AN ABSTRACT OF THE THESIS OF

FERDİ TAHİR OLGUN for the degree of MASTER OF SCIENCE in
(Name of Student) (Degree)

CHEMICAL ENGINEERING presented on Nov. 18, 1977
(Major Department) (Date)

Title: AN EXECUTIVE FOR SIMULATION OF PROCESS DYNAMICS AND CONTROL

Abstract approved by: Redacted for Privacy
Ferhan Kayıhan

A simple package of routines that can be easily implemented in small minicomputers used in process control, was developed to simulate dynamic systems and computer control of simple dynamic systems found in the process industries.

Two different systems, a gas absorber and a fifth-order transfer function were chosen as processes to be simulated. Several control models, like proportional-integral, direct digital control (DDC), steady-state and dynamic feedforward control and Dahlin's control algorithm were used to control these systems. The gas absorber and the fifth-order transfer function were simulated with an analog computer. The control actions were either done by the analog computer or the NOVA minicomputer. The results were also obtained by simulating the processes and the control actions by using the executive package. The effect of sampling time in DDC control was shown using very fast sampling and slow sampling times. The performance of Dahlin's algorithm for various values of the parameter λ was also shown.

The simulation results were compared with the results obtained from analog computer and NOVA minicomputer. It was observed that the results were the same in all cases which indicates that the executive package developed works effectively for systems similar to the test cases.

An Executive for Simulation of
Process Dynamics and Control

by

Ferdi Tahir Olgun

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed November 18, 1977

Commencement June 1978

APPROVED:

Redacted for Privacy

Professor of Chemical Engineering in charge of major

Redacted for Privacy

Head of Department of Chemical Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented Nov. 18, 1977

Typed by Joan Yvonne Ziobron for Ferdi Tahir Olgun

ACKNOWLEDGEMENT

I would like to express my sincere appreciations:

To Dr. Eugene Elzy for his advice, support and encouragement during the course of my studies.

To Dr. Ferhan Kayıhan for his helpful suggestions.

To Dr. Charles E. Wicks and the rest of the Chemical Engineering Department for their financial and moral support during the past two years.

And to, Mel Olson for his assistance in support of the NOVA minicomputer.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

LIST OF FIGURES

LIST OF FIGURES (cont.)

LIST OF TABLES

AN EXECUTIVE FOR SIMULATION OF PROCESS DYNAMICS AND CONTROL

I. INTRODUCTION

The objective of this work has been the development of a simple package of routines, to simulate dynamic systems and computer control of typical systems found in the process industries. The emphasis has been on the development of simple routines that can be easily implemented on a process control minicomputer.

Very elaborate executive systems are available for simulating dynamic chemical processes (Dynsys (1), Dyflo (2)). But these systems are usually very large because they include numerous, specific equipment routines, i.e. dynamic simulation of a distillation column using dynamic energy and mass balances, which make them difficult to implement on a small computer. In this system, the dynamic simulation is approached by utilizing lower transfer functions which would be determined experimentally. Most of the chemical processes can be approximated by first or second order transfer functions with dead time. (Murrill (3), Smith (4))

Another emphasis was on simulating Direct Digital Control (DDC) algorithms in order to predict the performance of a process control computer before actual implementation. In actual practice, the economic justification of a process control computer in industry, is based upon the benefits which are expected from improved control. This research will assist the engineer in developing an economic justification by simulating computer control in order to predict the improvement that might be obtained. It will also assist in investigating the alternative control techniques so that the method yielding the greatest improvement could be implemented. In this study, supervisory computer control was not implemented but it can also be simulated as well with no difficulty. In practice, the simulation of supervisory digital computer control would utilize different variables and input information such as cost of raw materials, value of products, constraints on the operation and specifications on products, in a model to compute optimum control set points which would be used to modify the set points and the parameters of the control algorithms. The model to determine set points depends on each specific application and once the model is available the simulation is straightforward.

In some applications, the use of material and energy balances may be required or preferred to the use of simple transfer functions, which would require the solution of a set of differential equations. These routines can be implemented easily utilizing the subroutine DSIM to solve the dynamic set of equations. But it is outside the scope of this research to provide equipment routines based on material and energy balances.

The computer programs of the executive system were developed on the Oregon State University Computer (Control Data Corporation (CDC) 3300). The simulation of the processes and their control were done on the Oregon State University Chemical Engineering Department's Computer (Data General NOVA 840) and analog computer (EAI TR.20).

II. THE EXECUTIVE SYSTEM LIBRARY

Basic approach used in the development of the routines in this research is the collection of subroutines which can be utilized to simulate a wide variety of dynamic systems with or without control. The simulation of the system can be either done by solving the differential equations of the system or by using first or second order transfer functions with dead time. The simulation of the control can be done by computing the equations representing the control action or by using appropriate control routines. In every case, the user writes a main program which calls the necessary subroutines to accomplish the simulation required for a particular study.

Main Program

The way the main program is structured for calling subprograms, printing and plotting is the same as Evans' (5) computer program for dynamic simulation. However, it is internally different.

A flow chart of the main program is shown in Fig. 1. Listing of the main program for an example problem is given in Appendix A. The main program consists of five steps:

1. Initialization: Initial conditions, constants, and control parameters are set or read as data in this section.
2. Derivative Calculations: The values of the derivatives are computed using current values of the state variables and time.

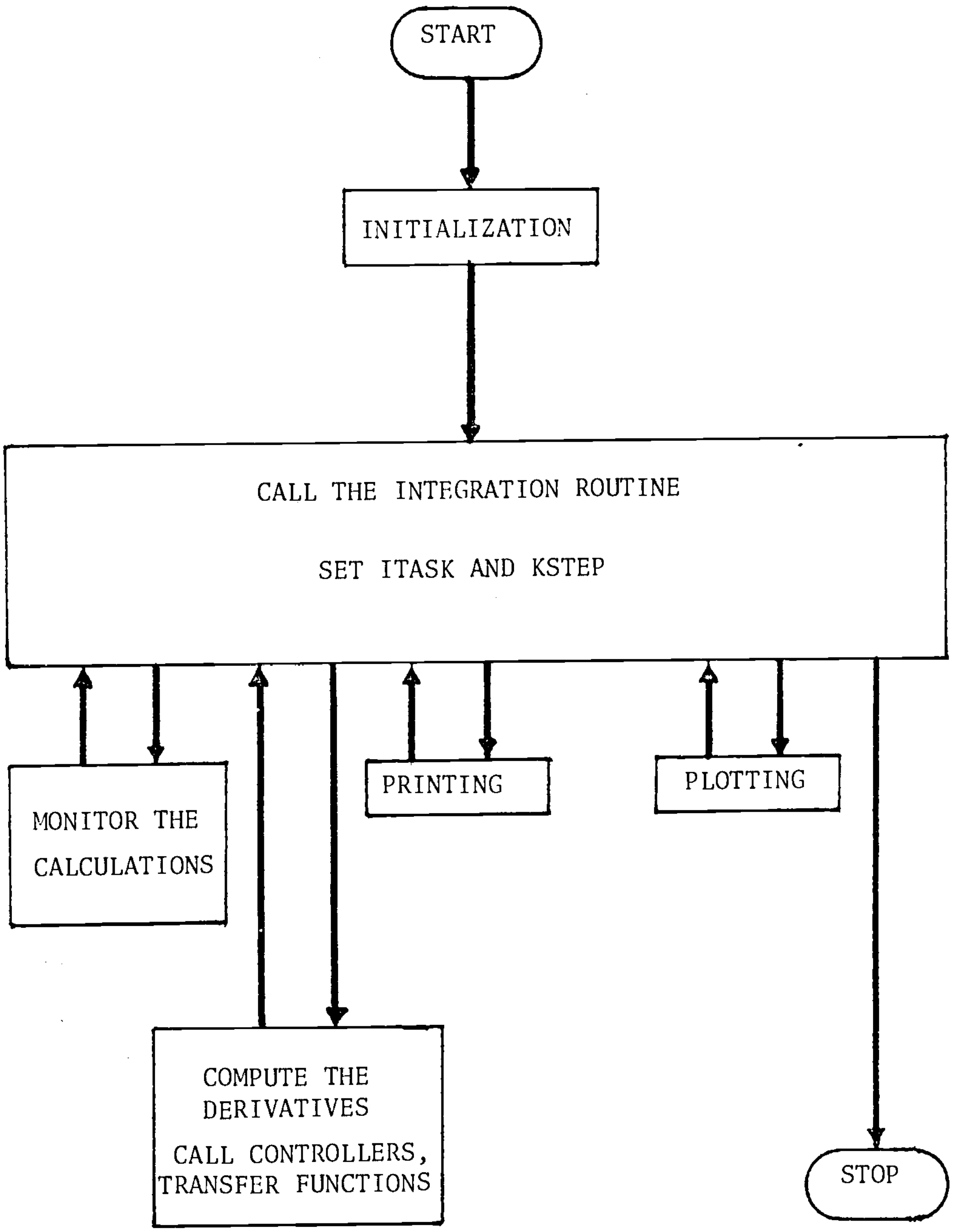


Figure 1. Flow chart of the main calling program

3. Printing: Necessary results are computed and values are printed for any given print interval.
4. Plotting: Necessary results are computed and values are saved for plotting, if desired.
5. Termination: Any remaining calculation is performed and the simulation is completed.

The subroutine DSIM which is called after the initialization conducts the flow of the simulation program. A more general explanation about DSIM and the flow of the simulation is presented in the next section.

Numerical Integration Routine

The key routine in the present study is the subroutine for solving systems of ordinary differential equations (O.D.E.). Numerous techniques are available and have been extensively discussed in the literature. Only algorithms capable of automatic step-size adjustment to maintain a desired accuracy were considered in this work. The state of the art technique most widely recognized for numerical solution of systems of O.D.E. is Gear's method. However, the complexity of Gear's method makes its use very difficult for small minicomputer. The fourth-order Runge-Kutta technique proposed by England (6) has been demonstrated to be efficient and convenient for step-size adjustment when compared with other methods. England discusses several algorithms but the one algorithm using eight evaluations per step was more efficient than others discussed. This Runge-Kutta-England algorithm was implemented in the present research as subroutine DSIM. The coefficients for this method are:

$$k_0 = hf(x_0, y_0)$$

$$k_1 = hf(x_0 + \tfrac{1}{2}h,\ y_0 + \tfrac{1}{2}k_0)$$

$$k_2 = hf(x_0 + \tfrac{1}{2}h,\ y_0 + \tfrac{1}{4}(k_0 + k_1))$$

$$k_3 = hf(x_0 + h,\ y_0 - k_1 + 2k_2)$$

At this point, the value of the variable is calculated:

$$y_1 = y_0 + 1/6(k_0 + 4k_2 + k_3)$$

$$k_4 = hf(x_0 + h,\ y_1)$$

$$k_5 = hf(x_0 + 3/2h,\ y_1 + \tfrac{1}{2} k_4)$$

$$k_6 = hf(x_0 + 3/2h,\ y_1 + \tfrac{1}{4}(k_4 + k_5))$$

$$k_7 = hf(x_0 + 2h,\ y_0 + 1/6(-k_0 - 96k_1 + 92k_2 - 121k_3 + 144k_4 + 6k_5 - 12k_6))$$

The error estimate is calculated at this point:

Error: $r = r_1 + r_2 = \frac{1}{90}(-k_0 + 4k_2 + 17k_3 - 23k_4 + 4k_6 - k_7)$

The absolute value of the estimated error is compared with the given tolerance and if absolute r is greater than the tolerance, all values are discarded and the computation proceeds with an interval of h/2. If r is tolerable, the computation is completed:

$$k_8 = hf(x + 2h,\ y_1 - k_5 + 2k_6)$$

$$y_2 = y_1 + 1/6(k_4 + 4k_6 + k_8)$$

If the absolute value of the error estimate divided by tolerance is less than 0.003 the step-size is doubled for the next step.

DSIM

Subroutine DSIM solves a system of ordinary differential equations using the method discussed above. The organization and the method of calling upon the main program are indicated in Fig. 2.

DSIM is always called by a pair of statements:

```
CALL DSIM (N, T, HPRINT, HPLOT, H, HMIN, TMAX, ERR , E, IERR, ITASK)
GO TO (1, 2, 3, 4, 5) ITASK
```

Where:

N = number of differential equations

T = independent variable, time

HPRINT = intervals at which values are to be printed

HPLOT = intervals at which values are to be plotted

H = current value of the step-size

HMIN = minimum step-size allowable

TMAX = maximum time at which the simulation is to be terminated

ERR = accuracy desired, i.e., if the solution is desired to 0.0001 then ERR = 0.0001

E = estimated errors of the state variables

IERR = flag set to 1 if the step-size is less than HMIN

ITASK = flag set to 1, 2, 3, 4, 5 depending on the task to be performed by the main program

1. Monitor the calculations at each step
2. Compute the derivatives
3. Print out results

```
      PROGRAM MAIN
C
C     INITIALIZATION SECTION
C
C
   10 CALL DSIM(N,T,HPRINT,HPLOT,H,HMIN,TMAX,
     1 E,ERR,IERR,ITASK)
C
C
    1 SECTION TO MONITOR CALCULATIONS
      GO TO 10
C
    2 SECTION TO COMPUTE THE DERIVATIVES
      GO TO 10
C
    3 SECTION TO PRINT RESULTS
      GO TO 10
C
    4 SECTION TO SAVE RESULTS FOR PLOTTING
      GO TO 10
C
    5 TERMINATION
C
C
      END
```

Figure 2. Organization of the main program using DSIM

4. Save results for plotting

5. Terminate the simulation

After the designated task has been performed, control is transferred back to the statement which calls DSIM, except when ITASK = 5.

The values of the state variables X, derivative function values F, and KSTEP values are passed through COMMON/VAL/ during the simulation.

Transfer Functions

TRFN

This routine is implemented to calculate the derivative values for the differential equations representing N first-order transfer functions having the same gains and constants.

TRFN (NXI, NXO, TC, GAIN, N)

NXI = input variable number

NXO = output variable number

TC = time constant

GAIN = gain

N = number of first-order transfer functions

NOTE: The use of the state variable numbers in TRFN containing more than one first-order transfer function can be ambigious. A simple illustration of the use of the state variables for TRFN is given in Fig. 3. The variable numbers from NXO + 1, to NXO + N should not be used by other routines, since they are used internally by TRFN.

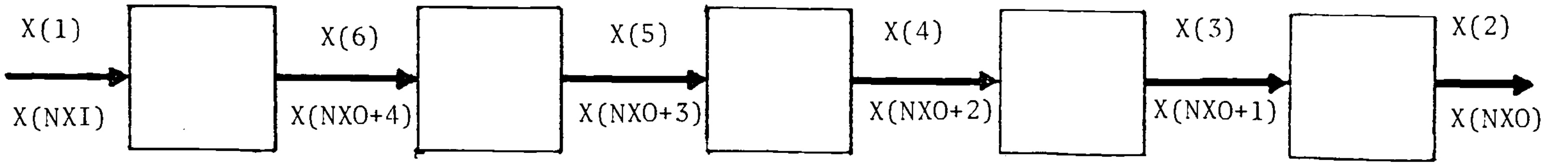


Figure 3. Use of the state variable numbers with the subroutine TRFN

TRF2

This routine is implemented to calculate the derivative values for a second-order transfer function. The denominator of the transfer function can be in the following forms

a) $$\frac{GAIN}{TC^2\ S^2 + 2(TC)(DAMP)S + 1}$$ or

b) $$\frac{GAIN}{((TC)\ S + 1)((TC2)\ S + 1)}$$ or

The first equation will be computed if TC2 is given as zero.

TRF2 (NXI, NXO, TC, DAMP, GAIN, TC2)

NXI = input variable number

NXO = output variable number

TC = first time constant

TC2 = second time constant

DAMP = damping ratio

GAIN = gain

NOTE: The calculations of the derivatives are performed on the space of the variable, NXO + 1. Additional care should be taken not to use the variable number for other purposes.

Controllers

Subroutines are developed to simulate the action for each of the three basic modes of industrial controllers which are:

1. Single mode or proportional
2. Two mode or proportional, integral
3. Three mode or proportional, integral plus derivative

There are two different terminologies used for the parameters of these controllers (2), (7). In the present work the terminology used by Franks in his dynamic simulation package, DYFLO will be used. In Table 1, a listing of the terms and their relations is given.

In addition to these routines, another set of control routines, performing the same action without calculating the derivative values are implemented. These routines allow the user to simulate the effect of sampling time in DDC control for slow sampling times.

PRCONTR (proportional controller)

This routine is implemented to simulate the action of a proportional controller. The input signal is normalized based on the reference point and range of the instrument which are usually zero and 100, and the control equation calculates the output. This subroutine can be also used in DDC control with slow sampling time.

PRCONTR (NXI, NXO, ZR, RAN, ACT, SP, PB, XMN)

PRCONTR (NXI, NXO, ZR, RAN, ACT, SP, GAIN, XMN)

NXI	=	input variable number
NXO	=	output variable number
ZR	=	zero
RAN	=	range
ACT	=	action: + 1. direct, - 1. reverse
PB	=	proportional band
GAIN	=	gain
XMN	=	manual reset, set to a value that will reduce the error close to zero under normal steady-state conditions

PICONTR (Proportional, integral controller)

This routine is implemented to simulate the action of a proportional, integral controller. The equation describing the action is:

$$X\ (NXO) = 100/PB\ (ACT)\ (ERROR + RT \int ERRORdt)$$

The error is the difference between the current value of the state variable X(NXI) and the set point SP.

PICONTR (NXI, NXO, ZR, RAN, SP, ACT, PB, RT)

PRCONTR (NXI, NXO, ZR, RAN, SP, ACT, GAIN, TI)

NXI = input variable number

NXO = output variable number

ZR = zero of the instrument

RAN = range of the instrument

ACT = action: + 1. direct, - 1. reverse

PB = proportional band

GAIN = gain

RT = repeats/unit time

TI = time/repeat

NOTE: The calculations of the integral part are performed on the space of the variable, NSI + 1. Additional care should be taken not to use this variable number for other purposes.

PIDCON (proportional, integral, plus derivative action controller)

This routine is implemented to simulate the action of a proportional, integral, plus derivative action controller. This can be regarded as being merely a PI Controller with derivative action added. This derivative action can be either acting on the error signal or on the input variable. If the action is on the error signal, it will be applied to any change in the input signal as well as in the set-point. Since, this is not desired in the implementation of these routines, the alternate form was used. The derivative section provides an additional change in the output, which is proportional to the rate of change in the input variable. The following integral equation describes the derivative action:

$$VID = RA\ (VI + \frac{1}{RT} \int (VI - VID))dt$$

PIDCON (NXI, NXO, ZR, RAN, SP, ACT, PB, RPT, RT, RA)

PIDCON (NXI, NXO, ZR, RAN, SP, ACT, GAIN, TI, TD, ALFA)

NXI = input variable number

NXO = output variable number

ZR = zero

RAN = range

SP = set point

ACT = action: + 1. direct, - 1. reverse

PB = proportional band

GAIN = gain

RPT = repeats/unit time

TI = time/repeat

RT = rate time

TD = derivative time

RA = rate amplitude

ALFA = alfa

NOTE: The calculations of the integral and derivative parts are performed on the spaces of the variables, NXI + 1 and NXI + 2. Additional care should be taken not to use these variable numbers for other purposes.

DPICON (proportional, integral controller)

This routine is implemented to simulate the action of a proportional, plus integral controller used in direct digital control. The equations used in the subroutine are as follows:

Sum of errors = SUME = SUME + ERROR

X (NXO) = (100/PB) (ERR + SUME (TI) DELT)

DPICON (NXI, NXO, ZR, RAN, SP, ACT, PB, TI, DELT)

NXI = input variable number

NXO = output variable number

ZR = zero

RAN = range

SP = set point

ACT = action = + 1. direct, - 1. reverse

PB = proportional band

TI = repeats/ unit time

DELT = sampling time

NOTE: This routine should be called every time period equal to DELT.

DPIDCON (proportional, integral plus derivative controller

This routine is implemented to simulate the action of a proportional, integral plus derivative action controller used in direct digital control. The derivation for the control equation is given in Appendix E-1.

DPIDCON (NXI, NXO, ZR, RAN, SP, ACT, PB, TI, TR, DELT)

NXI = input variable number

NXO = output variable number

ZR = zero

RAN = range

SP = set point

ACT = action: +1. direct, - 1. reverse

PB = proportional band

TI = repeats/unit time

TR = rate time

DELT = sampling time

NOTE: This routine should be called every time period equal to DELT.

Table 1: Controller Arguments

Proportional band, PB	Gain, GAIN	GAIN = 100/PB
Repeats/unit time, RPT	Time/repeat, TI	TI = I/RPT
Rate time, RT	Derivative time, TD	TD = RT
Rate amplitude, RA	Alfa, ALFA	ALFA = I/RA

Valve Routine

This routine is implemented to calculate the output flow rate of a valve given the stem position. Three different port characteristics are used (2), (8):

1. Linear
2. Square Root
3. Equal Percentage

VALVE (NXI, NXO, P1, P2, LV, KT, VC,R)

NXI = state variable number which corresponds to the stem position

NXO = state variable number which corresponds to the flow rate

P1 = upstream pressure

P2 = downstream pressure

LV = liquid or vapor (0 = vapor)

KT = port characteristics, 1 = linear, 2 = equal %
3 = square root

VC = valve capacity

R = rangeability = $1/A_0$, where A_0 is the valve opening with fully closed stem position

Function Generator Routines

STEP

This routine generates a step function of amplitude (X2 - X1) from time A to time B. At time B the value of the function is set to X3.

STEP (T, A, B, X1, X2, X3)

T = time

A = time value when the step change starts

B = time value when the step change terminates

X1 = function value before time A

X2 = function value for A < time < B

X3 = function value after time B

PEAK

This routine generates a symmetrical triangular peak starting at time A and terminating at time B.

PEAK (T, A, B, X1, X2, X3)

T = time

A = time value when the change starts

B = time value when the change terminates

X1 = function value before time A

X2 = maximum or minimum function value

X3 = function value after time B

SPULSE

This routine generates a half sinusoidal function of amplitude (X2 - X1) starting at time A and terminating at time B.

SPULSE (T, A, B, X1, X2, X3)

T	=	time
A	=	time value when the change starts
B	=	time value when the change terminates
X1	=	function value before time A
X2	=	maximum or minimum function value
X3	=	function value after time B.

Convergence and Arbitrary Function Generator Routines

These routines are identical to the routines used in the dynamic simulation executive DYFLO (2). They were included in this executive system to enlarge the capacity. Their use is shown in the example problem given in Appendix B-2.

CONV

This routine is used for the algebraic convervence of a variable. The method is based on Weigstein's technique for algebraic convergence.

CONV (X, Y, NR, NC)

X = trial value

Y = calculated value

NR = routine call number

NC = converge index (NC = 1, convergence)

FUN

This routine calculates a value of Y for a particular value of X. The coordinates of each point should be stored in X and Y arrays in the calling program. A linear interpolation is done around the adjacent coordinate points of X for intermediate values.

FUN (A, N, X, Y)

A = input variable

N = total number of coordinate points

X = X array

Y = Y array

The Delay Routine

XDEL

In chemical processes, the presence of a transport lag is very common. The approximation of these processes can be done by using simple transfer functions with dead time. XDEL is implemented to simulate the delay of 25 different variables for a given time period in the simulation of a process.

The implementation of such routines is usually very simple for systems using constant step-size integration routines. The values are stored in an array and the number of the values to be stored is equal to the delay time divided by the step-size. (This number has to be an integer) If a changing step-size integration routine is used, the number of the stored values will change every time the step-size is changed. Other problems, i.e., the last values are discarded if the step-size is halved, the step-size is adjusted for exact printing time or plotting time, make the use of these simple routines impossible. The objective was to solve these problems and write a subroutine which would be completely compatible with the integration routine DSIM. The subroutine DSIM is implemented for this purpose, however it has a highly complex logic. A diagram showing the flow chart is given in Fig. 4.

The operating principle of this routine is to allocate N spaces for the values of the variable to be delayed and N spaces for the time the value was stored. Every time the routine is called readin and readout move from one space to the next one. The stored time value is equal to actual time added to delay time (this is actually the time when the variable should be readout). For the readout of a value, the present time is compared to the stored time values until the same or the closest is found. The delayed variable value corresponding to this time or the value found by interpolation between the two closest times is readout as X(NXO). In case, the step-size is halved, the values that are discarded in the routine DSIM, are also discarded in XDEL.

XDEL (NXI, NXO, DELAY, T, JC)

NXI	=	input variable number
NXO	=	output variable number
DELAY	=	delay time
T	=	actual time
JC	=	subroutine call number

NOTE: XDEL should be considered as an equipment having an input and an output. Two different numbers should be assigned as input and output variables.

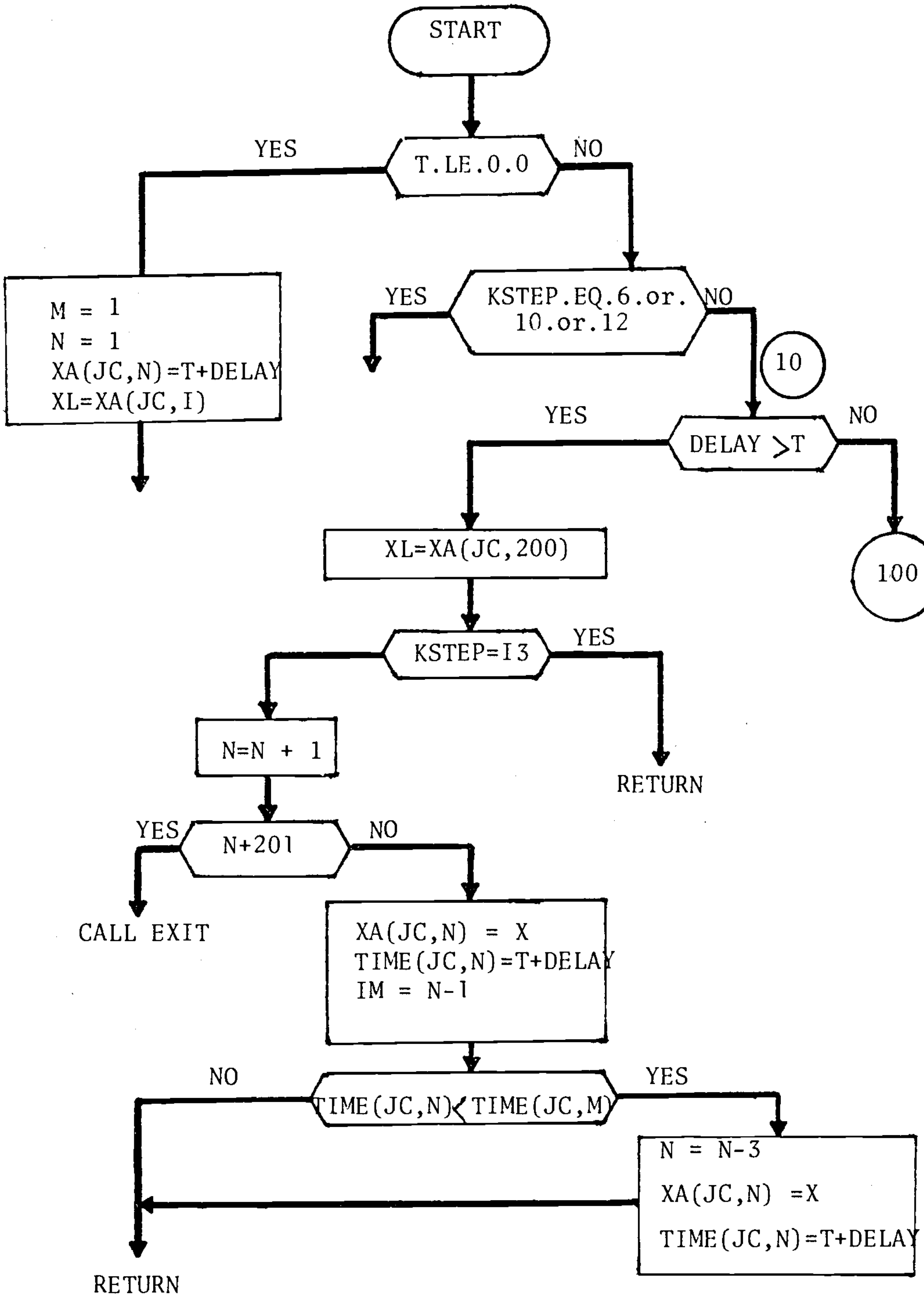
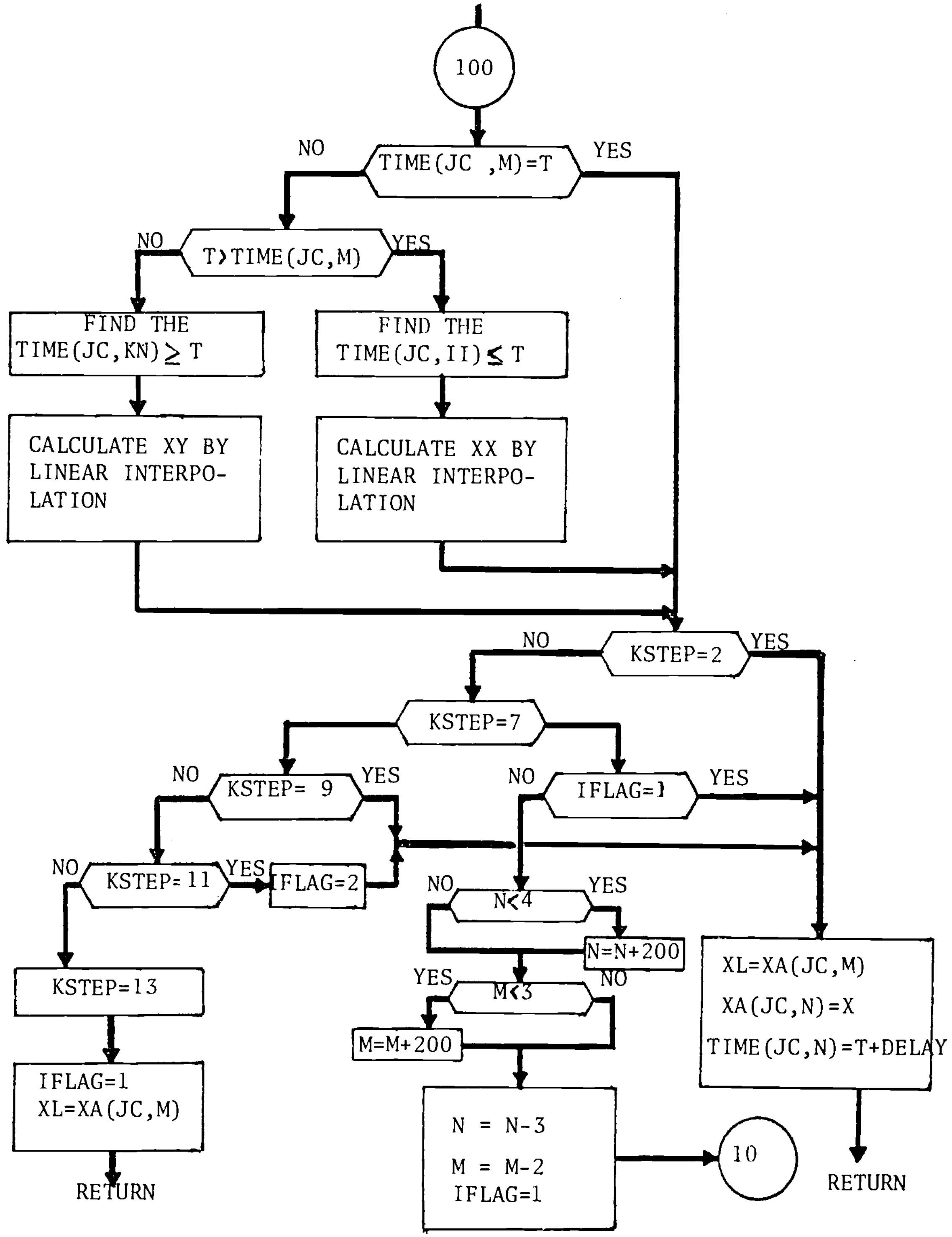


Figure 4. Flow chart of the delay routine, XDEL


100
NO
TIME(JC ,M)=T
YES
NO
T>TIME(JC,M)
YES
FIND THE TIME(JC,KN) ≥ T
FIND THE TIME(JC,II) ≤ T
CALCULATE XY BY LINEAR INTERPOLATION
CALCULATE XX BY LINEAR INTERPOLATION
NO
KSTEP=2
YES
KSTEP=7
NO
KSTEP= 9
YES
NO
IFLAG=1
YES
NO
KSTEP=11
YES
IFLAG=2
NO
N<4
YES
N=N+200
YES
M<3
NO
M=M+200
KSTEP=13
IFLAG=1
XL=XA(JC,M)
RETURN
N = N-3
M = M-2
IFLAG=1
10
XL=XA(JC,M)
XA(JC,N)=X
TIME(JC,N)=T+DELAY
RETURN

III. ANALOG AND DIGITAL SIMULATION OF A NONLINEAR SYSTEM

The system used for the simulation was a gas absorber. The analysis and the dynamic behavior are thoroughly discussed by Coughanowr and Koppel (7). They also developed an analog computer circuit to simulate the gas absorber. The description of the gas absorber, the assumptions made to simplify the modeling, and the equations describing the absorber are given in the Appendix C-1.

