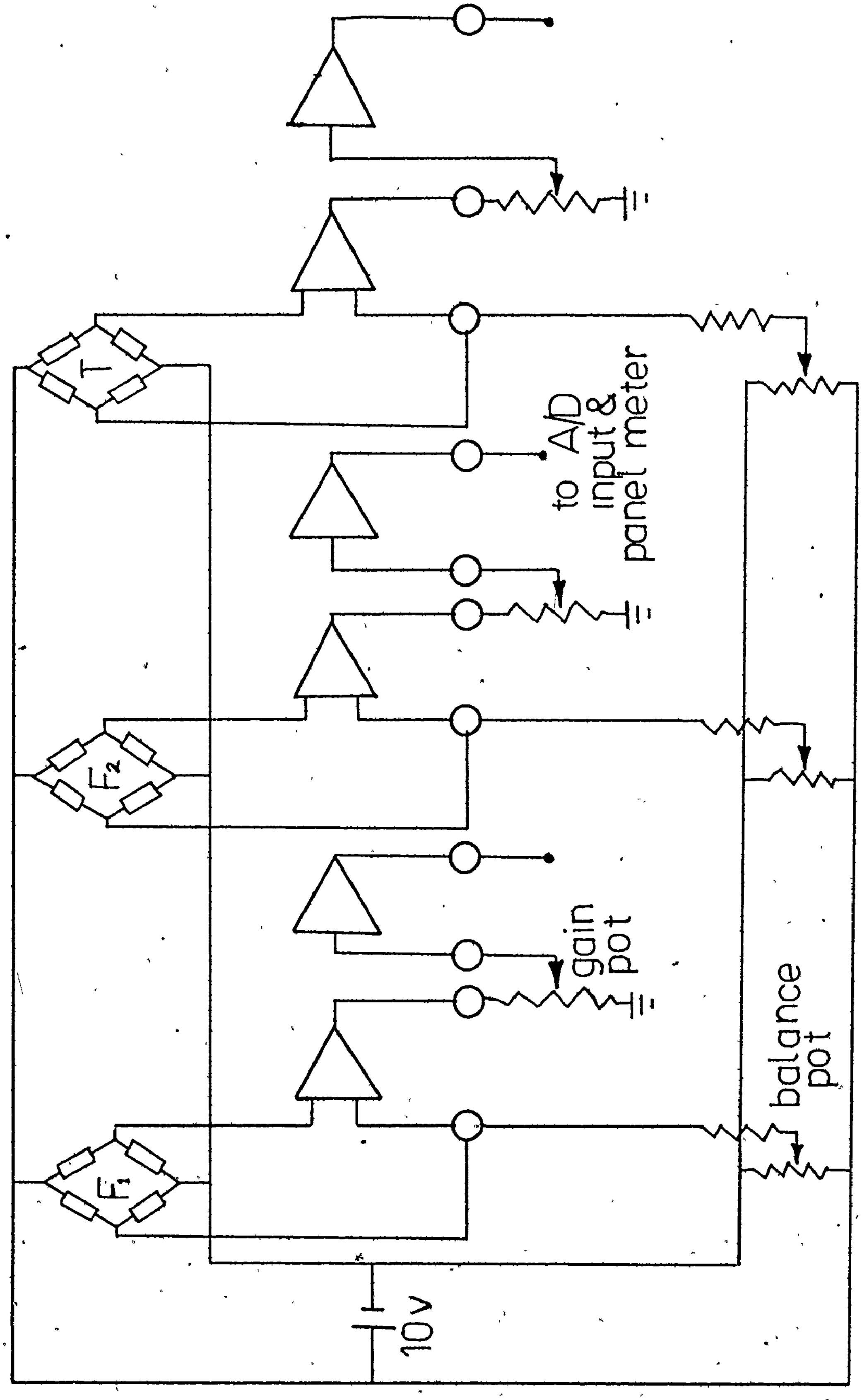


Figure 4

Bridges and Amplification in ADP

be sensitive to the small forces which act on a long and slender end mill. (In this transducer, the sensitivity depends largely upon the flexibility of the spokes in plane B.)

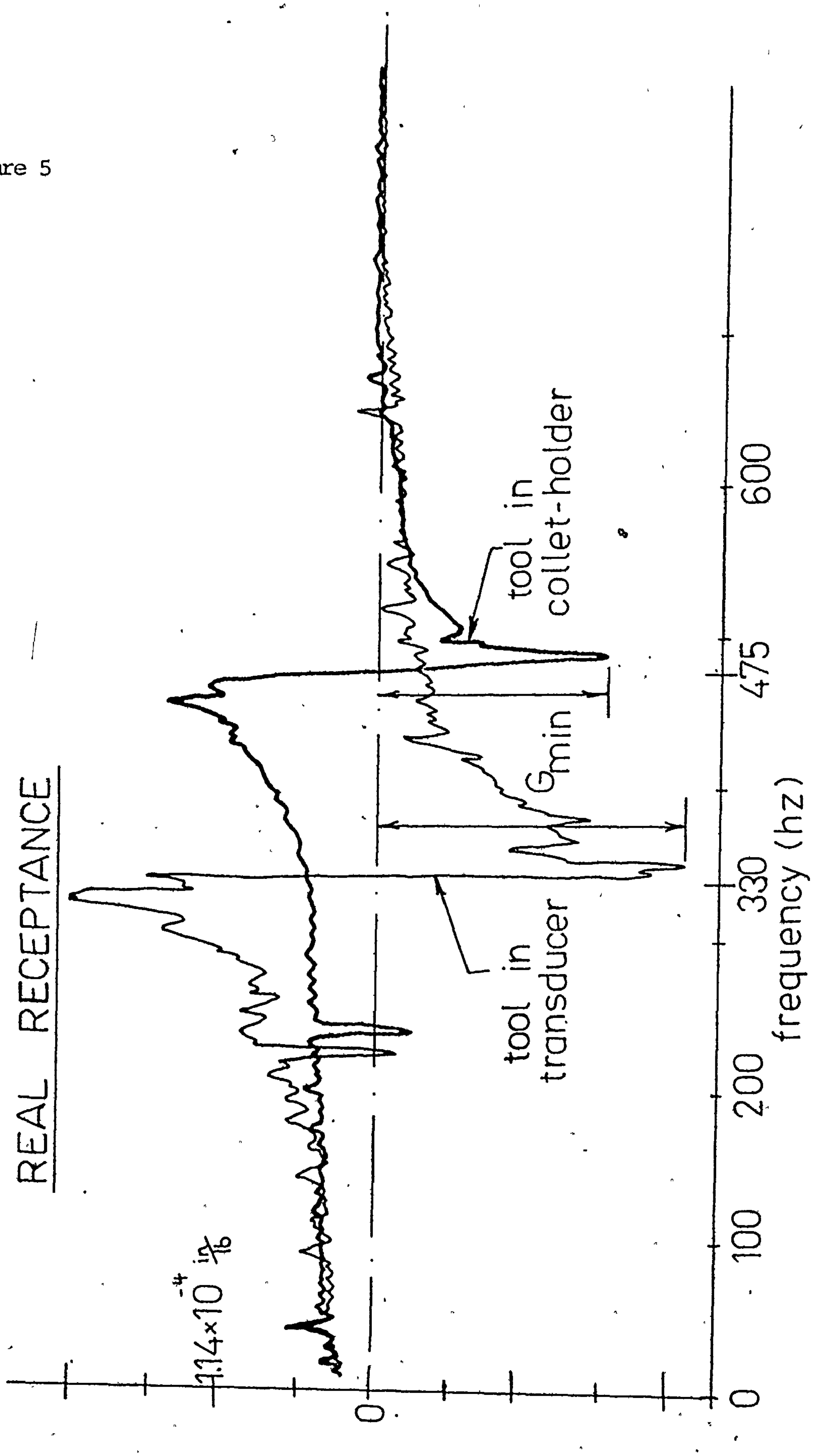
Therefore, to show that the final form of the transducer was sufficiently stiff, the dynamic characteristics of the transducer were analyzed using the method of shock excitation with a Fourier Analyzer. In this test, the tool was first clamped (on the milling machine) in the collet holder normally used in N/C milling. Then the same tool was clamped in the A/C transducer with the same overhang. A 5/8 inch diameter bar was placed in the collet with an overhang of 3 inches to simulate an end mill. Attached to the end of the bar was a small (0.4 lb) weight, bringing the total overhang to 3.5 inches. (The small weight was added to the bar in an effort to reduce the frequency of the mode in order to improve the accuracy of the Fourier analysis. Adding the weight, of course, does not alter the stiffness which was of central interest in the test.)

Figure 5 shows a plot of the real part of the receptance (in the frequency domain) generated by hitting the end of the simulated tool with a force transducer and measuring the vibration at that point with a capacitive probe. The figure shows the same result obtained with both the transducer and the conventional collet holder. Assuming that the bar is clamped in a rigid support, and that all of the mass of the block plus about 1/3 of the mass of the bar are concentrated in the block,

$$f_n = \frac{1}{2\pi}\sqrt{\frac{K}{M}}$$

$$\doteq 443 \text{ Hz}$$

Figure 5


REAL RECEPTANCE
1.14×10^-4 in/lb
0
tool in transducer
tool in collet-holder
G min
0
100
200
330
475
600
frequency (hz)

where: K = static stiffness

and M = mass.

This is rather close to the actual 475 Hz observed with the collet holder. This tends to indicate that this mode seen in figure 5 is mainly that of the overhanging tool. Nonetheless, it is obvious from the figure that the collet holder is stiffer than the transducer. The natural frequency of the transducer is about 1.4 times lower than that of the collet holder. Considering the equation relating the limit width of cut for chatter to the minimum of the real receptance

$$b_{lim} = \frac{-1}{2 R G_{lim}} \qquad [21]$$

where: R is the cutting force coefficient

and G_{lim} is the minimum real receptance

it can be seen that b_{lim} would be smaller by a factor of only 1.3 when using the transducer.

The next part of the test involved both hitting and measuring at the collet holder (and the transducer). In doing so, it was found that the G_{lim} value of the transducer was about 2.5 times larger than that of collet holder. Therefore, approaching a zero tool length, there would be more likelihood of chatter with the transducer. However, the overall scale of the resulting plot had to be increased by a factor of 250 to obtain any reasonable trace. Therefore, it seems that the above-mentioned effect is not very important. The conclusion is then that chatter is probably mainly due to the tool itself, and that the addition of the transducer to the milling machine makes it only slightly more susceptible to chatter.

The sensitivity of the transducer is indicated by the calibration curves included as figures 6, 7 and 8. The figures show the relation between the applied force or torque and the digitized computer input for each channel. The slope of the curves may be increased by increasing the amplifier gain. The 10-bit A/D converters operate on ± 10 volts, and thus, their output range is ± 511 units. Since an input in excess of 10 volts saturates the A/D converters, the ADP could supply force data well below the actual cutting force. Therefore, to avoid this, the amplifier gain was set so as to saturate the A/D converters only at a force of 1500 lb. on each of F_1 and F_2. This corresponds to the noted potentiometer settings of 1.5 for F_1 and 1.0 for F_2. To obtain the torque sensitivity shown in figure 8, maximum amplification was required with a potentiometer setting of 10.0. However, the noise caused by spindle rotation, also magnified by this high gain, effectively obliterates the true torque signal. Therefore, in the work that follows, the torque signal was not used. To use this signal, a preamplifier would be required within the transducer itself.

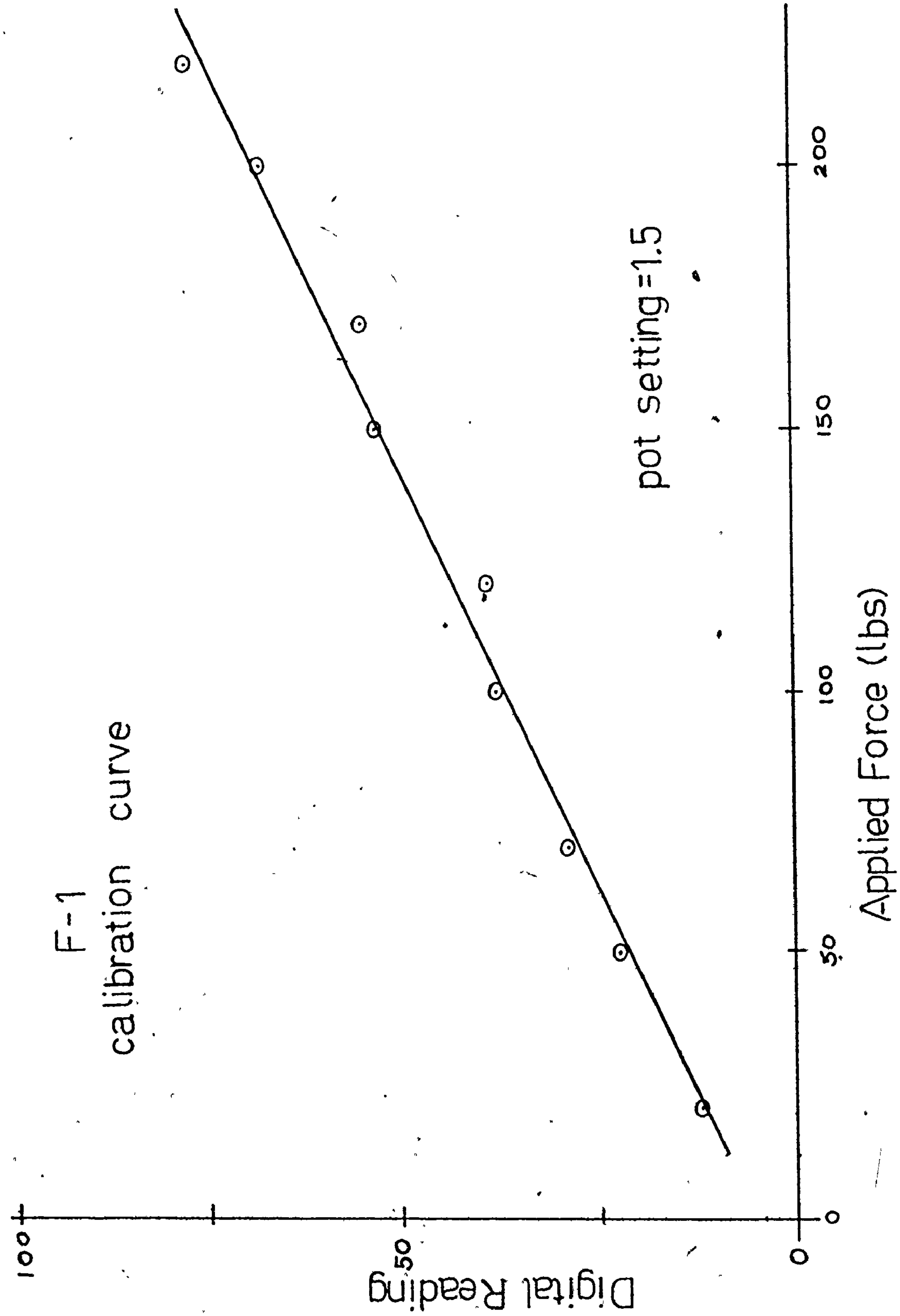

F-1
calibration curve
pot setting=1.5
Applied Force (lbs)
Digital Reading
0
50
100
150
200
0
50
100

Figure 6

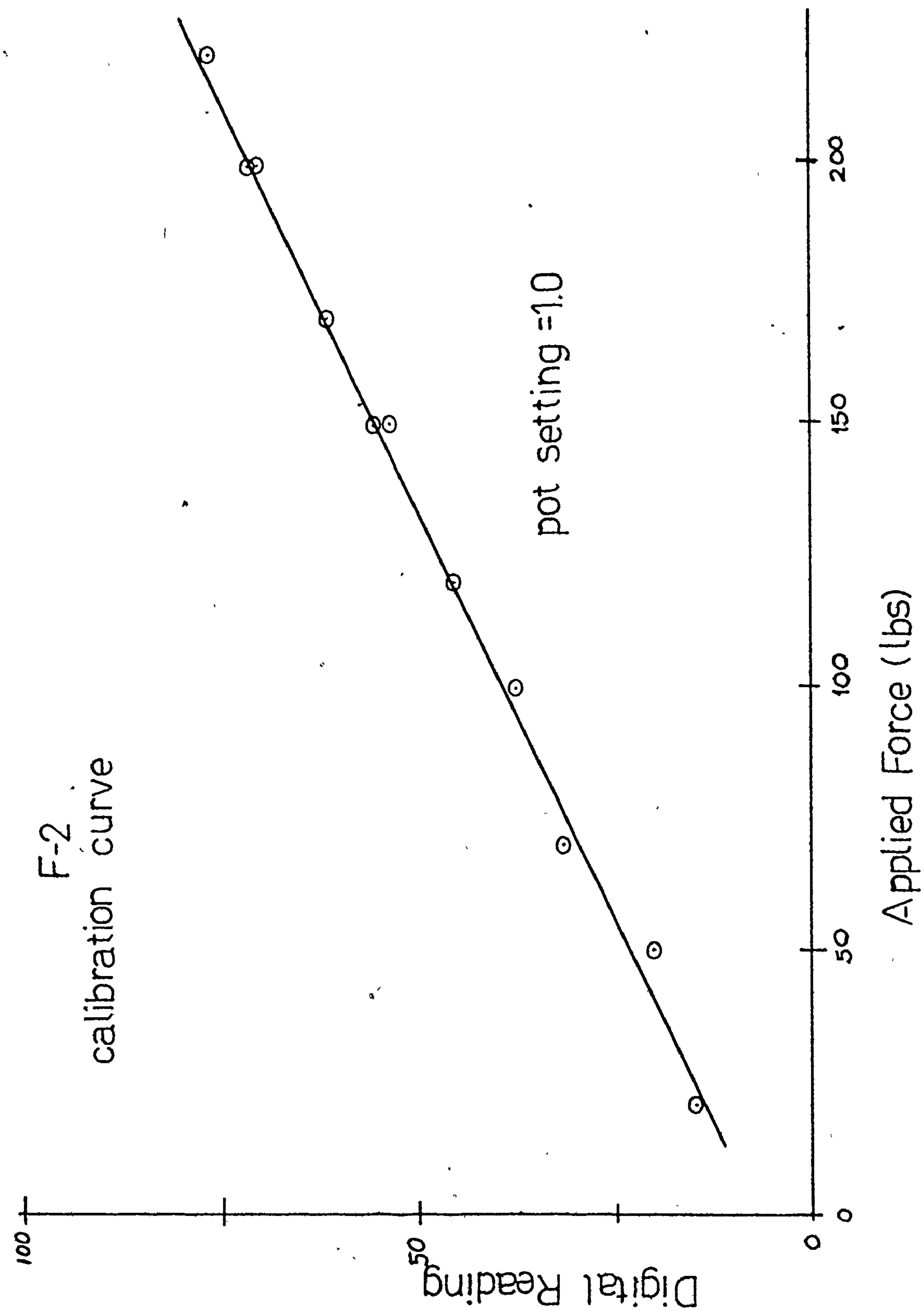

F-2
calibration curve
pot setting = 1.0
Applied Force (lbs)
0
50
100
150
200
Digital Reading
0
50
100

Figure 7

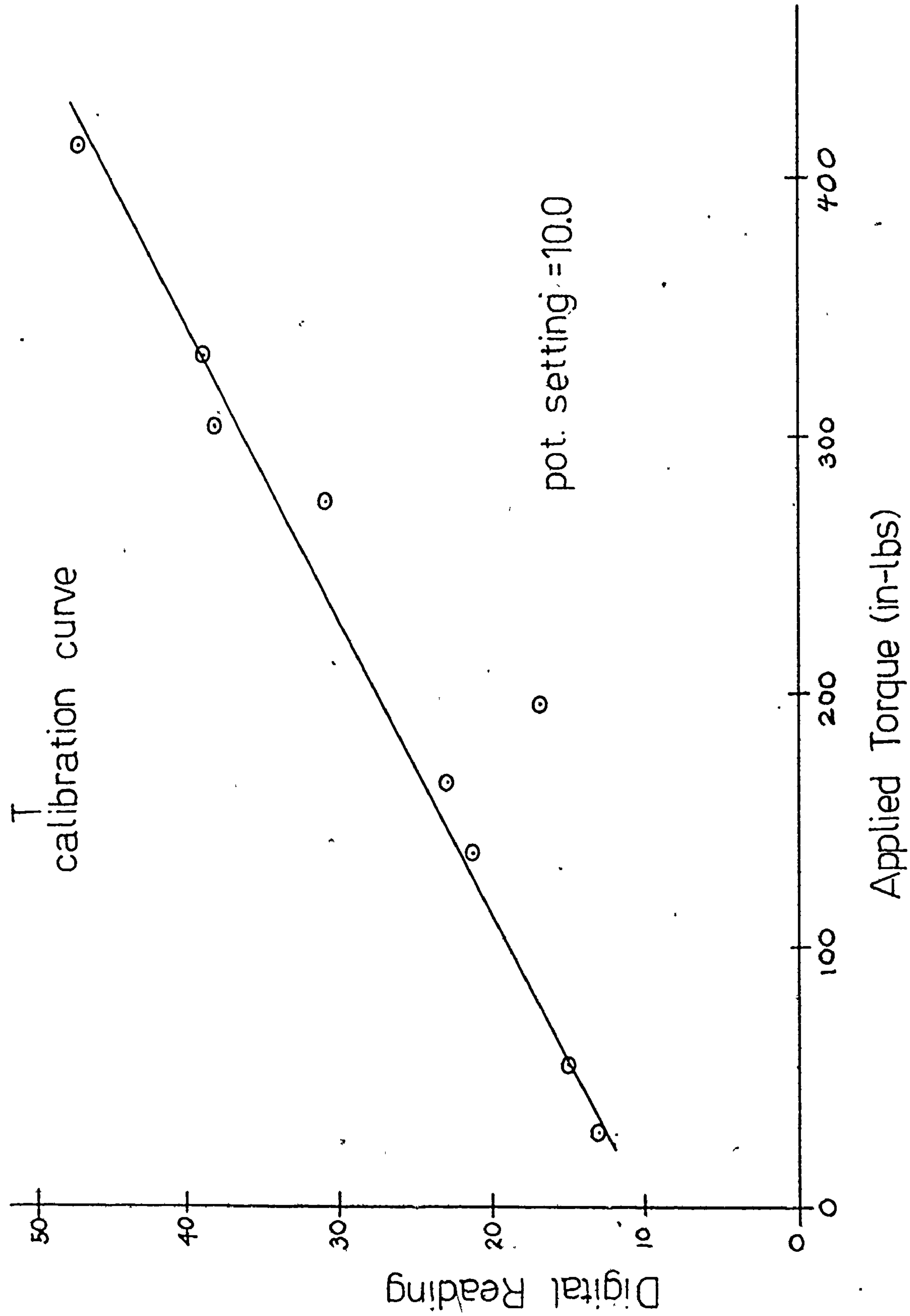

T
calibration curve
pot. setting =10.0
Applied Torque (in-lbs)
Digital Reading
0
100
200
300
400
10
20
30
40
50

Figure 8

Chapter 4

Cutting Forces

Before any simulation of the Adaptive Control system could begin, the nature of the forces produced in milling required examination. For the purpose of simulation, a mathematical model was required which was capable of simulating both the total force and torque acting on an end mill since both quantities were to be measured by the A/C transducer. To simplify the task somewhat, drilling or sinking operations were not modelled, and only conventional or "up" milling was analysed. The model began with a consideration of the radial and tangential forces acting on a single tooth, given its relative position in the cut. The other required vaiables include the specific cutting force of the material K, the cutter radius r, the width of cut b, the depth of cut a, the feed per tooth S_t and the helix angle of the end mill teeth β. The fundamental relation for an increment of tangential force acting on an incremental length of cutting edge dy in angular position ϕ is:

$$dF_t = K \cdot S_t \cdot \sin\phi \, dy \qquad \text{for } 0 \leqslant \phi \leqslant \phi_a$$

and

$$dF_t = 0 \qquad \text{for } \phi > \phi_a .$$

Similarly, the basic equation for the increment of radial force is:

$$dF_r = 0.3 \cdot K \cdot S_t \cdot \sin\phi \, dy \qquad \text{for } 0 \leqslant \phi \leqslant \phi_a$$

and

$$dF_r = 0$$

for $\phi > \phi_a$

(The constant 0.3 relating the magnitudes of radial and tangential force is an average empirical value which is generally accepted.) Figure 9 illustrates the physical stiuation. An analysis of the process is facilitated by developing the cut surface; in the developed surface, the helical tooth appears as a straight line, as shown in figure 10. From such a development, it is clearly seen that three distinct phases exist in the cut. As shown in figure 10, the width of cut increases from zero to b in phase A, remains constant throughout phase B and decreases from b to zero in phase C. However, figure 11 illustrates another possible situation in which the depth of cut a is small in comparison to the width of cut b and the full width of cut is never really achieved. The three distinct phases do remain wherein width of cut increases from zero to $b' < b$ in phase A, remains constant at b' in phase B and decreases from b' to zero in phase C.

To determine the total force and torque acting on the end mill, the directionality of these radial and tangential tooth forces is important. Referring to figure 12, it may be seen that a vector from the point of tooth-workpiece contact to the centre line of the cutter coincides with the direction of dF_r. The direction of dF_t is then perpendicular to the direction of dF_r and pointing in a direction opposite to that of the cutter rotation. The torque acting on the end mill is determined directly from the tangential force acting on all teeth involved in cutting, while the total force acting on the cutter is the resultant of both the tangential and radial components of force. Thus, to obtain

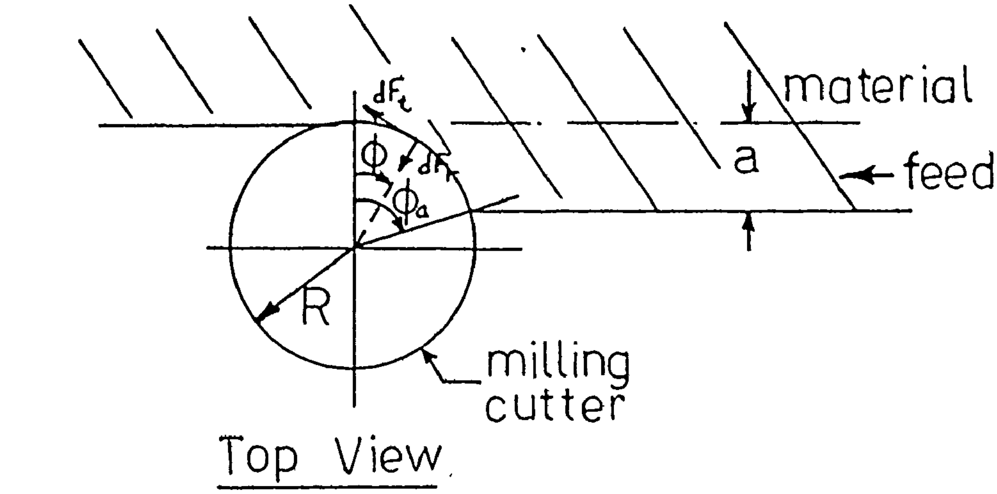
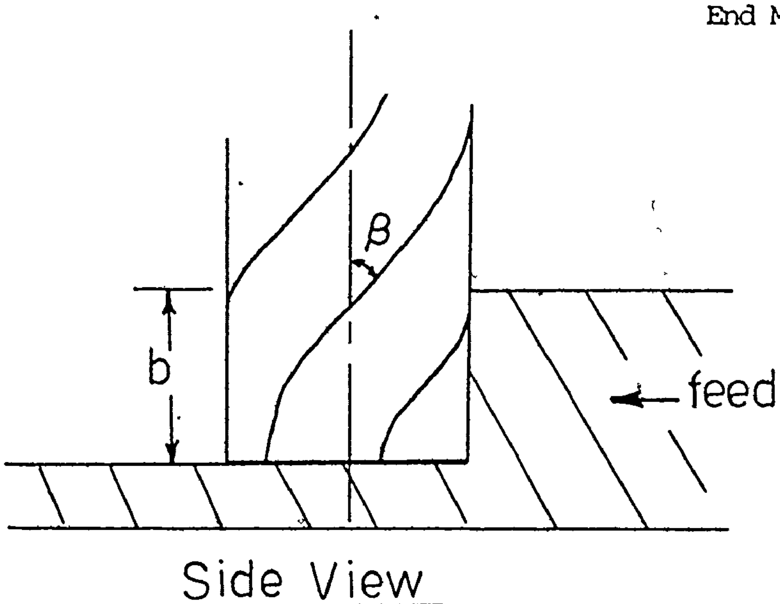


Figure 9

End Mill



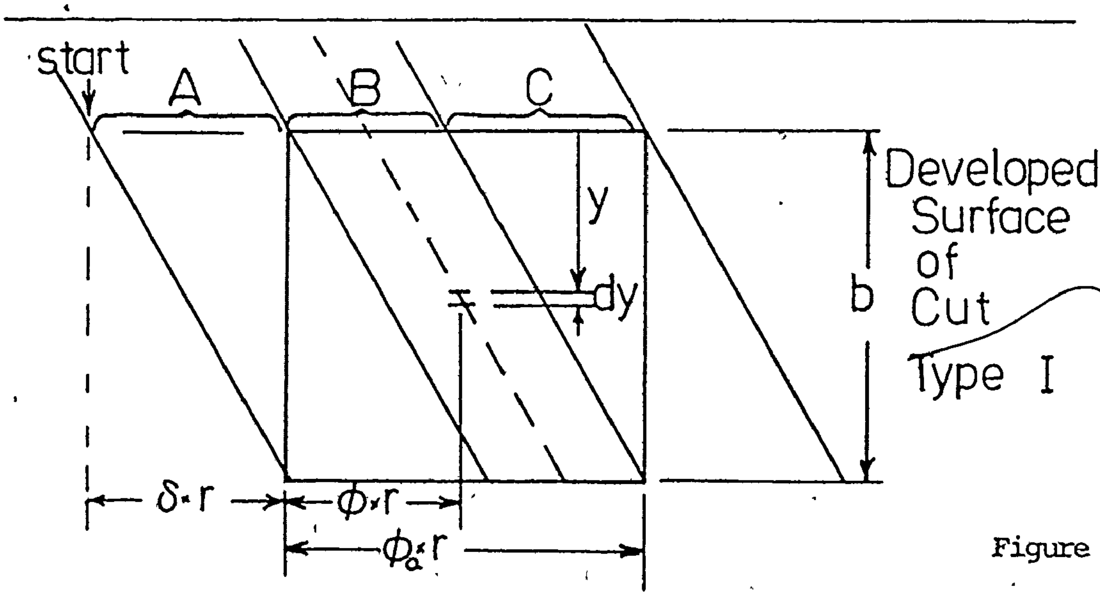
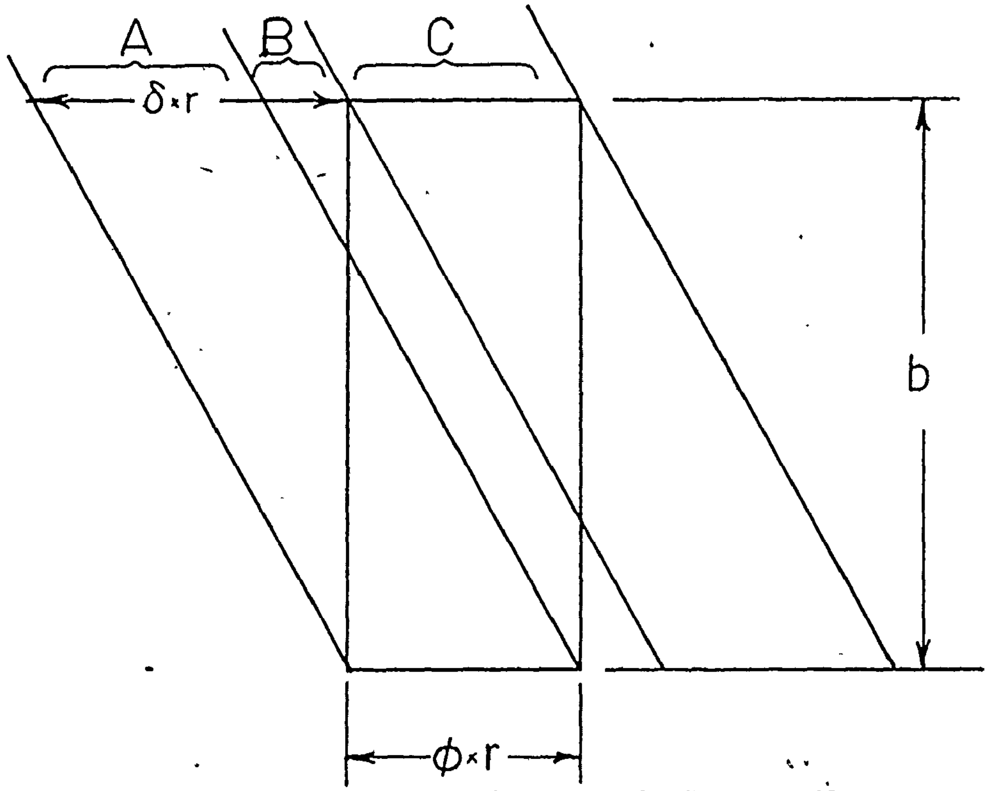
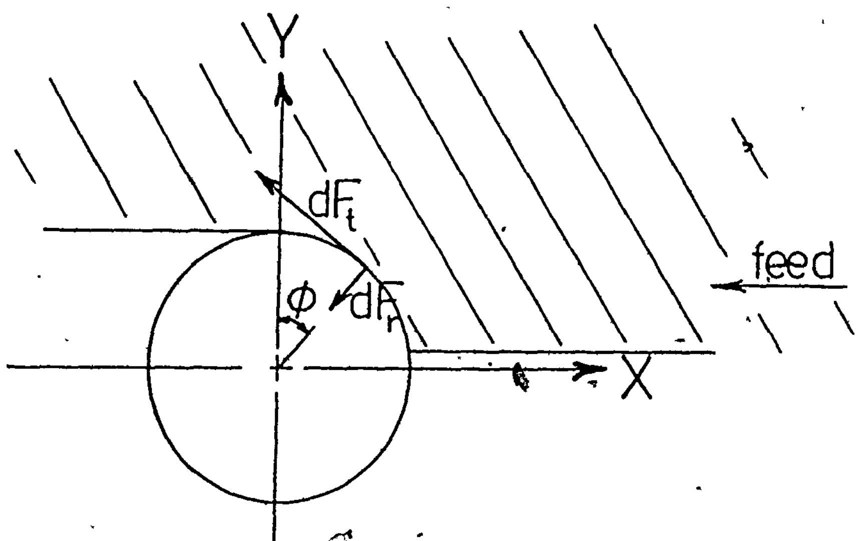


Figure 10



Developed Surface of Cut - Type II

Figure 11



Coordinate System

Figure 12

this resultant, the forces are superimposed on an X-Y coordinate system as shown in figure 12. Thus, the fundamental relations for the four components of force become:

$$dF_{rx} = -0.3\,K \cdot S_t \cdot \sin^2\phi\,dy$$

$$dF_{ry} = -0.3\,K \cdot S_t \cdot \sin\phi\cos\phi\,dy$$

$$dF_{tx} = -K \cdot S_t \cdot \sin\phi\cos\phi\,dy$$

$$dF_{ty} = K \cdot S_t \cdot \sin^2\phi\,dy$$

It can be seen that dF_{rx} and dF_{ty} differ only by the factor 0.3 and similarly with dF_{ry} and dF_{tx}. Thus, the complete development in Appendix A uses only the relations for dF_{tx} and dF_{ty}.

The basic equations for dF_t and its two components are integrated over the length of the tooth in contact with the workpiece during each phase of the cut. The final integrated relations give the tangential force and the total force acting on a single tooth for a given angle of rotation of the cutter. Since each phase of the cut has different limit conditions, a different relation is required for each phase. Thus, in using these relations, the initial data must first be used to divide the cut into its three phases before calculating the forces on the tooth. The final relations are as follows:

Phase A $\quad 0 \leqslant \phi \leqslant \delta$

$$F_t = F_u\left\{1 - \cos\phi\right\}$$

$$F_x = -\frac{F_u}{2}\left\{0.3\,\phi - 0.3\,\frac{\sin 2\phi}{2} + \sin^2\phi\right\}$$

$$F_y = \frac{F_u}{2}\left\{\phi - \frac{\sin 2\phi}{2} - 0.3\,\sin^2\phi\right\}$$

Phase B $\quad \delta \leqslant \phi \leqslant \phi_a$

$$F_t = F_u\left\{\cos(\phi - \delta) - \cos\phi\right\}$$

$$F_x = -\frac{F_u}{2}\left\{0.3\delta - 0.3\frac{\sin 2\phi}{2} + 0.3\frac{\sin 2(\phi-\delta)}{2} + \sin^2\phi - \sin^2(\phi-\delta)\right\}$$

$$F_y = \frac{F_u}{2}\left\{\delta - \frac{\sin 2\phi}{2} + \frac{\sin 2(\phi-\delta)}{2} - 0.3\sin^2\phi + 0.3\sin^2(\phi-\delta)\right\}$$

Phase C $\phi_a \leqslant \phi \leqslant \phi_a + \delta$

$$F_t = F_u\left\{\cos(\phi-\delta) - \cos\phi_a\right\}$$

$$F_x = -\frac{F_u}{2}\left\{0.3(\phi_a+\delta-\phi) - 0.3\frac{\sin 2\phi_a}{2} + 0.3\frac{\sin 2(\phi-\delta)}{2} + \sin^2\phi_a - \sin^2(\phi-\delta)\right\}$$

$$F_y = \frac{F_u}{2}\left\{(\phi_a+\delta-\phi) - \frac{\sin 2\phi_a}{2} + \frac{\sin 2(\phi-\delta)}{2} - 0.3\sin^2\phi_a + 0.3\sin^2(\phi-\delta)\right\}$$

where: F_t = tangential force on tooth

F_x & F_y = components of total force on tooth

$$F_u = \frac{K \cdot S_t \cdot r}{\tan\beta}$$

$$\delta = \frac{b \cdot \tan\beta}{r}$$

$$\phi_a = \arccos\left(\frac{r-a}{r}\right)$$

These terms are illustrated in figures 9 through 11.

A cut of type II as indicated in figure 11 will have $\delta > \phi_a$. Thus, two changes are caused in the above series of equations; the first change is a re-definition of the limits of the three phases:

Phase A $0 \leqslant \phi \leqslant \phi_a$

Phase B $\phi_a \leqslant \phi \leqslant \delta$

Phase C $\delta \leqslant \Phi \leqslant \Phi_a + \delta$

The second change is that the force remains constant throughout phase B. This can be seen quite easily by comparing the developed surface of the cut and a top view of the actual cut as shown in figure 13. In the type I cut, it is obvious that B1 and B2 act over different sections of the chip. The chip thickness therefore varies between B1 and B2. However, in the type II cut, B1 and B2 cover the same section of the chip, just at different heights on the cut surface. Thus, the chip thickness and therefore the force remain constant throughout phase B. Therefore, given the new definition of limits for phases, the above listed equations for tooth forces apply for a type II cut in phases A and C while the force in phase B remains constant.

Some examples of typical forces on a tooth are given in figures 14 to 21. This type of information, coupled with information about the distribution of force along the length of a tooth could be used in an analysis of the limit strength of an end mill tooth. Such an analysis could be very useful in establishing limits for adaptive control milling.

Figures 14, 15, 18 and 19 illustrate examples of a type II cut while 16, 17, 20 and 21 illustrate examples of type I. (In all of the figures, the force is plotted against the angle of rotation of the tooth where PHI = 0 is considered the start of the cut.)

The force acting on a single tooth may be useful in determining the limit strength of a tooth, but it is not sufficient for a simulation of forces encountered in milling; at any given time, more than one tooth may be involved in cutting. To determine the total force and torque acting

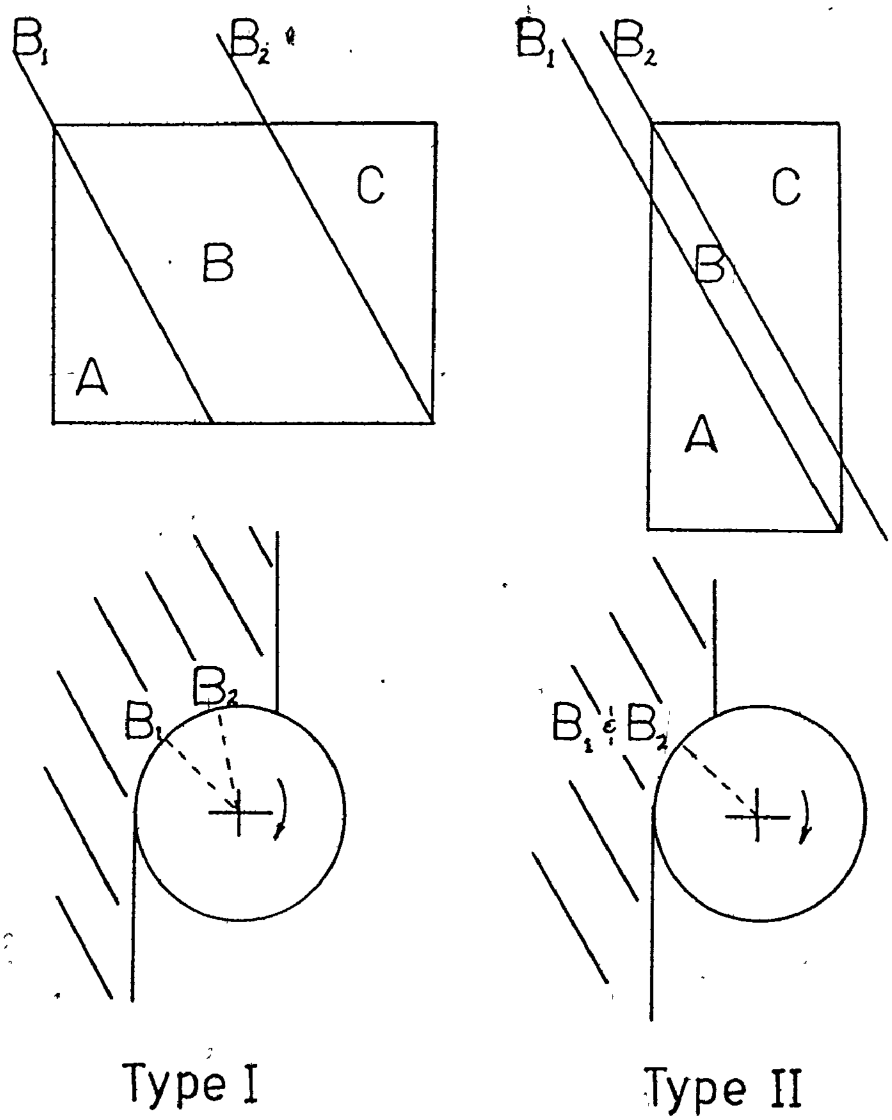
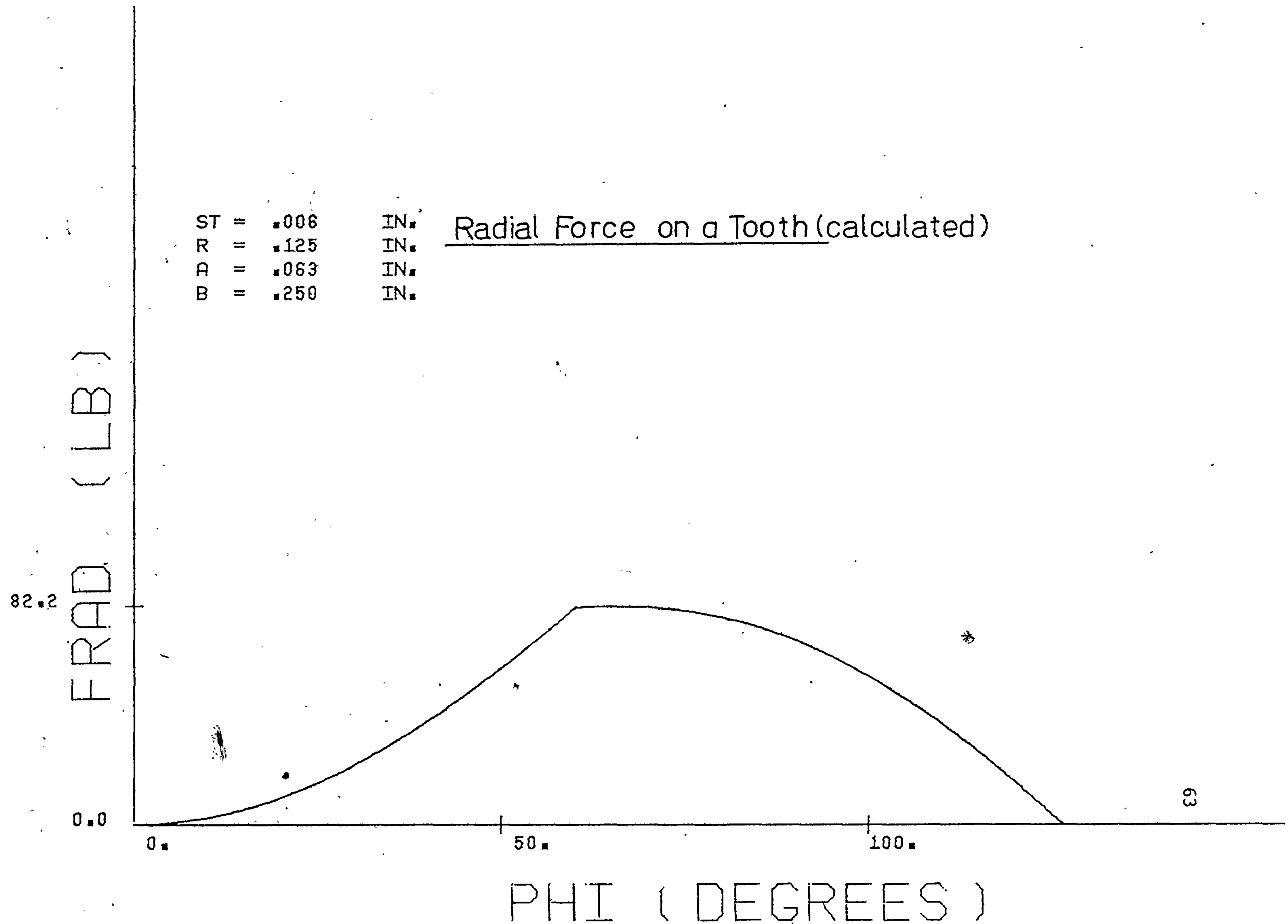


Figure 13

Phase B Comparison


ST = .006 IN.
R = .125 IN.
A = .063 IN.
B = .250 IN.
Radial Force on a Tooth (calculated)
82.2
FRAD (LB)
0.0
0.
50.
100.
PHI (DEGREES)

