TO MY BELOVED

FATHER AND MOTHER

ABSTRACT

The dynamics of a semi-works scale binary distillation column, separating methanol and water, was studied. Non-linear steady-state and dynamic models of this column based on material and energy balance were developed and subsequently solved. The predicted transient and steady-state behavior agreed with the results from experimental testing.

Two optimal control policies based on a linearized dynamic model were formulated. The first optimal policy required the control to bring the product compositions to their desired level while minimizing the following quadratic performance index.

$$J_1 = \underline{x'(t_f)}\,\underline{S}\,\underline{x(t_f)} + \int_0^{t_f} \tfrac{1}{2}(\underline{x'(t)}\,\underline{Q}\,\underline{x(t)} + \underline{u'(t)}\,\underline{R}\,\underline{u(t)})dt \quad \text{(A-1)}$$

The second optimal policy required the control to bring the product compositions to their desired level while minimizing the same J_1 function and satisfying the following integral constraints.

$$I_1 = x_{D,ave} = \int_0^{t_f} x_N(t)dt = 0 \quad \text{(A-2)}$$

$$I_2 = x_{B,ave} = \int_0^{t_f} x_{\bar{1}}(t)dt = 0 \quad \text{(A-3)}$$

The resulting optimal control trajectories of these two optimal policies were tested with the forementioned non-linear model and the results were satisfactory. Also, the optimal control trajectories showed a definite trend and shape which depended only on the nature of the disturbance. Because of this particular feature, simplified optimal control schemes can be derived and implemented with a control computer for direct digital computer control of the distillation column operation.

ACKNOWLEDGEMENT

The author wishes to express his sincere gratitude to his supervisor, Dr. F. D. F. Talbot who has not only provided a great deal of technical guidance but also much appreciated encouragement and patience throughout the course of this work.

He is grateful to Mr. G. Gasperetti and the rest of the technical staff of the Chemical Engineering Department for their help in constructing and maintaining the equipment, also to Miss N. Sadidi for her help in part of the experimental work.

He also wishes to thank Dr. J. Polak for his help in calibrating the Beckman thermometer.

The many discussions with Dr. S. D. Chang and Mr. H. A. Laudie have augmented the author's skill and knowledge in numerical techniques as well as computer programming.

The ICOS program, supplied by Dr. L. Birta is also gratefully acknowledged.

Mr. K. T. Heng deserves special thanks for his assistance in preparing many of the figures contained in this work.

Last, but not the least, the patience and understanding of the author's wife is also highly appreciated.

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$\underline{A}$	N x N coefficient matrix of the state.
$\underline{\bar{A}}$	N x N coefficient matrix (equation (5-16)).
$\underline{AA}$	N x N coefficient matrix (equation (3-6)).
A_1	Constant term of equation (2-6).
Ab	Coefficient matrix of equation (2-9).
$\underline{B}$	N x m coefficient matrix of the control.
$\underline{\bar{B}}$	N-dimensional vector (equation (5-16)).
$\underline{BB}$	N-dimensional vector (equation (3-6)).
B_1	Constant term of equation (2-6).
b_i	Bottom product withdrawal rate of component, i lb-mole/hr.
B	Bottom product withdrawal rate, lb-mole/hr.
C_a, C_b	Constants.
C	Total number of components (2).
D	Top product withdrawal rate (Distillate rate), lb-mole/hr.
d_i	Top product withdrawal rate of component i, lb-mole/hr.
E	Error function (equation (B-15)).
E(t)	Quantity defined by equation (2-6).
E_j	Efficiency of tray j.
F	Feed flow rate, lb-mole/hr.
$\underline{FF}$	Feed disturbance vector.
$\underline{Ff}$	Feed disturbance vector after transformation.

ff	Non-zero element of $\underline{Ff}$.
$H_{j,i}$	Enthalpy of vapor stream of component i and tray j, Btu/lb-mole.
$h_{j,i}$	Enthalpy of liquid stream of component i and tray j, Btu/lb-mole.
H_D H_B	Enthalpy of the top and bottom flow streams.
$\underline{H}$	Hamiltonian defined by equation (7-18).
i	Component number i = 1 for methanol, i = 2 for water.
$\underline{i_5(t)}$	The fifth column of matrix $\underline{J}$, N-dimensional vector.
I_1 I_2	Integral constraints defined by equations (A-2) and (A-3).
j	Tray number.
$\underline{J(t)}$	N x N matrix defined by equation (7-52).
J_1	Quadratic performance defined by equation (A-1).
J_2	Performance index defined by equation (B-4).
k	Integer used for counting.
$\underline{k_1(t)}$	N-dimensional vector defined by equation (7-53).
$\underline{k_N(t)}$	N-dimensional vector defined by equation (7-54).
$\underline{K(t)}$	N x N matrix.
$K_{j,i}$	Equilibrium constants.
$l_{j,i}$	Liquid flow rate of component i, lb-mole/hr.
L, L_j	Liquid flow rate, lb-mole/hr.
$\underline{L(t)}$	N x N matrix.

$\underline{l_5(t)}$	N-dimensional vector defined by equation (7-55).
m	Number of control variable (2).
n	Number of time step.
nn	Order of the system (number of state variable equation (B-1)).
N	Total number of tray + condenser + reboiler (12).
$\underline{Q}$	N x N weighting matrix.
Q_j	Heat flow in and out of a tray.
Q_c	Heat removed by condenser, Btu/hr.
Q_r	Heat input to the reboiler, Btu/hr.
$\underline{R}$	m x m weighting matrix.
r	Number of control variable (equation (B-1)).
$\underline{S}$	N x N weighting matrix.
t	Time in minutes.
t_f	Control period in minutes.
T_j	Temperature of tray j in °F.
TT_1, TT_2	Time constants of equation (2-6).
$\underline{\bar{u}(t)}$	Control vector (deviation variable).
$\underline{u(t)}$	Control variable after transformation $\underline{u(t)} = \underline{\bar{u}(t)} - \underline{\bar{u}_d}$.
$u_{j,i}$	Liquid tray holdup of component i and tray j, lb-mole.
$\underline{u^*(t)}$	Optimal control trajectory.
U_j	Liquid holdup of tray j, lb-mole.
U_j^V	Vapor holdup of tray j, lb-mole.

$v_{j,i}$	Vapor flow rate of component i, lb-mole/hr.
$\underline{v}$	N-dimensional vector with element v_j, for a component i.
V_j	Vapor flow rate, lb-mole/hr.
W_j	Liquid draw off rate from tray j.
w	Initial value of the co-state vector (equation (B-14)).
X_F	Feed composition, mole fraction.
$\underline{X(t)}$	Liquid compositions, mole fraction.
$\underline{\bar{x}(t)}$	Liquid compositions (deviation variables).
$\underline{\bar{\bar{x}}(t)}$	Variables after the following transformation $\underline{\bar{\bar{x}}}(t) = \underline{U\bar{x}}(t)$.
$\underline{x}(t)$	State variables after transformation $\underline{x}(t) = \underline{\bar{\bar{x}}}(t) - \underline{\bar{\bar{x}}_d}$.
$Y_j(t)$	Vapor composition mole fraction.
$\underline{Z(t)}$	N-dimensional co-state vector of the state $\underline{x(t)}$.
$\underline{Z_c}$	N-dimensional co-state vector of $\underline{x_{ave}}$.
$z_{c,1} z_{c,N}$	Non-zero elements of $\underline{Z_c}$.
μ	Constant of the implicit method.
θ_j	Multiplier associated with tray holdups and distillate rate.
$\underline{\theta_1 \theta_2}$	N-dimensional vectors defined by equation (7-49).
$\Phi(\underline{t_1 t_0})$	2N x 2N fundamental matrix of equation (7-25).

$\gamma_{j,i}$ Activity coefficient of component i, tray j.

2N x 2N matrix defined by equation (7-28).

ΔB Deviation of bottom product withdrawal rate, lb-mole/min.

ΔD Deviation of top product withdrawal rate, lb-mole/min.

ΔL Deviation of liquid flow rate, lb-mole/min.

Δt Time increment (step size) in minutes.

ΔV Deviation of vapor flow rate, lb-mole/min.

$$\text{matrix I} = \begin{bmatrix} 1 & & & & & & & & & \\ & 0 & & & & & & & & \\ & & 0 & & & & & & & \\ & & & 0 & & & & & & \\ & & & & 0 & & & & & \\ & & & & & 0 & & & & \\ & & & & & & 0 & & & \\ & & & & & & & 0 & & \\ & & & & & & & & 0 & \\ & & & & & & & & & 1 \end{bmatrix} \quad \text{matrix II} = \begin{bmatrix} 1 & & & 0 \\ & & & \\ & & & \\ 0 & & & 1 \end{bmatrix}$$

SUBSCRIPTS:

ave	Average.
B	Bottom product.
ca	Calculated value.
co	Corrected value.
D	Top product.
d	Desired state.
F	Feed stream.
o	Initial time.
S.S.	Steady state.

SUPERSCRIPTS:

o	Previous value of a variable.
*	Optimal state.

CHAPTER I

INTRODUCTION

Continuous distillation has played a very important role in the chemical and petroleum industries. It is the type of mass transfer operation that physically separates a mixture of two liquids of different volatility by applying energy to the system. It is a fairly complex operation because simultaneous transfer of energy and mass is involved. Because of the complexity of the operation and the lack of advanced equipment, early researchers could not have a thorough picture of the distillation operation. The design engineers of the past had to resolve this by overdesigning the column, that is, they put more trays in the column and made it with a larger diameter than was necessary. With the advent of sophisticated analytical equipment and high speed digital and analog computers, chemical engineers have the opportunity to understand more about the distillation operation than they ever had before.

The analysis of a distillation operation can mainly be divided into two parts, the transient and the steady state. Early graphical methods, like the McCabe-Thiele diagram and the Ponchon-Savarit enthalpy concentration diagram are for the steady state analysis. Transient analysis is more involved because the column conditions change with time. The equations involved in the transient state are usually a

system of differential and algebraic equations of non-linear nature. Analytical solutions of these equations are next to impossible. In order to obtain the analytical solutions of these equations certain assumptions were made by early researchers. Because of these assumptions, these equations were simplified and became linear and thus the Laplace transform could be applied to obtain the solution.

With the arrival of the high speed digital computer, numerical solutions can readily be obtained for these non-linear equations that describe the transient state of the distillation operation. The assumptions that were once necessary can now be relaxed. The mathematical dynamic model becomes a better and closer representation of the column operation. Also, rigorous calculation methods can be used instead of the simplified graphical methods to evaluate the steady state operation of the distillation column.

In order to keep the distillation column operating at a certain steady state condition, the streams that flow in and out of the column, as well as, the internal flows must be kept constant. Therefore, some kind of control system must be incorporated on the column to maintain the desired operating condition. Besides, it is not unusual that disturbances occur in the feed stream. Whenever a disturbance enters the column, the top and bottom product compositions will deviate from their desired values. The control system must bring them back to their desired values. If only one of the two

products is important, the control system with one manipulated variable will be good enough, i.e. reflux flow rate to control the top product composition, and heat input to the reboiler (steam flow rate to the reboiler) to control the bottom product composition. However, when both product compositions are critical, control loops have to be set up around the two ends of the column. The pair of manipulated variables usually chosen by people in the chemical industries are either the top product withdrawal rate and the reboiler steam rate, or the reflux rate and the bottom product withdrawal rate. Besides keeping both terminal compositions at the desired level, this two-point control system can also maintain the yield of the distillation operation.

With the help of the analog and digital computers, a new kind of control scheme has been developed. It is the so called "feedforward control". The advantage of this one over the conventional feedback control is that disturbances are detected before they enter the column. Therefore, the required control action can be anticipated and better control of the column will be obtained. Before a feedforward control system can be constructed, dynamic and steady state models are necessary to describe the column operation. When a disturbance is sensed, the signal will be transmitted to the system and the necessary control action will be calculated based on the mathematical model of the column. Thus, the effectiveness of the feedforward control scheme is highly

depended on the accuracy of the mathematical model.

Whan a control system is designed, it is difficult to assess whether it is the "best control" or not. With the arrival of the state space technique and modern control theory, true optimal (the best) control can be obtained for a linear system with respect to a certain performance criterion. The concept of the optimal control strategy and the idea of feedforward control, blend into each other and emerge as a powerful control scheme for the chemical industry. Many binary distillation column operations can be described by a linear system, therefore, the truly optimal feedforward controller can readily be synthesized through the present knowledge of modern control theory. Although the foremention two-point control system has widely been used in the chemical industries, no literature has yet been found dealing directly with implementing optimal control policies with this two-point control system for controlling the operation of a distillation column.

The present work was centered around a piece of existing distillation equipment belonging to the Department of Chemical Engineering, University of Ottawa. It is a semi-works scale 10- tray distillation column used to separate mixture of methanol and water.

The first part of this study was devoted to the development of dynamic and steady state mathematical models to represent the operation of this column. Because of the non-

ideal nature of the mathanol-water system and the extreme operating conditions of the column, it was found that a non-linear dynamic model was required to represent the distillation column. The non-linear distillation model developed by Holland and Waggoner (1) was successfully adopted and modified in this work. This mathematical model was eventually used to test the effectiveness of the optimal controllers which were synthesized using a simpler linear dynamic model.

The second part of this study was devoted to the development of the optimal control policies. The state-space approach and the linear synthesis techniques of the modern control theory was used in this development. A simple linear dynamic model based on material balance only was used. The system of differential equations which made up this model was transformed into the linear time-invariant state-space equations. The techniques of the minimum principle of Pontryagin and quadratic performance index was used to synthesize the optimal controllers. Two optimal controllers based on different control philosophies were constructed using the linear dynamic model. These were incorporated into a simulation of the forementioned feedforward two-point distillation control scheme using the non-linear dynamic model of the distillation column. The effectiveness of the optimal control scheme was evaluated when set-point and load disturbances were introduced into the column and were found to be satisfactory.

The linear synthesis and quadratic performance index

techniques of the modern control theory was found to be a versatile method for constructing optimal feedforward controllers. It permits one to handle problems of set-point and load disturbances in one formulation. And because of the properties of the linear systems, an optimal solution of the problem is guaranteed. Also from the numerical calculation point of view, the calculation is straightforward. This is an advantage over non-linear synthesis which is more complex and in which there was no guarantee that an optimal solution exists, especially for high-order systems like the distillation column in this study. Until more advanced and sophisticated numerical techniques which guaranteed a better result have been developed, the approach used in this study is a very valuable alternative.

CHAPTER II

LITERATURE SURVEY

A) STEADY STATE CALCULATION

Steady state calculation of distillation operations has always been important to chemical industry. Before the availability of the high speed computer, graphical and short cut methods had been widely used. For binary systems, there are graphical methods like the McCabe-Thiele diagram (3) and the Ponchon-Savarit enthalpy concentration diagram (3). For multicomponent systems, there are short cut methods like the Edmister method (3), the Fenske-Underwood's total reflux equation (3), Underwood's minimum reflux equation (3) and the Gilliland correlation (3). Some rigorous calculation methods which are based on tray to tray calculation like the Lewis-Matheson method (3) and the Thiele-Geddes method(3) were also proposed. With the arrival of digital computer, these rigorous methodshave become more popular. Lyster et al. (4) proposed a new convergence procedure: - the θ-method of convergence -. This convergence procedure together with the Thiele-Geddes method have turned into a powerful tool for solving steady state distillation problems. In 1958, Amundson and Pontinen (5) aptly described the distillation problem as a system of simultaneous algebraic equations and solved them with the help of matrix algebra. The major difference between this approach and the conventional ones is

that this method does not involve the tray to tray calculation, it solves the equations directly.

Rose, Sweeny and Schrodt (6) have applied the familiar relaxation method to distillation problems. Because of the nature of this method, it also gives an approximation of the transient response of the column while performing the steady state calculations. But this method is considerably slower than the others. Ball (8) has modified this method so that steady state can be obtained in a shorter computation time.

Various convergence procedures have been reported and modified by researchers like Peiser (9), Bonner (10), Greenstadt et al. (11). A comparison of the convergence procedures when applied to various calculation methods was reported by Holland in his text (7). Recently, Tomich (12) (1970), Billingsley (13), Gentry (14), and Seppala and Luus (15) (1972) have reported improved calculation procedures for the steady state distillation processes. The texts by Smith (3), Van Winkle (16) and Holland (7) are good sources of reference on the steady state distillation calculation methods.

Because of the availability of large scale high speed digital computer, obtaining the solution of the steady state equation is no longer a problem. Recently, researchers have concentrated on improvement of the convergence techniques in order to shorten the computation time.

B) UNSTEADY STATE CALCULATION

1) LAPLACE TRANSFORM

Marshall and Pigford (17) were the earliest investigators who analysed the unsteady state distillation operation quantitatively. They formulated the differential equations which described the rate of mass transfer on a single distillation tray j as follows:

$$U\frac{dx_j}{dt} + U^V\frac{dY_j}{dt} = V(Y_{j-1}-Y_j) + L(X_{j+1}-X_j) \qquad (2\text{-}1)$$

Where U and U^V are the holdups of liquid and vapor respectively. X_j and Y_j are the liquid and vapor compositions leaving tray j. V and L are the total vapor and liquid rates.

The set of differential equations was solved analytically by Laplace transformation with the help of the following assumptions.

a) Linear equilibrium relationship of the form $Y_j = C_a X_j + C_b$ existed.

b) Vapor and liquid leaving the tray were assumed to be in equilibrium.

c) Total vapor and liquid rates entering and leaving each tray were independent of time and tray number.

d) Liquid and vapor holdups were independent of time and tray number.

e) The contents of the vapor and liquid holdups were perfectly mixed.

Lapidus and Amundson (18) applied the same methods to obtain a transient solution of an absorber problem. Mickley, Sherwood and Reed (19), also utilized the Laplace transform to obtain an analytical solution to the problem which described the rate of approach to the steady state of a plate absorber. The assumption of constant liquid and vapor rates, constant liquid and vapor holdups and linear equilibrium relationship for every tray were made as in the previous cases. The development was carried out in detail for both 100% tray efficiency and trays other than perfect.

Wilkinson and Armstrong (20) (21) applied the same treatment to a five-tray binary enriching column operating at total reflux. The theoretical responses of the column under feed composition step changes were confirmed by the experimental results. Armstrong and Wood (22), then applied the same Laplace transform to a 21 trays binary distillation column.

Voetter (23) investigated the propagation of a concentration disturbance through an enriching sieve tray column both experimentally and theoretically. He then extended the mathematical analysis to the transient response of a complete conventional column. The work by Voetter, Wilkinson and Armstrong all involved the common assumptions of 100% tray efficiency, constant liquid tray holdups, adiabatic operation and constant

molal overflow.

In 1967, Osborne and Maddox proposed a simplified model in which the column was divided into three sections. They were: the rectifying section, the feed section and the stripping section. Each section was represented by a component material balance and an enthalpy balance equation. Transient experimental results of a bench scale column, with ten perforated trays, separating binary and ternary systems were obtained and compared with the models results. Osborne and Maddox claimed that the accuracy of their model depended highly on the accuracy of the flow rate measurements.

In 1969, Pohjola and Norden (24) proposed a linearized dynamic model for binary distillation. In their proposal, the distillation process was divided into two subprocesses, namely, the mass transfer process and the liquid flow process. The transient response was obtained by the superposition of the responses of the two subprocesses. The assumptions of constant molal overflow, perfectly mixed tray holdups and negligible vapor holdups, were also made.

The Laplace transform methods are useful in those cases where there is only one volatile component such as the absorber and the binary distillation column. As far as the multicomponent distillation problem is concerned, this technique fails to produce an analytical solution.

2) ANALOG COMPUTER

Analog computer has been one of the tools for solving differential equations of the type given by equation (2-1). The absorber problem studied by Lapidus and Amundson (18) was again solved by Acrivos and Amundson (26), using an analog computer in 1953. The cases they investigated included linear equilibrium relationships as well as non-linear ones. Yesberg and Johnson (27), using the analog of Liebman which consisted only of resistance elements, simulated the start-up period of a six plate absorber. Again, the system under study had only one volatile component.

Using a "Differential Analyzer" Pigford, Tepe and Garrahn (28) obtained the transient profile of the batch distillation of a binary mixture. The relative volatility as well as the vapor and the liquid flow rates were taken to be constant with respect to time and tray number.

Rose, working with Johnson and Williams (29) (30) demonstrated the use of an analog computer in solving the equations which described the continuous distillation of a binary mixture. The transient conditions produced by changes in the feed composition were presented. Again this procedure was restricted to binary systems. Assumptions were also made of constant liquid and vapor flow rates of the rectifying and

stripping sections.

Lamb, Pigford and Rippin (31) presented a new approach for solving unsteady state distillation problems. The system of equations used to represent the dynamic characteristic of a distillation column was developed by using the following assumptions:

a) By small perturbation, equations could be linearized around the steady state values.

b) Negligible vapor holdup and perfectly mixed liquid holdups.

c) At steady state, the liquid and vapor rates were constant throughout the stripping and rectifying sections.

d) The tray efficiencies were allowed to vary from tray to tray but were independent of time.

The equations were solved on an analog computer and verified with experimental results. Important contributions of this work were: a) it treated non-constant liquid holdups, b) the linearization of the system equations allowed the analog solution to be accomplished with considerably less analog equipment than the solution of non-linear equations.

In 1959, Rijnsdorp and Maarleveld (32) and Rademaker and Rijnsdorp (33) presented an electrical analog of a distillation column which included an interconnected pressure network, vapor flow network, concentration network and liquid flow network. The merit of this

electrical analog was that it consisted chiefly of passive elements with a minimum use of operational amplifiers. The main assumptions made in developing these model equations were

a) Linear behavior

b) Binary mixture

c) Theoretical trays

d) Equal molal overflows

In 1964, Luyben, Vernevill and Gerster (34) extended the work of Lamb, Pigford and Rippen to a pilot scale binary distillation column with ten trays. Their study showed that a set of linear perturbation type differential equations could predict the unsteady state performance of their column with reasonably good precision and they could also be easily simulated on an analog computer. However, the accuracy deteriorated as the magnitude of the perturbation became larger. The merit of the analog computer is that, the differential equations which represent the distillation column can be integrated directly. The capacity of the analog machine is usually not large enough to handle multicomponent or even non-linear binary distillation problems. In recent years, researchers have been inclined to use digital computers instead of the analog computers to attack distillation problems.

3) DIGITAL COMPUTER

The analytical solution of a distillation problem seldom exists unless it has been simplified by making assumptions. The application of the analog computer to this problem is also limited by the capacity of this machine. Naturally, the choice for this work is the digital computer. Since this computer cannot integrate the differential equations directly, a numerical procedure must be used to obtain an approximation of the differential equations which is acceptable to the digital machine.

In 1956, Jackson and Pigford (35) illustrated the use of a digital computer to obtain a numerical solution corresponding to the start up period of a column separating a binary mixture at total reflux. At total reflux, L = V, and assuming the vapor holdups are negligible, equation (2-1) reduces to

$$Y_{j-1,i} - Y_{j,i} + X_{j+1,i} - X_{j,i} = \frac{dX_{j,i}}{dt} \qquad (2\text{-}5)$$

On the basis of constant relative volatility and Murphree efficiencies, transient solutions were obtained for two types of reboiler.

By assuming constant relative volatility, flow rates and holdups within each section of a binary column, Gilliland and Mohr (36) predicted the transient behavior of the column. A digital computer was used to solve a system

of equations similar to equation (2-1). These results were used to evaluate the time constants TT_1 and TT_2,

$$E(t) = \frac{X(t) - X(\infty)}{X(0) - X(\infty)} = A_1 e^{-t/TT_1} + B_1 e^{-t/TT_2} \qquad (2\text{-}6)$$

$X(t)$ denotes the value of X at time t. The quantities A_1 and B_1 may be stated in terms of TT_1 and TT_2. Equation (2-6) was combined with equations for the change in feed composition to give transfer functions for sections of the column. By using these transfer functions, the responses of the column to step changes in feed composition were predicted and compared with the responses calculated by the numerical integration of the differential equations for the column.

The relaxation method proposed by Rose, Sweeney and Schrodt (6) was originally used to determine the steady state column conditions. Using the normal simplifying assumptions of constant molal overflow, negligible vapor holdup, constant liquid holdup and perfect tray mixing, the following equation was presented.

$$U_j X_{j,i} = U_j X^{\circ}_{j,i} + \Delta t(VY^{\circ}_{j-1,i} + LX^{\circ}_{j+1,i} - VY^{\circ}_{j,i} - LX^{\circ}_{j,i}) \qquad (2\text{-}7)$$

where the superscript $^{\circ}$ denotes the compositions at the old time level. The working equation which was solved explicitly for the composition of tray j at the new time level, was

$$X_{j,i} = X^{\circ}_{j,i} + \frac{1}{U_j}\Delta t(VY^{\circ}_{j-1,i} + LX^{\circ}_{j+1,i} - VY^{\circ}_{j,i} - LX^{\circ}_{j,i}) \qquad (2\text{-}8)$$

An equation such as this could be written for each component and for each tray. The computation started with all tray compositions equal to the feed concentration. Each tray composition slowly approached its steady state value. It was suggested that this method could be used to solve the unsteady state distillation problem. This method is similar to the one proposed by Rosenbrock (37) to solve the unsteady state binary distillation problem.

In 1965, Albritton and Murrill (38) used the relaxation method to solve an unsteady state distillation problem in which a change in reflux flow rate had occurred. A serious disadvantage of this method as pointed out by Rose et al. (7) was that an extremely small time increment must be chosen to avoid distortion. Therefore, a large amount of computation time was required to reach the steady state. These observations were confirmed by Sweeney (39) in a later paper. In order to reduce the number of time periods required by the relaxation method to approach the steady state solution, Ball (8) modified the process which permitted the time period to be lengthened appreciably. Unfortunately, this modified method may lead to an incorrect transient solution even though the steady state solution is corrected.

These are the works done by early researchers who had utilized the digital machine to obtain the numerical solutions of dynamic distillation equations. Most of the problems involved binary systems.

4) MATRIX METHOD

Acrivos and Amundson (40) have brought the important branch of mathematics--matrix algebra--to the field of chemical engineering. They demonstrated the use of matrix operation to solve the system of equations representing the unsteady state behavior of a distillation column with multicomponent systems.

For each component, an equation of the form of equation (2-1) can be written for each tray as well as for the condenser and reboiler. By assuming negligible vapor holdup equation (2-1) can be written in the following form:

$$\frac{dX_{j,i}}{dt} = \frac{V_{j-1}K_{j-1,i}X_{j-1,i}}{U_j} - \frac{L_jX_{j,i}}{U_j} - \frac{V_jK_{j,i}X_{j,i}}{U_j} + \frac{L_{j+1}X_{j+1,i}}{U_j} \qquad (2\text{-}8a)$$

where $K_{j,i}$'s are the equilibrium constants relating the vapor and liquid compositions. The assumption that molal holdup is independent of time was also made.

The coefficients of all X's of equations (2-2) over the entire column can be grouped together into a matrix denoted by Ab. All the X's can be put into a vector denoted by X(t), and the feed input by F(t). With

the help of these notations, the system of equations for a given component over the entire column can be represented by the following vector-matrix differential equation

$$\frac{d\underline{X}(t)}{dt} = \underline{Ab}\ \underline{X}(t) + \underline{F}(t) \quad , \qquad (2\text{-}9)$$

Acrivos and Amundson (40) further stated that if equation (2-9) is linear, the solution of this equation can be given by the following expression

$$\underline{X}(t) = \underline{M(Ab)}\ \underline{X}(0) + \underline{M(Ab)} \int_0^t \underline{M(Ab)}^{-1}\ \underline{A}\ \underline{F}(t)\ dt \qquad (2\text{-}10)$$

Where $\underline{M(Ab)}$ is the Matrizer of $\underline{Ab}$. The development of this well known relationship is given by Frazer et al. (41).

Equation (2-10) is an expression for the composition of the liquid stream (leaving the absorber) and it is a function of time. In this development, it was assumed that the elements of the matrix $\underline{Ab}$ which contained the flow rates, equilibrium constants and the tray holdups remained constant with respect to both time and tray number. They also showed that for an absorber with a single volatile component transferring between two inert streams, the matrizer of $\underline{Ab}$ could be evaluated to give an analytical solution of equation (2-10).

The work of Acrivos and Amundson lead to the procedure presented by Mah, Michaelson and Sargent (42) for the

solution of a multicomponent, multi-phase system in 1962. Again, the system equations were represented by equation (2-9). By making the assumption that the matrix $\underline{A}$ was constant throughout each time period, it was possible to use the solution for a linear matrix equation as given by Frazer et al. (41). A set of new flow rates and equilibrium constants were calculated at the end of each time period, and these values were used for the next time period. These authors noted that the procedure led to sets of tray compositions for which $\sum_{i=1}^{c} X_{j,i} \neq 1.0$. This problem was cured by normalizing the tray compositions.

Sargent (41) acknowledged that the validity of this approximation had not been substantiated and proposed a variation of the earlier procedure. The system was described by equations of the same form except that the $U_{j,i}$'s were taken to be the dependent variables rather than the $X_{j,i}$'s. The coefficients in the material balance equations were permitted to be linear functions of time in the revised solution. It was proposed that the corrected functions for each time period could be obtained by an iteration scheme. No example was presented to demonstrate the use of the proposed calculation procedure.

In 1964, Waggoner (44) used the matrix equations proposed by Sargent to solve an unsteady state distillation problem. The procedure, used to obtain a numerical solution from the matrix equation, differed from that proposed by Sargent.

Waggoner showed that there were a few limitations to this method. The most significant were:

a) The solutions obtained differed when the length of the time period varied.

b) The solutions did not converge to the corrected final steady state.

Waggoner suggested that the limitations he encountered would apply equally to the numerical method proposed by Sargent. As a matter of fact, Waggoner had tried the same numerical procedure suggested by Sargent to obtain the solution of the same matrix equation. It was found that differences in the numerical results could be attributed to those which arose in computation.

5) RUNGE-KUTTA METHOD

Numerical integration by the Runge-Kutta method has been used to solve the unsteady state multicomponent distillation problem. Mah et al. (42), Duffin (45), Duffin and Gamer (46) have used this method. As pointed out by Mah et al., the step size used for this integration scheme is extremely critical from the stability standpoint. This method, however, does have the advantage of not requiring any special starting procedure.

The model used by Duffin (45) assumed constant liquid holdup, no liquid or liquid holdup between trays, perfect mixing, negligible vapor holdup and instantaneous equilibrium between the vapor and the liquid at each tray. The solution of the differential material and energy balance equations proceeded as follows:

a) The Runge-Kutta-Gill (47) algorithm was used to integrate each component material balance for one time step. During this integration, the column liquid and vapor flow rates were assumed constant.

b) From the composition obtained at the end of the time step, new temperatures and stream enthalpies were computed and the component material balance integration was repeated until the enthalpy balance was also satisfied. The internal column flow rates were allowed to vary in order to satisfy the tray enthalpy balance.

Duffin and Gamer (46) generalized this by allowing variable liquid holdup, as determined by the tray hydraulics. This solution procedure was essentially the same as that used by Duffin (45)

The advantage of this method is, that it is relatively simple and easy to program. It is rather a time consuming process because each point has to be iterated n times (n is the order of the Runge-Kutta routine) before it can proceed to the next point. Also, it does not provide any check for the truncation error as in the predictor-corrector method.

6) PREDICTOR-CORRECTOR METHOD

Huckaba, Franke and May (48) (49) have presented the transient solution of a binary distillation column and verified the results with experimental data. The following assumptions were made during the derivation of the equations.

a) Perfect tray mixing

b) Constant holdup for all trays

c) Negligible vapor holdup

d) Negligible liquid and vapor entrainment

e) Tray efficiencies independent of time.

The equations used by Huckaba et al. to describe the process were of the following form:

Total material balance

$$-V_j + V_{j-1} + L_{j+1} - L_j + F_j = 0 \qquad (2\text{-}11)$$

Component material balance

$$V_{j-1}Y_{j-1,i} + L_{j+1}X_{j+1,i} - V_{j,i}Y_{j,i} - L_jX_j + FX_F = U_j\frac{dX_{j,i}}{dt} \qquad (2\text{-}12)$$

Enthalpy balance

$$V_{j-1}H_{j-1,i} + L_{j+1}h_{j+1,i} - V_jH_{j,i} - L_jh_{j,i} + Q_j = h_j\frac{dX_{j,i}}{dt} \qquad (2\text{-}13)$$

The differential equations were numerically integrated using a fifth order predictor-corrector technique developed by Shell (50). The method was not self-

starting and required four history points. In this procedure, the four history points were supplied by the Runge-Kutta-Gill integration routine before switching to the predictor-corrector method. Also, four new history points were supplied by the same routine each time when the time increment size was changed.

In 1964, Distefano (50) extended the work of Huckaba et al. by generalizing the model to include variable liquid holdup and presented new experimental verifications. It must be pointed out that in the calculation procedure of Distefano (50), the model equations used were identical to those of Huckaba et al. That is, equations (2-11), (2-12) and (2-13) were used. The assumption was made that the change in liquid holdup with time was negligible compared to all other terms in the equations. In this calculation procedure, the liquid tray holdups at the beginning were assumed constant over the entire step, and then updated at the end of the step with empirical correlations for the next time step. In this manner, the use of mean holdups over a time step which required iteration was avoided.

The predictor-corrector calculation used by Distefano (50) did not require a starting method such as the Runge-Kutta-Gill routine as in the calculation procedure of Huckaba (48) (49). Distefano (50) used a self-

starting procedure which was described in part two of the paper by Shell (50), thereby eleminating the need of the Runge-Kutta-Gill algorithm.

The merit of the predictor-corrector method is that it provides an available measure of the truncation error. This was used as a guideline for setting the time step size.

7) IMPLICIT METHOD

A powerful method for solving the unsteady state multicomponent distillation problem called the implicit method was first used by Rosenbrock et al. (51) and Ball (8). This method was then developed in detail by Waggoner and Holland (1) (44) (51) later in 1964. In the development of the model by Rosenbrock et al. (51), the following assumptions were made.

a) The vapor and the liquid flow rates were assumed to be constant during the transient period. These flow rates were allowed to vary with tray number to satisfy an initial heat balance, but no variation was allowed during the transient period.

b) Vapor holdup was negligible and liquid holdup was constant for a particular tray but varied from tray to tray.

c) The tray holdups were perfectly mixed.

d) The vapor and the liquid leaving a tray were in equilibrium.

With these assumptions, an i component material balance around tray j yielded:

$$U_j \frac{dX_{j,i}}{dt} = V_{j-1}Y_{j-1,i} - V_j Y_{j,i} - (L_j + W_j)X_{j,i} + L_{j+1}X_{j+1,i} + FX_{F,i} \qquad (2\text{-}14)$$

Where W_j was the liquid draw off rate from tray j. Rosenbrock (51) following his earlier development for

transient binary distillation (53) (54) (55) replaced the derivative in equation (2-14) by its slope equivalent.

$$\frac{dX_{j,i}}{dt} = \frac{X_{j,i} - X^{\circ}_{j,i}}{\Delta t} = \frac{1}{U_j}\left[\mu\,(\phi\phi) + (1-\mu)\phi\phi^{\circ}\right] \qquad (2\text{-}15)$$

Where $\phi\phi$ was the right hand side of equation (2-14), the superscript $^{\circ}$ denoted the values at the beginning of the time step and $0 \leq \mu \leq 1.0$. Introducing the equilibrium relationship $Y_{j,i} = K_{j,i}X_{j,i}$ where $K_{j,i} = f(T_j)$; into equation (2-15) yields

$$Aa\,X_{j-1,i} + Bb\,X_{j,i} + Cc\,X_{j+1,i} = Dd \qquad (2\text{-}16)$$

Where

$$\begin{aligned} Aa &= -\mu\,V_{j-1}\,K_{j-1,i} \\ Bb &= \frac{U_j}{\Delta t} + \mu\,(V_jK_{j,i} + L_j + W_j) \\ Cc &= -\mu L_{j+1} \\ Dd &= (1-\mu)\phi\phi^{\circ} + \frac{U_jX^{\circ}_{j,i}}{\Delta t} + \mu FX_{F,i} \end{aligned} \qquad (2\text{-}16a)$$

For each component i, this equation (16) was combined with similar equation for each tray and solved simultaneously for the composition of each component at the new time level. The temperature, and the equilibrium constant $K_{j,i}$ were held constant at their initial values through the time step. The temperature was then computed again at the end of the time step and a new equilibrium constant was evaluated

for the next time step.

For the case where $\mu = 0$, all terms at the new time level disappeared from equation (2-15) and this became the explicit method and was identical to the relaxation method proposed by Rose et al. (6). Rosenbrock (51) used $\mu = \frac{1}{2}$ to solve the unsteady state problem and noted that the error in the time step was of the order of Δt^3. For $\mu = 1.0$, the error was of the order of Δt^2 but the process was better for finding the steady state solution. This implicit method was further developed by Waggoner (44) and Holland (1).

The model developed by Waggoner (44) was more general than that used by Rosenbrock. The internal vapor and liquid flow rates were allowed to vary during the transient period in order to satisfy the tray enthalpy balance. The calculation procedure developed by Waggoner was

a) Assumed values of total flow rates and temperatures at the new time level. For the first trial, these were assumed to be the values at the old time level.

b) Computed the compositions at the new time level by solving equation (2-16) simultaneously for all trays.

c) Using the calculated compositions, determined the corrected temperatures at the new time level.

d) From the enthalpy balance around the tray, determined the corrected total flow rates at the new time

level.

e) Repeated step 1-4 until a specified convergence criterion has been satisfied.

In 1966, Tetlow (56) (57) generalized the model of Waggoner by allowing variable tray holdup. He also introduced a liquid downcomer holdup between the tray holdups.

In 1967, Picou (58) further generalized Tetlow's model by dividing the tray holdup into three distinct parts. They were:

a) A liquid holdup in which mass and heat transfer took place.

b) A liquid holdup in the downcomer.

c) A vapor holdup between trays.

Picou applied the modified model to a complex column and claimed that improved results were obtained.

The most important feature of this method is, when $\frac{1}{2} \leq \mu \leq 1.0$, the solutions always converge to the right steady state no matter how large the step size. This has been shown in the text by Holland (1).

C) CONTROL OF A DISTILLATION COLUMN

1) FEEDBACK AND FEEDFORWARD CONTROLS.

An excellent source of feedback control schemes is given by Harriott (59). Hoffman (60) presented a good discussion on certain types of feedback control system in 1963. Shinskey (61) and Buckley (62) have broadened the scope by including the feedforward control schemes in their text books.

The feedback control schemes reported in literature are satisfactory for single terminal composition control. For example, the top product composition is regulated by the reflux rate, and the bottom product composition is regulated by the reboiler steam rate. However, the simultaneous control of top and bottom product compositions, using reflux and reboiler steam flows as the manipulated variables proved to be particular difficult because of the coupling effect inherent in a distillation column. This will cause the two control loops to interact with each other, thereby deteriorating the performances of both control loops. This has been clearly shown in Rosenbrock's work (63). In 1962, Rosenbrock (63) proposed a control scheme with the top product flow rate and the reboiler steam rate as the two manipulated variables. In this proposal, the top loop controlled the sum of the two measured compositions and the bottom loop controlled the

difference between the two measured compositions. Extensive study in this respect had been done by Rijnsdorp (64) (65). He proposed to use a ratio controller to control the ratio of the overhead vapor rate to the reflux flow rate with the setpoint of the ratio controller being adjusted by the signal from the overhead composition controller. This scheme thus caused the overhead composition to be insensitive to variations in the reboiler flow rate which caused the overhead vapor rate to change.

The studies of Rijnsdorp (64) (65) and Davidson (66) outline the conditions which give rise to interaction between the two composition control loops. Davidson (66) has gone even further to study the interaction between pressure and temperature control loops.

In 1970, Luyben (67) using digital simulation, considered two different approaches to handle the interaction of these two control loops. He studied the ideal decoupling where the closed loop response was the same as it would be if the other loop were under manual control, and "simplified decoupling" which involved the use of non-interacting compensators to isolate the two loops.

Wood and Berry (68) applied the Rijnsdorp ratio control scheme and the non-interacting control system which employed non-interacting compensators to a binary

distillation column of pilot scale. These authors found that the control performance obtained by using the non-interacting control was only marginally better than that using the ratio control scheme.

Feedforward control is another method to achieve the dual quality control of a distillation column. In 1962 and 1964, Lupfer, Parson and Johnson (69) (70) proposed the feedforward control scheme in which the column dynamics were represented by simple transfer functions and the manipulated variables were the bottom product rate and the reflux rate. The bottom product rate was used instead of the reboiler steam rate because the authors claimed that it was easier to predict the bottom product rate than the reboiler steam rate. Finally, they combined a feedback loop with this feedforward scheme to account for the inaccuracy of the dynamic model. These control schemes were implemented in a debutanizing operation at Phillips Petroleum Company and a substancial saving was reported. Weiss, Archer and Burt (71) reportedly used a computer to search for the best control of a deisobutanizer. They defined an operating profit rate as the following:

$$P = (V_d - V_F)\, D - V_s S \qquad (2\text{-}17)$$

Where V_d, V_F, Vs denotes the unit values of the distillate, feed and the reboiler steam rates. S is the steam rate to the reboiler and D is the distillate

rate. The pertinent informations were fed into the computer. The profit rate P was maximized at all instances by manipulating the top product flow rate and the reboiler steam rate. The beauty of this scheme is that no dynamic model is needed. However, this type of search process is very time consuming.

Luyben and Gerster (72) utilized a linearized dynamic model to construct the feedforward control scheme for a binary distillation column under disturbances in feed rate and in feed composition. Their manipulated variables were the vapor rate and the reflux rate to the column. The performances of the controllers were determined by analog simulation as well as experimental work.

Shinskey (73) also proposed a feedforward scheme with the reboiler steam rate and top product rate as the two manipulated variables for dual quality control.

Cadman, Rothfus and Kermode (74) synthesized a feedforward control scheme for a multicomponent distillation column. The reflux and the vapor rate to the column were the two manipulated variables. The transfer functions relating all the pertinent variables were obtained from the basic material and energy balance of the trays.

Utilizing a digital simulation, Luyben (75) investigated the feedforward control scheme for a 20-tray binary

column in 1969. Non-linear steady state and linear dynamic synthesis techniques were used to design the required feedforward controllers. Again, the vapor rate and the reflux rate to the column were the manipulated variables. Luyben concluded that the combined feedforward and feedback control scheme was more effective than either feedback or feedforward control alone.

Shunta and Luyben (109) studied the dynamics of a binary distillation column using non-linear and linear dynamic models. This work also included the effect of various temperature control tray locations on the operation of the column. These same authors (110) have reported the design of sample-data controllers for a theoretical binary distillation column using computer simulation techniques. They have proposed a dual control scheme which compensates for both set point and load disturbances. It was claimed that this scheme was superior to the continuous control particularly when both disturbances were encountered at once. They also pointed out that different sets of compensators were needed to account for the different types of disturbances because these compensators were only good for the particular disturbance for which they were designed.

Clear (111) studied the dynamic modelling and the control aspects of a 6 and a 7 tray binary distillation

column which separated n-butane and n-pentane. Tramontin (112) proposed a non-linear model and four linear models with various degrees of complexity to represent a ternary column with five trays. He concluded that the linear models were applicable for the design of conventional feedback and feedforward control systems. He also claimed that the non-ideal mixtures produce non-linear effects which give rise to control instabilities.

Toijala et al. (113) studied the dynamic responses of a 4- component 10- tray column and a 3-component 30-tray column by introducing step sinsoidal and stochastic changes in the reflux rate, vapor rate, feed enthalpy and composition. These authors (114) also studied the two-point feedback control of the distillation column by digital computer simulation.

In recent years, feedforward control schemes have become quite popular for distillation column control. The merit of this control scheme is that it can anticipate the necessary control action before the disturbance enters the column. The success of this type of control scheme depends on the accuracy of the dynamic model.

2) OPTIMAL CONTROLLER

State space representation of the dynamic system has played an important role in modern control practice. The original statement of the problem in this form seem to have been made by Kalman, Bertram and Keopecke (76) (77).

The major advantage of the state space technique is, it presents the design of controllers by analytical procedure as opposed to the classical methods like the Bode diagram and the root locus method (78) which are graphical techniques.

Kalman and Bertram (77) proposed a method of designing state space controllers for digital systems which could bring a system from the initial upset state to the equilibrium state with all state variables equal to zero. At the same time, this controller would minimize a quadratic function formed by the state variables. Only one manipulated variable was considered in this study. The basic work was extented by Kalman and Keopecke (79) in 1959 to multi-variable systems having more than one manipulated variable and output. These two articles by Kalman et al. formed the basis for the design of optimal controllers. They are revolutionary because the state space equations are used and an analytical approach to controller design is provided. Later, Kalman's theory was extended by

authors like Tou (80), who in his text, has summarized the state of optimal digital controller design.

Kalman's digital controller design technique appeared in chemical engineering literature as early as 1961. Amundson, Aris, Kalman and Lapidus (81) prepared a set of lecture notes on applied mathematics which included the design of a controller for a train of a six stages counter-current extraction unit. This extraction train was to be driven from an initial equilibrium state to a new desired equilibrium state. Lapidus later presented this example in his textbook. Lapidus, Shapiro, and Shapiro and Stillman (82) then published a paper concerning the application of the so-called Kalman controller to a six-tray absorber. Then, Lapidus extended this example to examine the effects of various performance criterions on system behavior. In 1967, Lapidus and Luus published a textbook (83). In this book, they expanded the treatment to non-linear systems such as the continuous stirred tank reactor. This book has a very thorough treatment of optimal control theory and is a good source of reference.

Heidmann (84) applied the Kalman optimal controller to a four-tray distillation column. The dynamics of the column was represented by transfer functions which were obtained from experimental frequency response data.

Rafal (85) studied the optimal control of two non-linear processes in which physical limits on the set of admissible controls were presented. These two processes were: a stirred tank reactor and a three-tray distillation column. He claimed that the boundary search technique for determining the optimal control was very effective and the method was also good for on-line direct digital computer control of chemical processes.

Brosilow and Naphtali (86) investigated the optimal on-off control of the same extraction train used by Lapidus. Later he and Handley (87) applied the same controller to a 15-tray binary rectifying column under feed composition disturbances. This rectifying column was represented by a linearized dynamic model. In 1972, Merluzzi and Brosilow (88) applied the same on-off controller to a distillation column. They claimed the results were successful. Zahradnik, Archer and Rothfus (89) used a simple mathematical model--a first order system or first order system with time lag --to represent the dynamics of a binary distillation column. They defined a profit criterion which was a function of the manipulated variables, the reboiler steam rate and the top product rate, and the top product composition and feed stream specification. This criterion was optimized by manipulation of the

reboiler steam and the top product rates. The techniques of dynamic programming and maximum principle were applied to obtain the optimal control.

McNeill and Sacks (90) of Monsanto Co. reported that they applied a "bang-bang" control scheme to a ethylbenzene column that had a history of unsteady operation and slow response. The result was an increase in column throughput by 10%, product recovery by 4% and essentially eliminated off-specific product. The authors claimed the Pontryagin maximum principle has shown that true optimal control can be achieved only by using "bang-bang" control. The fact is, when the control enters the Hamiltonian linearly and is bound, the resulting optimal control is the "bang-bang" type (91).

In 1972, Hu and Ramirez (92) synthesized optimal controllers for a binary distillation column with five trays. The column was represented by a linearized model. The authors showed that the results were also satisfactory when tested on a non-linear dynamic model.

In 1971, Pang and Johnson (93) have extended the Kalman controller design to include integral constraint on the state. These authors used a laboratory scale extraction column as an example. The integral constraint was introduced because the average

extraction concentration was to be kept at some constant value during the change-over period. This type of problem is referred to as the isoperimetric problem in the classical calculus of variations (94) (problem of Dido). The original problem was to find a maximum area that a curve of fixed length could encircle.

Whatley (115) studied the suboptimal control scheme for a batch distillation with a multicomponent system. The distillation process was represented by a linear time-varying state equation and the suboptimal policy was obtained for manipulation of the reflux rate. Mayur (116) also studied the optimal control policy of binary and ternary batch distillation processes. The optimal control policy he used was a time-optimal reflux policy. Eschenbacher (117) using a first-order linear system to represent the binary distillation column, designed an optimal controller using the technique of dynamic programming. This optimal controller was implemented with the digital computer to control the column. He claimed that noise was a major problem in the direct digital computer control and in spite of the noise problem, the optimal controller was superior than the conventional feedback control.

Most of the optimal controller synthesis for distillation and extraction processes were based on linear mathematical models. Because of the large number of variables involved in these processes, synthesizing the optimal controller with non-linear dynamic model is very difficult if not impossible.

CHAPTER III

FORMULATION AND SOLUTION OF THE STEADY STATE MODEL

A non-linear steady state model was developed and subsequently solved. With this model, the starting and the final steady state conditions of the distillation column were generated and used in the dynamic calculation. The matrix method of Amundson and Pontinen (5) and the constant composition method of Holland (7) were employed to set up and solve the component material balance and the enthalpy balance equations.

A) FORMULATION OF THE COMPONENT MATERIAL BALANCE EQUATIONS

With the advent of high speed digital computers, rigorous calculation methods for steady state distillation operations have been widely used by chemical engineers. One of them is the matrix method proposed by Amundson and Pontinen (5). In this method, the steady state distillation problems have been described as a system of simultaneous algebraic equations in the liquid composition in which a set of parameters, the tray temperature must be fixed such that the over-all meterial and energy balance will not be violated. Because of the availability of high speed, large memory capacity digital computers, these equations can be solved directly, with the help of matrix algebra, to give the steady state tray compositions of the distillation column.

The iteration variables that must be assumed to begin the computation are the temperature and the vaporrate profiles. Figure 1 shows a schematic diagram of the binary distillation column under study.

A component i balance around the reboiler yields the following equations

$$-(B + V_1 K_{1,i}) X_{1,i} + (V_1 + B) X_{2,i} = 0 \qquad (3\text{-}1)$$

Moving up to the stripping section, the equation which represents a component material balance around any tray in this section is

$$(V_{j-1} K_{j-1,i}) X_{j-1,i} - (V_{j-1} + B + V_j K_{j,i}) X_{j,i} + (V_j + B) X_{j+1,i} = 0. \qquad (3\text{-}2)$$

For the feed tray, the equation is

$$(V_{f-1} K_{f-1,i}) X_{f-1,i} - (V_{f-1} + B + V_f K_{f,i}) X_{f,i} + (V_f - D) X_{f+1,i} = -FX_{F,i} \qquad (3\text{-}3)$$

For any tray in the rectifying section, the equation is

$$(V_{j-1} K_{j-1,i}) X_{j-1,i} - (V_{j-1} - D + V_j K_{j,i}) X_{j,i} + (V_j - D) X_{j+1,i} = 0. \qquad (3\text{-}4)$$

At the top of the column, there is a total condenser. The equation that represents it, is as follows:

$$(V_{N-1} K_{N-1,i}) X_{N-1,i} - V_{N-1} X_{N,i} = 0 \qquad (3\text{-}5)$$

Equations (3-1) to (3-5) can be grouped together to yield the following vector-matrix form.

$$\underline{AA}\ \underline{X} = \underline{BB} \qquad (3\text{-}6)$$

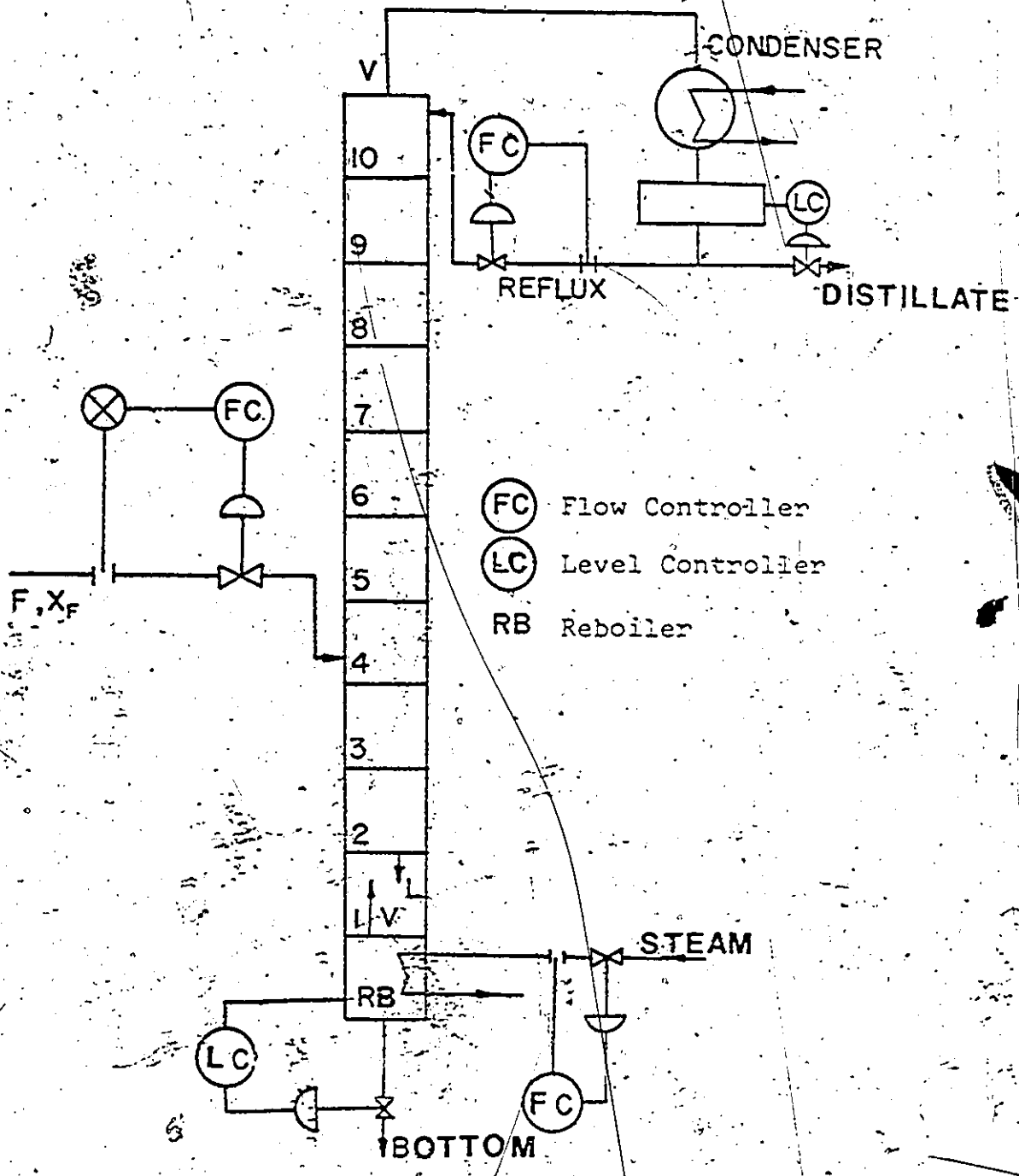


Figure 1. Sketch of the binary distillation column.

Where matrix $\underline{AA}$ is an N x N coefficient matrix and $\underline{X}$ is a N-dimensional vector representing the liquid tray composition and $\underline{BB}$ is a N-dimensional vector which describes the feed condition.

Equation (3-6) can be solved by inverting the $\underline{AA}$ matrix and pre-multiplying the inverse to the $\underline{BB}$ vector, provided that matrix $\underline{AA}$ is non-singular.

$$\underline{X} = \underline{AA}^{-1}\ \underline{BB} \qquad (3\text{-}7)$$

$$\underline{X} = \begin{bmatrix} X_{1,i} \\ X_{2,i} \\ \cdot \\ \cdot \\ \cdot \\ X_{N-1,i} \\ X_{N,i} \end{bmatrix} \qquad \underline{BB} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ FX_{F,i} \\ 0 \\ \cdot \\ 0 \end{bmatrix}$$

$$\underline{AA} = \begin{bmatrix} -(B+V_1K_{1,i}) & (V_1+B) & & & \\ (V_1K_{1,i}) & -(V_1-D+V_2K_{2,i}) & (V_2+B) & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & & -V_{N-1} \end{bmatrix}$$

5) CONVERGENCE PROCEDURE

This method does not provide any correction for the calculated tray concentrations to better the values before they are used to evaluate the new temperature profile. The only thing that Amundson and Pontinen suggested was to change any negative number to zero and then normalize the resulting tray compositions so that $\sum_{i=1}^{c} X_i = 1.0$ for each tray. These normalized values will then be used to calculate the new temperature profile by the non-ideal bubble point calculation.

C) DETERMINATION OF THE NEW TEMPERATURE PROFILE

The bubble point calculation was used to determine the new temperature profile. For this calculation, the vapor phase tray compositions have to be known. Since the methanol-water system is non-ideal, the ideal equilibrium constants, evaluated from the Antoine equation had to be modified before they were used. The activity coefficients γ_1 and γ_2, which were calculated from the Van Laar third order equations, were employed to modify the ideal equilibrium constants. Physical data concerning the non-ideal bubble point calculation can be found in appendices E and F.

The Newton's approximation method used by Amundson and Pontinen (3) was not applicable in this case. The methanol-water system, being non-ideal, did not permit a direct evaluation of the derivatives required in the Newton's method. Therefore, an alternative method--the false-position method--was used.

The false-position method is an interpolation method which approximates the function by a straight line. Since it is an interpolation process, two values are needed to start the iteration process. Therefore, besides the temperature of the previous trial, another temperature is needed to obtain the next trial value.

At a given set of $X_{i,j}$'s, the vapor compositions are assumed to be a function of temperature. Let

$$T_j^2 = T_j^1 + \Delta T \qquad (3\text{-}7a)$$

where ΔT is a small pre-set quantity. T_j^1 is the temperature of the previous trial. Evaluating each $Y_{i,j}$ at these two temperatures, the next temperature T_j^3 is obtained using the following equation:

$$T_j^3 = \frac{\Delta T \left(1.0 - \sum_{i=1}^{c} Y^1_{i,j}\right)}{\sum_{i=1}^{c} Y^2_{i,j} - \sum_{i=1}^{c} Y^1_{i,j}} + T_j^1 \qquad (3\text{-}7b)$$

Where $Y^1_{i,j} = K_{i,j}\,(T_j^1)\,X_{i,j}$

$Y^2_{i,j} = K_{i,j}\,(T_j^2)\,X_{i,j}$

The next best value of the tray temperature; T_j^4 is calculated by replacing T_j^1 with T_j^3, subsequent values of T_j^n are attained in the same fashion until $\sum_{i=1}^{c} Y^n_{i,j} = 1.0$, at which time, the corrected tray temperature is obtained. This set of new tray temperatures, the enthalpy and the total material balance, are used to determine the new liquid and vapor flow rate profiles of the distillation column.

D) FORMULATION OF THE ENTHALPY BALANCE EQUATIONS

The enthalpy balance equations can also be set up in the vector-matrix form with the vapor flow rates as the iteration variables. A careful survey of the available methods suggests that Holland's constant composition method (7) for enthalpy balance calculation is considerably better than the others. Holland's method has been adopted in this calculation to obtain the new vapor and the new liquid flow rates for this steady state calculation.