The Problem Description

An air- SO_2 mixture containing 2 mole percent SO_2 enters the column at a flow rate of V = 0.051 lb moles/min. of gas mixture. Pure water enters at the top with a rate of X_3 = 0.4 lb. moles/min. The equilibrium relation at 25^oC and 1 atm is:

$$y = 27X - 0.00324$$

The holdup, same for each plate, is H = 0.1666. The liquid dynamics time constant for each plate is, TC = 0.1 min.

For the gas absorber described above, the dynamic response of the column was found for the following step changes:

Run a. A step change from 0.02 to 0.0226 mole fr. SO_2 was made in Y_0, all other conditions remaining the same.

Run b. A step change from 0.4 to 0.2 lb. moles/min. was made in X_3 all other conditions remaining the same.

Analog Computer Simulation

A computer diagram for simulating the problem is shown in Fig. 5. The settings of the coefficients are shown in Table 2. The time scale factor β, has been set to 60 in order to slow down the response of the analog computer.

The procedure for getting the circuit into operation is given by Coughanowr and Koppel.

Fig. (6), and (7) show the response of X_1 and X_5 for changes in inlet gas composition and in liquid flow rate as described by Run a and b. As expected, the response of X_5 to a change in concentrating was an overdamped second-order response and the response of X_1 appeared as first-order. The nonlinearity of the system is seen in the response of X_1 to the liquid flow rate change, where X_1 dropped slightly before rising to its new steady-state value.

Digital Computer Simulation

The simulation was done by solving the differential equations using subroutine DSIM. The values of the variables were not changed in order to be able to compare the responses obtained by analog simulation with the responses of the simulation.

The main program and the results are given in Appendix C-2. Fig. (6) and (7) show the plotted results.

The perfect matching of the responses proved the

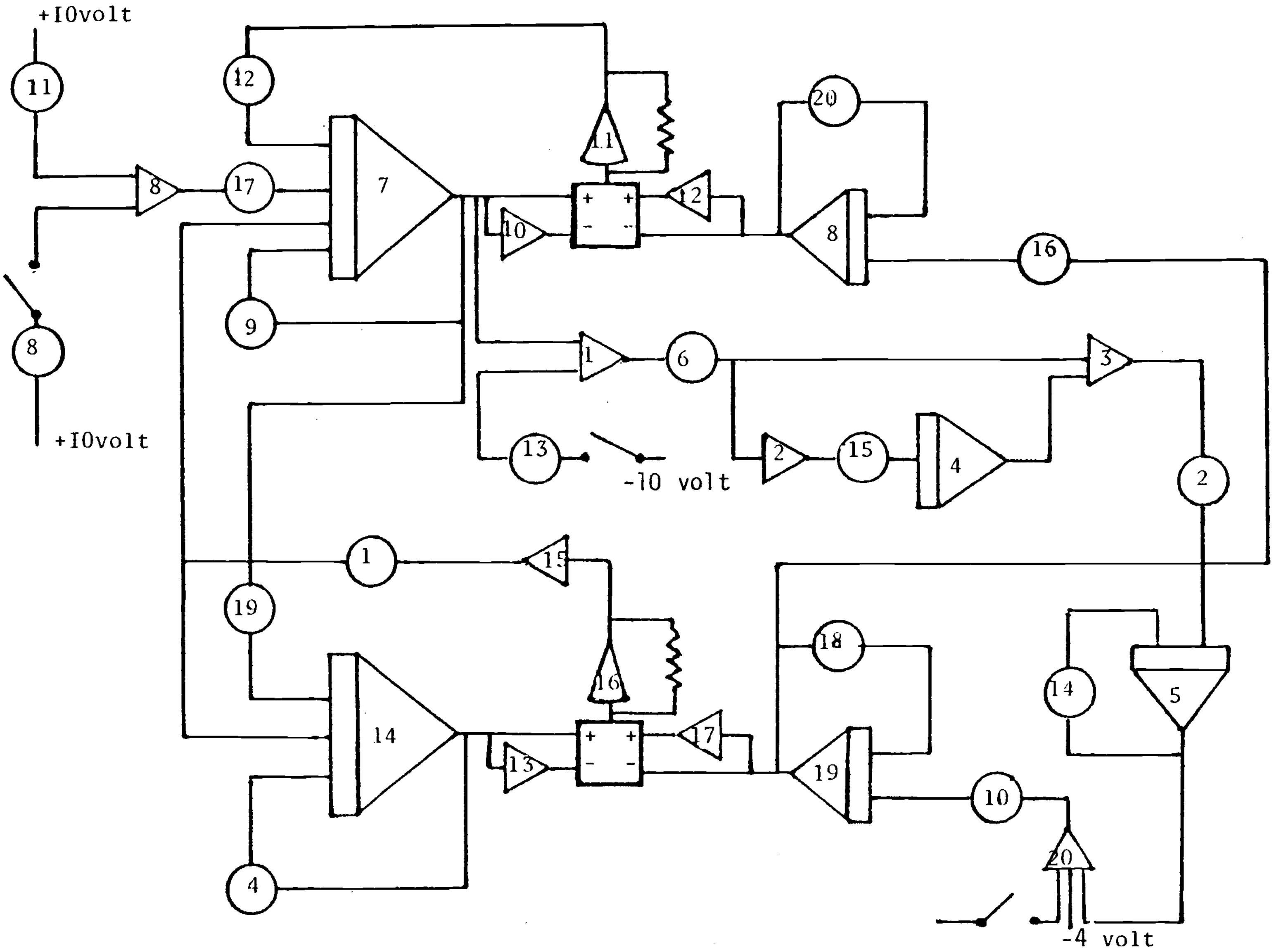


Figure 5. Computer circuit for the analog simulation of the gas absorber

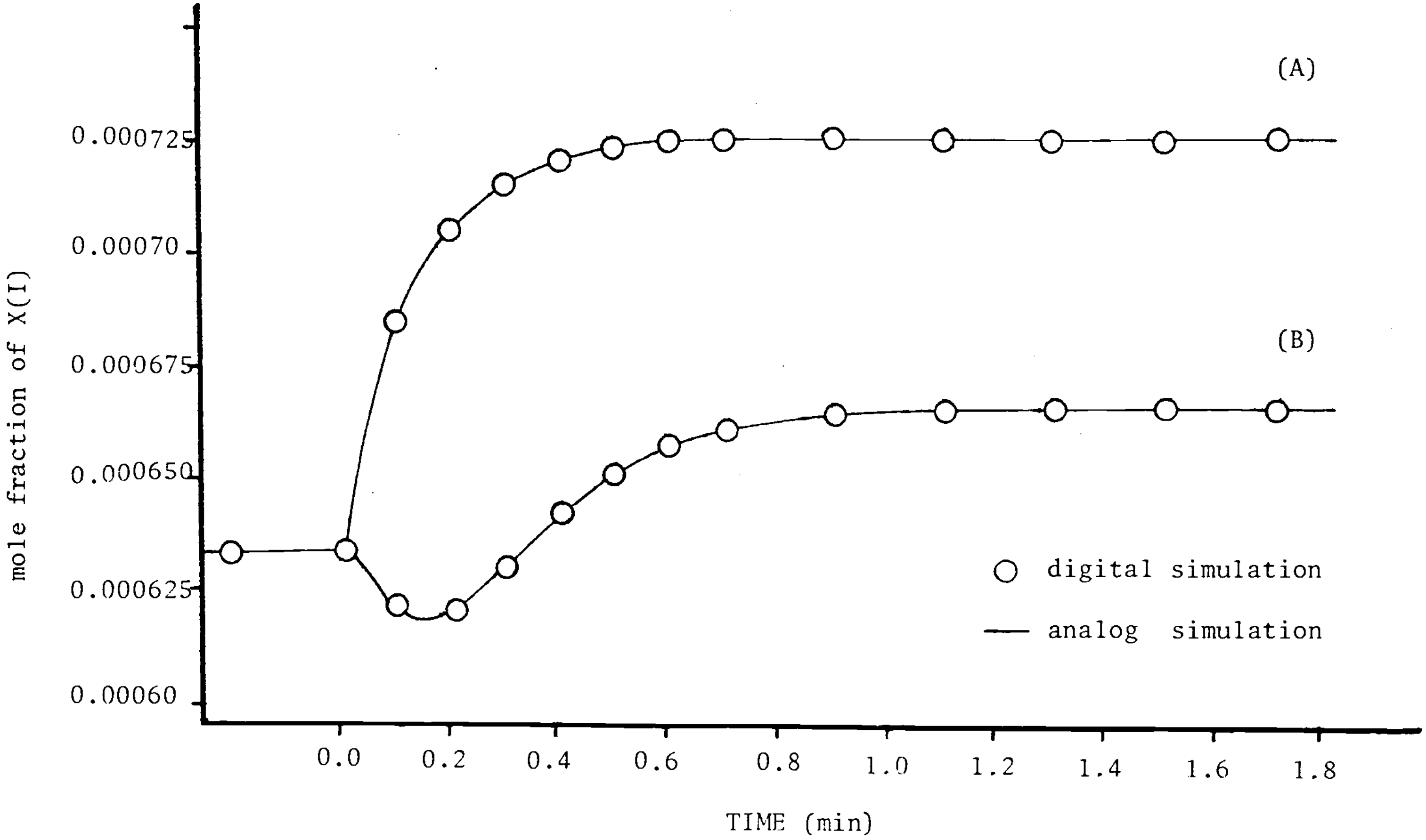


Figure 6. Response of X(1) to step changes in inlet gas concentration and flow rate (A) inlet gas concentration step change , (B) flow rate step change

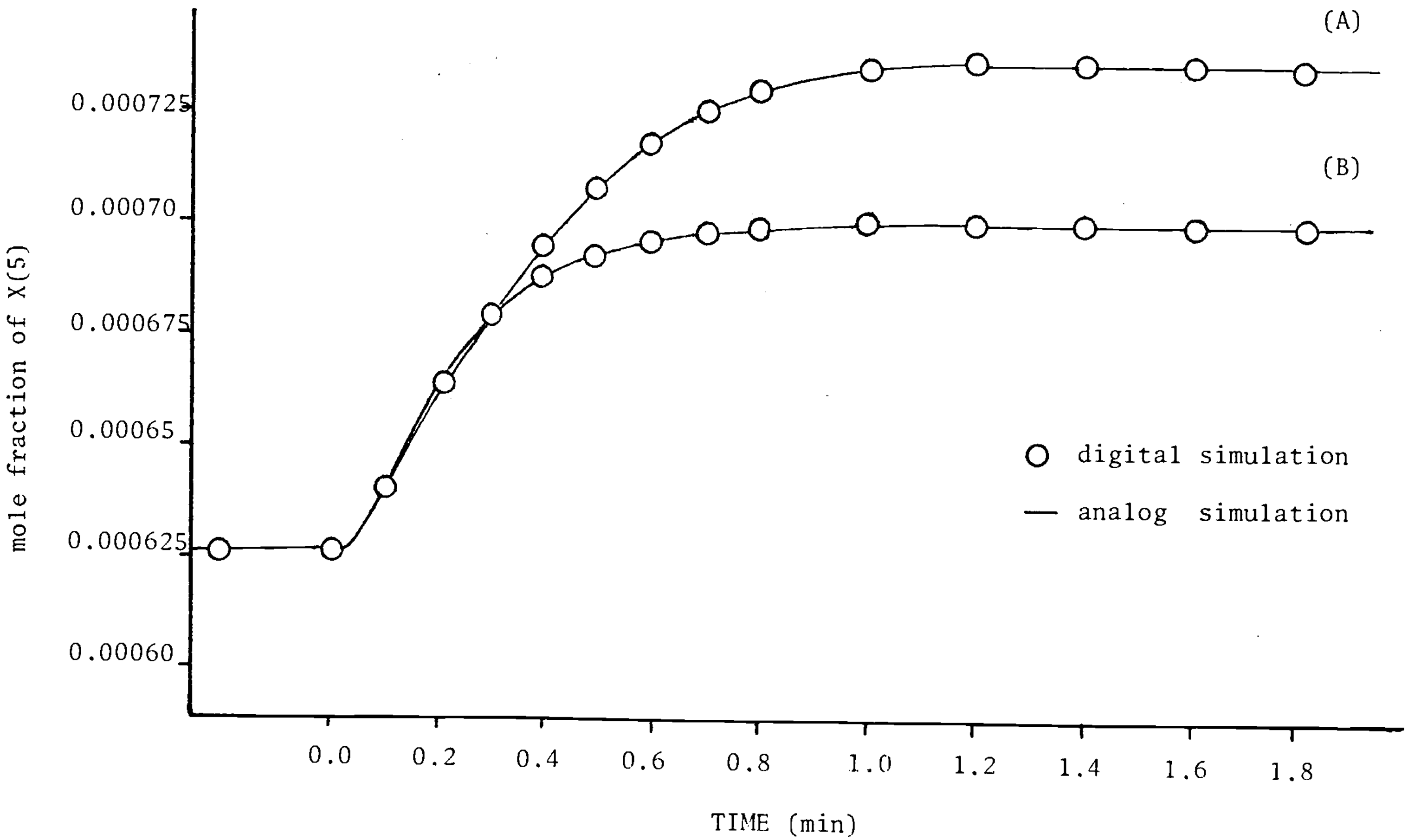


Figure 7. Response of X(5) to step changes in inlet gas concentration and flow rate
(A) flow rate step change , (B) inlet gas concentration step change

capability of the executive system in simulating a system described by its dynamic equations. The next step was a successful simulation of the control of a process. This is shown in the next chapter.

IV. SIMULATION OF THE CONTROL OF A GAS ABSORBER

The purpose of this part of the research was to simulate the control of the gas absorber and to study the effect of the gain and time constant of the controller on the analog computer, and then simulate the same controller with the digital computer.

A proportional-integral controller was chosen because of the limitations of the analog computer used for this simulation and its relevant application to industry.

Along with the controller, a value having first-order dynamics with time constant T_v and gain K_v was used to complete the simulation. The values used for different runs are shown in Table 2.

Analog Computer Simulation

The computer circuit for the simulation is shown in Fig. 5. The values of the potentiometers are in Table 3.

The response obtained with the controller having K = gain = 0.5 and time constant T_I = 10 sec. was chosen as a reasonably good control action and higher and lower values of K and T_I were used to show the effect of the gain and the time constant. Plots showing these responses are in Fig. 8-12. Only the response of the variable X_1 was plotted for the reason that X_1 was the controlled variable.

Table 2: Controller and Valve Arguments for Different Runs

Run	K_c	T_I	K_v	T_v
1	2500	0.1666	0.705	0.0833
2	2500	0.126	0.705	0.0833
3	2500	0.233	0.075	0.0833
4	3250	0.1666	0.705	0.0833
5	1750	0.1666	0.705	0.0833

Table 3: Values of the Potentiometers Used in the Simulation

Potentiometer	Value	Potentiometer	Value
1	0.1	12	0.1
2	0.141	13	0.4
4	0.1375	14	0.2
6	0.5	15	0.1666
8	0.05	16	0.1666
9	0.1375	17	0.1375
10	0.1666	18	0.1
11	0.43	19	0.1375
		20	0.1666

Digital Computer Simulation

The routines PICONTR and TRFN were used for the simulation of the controller, and the valve. These routines were called in the derivative section of the main program. The values of the gain and the time constant of the controller were kept the same as in Table 2. The results for these simulations were obtained and plotted. Fig. 8-12 show the responses of the variable X_1. The identical responses obtained for both cases show that the executive system can be effectively used as a tool to improve the control of a process.

A listing of the main program and the results are given in Appendix D.

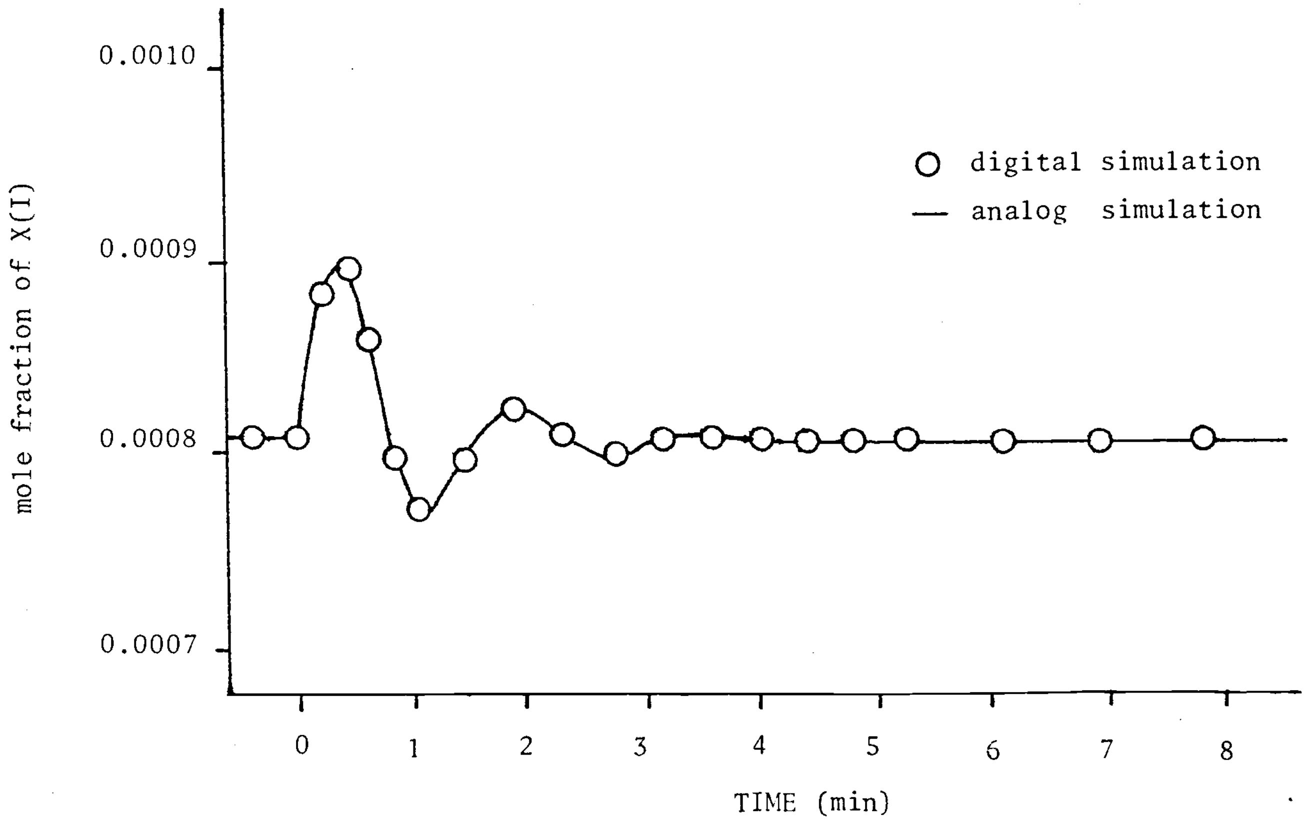


Figure 8. Transient response of the gas absorber (K_c = 2500 , T_I = 0.1666)

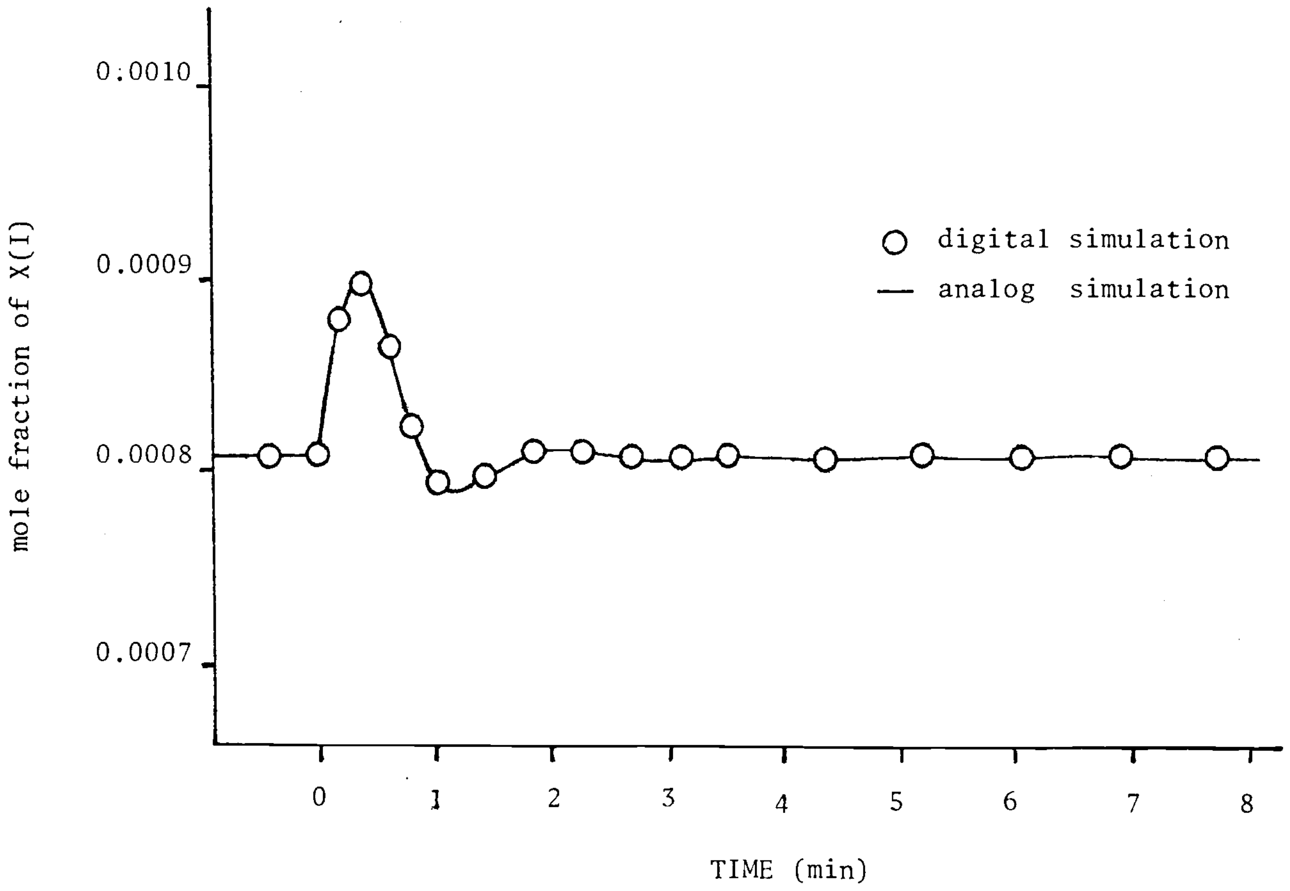


Figure 9. Transient response of the gas absorber (K_c =2500 , T_I =0.233)

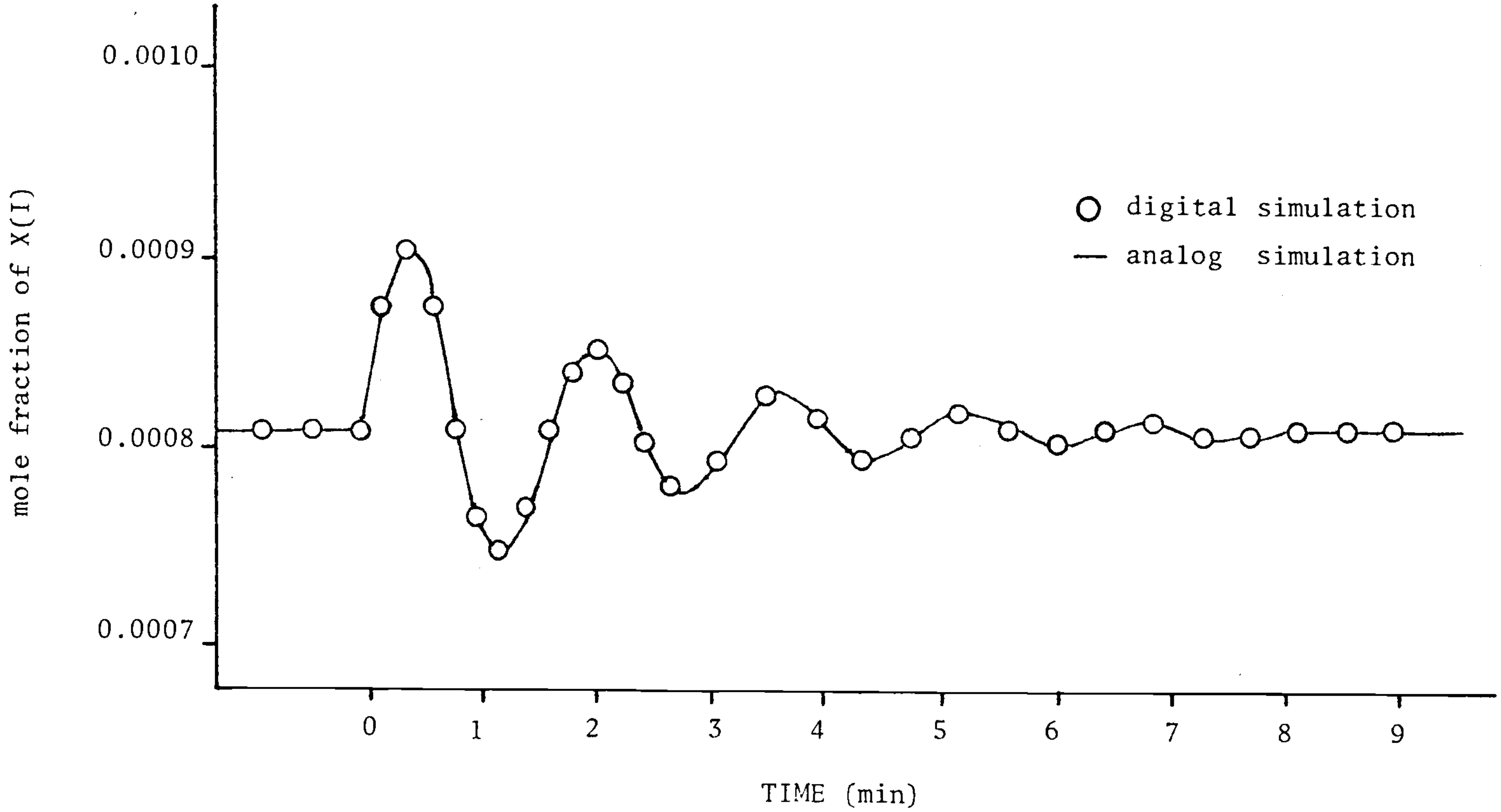


Figure 10. Transient response of the gas absorber (K_c = 2500 , T_I = 0.I26)