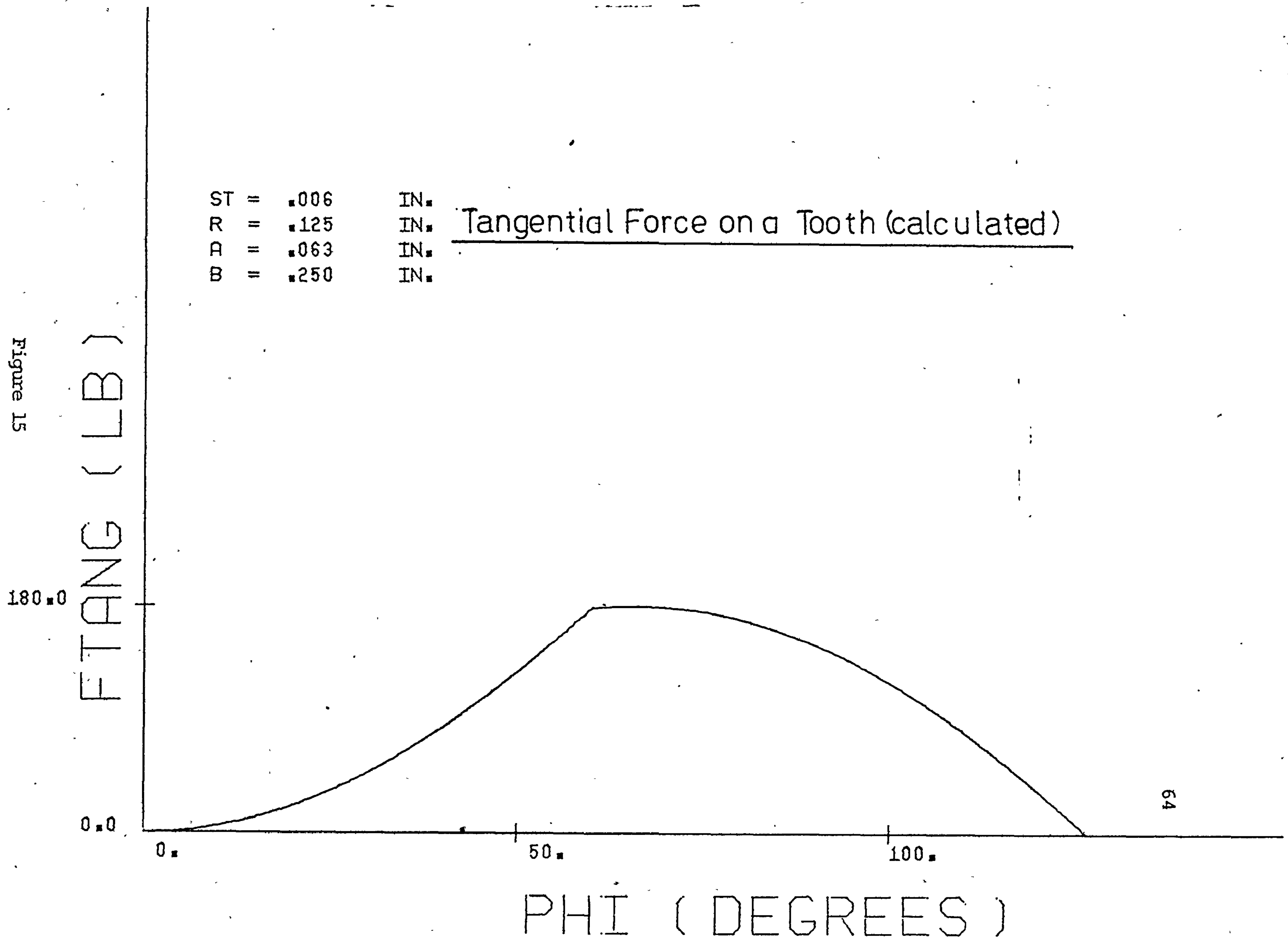

Tangential Force on a Tooth (calculated)
ST = .006 IN.
R = .125 IN.
A = .063 IN.
B = .250 IN.
FTANG (LB)
180.0
0.0
0.
50.
100.
PHI (DEGREES)

Figure 15

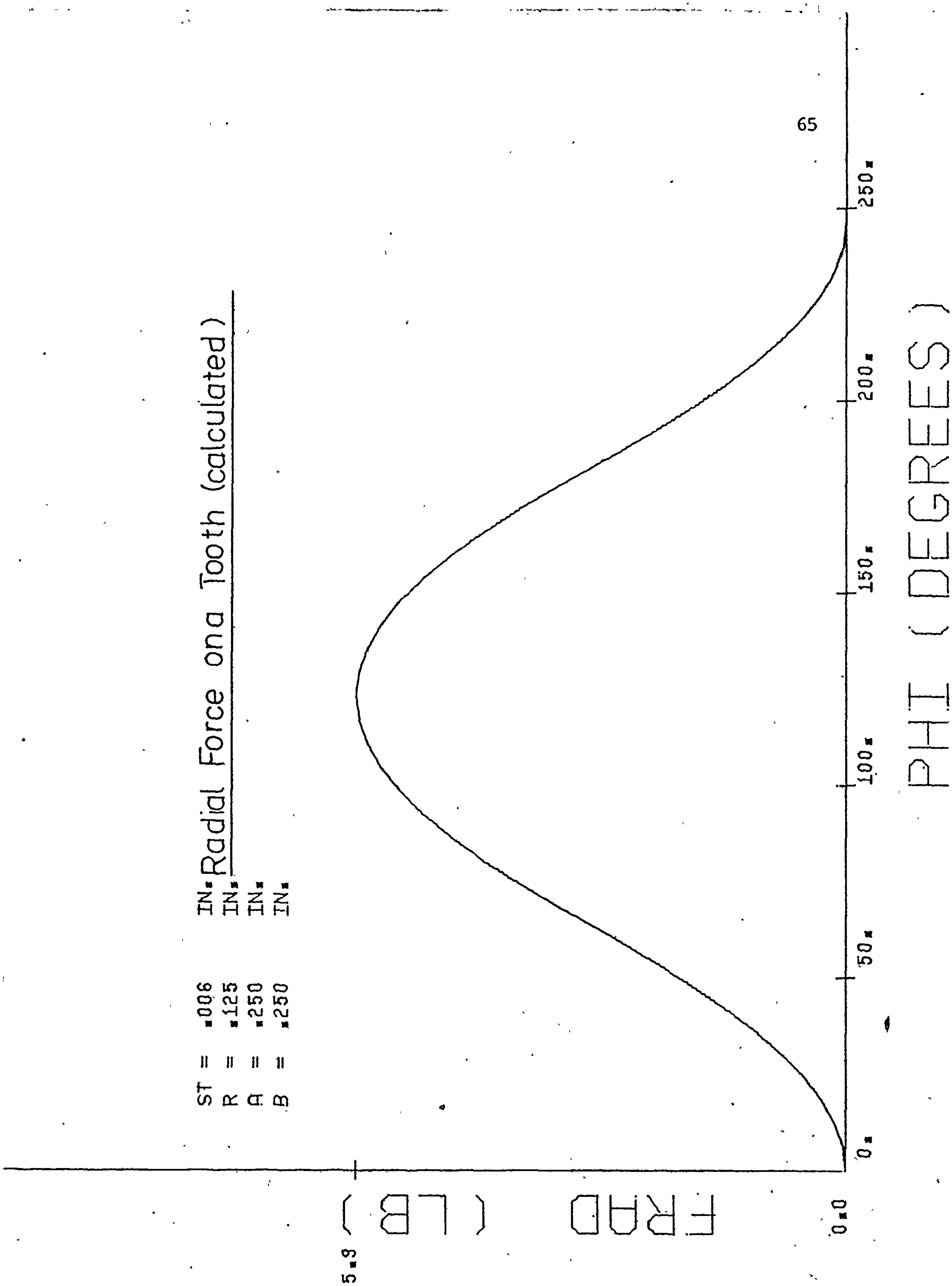

Radial Force ona Tooth (calculated)
ST = .006 IN.
R = .125 IN.
A = .250 IN.
B = .250 IN.
FRAD (LB)
385.9
0.0
0.
50.
100.
150.
200.
250.
PHI (DEGREES)

Figure 16

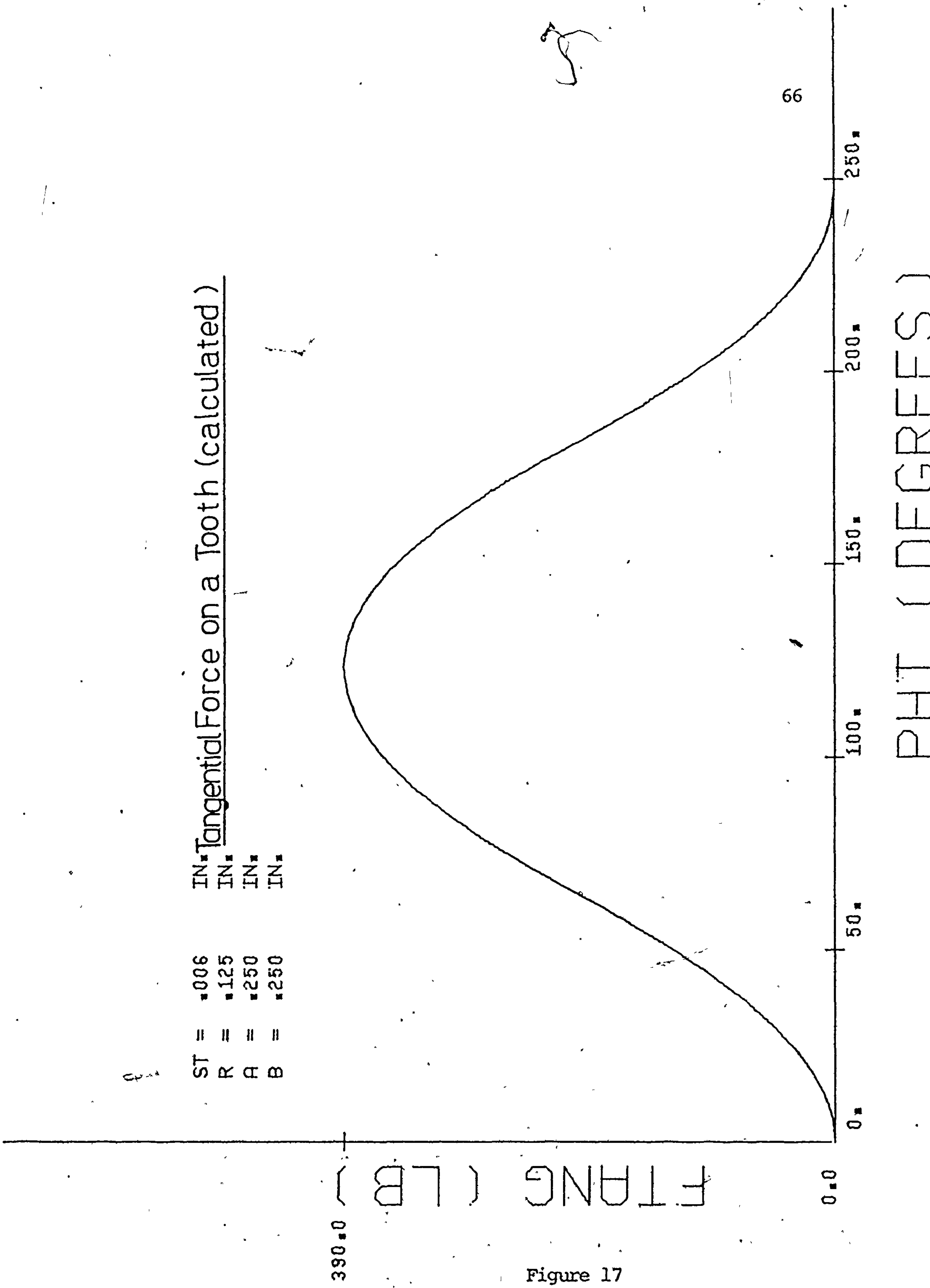
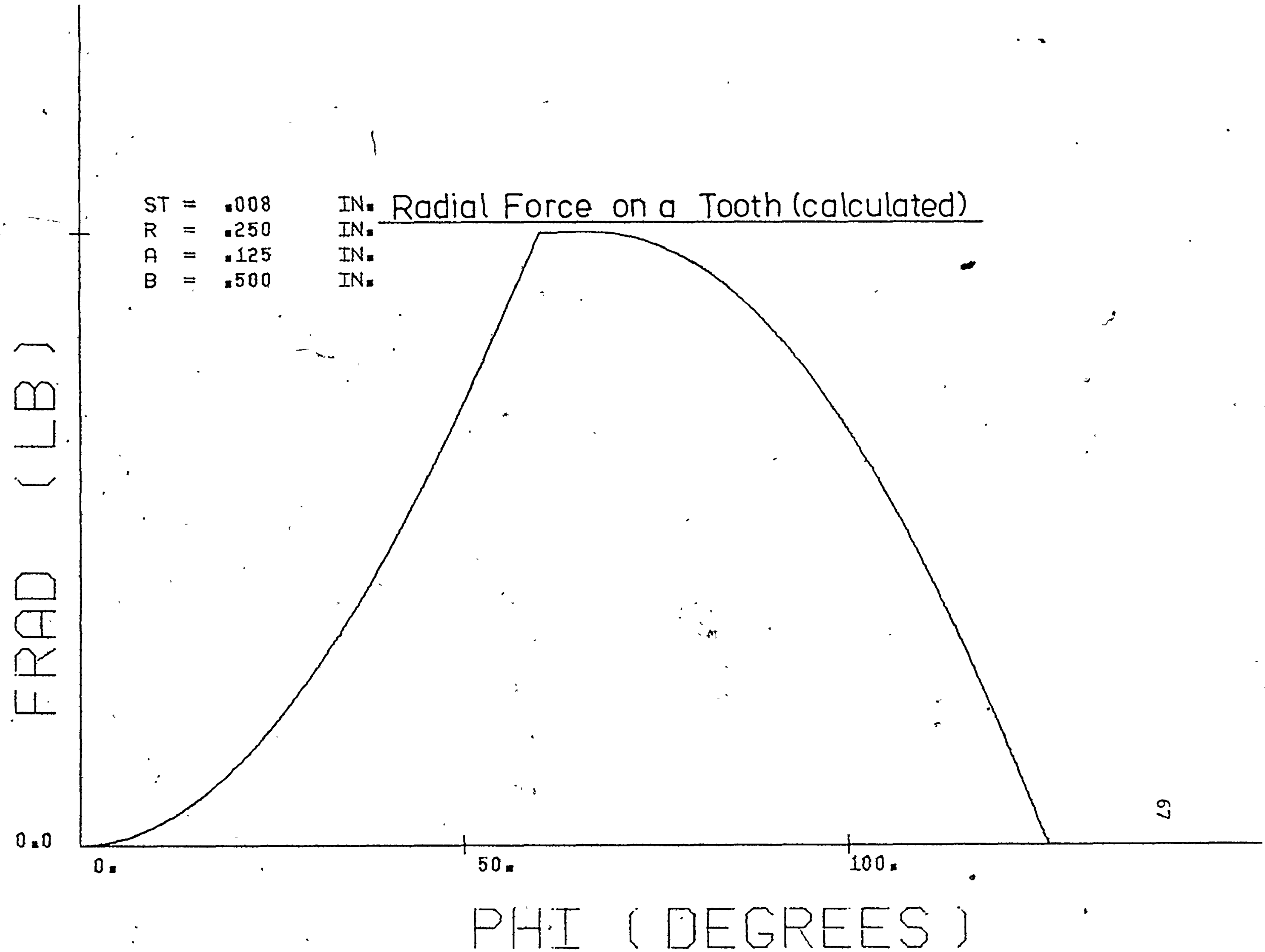

Tangential Force on a Tooth (calculated)
ST = .006 IN.
R = .125 IN.
A = .250 IN.
B = .250 IN.
FTANG (LB)
390.0
0.0
0.
50.
100.
150.
200.
250.
PHI (DEGREES)

Figure 17


Radial Force on a Tooth (calculated)
ST = .008 IN.
R = .250 IN.
A = .125 IN.
B = .500 IN.
482.1
0.0
FRAD (LB)
0.
50.
100.
PHI (DEGREES)

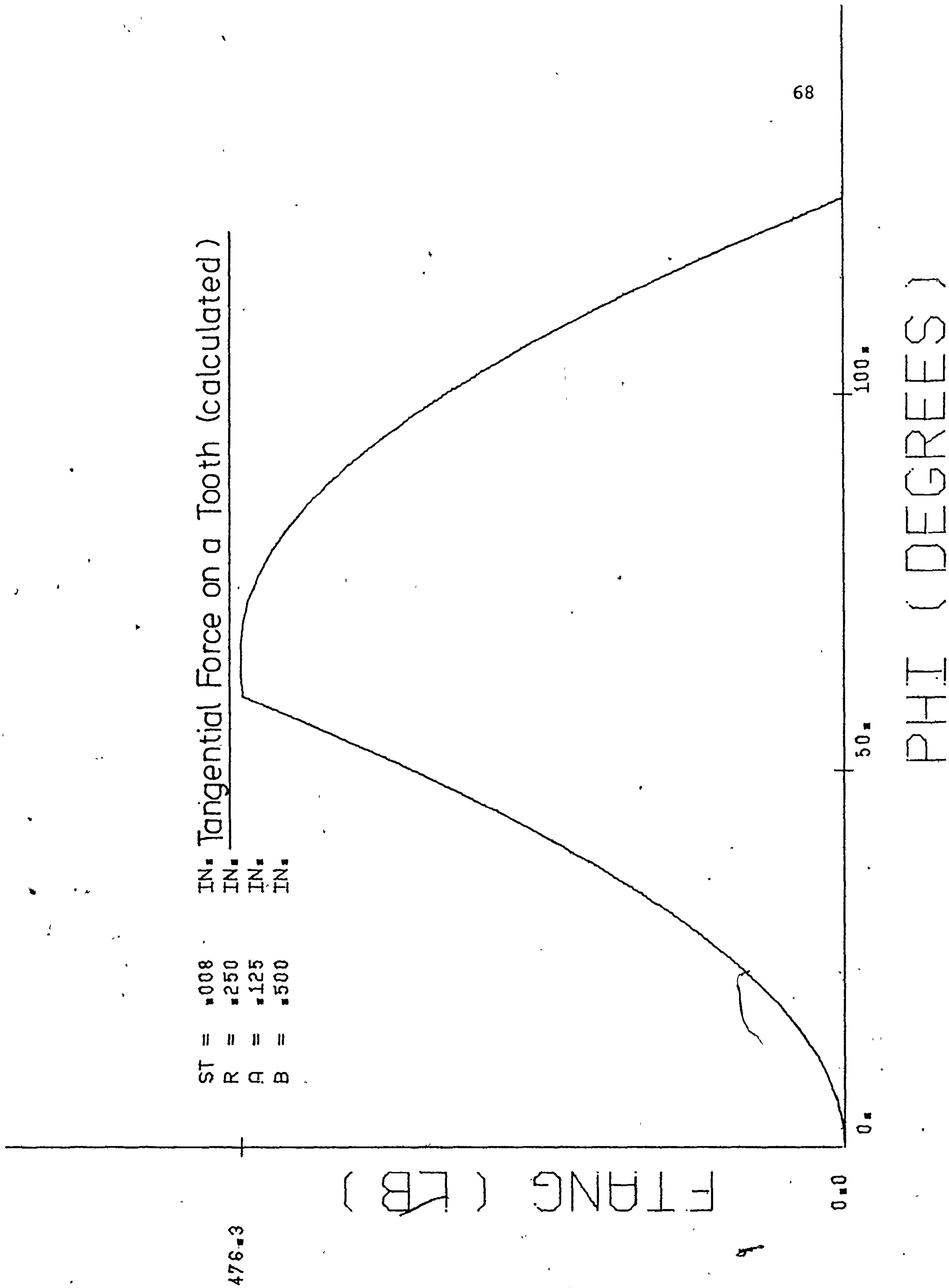
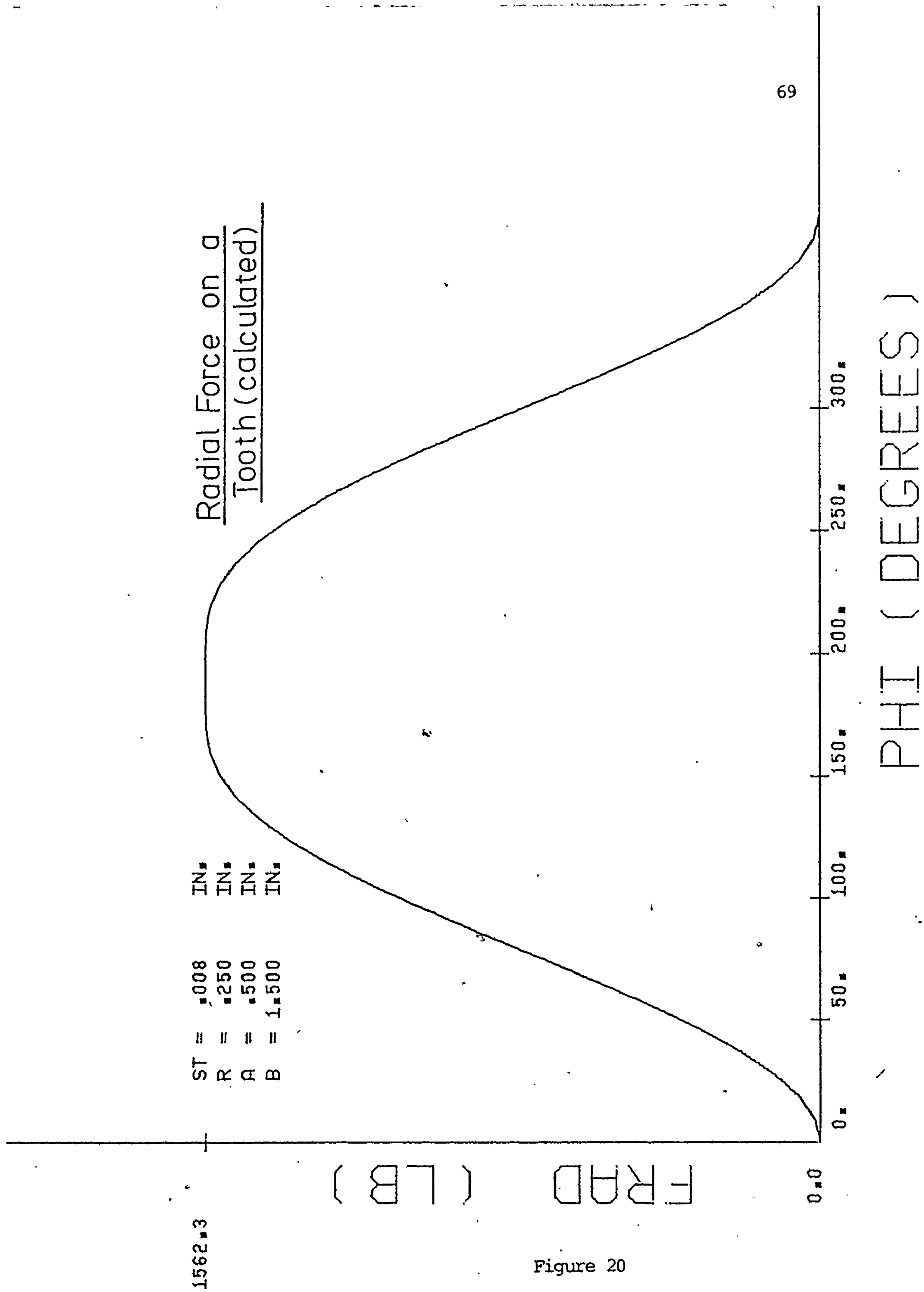
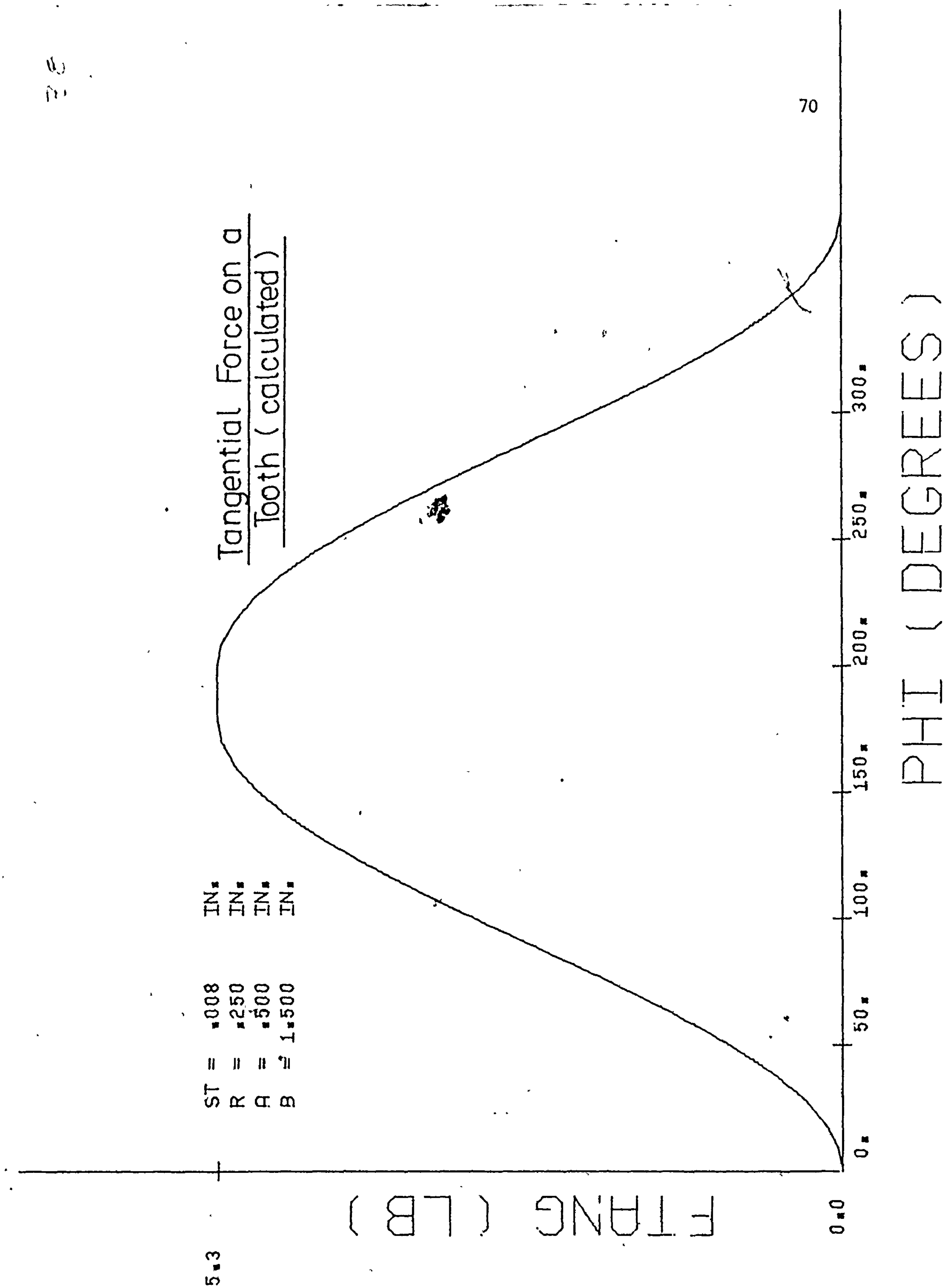

Tangential Force on a Tooth (calculated)
ST = .008 IN.
R = .250 IN.
A = .125 IN.
B = .500 IN.
PHI (DEGREES)
0.
50.
100.
FTANG (LB)
0.0
476.3

Figure 19



Radial Force on a Tooth (calculated)
ST = .008 IN.
R = .250 IN.
A = .500 IN.
B = 1.500 IN.
0.
50.
100.
150.
200.
250.
300.
PHI (DEGREES)
FRAD (LB)
0.0
1562.3

Figure 20



Tangential Force on a Tooth (calculated)
ST = .008 IN.
R = .250 IN.
A = .500 IN.
B = 1.500 IN.
FTANG (LB)
1905.3
0.0
PHI (DEGREES)
0.
50.
100.
150.
200.
250.
300.

Figure 21

on an end mill another parameter is required, the number of teeth on the cutter z. Thus, at any moment of time, it must be determined how many teeth are in the cut. Then F_t , F_x and F_y are determined for each tooth in the cut, based upon the angle ϕ of each tooth relative to the start of the cut. Then, the final quantities desired in the model are obtained for any given time t:

$$T = r \cdot \sum^{n} F_{t_i}$$

$$F = \sqrt{\left[\sum^{n} F_{t_{x_i}} + F_{r_{x_i}}\right]^2 + \left[\sum^{n} F_{t_{y_i}} + F_{r_{y_i}}\right]^2}$$

where: T = Torque (total)

F = Force (total)

F_{t_i} , F_{x_i} , F_{y_i} represent the tooth forces on the ith tooth in the cut

and n = number of teeth in the cut.

Of course, T, F and n are time varying quantities, so, a computer program was written to calculate T and F and plot their variation with cutter rotation. Several examples were computed in order to examine the various types of force and torque functions which may occur in milling. Figures 22, 23, 24, 25, 26 and 27 represent type II situations. The constant force in phase B is accentuated in figures 24 and 25. Figures 26 to 31 illustrate the smoothing effect which increasing the number of teeth has on the force and torque.

A slightly different situation exists in the case of the cutter entering a workpiece as opposed to the steady-state milling model already described. Figure 32 illustrates that the depth of cut a increases from

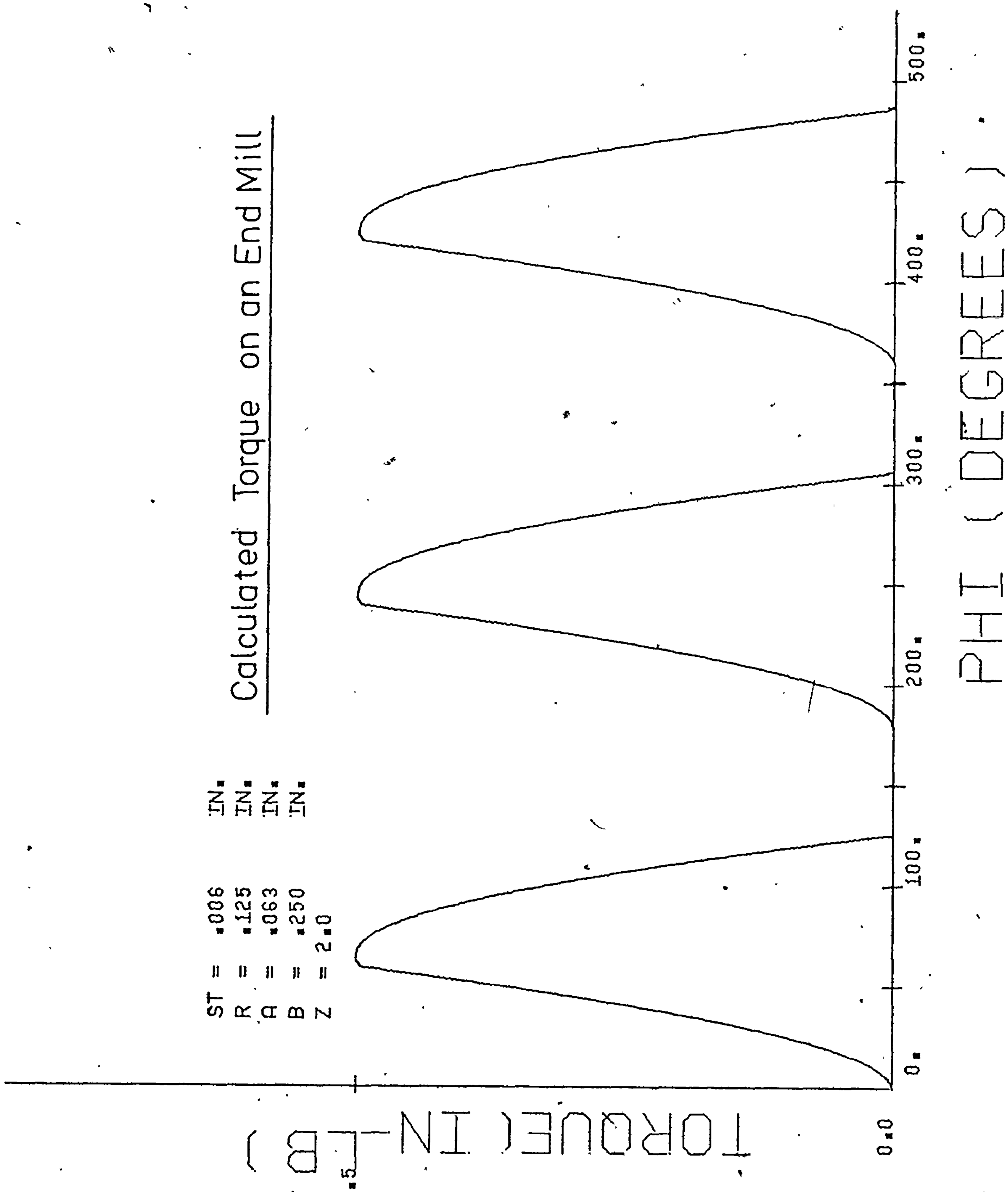
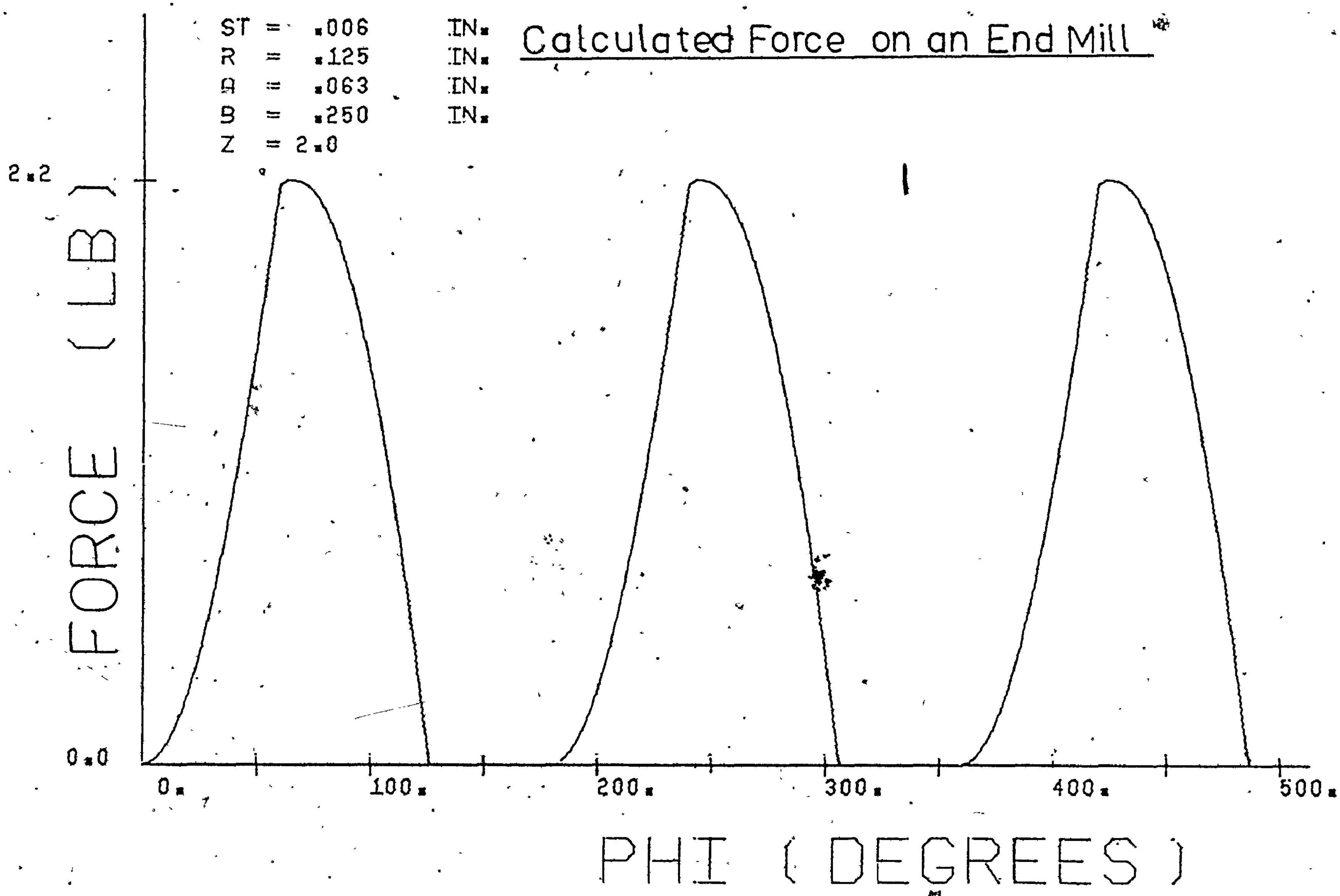
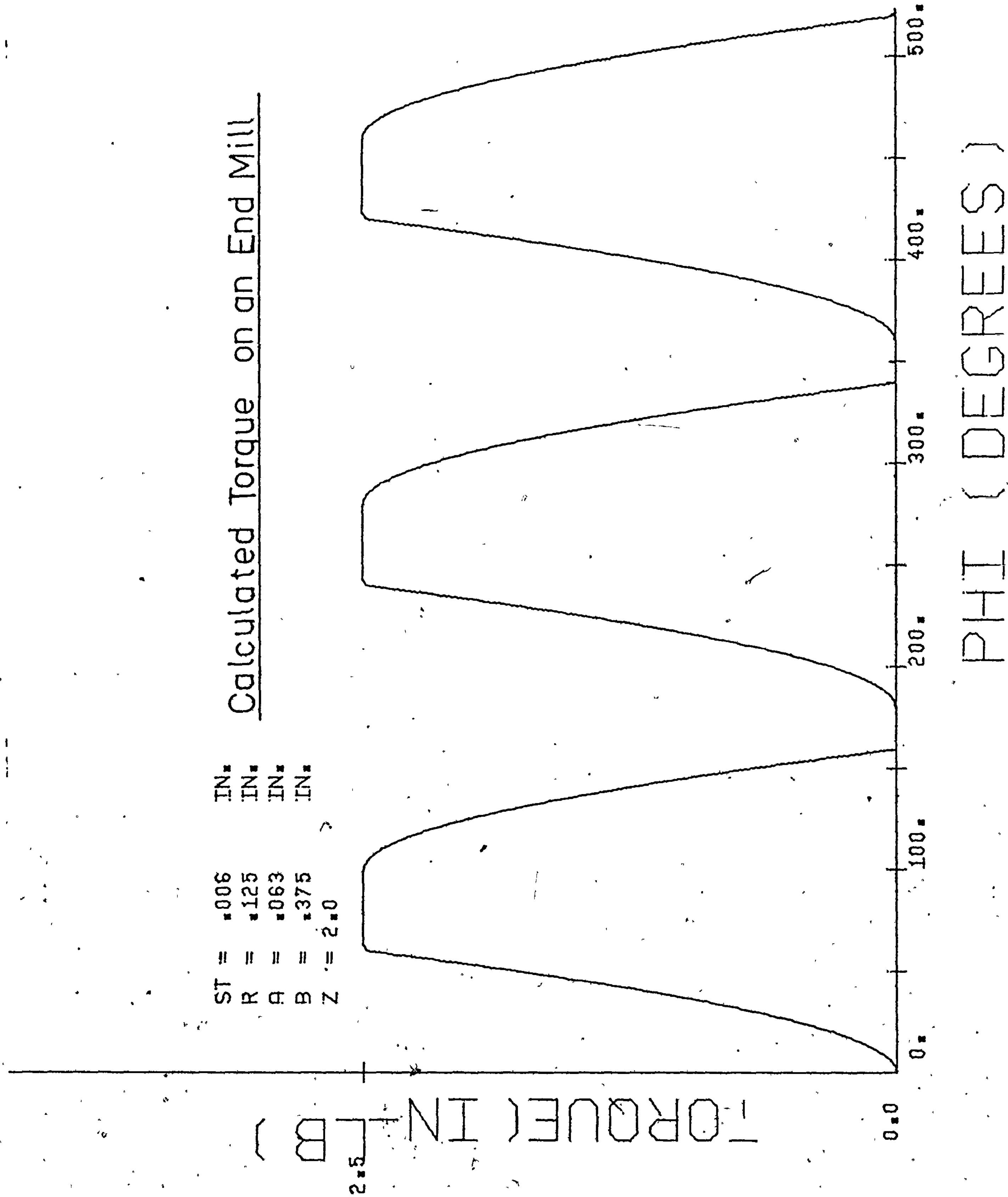

Calculated Torque on an End Mill
ST = .006 IN.
R = .125 IN.
A = .063 IN.
B = .250 IN.
Z = 2.0
TORQUE(IN-LB)
0.0
22.5
PHI (DEGREES)
0.
100.
200.
300.
400.
500.

Figure 22


ST = .006 IN.
R = .125 IN.
A = .063 IN.
B = .250 IN.
Z = 2.0
Calculated Force on an End Mill
FORCE (LB)
2.2
0.0
0.
100.
200.
300.
400.
500.
PHI (DEGREES)



Calculated Torque on an End Mill
ST = .006 IN.
R = .125 IN.
A = .063 IN.
B = .375 IN.
Z = 2.0
TORQUE(IN-LB)
22.5
0.0
0.
100.
200.
300.
400.
500.
PHI (DEGREES)

Figure 24

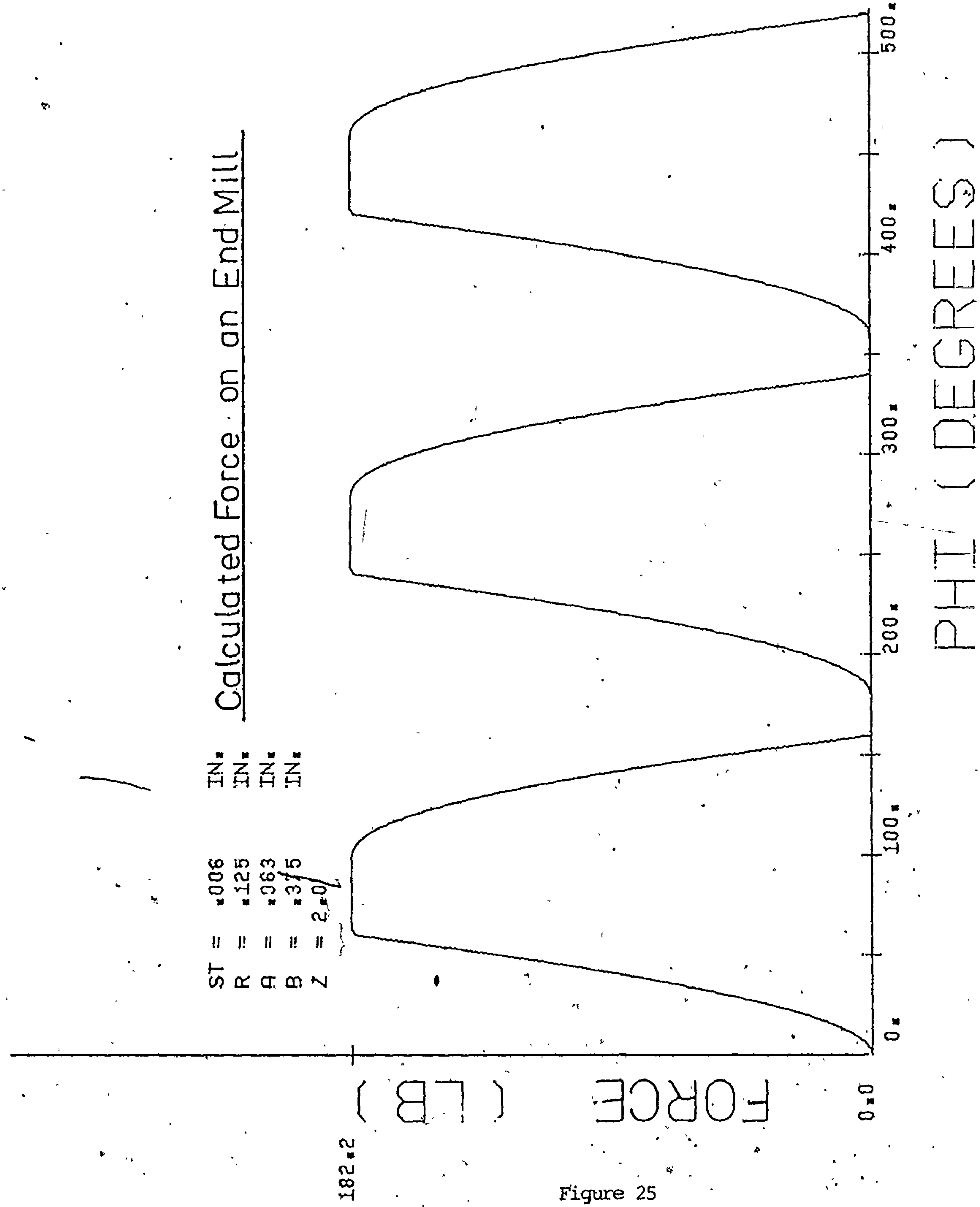

Calculated Force on an End Mill
ST = .006 IN.
R = .125 IN.
A = .363 IN.
B = .375 IN.
Z = 2.0
FORCE (LB)
0.0
182.2
PHI (DEGREES)
0.
100.
200.
300.
400.
500.