The vapor leaving tray j-1 of the rectifying section of the column consists of the distillate and the liquid entering tray j-1 from tray j. Thus, the total enthalpy of the vapor stream can be expressed in the following forms:

$$V_{j-1}H_{j-1} = \sum_{i=1}^{C} H_{j-1,i}V_{j-1,i} = \sum_{i=1}^{C} H_{j-1,i}\,(l_{j,i} + d_i) \qquad (3\text{-}8)$$

$$V_{j-1}H_{j-1} = L_jH(X_j)_{j-1} + DH(X_D)_{j-1} \qquad (3\text{-}9)$$

where

$$H(X_j)_{j-1} = \sum_{i=1}^{C} H_{j-1,i}X_{j,i}$$

$$H(X_D)_{j-1} = \sum_{i=1}^{C} H_{j-1,i}X_{D,i}$$

An enthalpy balance enclosing tray j and the top of the column yields the following equation:

$$V_{j-1}H_{j-1} = L_jh_j + DH_D + Q_c \qquad (3\text{-}10)$$

Combining equations (3-9) and (3-10), the equation for liquid flow rate is obtained

$$L_j = \frac{D\left[H_D - H(X_D)_{j-1}\right] + Q_c}{\left[H(X_j)_{j-1} - h_j\right]} \qquad (3\text{-}11)$$

The equation for vapor flow rates in the rectifying section is

$$V_{j-1} = L_j + D. \qquad (3\text{-}12)$$

The equation for the condenser duty is

$$Q_c = L_N\left[H(X_N)_{N-1} - h_N\right] + D\left[H(X_D)_{N-1} - H_D\right] \qquad (3\text{-}13)$$

For the feed tray, the equation is

$$L_f = \frac{D\left[H_D - H(X_D)_{f-1}\right] + Q_c - Fh_f}{\left[H(X_f)_{f-1} - h_f\right]} \qquad (3\text{-}14)$$

For the distillation column under study, the feed entered the column as a liquid, therefore, no vapor feed term appeared in equation (3-14).

Applying the same technique to the stripping section of the column, the following equations for the liquid and the vapor flow rates result:

$$L_{j+1} = \frac{B\left[H(X_B)_j - H_B\right] + Q_R}{\left[H(X_{j+1})_j - h_{j+1}\right]} \qquad (3\text{-}15)$$

$$V_j = L_{j+1} - B \qquad (3\text{-}16)$$

Where

$$H(X_B)_j = \sum_{i=1}^{C} H_{j,i} X_{B,i}$$

$$H(X_{j+1})_j = \sum_{i=1}^{C} H_{j,i} X_{j+1,i}$$

The reboiler heat duty can be calculated from the overall enthalpy balance equation.

The merit of this constant composition method is in its great stability. A careful examination of equations (3-11), (3-14) and (3-15) reveals the fact that the denominators of these equations are of the order of magnitude of the latent heat of vaporization, whereas the corresponding denominators of the conventional calculation processes (7) take on a wide range of values which include positive and negative numbers. This will tend to decrease the stability of the calculation and may also result in negative flow rates.

The new set of vapor flow rates, together with the new temperature profile were used to renew the AA matrix, and then new tray compositions were calculated. This process should be carried on until the tray compositions from two successive trials have identical values. Then the steady state solution is reached.

E) CALCULATION PROCEDURE

The calculation procedure can be summarized as follows

1) Assume a vapor flow profile and a temperature profile of the column.

2) Calculate the non-ideal equilibrium constants with the Antoine equation and the Van Laar third order equations.

3) Calculate the elements of the matrix AA and the vector BB.

4) Invert AA and premultiply this inverse to the vector BB.

5) Check whether the difference between the old and the new tray compositions are smaller than the error criterion. If it is, the required steady state is obtained. If not, proceed to step 6.

6) Normalize all the component tray compositions

$$X_{j,i} = \frac{X_{j,i}}{\sum_{i=1}^{C} X_{j,i}}$$

7) Compute the new temperature profile based on the normalized tray compositions with the non-ideal bubble point calculation.

8) Calculate the enthalpies of the vapor and the liquid streams H_j's and h_j's.

9) Calculate the new vapor and the liquid flow rates and return to step 2 until the required criterion is attained.

CHAPTER IV

CALCULATION OF THE STEADY STATE TWO-POINT CONTROL SCHEMES

Utilizing the steady state mathematical model developed in Chapter III, the performance of the two-point control schemes were evaluated when the distillation column was under load or set point disturbance. The steady state operating conditions of the column were also computed in these calculations. These steady state conditions would be used in the dynamic optimization of the two-point control systems.

A) GENERAL DESCRIPTION OF THE TWO-POINT CONTROL SCHEMES

The control schemes shown in figures 2 and 3 are the ones that were used in this study. The pairs of manipulated variables were either the top product withdrawal rate and the steam rate to the reboiler, or the reflux rate and the bottom product withdrawal rate. These control schemes were of feedforward configuration. The manipulated variables which were directly supervised by the computer were under flow control. The other pair was manipulated by the level controller which maintained the level of the reboiler and the condenser.

When a disturbance entered the column through the feed stream, or a change in product composition was needed, the signal would be transfered to the computer. The necessary control actions were calculated based on the mathematical model and sent to the pertinent controllers.

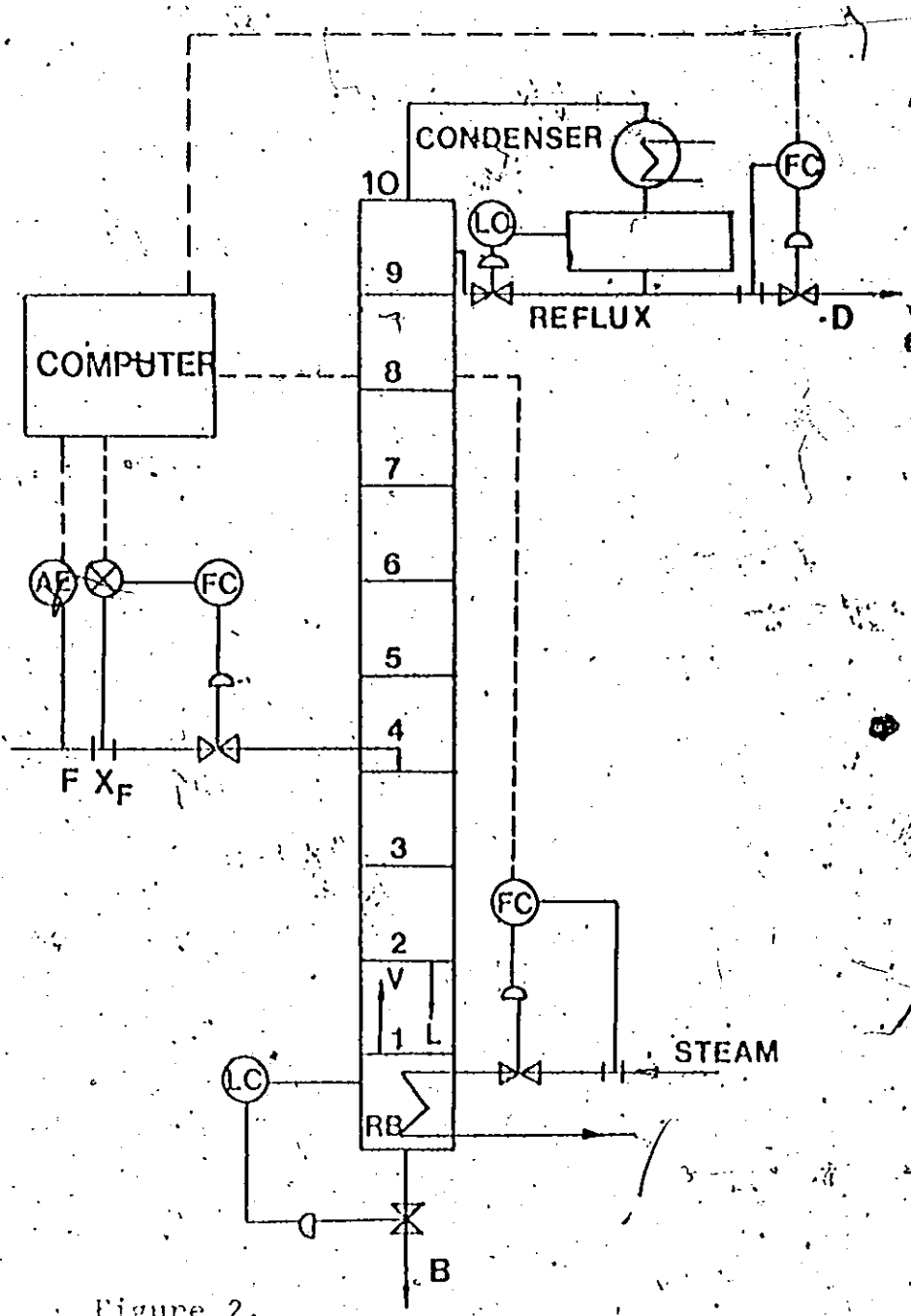


Figure 2.

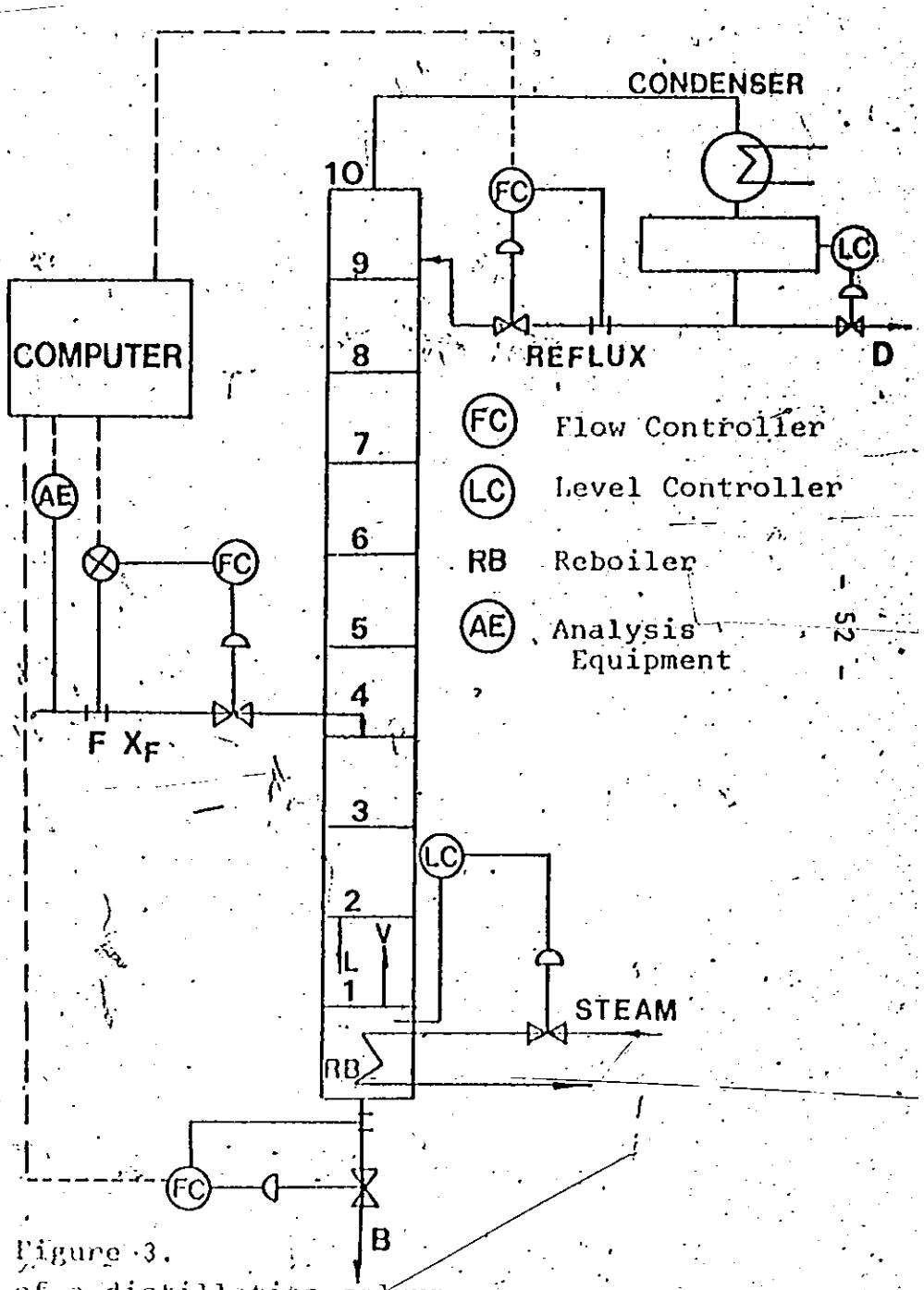


Figure 3.

Sketch of feedforward two-point control system of a distillation column

B) COMPUTATION OF THE MANIPULATED VARIABLES

Among the four manipulated variables, two can readily be calculated from the over-all material balance of the column. They are the top and the bottom product withdrawal rates. Total and component over-all material balances of the column result in the following equations:

$$F = D + B, \qquad (4\text{-}1)$$

$$X_F F = X_D D + X_B B. \qquad (4\text{-}2)$$

Combination of equations (4-1) and (4-2) yields:

$$D = \frac{F\,(X_F - X_B)}{(X_D - X_B)} \qquad (4\text{-}3)$$

Since the feed condition, the specifications of the top and bottom product compositions have all been fixed, the top product rate can be computed from equation (4-3). The bottom product rate can then be obtained from equation (4-1).

The calculation of the two-point control now reduces to the search of one control variable, the reboiler steam rate or the reflux rate, depending on which control scheme is employed. A careful review of the situation shows that both the reboiler steam rate and the reflux rate can be obtained in just one calculation. Since the feed condition, the top and the bottom product compositions and the external flow rates have all been fixed, there is only one set of reboiler steam rate and reflux rate that will fulfill the requirement. Also, the reboiler steam rate and the reflux

rate are related to each other through the enthalpy balance. Therefore, one calculation is enough to evaluate the two pairs of control variables for the two different control schemes. This of course is based on the assumption that all tray efficiencies are constant. Only one manipulated variable among the four has to be found through the search process. In this study, the sole search variable was the reboiler steam rate and the reflux rate was calculated from the enthalpy balance.

Four different search procedures have been used to locate the desired reboiler steam rate. Also, this search problem was used as a criterion to evaluate the efficiency of the four search methods. Among the four, two were one-dimensional search methods, the other two were multi-dimensional search procedures.

1) The dichotomous search (95)

This is a one-dimensional search method. In this process, the step size is reduced by half for each iteration. It is a simple but effective method and the only drawback is that it is fairly time consuming.

2) The false-position method (96)

This is also a one-dimensional search method. It is similar to Newton's method except that it does not require the knowledge of the derivative. In this procedure, the non-linear equation is approximated by a straight line through interpolation. This is a

better method than the Newton's method because it works in cases where Newton's method has failed to obtain the solution (96).

3) The pattern search method (95)

This is a multi-dimensional search method. It was proposed by Hooke and Jeeves (95) and later modified by the same authors. This technique is based on the hopeful conjecture that any adjustment of the parameters which have been successful in early trials will be worth trying again. One of the beauties of this method is, its computation time increases with the first power of the dimension, whereas, for the classical search techniques, the computation time increases with the cube power of the dimension. Thus, this process is good for high dimension systems.

4) The method by Marquardt (97)

This is a combination of the steepest descent and the Newton's method. The steepest descent method is used at the beginning of the search, as the minimum is approached, the calculation will switch to the fast convergence of Newton's method.

This method was originally set up for solving system of non-linear algebraic equations and non-linear regression of multi-dimension systems.

From this study, it was observed that this method rapidly converged to the solution. The solution was usually obtained after a few trials. Derivative information is necessary in this search process and they were evaluated numerically. It was found that the convergence did depend highly on the accuracy of the evaluation of the derivatives.

These four search methods were tried and the false-position method was found to be the most efficient one. Therefore, most of the calculations were done by employing the false-position method.

C) ITERATION OF THE CONTROL VARIABLE USING THE FALSE-POSITION METHOD

As it was mentioned in the previous chapter, the false-position algorithm is an interpolation method and therefore required two points to begin the iteration process. These two points are obtained by assuming two steam rates to the reboiler. Steady state calculations are then carried out with these two values yielding the two corresponding top product compositions. Let the steam rates to the reboiler be Q_R^1 and Q_R^2, the top product compositions be X_D^1 and X_D^2; and

$$X_D^1 - X_{D,\ \text{desired}} = \Delta X_D^1 \qquad (4\text{-}4)$$

$$X_D^2 - X_{D,\ \text{desired}} = \Delta X_D^2 \qquad (4\text{-}5)$$

The steam rate to the reboiler which makes $\Delta X_D^n = 0$ is the required one. Implementation of false-position method results in an equation similar to equation (3-7b). The new steam rate to the reboiler Q_R^3 which is given by the equation (4-6) is the next best trial value:

$$Q_R^3 = \frac{\Delta Q_R (0 - \Delta X_D^1)}{\Delta X_D^2 - \Delta X_D^1} + Q_R^1 \qquad (4\text{-}6)$$

ΔQ_R is a small pre-set quantity.

Iteration of equation (4-6) stops when Q_R^n results in the value of $\Delta X_D^n = 0$.

D) CALCULATION PROCEDURE

The calculation procedure for the two-point control schemes can be summarized in the following steps.

1) Assume temperature and vapor flow profiles.

2) Calculate D and R from equations (4-1), (4-2) and (4-3).

3) Assume a value for the steam rate to the reboiler.

4) Calculate the component tray composition with the matrix method.

5) Check the differences of the values of the tray compositions of two successive trials. If the differences are smaller than the error criterion, proceed to step 6. If not, calculate new temperature and flow profiles of the column and return to step 4.

6) Check if X_D is equal to the desired value. If it is, the calculation is ended. If not, obtain a new value of the reboiler steam rate by the false-position method and return to step 4 until X_D reaches the desired value.

CHAPTER V

FORMULATION AND SOLUTION OF THE NON-LINEAR DYNAMIC MODEL

A non-linear dynamic model of the binary distillation column was developed and subsequently solved by various numerical methods. This mathematical model was used to predict the transient behaviors of the distillation column under disturbances in the feed stream, the reflux, and the heat input to the reboiler.

A) GENERAL DESCRIPTION OF A DYNAMIC MODEL FOR A DISTILLATION COLUMN

The unsteady state model that describes the operation of a distillation column is based on the law of conservation of mass and energy. The law of conservation of mass (energy) of a system (system can be used to represent a single tray or the whole distillation column) between time period t_n and $t_n + \Delta t$ can be put in the following form:

Input of material (energy) to the system during Δt	−	Output of material (energy) from the system during Δt	=	Accumulation of material within (energy) the system during Δt

(5-1)

The accumulation term is defined as follows

$$\left[\begin{array}{l}\text{Accumulation of}\\ \text{material within}\\ \text{(energy)}\\ \text{the system during}\\ \Delta t\end{array}\right] = \left[\begin{array}{l}\text{Amount of material}\\ \text{(energy)}\\ \text{in the system at}\\ \text{time } t_n + \Delta t\end{array}\right] - \left[\begin{array}{l}\text{Amount of material}\\ \text{(energy)}\\ \text{in the system at}\\ \text{time } t_n\end{array}\right] \qquad (5\text{-}2)$$

For a system at unsteady state operation, equations (5-1) and (5-2) can be put into the following form:

$$\int_{t_n}^{t_n+\Delta t} \left[\left(\begin{array}{l}\text{Input of material}\\ \text{(energy)}\\ \text{per unit time}\end{array}\right) - \left(\begin{array}{l}\text{Output of material}\\ \text{(energy)}\\ \text{per unit time}\end{array}\right)\right] dt$$

$$= \left.\begin{array}{l}\text{Amount of}\\ \text{material}\\ \text{(energy)}\end{array}\right|_{t_n+\Delta t} - \left.\begin{array}{l}\text{Amount of}\\ \text{material}\\ \text{(energy)}\end{array}\right|_{t_n} \qquad (5\text{-}3)$$

Applying equation (5-3) to a distillation tray at unsteady state operation, the total material balance can be represented by the following equation:

$$\int_{t_n}^{t_n+\Delta t} \left[(V_{j-1} + L_{j+1}) - (V_j + L_j)\right] dt = (U_j + U_j^V)\Big|_{t_n+\Delta t} - (U_j + U_j^V)\Big|_{t_n} \qquad (5\text{-}4)$$

By applying the mean-value theorem of differential and integral calculus (98), equation (5-4) is transformed into:

$$V_j + L_{j-1} - V_j - L_{j+1} = \frac{d\left[U_j + U_j^V\right]}{dt} \qquad (5\text{-}5)$$

For component material balance of a distillation tray at transient operation, the equation is:

$$\int_{t_n}^{t_n+\Delta t} (v_{j-1,i} + l_{j+1,i} - v_{j,i} - l_{j,i})\, dt = (u^l_{j,i} + u^v_{j,i})\Big|_{t_n+\Delta t} - (u^l_{j,i} + u^v_{j,i})\Big|_{t_n} \qquad (5\text{-}6)$$

An enthalpy balance of the tray is essential to the determination of the liquid and the vapor flow rates that leave and enter the tray. The equation which represents this balance is as follows:

$$\int_{t_n}^{t_n+\Delta t} (V_{j-1}H_{j-1} + L_{j+1}h_{j+1} - V_jH_j - L_jh_j)\, dt = (U_jh_j + U^v_jH_j)\Big|_{t_n+\Delta t} - (U_jh_j + U^v_jH_j)\Big|_{t_n} \qquad (5\text{-}7)$$

Since binary distillation is a vapor-liquid contacting operation, the equilibrium relationship between the two phases cannot be neglected. It can be expressed in the following form:

$$Y_{j,i} = E_{j,i}\, \gamma_{j,i} K_{j,i} X_{j,i} \qquad (5\text{-}8)$$

Where

$$\sum_{i=1}^{c} Y_{j,i} = 1.0 \qquad (5\text{-}9)$$

It follows from equations (5-8) and (5-9) that the liquid temperature leaving tray j at any time t is that $T_j > 0$, which makes $f(T_j) = 0$ where

$$f(T_j) = \sum_{i=1}^{c} E_{j,i} \gamma_{j,i} K_{j,i} X_{j,i} - 1.0 \qquad (5\text{-}10)$$

Various combinations of equation (5-4) to (5-7), together with some form of equilibrium relationship have been used to study the transient behavior of continuous distillation process. Because of the non-linear nature as well as the large number of equations, analytical solutions are almost impossible unless these equations are highly simplified by making a large number of assumptions. With the availability of high speed large scale digital computers, numerical solutions of these equations can easily be obtained. Still, some kind of assumptions are necessary in order to simplify the calculation processes. The validity of the assumptions can be reflected from the discrepancy between the experimental and the calculated results.

B) ASSUMPTIONS FOR DEVELOPING THE NON-LINEAR MODEL

The following assumptions have been made in order to develop the dynamic equations.

1) Adiabatic operation

2) Negligible vapor holdup on all trays

3) Liquid tray holdups are constant with respect to time for all trays.

4) Tray efficiencies are 100%

5) Ideal solution is assumed during the calculation of enthalpies.

6) Equilibrium constants are function of temperature and pressure alone.

7) Reflux enters the column at saturation temperature.

In reality, the efficiencies of the tray can seldom be exactly 100%. This assumption was made in order to simplify the dynamic calculation. A large number of numerical calculations will be involved if the actual transient efficiencies have to be calculated.

The assumptions that the reflux entered the column at saturation temperature was made in order to make the calculation simpler. From experimental observation, the reflux temperature was constant at about 5-10° F below its saturation temperature.

C) BRIEF DESCRIPTION OF THE COMPUTATION METHOD

The combination of the implicit method and the θ-method of convergence developed by Waggoner and Holland (1) for multicomponent system has been adopted in this study to obtain the transient behavior of the distillation column which separated methanol and water. Before going into the detailed numerical procedures, a brief description of the over-all approach of this method is essential.

At some particular time, i.e. $t=t_n$, suppose the temperatures, compositions and flow rates are known for each component throughout the whole column. If the values of these variables are needed at some future time, say $t=t_n+\Delta t$, they can be obtained in the following way. Applying the implicit method to the integral equations mentioned above, these equations will then be converted into a set of algebraic equations for each time period Δt. A careful examination of these algebraic equations reveals that they are very similar in form to the equations which describe the steady state operation. It is conceivable that a method for solving the steady state equations can also be applied in this case. It was proved by Waggoner and Holland (1) that the calculation procedure of Thiele and Geddes for steady state calculation (3) can also be used to solve the component material balance equation for each Δt. In this method, the tray temperatures at the end of the period ($t = t_n + \Delta t$) are used as the independent variables.

For the first trial, a L/V profile has to be assumed. Based on this assumed temperature profile for time $t_n + \Delta t$, the component material balance equations are solved for the component holdups and component flow rates. Then, the θ-method of convergence (4) is used to calculate a new set of component holdups $u_{j,i}$'s and component distillate rate d_i's. These sets of $u_{j,i}$'s and d_i's, have to be in over-all material balance, and at the same time equal to the specified values of the total holdups and distillate rate at time $t_n + \Delta t$. With these corrected rates, the corresponding corrected tray compositions are calculated and used to determine a new temperature profile. Then, the corrected compositions and the corresponding temperatures are used in the enthalpy balance to obtain the liquid (or vapor) rates for the next trial. The total material balance will then be used to find the corresponding vapor (or liquid) flow rates.

For each time period Δt, these calculation procedures are repeated until the differences between the assumed and calculated temperatures are less than some preassigned numbers. As soon as this condition has been achieved, the calculation procedure will be repeated for the next time period. This computation is carried on until the next steady state has been reached.

D) FORMULATION OF THE MATERIAL BALANCE EQUATIONS

A component material balance can be drawn around each tray. Utilizing the assumptions that have been made, the following equation represents any tray (except the feed tray) of the column:

$$\int_{t_n}^{t_n+\Delta t} (v_{j-1,i} + l_{j+1,i} - v_{j,i} - l_{j,i})\, dt = u_{j,i}\Big|_{t_n+\Delta t} - u_{j,i}\Big|_{t_n} \qquad (5\text{-}11)$$

This integral equation can be approximated by using the implicit method to yield the following equation:

$$\mu (v_{j-1,i} + l_{j+1,i} - v_{j,i} - l_{j,i}) + (1.0-\mu)(v^{\circ}_{j-1,i} + l^{\circ}_{j+1,i} - v^{\circ}_{j,i} - l^{\circ}_{j,i}) = \frac{u_{j,i} - u^{\circ}_{j,i}}{\Delta t} \qquad (5\text{-}12)$$

Using the relationships:

$$l_{j,i} = p_{j,i} v_{j,i}$$

$$u_{j,i} = \frac{U_j p_{j,i-1} v_{j,i}}{L_j}$$

Where

$$p_{j,i} = \frac{L_j}{K_{j,i} V_j}$$

the following equation results:

$$\mu (v_{j-1,i} - p_{j,i} v_{j,i} + p_{j+1,i} v_{j+1,i} - v_{j,i}) + (1-\mu)(v^{\circ}_{j-1,i} + p^{\circ}_{j+1,i} v^{\circ}_{j+1,i} - v^{\circ}_{j,i} - p^{\circ}_{j,i} v^{\circ}_{j,i}) = \frac{U^{\circ}_j}{L^{\circ}_j} p^{\circ}_{j,i} v^{\circ}_{j,i} - \frac{U^{\circ}_j}{L^{\circ}_j} p^{\circ}_{j,i} v^{\circ}_{j,i} \qquad (5\text{-}13)$$

Since the unknown $v_{j,i}$'s appear in both sides of equation (5-13), it is said to be of implicit form. Every component i on each tray j and in the reboiler and condenser generates an equation of similar form. Thus for each component, there will be a set of N simultaneous equations of non-linear nature. This set of equations is:

$$\begin{aligned} -hh_{1,i}v_{1,i} + p_{2,i}v_{2,i} &= -\bar{b}_{1,i} \\ v_{j-1,i} - hh_{j,i}v_{j,i} + p_{j+1,i}v_{j+1,i} &= -\bar{b}_{j,i}, \qquad (5\text{-}14) \\ v_{N-1,i} - hh_{N,i}v_{N,i} &= -\bar{b}_{N,i} \end{aligned}$$

where $hh_{j,i} = 1.0 + p_{j,i}(1.0 + \tau\tau_j/\mu)$,

$$\bar{b}_{j,i} = zz_{j,i} + \sigma\left[zz^{\circ}_{j,i} + v^{\circ}_{j-1,i} - (p^{\circ}_{j,i} + 1)\, v^{\circ}_{j,i} + p^{\circ}_{j+1,i}v^{\circ}_{j+1,i}\right] + \tau\tau^{\circ}_j\, p^{\circ}_{j,i}v^{\circ}_{j,i}/\mu,$$

$$zz_{f+1,i} = v_{F,i},$$

$$zz_{f,i} = l_{F,i}$$

and for all other j and i $\quad zz_{j,i} = 0$,

$$\sigma = \frac{1-\mu}{\mu}.$$

$\tau\tau_j = \dfrac{(U_j)}{L_j \Delta t}$ is the mean time required to sweep out the liquid holdup of tray j divided by the increment of time.

$\tau\tau^{\circ}_j = \dfrac{(U^{\circ}_j)}{L_j \Delta t}$ where Δt is the time increment under consideration and not the previous one.

For a total condenser:

$$v_{N,i} = d_i \,,$$

$$l_{N,i} = p_{N,i} v_{N,i} \,,$$

$$p_{N,i} = \frac{L_N}{D} \,.$$

In vector-matrix notation, equation (5-14) has the following form:

$$\begin{bmatrix} -hh_{1,i} & p_{2,i} & 0 & 0 & \\ 1 & -hh_{2,i} & p_{3,i} & & \\ & 1 & -hh_{3,i} & p_{4,i} & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -hh_{N,i} \end{bmatrix} \begin{bmatrix} v_{1,i} \\ v_{2,i} \\ v_{3,i} \\ \vdots \\ v_{N,i} \end{bmatrix} = \begin{bmatrix} -b_{1,i} \\ -b_{2,i} \\ -b_{3,i} \\ \vdots \\ -b_{N,i} \end{bmatrix} \qquad (5\text{-}15)$$

or

$$\bar{A}\,\underline{v} = \bar{b} \qquad (5\text{-}16)$$

E) FORMULATION OF THE ENTHALPY BALANCE EQUATIONS

Based on the law of conservation of energy, the energy balance of each tray can also be written in the form of an integral-difference equation. Take the condenser as an example, the following equation represents the enthalpy balance of the condenser:

$$\int_{t_n}^{t_n+\Delta t}\left[V_{N-1}H_{N-1}-L_Nh_N-DH_D-Q_c\right]dt=U_Nh_N\Big|_{t_n+\Delta t}-U_Nh_N\Big|_{t_n} \qquad (5\text{-}17)$$

Application of the implicit method transforms equation (5-17) into the following expression:

$$V_{N-1}H_{N-1}-L_Nh_N-DH_D-Q_c+\delta\left[V^\circ_{N-1}H^\circ_{N-1}-L^\circ_Nh^\circ_N-D^\circ H^\circ_D-Q^\circ_c\right]$$

$$=\frac{1}{\mu\Delta t}\left[U_Nh_N-U^\circ_Nh^\circ_N\right] \qquad (5\text{-}18)$$

Applying the constant composition method as in the steady state calculation, $V_{N-1}H_{N-1}$ is replaced by:

$$V_{N-1}H_{N-1}=\sum_{i=1}^{C}H_{N-1,i}V_{N-1,i}$$

$$=\sum_{i=1}^{C}H_{N-1,i}\left[l_{N,i}+d_i+\delta(L^\circ_{N,i}+d^\circ_i-V^\circ_{N-1,i})+\frac{1}{\mu\Delta t}(u_{N,i}-u^\circ_{N,i})\right]$$

$$V_{N-1}H_{N-1}=L_NH(X_N)_{N-1}+DH(X_D)_{N-1}+\delta\left[L^\circ_NH(X^\circ_N)_{N-1}+D^\circ H(X^\circ_N)_{N-1}\right.$$

$$\left.-V^\circ_{N-1}H(Y^\circ_{N-1})_{N-1}\right]+\frac{1}{\mu\Delta t}\left[U_NH(X_N)_{N-1}-U^\circ_NH(X^\circ_N)_{N-1}\right]$$

Where

$$H(X_N)_{N-1} = \sum_{i=1}^{c} H_{N-1,i}\, X_{N,i},$$

$$H(X_D)_{N-1} = \sum_{i=1}^{c} H_{N-1,i}\, X_{D,i},$$

$$H(Y_{N-1})_{N-1} = \sum_{i=1}^{c} H_{N-1,i}\, Y_{N-1,i}$$

Substituting these into equation (5-18) the following equations result

$$Q_c = L_N \left[H(X_N)_{N-1} - h_N\right] + D\left[H(X_D)_{N-1} - H_D\right] + \sigma\left\{L_N^\circ\left[H(X_N^\circ)_{N-1} - h_N^\circ\right] + D^\circ\left[H(X_D^\circ)_{N-1} - H_D^\circ\right] - V_{N-1}\left[H(Y_{N-1}^\circ)_{N-1} - H_{N-1}^\circ\right] - Q_c\right\}$$
$$+ \frac{U_N}{\mu\Delta t}\left[H(X_N)_{N-1} - h_N\right] - \frac{U_N^\circ}{\mu\Delta t}\left[H(X_N^\circ)_{N-1} - h_N^\circ\right] \qquad (5\text{-}19)$$

$$L_j = \frac{1}{\left[H(X_j)_{j-1} - h_j\right]}\left\{-D\left[H(X_D)_{j-1} - H_D\right] + Q_c + \sigma\left[V_{j-1}^\circ\left(H(Y_{j-1}^\circ)_{j-1} - H_{j-1}^\circ\right) - L_j^\circ\left(H(X_j^\circ)_{j-1} - h_j^\circ\right) - D^\circ\left(H(X_D^\circ)_{j-1} - H_D^\circ\right) + Q_c\right]\right\}$$
$$+ \frac{1}{\mu\Delta t\left[H(X_j)_{j-1} - h_j\right]}\left\{\sum_{k=j}^{N}\left[U_k\left(H(X_k)_{j-1} - h_k\right) - U_k^\circ\left(H(X_k^\circ)_{j-1} - h_k^\circ\right)\right]\right\} + \frac{fd}{\left[H(X_j)_{j-1} - h_j\right]}, \qquad (5\text{-}20)$$

Where $1.0 < j \leq N$,

$fd = 0$ for $j > f$

$$fd = F\left[H(X_f)_{j-1} - H_F\right] + \delta F^{\circ}\left[H(X_f^{\circ})_{j-1} - H_F^{\circ}\right], \quad \text{for } j \leq f$$

$$V_{j-1} = L_j + D - ZF - \delta(V_{j-1}^{\circ} - L_j^{\circ} - D^{\circ} + ZF^{\circ})$$

$$+ \frac{1}{\mu\Delta t} \sum_{k=j}^{N} (U_k - U_k^{\circ}), \qquad (5\text{-}21)$$

$$ZF = 0 \quad \text{for } j > f$$

$$ZF = F \quad \text{for } j \leq f$$

Equation (5-21) results from the application of the implicit method to the total material balance of tray j. The material and the energy balances together form the basis of the mathematical model of the binary distillation column.

F) θ - METHOD OF CONVERGENCE

Convergence method for the distillation operation is a way of choosing a set of temperatures after each trial, so that eventually, there will be no change in the temperatures, compositions and flow rates between the successive trials.

The θ - method of convergence for the steady state calculation (4) which has been modified by Holland and Waggoner (1) for dynamic calculation, was adopted in this work. This method, for columns at unsteady state operation, is analogous to the θ - method of convergence for a complex column at steady state operation. For a column at unsteady state operation, each holdup specification gives rise to an additional θ - multiplier. This method is an indirect way of choosing a new set of temperatures on the basis of the calculated results obtained from the last assumed set of temperatures. This procedure alters or corrects the mole fractions of the calculated compositions. With these mole fractions, new temperatures are computed. The corrected set of compositions reflects the fact that each component must be in over all material balance and in agreement with the specified values for the total distillate and the total holdups (the conditions of restraint).

With the exception of the reboiler holdup, all the tray holdups, distillate and the condenser holdup give rise to a θ - multiplier:

$$\left(\frac{b_i}{d_i}\right)_{co} = \theta_{N-1}\left(\frac{b_i}{d_i}\right)_{ca} \qquad (5\text{-}22)$$

$$\left(\frac{u_{j,i}}{d_i}\right)_{co} = \theta_j \left(\frac{u_{j,i}}{d_i}\right)_{ca} \qquad 1 \leq j \leq N-2$$

Knowledge of the holdup of the reboiler does not result in a corresponding θ, because $(u_{1,i}/d_i)_{co}$ can be expressed in terms of $(b_i/d_i)_{co}$ as follows:

$$\left(\frac{u_{1,i}}{d_i}\right)_{co} = \left(\frac{U_1}{B}\right)\left(\frac{b_i}{d_i}\right)_{co} = \left(\frac{U_1}{B}\right)\left(\frac{b_i}{d_i}\right)_{ca}\theta_{N-1} \qquad (5\text{-}23)$$

The corrected tray compositions can be calculated from the following equations:

$$X_{j,i} = \frac{(l_{j,i}/d_i)_{ca}\,(d_i)_{co}}{\sum_{i=1}^{c}(l_{j,i}/d_i)_{ca}\,(d_i)_{co}} \qquad (5\text{-}24)$$

$$Y_{j,i} = \frac{(v_{j,i}/d_i)_{ca}\,(d_i)_{co}}{\sum_{i=1}^{c}(v_{j,i}/d_i)_{ca}\,(d_i)_{co}} \qquad (5\text{-}25)$$

The expression for $(d_i)_{co}$ is based on the requirement that each component be in over all material balance, that is

$$\int_{t_n}^{t_n+\Delta t}\left[FX_i - (b_i)_{co} - (d_i)_{co}\right]dt = \sum_{j=1}^{N}(u_{j,i})_{co}\Big|_{t_n+\Delta t} - \sum_{j=1}^{N}(u_{j,i})_{co}\Big|_{t_n} \qquad (5\text{-}26)$$

Applying the implicit method again, the resulting expression for $(d_i)_{co}$ is:

$$(d_i)_{co} = \frac{FX_i + \sigma\left[F^\circ X_i^\circ - d_i^\circ - b_i^\circ\right] + \left(\frac{1}{\mu\Delta t}\right)\sum_{j=1}^{N} u_{j,i}^\circ}{1+\theta_{N-1}\left(\frac{b_i}{d_i}\right)_{co}\left[1 + \left(\frac{U_N/B}{\mu\Delta t}\right)\right] + \frac{1}{\mu\Delta t}\sum_{j=1}^{N}\theta_j\left(\frac{u_{j,i}}{d_i}\right)_{ca}} \quad (5\text{-}27)$$

For a total condenser, the term $(u_{N,i}/d_i)_{co}$ can be replaced by (U_i/D).

Except the θ_j's, all the quantities of equation (5-27) are either known or can be calculated from the component material balance equations. Combining equations (5-22) and (5-26), the following set of simultaneous equations result:

$$g_i(\theta_1\theta_2\cdots\theta_{N-1}) = \sum_{i=1}^{c}(u_{j,i})_{co} - U_j \qquad 1 \leq j \leq N-2$$

$$g_{N-1}(\theta_1,\theta_2\cdots\theta_{N-1}) = \sum_{i=1}^{c}(d_i)_{co} - D \quad (5\text{-}28)$$

The desired set of θ_j's is the set that makes equation (5-28) equal to zero. This set of θ_j's can be found by applying the Newton-Raphson method (96) to equation (5-28). In this process, the following system of simultaneous equations is solved for $\Delta\theta_1, \Delta\theta_2 \cdots \Delta\theta_{N-1}$.

$$\begin{bmatrix} \frac{\partial g_1}{\partial \theta_1} & \frac{\partial g_1}{\partial \theta_2} & \cdots & \frac{\partial g_1}{\partial \theta_{N-1}} \\ \frac{\partial g_2}{\partial \theta_2} & \frac{\partial g_2}{\partial \theta_2} & \cdots & \frac{\partial g_2}{\partial \theta_{N-1}} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{\partial g_{N-1}}{\partial \theta_1} & \frac{\partial g_{N-1}}{\partial \theta_2} & & \frac{\partial g_{N-1}}{\partial \theta_{N-1}} \end{bmatrix} \begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \cdot \\ \cdot \\ \cdot \\ \Delta\theta_{N-1} \end{bmatrix} = \begin{bmatrix} -g_1 \\ -g_2 \\ \cdot \\ \cdot \\ \cdot \\ -g_{N-1} \end{bmatrix} \qquad (5\text{-}29)$$

where $\Delta\theta_j = \theta_{j,n+1} - \theta_{j,n}$.

After the component material balance has been solved for the $v_{j,i}$'s, the desired set of θ_j's that make $g_1 = g_2 = g_3 = \cdots = 0$ simultaneously, is the set of θ_j's that must be larger than zero to satisfy the Newton-Raphson equation (5-29). With the assumed set of $\theta_{j,n}$'s, the g - functions and their derivatives can be evaluated. The equation (5-29) is then solved for $\Delta\theta_j$'s. A new set of θ_j's that are equal to the old set plus the $\Delta\theta_j$'s will be used for the next trial solution of equation (5-29). This process is carried on until a set of θ_j's with the desired degree of accuracy has been found. All the θ_j's are taken equal to 1.0 for the first trial. During the calculation, whenever a $\theta_{j,n} = 0$ is encountered, it is discarded and half of the last positive value of the $\theta_{j,n}$ is used.

The values of the derivatives are evaluated by using the numerical approximation of the derivatives

$$\frac{\partial g_1}{\partial \theta_1} = \frac{g_1\left[\theta_{1,n} + e_1, \theta_{2,n} \text{ --- } \theta_{N-1,n}\right] - g_1\left[\theta_{1,n}, \theta_{2,n} \text{ --- } \theta_{N-1,n}\right]}{e_1} \qquad (5\text{-}30)$$

One of the criterion for selecting the e_j is that it should be smaller than the required accuracy of the θ_j. For example, if $\left| 1 - \theta_j \right| \leq 10^{-2}$ then e_j should be at least smaller than 10^{-2}.

G) DETERMINATION OF NEW TEMPERATURE PROFILE

The method used in this calculation is the same as that used in the steady state calculation.

After the Θ_j's have been determined; the corrected tray temperatures were computed. The new tray temperatures were determined by the non-ideal bubble point calculation. This was done by evaluating each $Y_{j,i}$, then, the corrected temperature was obtained when $\sum_{i=1}^{c} Y_{j,i} = 1.0$.

In order to compute $Y_{j,i}$'s, the equilibrium constants $K_{j,i}$'s have to be determined. For a non-ideal system such as methanol-water, the equilibrium constants from Antoine equation (98) have to be modified by the activity coefficients before they can be used to compute $K_{j,i}$'s. Van Laar third order equations (98) were used to obtain the activity coefficients $\gamma_{j,1}$ and $\gamma_{j,2}$.

With the help of the modified $K_{j,i}$'s the corrected tray temperatures were evaluated with the false position method. Each $Y_{j,i}$ is evaluated at two temperatures T_j^1 and T_j^2. The first is the temperature used in the preceeding time step. The second is:

$$T_j^2 = T_j^1 + \Delta T,$$

Where ΔT is a small pre-set quantity. Then a new approximation T_j^3 is calculated by the following equation:

$$T_j^3 = \frac{\Delta T\left(1.0 - \sum_{i=1}^{c} Y_{j,i}^1\right)}{\sum_{i=1}^{c} Y_{j,i}^2 - \sum_{i=1}^{c} Y_{j,i}^1} + T_j^1 \qquad (5\text{-}31)$$

where $Y_{j,i}^1 = K_{j,i}\,(T_j^1)\,X_{j,i}$

$Y_{j,i}^2 = K_{j,i}\,(T_j^2)\,X_{j,i}$.

The value of T_j^1 is replaced by T_j^3 and the whole process is repeated until $\sum_{i=1}^{c} Y_{j,i} = 1.0$. Then, the corrected tray temperature is obtained. With the new set of tray temperatures, enthalpy and total material balances are used to compute the new set of liquid and vapor flow rates.

H) CALCULATION PROCEDURE

The calculation procedure can be summarized in the following steps:

1) Take μ = .6 as suggested by Holland (1). Choose the first Δt.

2) Assume values for the temperature and the (L/V) profiles at time $t_n+\Delta t$. The point-slope predictor will be used to predict the values of the temperature and vapor rate profiles for the second and subsequent time increments. The point slope formulas for T_j and V_j are as follows:

$$T_j\Big|_{t_n+\Delta t} = T_j\Big|_{t_n-\Delta t} + 2\Delta t \left.\frac{dT_j}{dt}\right|_{t_n} \qquad (5\text{-}32)$$

$$V_j\Big|_{t_n+\Delta t} = V_j\Big|_{t_n-\Delta t} + 2\Delta t \left.\frac{dT_j}{dt}\right|_{t_n}$$

The derivatives of T_j's and V_j's are evaluated numerically.

3) Compute d_1's, $l_{j,i}$'s, $u_{j,i}$'s and b_1's at the end of every time increment. This is done by using the component material balance equations.

4) Find the θ_j's which make $g_1 = g_2 = g_3$ ----- $= g_{n-1} = 0$, by means of the Newton-Raphson method.

5) Calculate the corrected liquid compositions and

use these values to evaluate the new temperature profile.

6) Calculate the new liquid and the vapor flow rates with the enthalpy balance and the total material balance equations.

7) Check whether $|1 - \theta_j| \leq 10^{-2}$ is true or not. If not, return to step 2. If it is, proceed to the next time increment by returning to step 1.

8) Repeat steps 1 to 6 until there is no changes in tray temperatures, compositions and flow rates between two successive trials.

The value of μ in fact can be anywhere between .5 and 1.0. Holland (1) has proved that for $\mu \geq .5$, the calculation is stable and bound, even though Δt increases without bound.

CHAPTER VI

FORMULATION AND SOLUTION OF THE LINEAR DYNAMIC MODEL

A linear dynamic mathematical model of this binary distillation column was developed. This model was used to represent the distillation column in the synthesis of the optimal control policies when the column was under load or set point disturbance.

A) ASSUMPTIONS FOR DEVELOPING THE LINEAR DYNAMIC MODEL

The linear dynamic model is based solely on the component material balance of the binary distillation column with the help of the following assumptions:

1) Adiabatic operation.
2) Negligible vapor holdup on all trays.
3) Liquid holdups are constant with respect to time for all trays.
4) Tray efficiencies are 100%.
5) Changes of vapor and liquid flow rates are instantaneous.
6) Equal molal overflow.
7) Equilibrium constants remain unchanged when the operating condition changes.

B) FORMULATION OF THE COMPONENT MATERIAL BALANCE EQUATIONS.

A component material balance around the reboiler leads to equation (6-1).

$$U_1 \frac{dX_1}{dt} = L_2X_2 - V_1Y_1 - B_1X_1$$

$$U_1 \frac{dX_1}{dt} = L_2X_2 - (V_1K_1 + B_1) X_1 \qquad (6-1)$$

For the stripping section of the column, the general equation that represent any tray of this section is the following equation:

$$U_j \frac{dX_j}{dt} = V_{j-1}K_{j-1}X_{j-1} - (V_jK_j + L_j) X_j + L_{j+1}X_{j+1} \qquad (6-2)$$

For tray where feed is introduced into the column, the equation is:

$$U_f \frac{dX_f}{dt} = V_{f-1}K_{f-1}X_{f-1} - (V_fK_f + L_f) X_f + L_{f+1}X_{f+1} + FX_F \qquad (6-3)$$

Applying the same component material balance to the rectifying section, the equation that represents any tray of this section is:

$$U_j \frac{dX_j}{dt} = V_{j-1}K_{j-1}X_{j-1} - (V_jK_j + L_j) X_j + L_{j+1}X_{j+1} \qquad (6-4)$$

Moving to the top of the column, the equation for the condenser is :

$$U_N \frac{dX_N}{dt} = V_{N-1} K_{N-1} X_{N-1} - (D + L_N) X_N \qquad (6\text{-}5)$$

Equations (6-1 to (6-5) are non-linear in nature. In order to linearize these equations, the non-linear terms are expanded into a Taylor series with the linear terms being retained. The following set of linearized differential equations result:

$$U_1 \frac{d\bar{x}_1}{dt} = (L_2)\,\bar{x}_2 - (K_1V_1 + B)\,\bar{x}_1 + (X_2 - X_1K_1)\,\Delta v + (X_2)\,\Delta L - (X_1)\,\Delta B \qquad (6\text{-}6)$$

$$U_j \frac{d\bar{x}_j}{dt} = (V_{j-1}K_{j-1})\,\bar{x}_{j-1} - (V_jK_j + L_j)\,\bar{x}_j + (L_{j+1})\,\bar{x}_j + (K_{j-1}X_{j-1} - K_jX_j)\,\Delta v + (X_{j+1} - X_j)\,\Delta L \qquad (6\text{-}7)$$

$$U_f \frac{dx_f}{dt} = (V_{f-1}K_{f-1})\,\bar{x}_{f-1} - (V_fK_f + L_f)\,\bar{x}_f + (L_{f+1})\,\bar{x}_{f+1} + (K_{f-1}X_{f-1} - K_fX_f)\,\Delta v + (X_{f+1} - X_f)\,\Delta L + (F)\,\Delta X_F + (X_F)\,\Delta F \qquad (6\text{-}8)$$

$$U_N \frac{dX_N}{dt} = (V_{N-1}K_{N-1})\,\bar{x}_{N-1} - (D + L_N)\,\bar{x}_N - (X_N)\,(\Delta D + \Delta L) + (K_{N-1}X_{N-1})\,\Delta v \qquad (6\text{-}9)$$

where $\Delta v = \Delta D + \Delta L \qquad (6\text{-}10)$

$\Delta B = \Delta L - \Delta v = - \Delta D \qquad (6\text{-}11)$

C) SOLUTION OF THE COMPONENT MATERIAL BALANCE EQUATIONS

Equations (6-6) to (6-11) formulate the linear dynamic model of the binary distillation. These equations can be put into the compact vector-matrix form. The fourth order Runge-Kutta integration routine (100) was employed to solve this set of differential equations to give the linear unsteady state responses of the column.

CHAPTER VII

SYNTHESIS OF THE OPTIMAL CONTROLLERS

A) DESCRIPTION OF THE OPTIMAL CONTROL PROBLEMS

Two different types of optimal control problems have been studied in this work, they were:

1) THE SERVOMECHANISM PROBLEM

This is also referred to as the set-point change problem. The set points in this case are the top or bottom product specification. The problem here is to change either one of the product specifications from one value to another. This is essentially the same as changing the column's operating condition from one level to another. The purpose of the controllers is to make this change-over in an optimal fashion.

2) THE REGULATOR PROBLEM

This is also referred to as the load disturbance problem. In this case, the controller should be able to keep the top and the bottom product compositions on specifications in an optimal fashion in spite of external disturbances such as feed rate and composition.

3) THE OPTIMAL CONTROLLERS

Two different types of optimal feedforward controllers based on different control philosophies were synthesized in this study.

a) The first controller was originally proposed by Kalman and co-workers (76) (77) and is called the minimum energy regulator problem by control engineers. When a disturbance enters the column through the feed stream, or a change in the product specifications is required, the compositions of the two product streams tend to change. The function of this type of controller is to bring these two terminal product compositions back to their respective desired values while minimizing the following function.

$$J_1 = \underline{x}'(t_f)\,\underline{S}\,\underline{x}(t_f) + \int_0^{t_f} \tfrac{1}{2}\,(\underline{x}'(t)\,\underline{Q}\,\underline{x}(t) + \underline{u}'(t)\,\underline{R}\,\underline{u}(t))\,dt \qquad (7\text{-}1)$$

Function J_1 is called a quadratic performance index. The controller is optimal in the sense that it performs its duty while optimizing the above performance criterion J_1.

From the definition of the quadratic performance index J_1, the weighting matrix $\underline{S}$, which must be positive semi-definite, can be treated as the penalty function for the state variables - the deviations of the top and

bottom product compositions at terminal time t_f. The weighting matrix Q, which must be positive semi-definite, can be viewed as the penalty function for the two product deviations during the whole control period. The weighting matrix $\underline{R}$, which must be positive definite, governs the control actions between time zero and time t_f. From now on this type of controller will be referred to as the first controller.

b) The second type of controller is more complex than the first one. Besides minimizing the quadratic performance index J_1 (equation (7-1)), this controller also ensures that the average composition of the two product streams at the end of the control period, t_f, are equal to the desired value. Thus on the average, no off-specification products are produced during the control period. In terms of deviation variables, this means that the average deviations of the two product compositions between t = 0 and t_f should be zero. These additional restrictions, when translated into the language of mathematics, yields the following equations

$$x_{1,ave} = I_1 = \frac{\int_0^{t_f} x_1(t)\,dt}{\int_0^{t_f} dt} = 0 \qquad (7\text{-}1a)$$

$$x_{N,ave} = I_2 = \frac{\int_0^{t_f} x_N(t)\, dt}{\int_0^{t_f} dt} = 0 \qquad (7\text{-}1b)$$

Where $x_1(t)$ and $x_N(t)$ are the bottom and top product composition deviations at time t. For $x_{1,ave}$ and $x_{N,ave}$ equal to zero, the numerators on the right hand sides of equations (7-1a) and (7-1b) have to be zero. The above equations can be simplified to

$$x_{1,ave} = I_1 = \int_0^{t_f} x_1(t)\, dt = 0 \qquad (7\text{-}2)$$

$$x_{N,ave} = I_2 = \int_0^{t_f} x_N(t)\, dt = 0 \qquad (7\text{-}3)$$

These two equations, when put into vector form, result in the following equation

$$\underline{x_{ave}} = \int_0^{t_f} \underline{x(t)}\, dt = 0 \qquad (7\text{-}4)$$

Where

$$\underline{x_{ave}} = \begin{bmatrix} x_{1,ave} \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ x_{N,ave} \end{bmatrix} \qquad \underline{x(t)} = \begin{bmatrix} x_1(t) \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ x_N(t) \end{bmatrix}$$

From now on this type of controller will be referred to as the second controller.

Differentiating both sides of equation (7-4) results in the following differential equation

$$\dot{\underline{x}}_{ave} = \underline{x(t)} \qquad (7\text{-}5)$$

with the following terminal conditions

$$\underline{x_{ave}(0)} = \underline{0} \qquad (7\text{-}6)$$

$$\underline{x_{ave}(t_f)} = 0 \qquad (7\text{-}7)$$

The synthesis of the second controller can be referred to as the isoperimetric problem (94) (problem of Dido) in the classical calculus of variations. The original problem involves the find of a maximum area which a curve of fixed length can enclose.

Figure (4) shows the typical responses of the product

composition deviation for these two different controllers for a step increase in column feed composition. As can be seen from this figure, the product composition deviations of the second controller average to zero between time zero and time equal to 70 minutes. (The area above and below the zero line equal to each other). This is not the case for the first controller.

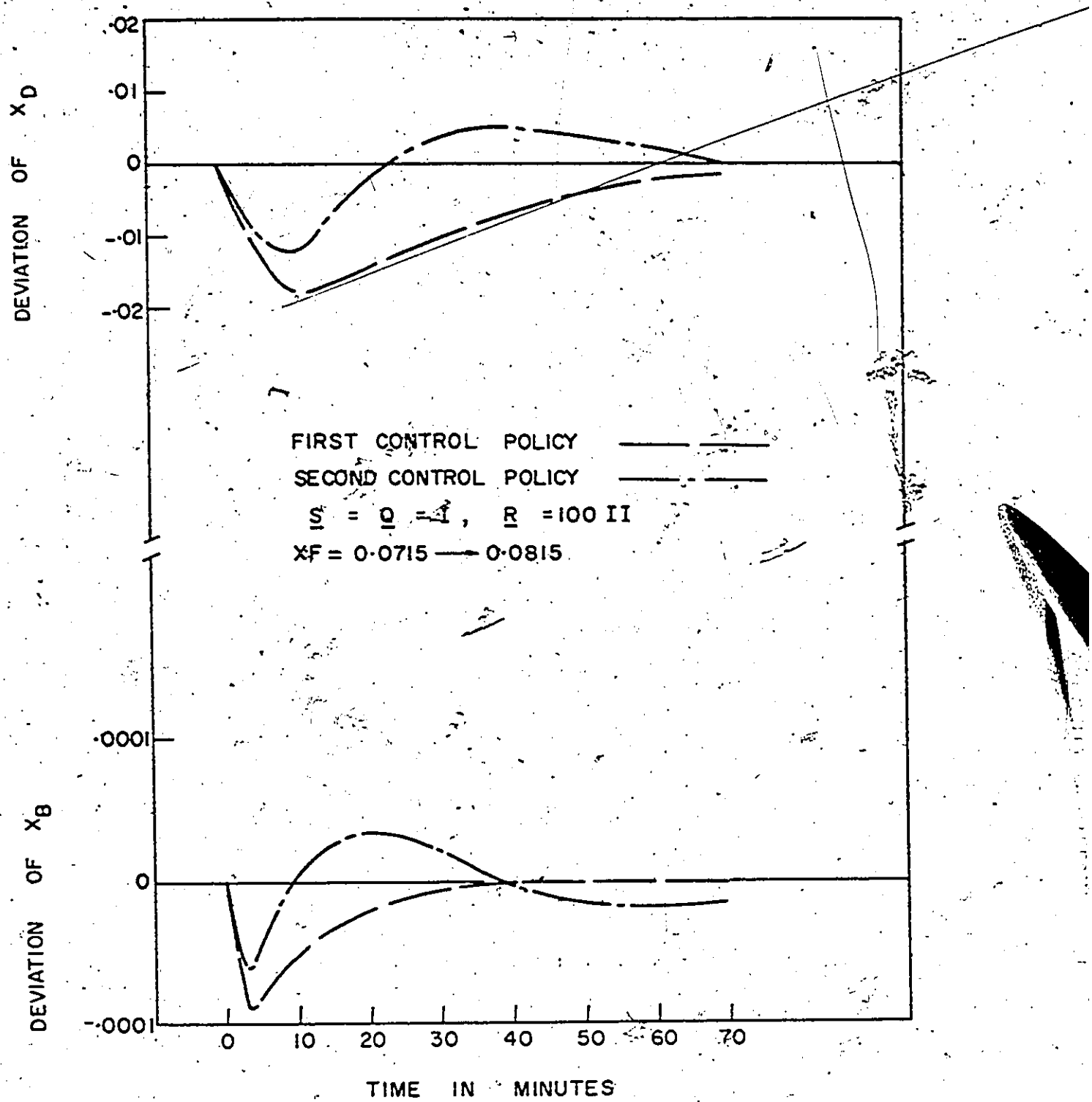


Figure 4. Typical response of top and bottom product deviations for the two different optimal controllers.

B) STATE SPACE REPRESENTATION OF THE LINEAR DYNAMIC MODEL

Equation (6-10) can be rearranged into the following form:

$$\Delta L = \Delta v - \Delta D \qquad (7-8)$$

Substituting equations (6-10) and (7-8) into equations (6-6) to (6-9), the following equations result:

$$U_1 \frac{d\bar{x}_1}{dt} = (L_2)\,\bar{x}_2 - (K_1V_1 + B)\,\bar{x}_1 + (X_2 - X_1K_1)\,\Delta v + (X_1 - X_2)\,\Delta D \qquad (7-9)$$

$$U_j \frac{d\bar{x}_j}{dt} = (V_{j-1}K_{j-1})\,\bar{x}_{j-1} - (V_jK_j + L_j)\,\bar{x}_j + (L_{j+1})\,\bar{x}_{j+1} + (K_{j-1}X_{j-1} - K_jX_j + X_{j+1} - X_j)\,\Delta v + (X_j - X_{j+1})\,\Delta D \qquad (7-10)$$

$$U_f \frac{d\bar{x}_f}{dt} = (V_{f-1}K_{f-1})\,\bar{x}_{f-1} - (V_fK_f + L_f)\,\bar{x}_f + (L_{f+1})\,\bar{x}_{f+1} + (K_{f-1}X_{f-1} - K_fX_f + X_{f+1} - X_f)\,\Delta v + (X_f - X_{f+1})\,\Delta D + (F)\,x_F + (X_F)\,\Delta F \qquad (7-11)$$

$$U_N \frac{d\bar{x}_N}{dt} = -(K_{N-1}V_{N-1})\,\bar{x}_{N-1} - (D + L_N)\,\bar{x}_N + (K_{N-1}X_{N-1} - X_N)\,\Delta v \qquad (7-12)$$

These linear differential equations can be grouped together and put into the following vector-matrix form:

$$\begin{bmatrix} U_1\dot{\bar{x}}_1 \\ U_2\dot{\bar{x}}_2 \\ \vdots \\ U_N\dot{\bar{x}}_N \end{bmatrix} = \begin{bmatrix} -(K_1V_1 + B) & L_2 & & \\ & \ddots & & \\ & & K_{N-1}V_{N-1} & -(D + L_N) \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_N \end{bmatrix}$$

$$+ \begin{bmatrix} (X_2 - X_1K_1) & (X_1 - X_2) & & \\ & \ddots & & \\ & & (K_{N-1}X_{N-1} - X_N) & 0 \end{bmatrix} \begin{bmatrix} \Delta V \\ \Delta D \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ (F)x_F + (X_F)\Delta F \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (7\text{-}13)$$

$$\underline{U}\dot{\bar{x}} = \underline{A}\,\bar{x} + \underline{B}\,\bar{u} + \underline{FF} \qquad (7\text{-}14)$$

Where $\bar{x}$ and $\underline{FF}$ are N-dimensional vectors and $\bar{u}$ is a

m-dimensional vector, $\underline{A}$ is a N x N matrix and $\underline{B}$ is a N x m matrix.