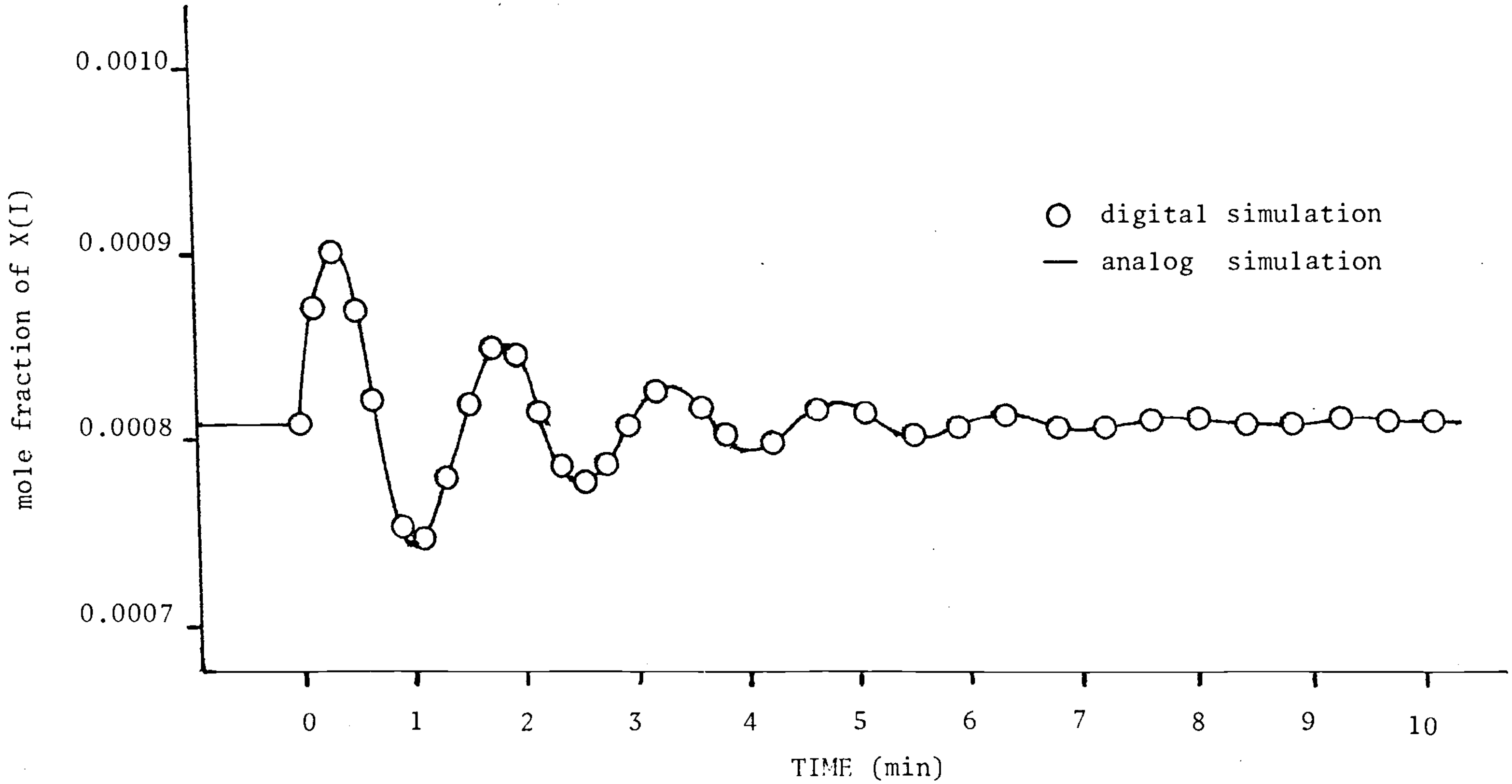


Figure 11. Transient response of the gas absorber (K_c = 3250 , T_I =0.1666)

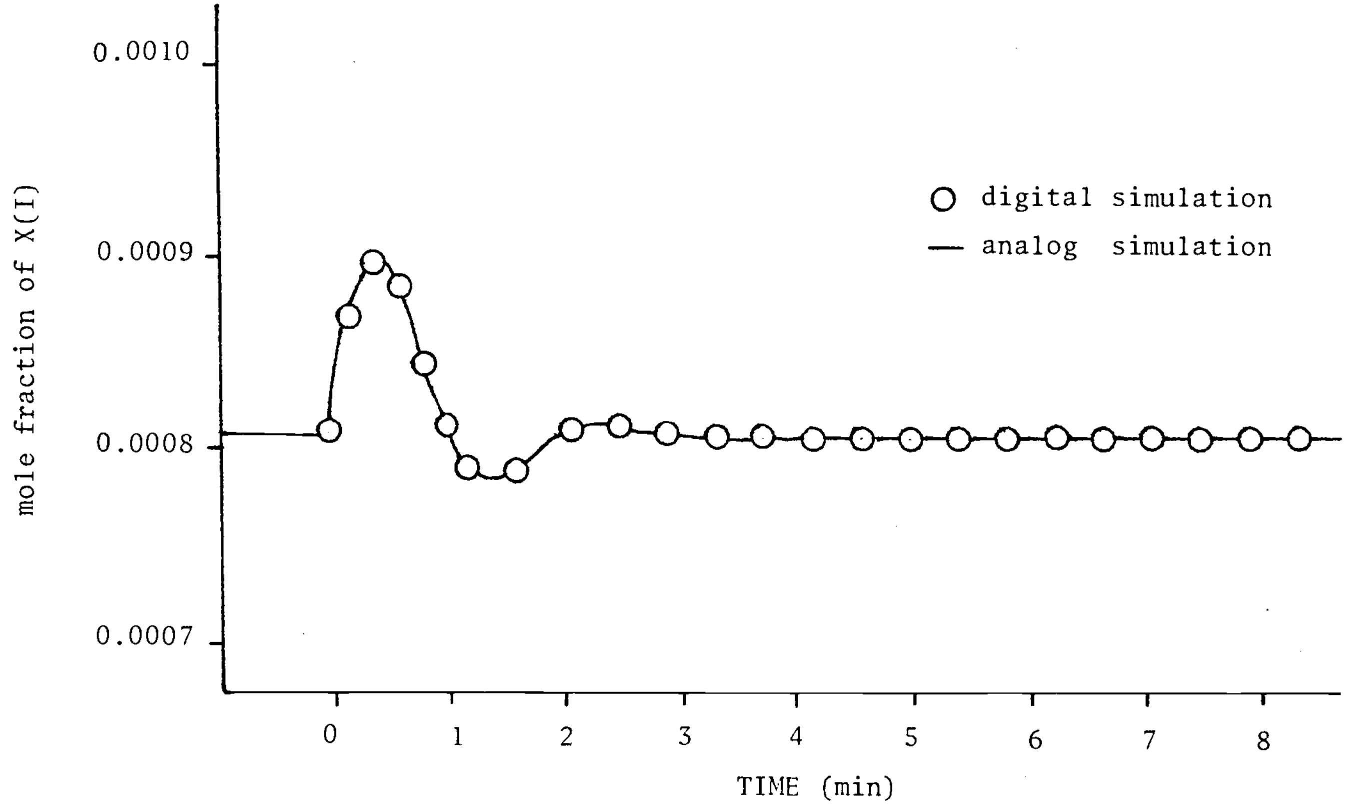


Figure 12. Transient response of the gas absorber (K_c =I750 , T_I = 0.I666)

V. DIRECT DIGITAL COMPUTER CONTROL OF A GAS ABSORBER

There are two approaches for digitally representing the conventional or electronic controllers. The first one is the position algorithm where the computer output is the corrected valve position, and the integration is done in the computer. The second one is the velocity algorithm, where the computer output is the change that the valve should have between sampling times and the integration is done by the final element. In this study, the position algorithm is used because of the limitations on the use of the velocity algorithm. Although, equations describing PID control are given in Appendix E-1 only PI control action was used in order to compare the responses obtained with the responses already obtained from analog computer.

NOVA Computer Control of the Simulated Gas Absorber with Fast Sampling Time

The input and output operations used between NOVA and the analog computer were done by using the Real-Time Subroutine Package developed by Stan Fukui (9).

The control of the system was done by sending a voltage signal representing the concentration X_1 to NOVA from the analog computer, and sending back a voltage signal representing the value of the valve setting calculated by NOVA back to the analog computer. The valve was simulated in the analog computer as before, using the same K_V and T_V values. A listing of the computer program used in the control is given in Appendix E-2.

A sampling time of 1/100 of a second was found satisfactory for controlling the system as an analog controller.

Fig. 8-12 show the transient response of the DDC controlled absorber to a step change in inlet concentration. These runs were made by using the tabulated values of K_C and T_I listed in Table 2, for the same step input change of the concentration. The identity of the responses show that the control action of an analog controller is obtained by DDC control for fast sampling time.

Effect of Sampling Time on DDC Control

A very large sample time, such as 10 seconds was used in the control of the same process. The responses were obtained for different gains and time constants of the controller. The high oscillatory behavior of the control response is shown in Fig. 13-15. The gain and time constant values used in each case are listed in Table 4.

Simulation of the Effect of Sampling Time with Digital Computer

The simulation of the effect of sampling time was done by using DPICON routine instead of PICONTR routine. The call was done in the plotting section and HPLOT was set to 0.1666. The results were plotted and a perfect matching was obtained as it is seen in Figures 13, 14, and 15. A complete listing of the main program is given in Appendix E-2. It is seen that the executive system can be satisfactorily used in finding the performance of digital controller for different sampling times.

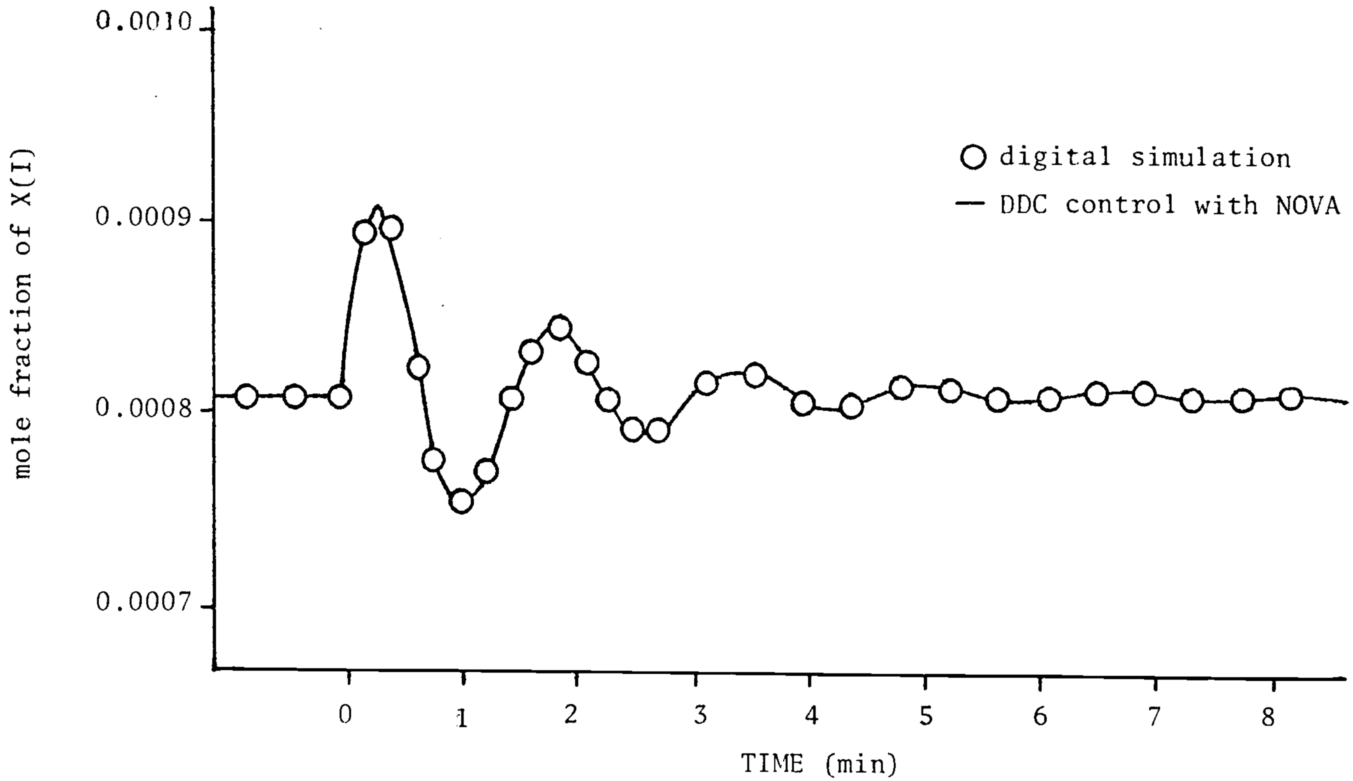


Figure 13. DDC Control with sampling time = 10 sec (K_c = 2500 , T_I = 0.1666)

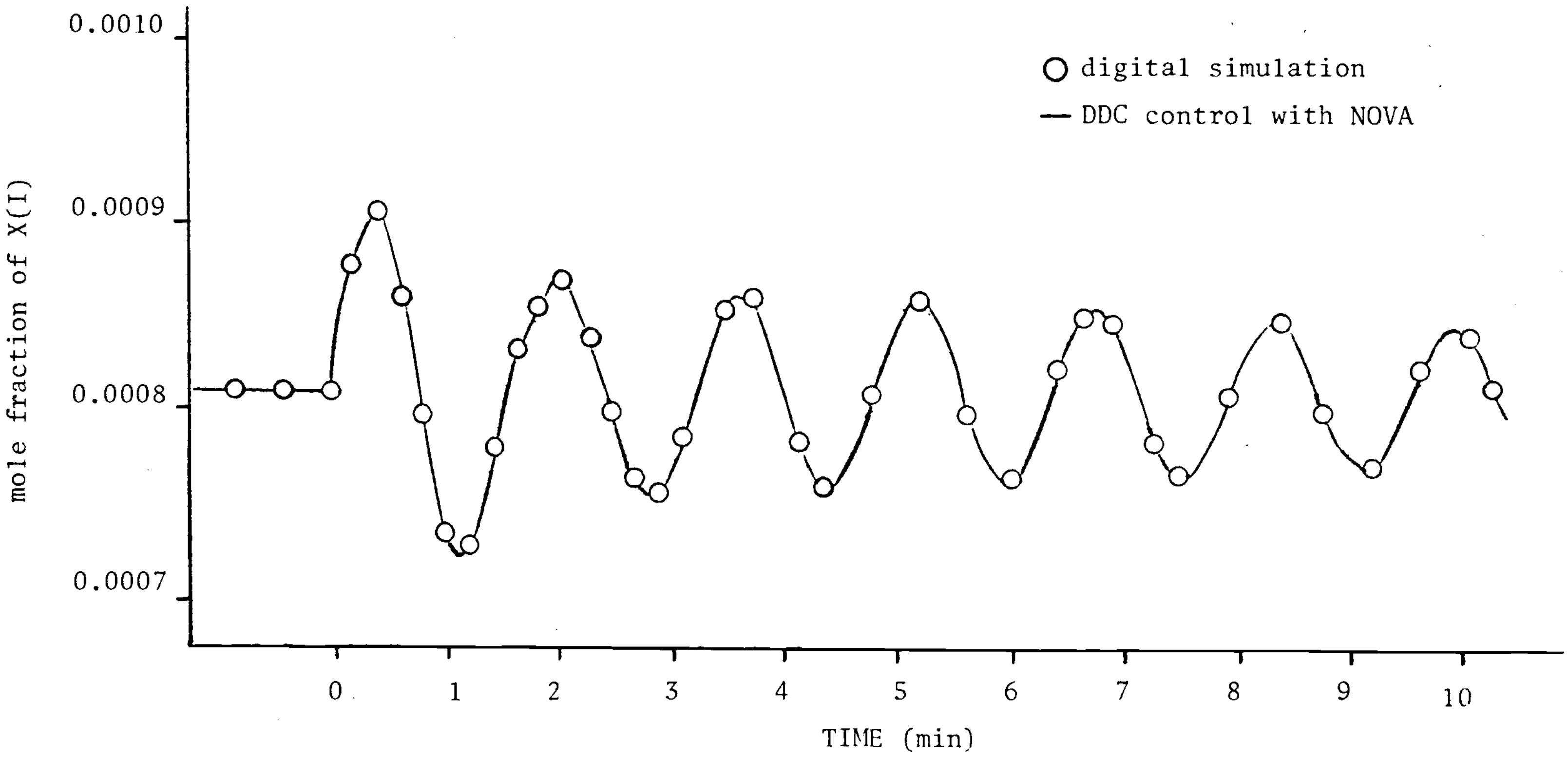


Figure 14. DDC Control with sampling time = 10 sec (K_c = 2500 , T_I = 0.I26)

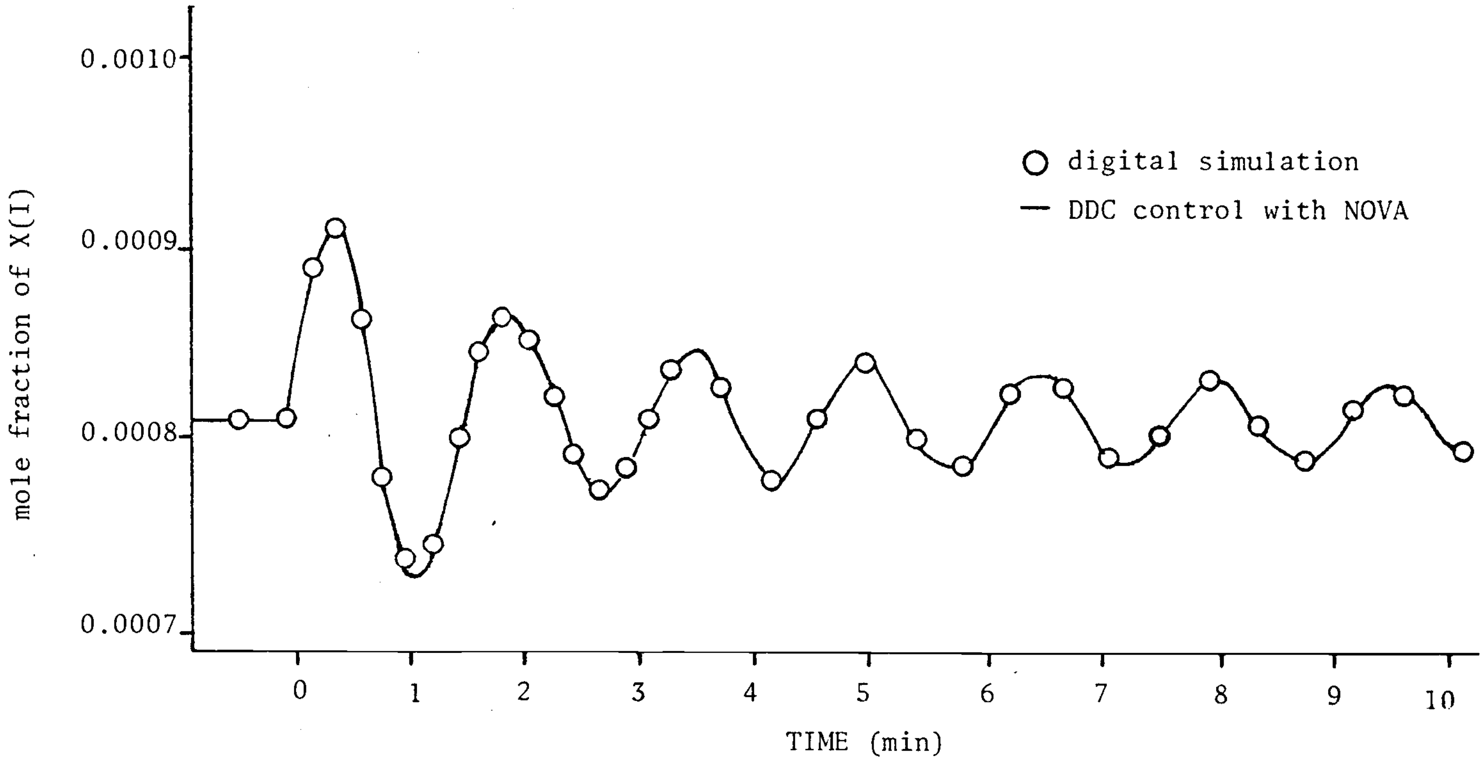


Figure 15. DDC Control with sampling time = 10 sec (K_c = 3250 , T_I = 0.1666)

Table 4: K_c and T_I Values for DDC Control

Run	K_c	T_I
1	2500	0.1666
2	3250	0.1666
3	2500	0.126

VI. FEEDFORWARD CONTROL OF A GAS ABSORBER AND A FIFTH ORDER SYSTEM

Feedback control is based on the measurement of the controlled variable, its comparison with the desired value, and the use of the difference as a mean to compute an input to the process, in order to eliminate this difference. The feedforward control is based on the measurement of a variable which is subject to upsets, and the compensation of any deviation in its value by manipulating another input before the upset affects the controlled variable.

In practice, the feedforward control systems make energy and material balances to compute the necessary changes. The computer should be programmed in order to maintain the balances in the steady-state and also in transient intervals between steady-states. It must have a model of the process, consisting of a steady-state and dynamic parts. In this study both steady-state feedforward control and feedforward control with dynamic compensation were applied.

Steady-State Feedforward Control

In this partial feedforward control, the appropriate change in the manipulative variable is made only by considering the energy and material balances of the system. For nonlinear systems, the steady-state changes can be easily found numerically by using the nonlinear steady-state equations of the process. The differential equations representing the process are set equal to zero and the resulting equations are solved for the manipulative variables as a function of the disturbed variables.

In this study a steady-state feedforward control of the gas absorber was done both by using NOVA computer as a controller of the system simulated on analog computer, and by simulating the gas absorber and the controller on the digital computer.

Control With NOVA

The computer was programmed to control the value of concentration X_1 for changes in inlet gas concentrations by manipulating the flow rate of the liquid input. The derivation of the equation representing the relation between the flow rate and the inlet gas concentration is shown below.

The derivative equations of the gas absorber are set equal to zero,

$$\frac{dX1}{dt} = 0 = \frac{1}{H}(L_2X_2 - L_1 X_1) + \frac{Vm}{H}(X_o - X_1)$$

$$\frac{dX2}{dt} = 0 = \frac{Vm}{H}(X_1 - X_2) - \frac{1}{H}(L_2X_2)$$

$$\frac{dL1}{dt} = 0 = L_3/T_2 - L_2/T_2$$

$$\frac{dL2}{dt} = 0 = L_2/T_2 - L_1/T_2$$

Where $L_1 = X_3$ and $L_2 = X_4$

The solution of these equations by substitutions gives

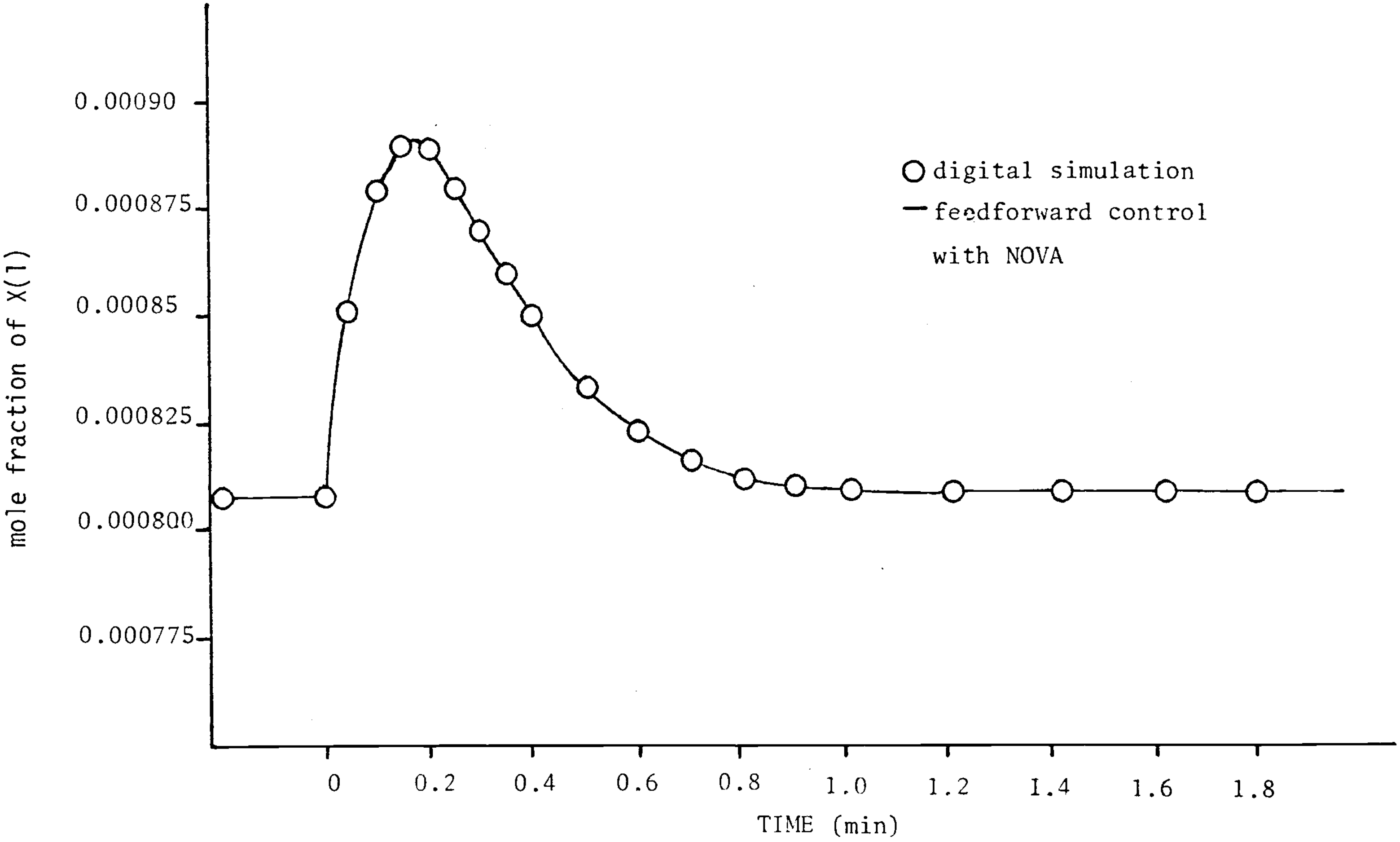


Figure 16. Transient response of the gas absorber with steady-state feedforward control

$$L^2 X_1 - L (V_m X_o - V_m X_1) - V_m^2 (X_o - X_1) = 0$$

Using the quadratic solution technique,

$$L = \frac{V_m (X_o - X_1) + \sqrt{V_m^2 (X_o - X_1)(X_o - X_1 + 4X_1)}}{2X_1}$$

L is the flow rate that should be used to compensate the inlet gas concentration change.

Fig. 16 shows the transient response of the absorber to the step input change on the gas concentration. The fast return of the concentration to the set point value shows the successful use of feedforward control if the response of the system to a change is known.

Simulation of the Control

The equation describing the relation of the liquid flow rate with the gas inlet concentration was added to the derivative section of the simulation program. The listing of this program and the simulation results are given in Appendix F-2.

The results were plotted and found identical to the ones obtained by NOVA.

Feedforward Control With Dynamic Compensation

Different approaches have been taken to formulate the dynamic compensation, easy to apply to real processes (3), (8), (10). The practical model suggested by Shinskey was implemented in this study.