Figure 25

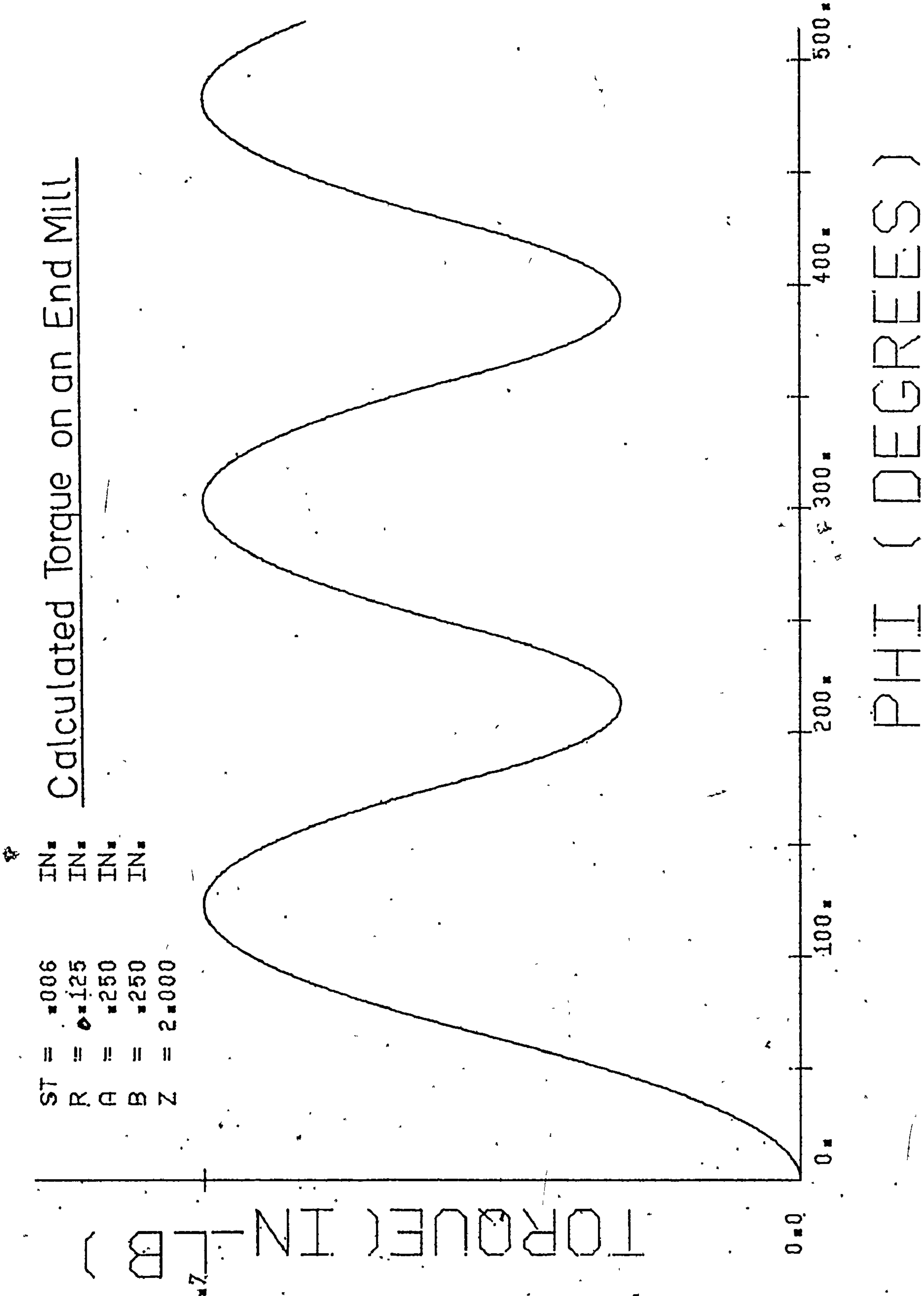

Calculated Torque on an End Mill
ST = .006 IN.
R = .125 IN.
A = .250 IN.
B = .250 IN.
Z = 2.000
TORQUE(IN-LB)
0.0
48.7
PHI (DEGREES)
0.
100.
200.
300.
400.
500.

Figure 26

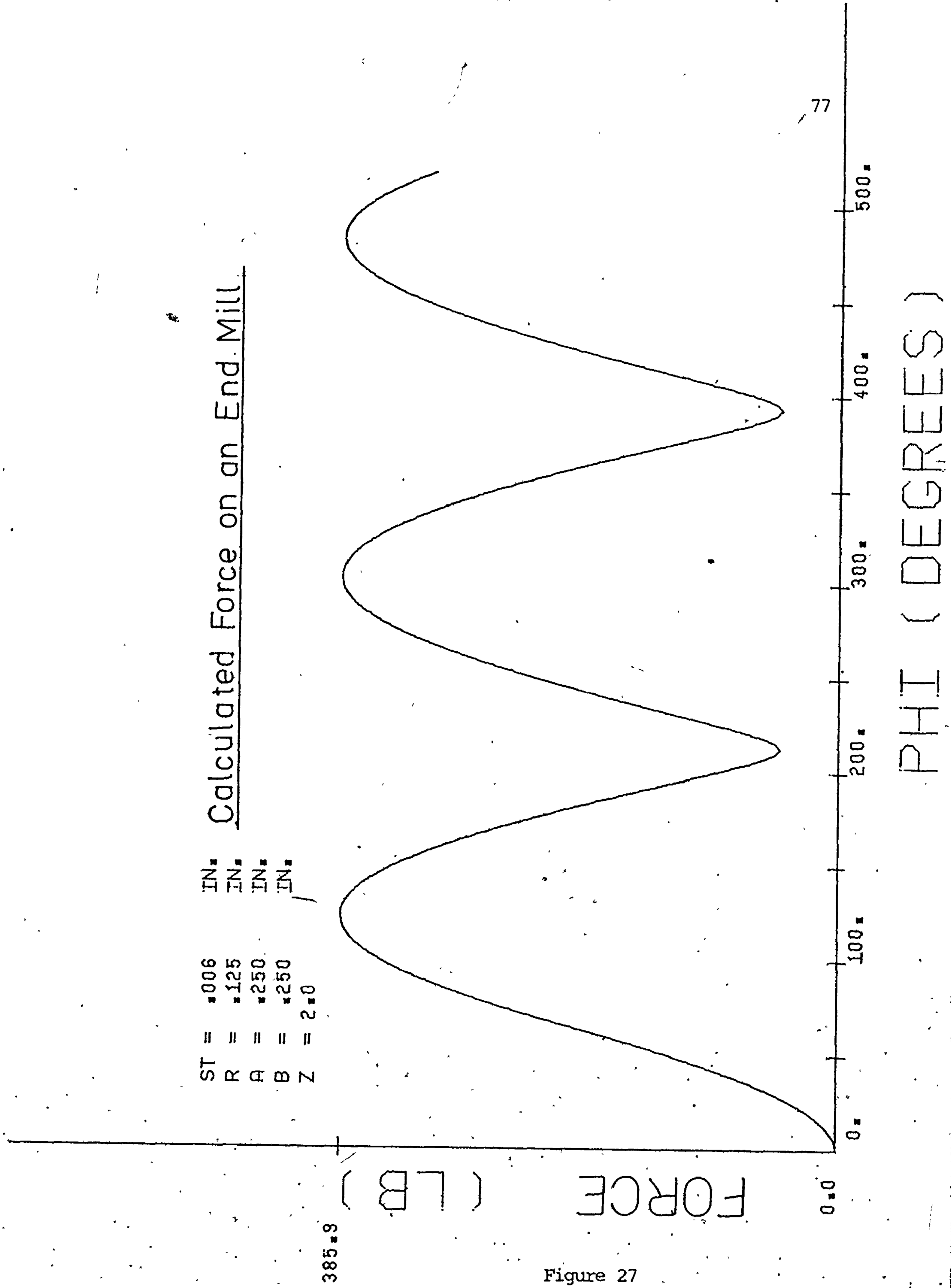

Calculated Force on an End Mill
ST = .006 IN.
R = .125 IN.
A = .250 IN.
B = .250 IN.
Z = 2.0
FORCE (LB)
385.9
0.0
PHI (DEGREES)
0.
100.
200.
300.
400.
500.

Figure 27

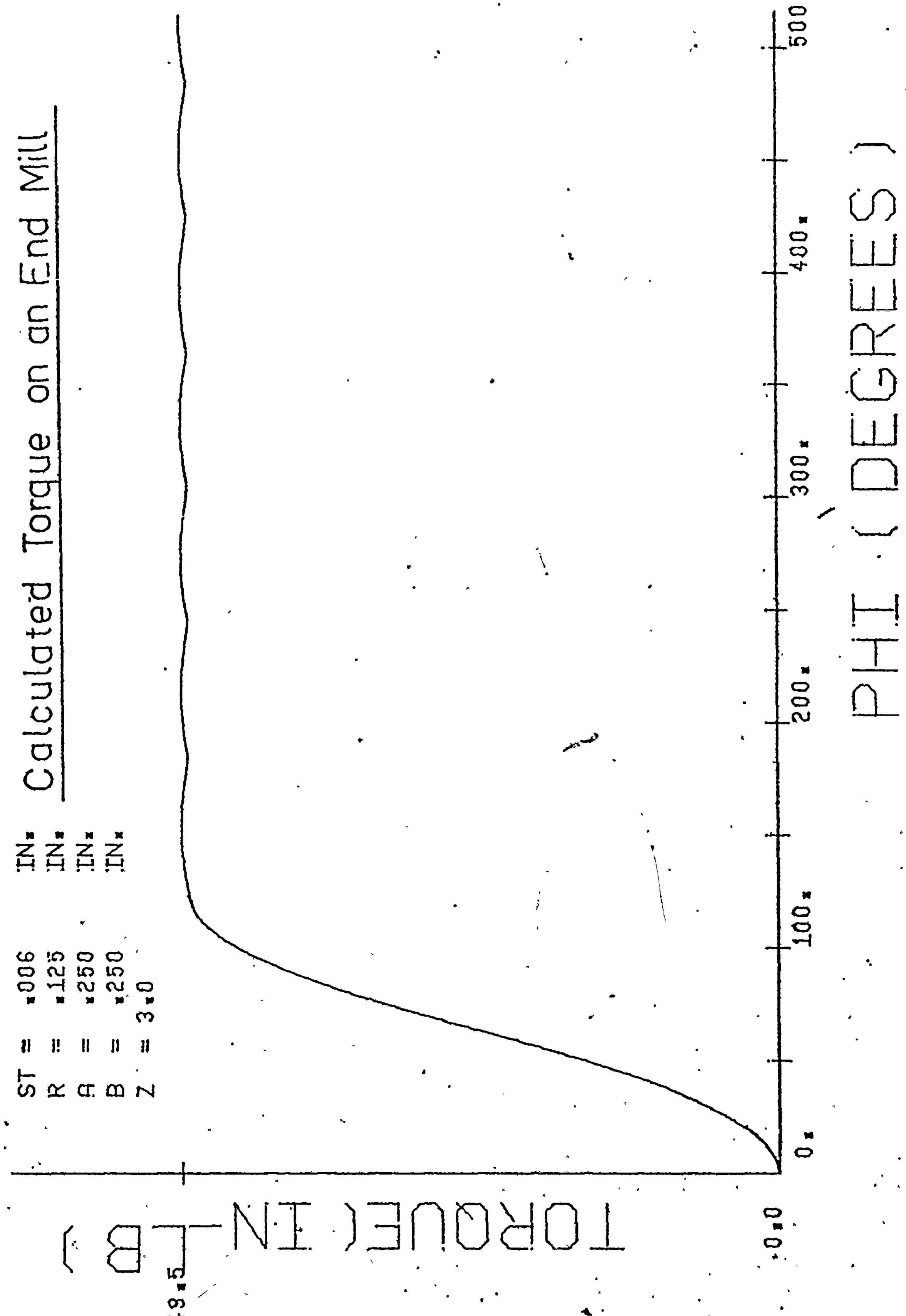

Calculated Torque on an End Mill
ST = .006 IN.
R = .125 IN.
A = .250 IN.
B = .250 IN.
Z = 3.0
TORQUE(IN-LB)
0.0
49.5
0.
100.
200.
300.
400.
500.
PHI (DEGREES)

Figure 28

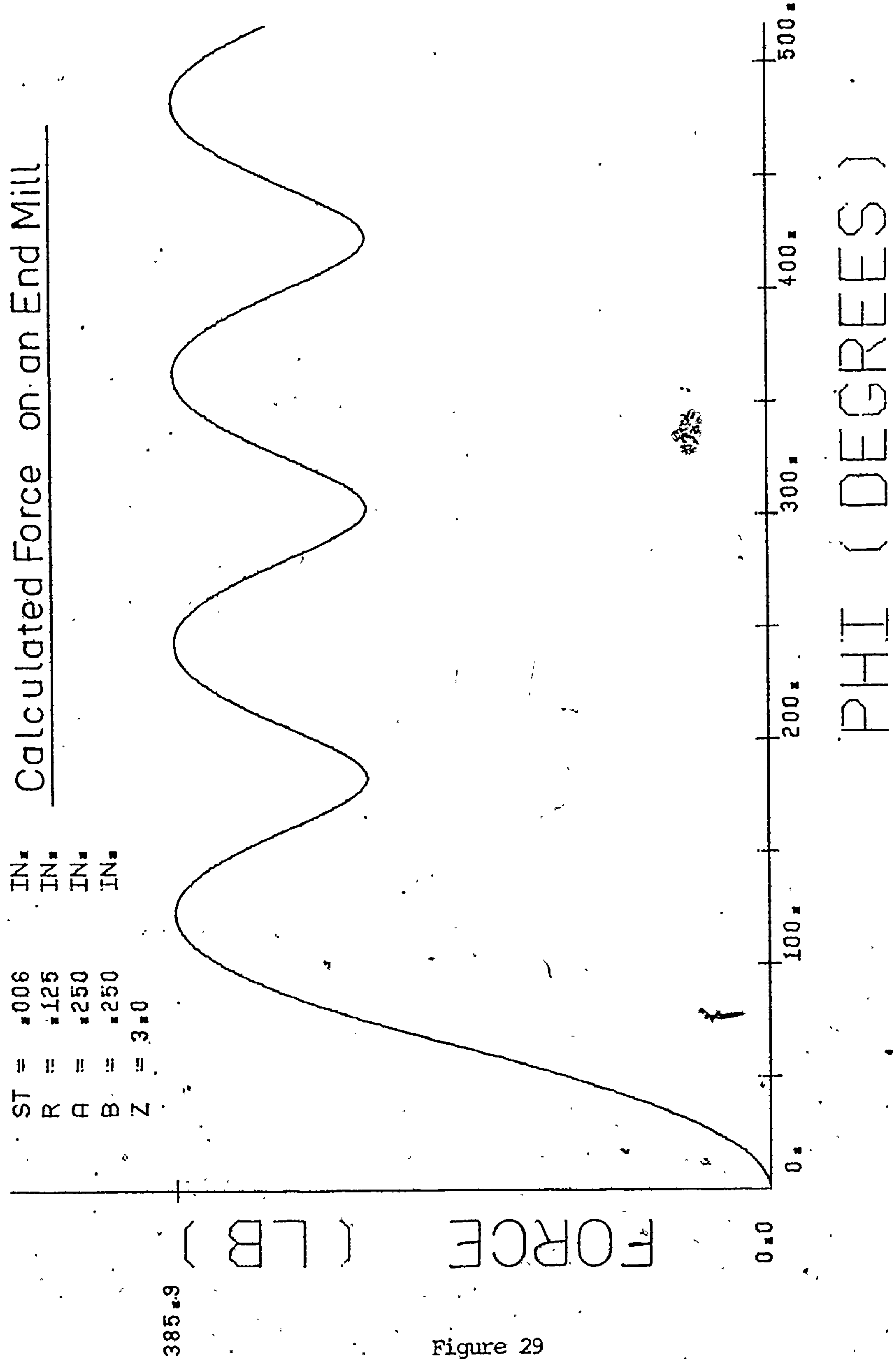

Calculated Force on an End Mill
ST = .006 IN.
R = .125 IN.
A = .250 IN.
B = .250 IN.
Z = 3.0
FORCE (LB)
0.0
385.9
PHI (DEGREES)
0.
100.
200.
300.
400.
500.

Figure 29

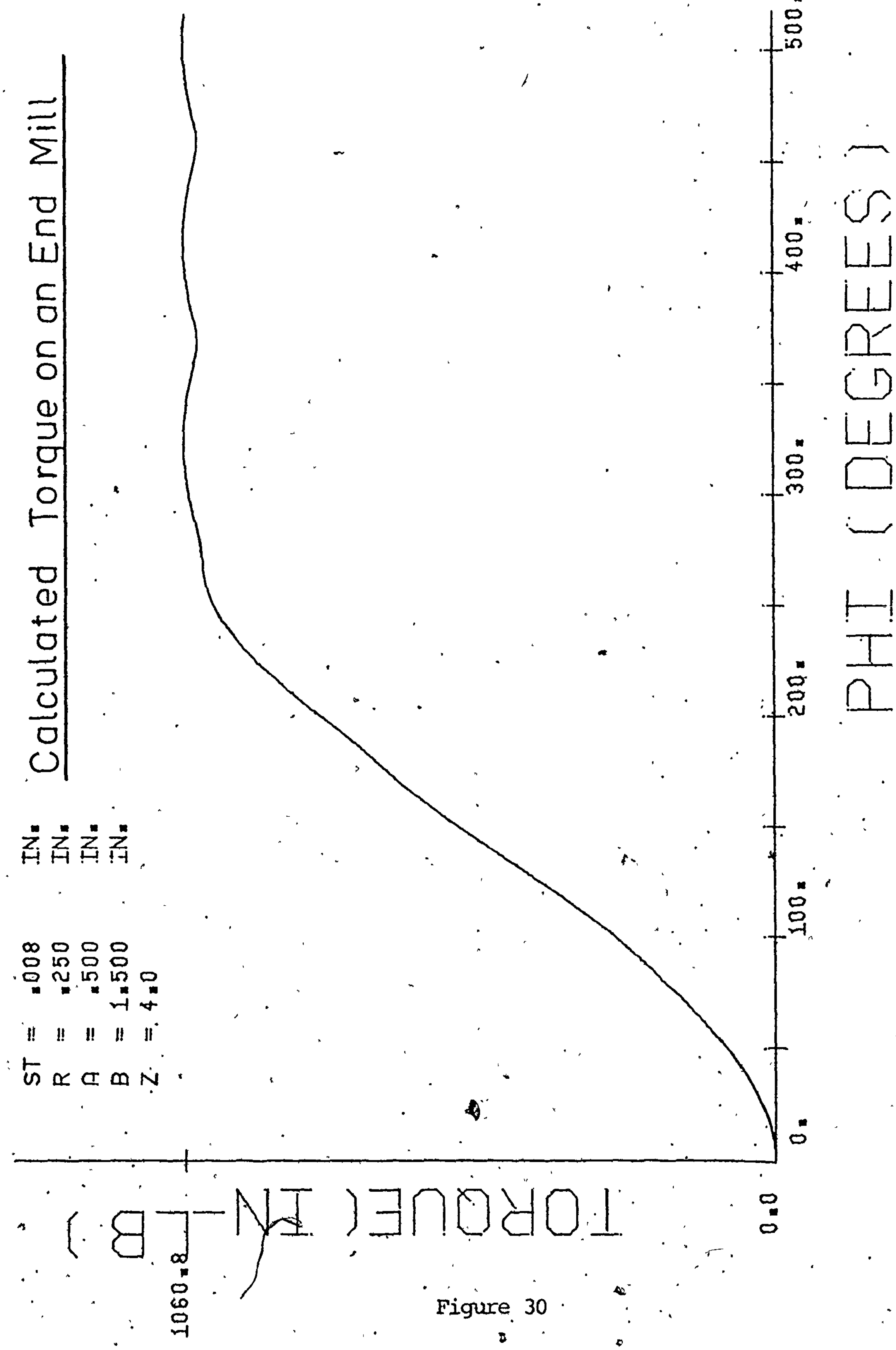

Calculated Torque on an End Mill
ST = .008 IN.
R = .250 IN.
A = .500 IN.
B = 1.500 IN.
Z = 4.0
TORQUE (IN-LB)
0.0
1060.8
PHI (DEGREES)
0.
100.
200.
300.
400.
500.

Figure 30

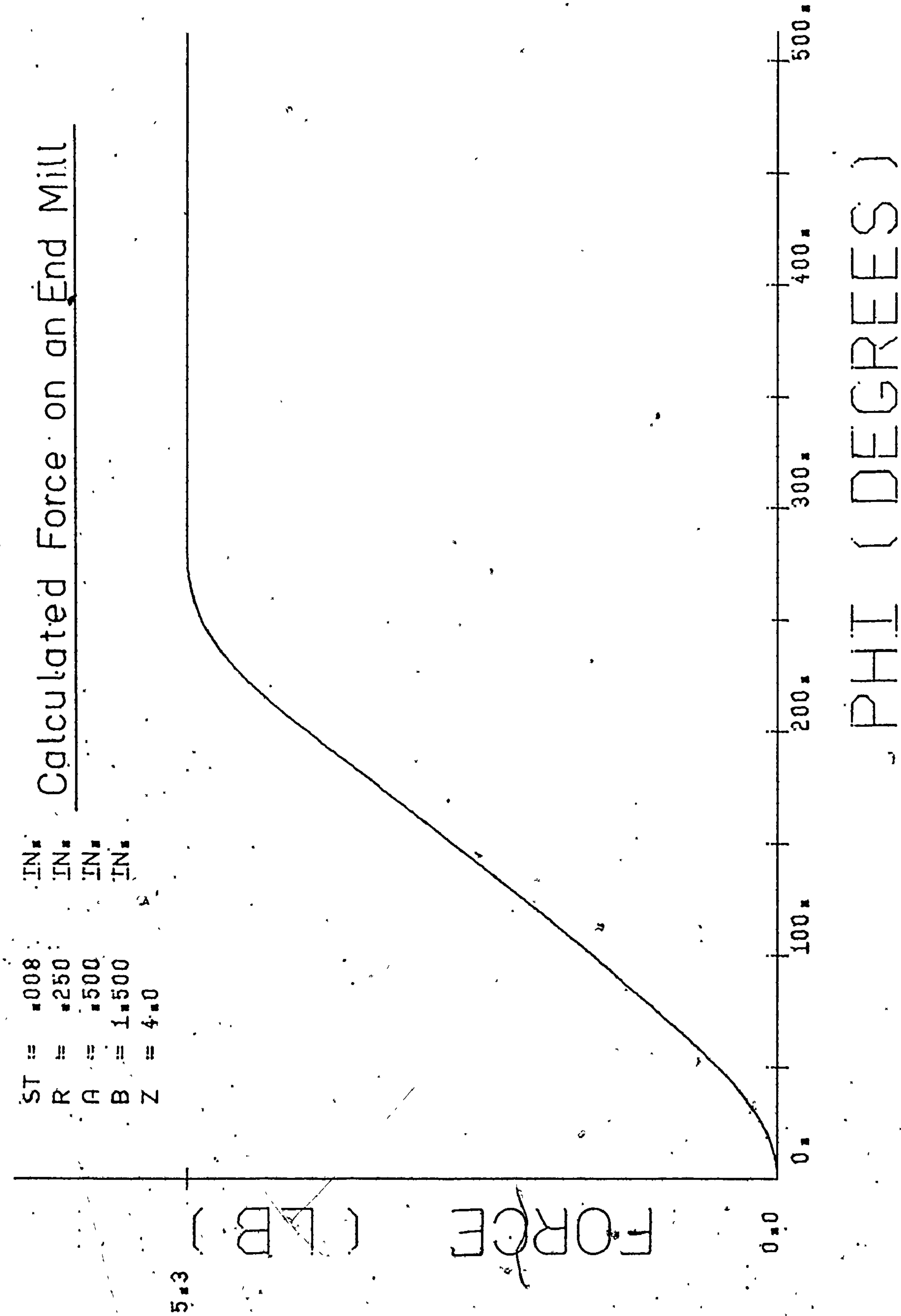

Calculated Force on an End Mill
ST = .008 IN.
R = .250 IN.
A = .500 IN.
B = 1.500 IN.
Z = 4.0
FORCE (LB)
3445.3
0.0
PHI (DEGREES)
0.
100.
200.
300.
400.
500.

Figure 31

zero to a maximum as the cutter advances into the material. To aid in the analysis, a transient depth of cut a' was defined as shown in figure 32. This quantity a' increases from zero to r. To further simplify the problem, a fourth phase was introduced into the cutting process; phase A_1 exists when $0 \leqslant \phi \leqslant \phi_1$ and it decreases in size as the cutter advances into the material. This phase covers the period from $\phi = 0$ to $\phi = \phi_1$ when the tooth begins to cut the workpiece.

Another difference lies in the type of cut; whereas in the steady-state milling model the cut was either of type I or type II, here the cut changes from type II to type I as the cutter advances into the material.

Again, the fundamental relations governing the cutting force are:

$$dF_t = K \cdot S_t \cdot \sin\phi \, dy$$

$$dF_{t_x} = K \cdot S_t \cdot \sin\phi \cos\phi \, dy$$

$$dF_{t_y} = K \cdot S_t \cdot \sin^2\phi \, dy$$

Again, these equations were integrated over the entire length of the tooth, given the limitations of the four phases of the cut. The resulting expressions differ in form from the steady-state model only in phase A.

Phase A_1 $\quad 0 \leqslant \phi \leqslant \phi_1$

$$F_t = F_{t_x} = F_{t_y} = 0$$

Phase A $\quad \phi_1 \leqslant \phi \leqslant \phi_1 + \delta$

$$F_t = F_u \left\{ \cos\phi_1 - \cos\phi \right\}$$

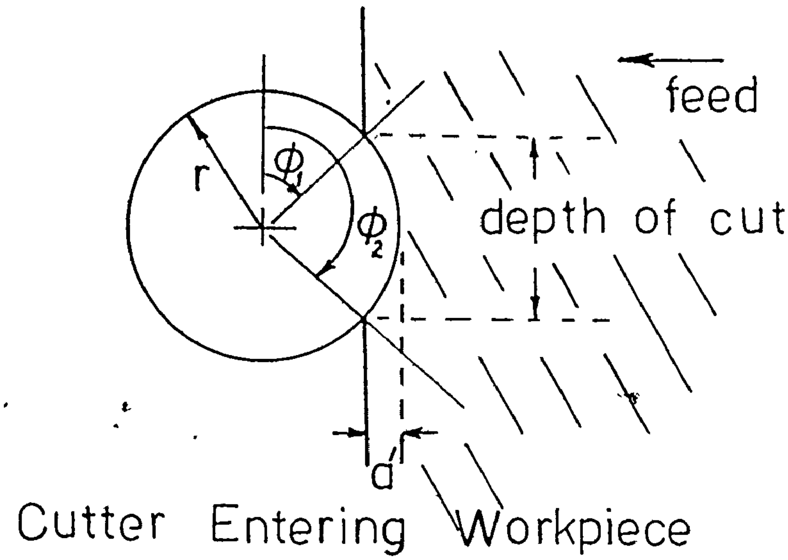
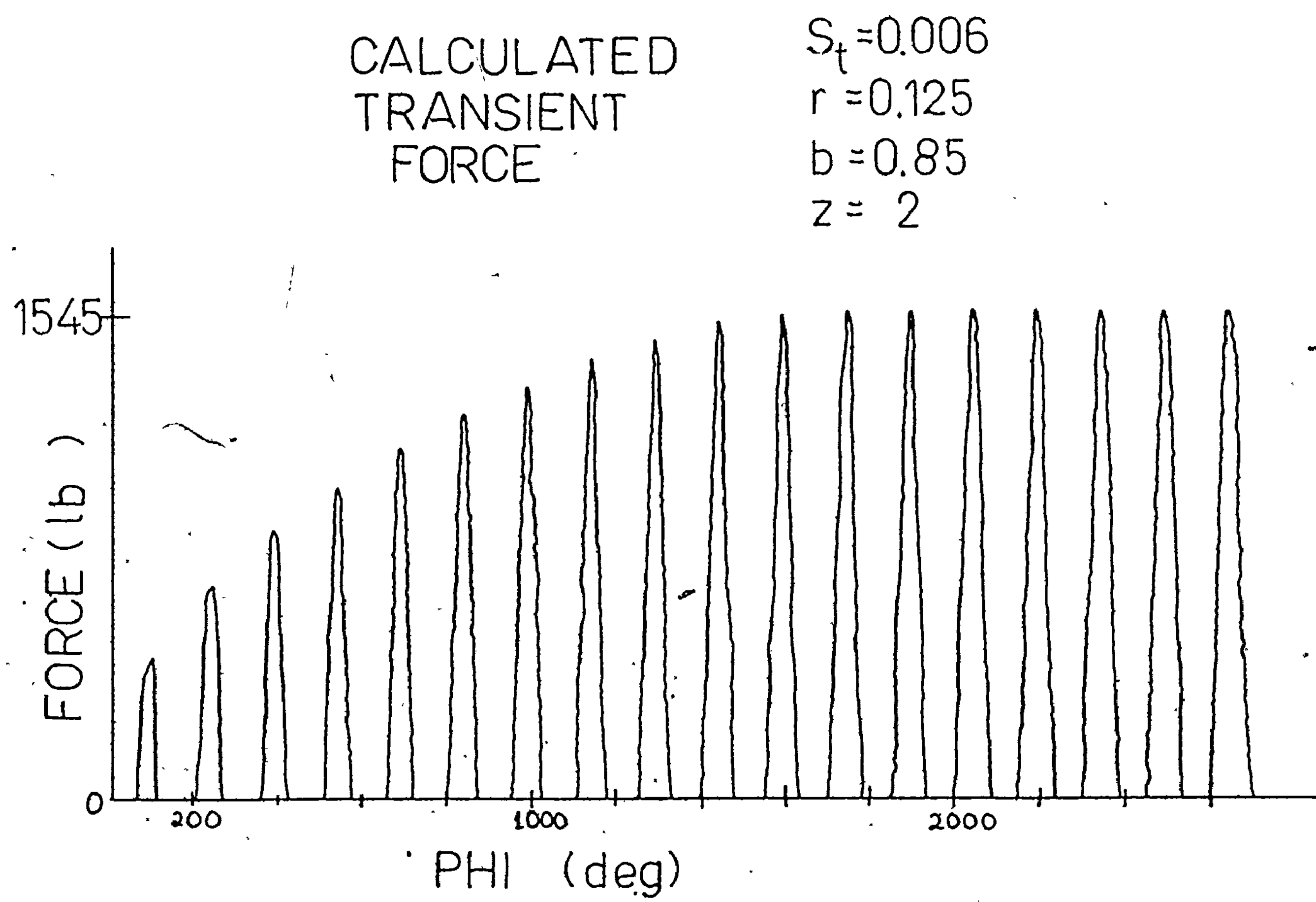

feed
r
ϕ_1
ϕ_2
depth of cut
d
Cutter Entering Workpiece

Figure 32


CALCULATED TRANSIENT FORCE
S_t = 0.006
r = 0.125
b = 0.85
z = 2
1545
0
200
1000
2000
FORCE (lb)
PHI (deg)

$$F_{t_x} = -\frac{F_u}{2}\left\{\sin^2\phi - \sin^2\phi_1\right\}$$

$$F_t = \frac{F_u}{2}\left\{\phi - \phi_1 - \frac{\sin 2\phi}{2} + \frac{\sin 2\phi_1}{2}\right\}$$

where: F_t = tangential force on a tooth

F_{t_x} & F_{t_y} = components of F_t

$$\phi_1 = \arcsin\left[\frac{r-a}{r}\right]$$

$$\phi_a = \phi_2 = \pi - \phi_1$$

$$\delta = \frac{b \cdot \tan\beta}{r}$$

$$F_u = \frac{K \cdot S_t \cdot r}{\tan\beta}$$

These equations are listed as for a type I cut. As in the steady-state model, the changes for a type II cut involve changes in the limits of the phases and a constant force in phase B. These tooth forces may then be combined to determine the total force and torque in the same manner as in the steady-state model. The transient force and torque thus differ from the steady-state model only in the fact that, as the depth of cut increases with time, the situation changes from type II to type I and the magnitude of the force and torque increase, as may be seen in figure 33.

The transient case just discussed represents an important part of any model used in the simulation of A/C milling. In optimizing the milling process, the non-cutting time must be minimized; therefore, when not in contact with a workpiece, the cutter should be moving at the maximum permissable feed. Upon encountering a workpiece then, the feed per tooth S_t will be very high, generating very high cutting forces. The

Figure 34

Comparison of Actual & Computed Forces

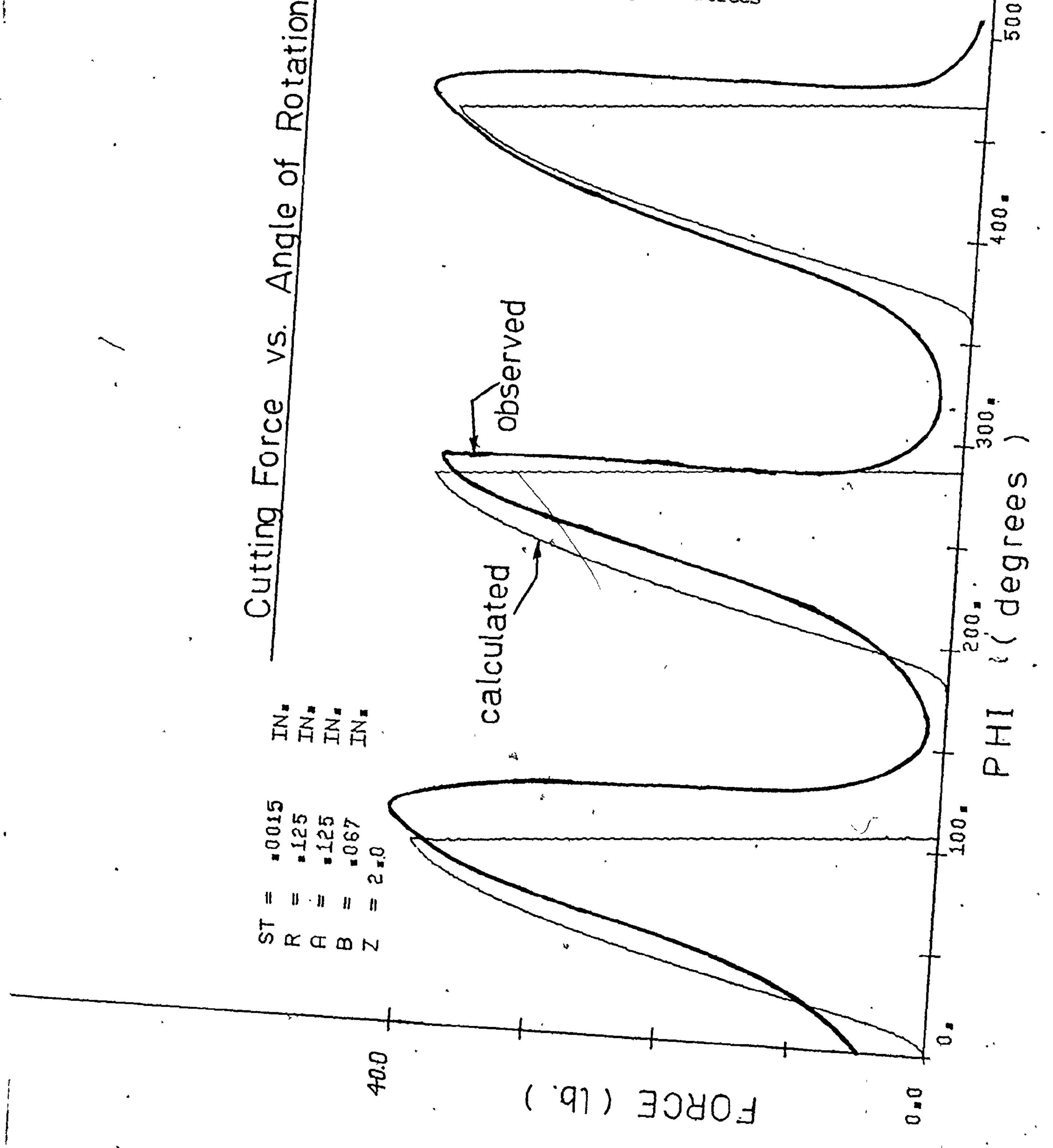


Figure 35

Comparison of Actual & Computed Forces

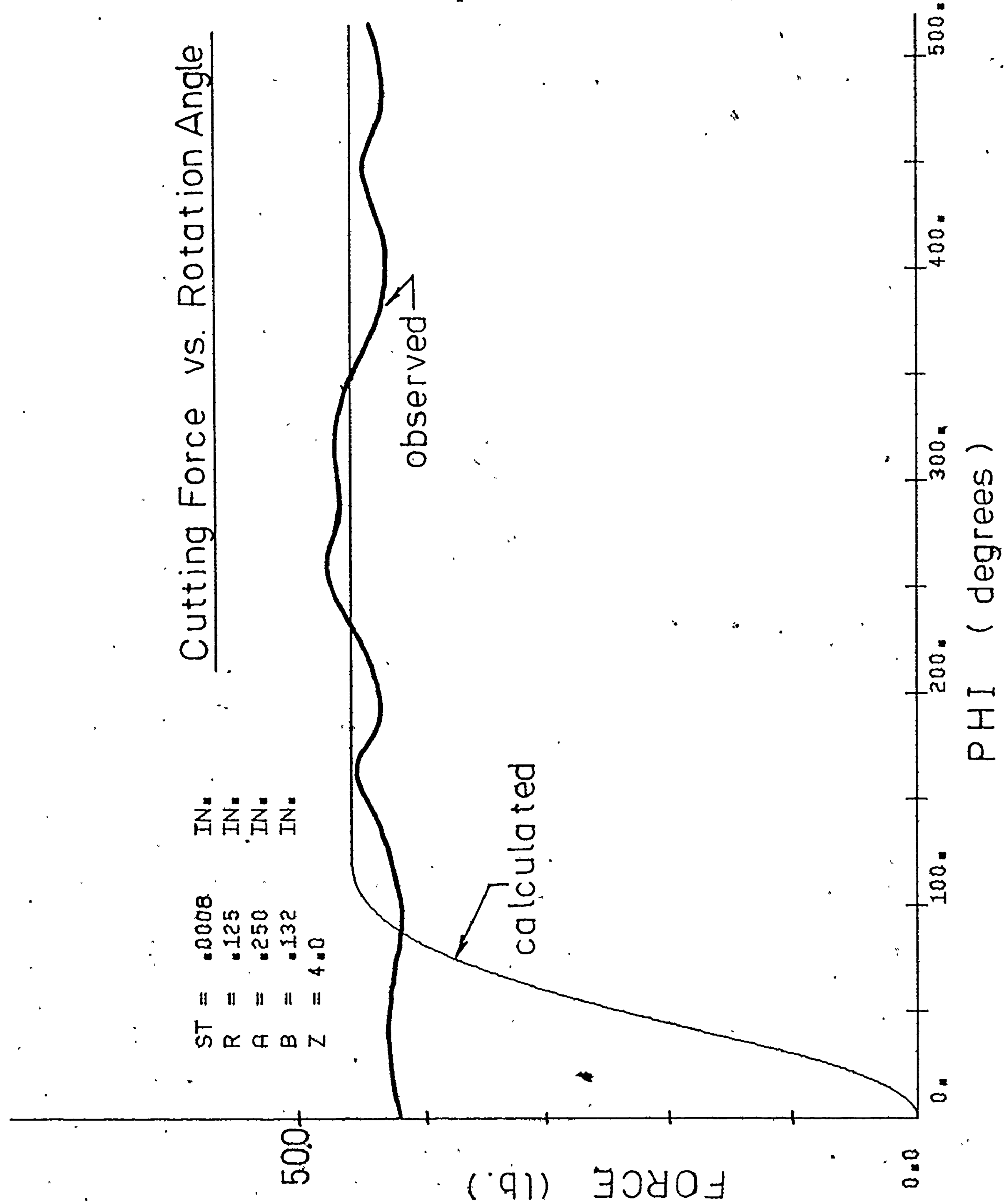


Figure 36
Comparison of Actual & Computed Forces

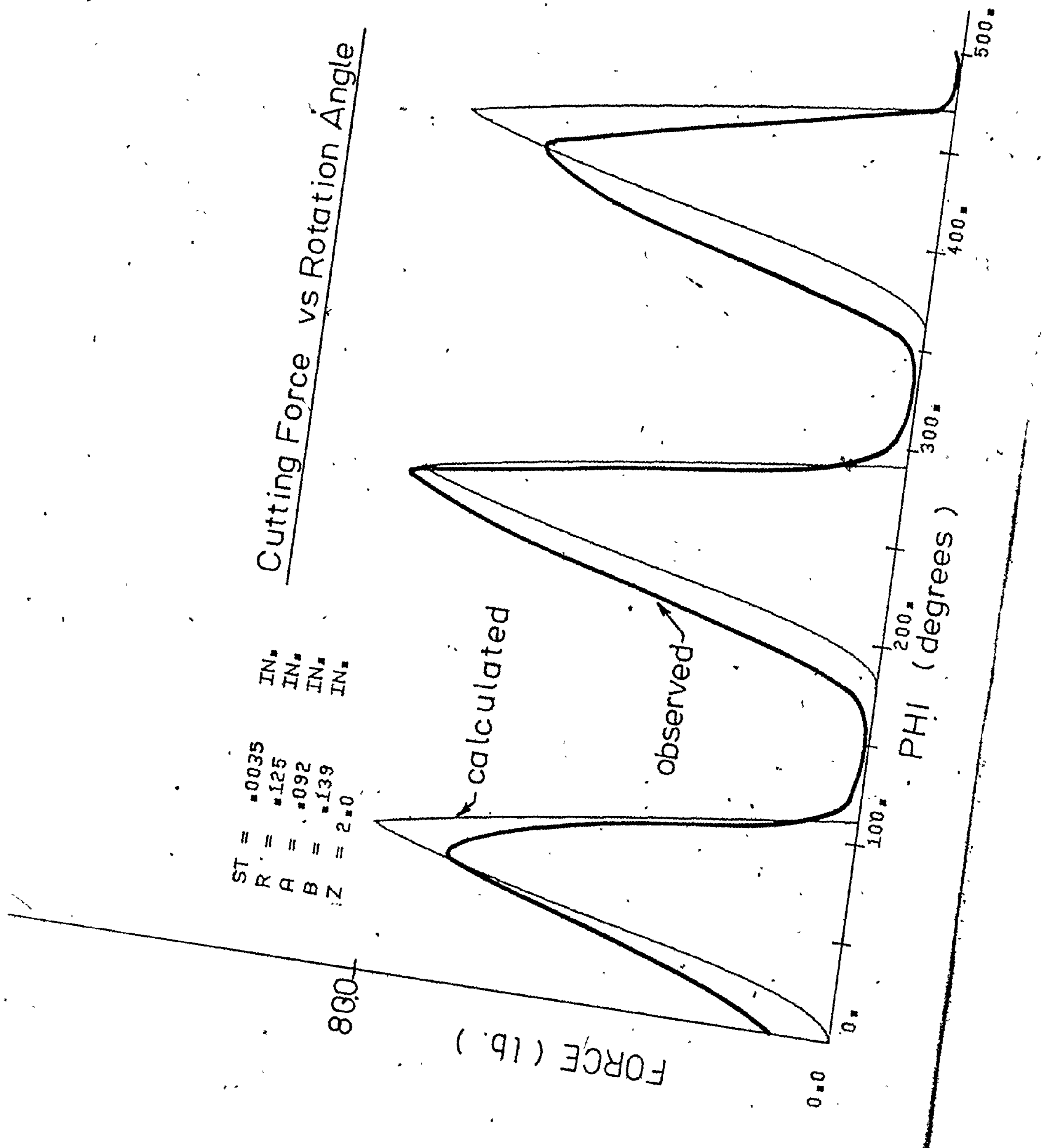
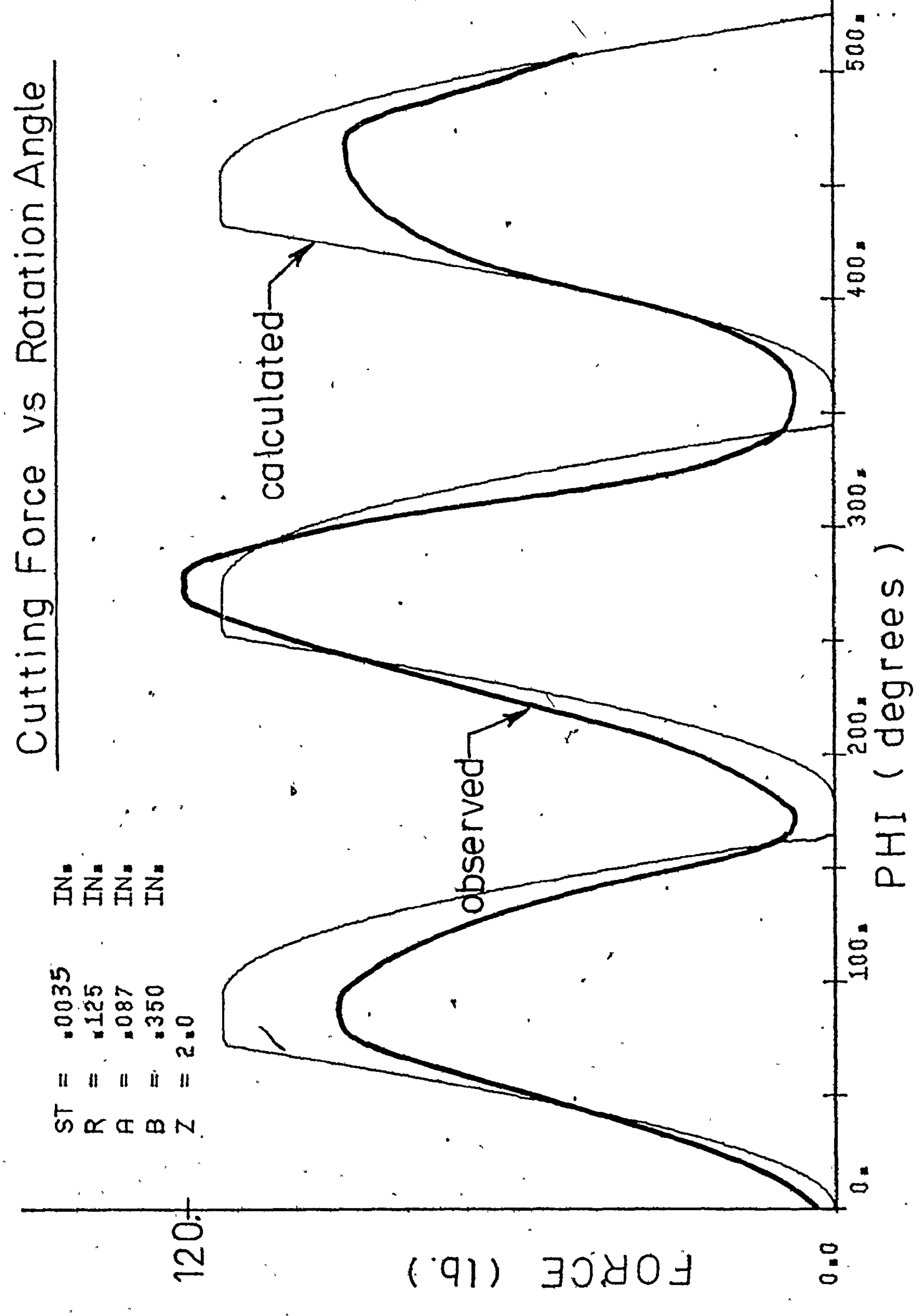


Figure 37

Comparison of Actual & Computed Forces



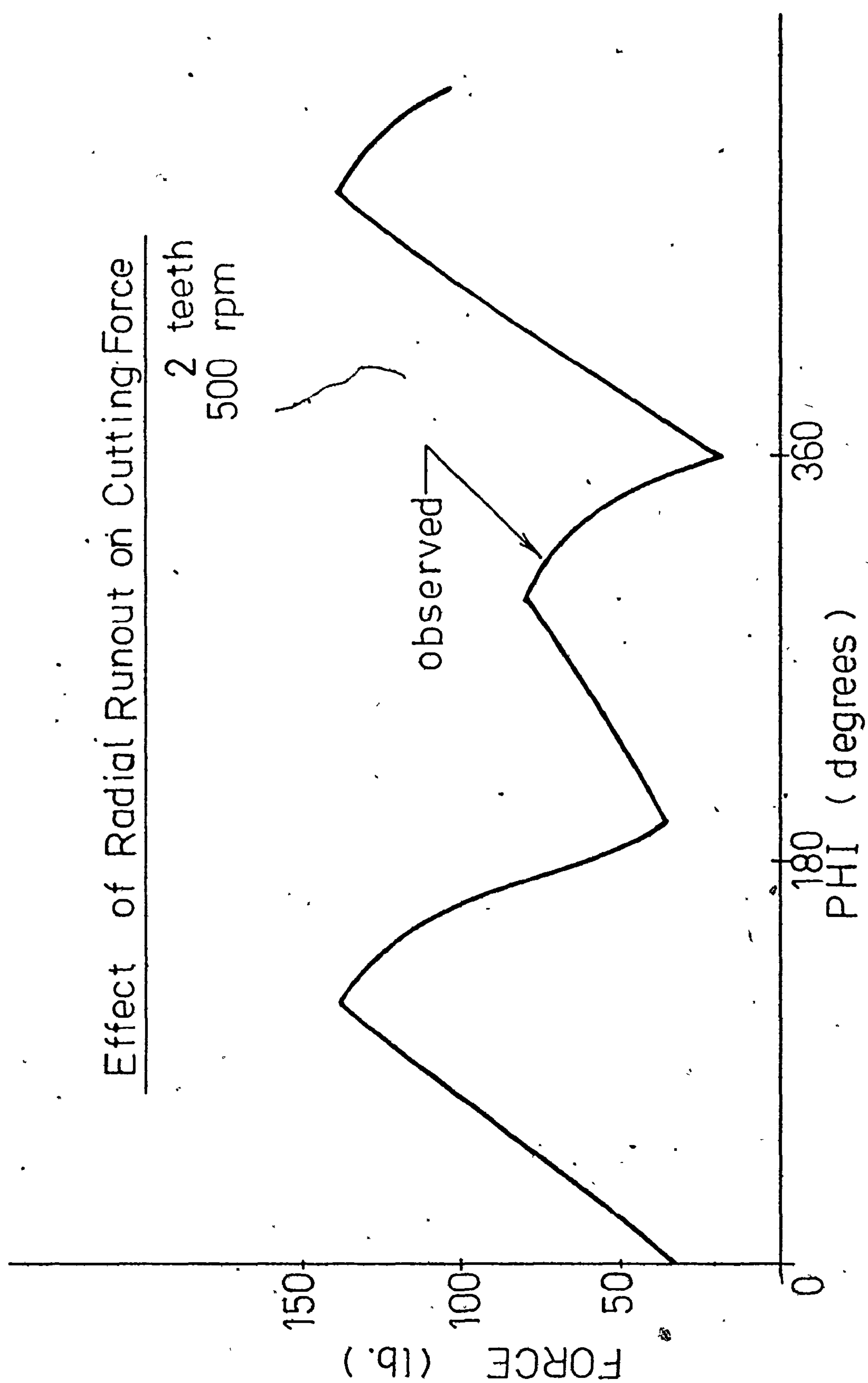

Effect of Radial Runout on Cutting Force
2 teeth
500 rpm
observed
FORCE (lb.)
150
100
50
0
180
360
PHI (degrees)

Figure 38

A/C control system must first recognize the excessive force and then promptly react so as to reduce the force. Thus, the response of an A/C system to such a rapid traverse situation is surely one important measure of its effectiveness.