The liquid holdups, U_j's can be absorbed into the right hand side of equation (7-14) through the following transformation of variables, $U_j \bar{x}_j = \bar{\bar{x}}_j$. Equation (7-14) becomes

$$\dot{\underline{\bar{\bar{x}}}}(t) = \underline{A}\,\underline{\bar{\bar{x}}}(t) + \underline{B}\,\underline{\bar{u}}(t) + \underline{FF} \qquad (7\text{-}15)$$

Applying the following transformation of variables to equation (7-15),

$$\underline{x}(t) = \underline{\bar{\bar{x}}}(t) - \underline{\bar{\bar{x}}}_d,$$

$$\underline{u}(t) = \underline{\bar{u}}(t) - \underline{\bar{u}}_d,$$

$$\underline{Ff} = \underline{FF} - \underline{F}_d,$$

it becomes:

$$\dot{\underline{x}}(t) = \underline{A}\,\underline{x}(t) + \underline{B}\,\underline{u}(t) + \underline{Ff} \qquad (7\text{-}16)$$

$\underline{\bar{\bar{x}}}_d$, $\underline{\bar{u}}_d$, and $\underline{F}_d$ are the desired steady state values. The essence of this transformation is to transfer the desired state into the origin while moving the initial state to some other value. The initial condition of this equation becomes: $\underline{x}(0) = \underline{x}_o$ (7-17)

Equation (7-16) is the so-called state space representation of the binary distillation column. Under the terminology of modern control theory, $\underline{x}(t)$ is called the state vector, $\underline{u}(t)$ is the control vector. $\underline{A}$ and $\underline{B}$ are the

coefficient matrices for the state and control vectors. The equation (7-9) was used in this study to represent the binary distillation column in synthesizing the optimal controllers.

C) FORMULATION OF THE OPTIMAL REGULATOR PROBLEM

1) DERIVATION OF THE CO-STATE EQUATION

Applying the minimum principle of Pontryagin (91) to the state equation (equation (7-16)), the quadratic performance index (equation (7-1)) and the two integral constraints (equations (7-2) and (7-3)), the Hamiltonian is defined as:

$$\underline{H} = \underline{Z(t)}\left(\underline{A}\ \underline{x(t)} + \underline{B}\ \underline{u(t)} + \underline{Ff} \right) + \tfrac{1}{2}\ \underline{x'(t)}\ \underline{Q}\ \underline{x(t)} + \tfrac{1}{2}\ \underline{u'(t)}\ \underline{R}\ \underline{u(t)} + \underline{Z_c}\ x(t). \qquad (7\text{-}18)$$

$\underline{Z(t)}$ and $\underline{Z_c}$ are N-dimensional co-state vectors corresponding to the vectors $\underline{x(t)}$ and x_{ave}

$$\underline{Z(t)} = \begin{bmatrix} z_1(t) \\ z_2(t) \\ \cdot \\ \cdot \\ \cdot \\ z_N(t) \end{bmatrix} \qquad \underline{Z_c} = \begin{bmatrix} z_{1,c} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ z_{N,c} \end{bmatrix}$$

The co-state equation is defined as:

$$\dot{\underline{Z}}(t) = -\frac{\delta \underline{H}}{\delta \underline{x}} = -\underline{Q}\ \underline{x(t)} - A'\ \underline{Z(t)} - \underline{Z_c} \qquad (7\text{-}19)$$

Application of the transversality condition (91) results in the following terminal condition for the co-state vector:

$$\underline{Z}(t_f) = \underline{S}\,\underline{x}(t_f) \qquad (7\text{-}20)$$

The other co-state vector $\underline{Z}_c$ is constant throughout the whole trajectory and is unspecified.

2) DERIVATION OF THE OPTIMAL CONTROL TRAJECTORY

In order to minimize the function J_1, it is necessary to minimize the Hamiltonian $\underline{H}$. For unconstrained $\underline{u}$, the minimum can be found by setting $\delta \underline{H}/\delta \underline{u}$ equal to zero.

$$\frac{\delta \underline{H}}{\delta \underline{u}} = \underline{R}\ \underline{u(t)} + \underline{B}'\underline{Z(t)} \qquad (7\text{-}21)$$

$$\underline{u}^* = -\underline{R}^{-1}\ \underline{B}'\underline{Z(t)} \qquad (7\text{-}22)$$

Again, differentiating equation (7-21) with respect to $\underline{u}$, the following expression results:

$$\frac{\delta \underline{H}^2}{\delta^2 \underline{u}} = \underline{R} \qquad (7\text{-}23)$$

From the definition of the function J_1, $\underline{R}$ is positive definite, so $\underline{H}$ has its minimum at $\underline{u(t)} = \underline{u}^*$. Substituting equation (7-22) into (7-16), the state equation becomes

$$\underline{\dot{x}(t)} = \underline{A}\ \underline{x(t)} - \underline{B}\ \underline{R}^{-1}\ \underline{B}'\underline{Z(t)} + \underline{Ff} \qquad (7\text{-}24)$$

The synthesis of optimal controller now reduced to finding the solution of two sets of coupling vector differential equations (7-19) and (7-24), with mixed terminal conditions. This system of equations is called the two-point boundary value problem. Once this is solved, the optimal control trajectory can be generated from equation (7-22).

3) TECHNIQUES FOR SOLVING THE OPTIMAL CONTROL PROBLEMS WITH INTEGRAL CONSTRAINTS

Various methods can be used to solve the two point boundary value problem. The Riccati transformation method (91), was developed for control problem with quadratic performance index and no integral constraint. Kalman, then modified this method and applied the modified Riccati transformation to the tracking problem (91). Pang and Johnson (93) have implemented this method successfully for the control of an extraction column with only one integral constraint and no forcing function i.e. the vector $\underline{Ff} = 0$. In this study the same approach has been extended to the present system with forcing function and two integral constraints.

Equations (7-19) and (7-24) can be arranged into the following form:

$$\begin{bmatrix} \dot{\underline{x}}(t) \\ \hline \dot{\underline{Z}}(t) \end{bmatrix} = \left[\begin{array}{c|c} \underline{A} & -\underline{B}\,\underline{R}^{-1}\underline{B}' \\ \hline -\underline{Q} & -\underline{A}' \end{array}\right] \begin{bmatrix} \underline{x}(t) \\ \hline \underline{Z}(t) \end{bmatrix} + \begin{bmatrix} \underline{Ff} \\ \hline -\underline{Z}_c \end{bmatrix} \quad (7\text{-}25)$$

Utilizing the fundamental matrix (101), the solution of equation (7-25) is:

$$\begin{bmatrix} \underline{x(t)} \\ \underline{Z(t)} \end{bmatrix} = \underline{\Phi}(t_1 t_o) \begin{bmatrix} \underline{x(t)} \\ \underline{Z(t)} \end{bmatrix} + \int_{t_o}^{t} \underline{\Phi}(t_1 \zeta) \begin{bmatrix} \underline{Ff} \\ -\underline{Z_c} \end{bmatrix} d\zeta \qquad (7\text{-}26)$$

Where $\underline{\Phi}(t_1 t_o)$ is the 2N x 2N fundamental matrix of equation (7-25).

At $t = t_f$

$$\begin{bmatrix} \underline{x(t_f)} \\ \underline{Z(t_f)} \end{bmatrix} = \underline{\Phi}(t_f, t) \begin{bmatrix} \underline{x(t)} \\ \underline{Z(t)} \end{bmatrix} + \int_{t}^{t_f} \underline{\Phi}(t, \zeta) \begin{bmatrix} \underline{Ff} \\ -\underline{Z_c} \end{bmatrix} d\zeta \qquad (7\text{-}27)$$

$$\text{Let } \underline{\Delta}(t,\zeta) = \int_{t}^{t_f} \underline{\Phi}(t,\zeta)\, d\zeta \qquad (7\text{-}28)$$

Combination of equations (7-26) (7-27) (7-28) results in:

$$\begin{bmatrix} \underline{x(t_f)} \\ \underline{S\ x(t_f)} \end{bmatrix} = \begin{bmatrix} \underline{\Phi_{1,1}} & \underline{\Phi_{1,2}} \\ \underline{\Phi_{2,1}} & \underline{\Phi_{2,2}} \end{bmatrix} \begin{bmatrix} \underline{x(t)} \\ \underline{Z(t)} \end{bmatrix} + \begin{bmatrix} \underline{\Delta_{1,1}} & \underline{\Delta_{1,2}} \\ \underline{\Delta_{2,1}} & \underline{\Delta_{2,2}} \end{bmatrix} \begin{bmatrix} \underline{Ff} \\ -\underline{Z_c} \end{bmatrix} \qquad (7\text{-}29)$$

Equation (7-29) can be separated into the following equation:

$$\underline{x(t_f)} = \underline{\Phi_{11}}\ \underline{x(t)} + \underline{\Phi_{12}}\ \underline{Z(t)} + \underline{\Delta_{11}}\ \underline{Ff} - \underline{\Delta_{12}}\ \underline{Z_c} \quad (7\text{-}30)$$

$$\underline{S}\ \underline{x(t_f)} = \underline{\Phi_{21}}\ \underline{x(t)} + \underline{\Phi_{22}}\ \underline{Z(t)} + \underline{\Delta_{21}}\ \underline{Ff} - \underline{\Delta_{22}}\ \underline{Z_c} \quad (7\text{-}31)$$

A combination of equations (7-30) and (7-31) yields the following equation

$$(\underline{S}\,\underline{\Phi_{11}} - \underline{\Phi_{21}})\ \underline{x(t)} + (\underline{S}\,\underline{\Phi_{12}} - \underline{\Phi_{22}})\ \underline{Z(t)} + (\underline{S}\,\underline{\Delta_{11}} - \underline{\Delta_{21}})\ \underline{Ff}$$
$$- (\underline{S}\,\underline{\Delta_{12}} - \underline{\Delta_{22}})\ \underline{Z_c} = 0 \quad (7\text{-}32)$$

Rearrange equation (7-32) into

$$\underline{Z(t)} = (\underline{S}\,\underline{\Phi_{12}} - \underline{\Phi_{22}})^{-1} \left[- (\underline{S}\,\underline{\Phi_{11}} - \underline{\Phi_{21}})\ \underline{x(t)} + (\underline{S}\,\underline{\Delta_{12}} - \underline{\Delta_{22}})\ \underline{Z_c} - (\underline{S}\,\underline{\Delta_{11}} - \underline{\Delta_{21}}) \right] \underline{Ff} \quad (7\text{-}33)$$

Let

$$\underline{J(t)} = - (\underline{S}\,\underline{\Phi_{12}} - \underline{\Phi_{22}})^{-1} (\underline{S}\,\underline{\Phi_{11}} - \underline{\Phi_{21}})\ \text{N x N matrix}$$

$$\underline{K(t)} = (\underline{S}\,\underline{\Phi_{12}} - \underline{\Phi_{22}})^{-1} (\underline{S}\,\underline{\Delta_{12}} - \underline{\Delta_{22}})\ \text{N x N matrix}$$

$$\underline{L(t)} = - (\underline{S}\,\underline{\Phi_{12}} - \underline{\Phi_{22}})^{-1} (\underline{S}\,\underline{\Delta_{11}} - \underline{\Delta_{21}})\ \text{N x N matrix}$$

Substituting the above quantities into equation (7-33) yields the following equations:

$$\underline{Z(t)} = \underline{J(t)}\ \underline{x(t)} + \underline{K(t)}\ \underline{Z_c} + \underline{L(t)}\ \underline{Ff} \quad (7\text{-}34)$$

At $t = t_f$

$$\underline{Z(t_f)} = \underline{J(t_f)}\ \underline{x(t_f)} + \underline{K(t_f)}\ \underline{Z_c} + \underline{L(t_f)}\ \underline{Ff} \qquad (7\text{-}35)$$

Comparing equation (7-35) with (7-20), the following equalities result:

$$\underline{J(t_f)} = \underline{S} \qquad (7\text{-}36)$$

$$\underline{K(t_f)} = 0 \qquad (7\text{-}37)$$

$$\underline{L(t_f)} = 0 \qquad (7\text{-}38)$$

The product of the matrix $\underline{K(t)}$ and the vector $\underline{Z_c}$, $\underline{K(t)}\ \underline{Z_c}$, can be simplified into the following form:

$$\underline{K(t)} \begin{bmatrix} z_{c,1} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ z_{c,N} \end{bmatrix} = z_{c,1} \begin{bmatrix} k_{1,1} \\ k_{2,1} \\ \cdot \\ \cdot \\ \cdot \\ k_{N,1} \end{bmatrix} + z_{c,N} \begin{bmatrix} k_{1,N} \\ k_{2,N} \\ \cdot \\ \cdot \\ \cdot \\ k_{N,N} \end{bmatrix}$$

$$\underline{K(t)}\ \underline{Z_c} = z_{c,1}\ \underline{k_1(t)} + z_{c,N}\ \underline{k_N(t)} \qquad (7\text{-}39)$$

Where $\underline{k_1(t)}$ and $\underline{k_N(t)}$ are N-dimensional vectors.

Similarly the product $\underline{L(t)}\ \underline{Ff}$ can be simplified into the following vector form:

$$\underline{L(t)}\ \underline{Ff} = ff\ \underline{l_5(t)} \qquad (7\text{-}40)$$

$\underline{l_5(t)}$ is the fifth column of the matrix $\underline{L(t)}$, therefore, it is a N-dimensional vector.

Substituting equations (7-39) and (7-40) into equation (7-35) the following equation results:

$$\underline{Z(t)} = \underline{J(t)}\ \underline{x(t)} + z_{c,1}\ \underline{k_1(t)} + z_{c,N}\underline{k_N(t)} + ff\ \underline{l_5(t)} \qquad (7\text{-}41)$$

At $t = t_f$,

$$\underline{Z(t_f)} = \underline{J(t_f)x(t_f)} + z_{c,1}\underline{k_1(t_f)} + z_{c,N}\ \underline{k_N(t_f)} + ff\ \underline{l_5(t_f)} \qquad (7\text{-}42)$$

Comparing this equation with equation (7-20), the following equalities result:

$$\underline{J(t_f)} = \underline{S} \qquad (7\text{-}36)$$

$$\underline{k_1(t_f)} = 0 \qquad (7\text{-}43)$$

$$\underline{k_N(t_f)} = 0 \qquad (7\text{-}44)$$

$$\underline{l_5(t_f)} = 0 \qquad (7\text{-}45)$$

Differentiating equation (7-41) with respect to t, yields the following equation:

$$\dot{\underline{Z}}(t) = \dot{\underline{J}}(t)\ \underline{x(t)} + \underline{J(t)}\ \dot{\underline{x}}(t) + z_{c,1}\ \dot{\underline{k}}_1(t) + z_{c,N}\ \dot{\underline{k}}_N(t) + ff\ \dot{\underline{l}}_5(t) \qquad (7\text{-}46)$$

By combining equations (7-24), (7-41) and (7-46) the following equation results

$$\underline{\dot{Z}(t)} = (\underline{J(t)} + \underline{J(t)}\ \underline{A} - \underline{J(t)}\ \underline{B}\ \underline{R}^{-1}\ \underline{B}'\ \underline{J(t)}\ )\ \underline{x(t)}$$

$$+ z_{c,1}\underline{k_1(t)} + z_{c,N}\underline{k_N(t)} - z_{c,1}\ \underline{J(t)}\ \underline{B}\ \underline{R}^{-1}\ \underline{B}'\ \underline{k_1(t)}$$

$$- z_{c,N}\ \underline{J(t)}\ \underline{B}\ \underline{R}^{-1}\ \underline{B}'\ \underline{k_N(t)} + ff\ \underline{\dot{l}_5(t)}$$

$$+ ff - ff\ \underline{J(t)}\ \underline{B}\ \underline{R}^{-1}\ \underline{B}'\underline{l}_5\ (t) \qquad (7\text{-}47)$$

The N-dimensional vector $\dot{l}_5(t)$ is the fifth column of the N x N matrix $\underline{J(t)}$.

Combining equations (7-19) and (7-41) yields the following equation:

$$\underline{\dot{Z}(t)} = (-\ \underline{A}'\ \underline{J(t)} - \underline{Q}\ )\ \underline{x(t)} - z_{c,1}\underline{A}'\ \underline{k_1(t)}$$

$$- z_{c,N}\ \underline{A}'\ \underline{k_N(t)} - \underline{Z_c} - ff\ \underline{A}'\ \underline{l_5(t)} \qquad (7\text{-}48)$$

The N-dimensional vector $\underline{Z_c}$ can be decomposed into the following vectors:

$$\underline{Z_c} = \begin{bmatrix} z_{c,1} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ z_{c,N} \end{bmatrix} = z_{c,1} \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} + z_{c,N} \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} \qquad (7\text{-}49)$$

Let

$$\underline{\theta}_1 = \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \qquad \underline{\theta}_N = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix}$$

Then

$$\underline{Z}_c = z_{c,1}\,\underline{\theta}_1 + z_{c,N}\,\underline{\theta}_N \qquad (7\text{-}50)$$

Combination of equations (7-48) and (7-50) yields the following equation:

$$\dot{\underline{Z}}(t) = (-\underline{A}'\,\underline{J}(t) - \underline{Q})\,\underline{x}(t) - z_{c,1}\,\underline{A}'\,\underline{k}_1(t) - z_{c,N}\,\underline{A}'\,\underline{k}_N(t) - z_{c,1}\,\underline{\theta}_1 - z_{c,N}\,\underline{\theta}_N - ff\,\underline{A}'\,\underline{1}_5(t) \qquad (7\text{-}51)$$

Since the existance of an optimal control solution has been established (93), the corresponding terms of equations (7-47) and (7-51) must be equal to each other. Then, the following equalities are obtained.

$$\dot{\underline{J}}(t) + \underline{A}'\underline{J}(t) + \underline{J}(t)\,\underline{A} - \underline{J}(t)\,\underline{B}\,\underline{R}^{-1}\,\underline{B}'\,\underline{J}(t) + \underline{Q} = 0 \qquad (7\text{-}52)$$

$$\dot{\underline{k}}_1(t) + (\underline{A}' - \underline{J}(t)\,\underline{B}\,\underline{R}^{-1}\,\underline{B}')\,\underline{k}_1(t) + \underline{\theta}_1 = 0 \qquad (7\text{-}53)$$

$$\dot{\underline{k}}_N(t) + (\underline{A}' - \underline{J}(t)\,\underline{B}\,\underline{R}^{-1}\,\underline{B}')\,\underline{k}_N(t) + \underline{\theta}_N = 0 \qquad (7\text{-}54)$$

$$\underline{\dot{l}_5(t)} + (\underline{A'} - \underline{J(t)}\ \underline{B}\ \underline{R}^{-1}\ \underline{B'})\ \underline{l_5(t)} + \underline{i_5(t)} = 0 \qquad (7\text{-}55)$$

Substituting the information of $\underline{Z(t)}$ into equation (7-22), the optimal control trajectory will be generated from the following equation:

$$\underline{u^*(t)} = -\underline{R}^{-1}\ \underline{B'}\left[\underline{J(t)}\ \underline{x(t)} + z_{c,1}\ \underline{k_1(t)} + z_{c,N}\ \underline{k_N(t)} + ff\ \underline{l_5(t)}\right] \qquad (7\text{-}56)$$

Combining equations (7-24) and (7-56), yields the equation which represents the optimal state trajectory:

$$\underline{\dot{x}(t)} = \left[\underline{A} - \underline{B}\ \underline{R}^{-1}\ \underline{B'}\ \underline{J(t)}\right]\underline{x(t)} - z_{c,1}\ \underline{B}\ \underline{R}^{-1}\ \underline{B'}\ \underline{k_1(t)} - z_{c,N}\ \underline{B}\ \underline{R}^{-1}\ \underline{B'}\ \underline{k_N(t)} - ff\ \underline{B}\ \underline{R}^{-1}\ \underline{B'}\ \underline{l_5(t)} + \underline{Ff} \qquad (7\text{-}57)$$

The Riccati transformation has helped to transform the two point boundary value problem into four sets of decoupled linear differential equations. The original two point boundary value problem has mixed terminal conditions, whereas, the four uncoupled equations have terminal condition at terminal time t_f. These differential equations can be integrated backward with respect to time, by any existing numerical integration routine to obtain their time trajectories. After these four sets of differential equations have been

integrated, a search process will be used to locate the value of the co-state vector $\underline{Z_c}$ which satisfies the two integral constraints (equations (7-2) and (7-3)). Then, this particular $\underline{Z_c}$ will be used to generate the optimal control and state trajectories from equations (7-56) and (7-57).

4) OPTIMAL CONTROL PROBLEM WITH NO INTEGRAL CONSTRAINT

When the integral constraint does not exist, the co-state vector $\underline{Z}_c$ is no longer necessary. It can be treated as a special case with $\underline{Z}_c$ equal to zero. This will make the solution simpler than the previous one. The state equation is

$$\dot{\underline{x}}(t) = \underline{A}\,\underline{x}(t) + \underline{B}\,\underline{u}(t) + \underline{Ff}, \qquad (7\text{-}16)$$

$$\underline{x}(0) = \underline{x}_0.$$

The co-state equation is

$$\dot{\underline{Z}}(t) = -\underline{Q}\,\underline{x}(t) - \underline{A}'\,\underline{Z}(t) \qquad (7\text{-}58)$$

$$\underline{Z}(t_f) = \underline{S}\,\underline{x}(t_f).$$

The equation for the control is

$$\underline{u}^*(t) = -\underline{R}^{-1}\,\underline{B}'\,\underline{Z}(t) \qquad (7\text{-}22)$$

After the application of the Riccati transformation, the following working equations are obtained:

$$\underline{Z}(t) = \underline{J}(t)\,\underline{x}(t) + ff\,\underline{l}_5(t) \qquad (7\text{-}59)$$

$$\dot{\underline{J}}(t) + \underline{A}'\underline{J}(t) + \underline{J}(t)\,\underline{A} - \underline{J}(t)\,\underline{B}\,\underline{R}^{-1}\,\underline{B}'\,\underline{J}(t) + \underline{Q} = 0 \qquad (7\text{-}52)$$

$$\underline{J}(t_f) = \underline{S} \qquad (7\text{-}36)$$

$$\underline{u}^*(t) = -\underline{R}^{-1}\,\underline{B}'\left[\underline{J}(t)\,\underline{x}(t) + ff\,\underline{l}_5(t)\right] \qquad (7\text{-}60)$$

$$\dot{\underline{x}}(t) = (\underline{A}' - \underline{B}\,\underline{R}^{-1}\,\underline{B}'\,\underline{J}(t))\underline{x}(t) - ff\,\underline{B}\,\underline{R}^{-1}\,\underline{B}'\,\underline{l}_5(t) + \underline{Ff} \qquad (7\text{-}61)$$

$$\underline{x}(0) = \underline{x}_o$$

No search process is involved because the co-state vector, $\underline{Z}_c$ does not appear in the equations. Only one set of differential equations, instead of four as the previous case has to be integrated before the optimal control and state trajectories can be generated from equations (7-60) and (7-61).

5) NUMERICAL METHODS FOR OBTAINING THE SOLUTIONS

The two point boundary value problem has been transformed into sets of uncoupled linear differential equations, these equations were integrated numerically by applying the Runge-Kutta fourth order integration routine (100).

For the case where integral constraints were presented, two different search methods have been tried to locate the value of the co-state vector $\underline{Z_c}$. The desired $\underline{Z_c}$ was the ore that equations (7-2) and (7-3) were satisfied. The first search method used in this calculation was the pattern search technique (95). It was found that this method was very time consuming. Therefore, an algorithm which would locate the right $\underline{Z_c}$ in a shorter computation time was desired. The Newton-Raphson algorithm was tried and was found to be far superior to the pattern search routine because of its fast convergence. Usually, the required answer was located in less than seven iterations. So far as the pattern search method was concerned, it took close to twenty iterations to reach the solution.

The routine used to integrate equations (7-2) and (7-3) was the combination of Simpson's method and Newton's 3/8 method (102).

Different step sizes have been used. It was found that step size of one minute was most suitable for this study.

6) CALCULATION PROCEDURE

The calculation procedure can be summarized into the following steps.

a) Obtain the new steady state values for the state and control vectors via a steady state calculation.

b) Make a change of variable such that $x_{desired} = 0$.

c) Integrate equations (7-52) to (7-55) backward in time with the Runge-Kutta integration routine.

d) Search for the co-state variables $z_{c,1}$ and $z_{c,N}$.

e) Integrate equation (7-57) to obtain the state trajectory.

f) Integrate equation (7-2) and (7-3) and check if the error criterion is satisfied (i.e. $x_{ave} \leq 10^{-4}$). If not, return to step (d) and repeat the process until the error criterion is satisfied.

g) Calculate the optimal control trajectory with equation (7-56).

All the calculations were done on an IBM 360/65 computer. Double precision arithmetic was used to ensure maximum accuracy. The program for solving this problem utilized about 200 K of the computer core space.

D) FORMULATION AND SOLUTION OF THE SERVOMECHANISM PROBLEM

1) CHANGE OVER POLICY WITH INTEGRAL CONSTRAINTS

The state equation that represents this operation is

$$\dot{\underline{x}}(t) = \underline{A}\,\underline{x}(t) + \underline{B}\,\underline{u}(t) \qquad (7\text{-}62)$$

$$\underline{x}(0) = \underline{x}_0$$

The $\underline{Ff}$ term does not appear here, because there is no feed disturbance entering the column in this case. Applying the same techniques--the minimum principle of Pontryagin and the modified Riccati transformation--and following the same argument as in the regulator problem, the following working equations result:

$$\dot{\underline{Z}}(t) = -\underline{Q}\,\underline{x}(t) - \underline{A}'\underline{Z}(t) - \underline{Z}_c \qquad (7\text{-}19)$$

$$\underline{Z}(t_f) = \underline{S}\,\underline{x}(t_f) \qquad (7\text{-}20)$$

$$\dot{\underline{J}}(t) + \underline{A}'\,\underline{J}(t) + \underline{J}(t)\,\underline{A} - \underline{J}(t)\,\underline{B}\,\underline{R}^{-1}\,\underline{B}'\,\underline{J}(t) + \underline{Q} = 0 \qquad (7\text{-}52)$$

$$\dot{\underline{k}}_1(t) + \underline{A}'\,\underline{k}_1(t) - \underline{J}(t)\,\underline{B}\,\underline{R}^{-1}\,\underline{B}'\,\underline{k}_1(t) + \underline{\theta}_1 = 0 \qquad (7\text{-}53)$$

$$\dot{\underline{k}}_N(t) + \underline{A}'\,\underline{k}_N(t) - \underline{J}(t)\,\underline{B}\,\underline{R}^{-1}\,\underline{B}'\,\underline{k}_N(t) + \underline{\theta}_N = 0 \qquad (7\text{-}54)$$

$$\underline{J}(t_f) = \underline{S} \qquad (7\text{-}42)$$

$$\underline{k}_1(t_f) = 0 \qquad (7\text{-}43)$$

$$\underline{k}_N(t_f) = 0 \qquad (7\text{-}44)$$

The optimal control trajectory is

$$\underline{u}^*(t) = - \underline{R}^{-1} \underline{B}' \left(\underline{J(t)} \, \underline{x(t)} + z_{c,1} \, \underline{k_1(t)} + z_{c,N} \, k_N(t) \right) \tag{7-63}$$

The optimal state trajectory is

$$\dot{\underline{x}}(t) = \left(\underline{A} - \underline{B} \, \underline{R}^{-1} \, \underline{B}' \, \underline{J(t)} \right) \underline{x(t)} + z_{c,1} \, \underline{B} \, \underline{R}^{-1} \, \underline{B}' \, \underline{k_1(t)}$$

$$- z_{c,N} \, \underline{B} \, \underline{R}^{-1} \, \underline{B}' \, \underline{k_N(t)} \tag{7-64}$$

2) CHANGE OVER POLICY WITH NO INTEGRAL CONSTRAINT

Following the same method and argument as in the regulator problem, the following working equations result:

$$\underline{Z(t)} = \underline{J(t)}\ \underline{x(t)} \tag{7-65}$$

$$\dot{\underline{J}}(t) + \underline{A}'\ \underline{J(t)} + \underline{J(t)}\ \underline{A} - \underline{J(t)}\ \underline{B}\ \underline{R}^{-1}\ \underline{B}'\ \underline{J(t)} + \underline{Q} \tag{7-52}$$

$$\underline{J(t_f)} = \underline{S} \tag{7-42}$$

$$\underline{u^*(t)} = -\ \underline{R}^{-1}\ \underline{B}'\ \underline{J(t)}\ \underline{x(t)} \tag{7-66}$$

$$\dot{\underline{x}}(t) = (\ \underline{A} - \underline{B}\ \underline{R}^{-1}\ \underline{B}'\ \underline{J(t)}\ )\ \underline{x(t)} \tag{7-67}$$

$$\underline{x(0)} = \underline{x_o}$$

The same numerical techniques used in the regulator problem were applied in this case to obtain the numerical solutions of these equations.

CHAPTER VIII

EXPERIMENTAL DYNAMIC STUDY OF THE DISTILLATION COLUMN

The purpose of the experimental investigation is to check the validity of the proposed dynamic and steady state models. Step changes in the feed rate, composition, the reflux rate and the steam rate to the reboiler were introduced into the column. Transient as well as steady state temperature profiles of the column were monitored. These experimental data were then compared with the ones generated from the digital computer simulation of the mathematical models.

A) DESCRIPTION OF MATERIALS AND APPARATUS

The piece of equipment used in this work is located in the Department of Chemical Engineering, University of Ottawa. It is a total distillation column of semi-works scale. The binary mixture which this column separated was methanol and water. The methanol used in this work was grade AC-5730, from Anachemia Chemical Limited. A simple schematic diagram of the equipment is shown in figure 5.

1) THE DISTILLATION COLUMN

The column itself was constructed of Q. V. F. 9-inch glass pipe with an inside diameter of 8.38 inches and an outside diameter of 9.25 inches. The working drawing of this glass column can be found in appendix I. Inside the column, there were ten bubble-cap trays, and two distribution plates. One of the distribution plates was for the reflux and the other was for the feed. Both the trays and the distribution plates were made of brass with a thickness of 3/8 inches. There were two bubble-caps on each bubble-cap tray. The bubble-cap risers were made of 1½-inch type "K" seamless copper water pipe. The bubble-caps were made from 2-inch type "K" seamless copper water pipe caps. Each cap contained 28 1/8-inch x ½-inch slots located 1/16 inches from the bottom of the cap. The circular downers were also made from 2-inch type "K" seamless copper water pipe. They extended 1½ inches through the tray providing the necessary weir height. The tray spacing including the distribution plate was one foot. On each tray, there was a copper-constantan thermocouple for measuring the temperature of the liquid. It was located in the space between the downcomer and the bubble-caps. Samples of tray liquid were also drawn from the same place where the thermocouple was located. Remotely operated solenoid valves were used to withdraw the liquid samples

from the trays. These samples were passed through water coolers before entering the sample bottle. A detailed working drawing of the bubble-cap tray can be found in appendix I.

The overhead system consisted of a total condenser, overhead reflux drum and two product tanks. The condenser contained two shell and tube copper heat exchangers operated in series. The shell sides of these exchangers were made of 3-inch type "K" seamless copper tube. Each exchanger contained ten 3/8 inches type "K" seamless copper tubing. The length of the two exchangers was 4 feet. The detailed working drawings of the condenser system can be found in appendix I.

Cooling water to the condenser was manipulated by a hand valve and a rotameter. Part of the overhead condenser was withdrawn as overhead product. The overhead product flow rate was manipulated by a liquid level control system on the overhead product drum which was under the condenser. The product tanks of stainless steel, were horizontal, 18-inch diameter x 28 inches long. Part of the overhead condensate was pumped through the reflux flow control valve into the reflux distribution plate.

The bottom system consisted of a thermosyphon reboiler, bottom product cooler and two storage tanks. The reboiler was an 8-inch diameter shell and tube copper heat-exchanger mounted on the bottom of the column.

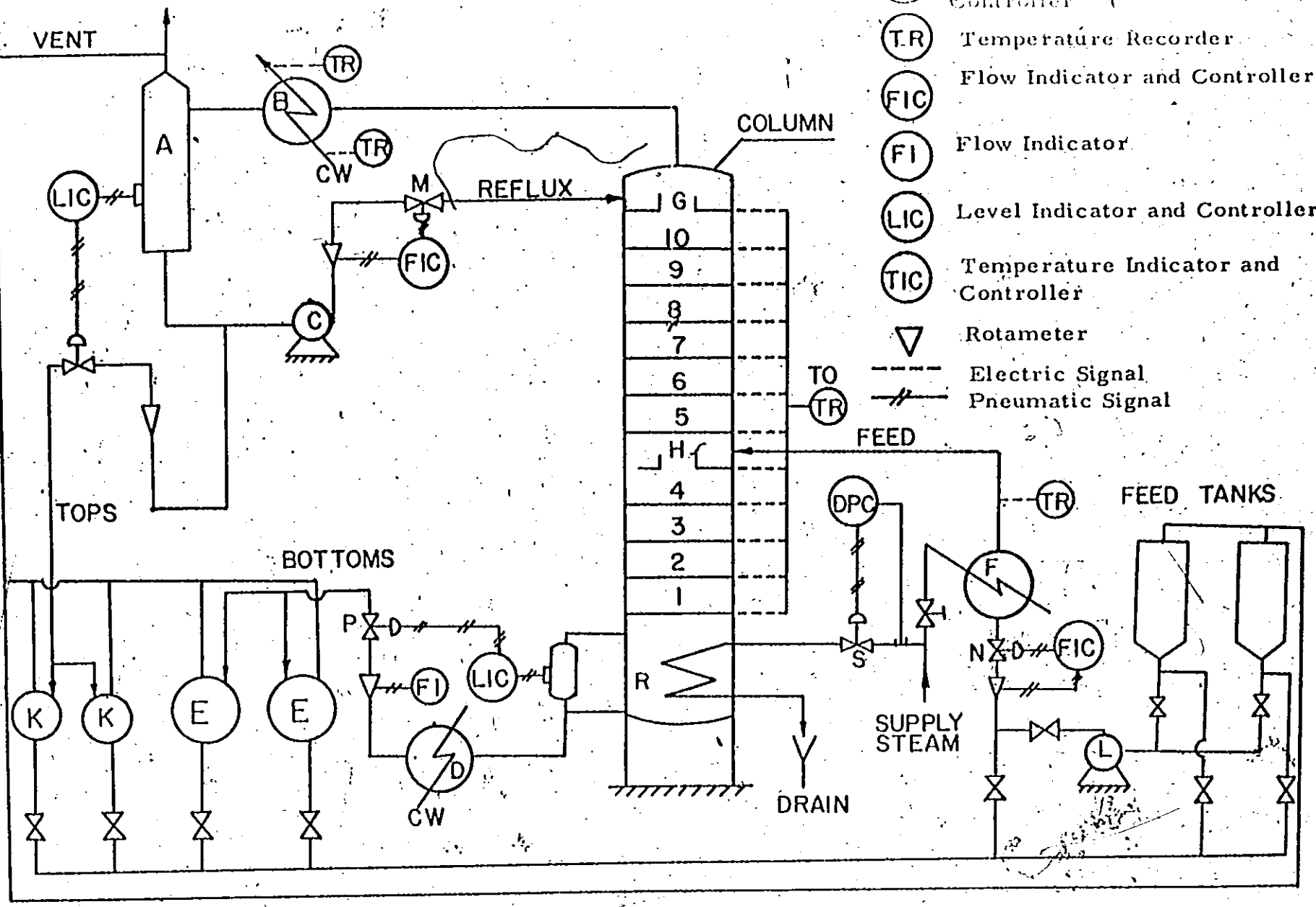


Figure 5. Schematic diagram of the experimental column.

NOTATION FOR FIGURE

A	Overhead Condensate Drum.
B	Overhead Condenser.
C	Reflux Pump.
D	Bottoms Cooler.
E	Bottoms Product Tank.
F	Feed Preheater.
H	Feed Distribution Plate.
K	Top Product Tank.
L	Feed Pump.
M	Reflux Control Valve.
N	Feed Control Valve.
P	Bottoms Control Valve.
R	Reboiler.
S	Steam Control Valve.
G	Reflux Distribution Plate.

The thickness of the shell was .125 inches. There were 48 copper tubes of 5/8 inches outside diameter and a wall thickness of .045 inches inside the shell. The length of these tubes was 18 inches. The steam was introduced in the shell side and the liquid in the tube side. Detailed drawings of the reboiler can be found in appendix I.

The steam rate to the reboiler was controlled by a flow control system. The bottom product withdrawal rate was manipulated by a level control system. The bottom product was passed through a bottom cooler before being stored in the two product tanks. The bottom cooler consisted of two shell and tube heat exchangers in series. It had the same design as the overhead condenser. Cooling water to this bottom cooling system was manipulated by a hand valve. The two storage tanks of stainless steel, were horizontal, 24-inch diameter x 48-inch long.

Two vertical stainless steel tanks (24-inch diameter x 68-inch high) were used to store the feed stock. Well mixed feed was pumped through the pre-heater on to the feed distribution plate located above the fourth tray from the bottom. The feed-preheater was a shell and tube heat exchanger. The design of the feed-preheater was the same as the overhead condenser. 3-inch carbon steel pipe was used for the shell side instead of the

3-inch type "K" copper pipe as in the overhead condenser. Steam was used as the heating medium. It was controlled manually with a pressure reducer. The piping system of this equipment was built in such a way that liquid could be transfered directly between the feed and the top and bottom product tanks. With this arrangement, the methanol and water solution could be recycled and blended conveniently.

2) AUXILIARY INSTRUMENTS

A constant temperature bath and a 3-ml. capillary U-tube pycnometer were used to determine the specific gravity of the methanol-water system at 20° c. Detailed description of this method can be found in a paper by Lipkin et al. (103).

Following is a list of auxiliary instruments used in this work:

ROTAMETERS

Four Brooks Full View rotameters were used for the following services:

Reflux flow rate: Type 6-1110-5500-A E, indicating maximum flow of .5 USGPM (sp. gr. 0.73). The float was integrated to a transmitter which transduced float lift into pneumatic signal.

Feed flow rate: Type 7-1110-5500-A, indicating maximum flow of .8 USGPM (sp. gr. .98). The flow was integrated to a transmitter which transduced float

lift displacement into pneumatic signal.

Bottom product flow rate: Type 8-1110, indicating maximum flow of .75 USGPM (sp. gr..98).

Top product flow rate: Type 6-1110, indicating maximum flow of .14 USGPM (sp. gr..73).

LEVEL TRANSMITTERS

Foxboro Differential Pressure (D/p) cell:

Type 15A-LS2 with output range of 3-15 psig. at 10-inch water gage setting, to transmit the bottom level of the column.

Honeywell Direct Action Diaphragm Type Liquid Level Transmitter, with output range of 3-15 psig at 10-inch water gage setting, to transmit overhead condenser reflux drum level. The diaphragm was of 316 S.S.

CONTROLLERS/RECEIVERS

Four Foxboro Consotrol Unit, consisting of Model 52A controller output with model 53 receiver for indicating and controlling of feed flow rate, reflux flow rate, reflux drum level and bottom level of the column.

A Honeywell Pneumalik Tel-O-Set two pen recorder with automatic-manual control station model 52302-16-80, was used to control the rate of steam input to the reboiler.

CONTROL VALVES

Five model 1403, Honeywell Control Valves were used for the following services in the experimental set up. All valves had specification of ½ inch body size, 9/16 inch lift, linear contour, non-lubricated, teflon packed, 316 S. S. trim C. I. body.

Reflux control valve: with CV of 0.63

Feed control valve: with CV of 0.63

Bottom level control valve: with CV of 1.6

Reflux drum level control valve: with CV of .36

Steam control valve: with CV of 4.0

THERMOCOUPLE REFERENCE JUNCTIONS

Acromag model 335: A 25-channel, with 0° C reference for type T (copper-constantan) thermocouple reference junction, operating for all thermocouple (one at a time) connected to a Honeywell multipoint temperature recorder.

TEMPERATURE RECORDER

Honeywell Type 153 Electronik Multipoint Recorder: A 16-point connected up with the Acromag model 335 reference junction to record all temperatures. Input range is .5 to 5.5 mV.

FLOW RATE AND LEVEL RECORDER

A Foxboro model 5330F Three Pen Recorder. The recorder was connected with the Foxboro Consotrol units in order to monitor the flow rates and levels of the column. Input range is 3-15 psig.

PUMPS

Two pumps were used for transfering the feed and the reflux into the column. The feed pump also acted as a transfer pump, returning the liquid from the product tanks to the feed tanks. The pumps were Fairbanks and Morse cast iron body centrifugal type. Two Wagne-Leland's Type SPAKJ1, model 911156-02, 1725 rpm 115V, 1/3 H.P. motors were used to drive these pumps.

SOLENOID VALVES

Ten ASCO Red-Had 3-way Solenoid valves, type 8320A3 from Ascolectric Brantford Limited were used for withdrawing liquid samples from each tray. The pipe size was 1/8 inch N. P. T., the valves operated on 110 V A. C. and 60 cycles.

THERMOCOUPLES

Twenty of them were installed in the trays and in various locations of the column. These thermocouples were from Thermo Electric Canada Limited. They were 1/25 inch O. D. Thermo Electric Type 5T0100H copper

constantan with probe length of 6 inches. These thermocouples were installed in the column together with Thermo Electric type MSJ-TX jacks and Thermo Electric No. 8 adjustable mounting brass bushings of 1/8 inch N. P. T. PP-20-TX-thermocouple extension wire was used to connect the thermocouples to the Acromag reference junctions.

THERMOMETER

Beckman type, model T2202-1c, with accuracy to .01 °C from Canlab was used to indicate the bath temperature.

TEMPERATURE CONTROLLER

Bayley Instrument Company model 225 Proportional type with sensing element, the 300 watt heater was connected to this controller.

PYCNOMETER

Ace Glass Incorporated model 5437-10, 3ml pycnometer Type B, was used to determine the specific gravity of the methanol-water solution.

BALANCE

A Mettle H20 electric balance from Fisher Scientific Company with maximum reading of 160 gr. and a minimum reading of .01 mg was used to measure the weights of

the samples.

PNEUMATIC D/P CELL TRANSMITTER

Foxboro Differential Pressure (D/P) Cell: Model 13A, with output range of 3-15 psig calibrated at 0-100 inches of water which was a part of the steam flow control system.

B) EXPERIMENTAL WORK

1. CALIBRATIONS

Some calibrations were required before the dynamic testing was attempted. The volumes of the pycnometers were calibrated against the pycnometers' scale reading. This was done by using deaerated distilled water. The calibration curves can be found in appendix D.

A steam gauge which was used to indicate the pressure of the steam line was calibrated against a dead-weight gauge. This calibration curve is also shown in appendix D.

An orifice meter, which was part of the steam flow control system was installed in the steam line upstream of the control valve. The diameter of this orifice meter was .25 inch. The calibration curve of this orifice meter (recorder reading verses steam flow rate) can be found in appendix D.

2. DYNAMIC TESTING OF THE COLUMN

The first step in this experimental work was to blend the feed stock, then measure its concentration by using the pycnometers. The feed was pre-heated while passing through the feed pre-heater. Then, it was transfered into the column. As soon as the liquid in the reboiler reached a certain level, steam was introduced into the reboiler through the control valve. When there was enough condensate accumulated in the

reflux drum, a pre-determined part of it was pumped back into the column as reflux. The feed and the reflux flow rates were under flow control. The top and the bottom product flow rates were under level control. All these controllers were tuned to the automatic mode. The tray temperature, as well as the reflux temperature were monitored continuously with the help of the multi-point recorder. No attempt was made to automatically control the reflux temperature. The rate of cooling water to the overhead condenser was set at a fixed value such that the reflux temperature was not too far from their saturation temperature for all experimental runs. The pressure in the main steam line was kept high, around 120 psig, before passing through the orifice meter and the control valve. However, the pressure inside the reboiler was relatively low, around 5 psig. The steam rate was quite constant throughout all the experimental runs. Usually, it took about three to four hours before a steady state was attained.

The steady state temperature profiles were recorded for different operating conditions. This was done mainly for checking the validity of the steady state mathematical model.

The dynamic responses of the column were obtained by introducing step changes in the feed rate, feed composition, reflux rate and reboiler steam rate

separately after the first steady state had been reached. For the first trial, liquid samples were taken from tray No. 6 during the whole transient period. The concentrations of these samples were determined by using the pycnometers. At the same time, the tray temperatures were measured and recorded using the multi-point recorder. These measured tray concentrations were compared with the corresponding values obtained from digital simulation of the non-linear dynamic model. It was found that both the experimental and the calculated values agreed quite well with each other as shown in figure 6. Tray No. 6 was chosen because this one had the largest changes in composition and temperature.

For a two-phase binary distillation, once the operating pressure is fixed, the specification of only one additional variable is sufficient to fix all other variables in the system. In this study, the operating pressure was fixed at one atmospheric pressure. According to the phase rule, for a binary system at equilibrium, knowing the temperatures it gives the knowledge of the compositions.

Therefore, the temperature response, instead of the composition response were recorded for all dynamic testing which were performed subsequently.

The thermocouples used in this study were not calibrated. According to the information supplied by Thermo-Electric Co., the limits of error were ± 1½°F between the range of -75 and +200°F. To check the accuracy of the thermocouple, a steady state temperature profile was recorded. The corresponding tray compositions were obtained from the temperature concentration diagram. These values were compared with the ones obtained from analysis of the liquid samples withdrawn from the column at steady state by the specific gravity method. The differences between these two values were very small. Therefore, the accuracy of the thermocouples was considered good enough for this work without being calibrated.

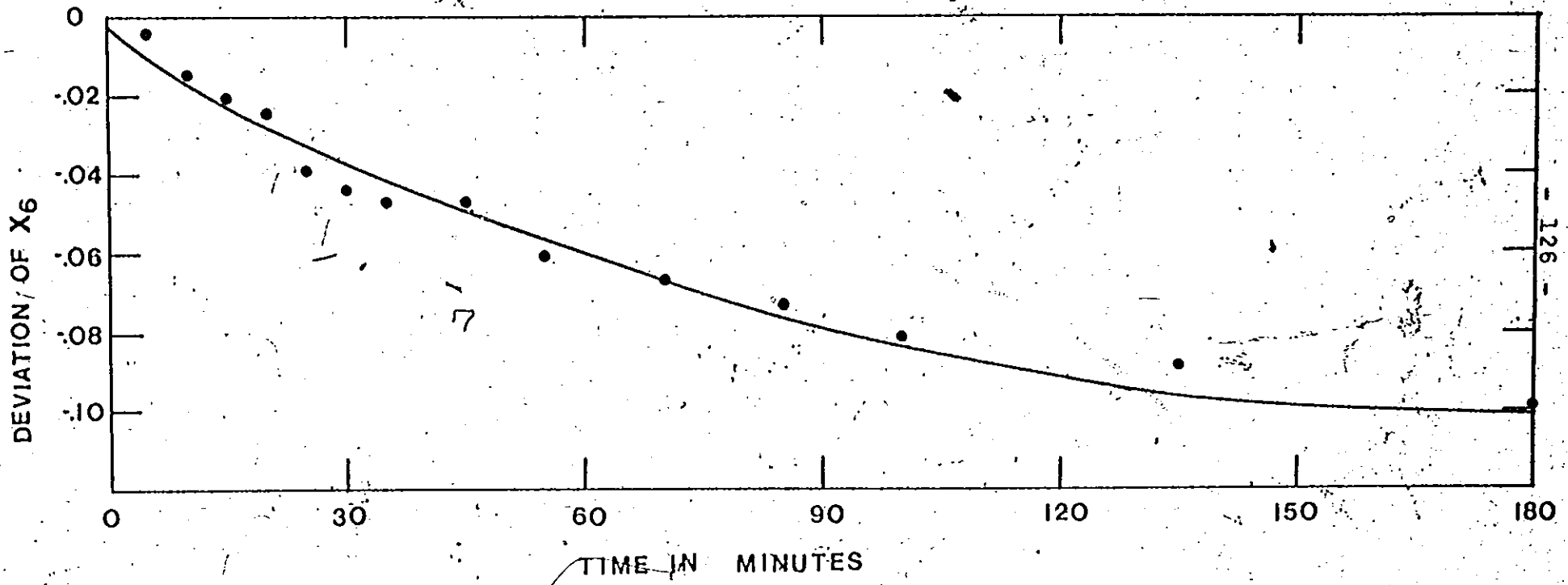


Figure 6. Transient composition response of tray No. 6 under step decrease in feed composition.

CHAPTER IX

RESULTS AND DISCUSSIONS

A) STEADY STATE MODEL

The steady state temperature profiles calculated from the non-linear mathematical model and those from experimental work were plotted and shown in Figures 7 and 8. From these plots, it can be observed that both values agreed quite well with each other. The differences between the experimental and calculated values were not more than 5%. Therefore, the steady state model, used in the work can adequately represent the steady state behavior of this binary distillation column.

The most important factor that dominated the steady state calculation (also the dynamic calculation) was the prediction of the vapor composition with the vapor-liquid equilibrium relationship. Because methanol-water is a non-ideal system, the ideal equilibrium constants calculated from the Antoine equation have to be modified before they can be used. These ideal equilibrium calculations have been used in the steady state calculation, the computation would either converge to the wrong steady state or did not converge at all. The activity coefficients evaluated from the Van Laar third order equations were then used to modify these equilibrium constants, the calculations always converged to the right steady states.

With this type of steady state calculation, it was necessary to assume a temperature as well as a vapor flow

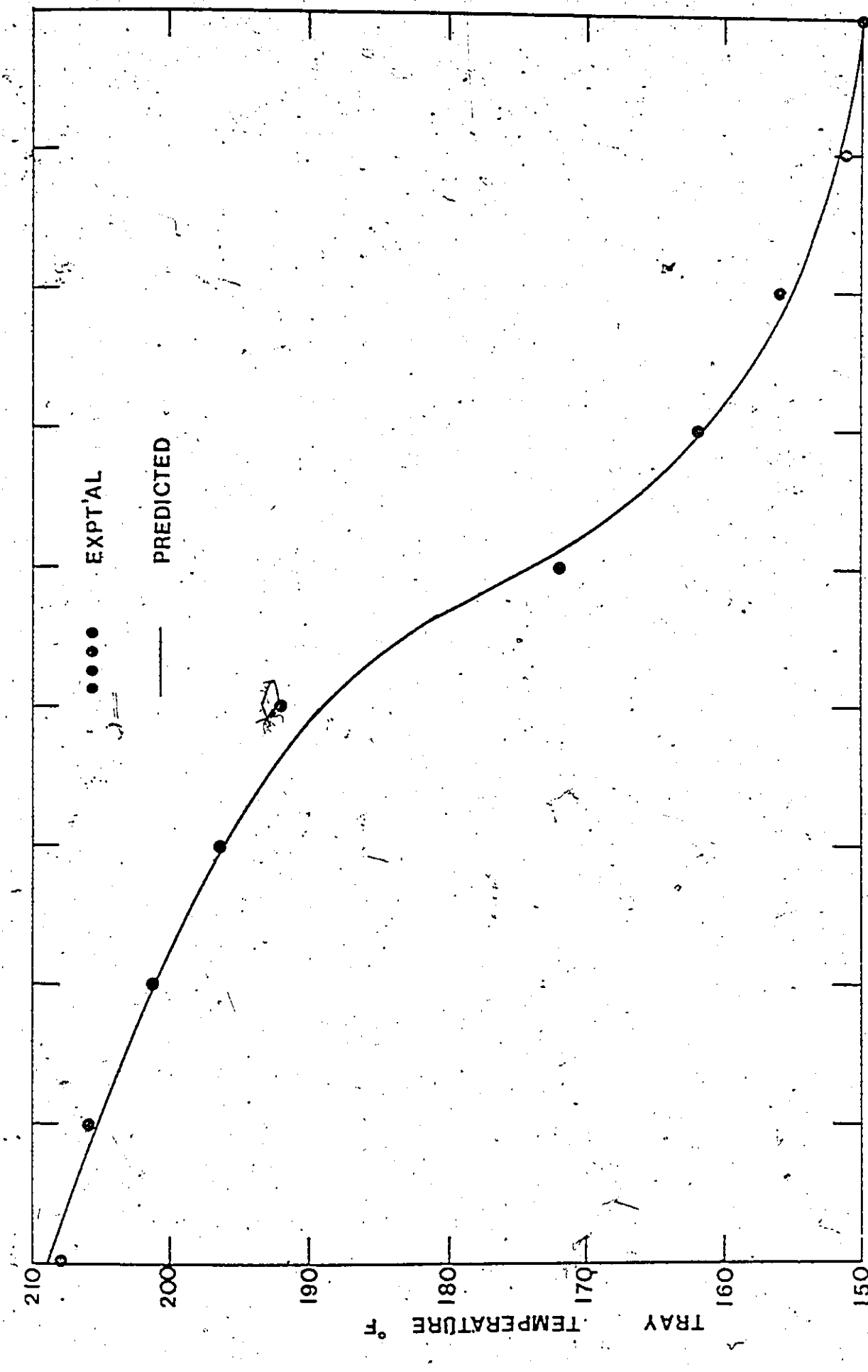


Figure 7. Steady state temperature profile of the column.

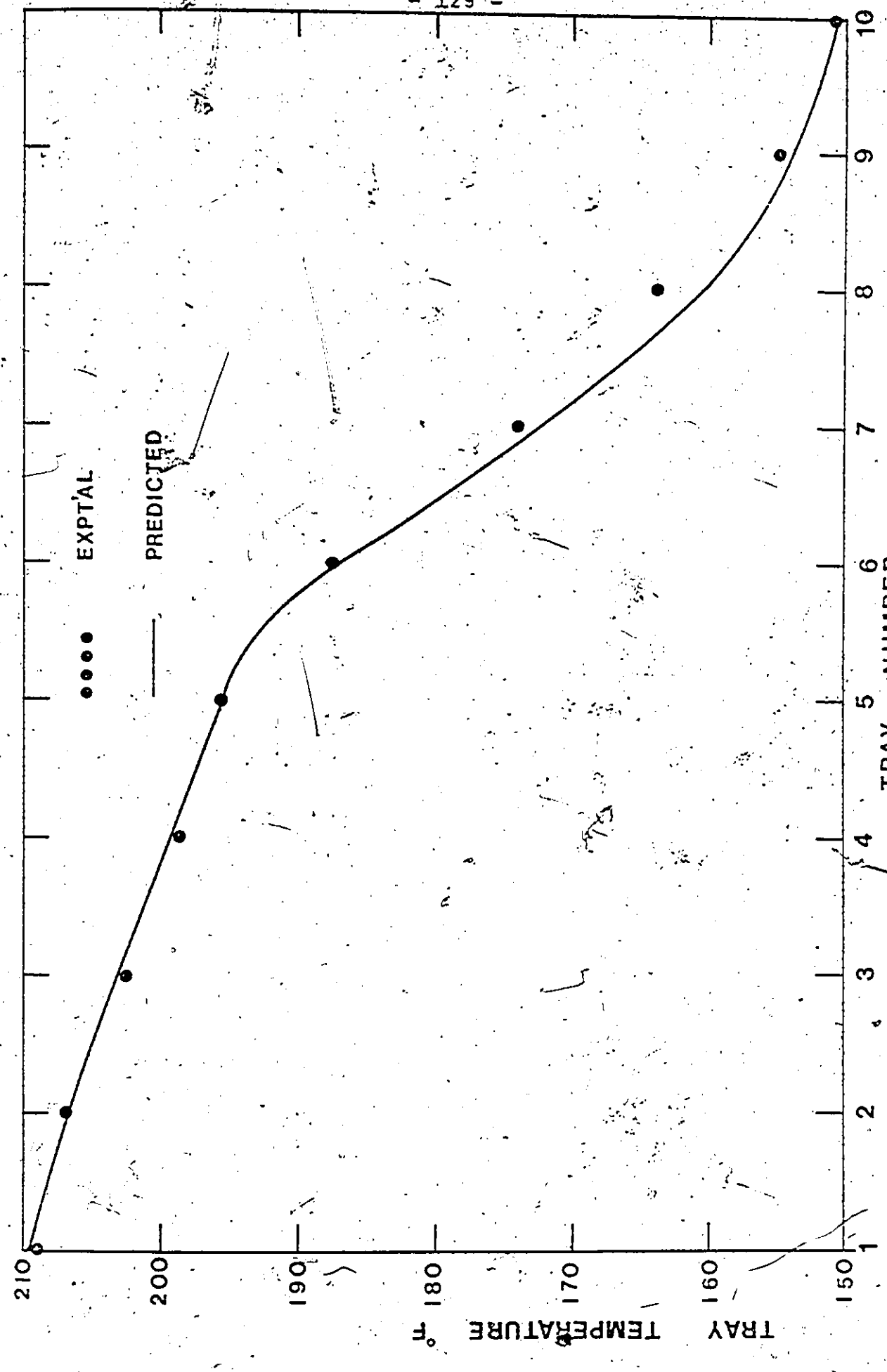


Figure 8. Steady state temperature profile of the column.

profile in order to start the computation. This was considerably simpler than other methods such as the Thiele-Geddes, and the Lewis-Matheson methods. Both of which required the assumption of more variables before calculation could be started.

As far as the computation time is concerned, this method was comparable to other methods. The Thiele-Geddes method was tried. It was found that both methods took more or less the same amount of computation time and computer memory. A typical calculation with the method used in this study tookabout 10 CPUseconds on an IBM 360/65 model computer to obtain the results.

B) STEADY STATE TWO-POINT CONTROL CALCULATION

With the same non-linear steady state model, the values of the control variables were computed when the column was under feed rate, composition disturbance or change in operating condition was demanded. These control variables were the reflux and the bottom product flow rates or the reboiler steam rate and the top product flow rate.

As was mentioned in an early chapter, the values of these two pairs of control variables could be obtained in only one calculation. In this study, the steam rate to the reboiler was picked as the iterating variable. Four different search procedures were tried, two were one-dimensional search methods, the others were multi-dimensional search methods.

Among the four methods, the dichotomous search was the simplest and the most straightforward method. An upper and a lower bound of the search variable had to be assumed at the beginning of the search. Normally, the range between the upper and the lower bound was set at a large value in order to include the desired value. The convergence speed was directly proportional to this range. A large range automatically implied longer computation time. Although this method was simple and straightforward, its speed of convergence was considerably slower than some of the other methods. The only merit of this method was its ease of programming.

The modified pattern search process was on par with the dichotomous search, as far as speed of convergence was concerned. The beauty of this method was that it could always locate the answer regardless of the initial guess. Also, the computation time increased only as the first power of the number of variables rather than to the cube as in other methods. Surely, this method is very powerful for problems with a large number of variables. Judging from the result of this study, this method was not particularly attractive to one dimensional search problems.