The ratio D_2/D_1 described in Appendix F-1 representing the dynamic elements of the system can be approximated by a lead-lag unit, if the dead-times for D_1 and D_2 are close enough to provide nearly complete cancellation. The output m(t) of this unit follows a step input m as,

$$m\ (t) = m(1+ \frac{\tau_1 - \tau_2}{2} e^{-t/\tau 2})) \quad \text{where}$$

τ_1 is the lead time and τ_2 the lag time. This can be digitally realized simply by iterative procedure. The differential equations for lead and lag units will be:

$$n = y + \tau_2 \frac{dy}{dt} \qquad z = y + \tau_1 \frac{dy}{dt} \qquad \text{or}$$

$$z = x + (\tau_1 - \tau_2)\ dy/dt \qquad \frac{dy}{dt} = \frac{1}{\tau_2}(x - y)$$

where x is the input, y is the input lagged by τ_2 and z is y led by τ_1. The differentials above must be written as difference equations because digital computers can only compute at every interval Δt. First, the value of z_n is calculated from the values of x_n and y_n for that interval

$$z_n = x_n + \frac{\tau_1 - \tau_2}{2} (x_n - y_n)$$

Then, y_n is incremented before the next calculation of z_{n+1}:

$$y_{n+1} = y_n + \frac{\Delta t}{\tau_2} (x_n - y_n)$$

The NOVA computer was used to compute the corrective action from these equations and to send the voltage signal necessary to correct the step input change to the system simulated on the analog computer. As the next step, the system and the controller were simulated on the digital computer and the results were compared.

The Fifth-Order System

The steady-state feedforward and PI control of a system represented by its dynamic equations being already done, the simulation and control of a different system was developed. The system consisted of five tanks with first-order transfer functions. In the simulation of the process by the executive system, the process was approximated as a first-order transfer function with dead-time followed by another first-order transfer function.

Both analog and digital simulations of the system were obtained and results were shown in Appendix C-3. A block diagram showing the system is in Fig. 17. The time constant values and the dead-time value used in the approximation of the process were obtained from the response of the system to a step change as it is

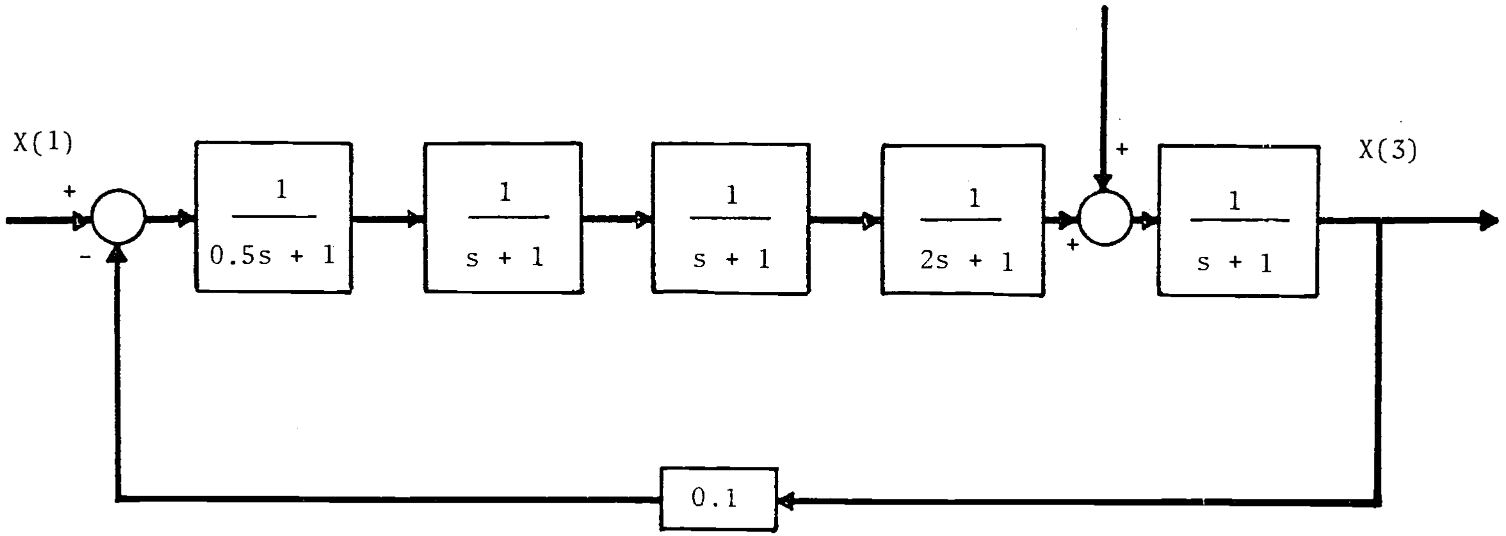


Figure I7. Block diagram of the fifth-order system shown in Fig. 19. Fig. 18 shows the computer circuit used in the analog simulation.

The necessary values used to determine the ratio D_2/D_1 were obtained from the response curves of the system to step change in variables X_1 and X_2 as shown in Fig. 20.

The dynamic correction was obtained as:

$$\frac{D_2(s)}{D_1(s)} = \frac{e^{-\tau_0 s} / (1 + \tau_2 s)}{1/ \ (1 + \tau_2 s)} = \frac{e^{-\tau_0 s} \ (1 + \tau_1 s)}{(1 + \tau_2 s)}$$

where

τ_0 = 0.26 seconds

τ_1 = 0.11 seconds

τ_2 = 0.35 seconds

Dynamic Feedforward Control With NOVA

The sampling time was chosen as 0.01 seconds for this control. The lead-lag function was realized as discussed above, and the action obtained was delayed for the time τ_0. The logic for delaying the values is shown below.

The maximum number of stored values was obtained by

N = (delay time) / (sampling time)

The input values were stored in an array. At each sampling

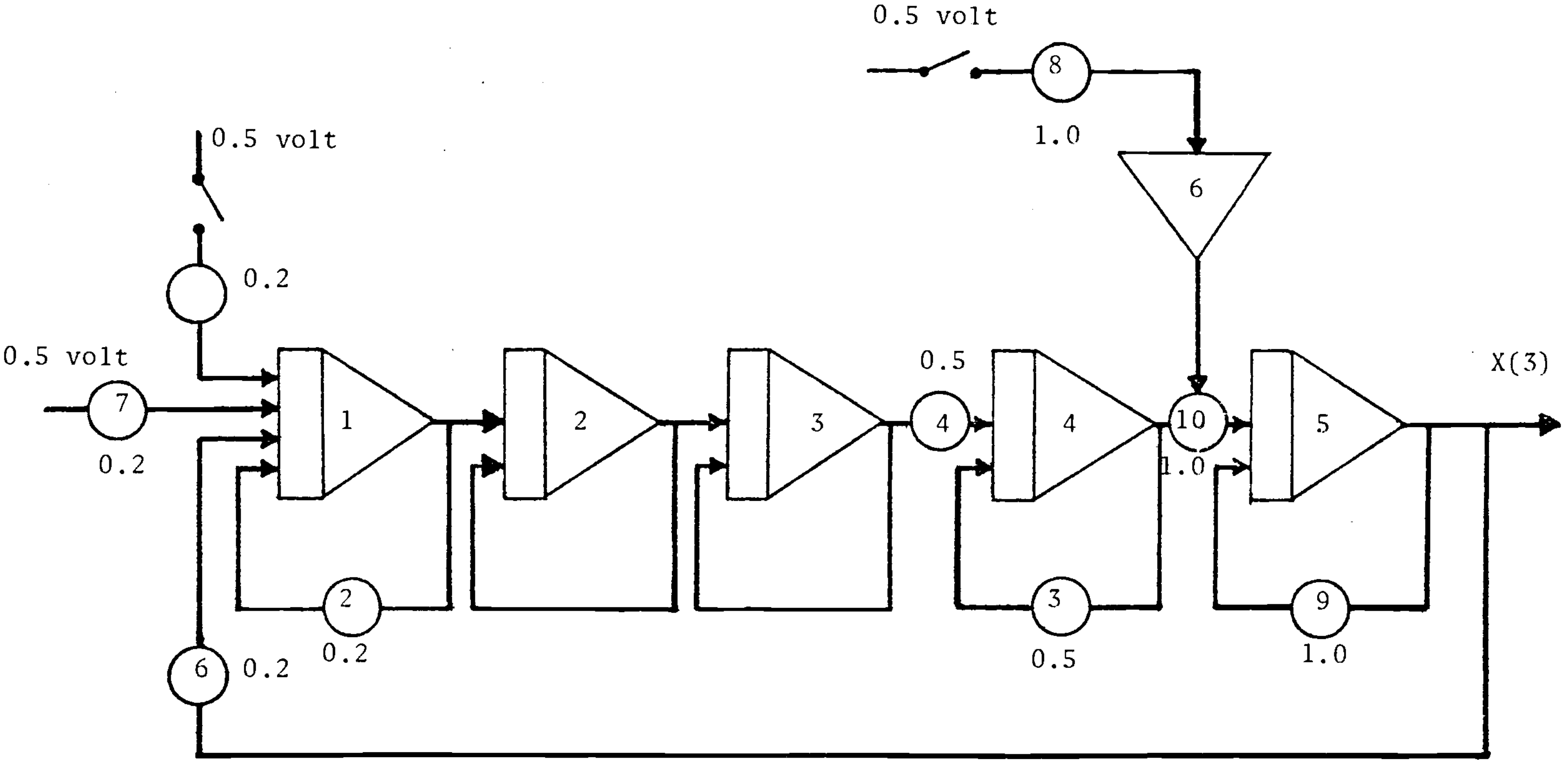


Figure 18. Computer circuit for the analog simulation of the fifth-order system

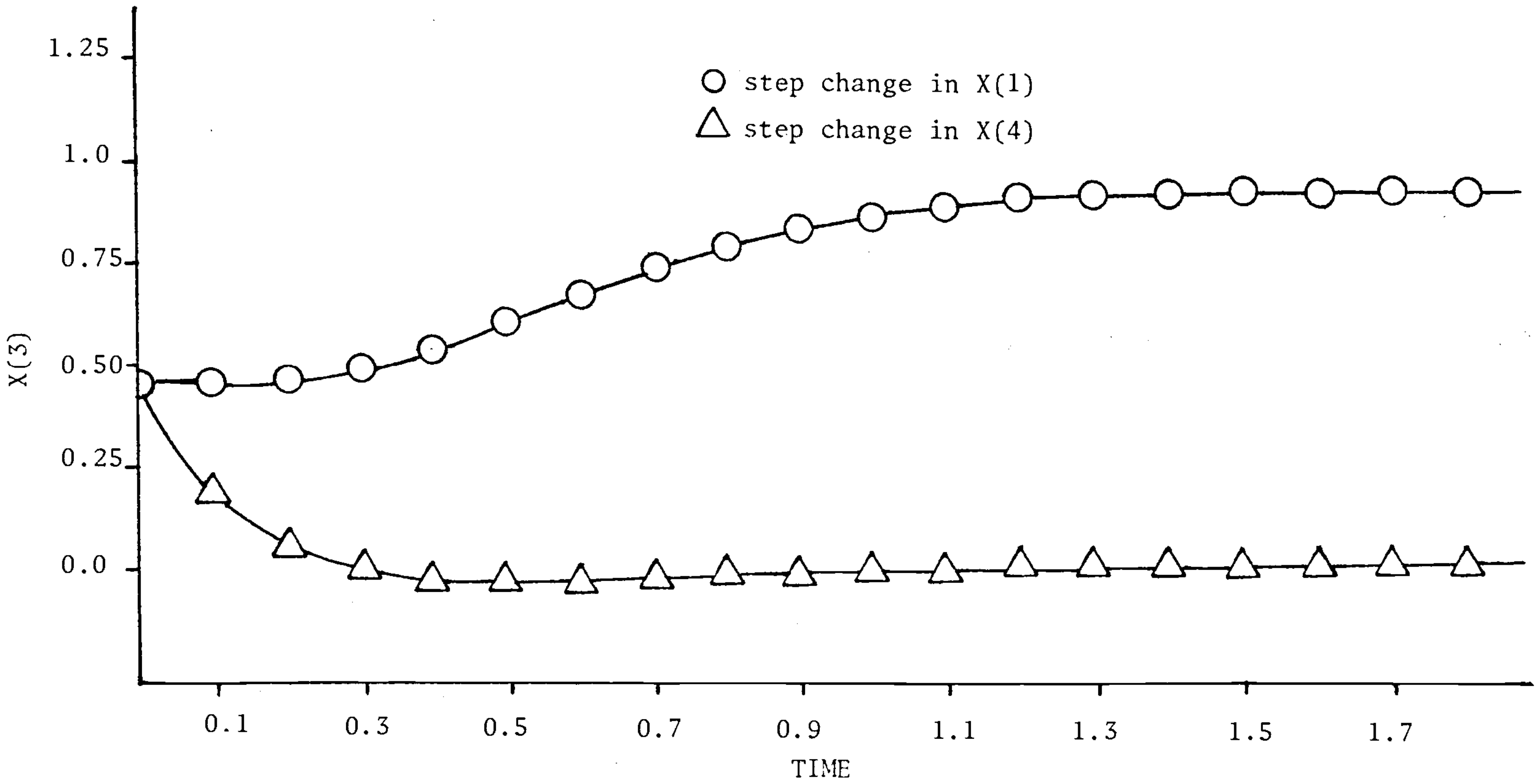


Figure 19. Response of the fifth-order system to step changes

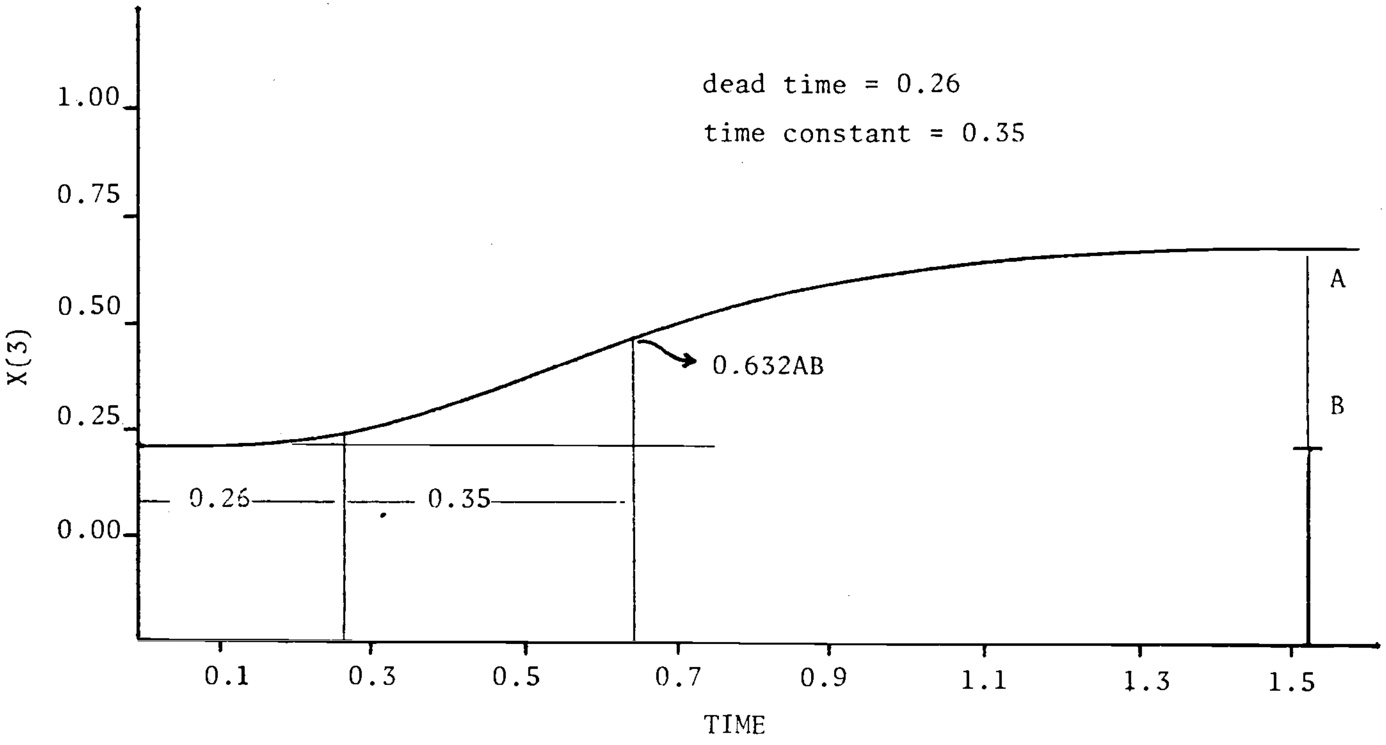


Figure 20. Determination of the parameters for dynamic feedforward control

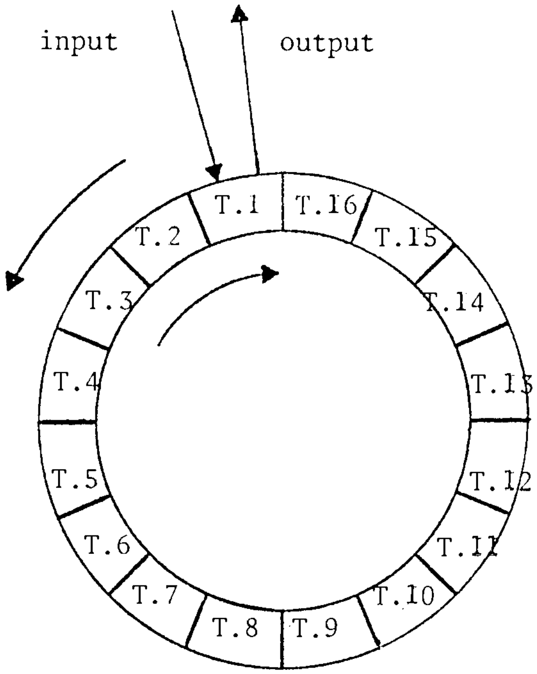
time, the readin and readout was made on this array, represented schematically in Fig. 21. Each time, one value was readout, and its place was replaced by the new value. A computer listing of the computer program is given in Appendix F-3.

Fig. 22 shows the response of the process to a step change of magnitude 0.5.

Simulation of the Control

The simulation of the system was first made using the values obtained from Fig. 19. The results were plotted and the same response was obtained as the one obtained with analog computer. The calculations done for the control action were kept the same, and they were computed every 0.01 seconds by solving them in the plotting section. The dead-time of the process and the dead-time used in the control were successfully simulated by the routine XDEL.

The results of the control action were obtained for a step change of 0.5 in variable X_1 and were plotted. Fig. 22 shows the response of the system. The similarity of the two responses obtained from NOVA and from the simulation, showed the successful use of the executive system.



$$\sum_{i=1}^{n} T_i = \text{delay time}$$

Figure 21. Schematic for simple time-delay logic

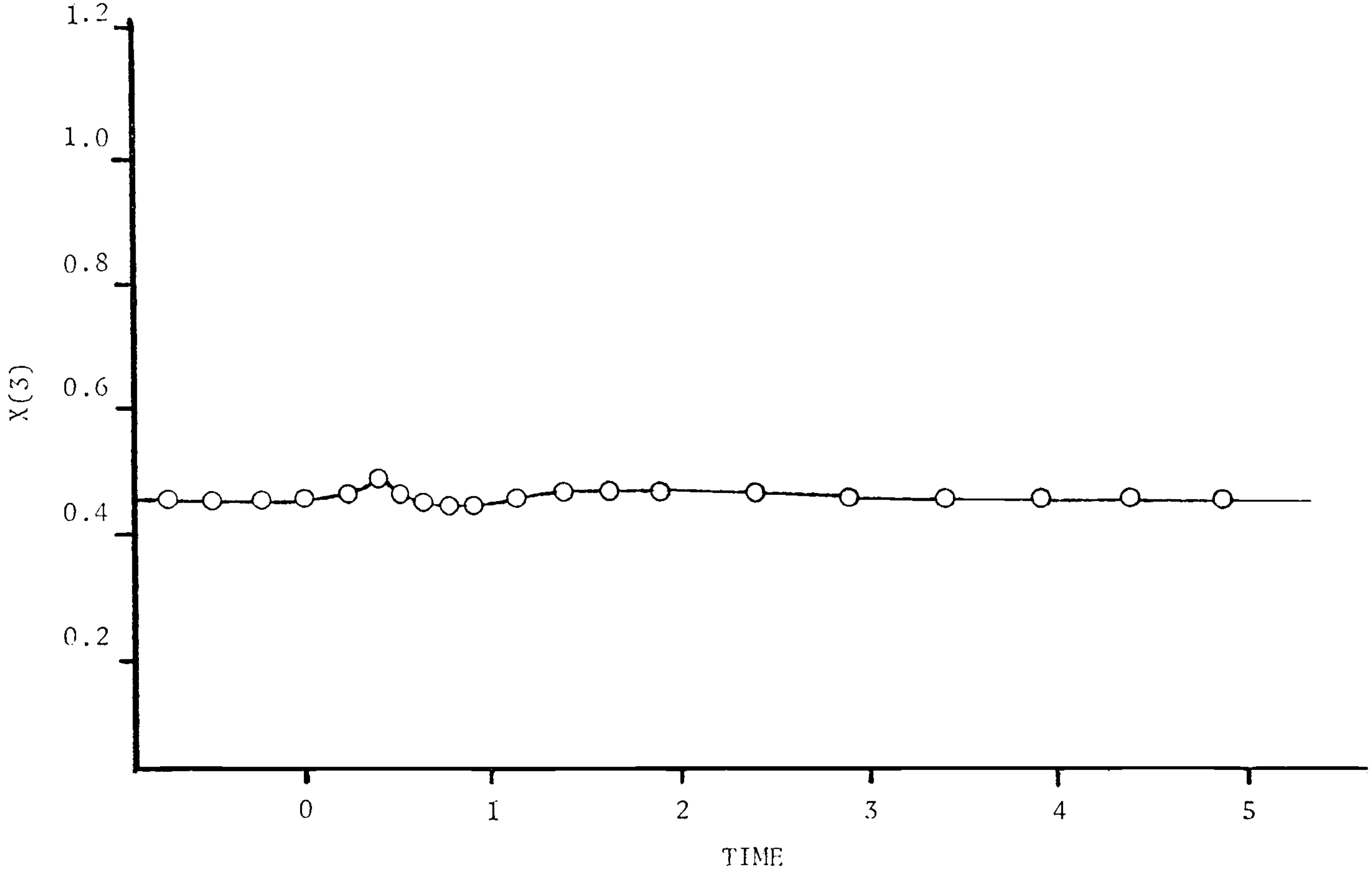


Figure 22. Response of the fifth-order system with dynamic feedforward control to a step change in input

VII. AN ADVANCED CONTROL TECHNIQUE

Design of Control Algorithms Using Z - Transforms

The closed loop system of Fig. 23 can be approximated as a first-order-lag-dead-time which can be written in discrete form as follows:

$$\frac{C(z)}{R(z)} = \frac{(1 - e^{-T/\lambda})\, z^{-N-1}}{(1 - e^{-T/\lambda}\, z^{-1})}$$

where T is the sampling time

λ is the lag time

For a given process described as first-order-lag-plus-dead-time, the controller equation can be written as follows:

$$G(S) = \frac{Ke^{-\theta s}}{\tau s + 1} \qquad \text{(the process)}$$

$$D(z) = \frac{(1 - e^{-T/\lambda})\,(1 - e^{-T/\tau}\, z^{-1})}{K(1 - e^{-T/\lambda}\, z^{-1} - (1 - e^{-T/\lambda})\, z^{-N-1})\,(1 - e^{-T/\tau})}$$

or as

$$D(z) = \frac{\tau}{\lambda}\; \frac{(1 + (\frac{T}{\lambda} - 1)\, z^{-1})}{(1 - (1 - T/\lambda)\, z^{-1} - (T/\lambda)\, z^{-N-1})}$$

A detailed derivation of these equations is given in Appendix G-1.

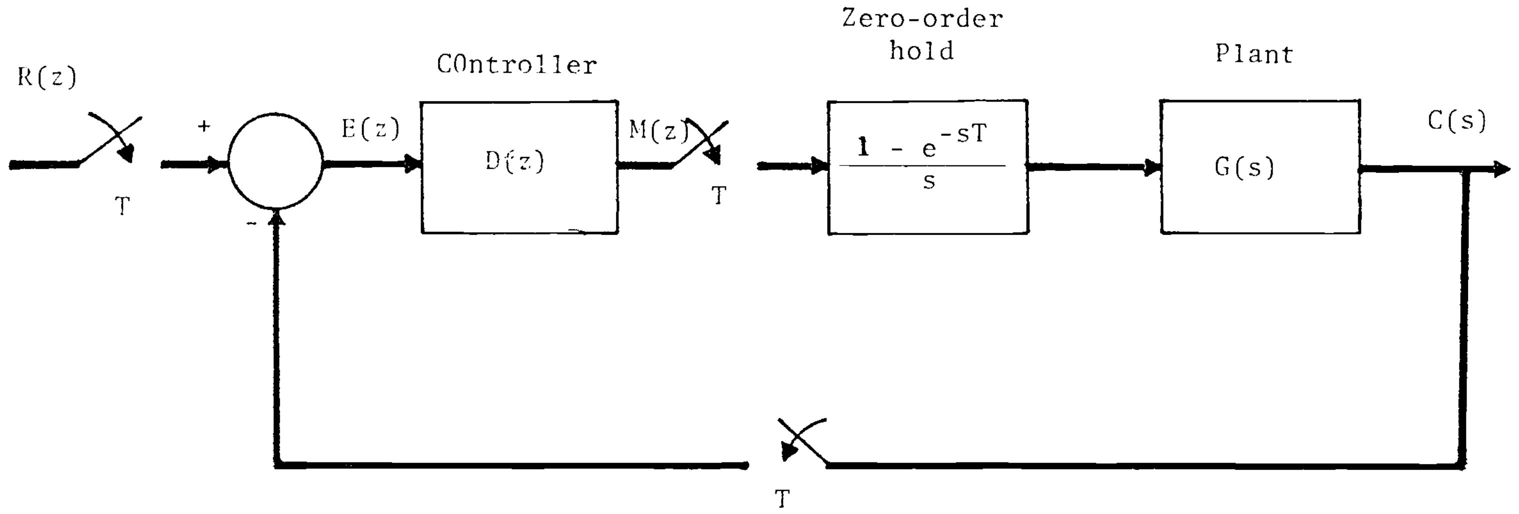


Figure 23. Block diagram for a digitally controlled system

Dahlin's Method With NOVA

The transfer function describing the system is obtained from its transient response to a unit step change and it is:

$$G(z) = \frac{e^{-0.25s}}{0.35\ s + 1}$$

The closed-loop response is formulated as:

$$C(z) = \frac{e^{-0.25s}}{(\lambda s + 1)}$$

Given these equations, the controller D(z) was calculated for sampling time T equal to 0.05 seconds:

$$D(z)_1 = \frac{(1 - e^{-0.05/\lambda})\ (- e^{-0.05/0.35}\ z^{-1})}{(1 - e^{-0.05/\lambda}\ z^{-1} - (1 - e^{-0.05/\lambda})\ z^{-5}\ (1 - e^{-0.05/0.35})}$$

or

$$D(z)_2 = \frac{0.35}{\lambda}\ \frac{1 + (\frac{0.05}{0.35} - 1)\ z^{-1}}{1 - (1 - \frac{0.05}{\lambda}\ z^{-1} - (\frac{0.05}{\lambda})\ z^{-5}}$$

Since $\quad D(z) = \frac{M(z)}{E(z)}$

M at each sampling time was calculated as:

$$M_1 = \frac{(1 - e^{-0.05/\lambda})}{(1 - e^{-0.05/0.35})} (E_n - e^{-0.05/0.35} E_{n-1}) + e^{-0.05/\lambda} (M_{n-1})$$

$$+ (1 - e^{-0.05/\lambda}) M_{n-5}$$

or

$$M_2 = \frac{0.35}{\lambda} (E_n + (0.05/0.35 - 1) (E_{n-1})) + (1 - 0.05/\lambda) M_{n-1}$$

$$+(0.05/\lambda) (M_{n-5})$$

The transient response of the system to a unit step change was obtained using different λ values such as 0.4, 0.24, 0.1. Fig. 24 shows the response of the system.

The values of the error and M were stored at each sampling time because of the use of past values in the computation of M. A listing of the computer program is given in Appendix G-2.

The responses of the system using M_1 and M_2 were identical which showed either of the equation could be used.