Using the A/C transducer described in chapter three and a high speed U-V chart recorder, the two force signals were recorded while cutting steel under various conditions. Figures 34 to 37 show the comparison between the derived total force and the calculated total force. Figure 37 illustrates a type II cut wherein the actual force does indicate a constant force in phase B. Figure 38 illustrates the effect of radial runout of the end mill on the total force. This is of course quite reasonable because it really means that each tooth cuts a chip of different width.

Chapter 5

N/C Control Theory

From a control systems point of view, adaptive control represents merely a control loop containing both feed-forward and feedback elements. The loop examines certain process variables such as force and torque, perceives an error between the desired and actual values of these variables, and reacts according to some preset policy to reduce this error. However, the adaptive controller often must work in cooperation with other control functions. In the case of A/C milling, the adaptive controller has a supervisory role over the N/C controller. The task of the latter is to produce a dimensionally correct part while the task of the former is to improve the efficiency or the tolerances of the latter. It is therefore obvious that both control functions must be compatible.

In adding an adaptive control loop to the present CNC system, an attempt was made to ensure this compatibility. The complete N/C control function was analysed before the addition of the A/C loop. Both loops were simulated by a digital computer program in order to more easily observe the effects of changes in either cutting or controlling conditions. This chapter discusses the simulation of the N/C loop.

The first step towards simulation involved an evaluation of the N/C loop by classical control theory. Figure 39 illustrates the block diagram of any axis of motion of the system. The three axes of motion may be considered as parallel systems, each one complete in itself, with

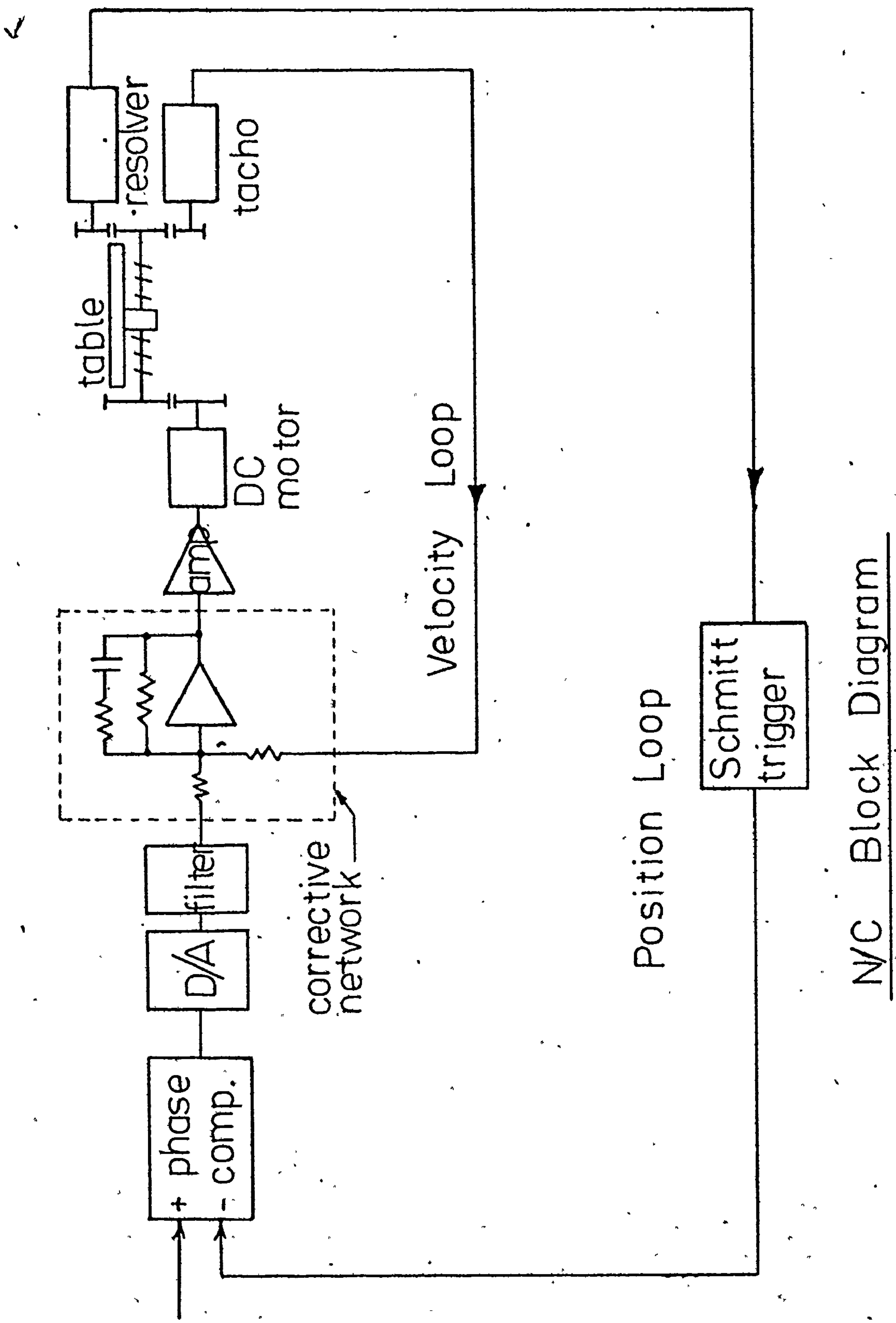

resolver
tacho
table
DC motor
Velocity Loop
amp
corrective network
filter
D/A
+ phase
- comp.
Position Loop
Schmitt trigger
N/C Block Diagram

Figure 39

each receiving its input from the minicomputer. Input to the system is in the form of a distance command. Subtracted from this distance input is the actual position of the machine slide given by the resolver, a position feedback device connected to the leadscrew. The subtraction occurs in a phase comparator which produces a control error proportional to the phase difference between command and feedback. The control error is converted into an analog signal in the D/A converter, and filtered to reduce the noise content of the signal. The correcting circuitry acts as a variable gain amplifier, giving a gain which decreases with frequency; this provides the velocity loop with a fast initial response without serious overshoot problems. The correcting circuit also acts as a summing junction for the velocity loop. It is at this point that the actual slide velocity indicated by the tachogenerator is compared with the command. The output of this circuitry is then amplified to the level required to excite the rotor of the D.C. motor so as to control the desired axis of motion. The servomotor is rated at 54.4 inches-pounds of torque which may not be sufficient to overcome the inertia of the table or slide when loaded with a large workpiece. Thus, the motor turns the ball bearing leadscrew through a transmission ratio of 2.5:1. Both the resolver and tachogenerator then are connected to the leadscrew through a 1:2.5 gear ratio, giving both feedback devices the same speed as the servomotor. The output of the resolver, being of a sinusoidal nature, is converted by a Schmidt trigger into a pulse train.

In analysing this N/C control system, several simplifications were made. First, the phase comparator was considered as a simple summing point having no significant frequency characteristics. Secondly, the filter had a time constant of $T = 0.4$ milliseconds which was considered negligible in comparison with time constants elsewhere in the system. A third assumption was that the Schmidt trigger which had a gain of unity had no significant time delay or frequency characteristic. Finally, in the mechanical part of the system, backlashes and dead zones were neglected. The final block diagram considered is shown in figure 40.

The next step in the classical analysis was the determination of the transfer functions of each block.

D/A Converter

This has a simple gain of 4.5 volt per cycle of the phase comparator wherein one cycle may be equated to 0.1 inch of slide motion. Thus, the gain is 45 volts/inch.

Filter

This has a simple gain of unity.

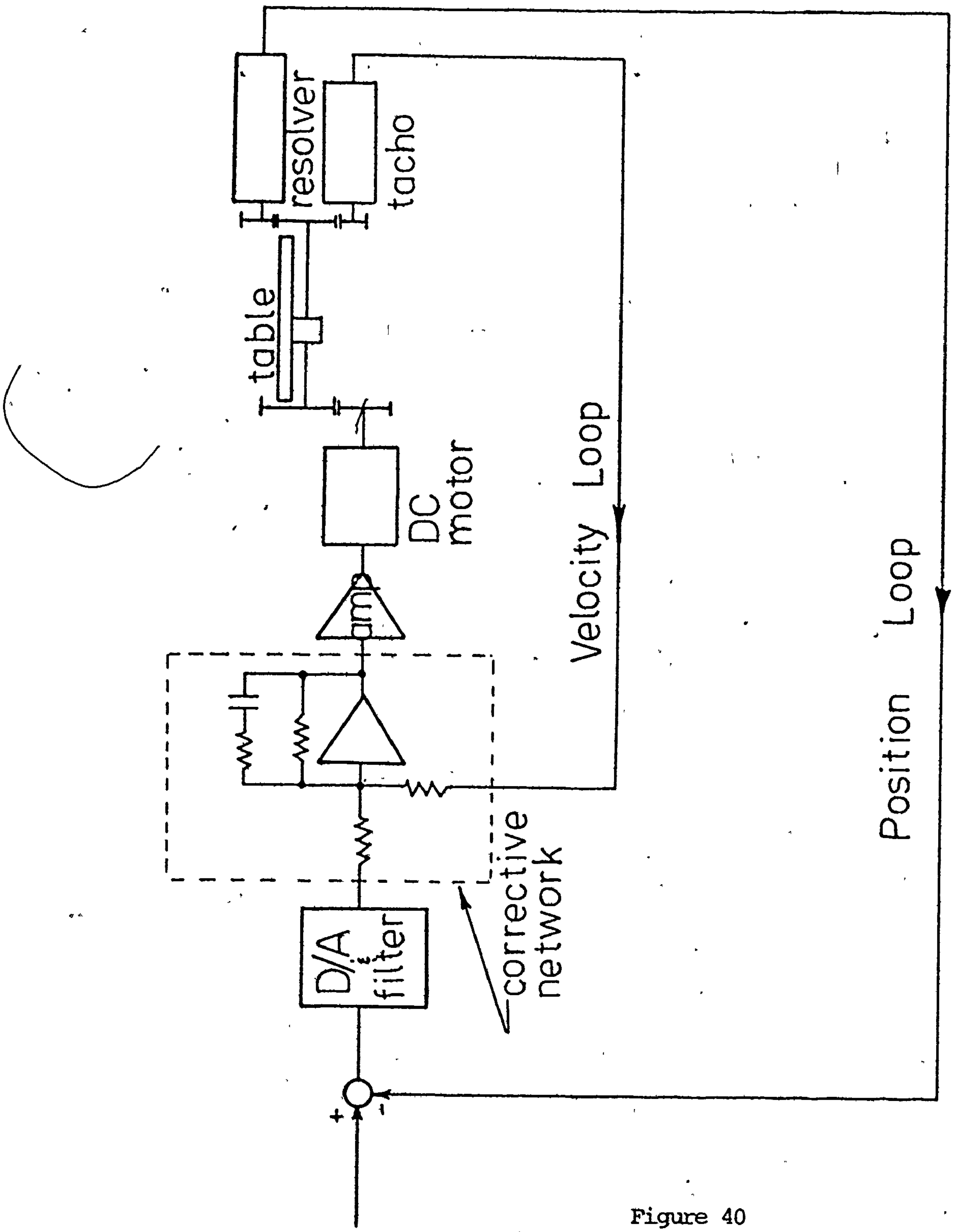
Correcting Network

To aid in the analysis of the closing of the velocity loop, the correcting network is modified as shown in figure 41. This will then alter the gain of the velocity feedback loop by a factor of R_1/R_4. Given this change, the transfer function of the correcting circuitry becomes:

$$T(s) = 2.178 \left\{ \frac{s + 45.45}{s + 0.45} \right\}$$

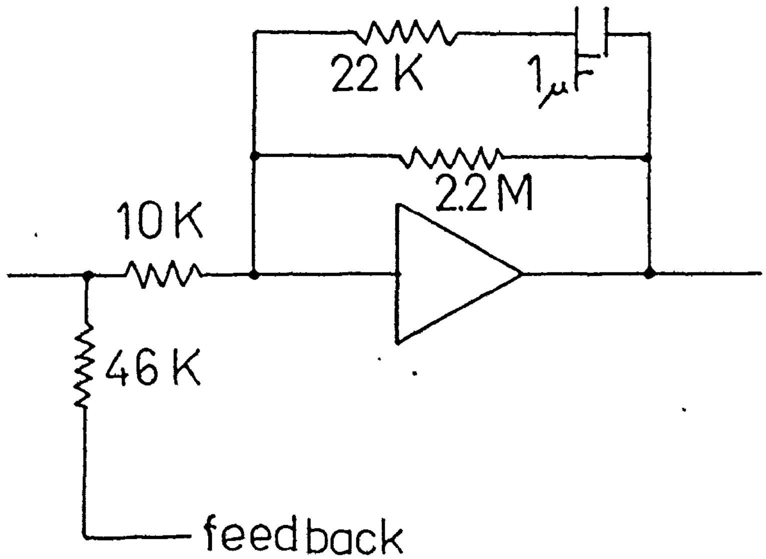
Amplifier

This power amplifier has a very flat frequency characteristic with a gain of 10.0.



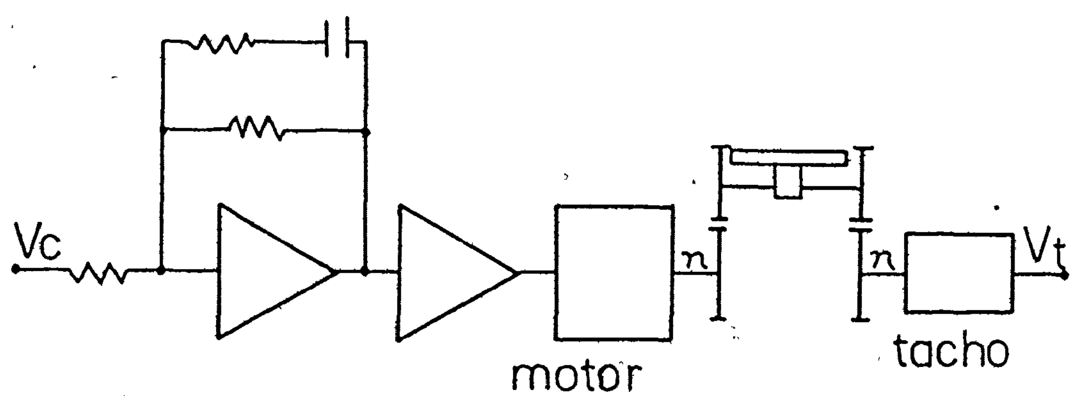
Simplified N/C Block Diagram

Figure 40



Alteration to Corrective Network

Figure 41



Open Velocity Loop

Figure 42

D.C. Motor

Neglecting the inductance of the armature coil, the general expression for the transfer function of a rotor-excited D.C. motor is:

$$T\ (s) = \frac{K}{1 + s\tau} \qquad [18]$$

The value of the motor time constant was measured to be 32 milliseconds and the gain was stated by the manufacturer to be 0.35 rev/sec/volt. The transfer function is then:

$$T\ (s) = \frac{0.35}{1 + 0.32\ s}$$

Gear Transmission

From the motor output shaft to either resolver or tacho input, the gain is unity. However, the relation between actual table feed in inches per second and motor speed in revolutions per second is determined by the gear ratio of 2.5:1 and the leadscrew pitch of 0.25. Therefore, table feed is 1/10 of the motor speed.

Tachogenerator

The gain of the tacho is 1.24 volts/rev/sec and the gain of the modified feedback loop is $R_1/R_4 = 0.2174$.

Resolver

The position feedback device is basically an integrator, determining a distance output from a shaft rotation. As mentioned in the motor section, the transmission ratio between resolver and table is 10:1. Therefore:

$$T\ (s) = \frac{1}{10\ s}$$

Proceeding with the classical control theory approach, each loop within the N/C controller was systematically analysed. First, the open velocity loop as shown in figure 42 was considered. Given the transfer functions of each element within the loop, the transfer function of the open loop is:

$$T\ (s) = \frac{V_t}{V_c}\ (s)$$

$$= 954.8 \ \frac{(1 + 0.022\ s)}{(1 + 2.222\ s)\ (1 + 0.032\ s)}$$

Examining the step input response of the open loop gives a result which may easily be verified experimentally. The response of the open loop to a step input of 1.0 volt is:

$$V_t\ (s) = 954.8 \ \frac{(1 + 0.022\ s)}{(1 + 2.222\ s)\ (1 + 0.032\ s)} \cdot \frac{1}{s}$$

In the time domain, the response is:

$$V_t\ (t) = 954.8 \left\{ 1.0 + 0.0045\ e^{-t/.032} - 1.0045\ e^{-t/2.222} \right\}$$

Therefore, as can be seen in figure 43, the significant time constant is about 2.3 seconds. The same response as modelled by the state-space analysis method is seen to be identical. (This method is discussed later in this chapter.) The open velocity loop response was examined experimentally by disconnecting the velocity feedback and applying a step voltage to the correcting network. The time constant observed was approximately 2.8 seconds. The gain was 955.5.

Closing the velocity loop as shown in figure 44, the transfer function of the loop becomes:

$$T\ (s) = 295.45 \left\{ \frac{S + 45.45}{s^2 + 95.94\ s + 2933.66} \right\}$$

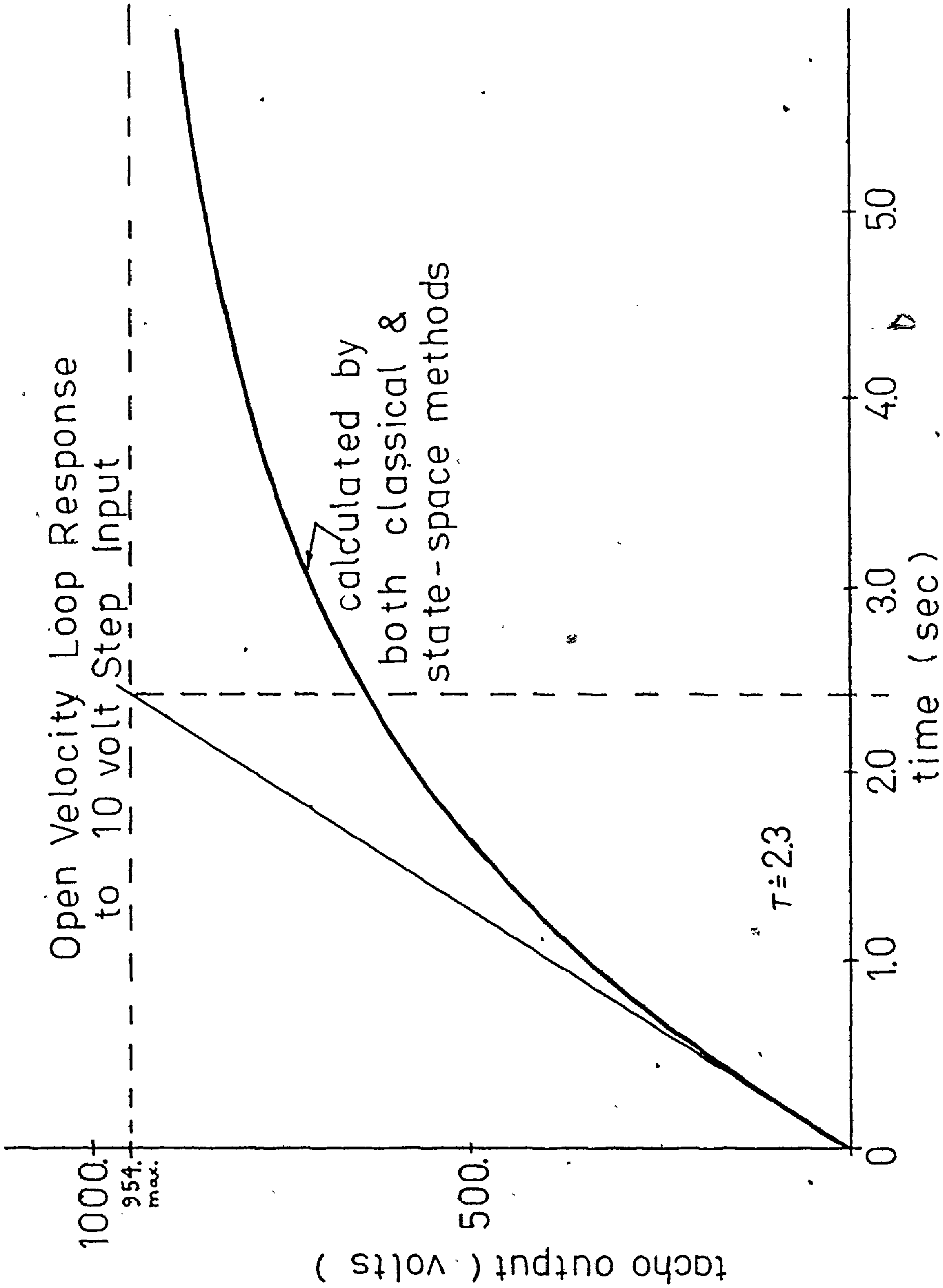

Open Velocity Loop Response
to 10 volt Step Input
calculated by
both classical &
state-space methods
τ ≐ 2.3
time (sec)
tacho output (volts)
1000.
954.
max.
500.
0
1.0
2.0
3.0
4.0
5.0

Figure 43

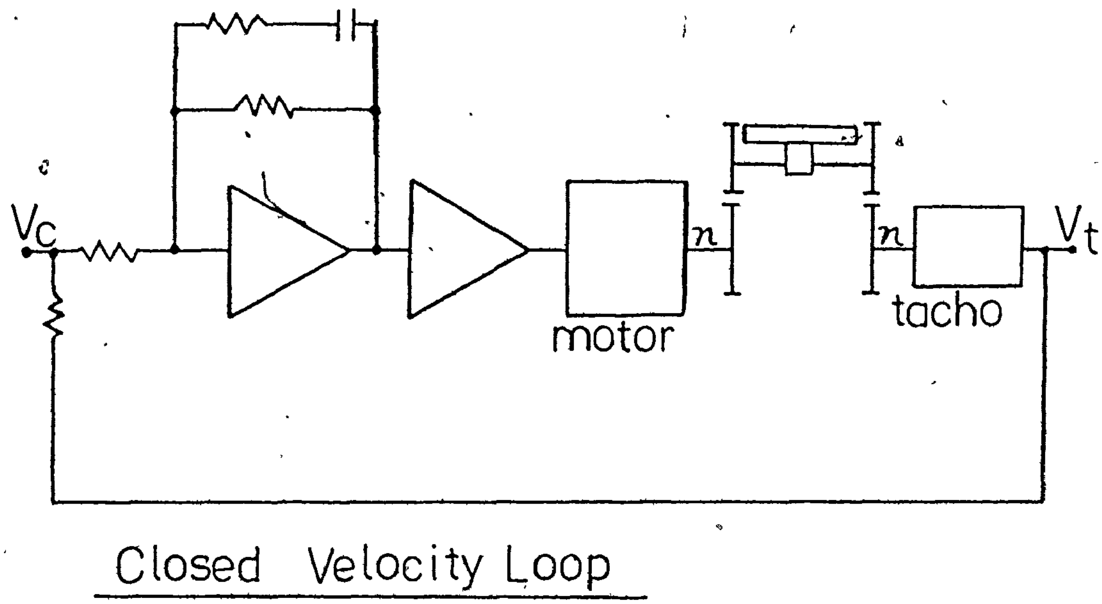


Figure 44

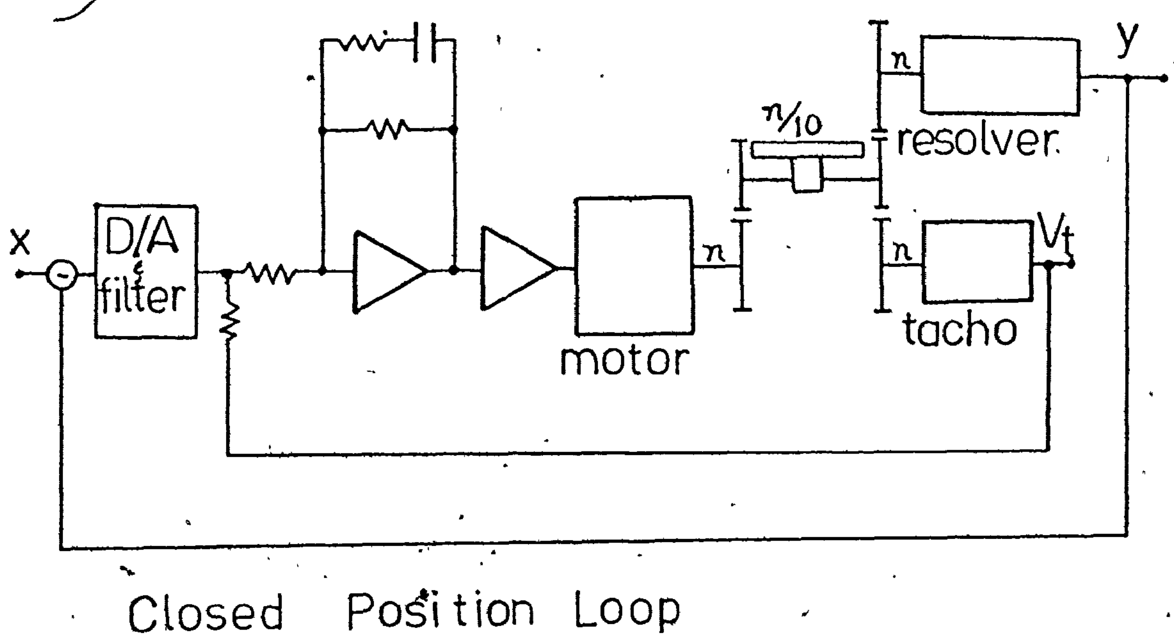
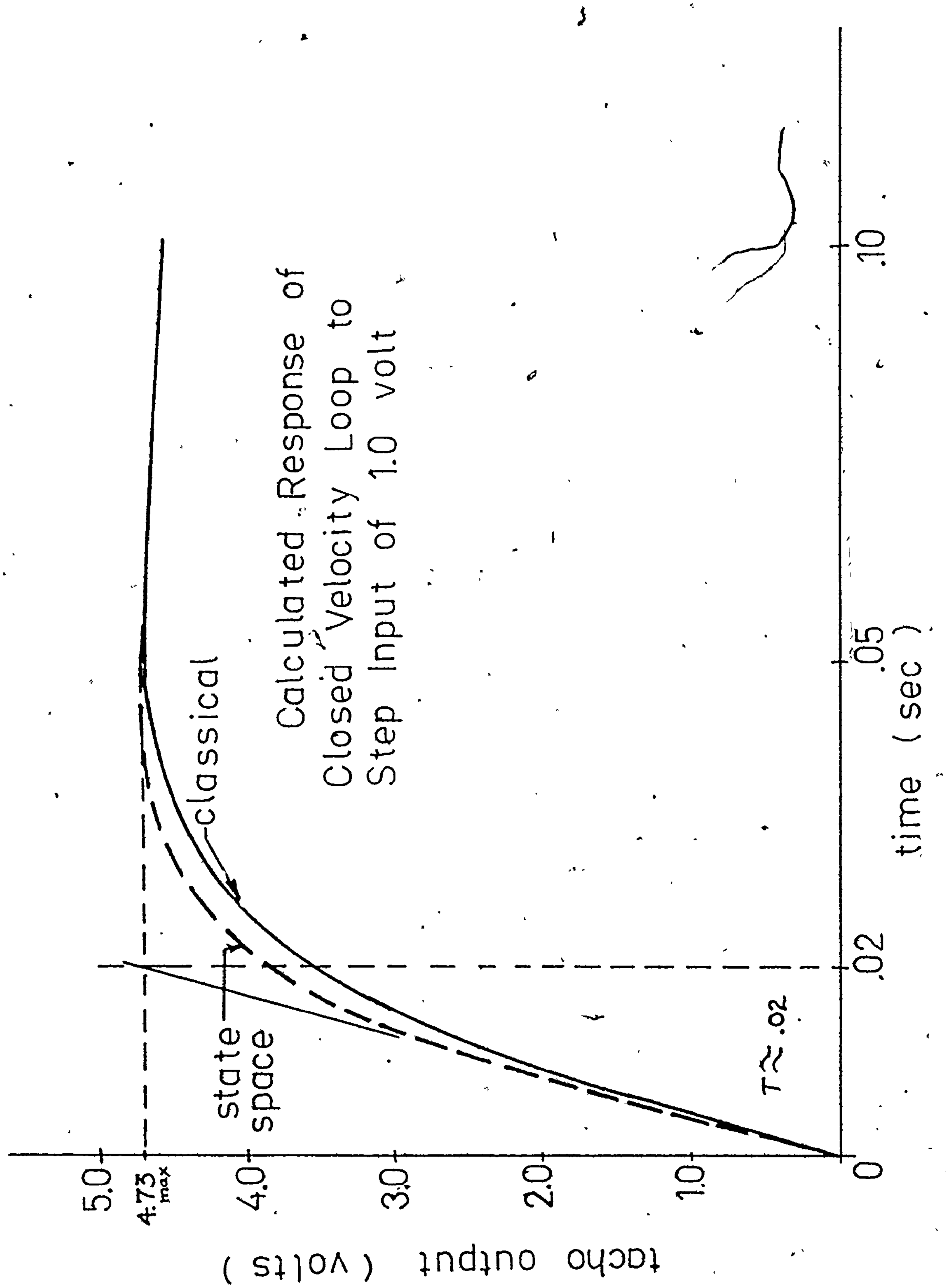


Figure 46



Calculated Response of
Closed Velocity Loop to
Step Input of 1.0 volt
classical
state space
$\tau \approx .02$
tacho output (volts)
5.0
4.73 max
4.0
3.0
2.0
1.0
0
.02
.05
.10
time (sec)

Figure 45

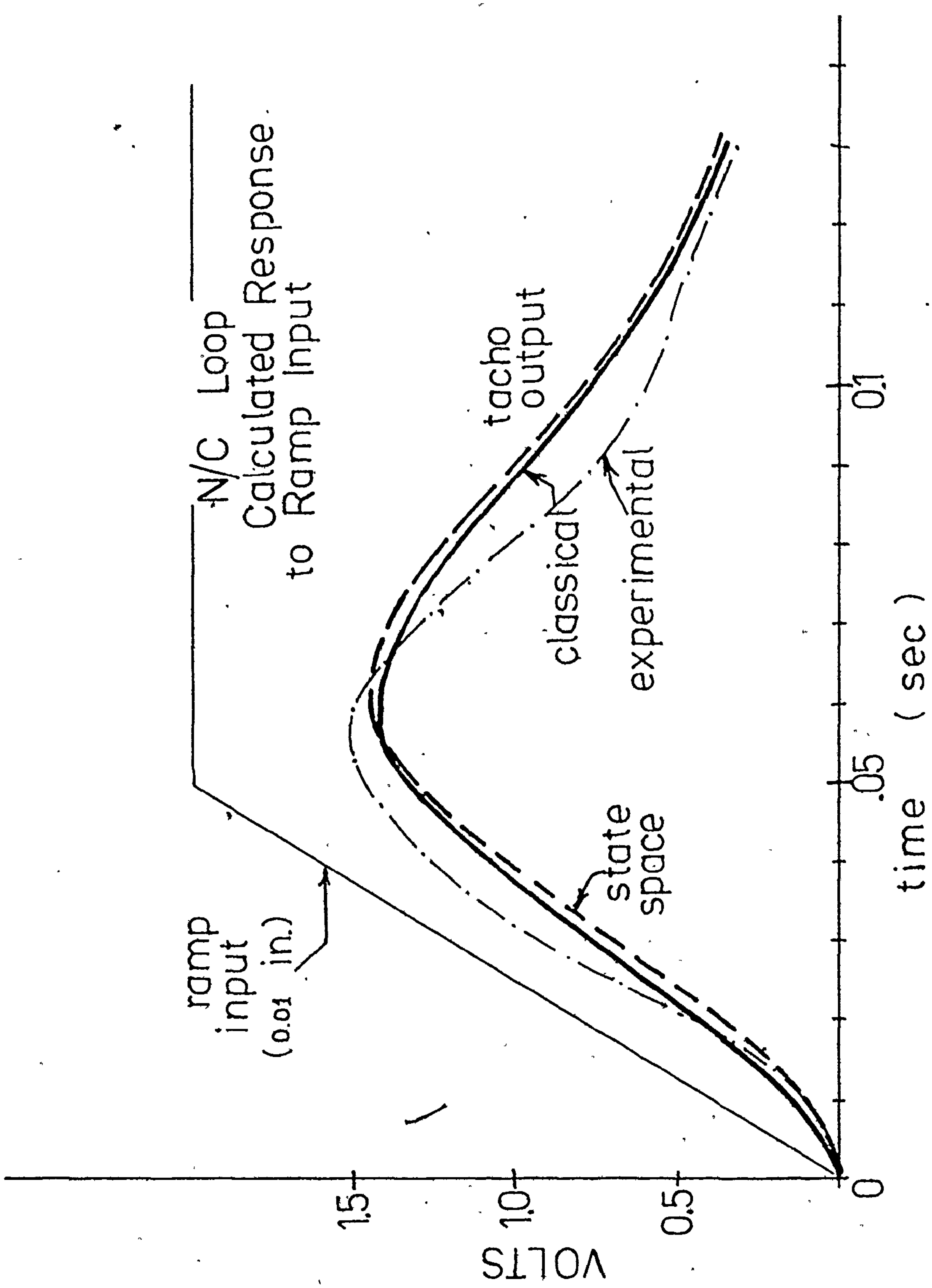

N/C Loop
Calculated Response
to Ramp Input
ramp input (0.01 in.)
state space
classical
experimental
tacho output
time (sec)
VOLTS
0
0.5
1.0
1.5
.05
0.1

Figure 48

was examined. As seen from the resolver:

$$R(t) = 1.00026 - 1.0238\, e^{-t/.0519} + 0.02369\, e^{-t/.0261} \cdot \cos(32.537\, t) - 0.577\, e^{-t/.0261} \sin(32.537\, t)$$

and, as seen from the tacho:

$$V_t(t) = 244.74 \left\{ e^{-t/.0519} - e^{-t/.0261} \cos(32.537\, t) + 1.083\, e^{-t/.0261} \sin(32.537\, t) \right\}$$

These responses are shown in figure 47 where it may be noticed that the overall time constant of the system seems to be 80 msec. This certainly gives some insight into the response of the system. However, the actual N/C system can supply a distance command only over a finite time. Of course a step input command equal to one basic length unit (0.0001 inch) could be applied, but in such a case, factors not included in the model such as static friction in the guideways and windup in the leadscrew may considerably distort the response.

An input which the actual system is capable of supplying is a ramp input. The ramp function shown in figure 48 was applied during actual tests. The function is described by:

$$f(t) = 0.2\, t \text{ inch for } 0 \leqslant t \leqslant T$$

and

$$f(t) = .01 \text{ inch for } T < t < \infty$$

The calculated response of the system to the ramp input as seen from the tacho output is then:

$$V_t(t) = 2.48 - 2.54\, e^{-t/.0519} + .0595\, e^{-t/.0261} \cdot \cos 32.53\, t - 1.432\, e^{-t/.0261} \sin 32.537\, t - \left\{ 2.48 - 2.54\, e^{-(t-.05)/.0519} + .0595\, e^{-(t-.05)/.0261} \cos 32.53\,(t-.05) - 1.432\, e^{-(t-.05)/.0261} \right.$$

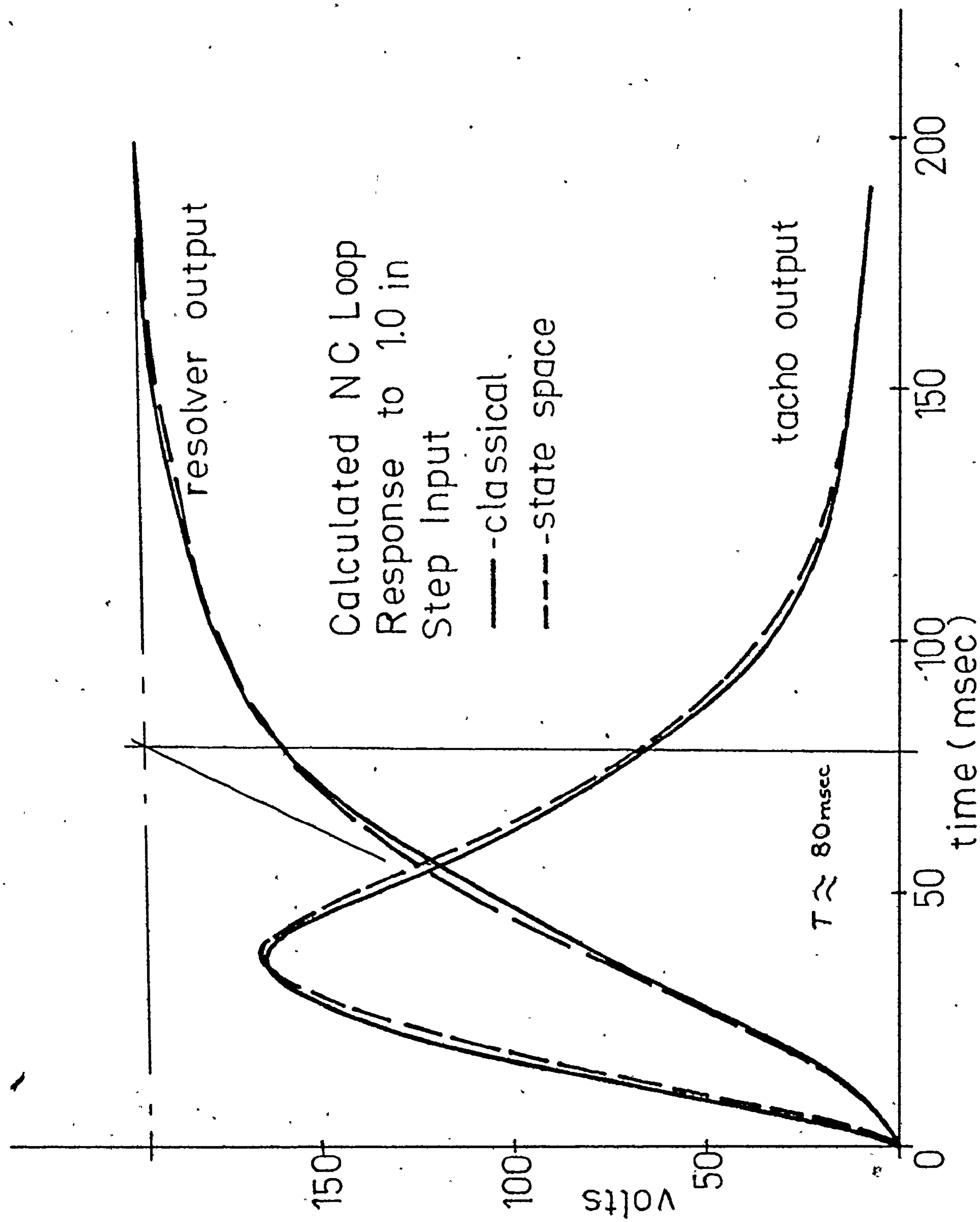

resolver output
Calculated N C Loop
Response to 1.0 in
Step Input
-classical
-state space
tacho output
τ ≈ 80msec
volts
150
100
50
0
50
100
150
200
time (msec)

Figure 47

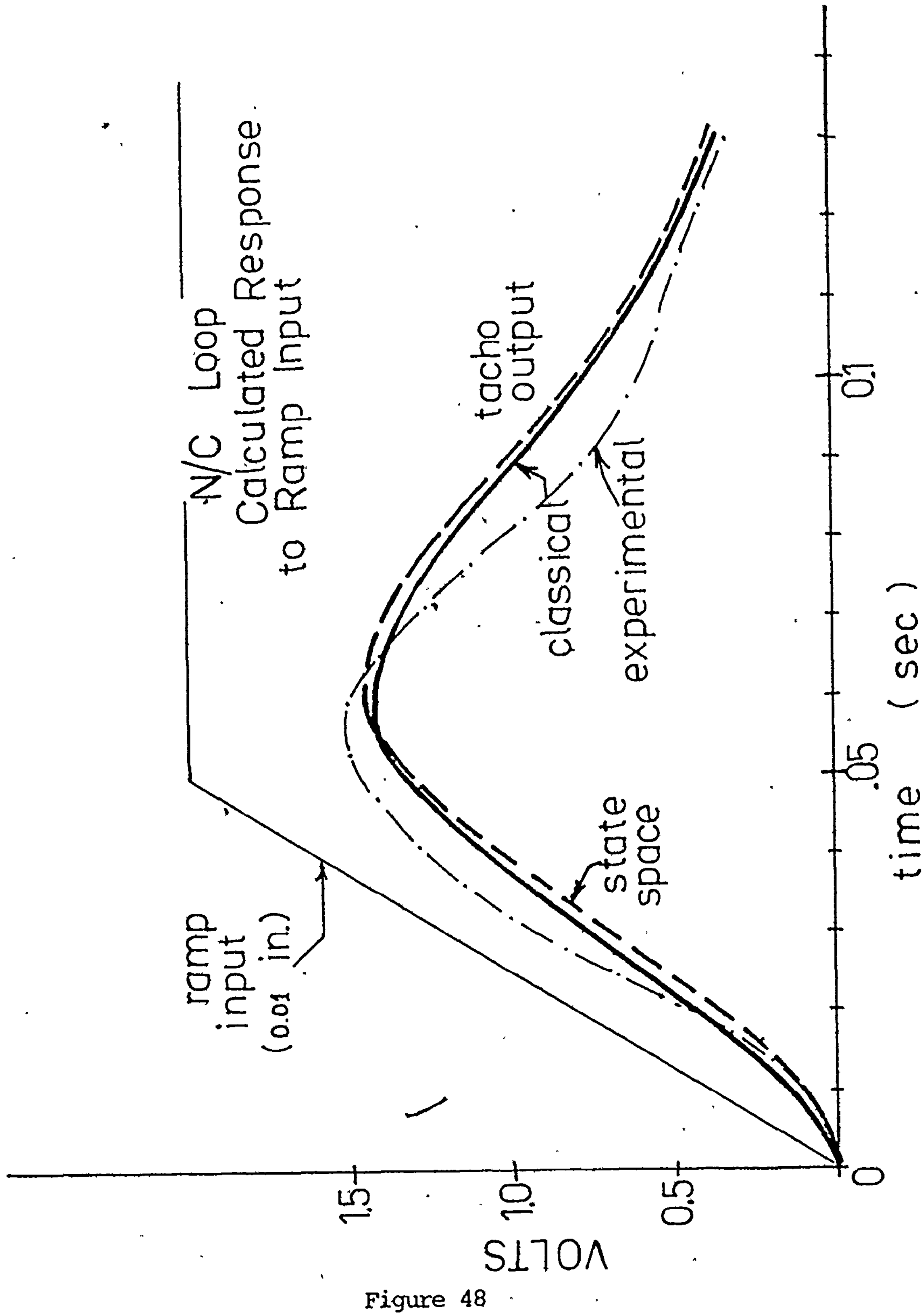

N/C Loop
Calculated Response
to Ramp Input
ramp input (0.01 in.)
state space
classical
experimental
tacho output
VOLTS
1.5
1.0
0.5
0
.05
0.1
time (sec)

Figure 48

$$\sin 32.537\,(t - .05)\Big\}$$

where the bracketed term is zero for all $t < .05$ sec. Since the resolver output was not measured experimentally, only the tacho output is discussed here. The response is plotted in figure 48 along with the equivalent response determined using the state-space method, and the experimental result.

The classical model of the N/C control system was complete with the closing of the position loop. The next step in the model was then the A/C loop. However, the A/C algorithm, described in chapter six, includes a delay of about 10.5 milliseconds between feedrate corrections. The complexity which this delay introduced into the classical calculations made the inevitable tuning of the A/C algorithm seem a formidable task. Thus, a simpler model was developed which was based on the state-space technique.[19]

"In the analysis of a system via the state-space approach, the system is characterized by a set of first-order differential or difference equations that describe its 'state' variables. System analysis and design can be accomplished by solving a set of first-order equations rather than a single, higher-order equation."[7] The solution to a set of first-order equations may be determined rather simply with the use of a digital computer. This would then simplify the task of 'tuning' the A/C loop.