The method of Marquardt is a sophisticated process which was originally designed for non-linear regression and solving simultaneous non-linear algebraic equations. It was adopted in this work without any modification and was found to be very satisfactory in locating the answer. Usually, it took about four iterations to reach the right answer. However, this method required the knowledge of derivatives of the functions involved. In this work, the derivatives could only be evaluated numerically. When the estimation of the derivatives was not accurate enough, this process collapsed. This would be an extremely efficient method providing accurate informations of the derivatives could be obtained. Another drawback of this method was the fact that it required a fair amount of skill to incorporate it into a computer program.

The last method tried in this study was the false position method. Of the four, it was as far as this calculation was concerned, the best method. Its convergence speed could match that of the Marquadt's method. For the cases that have been studied, six iterations were enough to obtain the desired answers. Knowledge of the derivative was not necessary. The initial guess and step size had no bearing on the speed of convergence. Above all, this method was very easy to incorporate into a computer program.

The same false position method was also used in the bubble point calculation. The convergence of this method did depend on the initial guess in this particular calculation. When the initial guess was too far from the desired value, this method would collapse. Usually, in the bubble point calculation, the starting values of the temperature profile would not be too far from the desired values. So, this still remains a very good method for the bubble point calculation.

C) DYNAMIC MODELS

Two different dynamic models were studied in this work. The non-linear model was used in predicting the transient behavior of the distillation column under step disturbance or a sequential disturbance in the feed stream, the reflux and the reboiler steam rate. The linear dynamic model was used in the synthesis of the optimal controllers for the control systems of the column.

The non-linear model used in this study was more or less the same as the one developed by Holland and Waggoner (1). The sole difference was that the tray efficiencies were assumed to be unity in this work.

The transient temperature profiles of all trays generated by the non-linear model and the ones that were acquired from experimental testing were plotted in figures 9 to 14. Good agreement between the predicted and the experimental data can be observed from these plots. In most cases, the difference between the experimental and calculated values were less than 10%. However in some cases, the predicted and the experimental values did not agree very well for the trays in the middle part of the column, their differences were as high as 30%. The discrepancies between the two values could be due to the assumptions which were made during the development of the mathematical model, such as unity tray efficiencies and saturation reflux temperature. The limitation of the experimental equipment also contributes in part to these discrepencies.

FIGURE 9

TRANSIENT TEMPERATURE RESPONSE OF THE COLUMN

UNDER STEP INCREASE IN REFLUX RATE

(2.484 $\longrightarrow$ 2.58 lb-mole/hr)

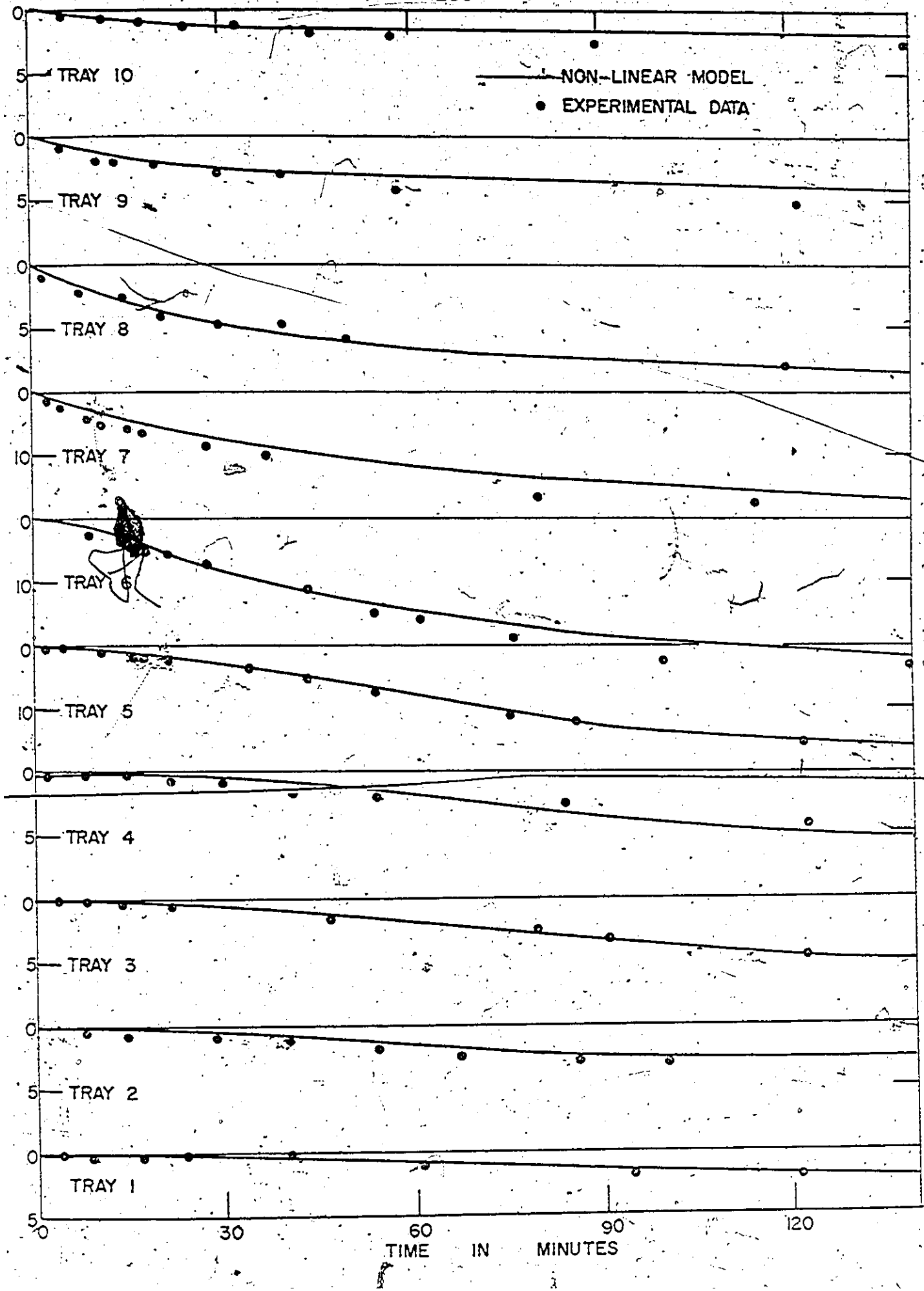


Figure 9. Transient temperature response of the column under step increase in reflux rate.

FIGURE 10

TRANSIENT TEMPERATURE RESPONSE OF THE COLUMN

UNDER STEP DECREASE IN REFLUX RATE

(2.484 ⟶ 2.14 lb-molc/hr)

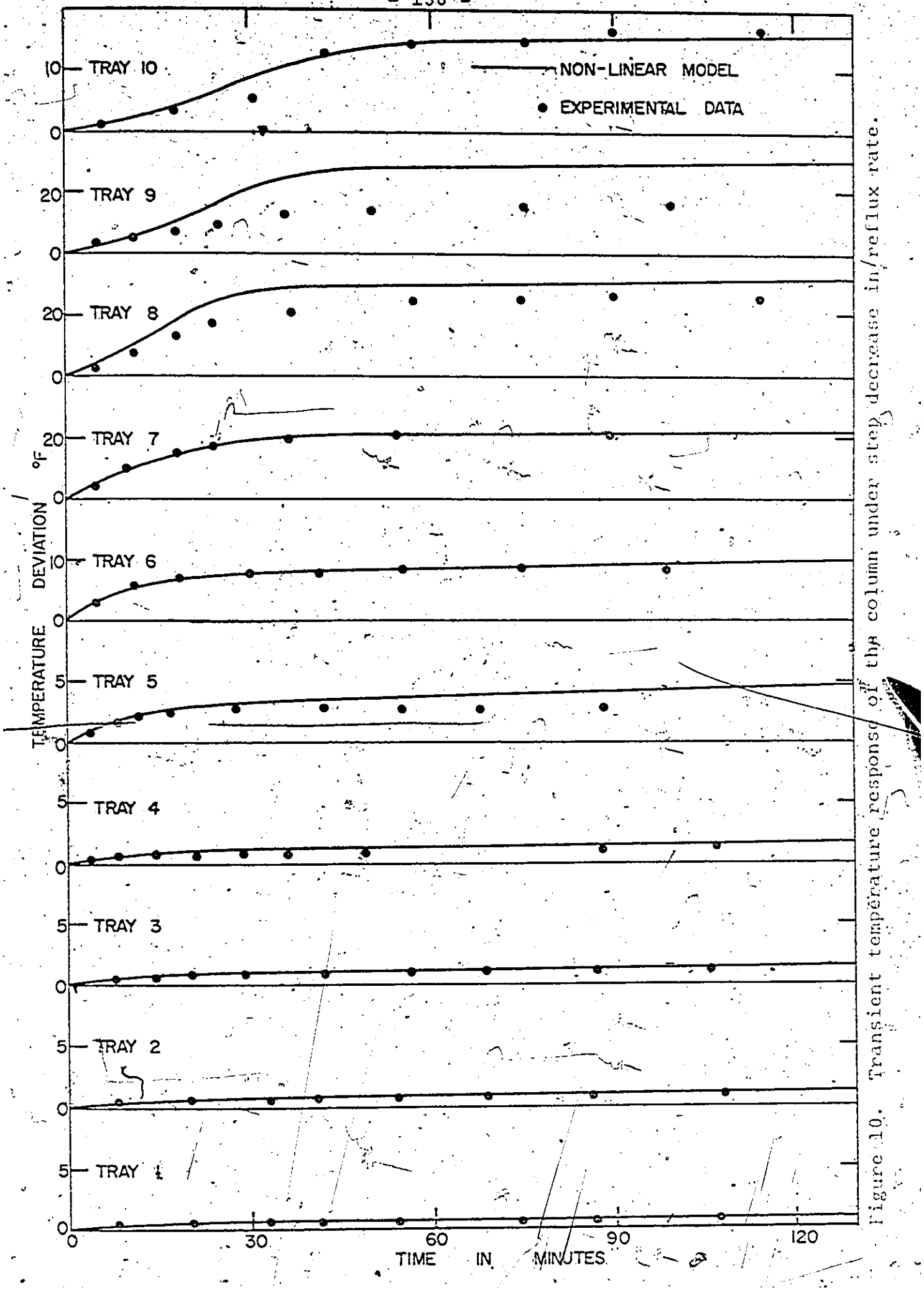


Figure 10. Transient temperature response of the column under step decrease in reflux rate.

FIGURE 11

TRANSIENT TEMPERATURE RESPONSE OF THE COLUMN

UNDER STEP DECREASE IN FEED RATE

(11.5 ⟶ 9.728 lb-mole/hr)

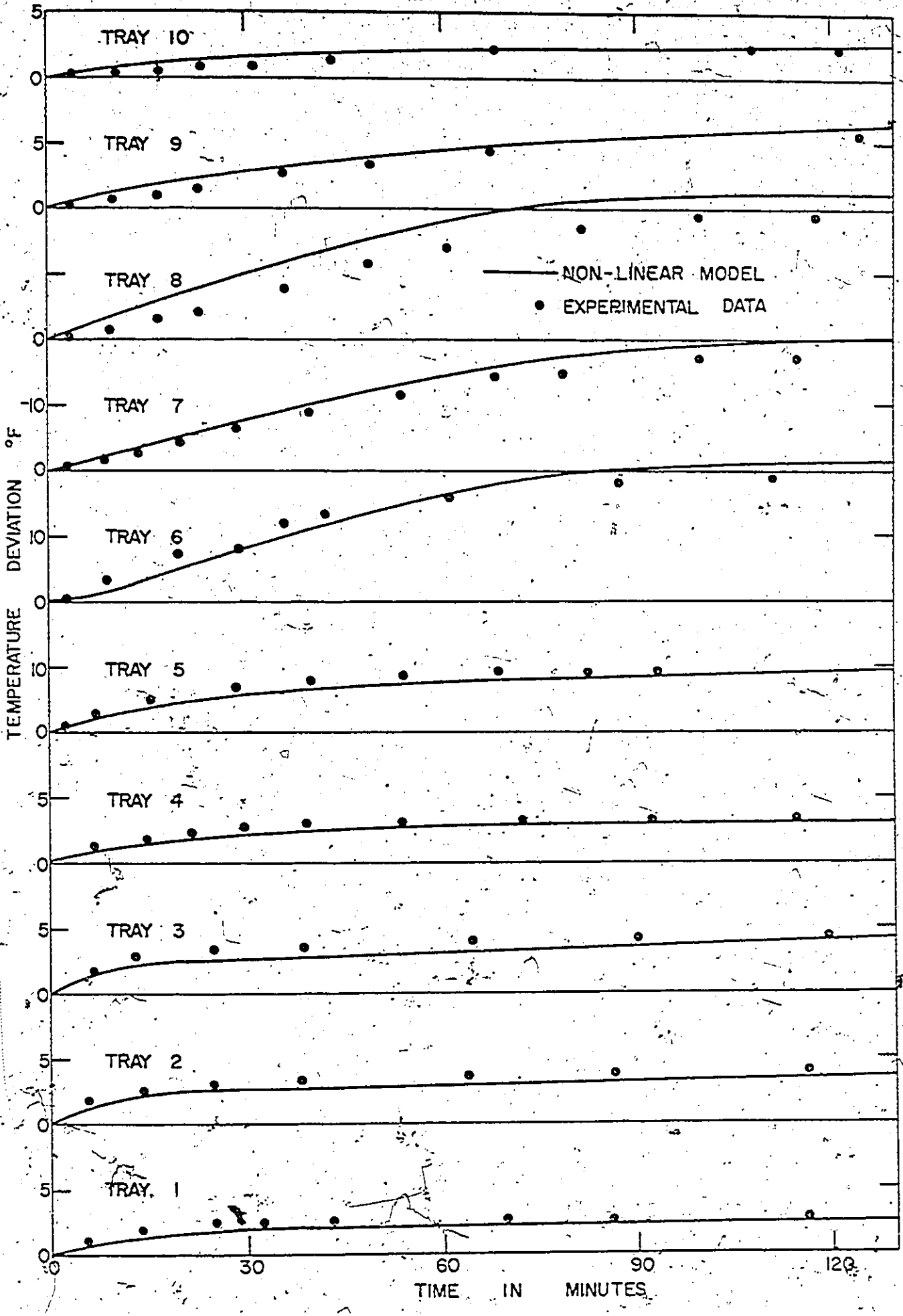


Figure 11. Transient temperature response of the column under step decrease in feed rate.

FIGURE 12

TRANSIENT TEMPERATURE RESPONSE OF THE COLUMN

UNDER STEP INCREASE IN FEED RATE

(9.728 $\longrightarrow$ 11.5 lb-mole/hr)

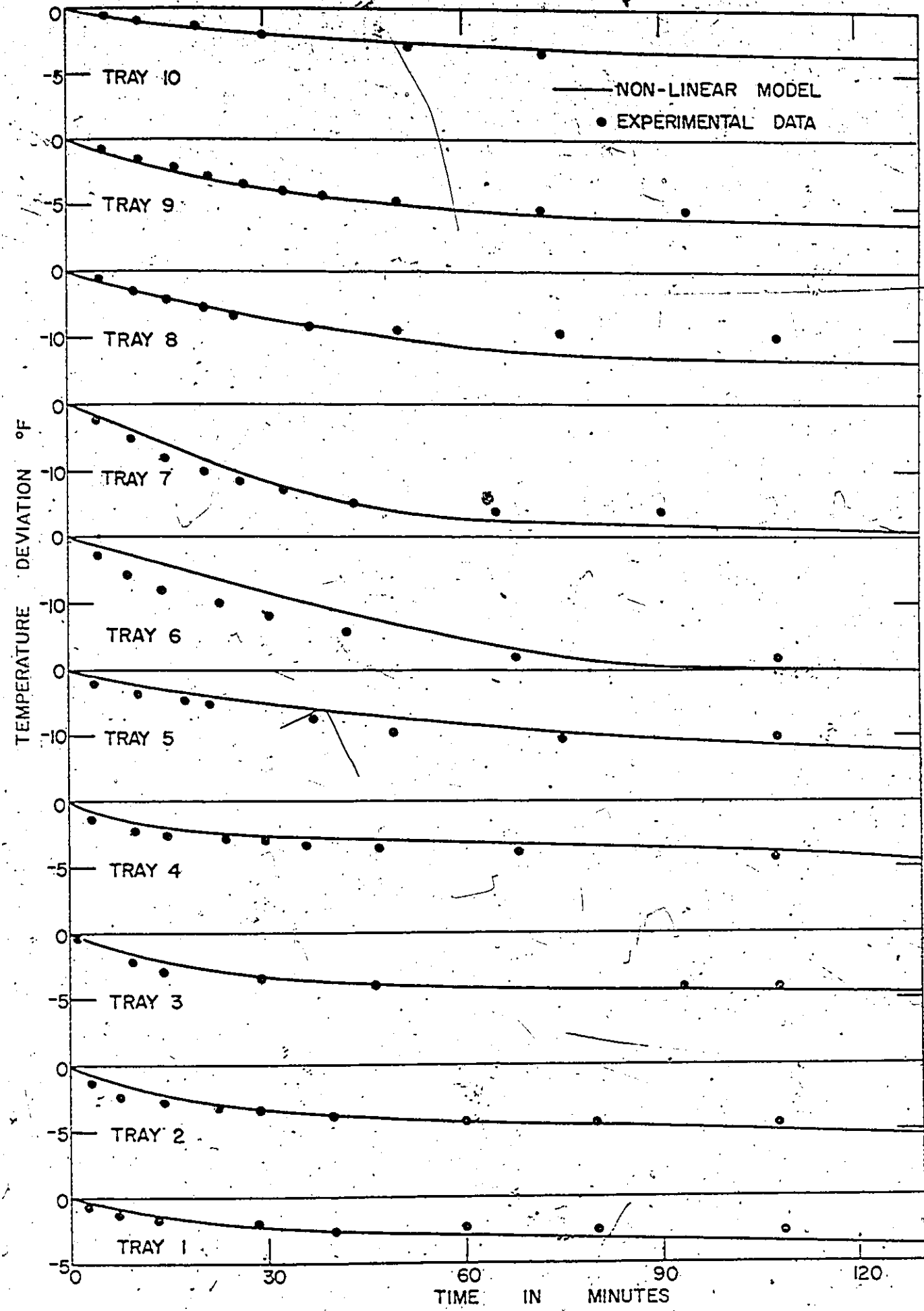


Figure 12. Transient temperature response of the column under step increase in feed rate.

FIGURE 13

TRANSIENT TEMPERATURE RESPONSE OF THE COLUMN

UNDER STEP DECREASE IN STEAM RATE

(51986.0 ⟶ 51009.0 Btu/hr)

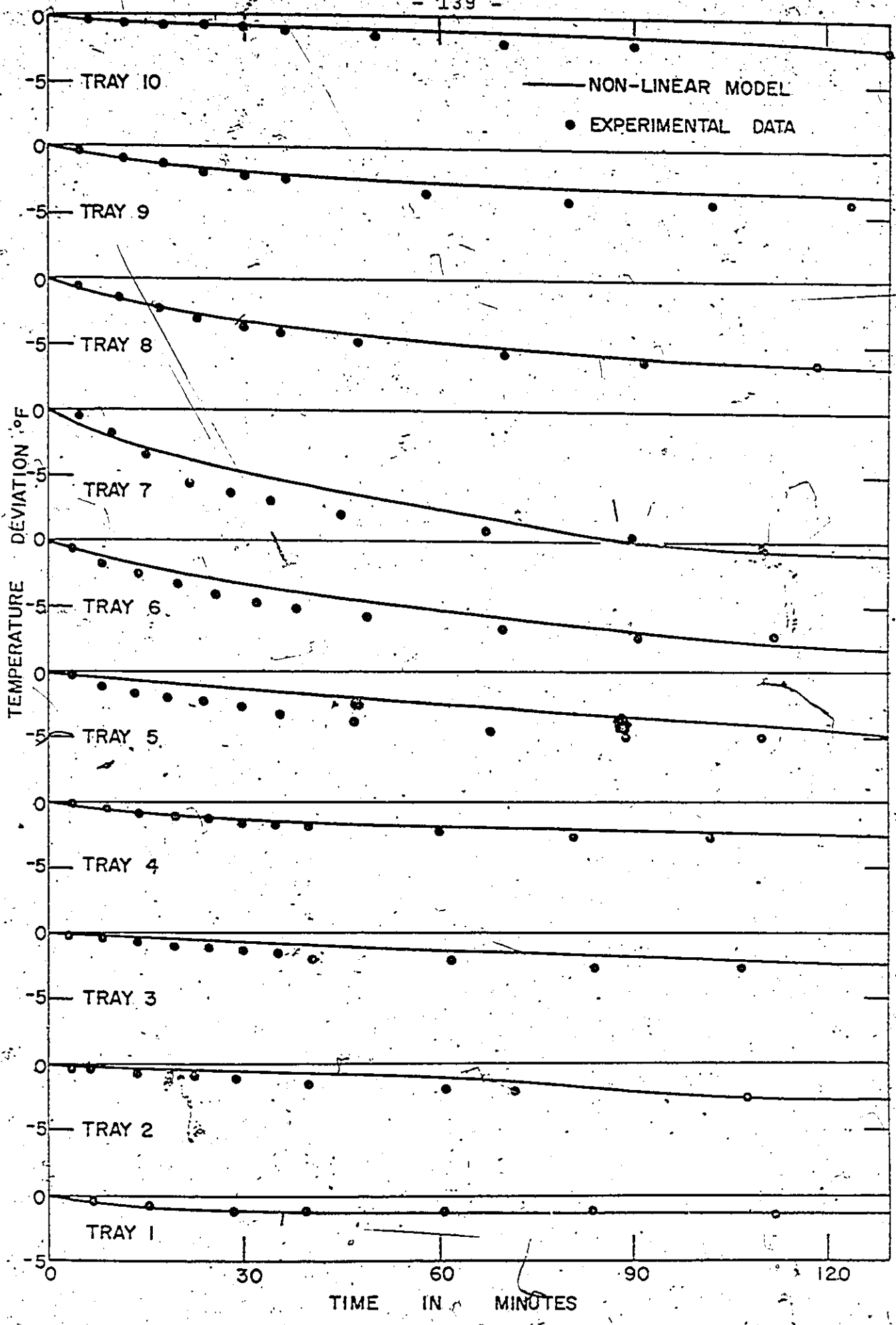


Figure 13. Transient temperature response of the column under step decrease in steam rate.

FIGURE 14

TRANSIENT TEMPERATURE RESPONSE OF THE COLUMN

UNDER STEP DECREASE IN FEED COMPOSITION

(.0713 $\longrightarrow$ .06812 mole fraction)

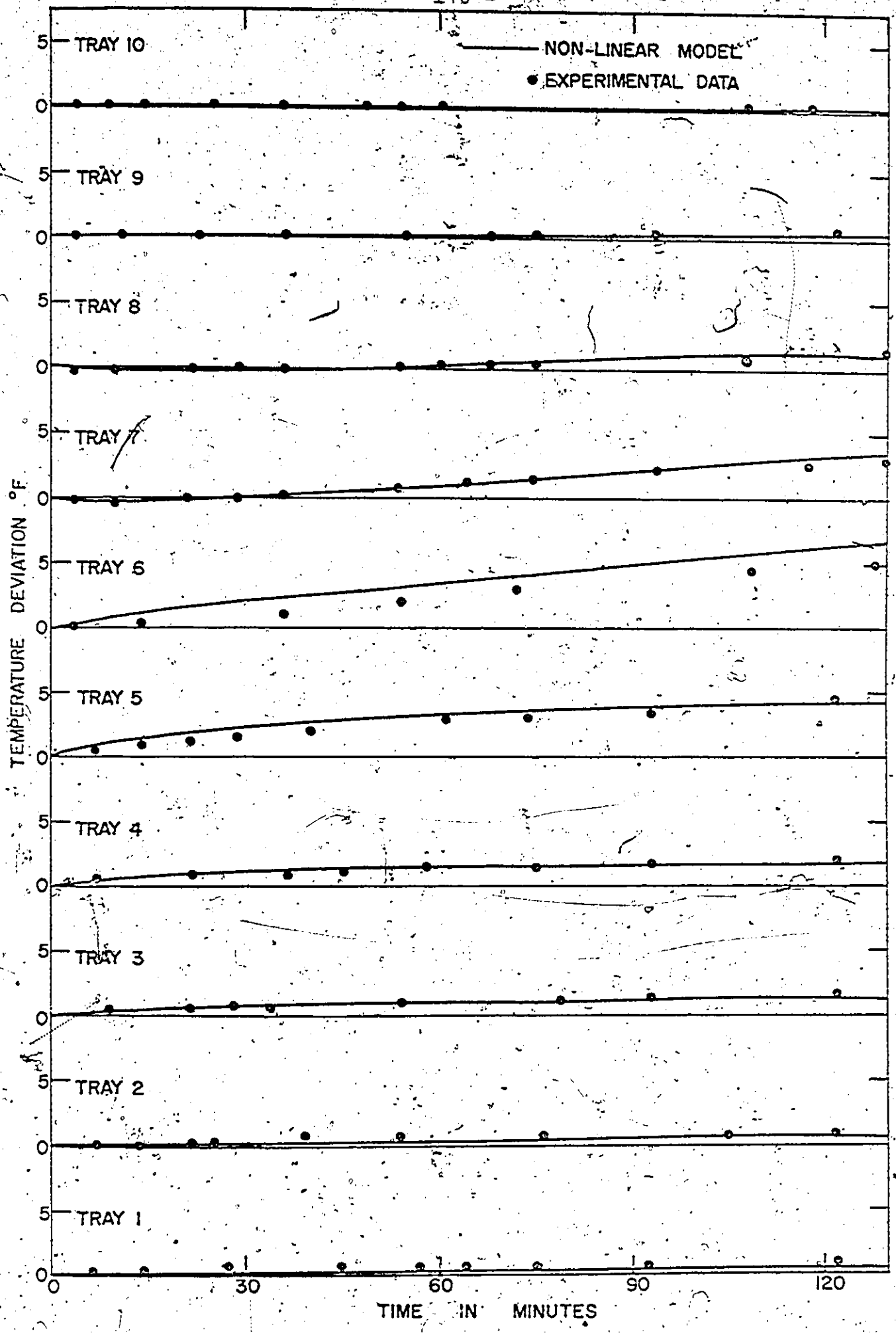


Figure 14. Transient temperature response of the column under step decrease in feed composition.

A better match between the predicted and the experimental values could be obtained if some of the assumptions were relaxed, i.e. try to predict the actual tray efficiencies instead of assuming unity efficiencies. Since the prediction of transient tray efficiencies was fairly involved, no attempt was made to incorporate it into this model.

The θ - method of convergence combined with the implicit method and the constant composition method have proven to be a very powerful way of predicting transient behavior of the column operation. The step size did not have any bearing on the convergence of the solution as long as the value of μ was larger than or equal to .5. This has been proven by Holland (1). Other methods, such as direct numerical integration of the non-linear differential equations depended greatly on the step size. Too small a step size would prolong the computation time and increase the truncation error. Too large a step size would make the calculation diverge instead of converge to the right steady state. The method used in this work would always have the solution converge to the right steady state regardless of the step size. However, the step size did affect the transient response to some extent. A large step size would result in a slower response and a smaller step size would result in a faster response. Also, the response became more oscillatory when the step size grew larger. A step size of

five minutes was used for most of the calculations. This particular value was chosen because the resulting transient response still matched that of the experimental data. A typical calculation with step size equal to five minutes on the IBM 360-65 computer took about two minutes CPU time. The same calculation with step size of one minute took about ten minutes of CPU time. In order to minimize the computing time, the largest possible step size was used in the calculation.

The value of μ had some effect, though not very pronounced on the transient response. A larger value of μ implied a slower response and a smaller value of μ (at least equal to .5) would make the response faster. Yet, μ did not affect the final steady state value at all. In this work, μ was set equal to .6 for all calculation as recommended by Holland (2).

The values of the θ_j's obtained in this work were between .965 to 1.02. In most cases, they were smaller than 1.0, and only a smaller portion of the calculation had θ_j's values larger than 1.0. The step size and μ did affect the values of θ_j's. Large step size and μ tended to drive the θ_j's closer to unity and vice versa. Yet the effects of step size and μ on θ_j's were not pronounced enough to pay special attention to it.

In order to show the dynamic calculation converge to the right steady state, the resulting steady state column

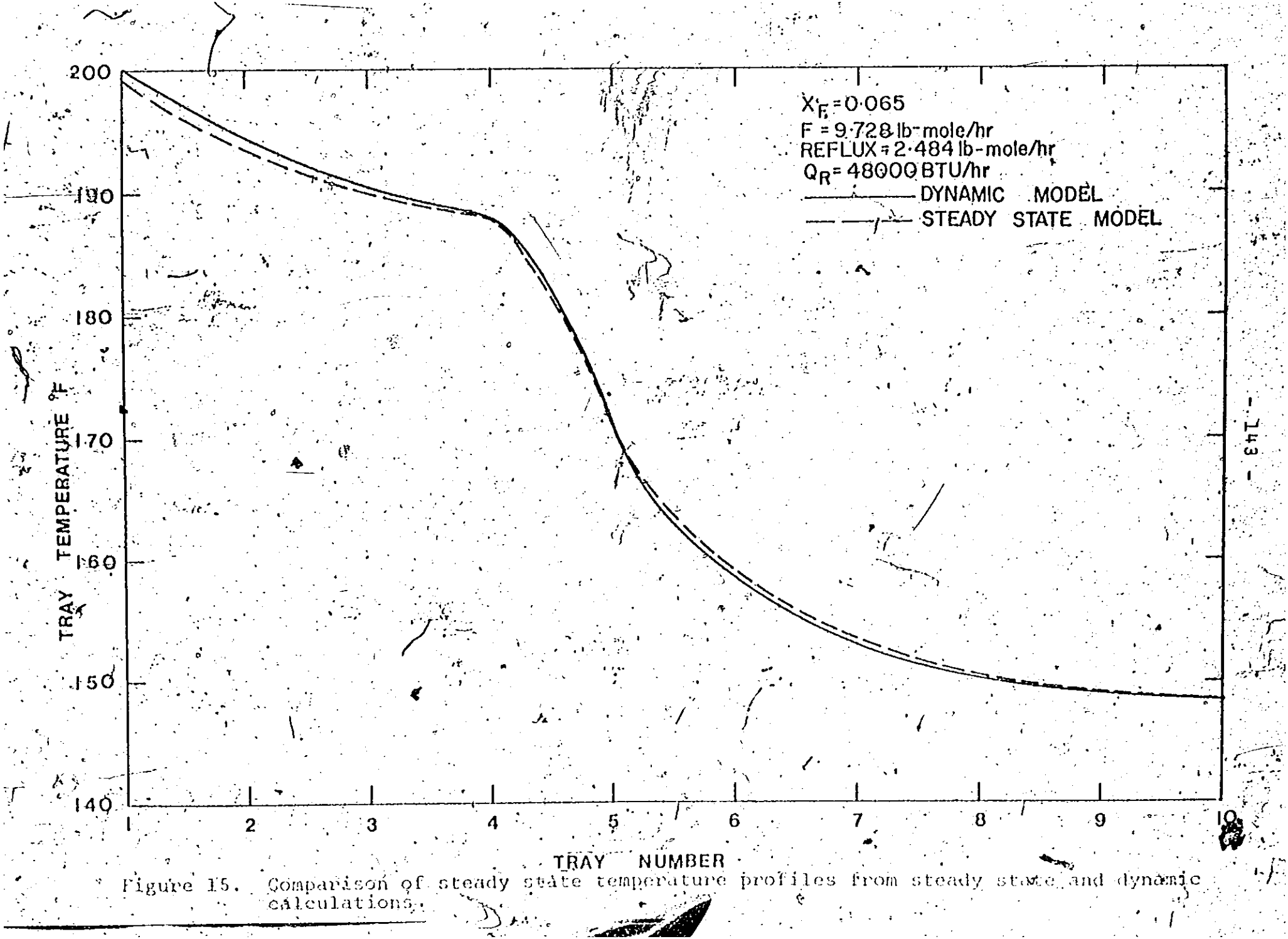


Figure 15. Comparison of steady state temperature profiles from steady state and dynamic calculations.

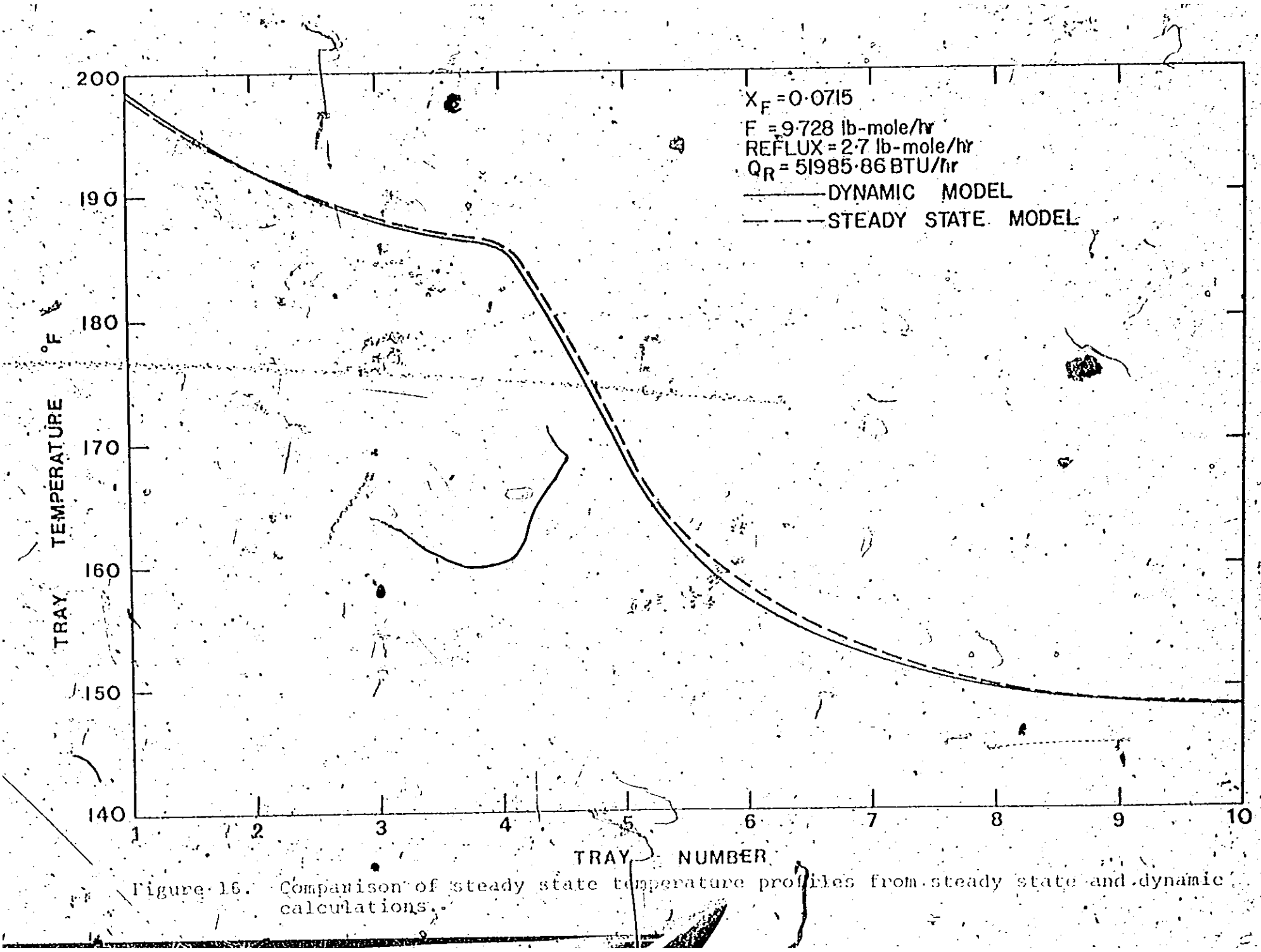


Figure 16. Comparison of steady state temperature profiles from steady state and dynamic calculations.

temperature profiles generated from the steady state and the dynamic models were plotted and shown in figures 15 and 16. Good agreement can be observed from these plots.

The linear dynamic model based on material balance was also used to generate the transient composition responses when a step change in the feed composition was introduced. These transient and steady state responses, generated from the linear dynamic model did not agree with the corresponding responses generated from the non-linear dynamic model. Figure (34) shows the dynamic responses of the compositions of tray No. 6 from the two mathematical models and the experimental results. This plot shows that the non-linear dynamic model is a better representation of the distillation column in this study.

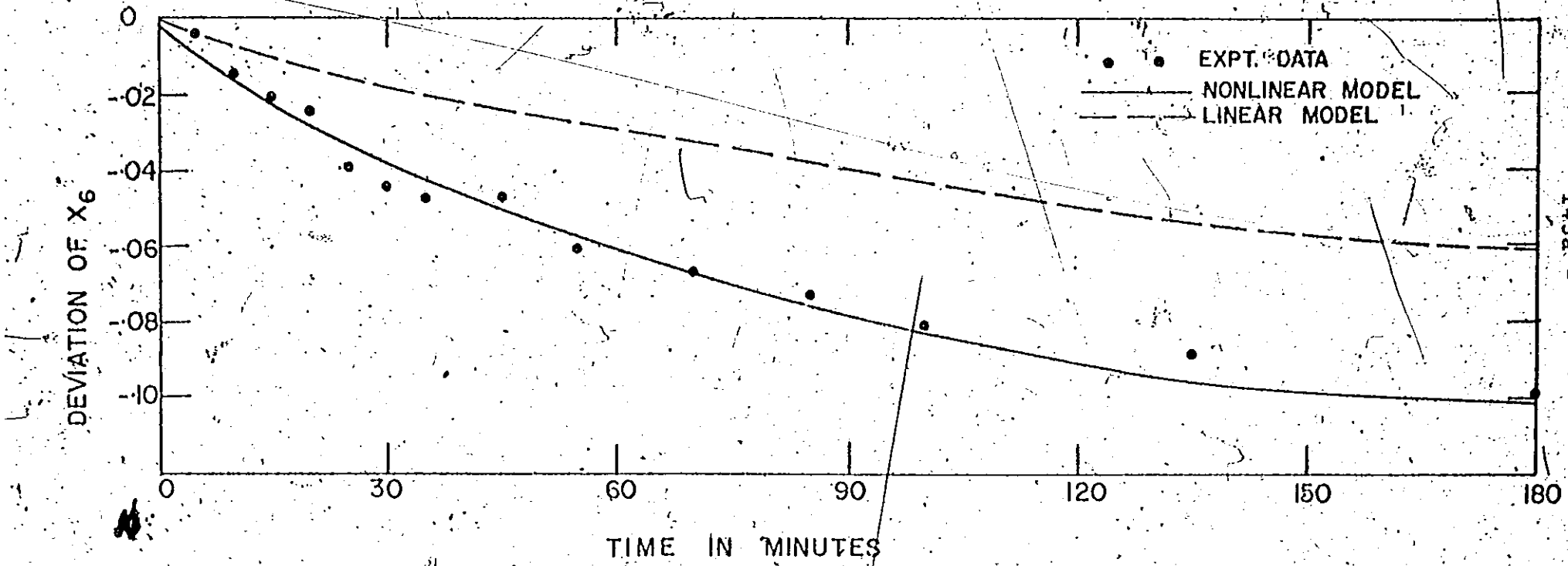


Figure 34. Comparison of the composition responses of tray No. 6 from the linear, non-linear dynamic models and the experimental result for a step decrease in feed composition (.0715 ⟶ .06812 mole fraction methanol)

D) EXPERIMENTAL TESTING

Steady state and transient experimental temperature profiles were obtained. These data were plotted in figures which have been discussed in the early part of this chapter.

It was observed that the end trays reached their new steady state values faster than the ones in the middle. The magnitudes of the changes were smaller for the trays at both ends than those in the middle. In most cases, the tray temperatures tended to oscillate around their steady state values rather than rest at these values when the new steady state arrived. The amplitudes of the oscillations varied from tray to tray. Those trays in the middle had larger amplitudes than the ones at both ends. In some cases, oscillations occurred only in the middle trays, the end trays did not show any detectable oscillation at all.

The feed concentration of the column was quite low. This meant that the amount of methanol in the column was fairly small. Tray temperatures were extremely sensitive to the two product flow rates. It was observed that a small change in the bottom product flow rate shifted considerably the steady state temperature profile either upward or downward.

No attempt was made to control the reflux temperature automatically. The hand valve and the rotameter that manipulated the flow of cooling water to the condenser was set at a fixed value such that the reflux temperature was close to its saturation temperature. As the cooling water temperature changed from day to day and also the water line pressure varied from time to time, it was very difficult to control the reflux temperature precisely. However, it was observed that the reflux temperature did not change too much, and the changes were always less than 10 °F. This might be due to the fact that the experiments were carried out at a time when the ambient temperature and the cooling water temperature were fairly constant. A better agreement between the experimental and the predict data can be expected if more strigent control of the reflux temperature is implemented.

E) SYNTHESIS OF THE OPTIMAL CONTROLLERS

In this study, quantitative results of the optimal controllers were generated using a linear model of the binary distillation column. Disturbances in feed composition and rate as well as product specification changes were studied.

The controllers were optimal in a sense that they minimized the performance index J_1 (equation 7-1). In this equation, there were four variables that could affect the performances of the controllers. These four variables were: t_f, the control period and the weighting matrices $\underline{R}$, $\underline{S}$ and $\underline{Q}$.

1) EFFECTS OF t_f ON THE OPTIMAL CONTROL AND STATE TRAJECTORIES

Of the four, t_f is tied in much closer to the physical situation than the others. Ideally, the faster it is for the product compositions to reach their desired values the better it is for operation of distillation column. Unfortunately, this was not the case in the practical situation. The value of t_f should not be smaller than the major time constant of the distillation column. Because, for a t_f which was smaller than the column's major time constant, the column just would not have enough time to bring the products to their desired compositions no matter how large the control action

was. From experimental dynamic testing, it was shown that the major time constant of the column was around 40 minutes. Also the t_f chosen in this study were longer than 40 minutes.

When t_f was small, more control action would be needed to achieve the goal. Yet there was a physical limitation to the amount of control action that could be generated. This was another reason for having a lower limit set for the value of t_f.

Figure 17 shows how the control period t_f affects the state and the control trajectories of the second optimal controller. This plot shows another reason why t_f could not be too small. When t_f was equal to 30 minutes, at the 30 minute mark, the positive and the negative deviations cancelled each other, and the deviation curve went away from the zero line instead of approaching it. Furthermore, at terminal time, the controls would be fixed at the new steady state values, and the product deviations would eventually return to zero. Then, there would be a large amount of off-specification material produced between time equal to 30 minutes and the time that the product deviation equal to zero.

This would ruin one of the purposes of this optimal controller.

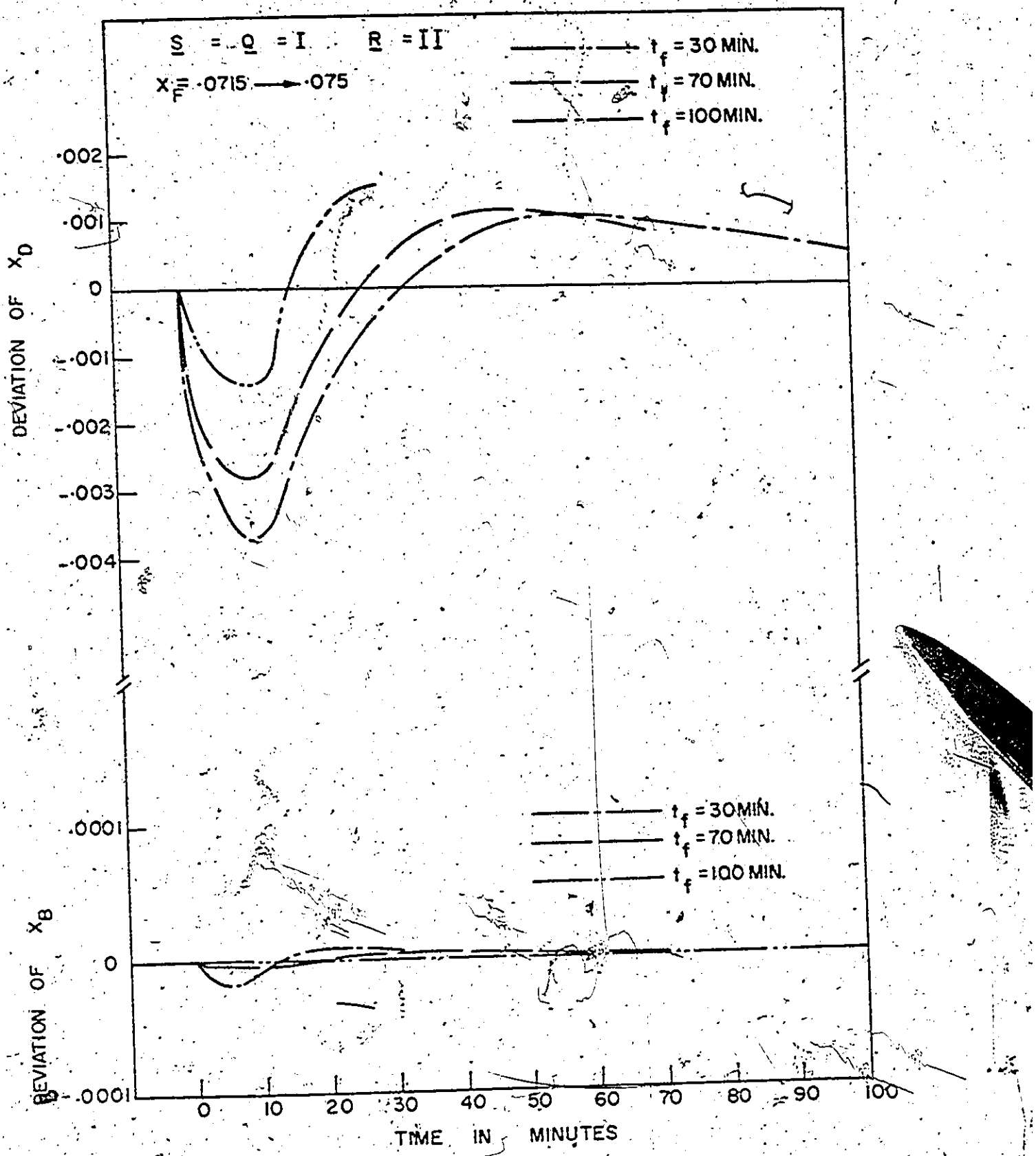


Figure 17. Comparison of top and bottom product deviations with different t_f (second controller).

An examination of the response curves for t_f equal to 70 and 100 minutes indicated that they were more desirable than the one with t_f = 30 minutes. For instance, the deviations at time equal to 70 minutes was much smaller. Even though the control would be set at the steady state value at terminal time, the amount of off specification material would be too small to be of any significance. For the case where t_f = 100 minutes, the performance was even better. For practical purposes, t_f should not be chosen too large because it might affect the operations of the downstream equipments.

2) EFFECTS OF $\underline{S}$, $\underline{Q}$, $\underline{R}$ ON THE OPTIMAL CONTROL AND STATE TRAJECTORIES

The results of this study showed that $\underline{S}$ did not exert as much influence on the optimal control and state trajectories as the other two matrices. This was

obvious because the prime purpose of $\underline{S}$ was to govern the product deviations at terminal time. As the value of $\underline{S}$ grew larger, the terminal product deviations would become smaller. In order to achieve this, more control action was needed. The resulting control would have a larger value. By decreasing the value of $\underline{S}$, the outcome would be just the opposite. Table 10 shows the effect of $\underline{S}$ on the terminal product deviations.

The ways that the matrix $\underline{Q}$ influenced the control and the state trajectories were more or less the same as $\underline{S}$ except to a greater extent. A larger $\underline{Q}$ implied smaller product deviations throughout the whole control period. Also, the magnitude of the control would grow as $\underline{Q}$ became larger. Figure 18 shows how the product deviations change with the value of $\underline{Q}$.

Among the three weighting matrices, $\underline{R}$ exerts more influence on the state and the control trajectories than any of the other two. The matrix $\underline{R}$ which can be viewed as the penalty function for the control vector also affects the state vector. Any change in the magnitude of the control will certainly result in a change of the state. When the value of $\underline{R}$ was small, the control was penalized lightly, in other words, the control would have more freedom to manoeuver. Because of this, smaller product deviations could be

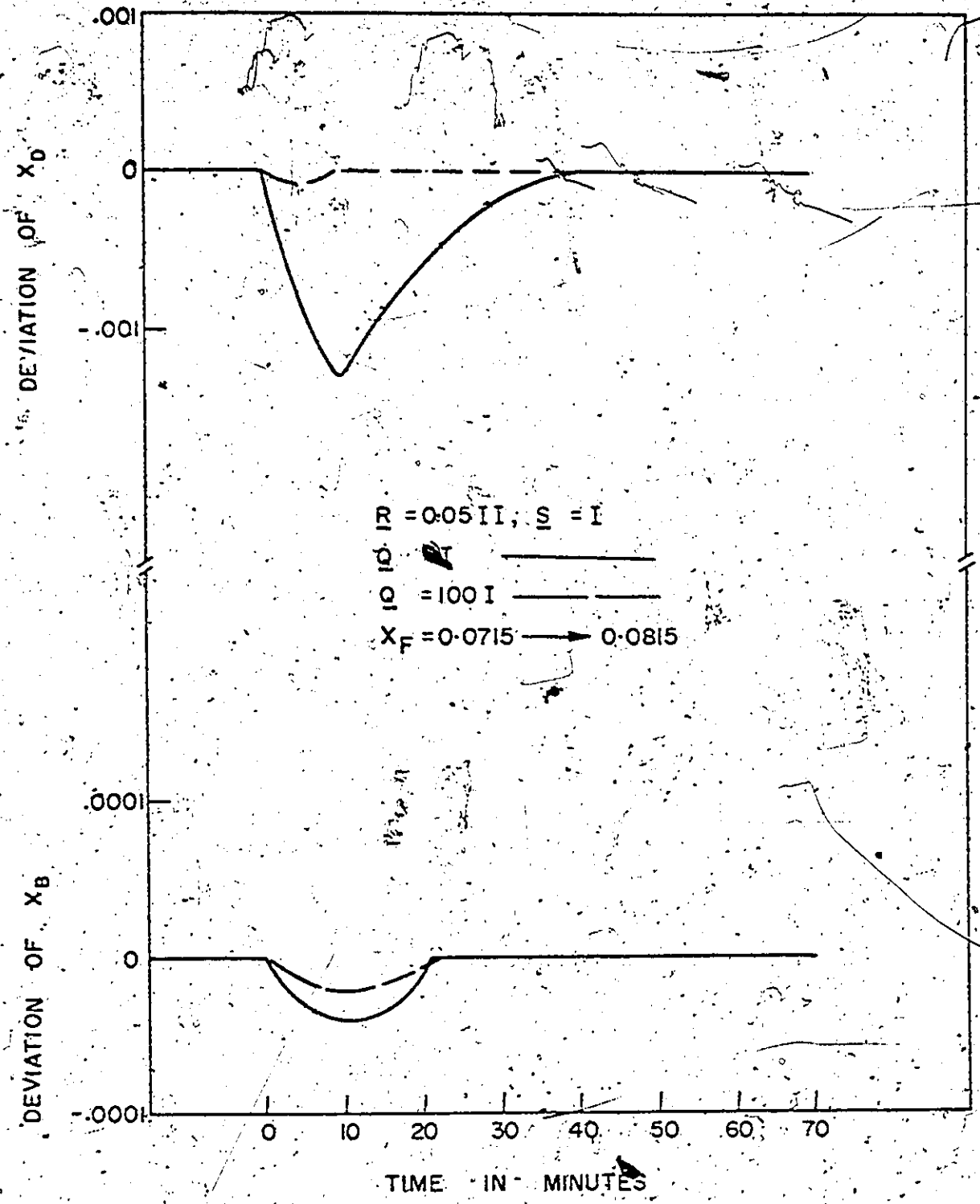


Figure 18. Comparison of top and bottom product deviations with different Q (first controller).

expected. When the value of R was large, it implied that the control action was heavily penalized. In this case, a small control as well as larger product deviations would result. Figure 19 shows how the control trajectories change with the values of R and Q. The magnitude of these three matrices also affected the rate of the response of the product deviations. For example, the product deviations would be driven to zero faster for a small R than for a large R. Increasing the values of S and Q would have the same effect on the rates of response of the product deviations as decreasing the value of R.

In the case of the second controller, satisfying equations (7-2) and (7-3) does not necessarily imply that the resulting product compositions from blending the products produced during the control period will be at the desired levels. This is because the product flow rates vary during the control period. However, the resulting values obtained from integrating the deviations of the component distillate rate ($D\,x_N(t)$) and the component bottom withdrawal rate ($Bx_1(t)$) between time zero and time t_f were very closed to zero. One of the cases is plotted in figure 20. It shows that the negative and the positive deviations did cancel each other. This probably will not be true when the magnitude of the disturbance is very large.

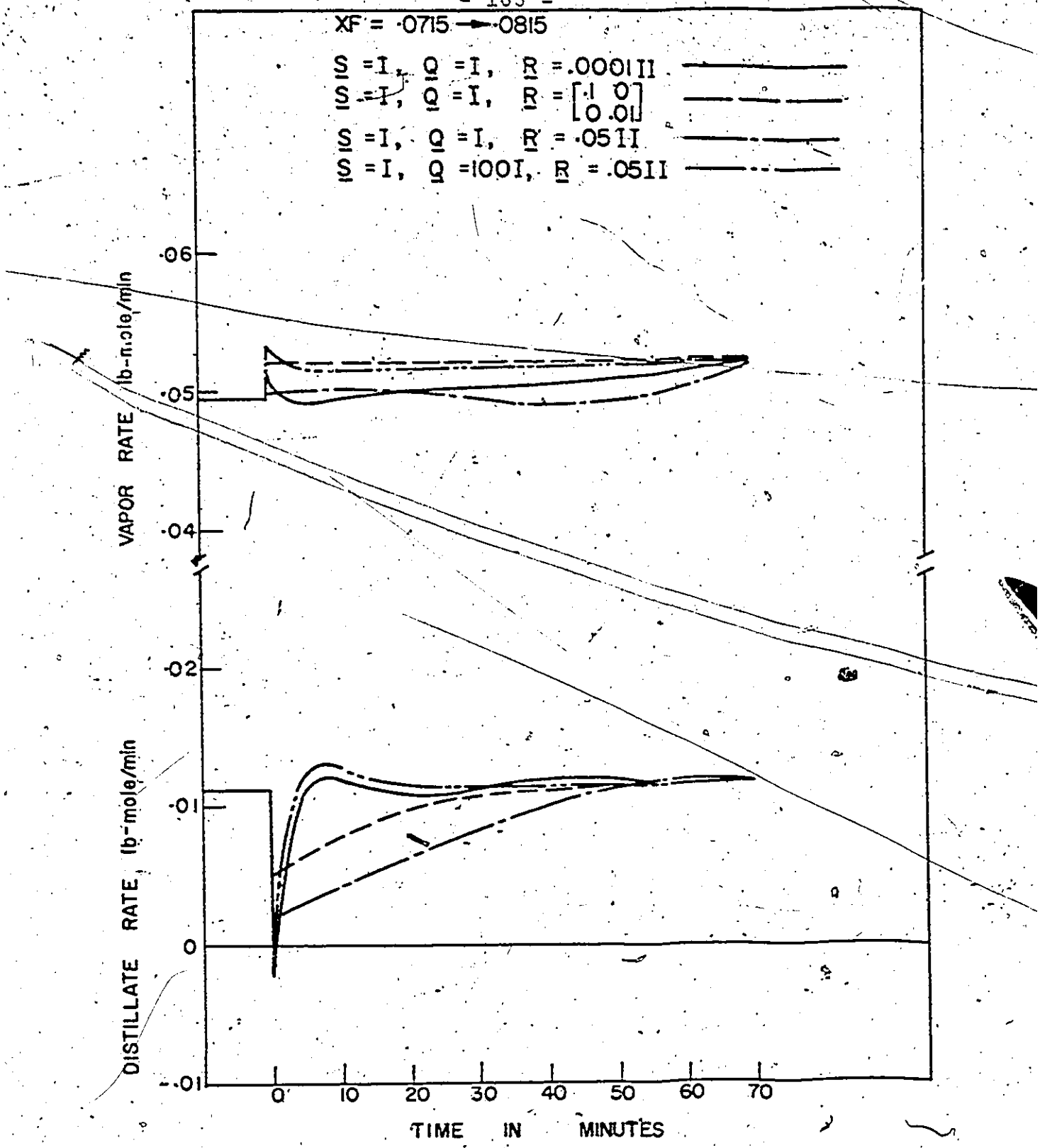


Figure 19. Comparison of optimal control trajectories with different $\underline{Q}$ (second controller).

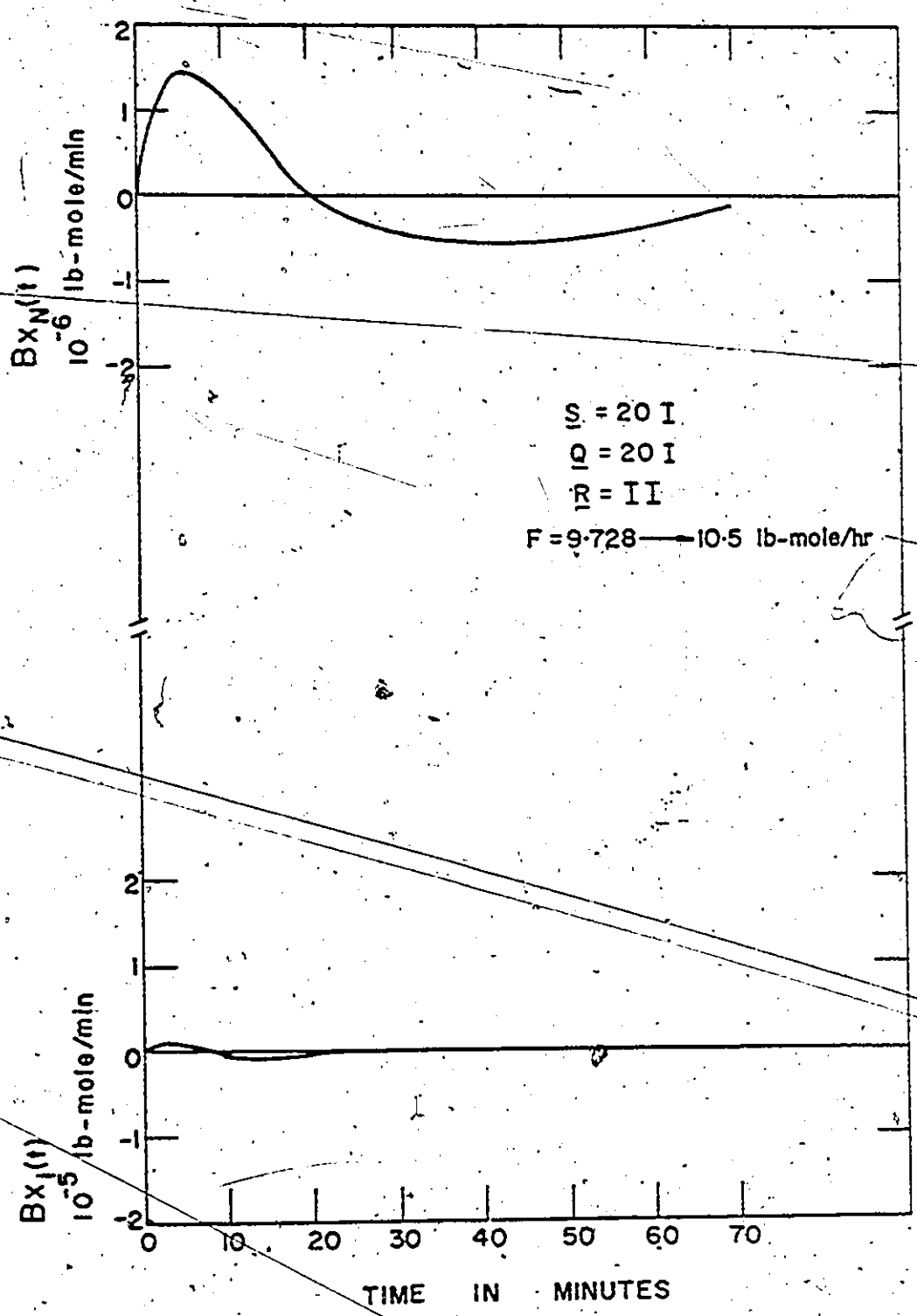


Figure 20. Deviations of the top and bottom product component withdrawal rates (second control).