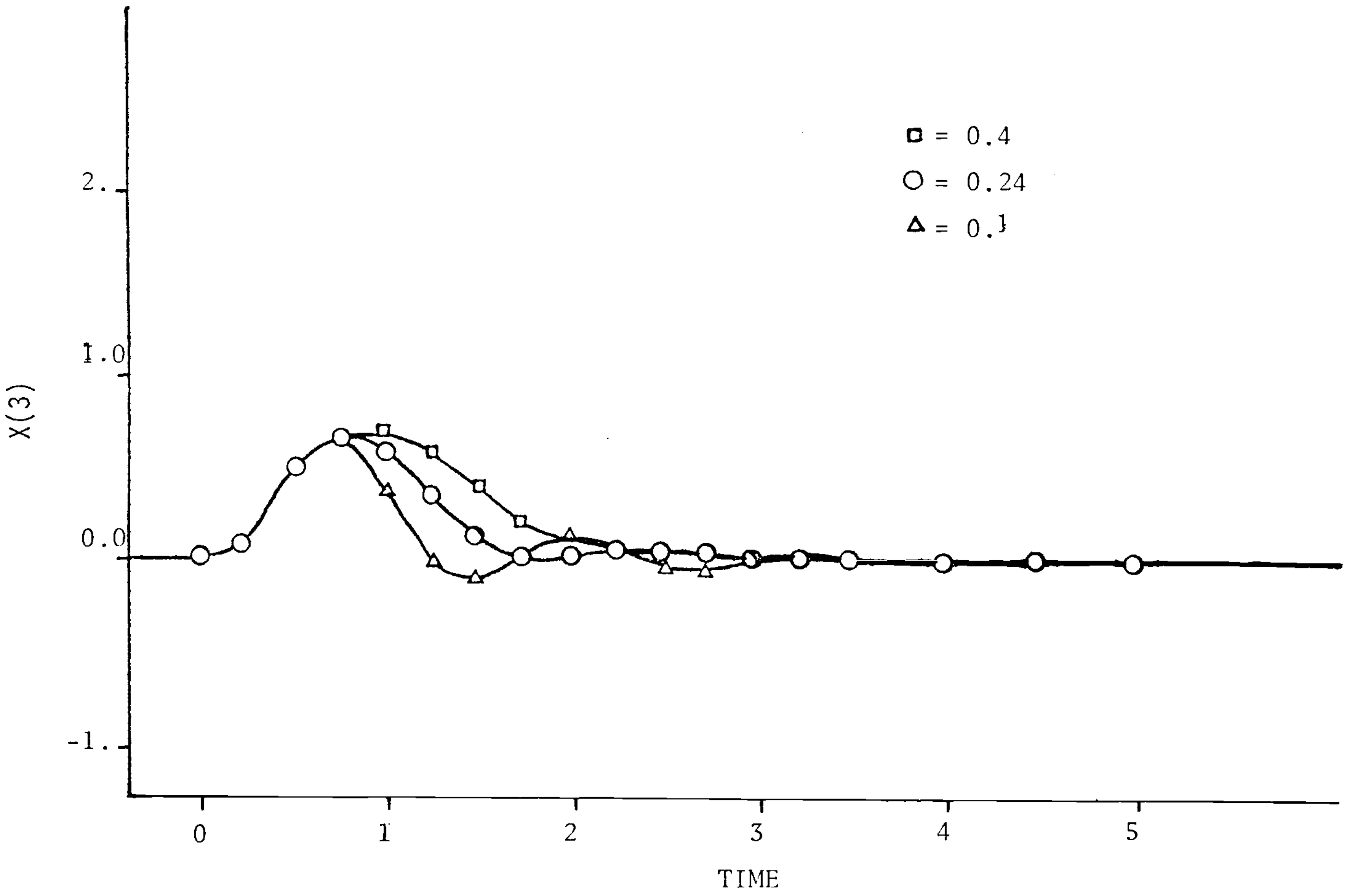


Figure 24. Response of the fifth-order system for different λ values

Simulation of the Control

The system was simulated as a first-order-lag-plus-dead-time and the control equation M was calculated for every 0.05 seconds. A block diagram showing the system is given in Fig. 25. The control action was calculated in plotting section and HPLOT was set to 0.05. A complete listing of the program is given in Appendix G-2.

The results obtained for $\lambda_1 = 0.4$, $\lambda_2 = 0.24$ and $\lambda_3 = 0.1$ were plotted and the responses were found very close to responses obtained with NOVA. Fig. 24 shows the transient responses plotted.

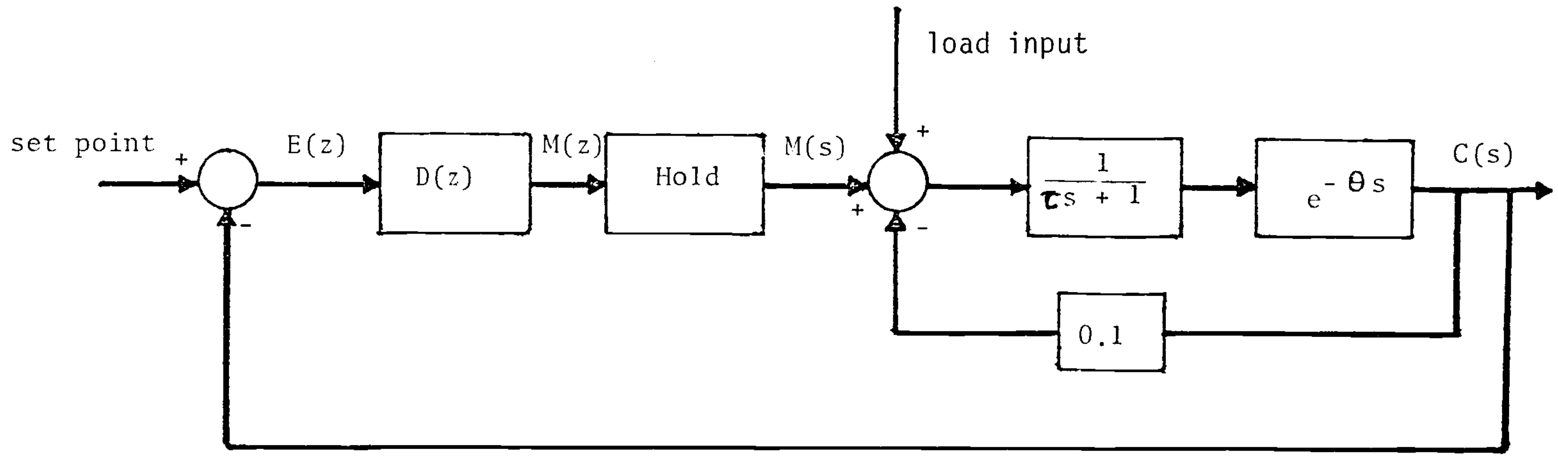


Figure 25. Block diagram of the fifth-order system with DAHLIN's control algorithm

VIII. CONCLUSIONS

An executive system for simulation of dynamic systems and their control was developed. The routines were explicitly tested using different simulation and control techniques. The simulation results of two different systems using conventional and modern control techniques were compared with results obtained from analog computer and NOVA minicomputer for the same systems, and exact agreement was observed in all of the examples. These results demonstrated that the executive system has performed reliable simulation of dynamic systems with or without control. The simplicity of the routines and the ease of implementation on a process control minicomputer make this executive system a useful tool for laboratory and industrial use.

BIBLIOGRAPHY

1. Johnson, A., Dynsys Manual, McMaster University, (1971).

2. Franks, R. G. E., Modeling and Simulation in Chemical Engineering, Wiley-Interscience, (1972).

3. Murrill, P. W., Automatic Control of Processes, Intext Educational Publishers, Scranton, Pa., (1967).

4. Smith, L. S., Digital Computer Process Control, Intext Educational Publishers, Scranton, Pa., (1972).

5. Evans, B. L., A Computer Program for Dynamic Simulation, Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts.

6. England, R., Computer Journal, V. 12, pp. 166-170, (1969).

7. Coughanowr, R. D., Koppel, B. L., Process System Analysis and Control, McGraw-Hill, New York, (1965).

8. Luyben, W. L., Process Modeling, Simulation, and Control for Chemical Engineers, McGraw-Hill, New York, (1973).

9. Fukui, C. S., Development of a Real-Time Digital Computing Facility in Chemical Engineering, Master Thesis, Oregon State University, (1977).

10. Shinskey, F. G., ISA Journal, pp. 61-65, November, (1963).

11. Shinskey, F. G., Process Control Systems, McGraw-Hill, New York, (1967).

12. Lapidus, L., Luus, R., Optimal Control of Engineering Processes, Blaisdell Publishing Co., p. 61, (1967).

13. Cox, J. B., Hellums, L. J., Williams, T. S., Banks, R. S., Kirk, G. J., ISA Journal, pp. 65-72, October, (1966).

14. Dahlin, E. B., Instruments and Control Systems, V. 41, pp. 77-83, June, (1968).

15. Chiu, K., Corripio, A. B., Smith, C. L., Instruments and Control Systems, pp. 57-59, October, (1973).

APPENDICES

APPENDIX A-1

Example Problem for the Use

of the Executive System

Photochemical Reaction carried out isothermally in a CSTR

From L. Lapidus and R. Luus (12)

The molecular equations for the interactions of species A, B, C. D, E, F, and G are given by,

$$A + B \xrightarrow{K_1} 2D$$

$$C + B \xrightarrow{K_2} CB$$

$$CB + B \xrightarrow{K^*} 3E$$

$$E + D \xrightarrow{K_3} 2F$$

$$F + A \xrightarrow{K_4} 2G$$

CB is an intermediate which is present in immesurable quantities, so;

$$\frac{d\ (CB)}{dt} = 0$$

Desired parameter values are:

$K_1 = 17.6 \quad K_2 = 73.0 \quad K_3 = 51.3 \quad K_4 = 23.0$

K^* very large $\quad F_1 = 3.00 \quad F_2 = 4.75 \quad F_3 = 1.25$

$I = 0.60$

where

K's are the reaction rate constants

F's are inlet feed rates

I is light intensity

The rates of reaction can be written for all components in terms of the system parameters and rate constants. After simplification one obtains:

$$\dot{x}_1 = F_1 - F\,x_1 - k_1\,x_1\,x_2 - k_4\,x_1\,x_6\,I^{\frac{1}{2}}$$

$$\dot{x}_2 = F_2 - F\,x_2 - k_1\,x_1\,x_2 - 2\,k_2\,x_2\,x_3$$

$$\dot{x}_3 = F_3 - F\,x_3 - k_2\,x_2\,x_3$$

$$\dot{x}_4 = -F\,x_4 - 2\,k_1\,x_1\,x_2 - k_3\,x_4\,x_5$$

$$\dot{x}_5 = -F\,x_5 + 3\,k_2\,x_2\,x_3 - k_3\,x_4\,x_5$$

$$\dot{x}_6 = -F\,x_6 + 2\,k_3\,x_4\,x_5 - k_4\,x_1\,x_6\,I^{\frac{1}{2}}$$

$$\dot{x}_7 = -F\,x_7 + 2\,k_4\,x_1\,x_6\,I^{\frac{1}{2}}$$

with initial conditions:

$$\text{at } t = 0,\ x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = x_7 = 0$$

A listing of the main program used in the calculations is given on the next page. Also, results are given for the case having values as follows:

HPRINT = 0.1

HPLOT = 0.0

HMIN = 0.001

TMAX = 1.5

ERR = 0.0001

```
      PROGRAM EXAMP1
C
      COMMON/VAL/X(1Ø),G(1Ø),KSTEP
C     EXAMPLE PROGRAM FOR A PHOTOCHEMICAL REACTION CARRIED OUT
C     ISOTHERMALLY IN A CSTR
C
      DIMENSION E(1Ø)
      REAL K1,K2,K3,K4,I
      N=7
      K1=17.6$K2=73.Ø$K3=51.3$K4=23.Ø
      F1=3.ØØ$F2=4.75$F3=1.25$I=Ø.6Ø
      T=Ø.Ø
      DO 6 M=1,7
      E(M)=Ø.Ø
    6 X(M)=Ø.Ø
      READ(6Ø,1ØØ) HPRINT,HPLOT,HMIN,TMAX,H,ERR
  1ØØ FORMAT(6F8.6)
      WRITE(61,2ØØ)
  2ØØ FORMAT(8X,'T'4X'X(1)'4X'X(2)'4X'X(3)'4X'X(4)'4X'X(5)'
     1 4X'X(6)'4X'X(7)')
   1Ø CALL DSIM(N,T,HPRINT,HPLOT,H,HMIN,TMAX,ERR·E,
     1 IERR,ITASK)
      GO TO(1·2,3,4,5) ITASK
    1 GO TO 1Ø
C
    2 F=F1+F2-F3
      G(1)=F1-F*X(1)-K1*X(1)*X(2)-K4*X(1)*X(6)*I**Ø.5
      G(2)=F2-F*X(2)-K1*X(1)*X(2)-2.*K2*X(2)*X(3)
      G(3)=F3-F*X(3)-K2*X(2)*X(3)
      G(4)=-F*X(4)+2.*K1*X(1)*X(2)-K3*X(4)*X(5)
      G(5)=-F*X(5)+3.*K2*X(2)*X(3)-K3*X(4)*X(5)
      G(6)=-F*X(6)+2.*K3*X(4)*X(5)-K4*X(1'*X(6)*I**Ø.5
      G(7)=-F*X(7)+2.*K4*X(1)*X(6)*I**Ø.5
      GO TO 1Ø
    3 WRITE(61,3ØØ) T,X(1),X(2),X(3),X(4),X(5),X(6),X(7)
  3ØØ FORMAT(1X,8F8.6)
    4 GO TO 1Ø
    5 CONTINUE
      END
```

Ø.1 Ø.ØØ1 1.5 Ø.ØØØ1

T	X(1)	X(2)	X(3)	X(4)	X(5)	X(6)	X(7)
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
.1ØØØØ	.17216	.22Ø67	.Ø4879	.Ø46Ø6	.Ø9642	.ØØ855	.ØØØ77
.2ØØØØ	.2Ø458	.24ØØ3	.Ø4762	.Ø8858	.16571	.Ø6866	.Ø1952
.3ØØØØ	.2ØØ47	.24377	.Ø4683	.Ø9512	.177Ø9	.11479	.Ø5472
.4ØØØØ	.19422	.2463Ø	.Ø4645	.Ø9352	.17996	.13173	.Ø8Ø49
.5ØØØØ	.19114	.24795	.Ø462Ø	.Ø9175	.18187	.13635	.Ø9365
.6ØØØØ	.18997	.24881	.Ø46Ø5	.Ø9Ø8Ø	.183Ø9	.13738	.Ø9938
.7ØØØØ	.18957	.24921	.Ø4599	.Ø9Ø39	.18373	.13759	.1Ø17Ø
.8ØØØØ	.18943	.24938	.Ø4596	.Ø9Ø23	.184Ø2	.13764	.1Ø261
.9ØØØØ	.18938	.24944	.Ø4594	.Ø9Ø16	.18414	.13766	.1Ø297
1.ØØØØØ	.18936	.24947	.Ø4594	.Ø9Ø14	.18419	.13766	.1Ø311
1.1ØØØØ	.18935	.24948	.Ø4594	.Ø9Ø13	.18421	.13767	.1Ø317
1.2ØØØØ	.18935	.24949	.Ø4594	.Ø9Ø13	.18422	.13767	.1Ø319
1.3ØØØØ	.18935	.24949	.Ø4593	.Ø9Ø13	.18422	.13767	.1Ø32Ø
1.4ØØØØ	.18935	.24949	.Ø4593	.Ø9Ø12	.18422	.13767	.1Ø32Ø
1.5ØØØØ	.18935	.24949	.Ø4593	.Ø9Ø12	.18422	.13767	.1Ø32Ø

APPENDIX B-1

Listings of Executive Library System Routines

```
      SUBROUTINE DSIM(N,T,HPRINT,HPLOT,H,HMIN,TMAX,ERR,E,
     1 IERR,ITASK)
      COMMON/VAL/ X(10),F(10),KSTEP
C     THIS SUBROUTINE SOLVES A SYSTEM OF ORDINARY FIRST ORDER
C     DIFFERENTIAL EQUATIONS USING RUNGE KUTTA ENGLAND ALGORITHM
      DIMENSION SAVEX(10),SAVEX2(10),AK(10,10),E(10)
      IF (ITASK) 5,5,6
    6 GO TO (101,102,103,104,105,106,107,108,109,110,111,112,113
     1 )KSTEP
    5 IERR=0
      ACCUM1=0.0
      ACCUM2=0.0
      INDEX1=0
      INDEX2=0
C     THE SECTION FOR THE CALCULATION OF THE STARTING STEP SIZE
C
      HH=H
      H=HPLOT/2.
      IF(HPRINT.EQ.0.0.AND.HPLOT.EQ.0.0) GO TO 8
      IF(HPRINT.EQ.0.0) GO TO 13
      IF(HPLOT.EQ.0.0) GO TO 7
      IF(HPRINT-HPLOT) 7,13,13
    7 H=HPRINT/2.
      GO TO 13
    8 IF(HMIN.EQ.0.0.AND.HH.EQ.0.0) 12,9
    9 IF(HMIN.EQ.0.0.AND.HH.NE.0.0) 11,10
   10 H=HMIN
      GO TO 13
   11 H=HH
      GO TO 13
   12 H=0.1
C
C     FIRST PASS
   13 ITASK=1
      KSTEP=1
      RETURN
C
C     SECOND PASS
  101 ITASK=2
      KSTEP=2
      RETURN
C
```

```
C       3TH PASS AND THE LOGIC TO DECIDE FOR PRINTING
  102 IF(HPRINT.EQ.0.0) GO TO 118
      IF(HPRINT.GT.TMAX) GO TO 103
      IF(ABS(ACCUM1-HPRINT).LT.1.E-9) 16,17
   16 INDEX1=0 $ACCUM1=0.
   17 ACCUM1=ACCUM1+2.*H
      IF(ACCUM1.LE.HPRINT) GO TO 18
      IF(ABS(ACCUM1-HPRINT).LT.1.E-9) GO TO 18
      HLAST=H $IH=2
      H=(HPRINT-ACCUM1+2.*H)/2.
      ACCUM1=HPRINT
   18 IF(INDEX1.EQ.0) 117,103
  117 INDEX1=1
  118 ITASK=3
      KSTEP=3
      RETURN
C
C       4TH PASS AND THE LOGIC TO STORE THE VALUES FOR PLOTTING
  103 IF(HPLOT.EQ.0.0) GO TO 121
      IF(HPLOT.GT.TMAX) GO TO 104
      IF(ABS(ACCUM2-HPLOT).LT.1.E-9) 19,20
   19 INDEX2=0 $ACCUM2=0.
   20 ACCUM2=ACCUM2+2.*H
      IF(ACCUM2.LE.HPLOT) GO TO 21
      IF(ABS(ACCUM2-HPLOT).LT.1.E-9) GO TO 21
      IF(IH.EQ.2) GO TO 221
      HLAST=H $IH=2
  221 HACCUM2=H
      H=(HPLOT-ACCUM2+2.*H)/2.
      ACCUM2=HPLOT
      ACCUM1=ACCUM1-HACCUM2*2.+H*2.
   21 IF(INDEX2.EQ.0) 120,104
  120 INDEX2=1
  121 ITASK=4
      KSTEP=4
      RETURN
C
C       5TH PASS AND THE CHECK FOR THE END OF THE CALCULATIONS
  104 IF((TMAX-T).GT.1.E-9) GO TO 23
      ITASK=5
      KSTEP=5
      RETURN
```

```
  105 CONTINUE
C
C       CALCULATION SECTION
C
C       FIRST STEP
   23 T=T+H/2.
      DO 24 J=1,N
      SAVEX(J)=X(J)
      AK(J,1)=F(J)*H
      E(J)=-AK(J,1)
   24 X(J)=SAVEX(J)+0.5*AK(J,1)
      ITASK=2
      KSTEP=6
      RETURN
C       2ND STEP
  106 DO 25 J=1,N
      AK(J,2)=F(J)*H
   25 X(J)=SAVEX(J)+0.25*(AK(J,1)+AK(J,2))
      ITASK=2
      KSTEP=7
      RETURN
C       3RD STEP
  107 T=T+H/2.
      DO 26 J=1,N
      AK(J,3)=F(J)*H
      X(J)=SAVEX(J)-AK(J,2)+2.*AK(J,3)
   26 E(J)=E(J)+4.*AK(J,3)
      ITASK=2
      KSTEP=8
      RETURN
C       4TH STEP
  108 DO 27 J=1,N
      AK(J,4)=F(J)*H
      E(J)=E(J)+17.*AK(J,4)
      X(J)=SAVEX(J)+(AK(J,1)+4.*AK(J,3)+AK(J,4))/6.
   27 SAVEX2(J)=X(J)
      ITASK=2
      KSTEP=9
      RETURN
C       5TH STEP
  109 T=T+H/2.
      DO 28 J=1,N
```

```
      AK(J,5)=F(J)*H
      E(J)=E(J)-23.*AK(J,5)
   28 X(J)=SAVEX2(J)+0.5*AK(J,5)
      ITASK=2
      KSTEP=10
      RETURN
C     6TH STEP
  110 DO 29 J=1,N
      AK(J,6)=F(J)*H
   29 X(J)=SAVEX2(J)+0.25*(AK(J,5)+AK(J,6))
      ITASK=2
      KSTEP=11
      RETURN
C     7TH STEP
  111 T=T+H/2.
      DO 30 J=1,N
      AK(J,7)=F(J)*H
      E(J)=E(J)+4.*AK(J,7)
   30 X(J)=SAVEX(J)+(-AK(J,1)-96.*AK(J,2)+92.*AK(J,3)-
     1121.*AK(J,4)+144.*AK(J,5)+6.*AK(J,6)-12.*AK(J,7))/6.
      ITASK=2
      KSTEP=12
      RETURN
C     CHECK FOR THE ERROR ESTIMATE
  112 DO 31 J=1,N
      AK(J,8)=F(J)*H
      E(J)=(E(J)-AK(J,8))/90.
      IF(ABS(E(J)).GT.ERR) GO TO 32
C     8TH STEP
   31 X(J)=SAVEX2(J)-AK(J,6)+2.*AK(J,7)
      IF(IERR.EQ.0) GO TO 34
      WRITE(61,1010) T
 1010 FORMAT("  SP.ERR.CAN BE OBTAINED FOR H.GE.HMIN ATT- T=   "
     1 E11.5)
      IERR=0
      GO TO 34
C     SECTION FOR HALVING THE STEP SIZE AND GIVING TO THE
C     VARIABLES THEIR PRECEDENT VALUES
   32 IF(IERR.EQ.1) GO TO 34
      T=T-H*2.
      ACCUM1=ACCUM1-H
      ACCUM2=ACCUM2-H
```

```
      DO 33 JJ=1,N
      X(JJ)=SAVEX(JJ)
   33 F(JJ)=AK(JJ,1)/H
      H=H/2.
      IF(H.GE.HMIN) GO TO 14
      ACCUM1=ACCUM1-H*2.+2.*HMIN
      ACCUM2=ACCUM2-H*2.+HMIN*2.
      H=HMIN
      IERR=1
      WRITE(61,1000) T
 1000 FORMAT("  H.LT.HMIN FOR SPECIFIED ERROR AT T="
     1 E11.5)
   14 INDEX1=1
      INDEX2=1
      IH=1
      GO TO 23
   34 CONTINUE
      ITASK=2
      KSTEP=13
      RETURN
C     9TH STEP AND THE LOGIC FOR DOUBLING THE STEP SIZE.
  113 DO 35 J=1,N
      AK(J,9)=F(J)*H
   35 X(J)=SAVEX2(J)+(AK(J,5)+4.*AK(J,7)+AK(J,9))/6.
      DO 36 J=1,N
      IF(ABS(E(J)/ERR).GT. 0.001) GO TO 39
   36 CONTINUE
      IF(IH.EQ.2) 37,38
   37 H=HLAST
      IH=1
      GO TO 13
   38 H=2.*H
      GO TO 13
   39 IF(IH.EQ.2) 40,41
   40 H=HLAST
      IH=1
   41 GO TO 13
      END
```

```
      SUBROUTINE TRFN(NXI,NXO,TC,GAIN,N)
      COMMON/VAL/ X(10),F(10),KSTEP
      M=NXO+N-1
      DO 1 I=NXO,M
      IO=M-I+NXO
      II=IO+1
      IF(I.EQ.NXO) II=NXI
    1 F(IO)=(GAIN*X(II)-X(IO))/TC
      RETURN
      END
```

```
      SUBROUTINE TRF2(NXI,NXO,TC,DAMP,GAIN,TC2)
      COMMON/VAL/ X(10),F(10),KSTEP
      IF(TC2.LT.1.E-9) 1,3
    1 A1=1./(TC**2.)
      A2=2.*DAMP/TC
    2 NXD=NXO+1
      F(NXD)=(GAIN*X(NXI)-X(NXO))*A1-A2*X(NXD)
      F(NXO)=X(NXD)
      RETURN
    3 A1=1./(TC*TC2)
      A2=(TC+TC2)/(TC*TC2)
      GO TO 2
      END
```

```
SUBROUTINE PRCONTR(NXI,NXO,ZR,RAN,ACT,SP,PB,XMN)
COMMON/VAL/ X(10),F(10),KSTEP
RANGE=ABS(RAN-ZR)
YN=X(NXI)
IF(X(NXI).GT.RAN) YN=RAN
IF(X(NXI).LT.ZR) YN=ZR
ERR=100.*ACT*(YN-SP)/RANGE
X(NXO)=100./PB*ERR+XMN
IF(X(NXO).LT.0.0) X(NXO)=0.0
IF(X(NXO).GT.100.) X(NXO)=100.0
RETURN
END
```

```
SUBROUTINE PICONTR(NXI,NXO,ZR,RAN,SP,ACT,PB,RT)
COMMON/VAL/ X(10),F(10),KSTEP
RANGE=ABS(RAN-ZR)
YI=X(NXI)
IF(X(NXI).LT.ZR) YI=ZR
IF(X(NXI).GT.RAN) YI=RAN
ERR=100.*ACT*(YI-SP)/RANGE
NXD=NXI+1
F(NXD)=ERR*RT*100./PB
IF(X(NXD).LT.0.) X(NXD)=0.0
IF(X(NXD).GT.100.) X(NXD)=100.
X(NXO)=100./PB*ERR+X(NXD)
IF(X(NXO).GT.100.) X(NXO)=100.
IF(X(NXO).LT.0.) X(NXO)=0.
RETURN
END
```

```
      SUBROUTINE PIDCON(NXI,NXO,ZR,RAN,SP,ACT,PB,RPT,RT,RA)
      COMMON/VAL/ X(10),F(10),KSTEP
      YI=100.*(X(NXI)-ZR)/ABS(RAN-ZR)
      IF(YI.LT.0.) YI=0.
      IF(YI.GT.100.) YI=100.
      IF(IFLAG.EQ.1) GO TO 1
      IFLAG=1
      NXL=NXI+1
      X(NXL)=YI*(1./RA-1.)
    1 YD=RA*(YI+X(NXL))
      F(NXL)=(YI-YD)/RT
      VNOR=100.*(SP-ZR)/ABS(RAN-ZR)
      ERR=(YD-VNOR)*ACT*100./PB
      NXD=NXL+1
      IF(X(NXD).LT.0.) X(NXD)=0.
      IF(X(NXD).GT.100.) X(NXD)=100.
      F(NXD)=ERR*RPT
      X(NXO)=ERR+X(NXD)
      IF(X(NXO).LT.0.) X(NXO)=0.
      IF(X(NXO).GT.100.) X(NXO)=100.
      RETURN
      END
```

```
      SUBROUTINE DPICON(NXI,NXO,ZR,RAN,SP,ACT,PB,TI,DELT)
C
      COMMON/VAL/ X(10),F(10),KSTEP
      COMMON/E/SUME(10)
C
      YI=X(NXI)
      IF(X(NXI).LT.ZR) YI=ZR
      IF(X(NXI).GT.RAN) YI=RAN
C
      ERR=ACT*(YI-SP)
C
      SUME(NXI)=SUME(NXI)+ERR
      X(NXO)=100./PB*(ERR+SUME(NXI)*TI*DELT)
C
      RETURN
      END
```

```
      SUBROUTINE DPIDCON(NXI,NXO,ZR,RAN,SP,ACT,PB,TI,TR,DELT)
C
      COMMON/VAL/X(10),F(10),KSTEP
      COMMON/E/SUME(10)
      DIMENSION XERR(10,4)
C
      YI=X(NXI)
      IF(X(NXI).LT.ZR) YI=ZR
      IF(X(NXI).GT.RAN) YI=RAN
C
      XERR(NXI,4)=ACT*(YI-SP)
      SUME(NXI)=SUME(NXI)+XERR(NXI,4)
      DELTA=1./(6.*DELT)*(XERR(NXI,4)-XERR(NXI,1)+3.*XERR
     $(NXI,3)-3.*XERR(NXI,2))
      X(NXO)=100./PB*(XERR(NXI,4)+SUME(NXI)*TI*DELT+TR*DELTA)
C
      DO 1 I=1,3
      N=I+1
    1 XERR(NXI,I)=XERR(NXI,N)
      RETURN
      END
```

```
      FUNCTION STEP(T,A,B,X1,X2,X3)
      IF((A-T).GT.1.E-9) 1,2
    1 STEP=X1
      RETURN
    2 IF(B.EQ.0.) GO TO 3
      IF((T-B).GT.1.E-9) 4,3
    3 STEP=X2
      RETURN
    4 STEP=X3
      RETURN
      END
```

```
      FUNCTION PEAK(T,A,B,X1,X2,X3)
      IF((A-T).GT.1.E-9) 1,2
    1 PEAK=X1
      RETURN
    2 AB=A+(B-A)/2.
      IF((AB-T).GT.1.E-9) 3,4
    3 PEAK=X1+(X2-X1)*(T-A)/(AB-A)
      RETURN
    4 IF((B-T).GT.1.E-9) GO TO 5
      PEAK=X3
      RETURN
    5 PEAK=X2+(X3-X2)*(T-AB)/(B-AB)
      RETURN
      END
```

```
      FUNCTION SPULSE(T,A,B,X1,X2,X3)
      IF((A-T).GT.1.E-9) 1,2
    1 SPULSE=X1
      RETURN
    2 IF((B-T).GT.1.E-9) GO TO 3
      SPULSE=X3
      RETURN
    3 PI=3.141592
      SPULSE=(X2-X1)*SIN(PI/(B-A)*(T-A))+X1
      RETURN
      END
```

```
      SUBROUTINE CONV(X,Y,NR,NC)
      DIMENSION XA(10),YA(10)
      IF(ABS((X-Y)/(X+Y)).LT..0001) GO TO 6
      IF(NC.LE.1) GO TO 5
      XT=(XA(NR)*Y-YA(NR)*X)/(XA(NR)-X+Y-YA(NR))
      XA(NR)=X
      YA(NR)=Y
      X=XT
      RETURN
    5 XA(NR)=X
      YA(NR)=Y
      X=Y
      NC=2
      RETURN
    6 X=Y
      NC=1
      RETURN
      END
      FUNCTION STEP(T,A,B,X1,X2,X3)
      IF((A-T).GT.1.E-9) 1,2
    1 STEP=X1
      RETURN
    2 IF(B.EQ.0.) GO TO 3
      IF((T-B).GT.1.E-9) 4,3
    3 STEP=X2
      RETURN
    4 STEP=X3
      RETURN
      END
```

```
      FUNCTION FUN(A,N,X,Y)
      DIMENSION X(10),Y(10)
      IF(A-X(1)) 5,5,6
    6 IF(A-X(N)) 1,2,2
    2 FUN=Y(N)
      RETURN
    5 FUN=Y(1)
      RETURN
    1 DO 3 I=2,N
      IF(A.LT.X(I)) GO TO 4
    3 CONTINUE
    4 II=I-1
      FUN=Y(II)+(A-X(II))*(Y(I)-Y(II))/(X(I)-X(II))
      RETURN
      END
```

```
      SUBROUTINE VALVE(NXI,NXO,P1,P2,LV,KT,VC,R)
      COMMON/VAL/ X(10),F(10),KSTEP
      SP=X(NXI)
      PD=ABS(P1-P2)
      IF(PD.LT.1.E-9) GO TO 7
      GO TO (1,2,3) KT
    1 A=SP/100.
      GO TO 4
    2 A=R*EXP(-SP*ALOG(R))/100.
      GO TO 4
    3 A=R*SQRT(SP)/100.
    4 IF(LV.EQ.0) GO TO 5
      X(NXO)=A*VC*SQRT(PD)*(P1-P2)/PD
      RETURN
    5 IF(P2.LT.0.53*P1) GO TO 6
      X(NXO)=A*VC*SQRT(P1*PD)*(P1-P2)/PD
      RETURN
    6 X(NXO)=A*VC*P1*0.85
      RETURN
    7 X(NXO)=0.0
      RETURN
      END
```

```
      SUBROUTINE XDEL(NXI,NXO,DELAY,T,JC)
      COMMON/VAL/ X(10),F(10),KSTEP
      DIMENSION XA(25,200),TIME(25,200),MM(25),NN(25)
C
C     THIS SUBROUTINE DELAYS THE VARIABLE X(NXI) FOR A CERTAIN
C     TIME PERIOD,DELAY. THE OUTPUT IS THE DELAYED
C     VARIABLE X(NXO).
C
C     XA=TWO DIMENSIONAL ARRAY TO STORE STATE VARIABLES
C     X(NXI)=INPUT TO STORE
C     X(NXO)=OUTPUT DELAYED
C     DELAY=DELAY TIME
C     T=TIME
C     JC=SUBROUTINE NUMBER
C
      M=MM(JC)
      N=NN(JC)
      IF(T.LE.0.0) GO TO 1
      IF(KSTEP.EQ.6.OR.KSTEP.EQ.8) GO TO 3
      IF(KSTEP.EQ.10.OR.KSTEP.EQ.12) GO TO 3
   10 IF((DELAY-T).GT.1.E-9) GO TO 2
      IF(ABS(TIME(JC,M)-T).LT.1.E-9) GO TO 100
      IF((T-TIME(JC,M)).GT.1.E-9) GO TO 200
      IF((TIME(JC,M)-T).GT.1.E-9) GO TO 300
C
    1 M=1
      N=1
      IFLAG=1
      DO 13 I=1,200
   13 XA(JC,I)=X(NXI)
      XA(JC,N)=X(NXI)
      TIME(JC,N)=T+DELAY
      X(NXO)=XA(JC,N)
      MM(JC)=M
      NN(JC)=N
      RETURN
C
    2 X(NXO)=XA(JC,200)
      IF(KSTEP.EQ.13) GO TO 3
      N=N+1
      IF(N.EQ.201) 21,22
   21 WRITE(61,1000)
```

```
 1000 FORMAT(1X," DIMENSION FOR STORAGE SHOULD BE MADE LARGER")
      CALL EXIT
   22 XA(JC,N)=X(NXI)
      TIME(JC,N)=T+DELAY
      IM=N-1
      IF(TIME(JC,N).LT.TIME(JC,IM)) 20,3
   20 N=N-3
      XA(JC,N)=X(NXI)
      TIME(JC,N)=T+DELAY
    3 MM(JC)=M
      NN(JC)=N
      RETURN
C
C     CHECK FOR THE STEP
  100 GO TO(3,55,3,3,3,3,4,3,5,3,6,3,7)KSTEP
    4 IF(IFLAG.EQ.1) GO TO 5
      IF(N.LT.4) N=200+N
      IF(M.LT.3) M=200+M
      N=N-3
      M=M-2
      IFLAG=1
      GO TO 10
    5 X(NXO)=XA(JC,M)
   55 N=N+1
      IF(N.EQ.201) N=1
      M=M+1
      IF(M.EQ.201) M=1
      XA(JC,N)=X(NXI)
      TIME(JC,N)=T+DELAY
      MM(JC)=M
      NN(JC)=N
      RETURN
    6 IFLAG=2
      GO TO 5
    7 IFLAG=1
      X(NXO)=XA(JC,M)
      RETURN
C
C     SEARCH FOR THE EXACT TIME WHICH IS EQUAL TO ACTUAL TIME.
C
  200 NM=M+100
      DO 8 I=M,NM
```

```
      II=I
      IF(I.GT.200) 14,15
   14 II=I-200
   15 IF(TIME(JC,II).GE.T) GO TO 9
    8 CONTINUE
    9 TL1=TIME(JC,II)-T
      KI=II-1
      TL2=TIME(JC,II)-TIME(JC,KI)
C     LINEAR INTERPOLATION
      XX=XA(JC,II)-(XA(JC,II)-XA(JC,KI))*TL1/TL2
      XA(JC,II)=XX
      TIME(JC,II)=T
      M=II
      GO TO 100
C
C     SEARCH FOR THE EXACT TIME WHICH IS EQUAL TO ACTUAL TIME.
C
  300 KK=M+100
      DO 11 K=M,KK
      KN=2*M-K-1
      IF(KN.LE.1) GO TO 100
      IF(T.GE.TIME(JC,KN)) GO TO 12
   11 CONTINUE
   12 TG1=T-TIME(JC,KN)
      KNN=KN+1
C     LINEAR INTERPOLATION
      TG2=TIME(JC,KNN)-TIME(JC,KN)
      XY=XA(JC,KN)+(XA(JC,KNN)-XA(JC,KN))*TG1/TG2
      TIME(JC,KN)=T
      XA(JC,KN)=XY
      M=KN
      GO TO 100
      END
```