"The state-variable diagram provides a physical picture that is useful in understanding the state-space concept. In addition, the differential equations relating the state variables are easily obtained

[7]Stanley M. Shinners. Modern Control System Theory and Application (Don Mills, June, 1973), p. 62.

by inspection of the diagram. A state-variable diagram consists of integrators, summing devices, and amplifiers. Outputs from the integrators denote the state variables."[8] It is important to note that not all of the state variables will be physically observable or measurable; they may represent intermediate states within the system. This technique is now applied to the block diagram of the N/C loop as shown in figure 40.

Let the initial distance input to the N/C loop be labelled as X1. As in the classical analysis, the correcting network is examined first. The transfer function of the correcting network is used with its input and output labelled as X2 and X3 respectively:

$$\frac{X3}{X2} = 2.178\left[\frac{s + 45.45}{s + 0.45}\right]$$

Proceeding with the technique described by Shinners[19], the numerator and denominator are divided by s.

$$\frac{X3}{X2} = 2.178\left[\frac{1 + 45.45\ s^{-1}}{1 + 0.45\ s^{-1}}\right] \qquad \text{—— (1)}$$

The function E (s) is then defined:

$$E\ (s) = \frac{X2}{(1 + 0.45\ s^{-1})}$$

Equation 1 may then be re-written:

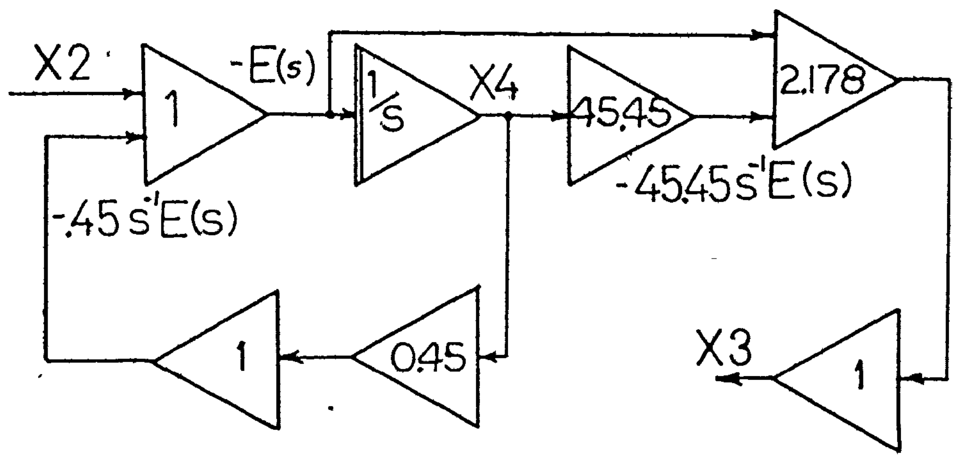
$$X3 = 2.178\ (1 + 45.45\ s^{-1})\ E\ (s)$$

Then using the relation

$$E\ (s) = X2 - 0.45\ s^{-1}\ E\ (s)$$

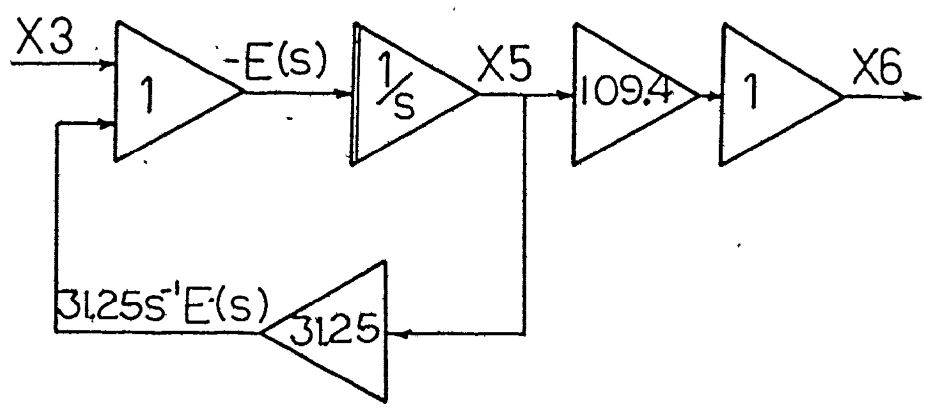
the state variable diagram may be drawn as in figure 49. Note that since the "outputs from the integrators denote state variables", a new state variable X4 is defined. Therefore, the correcting network may be defined by:

[8] Ibid, p. 69.



STATE SPACE DIAGRAM-CORRECTING NETWORK

Figure 49



State Space Diagram — Motor & Amp.

Figure 50

$$\dot{X4} = X2 - 0.45\,X4$$

$$X3 = -\,2.178\,\dot{X4} - 98.99\,X4$$

or $$X3 = -\,2.178\,X2 - 98.01\,X4$$

Proceeding to the next elements of the block diagram, the amplifier and the D.C. motor are combined. Denoting the output of the motor as X6,

$$\frac{X6}{X3} = \frac{3.5}{1 + 0.032\,s}$$

Dividing numerator and denominator by s,

$$\frac{X6}{X3} = \frac{3.5\,s^{-1}}{0.032 + s^{-1}} = \frac{109.4\,s^{-1}}{1 + 31.25\,s^{-1}} \qquad (2)$$

The function E (s) is then defined

$$E\,(s) = \frac{X3}{(1 + 31.25\,s^{-1})}$$

Equation 2 may be re-written

$$X6 = 109.4\,s^{-1}\,E\,(s)$$

Then, using the relation

$$E\,(s) = X3 - 31.25\,s^{-1}\,E\,(s)$$

the state-space diagram for the amplifier and motor may be constructed as shown in figure 50. Again, an intermediate state space variable X5 is created. By observation of the diagram:

$$\dot{X5} = X3 - 31.25\,X5$$

$$X6 = 109.4\,X5$$

Considering the transfer function of the resolver, it is seen that the input to the resolver is X6. The output is labelled X7.

$$\frac{X7}{X6} = \frac{1}{10\,s}$$

Thus, without the aid of the state variable diagram, it is obvious that

$$\dot{X}7 = \frac{X6}{10}$$

Similarly the tacho transfer function directly gives

$$X8 = 1.24 \cdot X6$$

where X8 is the output voltage of the tachogenerator.

The open velocity loop response is then quite simply represented by a series of first-order difference equations. Referring to figure 51, the required equations are:

$$\dot{X}4 = X2 - 0.45\ X4$$

$$X3 = -\ 2.178\ X2 - 98.01\ X4$$

$$\dot{X}5 = X3 - 31.25\ X5$$

$$X6 = 109.4\ X5$$

$$X8 = 1.24\ X6$$

By definition,

$$\dot{X}\ (t) = \lim_{\Delta t \to 0} \frac{X\ (t + \Delta t) - X\ (t)}{\Delta t}$$

For a small increment of time $\Delta t = T$,

$$\dot{X} \doteq \frac{X\ (t + T) - X\ (t)}{T}$$

Therefore, the required set of equations becomes:

$$X4\ (t + T) = T \cdot X2(t) + X4(t)\ (1 - 0.45T)$$

$$X5\ (t + T) = T \cdot X3(t) + X5(t)\ (1 - 31.25T)$$

$$X3\ (t) = -\ 2.178 \cdot X2(t) - 98.01 \cdot X4(t)$$

$$X6\ (t) = 109.4 \cdot X5(t)$$

$$X8\ (t) = 1.24 \cdot X6(t)$$

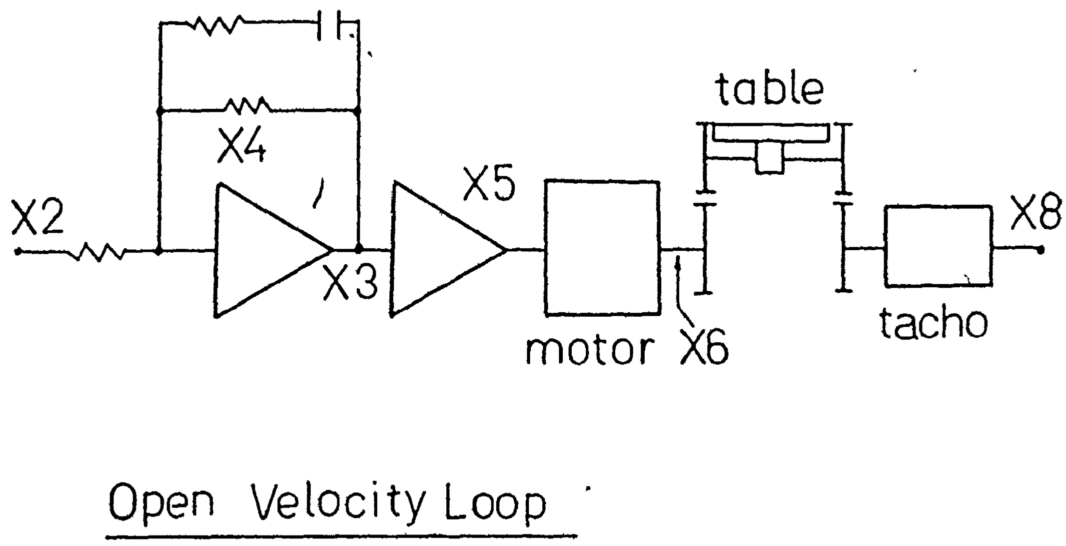


Figure 51

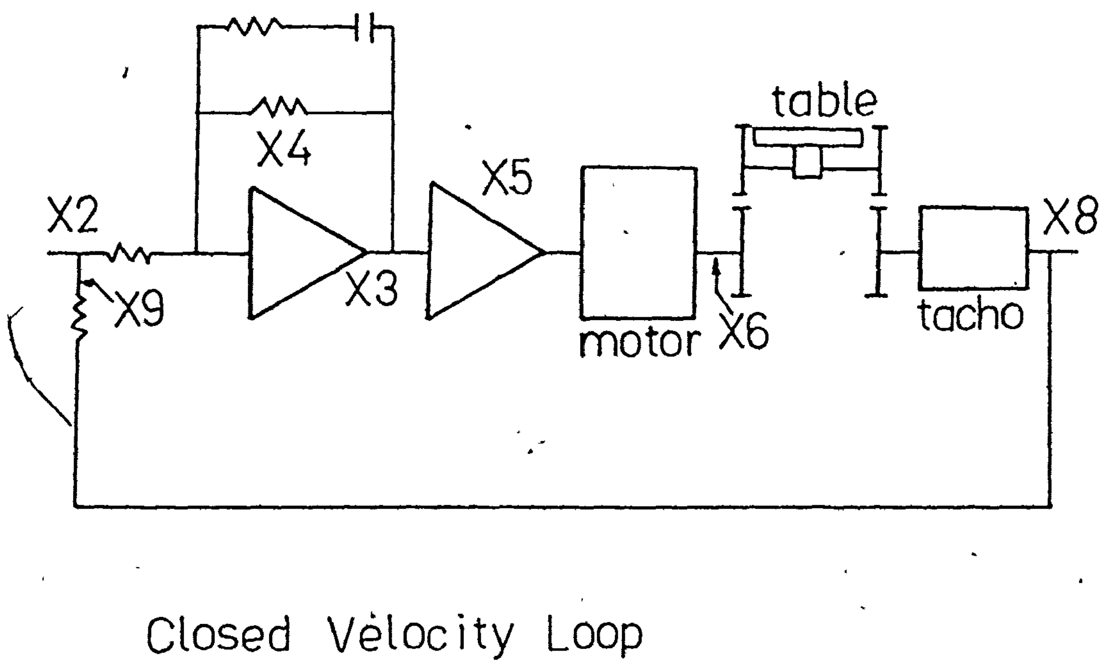


Figure 52

Given a set of initial conditions $X_i(0)$, and using the above set of equations, the output $X8(t)$ (or any function $X_i(t)$) may be determined for any given input $X2 = f(t)$. Applying this procedure, the response of the open velocity loop to a step input of one volt is determined. The result, using $T = 0.05$ seconds, is shown in figure 43 together with the classically-determined equivalent.

Closure of the velocity loop as shown in figure 52 may be affected by adding the two equations:

$$X9\ (t) = 0.2174 \cdot X8(t)$$

and

$$X2\ (t) = X10(t) - X9(t)$$

where X9 is the velocity feedback and X10 becomes the input to the velocity loop. Again, sequentially using the new series of equations, the response of the closed velocity loop to any input $X10 = f(t)$ can be determined. The response to a step input of one volt is shown in figure 45 together with the response determined by classical control theory. The time increment used was $T = 0.002$ seconds.

Finally, closure of the position loop, as shown in figure 53, may be affected by adding the following equations:

$$X7\ (t + T) = 0.1 \cdot T \cdot X6(t) + X7(t)$$

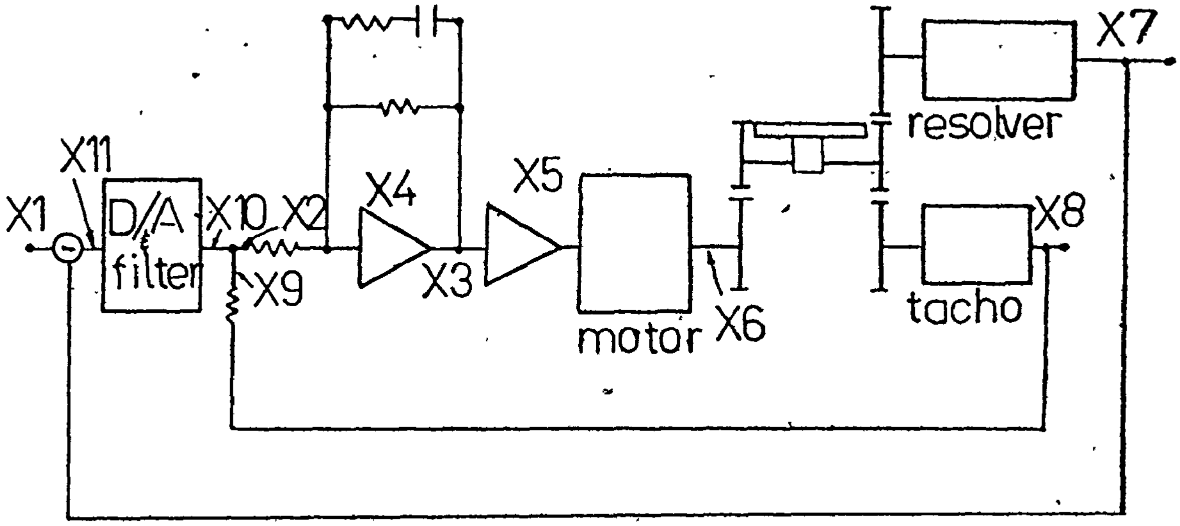
$$X10\ (t) = 45.0 \cdot X11(t)$$

$$X11\ (t) = X1(t) - X7(t).$$

Again, the response of any $X_i(t)$ to the input $X1 = f(t)$ may be observed by sequential application of the entire set of equations. Figure 47 shows the response to a 1.0 inch step input as seen from both resolver and tacho using $T = 0.0005$ seconds. Also shown is the same response determined by classical methods.

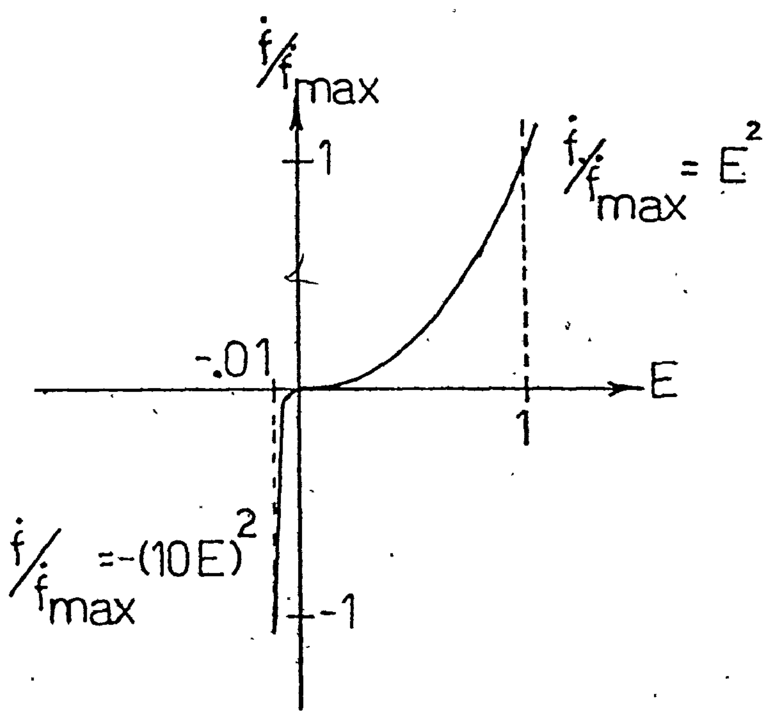
The ramp input response of the closed position loop may be easily examined by altering the input X1 = f (t) and sequentially applying the same set of first-order equations. Thus, the response to the ramp function of figure 48 is shown in figure 48 along with the classical results. The time increment used was T = 0.001 seconds.

The accuracy with which the state-space responses match the responses calculated by the more familiar classical method lends confidence to the application of the state-space method of analysis to the adaptive control loop. This topic is discussed in chapter six.



Closed Position Loop

Figure 53



Parabolic Acceleration Strategy

Figure 54

Chapter 6

A/C Simulation

An adaptive control algorithm for tne HP-2100 minicomputer was suggested in references 16 and 20. The algorithm uses three inputs- two components of force separated by 90 degrees, F_1 and F_2, and a torque, T. The resultant force in the plane of the table is calculated from the two component forces.

$$\text{FORCE} = \sqrt{F_1^2 + F_2^2}$$

These actual values of force and torque are then compared with the pre-selected 'optimal' values of force and torque to determine a relative error upon which to base the adaptive control.

$$\text{Error}_F = \frac{\text{Foptimal} - \text{Force}}{\text{Foptimal}}$$

and

$$\text{Error}_T = \frac{\text{Toptimal} - T}{\text{Toptimal}}$$

The smaller (or larger negative) of the two errors then becomes the controlling error. However, as discussed in chapter three, the torque signal supplied by the transducer was entirely obscured by the electrical noise of the slip rings and brushes. Therefore, a reasonable torque error was not available for control, so the force error was used as the controlling error.

The A/C algorithm calculates a new feedrate approximately every 10.5 milliseconds. The time delay between feedrate corrections is measured by a software counter which counts the number of Assembly language instructions which have been executed, each instruction requiring approximately 2 microseconds. When the software counter indicates that a feedrate correction is required, the resultant force and force error are calculated. A change in feedrate which is proportional to the force error squared is then calculated. The proportionality constant, or A/C gain is dependent upon the sign of the error; a negative error indicates an excessive force which must be corrected quickly to avoid breaking the tool. Therefore, the A/C gain for a negative error was arbitrarily selected 100 times larger than the positive gain.

The suggested gain was derived from the maximum possible acceleration of the existing feed drives, 750 inches/min/sec. Therefore, the desired acceleration was suggested as

$$\dot{f} = \dot{f}_{max} \cdot (\text{Error})^2$$

and, for a negative error,

$$\dot{f} = -\dot{f}_{max} \cdot (10 \cdot \text{Error})^2$$

Thus, the new commanded feedrate becomes

$$f\,(t + T_d) = f(t) + \dot{f} \cdot T_d$$

where: f = commanded feedrate in inches per min.

$\dot{f}$ = commanded acceleration in ipm/sec.

T_d = A/C delay in seconds.

The commanded acceleration can then be seen to follow a parabolic strategy as shown in figure 54. Very simply, the policy may be summaraized by:

$$f_{new} = f_{old} + \Delta f$$

where: $\Delta f = G \cdot (\text{Error})^2$ for a positive error

and $\Delta f = -G \cdot (10 \cdot \text{Error})^2$ for a negative error

where: $G = \dot{f}_{max} \cdot$ actual delay

$\doteq 7.77$ inches/min.

(Since the format of the computer feed-word is inches/min x 100, G appears as 777 in the actual A/C algorithm.)

The effect of this very large negative gain is severely limited by the system constraint of a minimum feedrate of 0.3 ipm. Finally, the new commanded feedrate is checked to ensure that it lies between 0.3 and 30 ipm. before it is supplied to the N/C routine. Control then returns to the routine which accepts the force and torque inputs. The computer continues looping in this routine for the duration of the delay. A listing of the final A/C algorithm is contained in appendix C.

The simulation of the A/C loop begins with the block diagram shown in figure 55. It can be seen that the A/C loop represents another feedback loop in the system. Within the new loop are three elements.

The first element is the cutting process which produces a cutting force dependent upon the cutting conditions. For a given depth of cut a, width of cut b and spindle speed n, the cutting force is dependent upon feedrate. As a first approximation, it may be assumed that the force is proportional to feedrate. Certainly, some of the calculated forces, as discussed in chapter four, seem to follow this simple relation while others obviously have some superimposed periodic component. Therefore, in the simplest case:

$$\text{Force} = X12 = C \cdot X6/10$$

The actual feedrate is equal to X6/10 because the machine slide is driven from the motor through a 2.5:1 gear ratio and a leadscrew of 0.25 pitch. The constant C is chosen to provide reasonable values for X12. However, as may be seen from figure 56 the cutting force does not keep pace with

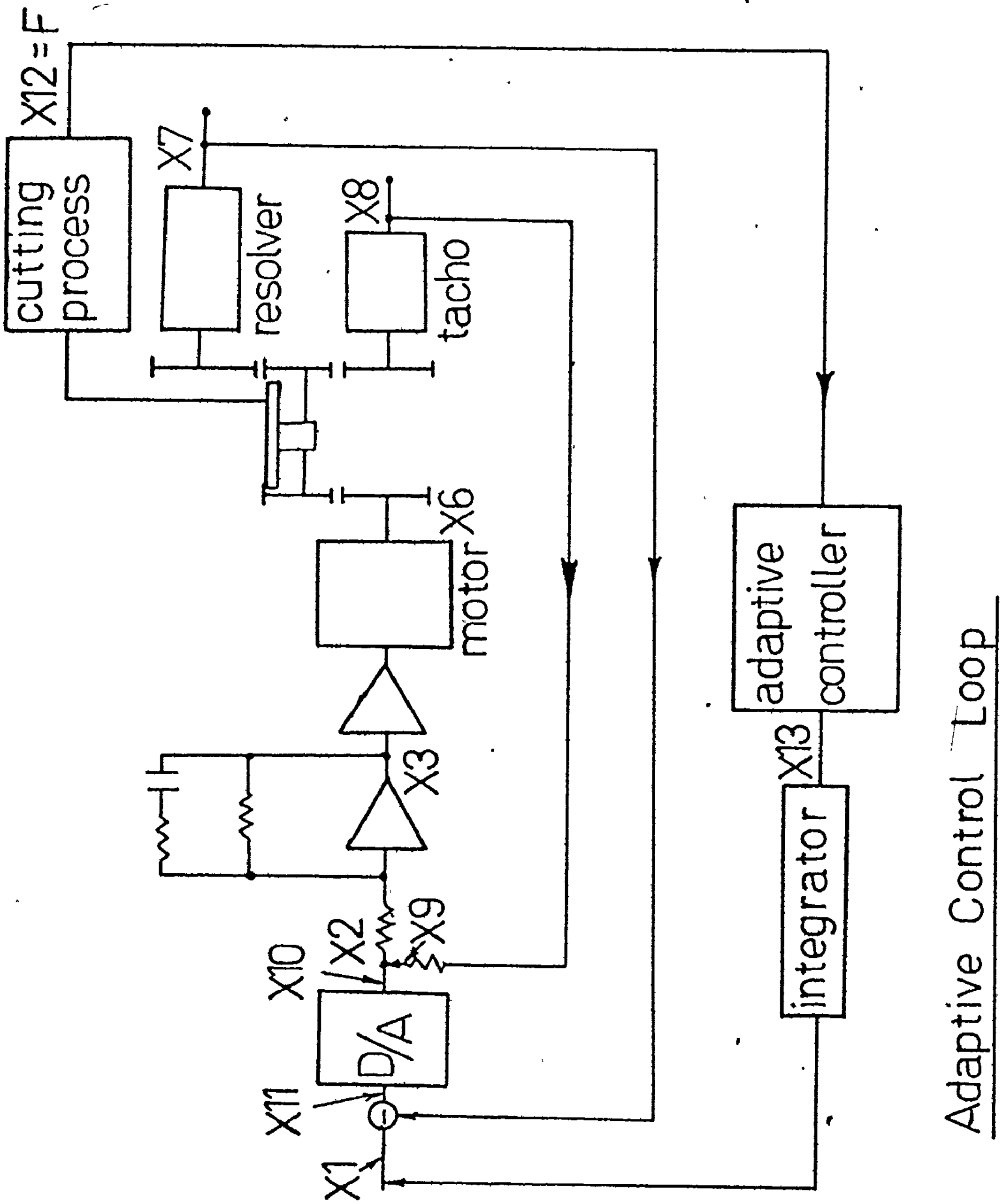

X12 = F
cutting process
resolver
X7
tacho
X8
motor
X6
X3
X2
X9
X10
D/A
X11
X1
adaptive controller
X13
integrator
Adaptive Control Loop

Figure 55

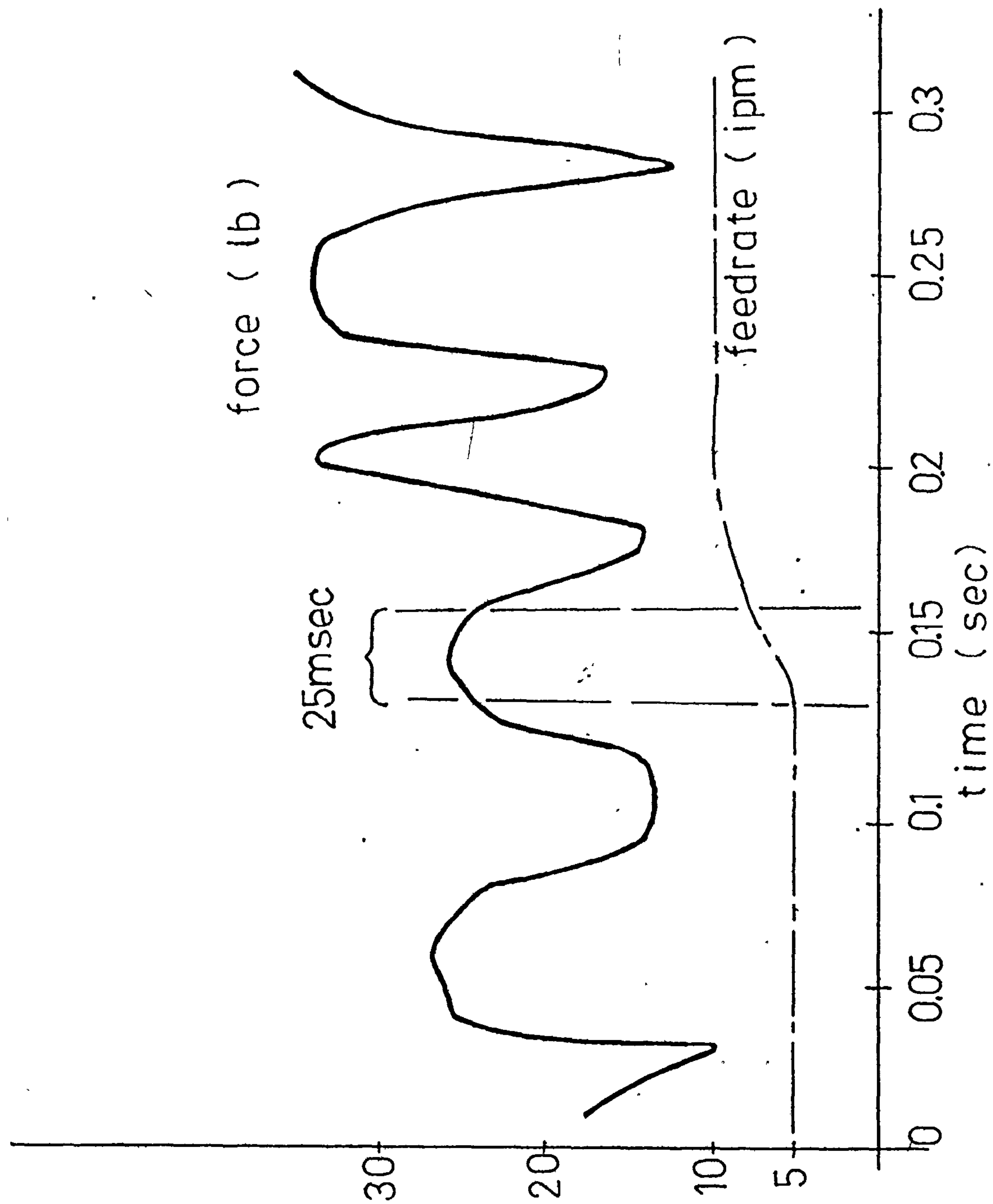


Figure 56
Cutting Force Time Lag

the feedrate at least over the indicated time span. The magnitude of this time lag is dependent upon the spindle speed n and the number of teeth on the cutter z. This time lag is basically due to the finite time required for a new tooth to begin cutting at a new feedrate. Increasing n and z therefore tend to reduce this delay. Thus, the cutting process may be more correctly described by:

$$X12(t) = 0.1C \cdot X6(t - T_L)$$

where T_L is the force time lag.

The second element of the A/C loop may be termed the adaptive controller. This function is performed by the adaptive control algorithm in the HP-2100. As previously discussed, it determines the error between the actual and desired forces and calculates a corrected feedrate command based upon this error. This function may be simulated by

$$\text{Error} = 1.0 - X12/_{\text{Foptimal}}$$

$$\Delta X13 = 12.5 \cdot (\text{Error})^2 \cdot T_d \qquad \text{for Error} > 0$$

$$\Delta X13 = -12.5 \cdot (10 \cdot \text{Error})^2 \cdot T_d \qquad \text{for Error} < 0$$

and $$X13(t) = X13(t - T_d) + \Delta X13$$

where: X13 = feedrate in inches/sec.

T_d = A/C delay in seconds.

(Note that here, the A/C gain for a positive error is $G = 12.5 \cdot T_d$. This differs from the HP-2100 algorithm solely because feedrate here is in inches/sec. Thus, 750 ipm = 12.5 ips.)

The third element of the A/C loop must be added to convert the feedrate command into a distance command added to X1. Thus, it is an integrator, determining the addition to X1 over each basic time increment.

The expression for this is

$$X1(t) = X1(t - T) + X13(t) \cdot T$$

where X13 is the commanded feedrate and T is the basic time increment in the state-space analysis. This integrator is actually resident within the CNC computer routine.

Thus, the equations discussed in this chapter for the simulation of the A/C loop, when combined with the N/C simulation, provide a total state-space representation of the CNC-A/C control of the TOS milling machine. Computer simulation of this complete control function began with the generation of two sets of initial conditions for the CNC-A/C simulation. The initial conditions represent the steady-state operation of the N/C loop as determined by the N/C simulation. For a desired force of Foptimal = 200 lb, the two cases represent both an excessive and a sub-optimal initial cutting force. The A/C loop is then allowed to react to the initial error. The computer simulation routine generates a plot of both actual and commanded feedrate versus time. Thus, both the controlled parameter and the command are recorded. The CNC-A/C simulation using the suggested A/C algorithm is shown in figures 57 and 58. In these simulations, it was assumed that the cutter would have four flutes and be rotating at 500 rpm. Thus, for a small width of cut, the force time lag T_L could be as much as 30 milliseconds. It is obvious from figures 57 and 58 that the system is unstable with a frequency of about 5.0 hz.

At this point, it is worthwhile to note that this CNC-A/C control system is decidedly non-linear because of the parabolic feedrate correction strategy, the A/C delay and cutting force time lag. Whereas the stability of a linear system depends only upon the system's parameters, the

Simulated A/C Response

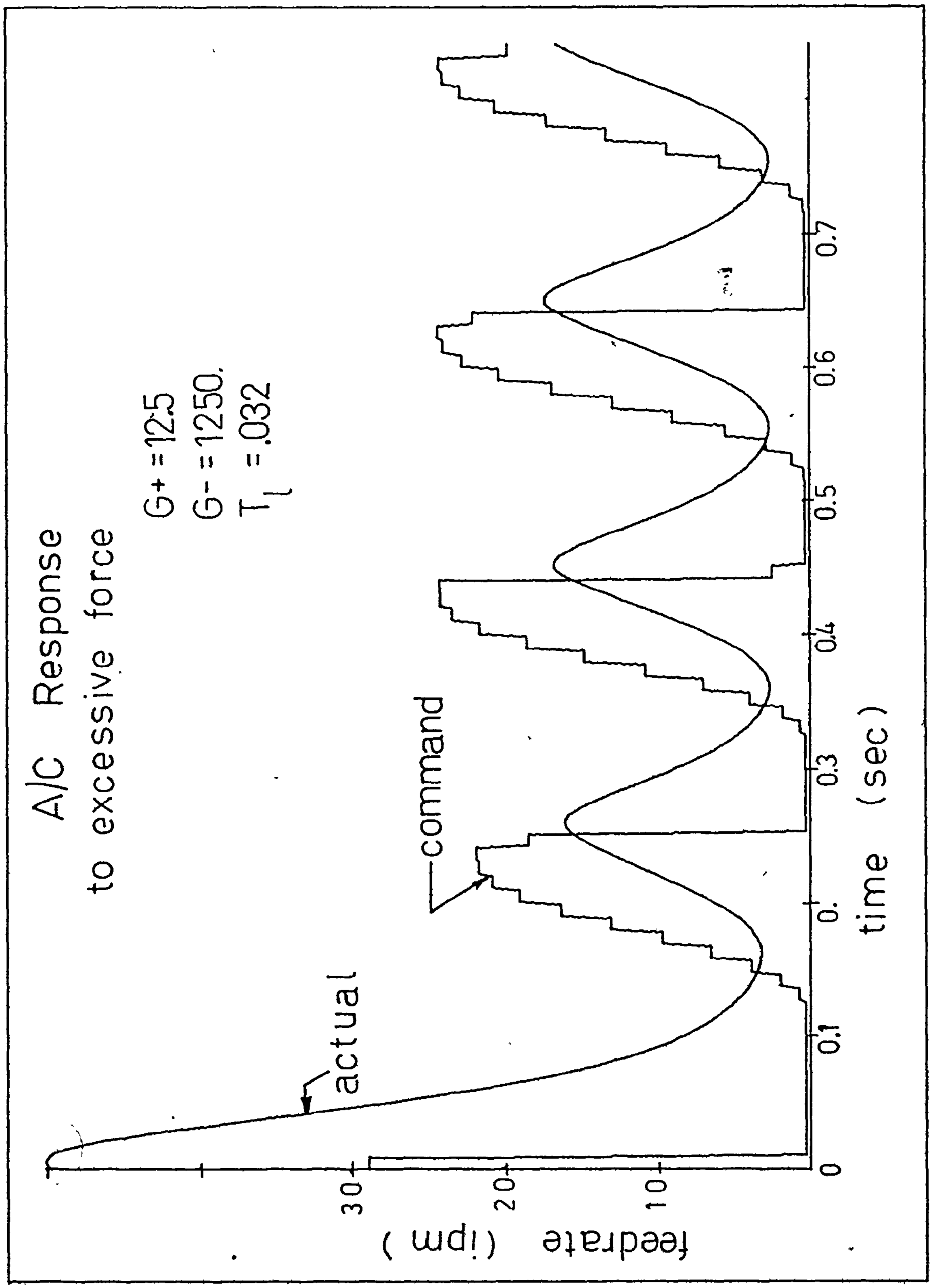


Figure 57

Simulated A/C Response

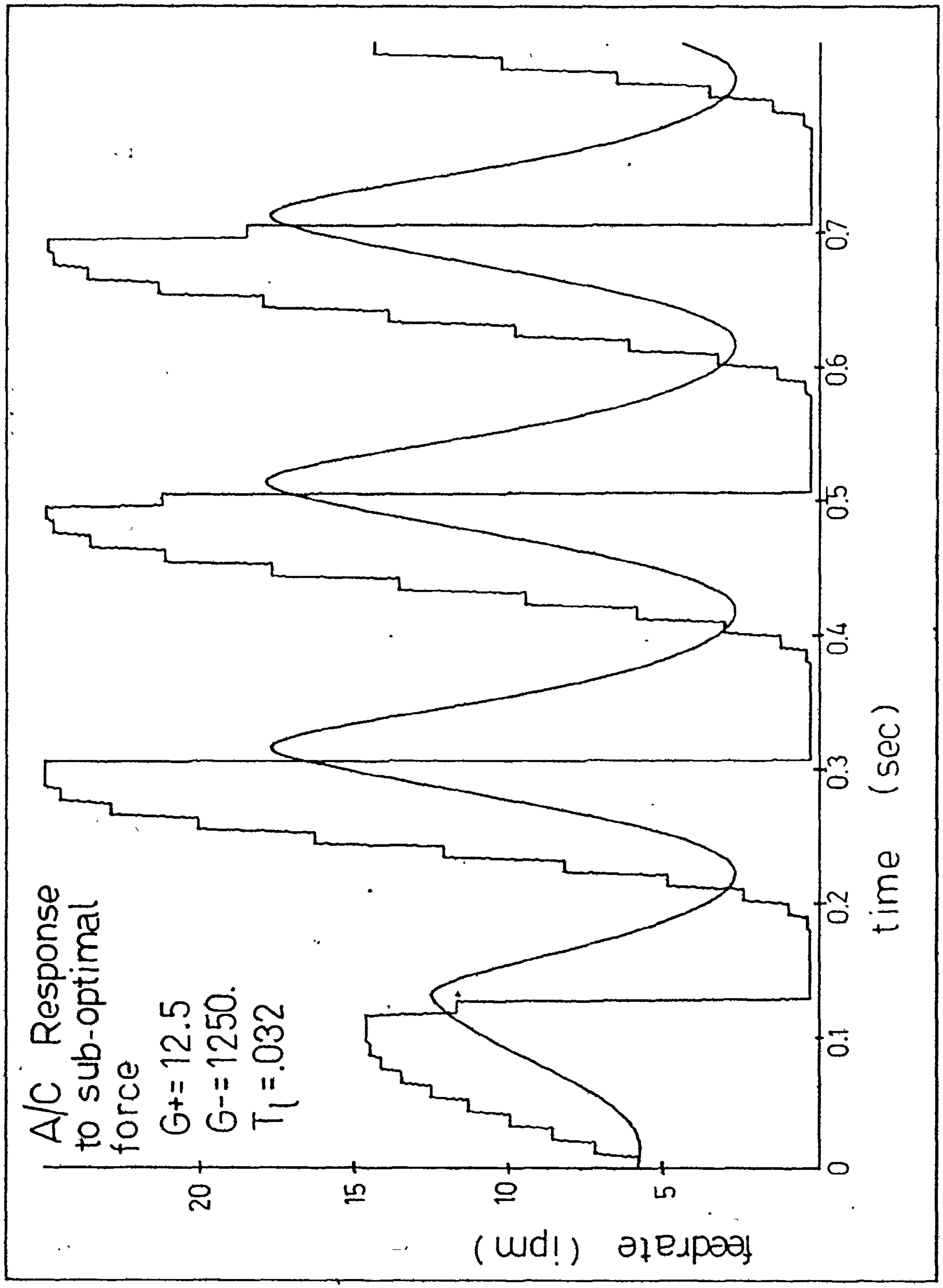


Figure 58

stability of a non-linear system may vary with its initial conditions. For this reason, two completely different sets of initial conditions are tested. Even the question of stability itself in a non-linear system may not be clear. Whereas a linear system is, in general, clearly either stable or unstable with the amplitude of any oscillation either decreasing or increasing, a non-linear system may be considered stable with an oscillation of constant amplitude and frequency. Such oscillations are called limit cycles. "The occurrence of limit cycles in non-linear systems makes it necessary to define instability in terms of acceptable magnitudes of oscillation, since a very small non-linear oscillation may not be detrimental to the performance of a system."[9] The oscillations seen in figures 57 and 58 are limit cycles. It is arbitrarily decided that oscillations with amplitude of less than 0.5 inch per minute are stable because they will probably have minimal effect on the surface finish of a milled part. Therefore, the system performance is considered unacceptable.

Before attempting to stabilize the simulated system, some indication of how well it imitated the actual system was required. Therefore the actual CNC-A/C system was tested using a four-fluted, 0.25 inch diameter end mill at a spindle speed of 500 rpm. During milling, the adaptive controller could not be suddenly activated as was artificially done in the simulation. Thus, the experimental test began with a zero feedrate. Again, the system was found to be unstable as shown by the tacho output in figure 59. The frequency of the observed instability was

[9] Ibid, p. 361.

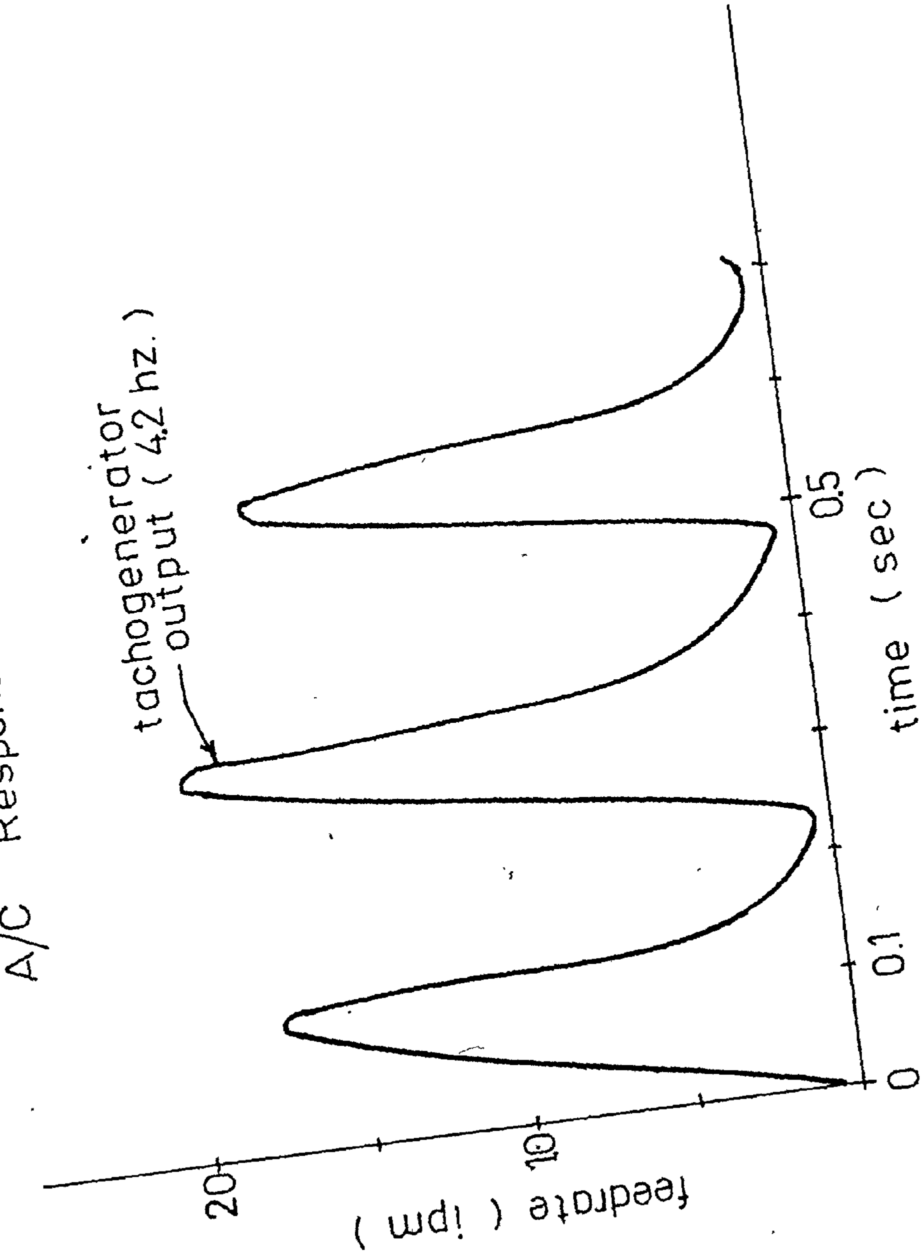


Figure 59
Actual A/C Response

approximately 4.2 hz. Since the simulation predicted both the instability and its approximate frequency, it was judged that the simulation routine could be used in stabilizing the A/C loop.

The time delay in the cutting process creates a considerable phase shift between cutting force and feedrate. With a four-fluted cutter rotating at 500 rpm, the phase shift is 90° for a time lag of 30 milliseconds. Simulating a hypothetical situation having no delay in the cutting force again showed instability as shown in figure 60, although of a slightly higher frequency - 6.7 hz. Thus, the delay in cutting force is not the sole cause of instability.

In attempting to stabilize the system, two methods were employed. The first method involved reducing the system gain. However, so as to leave the N/C system operable as a separate entity, only the A/C loop gains were reduced. The second method involved the introduction of rate feedback which effectively increases the equivalent damping factor of the system. A third possible method of stabilization (which was not tried) requires the addition of a phase-lead network such as in figure 61. Its purpose is to shift the phase of the control signal so that the phase of the output leads the phase of the input.