If the distillate and the bottom product withdrawal rates are constant throughout the whole control period and equations (7-2) and (7-3) are satisfied, the products produced during this period will definitely have the desired average concentrations. This can be done by manipulating the weighting matrix $\underline{R}$ because either the top or the bottom product flow rate is always one of the manipulated variables. By heavily penalizing this particular product flow rate, it will remain at the desired steady state value throughout the whole period. For example, when the feed flow rate is increased from 9.728 to 10.5 lb-mole/hr and the reflux and the bottom product withdrawal rates are the control variables. By setting $\underline{R} = \begin{bmatrix} 1.0 & 0 \\ 0 & 100.0 \end{bmatrix}$ the resulting bottom product withdrawal rate is constant as shown in figure 21. Because of the nature of the control system, the top product withdrawal rate also remained constant at its final value.

Thus, the objectives of the control system are achieved by direct manipulation of the reflux rate. Yet, the performance of the controller is not affected by this type of manipulation at all, as can be observed from figure 22.

In this study, there was no constraint explicitly imposed on the control vector. The inherent danger of this practice was that the resulting optimal control

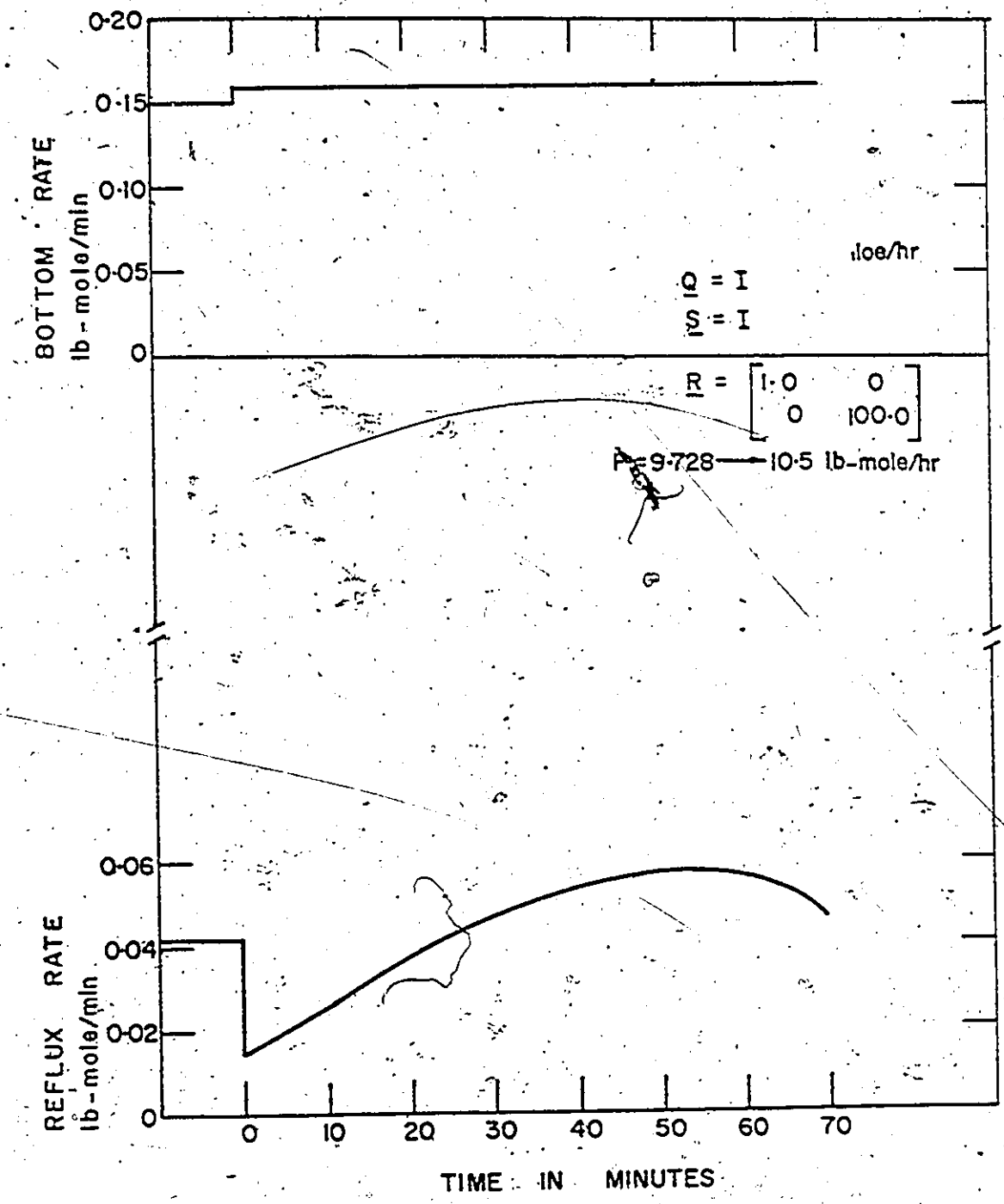


Figure 21. Optimal control trajectories of a case where the bottom product withdrawal rate was heavily penalized (second controller).

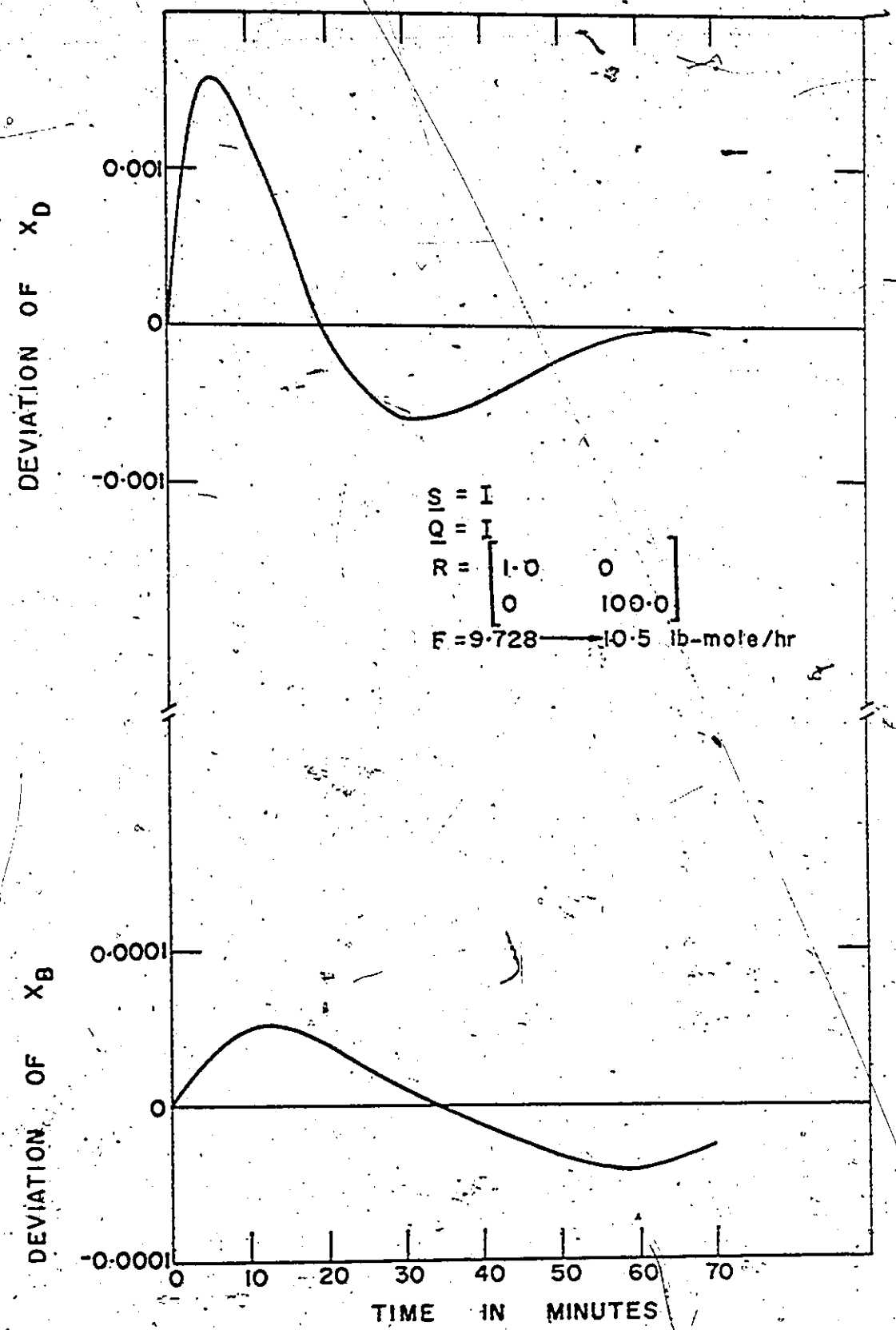


Figure 22. Optimal state trajectories of a case where the bottom product withdrawal rate was heavily penalized (second controller).

might be outside the physical limitation of the system. Fortunately, the optimal control could be regulated through manipulation of these three weighting matrices. Because of this, $\underline{S}$, $\underline{Q}$ and $\underline{R}$ could be regarded as indirect constraints to the control vector. The resulting optimal control and state trajectories of the first controller were more sensitive to the magnitudes of the weighting matrices than that of the second controller. This might be explained in the following way: The co-state vector $\underline{Z_c}$ of the second controller changed with the changes of the matrices $\underline{S}$, $\underline{Q}$ and $\underline{R}$. The change in the value of $\underline{Z_c}$ might be enough to offset the changes of the weighting matrices in such a way that the resulting trajectories would remain fairly intact. In the case of the first controller, the co-state vector $\underline{Z_c}$ did not exist. Hence, the resulting optimal control and state trajectories would be more sensitive to the changes of the three weighting matrices. When $\underline{R}$ approached zero and (or) $\underline{S}$ and $\underline{Q}$ approached infinity. The absolute value of the co-state vector $\underline{Z_c}$ approached zero and the performances of the two controllers tended to be identical with each other. In the process of synthesizing the first controller, the co-state vector $\underline{Z_c}$ did not appear in the equations. It did appear in the second controller because of the two integral constraints. Therefore, when $\underline{Z_c}$

approached zero, it would be the same as the case where there was no integral constraint at all. When this happened, the two optimal controllers tended to be identical to each other and also their performances.

3) EFFECTS OF THE INITIAL STATE VALUES ON THE OPTIMAL CONTROL TRAJECTORIES

Another important fact observed from the results of this study was that when the values of R, S, Q and t_f were fixed, the resulting optimal control trajectory had a definite trend and shape according to the sign of the initial values of the state vector. For example, when the feed composition decreased from .0715 to .065 mole fraction (methanol), all the initial values of the state vector had positive signs. The trajectories of the control vector (reflux and bottom product flow rates) jumped to certain values and then increased gradually to their steady state values. When the disturbance entered the column in the form of feed rate increased, (9.728 to 10.5 lb-mole/hr), the initial values of the state vector were all positive. The resulting optimal trajectories of the same control vector had identical trend and shape as in the previous case. On the contrary, when the feed concentration increased from .0715 to .0815 mole fraction (methanol), the initial values of the state

vector were mixed with positive and negative values. The resulting optimal control trajectories of the same control vector jumped to certain values and then decreased gradually to their steady state values. Figure 23 shows how these control trajectories vary with the signs of their corresponding initial values of the state vector. As soon as the type and magnitude of the feed disturbance are known, the initial values of the state vector can be calculated. Then the resulting optimal control trajectories can be estimated without recourse to the lengthy calculation. Figure 24 shows the state and the control trajectories of the second controller when a change in the top product composition is demanded. In this case, signs of the initial values of the state vector were all negative. The resulting optimal control trajectories followed the same trend as in the case of increased feed composition.

Figures 25 and 26 show the optimal state and control trajectories of other cases of the optimal change over policy.

In the case of the first controller, the liquid tray holdups had no bearing on the resulting optimal control. This was mentioned by Brosilow and Handley (87) in their work. Table 11 shows the values of the resulting optimal control vectors with constant S R Q

and t_f. The type and magnitude of the disturbance were the same but the values of the liquid tray holdups were different. However, the magnitudes of the liquid tray holdups did affect the optimal state trajectories.

F) VALIDITY TEST FOR THE OPTIMAL CONTROLLERS WITH THE NON-LINEAR DYNAMIC MODEL

The design of the two optimal controllers was based on a linear model of the distillation column. However, it was understood that the linear model could not truly represent the operation of the binary distillation column. Good results with the linear model did not necessarily imply that good results would be obtained from the actual distillation column when these control strategies were applied, since it has been proved that the non-linear mathematical model could closely represent the actual column operation. It was logical to apply these optimal control schemes to the non-linear model in order to check their validities.

Figure 27 shows the comparison of the product deviations between the two models with the first controller. It is observed that the performance of the controller deteriorates for the non-linear model. But the product composition deviations are still very much acceptable. It can be said that the optimal control trajectories generated from the linear dynamic model are good enough for the real

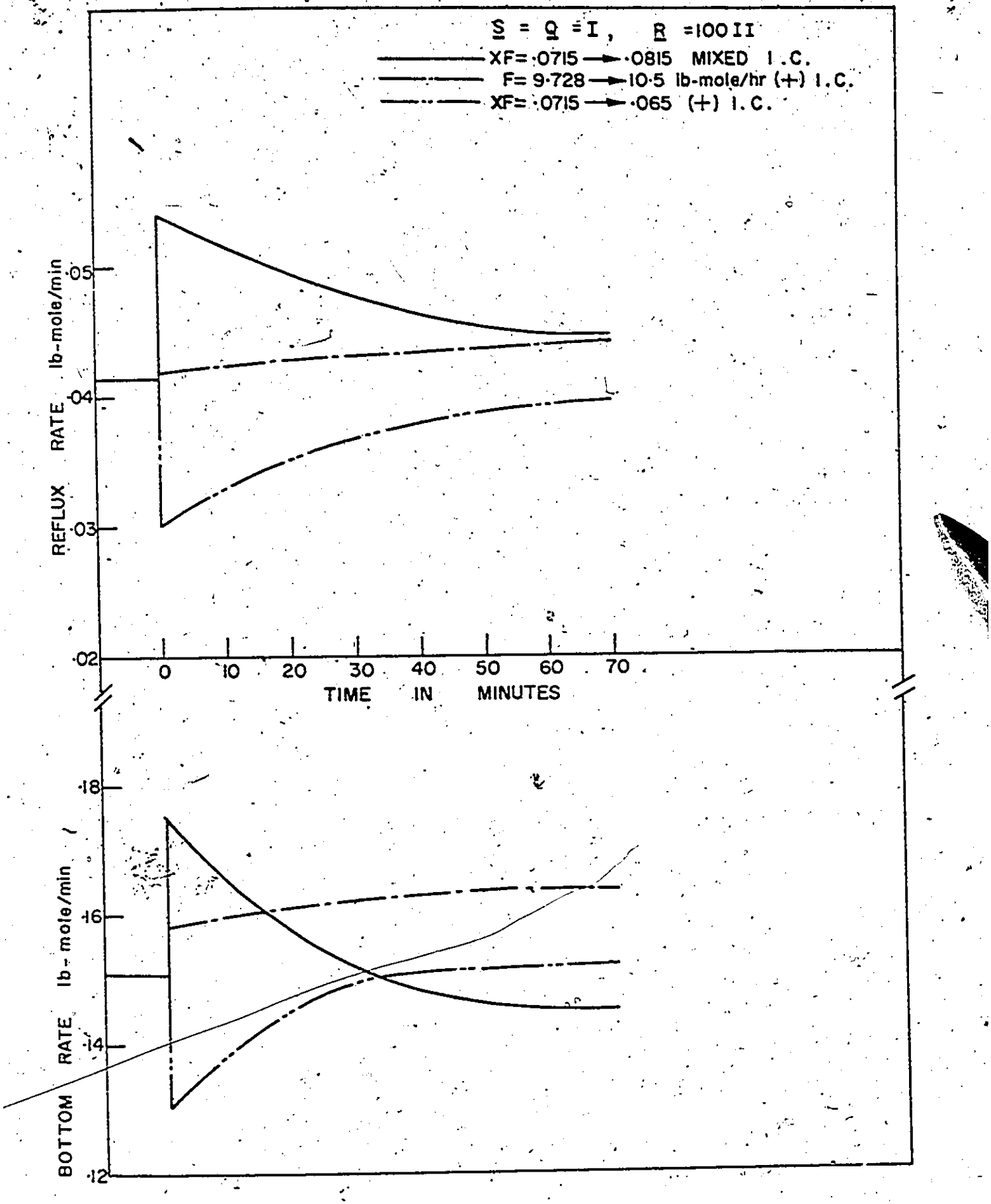


Figure 23. Trends of the optimal control trajectories (second controller).

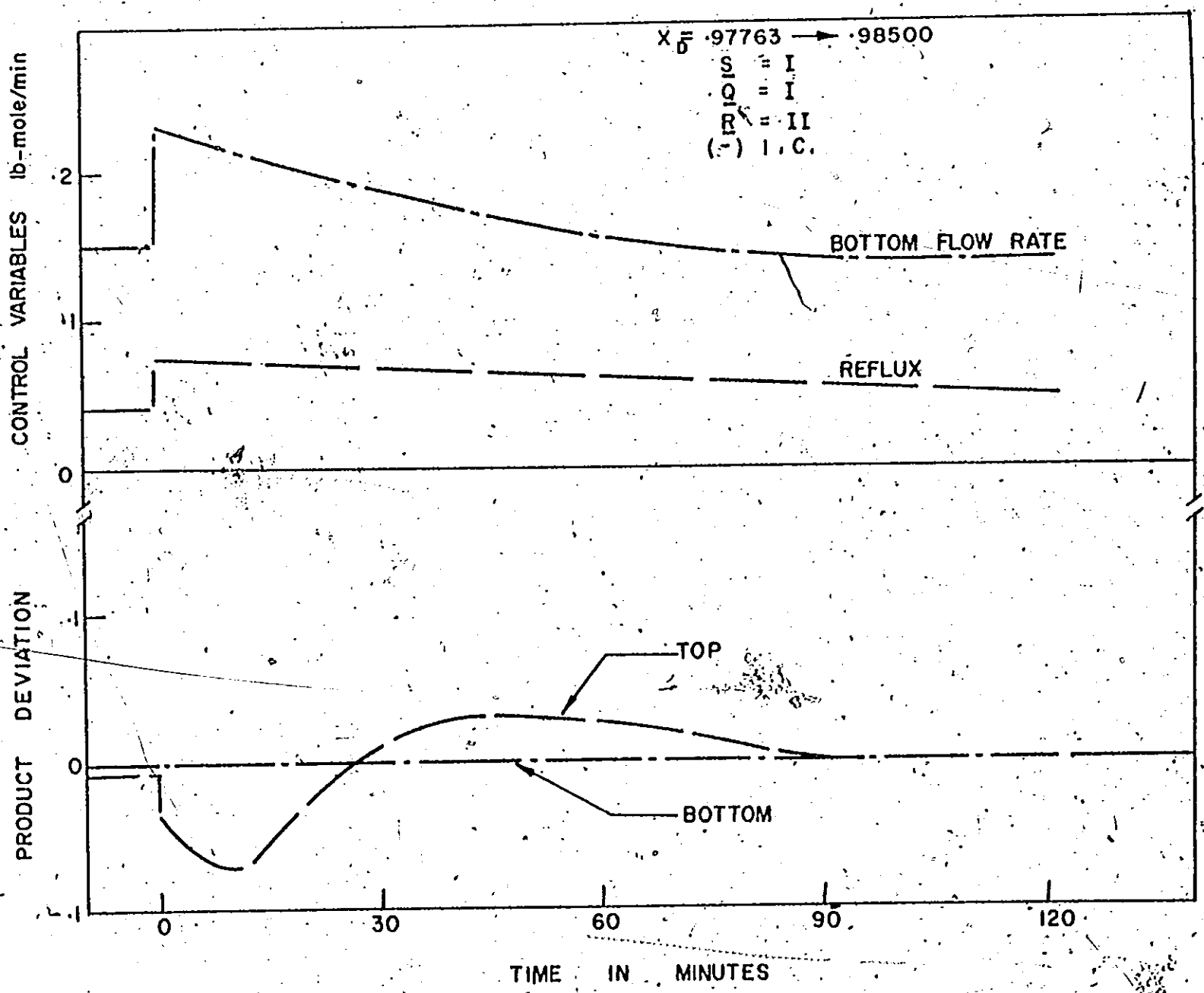


Figure 24. Optimal control and state trajectories of a change over policy (second controller).

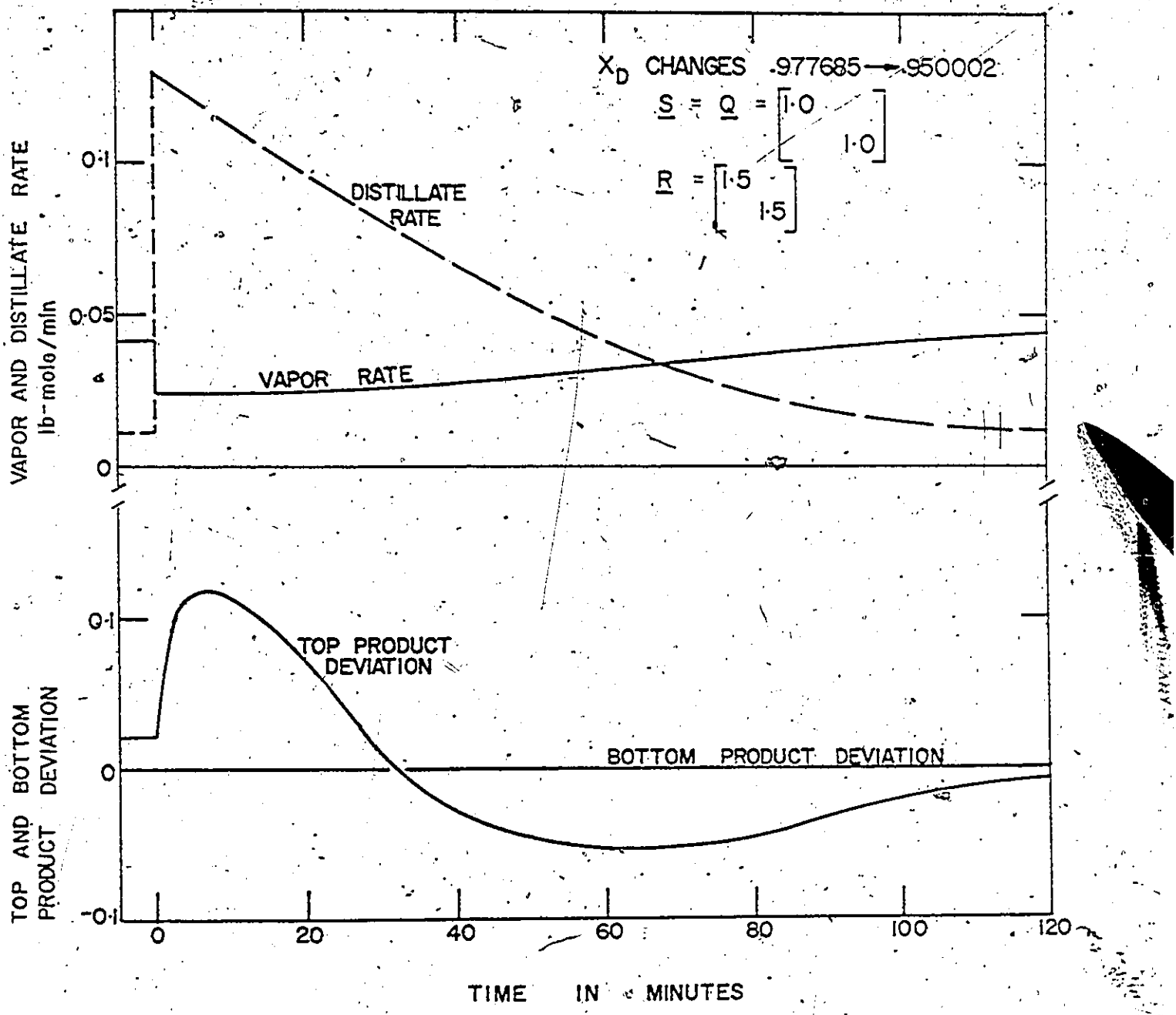


Figure 25. Optimal control and state trajectories of a change over policy (second controller).

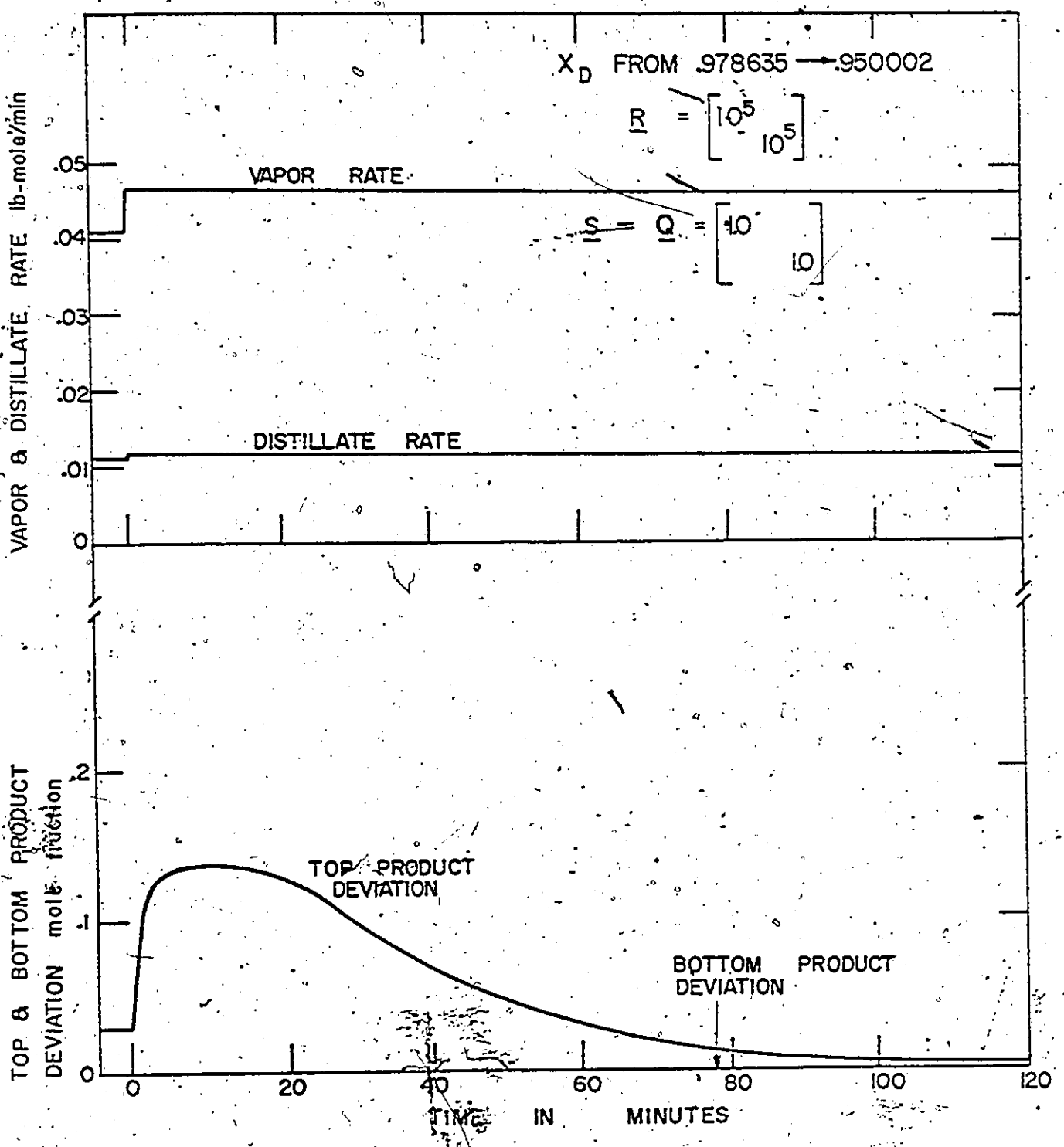


Figure 26. Optimal control and state trajectories of a change over policy (first controller).

situation. The optimal control trajectories generating from the second controller were also tested with the non-linear model. The resulting product deviations' responses were not as satisfactory as in the case of the first controller. This might be on account of the presence of the integral constraints. The dynamic information of the column would be more critical to the performance of the second controller than to the first one.

G.) COMPARISON OF THE OPEN AND CLOSED LOOP RESPONSES OF THE NON-LINEAR DYNAMIC MODEL

Figure (35) shows the simulated open and closed loop responses of the top and bottom product composition deviations under step disturbance in feed composition using the non-linear column model.

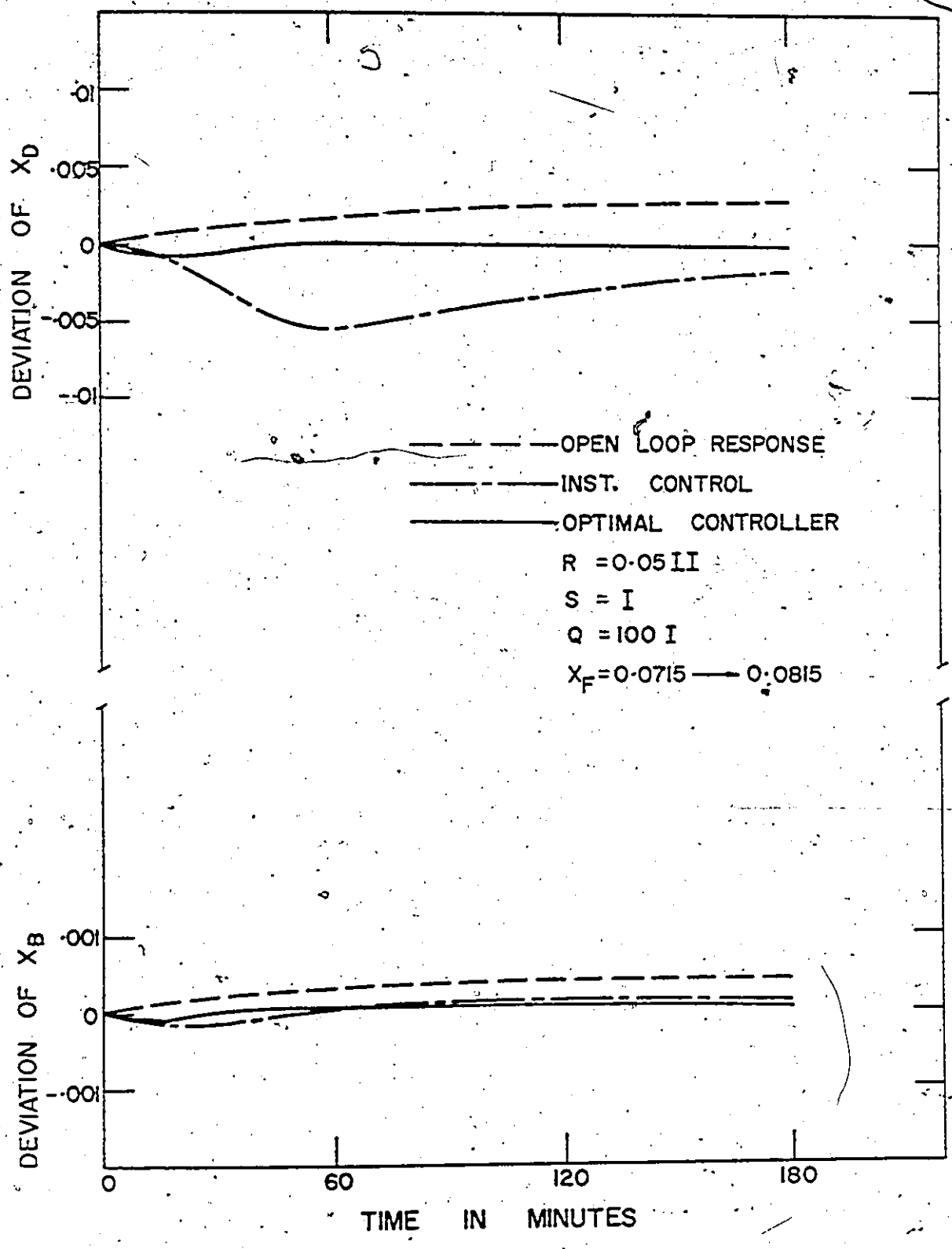
The open loop response is the response if no control action is taken. Two types of closed loop responses are shown on the figure; instantaneous feedforward control and optimal control (first controller). Instantaneous feedforward control consists of merely changing the manipulated variables to their final control values the instant the feed disturbance enters the column. The figure demonstrates the effectiveness of the optimal control policy. This type of results has also been reported for feedforward control in which instantaneous feedforward control is compared with feedforward control with dynamic compensation (72).

A comparison of the instantaneous control with the second control policy is not included because they are not considered to be equal alternatives.

FIGURE 35

COMPARISON OF THE OPEN AND CLOSED LOOP RESPONSES OF THE NON-LINEAR DYNAMIC MODEL UNDER STEP INCREASE IN FEED COMPOSITION

(.0715 ⟶ 0.0815 mole fraction methanol)


DEVIATION OF X_D
·01
·005
0
-·005
-·01
OPEN LOOP RESPONSE
INST. CONTROL
OPTIMAL CONTROLLER
R = 0·05 II
S = I
Q = 100 I
X_F = 0·0715 ⟶ 0·0815
DEVIATION OF X_B
·001
0
-·001
0
60
120
180
TIME IN MINUTES

H) GENERAL COMMENTS

The linear synthesis and quadratic performance index approach used in this study is a valuable process for constructing optimal feedforward controllers. It is a versatile method because it handles both the set-point and load disturbances in one formulation. This linear synthesis approach is more attractive than the non-linear synthesis from the numerical calculation point of view especially for high-order systems. One of the non-linear synthesis methods is discussed in appendix B. This method was applied to the problem in this study. The result was not satisfactory because of the instability of the numerical calculation. On the other hand, the linear synthesis approach guaranteed an optimal solution and the numerical calculation was straightforward.

Until more efficient and sophisticated numerical procedures have been developed, the approach used in this work is a viable alternative for the synthesis of optimal controllers of high-order non-linear system.

Because of the large system of equations involved in synthesizing the optimal controller, the optimal control trajectories could only be generated using a high speed computer with a large memory (i.e. IBM 360/65). Thus it is not economical to implement these optimal control schemes directly for on-line direct digital computer control. However, it is conceivable that simple algebraic relationships between the optimal control trajectories and the disturbances

could be established. These simplified algebraic expressions could then be used as the control law and implemented on a digital control computer for on-line direct digital computer control of the distillation column operation. By using this approach, the optimal controller synthesis described in this study can be viewed as a very valuable intermediate step for the development of optimal on-line direct digital computer control of a distillation column.

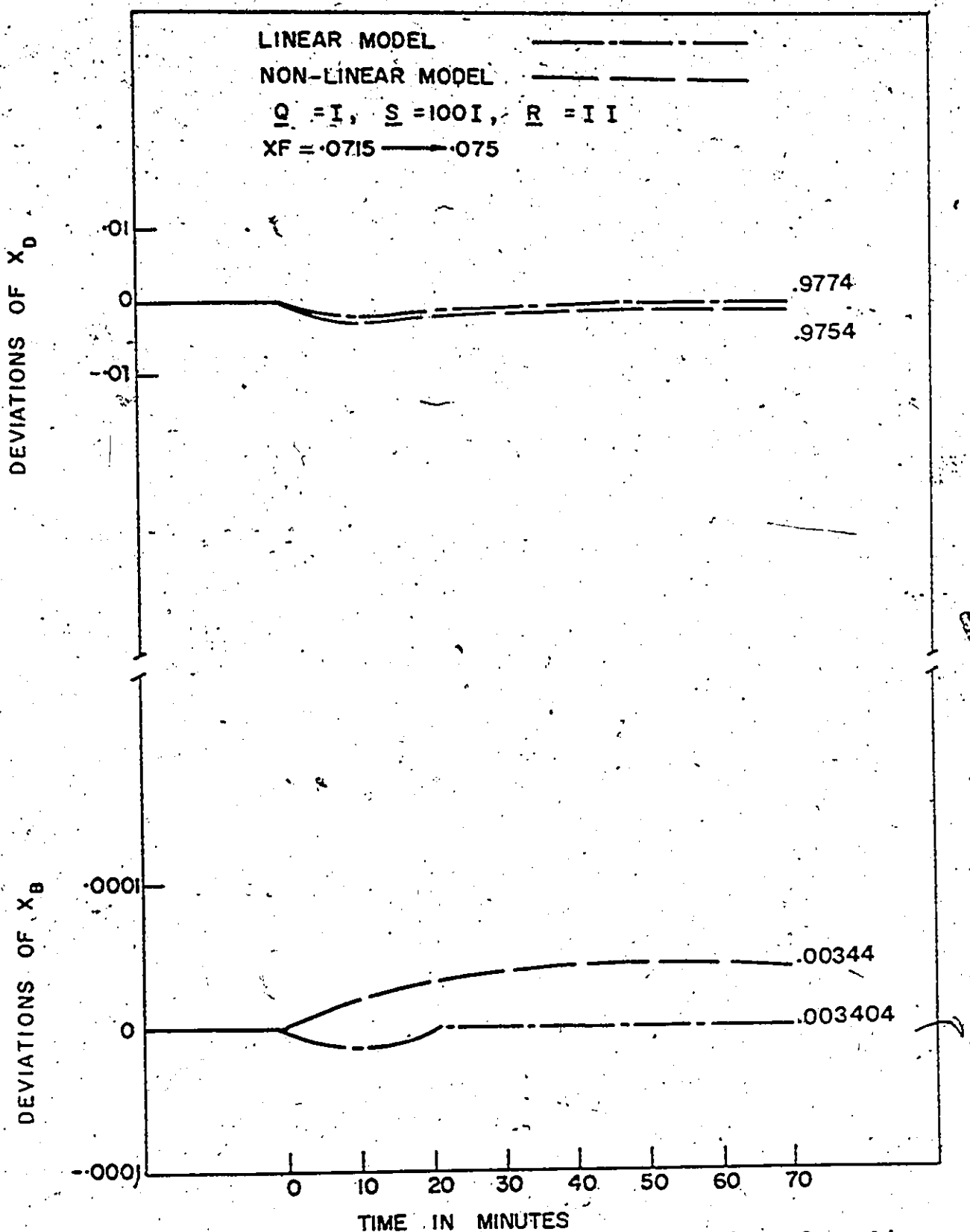


Figure 27. - Comparison of product deviations from linear and non linear model (first policy).

CHAPTER X

CONCLUSIONS AND RECOMMENDATIONS

The dynamic study in this work revealed that the binary distillation column with methanol-water system could adequately be represented by a non-linear mathematical model based on material and energy balance. The simplified linear model which was based on component material balance could not truly describe the transient behavior of the binary distillation column operation. The combination of the implicit method and the θ-method of convergence have proven to be a better method than the conventional direct numerical integration of the differential equations which constituted the dynamic model.

The steady state model used in this study has also been proved that it could aptly describe the steady state column behavior. The combination of the matrix method and constant composition method for enthalpy balance was not superior to the methods by Thiele-Geddes and Lewis-Methason, but at least it was comparable. The basic advantage of this matrix method was it solved the material balance directly rather than the tray to tray calculation as in the other processes. Also, the number of variables which had to be assumed to start the calculation were not as many as in the other methods.

Among the four search procedures tested in the steady

state two-point control calculation, the false position method was best suited for this particular computation. This does not imply that the other three methods are not good. They can be the best on some other occasions. For instance, the pattern search is good for cases where a large number of variables have to be searched.

The optimal two-point control of this binary distillation column under various feed disturbances and product compositions changes have been investigated. The minimum principle of Pontryagin together with the modified Riccati transformation were used to solve this control problem. A unique feature of this Riccati transformation was that the resulting optimal control was always a function of the state vector. Thus, it made the control problem a feedback type in nature.

Two optimal controllers based on different control philosophies were synthesized and their performances evaluated. Even though the simplified linear dynamic model was used to construct the optimal controllers, the resulting optimal control trajectories worked well when applied to the non-linear model.

When the control vector was allowed to move freely, and the deviations of the state were penalized severely, the performances of the two controllers tended to become identical. The co-state vector $\underline{Z}_c$ would approach zero and this implied that the two integral constraints were not

enforced. So, there was no difference between these two controllers and they would tend to become identical. This extreme case might not be obtainable in reality due to the physical limitation of the column.

The second optimal controller was not as sensitive to the weighting matrices $\underline{S}$, $\underline{Q}$ and $\underline{R}$ as the first one. Usually, more control action was needed for the second controller than for the first one to combat the same type and amount of disturbance. The two additional integral constraints did change the performances of the controller.

The linear synthesis and quadratic performance index technique used in this study resulted in a satisfactory optimal controller. This technique is versatile from the design of the optimal feedforward controller point of view because it is able to handle the set-point and load disturbance problems in one mathematical formulation. From the numerical calculation point of view, the method is straightforward and stable. The optimal solution is guaranteed by the properties of linear system with quadratic performance index. The more complicated non-linear synthesis does not possess this salient property and is also highly dependent on the numerical procedures. This is especially true for high-order system like the distillation column in this work.

It is not practical or economical to implement the optimal control synthesis described in this work directly on a present day on-line digital computer. However, it is a

valuable intermediate step for obtaining simpler control laws which can be readily implemented for on-line control.

It is recommended that future works should be focused on the following areas:

1) The simplified optimal control schemes obtained from the optimal control calculation should be verified experimentally. This can be done by implementing these control schemes with a control computer which is connected to the existing column.

2) Try to put actual constraints on the control vector rather than setting them through the weighting matrices $\underline{R}$, $\underline{S}$ and $\underline{Q}$.

3) The time optimal control problem that leads to the famous "bang-bang" control should be tried and then checked for the feasibility of actual implementation of this simple control scheme on the existing distillation column.

REFERENCES

Holland, C.D., 'Unsteady State Processes with Application in Multicomponent Distillation', Prentice-Hall Inc., (1966).

2. Lapidus, L., 'Digital Computation for Chemical Engineers', McGraw-Hill Book Co., (1962).

3. Smith, B.D., 'Design of Equilibrium Stage Processes', McGraw-Hill Book Co., (1963).

4. Lyster, W.N., Sullivan Jr., S.L., Billingsley, D.S., and Holland, C.D., Petroleum Refiner, 38 (5), 271 (1959); 38 (7), 151 (1959); 38 (10), 139 (1959).

5. Amundson, N.R., and Pontinen, A.J., Ind. Eng. Chem. 50, 730 (1958).

6. Rose, A., Sweeny, R.F., and Schrodt, V.N., Ind. Eng. Chem. 50, 737 (1958).

7. Holland, C.D., 'Multicomponent Distillation', Prentice-Hall Inc., (1963).

8. Ball, W.E., paper presented at the Machine Computation Special Workshop Session on Multicomponent Distillation at the 44th National Meeting of the A.I.Ch.E. in New Orleans, La. (Feb. 1961).

9. Peiser, A.M., Chemical Engineering, 67 129 (July 1960).

10. Bonner, J.S., American Petroleum Institute Quarterly, Division of Refining, 36 No. 3, 238 (1956).

11. Greenstadt, J., Bard, Y., and Morse, B., Ind. Eng. Chem., 50 1644 (1958).

12. Tomich, J.F., A.I.Ch.E., 16 No. 2, 229 (1970).

13. Billingsley, D.S., A.I.Ch.E., 16 No. 3, 441 (1970).

14. Gentry, J.W., Can. J. of Chem. Eng., 48 No. 4, 451 (1970).

15. Seppala, R.E., and Luus, R., J. of the Franklin Inst., 293 325 (1972).

16. Van Winkle, 'Distillation', McGraw-Hill Book Co., (1967).

17. Marshall, W.R., and Pigford, R.L., 'The Application of Differential Equations to Chemical Engineering Problems', University of Delaware, Newark, Del. 146 (1947).

18. Lapidus, L. and Amundson, N.R., Ind. Eng. Chem., 42 1070 (1950).

19. Mickley, H.S., Sherwood, T.K., and Reed, C.E., 'Applied Mathematics in Chemical Engineering', McGraw-Hill Book Co., (1957).

20. Wilkinson, W.L. and Armstrong, W.D., 'Plant and Process Dynamic Characteristics, New York; Academic, (1957).

21. Wilkinson, W.L. and Armstrong, W.D., Chem. Eng. Sci., 7 (1957).

22. Armstrong, W.D. and Wood, R.M., Trans. Inst. Chem. Engrs., 39, 65 (1961).

23. Votter, H., 'Plant and Process Dynamic Characteristics', London: Butterworths, 73 (1957).

24. Osborne Jr., W.G. and Maddox, R.N. 'System and Process Control', a CEP Technical Manual, A.I.Ch.E. 119 (1967).

25. Pohjola, V.J. and Norden, H.V., Chem. Eng. Sci., 24 1687 (1969).

26. Acrivos, A. and Amundson, N.R., Ind. Eng. Chem. 45, 467 (1953).

27. Yesberg, D., and Johnson, A.I., Can. J. Chem. Eng. 46 No. 2 (1960).

28. Pigford, R.L., Tepe, J.B. and Garrahan, C.J., Ind. Eng. Chem. 43, 2592 (1951).

29. Rose, A., and Williams, T.J., Ind. Eng. Chem., 47, 2284 (1955).

30. Rose, A., Johnson, R.C., and Williams, T.J., Ind. Eng. Chem., 48, 1173 (1956).

31. Lamb, D.E., Pigford, R.L., and Rippin, D.W.T., Chem. Eng. Progr. Symposium Series, 57, No. 36, 132 (1961).

32. Rijnsdorp, J.E., and Maarleveld, A., 'Symposium on Instrumentation in Process Development and Plant Design', Institution of Chemical Engineers, Society of Instrument Technology and British Computer Society, London, England, (May, 1959).

33. Rademaker, O., and Rijnsdorp, J.E., paper presented at the 1959 Fifth World Petroleum Congress, N.Y. (May 1959).

34. Luyben, W.L., Vernevill, V.S., and Gerster, J.A., A.I.Ch.E.J. 10, 357 (1964).

35. Jackson, R.F., and Pigford, R.L., Ind. Eng. Chem. 48, 1020 (1956).

36. Gilliland, E.R., and Mohr, C.M., Chem. Eng. Progr. 58, 59 (1962).

37. Rosenbrock, H.H., Trans. Instn. Chem. Engrs. 35, 347 (1957).

38. Albritton, J.A., and Murrill, P.W., paper presented at the 55th National Meeting of the American Institute of Chemical Engineers, Houston, Texas, (Feb. 1965).

39. Sweeney, R.F., paper presented at the Machine Computation Special Workshop Session on Multicomponent Distillation at the 44th National Meeting of the A.I.Ch.E. in New Orleans, La. (Feb. 1961).

40. Acrivos, A., and Amundson, N.R., Ind. Eng. Chem. 47, 1533 (1955).

41. Frazer, B., Duncan, W.T., and Collar, A.R. 'Elementary Matrices and Some Applications to Dynamics and Differential Equations.' Cambridge University Press, (1938).

42. Mah, R.S.H., Michaelson, M., and Sargent, R.W.H., Chem. Eng. Sci. 17, 619 (1962).

43. Sargent, R.W.H., Trans. Instn. Chem. Engrs. 41, 51 (1963).

44. Waggoner, R.C., Ph.D. Dissertation, 1964, Texas A & M University.

45. Duffin, J.H., Paper presented at the 56th National Meeting of the American Institute of Chemical Engineers, San Francisco, Calif., (May, 1965).

46. Duffin, J.H., and Gamer, J.D., Paper presented at the 58th National Meeting of the American Institute of Chemical Engineers, Dallas, Texas, (Feb., 1966).

47. Gill, S., Camb. Phil. Soc. Proc. 47, 96 (1951).

48. Huckaba, C.E., and Franke, F.R., Chem. Eng. Progr. Symposium Series 59 No. 46, 38 (1963).

49. Huckaba, C.E., Franke, F.R., and May R.P., Paper presented at the 55th Annual Meeting of A.I.Ch.E., Chicago, Ill. (Dec. 1962).

50. Distefano, G.P., Ph.D. Dissertation, 1964, University of Florida.

51. Rosenbrock, H.H., Tavendal, A.B., Storey, C., and Challis, J.A., Proc. First Intern. Congr. Federation Automatic Control, (J.F. Coales, ed.) Vol. IV London: Butterworths, 303 (1961).

52. Waggoner, R.C., and Holland, C.D., A.I.Ch. E. J. 11, 112 (1965).

53. Rosenbrock, H.H., Brit. Chem. Eng. 3, 364 (1958).

54. Rosenbrock, H.H., Brit. Chem. Eng. 3, 432 (1958).

55. Rosenbrock, H.H., Brit. Chem. Eng. 3, 491 (1958).

56. Tetlow, N.J., Ph.D. Dissertation, 1966, Texas A & M University.

57. Tetlow, N.J., Groves, D.M., and Holland, C.D., A.I.Ch.E.J. 13, 476 (1967).

58. Picou, C.F., Ph.D. Dissertation, 1967, Tulane University.

59. Harriott, P. 'Process Control', McGraw-Hill Book Co. (1964).

60. Hoffman, H.L., Hydrocarbon Processing & Petroleum Refiner 42, 108 (1963).

61. Shinskey, F.G., 'Process Control System' McGraw-Hill Book Co., (1967).

62. Buckley, P.S., 'Techniques of Process Control', John Wiley & Sons, (1964).

63. Rosenbrock, H.H., Trans. Inst. Chem. Engrs. 40, 35 (1962).

64. Rijnsdorp, J.E., Chem. Eng. Progr. 63 No. 7, 97 (1967).

65. Rijnsdorp, J.E. Automatica 1, 15 (1965).

66. Davisory, E.J., Automatica 6, 447 (1970).

67. Luyben, W.L., A.I.Ch.E. Journal 16, 198 (1970).

68. Wood, R.K., and Berry, M.W., Paper presented at the 21st Canadian Chemical Engineering Conference, Montreal, Que. (Oct. 1971).

69. Lupfer, D.E., and Parsons, J.R., Chem. Eng. Progr. 58, No. 9, 37 (1962).

70. Lupfer, D.E., and Johnson, M.L., ISA Trans. 3, No. 2, 165 (1964).

71. Weiss, E.A., Archer, D.H., and Burt, D.A. Hydrocarbon Processing & Petroleum Refiner 40 No. 10, 169 (1961).

72. Luyben, W.L., and Gerster, J.A., Ind. Eng. Chem., Proc. Design and Dev., 3, No. 4, 374 (1964).

73. Shinskey, F.G., The Oil and Gas Journal, 4, 45 (1965).

74. Cadman, T.W., Rothfus, R.R., and Kermode, R.I., ISA Trans. 6 No. 1,74 (1967).

75. Luyben, W.L., Chem. Eng. Sci. 24, 997 (1969).

76. Kalman, R.E., and Koepecke, R.W., Trans. ASME, 80, 1820 (1958).

77. Kalman, R.E., and Bertram, J.E., AIEE Trans., 77 602 (1959).

78. Coughanowr, D.R., and Koppel, L.B., 'Process Systems Analysis and Control', McGraw-Hill Book Co. (1965).

79. Kalman, R.E., and Koepecke, R.W., Proc. Western Joint Computer Conf., San Francisco, Calif., (March, (1959).

80. Tou, J.T., 'Modern Control Theory', McGraw-Hill Book Co. (1964).

81. Amundson, N.R., Aris, R., Kalman, R.E., and Lapidus, L., 'Applied Mathematics in Chemical Engineering', Special Lecture Series, A.I.Ch.E. Meeting, Washington D.C., (Dec. 1960).

82. Lapidus, L., Shapiro, E., Shapiro, S., and Stillman, R.E., A.I.Ch. E. Journal, 7, 288 (1961).

83. Lapidus, L., and Luus, R., 'Optimal Control of Engineering Processes', Blaisdell Publishing Co. (1967).

84. Heidemann, R.A., D.Sc. Dissertation, 1966, Washington University.

85. Rafal, M.D., Ph.D. Dissertation, 1966, Northwestern University.

86. Brosilow, C.B., and Naphtali, L.M., Chem. Eng. Sci. 22 1125 (1965).

87. Brosilow, C.B., and Handley, K.R., A.I.Ch.E.Journal, 14, 467 (1968).

88. Merluzzi, P., and Brosilow, C.B., A.I.Ch.E. Journal, 18, 739 (1972).

89. Zahradnik, R.L., Archer, D.H., and Rothfus, R.R., Chem. Eng. Progr. Symposium Series, 59, No. 46 132 (1963).

90. McNeill, G.A., and Sacks, J.D., Chem. Eng. Progr. 65 No. 3, 33 (1969).

91. Athan, M., and Falb, P.L., 'Optimal Control', McGraw-Hill Book Co. (1966).

92. Hu, Y.C., and Ramirez, W.F., A.I.Ch.E. Journal, 18, 479 (1972).

93. Pang. K.H., and Johnson, A.I., Can. J. Chem. Eng. 49, 398 (1971).

94. Ewing, G.M., 'Calculus of Variations with Applications', W.W. Norton Co. (1969).

95. Wilde, D.J., 'Optimal Seeking Method', Prentice-Hall Inc., (1964).

96. Rosenbrock, H.H., and Storey, C., 'Computational Techniques for Chemical Engineers', Pergamon Press (1966).

97. Marquardt, D.W., Journal of Soc. Indust. Applied Math., 11, No. 2, 431 (1963).

98. Thomas. Jr., G.B., 'Calculus' Addison-Wesley Publishing Co. Inc., (1962).

99. Hala, G., Wichterle, I., Polak, J., and Boublik, T., 'Vapor-Liquid Equilibrium Data at Normal Pressures', Pergamon Press (1968).

100. Milne, W.E., 'Numerical Solution of Differential Equations', Dover Publication Inc., (1970).

101. Ogata, K., 'State Space Analysis of Control Systems', Prentice-Hall Inc., (1967).

102. I.B.M. Application Program 'System 360 Scientific Subroutine Package' Verson III, I.B.M. Corporation.

103. Lipkin, M.R., Davison, J.A., Harvey, W.T., Kurtz Jr., S.S., Ind. and Eng. Chem., 16 No. 1, 55 (1944).

104. Thusoo, A.K., M.Sc. Thesis, 1969, University of Ottawa.

105. Birta, L.G. and Trushel, P.J., National Research Council Report, MK 25 (1969).

106. Birta, L.G., Personal communication.

107. Perry, J.H., Chemical Engineers' Handbook, McGraw-Hill Book Co. (1963) 4th edition.

108. Doan, M.H., B.A.Sc. Thesis, 1969, University of Ottawa.

109. Shunta, J.P., Luyben, W.L., A.I.Ch.E., 17 No.1, 92 (1971).

110. Shunta, J.P., Ph.D. Thesis, 1971, Lehigh University.

111. Clear, P.F., Sc.D. Thesis, 1970, Stevens Institute of Technology.

112. Tramontin, F.J., Ph.D. Thesis, 1971, University of Maryland.

113. Toijala, K. et al., Kem. Teollisuus, 29 No. 2, 95 (1972).

114. Toijala, K. et al., Kem. Teollisuus, 29 No. 1, 5 (1972).

115. Whatley Jr., M.J., Ph.D. Thesis, 1971, University of Florida.

116. Mayur, D.N., Ph.D. Thesis, 1971, Rice University.

117. Eschenbacher, R.C., Ph.D. Thesis, 1970, University of Florida.

118. McCabe, W.L., Smith, C.S., 'Unit Operations of ChemicalEngineering', McGraw Hill Book Co. (1967) 2nd edition.

APPENDICES

APPENDIX A

TABLES

TABLE 1

TRANSIENT COMPOSITION RESPONSE OF TRAY NO. 6
UNDER STEP DECREASE IN FEED COMPOSITION.
(.0715 ⟶ .06812 mole fraction Methanol)

TIME (Minutes)	Composition Deviation from 1st steady state (mole fraction)
5	- .00417
10	- .01486
15	- .02056
20	- .02461
25	- .04059
30	- .04217
35	- .04846
45	- .04768
55	- .06242
70	- .06778
85	- .07310
100	- .07914
135	- .08661
180	- .11076

These experimental data are plotted on figure 6.

TABLE 2

STEADY STATE TEMPERATURE PROFILE OF THE COLUMN

Feed rate	9.728 lb-mole/hr
Feed composition	.0715 mole fraction methanol
Reflux rate	2.484 lb-mole/hr
Steam rate	51985.86 Btu/hr

Tray number	Temperature, °F (experimental values)
1	207.5
2	206.5
3	196.0
4	192.2
5	187.0
6	172.2
7	162.0
8	156.0
9	151.0
10	150.0

These experimental data are plotted on figure 7.

TABLE 3

STEADY STATE TEMPERATURE PROFILE OF THE COLUMN

Feed rate 9.728 lb-mole/hr

Feed composition .06812 mole fraction methanol

Reflux 2.484 lb-mole/hr.

Steam rate 51985.86 Btu/hr

Tray number	Temperature °F (experimental values)
1	209.0
2	207.0
3	202.3
4	198.7
5	195.6
6	187.5
7	174.0
8	164.0
9	155.0
10	151.0

These experimental data are plotted on figure 8.

TABLE 4

TRANSIENT TEMPERATURE RESPONSE OF THE DISTILLATION COLUMN UNDER STEP INCREASE IN REFLUX RATE

(2.484 ⟶ 2.58 lb-mole/hr)

First steady-state condition

Feed composition	.6812 mole fraction methanol
Feed rate	9.728 lb-mole/hr
Heat input to reboiler	51986.0 Btu/hr
Reflux rate	2.484 lb-mole/hr

The experimental data are plotted on figure 9.

TABLE 4

Tray Number	Tray temperature deviation from 1st steady state °F												
1	-.10	-.22	-.22	-.25	-.75	-1.0	-1.8	-2.3					
2	-.30	-.70	-.75	-1.2	-1.75	-2.4	-2.8	-3.0					
3	-.15	-.20	-.35	-.50	-1.5	-2.6	-3.3	-4.75					
4	-.10	-.10	-.15	-.25	-.80	-1.7	-1.9	-2.5	-4.75				
5	-.40	-.40	-1.40	-1.60	-1.66	-2.2	-2.5	-5.4	-7.40	-11.3	-12.4	-14.8	-16.0
6	-2.6	-5.0	-5.6	-7.1	-11.3	-14.8	-16.0	-19.0	-23.0	-23.5			
7	-1.0	-2.6	-4.2	-5.0	-5.5	-6.3	-8.4	-10.0	-17.0	-18.0			
8	-1.0	-2.25	-2.5	-4.0	-4.6	-4.7	-6.0	-8.0					
9	-.75	-1.75	-1.80	-2.2	-2.8	-3.0	-4.2	-5.3					
10	-.50	-.75	-1.0	-1.25	-1.25	-1.75	-2.0	-2.60	-2.90				

TABLE 5

TRANSIENT TEMPERATURE RESPONSE OF THE COLUMN
UNDER STEP DECREASE IN REFLUX
(2.484 ⟶ 2.14 lb-mole/hr)

First steady state condition

Feed composition	.065 mole fraction methanol
Feed rate	9.728 lb-mole/hr
Heat input to reboiler	51986.0 Btu/hr
Reflux rate	2.484 lb-mole/hr

The experimental data are plotted on figure 10.