APPENDIX B-2

Sample Problem for Testing Some of the Executive Library System Routines

```
      PROGRAM CASE
      COMMON/VAL/ X(10),F(10),KSTEP
      DIMENSION E(10),SPA(10),AVA(10)
      REAL K
      DATA((SPA(I),I=1,10)=0.,20.,40.,55.,67.5,77.,86.,93.,
     * 97.,100.)
      DATA((AVA(I),I=1,10)=0.,4.,13.,20.5,30.,40.,55.,71.,
     * 84.,100.)
C     INITIAL CONDITIONS AND PARAMETERS
      TW=40.0 $ FW=6877.0
      TC2=0.0
      T=0.0 $ TMAX=1.0 $ HPRINT=0.1
      NC=1
      HMIN=0.0001 $ ERR=0.0001
      X(1)=CA=1.0 $ X(2)=CB=0.8
      X(3)=TJ=40.0 $ X(4)=TC=70.0
      X(5)=TCM=70.0
      X(9)=FC=0.0 $ X(6)=DFC=60.0
      X(7)=SP=60.0 $ X(8)=DSP=0.0
      PB=10.0 $ RPM=0.5
      PD=30.0
   10 CALL DSIM(8,T,HPRINT,0.,0.,HMIN,TMAX,ERR,
     * E,IERR,ITASK)
      GO TO(1,2,3,4,5) ITASK
    1 GO TO 10
    2 CALL TRFN(4,5,0.2,1.0,1)
      CALL PICONTR(5,9,30.,130.,70.,1.,PB,RPM)
      CALL TRF2(9,7,0.1,1.0,1.,TC2)
      AV=FUN(X(7),10,SPA,AVA)
    6 FW=60.*AV*SQRT(40.-PD)
      PD1=0.2E-5*FW**1.6+15.
      CALL CONV(PD,PD1,1,NC)
      IF(NC.EQ.2) GO TO 6
      K=2.58E5*EXP(-5000./(X(4)+273.))
      R=K*X(1)*X(2)
      Q=4.E4*(X(3)-X(4))
      F(1)=-R
      F(2)=-R
      F(3)=(FW*(TW-X(3))-Q)/2000.0
      F(4)=200.*R+Q/16200.0
      GO TO 10
    3 WRITE(61,100) T,X(4),X(3),X(1),X(5),X(9),X(7),AV,FW,PD
  100 FORMAT(F10.4,2X,5E12.5/12X,5E12.5)
    4 GO TO 10
    5 CONTINUE
      END
```

Case Study : Temperature Control of a Batch Reactor

The exothermic reaction $A + B \longrightarrow C + D$ is carried out in a jacketed reactor. A variable flow of cooling water F_w passes through the jacket entering at a temperature of T_{ji} (40^0C)

The control of the temperature is obtained using the following data:

1. Total volume of the liquid = $30Ft^3$,no density change with reaction

2. Initial charge: 30 moles A

   24 moles B

3. The reaction rate is second order, proportional to the concentration(mole/Ft^3) of each component. The rate coefficient is

   $k = 2.58.10^5.e^{-5000/T(^0K)}$ (Ft^3/min mole)

4. The heat of reaction is $10.8.10^4$ PCU/mole of A or B reacting and the average heat capacity of the reaction mass is 300PCU/mole^{0}C

5. The overall heat transfer coefficient between jacket and reactor contents is 4000PCU/^{0}C min. The heat capacity of the water in the jacket is 2000 PCU/^{0}C.

6. The pressure drop across the jacket is a function of the coolant flow F_w: $P_D - P_0 = 0.2.10^{-5} F_w^{1.6}$

7. Upstream pressure PU :40.PSIA

   15.PSIA

8. Control valve CV : 6000(lbs/min PSI$^{\frac{1}{2}}$)

The area characteristics are given as:

% stem position	valve area % open
0	0
20	4
40	13
55	20.5
67.5	30
77	40
86	55
93	71
97	84
100	100

9. The thermowell can be approximated by a first-order response with a 0.2 min time constant

10. The dynamic response of the valve follows a second-order response ; i.e,

$$\frac{OUT}{IN} = \frac{1}{0.01s^2 + 0.2s + 1}$$

11. Control instrument has a range of 30-130^{0}C

Its proportional band is 10.0 and its RPM(repeats per minute) is 0.5.

0	7.00000E 01	4.00000E 01	1.00000E 00	7.00000E 01	6.00000E 01
	6.00000E 01	2.43000E 01	6.87597E 03	1.77591E 01	
.1000	6.84468E 01	6.20657E 01	9.90939E-01	6.95047E 01	5.49395E 01
	5.95871E 01	2.39862E 01	6.79507E 03	1.77073E 01	
.2000	6.89550E 01	6.44666E 01	9.82110E-01	6.91830E 01	5.13769E 01
	5.74358E 01	2.23512E 01	6.36956E 03	1.74412E 01	
.3000	6.96682E 01	6.54804E 01	9.73220E-01	6.92423E 01	5.15630E 01
	5.47587E 01	2.03794E 01	5.84768E 03	1.71291E 01	
.4000	7.04446E 01	6.63287E 01	9.64230E-01	6.95740E 01	5.45751E 01
	5.33345E 01	1.96673E 01	5.65690E 03	1.70191E 01	
.5000	7.12413E 01	6.70418E 01	9.55130E-01	7.00863E 01	5.96065E 01
	5.39800E 01	1.99900E 01	5.74352E 03	1.70688E 01	
.6000	7.20183E 01	6.75530E 01	9.45922E-01	7.07077E 01	6.60166E 01
	5.67452E 01	2.18263E 01	6.23157E 03	1.73571E 01	
.7000	7.27153E 01	6.77453E 01	9.36632E-01	7.13756E 01	7.32154E 01
	6.13031E 01	2.52903E 01	7.12966E 03	1.79237E 01	
.8000	7.32767E 01	6.76706E 01	9.27305E-01	7.20275E 01	8.05866E 01
	6.71673E 01	2.97471E 01	8.24009E 03	1.86857E 01	
.9000	7.36565E 01	6.72433E 01	9.18006E-01	7.26070E 01	8.75445E 01
	7.37809E 01	3.66114E 01	9.84790E 03	1.99021E 01	
1.0000	7.38037E 01	6.64813E 01	9.08817E-01	7.30596E 01	9.34933E 01
	8.05645E 01	4.59408E 01	1.18318E 04	2.15752E 01	

APPENDIX C-1

The Gas Absorber

The gas absorber consists of two plates, where air containing SO_2 gas is contacted with fresh water in order to remove part of SO_2 from the gas.

The symbols used in the formulation are:

H = holdup for each plate

TC = the liquid dynamic time constant

V = flow rate of air - SO_2 mixture

X = concentration of liquid

Y = concentration of gas

Fig. 26 shows the gas absorber described. The dynamic equations of the system are:

$$\frac{dX(1)}{dt} = \frac{1}{H}(X(4)\ X(5) - X(3)\ X(1)) + \frac{Vm}{H}(X_o - X(1))$$

$$\frac{dX(5)}{dt} = \frac{Vm}{H}(X(1) - X(5)) - \frac{1}{H}X(4)\ X\ (5)$$

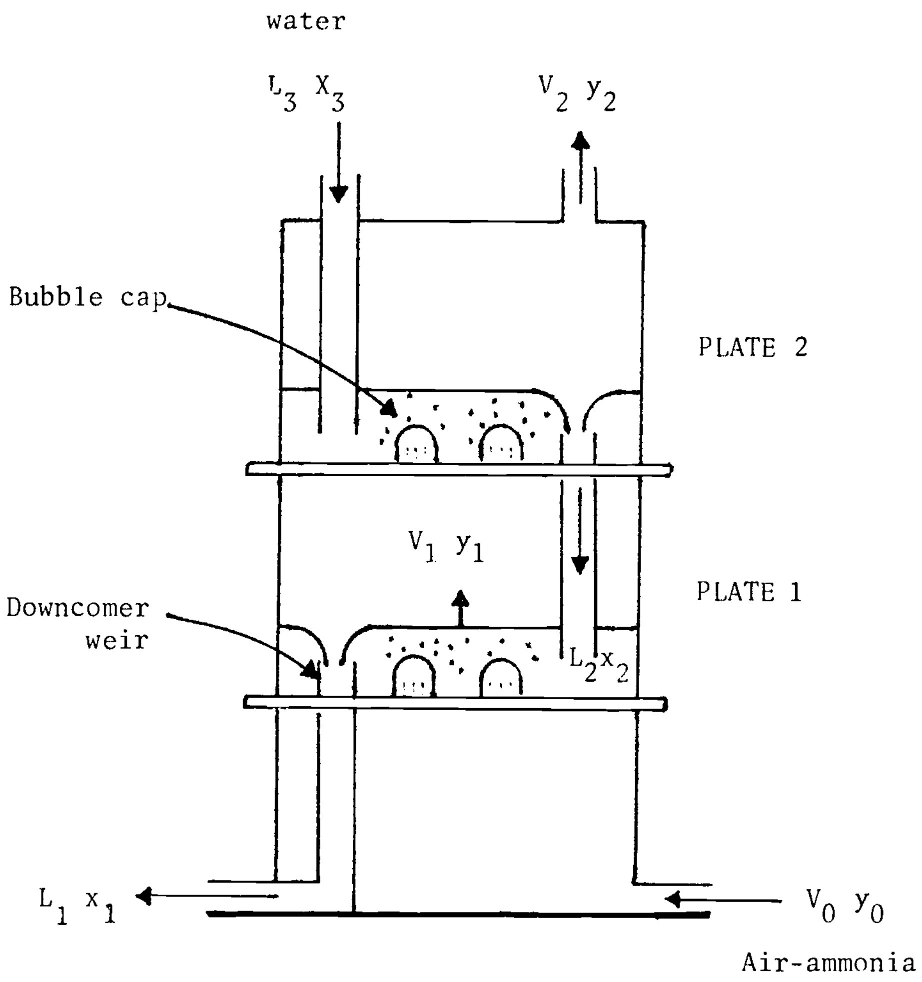
$$\frac{dX(3)}{dt} = \frac{X(8)}{TC} - \frac{X(4)}{TC}$$

$$\frac{dX(4)}{dt} = \frac{X(3)}{TC} - \frac{X(4)}{TC}$$

The following assumptions were made:

1) Temperature and pressure are constant throughout the column.

2) The plate efficiency is 100 percent; the plates are ideal.



L = liquid flow rate

V = gas flow rate

x = liquid concentration

y = gas concentration

Figure 26. The gas absorber

3) The equilibrium relation is:

$$y_n = mx_n + b$$

where m and be are constants.

4) The holdup of liquid H, is constant and is the same for each plate.

5) The gas flow rate V is constant on each plate.

APPENDIX C-2

Main Program Listing and Results

for the Simulation of the Gas Absorber

```
      PROGRAM CONTROL
      COMMON/VAL/ X(10),F(10),KSTEP
      DIMENSION E(10)
      N=5
      H=0.1666 $ TC=0.1
      VM=1.377 $ X(8)=0.4
      A=1.0 $ B=2.5
      Y1=0.00086
      Z1=0.4
      C=4.0 $ D=6.0
      X(1)=0.0008072
      X(3)=0.40
      X(4)=0.40
      X(5)=0.0006255
      READ (60,100) HPRINT,HMIN,TMAX,ERR
      READ(60,100) Y2,Y3,Z2,Z3
  100 FORMAT(4F8.6)
   10 X0=STEP(T,A,B,Y1,Y2,Y3)
      X(8)=STEP(T,C,D,Z1,Z2,Z3)
      CALL DSIM(N,T,HPRINT,0.0,0.0,HMIN,TMAX,ERR,
     $ E,IERR,ITASK)
      GO TO(1,2,3,4,5) ITASK
    1 GO TO 10
    2 F(1)=1./H*(X(3)*X(5)-X(4)*X(1))+VM/H*(X0-X(1))
      F(5)=VM/H*(X(1)-X(5))-1./H*X(3)*X(5)
      F(3)=X(8)/TC-X(3)/TC
      F(4)=X(3)/TC-X(4)/TC
      GO TO 10
    3 WRITE(61,200) T,X(1),X(5),X(3),X(4)
  200 FORMAT(5F10.7)
    4 GO TO 10
    5 CONTINUE
      END
```

Ø.1 Ø.ØØ1 3.Ø Ø.ØØØ1

Ø.ØØØ96 Ø.ØØØ96 Ø.4 Ø.4

C

Ø	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
.1ØØØØØØ	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
.2ØØØØØØ	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
.3ØØØØØØ	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
.4ØØØØØØ	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
.5ØØØØØØ	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
.6ØØØØØØ	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
.7ØØØØØØ	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
.8ØØØØØØ	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
.9ØØØØØØ	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
1.ØØØØØØØ	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
1.1ØØØØØØ	.ØØØ8593	.ØØØ6431	.4ØØØØØØ	.4ØØØØØØ
1.2ØØØØØØ	.ØØØ88Ø5	.ØØØ6651	.4ØØØØØØ	.4ØØØØØØ
1.3ØØØØØØ	.ØØØ89Ø5	.ØØØ6796	.4ØØØØØØ	.4ØØØØØØ
1.4ØØØØØØ	.ØØØ8955	.ØØØ6881	.4ØØØØØØ	.4ØØØØØØ
1.5ØØØØØØ	.ØØØ8981	.ØØØ6927	.4ØØØØØØ	.4ØØØØØØ
1.6ØØØØØØ	.ØØØ8995	.ØØØ6953	.4ØØØØØØ	.4ØØØØØØ
1.7ØØØØØØ	.ØØØ9ØØ2	.ØØØ6967	.4ØØØØØØ	.4ØØØØØØ
1.8ØØØØØØ	.ØØØ9ØØ6	.ØØØ6974	.4ØØØØØØ	.4ØØØØØØ
1.9ØØØØØØ	.ØØØ9ØØ8	.ØØØ6978	.4ØØØØØØ	.4ØØØØØØ
2.ØØØØØØØ	.ØØØ9ØØ9	.ØØØ698Ø	.4ØØØØØØ	.4ØØØØØØ
2.1ØØØØØØ	.ØØØ9Ø1Ø	.ØØØ6981	.4ØØØØØØ	.4ØØØØØØ
2.2ØØØØØØ	.ØØØ9Ø1Ø	.ØØØ6982	.4ØØØØØØ	.4ØØØØØØ
2.3ØØØØØØ	.ØØØ9Ø11	.ØØØ6982	.4ØØØØØØ	.4ØØØØØØ
2.4ØØØØØØ	.ØØØ9Ø11	.ØØØ6982	.4ØØØØØØ	.4ØØØØØØ
2.5ØØØØØØ	.ØØØ9Ø11	.ØØØ6982	.4ØØØØØØ	.4ØØØØØØ
2.6ØØØØØØ	.ØØØ9Ø11	.ØØØ6982	.4ØØØØØØ	.4ØØØØØØ
2.7ØØØØØØ	.ØØØ9Ø11	.ØØØ6982	.4ØØØØØØ	.4ØØØØØØ
2.8ØØØØØØ	.ØØØ9Ø11	.ØØØ6982	.4ØØØØØØ	.4ØØØØØØ
2.9ØØØØØØ	.ØØØ9Ø11	.ØØØ6982	.4ØØØØØØ	.4ØØØØØØ
3.ØØØØØØØ	.ØØØ9Ø11	.ØØØ6982	.4ØØØØØØ	.4ØØØØØØ

0.1 0.001 3.0 0.0001

0.00086 0.00086 0.2 0.2

0	.0008072	.0006255	.4000000	.4000000
.1000000	.0008072	.0006255	.4000000	.4000000
.2000000	.0008072	.0006255	.4000000	.4000000
.3000000	.0008072	.0006255	.4000000	.4000000
.4000000	.0008072	.0006255	.4000000	.4000000
.5000000	.0005072	.0006255	.4000000	.4000000
.6000000	.0008072	.0006255	.4000000	.4000000
.7000000	.0008072	.0006255	.4000000	.4000000
.8000000	.0008072	.0006255	.4000000	.4000000
.9000000	.0008072	.0006255	.4000000	.4000000
1.0000000	.0008072	.0006255	.4000000	.4000000
1.1000000	.0007962	.0006421	.2735774	.3471469
1.2000000	.0007971	.0006633	.2270682	.2811987
1.3000000	.0008081	.0006824	.2099580	.2398257
1.4000000	.0008201	.0006988	.2036634	.2183156
1.5000000	.0008294	.0007118	.2013477	.2050856
1.6000000	.0008357	.0007213	.2004955	.2034704
1.7000000	.0008395	.0007276	.2001824	.2014591
1.8000000	.0008417	.0007316	.2000671	.2006039
1.9000000	.0008429	.0007341	.2000247	.2002468
2.0000000	.0008436	.0007355	.2000091	.2000999
2.1000000	.0008440	.0007363	.2000033	.2000401
2.2000000	.0008442	.0007368	.2000012	.2000160
2.3000000	.0008443	.0007370	.2000005	.2000063
2.4000000	.0008444	.0007372	.2000002	.2000025
2.5000000	.0008444	.0007373	.2000001	.2000010
2.6000000	.0008444	.0007373	.2000000	.2000004
2.7000000	.0008444	.0007373	.2000000	.2000001
2.8000000	.0008444	.0007373	.2000000	.2000001
2.9000000	.0009444	.0007373	.2000000	.2000000
3.0000000	.0008444	.0007373	.2000000	.2000000

APPENDIX C-3

Main Program Listing and Results

for the Simulation of the Fifth-Order System

```
      PROGRAM ALTI
C
      COMMON/VAL/ X(10),F(10),KSTEP
      DIMENSION E(10)
C
C     INITIAL CONDITIONS AND CONSTANTS
C
      N=3
      A=0.0 $ B=5.0
      Y1=0.5 $ Y2=1.0 $ Y3=1.0
      X(2)=0.455 $ X(3)=0.455
      X(4)=0.0
      X(5)=0.455 $ X(8)=0.0
      READ(61,100) HPRINT,HPLOT,HMIN,TMAX,ERR
      READ(60,100) DEL2,T1,T2,TT1,TT2
  100 FORMAT(5F8.6)
      WRITE(61,200)
  200 FORMAT(3X,"TIME",6X,"X(1)",6X,"X(2)",6X,"X(3)",6X,"X(7)",
     $6X,"X(5)",///)
C
   10 X(1)=STEP(T,A,B,Y1,Y2,Y3)
      CALL DSIM(N,T,HPRINT,HPLOT,0.0,HMIN,TMAX,ERR,E,
     $ IERR,ITASK)
      GO TO (1,2,3,4,5) ITASK
    1 GO TO 10
C
C     DERIVATIVE SECTION
C
    2 X(7)=X(1)-X(9)
      CALL TRFN(7,2,T1,1.,1)
      CALL XDEL(2,5,0.146,T,1)
      X(6)=X(5)+X(4)
      CALL TRFN(6,3,T2,1.,1)
      X(9)=X(3)*0.1
      GO TO 10
C
C     PRINTING SECTION
    3 WRITE(61,300) T,X(1),X(2),X(3),X(7),X(5)
  300 FORMAT(6F10.7)
      GO TO 10
    4 CONTINUE
      GO TO 10
```

```
 5 CONTINUE
   END
```

Ø.1 Ø.Ø Ø.ØØ1 5.Ø Ø.ØØ1

Ø.Ø Ø.334 Ø.1 Ø.Ø Ø.Ø

TIME	X(1)	X(2)	X(3)	X(7)	X(5)
Ø	1.ØØØØØØØ	.455ØØØØ	.455ØØØØ	1.ØØØØØØØ	.455ØØØØ
.1ØØØØØØ	1.ØØØØØØØ	.585Ø8Ø3	.455ØØØØ	.9545ØØØ	.455ØØØØ
.2ØØØØØØ	1.ØØØØØØØ	.68Ø5625	.4718954	.95Ø7551	.53Ø37Ø7
.3ØØØØØØ	1.ØØØØØØØ	.7481436	.5486743	.9435256	.64Ø1Ø86
.4ØØØØØØ	1.ØØØØØØØ	.7964779	.635816Ø	.9352Ø19	.72Ø8583
.5ØØØØØØ	1.ØØØØØØØ	.83Ø6527	.7Ø97676	.9281347	.7771361
.6ØØØØØØ	1.ØØØØØØØ	.8545869	.7666622	.9226989	.817ØØ72
.7ØØØØØØ	1.ØØØØØØØ	.871273Ø	.8Ø85639	.9186956	.845Ø58Ø
.8ØØØØØØ	1.ØØØØØØØ	.8828754	.838669Ø	.9158193	.8646431
.9ØØØØØØ	1.ØØØØØØØ	.89Ø9289	.8599836	.9137828	.87827Ø6
1.ØØØØØØØ	1.ØØØØØØØ	.896513Ø	.8749434	.9123535	.8877348
1.1ØØØØØØ	1.ØØØØØØØ	.9ØØ3822	.8853875	.9113556	.8942992
1.2ØØØØØØ	1.ØØØØØØØ	.9Ø3Ø62Ø	.8926553	.91Ø6612	.8988487
1.3ØØØØØØ	1.ØØØØØØØ	.9Ø49176	.8977Ø25	.91Ø179Ø	.9Ø2ØØØ1
1.4ØØØØØØ	1.ØØØØØØØ	.9Ø62Ø22	.9Ø12Ø32	.9Ø98445	.9Ø41824
1.5ØØØØØØ	1.ØØØØØØØ	.9Ø7Ø915	.9Ø36292	.9Ø96127	.9Ø56933
1.6ØØØØØØ	1.ØØØØØØØ	.9Ø77Ø7Ø	.9Ø53Ø97	.9Ø94522	.9Ø67392
1.7ØØØØØØ	1.ØØØØØØØ	.9Ø81331	.9Ø64735	.9Ø9341Ø	.9Ø74632
1.8ØØØØØØ	1.ØØØØØØØ	.9Ø8428Ø	.9Ø72791	.9Ø9264Ø	.9Ø79643
1.9ØØØØØØ	1.ØØØØØØØ	.9Ø86321	.9Ø78369	.9Ø921Ø7	.9Ø83112
2.ØØØØØØØ	1.ØØØØØØØ	.9Ø87734	.9Ø82229	.9Ø91738	.9Ø85512
2.1ØØØØØØ	1.ØØØØØØØ	.9Ø88711	.9Ø849Ø2	.9Ø91483	.9Ø87174
2.2ØØØØØØ	1.ØØØØØØØ	.9Ø89388	.9Ø86751	.9Ø913Ø6	.9Ø88324
2.3ØØØØØØ	1.ØØØØØØØ	.9Ø89856	.9Ø88Ø31	.9Ø91184	.9Ø8912Ø
2.4ØØØØØØ	1.ØØØØØØØ	.9Ø9Ø18Ø	.9Ø88917	.9Ø91Ø99	.9Ø89671
2.5ØØØØØØ	1.ØØØØØØØ	.9Ø9Ø4Ø5	.9Ø89531	.9Ø91Ø41	.9Ø9ØØ52
2.6ØØØØØØ	1.ØØØØØØØ	.9Ø9Ø56Ø	.9Ø89955	.9Ø91ØØØ	.9Ø9Ø316
2.7ØØØØØØ	1.ØØØØØØØ	.9Ø9Ø668	.9Ø9Ø249	.9Ø9Ø972	.9Ø9Ø499
2.8ØØØØØØ	1.ØØØØØØØ	.9Ø9Ø742	.9Ø9Ø452	.9Ø9Ø953	.9Ø9Ø625
2.9ØØØØØØ	1.ØØØØØØØ	.9Ø9Ø793	.9Ø9Ø593	.9Ø9Ø939	.9Ø9Ø712
3.ØØØØØØØ	1.ØØØØØØØ	.9Ø9Ø829	.9Ø9Ø69Ø	.9Ø9Ø93Ø	.9Ø9Ø773