Employing the first method, the A/C loop gains were decreased with the force delay held at zero. Using a positive error gain of 8.0 and a negative error gain of 800.0, the result shown in figure 62 was obtained. The oscillation in the feedrate is found to diminish with time, taking about 550 msec to disappear. Reducing the A/C loop gains further to a positive error gain of 4.0 and a negative error gain of 400.0

Simulated A/C Response

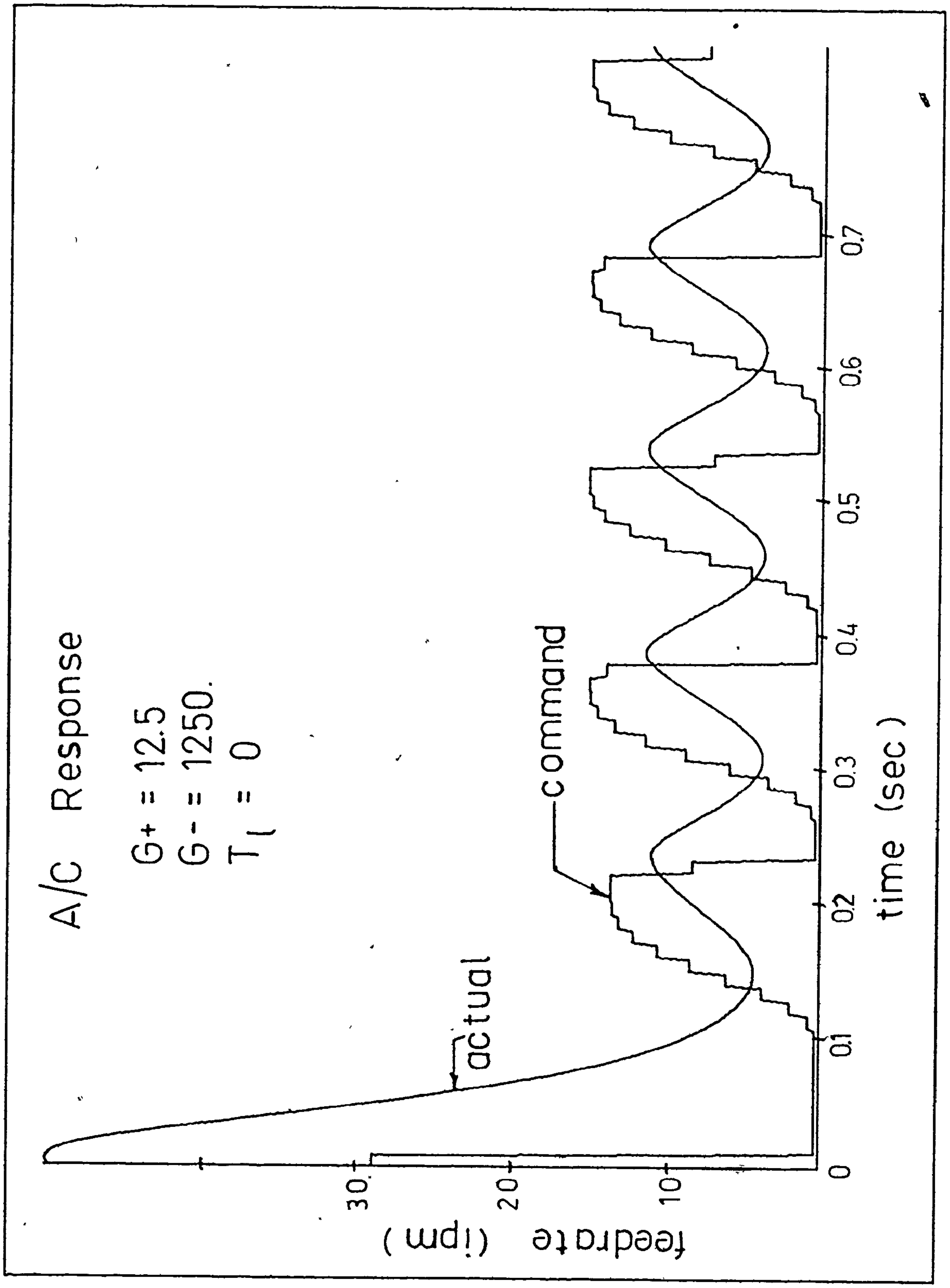


Figure 60

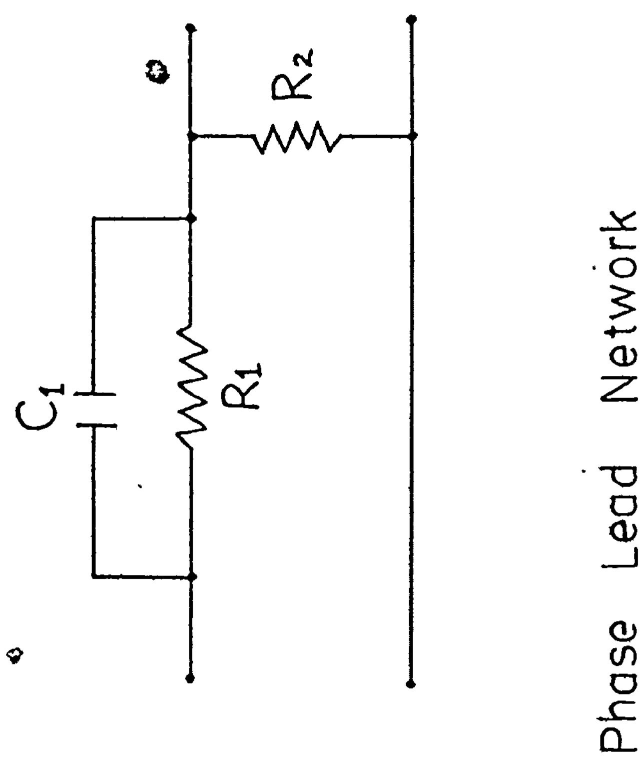


Figure 61

Simulated A/C Response

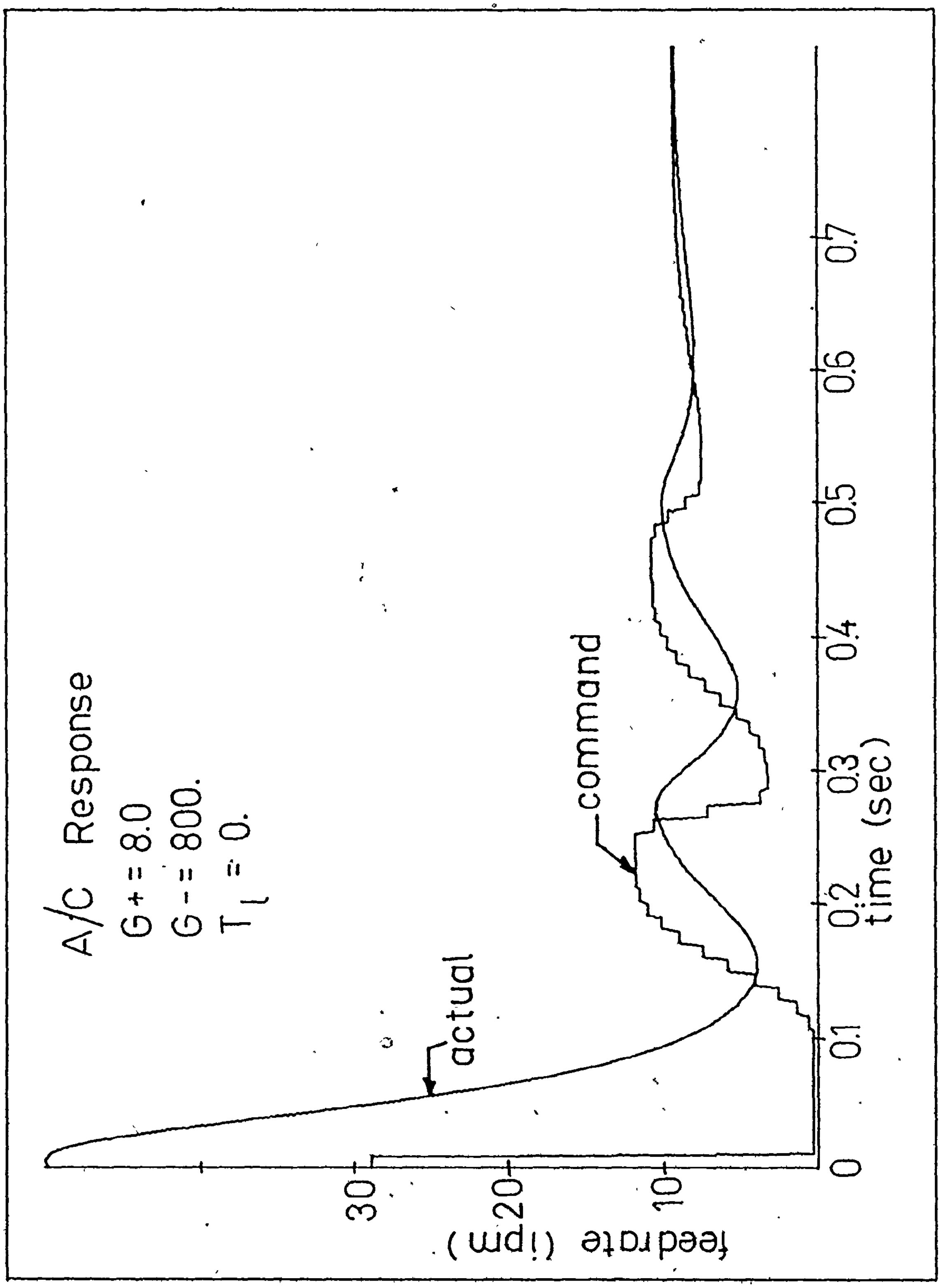


Figure 62

produces a considerably more stable output as shown in figure 63 with the system being quite stable within 300 msec.

It may be noted that the commanded feedrate falls to its minimum value of 0.3 ipm almost immediately, regardless of the A/C loop gains used so far. Also, the shape of the actual feedrate curve for the first 130 msec is identical for the three sets of A/C gains used thus far. In effect, the minimum feedrate constraint has nullified the effect of the very large negative gain. It therefore seems reasonable that the negative gain need not be so large. Using a positive gain of 4.0 and a negative gain of 4.0, the output appears as in figure 64. As expected, the response is identical to that of figure 63 with gains of 4.0 and 400.0. Again, it is stable within 300 msec. A final reduction of gains giving both positive and negative A/C gains equal to unity produced the response of figure 65. The shape of the actual feedrate curve within the first 130 msec does not change, and, as expected, the feedrate oscillations decay very quickly. However, the actual feedrate recovers from the initial overshoot much more slowly than the response of figure 64, requiring about 700 msec to reach a steady state.

In reality, the A/C loop must be capable of stable control in the face of some cutting force time lag as opposed to the rather hypothetical case of zero force time lag. Using a force time lag of 60 msec, which may be caused by a two-fluted cutter rotating at 500 rpm, the output shown in figure 66 is obtained for A/C gains of 4.0 and 400.0. Again, the response exhibits instability with a 3.2 hz frequency. The reduction of the negative A/C gain to 4.0 produces the response in figure 67. The amplitude of the instability decreases with time, but the

Simulated A/C Response

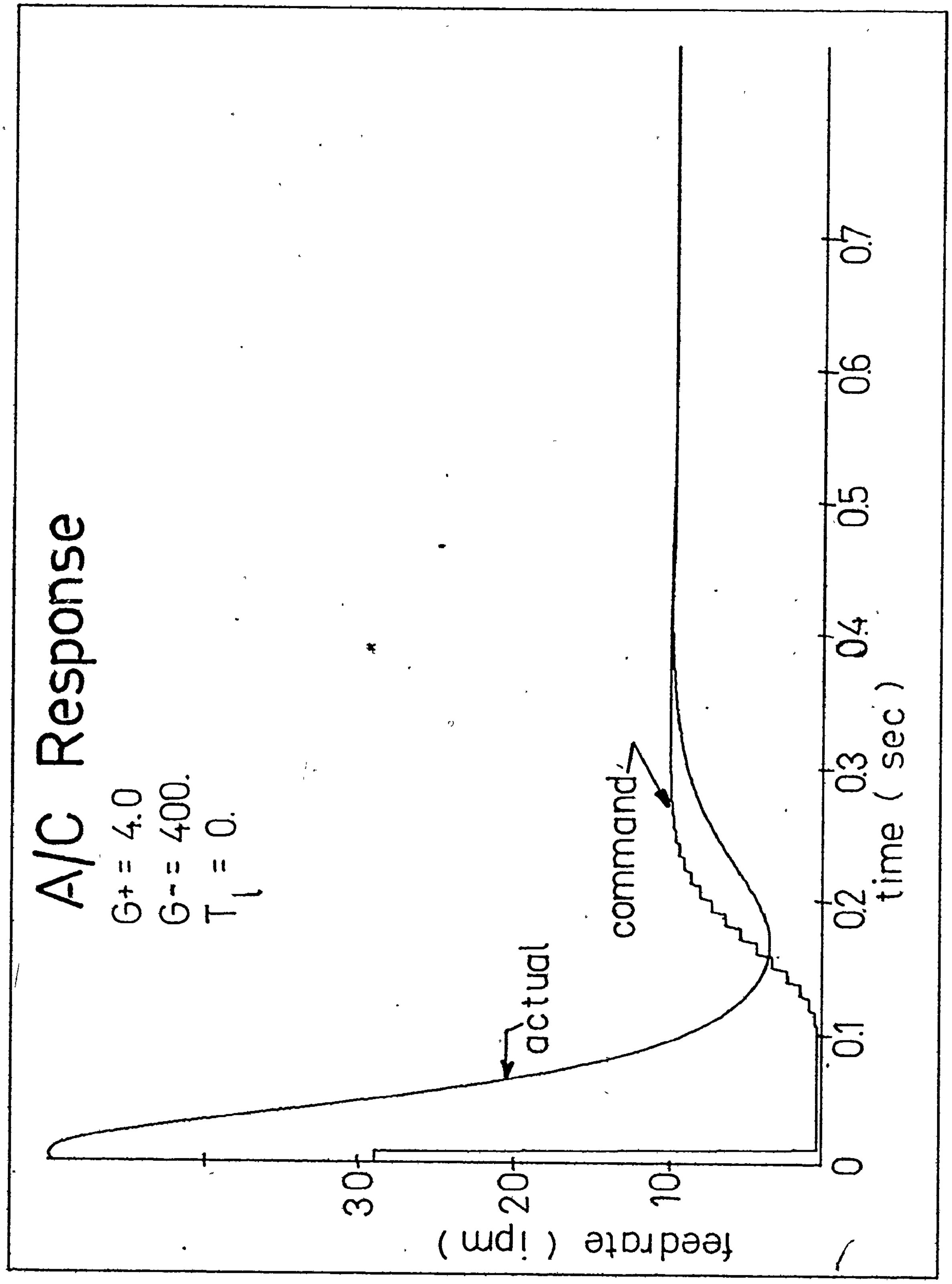


Figure 63

Simulated A/C Response

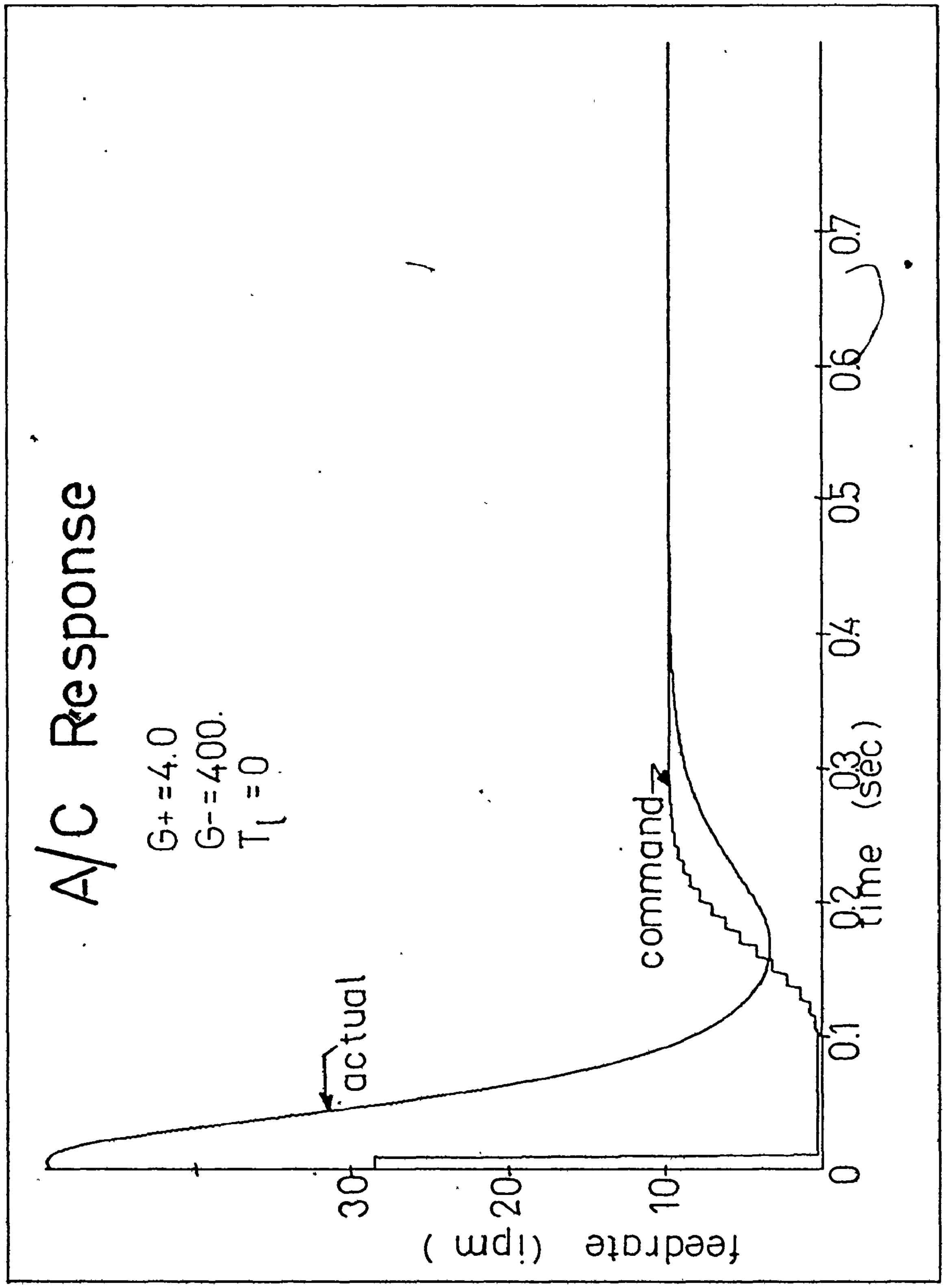


Figure 64

Simulated A/C Response

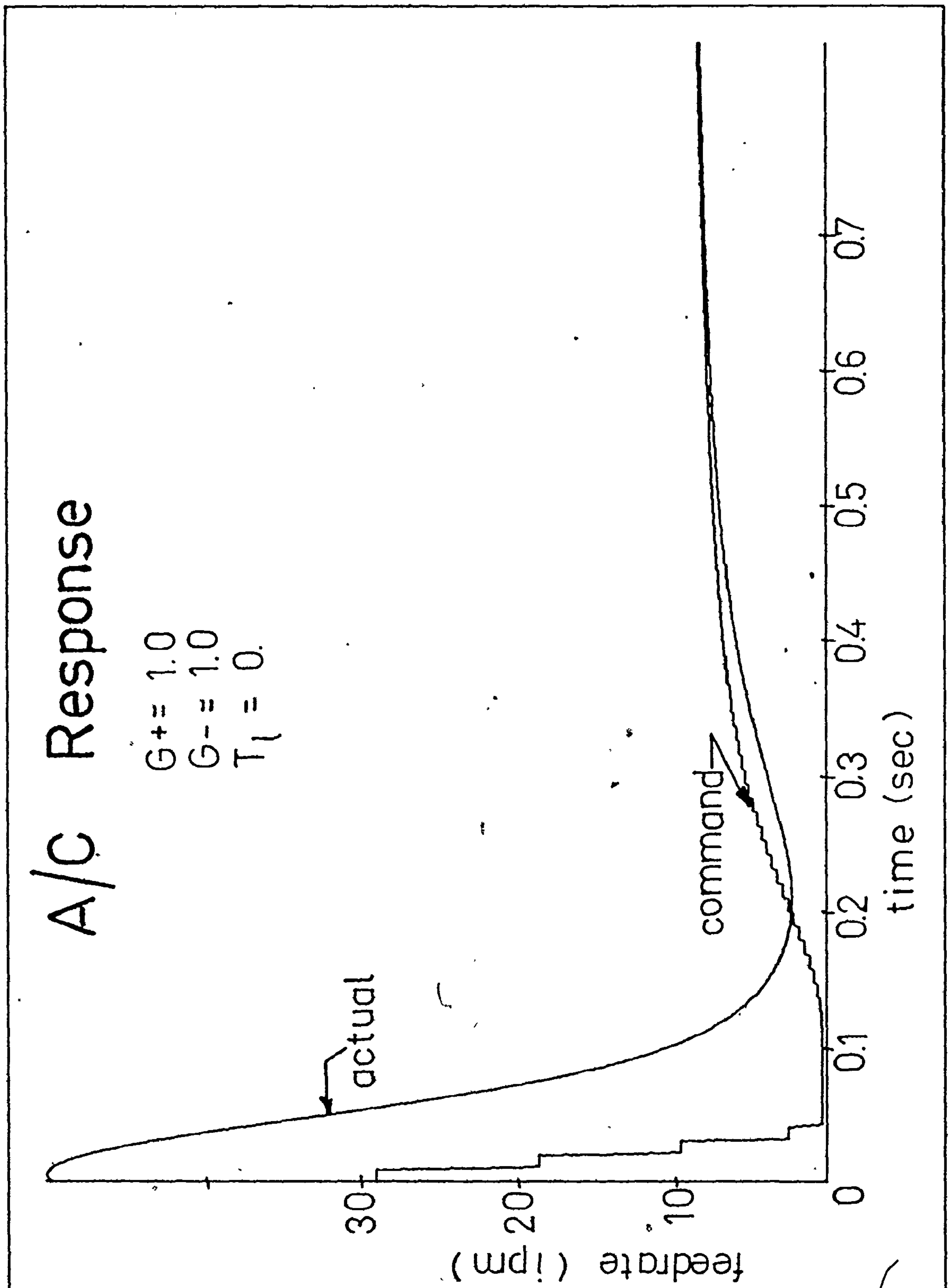


Figure 65

Simulated A/C Response

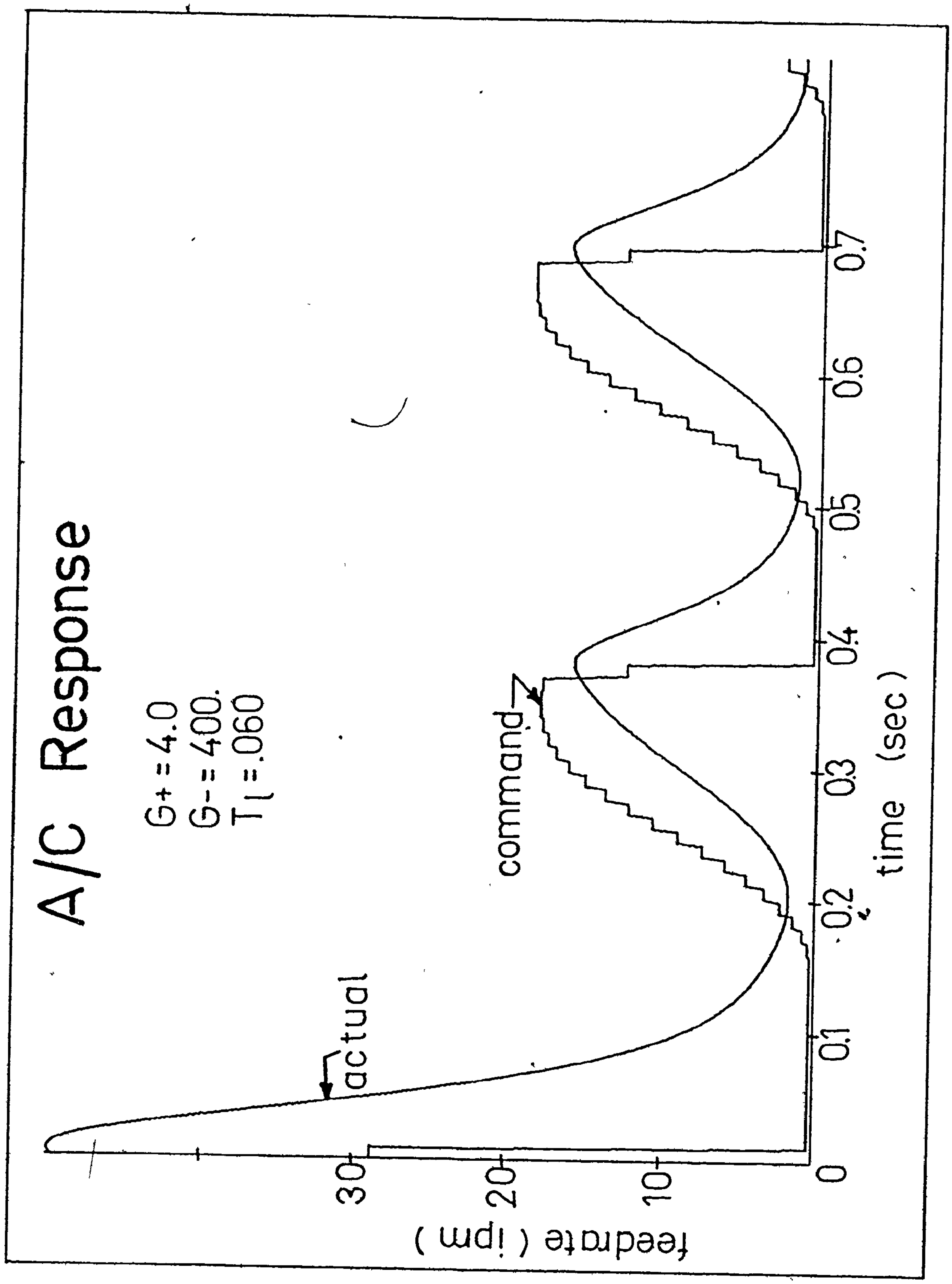


Figure 66

Simulated A/C Response

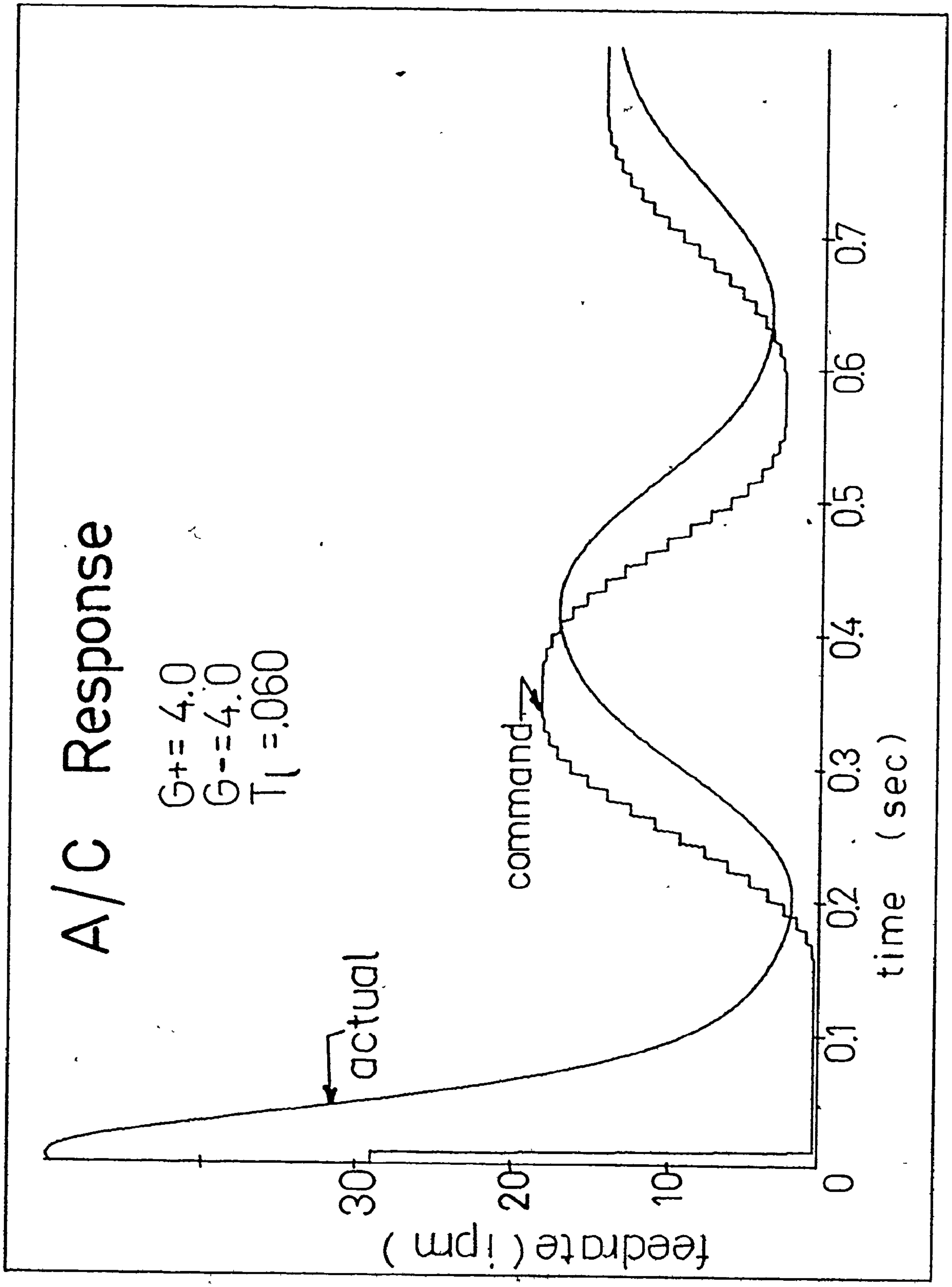


Figure 67

instability persists well beyond one second. Further reduction of the A/C gains tends to produce an undesirably slow response as indicated by figure 65.

The second method of stabilization is then introduced- the addition of rate feedback. The rate feedback term selected is $B \cdot \frac{d(\text{Error})}{dt}$ where B may be considered a damping factor. Thus, the expression for the change in feedrate becomes:

$$\Delta X13 = \text{Gain} \cdot (\text{Error}(t))^2 \cdot T_d + \frac{B \cdot (\text{Error}(t) - \text{Error}(t - T_d))}{T_d} \cdot T_d$$

$$\text{where: } \frac{de}{dt} \doteq \frac{e(t) - e(t - \Delta t)}{\Delta t}$$

Using damping factors of 0.1 and 0.2 with A/C gains of 4.0 and 400.0 left the system unstable as shown in figures 68 and 69. Again decreasing the negative A/C gain to 4.0 with a damping factor of 1.0 was seen to stabilize the system within 500 msec as in figure 70. Using the same damping factor with an A/C gain of unity stabilized the system within about 450 msec as seen in figure 71. Also, the initial overshoot can be seen to be corrected about 50 msec faster.

Further simulations showed a damping factor of 0.8 combined with an A/C gain of unity to give the best system response. Figures 72, 73, 74, 75, 76 and 77 illustrate the response of the final system to several different conditions. In all cases, the system is seen to be quite stable.

The actual A/C algorithm was then altered, reducing the A/C gain to approximately 0.9 or 52 in the units of the actual routine. It was found that the system remained quite stable, having a maximum feedrate