TABLE 5

Tray temperature deviations from 1st steady state °F

Tray Number								
1	.30	.25	.45	.45	.48	.49	.50	.50
2	.60	.50	.60	.60	.70	.75	.75	.75
3	.50	.60	.75	.80	1.0	1.0	1.1	1.1	1.1
4	.25	.50	.80	.75	.80	.80	.80	1.0	1.25
5	.75	1.7	2.1	2.5	2.9	2.9	2.9	2.9	2.9
6	3.0	6.0	7.0	8.0	8.0	8.2	8.3	8.2	
7	4.0	10.0	15.0	17.0	20.0	20.5	21.0		
8	2.5	7.0	13.5	17.0	20.5	24.0	25.5	26.0	25.5
9	2.5	4.5	7.0	9.0	12.0	14.0	16.0	16.0	
10	.75	3.0	5.3	13.0	14.6	14.5	16.5	16.8	

TABLE 6

TRANSIENT TEMPERATURE RESPONSE OF THE COLUMN

UNDER STEP DECREASE IN FEED RATE

(11.5 ⟶ 9.728 lb-mole/hr)

First steady state condition

Feed composition	.065 mole fraction methanol
Feed rate	11.5 lb-mole/hr
Heat input to reboiler	51986.0 Btu/hr
Reflux rate	2.484 lb-mole/hr

The experimental data are plotted on figure 11.

TABLE 6

Tray Number	Tray temperature deviation from 1st steady state °F										
1	1.2	2.0	2.5	2.6	2.5	2.5	2.5	2.5			
2	2.0	2.5	2.9	3.4	3.6	3.75	3.9				
3	1.20	2.75	3.3	3.5	4.0	4.2	4.25				
4	1.25	1.75	2.3	2.9	3.0	3.1	3.2	3.2	3.1		
5	.75	2.6	5.0	6.9	8.0	9.0	9.2	9.2	9.2		
6	.10	3.4	7.5	8.2	12.0	13.2	16.0	18.0	19.0		
7	.50	1.60	2.8	4.0	6.5	9.0	11.5	14.0	15.0	17.0	17.0
8	.10	1.4	3.0	4.1	7.8	11.4	14.0	17.0	19.0	19.0	
9	.10	.60	1.0	1.4	2.8	3.4	4.4	5.5	5.5		
10	.20	.30	.50	.70	.90	1.5	2.10	2.5	2.5		

TABLE 7

TRANSIENT TEMPERATURE RESPONSE OF THE COLUMN
UNDER STEP INCREASE IN FEED RATE
(9.728 ⟶ 11.5 lb-mole/hr)

First steady state condition

Feed composition	.065 mole fraction methanol
Feed rate	9.728 lb-mole/hr
Heat input to reboiler	51986.0 Btu/hr
Reflux rate	2.484 lb-mole/hr

The experimental data are plotted on figure 12.

TABLE 7

Tray Number	Tray temperature deviation from 1st steady state °F								
1	- .75	-1.3	-1.8	- 2.0	- 2.5	- 2.5	- 2.5	- 2.5	
2	-1.25	-2.45	-2.75	- 3.2	- 3.3	- 4.0	- 4.5	- 4.5	- 4.5
3	- .3	-2.0	-2.9	- 3.5	- 4.0	- 4.2	- 4.2		
4	-1.4	-2.2	-2.5	- 2.75	- 2.9	- 3.2	- 3.5	- 3.8	- 3.8
5	-2.0	-3.6	-4.6	- 5.3	- 7.5	- 9.5	-10.2	-10.2	
6	-2.5	-5.6	-7.8	-10.0	-11.6	-14.2	-18.0	-18.2	
7	-2.2	-5.0	-8.0	-10.0	-11.5	-12.8	-14.8	-16.0	-16.0
8	-1.0	-3.0	-4.6	- 5.8	- 7.0	- 8.0	- 8.5	- 9.0	- 9.5
9	- .70	-1.3	-2.1	- 2.8	- 3.5	- 4.0	- 4.4	- 4.6	- 5.4 - 5.4
10	- .60	-1.0	-1.5	- 2.0	- 2.70	- 3.25			

TABLE 8

TRANSIENT TEMPERATURE RESPONSE OF THE COLUMN
UNDER STEP DECREASE IN STEAM RATE
(51986.0 ⟶ 51009.0 Btu/hr)

First steady state condition

Feed composition	.06812 mole fraction methanol
Feed rate	9.728 lb-mole/hr
Heat input to reboiler	51986.0 Btu/hr
Reflux rate	2.484 lb-mole/hr

The experimental data are plotted on figure 13.

TABLE 8

Tray temperature deviations from 1st steady state °F

Tray Number											
1	- .40	- .75	-1.2	-1.4	-1.4	-1.5	- 1.5				
2	- .20	- .20	- .60	- .90	-1.0	-1.5	- 1.85	- 2.0	- 2.0		
3	- .10	- .30	- .75	- .90	-1.0	-1.30	- 1.50	- 2.0	- 2.0	- 2.6	- 2.6
4	- .20	- .50	- .85	-1.0	-1.40	-1.60	- 1.60	- 1.90	- 2.30	- 2.60	- 2.60
5	- .40	-2.0	-3.4	-4.0	-4.6	-5.8	- 6.5	- 7.5	- 9.0	-10.0	-10.0
6	- .50	-3.0	-5.0	-6.6	-8.4	-9.5	-10.2	-11.6	-13.2	-14.5	-14.5
7	- .40	-2.0	-3.6	-5.7	-6.3	-7.0	- 8.0	- 9.25	-10.0	-10.5	
8	- .50	-1.5	-2.5	-3.2	-3.8	-4.30	- 5.0	- 5.8	- 6.5	- 6.5	
9	- .40	- .90	-1.5	-2.0	-2.25	-2.5	- 3.5	- 3.9	- 4.2	- 4.0	
10	- .10	- .50	- .75	- .80	- .80	-1.0	- 1.5	- 1.90	- 2.0	- 2.0	

TABLE 9

TRANSIENT TEMPERATURE RESPONSE OF THE COLUMN
UNDER STEP DECREASE IN FEED COMPOSITION
(.0713 ⟶ .06812 mole fraction)

First steady state condition

Feed composition	.0713 mole fraction methanol
Feed rate	9.728 lb-mole/hr
Heat input to reboiler	51986.0 Btu/hr
Reflux rate	2.484 lb-mole/hr

The experimental data are plotted on figure 14.

TABLE 9

Tray temperature deviations from 1st steady state °F

Tray Number											
1	.10	.45	.70	.75	.75	.75	.75	.75	.80		
2	0.0	0.0	.20	.40	.75	.75	.75	.75	.85		
3	.40	.60	.60	.60	1.0	1.30	1.30	1.5			
4	.50	1.0	1.0	1.1	1.4	1.5	1.7	2.0			
5	.60	1.0	1.4	1.5	2.25	2.5	3.0	3.5	4.5		
6	- .20	.20	1.0	2.0	2.5	4.5	5.0				
7	- .20	- .40	0.0	0.0	.20	.75	1.20	1.50	2.25	2.5	3.0
8	- .20	- .25	- .25	- .15	0.0	0.1	.20	.25	.25	.80	1.3
9	0.0	0.0	0.0	0.0	0.0	0.0	.20	.30	.50		
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	.25	.25	

TABLE 10

EFFECT OF S ON THE TERMINAL PRODUCT DEVIATIONS

Disturbance: Feed Composition Decrease

.0715 ⟶ .065 mole fraction of methanol

t_f = 70 minutes

Q = I

R = .05II

	S = I	S = 100 II
Top Product deviation (mole fraction)	$.83 \times 10^{-4}$	$.11 \times 10^{-6}$
Bottom Product deviation (mole fraction)	$.17 \times 10^{-7}$	$.17 \times 10^{-8}$

TABLE 11

COMPARISON OF THE OPTIMAL CONTROL TRAJECTORIES CALCULATED FROM TWO DIFFERENT SETS OF LIQUID TRAY HOLDUPS

TIME (MINUTES)	REFLUX(lb-mole/min) A	B	Bottom rate(lb-mole/min) C	D
0	.0432104	.0432104	.149166	.149166
10	.0432185	.0432098	.149193	.149158
20	.0432114	.0432088	.149160	.149150
30	.0432012	.0432075	.149142	.149144
40	.0432052	.0432061	.149134	.149138
50	.0432043	.0432050	.149131	.149134
60	.0432040	.0432043	.149130	.149131
70	.0432040	.0432040	.149130	.149130

Column A and C are the optimal control trajectories calculated from actual liquid tray holdups. Column B and D are the optimal control trajectories calculated from liquid tray holdups equal to one pound mole.

APPENDIX B

AN OPTIMAL CONTROL ALGORITHM BASED ON AN INITIAL CO-STATE SEARCH

A broad class of optimal control problems can, via the necessary conditions provided by the Pontrayagin minimum principle, be reformulated as two point boundary value problems. Various ways can be employed to obtain the solution of the two boundary value problems. The Riccati transformation is one among these methods. Birta and Trushel (105) followed another approach which was referred to as the "terminal error function" (TEF) approach. In this approach, the two point boundary value problem is reformulated into a multi-dimensional minimization problem. Then, a suitable "error" function is defined. The minimum of this error function should coincide with the solution of the boundary value problem.

Consider the dynamic system which can be described by the following nn--first order differential equations.

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t)) \qquad \text{(B-1)}$$

The nn- vector $\underline{x}(t)$ and the r-vector $\underline{u}(t)$ represent the state and the control vector respectively; the independent variable t denotes time.

It is assumed that the initial state vector are given $\underline{x}(t_o) = \underline{x}_o$ and m_1, $(0 < m_1 < nn)$ component of the state vector

are prescribed at some terminal time t_1 of the process. The boundary conditions of equation (B-1) will be:

$$\underline{x(t_o)} = \underline{x_o} \qquad \text{(B-2)}$$

$$\underline{x(t_1)} = \underline{\sigma_j} \qquad j = 1, 2 \ldots m_1 \qquad \text{(B-3)}$$

The optimal control problem is to find a set of control which subjected to the differential constraint of (B-1) and the boundary conditions (B-2) and (B-3) will minimize the following performance function:

$$J_2 = \underline{MM\ (x(t_1))} + \int_{t_o}^{t_1} \underline{LL\ (x(t), u(t))}\, dt \qquad \text{(B-4)}$$

The solution of this problem is then termed as the optimal control and is assumed to be existent.

The necessary conditions which the optimal control has to satisfy are provided by the Pontrayagin minimum principle. In order to establish these necessary conditions, it is essential to introduce a nn-vector $\underline{Z(t)}$ (co-state variable) and the Hamiltonian function which is defined as

$$\underline{H\ (x(t), Z(t), u(t))} = \underline{Z(t)}\ \underline{f\ (x(t), u(t)} + \underline{LL\ (x(t), u(t))} \qquad \text{(B-5)}$$

Let x* be the solution of (B-1) which corresponds to $\underline{x^*(t_o)} = \underline{x_o}$ and $\underline{u} = \underline{u^*}$. For u* to be the optimal control, it is necessary that there exists a vector function $\underline{Z^*(t)}$, such that

$$\dot{x}^*(t) = H_Z\,(x^*(t),\ Z^*(t),\ u^*(t)\ ) \qquad \text{(B-6)}$$

$$\dot{Z}^*(t) = -H_x\,(x^*(t),\ Z^*(t),\ u^*(t)\ ) \qquad \text{(B-7)}$$

$$x^*_j(t_1) = \sigma_j,\ j = 1,\ 2,\ 3,\ \text{---}m_1 \qquad \text{(B-8)}$$

$$Z^*_j(t_1) = MM_{xj}(x^*(t_1)\ ),\ j = m_1 + 1,\ m_1 + 2\ \text{----}nn \qquad \text{(B-9)}$$

and

$$H\,(x^*(t),\ Z^*(t),\ u^*(t)\ ) \leq H\,(x^*(t),\ Z^*(t),\ u\ (t)\ ) \qquad \text{(B-10)}$$

for each $t \in [t_o,\ t_1]$.

If the terminal time t_1 is prescribed and the equation $H_u(x(t),\ Z(t)\ ) = 0$ can be explicitly solved for $u(t)$; i.e. $u(t) = U'(x(t);\ Z(t)\ )$, formulation of the TEF method results in the following $(r + 2nn + 1)$ equations.

$$H_u\,(x^*(t),\ Z^*(t),\ u^*(t)\ ) = 0 \qquad \text{(B-11)}$$

$$u(t) = U'(x(t),\ Z(t)\ ) \qquad \text{(B-12)}$$

$$\dot{x}(t) = H_Z\,(x(t),\ Z(t),\ u(t)\ ) \qquad \text{(B-13)}$$

$$x(t_o) = x_o$$

$$\dot{Z}(t)) = -\,H_X(x(t),\ Z(t),\ u(t)\ ) \qquad \text{(B-14)}$$

$$Z(t_o) = w$$

$$E = \sum_{j=m_1+1}^{nn} (x_j(t_1) - \sigma_j)^2 + \sum_{j=m_1+1}^{n} (Z_j(t_1) - MM_{Xj}\ (x(t_1)\ )^2 \qquad \text{(B-15)}$$

When $\underline{u}$ is given by equation (B-12), and $\underline{\Omega}$ represents the set of n-vector $\underline{w}$, for which the differential systems (B-13) and (B-14) have unique solutions in the interval $[t_o, t_1]$. Then, for every $w \in \underline{\Omega}$, there corresponds a unique non-negative value for the scalar function E. This function, which is the terminal error function, can be regarded as a function of the initial co-state $\underline{w} = \underline{Z}(t_o)$.

If a $w^* \in \underline{\Omega}$ can be found, of which $E(w^*) = 0$, the resulting solutions of equations (B-13) and (B-14), subjected to (B-12) will satisfy the necessary conditions of equations (B-6), (B-7), (B-8), (B-9) and (B-11). Then, the control $\underline{u}^*$ specified by equation (B-12) will be at least a local minimum for the original dynamic optimization problem. Furthermore, $\underline{w}^*$ also give E its minimum value. Consequently, the optimal control problem has now been transformed into the problem of minimizing the terminal error function by looking for the $\underline{w}^*$. Because $\underline{w}^*$is the initial condition of the co-state vector, this minimization problem is also termed as the "initial co-state search".

The above has just outlined the essential concepts of the TEF method for obtaining the solution of the optimal control problem. Notice that in this approach, two of the three necessary conditions (equations (B-6), (B-7) and (B-11)) are forced to hold and the iterative scheme is geared toward satisfying the third necessary condition (equations (B-8) and (B-9)). This method of forcing some

of the necessary conditions and then iterating to satisfy the remaining is common to a lot of optimal control algorithms suggested in recent years.

Birta and Trushel (105) presented solutions of the optimal control problems of different complexity with the TEF approach. Their examples included linear as well as non-linear systems (such as the CSTR), problems with bounded and unbounded control. Birta and Trushel (105) wrote a computer program which handled optimization problems with this initial co-state search technique. Birta (106) claimed that their program has been highly successful with systemsup to the fourth order.

A copy of this program has been given to the author by Birta and has been tried on the optimal distillation control calculation.

The transient responses of the column were obtained through simulation of the non-linear dynamic model. Second order transfer function were used to fit all these response curves. The second order systems were then used to represent the dynamics of the distillation column in the optimal control calculation with Birta's program. Only one control variable (Reflux or reboiler steam rate) was considered in this calculation. Both the bound and unbound control have been tried in this calculation. It was found, in some cases, the convergence of the calculation depended highly on the initial guess of the co-state variables.

For a guess which was too far from the desired values, the calculation would take a long computation time to converge to the right answer.

The linear model used in the two point optimal control calculation has also been used to represent the column in here. Again, only one control variable was considered. Since the linear dynamic model has twelve state variables (12^{th} order system), the calculation would be more complex and difficult. It was observed that in this case, the convergence was again highly dependant on the choice of the initial co-state variables. The calculation would collapse when the initial guess was too far from the desired values.

The method of Birta and Truchel is good for non-linear system with low order. It guaranteed at least a local minimum. Also, it did not require large amount of computation time and computer memory (about 120K on the IBM 360/65 model). On the other hand, for high order linear systems, this method may collapse due to the large number of differential equations that have to be integrated and the high number of unknowns to be searched through the search process. In this case the method is very time consuming.

The Riccati transformation used in the two point control calculation is applicable only to linear systems. No search procedure is involved and the resulting optimal control is always the global optimal (91). So, for a

linear system with high order, the Riccati transformation method has an edge over the TEF method of Birta and Trushel. The result of this work has supported this point.

APPENDIX C

STUDY OF COLUMN DYNAMICS BY PULSE TESTING

Thusoo (104) studied the dynamic behavior of the same distillation column in the frequency domain. The dynamic of the column was obtained experimentally by pulse testing technique.

In this work, the same pulse testing technique was used on the mathematical models rather than on the actual column itself.

The linear dynamic model which was based on component material was used to represent the column. Disturbance in the form of pulse was introduced into the column in either the feed rate, feed composition, the reflux or the vapor rate from the reboiler. Tray composition response was compared with the temperature response of Thusoo's work in the frequency domain. The response curve from the simulation was found to agree with the experimental results qualitatively, but not quantitatively. This result confirmed the fact that the linear dynamic model could not actually represent the transient behavior of the column.

Pulse testing was then performed on the non-linear model. Same type and size of pulses were introduced into the column. The transient responses of the tray temperatures were obtained. They were transformed into the frequency domain by the same method used in the previous cases. It

was found that both the results from simulation and Thusoo's experimental work agreed with each other. From the results of the pulse testing, it was proved, once more, that a non-linear mathematical model was needed to truly represent the dynamics of this binary distillation column.

Detailed description of the techniques used in transforming the dynamic information from time domain to frequency domain as well as the experimental pulse testing results can be obtained from Thusoo's thesis (104).

APPENDIX D

TABLE 12

1. CALIBRATION OF THE STEAM PRESSURE GAUGE

Reading from Dead weight Tester (psig)	Reading from Gauge
20.0	35.0
30.0	45.0
40.0	55.0
50.0	65.0
60.0	75.0
70.0	88.0
80.0	99.0
90.0	109.0
100.0	120.0
110.0	130.0

These experimental data are plotted on figure 28.

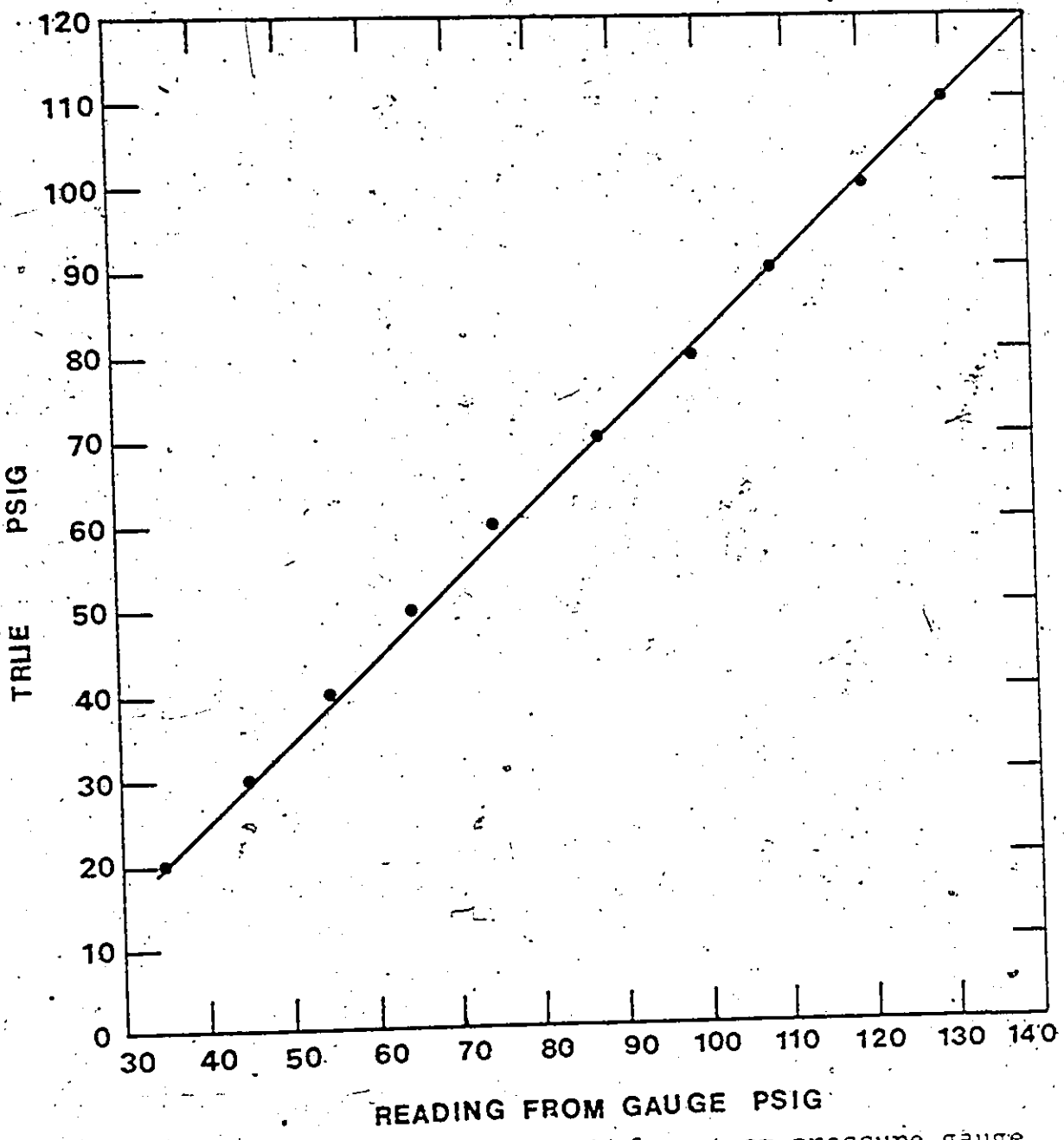


Figure 28. Calibration curve for steam pressure gauge.

TABLE 13

2. CALIBRATION OF THE ORIFICE METER

Steam pressure 120 psig

Reading of Chart Recorder	Flow rate of Steam (ft^3/sec)
15.0	.0587
31.0	.0528
39.0	.0595
41.0	.0605
49.0	.0655
53.0	.0673
63.0	.0749
69.0	.0772
77.0	.0825
81.0	.0824
85.0	.0864
95.0	.0890
97.0	.0913

These experimental data are plotted on figure 29.

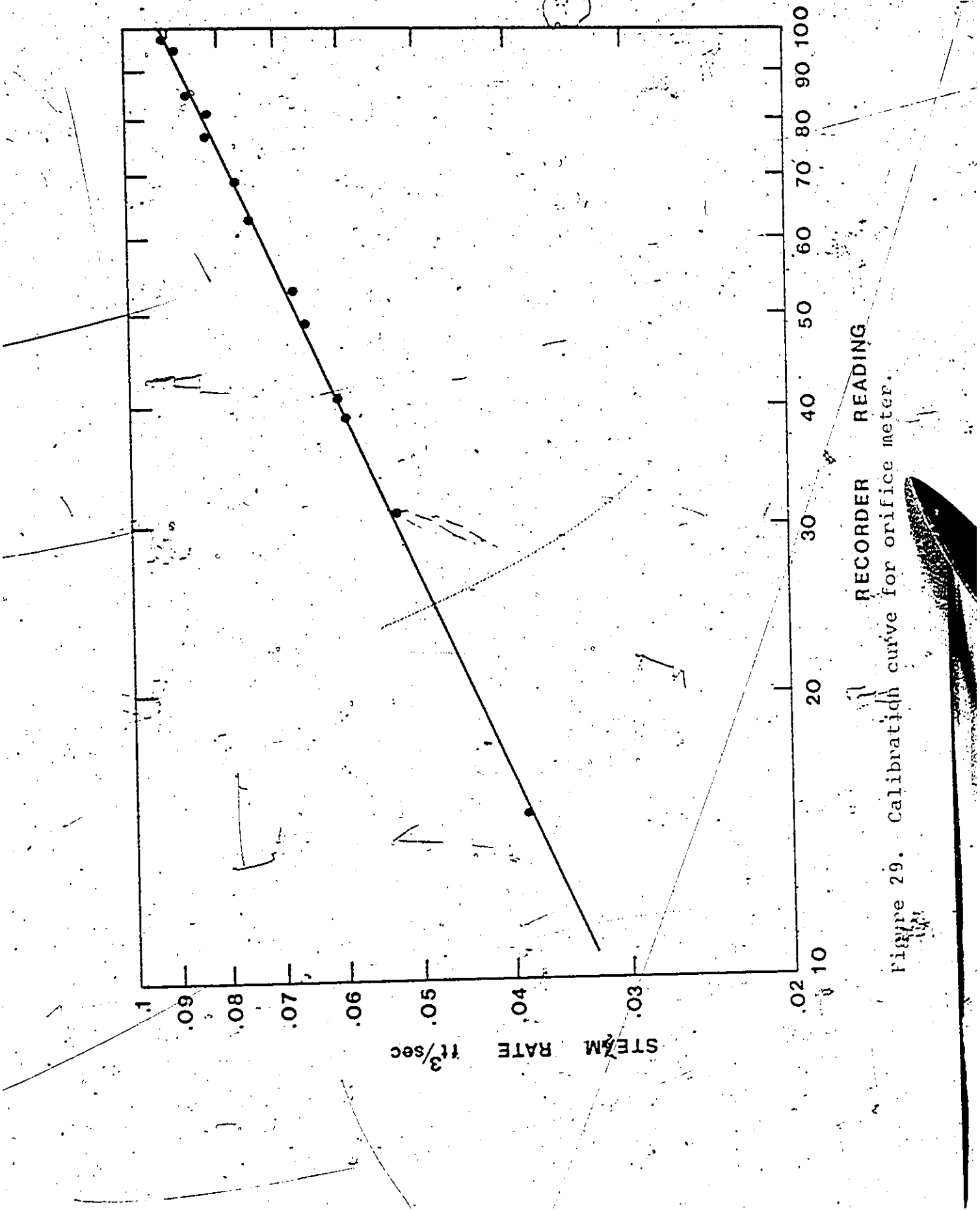


Figure 29. Calibration curve for orifice meter.

TABEE 14

3. CALIBRATION OF PYCNOMETERS

Dry weight of Pycnometer No. 1 18.15127 gm

Calibration Temperature 20°C

Total reading of the two arms	Actual Volume in ml
.810	3.0273
3.850	3.0502
4.050	3.0516
8.650	3.0881
8.730	3.0883
13.10	3.1219
13.580	3.1257
14.410	3.1322

These experimental data were plotted on figure 30.

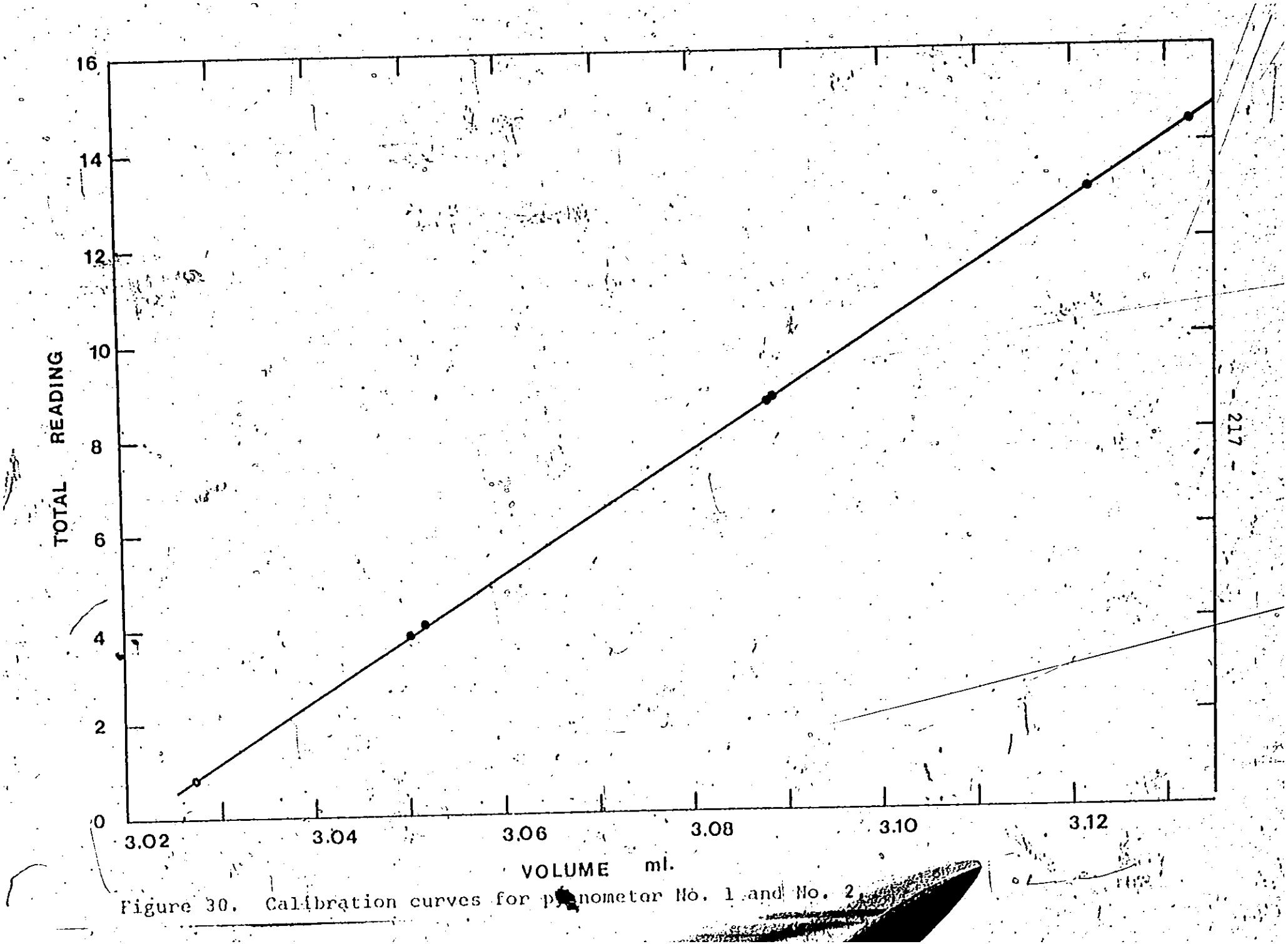


Figure 30. Calibration curves for pycnometer No. 1 and No. 2.

TABLE 15

Dry weight of pycnometer No. 2 19.21915 gm

Calibration Temperature 20° C.

Total reading of the two arms	Actual volume in ml
7.00	3.1228
8.170	3.1310
9.730	3.1436
12.620	3.1662
12.980	3.1683
14.980	3.1662

These experimental data were plotted on figure 31.

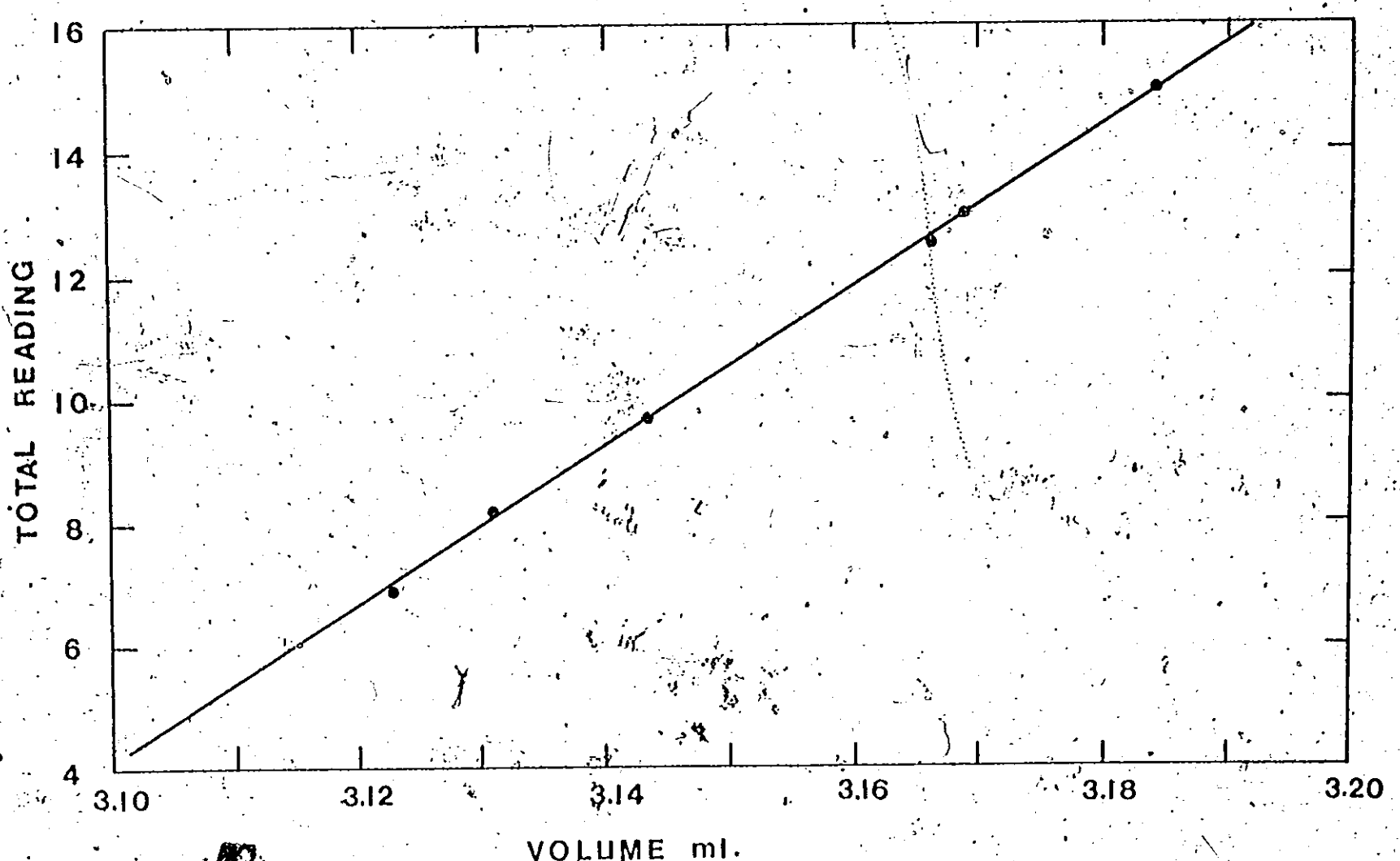


Figure 31. Calibration curves for pycnometer No. 1 and No. 2

APPENDIX E

VAPOR LIQUID EQUILIBRIUM DATA FOR METHANOL-WATER SYSTEM AT ATMOSPHERIC PRESSURE (99)

Antoine's equation

$$\text{Log } p = A' - B'/(T + C') \qquad \text{(D-1)}$$

p = mmHg

T = °C

Antoine vapor pressure constants.

	A'	B'	C'
(1)	8.07246	1574.99	238.86
(2)	7.96681	1668.21	228.0

Ideal equilibrium constant.

$$K_{Ideal} = p/P \qquad \text{(D-2)}$$

P = 760 mm Hg

Van Laar third order equation.

$$\log \gamma_1 = \frac{A_{12} X_2^2}{X_1 (A_{12}/A_{21}) + X_2^2} \qquad \text{(D-3)}$$

$$\log \gamma_2 = \frac{A_{21} X_1^2}{X_2 (A_{21}/A_{12}) + X_1^2} \qquad \text{(D-4)}$$

Van Laar equation constant

A_{12} = .3625 $\qquad$ A_{21} = .2418

TABLE 16

TEMPERATURE AND VAPOR-LIQUID COMPOSITION DATA FOR METHANOL-WATER SYSTEM AT 760 mm Hg (99)

Mole fraction of MeOH X	Mole fraction of MeOH Y	T°C
.0293	.1831	95.20
.0346	.2107	94.50
.0406	.2363	93.70
.0422	.2652	92.80
.0557	.2978	91.80
.0644	.3265	90.90
.0737	.3608	90.00
.0838	.3861	89.10
.0948	.4142	89.20
.2801	.6621	78.80
.3004	.6882	77.60
.3212	.6882	77.60
.3435	.7002	76.90
.3664	.7178	76.20
.3909	.7274	75.70
.4141	.7428	75.10
.4391	.7597	74.60
.4637	.7668	74.00
.8457	.9360	67.20
.8867	.9682	66.60
.9293	.9771	65.70

TABLE 17

TEMPERATURE AND VAPOR-LIQUID COMPOSITION DATA FOR METHANOL-WATER SYSTEM AT 760 mm Hg (107)

Mole fraction of MeoH X	Mole fraction of MeoH Y	T°C
.020	.134	96.40
.040	.230	93.50
.060	.304	91.20
.080	.365	89.30
.10	.418	87.70
.15	.517	84.40
.20	.579	81.70
.30	.665	78.0
.40	.729	75.30
.50	.779	73.10
.60	.825	71.20
.70	.87	69.30
.80	.915	67.60
.90	.958	66.0
.95	.979	65.0
1.00	1.00	64.50

Experimental data of tables 16 and 17 are plotted on figures 32 and 33.

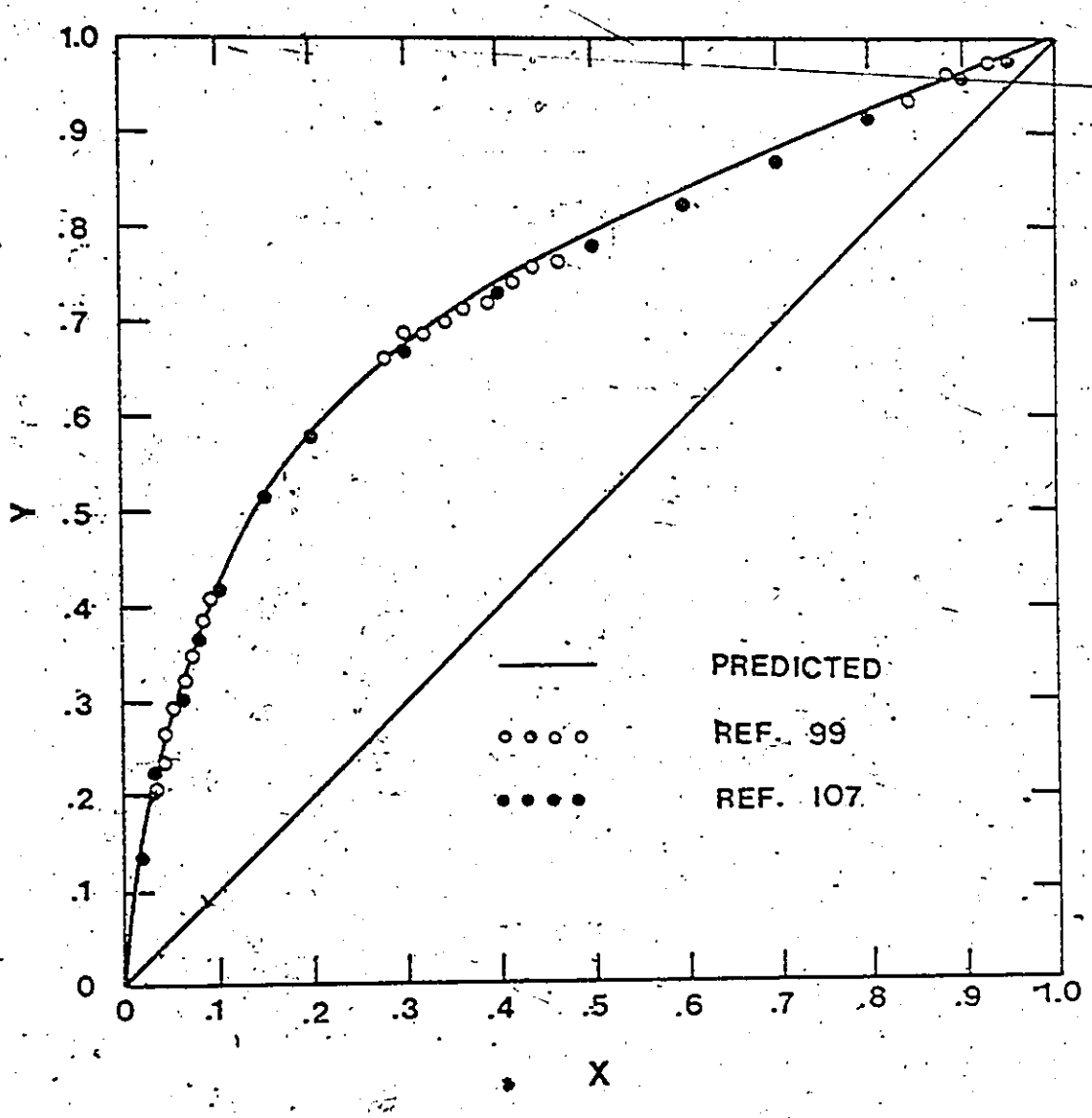


Figure 32. X - Y diagram for methanol-water system.

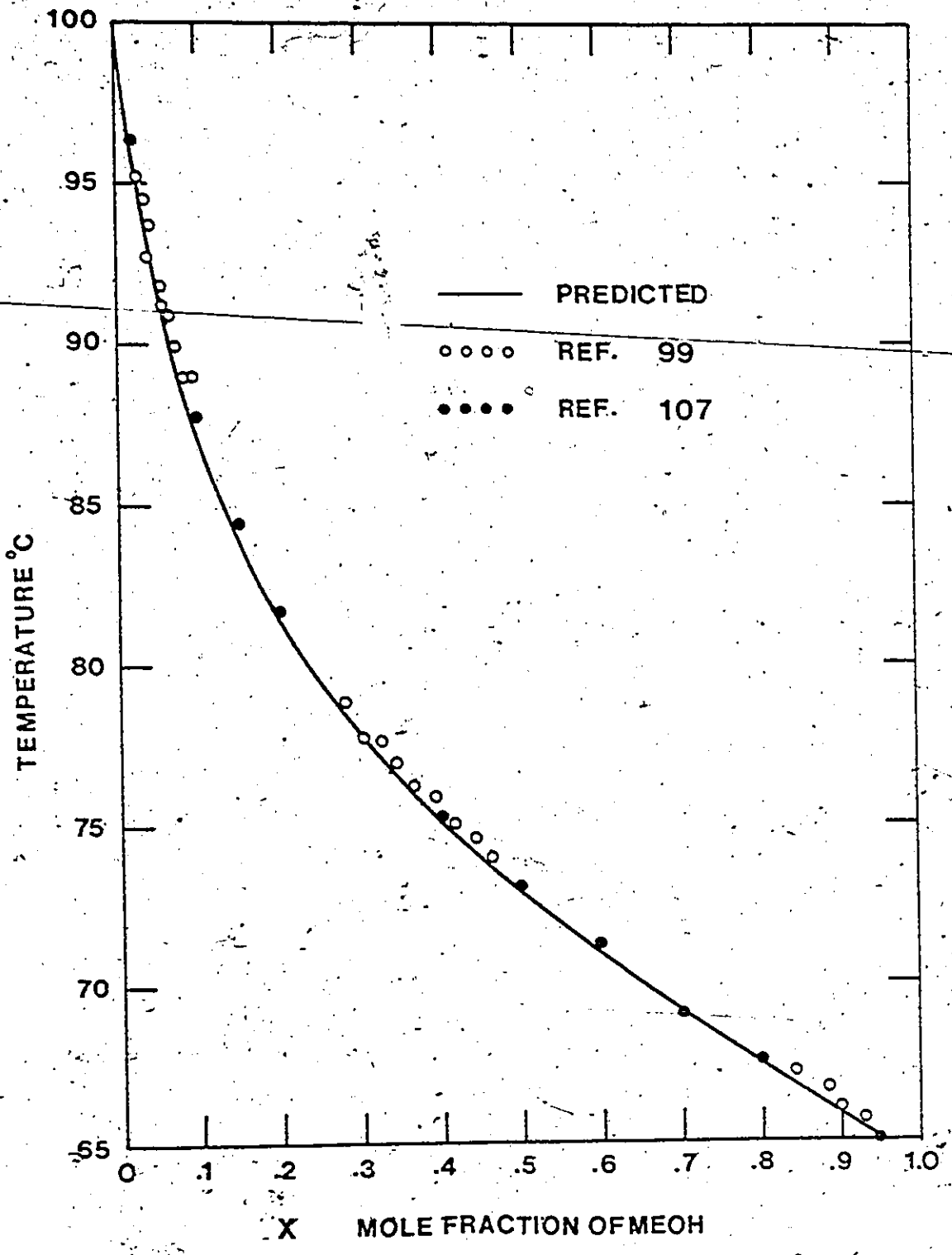


Figure 33. T - X diagram for methanol-water system.

APPENDIX F

TABLE 18

ENTHALPY DATA FOR METHANOL-WATER SYSTEM

Enthalpies of saturated water and steam (118)

Temperature (°F)	Liquid h Btu/lb	Vapor H Btu/lb
140.0	107.89	1122.0
150.0	117.89	1126.10
160.0	127.89	1130.20
170.0	137.90	1134.20
180.0	147.92	1138.10
190.0	157.95	1142.00
200.0	167.99	1145.90
210.0	178.05	1149.70
212.0	180.07	1150.40
220.0	188.13	1153.40

Base temperature = 32 °F

Using least square curve fitting, the following equations result

Saturated water

$$h = -586.62 + 18.054\ T \quad \text{(Btu/lb-mole)} \qquad \text{(E-1)}$$

Saturated steam

$$H = 19212.48 + 7.056\ T \quad \text{(Btu/lb-mole)} \qquad \text{(E-2)}$$

T = temperature in °F

TABLE 19

ENTHALPIES OF SATURATED LIQUID AND VAPOR METHANOL (108)

Temperature °F	Liquid h Btu/lb	Vapor H Btu/lb
140.0	65.34	542.59
149.0	71.07	544.50
158.0	76.86	544.72
167.0	82.68	545.46
176.0	88.56	544.93
185.0	94.47	545.75
194.0	100.04	545.76
203.0	106.44	546.67
212.0	112.50	547.20

Using least square curve fitting

Liquid Methanol

$$h = -850.23 + 20.96\ T \quad \text{(Btu/lb-mole)} \qquad \text{(E-3)}$$

Vapor Methanol

$$H = 17164.41 + 1.62\ T \quad \text{(Btu/lb-mole)} \qquad \text{(E-4)}$$

APPENDIX G

STEADY STATE OPERATING CONDITION OF THE BINARY DISTILLATION COLUMN FOR OPTIMAL CONTROL CALCULATION

Feed composition	.0715 mole fraction methanol
Feed rate	9.728 lb-mole/hr
Reflux rate	2.484 lb-mole/hr
Heat input to reboiler	51,986 Btu/hr
Top product withdrawal rate	.67997 lb-mole/hr
Bottom product withdrawal rate	9.04803 lb-mole/hr

TABLE 20

STEADY STATE INFORMATION OF THE COLUMN

Tray Number	Vapor Rate (lb-mole/hr)	Liquid rate (lb-mole/hr)	Holdup (lb-mole)	Composition (mole fraction)	Temperature °F	Equilibrium (99) Constant
Reboiler	2.9650	9.04803	2.0	.003405	210.761	8.09465
1	2.95208	12.0130	0.147	.008366	208.725	7.66782
2	2.94094	12.0001	0.142	.020233	205.377	6.99436
3	2.93888	11.9890	0.137	.037284	200.897	6.14639
4	2.84540	11.9869	0.130	.058755	196.214	5.33376
5	2.88807	2.16543	0.108	.104808	188.859	4.16286
6	2.98751	2.2081	0.085	.269607	174.555	2.38392
7	3.07212	2.30754	0.075	.544039	162.305	1.47967
8	3.12261	2.39215	0.072	.755929	155.248	1.19086
9	3.14985	2.44264	0.070	.878661	151.557	1.08264
10	3.16397	2.46988	0.065	.944019	149.078	1.03559
Condenser	.677997	2.4840	0.075	.97763	148.729	1.01364

In the optimal control calculation, the flow rates were in lb-mole/min.

The followings are the numerical values of the coefficient matrices of the state equation.

When the control variables are the reflux and the bottom flow rate, the $\underline{B}$ matrix is

$$\underline{B} = \begin{bmatrix} -.000303 & .000403 \\ -.000555 & .000903 \\ -.000980 & .00116 \\ -.0010 & .00146 \\ -.000636 & .00140 \\ -.000698 & .0020 \\ -.00113 & .00344 \\ -.00082 & .00270 \\ .00045 & .00159 \\ .000239 & .000851 \\ .000121 & .000439 \\ 0 & 0 \end{bmatrix}$$

When the control variables are the distillate and vapor rate, the $\underline{B}$ matrix is

$$\underline{B} = \begin{bmatrix} -.000303 & -.000099 \\ -.000555 & -.000018 \\ -.000980 & -.000284 \\ -.00110 & -.000354 \\ -.00263 & -.000768 \\ .000698 & -.00275 \\ .00113 & -.00457 \\ .000827 & -.00353 \\ .000459 & -.00205 \\ .000239 & -.00109 \\ .000121 & -.000559 \\ 0 & 0 \end{bmatrix}$$

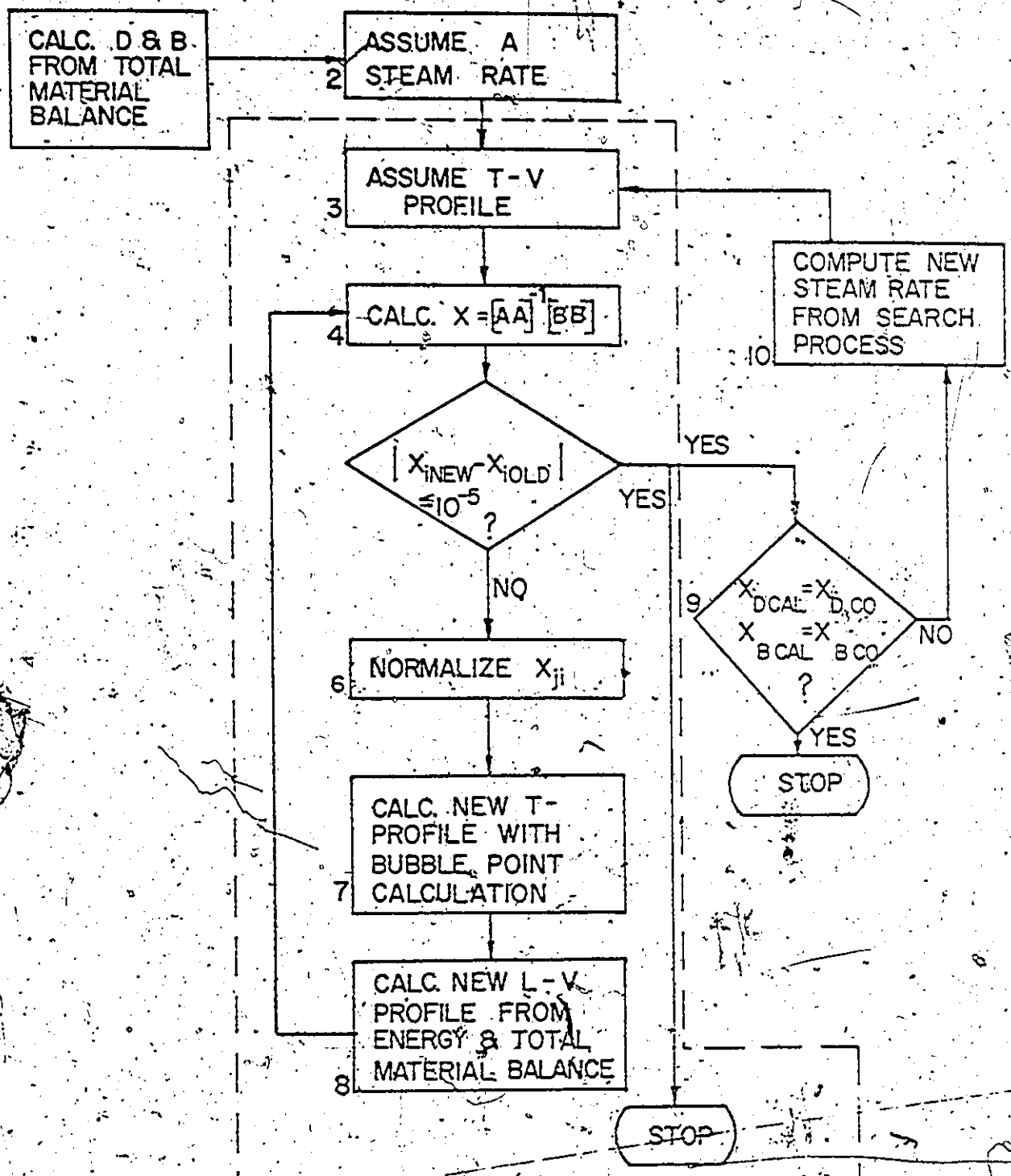
A $_T$ MATRIX

-.5508	.2002	
.40001	-.5774	.2000
.3773	-.5428	.1998
.3428	-.5009	.1998
.3011	-.4527	.0361
.2529	-.2364	.0385
.3003	-.1555	.0385
.1187	-.1142	.0399
.0758	-.1019	.0407
.0620	-.0975	.04116
.0569	-.0957	.0414
.0546	-.0527	

APPENDIX H

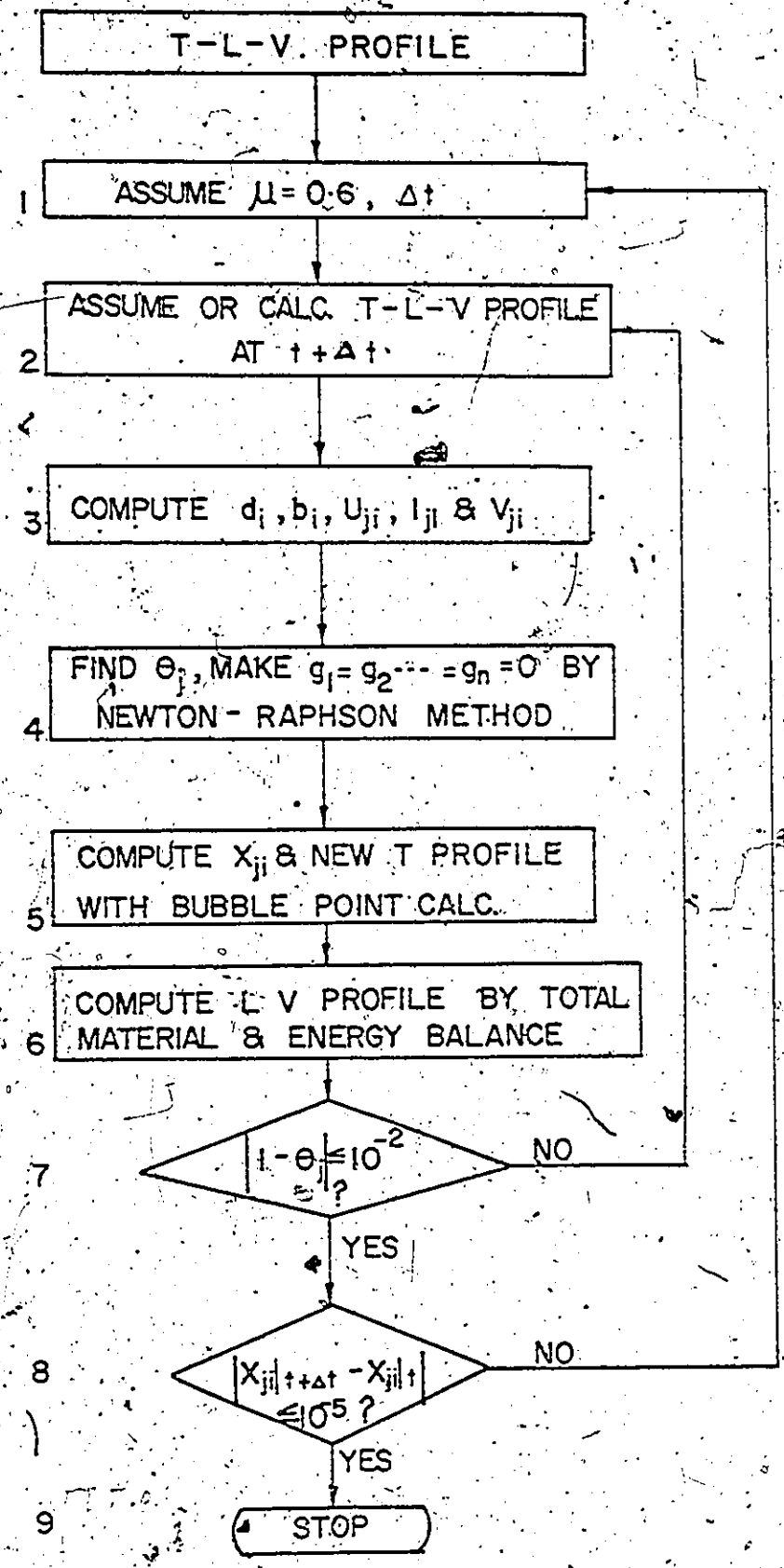
FLOW CHARTS AND COMPUTER PROGRAMS

FLOW CHART FOR STEADY STATE CONTROL CALCULATION

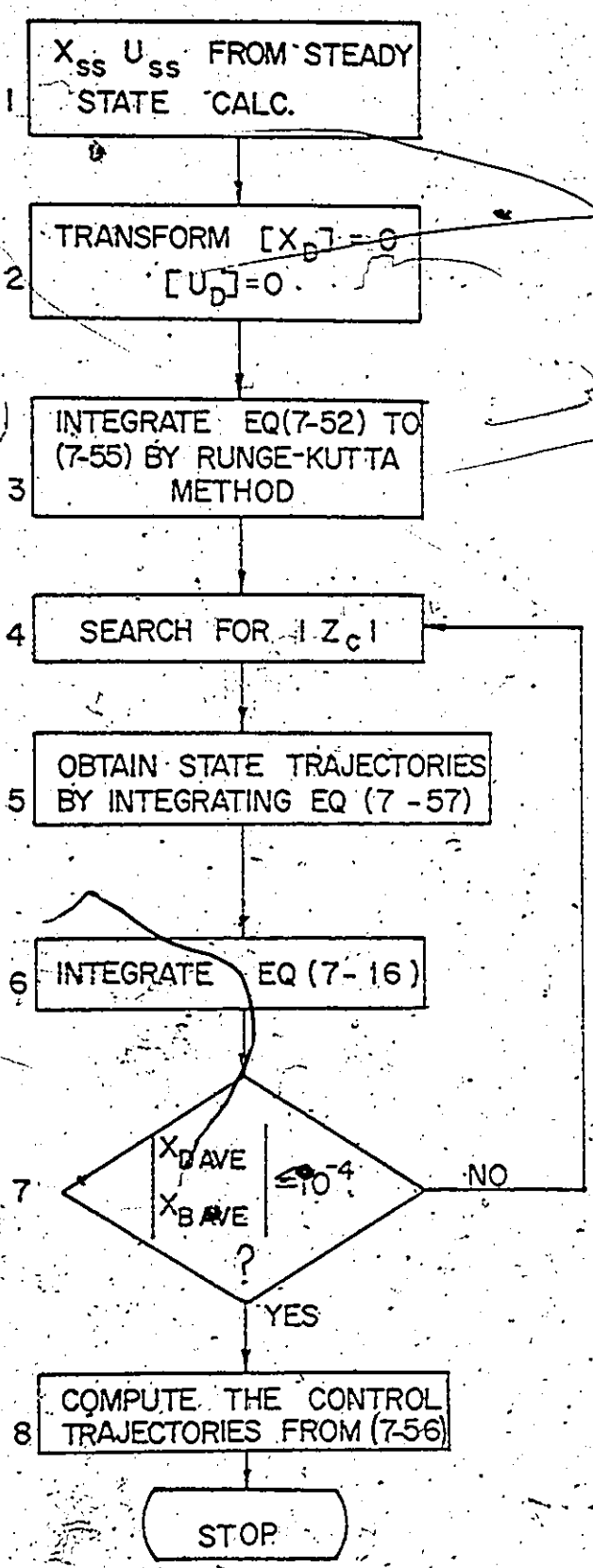


FLOW CHART FOR STEADY STATE CALC.