No Page 111

3.1000000	1.0000000	.9090854	.9090758	.9090924	.9090815
3.2000000	1.0000000	.9090871	.9090804	.9090919	.9090844
3.3000000	1.0000000	.9090883	.9090837	.9090916	.9090864
3.4000000	1.0000000	.9090891	.9090859	.9090914	.9090878
3.5000000	1.0000000	.9090896	.9090874	.9090912	.9090887
3.6000000	1.0000000	.9090900	.9090885	.9090911	.9090894
3.7000000	1.0000000	.9090903	.9090892	.9090911	.9090899
3.8000000	1.0000000	.9090905	.9090898	.9090910	.9090902
3.9000000	1.0000000	.9090906	.9090901	.9090910	.9090904
4.0000000	1.0000000	.9090907	.9090904	.9090910	.9090906
4.1000000	1.0000000	.9090908	.9090905	.9090909	.9090907
4.2000000	1.0000000	.9090908	.9090906	.9090909	.9090907
4.3000000	1.0000000	.9090908	.9090907	.9090909	.9090908
4.4000000	1.0000000	.9090909	.9090908	.9090909	.9090908
4.5000000	1.0000000	.9090909	.9090908	.9090909	.9090909
4.6000000	1.0000000	.9090909	.9090908	.9090909	.9090909
4.7000000	1.0000000	.9090909	.9090909	.9090909	.9090909
4.8000000	1.0000000	.9090909	.9090909	.9090909	.9090909
4.9000000	1.0000000	.9090909	.9090909	.9090909	.9090909
5.0000000	1.0000000	.9090909	.9090909	.9090909	.9090909

APPENDIX D

Main Program Listing and Results for the Simulation of the PI Controller

```
      PROGRAM CONTROL
      COMMON/VAL/ X(10),F(10),KSTEP
      DIMENSION E(10)
      N=7
      H=0.1666 $ TC=0.1
      VM=1.377 $ X(8)=0.4
      A=0.1 $ B=2.5
      Y1=0.00086
      X(1)=0.0008072
      X(3)=0.40
      X(4)=0.40
      X(5)=0.0006255
      X(6)=0.0
      X(7)=0.0
      X(2)=0.0
      ZR=0.0 $ RAN=1.0
      SP=0.0008072
      READ (60,100) HPRINT,HMIN,TMAX,ERR
      READ(60,100) Y2,Y3,PD,TI
      WRITE(61,300)
  300 FORMAT(5X,"T",9X,"X(1)",7X,"X(5)",7X,"X(3)",7X,"X(4)",///)
  100 FORMAT(4F8.6)
   10 XO=STEP(T,A,B,Y1,Y2,Y3)
      CALL DSIM(N,T,HPRINT,0.0,0.0,HMIN,TMAX,ERR,
     $ E,IERR,ITASK)
      GO TO(1,2,3,4,5) ITASK
    1 GO TO 10
    2 ERROR=(X(1)-0.0008072)*100.0
      CALL PICONTR(1,7,ZR,RAN,SP,1.0,PD,TI)
      CALL TRFN(7,6,0.08333,0.705,1)
      F(1)=1./H*(X(3)*X(5)-X(4)*X(1))+VM/H*(XO-X(1))
      F(5)=VM/H*(X(1)-X(5))-1./H*X(3)*X(5)
      F(3)=(X(8)+X(6))/TC-X(3)/TC
      F(4)=X(3)/TC-X(4)/TC
      GO TO 10
    3 WRITE(61,200) T,X(1),X(5),X(3),X(4)
  200 FORMAT(5F11.7)
    4 GO TO 10
    5 CONTINUE
      END
```

0.1	0.001	7.5	0.0001
0.00096	0.00096	4.0	6.0

T	X(1)	X(5)	X(3)	X(4)
0	.0008072	.0006255	.4000000	.4000000
.1000000	.0008072	.0006255	.4000029	.4000007
.2000000	.0008603	.0006419	.4157363	.4036305
.3000000	.0008859	.0006550	.4751843	.4311354
.4000000	.0008985	.0006496	.5645243	.4907813
.5000000	.0008980	.0006279	.6663919	.5746580
.6000000	.0008844	.0005959	.7647844	.6694311
.7000000	.0008608	.0005600	.8453763	.7606015
.8000000	.0008327	.0005265	.8979887	.8351637
.9000000	.0008058	.0004997	.9187115	.8841407
1.0000000	.0007945	.0004819	.9100548	.9041717
1.1000000	.0007711	.0004735	.8791804	.8974439
1.2000000	.0007656	.0004736	.8353264	.8701960
1.3000000	.0007670	.0004805	.7875463	.8305578
1.4000000	.0007735	.0004921	.7432847	.7867421
1.5000000	.0007832	.0005059	.7077608	.7456512
1.6000000	.0007941	.0005198	.6838998	.7122765
1.7000000	.0008047	.0005319	.6725630	.6895278
1.8000000	.0008138	.0005409	.6729094	.6783728
1.9000000	.0008204	.0005460	.6828115	.6781406
2.0000000	.0008241	.0005471	.6992978	.6869054
2.1000000	.0008248	.0005448	.7190157	.7019157
2.2000000	.0008230	.0005401	.7386993	.7200483
2.3000000	.0008192	.0005339	.7555970	.7382590
2.4000000	.0008144	.0005276	.7677965	.7539842
2.5000000	.0008094	.0005220	.7743880	.7654375
2.6000000	.0008050	.0005177	.7754410	.7717528
2.7000000	.0008018	.0005151	.7718221	.7729614
2.8000000	.0007999	.0005143	.7649124	.7698279
2.9000000	.0007994	.0005149	.7562994	.7635979
3.0000000	.0008001	.0005168	.7475037	.7557213
3.1000000	.0008016	.0005193	.7397775	.7476019
3.2000000	.0008036	.0005221	.7339854	.7404091
3.3000000	.0008057	.0005246	.7305650	.7349629
3.4000000	.0008077	.0005266	.7295502	.7316902
3.5000000	.0008092	.0005279	.7306425	.7306427

3.6000000	.0008102	.0005285	.7333105	.7315593
3.7000000	.0008106	.0005284	.7369023	.7339579
3.8000000	.0008104	.0005277	.7407544	.7372395
3.9000000	.0008099	.0005266	.7442849	.7407900
4.0000000	.0008090	.0005254	.7470601	.7440678
4.1000000	.0008081	.0005243	.7488294	.7466657
4.2000000	.0008072	.0005233	.7495284	.7483448
4.3000000	.0008065	.0005227	.7492545	.7490377
4.4000000	.0008060	.0005223	.7482237	.7488276
4.5000000	.0008057	.0005223	.7467182	.7479086
4.6000000	.0008058	.0005226	.7450342	.7465383
4.7000000	.0008060	.0005230	.7434384	.7449897
4.8000000	.0008063	.0005235	.7421359	.7435105
4.9000000	.0008067	.0005240	.7412536	.7422933
5.0000000	.0008071	.0005245	.7408371	.7414588
5.1000000	.0008075	.0005248	.7408590	.7410529
5.2000000	.0008077	.0005250	.7412366	.7410538
5.3000000	.0008078	.0005250	.7418528	.7413871
5.4000000	.0008078	.0005249	.7425787	.7419462
5.5000000	.0008078	.0005248	.7432937	.7426125
5.6000000	.0008076	.0005245	.7439003	.7432739
5.7000000	.0008075	.0005243	.7443341	.7438395
5.8000000	.0008073	.0005241	.7445661	.7442481
5.9000000	.0008071	.0005240	.7446010	.7444716
6.0000000	.0008070	.0005239	.7444697	.7445125
6.1000000	.0008069	.0005238	.7442208	.7443985
6.2000000	.0008069	.0005239	.7439104	.7441736
6.3000000	.0008069	.0005239	.7435927	.7438894
6.4000000	.0008070	.0005240	.7433133	.7435961
6.5000000	.0008071	.0005241	.7431038	.7433360
6.6000000	.0008071	.0005242	.7429807	.7431393
6.7000000	.0008072	.0005243	.7429453	.7430217
6.8000000	.0008073	.0005243	.7429865	.7429850
6.9000000	.0008073	.0005243	.7430846	.7430197
7.0000000	.0008073	.0005243	.7432157	.7431079
7.1000000	.0008073	.0005243	.7433555	.7432276
7.2000000	.0008073	.0005243	.7434831	.7433565
7.3000000	.0008073	.0005242	.7435829	.7434750
7.4000000	.0008072	.0005242	.7436462	.7435685
7.5000000	.0008072	.0005242	.7436708	.7436286

Ø.1 Ø.ØØ1 7.5 Ø.ØØØ1

Ø.ØØØ96 Ø.ØØØ96 4.Ø 7.8

T	X(1)	X(5)	X(3)	X(4)
Ø	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
.1ØØØØØØ	.ØØØ8Ø72	.ØØØ6255	.4ØØØØ31	.4ØØØØØ7
.2ØØØØØØ	.ØØØ86Ø3	.ØØØ6418	.4164448	.4Ø37634
.3ØØØØØØ	.ØØØ8864	.ØØØ6542	.481796Ø	.4334Ø89
.4ØØØØØØ	.ØØØ8997	.ØØØ6464	.5854425	.5ØØ6Ø85
.5ØØØØØØ	.ØØØ8988	.ØØØ62Ø1	.7Ø93812	.5992899
.6ØØØØØØ	.ØØØ8824	.ØØØ5815	.8336Ø71	.7151355
.7ØØØØØØ	.ØØØ8531	.ØØØ5384	.9371496	.8297137
.8ØØØØØØ	.ØØØ8176	.ØØØ4985	1.ØØ25738	.924ØØ86
.9ØØØØØØ	.ØØØ7839	.ØØØ4676	1.Ø2Ø8374	.9828948
1.ØØØØØØØ	.ØØØ7583	.ØØØ449Ø	.9932723	.9989114
1.1ØØØØØØ	.ØØØ7439	.ØØØ4435	.9298534	.9735Ø45
1.2ØØØØØØ	.ØØØ74Ø9	.ØØØ45Ø1	.8453112	.91548Ø1
1.3ØØØØØØ	.ØØØ7477	.ØØØ4667	.7552186	.8378761
1.4ØØØØØØ	.ØØØ7619	.ØØØ49Ø4	.673246Ø	.7547377
1.5ØØØØØØ	.ØØØ78Ø5	.ØØØ5177	.6Ø9726Ø	.67869Ø2
1.6ØØØØØØ	.ØØØ8ØØ8	.ØØØ5445	.57117Ø7	.619511Ø
1.7ØØØØØØ	.ØØØ82Ø5	.ØØØ5669	.56Ø33ØØ	.583518Ø
1.8ØØØØØØ	.ØØØ8372	.ØØØ5816	.5764743	.5734768
1.9ØØØØØØ	.ØØØ8489	.ØØØ5866	.6157144	.5887651
2.ØØØØØØØ	.ØØØ854Ø	.ØØØ5818	.6713551	.6256411
2.1ØØØØØØ	.ØØØ8516	.ØØØ5685	.7344796	.6776376
2.2ØØØØØØ	.ØØØ8421	.ØØØ5496	.795Ø362	.7362485
2.3ØØØØØØ	.ØØØ8275	.ØØØ5287	.8434921	.792Ø847
2.4ØØØØØØ	.ØØØ81Ø7	.ØØØ5Ø93	.8727Ø48	.8364538
2.5ØØØØØØ	.ØØØ7948	.ØØØ4942	.8793531	.862998Ø
2.6ØØØØØØ	.ØØØ7826	.ØØØ4851	.86437ØØ	.8688524
2.7ØØØØØØ	.ØØØ7755	.ØØØ4827	.8322938	.8549367
2.8ØØØØØØ	.ØØØ7739	.ØØØ4866	.78991Ø5	.8253648
2.9ØØØØØØ	.ØØØ7774	.ØØØ4957	.744727Ø	.78628Ø2
3.ØØØØØØØ	.ØØØ7848	.ØØØ5Ø84	.7Ø36797	.7445352
3.1ØØØØØØ	.ØØØ7946	.ØØØ5225	.67225Ø1	.7Ø65344
3.2ØØØØØØ	.ØØØ8Ø54	.ØØØ5359	.653981Ø	.6774Ø13
3.3ØØØØØØ	.ØØØ8155	.ØØØ5466	.65Ø3Ø6Ø	.66Ø4866
3.4ØØØØØØ	.ØØØ8238	.ØØØ5533	.66Ø5996	.6571643
3.5ØØØØØØ	.ØØØ8292	.ØØØ555Ø	.6823911	.66685Ø5

3.6000000	.0008309	.0005519	.7117367	.6872029
3.7000000	.0008288	.0005449	.7437693	.7144937
3.8000000	.0008236	.0005353	.7734240	.7441608
3.9000000	.0008161	.0005249	.7962568	.7715102
4.0000000	.0008078	.0005154	.8091887	.7924753
4.1000000	.0008002	.0005081	.8109803	.8042757
4.2000000	.0007944	.0005038	.8023131	.8058188
4.3000000	.0007910	.0005030	.7854839	.7977519
4.4000000	.0007905	.0005054	.7638329	.7821829
4.5000000	.0007924	.0005104	.7410779	.7621716
4.6000000	.0007963	.0005171	.7207008	.7411355
4.7000000	.0008015	.0005243	.7054751	.7222924
4.8000000	.0008071	.0005310	.6971677	.7082194
4.9000000	.0008122	.0005361	.6964079	.7005657
5.0000000	.0008162	.0005391	.7027056	.6999195
5.1000000	.0008186	.0005395	.7145969	.7058184
5.2000000	.0008191	.0005377	.7299022	.7168877
5.3000000	.0008178	.0005339	.7460729	.7310881
5.4000000	.0008149	.0005290	.7605906	.7460486
5.5000000	.0008110	.0005238	.7713587	.7594411
5.6000000	.0008069	.0005191	.7770166	.7693418
5.7000000	.0008032	.0005156	.7771113	.7745122
5.8000000	.0008004	.0005137	.7720997	.7745497
5.9000000	.0007989	.0005135	.7631909	.7698834
6.0000000	.0007988	.0005149	.7520794	.7616296
6.1000000	.0007999	.0005176	.7406312	.7513495
6.2000000	.0008020	.0005211	.7305847	.7407646
6.3000000	.0008047	.0005248	.7233100	.7314819
6.4000000	.0008076	.0005281	.7196482	.7247696
6.5000000	.0008101	.0005305	.7198424	.7214078
6.6000000	.0008120	.0005318	.7235526	.7216192
6.7000000	.0008131	.0005318	.7299491	.7250842
6.8000000	.0008132	.0005307	.7378654	.7310276
6.9000000	.0008123	.0005287	.7459925	.7383637
7.0000000	.0008108	.0005262	.7530850	.7458786
7.1000000	.0008088	.0005236	.7581485	.7524222
7.2000000	.0008068	.0005213	.7605763	.7570801
7.3000000	.0008050	.0005196	.7602161	.7592983
7.4000000	.0008036	.0005188	.7573561	.7589414
7.5000000	.0008030	.0005188	.7526414	.7562791

Ø.1 Ø.ØØ1 8.Ø Ø.ØØØ1

Ø.ØØØ96 Ø.ØØØ96 5.71 6.Ø

T	X(1)	X(5)	X(3)	X(4)
Ø	.ØØØ8Ø72	.ØØØ6255	.4ØØØØØØ	.4ØØØØØØ
.1ØØØØØØ	.ØØØ8Ø72	.ØØØ6255	.4ØØØØ2Ø	.4ØØØØØ5
.2ØØØØØØ	.ØØØ86ØØ	.ØØØ6422	.411ØØ81	.4Ø25411
.3ØØØØØØ	.ØØØ8843	.ØØØ658Ø	.4522839	.4217Ø27
.4ØØØØØØ	.ØØØ8962	.ØØØ6587	.5135825	.4628855
.5ØØØØØØ	.ØØØ898Ø	.ØØØ646Ø	.5834Ø56	.52Ø433Ø
.6ØØØØØØ	.ØØØ89Ø6	.ØØØ6244	.6528836	.586Ø22Ø
.7ØØØØØØ	.ØØØ8763	.ØØØ5983	.7149912	.6516715
.8ØØØØØØ	.ØØØ858Ø	.ØØØ5717	.7647578	.71Ø7368
.9ØØØØØØ	.ØØØ8387	.ØØØ5475	.7996367	.7585126
1.ØØØØØØØ	.ØØØ8211	.ØØØ5278	.8194714	.7925523
1.1ØØØØØØ	.ØØØ8Ø68	.ØØØ5133	.826ØØ62	.8125911
1.2ØØØØØØ	.ØØØ7965	.ØØØ5Ø39	.822144Ø	.82Ø1ØØ9
1.3ØØØØØØ	.ØØØ79Ø2	.ØØØ4992	.8112Ø43	.8176464
1.4ØØØØØØ	.ØØØ7875	.ØØØ4984	.7963516	.8Ø82359
1.5ØØØØØØ	.ØØØ7875	.ØØØ5ØØ3	.78Ø248Ø	.7948Ø4Ø
1.6ØØØØØØ	.ØØØ7896	.ØØØ5Ø41	.7649Ø16	.779876Ø
1.7ØØØØØØ	.ØØØ7928	.ØØØ5Ø89	.7516533	.7654Ø28
1.8ØØØØØØ	.ØØØ7966	.ØØØ5139	.7412451	.7527245
1.9ØØØØØØ	.ØØØ8ØØ4	.ØØØ5186	.7339269	.7426161
2.ØØØØØØØ	.ØØØ8Ø38	.ØØØ5226	.7295764	.7353775
2.1ØØØØØØ	.ØØØ8Ø66	.ØØØ5257	.7278154	.73Ø9417
2.2ØØØØØØ	.ØØØ8Ø88	.ØØØ5278	.7281159	.7289848
2.3ØØØØØØ	.ØØØ81Ø2	.ØØØ5289	.7298897	.729Ø275
2.4ØØØØØØ	.ØØØ81Ø9	.ØØØ5293	.73256Ø7	.73Ø5232
2.5ØØØØØØ	.ØØØ811Ø	.ØØØ529Ø	.7356172	.7329277
2.6ØØØØØØ	.ØØØ81Ø8	.ØØØ5283	.7384449	.7357523
2.7ØØØØØØ	.ØØØ81Ø2	.ØØØ5274	.7413437	.7385967

2.8000000	.0008095	.0005264	.7435268	.7411661
2.9000000	.0008087	.0005255	.7451109	.7432729
3.0000000	.0008080	.0005246	.7460962	.7448270
3.1000000	.0008074	.0005240	.7465445	.7458196
3.2000000	.0008070	.0005235	.7465557	.7463015
3.3000000	.0008066	.0005233	.7462476	.7463622
3.4000000	.0008065	.0005232	.7457385	.7461099
3.5000000	.0008064	.0005232	.7451349	.7456554
3.6000000	.0008065	.0005233	.7445241	.7451001
3.7000000	.0008066	.0005235	.7439701	.7445280
3.8000000	.0008067	.0005237	.7435139	.7440017
3.9000000	.0008069	.0005239	.7431753	.7435625
4.0000000	.0008070	.0005241	.7429568	.7432313
4.1000000	.0008071	.0005242	.7428479	.7430125
4.2000000	.0008072	.0005243	.7428299	.7428977
4.3000000	.0008073	.0005244	.7428800	.7428704
4.4000000	.0008073	.0005244	.7429745	.7429095
4.5000000	.0008074	.0005244	.7430918	.7429931
4.6000000	.0008073	.0005244	.7432138	.7431006
4.7000000	.0008073	.0005243	.7433268	.7432145
4.8000000	.0008073	.0005243	.7434217	.7433215
4.9000000	.0008073	.0005243	.7434937	.7434125

0.1 0.001 6.0 0.0001

0.00096 0.00096 4.0 4.29

T	X(1)	X(5)	X(3)	X(4)
0	.0008072	.0006255	.4000000	.4000000
.1000000	.0008072	.0006255	.4000028	.4000007
.2000000	.0008602	.0006419	.4150632	.4035044
.3000000	.0008855	.0006557	.4689123	.4289775
.4000000	.0008974	.0006526	.5447481	.4814765
.5000000	.0008971	.0006354	.6259093	.5514172
.6000000	.0008859	.0006097	.7000161	.6264051
.7000000	.0008673	.0005813	.7582955	.6953596
.8000000	.0008458	.0005547	.7962788	.7502075
.9000000	.0008257	.0005330	.8139606	.7868194
1.0000000	.0008095	.0005177	.8148196	.8050305
1.1000000	.0007986	.0005088	.8041583	.8077412
1.2000000	.0007928	.0005053	.7874910	.7995203
1.3000000	.0007913	.0005059	.7694476	.7852488
1.4000000	.0007927	.0005092	.7532821	.7691543
1.5000000	.0007959	.0005137	.7408377	.7543230
1.6000000	.0007998	.0005184	.7327687	.7426026
1.7000000	.0008036	.0005226	.7288621	.7347501
1.8000000	.0008067	.0005256	.7283636	.7306940
1.9000000	.0008089	.0005275	.7302637	.7298240
2.0000000	.0008102	.0005284	.7335222	.7312565
2.1000000	.0008106	.0005283	.7372266	.7340493
2.2000000	.0008104	.0005276	.7406842	.7373567
2.3000000	.0008098	.0005267	.7434563	.7405228
2.4000000	.0008090	.0005256	.7453446	.7431192
2.5000000	.0008082	.0005247	.7463461	.7449378
2.6000000	.0008075	.0005240	.7465927	.7459528
2.7000000	.0008069	.0005236	.7462880	.7462678
2.8000000	.0008066	.0005234	.7456540	.7460587
2.9000000	.0008065	.0005233	.7448913	.7455243
3.0000000	.0008065	.0005234	.7441556	.7448488
3.1000000	.0008066	.0005236	.7435488	.7441789
3.2000000	.0008068	.0005238	.7431208	.7436134
3.3000000	.0008070	.0005240	.7428789	.7432038
3.4000000	.0008071	.0005242	.7427998	.7429618
3.5000000	.0008072	.0005243	.7428436	.7428704

3.6000000	.0008073	.0005244	.7429645	.7428951
3.7000000	.0008073	.0005244	.7431197	.7429953
3.8000000	.0008073	.0005243	.7432751	.7431319
3.9000000	.0008073	.0005243	.7434073	.7432727
4.0000000	.0008073	.0005243	.7435039	.7433953
4.1000000	.0008073	.0005242	.7435617	.7434872
4.2000000	.0008072	.0005242	.7435843	.7435442
4.3000000	.0008072	.0005242	.7435797	.7435691
4.4000000	.0008072	.0005242	.7435572	.7435682
4.5000000	.0008072	.0005242	.7435259	.7435500
4.6000000	.0008072	.0005242	.7434933	.7435227
4.7000000	.0008072	.0005242	.7434647	.7434933
4.8000000	.0008072	.0005242	.7434431	.7434668
4.9000000	.0008072	.0005242	.7434295	.7434464
5.0000000	.0008072	.0005242	.7434234	.7434331
5.1000000	.0008072	.0005242	.7434234	.7434267
5.2000000	.0008072	.0005242	.7434275	.7434260
5.3000000	.0008072	.0005242	.7434337	.7434291
5.4000000	.0008072	.0005242	.7434405	.7434345
5.5000000	.0008072	.0005242	.7434467	.7434406
5.6000000	.0008072	.0005242	.7434515	.7434463
5.7000000	.0008072	.0005242	.7434546	.7434508
5.8000000	.0008072	.0005242	.7434562	.7434539
5.9000000	.0008072	.0005242	.7434564	.7434555
6.0000000	.0008072	.0005242	.7434557	.7434558

APPENDIX E-1

Position Algorithm in DDC Control

A PID Controller can be presented as (13):

$$P_n = K_c \left(e + T_D \frac{\Delta e}{\Delta t} + \frac{1}{T_I} \sum_0^n e(\Delta t)\right) + P_m$$

$$e = S - V$$

where

P_n = valve position at time n

P_m = initial valve position

K_c = proportional gain

T_D = derivative time

T_I = integral or reset time

Δ = change or difference

e = error

S = set point

V = variable

The derivative expression is usually calculated with a four-point difference technique as:

$$V = (V_n + V_{n-1} + V_{n-2} + V_{n-3})/4$$

where V is the variable

$$\frac{\Delta V}{\Delta t} = \left(\frac{V_n - V}{1.5 \Delta t} + \frac{V_{n-1} - V}{0.5 \ \Delta t} + \frac{V - V_{n-2}}{0.5 \ \Delta t} + \frac{V - V_{n-3}}{1.5 \ \Delta t} \right) \Big/ 4$$

or

$$\frac{\Delta V}{\Delta t} = \frac{1}{6 \Delta t} \quad (V_n - V + 3V_{n-1} - 3V + 3V - 3V_{n-2} + V - V_{n-3})$$

$$\frac{V}{\Delta t} = \frac{1}{6 \Delta t} \ (V_n - V_{n-3} + 3V_{n-1} - 3V_{n-2})$$

if $\frac{\Delta V}{\Delta t} = \frac{\Delta e}{\Delta t}$;

$$\frac{\Delta e}{\Delta t} = \frac{1}{6 \Delta t} (e_n - e_{n-3} + 3e_{n-1} - 3e_{n-2})$$

The term $\sum_{e}^{n} e \Delta t$ can be obtained by using the expressions:

$$SUME = SUME + ERROR$$

$$\sum e \Delta t = SUME \ (\Delta t)$$

where Δt is the sampling time.