Simulated A/C Response

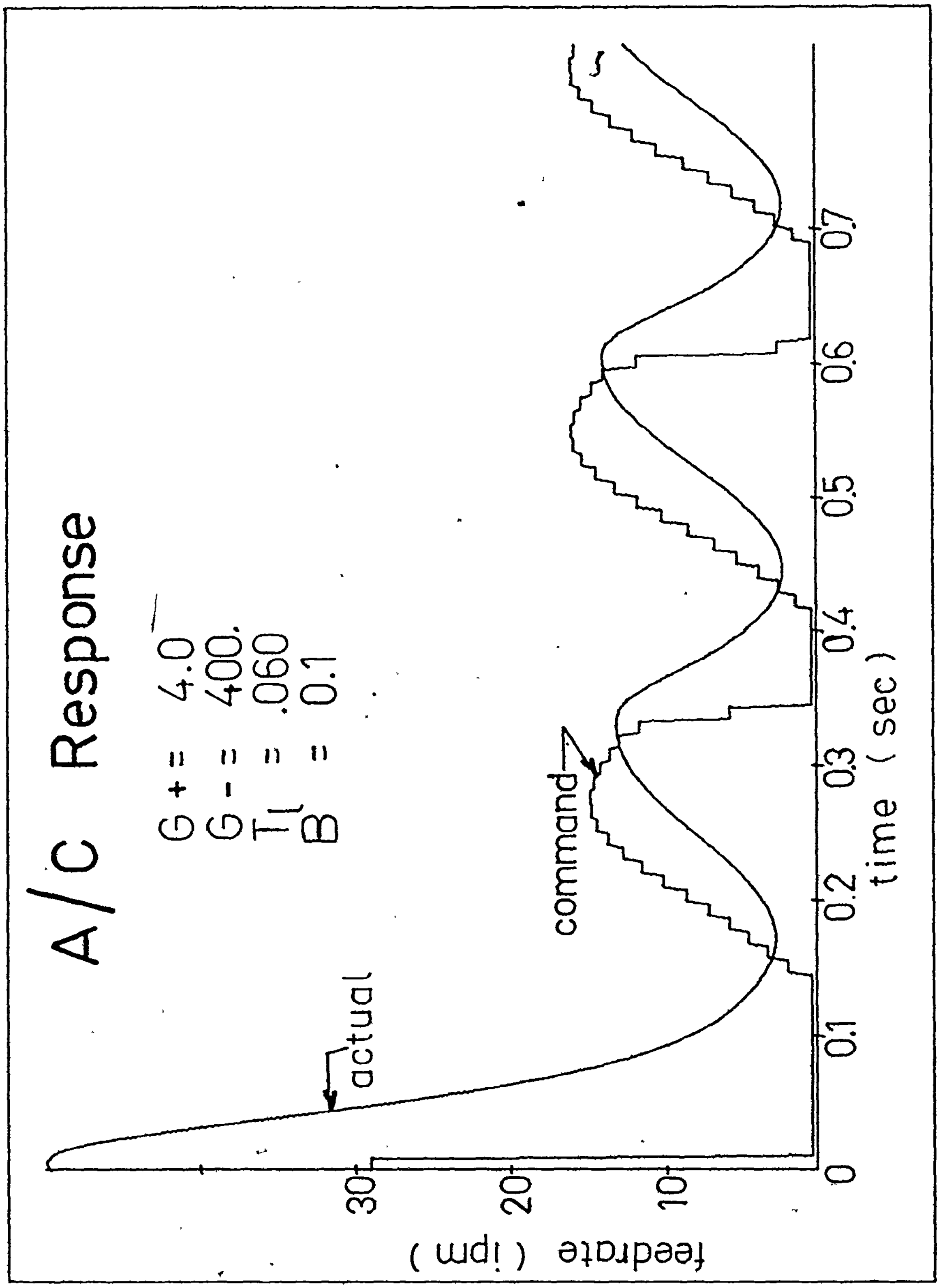


Figure 68

Simulated A/C Response

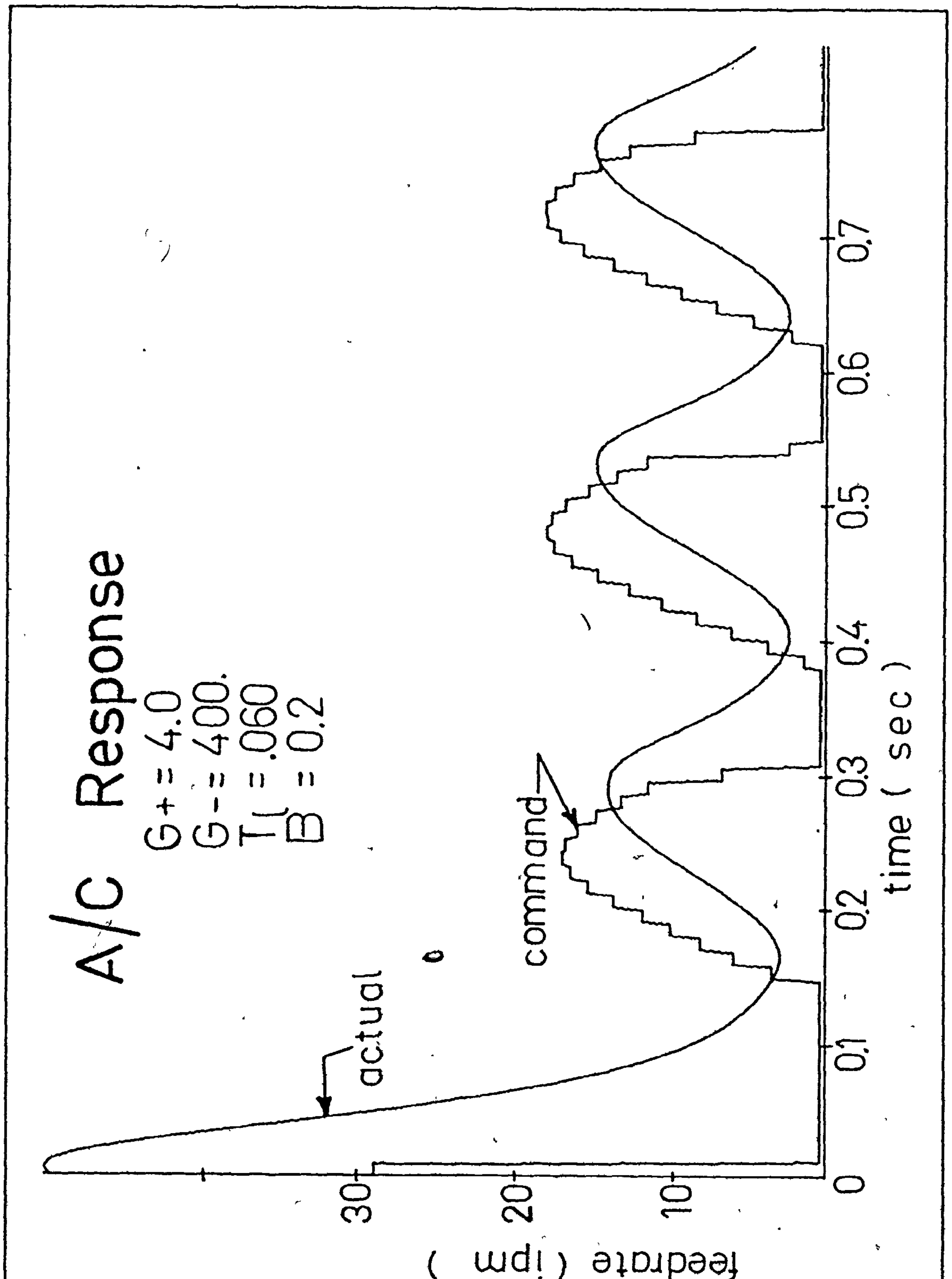


Figure 69

Simulated A/C Response

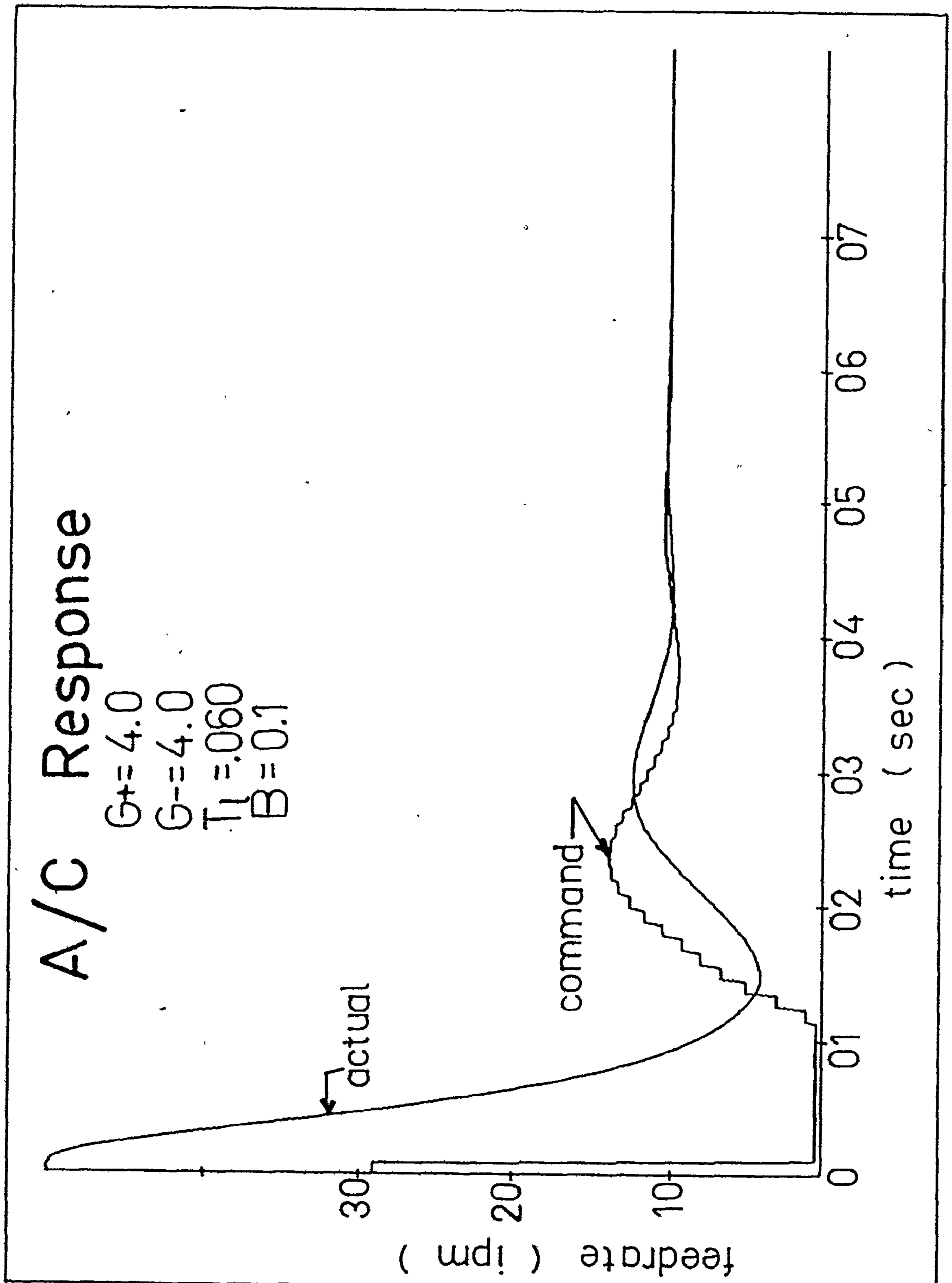


Figure 70

Simulated A/C Response

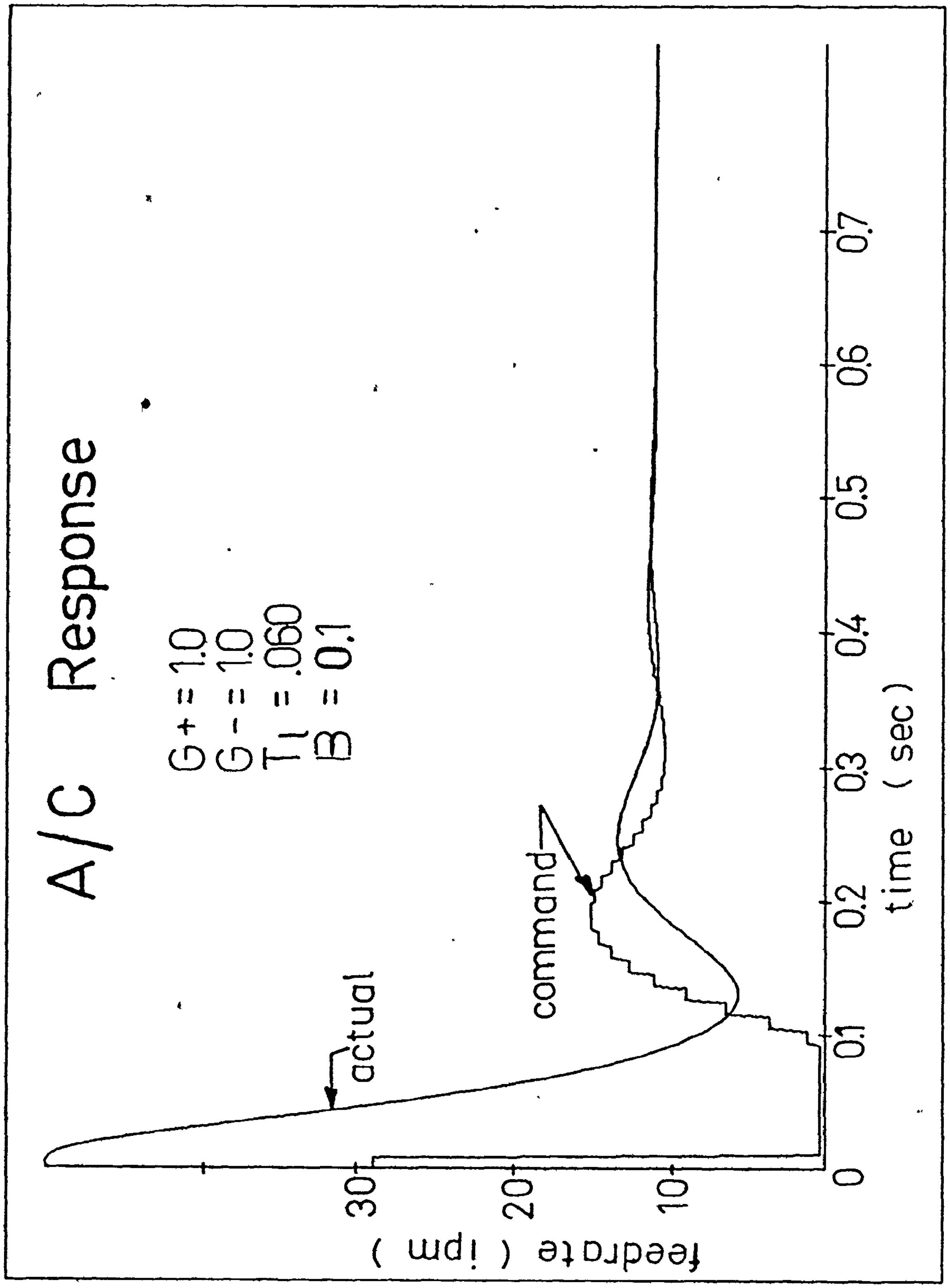


Figure 71

Simulated Response of Final A/C System

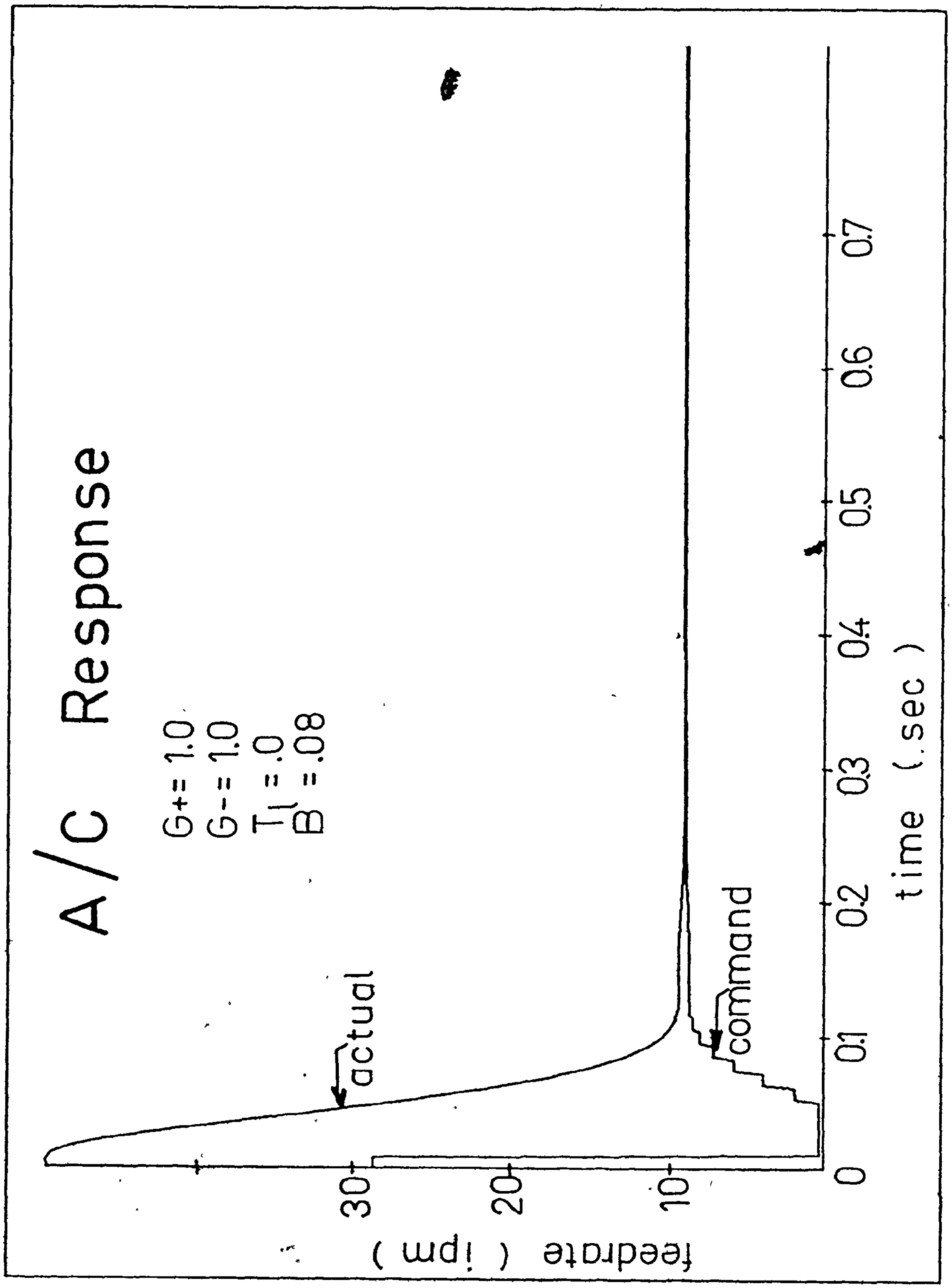


Figure 72

Simulated Response of Final A/C System

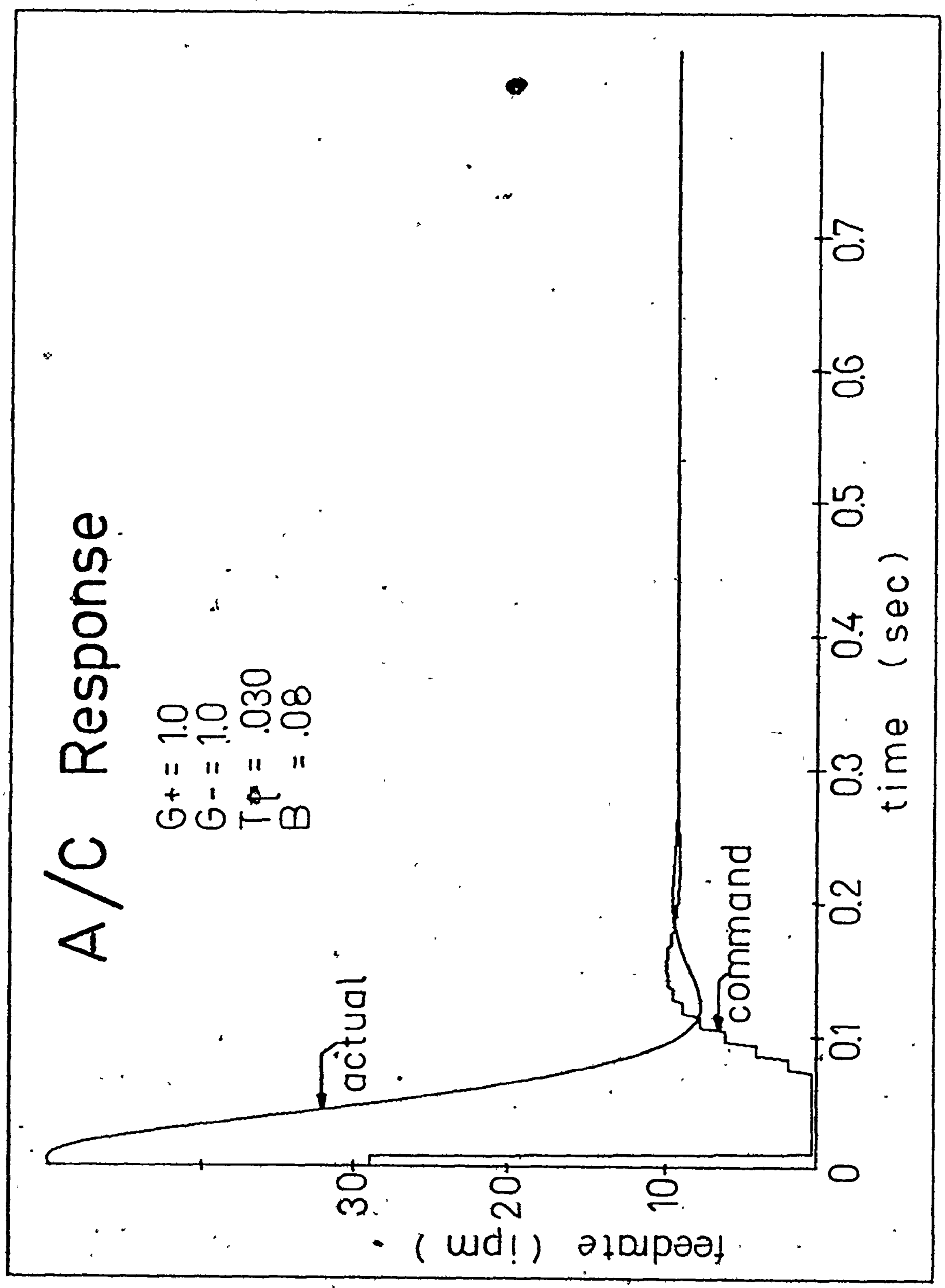


Figure 73

Simulated Response of Final A/C System

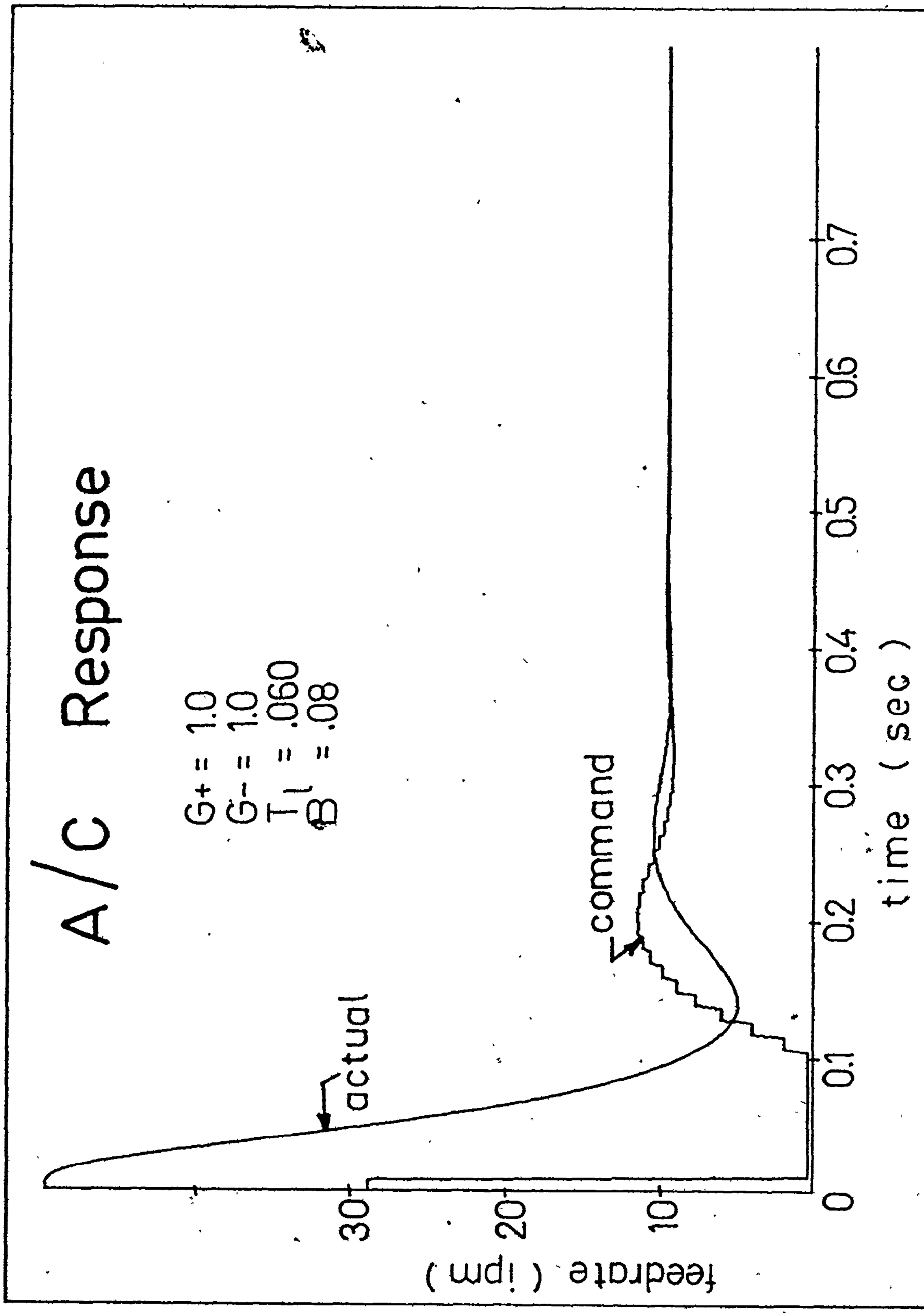


Figure 74

Simulated Response of Final A/C System

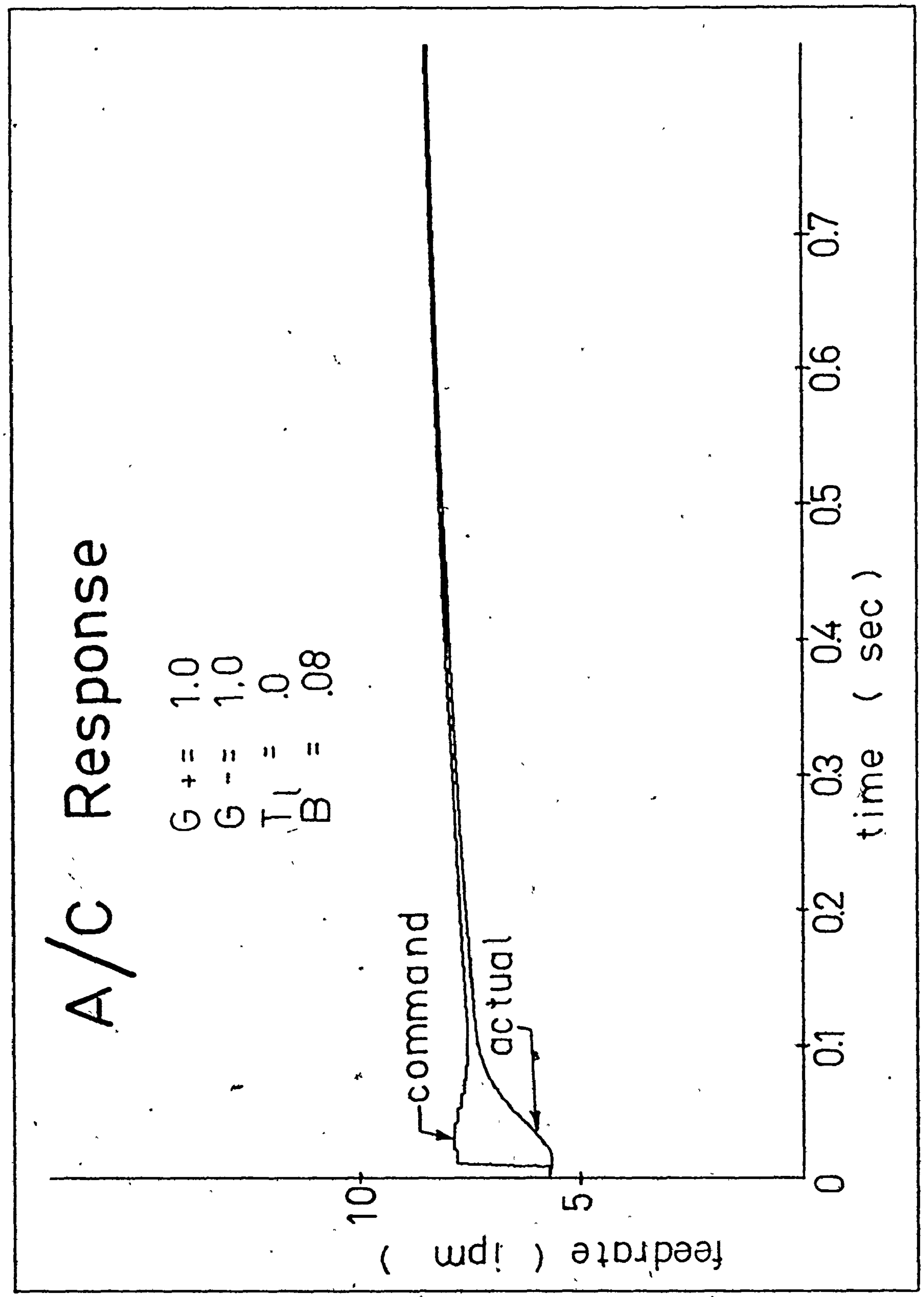


Figure 75

Simulated Response of Final A/C System

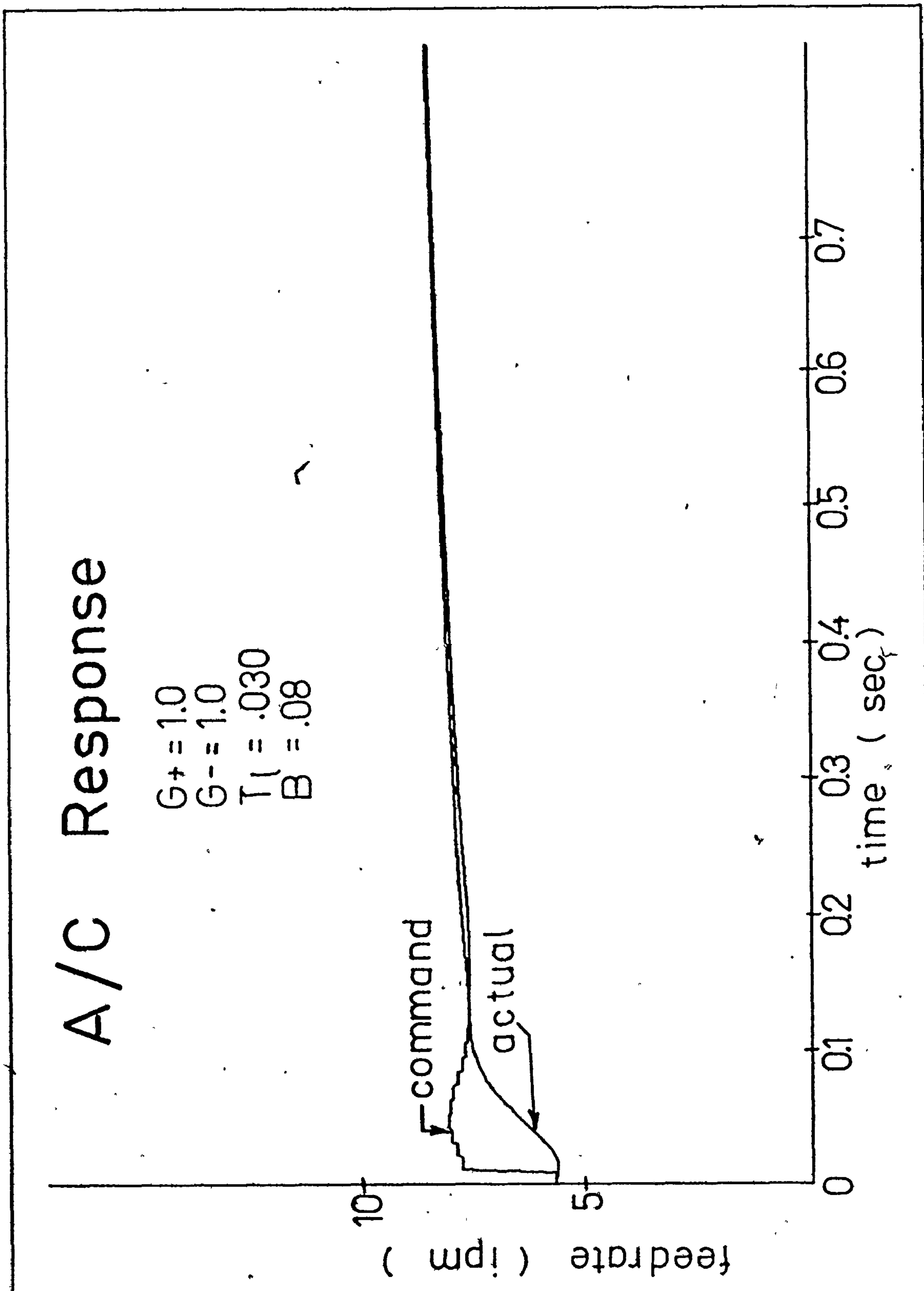


Figure 76

Simulated Response of Final A/C System

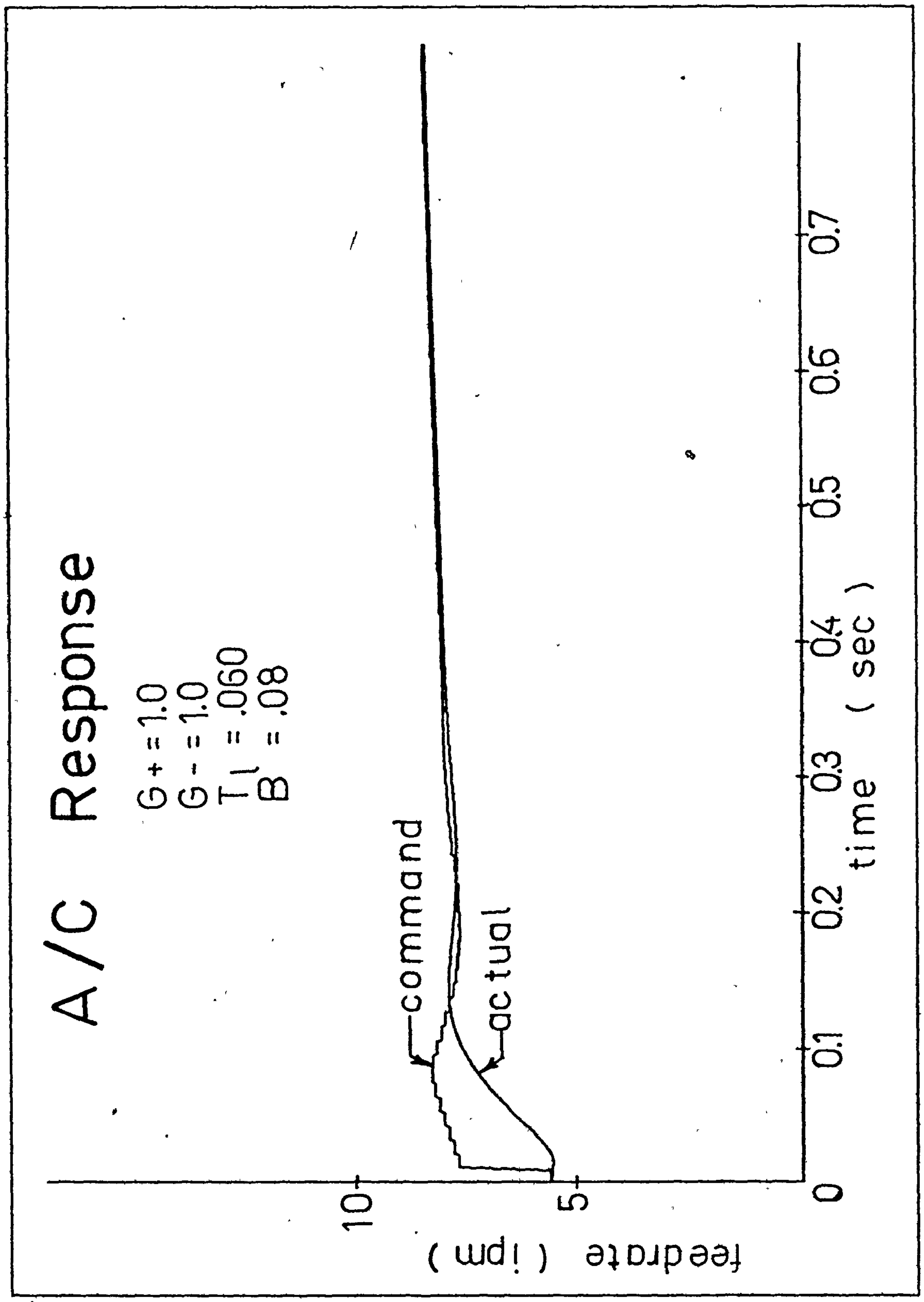


Figure 77

variation of less than 0.9 ipm peak to peak with no dominant frequency, during steady-state milling. The actual A/C gain used was slightly less than that suggested by the A/C simulation and the damping factor B was zero. The resulting system was seen to be stable without any damping term probably because the force time lag T_L was not great. The cutter had two teeth and a spindle speed of 500 rpm, allowing a maximum of T_L = .060 sec. However, since the workpiece material was aluminum, a large width of cut b was used, thereby greatly reducing the actual value of T_L.

Subjecting the A/C loop to an excessive force by allowing the end mill to enter the workpiece at a high feedrate, produced the result shown in figure 78. The time required to correct the excessive force error is seen to be 1.38 seconds. Figure 79, produced under non-adaptive control conditions shows that, at 5 ipm, about 570 msec are required to reach the maximum force when running into a workpiece. Thus, the very slow response exhibited in figure 78 would surely break a cutter when entering a workpiece at a high feedrate.

The actual A/C system was not tested further, having stabilized the system. To improve the response time of the system, both the A/C loop gain and the damping term discussed in the A/C simulation require further experimental determination.

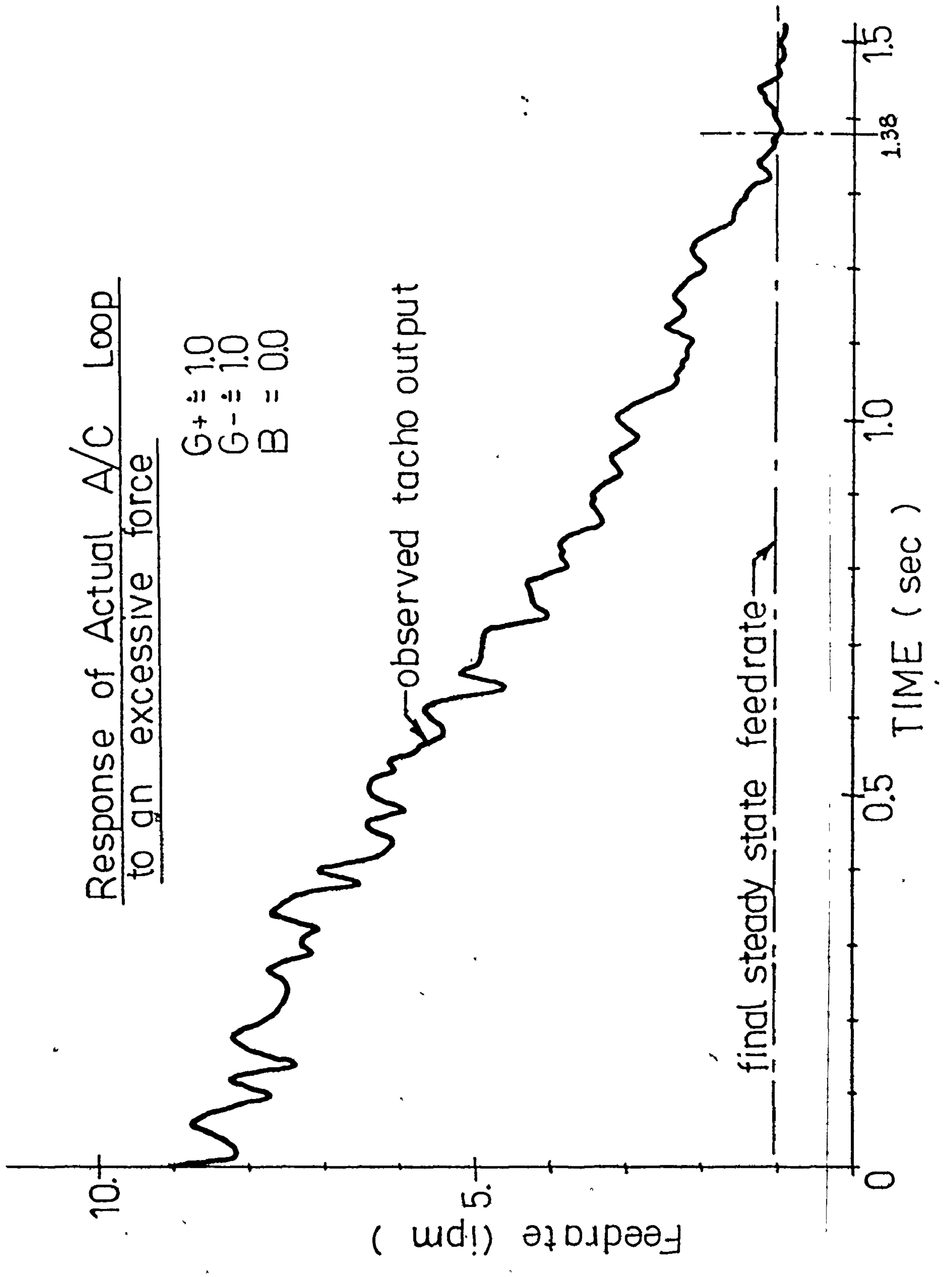

Response of Actual A/C Loop
to an excessive force
G+ ≐ 1.0
G- ≐ 1.0
B = 0.0
observed tacho output
final steady state feedrate
TIME (sec)
0
0.5
1.0
1.38
1.5
Feedrate (ipm)
10.
5.

Figure 78

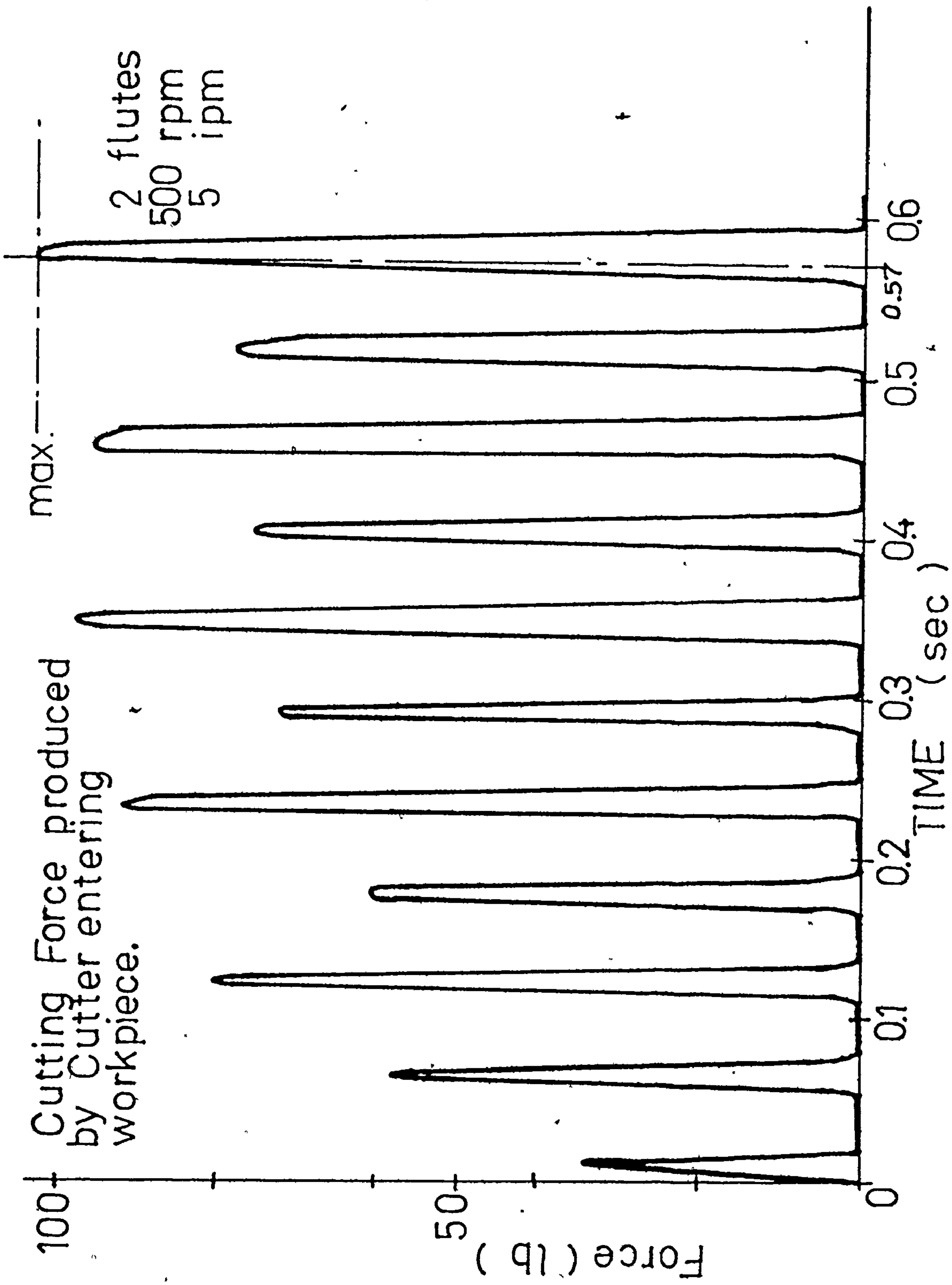

Cutting Force produced
by Cutter entering
workpiece.
max.
2 flutes
500 rpm
5 ipm
100
50
0
Force (lb)
0
0.1
0.2
0.3
0.4
0.5
0.57
0.6
TIME (sec)

Figure 79

Chapter 7

Summary

An adaptive control system controlling the feedrate of a CNC milling machine has been simulated and developed. The A/C strategy has been altered sufficiently to produce stable control in at least the tested cases. However, to produce an efficient A/C system capable of approaching a workpiece at maximum feedrate, further work must be done. Experimentation with the A/C loop gain and the damping factor B is required to improve the system response. A further modification may involve using the input channel now occupied by the torque signal to determine whether the tool is in or out of the cut; by placing an accelerometer on the headstock, the A/C loop could be given an early warning when entering a workpiece at high feedrate. Rather than waiting for the cutting force to reach an excessive value, the vibration caused by encountering the workpiece would be sufficient to initiate deceleration.

Further work need not be limited to the particular A/C algorithm discussed in chapter six. Rather than commanding an acceleration or deceleration for instance, the commanded feedrate may be based directly upon the actual feedrate by using the output of the tachogenerator as an input to the computer. Thus, rather than base the new feedrate command upon the previous command which may not correspond at all to the actual feedrate which is producing the force error, the new algorithm may determine the new feedrate command by:

$$X13 = X6/_{10} + \text{Error} * \text{Gain}$$

where: $X6/_{10}$ is the actual feedrate.

Thus, the proposed algorithm is linear. One example of the simulated response of such a linear controller is indicated in figure 80. It is seen to be quite stable without the need for a damping term. In actual A/C milling, the tacho output may be used to provide the actual feedrate.

Further work may be useful on the A/C transducer itself. Its applicability to die sinking could be greatly enhanced by making it smaller and simpler. Similarly, the ADP could be improved to accept more analog inputs for digitizing, such as the tacho output.

The field of adaptive control has witnessed many very complex and cumbersome systems. However, there remains a need for simple yet efficient A/C systems. It is towards this goal that further research should be directed.

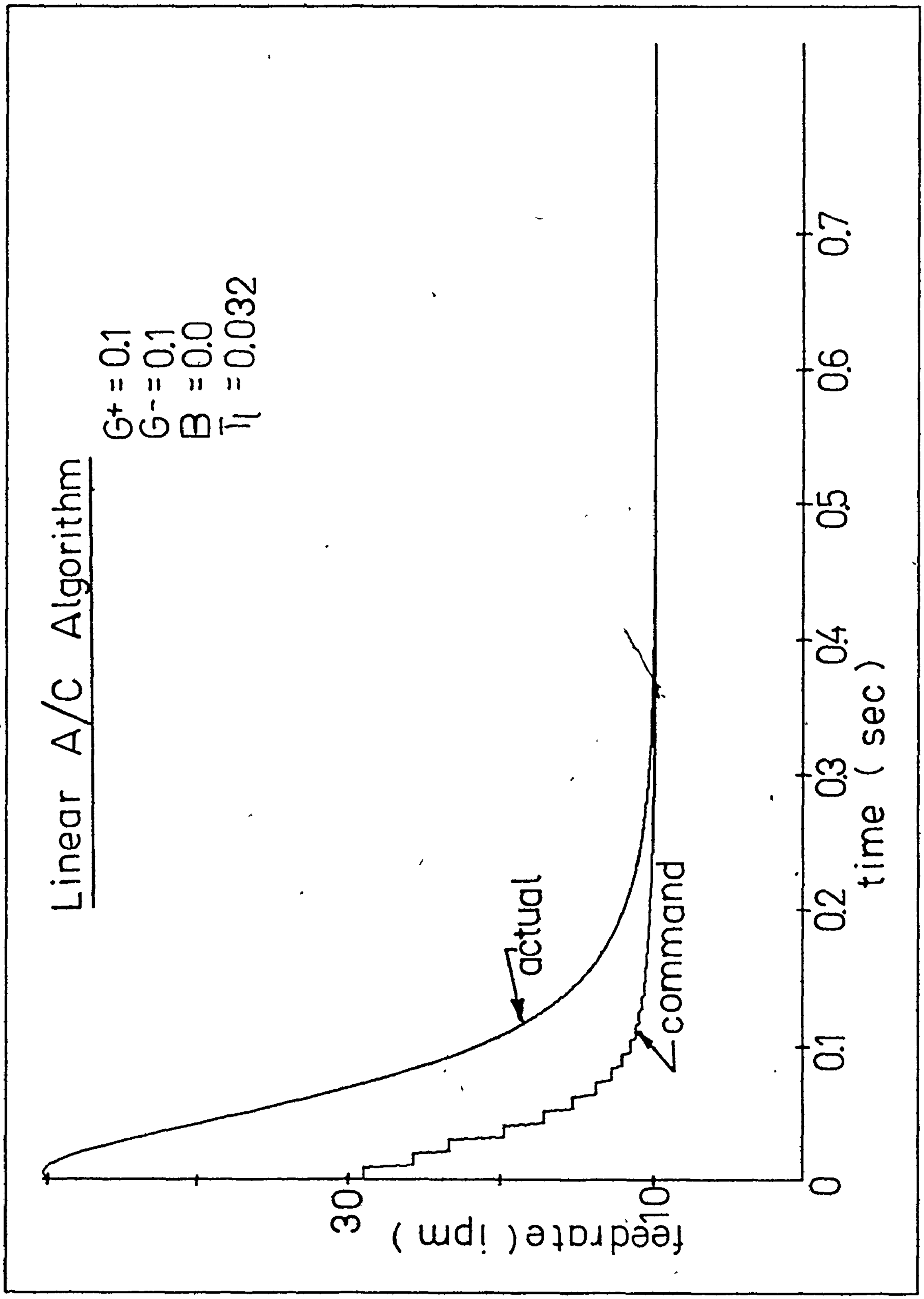

Linear A/C Algorithm
G+ = 0.1
G- = 0.1
B = 0.0
Tl = 0.032
actual
command
feedrate (ipm)
30
10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
time (sec)

Figure 80

APPENDIX A

Derivation of the Cutting Force Equations

Appendix A

Derivation of the Cutting Force Equations

The basic relation governing the tangential force applied to a length of the cutting edge dy in angular position ϕ is

$$dF_t = K \cdot S_t \cdot \sin\phi dy$$

The magnitude of each increment of radial force is $0.3 \cdot dF_t$.

In determining the total force acting on an end mill, both the tangential and radial components of force must be included. It is obvious that the radial and tangential forces change not only in magnitude, but also in direction as the tool rotates. To simplify the problem of directionality, the forces are superimposed on an X-Y coordinate system, and their x and y components are determined. From figure 12 it is obvious that the components are:

$$dF_{rx} = - dF_r \cdot \sin\phi$$

$$dF_{ry} = - dF_r \cdot \cos\phi$$

$$dF_{tx} = - dF_t \cdot \cos\phi$$

$$dF_{ty} = dF_t \cdot \sin\phi$$

Using the basic relations governing the force on a tooth, the steady state cutting forces are derived. The fundamental relations are:

$$dF_t = K \cdot S_t \cdot \sin\phi dy$$

$$dF_{tx} = -K \cdot S_t \cdot \sin\phi\cos\phi dy$$

$$dF_{ty} = K \cdot S_t \cdot \sin^2\phi dy$$

These relations are then integrated over the length of the cutting edge in each phase of cut.

Phase A

The variable determining the position of the cutting edge is the angle ϕ. Phase A is specified by:

$$0 \leq \phi \leq \delta$$

where δ is the angle encompassed by the cutting edge over the cut width b, as shown in figure 81. It can be seen from the figure that:

$$\delta = \frac{b}{r} \cdot \tan\beta$$

where β is the helix angle of the end mill teeth.

The cut extends, with edge in position ϕ, from y = c to y = b. Therefore, over the entire cutting edge:

$$F_t = \int_c^b dF = K \quad S_t \int_c^b \sin\phi dy$$

$$\text{but: } y = \frac{r}{\tan\beta} \cdot \phi + c$$

$$\text{thus: } dy = \frac{r}{\tan\beta} \cdot d\phi$$

Integrating with respect to ϕ,

$$F = \frac{K \cdot S_t \cdot r}{\tan\beta} \int_0^{\phi} \sin\phi d\phi$$

$$= \frac{-K \cdot S_t \cdot r}{\tan\beta} \cos\phi \Big|_0^{\phi}$$

$$= F_u \left[1 - \cos\phi \right]$$

Similarly,

$$F_{tx} = \frac{-K \cdot S_t \cdot r}{\tan\beta} \int_0^{\phi} \sin\phi\cos\phi d\phi$$

$$= \frac{-F_u}{2} \sin^2\phi \Big|_0^{\phi}$$

$$= \frac{-F_u}{2} \sin^2\phi$$

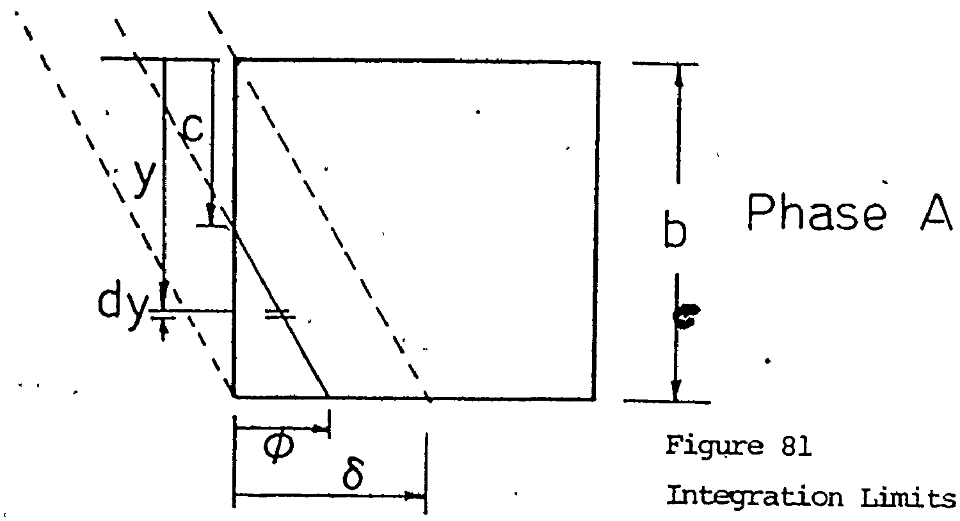


Figure 81
Integration Limits -
Phase A

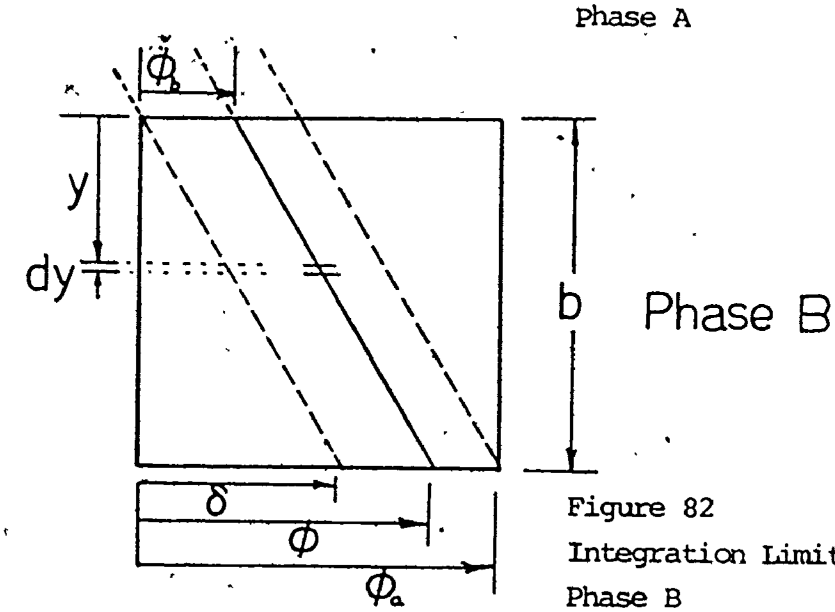


Figure 82
Integration Limits -
Phase B

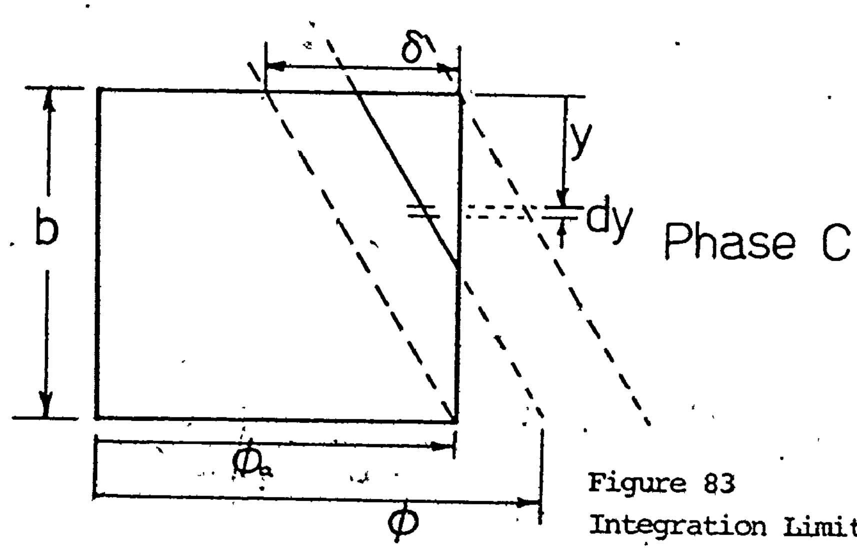


Figure 83
Integration Limits -
Phase C

$$F_{ty} = \frac{K \cdot S_t \cdot r}{\tan\beta} \int_0^{\phi} \sin^2\phi \, d\phi$$

$$= F_u \quad \frac{\phi}{2}\Big|_0^{\phi} - \frac{\sin 2\phi}{4}\Big|_0^{\phi}$$

$$= \frac{F_u}{2}\left[\phi - \frac{\sin 2\phi}{2}\right]$$

Phase B

This phase, illustrated in figure 82, is specified by:

$$\delta \leq \phi \leq \phi_a$$

Here, it becomes somewhat simpler to measure the position of the cutting edge by the angle ϕ_T which gives the position of the trailing edge. It is then obvious that:

$$y = \frac{r\ (\phi - \phi_T)}{\tan\beta}$$

or $$\phi = \frac{y \tan\beta}{r} + \phi_T$$

Then integrating with respect to y:

$$F_t = K \cdot S_t \int_0^b \sin\left\{\frac{\tan\beta}{r}y + \phi_T\right\}dy$$

$$= K \cdot S_t \int_0^b \left[\sin \frac{\tan\beta}{r} y \cos\phi_T + \cos \frac{\tan\beta}{r} y \sin\phi_T\right] dy$$

$$= K \cdot S_t \left[\frac{-r}{\tan\beta} \cos\phi_T \cos \frac{\tan\beta}{r} y \Big|_0^b + \frac{r}{\tan\beta} \sin\phi_T \sin \frac{\tan\beta}{r} y \Big|_0^b\right]$$

$$= F_u \left[\cos\phi_T \left\{1 - \cos\frac{\tan\beta}{r} b\right\} + \sin\phi_T \sin\frac{\tan\beta}{r} b\right]$$

But, $\delta = \frac{\tan\beta}{r} b$

Thus, $$F_t = F_u \left[\cos\phi_T - \cos\left\{\phi_T + \delta\right\}\right]$$

$$= F_u\left[\cos\left\{\phi - \delta\right\} - \cos \phi\right]$$

Now, integrate the x component of the tangential force.

$$F_{t_x} = -K \quad S_t \int_0^b \sin\left\{y \cdot \frac{\tan\beta}{r} + \phi_T\right\} \cos\left\{y \quad \frac{\tan\beta}{r} + \phi_T\right\} dy$$

Let $\frac{\tan\beta}{r} = m$ for simplicity.

$$\therefore \quad F_{t_x} = -K \cdot S_t \int_0^b \left\{\sin my \cos\phi_T + \cos my \sin\phi_T\right\}\left\{\cos my \cos\phi_T - \sin my \sin\phi_T\right\} dy$$

$$= -K \cdot S_t \left[\cos^2\phi_T \int_0^b \sin my \cos my \, dy - \sin\phi_T \cos\phi_T \cdot \int_0^b \sin^2 my \, dy + \sin\phi_T \cos\phi_T \int_0^b \cos^2 my \, dy - \sin^2\phi_T \int_0^b \cos my \sin my \, dy\right.$$

$$= \frac{-K \cdot S_t}{m}\left[\cos^2\phi_T \frac{\sin^2 my}{2}\Big|_0^b + \sin\phi_T \cos\phi_T \left\{\frac{my}{2} + \frac{1}{4}\sin 2my - \frac{my}{2} + \frac{1}{4}\sin 2my\right\} - \sin^2\phi_T \frac{\sin^2 my}{2}\Big|_0^b\right]$$

$$= \frac{-Fu}{2}\left[\left\{\cos^2\phi_T - \sin^2\phi_T\right\}\left\{\sin^2 mb - 0\right\} + \sin\phi_T \cos\phi_T (\sin 2mb - 0)\right]$$

But, $mb = \delta$ and $\phi_T = \phi - \delta$

Thus, $$F_{t_x} = -\frac{Fu}{2}\left[\left\{\cos^2(\phi - \delta) - \sin^2(\phi - \delta)\right\} \sin^2\delta + \sin(\phi - \delta) \cdot \cos(\phi - \delta) \cdot \sin 2\delta\right]$$

Using trigonometric identities, the result is:

$$F_{t_x} = \frac{-Fu}{2}\left[\sin^2\phi_T - \sin^2(\phi - \delta)\right]$$

Then, the y component of the tangential force is integrated in phase B.

$$dF_{t_y} = K \cdot S_t \sin^2\phi \, dy$$

Using $\phi = \frac{y \tan\beta}{r} + \phi_T$, the integration is carried out with respect to y. Also, using trigonometric identities and letting $m = \frac{\tan\beta}{r}$,

$$\sin^2\phi = \sin^2 my \cos^2\phi_T + 2 \sin my \cos my \sin\phi_T \cos\phi_T + \cos^2 my \sin^2\phi_T$$

Thus,

$$F_{t_y} = K \cdot S_t \left[\cos^2\phi_T \int_0^b \sin^2 my\, dy + 2 \sin\phi_T \cos\phi_T \cdot \int_0^b \sin my \cos my\, dy + \sin^2\phi_T \int_0^b \cos^2 my\, dy \right]$$

$$= \frac{K \cdot S_t}{m} \left[\cos^2\phi_T \left\{ \frac{my}{2} \Big|_0^b - \frac{\sin 2my}{4} \Big|_0^b \right\} + 2 \sin\phi_T \cos\phi_T \cdot \frac{1}{2} \sin^2 my \Big|_0^b + \sin^2\phi_T \left\{ \frac{my}{2} \Big|_0^b + \frac{\sin 2my}{4} \Big|_0^b \right\} \right]$$

$$= \frac{Fu}{2} \left[\delta \left\{ \cos^2\phi_T + \sin^2\phi_T \right\} + \frac{\sin 2\delta}{2} \left\{ \sin^2\phi_T - \cos^2\phi_T \right\} + 2 \sin\phi_T \cos\phi_T \sin^2\delta \right]$$

Again, employing various trigonometric identities and using the identity $\phi_T = \phi - \delta$, the final equation becomes:

$$F_{t_y} = \frac{Fu}{2} \left[\delta - \frac{\sin 2\phi}{2} + \frac{\sin 2(\phi - \delta)}{2} \right]$$

Phase C

Figure 83 illustrates the situation in phase C which is specified by:

$$\phi_a \leq \phi \leq \phi_a + \delta$$

Here, the relation $\frac{\tan\beta}{r} dy = d\phi$ is used to integrate dF_t with respect to ϕ.

$$F_t = K \cdot S_t \int_{\phi-\delta}^{\phi_a} \sin\phi \cdot \frac{r}{\tan\beta} d\phi$$

$$= - F_u \cos\phi \Big|_{\phi-\delta}^{\phi_a}$$

$$= F_u \left[\cos(\phi - \delta) - \cos\phi_a \right]$$

The x component of the tangential force is integrated in a similar fashion.

$$F_{t_x} = - F_u \int_{\phi-\delta}^{\phi_a} \sin\phi \cos\phi \, d\phi$$

$$= - \frac{F_u}{2} \left\{ \sin^2\phi \Big|_{\phi-\delta}^{\phi_a} \right\}$$

$$= - \frac{F_u}{2} \left[\sin^2\phi_a - \sin^2(\phi - \delta) \right]$$

$$F_{t_y} = F_u \int_{\phi-\delta}^{\phi_a} \sin^2\phi \, d\phi$$

$$= F_u \left[\frac{\phi}{2} \Big|_{\phi-\delta}^{\phi_a} - \frac{1}{4} \sin 2\phi \Big|_{\phi-\delta}^{\phi_a} \right]$$

$$= \frac{F_u}{2} \left[\phi_a + \delta - \phi - \frac{\sin 2\phi_a}{2} + \frac{\sin 2(\phi-\delta)}{2} \right]$$

The relations as developed are for a type I cut, and are summarized in chapter four. For a type II cut, the angular limits of each phase are altered and the force in phase B is constant, as discussed in chapter four.