FLOW CHART FOR DYNAMIC CALCULATION



FLOW CHART FOR OPTIMAL CONTROL CALCULATION (2nd CONTROLLER)



COMPUTER PROGRAM FOR

STEADY STATE CALCULATION

```
FORTRAN IV G LEVEL 18          MAIN          DATE = 70341      21/38/10          PAGE 0001
      C     HEAT AND MATERIAL BALANCE-STEADY-STATE CALCULATION FOR MEOH-WATER
      C     THIS METHOD WAS PROPOSED BY AMUNDSON AND PONTINEN
      C     REFERENCE--PP348-354 CHAPTER 10 "DESIGN OF EQUILIBRIUN STAGE
      C     PROCESSES" BY B.D.SMITH
      C     THIS DISTILLATION PROCESS IS DESCRIBED BY TWO SYSTEMS OF
      C     SIMULTANEOUS ALGEBRAIC EQUATION.ONE IN THE COMPOSITION ONE IN
      C     THE VAPOR FLOW RATE. THESE TWO SET OF EQUATIONS ARE SOLVED BY MATRIX
      C     ALGEBRA.
      C     THE ITERATION VARIABLES THAT HAVE TO BE ASSUMED ARE THE TEMPERATURE
      C     AND VAPOR FLOW PROFILE.
0001        IMPLICIT REAL*8(A-H,O-Z)
0002        DIMENSION AG(11),AM(11,11)
0003        DIMENSION HLW(12),HLM(12),HVW(12),HVM(12)
0004        DIMENSION DLDXT(12),XTRAY(12),HF(12),HG(12),AV(12,12),AX(12,12)
0005        DIMENSION DELX(12),HV(12),HL(12),V(12),YTRAY(12),TEMPB(12)
0006        DIMENSION GAMA(12),GAMB(12),AL(12)
0007        DIMENSION BK(12),AK(12),Y2(12),XT(12)
0008        DIMENSION BX(12,12),X2(12),DF(12),XTAL(12)
0009        L=2
0010        N=12
0011        NN=N-1
0012        KM=N-1
0013        READ(1,401)F,D,B,ERR,RDERR
0014        READ(1,500)(XTRAY(I),I=1,N)
0015        READ(1,500)(TEMPB(I),I=1,N)
0016        CR=48000.00
0017        REFLUX=2.200
0018        F=0.000
0019        XF=.06500
0020        XF=.071500
0021        F=0.72900
0022        REFLUX=2.48400
0023        CR=52000.00
0024        TF=70.00
0025        F=.47755800
0026        B=F-D
0027        DO 48 I=1,NN
0028        DO 48 J=1,NN
0029        AM(I,J)=0.00
0030     48 CONTINUE
0031        DO 49 I=1,NN
0032        AG(I)=0.00
0033     49 CONTINUE

0034        ERR=1.D-7
0035        Y1=0.00
0036        Y2=0.00
0037        ABF=0.00
0038        ABH=0.00
0039        VF=0.00
0040        A1=1.0061400
0041        A2=-.0007180C
0042        A3=.0000034CC
0043        A4=-.0004720C
0044        A5=.58800
0045        B1=1.01 400
0046        B2=-.00588CC
0047        B3=.0000120C
0048        B4=.0004200
0049        DO 52 I=1,5
0050        AL(I)=1.58100
0051     52 CONTINUE
0052        DO 53 I=6,12
0053        AL(I)=2.4H400
0054     53 CONTINUE
0055        DO   I=1,N
0056        DO   J=1,N
0057        AX(I,J)=0.00
0058        BX(I,J)=0.00
0059     1  EV(I,J)=0.00
0060        DO 2 I=1,N
0061        HV(I)=0.00
0062        HL(I)=0.00
0063        BK(I)=0.00
0064        AK(I)=0.00
0065        X2(I)=0.00
0066        DF(I)=0.00
0067        BG(I)=0.00
0068        BF(I)=0.00
0069        DLDXT(I)=0.00
0070        CONTINUE
0071        DO 17 I=1,N
0072        X2(I)=1.00-XTRAY(I)
0073        V(I)=2.9A140C
0074     17 CONTINUE
0075        ITER=1
0076        XG=1.00-XF
```

```
      AL(12)=REFLUX
      CALL VASLA(XTRAY,X2,TEMPB,AK,BK,GAMA,GAMB,N)
   29 CONTINUE
      CALL COEFT(AX,BE,AK,V,B,D,F,XF,N)
      CALL MATINV(AX,N)
      CALL AMATM(AX,BE,XTRAY,N)
      IF(INDEX-10*(INDEX/10)) 41,61,41
   61 WRITE(3,6)(XTRAY(I),I=1,N)
   41 CALL COEFT(BX,CE,BKTV,B,D,F,XG,N)
      CALL MATINV(BX,N)
      CALL AMATM(BX,CE,X2,N)
      IN=1
   14 DELX(IN)=DABS(XTRAY(IN)-OLDXT(IN))
      IF(DELX(IN)-ERR)30,30,31
   30 IF(IN-N)15,16,16
   15 IN=IN+1
      GO TO 14
   31 DO 12 I=1,N
   12 OLDXT(I)=XTRAY(I)
      DO 32 I=1,N
      XT(I)=X2(I)+XTRAY(I)
      XTRAY(I)=XTRAY(I)/XT(I)
      X2(I)=X2(I)/XT(I)
      ABC=XTRAY(I)
      ABD=X2(I)
      ABE=TEMPB(I)
      CALL BUBPTG(ABC,ABD,ABE,Y1,Y3,ABF,ABH,L,BPERR)
      YTRAY(I)=Y1
      Y2(I)=Y3
      AK(I)=ABF
      BK(I)=ABH
      TEMPB(I)=(TEMPB(I)+ABE)/2.D0
   32 CONTINUE
   42 CALL ENCON(TEMPB,HV,HL,XTRAY,X2,YTRAY,Y2, HVW,HVM,HLW,HLM,N)
      CPWD=A1+A2*TEMPB(12)+A3*TEMPB(12)*TEMPB(12)
      CPMD=A4*TEMPB(12)+A5
      CPWB=A1+A2*TEMPB(1)+A3*TEMPB(1)*TEMPB(1)
      CPMB=A4*TEMPB(1)+A5
      CPWF=A1+A2*TF+A3*TF*TF
      CPMF=A4*TF+A5
      CPWD=CPWD*18.D0
      CPWB=CPWB*18.D0
      CPWF=CPWF*18.D0
      CPMD=CPMD*32.D0
      CPMB=CPMB*32.D0
      CPMF=CPMF*32.D0
      HF=(-586.124D0+18.052D0*TF)*XG+(-1064.693D0+23.153D0*TF)*XF
      HF=CPMF*(TF-32.D0)*XF+CPWF*(TF-32.D0)*XG
      HD=CPMD*(TEMPB(12)-32.D0)*XTRAY(12)+CPWD*(TEMPB(12)-32.D0)*X2(12)
      HB=CPMB*(TEMPB(1)-32.D0)*XTRAY(1)+CPWB*(TEMPB(1)-32.D0)*X2(1)
      HF=26757.D0/9.729D0
      HB=HL(1)
      HD=HL(12)
      AAB=AL(12)*(HVW(11)*X2(12)+HVM(11)*XTRAY(12)-HL(12))
      AAC=HVW(11)*X2(12)+HVM(11)*XTRAY(12)-HD
      F=(F*(HF-HB)+QR-AAB)/(HD+AAC-HB)
      D=F-D
      IF(D.LE.0.D0)  GO TO 36
      B=F-D
      GO TO 35
   36 D=.47765BD0-
      B=F-D
   35 CONTINUE
      CC=AL(12)*(HVW(11)*X2(12)+HVM(11)*XTRAY(12)-HL(12))+
     1D*(HVW(11)*X2(12)+HVM(11)*XTRAY(12)-HD) -
      ALN=B*(HVW(4)*X2(1)+HVM(4)*XTRAY(1))+D*HD+CC-F*HF
      ALD=HVW(4)*X2(5)+HVM(4)*XTRAY(5)-HL(5)
      AL(5)=ALN/ALD
      V(4)=AL(5)-B
      DO 50 I=1,3
      ALN=B*(HVW(I)*X2(1)+HVM(I)*XTRAY(1)-HB)+QR
      ALD=HVW(I)*X2(I+1)+HVM(I)*XTRAY(I+1)-HL(I+1)
      AL(I+1)=ALN/ALD
      V(I)=AL(I+1)-B
   50 CONTINUE
      DO 51 I=5,10
      ALN=D*(HD-(HVW(I)*X2(12)+HVM(I)*XTRAY(12)))+CC
      ALD=HVW(I)*X2(I+1)+HVM(I)*XTRAY(I+1)-HL(I+1)
      AL(I+1)=ALN/ALD
      V(I)=AL(I+1)+D
   51 CONTINUE
      AL(1)=B
      V(12)=D
      V(11)=D+AL(12)
      INDEX=INDEX+1
      IF(INDEX-10*(INDEX/10))44,61,44
```

```
   63 WRITE(3,59)C,D,B
      WRITE(3,109)CR
  109 FORMAT(1H0,2X,' CR = ',D17.6)
      WRITE(3,155)(XT(I),I=1,N)
      WRITE(3,5)(TEMPR(K),K=1,N)
      WRITE(3,11)(V(I),I=1,KM)
      WRITE(3,55)(AL(I),I=1,N)
      WRITE(3,156)(X2(I),I=1,N)
      WRITE(3,33)(HV(I),I=1,N)
      WRITE(3,34)(HL(I),I=1,N)
   64 IF(INDEX-100) 29,29,18
   16 CONTINUE
      WRITE(3,139) INDEX
  139 FORMAT(1H0,2X,'INDEX= ',I5)
      WRITE(3,6)(XTRAY(I),I=1,N)
      INDEX=111
      GO TO 63
   18 RETURN
   33 FORMAT(1H0,2X,'HV='/(10F12.6))
   34 FORMAT(1H0,2X,'HL='/(10F12.6))
   55 FORMAT(1H0,2X,'AL='/(10F12.6))
   59 FORMAT(1H0,2X,'C=',F12.6,5X,'D=',F12.6,5X,'B=',F12.6)
  156 FORMAT(1H0,2X, 'X2 =' /(10F12.6))
   40 FORMAT(6F10.5)
    5 FORMAT(1H0,2X,'TEMPR='/(10F12.6))
  155 FORMAT(1H0,2X,'XT(I)= '/(10F12.6))
   11 FORMAT(1H0,2X,'V='/(10F12.6))
    6 FORMAT(1H0,2X,'XTRAY='/(10F12.6))
  500 FORMAT(8F10.5)
      END
```

```
      SUBROUTINE VANLA(X,X2,T,AK1,AK2,GAMMAA,GAMMAB,N)
C     SUBROUTINE VANLA
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PPA(12),PPB(12),PA(12),PB(12),AKA(12),AKB(12),X(12)
      DIMENSION X2(12),GAMAA(12),GAMAB(12),GAMMAA(12),GAMMAB(12),Y(12)
      DIMENSION AK1(12),AK2(12),T(12)
      A12=.378500
      A21=.211500
      DO 10 I=1,N
      T(I)=5.D0*(T(I)-32.D0)/9.D0
      PPA(I)=8.0724600-1574.99D0/(T(I)+238.86)
      PPB(I)=7.9668100-1668.21D0/(T(I)+228.D0)
      PA(I)=DEXP(PPA(I)*2.302590D)
      PB(I)=DEXP(PPB(I)*2.30259D0)
      AKA(I)=PA(I)/760.D0
      AKB(I)=PB(I)/760.D0
      GAMAA(I)=A12*X2(I)*X2(I)/((X(I)*A12/A21+X2(I))**2)
      GAMAB(I)=A21*X(I)*X(I)/((X2(I)*A21/A12+X(I))**2)
      GAMMAB(I)=DEXP(GAMAB(I)*2.30259000)
      GAMMAA(I)=DEXP(GAMAA(I)*2.30259000)
      AK1(I)=AKA(I)*GAMMAA(I)
      AK2(I)=AKB(I)*GAMMAB(I)
   10 CONTINUE
      DO 11 I=1,N
   11 T(I)=1.8D0*T(I)+32.D0
      RETURN
      END
```

FORTRAN IV G LEVEL 1 MATINV DATE = 70341 21/38/10 PAGE 0001

```
      SUBROUTINE MATINV(A,N)
C     SUBROUTINE MATINV   (MATRIX INVERSION)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION INDEX(12,2),A(12,12)
      DO 108 I=1,N
  108 INDEX(I,1)=0
      II=0
  109 AMAX=-1.D0
      DO 110 I=1,N
      IF(INDEX(I,1))110,111,110
  111 DO 112 J=1,N
      IF(INDEX(J,1))112,113,112
  113 TEMP=DABS(A(I,J))
      IF(TEMP-AMAX)112,112,114
  114 IROW=I
      ICOL=J
      AMAX=TEMP
  112 CONTINUE
  110 CONTINUE
      IF(AMAX)225,115,116
  116 INDEX(ICOL,1)=IROW
      IF(IROW-ICOL)119,118,119
  119 DO 120 J=1,N
      TEMP=A(IROW,J)
      A(IROW,J)=A(ICOL,J)
  120 A(ICOL,J)=TEMP
      II=II+1
      INDEX(II,2)=ICOL
  118 PIVOT=A(ICOL,ICOL)
      A(ICOL,ICOL)=1.D0
      PIVOT=1.D0/PIVOT
      DO 121 J=1,N
  121 A(ICOL,J)=A(ICOL,J)*PIVOT
      DO 122 I=1,N
      IF(I-ICOL)123,122,123
  123 TEMP=A(I,ICOL)
      A(I,ICOL)=0.D0
      DO 124 J=1,N
  124 A(I,J)=A(I,J)-A(ICOL,J)*TEMP
  122 CONTINUE
      GO TO 109
  125 ICOL=INDEX(II,2)
      IROW=INDEX(ICOL,1)
      DO 126 I=1,N
      TEMP=A(I,IROW)
      A(I,IROW)=A(I,ICOL)
  126 A(I,ICOL)=TEMP
      II=II-1
  225 IF(II)125,127,125
  115 WRITE(6,1001)
 1001 FORMAT(1H0,2X,11H ZERO PIVOT)
  127 RETURN
      END
```

FCRIPAN IV G LEVEL 18 COEFT DATE = 70341 21/38/10 PAGE 0001

```
0001          SUBROUTINE COEFT(AX,F,AK,V,D,D,F,XK,N)
              C
0002          IMPLICIT REAL*8(A-H,O-Z)
0003          DIMENSION AX(12,12),F(12),AK(12),V(12)
0004          DO 1 I=1,N
0005          DO 1 J=1,N
0006          AX(I,J)=0.0
0007        1 CONTINUE
0008          DO 2 I=1,N
0009          F(I)=0.0
0010        2 CONTINUE
0011          DO 3 I=2,5
0012          AX(I,I-1)=V(I-1)*AK(I-1)
0013          AX(I,I)=-(V(I-1)+H+V(I)*AK(I))
0014        3 AX(I,I+1)=V(I)+R
0015          AX(1,1)=-(H+V(1)*AK(1))
0016          AX(1,2)=V(1)+R
0017          AX(5,5)=-(V(4)+R+V(5)*AK(5))
0018          AX(5,6)=V(5)=R
0019          DO 4 I=6,11
0020          AX(I,I-1)=V(I-1)*AK(I-1)
0021          AX(I,I)=-(V(I-1)=R+V(I)*AK(I))
0022        4 AX(I,I+1)=V(I)-R
0023          AX(12,12)=-V(11)
0024          AX(12,11)=V(11)*AK(11)
0025          F(5)=-2*XK
0026          RETURN
0027          END
```

```
              C
0001          SUBROUTINE AMATM(A6,R6,R6,N)
              C     SUBROUTINE AMATM  (MATRIX MULTIPLICATION)
0002          IMPLICIT REAL*8(A-H,O-Z)
0003          DIMENSION A(12,12),R6(12),R6(12)
0004          DO 310 I=1,N
0005        310 R6(I)=0.DO
0006          DO 311 I=1,N
0007          DO 311 J=1,N
0008        311 R6(I)=R6(I)+A6(I,J)*R6(J)
0009          RETURN
0010          END
```

```
FORTRAN IV G LEVEL 18          ENCON          DATE = 70341      21/30/10          PAGE 0001
0001          SUBROUTINE ENCON(TEMPR,HV,HL,XM,XW,YM,YW,HVW,HVM,HLW,HLM,N)
0002          C
0003          IMPLICIT REAL*8(A-H,O-Z)
0004          DIMENSION TEMPR(12),HV(12),HL(12),HLW(12),HLM(12),HVW(12)
0005          DIMENSION HVM(12),XM(12),XW(12),YM(12),YW(12)
0006          DIMENSION HLWW(12),HLMM(12),HVWW(12),HVMM(12)
0007          DO 1 I=1,N
0008          HLM(I)=-546.6200+18.05400*TEMPR(I)
0009          HLWW(I)=HLW(I)*XW(I)
0010          HLW(I)=-850.2300+20.9600*TEMPR(I)
0011          HLMM(I)=HLM(I)*XM(I)
0012          HVW(I)=(19212.2500+7.059600*TEMPR(I))
0013          HVWW(I)=HVW(I)*XW(I)
0014          HVM(I)=17164.00+1.6200*TEMPR(I)
0015          HVMM(I)=HVM(I)*YM(I)
0016          HV(I)=HVWW(I)+HVMM(I)
0017          HL(I)=HLWW(I)+HLMM(I)
0018        1 CONTINUE
0019          RETURN
0020          END
```

```
              C     SUBROUTINE FOR NON-IDEAL BUBBLE POINTS CALCULATION
0001          SUBROUTINE BUBPTG(X1,X3,T,Y1,Y2,ABF,ABH,L,BPERR)
0002          IMPLICIT REAL*8(A-H,O-Z)
0003          DIMENSION QUIDX(2),GAMMA(2),AKA(2),PPA(2),PA(2)
0004          A12=.378500
0005          A21=.231400
0006          GAMAA=A12*X3**2/((X1+A12/A21*X3)**2)
0007          GAMAB=A21*X1**2/((X3*A21/A12+X1)**2)
0008          GAMMAA=DEXP(GAMAA*2.302585000)
0009          GAMMAB=DEXP(GAMAB*2.302585000)
0010          T=5.00*(T-32.00)/9.00
0011          GAMMA(1)=GAMMAA
0012          GAMMA(2)=GAMMAB
0013          KTIMES=1
0014          QUIDX(1)=X1
0015          QUIDX(2)=X3
0016        3 SUMY=0.0
0017          PPA(1)=8.072460E-1674.9200/(T+238.8400)
0018          PPA(2)=7.96681E0-1668.2100/(T+228.000)
0019          DO 4 I=1,2
0020          PA(I)=DEXP(PPA(I)*2.302585000)
0021          AKA(I)=PA(I)/760.00
0022          Y=AKA(I)*GAMMA(I)*QUIDX(I)
0023        4 SUMY=SUMY+Y
0024          IF(DABS(SUMY-1.00).LE.BPERR) GO TO 8
0025          KTIMES = KTIMES +1
0026          IF (KTIMES.LT.5) GO TO 7
0027          SUMYO = SUMY
0028          TO=T
0029          T=T+.500
0030          GO TO 3
0031        7 SLOPE=(SUMY-SUMYO)/(T-TO)
0032          T=(1.00-SUMY)/SLOPE+T
0033          SUMYO=SUMY
0034          TO=T
0035          T=T
0036          GO TO 3
0037        8 T=.800*T+32.00
0038          Y1=AKA(1)*X1*GAMMA(1)
0039          Y2=AKA(2)*X3*GAMMA(2)
0040          ABF=AKA(1)*GAMMA(1)
0041          ABH=AKA(2)*GAMMA(2)
```

```
FORTRAN IV G LEVEL 18          RUNRTG          DATE = 70341      21/38/10          PAGE 0003
0042                  RETURN
0043                  END
```

COMPUTER PROGRAMS FOR STEADY STATE TWO POINT CONTROL CALCULATION

```
FORTRAN IV G LEVEL  20          MAIN            DATE = 72 89      22/32/13          PAGE 0001

                C     MAIN
0001                  IMPLICIT REAL*8(A-H,O-Z)
0002                  DIMENSION X( , ),MARG( ),A( ),STDEVA( ),K( ),ICONST(3),COV( , ),
                      RCONST(6),DELTA( ),KCOMP( ),W2( ),W3(1, ),W4( ),W( )
0003                  COMMON /NIPT/NY
                C
                C
                C
0004                  REAL*8 K,KCOMP
                C
                C     A(I) ARE GUESSED COEFFICIENTS  NO MUST BE ASSUMED BEFORSHAND
                C
0005                1 FORMAT(4F20.10)
0006               12 FORMAT(4F20.7)
0007                  N=
0008                  NY=0
0009                  M=
0010                  M=
0011                  A( )=60000.00
0012                  WRITE(3, 2)(A( ),I= ,N)
0013                  K( )=0.00
0014                  X( ,1)=0.00
0015                  DO 5 L= ,M
0016                  WRITE(3, 1) X( ,L),K(L)
0017                5 CONTINUE
0018                  ICONST( )=
0019                  ICONST(2)=500
0020                  ICONST(3)=3
0021                  RCONST( )=.00D
0022                  RCONST(2)= 0.000
0023                  RCONST(3)=5.2
0024                  RCONST(4)=.707
0025                  RCONST(5)=2.00
0026                  RCONST(6)=0.00
0027                  DO 20 JQ=1,N
0028               20 IPAR(JQ)=1
0029                  CALL MRQRDT(M,X,K,KCOMP,K,N,IPAR,A,STDEVA,COV,N,SUM,
                      ICONST,RCONST,DELTA,IEND,W ,W2,W3,W4)
0030                  CALL CNFOLM(X,X,2.10,4.00, ,3,2)
0031                  CALL CNFOLM(X,X,2.10,4.00, ,3, )
0032                  RETURN
0033                  END
```

```
0001                  SUBROUTINE DERIV(N,IPAR,M,N ,A,X,F,DF,ISW)
0002                  IMPLICIT REAL*8(A-H,O-Z)
0003                  DIMENSION X( ,M),A(N),DF(N,M),IPAR(N),F(M)
0004                  COMMON /NIPT/NY
                C     CALCULATION OF PARTIAL DERIVATIVES FOR  DYNAMIC EQUATION
                C     K(L)--DEPENDENT VARIABLE
                C
0005                  QR=A(1)
0006                  CALL ROOT(ERR,QR)
0007                  IF(ISW) 3,2,2
0008                3 QQR=QR-50.00
0009                  CALL ROOT(EER,QQR)
0010                  DF( ,1)=(EER-ERR)/(QQR-QR)
                C
0011                2 F( )=ERR
0012                  RETURN
0013                  END
```

```
FORTRAN IV G LEVEL  20          SOLVD          DATE = 72 89      22/32/13          PAGE 0001

0001          SUBROUTINE SOLVD (X,IX,M,N,PREC,DET,TEST)                          SOLVD  3
                                                                                 SOLVD  4
0002          IMPLICIT REAL*8(A-H,O-Z)
0003          INTEGER*2 I1(255), I2(255)                                         SOLVD  5
0004          REAL*8 X(1,M),T,DET,BIG,PREC,TEST,PIV,TEMP,DSQRT,DABS
0005          EQUIVALENCE (BIG,TEMP,PIV)                                         SOLVD  4
0006          DET = .DO                                                          SOLVD  4
0007          TEMP = PREC                                                        SOLVD  4
0008          IF (TEMP .EQ. 0.DO)  TEMP = 0.5D- 4                                SOLVD  4
0009          TEST = 0.DO                                                        SOLVD  4
0010          DO 1 I = 1,M                                                       SOLVD  4
0011          I2(I) = 0                                                          SOLVD  4
0012        1 DO 1 J = 1,M                                                       SOLVD  5
0013          TEST = TEST + X(I,J) * X(I,J)                                      SOLVD  5
0014          TEST = TEMP * DSQRT(TEST) / M                                      SOLVD  5
0015          DO 9 K = 1,M                                                       SOLVD  5
0016          BIG = 0.DO                                                         SOLVD  5
0017          DO 3 I = 1,M                                                       SOLVD  5
0018          IF (I2(I) .NE. 0) GO TO 3                                          SOLVD  5
0019          DO 2 J = 1,M                                                       SOLVD  5
0020          IF (I2(J) .NE. 0) GO TO 2                                          SOLVD  5
0021          T =DABS(X(I,J))                                                    SOLVD  5
0022          IF (T .LE. BIG) GO TO 2                                            SOLVD  6
0023          BIG = T                                                            SOLVD  6
0024          IROW = I                                                           SOLVD  6
0025          JCOL = J                                                           SOLVD  6
0026        2 CONTINUE                                                           SOLVD  6
0027        3 CONTINUE                                                           SOLVD  6
0028          IF (BIG .LE. TEST) RETURN                                          SOLVD  6
0029          I1(K) = JCOL                                                       SOLVD  6
0030          I2(JCOL) = IROW                                                    SOLVD  6
0031          IF (JCOL .EQ. IROW) GO TO 6                                        SOLVD  6
0032          DO 5 J = 1,N                                                       SOLVD  7
0033          TEMP=X(IROW,J)                                                     SOLVD  7
0034          X(IROW,J) = X(JCOL,J)                                              SOLVD  7
0035        5 X(JCOL,J) = TEMP                                                   SOLVD  7
0036          DET = -DET                                                         SOLVD  7
0037        6 DET = DET * X(JCOL,JCOL)                                           SOLVD  7
0038          PIV = 1.DO/X(JCOL,JCOL)                                            SOLVD  7
0039          DO 7 J = 1,N                                                       SOLVD  7
0040          IF (J .NE. JCOL)  X(JCOL,J) = PIV * X(JCOL,J)                      SOLVD  7
0041        7 CONTINUE                                                           SOLVD  7
0042          X(JCOL,JCOL) = PIV                                                 SOLVD  8

0043          DO 9 I = 1,M                                                       SOLVD  8
0044          IF (I .EQ. JCOL)  GO TO 9                                          SOLVD  8
0045          PIV = -X(I,JCOL)                                                   SOLVD  8
0046          DO 8 J = 1,N                                                       SOLVD  8
0047          IF (J .NE. JCOL)  X(I,J) = X(I,J) + PIV * X(JCOL,J)                SOLVD  8
0048        8 CONTINUE                                                           SOLVD  8
0049          X(I,JCOL) = PIV * X(JCOL,JCOL)                                     SOLVD  8
0050        9 CONTINUE                                                           SOLVD  8
0051          DO 11 K = 1,M                                                      SOLVD  8
0052          I = M+1 - K                                                        SOLVD  9
0053          JCOL = I1(I)                                                       SOLVD  9
0054          IROW = I2(JCOL)                                                    SOLVD  9
0055          IF (IROW .EQ. JCOL)  GO TO 11                                      SOLVD  9
0056          DO 10 I = 1,M                                                      SOLVD  9
0057          TEMP = X(I,IROW)                                                   SOLVD  9
0058          X(I,IROW) = X(I,JCOL)                                              SOLVD  9
0059       10 X(I,JCOL) = TEMP                                                   SOLVD  9
0060       11 CONTINUE                                                           SOLVD  9
0061          TEST = 0.DO                                                        SOLVD  9
0062          RETURN                                                             SOLVD 10
0063          END
```

```
FORTRAN IV G LEVEL 20          MRQRDT          DATE = 72089      22/32/13          PAGE 0001

0001          SUBROUTINE MRQRDT(M,X,Y,YCOMP,W,N,IPAR,AZERO,STDEVA,ASTAR,
             1N1,SUMOLD,ICONST,RCONST,DELTA,IE,DELTY,ABAR,D,A )
              C                                                                  MRQRD 9
0002 =        IMPLICIT REAL*8(A-H,O-Z)                                          MRQRD 9
              C                                                                  MRQRD 9
              C                                                                  MRQRD 9
              C                                                                  MRQRD 0
              C                                                                  MRQRD 0
0003          DIMENSION      Y(M),YCOMP(M),W(M),AZERO(N),STDEVA(N),ASTAR(N ,N1),
             1RCONST(6),DELTA(N),IPAR(N),ICONST(3),DELTY(M),ABAR(N1,X( , ),
             2D(N1,1),A1(N)
0004          COMMON /NIPT/NY
              C                                                                  MRQRD 0
0005          REAL*8 LMBDN,NU,K,KDIV,LMBD ,LMBDMN/0.5D-16/,LMBDMX/1.0D70/        MRQRD 0
              C
0006          INTEGER*4 R                                                       MRQRD 0
              C                                                                  MRQRD 0
0007          LOGICAL*1 IWT                                                     MRQRD 0
              C                                                                  MRQRD 0
0008          DATA LMAX/50/,PREC/1.D-20/                                        MRQRD11
              C                                                                  MRQRD11
0009          IE=0                                                              MRQRD11
00 0          R=0                                                               MRQRD11
00 1          IWT=ICONST(1).NE.1                                                MRQRD11
00 2          IMAX=ICONST(2)                                                    MRQRD11
0013          IPRINT=ICONST(3)                                                  MRQRD11
0014          LMBDN=RCONST(1)                                                   MRQRD11
00 5          COSMAX=RCONST(4)                                                  MRQRD11
              C                                                                  MRQRD12
              C          CALCULATE THE INITIAL SUM OF SQUARES                    MRQRD12
              C                                                                  MRQRD12
0016          CALL DERIV(N,IPAR,M,N1,AZERO,X,YCOMP,ASTAR, )
00 7          NY=1                                                              MRQRD12
00 8          ISW =1                                                            MRQRD12
00 9      100 R=R+1                                                             MRQRD12
0020          IF(R.GT.IMAX)GO TO 450                                            MRQRD12
0021          SUM1=0.000                                                        MRQRD12
0022          DO 101 L=1,M                                                      MRQRD12
0023          X1=(Y(L)-YCOMP(L))**2                                             MRQRD12
0024          IF(IWT)X1=X1*W(L)                                                 MRQRD13
0025      101 SUM1=SUM1+X1                                                      MRQRD13
0026          IF(ISW .LT.1) GO TO 250                                           MRQRD13
0027          SUMOLD=SUM

0028          I1=0                                                              MRQRD13
0029          K=1.000                                                           MRQRD13
              C                                                                  MRQRD13
              C          PRINT OUT THE INITIAL PARAMETERS AND ITERATION CONSTANTS MRQRD13
              C                                                                  MRQRD13
0030          IF(IPRINT.NE.0)WRITE(IPRINT,3009)                                 MRQRD13
0031     3009 FORMAT('1')                                                       MRQRD14
0032          IF(IPRINT.NE.0)WRITE(IPRINT,3000)I ,LMBDN,COSMAX,SUMOLD,K,        MRQRD14
             1(IPAR(J),AZERO(J),J=1 ,N)
0033     3000 FORMAT('0',I2,' LAMBDA=',D 8.3,' COS(GAMMA)=',F8.6,' SUM OF SQUA',MRQRD14
             1'RES=',D22.15,' K=',D22. 5/(' ',I4,D 8.3,D 8.3,I4,D 8.3,D 8. 3,MRQRD14
             2 I4,D18.10,I4,D18.10))                                           MRQRD14
              C                                                                  MRQRD14
0034          IF(IMAX.EQ. )RETURN                                               MRQRD14
0035          ISW1=-1                                                           MRQRD14
0036          NU=RCONST(2)                                                      MRQRD14
0037          CONV=RCONST(3)                                                    MRQRD15
0038          KDIV=RCONST(5)                                                    MRQRD15
0039          III=0                                                             MRQRD15
              C                                                                  MRQRD15
              C          WRITE OUT THE STATISTICS FOR THE R TH ITERATION         MRQRD15
              C                                                                  MRQRD15
0040      110 IF(IPRINT.NE.0)WRITE(IPRINT,300 )R,SUMOLD                         MRQRD15
0041     300  FORMAT('0',82X,'NOFEV=',I4,' SUM OF SQUARES=',D22. 5)             MRQRD15
0042          K=1.000                                                           MRQRD15
              C                                                                  MRQRD15
              C          SET UP THE MATRIX SYSTEM WITHOUT HAVING DETERMINED LAMBDA MRQRD16
              C                                                                  MRQRD16
0043          CALL DERIV(N,IPAR,M,N1,AZERO,X,YCOMP,D,3)
0044          DO 111 L=1,M                                                      MRQRD16
0045      111 DELTY(L)=Y(L)-YCOMP(L)                                            MRQRD16
0046          X5=0.000                                                          MRQRD16
0047          N2=0                                                              MRQRD16
0048          DO 120 I=1,N                                                      MRQRD16
0049          IF(IPAR(I).NE. ) GO TO 120                                        MRQRD16
0050          N2=N2+1                                                           MRQRD16
0051          X4=0.000                                                          MRQRD16
0052          DO 121 L=1,M                                                      MRQRD17
0053          YCOMP(L)=D(I,L)                                                   MRQRD17
0054          IF(IWT)YCOMP(L)=YCOMP(L)*W(L)                                     MRQRD17
0055      121 X4=X4+DELTY(L)*YCOMP(L)                                           MRQRD17
0056          J1=N2                                                             MRQRD17
0057          DO 122 J=I,N                                                      MRQRD17
```

```
0058            IF(IPAR(J).NE.1)GO TO 122                                MRQRD17
0059            X2=0.0D0                                                 MRQRD17
0060            DO 123 L=1,M                                             MRQRD17
0061        123 X1=X1+YCOMP(L)*D(J,L)                                    MRQRD17
0062            ASTAR(N2,J1)=X.                                          MRQRD18
0063            J1=J1+1                                                  MRQRD18
0064        122 CONTINUE                                                 MRQRD18
0065            STDEVA(N2)=DSQRT(ASTAR(N2,N2))                           MRQRD18
0066            ASTAR(N2,N2)=1.000                                       MRQRD18
0067            X4=X4/STDEVA(N2)                                         MRQRD18
0068            ABAR(N2)=X4                                              MRQRD18
0069            X5=X5+X4*X4.                                             MRQRD18
0070        120 CONTINUE                                                 MRQRD18
0071            NP1=N2+1                                                 MRQRD18
0072            IF (N2.EQ.1)GO TO 26                                     MRQRD19
0073            J1=N2-1                                                  MRQRD19
0074            DO 125 I=1,J1                                            MRQRD19
0075            X1=STDEVA(I)                                             MRQRD19
0076            IP1=I+1                                                  MRQRD19
0077            DO 125 J=IP1,N2                                          MRQRD19
0078        125 ASTAR(I,J)=ASTAR(I,J)/(X1*STDEVA(J))                     MRQRD19
                C                                                        MRQRD19
                C          SET UP CONVERGENCE CRITERIA                   MRQRD19
                C                                                        MRQRD19
0079         26 IF(M-N2)130,13.,3                                        MRQRD20
0080        130 IE=-3                                                    MRQRD20
0081            IF(IPRINT.NE.0)WRITE(IPRINT,3013)                        MRQRD20
0082       3013 FORMAT('0 M.LT.N -MAKING THE PROBLEM INDETERMINATE')     MRQRD20
0083            GO TO 460                                                MRQRD20
0084        131 CONV1=CONV*(SUMOLD+RCONST(6))                            MRQRD20
0085        200 LMBD1=LMBDN/NU                                           MRQRD20
0086            IF(LMBD1.LT.LMBDMN)GO TO 269                             MRQRD20
                C                                                        MRQRD20
                C          STEP 1.                                       MRQRD20
                C                                                        MRQRD21
0087            I1=1                                                     MRQRD21
0088        210 DO 211 I=1,N2                                            MRQRD21
0089            D(I,I)=1.000+LMBD1                                       MRQRD21
0090            D(I,NP1)=ABAR(I)                                         MRQRD21
0091            IF(I.EQ.N2)GO TO 21                                      MRQRD21
0092            IP1=I+1                                                  MRQRD21
0093            DO 212 J=IP1,N2                                          MRQRD21
0094            D(J,I)=ASTAR(I,J)                                        MRQRD21
0095        212 D(I,J)=ASTAR(I,J)                                        MRQRD21
0096         21 CONTINUE                                                 MRQRD22
                C                                                        MRQRD22
0097            CALL SOLVD (D,N1,N2,NP1,PREC,DET,TEST)                   MRQRD22
0098            IF(    TEST.NE.0.0D0)   GO TO 250                        MRQRD22
                C                                                        MRQRD22
0099            COS1=0.000                                               MRQRD22
0100            X1=0.000                                                 MRQRD22
0101            DO 220 I=1,N2                                            MRQRD22
0102            X2=D(I,NP1)                                              MRQRD22
0103            DELTA(I)=X2/STDEVA(I)                                    MRQRD23
0104            COS1=COS1+X2*ABAR(I)                                     MRQRD23
0105        220 X1=X1+X2*X2                                              MRQRD23
0106            COS1=COS1/DSQRT(X1*X5)                                   MRQRD23
0107        225 I2=0                                                     MRQRD23
0108            DO 230 I=1,N                                             MRQRD23
0109            IF(IPAR(I)-1)23,232,23                                   MRQRD23
0110         23 A1(I)=AZERO(I)                                           MRQRD23
0111            GO TO 230                                                MRQRD23
0112        232 I2=I2+1                                                  MRQRD23
0113            A1(I)=AZERO(I)+K*DELTA(I2)                               MRQRD24
0114        230 CONTINUE                                                 MRQRD24
                C                                                        MRQRD24
0115            CALL DERIV(N,IPAR,M,N1,A1,X,YCOMP,FSTAR,I)               MRQRD24
0116            GO TO 100                                                MRQRD24
0117        250 IF(IPRINT.NE.0)WRITE(IPRINT,3000)I1,LMBD,COS1,SUM1,K,    MRQRD24
                    (IPAR(I),A1(I),I=1,N)                                MRQRD24
                C                                                        MRQRD24
                C          TEST FOR CONVERGENCE                          MRQRD24
                C                                                        MRQRD24
0118            IF(SUM1.LE.SUMOLD.OR.DABS(SUM1-SUMOLD).LE.CONV )GO TO 400 MRQRD25
0119            IF(I1.EQ.2)GO TO 280                                     MRQRD25
0120            IF(I1.EQ.3)GO TO 300                                     MRQRD25
                C                                                        MRQRD25
0121        269 LMBD1=LMBDN                                              MRQRD25
0122        270 I1=2                                                     MRQRD25
0123            GO TO 210                                                MRQRD25
                C                                                        MRQRD25
                C          STEP 2.                                       MRQRD25
                C                                                        MRQRD25
                C          TRY LAMBDA=LAMBDA(R)**S AS LONG AS ALPHA GT ALPHA0 MRQRD25
                C                                                        MRQRD26
0124        280 IF(COS1-COSMAX)281,281,282                               MRQRD26
```

```
0 25          281 LMBD1=LMBD*NU                                          MRQRD26
0 26              GO TO 270                                              MRQRD26
0 27          282 K=1.000                                                MRQRD26
0 28              II=3                                                   MRQRD26
0 29              J=1                                                    MRQRD26
0 30          290 K=K/KDIV                                               MRQRD26
                  C                                                      MRQRD26
                  C          STEP 3                                      MRQRD26
                  C                                                      MRQRD27
                  C          EVALUATE S(DELTA/K**J)                      MRQRD27
                  C                                                      MRQRD27
0 31                  GO TO 225                                          MRQRD27
0 32          300 J=J+1                                                  MRQRD27
0 33              IF(J.LE.LMAX)GO TO 290                                 MRQRD27
0 34              IE=-1                                                  MRQRD27
0 35              GO TO 460                                              MRQRD27
                  C                                                      MRQRD27
                  C          MATRIX WOULD NOT INVERT, INCREASE LAMBDA    MRQRD27
                  C                                                      MRQRD28
0 36          350 LMBDN=LMBDN*NU                                         MRQRD28
0 37              IF(LMBDN.LE.LMBDMX)GO TO 250                           MRQRD28
0 38              IE=-2                                                  MRQRD28
0 39              GO TO 460                                              MRQRD28
                  C                                                      MRQRD28
                  C          TEST FOR CONVERGENCE                        MRQRD28
                  C                                                      MRQRD28
0 40          400 DO 401 J=1,N                                           MRQRD28
0 41          401 AZERO(J)=A(J)                                          MRQRD29
0 42              LMBDN=LMBD                                             MRQRD29
0 43              IF(DABS(SUM1-SUMOLD)-CONV)414,0,4 0,4                  MRQRD29
0 44          414 SUMOLD=SUM1                                            MRQRD29
0 45              GO TO 110                                              MRQRD29
0 46          410 IE=0                                                   MRQRD29
0 47              GO TO 460                                              MRQRD29
0 48          450 IE=1                                                   MRQRD29
0 49          460 RETURN                                                 MRQRD29
                  C                                                      MRQRD29
                  C                                                      MRQRD30
                  C                                                      MRQRD30
                  C          ENTRY POINT WHICH CALCULATES THE VARIOUS CONFIDENCE LIMITS  MRQRD30
                  C                                                      MRQRD30
                  C                                                      MRQRD30
                  C                                                      MRQRD30
```

```
0 50              ENTRY CNFDLM(AMIN,AMAX,TTEST,FTEST,ISW,IPRNT,JMAX)     MRQRD30
                  C                                                      MRQRD30
0 51              DIMENSION AMIN(2),AMAX(2)                              MRQRD30
                  C                                                      MRQRD30
0 52              IF(III.NE.0)GO TO 600                                  MRQRD31
0 53              III=1                                                  MRQRD31
0 54              X3=0.0D0                                               MRQRD31
0 55              IF(M.GT.N2)X3=DSQRT(SUMOLD/(M-N2))                     MRQRD31
0 56              IF(N2.EQ.1)GO TO 502                                   MRQRD31
0 57              II=N2-1                                                MRQRD31
0 58              DO 500 I=1,II                                          MRQRD31
0 59              IP1=I+1                                                MRQRD31
0 60              DO 500 J=IP1,N2                                        MRQRD31
0 61          500 ASTAR(J,I)=ASTAR(I,J)                                  MRQRD31
                  C                                                      MRQRD32
0 62          502 CALL SOLVD (ASTAR,N2,N2,N2,PREC,DET,TEST)              MRQRD32
0 63              IF(TEST.NE.0.0D0)WRITE(3,4000)                         MRQRD32
0 64          4000 FORMAT(1X,'MATRIX INVERSION FAILED')                  MRQRD32
0 65              II=N2+1                                                MRQRD32
0 66              I=N                                                    MRQRD32
                  C                                                      MRQRD32
0 67          510 IF(IPAR(I)-1)505,505,510                               MRQRD32
0 68              STDEVA(I)=0.0D0                                        MRQRD32
0 69              DELTA(I)=0.0D0                                         MRQRD32
0 70              GO TO 512                                              MRQRD32
0 71          511 II=II-1                                                MRQRD32
0 72              DELTA(I)=DELTA(II)                                     MRQRD32
0 73              ABAR(II)=DSQRT(DABS(ASTAR(II,II)))                     MRQRD33
0 74              STDEVA(I)=X3*ABAR(II)/STDEVA(II)                       MRQRD33
0 75          512 X1=ABAR(II)                                            MRQRD33
0 76              J1=N2+1                                                MRQRD33
0 77              J=N                                                    MRQRD33
                  C                                                      MRQRD33
0 78          520 IF(IPAR(I).EQ.1.AND.IPAR(J).EQ.1)GO TO 525             MRQRD33
0 79              X2=0.0D0                                               MRQRD33
0 80              GO TO 530                                              MRQRD33
0 81          525 J1=J1-1                                                MRQRD34
0 82              X2=ASTAR(II,J1)/(X1*ABAR(J1))                          MRQRD34
0 83          530 ASTAR(I,J)=X2                                          MRQRD34
0 84              J=J-1                                                  MRQRD34
0 85              IF(J.GE.1)GO TO 520                                    MRQRD34
0 86              I=I-1                                                  MRQRD34
0 87              IF(I.GE.1)GO TO 50                                     MRQRD34
```

```
                C                                                                MRQRD34
0188                DO 540 I=1,N                                                 MRQRD34
0189                DO 540 J=I,N                                                 MRQRD34
0190              540 ASTAR(J,I)=ASTAR(I,J)                                      MRQRD34
                C                                                                MRQRD35
0191                IF(ISW-NE-1)GO TO 600                                        MRQRD35
0192                IF(IPRNT-EQ-0)GO TO 56                                       MRQRD35
0193                WRITE(IPRNT,30 0)                                            MRQRD35
0194             3010 FORMAT('1',11X,'A(I)', 5X,'STANDARD ERROR', X,'LAST CORRECTION') MRQRD35
0195                DO 541 I=1,N                                                 MRQRD35
0196                IF(IPAR(I)-EQ- )WRITE(3,30 4)AZERO(I),STDEVA(I),DELTA(I)     MRQRD35
0197                IF(IPAR(I)-NE- )WRITE(3,30 4)AZERO(I)                        MRQRD35
0198             3014 FORMAT('0',3025-5)                                         MRQRD35
0199              541 CONTINUE                                                   MRQRD35
0200                WRITE(IPRNT,30 )                                             MRQRD36
0201              301 FORMAT('0 CORRELATION MATRIX')                             MRQRD36
0202                DO 550 I=1,N                                                 MRQRD36
0203              550 WRITE(IPRNT,3003)(ASTAR(I,J),J= ,I)                        MRQRD36
0204             3003 FORMAT('0',10F 3-0)                                        MRQRD36
0205                IF(-NOT-IWT)WRITE(IPRNT,3004)                                MRQRD36
0206             3004 FORMAT('0',20X,'OBSERVED',22X,'ESTIMATED',20X,'DIFFERENCE'//) MRQRD36
0207                IF(IWT)WRITE(IPRNT,30 2)                                     MRQRD36
0208             3012 FORMAT('0',20X,'OBSERVED',22X,'ESTIMATED',2?X,'DIFFERENCE',21X, MRQRD36
                    1    'WEIGHTS'//)                                           MRQRD37
0209                DO 560 L=1,M                                                 MRQRD37
0210                K=Y(L)-YCOMP(L)                                              MRQRD37
0211                IF(-NOT-IWT)WRITE(IPRNT,3005)L,Y(L),YCOMP(L),K               MRQRD37
0212                IF(IWT)WRITE(IPRNT,3005) L,Y(L),YCOMP(L),K,W(L)              MRQRD37
0213             3005 FORMAT(' ',I5,4D30-5)                                      MRQRD37
0214              560 CONTINUE                                                   MRQRD37
0215               56 RETURN                                                     MRQRD37
                C                                                                MRQRD37
0216              600 IF(ISW-3)601,610,620                                       MRQRD37
0217               60 X1=TTEST                                                   MRQRD38
0218                  GO TO 611                                                  MRQRD38
0219              610 X1=DSQRT(N2*FTEST)                                         MRQRD38
0220               611 DO 612 I=1,N                                              MRQRD38
0221                  X2=X1*STDEVA(I)                                            MRQRD38
0222                  AMIN(I)=AZERO(I)-X2                                        MRQRD38
0223              612 AMAX(I)=AZERO(I)+X2                                        MRQRD38
0224                  SCRIT=SUMOLD                                               MRQRD38
0225                  GO TO 800                                                  MRQRD38
                C
```

```
0226              620 SCRIT=SUMOLD*(1.0D0+N2*FTEST/(M-N2))                       MRQRD38
0227                  DO 630 I=1,N                                               MRQRD39
0228                  X2=AZERO(I)                                                MRQRD39
0229                  IF(IPAR(I)-EQ-0)GO TO 640                                  MRQRD39
0230                  AMIN(I)=X2                                                 MRQRD39
0231                  AMAX(I)=X2                                                 MRQRD39
0232                  GO TO 630                                                  MRQRD39
                C                                                                MRQRD39
0233              640 X3=TTEST*STDEVA(I)                                         MRQRD39
0234                  X1=X3                                                      MRQRD39
0235               64 ISW1=0                                                     MRQRD40
0236                  P0=0.0D0                                                   MRQRD40
0237                  PHI0=SUMOLD                                                MRQRD40
0238                  P1=1.0D0                                                   MRQRD40
0239                  DO 650 J=1,JMAX                                            MRQRD40
0240                  AZERO(I)=X2*P1+X                                           MRQRD40
0241                  CALL DERIV(N,IPAR,M,N1,AZERO,X,DELTY,ASTAR, )              MRQRD40
0242                  PHI1=0.0D0                                                 MRQRD40
0243                  DO 651 L=1,M                                               MRQRD40
0244                  X4=(Y(L)-DELTY(L))**2                                      MRQRD40
0245                  IF(IWT)X4=X4*W(L)                                          MRQRD41
0246               65 PHI1=PHI1+X4                                               MRQRD41
0247                  IF(ISW1)660,660,670                                        MRQRD41
0248              660 IF(PHI1-SCRIT)661,661,662                                  MRQRD41
0249               66 P2=2.0D0*P1                                                MRQRD41
0250                  GO TO 680                                                  MRQRD41
0251              662 ISW1=2                                                     MRQRD41
0252                  GO TO 680                                                  MRQRD41
                C                                                                MRQRD41
0253              670 IF(DABS(P1-P0).LT. .0D-2*DABS(P0))GO TO 690                MRQRD41
0254                  P2=P0+(SCRIT-PHI0)*(P1-P0)/(PHI1-PHI0)                     MRQRD42
                C                                                                MRQRD42
0255              680 P0=P1                                                      MRQRD42
0256                  P1=P2                                                      MRQRD42
                C                                                                MRQRD42
0257              650 PHI0=PHI1                                                  MRQRD42
0258                  AZERO(I)=X2                                                MRQRD42
                C                                                                MRQRD42
0259              690 IF(X1)691,691,692                                          MRQRD42
0260              692 AMAX(I)=AZERO(I)                                           MRQRD42
0261                  X1=-X3                                                     MRQRD43
0262                  GO TO 641                                                  MRQRD43
0263              691 AMIN(I)=AZERO(I)
```

```
FORTRAN IV G LEVEL 20          MRQRDT          DATE = 72189          22/32/13          PAGE 0009

0264          630 AZERO(I)=X2                                                        MRQRD43
                  C                                                                 MRQRD43
                  C          OUTPUT THE RESULTS IF DESIRED                          MRQRD43
                  C                                                                 MRQRD43
0265              800 IF(IPRNT.EQ.0)GO TO 850                                       MRQRD43
0266                  WRITE(IPRNT,3008)ISW,TTEST,FTEST,M,N2,SCRIT                   MRQRD43
0267              3008 FORMAT('1CONFIDENCE LIMIT CALCULATIONS USING TECHNIQUE NUMBER',MRQRD43
                     1    13/' T(M-N)=',D22.15,' F(N,M-N)=',D22.15,' FOR',I5,' POINTS',MRQRD43
                     2    ' USING',I5,' PARAMETERS'/'0PHI CRITICAL=',D22.15/'0PARA',  MRQRD44
                     3    8X,'LOWER A',16X,'A',16X,'UPPER A'//)                      MRQRD44
0268                  DO 801 I=1,N                                                  MRQRD44
0269                  IF(IPAR(I)-1)802,803,802                                      MRQRD44
                  C                                                                 MRQRD44
0270              802 WRITE (IPRNT,3007)I,AZERO(I)                                  MRQRD44
0271              3007 FORMAT(' ',I5,20X,D20.10)                                    MRQRD44
0272                  GO TO 801                                                     MRQRD44
0273              803 X1=AZERO(I)-AMIN(I)                                           MRQRD44
0274                  X2=AMAX(I)-AZERO(I)                                           MRQRD44
0275                  WRITE(IPRNT,3006)I,AMIN(I),AZERO(I),AMAX(I),X1,X2             MRQRD45
0276              3006 FORMAT(' ',I5,5D20.10)                                       MRQRD45
0277              801 CONTINUE                                                      MRQRD45
                  C                                                                 MRQRD45
0278              850 RETURN                                                        MRQRD45
0279                  END                                                           MRQRD45
```

```
0001                  SUBROUTINE ROOT(ERROR,QR)
                  C   MANIPULATED VARIABLE D---QR,REFLUX--B
                  C   IJK=1  FALSE POSITION METHOD
                  C   IJK=2 BISECTIONAL METHOD
                  C   ITERATION VARIABLE--QR
                  C   CALCULATION OF THE S.S. VALUE OF THE MANUPULATED VARIABLE
                  C   HEAT AND MATERIAL BALANCE STEADY-STATE CALCULATION FOR MEOH-WATER 00000010
                                                                                    00000020
                  C   THIS METHOD WAS PROPOSED BY AMUNDSON AND PONTINEN              00000030
                  C   REFERENCE--PP348-354 CHAPTER 10 'DESIGN OF EQUILIBRIUN STAGE   00000040
                  C   PROCESSES BY B.D.SMITH                                        00000050
                  C   THIS DISTILLATION PROCESS IS DESCRIBED BY TWO SYSTEMS OF      00000060
                  C   SIMULTANEOUS ALGEBRAIC EQUATION -ONE IN THE COMPOSITION ONE IN 00000070
                  C   THE VAPOR FLOW RATE. THESE TWO SET OF EQUATIONS ARE SOLVED BY MATR00000080
                  C   ALGEBRA.                                                     00000090
                  C   THE ITERATION VARIABLES THAT HAVE TO BE ASSUMED ARE THE TEMPERATUR00000100
                  C   AND VAPOR FLOW PROFILE.                                      00000110
0002                  IMPLICIT REAL*8(A-H,O-Z)
0003                  REAL*8 NQR                                                    00000130
0004                  DIMENSION HLW(12),HLM(12),HVW(12),HVM(12),QX2(12),QXT(12)     00000140
0005                  DIMENSION OLDXT(12),XTRAY(12),BF(12),BG(12),AV(12,12),AX(12,12) 00000150
0006                  DIMENSION DELX(12),HV(12),HL(12),V(12),YTRAY(12),TEMPB(12)    00000160
0007                  DIMENSION GAMA(12),GAMB(12),AL(12)                            00000170
0008                  DIMENSION BK(12),AK(12),Y2(12),XT(12)                         00000180
0009                  DIMENSION BX(12,12),X2(12),CF(12),XTAL(12)
0010                  COMMON /NIPT/NY
0011                  IF(NY.EQ.1)  GO TO 29
0012                  IJK=4                                                         00000190
0013                  L=2                                                           00000200
0014                  N=12                                                          00000210
0015                  NN=N-1                                                        00000220
0016                  KM=N-1                                                        00000240
0017                  READ(1,500)(XTRAY(I),I=1,N)                                   00000250
0018                  READ(1,501)(TEMPB(I),I=1,N)
0019                  ITT=1
0020                  DELTA=200.00
0021                  REFLUX=2.48400
0022                  QR=56800.00
0023                  QF=55250.00                                                   00001300
0024                  HF=26797.00/9.72800
0025                  XD=.9776300
0026                  F=9.72800
0027                  QF=.071500
0028                  XF=.06500
```

```
0029          XB=.340490D-2                                                    00000320
0030          TF=184.DO
0031          ERR=1.D-6
0032          BPERR=ERR
0033          FXTRAY=.9776300
0034          D=F*(XF-XB)/(XD-XB)
0035       26 CONTINUE
0036          IY=0
0037          B=F-D                                                            00000420
0038          Y1=0.DO                                                          00000430
0039          Y3=0.DO                                                          00000440
0040          ABF=0.DO                                                         00000450
0041          ABH=0.DO                                                         00000460
0042          VF=0.DO                                                          00000560
0043          DO 52 I=1,5                                                      00000570
0044          AL(I)=11.8722500                                                 00000580
0045       52 CONTINUE                                                         00000590
0046          DO 53 I=6,12                                                     00000600
0047          AL(I)=2.494200                                                   00000610
0048       53 CONTINUE                                                         00000620
0049          DO 1 I=1,N                                                       00000630
0050          DO 1 J=1,N                                                       00000640
0051          AX(I,J)=0.DO                                                     00000650
0052          BX(I,J)=0.DO                                                     00000660
0053        1 AV(I,J)=0.DO                                                     00000670
0054          DO 2 I=1,N
0055          OX2(I)=0.DO                                                      00000680
0056          HV(I)=0.DO                                                       00000690
0057          HL(I)=0.DO                                                       00000700
0058          BK(I)=0.DO                                                       00000710
0059          AK(I)=0.DO                                                       00000720
0060          Y2(I)=0.DO                                                       00000730
0061          CF(I)=0.DO                                                       00000740
0062          BG(I)=0.DO                                                       00000750
0063          BF(I)=0.DO                                                       00000760
0064          OLDXT(I)=0.DO
0065          XT(I)=0.DO                                                       00000770
0066        2 CONTINUE                                                         00000780
0067          DO 17 I=1,N                                                      00000790
0068          X2(I)=1.DO-XTRAY(I)                                              00000800
0069          V(I)=2.864DO                                                     00000810
0070       17 CONTINUE                                                         00000820
0071          INDEX=1

0072          XG=1.DO-XF                                                       00000830
0073          AL(12)=REFLUX                                                    00000840
0074          V(12)=D
0075          AL(1)=B
0076       29 CONTINUE
0077          KII=0
0078          CALL VANLA(XTRAY,X2,TEMPB,AK,BK,GAMA,GAMB,N)                     00000850
0079          CALL COEFT(AX,BF,AK,V,B,D,F,XF,N)                                00000870
0080          CALL MATINV(AX,N)                                                00000880
0081          CALL AMATM(AX,BF,XTRAY,N)                                        00000890
0082       41 CALL COEFT(BX,CF,BK,V,B,D,F,XG,N)                                00000920
0083          CALL MATINV(BX,N)                                                00000930
0084          CALL AMATM(BX,CF,X2,N)                                           00000940
0085          IN=1                                                             00000950
0086       14 DELX(IN)=DABS(XTRAY(IN)-OLDXT(IN))                               00000960
0087          IF(DELX(IN)-ERR)30,30,31                                         00000970
0088       30 IF(IN-N) 15,16,16
0089       15 IN=IN+1                                                          00000990
0090          GO TO 14                                                         00001000
0091       31 DO 12 I=1,N                                                      00001010
0092          OLDXT(I)=XTRAY(I)
0093       12 CONTINUE                                                         00001030
0094          DO 32 I=1,N                                                      00001040
0095          XT(I)=X2(I)+XTRAY(I)                                             00001050
0096          XTRAY(I)=XTRAY(I)/XT(I)                                          00001060
0097          X2(I)=X2(I)/XT(I)                                                00001160
0098       32 CONTINUE
0099        3 DO 39 I=1,N                                                      00001070
0100          ABC=XTRAY(I)                                                     00001080
0101          ABD=X2(I)                                                        00001090
0102          ABE=TEMPB(I)                                                     00001100
0103          CALL BUBPTG(ABC,ABD,ABE,Y1,Y3,ABF,ABH,L,BPERR)                   00001110
0104          YTRAY(I)=Y1                                                      00001120
0105          Y2(I)=Y3                                                         00001130
0106          AK(I)=ABF                                                        00001140
0107          BK(I)=ABH                                                        00001150
0108          TEMPB(I)=(TEMPB(I)+ABE)/2.DO
0109       39 CONTINUE                                                         00001170
0110       42 CALL ENCON(TEMPB,HV,HL,XTRAY,X2,YTRAY,Y2, HVW,HVM,HLW,HLM,N)     00001350
0111          HB=HL(1)                                                         00001360
0112          HD=HL(12)
0113          IF(IJK.NE.0) GO TO 198                                           00001370
0114          AAB=AL(12)*(HVW(11)*X2(12)+HVM(11)*XT( Y(12)-HL(12))
```

```
FORTRAN IV G LEVEL  20          ROOT          DATE = 72189      22/32/13          PAGE 0004

0115            AAC=HVW(11)*X2(12)+HVM(11)*XTRAY(12)-HD                          00001380
0116            D=(F*(HF-HB)+QR-AAB)/(HD+AAC-HB)                                 00001410
0117            B=F-D                                                            00001420
0118            QC=AL(12)*(HVW(11)*X2(12)+HVM(11)*XTRAY(12)-HL(12)+              00001490
               1D*(HVW(11)*X2(12)+HVM(11)*XTRAY(12)-HD)                          00001500
0119          198 CONTINUE
0120          200 QC=F*(HF-HB)+QR-D*(HD-HB)
0121          209 ALN=B*(HVW(1)*X2(1)+HVM(1)*XTRAY(1))+D*HQ+QC-F*HF              00001510
0122            ALD=HVW(1)*X2(1)+HVM(1)*XTRAY(1)-HL(1)                           00001520
0123            AL(1)=ALN/ALD                                                    00001530
0124            V(1)=AL(1)-B                                                     00001540
0125            DO 50 I=1,3                                                      00001550
0126            ALN=B*(HVW(I)*X2(1)+HVM(I)*XTRAY(1)-HB)+QR                       00001560
0127            ALD=HVW(I)*X2(I+1)+HVM(I)*XTRAY(I+1)-HL(I+1)                     00001570
0128            AL(I+1)=ALN/ALD                                                  00001580
0129            V(I)=AL(I+1)-B                                                   00001590
0130           50 CONTINUE                                                       00001600
0131            DO 51 I=5,11
0132            ALN=D*(HD-(HVW(I)*X2(12)+HVM(I)*XTRAY(12)))+QC                   00001620
0133            ALD=HVW(I)*X2(I+1)+HVM(I)*XTRAY(I+1)-HL(I+1)                     00001630
0134            AL(I+1)=ALN/ALD                                                  00001640
0135            V(I)=AL(I+1)*D                                                   00001650
0136           51 CONTINUE                                                       00001660
0137            AL(1)=B                                                          00001670
0138            V(12)=D                                                          00001680
0139            REFLUX=AL(12)
0140            INDEX=INDEX+1                                                    00001700
0141            GO TO 29
0142           16 ERROR=XTRAY(12)-FXTRAY
0143            WRITE(3,6)(XTRAY(I),I=1,N)
0144            WRITE(3,59) QC,D,B,REFLUX,QR,ERROR
0145            IF(DABS(ERROR).LE.1.D-5) GO TO 63
0146            IF(IJK.GE.3) GO TO 18
0147            IF(INDEX.GT.500) GO TO 67
0148            IF(IJK.EQ.2) GO TO 43
0149            KTT=KTT-1
0150            IF(KTT.LT.0) GO TO 70
0151            OERROR=ERROR
0152            OQR=QR
0153            QR=QR+DELTA
0154            GO TO 26
0155           70 SLOPE=(ERROR-OERROR)/(QR-OQR)
0156            QRN=((0.DO-ERROR)/SLOPE)+QR

0157            OQR=QR
0158            QR=QRN
0159            OERROR=ERROR
0160            GO TO 26
0161           43 IF(ERROR) 45,63,44
0162           44 OQR=QR
0163            QR=(QR+NQR)/2.DO
0164            GO TO 26
0165           45 NQR=QR
0166            QR=(QR+OQR)/2.DO
0167            GO TO 26
0168           63 WRITE(3,139) INDEX,HF,QR                                       00001880
0169            WRITE(3,6)(XTRAY(I),I=1,N)                                       00001790
0170            WRITE(3,156)(X2(I),I=1,N)                                        00001750
0171            WRITE(3,155)(XT(I),I=1,N)                                        00001760
0172            WRITE(3,5)(TEMPB(K),K=1,N)                                       00001770
0173            WRITE(3,11)(V(I),I=1,KM)                                         00001780
0174            WRITE(3,55)(AL(I),I=1,N)                                         00001800
0175            WRITE(3,33)(HV(I),I=1,N)                                         00001810
0176            WRITE(3,34)(HL(I),I=1,N)
0177            WRITE(3,135)(AK(I),I=1,N)
0178            GO TO 18
0179           67 WRITE(3,66) INDEX
0180           18 RETURN
0181          135 FORMAT(1HO,2X,'AK='/(10D12.6))
0182           66 FORMAT(1HO,5X,' THE S. S. VALUE CANNOT BE OBTAINED IN',I6,
               1'TRIALS ')
0183          139 FORMAT(1HO,2X,'INDEX=   ',I5,5X,'HF=',D12.6,5X,'QR=',D12.6)    00001870
0184           33 FORMAT(1HO,2X,'HV='/(10E12.6))                                 00001920
0185           34 FORMAT(1HO,2X,'HL='/(10E12.6))                                 00001930
0186           55 FORMAT(1HO,2X,'AL='/(10E12.6))                                 00001940
0187           59 FORMAT(1HO,2X,'QC=',E12.6,5X,'D=',E12.6,5X,'B=',E12.6,5X,      00001950
               1'REFLUX=',D12.6,5X,'QR=',D12.6,5X,'ERROR=',D14.8)
0188          156 FORMAT(1HO,2X,  'X'   ='  /(10E12.6))                          00001960
0189            5 FORMAT(1HO,2X,'TEMPB='/(10F12.6))                             00001980
0190          155 FORMAT(1HO,2X,'XT(I)=  '/(10E12.6))                           00001990
0191           11 FORMAT(1HO,2X,'V='/(10E12.6))                                 00002000
0192            6 FORMAT(1HO,2X,'XTRAY='/(10E12.6))                             00002010
0193          500 FORMAT(8F10.5)                                                00002020
0194          501 FORMAT(8F10.5)                                                00002030
0195              END
```

```
FORTRAN IV G LEVEL 20          VANLA          DATE = 72189      22/32/13          PAGE 0001

0001          SUBROUTINE VANLA(X,X2,T,AK1,AK2,GAMMAA,GAMMAB,N)                  00002040
              C  SUBROUTINE VANLA                                               00002050
0002          IMPLICIT REAL*8(A-H,O-Z)                                          00002060
0003          DIMENSION PPA(12),PPB(12),PA(12),PB(12),AKA(12),AKB(12),X(12)     00002070
0004          DIMENSION X2(12),GAMAA(12),GAMAB(12),GAMMAA(12),GAMMAB(12),Y(12)  00002080
0005          DIMENSION AK1(12),AK2(12),T(12)                                   00002090
0006          A12=.3785D0                                                       00002100
0007          A21=.2116D0                                                       00002110
0008          DO 10 I=1,N                                                       00002120
0009          T(I)=5.D0*(T(I)-32.D0)/9.D0                                       00002130
0010          PPA(I)=8.07246D0-1574.99D0/(T(I)+238.86)                          00002140
0011          PPB(I)=7.96681D0-1668.21D0/(T(I)+228.D0)                          00002150
0012          PA(I)=DEXP(PPA(I)*2.302585D0)                                     00002160
0013          PB(I)=DEXP(PPB(I)*2.302585D0)                                     00002170
0014          AKA(I)=PA(I)/760.DO                                               00002180
0015          AKB(I)=PB(I)/760.DO                                               00002190
0016          GAMAA(I)=A12*X2(I)*X2(I)/((X(I)+A12/A21*X2(I))**2)                00002200
0017          GAMAB(I)=A21*X(I)*X(I)/((X2(I)*A21/A12+X(I))**2)                  00002210
0018          GAMMAB(I)=DEXP(GAMAB(I)*2.302585D0)                               00002220
0019          GAMMAA(I)=DEXP(GAMAA(I)*2.302585D0)                               00002230
0020          AK1(I)=AKA(I)*GAMMAA(I)                                           00002240
0021          AK2(I)=AKB(I)*GAMMAB(I)                                           00002250
0022       10 CONTINUE                                                          00002260
0023          DO 11 I=1,N                                                       00002270
0024       11 T(I)=1.800*T(I)+32.00                                             00002280
0025          RETURN                                                            00002290
0026          END                                                               00002300
```

```
0001          SUBROUTINE MATINV(A,N)                                            00002310
              C  SUBROUTINE MATINV  (MATRIX INVERSION)                          00002320
0002          IMPLICIT REAL*8(A-H,O-Z)                                          00002330
0003          DIMENSION INDEX(12,2),A(12,12)                                    00002340
0004          DO 108 I=1,N                                                      00002350
0005       108 INDEX(I,1)=0                                                     00002360
0006          II=0                                                              00002370
0007       109 AMAX=-1.                                                         00002380
0008          DO 110 I=1,N                                                      00002390
0009          IF(INDEX(I,1))110,111,110                                         00002400
0010       111 DO 112 J=1,N                                                     00002410
0011          IF(INDEX(J,1))112,113,112                                         00002420
0012       113 TEMP=DABS(A(I,J))                                                00002430
0013          IF(TEMP-AMAX)112,112,114                                          00002440
0014       114 IROW=I                                                           00002450
0015          ICOL=J                                                            00002460
0016          AMAX=TEMP                                                         00002470
0017       112 CONTINUE                                                         00002480
0018       110 CONTINUE                                                         00002490
0019          IF(AMAX)225,115,116                                               00002500
0020       116 INDEX(ICOL,1)=IROW                                               00002510
0021          IF(IROW-ICOL)119,118,119                                          00002520
0022       119 DO 120 J=1,N                                                     00002530
0023          TEMP=A(IROW,J)                                                    00002540
0024          A(IROW,J)=A(ICOL,J)                                               00002550
0025       120 A(ICOL,J)=TEMP                                                   00002560
0026          II=II+1                                                           00002570
0027          INDEX(II,2)=ICOL                                                  00002580
0028       118 PIVOT=A(ICOL,ICOL)                                               00002590
0029          A(ICOL,ICOL)=1.DO                                                 00002600
0030          PIVOT=1.DO/PIVOT                                                  00002610
0031          DO 121 J=1,N                                                      00002620
0032       121 A(ICOL,J)=A(ICOL,J)*PIVOT                                        00002630
0033          DO 122 I=1,N                                                      00002640
0034          IF(I-ICOL)123,122,123                                             00002650
0035       123 TEMP=A(I,ICOL)                                                   00002660
0036          A(I,ICOL)=0.DO                                                    00002670
0037          DO 124 J=1,N                                                      00002680
0038       124 A(I,J)=A(I,J)-A(ICOL,J)*TEMP                                     00002690
0039       122 CONTINUE                                                         00002700
0040          GO TO 109                                                         00002710
0041       125 ICOL=INDEX(II,2)                                                 00002720
0042          IROW=INDEX(ICOL,1)                                                00002730
```

```
0043            DO 126 I=1,N                                            00002740
0044            TEMP=A(I,IROW)                                          00002750
0045            A(I,IROW)=A(I,ICOL)                                     00002760
0046          126 A(I,ICOL)=TEMP                                        00002770
0047            II=II-1                                                 00002780
0048          225 IF(II)125,127,125                                     00002790
0049          115 WRITE(6,1001)                                         00002800
0050         1001 FORMAT(1HO,2X,11H ZERO PIVOT)                         00002810
0051          127 RETURN                                                00002820
0052              END                                                   00002830
```

```
0001              SUBROUTINE COEFT(AX,E,AK,V,B,D,F,XK,N)                00002840
                  C                                                     00002850
                                                                        00002860
0002              IMPLICIT REAL*8(A-H,O-Z)                              00002870
0003              DIMENSION AX(12,12),E(12),AK(12),V(12)                00002880
0004              DO 1 I=1,N                                            00002890
0005              DO 1 J=1,N                                            00002900
0006              AX(I,J)=0.0                                           00002910
0007            1 CONTINUE                                              00002920
0008              DO 2 I=1,N                                            00002930
0009              E(I)=0.0                                              00002940
0010            2 CONTINUE                                              00002950
0011              DO 3 I=2,5                                            00002960
0012              AX(I,I-1)=V(I-1)*AK(I-1)                              00002970
0013              AX(I,I)=-(V(I-1)+B+V(I)*AK(I))                        00002980
0014            3 AX(I,I+1)=V(I)+B                                      00002990
0015              AX(1,1)=-(B+V(1)*AK(1))                               00003000
0016              AX(1,2)=V(1)+B                                        00003010
0017              AX(5,5)=-(V(4)+B+V(5)*AK(5))                          00003020
0018              AX(5,6)=V(5)-D                                        00003030
0019              DO 4 I=6,11                                           00003040
0020              AX(I,I-1)=V(I-1)*AK(I-1)                              00003050
0021              AX(I,I)=-(V(I-1)-D+V(I)*AK(I))                        00003060
0022            4 AX(I,I+1)=V(I)-D                                      00003070
0023              AX(12,12)=-V(11)                                      00003080
0024              AX(12,11)=V(11)*AK(11)                                00003090
0025              E(5)=-F*XK                                            00003100
0026              RETURN                                                00003110
0027              END
```

```
FORTRAN IV G LEVEL 20          MAIN          DATE = 72189      22/32/13          PAGE 0001

0001          C                                                              00003120
              C     SUBROUTINE AMATM(A6,B6,R6,N)                             00003130
              C     SUBROUTINE AMATM  (MATRIX MULTIPLICATION)                00003140
0002                SUBROUTINE AMATM(A6,B6,R6,N)                             00003150
0003                IMPLICIT REAL*8(A-H,O-Z)                                 00003160
0004                DIMENSION A6(12,12),B6(12),R6(12)                        00003170
0005                DO 310 I=1,N                                             00003180
0006          310   R6(I)=0.D0                                               00003190
0007                DO 311 I=1,N                                             00003200
0008                DO 311 J=1,N                                             00003210
0009          311   R6(I)=R6(I)+A6(I,J)*B6(J)                                00003220
0010                RETURN                                                   00003230
0011                END
```

```
0001          C     SUBROUTINE ENCONK(TEMPB,HV,HL,XM,XW,YM,YW, HVW,HVM,HLW,HLM,N)   00003240
              C                                                              00003250
0002                IMPLICIT REAL*8(A-H,O-Z)                                 00003260
0003                DIMENSION TEMPB(12),HV(12),HL(12),HLW(12),HLM(12),HVW(12)  00003270
0004                DIMENSION HVM(12),XM(12),XW(12),YM(12),YW(12)            00003280
0005                DIMENSION HLWW(12),HLMM(12),HVWW(12),HVMM(12)            00003290
0006                DO 1 I=1,N                                               00003300
0007                HLW(I)=-586.6200+18.0540*TEMPB(I)                        00003310
0008                HLWW(I)=HLW(I)*XW(I)                                     00003320
0009                HLM(I)=-850.2300+20.9600*TEMPB(I)                        00003330
0010                HLMM(I)=HLM(I)*XM(I)                                     00003340
0011                HVW(I)=(19212.25D0+7.0596D0*TEMPB(I))                    00003350
0012                HVWW(I)=HVW(I)*YW(I)                                     00003360
0013                HVM(I)=17164.D0+1.6200*TEMPB(I)                          00003370
0014                HVMM(I)=HVM(I)*YM(I)                                     00003380
0015                HV(I)=HVWW(I)+HVMM(I)                                    00003390
0016                HL(I)=HLWW(I)+HLMM(I)                                    00003400
0017              1 CONTINUE                                                 00003410
0018                RETURN                                                   00003420
0019                END                                                      00003430
```