Replacing the values in the control equation:

$$P_n = K_c \ (e + \frac{T_D}{6 \Delta t} (e_n - e_{n-3} + 3e_{n-1} - 3e_{n-2}) + \frac{1}{TR} \ SUME \ (\Delta t)$$

APPENDIX E-2

Main Program Listing for

DDC Control with NOVA

Main Program Listing

for Simulating Sampling Time Effect on DDC Control

```
C     INITIAL VALUES
C
      SUME=0.
      ERRV=0.
      SVAL=0.
C     SET POINT
C
1     CALL AIRDW(1,1,2,IVAL,IERR)
      SSV=FLOAT(IVAL)*0.005
      TYPE 'SSV= ',SSV
      IF(ABS(SSV-SVAL).LT.0.0001) GO TO 2
      SVAL=SSV
      CALL WAIT(5,2,IERR)
      GO TO 1
C
2     CONTINUE
      SETP=SVAL
      TYPE 'SET POINT= ',SETP
      ACCEPT 'GAIN= ',GAIN
      ACCEPT 'RT= ',RT
      DELT=0.01
      CONTINUE
C     CONTROL ACTION
C
3     CALL AIRDW(1,1,2,IVAL,IERR)
      SSV=FLOAT(IVAL)*0.005
      ERRV=SSV-SETP
      SUME=SUME+ERRV
      PM=GAIN*(ERRV+(SUME/RT)*DELT)
      INEW=IFIX(PM/0.005)
      CONTINUE
      CALL AOW(1,11,INEW,IERR)
      CALL WAIT(10,1,IERR)
      GO TO 3
      END
```

```
C     INITIAL VALUES
C
      SUME=0.
      ERRV=0.
      SVAL=0.
C     SET POINT
C
1     CALL AIRDW(1,1,2,IVAL,IERR)
      SSV=FLOAT(IVAL)*0.005
      TYPE 'SSV= ',SSV
      IF(ABS(SSV-SVAL).LT.0.0001) GO TO 2
      SVAL=SSV
      CALL WAIT(5,2,IERR)
      GO TO 1
C
2     CONTINUE
      SETP=SVAL
      TYPE 'SET POINT= ',SETP
      ACCEPT 'GAIN= ',GAIN
      ACCEPT 'RT= ',RT
      DELT=10.0
      CONTINUE
C     CONTROL ACTION
C
3     CALL AIRDW(1,1,2,IVAL,IERR)
      SSV=FLOAT(IVAL)*0.005
      ERRV=SSV-SETP
      SUME=SUME+ERRV
      PM=GAIN*(ERRV+(SUME/RT)*DELT)
      INEW=IFIX(PM/0.005)
      CONTINUE
      CALL AOW(1,11,INEW,IERR)
      CALL WAIT(10,2,IERR)
      GO TO 3
      END
```

```
      PROGRAM CONTROL
      COMMON/VAL/ X(10),F(10),KSTEP
      COMMON/E/SUME(10)
      DIMENSION E(10)
C     INITIAL VALUES AND CONSTANTS
C
      N=7
      H=0.1666 $ TC=0.1
      VM=1.377 $ X(8)=0.4
      A=0.1 $ B=2.5
      DO 11 I=1,10
   11 SUME(I)=0.0
      Y1=0.00086
      X(1)=0.0008072
      X(3)=0.40
      X(4)=0.40
      X(5)=0.0006255
      X(6)=0.0
      X(7)=0.0
      X(2)=0.0
      ZR=0.0 $ RAN=1.0
      SP=0.0008072
      READ (60,100) HPRINT,HMIN,TMAX,ERR
      READ(60,100) Y2,Y3,PD,TI
      WRITE(61,300)
  300 FORMAT(5X,"T",9X,"X(1)",7X,"X(5)",7X,"X(3)",7X,"X(4)",///)
  100 FORMAT(4F8.6)
   10 X0=STEP(T,A,B,Y1,Y2,Y3)
      CALL DSIM(N,T,HPRINT,0.1666,0.0,HMIN,TMAX,ERR,
     $ E,IERR,ITASK)
      GO TO(1,2,3,4,5) ITASK
    1 GO TO 10
C
C     DERIVATIVE SECTION
C
    2 F(1)=1./H*(X(3)*X(5)-X(4)*X(1))+VM/H*(X0-X(1))
      F(5)=VM/H*(X(1)-X(5))-1./H*X(3)*X(5)
      CALL TRFN(7,6,0.08333,0.705,1)
      F(3)=(X(8)+X(6))/TC-X(3)/TC
      F(4)=X(3)/TC-X(4)/TC
      GO TO 10
C     PRINTING SECTION
```

```
  3 WRITE(61,200) T,X(1),X(5),X(3),X(4)
200 FORMAT(5F11.7)
    GO TO 10
  4 CALL DPICON(1,7,ZR,RAN,SP,1.0,PD,TI,0.1666)
    GO TO 10
  5 CONTINUE
    END
```

APPENDIX F-1

Model for Feedforward Control

If we consider the system shown in Fig. 27, the output variable can be represented as:

$$C(s) = G_1(s)\ M(s) - G_2(s)\ U(s)$$

The object will be to keep $C(s) = R(s)$;

$$R(s) = G_1(s)\ M(s) - G_2(s)\ U(s)$$

then, the manipulative variable is:

$$M = \frac{R + G_2(s)\ U}{G_1(s)}$$

Since a transfer function can be broken down into a static and a dynamic part such as:

$$G(s) = SD(s)$$

the equation for M can be written in two different forms:

$M = \dfrac{R + S_2 U}{S_1}$ where the dynamic part is neglected. This control is called a steady-state feedforward control,

and,

$$M = \frac{R + S_2 D_2(s)\ U}{S_1\ D_1(s)}$$

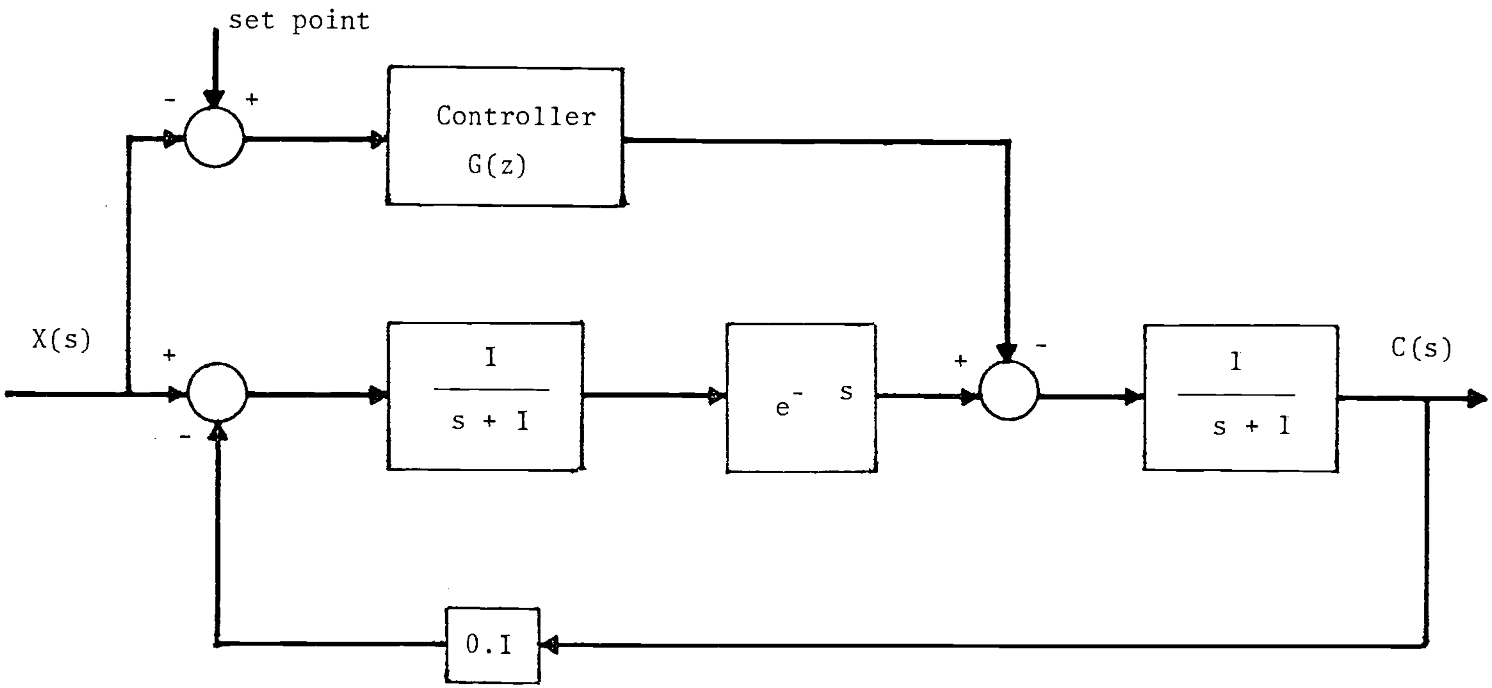


Figure 27. Block diagram of the fifth-order system with feedforward control

A practical version of this equation can be obtained by grouping the dynamic elements as a first approximation:

$$M \approx \left(\frac{R + \frac{S_2}{S_1} U}{}\right)\left(\frac{D_2(s)}{D_1(s)}\right)$$

This control system is called the dynamic feedforward control. The values of the dynamic elements are usually determined from the transient response data. The first part is the study of the change of the controlled variable resulting from a change in U. The second part is the study of the change of the controlled variable resulting from a change in M sufficient to compensate for the change in U. These data are usually presented as plots of the transient response of the variable. A first-order transfer function with dead-time is generally used to fit these curves. If this is done

$$\frac{D_2(s)}{D_1(s)} = \frac{e^{-t_{02}s}}{1 + \tau_2 s} \Big/ \frac{e^{-t_{01}s}}{1 + \tau_1 s}$$

$$= \frac{e^{-t_0 s}(1 + \tau_1 s)}{(1 + \tau_2 s)}$$

where $t_o = t_{o2} - t_{o1}$

APPENDIX F-2

a) Program Listing for Feedforward Control of the Gas Absorber with NOVA

b) Main Program Listing and the Results of the Simulation of Feedforward Control

```
      SVAL=0.0
1     CALL AIRDW(1,1,2,IVAL,IERR)
      SSV=FLOAT(IVAL)*0.005
      TYPE "SSV =",SSV
      IF(ABS(SSV-SVAL).LT.0.0001) GO TO 2
      SVAL=SSV
      CALL WAIT(1,3,IERR)
      GO TO 1
2     CONTINUE
      VM=1.377
      H=0.1666
      TC=0.1
      X1=SVAL/5000.
      SETP=SVAL
      TYPE "SET POINT =",SETP
3     CALL AIRDW(1,1,3,IVAL2,IERR)
      STEP=FLOAT(IVAL2)*0.005
      XO=STEP/5000.
      FLOW=(VM*(XO-X1)+SQRT(VM**2*(XO-X1)*(XO-X1+4.*X1)))/
     1(2.*X1)
      XFLOW=-1.*(FLOW*10.-4.0)
      INEW=IFIX(XFLOW/0.005)
      CALL AOW(1,11,INEW,IERR)
      CALL WAIT(500,1,IERR)
      GO TO 3
      END
```

```
      PROGRAM CONTROL
      COMMON/VAL/ X(10),F(10),KSTEP
      DIMENSION E(10)
      N=5
      H=0.1666 $ TC=0.1
      VM=1.377 $ X(8)=0.4
      A=0.5 $ B=3.0
      Y1=0.0008600 $ Y2=0.000960 $ Y3=0.0008600
      X0=Y1
      X(1)=0.0008079
      X(3)=0.40
      X(4)=0.40
      X(5)=0.0006261
      XX1=X(1)
      READ (60,100) HPRINT,HMIN,TMAX,ERR
  100 FORMAT(4F8.6)
   10 X0=STEP(T,A,B,Y1,Y2,Y3)
      IF(X0.EQ.Y1) GO TO 11
      X(8)=(VM*(X0-XX1)+SQRT(VM**2*(X0-XX1)*(X0-XX1+4.*XX1)))
     $/(2.*XX1)
   11 CONTINUE
      CALL DSIM(N,T,HPRINT,0.0,0.0,HMIN,TMAX,ERR,
     $ E,IERR,ITASK)
      GO TO(1,2,3,4,5) ITASK
    1 GO TO 10
    2 F(1)=1./H*(X(3)*X(5)-X(4)*X(1))+VM/H*(X0-X(1))
      F(5)=VM/H*(X(1)-X(5))-1./H*X(3)*X(5)
      F(3)=X(8)/TC-X(3)/TC
      F(4)=X(3)/TC-X(4)/TC
      GO TO 10
    3 WRITE(61,200) T,X(1),X(5),X(3),X(4)
  200 FORMAT(5F10.7)
    4 GO TO 10
    5 CONTINUE
      END
```

0.05 0.001 2.5 0.00001

0	.0008079	.0006261	.4000000	.4000000
.0500000	.0008077	.0006260	.4000000	.4000000
.1000000	.0008075	.0006259	.4000000	.4000000
.1500000	.0008074	.0006258	.4000000	.4000000
.2000000	.0008074	.0006258	.4000000	.4000000
.2500000	.0008073	.0006257	.4000000	.4000000
.3000000	.0008073	.0006257	.4000000	.4000000
.3500000	.0008073	.0006256	.4000000	.4000000
.4000000	.0008073	.0006256	.4000000	.4000000
.4500000	.0008072	.0006256	.4000000	.4000000
.5000000	.0008072	.0006256	.4000000	.4000000
.5500000	.0008480	.0006214	.5341709	.4307595
.6000000	.0008761	.0006154	.6155498	.4901053
.6500000	.0008878	.0006093	.6649085	.5507796
.7000000	.0008873	.0006022	.6948462	.6025492
.7500000	.0008797	.0005938	.7130043	.6430281
.8000000	.0008692	.0005846	.7240177	.6730864
.8500000	.0008582	.0005752	.7306977	.6946578
.9000000	.0008481	.0005664	.7347494	.7097672
.9500000	.0008393	.0005584	.7372068	.7201603
1.0000000	.0008321	.0005515	.7386973	.7272093
1.0500000	.0008263	.0005458	.7396013	.7319367
1.1000000	.0008218	.0005411	.7401497	.7350782
1.1500000	.0008183	.0005373	.7404823	.7371499
1.2000000	.0008156	.0005344	.7406840	.7385073
1.2500000	.0008136	.0005321	.7408063	.7393918
1.3000000	.0008121	.0005304	.7408805	.7399654
1.3500000	.0008110	.0005291	.7409255	.7403358
1.4000000	.0008102	.0005281	.7409528	.7405741
1.4500000	.0008095	.0005273	.7409694	.7407269
1.5000000	.0008091	.0005268	.7409794	.7408246

1.5500000	.0008088	.0005264	.7409855	.7408869
1.6000000	.0008085	.0005261	.7409892	.7409266
1.6500000	.0008084	.0005258	.7409915	.7409517
1.7000000	.0008082	.0005257	.7409928	.7409677
1.7500000	.0008081	.0005256	.7409936	.7409778
1.8000000	.0008081	.0005255	.7409941	.7409841
1.8500000	.0008080	.0005254	.7409945	.7409881
1.9000000	.0008080	.0005254	.7409946	.7409907
1.9500000	.0008080	.0005253	.7409947	.7409923
2.0000000	.0008079	.0005253	.7409948	.7409932
2.0500000	.0008079	.0005253	.7409949	.7409939
2.1000000	.0008079	.0005253	.7409949	.7409943
2.1500000	.0008079	.0005253	.7409949	.7409945
2.2000000	.0008079	.0005253	.7409949	.7409947
2.2500000	.0008079	.0005253	.7409949	.7409948
2.3000000	.0008079	.0005253	.7409949	.7409948
2.3500000	.0008079	.0005253	.7409949	.7409949
2.4000000	.0008079	.0005253	.7409949	.7409949
2.4500000	.0008079	.0005253	.7409949	.7409949
2.5000000	.0008079	.0005253	.7409949	.7409949

APPENDIX F-3

a) Program Listing for Dynamic Feedforward Control of a System with NOVA

b) Main Program Listing and the Results of the Simulation of Dynamic Feedforward Control

```
      DIMENSION XN(50),YN(50),ZN(50)
C     INITIAL VALUES AND CONSTANTS
      DELTA=0.1
      I=1
      DO 1 I=1,30
      XN(I)=0.
      YN(I)=0.
1     ZN(I)=0.
      ACCEPT "T1= ",T1
      ACCEPT "T2= ",T2
      ACCEPT "IT3= ",IT3
C     STEADY STATE VALUE
C
      CALL AIRDW(1,1,2,IVAL,IERR)
      SSV=FLOAT(IVAL)*0.005
      TYPE "SSV= ",SSV
      CALL WAIT(2,2,IERR)
C     CONTROL SECTION
2     CALL AIRDW(1,1,2,IVAL,IERR)
      SVAL=FLOAT(IVAL)*0.005
      INEW=IFIX(ZN(I)/0.005)
      CALL AOW(1,11,INEW,IERR)
      XN(I)=SVAL-SSV
      ZN(I)=XN(I)+(T1-T2)/T2*(XN(I)-YN(I))
      J=I+1
      IF(J.GT.IT3) J=1
      YN(J)=YN(I)+DELTA/T2*(XN(I)-YN(I))
      I=I+1
      IF(I.GT.IT3) I=1
      CALL WAIT(100,1,IERR)
      GO TO 2
      END
```

```
      PROGRAM ALTI
C
      COMMON/VAL/ X(10),F(10),KSTEP
      DIMENSION E(10)
C
C     INITIAL CONDITIONS AND CONSTANTS
C
      N=3
      A=0.0 $ B=5.0
      Y1=0.5 $ Y2=1.0 $ Y3=1.0
      X(2)=.455 $ X(3)=.455
      X(4)=0.0
      X(5)=0.455 $ X(8)=0.0
      READ(61,100) HPRINT,HPLOT,HMIN,TMAX,ERR
      READ(60,100) DEL2,T1,T2,TT1,TT2
  100 FORMAT(5F8.6)
      WRITE(61,200)
  200 FORMAT(3X,"TIME",6X,"X(1)",6X,"X(2)",6X,"X(3)",6X,"X(4)",
     $6X,"X(5)",///)
      AA=(TT1-TT2)/TT2
C
   10 X(1)=STEP(T,A,B,Y1,Y2,Y3)
      DELT=0.5-X(1)
      CALL DSIM(N,T,HPRINT,HPLOT,0.0,HMIN,TMAX,ERR,E,
     $ IERR,ITASK)
      GO TO (1,2,3,4,5) ITASK
    1 GO TO 10
C
C     DERIVATIVE SECTION
C
    2 X(7)=X(1)-X(9)
      CALL TRFN(7,2,T1,1.,1)
      CALL XDEL(2,5,0.146,T,1)
      CALL XDEL(8,4,DEL2,T,2)
      X(6)=X(5)+X(4)
      CALL TRFN(6,3,T2,1.,1)
      X(9)=X(3)*0.1
      GO TO 10
C
C     PRINTING SECTION
    3 WRITE(61,300) T,X(1),X(2),X(3),X(4),X(5)
  300 FORMAT(6F10.7)
```

```
      GO TO 10
    4 CONTINUE
      YN=YN+HPLOT/TT2*(DELT-YN)
      X(8)=DELT+AA*(DELT-YN)
      GO TO 10
    5 CONTINUE
      END
```

0.1 0.01 0.001 5.0 0.001

0.26 0.334 0.1 0.11 0.35

TIME	X(1)	X(2)	X(3)	X(4)	X(5)
0	1.0000000	.4550000	.4550000	0	.4550000
.1000000	1.0000000	.5843238	.4550000	0	.4550000
.2000000	1.0000000	.6800015	.4722984	0	.5296582
.3000000	1.0000000	.7499485	.4916390	-0.1946790	.6395807
.4000000	1.0000000	.8024111	.4617167	-0.2715109	.7204346
.5000000	1.0000000	.8417380	.4528949	-0.3290086	.7801099
.6000000	1.0000000	.8710015	.4513521	-0.3720375	.8250687
.7000000	1.0000000	.8926951	.4520899	-0.4042384	.8586157
.8000000	1.0000000	.9087486	.4532153	-0.4283361	.8835200
.9000000	1.0000000	.9206209	.4541463	-0.4463698	.9019610
1.0000000	1.0000000	.9294011	.4547643	-0.4598655	.9156014
1.1000000	1.0000000	.9358970	.4551113	-0.4699651	.9256886
1.2000000	1.0000000	.9407057	.4552667	-0.4775232	.9331500
1.3000000	1.0000000	.9442679	.4552958	-0.4831793	.9386718
1.4000000	1.0000000	.9469085	.4552549	-0.4874121	.9427609
1.5000000	1.0000000	.9488674	.4551790	-0.4905798	.9457912
1.6000000	1.0000000	.9503216	.4550904	-0.4929503	.9480384
1.7000000	1.0000000	.9514017	.4550023	-0.4947243	.9497060
1.8000000	1.0000000	.9522046	.4549204	-0.4960519	.9509444
1.9000000	1.0000000	.9528017	.4548487	-0.4970454	.9518646
2.0000000	1.0000000	.9532459	.4547879	-0.4977889	.9525488
2.1000000	1.0000000	.9535766	.4547374	-0.4983453	.9530577
2.2000000	1.0000000	.9538230	.4546964	-0.4987617	.9534365
2.3000000	1.0000000	.9540065	.4546633	-0.4990733	.9537186
2.4000000	1.0000000	.9541433	.4546369	-0.4993065	.9539287
2.5000000	1.0000000	.9542453	.4546162	-0.4994810	.9540853
2.6000000	1.0000000	.9543214	.4545999	-0.4996116	.9542021
2.7000000	1.0000000	.9543781	.4545873	-0.4997094	.9542891
2.8000000	1.0000000	.9544205	.4545775	-0.4997825	.9543541
2.9000000	1.0000000	.9544521	.4545699	-0.4998372	.9544025
3.0000000	1.0000000	.9544757	.4545641	-0.4998782	.9544387

3.1000000	1.0000000	.9544933	.4545596-0.4999088	.9544657
3.2000000	1.0000000	.9545065	.4545562-0.4999318	.9544859
3.3000000	1.0000000	.9545163	.4545536-0.4999489	.9545009
3.4000000	1.0000000	.9545237	.4545516-0.4999618	.9545122
3.5000000	1.0000000	.9545292	.4545501-0.4999714	.9545206
3.6000000	1.0000000	.9545333	.4545490-0.4999786	.9545269
3.7000000	1.0000000	.9545364	.4545481-0.4999840	.9545316
3.8000000	1.0000000	.9545387	.4545475-0.4999880	.9545351
3.9000000	1.0000000	.9545404	.4545470-0.4999910	.9545377
4.0000000	1.0000000	.9545417	.4545466-0.4999933	.9545396
4.1000000	1.0000000	.9545426	.4545463-0.4999950	.9545411
4.2000000	1.0000000	.9545433	.4545461-0.4999962	.9545422
4.3000000	1.0000000	.9545439	.4545459-0.4999972	.9545430
4.4000000	1.0000000	.9545443	.4545458-0.4999979	.9545436
4.5000000	1.0000000	.9545446	.4545457-0.4999984	.9545441
4.6000000	1.0000000	.9545448	.4545457-0.4999988	.9545444
4.7000000	1.0000000	.9545450	.4545456-0.4999991	.9545447
4.8000000	1.0000000	.9545451	.4545456-0.4999993	.9545449
4.9000000	1.0000000	.9545452	.4545455-0.4999995	.9545450
5.0000000	1.0000000	.9545452	.4545455-0.4999996	.9545451

APPENDIX G-1

The Derivations of the Controller Equations for Dahlin's Method

Direct Synthesis of Digital Controller Equations

The closed loop system of Fig. 23 may be assumed to behave like a continuous first-order-lag-with-dead-time, which is written in discrete form as follows:

$$\frac{C(z)}{R(z)} = \frac{(1 - e^{-T/\lambda})\, z^{-N-1}}{(1 - e^{-T/\lambda}\, z^{-1})}$$

where λ is the time constant.

The controller then becomes:

$$D_1(z) = \frac{1}{H(z)} \; \frac{(1 - e^{-T/\lambda})\, z^{-N-1}}{1 - e^{-T/\lambda}\, z^{-1} - (1 - e^{-T/\lambda})\, z^{-N-1}}$$

where H(z) is the pulse function.

The pulse function can be determined by assuming first or second-order models for the process. (14)

If the first-order-lag-with-dead-time is used as a process model,

$$G(s) = \frac{Ke^{-\theta s}}{\tau s + 1}$$

$$H(z) = \frac{K\,(1 - e^{-T/\tau})\, z^{-N-1}}{1 - (e^{-T/\tau})\, z^{-1}}$$

and the controller equation is:

$$D_1(z) = \frac{(1 - e^{-T/\lambda})\ (1 - (e^{-T/\tau})\ z^{-N-1})}{K\ (1 - (e^{-T/\lambda})\ z^{-1} - (1 - e^{-T/\lambda})\ z^{-N-1})\ (1 - e^{-T/\tau})}$$

Digital Equivalent of Analog Controller

Another route for obtaining this controller equation is to design an analog controller and then use its discrete equivalent (4), (14), (15).

If the transfer function G(s) of the system is:

$$G(s) = \frac{Ke^{-\Theta s}}{\tau s + 1}$$

The controller equation becomes:

$$D(s) = \frac{\tau s + 1}{Ke^{-\Theta s}} \cdot \left(\frac{C(s) / R(s)}{1 - C(s) / R(s)} \right)$$

If $\frac{C(s)}{R(s)}$ was chosen as:

$$\frac{C(s)}{R(s)} = \frac{e^{-\Theta s}}{\lambda s + 1}$$

$$D(s) = \frac{(\tau s + 1)/K}{\lambda s + (1 - e^{-\Theta s})} = \frac{M(s)}{E(s)}$$

In the time domain D(s) can be represented as:

$$\lambda \frac{dX(t)}{dt} + X(t) - X(t-\Theta) = (\tau \frac{de(t)}{dt} + e(t))/K$$

Expressing the derivatives in difference equations:

$$\frac{(M_n - M_{n-1})}{T} + M_{n-1} - M_{n-N-1} = (\tau \frac{(e_n - e_{n-1})}{T} + e_{n-1})/K$$

$$M_n = (1 - \frac{T}{\lambda})\, M_{n-1} + \frac{T}{\lambda} M_{n-N-1} + \frac{\tau}{K\lambda} (e_n + (\frac{T}{\tau} - 1)\, e_{n-1})$$

D(z) can be obtained in z domain from the equation above

$$D(z) = \frac{M(z)}{E(z)} = \frac{\frac{\tau}{K\lambda} \quad (1 + \frac{T}{\tau} - 1)\, z^{-1})}{1 - (1 - T/\lambda)z^{-1} - (T/\lambda)z^{-N-1}}$$

APPENDIX G-2

a) Program Listing for Control with Dahlin's Algorithm

b) Main Program Listing for the Simulation of Dahlin's Control Algorithm

```
      DIMENSION XL(5),E(5)
C     INITIAL VALUES
C
      DO 1 I=1,5
      XL(I)=0.0
1     E(I)=0.0
      T=0.5
C
C     SET POINT
C
      CALL AIRDW(1,1,2,IVAL,IERR)
      SETP=FLOAT(IVAL)*0.005
      TYPE 'SET POINT= ',SETP
      CALL WAIT(1,2,IERR)
C
      ACCEPT 'TC= ',TC
      ACCEPT 'LANDA= ',XLD
      TT1=T/XLD
      TT2=T/TC
C
C     CONTROL ACTION
C
2     CALL AIRDW(1,1,2,IVAL,IERR)
      SSV=FLOAT(IVAL)*0.005
      E(5)=SSV-SETP
      XL(5)=TC/LD*(E(5)+(TT2-1.)*E(4))+(1.-TT1)*XL(4)+
     $  TT1*XL(1)

      TYPE 'XL(5)= ',XL(5),'E(5)= ',E(5)

      INEW=IFIX(XL(5)/0.005)
      CALL AOW(1,11,INEW,IERR)
      E(4)=E(5)
      XL(1)=XL(2)
      XL(2)=XL(3)
      XL(3)=XL(4)
      XL(4)=XL(5)
      CALL WAIT(500,1,IERR)
      GO TO 2
      END
```

```
      DIMENSION XL(5),E(5)
C     INITIAL VALUES
C
      DO 1 I=1,5
      XL(I)=0.0
1     E(I)=0.0
      T=0.5
C
C     SET POINT
C
      CALL AIRDW(1,1,2,IVAL,IERR)
      SETP=FLOAT(IVAL)*0.005
      TYPE 'SET POINT= ',SETP
      CALL WAIT(1,2,IERR)
C
      ACCEPT 'TC= ',TC
      ACCEPT 'LANDA= ',XLD
      TT1=T/XLD
      TT2=T/TC
      AA=EXP(-TT1)
      BB=EXP(-TT2)
C     CONTROL ACTION
C
2     CALL AIRDW(1,1,2,IVAL,IERR)
      SSV=FLOAT(IVAL)*0.005
      E(5)=SSV-SETP
      XL(5)=(1.-AA)/(1.-BB)*(E(5)-BB*E(4))+AA*XL(4)+
     $   (1.-AA)*XL(1)

      TYPE 'XL(5)= ',XL(5),'E(5)= ',E(5)

      INEW=IFIX(XL(5)/0.005)
      CALL AOW(1,11,INEW,IERR)
      E(4)=E(5)
      XL(1)=XL(2)
      XL(2)=XL(3)
      XL(3)=XL(4)
      XL(4)=XL(5)
      CALL WAIT(500,1,IERR)
      GO TO 2
      END
```

```
      PROGRAM DAHLIN
      COMMON/VAL/ X(10),F(10),KSTEP
      DIMENSION E(10),XL(10),XERR(10)
C
C     INITIAL CONDITIONS
C
      N=4
      DO 6 I=1,10
      X(I)=0.0
      XL(I)=0.0
    6 XERR(I)=0.0
      A=0.0
      B=5.0
      Y1=0.0 $ Y2=1.0 $ Y3=1.0
      SP=0.0
      READ(60,100) HPRINT,HPLOT,HMIN,TMAX,ERR
      READ(60,100) TC1,XDET,TC2,TC,XLD
      TT1=0.05/XLD
      TT2=0.05/TC
      AA=EXP(-TT1)
      BB=EXP(-TT2)
      WRITE(61,200)
  100 FORMAT(5F8.6)
  200 FORMAT(5X,"T",6X,"X(0)",6X,"X(4)",6X,"X(6)",6X,"X(1)",///)
C
C     DERIVATIVE SECTION
C
   10 XO=STEP(T,A,B,Y1,Y2,Y3)
      CALL DSIM(N,T,HPRINT,HPLOT,0.0,HMIN,TMAX,ERR,E,IERR,
     $ ITASK)
      GO TO (1,2,3,4,5) ITASK
    1 GO TO 10
    2 X(1)=XO-X(7)+X(6)
      CALL TRFN(1,2,TC1,1.,1)
      CALL XDEL(2,3,XDET,T,1)
      CALL TRFN(3,4,TC2,1.,1)
      X(7)=X(4)*0.1
      GO TO 10
C
C     PRINTING SECTION
C
    3 WRITE(61,300) T,XO,X(4),X(6),X(1)
```

```
  300 FORMAT(5F10.7)
      GO TO 10
C
    4 XERR(2)=SP-X(4)
      X(6)=(1.-AA)/(1.-BB)*(XERR(2)-BB*XERR(1))+AA*XL(4)
     $ +(1.-AA)*XL(1)
      XL(1)=XL(2)
      XL(2)=XL(3)
      XL(3)=XL(4)
      XL(4)=X(6)
      XERR(1)=XERR(2)
      GO TO 10
C
    5 CONTINUE
      END
```