The relations thus far discuss the forces acting on a single tooth at various stages of the cut. Determining the number of teeth involved in the cut n, at each angle of cutter rotation ϕ, is the first step towards determining the total force and torque on the cutter. At each angle ϕ_i, the torque and force on each tooth in the cut are added:

$$T = r \cdot \sum^{n} F_{t_i}$$

$$F = \sqrt{\left\{\sum^{n} F_{t_{x_i}} + F_{r_{x_i}}\right\}^2 + \left\{\sum^{n} F_{t_{y_i}} + F_{r_{y_i}}\right\}^2}$$

The computer routine listed in appendix B calculates F_{t_i}, $F_{t_{x_i}}$, $F_{t_{y_i}}$, $F_{r_{x_i}}$, $F_{r_{y_i}}$ and n for each value of ϕ and produces plots of T (ϕ) and F (ϕ).

Now consider the forces produced as the cutter enters the workpiece. Figure 32 illustrates that the depth of cut a increases from zero to a maximum as the cutter advances into the material. Throughout the ensuing analysis, the transient depth of cut a' will be used. As before,

$$\delta = \frac{b}{r} \cdot \tan\beta$$

A new term ϕ_1 is introduced to delimit the initial phase A_1 which decreases as the tool enters the cut:

$$\phi_1 = \text{arc sin}\left(\frac{r - a'}{r}\right)$$

and $\quad \phi_a = \pi - \phi_1$

Throughout phase A_1, the tooth is not in contact with the workpiece. Thus, the cutting force is zero.

Phase A

In this phase which is specified by $\phi_1 \leq \phi \leq \phi_1 + \delta$, the cut extends from y = c to y = b where c is any arbitrary point along the edge.

$$F_t = K \cdot S_t \int_c^b \sin\phi \, dy$$

and $$\phi = \frac{(y - c)\tan\beta}{r} + \phi_1$$

$$F_t = K \cdot S_t \int_c^b \sin\left\{\frac{(y - c)\tan\beta}{r} + \phi_1\right\} dy$$

Let $\dfrac{\tan\beta}{r} = m$

$$F_t = K \cdot S_t \int_c^b \left\{\sin my \cos mc - \cos my \sin my\right\} \cdot \cos\phi_1 + \left\{\cos my \cos mc + \sin my \sin mc\right\} \cdot \sin\phi_1 \cdot dy$$

$$= K \cdot S_t\left[\frac{\cos\phi_1}{m}\left\{- \cos mc \cos my \Big|_c^b - \sin mc \sin my \Big|_c^b\right\}\right.$$

$$+ \frac{\sin\phi_1}{m}\left\{\cos mc \sin my \Big|_c^b - \sin mc \cos my \Big|_c^b\right\}\Bigg]$$

$$= K \cdot S_t\left[\frac{\cos\phi_1}{m}\left\{-\cos mc \cos mb + \cos^2 mc - \sin mc \sin mb\right.\right.$$

$$\left.\left. + \sin^2 mc\right\} + \frac{\sin\phi_1}{m}\left\{\cos mc \sin mb - \sin mc \cos mb\right\}\right]$$

$$= K \cdot S_t\left[\frac{\cos\phi_1}{m}\left\{1 - \cos (mb - mc)\right\} + \frac{\sin\phi_1}{m}\left\{\sin (mb - mc)\right\}\right]$$

But, $m = \dfrac{\tan\beta}{r}$

$$F_t = F_u\left[\cos\phi_1 - \cos\phi_1 \cos\left\{\frac{\tan\beta}{r}(b - c)\right\} + \sin\phi_1 \sin\right.$$

$$\left.\left\{\frac{\tan\beta}{r}(b - c)\right\}\right]$$

$$= F_u\left[\cos\phi_1 - \cos\left\{\phi_1 + \frac{\tan\beta}{r}(b - c)\right\}\right]$$

Since $\phi = \phi_1 + \dfrac{\tan\beta}{r}(b - c)$,

therefore;

$$F_t = F_u\left[\cos\phi_1 - \cos\phi\right]$$

Now, the x-component of the tangential force:

$$F_{t_x} = -K \cdot S_t \int_c^b \sin\phi \cos\phi \, dy$$

But, $d\phi = dy \cdot \dfrac{\tan\beta}{r}$

$$F_{t_x} = -F_u \int_{\phi_1}^{\phi} \sin\phi \cos\phi \, d\phi$$

$$= -F_u \cdot \frac{\sin^2\phi}{2}\Bigg|_{\phi_1}^{\phi}$$

Therefore:

$$F_{t_x} = -\frac{F_u}{2}\left[\sin^2\phi - \sin^2\phi_1\right]$$

Similarly, the y-component of the tangential force is integrated:

$$F_{t_y} = F_u \int_{\phi_1}^{\phi} \sin^2\phi \, d\phi$$

$$= \frac{F_u}{2}\left[\phi \Big|_{\phi_1}^{\phi} - \frac{\sin 2\phi}{2} \Big|_{\phi_1}^{\phi}\right]$$

Therefore:

$$F_{t_y} = \frac{F_u}{2}\left[\phi - \phi_1 - \frac{\sin 2\phi}{2} + \frac{\sin 2\phi_1}{2}\right]$$

The equations governing the force in phases B and C remain as in the steady state slab milling model. The equations listed govern a type I cut; the changes required for a type II cut are identical to those required in the steady state case.

These forces acting on a tooth are combined to give the total force and torque on the end mill in the same manner as the steady state model. However, the transient depth of cut a' is a function of time; it increases from zero to a maximum value equal to the radius of the tool when it is cutting with the full diameter. Therefore, the computer routine written to calculate the total force and torque increases a' (according to the feed per revolution) with the rotation of the cutter. Thus, the plots of force and torque indicate their transient nature as the tool enters the workpiece. The computer routine is listed in appendix B.

APPENDIX B

Cutting Force Computer Program

```
      PROGRAM FORCE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE10)
C
C ***************************************************************
C   CUTTING FORCE SIMULATION - END MILLING...
C   ***************************************************************
C  WRITTEN BY P. MACNEIL       FEB.1975
C   ***************************************************************
C
C
      DIMENSION FTANG(1800),FRAD(1800),ANGL(1800)
      DIMENSION AO(80)
      DIMENSION ARRAY(200),WORD(5),XNUMB(5)
      DIMENSION XLAB1(3),XLAB2(3)
      COMMON /BLOCK/ PHIA,PHI1,DELTA,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,
     1  TYPFLG,KLM,IKLM,FU,FU2,AFLAG
      REAL K
      INTEGER TYPFLG,TRANS
      DATA BETA / 0.5236 /
C   THIS ASSUMES THAT THE HELIX ANGLE OF CUTTER IS 30 DEGREES.
      DATA  WORD/5HST =  ,5HR  =  ,5HA  =  ,5HB  =  ,5HZ  =  /
      DATA XLAB1/10HTORQUE(IN-  ,10HFRAD  (LB)  ,10HALPHA (DEG  /
      DATA XLAB2 /3HLB)  ,3H    ,3H)   /
      DATA NPLOT / 1 /
      DO 804  MMM=1,7
      READ(5,20) TRANS
      READ(5,21) ST,R,A,B,Z
C   IF TRANS = 1, INPUT DATA REFERS TO A TRANSIENT CASE
C   OF RUNNING INTO THE WORKPIECE.
C   ST IS THE FEED PER TOOTH IN INCHES.
C   R IS THE RADIUS OF THE CUTTER IN INCHES.
C   A IS THE DEPTH OF CUT IN INCHES OR THE TRANSIENT
C   DEPTH OF CUT IN INCHES.
C   B IS THE WIDTH OF CUT IN INCHES.
C   ( WIDTH OF CUT IS PARALLEL TO THE AXIS OF THE CUTTER.)
C   Z IS THE NUMBER OF TEETH ON THE CUTTER.
      XLNK = 11.76 - 0.17* ALOG(ST)
C   XLNK IS THE NATURAL LOG OF K .
      K = EXP( XLNK )
C   K IS THE SPECIFIC FORCE IN LB/IN WHEN ST IS IN INCHES.
C
C   IF Z= 1, IT IS ASSUMED THAT ONLY THE FORCES ON
C   ONE TOOTH ARE REQUIRED.
C
      FU = K*ST*R/TAN(BETA)
      FU2 = FU/2.0
      DELTA = B*TAN(BETA)/R
      RAR = ((R-A)/R)
      IF(RAR.GT.1.0)  RAR = 1.0
      IF(RAR.LT.-1.0) RAR = -1.0
      IF(TRANS.EQ.1) GO TO 3
C   THIS IS FOR SLAB MILLING
      PHI1 = 0.
      PHIA = ACOS(RAR)
      GO TO 4
C   THIS IS FOR A TRANSIENT CASE.
3     PHI1 = ASIN(RAR)
      IF(PHI1.LT.0.0) PHI1 = 0.0
      PHIA = 3.14159 - PHI1
C   DELA IS SET FOR 5 IPM. AND 500 RPM.
```

```
      DELA = 0.0000833
4     DO 5 I=1,200
      FTANG(I) = 0.0
5     FRAD(I) = 0.0
      ANGL(I) = 0.0
      NN2=200
      IF(TRANS.EQ.1)NN2=1800
      NN3= NN2-1
      DO 2 I=1,NN3
2     ANGL(I+1) = ANGL(I) + 0.05236
C     DETERMINE VALUES OF Q CONSTANTS AND TYPE OF CUT.
      CALL CONST
C     NOW BEGIN CALCULATING FORCES.
      NN1=10
      IF(TRANS.EQ.1) NN1=80
      DO 10 IZ=1,NN1
      AQ(IZ) = (2. * 3.14159)*FLOAT(IZ - 1) /Z
10    CONTINUE
      DO 11 KLM = 1,NN2
      IKLM = KLM
      DO 12 IZ= 1,NN1
      ANAQ = ANGL(KLM) - AQ(IZ)
      IF(AQ(IZ).LE.ANGL(KLM).AND.ANGL(KLM).LE.(Q2+AQ(IZ)) )CALL
     1FSUB(FT,FRX,FRY   ,ANAQ)
12    CONTINUE
      FTANG(KLM) = FT
      FRAD(KLM) = SQRT(FRX*FRX + FRY*FRY)
      ANGL(KLM) = ANGL(KLM) * 180./3.14159
      IF(TRANS.NE.1) GO TO 11
C     IF A TRANSIENT CASE, INCREASE DEPTH OF CUT WITH
C     EACH INCREMENT OF PHI .
      A= A+ DELA
      IF(A.GT.R) A=R
      RAR = 1.-A/R
      IF(RAR.GT.1.0)RAR=1.0
      IF(RAR.LT.-1.0)RAR=-1.0
      PHI1=ASIN(RAR)
      IF(PHI1.LT.0.0)PHI1=0.0
      PHIA= 3.14159-PHI1
C     DTERMINE NEW Q VALUES AND TYPE OF CUT .
      CALL CONST
11    CONTINUE
C
20    FORMAT(I1)
21    FORMAT(F8.1)
C     ********************************************
C
C     NOW BEGIN PLOTTING ROUTINE.
C
C
C     FIND LARGEST Y COORDINATE TO BE PLOTTED.
      Y1=FRAD(NN2)
      DO 800 JJ=1,NN3
      Y1 = AMAX1(FRAD(JJ),Y1)
800   CONTINUE
      WRITE(6,444) Y1
444   FORMAT(1H , *  FORCE MAX =    *,F8.1)
C     DETERMINE SCALING FACTORS.
```

```
      VMAX = 5.0
      HMAX = 7.0
      IF(TRANS.EQ.1)HMAX = 20
      VSCALE = VMAX/Y1
      HSCALE= HMAX/ANGL(NN2)
      IF(NPLOT.NE.1)GO TO 801
      NPLOT = NPLOT + 1
C     LABEL PLOT WITH NAME OF USER.
      CALL PLOT(1.0,1.0,2)
      CALL LETTER(7,0.5,90.0,1.0,1.0,7HMACNEIL    )
C     RESET ORIGIN FOR AXES OF PLOT.
801   CALL PLOT(4.75,2.25,-3)
      CALL PLOT(0.0,5.0,2)
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(HMAX,0.0,2)
C     PLOT THE FORCE VERSUS ANGLE.
      Y2= FRAD(1)*VSCALE
      CALL PLOT(0.0,Y2,3)
      DO 802 KK=2,NN2
      Y2=FRAD(KK)*VSCALE
      A2=ANGL(KK)*HSCALE
      CALL PLOT(A2,Y2,2)
802   CONTINUE
      XLL = HMAX + 3.0
      CALL PLOT(XLL,-2.25,-3)
804   CONTINUE
      CALL LETTER(11,0.5,90.0,4.0,0.0,11HEND OF TAPE   )
      CALL PLOT(7.0,0.0,3)
      CALL PLOT(X,Y,999)
      STOP
      END
      SUBROUTINE FSUR(FT,FRX,FRY ,PHI)
      COMMON /BLOCK/ PHIA,PHI1,DELTA,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,
     1  TYPFLG,KLM,IKLM,FU,FU2,AFLAG
      INTEGER TYPFLG
      IF(IKLM.NE.KLM) GO TO 101
      FT = 0.0
      FX=0.0
      FY= 0.0
      FRX = 0.0
      FRY = 0.0
      FTNG = 0.0
101   IF(PHI.LE.PHI1) GO TO 10
      IF(PHI1.LT.PHI.AND.PHI.LE.Q1.AND.TYPFLG.EQ.1)GO TO 20
      IF(PHI1.LT.PHI.AND.PHI.LE.PHIA.AND.TYPFLG.EQ.2)GO TO 20
      IF(Q1.LT.PHI.AND.PHI.LE.PHIA.AND.TYPFLG.EQ.1)GO TO 30
      IF(PHIA.LT.PHI.AND.PHI.LE.Q1.AND.TYPFLG.EQ.2)GO TO 40
      IF(PHIA.LT.PHI.AND.PHI.LE.Q2.AND.TYPFLG.EQ.1)GO TO 50
      IF(Q1.LT.PHI.AND.PHI.LE.Q2.AND.TYPFLG.EQ.2)GO TO 50
      WRITE(6,6)PHI
6     FORMAT(1H0,*  ERROR     PHI = *,F13.5)
      GO TO 1111
C     PHASE A-PRIME.
10    FTNG  = 0.0
      C1 = 0.0
      C2 = 0.0
      GO TO 100
C     PHASE A
```

```
20     FTNG   = FU*(Q3 - COS(PHI))
       C1 = PHI - PHI1 -(SIN(2*PHI))/2.0 + Q5
       C2 =(SIN(PHI))**2 - Q6
        GO TO 100
C     PHASE B FOR TYPE 1
30     PHIDEL = PHI - DELTA
       FTNG = FU* ( COS(PHIDEL) - COS(PHI) )
       C1 = DELTA -(SIN(2*PHI))/2.0 +(SIN(2*(PHIDEL)))/2.0
       C2 = (SIN(PHI))**2 -(SIN(PHIDEL))**2
       GO TO 100
C     PHASE B FOR TYPE 2.
40     FTNG   = FU*(Q3-Q4)
       C1 = PHIA - PHI1 - Q7 + Q5
       C2 = Q8 - Q6
        GO TO 100
C     PHASE C.
50     PHIDEL = PHI - DELTA
       FTNG   = FU*(COS(PHIDEL)- COS(PHIA))
       C1 = Q2- PHI- Q7 +(SIN(2*PHIDEL))/2.0
       C2 = Q8 - (SIN(PHIDEL))**2
C     FINAL CALCULATION OF FRADIAL.
100    FX   =-FU2 *(0.3*C1   +  C2)
       FY   = FU2*(C1 -0.3*C2)
       FT = FT + FTNG
       FRX=FRX + FX
       FRY = FRY + FY
1111   IKLM = IKLM + 1
       RETURN
       END
       SUBROUTINE CONST
C
C    DETERMINES THE TYPE OF CUT AND VALUES FOR Q CONSTANTS.
C
       COMMON /BLOCK/ PHIA,PHI1,DELTA,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,
      1  TYPFLG,KLM,IKLM,FU,FU2,AFLAG
       INTEGER TYPFLG
       Q1 = PHI1 + DELTA
       Q2 = PHIA + DELTA
       Q3 = COS(PHI1)
       Q4= COS(PHIA)
       Q5 = (SIN(2*PHI1))/2.0
       Q6 = (SIN(PHI1))**2
       Q7 = (SIN(2*PHIA))/2.0
       Q8 = (SIN(PHIA))**2
C   IF DELTA LE PHIA, IT IS A TYPE 1.
       IF(  Q1  .LE.PHIA) TYPFLG = 1
       IF(  Q1  .GT.PHIA) TYPFLG = 2
       RETURN
         END
```

CDTOT   0224

APPENDIX C

A/C Assembly Language Routine

```
0001                       ASMB,A,B,L
0003  02024                      ORG 2024B
0004  00012                ADC EQU 12B
0005  02024 002400               CLA
0006  02025 172215               STA TIME,I
0007  02026 102112               STF ADC
0008  02027 026042               JMP INPUT
0009  02030 002400         WAIT  CLA
0010  02031 072252               STA COUNT
0011  02032 162215               LDA TIME,I
0012  02033 042237               ADA DELAY
0013  02034 002021               SSA,RSS
0014  02035 026042               JMP INPUT
0015  02036 162215               LDA TIME,I
0016  02037 042226               ADA TIME1
0017  02040 172215               STA TIME,I
0018  02041 026030               JMP WAIT
0019*  BEGIN READING A TO D CONVERTERS
0020  02042 002400         INPUT CLA
0021  02043 003000               CMA
0022  02044 102612               OTA ADC
0023  02045 102712               STC ADC
0024  02046 102312               SFS ADC
0025  02047 026046               JMP *-1
0026  02050 102512               LIA ADC
0027  02051 106712               CLC ADC
0028  02052 072256               STA TEMP1
0029  02053 012225               AND CODE
0030  02054 072257               STA TEMP2
0031  02055 066257               LDB TEMP2
0032  02056 062256               LDA TEMP1
0033  02057 012224               AND SORCE
0034  02060 052221               CPA CHAN1
0035  02061 076253               STB F1
0036  02062 052222               CPA CHAN2
0037  02063 076254               STB F2
0038  02064 052223               CPA CHAN3
0039  02065 076255               STB T
0040  02066 002400               CLA
0041  02067 102612               OTA ADC
0042  02070 103712               STC ADC,C
0043  02071 106712               CLC ADC
0044  02072 062252               LDA COUNT
0045  02073 052232               CPA THREE
0046  02074 026103               JMP ABSOL
0047  02075 036252               ISZ COUNT
0048  02076 062246               LDA NEGTN
0049  02077 072251               STA SLOW
0050  02100 036251               ISZ SLOW
0051  02101 026100               JMP *-1
0052  02102 026042               JMP INPUT
0053* TAKE THE ABSOLUTE VALUE OF INPUTS
0054  02103 062253         ABSOL LDA F1
0055  02104 016204               JSB ABS
0056  02105 072253               STA F1
0057  02106 062254               LDA F2
0058  02107 016204               JSB ABS
```

```
0059  02110 072254        STA F2
0060  02111 062255        LDA T
0061  02112 016204        JSB ABS
0062  02113 072255        STA T
0063* SORT F1 & F2 IN PREP. FOR SQUARE ROOT.
0064  02114 062253  SORT  LDA F1
0065  02115 003004        CMA,INA
0066  02116 042254        ADA F2
0067  02117 002021        SSA,RSS
0068  02120 026126        JMP SORT2
0069  02121 062253  SORT1 LDA F1      HERE,F1 > F2
0070  02122 072261        STA P
0071  02123 062254        LDA F2
0072  02124 072262        STA Q
0073  02125 026132        JMP ROOT
0074  02126 062253  SORT2 LDA F1   HERE, F1< F2
0075  02127 072262        STA Q
0076  02130 062254        LDA F2
0077  02131 072261        STA P
0078* CALC. SQUARE ROOT OF F1**2+F2**2
0079  02132 062262  ROOT  LDA Q
0080  02133 100200        MPY Q
      02134 002262
0081  02135 100400        DIV P
      02136 002261
0082  02137 001100        ARS
0083  02140 072263        STA Y
0084  02141 100200        MPY Y
      02142 002263
0085  02143 100400        DIV P
      02144 002261
0086  02145 001100        ARS
0087  02146 072264        STA Z
0088  02147 100200        MPY Y
      02150 002263
0089  02151 100400        DIV P
      02152 002261
0090  02153 042261        ADA P
0091  02154 042263        ADA Y
0092  02155 072260        STA RR
0093  02156 062264        LDA Z
0094  02157 003004        CMA,INA
0095  02160 042260        ADA RR
0096  02161 072267        STA FORCE
0097  02162 126214        JMP PQLCY,I
0098* FORCE = P+Q**2/2*P - Q**4/8*P**3 + Q**6/16*P**5
0099* THIS ROUTINE ENSURES THAT 30 < FEED <3000
0100  02163 062265  NEWFD LDA DELFD
0101  02164 142220        ADA FEED,I
0102  02165 172220        STA FEED,I
0103  02166 042242        ADA FMIN
0104  02167 002020        SSA
0105  02170 026177        JMP MIN
0106  02171 042243        ADA FMAX
0107  02172 002020        SSA
0108  02173 026201        JMP RETRN
      02174
```

```
02175 172220        STA FEED,I     FEED = 3000
02176 026201        JMP RETRN
02177 062241  MIN   LDA FDMIN
02200 172220        STA FEED,I     FEED = 30
02201 002400  RETRN CLA
02202 172215        STA TIME,I   CLEAR TIME COUNTER
02203 026042        JMP INPUT
* THIS ROUTINE PLACES THE SIGN BIT OF THE 10-BIT INPUT
* WORD IN THE M.S.B. AND TAKES THE ABS. VALUE.
02204 000000  ABS   NOP
02205 100040        LSL 16
02206 100046        LSL 6
02207 101020        ASR 16
02210 101026        ASR 6
02211 002020        SSA
02212 003004        CMA,INA
02213 126204        JMP ABS,I
*
*
*
02214 002500  POLCY OCT 2500
02215 020000  TIME  OCT 20000
02216 020001  FOPT  OCT 20001
02217 020003  TOPT  OCT 20003
02220 020004  FEED  OCT 20004
02221 010000  CHAN1 OCT 10000
02222 020000  CHAN2 OCT 20000
02223 040000  CHAN3 OCT 40000
02224 070000  SORCE OCT 70000
02225 001777  CODE  OCT 01777
02226 000012  TIME1 DEC 10
02227 000372  TIME2 DEC 250
02230 000001  ONE   DEC 1
02231 000002  TWO   DEC 2
02232 000003  THREE DEC 3
02233 000144  HUN   DEC 100
02234 000040  THIRT DEC 32
02235 000017  FIFTN DEC 15
02236 002000  TND24 DEC 1024
02237 166170  DELAY DEC -5000
02240 005670  FDMAX DEC 3000
02241 000036  FDMIN DEC 30
02242 177742  FMIN  DEC -30
02243 172146  FMAX   DEC -2970
02244 176600  NLIM  DEC -640
02245 001200  PLIM  DEC 640
02246 177766  NEGTN DEC -10
02247 000146  HUNTO DEC 102
02250 072460  BIG   DEC 30000
02251 000000  SLOW  NOP
02252 000000  COUNT NOP
02253 000000  F1    NOP
02254 000000  F2    NOP
02255 000000  T     NOP
02256 000000  TEMP1 NOP
02257 000000  TEMP2 NOP
02260 000000  RR    NOP
```

```
02261 000000  P      NOP
02262 000000  Q      NOP
02263 000000  Y      NOP
02264 000000  Z      NOP
02265 000000  DELFD  NOP
02266 000000  ERR.2  NOP
02267 000000  FORCE  NOP
02270 000000  F.ERR  NOP
02271 000000  ERROR  NOP
02272 000000  T.ERR  NOP
02273 000000  FACTR  NOP
02274 000000  ACC    NOP
02275 000000  EPREV  NOP
02276 000000  B      NOP
02277 000000  DAMP   NOP
*
*
*
02500                ORG 2500B
* CALC. FORCE & TORQUE ERRORS.
02500 062267         LDA FORCE
02501 003004         CMA,INA
02502 142216         ADA FOPT,I
02503 100200         MPY THIRT
02504 002234
02505 100400         DIV FOPT,I
02506 102216
02507 072270         STA F.ERR
02510 062255         LDA T
02511 003004         CMA,INA
02512 142217         ADA TOPT,I
02513 100200         MPY THIRT
02514 002234
02515 100400         DIV TOPT,I
02516 102217
02517 072272         STA T.ERR
* USE SMALLER OF F.ERR & T.ERR AS CONTROLLING ERROR
02520 003004         CMA,INA
02521 042270         ADA F.ERR
02522 002020         SSA
02523 026527         JMP FSMAL
02524 062270  TSMAL  LDA F.ERR
02525 072271         STA ERROR
02526 026531         JMP *+3
02527 062270  FSMAL  LDA F.ERR      F.ERR < T.ERR
02530 072271         STA ERROR
02531 100200         MPY ERROR
02532 002271
02533 072266         STA ERR.2
02534 042244         ADA NLIM
02535 002021         SSA,RSS
02536 026577         JMP EBIG
02537 062266  ESMAL  LDA ERR.2
02540 100200         MPY ONE
02541 002230
02542 072273         STA FACTR
02543 162215         LDA TIME,I
```

```
02544 103101          CLO
02545 042227          ADA TIME2
02546 006400          CLB
02547 100400          DIV HUN
02550 002233
02551 100200          MPY FACTR
02552 002273
02553 100400          DIV TND24
02554 002236
02555 102201          SOC
02556 026602          JMP DLMAX
02557 072274          STA ACC
02560 062275          LDA EPREV
02561 003004          CMA,INA
02562 042271          ADA ERROR
02563 100200          MPY B
02564 002276
02565 072277          STA DAMP
02566 062271          LDA ERROR
02567 002020          SSA
02570 016612          JSB NEG
02571 062277          LDA DAMP
02572 042274          ADA ACC
02573 072265    E1    STA DELFD
02574 062271          LDA ERROR
02575 072275          STA EPREV
02576 026163          JMP NEWFD
02577 062245    EBIG  LDA PLIM     ERR.2 > 640
02600 072266          STA ERR.2      ERR.2 = 640
02601 026537          JMP ESNAL
02602 062271    DLMAX LDA ERROR
02603 002020          SSA
02604 026607          JMP NMAX
02605 062250    PMAX  LDA BIG
02606 026573          JMP E1
02607 062250    NMAX  LDA BIG
02610 003004          CMA,INA
02611 026573          JMP E1
02612 000000    NEG   NOP
02613 062274          LDA ACC
02614 002021          SSA,RSS
02615 003004          CMA,INA
02616 072274          STA ACC
02617 126612          JMP NEG,I
                      END
**  NO ERRORS*
```

APPENDIX D

A/C Simulation Routine

```
      PROGRAM RESPONS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE10)
C
C     ************************************************************
C     THIS PROGRAM SIMULATES THE RESPONSE OF THE CNC-ADAPTIVE
C     CONTROL MILLING MACHINE IN THE METAL CUTTING RESEARCH LAB.
C     ************************************************************
C     COMPLETED FEB. 1975  BY P. MACNEIL.
C     ************************************************************
C
      DIMENSION X(14),CASE2(14),XPREV(14),CASE1(14)
      DIMENSION V(20),FEED(400),ACTUAL(400)
      DIMENSION TIMSTP(400)
      INTEGER DELFLG,FORFLG
      DATA CASE1/0.1,0.07,2.73,0.026,0.078,8.531,0.046,0.2,
     1 2.3,2.37,0.053,1023.,0.8531,0.0  /
      DATA CASE2/0.02,0.0021,0.2833,0.0028,0.0089,0.9786,0.014,
     $ 1.2135,0.2638,0.2659,0.0052,117.434,0.0979,0.0  /
      DATA TINCR/0.0021/
      DATA FMAX,FMIN/0.5,0.005/
C
C     THE TIME STEP IS SET AT 0.0021 SECONDS.
C
C     FMAX AND FMIN ARE THE MAX. AND MIN. FEEDRATES ALLOWED
C     IN UNITS OF INCHES/SEC.
C
      NPLOT = 1
C
C     FOR MULTIPLE RUNS, INSERT THE STATEMENT....  DO 804 MNO=1,(NO. RUNS)
C
      DO 804  MNO=1,6
      READ(5,31)ICASE,KDELAY,FORFLG
      READ(5,32)DELAY,CONST,FREQ,FOPT,B,GPOS,GNEG
C
      TIMDEL = DELAY*FLOAT(KDELAY+1)
      DO 30 II=1,14
      IF(ICASE.EQ.1) XPREV(II) = CASE1(II)
      IF(ICASE.EQ.2) XPREV(II) = CASE2(II)
30    CONTINUE
C
C     ICASE DETERMINES THE INITIAL CONDITIONS. I.E. CASE 1 OR CASE 2
C     FOR AN OPTIMAL FORCE OF 200 LBS, CASE 1 REPRESENTS AN EXCESSIVE
C     CUTTING FORCE AND CASE 2 REPRESENTS A SUB-OPTIMAL FORCE.
C
C     KDELAY DETERMINES THE TIME LAG BETWEEN CUTTING FORCE AND FEEDRATE
C     A VALUE OF KDELAY=N MEANS TIME LAG IS N+1 TIMES DELAY IN A/C LOOP
C
C     FORFLG IS A FLAG DENOTING THE DESIRED CUTTING FORCE FUNCTION.
C     I.E.  FORFLG = 0 GIVES A LINEAR FORCE AND FORFLG=1 GIVES A
C     SINUSOIDAL FORCE.
C
C     DELAY IS THE ACTUAL DELAY INTHE A/C LOOP GIVEN IN SECONDS.
C
C     CONST IS THE SCALING FACTOR IN THE FORCE FUNCTION.
C
C     FREQ IS THE FREQUENCY IN HZ OF THE FORCE.IT IS ZERO IF LINEAR.
C
C     FOPT IS THE DESIRED (OPTIMAL) CUTTING FORCE IN LBS.
C
C     B IS THE DAMPING FACTOR IN THE A/C ALGORITHM
```

```
C
C   GPOS AND GNEG ARE THE POSITIVE AND NEGATIVE GAINS RESP. IN THE
C   A/C ALGORITHM.
C
      DELFLG = 1
      IF(DELAY.EQ.0.0) DELFLG = 0
C
C   DELFLG = 1 INDICATES A DELAY IN THE A/C LOOP.
C
      WRITE(6,20)
      WRITE(6,21)ICASE
      WRITE(6,22)TINCR,FOPT
      WRITE(6,23)DELAY,CONST
      IF(FORFLG.EQ.0)WRITE(6,24)
      IF(FORFLG.EQ.1)WRITE(6,25) FREQ
      WRITE(6,26) GPOS,GNEG
      WRITE(6,27)B,TIMDEL
      OMEGA = 6.28318*FREQ
      IF(DELFLG.EQ.0) DELAY = TINCR
      T= -TINCR
      X(1) = XPREV(1)
      TIME = 0.0
      IDELAY = 1
      DO 29 IJ=2,14
29    X(IJ) = 0.0
C
C
C
      DO 100 I=1,400
      T = T+ TINCR
      IF(T.EQ.0.0) GO TO 12
C   X(1) IS THE DISTANCE COMMAND IN INCHES.
      X(1) = XPREV(1) + XPREV(13)*TINCR
12    X(4) = XPREV(2)*TINCR + XPREV(4)*(1.0-0.45*TINCR)
      X(5) = XPREV(3)*TINCR + XPREV(5)*(1.0-31.25*TINCR)
C   X(7) IS THE OUTPUT OF THE RESOLVER IN INCHES.
      X(7) = XPREV(7) + TINCR*XPREV(6)/10.0
C   X(11) IS THE INPUT TO THE D/A AND FILTER.
      X(11) = X(1) - X(7)
C   X(10) IS THE OUTPUT FROM THE D/A AND FILTER.
      X(10) = 45.0*X(11)
C   X(6) IS THE MOTOR SPEED IN REV/SEC.
      X(6) = 109.4*X(5)
C   X(8) IS THE TACHO OUTPUT IN VOLTS.
      X(8) = 1.24*X(6)
C   X(9) IS THE VELOCITY FEEDBACK IN VOLTS.
      X(9) = 0.2174*X(8)
C   X(2) IS THE INPUT TO THE CORRECTING NETWORK.
      X(2) = X(10) - X(9)
C   X(3) IS THE OUTPUT OF THE CORRECTING NETWORK.
      X(3) = 2.178*X(2) + 98.01*X(4)
C
C   NOW TO DETERMINE THE CUTTING FORCE X(12)
C
C   THIS LOGIC KEEPS THE CUTTING FORCE LAGGING BEHIND THE FEED BY
C   THE FACTOR TIMDEL.
C
      IF(DELFLG.EQ.0) GO TO 55
```

```
C     ONLY CALCULATE FORCE AND NEW FEEDRATE AFTER CORRECT
C     DELAY TIME.
        IF(T.LT.(DELAY-.0001)) GO TO 10
        IF(KDELAY.EQ.-1) GO TO 59
55      IF(IDELAY.LE.KDELAY) V(IDELAY) = X(6)/10.0
        IF(FORFLG.EQ.0)X(12)=CONST*V(1)
        IF(FORFLG.EQ.1)X(12)=100.0+ABS(CONST*SIN(OMEGA*TIME)*V(1))
        IF(IDELAY.LE.KDELAY) GO TO 56
        KKK = KDELAY -1
        DO 57 M=1,KKK
57      V(M)= V(M+1)
        V(KDELAY) = X(6)/10.0
56      IDELAY= IDELAY + 1
        GO TO 54
59      IF(FORFLG.EQ.0)X(12)=CONST*X(6)/10.0
        IF(FORFLG.EQ.1)X(12)=100.0+ABS(CONST*SIN(OMEGA*TIME)*X(6)/10.0)
C
C
C     **********    ADAPTIVE CONTROL POLICY    *************
C
C     COMPUTE EF THE RELATIVE FORCE ERROR.
C
C
54      EF =(1.0 - X(12)/FOPT  )
        X(14) = EF
C     COMPUTE THE DAMPING TERM B*D(EF)/DT
        DAMP = B *( X(14) - XPREV(14) )/DELAY
C
C     COMPUTE THE REQUIRED ACCELERATIOON
C
        IF(EF.LT.0.0) GO TO 8
        ACC = EF**2*GPOS    + DAMP
        GO TO 51
8       ACC= -EF**2*GNEG + DAMP
51      T = 0.0
        GO TO 9
10      ACC = 0.0
C
C     DETERMINE THE NEW COMMANDED FEEDRATE.
C
9       X(13) = XPREV(13) + ACC*DELAY
C     CHECK THE FEEDRATE LIMITS.
        IF(X(13).GT.FMAX)X(13) = FMAX
        IF(X(13).LT.FMIN)X(13) = FMIN
C
C
C
        DO 11 J = 1,14
11      X-REV(J) = X(J)
C
C     NOW DETERMINE THE ACTUAL AND COMMANDED FEEDRATES (IN UNITS OF I.P.M)
C
102     FEED(I) = X(13)*60.
        ACTUAL(I) = X(8)*60.0/12.4
        TIMSTP(I) = TIME
103     TIME = TIME + TINCR
100     CONTINUE
C
```

```
C
1000    CONTINUE
C**************************************************************************
C
C    NOW BEGIN PLOTTING ROUTINE.
C
C
C    FIND THE LARGEST Y COORDINATE TO BE PLOTTED.
       F1=FEED(400)
       A1=ACTUAL(400)
       DO 800  JJ=1,399
       F1=AMAX1(FEED(JJ),F1)
       A1=AMAX1(ACTUAL(JJ),A1)
800    CONTINUE
       Q1=AMAX1(F1,A1)
C    DETERMINE THE VERTICAL SCALING FACTOR
       VSCALE=5.0/Q1
       VMAX=5.0
       HMAX=7.5
C    DETERMINE THE HORIZONTAL SCALING FACTOR
       HSCALE=7.5/TIMSTP(400)
       IF(NPLOT.NE.1)GO TO 801
C    LABEL PLOT WITH NAME OF USER.
       CALL PLOT(1.0,1.0,3)
       CALL LETTER(7,0.5,90.0,1.0,1.0,7HMACNEIL  )
C    PUT A BORDER AROUND THE PLOT.
801    CALL PLOT(4.0,1.5,-3)
       CALL PLOT(8.5,0.0,2)
       CALL PLOT(8.5,6.0,2)
       CALL PLOT(0.0,6.0,2)
       CALL PLOT(0.0,0.0,2)
C    RESET ORIGIN FOR AXES OF PLOT.
       CALL PLOT(0.75,0.75,-3)
       CALL PLOT(0.0,5.0,2)
       CALL PLOT(0.0,0.0,3)
       CALL PLOT(7.5,0.0,2)
C    PLOT THE COMMANDED FEEDRATE VS. TIME.
       F2=FEED(1)*VSCALE
       CALL PLOT(0.0,F2,3)
       DO 802  KK=2,400
       F2=FEED(KK)*VSCALE
       T1=TIMSTP(KK)*HSCALE
       CALL PLOT(T1,F2,2)
802    CONTINUE
C    PLOT THE ACTUAL FEEDRATE VS. TIME.
       A2=ACTUAL(1)*VSCALE
       CALL PLOT(0.0,A2,3)
       DO 803  LL=2,400
       A2=ACTUAL(LL)*VSCALE
       T1=TIMSTP(LL)*HSCALE
       CALL PLOT(T1,A2,2)
803    CONTINUE
       CALL PLOT(11.0,-2.25,-3)
       NPLOT = NPLOT + 1
804    CONTINUE
       CALL LETTER(11,0.5,90.0,4.0,0.0,11HEND OF TAPE   )
       CALL PLOT(7.0,0.0,3)
       CALL PLOT(X,Y,999)
```

```
805   CONTINUE
      STOP
31    FORMAT(I2)
32    FORMAT(F7.1)
20    FORMAT(1H1,10H**********,* NON-LINEAR A/C ALGORITHM *,
     $ 10H********** //)
21    FORMAT(1H ,5X,* INITIAL CONDITIONS USED ARE CASE *,I1/)
22    FORMAT(1H ,5X ,* TIME INCREMENT = *,F6.4,* SEC*//6X,
     $ * OPTIMAL FORCE  = *,F6.1,* LB */)
23    FORMAT(1H ,5X ,* DELAY IN A/C LOOP = *,F6.4,* SEC*//6X,
     $ * FORCE CONSTANT =  *,F9.1/  )
24    FORMAT(1H ,5X,* LINEAR CUTTING FORCE */)
25    FORMAT(1H ,5X,* SINUSOIDAL CUTTING FORCE WITH FREQ = *,
     $ F8.3,* HZ */)
26    FORMAT(1H ,5X,* POSITIVE GAIN = *,F5.1/6X,* NEGATIVE GAIN = *,
     $ F6.1  / )
27    FORMAT(1H ,5X ,* DAMPING FACTOR =  *,F4.2//6X,
     $ * TIME LAG IN ACHIEVING NEW FORCE = *,F5.3,* SEC *//)
      END
```

REFERENCES

1. Childs, James J. Principles of Numerical Control. New York: Industrial Press Inc., 1965.

2. Akgerman, N. "BCLCNC - A Computer Numerical Control System", Manufacturing Technology Dept. Report. Columbus, Ohio: Battelle's Columbus Laboratories, 1974.

3. Pilafidis, E. J. "Adaptive Control Systems and the Manufacturing Process", Process Research and Development Report. Cincinnatti, Ohio: Cincinnatti Milacron, Inc., May 18, 1970.

4. Yamazaki, Kazuo, et al. "A Study on Adaptive Control in an N/C Milling Machine", 24th General Assembly of CIRP. Kyoto, Japan: Sept., 1974.

5. "On-Line Wear Sensor Readjusts Cutting Tools", Machine Design. Vol. 46, No. 7 (March 21, 1974), p. 31.

6. Dyke, R. M. "Advances in Adaptive Control", Canadian Controls and Instrumentation. (January, 1974), pp. 27-30.

7. "Adaptive Control of Milling Machines", Tool and Manufacturing Engineer. Vol. 62, No. 5 (May, 1969), pp. 50-54. Adapted from ASTME paper MS 68-638.

8. Valek, Robert J. "How to Go Adaptive Control", Tool and Manufacturing Engineer. Vol. 59, No. 1 (July, 1967), pp. 18-21.

9. Beadle, B. R. and J. G. Bollinger. "Computer Adaptive Control of a Machine Tool", Annals of the C.I.R.P.. Vol. XVIV (1971), pp. 61-65.

10. "Adaptive Control", Booklet #BICD-351C. Detroit, Michigan: Bendix Industrial Controls Div., n.d.

11. Centner, R. M. "Adaptive Control for Machine Tools", Frontiers in Manufacturing Technology, Vol. II, (Numerical Control Today), ed. by D. N. Smith and D. M. Peelle. Ann Arbor, Michigan: Industrial Development Div., Institute of Science and Technology, University of Michigan, 1967.

12. Mathias, R. A. "Adaptive Control of the Milling Process", IEEE National Machine Tools Industry Conference. Cleveland, Ohio: Oct., 1967.

13. Mathias, R. A. Adaptive Controlled Profile Milling , paper #MS70-563. Dearborn, Michigan: Society of Manufacturing Engineers, 1970.

14. MacNeil, Peter R. Retrofit of TOS Knee-Type Milling Machine. A Fourth Year Project. Hamilton: Mechanical Engineering Department, McMaster University, March, 1973.

15. Koren, Y. and J. Tlusty. "CNC/AC System for a Milling Machine - Part I", Metalworking Research Group Report No. 43. Hamilton: Mechanical Engineering Department, McMaster University, March, 1974.

16. Mailvaganam, G. N. An Adaptive Control Algorithm for a CNC Milling Machine. Master of Engineering Thesis. Hamilton: McMaster University, April, 1974.

17. Koren, Y. and J. Tlusty. "CNC/Ac System for a Milling Machine - Part III", Metalworking Research Group Report No. 46. Hamilton: Mechanical Engineering Department, McMaster University, June, 1974.

18. Smith, Ralph J. Circuits, Devices and Systems. 2nd ed. New York: John Wiley and Sons, Inc., 1971.

19. Shinners, Stanley M. Modern Control System Theory and Application. 2nd printing. Don Mills, Ont.: Addison-Wesley Publishing Co., June, 1973.

20. Tlusty, J., Y. Koren and P. MacNeil. "Numerical and Adaptive Control for Die Sinking", Proceedings of the International Conference on Production Engineering. Tokyo: Japan Society of Precision Engineering, 1974.

21. Koenigsberger, F. and J. Tlusty. Structures of Machine Tools, Vol. 1. 1st ed. Oxford: Pergamon Press, 1970.