FORTRAN IV G LEVEL 20 MAIN DATE = 72189 22/32/13 PAGE 0001

```
      C                                                                 00003440
      C     SUBROUTINE FOR NON-IDEAL BUBBLE POINTS CALCULATION          00003450
0001        SUBROUTINE BUBPTG(X1,X3,T,Y1,Y2,ABF,ABH,L,BPERR)            00003460
0002        IMPLICIT REAL*8(A-H,O-Z)                                    00003470
0003        DIMENSION QUIDX(2),GAMMA(2),AKA(2),PPA(2),PA(2)             00003480
0004        A12=.378500                                                 00003490
0005        A21=.211400                                                 00003500
0006        GAMAA=A12*X3*X3/((X1+A12/A21*X3)**2)                        00003510
0007        GAMAB=A21*X1*X1/((X3*A21/A12+X1)**2)                        00003520
0008        GAMMAA=DEXP(GAMAA*2.302590D0)                               00003530
0009        GAMMAB=DEXP(GAMAB*2.302590D0)                               00003540
0010        T=5.D0*(T-32.D0)/9.D0                                       00003550
0011        GAMMA(1)=GAMMAA                                             00003560
0012        GAMMA(2)=GAMMAB                                             00003570
0013        KTIMES=1                                                    00003580
0014        QUIDX(1)=X1                                                 00003590
0015        QUIDX(2)=X3                                                 00003600
0016      3 SUMY=0.0                                                    00003610
0017        PPA(1)=8.0724600-1574.9900/(T+238.8600)                     00003620
0018        PPA(2)=7.9668100-1668.2100/(T+228.D0)                       00003630
0019        DO 4 I=1,L                                                  00003640
0020        PA(I)= DEXP(PPA(I)*2.302590D0)                              00003650
0021        AKA(I)=PA(I)/760.D0                                         00003660
0022        Y=AKA(I)*GAMMA(I)*QUIDX(I)                                  00003670
0023      4 SUMY=SUMY+Y                                                 00003680
0024        IF(DABS(SUMY-1.D0).LE.BPERR) GO TO 8                        00003690
0025        KTIMES = KTIMES -1                                          00003700
0026        IF (KTIMES.LT.0)  GO TO 7                                   00003710
0027        SUMYO = SUMY                                                00003720
0028        TO=T                                                        00003730
0029        T=T+.5D0                                                    00003740
0030        GO TO 3                                                     00003750
0031      7 SLOPE=(SUMY-SUMYO)/(T-   TO)                                00003760
0032        TN=((1.D0-SUMY)/SLOPE)+T                                    00003770
0033        SUMYO=SUMY                                                  00003780
0034        TO=T                                                        00003790
0035        T=TN                                                        00003800
0036        GO TO 3                                                     00003810
0037      8 T=1.8D0*T+32.D0                                             00003820
0038        Y1=AKA(1)*X1*GAMMA(1)                                       00003830
0039        Y2=AKA(2)*X3*GAMMA(2)                                       00003840
0040        ABF=AKA(1)*GAMMA(1)                                         00003850
0041        ABH=AKA(2)*GAMMA(2)                                         00003860

0042        RETURN                                                      00003870
0043        END                                                         00003880
```

```
FORTRAN IV G LEVEL 20          MAIN          DATE = 72189      11/22/21      PAGE 0001

                C     MAIN                                                    00006730
0001                  IMPLICIT REAL*8(A-H,O-Z)                                00006740
                C                                                             00006750
                C     SUBROUTINE SEARCH(NDIM,H)                               00006760
                C     PATTERN SEARCH                                          00006770
                C                        REF.D.J.WILDE                        00006780
                C                        OPTIMUM SEEKING METHOD               00006790
                C                        PRENTICE-HALL (1964) PP.145          00006800
                C
0002                  COMMON /SECH/ NK,NSIG
0003                  READ(1,10) PO,CC,DO,EAR
0004                  READ(1,7) NOCAL,NOPAR
0005                  XO=0.DO                                                 00006850
0006                  NK=0
0007                  NYC=0                                                   00006880
0008                  NY=0                                                    00006890
0009                  NOCAL=NOCAL                                             00006900
0010                  CALL ROOT(XO,PO,NOCAL)
                C                                                             00006920
                C     LOCAL SEARCH                                            00006930
                C                                                             00006940
0011                  NK=1
0012                  NEC=0                                                   00006960
0013                  BESTE=XO                                                00006970
0014                1 NEC=NEC+1                                               00006980
0015                  IF(NEC.GE.20)  GO TO 81
0016                  T=PO
0017                  PO=PO+CC
0018                  CALL ROOT(XO,PO,NOCAL)
0019                  IF(NOCAL) 82,82,3                                       00007040
0020                3 IF(XO.LT.BESTE)  GO TO 4
0021                  PO=PO-2.DO*CC
0022                  CALL ROOT(XO,PO,NOCAL)
0023                  IF(NOCAL) 82,82,6                                       00007080
0024                6 IF(XO.GT.BESTE)      GO TO 8
0025                  CC=-CC
0026                4 IF(DABS(XO).LE.EAR)  GO TO 84
0027                9 BESTE=XO
0028                  GO TO 22
0029                8 PO=T
0030                  IF(NYC.EQ.1)  GO TO 36
0031                  IF(NY-1) 36,37,37
0032               35 CC=.75D0*CC

0033                  NY=1
0034                  GO TO 1
0035               37 CC=8.D0*CC/3.D0                                         00007320
0036                  NY=0                                                    00007330
0037                  GO TO 1                                                 00007340
                C                                                             00007350
                C     MOVING THE PATTERN                                      00007360
                C
0038               22 YT=PO
0039                  PO=PO+CC
0040                  CALL ROOT(XO,PO,NOCAL)                                  00007420
0041                  IF(NOCAL) 82,82,18
0042               18 IF(XO.GT.BESTE) GO TO 19                                00007440
0043                  BESTE=XO
0044                  GO TO 22
0045               19 PO=YT                                                   00007500
                C                                                             00007510
                C     DECREASE THE STEP SIZE                                  00007520
                C
0046                  CC=CC/3.DO                                              00007540
0047                  NEC=0                                                   00007570
0048                  NYC=1                                                   00007580
0049                  NY=0
0050                  IF(DABS(CC).GE.DO)  GO TO 1                             00007630
0051                  GO TO 83
0052               81 WRITE(3,33)                                             00007620
0053                  GO TO 80                                                00007630
0054               82 WRITE(3,25)  NOCAL                                      00007640
0055                  GO TO PO
0056               83  WRITE(3,26) CC                                         00007660
0057                  GO TO 80                                                00007670
0058               84 WRITE(3,27)
0059               80 WRITE(3,26) PO                                          00007690
0060                  WRITE(3,29) XO                                          00007700
0061                  NOCAL=NOCAL+NOCAL                                       00007710
0062                  WRITE(3,86) NOCAL                                       00007720
0063                  RETURN                                                  00007730
0064               27 FORMAT(1H0,5X,'NO. OF TRIALS =',I5)                     00007740
0065               33 FORMAT(1H0,5X,'LOCAL SEARCH FAILED')                    00007750
0066               25 FORMAT(1H0,5X,'SMALLEST CRITERION CAN NOT BE SATISFIED WITHIN',
                     1110,5X,'TRIALS')                                        00007760
0067               26 FORMAT(1H0,5X,'CC(1)=',D12.6,5X,'CC(2)=',D12.6,5X,'THE STEPS ARE TOO 00007770
                     1O SMALL TO CONTINUE THE SEARCH')
```

```
FORTRAN IV G LEVEL 20          MAIN          DATE = 72189      11/22/21          PAGE 0003

0068          27 FORMAT(1HO,5X,' THE MINIMUM HAS BEEN LOCATED')                  00007790
0069          29 FORMAT(1HO,5X,'ERROR=',D12.6)
0070          28 FORMAT(1HO,5X,'ZC(1)=',D12.6,5X,'ZC(2)=',D12.6)                 00007810
0071          10 FORMAT(4F20.7)
0072           7 FORMAT(2I10)                                                    00007820
0073             END
```

COMPUTER PROGRAM FOR DYNAMIC CALCULATION

```
        COMPILER OPTIONS - NAME= MAIN,OPT=02,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                C
                C
                C       MAIN
                C
ISN 0002                IMPLICIT REAL*8(A-H,O-Z)
                C       THE FIRST S.S. WAS WITH XF=.0715,F=9.728,REFLUX=2.7,QR=51985.86
                C       FEED COMPOSITION DISTURBANCE
                C       FEED COMPOSITION CHANGED FROM .0715---.065
                C       IDIST=0   FEED COMPOSITION DISTURBANCE
                C       IDIST=1   FEED RATE DISTURBANCE
                C       IDIST=2   REFLUX DISTURBANCE
                C       IDIST=3   REBOILER HEAT INPUT DISTURBANCE
                C       IDIST=4   FEED ENTHALPHY DISTURBANCE
ISN 0003                IDIST=0
ISN 0004                CALL DIFFT(IDIST)
ISN 0005                RETURN
ISN 0006                END
```

```
        COMPILER OPTIONS - NAME= MAIN,OPT=02,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002        SUBROUTINE DIFFT(IDIST)
                C
                C       SUBROUTINE DIFFT
                C       IMPLICIT METHOD FOR SOLVING SYSTEM OF FIRST ORDER
                C       DIFFERENTIAL EQUATIONS
                C       REFERENCE : CHAPTER 3 UNSTEADY STATE PROCESSES WITH APPLICATION
                C       IN MULTICOMPONENT DISTILLATION
                C       THE MATRIX METHOD PROPOSED BY AMUDSON AND PONTINEN FOR SOLVING
                C       STEADY STATE PROBLEM WAS USED TO FIND THE CONVERGENT
                C       MATERIAL AND ENERGY BALANCE SOLUTIONS FOR EACH TIME PERIOD
                C       OF THE TRANSIENT STATE
ISN 0003                IMPLICIT REAL*8(A-H,O-Z)
ISN 0004                REAL*8 XLAG(400),XLAG2(400),VV(12),CTEMP(12),DVAP(12)
ISN 0005                REAL*8 HCUPW(12),HCUPP(12),FHCUP(12),PINCT(12)
ISN 0006                REAL*8 HVM(12),HVW(12),HLM(12),HLW(12),FXTRAY(12),XTDIF(12)
ISN 0007                REAL*8 CZETAT(12),DZETA(11),GRAD(11,11)
ISN 0008                REAL*8 ZETA(12),ZETAN(12),UDM(12),UDW(12),GFUNT(12)
ISN 0009                REAL*8 CHDUPW(12),CHDUPP(12),GFUN(12)
ISN 0010                REAL*8 CVAPOR(12),CTEMPB(12),DXTRAY(12)
ISN 0011                REAL*8 FCCF(12),GCCF(12),FCCBF(12)
ISN 0012                REAL*8 GCOFF(12),Y2(12),Y2(12),DX2(12),CY1(12),DY2(12)
ISN 0013                REAL*8 CHCUP(12),CHV(12),CHL(12),HL(12),HV(12)
ISN 0014                REAL*8 XTRAY(12),TEMPB(12),FLOW(12),VAPOR(12),DAK1(12),DAK2(12)
ISN 0015                REAL*8 AK2(12),TAU(12),CCF(12,12),COFF(12,12),COFB(12),X2(12)
ISN 0016                REAL*8 VAA(12),VAB(12),ABFA(12),ABFB(12),HDUP(12),AK1(12)
ISN 0017                REAL*8 CFLOW(12),DVAA(12),DVAB(12),DABFA(12),DABFB(12),CTAU(12)
ISN 0018                REAL*8 CCFFB(12),ZFA(12),ZFB(12),ZFDA(12),ZFDB(12)
ISN 0019                COMMON  /CLD/ CHL,CHV,CTEMPB
ISN 0020                COMMON / ND/ N,NA,L
ISN 0021                COMMON /RATE/ VAPOR,FLOW,QQ
ISN 0022                COMMON /FXFD/XF,F,D,B
ISN 0023                COMMON /HP/ HDUP,DHCUP,FHCUP
ISN 0024                COMMON /CLOS/ QQ,QB,QF,QXF
ISN 0025                COMMON /DLCCOM/ CXTRAY,CX2,CY1,CY2
ISN 0026                COMMON /COMP/ XTRAY,X2,Y1,Y2
ISN 0027                COMMON /DWM/ DWGG,CMCC,UDW,UDM,BDW,BDM
ISN 0028                COMMON /VHF/ CVAPOR,CFLOW
ISN 0029                COMMON /VLWM/HVW,HLW,HVM,HLM
ISN 0030                COMMON /CITUB/XFC,FCIB
ISN 0031                COMMON /MMTT/ MTIME,MAB
ISN 0032                DATA BPERR/1.D-6 /,TF/193.DO/,IFF/5/
```

```
ISN 0033        54 FORMAT(1HO,2X,  'XTOIF(I)='/(10D12.6))
ISN 0034       173 FORMAT(1HO,2X,'  D=  ',D12.6,2X,'  B=  ',D12.6)
ISN 0035         8 FORMAT(1HO,2X,'CTEMPB='/(8D12.6))
ISN 0036        40 FORMAT(1HO,2X,'QC(I)='/(8D12.6))
ISN 0037       200 FORMAT(1HO,2X,'CHQUPM=  '/(8D12.6))
ISN 0038       201 FORMAT(1HO,2X,'CHQLPM=  '/(8D12.6))
ISN 0039       202 FORMAT(1HO,2X,4D12.6)
ISN 0040       204 FORMAT(1HO,2X,'HQUPM='/(8D12.6))
ISN 0041       203 FORMAT(1HO,2X,'HQUPM='/(8D12.6))
ISN 0042        80 FORMAT(4D20.10)
ISN 0043         1 FORMAT(8D10.5)
ISN 0044         2 FORMAT(8D10.5)
ISN 0045        16 FORMAT(1HO,2X,10D12.6)
ISN 0046        18 FORMAT(1HC,2X,'XTRAY(I)',5X,'TEMPB(I)',4X,'AK1(I)',5X,
                  1'AK2(I)',4X,'FLOW(I)',5X,'VAPOR(I)',6X,'HL(I)',7X,
                  1'HV(I)',7X,'HQUP(I)',7X,'ZETA(I)')
ISN 0047         4 FORMAT(1HO,2X,'TOTIME='/(10D12.6))
ISN 0048       104 FORMAT(1HO,20X,'ERROR')
ISN 0049       105 FORMAT(1HO,2X,8D12.6)
ISN 0050           READ(1,1)XF,F,D,B,REF
ISN 0051           L=2
ISN 0052           N=12
ISN 0053           IKK=0
ISN 0054            IAA=0
ISN 0055           NK=N
ISN 0056           NN=N-1
ISN 0057           KK=N-1
ISN 0058           LK=L
ISN 0059           MAB=0
ISN 0060           FDIB=9.72900
ISN 0061           XFD=.06500
ISN 0062           REFLUX=2.7000
ISN 0063           DWCO=0.00
ISN 0064           DMCO=0.00
ISN 0065           B=9.2322800
ISN 0066           D=.4957700
ISN 0067           DMUE=.600
ISN 0068           INC=1
ISN 0069           TINCT=1.00/60.00
ISN 0070           TOTIME=0.00
ISN 0071           MTIME=0
ISN 0072        82 READ(1,80)(XTRAY(I),I=1,N)
ISN 0073           READ(1,80)(TEMPB(I),I=1,N)

ISN 0074           READ(1,80)(VAPOR(I),I=1,N)
ISN 0075           READ(1,80)(FLOW(I),I=1,N)
ISN 0076           READ(1,2)( HQUP(I),I=1,N)
ISN 0077           READ(1,2)(ZETA(I),I=1,N)
ISN 0078           WRITE(3,2)(XTRAY(I),I=1,N)
ISN 0079           WRITE(3,2)(TEMPB(I),I=1,N)
ISN 0080           WRITE(3,2)(VAPOR(I),I=1,N)
ISN 0081           WRITE(3,2)(FLOW(I),I=1,N)
ISN 0082           WRITE(3,2)(HQUP(I),I=1,N)
ISN 0083           WRITE(3,2)(ZETA(I),I=1,N)
ISN 0084           SIGMA=(1.00-DMUE)/DMUE
ISN 0085           OXF=XF
ISN 0086           DO  3 I=1,N
ISN 0087           X2(I)=1.00-XTRAY(I)
ISN 0088           FHQUP(I)=HQUP(I)
ISN 0089           FXTRAY(I)=XTRAY(I)
ISN 0090         3 CONTINUE
ISN 0091           XST=XTRAY(12)
ISN 0092           XST2=X2(12)
ISN 0093           Y3=0.00
ISN 0094           Y4=0.00
ISN 0095           ABF=0.00
ISN 0096           ABH=0.00
ISN 0097           PER=TINCT
ISN 0098           DO 25 I=1,400
ISN 0099           XLAG1(I)=0.00
ISN 0100           XLAG2(I)=0.00
ISN 0101        25 CONTINUE
ISN 0102           DO 83 I=1,N
ISN 0103           PINCT(I)=1.D-10
ISN 0104           ZETA(I)=1.00
ISN 0105           ZETA1(I)=0.00
ISN 0106           ZETAN(I)=0.00
ISN 0107        83 CONTINUE
ISN 0108           DO 4 I=1,N
ISN 0109           OTEMPB(I)=0.00
ISN 0110           OVAPOR(I)=0.00
ISN 0111           HL(I)=0.00
ISN 0112           HV(I)=0.00
ISN 0113           OHL(I)=0.00
ISN 0114           OHV(I)=0.00
ISN 0115           FCOF(I)=0.00
ISN 0116           FCOFF(I)=0.00
```

```
ISN 0117          GCOF(I)=0.00
ISN 0118          GCOFF(I)=0.00
ISN 0119          VAA(I)=C.00
ISN 0120          VAB(I)=C.00
ISN 0121          UDM(I)=C.00
ISN 0122          UDW(I)=C.00
ISN 0123          OZETA(I)=0.00
ISN 0124          GFUNT(I)=0.00
ISN 0125          GFUN(I)=C.00
ISN 0126          GOFB(I)=C.00
ISN 0127          GOFFB(I)=0.00
ISN 0128          ZFA(I)=C.00
ISN 0129          ZFB(I)=C.00
ISN 0130          ZFDA(I)=0.00
ISN 0131          ZFDB(I)=C.00
ISN 0132          AK1(I)=C.00
ISN 0133          AK2(I)=0.00
ISN 0134          OAK1(I)=0.00
ISN 0135          OAK2(I)=0.00
ISN 0136          HV=(I)=C.00
ISN 0137          HVM(I)=0.00
ISN 0138          HLM(I)=C.00
ISN 0139          HLW(I)=C.00
ISN 0140          HOUPW(I)=0.00
ISN 0141          HOUPM(I)=0.00
ISN 0142          OHOUP(I)=HOUP(I)
ISN 0143        4 CONTINUE
ISN 0144          DO 24 I=1,N
ISN 0145          DO 24 J=1,N
ISN 0146          COF(I,J)=0.00
ISN 0147          COFF(I,J)=0.00
ISN 0148       24 CONTINUE
ISN 0149          DO 84 I=1,NN
ISN 0150          DO 84 J=1,NN
ISN 0151          OAADI(I,J)=0.00
ISN 0152       84 CONTINUE
ISN 0153          ZFA(5)=CXF*F
ISN 0154          ZFB(5)=F*(1.00-CXF)
ISN 0155          ZFDA(5)=XFD*FDIB
ISN 0156          ZFDB(5)=FDIB*(1.00-XFD)
ISN 0157          OF=F
ISN 0158          OD=D
ISN 0159          OB=B
```

```
ISN 0160          B=FDIB-C
ISN 0161       31 CONTINUE
ISN 0162          IJK=0
ISN 0163          F=FDIB
ISN 0164          XF=XFD
ISN 0165          DO 13 I=1,N
ISN 0166          OFLOW(I)=FLOW(I)
ISN 0167          OTEMP(I)=OTEMPB(I)
ISN 0168          OVAP(I)=CVAPOR(I)
ISN 0169          OVAPOR(I)=VAPOR(I)
ISN 0170          OTEMPB(I)=TEMPB(I)
ISN 0171          OXTRAY(I)=XTRAY(I)
ISN 0172          Ox2(I)=x2(I)
ISN 0173       13 CONTINUE
ISN 0174          DO 7 I=1,N
ISN 0175          ABY=OXTRAY(I)
ISN 0176          ABD=Ox2(I)
ISN 0177          ABE=OTEMPB(I)
ISN 0178          CALL BUBPTG(ABY,ABD,ABE,Y3,Y4,ABF,ABH,UX,BPERR)
ISN 0179          OY1(I)=Y3
ISN 0180          OY2(I)=Y4
ISN 0181          OTEMPB(I)=ABE
ISN 0182          OAK1(I)=ABF
ISN 0183          OAK2(I)=ABH
ISN 0184        7 CONTINUE
ISN 0185          DO 17 I=1,N
ISN 0186          OVAA(I)=CVAPOR(I)*CY1(I)
ISN 0187          OVAB(I)=OVAPOR(I)-CVAA(I)
ISN 0188          OHOUPM(I)=OHOUP(I)*CXTRAY(I)
ISN 0189          OHOUPW(I)=OHOUP(I)-OHOUPM(I)
ISN 0190          OABFA(I)=OFLOW(I)/(OAK1(I)*OVAPOR(I))
ISN 0191          OABFB(I)=OFLOW(I)/(OAK2(I)*OVAPOR(I))
ISN 0192          OTAU(I)=OHOUP(I)/(OFLOW(I)*TINCT)
ISN 0193       17 CONTINUE
ISN 0194          IA=0
ISN 0195          MTIME=MTIME+INCT
ISN 0196          OTIME=TOTIME+TINCT
ISN 0197          OABFB(12)=OFLOW(12)/OO
ISN 0198          OABFA(12)=OFLOW(12)/OO
ISN 0199          IF(INN.GT.2) GO TO 38
ISN 0200          DO 26 I=1,N
ISN 0201          TEMPB(I)=OTEMPB(I)+3.00
ISN 0202          VAPOR(I)=OVAPOR(I)+.0500
ISN 0203
```

```
ISN 0204        26 CONTINUE
ISN 0205           GO TO 35
ISN 0206        38 DO 78 I=1,N
ISN 0207           TEMPB(I)=OTEMP(I)+2.DO*TINCT*(OTEMPB(I)-OTEMP(I))/PER
ISN 0208           VAPOR(I)=OVAP(I)+2.DO*TINCT*(OVAPOR(I)-OVAP(I))/PER
ISN 0209        78 CONTINUE
ISN 0210        35 CONTINUE
ISN 0211           DO 70 I=2,N
ISN 0212           A50=0.DO
ISN 0213           DO 71 J=1,N
ISN 0214           A50=A50+HOUP(J)-CHOUP(J)
ISN 0215        71 CONTINUE
ISN 0216           A50=A50/(DMUE*TINCT)
ISN 0217           IF(I.GT.IPF)GO TO 72
ISN 0219           ZF=F
ISN 0220           OZF=OF
ISN 0221           GO TO 7C
ISN 0222        72 ZF=0.DO
ISN 0223           OZF=0.DC
ISN 0224        7C FLOW(I)=VAPOR(I-1)+ZF-D+SIGMA*(OVAPOR(I-1)+OZF-OFLOW(I)-DD)-A50
ISN 0225           FLOW(12)=REFLUX
ISN 0226           IF(IDIST.NE.0)  GO TO 499
ISN 0228           VAPOR(1)=OVAPOR(1)
ISN 0229      499  DO 6 I=1,N
ISN 0230           XTRAY(I)=CXTRAY(I)
ISN 0231           X2(I)=1.DO-XTRAY(I)
ISN 0232         6 CONTINUE
ISN 0233           BB=ZFA(1)-(OABFA(1)+1.DO)*OVAA(1)+OABFA(2)*OVAA(2)
ISN 0234           ABA=ZFB(1)-(OABFB(1)+1.DO)*OVAB(1)+OABFB(2)*OVAB(2)
ISN C235           BC=OTAU(1)*OABFA(1)*OVAA(1)/DMUE
ISN 0236           ACA=OTAU(1)*OABFB(1)*OVAB(1)/DMUE
ISN 0237           COFB(1)=-(SIGMA*BB+BC+ZFDA(1))
ISN 0238           COFFB(1)=-(SIGMA*ABA+ACA+ZFDB(1))
ISN 0239           DO 5 I=2,NN
ISN 0240           BB=ZFA(I)+OVAA(I-1)-(OABFA(I)+1.DO)*OVAA(I)+OABFA(I-1)*OVAA(I-1)
ISN 0241           ABA=ZFB(I)+OVAB(I-1)-(OABFB(I)+1.DO)*OVAB(I)+OABFB(I+1)*OVAB(I+1)
ISN 0242           BC=OTAU(I)*OABFA(I)*OVAA(I)/DMUE
ISN 0243           ACA=OTAU(I)*OABFB(I)*OVAB(I)/DMUE
ISN 0244           COFB(I)=-(SIGMA*BB+BC+ZFDA(I))
ISN 0245           COFFB(I)=-(SIGMA*ABA+ACA+ZFDB(I))
ISN 0246         5 CONTINUE
ISN 0247           BB=ZFA(12)+OVAA(11)-(OABFA(12)+1.DO)*OVAA(12)
ISN 0248           ABA=ZFB(12)+OVAB(11)-(OABFB(12)+1.DO)*OVAB(12)

ISN 0249           BC=OTAU(12)*OABFA(12)*OVAA(12)/DMUE
ISN 0250           ACA=OTAU(12)=OABFB(12)*OVAB(12)/DMUE
ISN 0251           COFB(12)=-(SIGMA*BB+BC+ZFDA(12))
ISN 0252           COFFB(12)=-(SIGMA*ABA+ACA+ZFDB(12))
ISN 0253           CALL VANLATTEMP8,AK1,AK2,0)
ISN 0254        30 CONTINUE
ISN 0255           DO 92 I=1,N
ISN 0256           ZETA(I)=1.00
ISN 0257        92 CONTINUE
ISN 0258           DO 14 I=1,N
ISN 0259           ABFA(I)=FLOW(I)/(AK1(I)*VAPOR(I))
ISN 0260           ABFB(I)=FLOW(I)/(AK2(I)*VAPOR(I))
ISN 0261           TAU(I)=HOLP(I)/(FLOW(I)*TINCT)
ISN 0262        14 CONTINUE
ISN 0263           ABFA(12)=FLOW(12)/D
ISN 0264           ABFB(12)=FLOW(12)/D
ISN 0265           CALL COEFT(ABFA,TAU,DMUE,COFB,COF,GCOF,FCOF,NK,KK)
ISN 0266           CALL COEFT(ABFB,TAU,DMUE,COFFB,COFF,GCOFF,FCOFF,NK,KK)
ISN 0267           VAA(12)=GCOF(12)
ISN 0268           VAB(12)=GCOFF(12)
ISN 0269           DO 66 I=1,NN
ISN 0270           K=N-I
ISN 0271           VAA(K)=GCOF(K)-FCOF(K)*VAA(K+1)
ISN 0272           VAB(K)=GCOFF(K)-FCOFF(K)*VAB(K+1)
ISN 0273        66 CONTINUE
ISN 0274           CM=VAA(12)
ISN 0275           CW=VAB(12)
ISN 0276           OFF=CM+CW
ISN 0277           BM=VAA(1)*ABFA(1)
ISN 0278           BW=VAB(1)*ABFB(1)
ISN 0279           BF=BW+BM
ISN 0280           BDM=BM/CM
ISN 0281           BDW=BW/CW
ISN 0282           DO 50 I=2,NN
ISN 0283           HOUPM(I)=HOUP(I)*ABFA(I)*VAA(I)/FLOW(I)
ISN 0284           HOUPW(I)=HOUP(I)*ABFB(I)*VAB(I)/FLOW(I)
ISN 0285           HOUP(I)=HOUPM(I)+HOUPW(I)
ISN 0286           UDM(I)=HOUPM(I)/CM
ISN 0287           UDW(I)=HOUPW(I)/CW
ISN 0288        50 CONTINUE
ISN 0289           CALL FUNT(ZETA,GFUNT,SIGMA,DMUE,TINCT,TRAD)
ISN 0290        10 CONTINUE
ISN 0291           DO 55 I=2,N
```

```
ISN 0292          ZETAN(I)=ZETA(I)
ISN 0293       55 CONTINUE
ISN 0294          DO 56 I=2,N
ISN 0295          ZETAN(I)=ZETA(I)+PINCT(I)
ISN 0296          CALL FUNT(ZETAN,GFUN,SIGMA,DMUE,TINCT,GRAD)
ISN 0297          DO 57 J=1,NN
ISN 0298          GRAD(J,I-1)=(GFUN(J)-GFUNT(J))/PINCT(I)
ISN 0299       57 CONTINUE
ISN 0300          ZETAN(I)=ZETAN(I)-PINCT(I)
ISN 0301       56 CONTINUE
ISN 0302          CALL MATINV(GRAD,KK,NXK)
ISN 0303          IF(NXK.EQ.1)  GO TO 95
ISN 0305          DO 58 I=1,NN
ISN 0306          GFUNT(I)=-GFUNT(I)
ISN 0307       58 CONTINUE
ISN 0308          CALL AMATM(GRAD,GFUNT,DZETA,KK)
ISN 0309          DO 59 I=2,N
ISN 0310          DZETA(I)=ZETA(I)
ISN 0311          ZETA(I)=ZETA(I)+DZETA(I-1)
ISN 0312          IF(ZETA(I).GT.0.0D0) GO TO 59
ISN 0314          ZETA(I)=0.5D0*DZETA(I)
ISN 0315       59 CONTINUE
ISN 0316          IY=1
ISN 0317          CALL FUNT(ZETA,GFUNT,SIGMA,DMUE,TINCT,GRAD)
ISN 0318       52 IF(GFUNT(IY).GT.1.D-13) GO TO 10
ISN 0320          IY=IY+1
ISN 0321          IF(IY.LE.NN) GO TO 52
ISN 0323          BMC=ZETA(12)*BDM*DMCO
ISN 0324          BWC=ZETA(IZ)*BDW*DWCO
ISN 0325          DFF=DMCC+DWCO
ISN 0326          BF=BMC+BWC
ISN 0327          FLOW(1)=BF
ISN 0328          VAPOR(12)=DFF
ISN 0329          XTRAY(1)=BMC/BF
ISN 0330          X2(1)=BWC/BF
ISN 0331          XTRAY(12)=DMCO/DFF
ISN 0332          X2(12)=DWCO/DFF
ISN 0333          DO 74 I=2,NN
ISN 0334          HOUPM(I)=ZETA(I)*UCM(I)*DMCO
ISN 0335          HOUPW(I)=ZETA(I)*UCW(I)*DWCO
ISN 0336          HOUP(I)=HOUPW(I)+HCUPM(I)
ISN 0337          XTRAY(I)=HCUPM(I)/HCUP(I)
ISN 0338          X2(I)=HCUPW(I)/HCUP(I)

ISN 0339       74 CONTINUE
ISN 0340          HOUPW(1)=ZETA(12)*( HCUP(1)/B)*BDW*DWCO
ISN 0341          HOUPM(1)=ZETA(12)*( HCUB(1)/B)*BDM*DMCO.
ISN 0342          HOUP(1)=HCUPW(1)+HCUPM(1)
ISN 0343          XTOPZ=X2(12)
ISN 0344          XTOP=XTRAY(12)
ISN 0345          IF(MTIME.GT.10)  GO TO 85
ISN 0347          XTRAY(12)=XS1
ISN 0348          X2(12)=XS12
ISN 0349          GO TO 86
ISN 0350       85 XTRAY(12)=XLAG(MTIME-10)
ISN 0351          X2(12)=XLAG2(MTIME-10)
ISN 0352       86 DO 62 I=1,N
ISN 0353          ABY=XTRAY(I)
ISN 0354          ABD=X2(I)
ISN 0355          ABE=TEMPB(I)
ISN 0356          CALL BUBPTG(ABY,ABD,ABE,Y3,Y4,ABF,ABH,LK,BPERR)
ISN 0357          Y1(I)=Y3
ISN 0358          Y2(I)=Y4
ISN 0359          AK1(I)=ABF
ISN 0360          AK2(I)=ABH
ISN 0361          TEMPB(I)=ABE
ISN 0362       62 CONTINUE
ISN 0363          CALL ENCON(TEMPB,HV,HL)
ISN 0364          IA=IA+1
ISN 0365          IF(MTIME.GT.400)  GO TO 34
ISN 0367          IF(IA.GT.12)  GO TO 51
ISN 0369       65 IF(IA.EQ.1)  GO TO 73
ISN 0371          IJK=1
ISN 0372       73 CALL ETCMB(DMUE,TINCT,SIGMA,HV,HL,TF,TEMPB,IFF,IJK,IDIST,0)
ISN 0373          GO TO 30
ISN 0374       51 CONTINUE
ISN 0375          CALL ETCMU(DMUE,TINCT,SIGMA,HV,HL,TF,TEMPB,IFF,IJK,IDIST,1)
ISN 0376          IF(MTIME.LE.30)  GO TO 28
ISN 0378          IF(MTIME.GT.120)  GO TO 48
ISN 0380          IF(MTIME-5*(MTIME/5))27,28,27
ISN 0381       48 IF(MTIME-10*(MTIME/10))27,28,27
ISN 0382       28 CONTINUE
ISN 0383          WRITE(3,9)TOTIME
ISN 0384          WRITE(3,173)  D,B
ISN 0385          WRITE(3,9)(CTEMPB(I),I=1,N)
ISN 0386          WRITE(3,46)(CQ(I),I=1,N)
ISN 0387          WRITE(3,18)
```

```
ISN 0388          DO 15 I=1,N
ISN 0389          XTOIF(I)=XTRAY(I)+XTRAY(I)
ISN 0390          WRITE(3,16)(XTRAY(I),TEMPB(I),AK1(I),AK2(I),FLOW(I),VAPOR(I),
                  1HL(I),HV(I),HCUP(I),ZETA(I))
ISN 0391       15 CONTINUE
ISN 0392          WRITE(3,54)(XTOIF(I),I=1,N)
ISN 0393          WRITE(3,53)(GFUN(I),I=1,NN)
ISN 0394       53 FORMAT(1H0,2X,'GFUN=',/(8D14.8))
ISN 0395       27 CONTINUE
ISN 0396          III=1
ISN 0397       37 IF(DABS(XTRAY(III)-OXTRAY(III)).GT.1.D-6) GO TO 32
ISN 0399          IF(III-N)33,81,81
ISN 0400       33 III=III+1
ISN 0401          GO TO 37
ISN 0402       32 ZFA(5)=ZFDA(5)
ISN 0403          ZFB(5)=ZFDB(5)
ISN 0404          OF=FDIB
ISN 0405          OXF=XFD
ISN 0406          OFLOW(12)=BFLOX
ISN 0407          DD=D
ISN 0408          OB=B
ISN 0409          XLAG(MTIME)=XTOP
ISN 0410          XLAG2(MTIME)=XTOP2
ISN 0411       39 CONTINUE
ISN 0412          GO TO 31
ISN 0413       34 CONTINUE
ISN 0414       96 WRITE(3,97)(PINCT(I),I=1,N)
ISN 0415       97 FORMAT(1H0,2X,'PINCT(I)= '/(8D12.6))
ISN 0416       81 DO 95 I=1,N
ISN 0417          WRITE(3,16)(XTRAY(I),TEMPB(I),AK1(I),AK2(I),FLOW(I),VAPOR(I),
                  1HL(I),HV(I),HCUP(I),ZETA(I))
ISN 0418       95 CONTINUE
ISN 0419          RETURN
ISN 0420          END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,I-O,NOXRI
ISN 0002          SUBROUTINE VANLA(T,AK1,AK2,K1)
                  C
                  C     SUBROUTINE VANLA
ISN 0003          IMPLICIT REAL*8(A-H,O-Z)
ISN 0004          REAL*8 PPA(12),PPB(12),PA(12),PB(12),AKA(12),AKB(12),X(12)
ISN 0005          REAL*8 X2(12),GAMAA(12),GAMAB(12),GAMMAA(12),GAMMAB(12),Y(12)
ISN 0006          REAL*8 AK1(12),AK2(12),T(12),Y2(12) ,C(12),C2(12)                    00005100
ISN 0007          REAL*8 OXTRAY(12),OX2(12),OY1(12),OY2(12)
ISN 0008          COMMON /NO/ N,NN,L
ISN 0009          COMMON /COMP/X,X2,Y,Y2
ISN 0010          COMMON /OLDCOM/OXTRAY,OX2,OY1,OY2
ISN 0011          DATA A12/.362500/,A21/.241800/
ISN 0012          IF(K1)    1,2,1
ISN 0013       1  DO 3 I=1,N
ISN 0014          C(I)=OXTRAY(I)
ISN 0015          C2(I)=OX2(I)
ISN 0016       3  CONTINUE
ISN 0017          GO TO 4
ISN 0018       2  DO 5 I=1,N
ISN 0019          C2(I)=X2(I)
ISN 0020          C(I)=X(I)
ISN 0021       5  CONTINUE
ISN 0022       4  DO 10 I=1,N
ISN 0023          T(I)=5.DO*(T(I)-32.DO)/9.DO
ISN 0024          PPA(I)=8.0726800-1574.7900/(T(I)+238.8600)
ISN 0025          PPB(I)=7.9668100-1668.2100/(T(I)+228.DO)
ISN 0026          PA(I)=DEXP(PPA(I)*2.3025900)
ISN 0027          PB(I)=DEXP(PPB(I)*2.3025900)
ISN 0028          AKA(I)=PA(I)/760.DO
ISN 0029          AKB(I)=PB(I)/760.DO
ISN 0030          GAMAA(I)=A12*C2(I)*C2(I)/((C(I)+A12/A21*C2(I))**2)
ISN 0031          GAMAB(I)=A21*C(I)*C(I)/((C2(I)*A21/A12+C(I))**2)
ISN 0032          GAMMAA(I)=DEXP(GAMAA(I)*2.3025900)
ISN 0033          GAMMAB(I)=DEXP(GAMAB(I)*2.3025900)
ISN 0034          AK1(I)=AKA(I)*GAMMAA(I)
ISN 0035          AK2(I)=AKB(I)*GAMMAB(I)
ISN 0036       10 CONTINUE
ISN 0037          DO 11 I=1,N
ISN 0038       11 T(I)=1.8DO*T(I)+32.DO
ISN 0039          RETURN
ISN 0040          END
```

```
.EVEL 18 ( SEPT.69 )          OS/360  FORTRAN H                    DATE  70.344/2

        COMPILER OPTIONS - NAME= MAIN,OPT=02,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXRE
ISN 0002            SUBROUTINE BUBPT(X1,X3,T,Y1,Y2,ABF,ABM,L,BPERR)
                    C
                    C SUBROUTINE FOR NON IDEAL BUBBLE POINTS
                    C       SUBROUTINE FOR NON-IDEAL BUBBLE POINTS CALCULATION
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
ISN 0004            REAL*8 CUIDX(2),GAMMA(2),AKA(2),PPA(2),PA(2)
ISN 0005            DATA A12/.3625D0/,A21/.2418D0/
ISN 0006            GAMAA=A12*X3*X3/((X1*A12/A21+X3)**2)
ISN 0007            GAMAB=A21*X1*X1/((X3*A21/A12+X1)**2)
ISN 0008            GAMMAA=DEXP(GAMAA-2.30259D0)
ISN 0009            GAMMAB=DEXP(GAMAB-2.30259D0)
ISN 0010            T=5.D0*(T-32.D0)/9.D0
ISN 0011            GAMMA(1)=GAMMAA
ISN 0012            GAMMA(2)=GAMMAB
ISN 0013            KTIMES=1
ISN 0014            CUIDX(1)=X1
ISN 0015            CUIDX(2)=X3
ISN 0016          3 SUMY=0.D0
ISN 0017            PPA(1)=8.07246D3-1574.99D0/(T+238.86D0)
ISN 0018            PPA(2)=7.96681D0-1668.21D0/(T+228.D0)
ISN 0019            DO 4 I=1,L
ISN 0020            PA(I)=DEXP(PPA(I)-2.30259D0)
ISN 0021            AKA(I)=PA(I)/760.D0
ISN 0022            Y=AKA(I)*GAMMA(I)*CUIDX(I)
ISN 0023          4 SUMY=SUMY+Y
ISN 0024            IF(DABS(SUMY-1.D0).LE.BPERR) GO TO 8
ISN 0026            KTIMES = KTIMES -1
ISN 0027            IF (KTIMES.LT.0)   GO TO  7
ISN 0028            SUMYO = SUMY
ISN 0030            TO=T
ISN 0031            T=T+.1D0
ISN 0032            GO TO 3
ISN 0033          7 SLOPE=(SUMY-SUMYO)/(T-   TO)
ISN 0034            TN=(1.D0-SUMY)/SLOPE+T
ISN 0035            SUMYO=SUMY
ISN 0036            TO=T
ISN 0037            T=TN
ISN 0038            GO TO 3
ISN 0039          8 T=1.8D0*T+32.D0
ISN 0040            Y1=AKA(1)*X1*GAMMA(1)
ISN 0041            Y2=AKA(2)*X3*GAMMA(2)

ISN 0042            ABF=AKA(1)*GAMMA(1)
ISN 0043            ABM=AKA(2)*GAMMA(2)
ISN 0044            RETURN
ISN 0045            END
```

```
LEVEL 18 ( SEPT 69 )          OS/360  FORTRAN H                    DATE  70.344/2

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXRE
ISN 0002                SUBROUTINE AMATM(A,B6,R6,N)
                      C SUBROUTINE AMATM  (MATRIX MULTIPLICATION)
ISN 0003                IMPLICIT REAL*8(A-H,O-Z)
ISN 0004                REAL*8 A6(11,11),B6(11),R6(11)
ISN 0005                DO 310 I=1,N
ISN 0006            310 R6(I)=0.0
ISN 0007                DO 311 I=1,N
ISN 0008                DO 311 J=1,N
ISN 0009            311 R6(I)=R6(I)+A6(I,J)*B6(J)
ISN 0010                RETURN
ISN 0011                END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREI
                      C
ISN 0002                SUBROUTINE ENCON(TEMPB,HV,HL)
                      C
                      C SUBROUTINE ENCON
ISN 0003                IMPLICIT REAL*8(A-H,O-Z)
ISN 0004                REAL*8 CHV(12),CHL(12),CTEMPB(12),OXW(12),OXM(12),OYM(12),OYW(12)
ISN 0005                REAL*8 TEMPB(12),HV(12),HL(12),HLW(12),HLM(12),HVW(12)
ISN 0006                REAL*8 HVM(12),XM(12),XW(12),YM(12),YW(12)
ISN 0007                REAL*8  HVMM(12),HVWW(12),HLMM(12),HLWW(12)
ISN 0008                COMMON /VLWM/HVW,HLW,HVM,HLM
ISN 0009                COMMON /CLC/ CHL,CHV,CTEMPB
ISN 0010                COMMON /NC/ N,NN,L
ISN 0011                COMMON /CLOCCN/ CXM,OXW,CYM,OYW
ISN 0012                COMMON /COMP/ XM,XW,YM,YW
ISN 0013                DO 1 I=1,N
ISN 0014                HLW(I)=(-586.6200+18.054D0*CTEMPB(I))*CXW(I)
ISN 0015                HLMM(I)=(-850.2300+20.96D0*CTEMPB(I))*OXM(I)
ISN 0016                HVW(I)=(19212.250D0+7.056D0*CTEMPB(I))*OYW(I)
ISN 0017                HVMM(I)=(17164.410D0+1.62D0*CTEMPB(I))*OYM(I)
ISN 0018                CHL(I)=HLW(I)+HLMM(I)
ISN 0019                CHV(I)=HVWW(I)+HVMM(I)
ISN 0020              1 CONTINUE
ISN 0021                DO 2 I=1,N
ISN 0022                HLW(I)=-586.6200+18.054D0*TEMPB(I)
ISN 0023                HLWW(I)=HLW(I)*XW(I)
ISN 0024                HLM(I)=-850.2300+20.96D0*TEMPB(I)
ISN 0025                HLMM(I)=HLM(I)*XM(I)
ISN 0026                HVW(I)=19212.250D0+7.056D0*TEMPB(I)
ISN 0027                HVWW(I)=HVW(I)*YW(I)
ISN 0028                HVM(I)=17164.410D0+1.62D0*TEMPB(I)
ISN 0029                HVMM(I)=HVM(I)*YM(I)
ISN 0030                HV(I)=HVWW(I)+HVMM(I)
ISN 0031                HL(I)=HLMM(I)+HLWW(I)
ISN 0032              2 CONTINUE
ISN 0033                RETURN
ISN 0034                END
```

```
EVEL 18 ( SEPT 69 )          OS/360  FORTRAN H                    DATE  70.344/2.

     COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002            SUBROUTINE MATINV(A,N,NXK)
                 C  SUBROUTINE MATINV   (MATRIX INVERSION)
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
ISN 0004            REAL*8 A(11,11)
ISN 0005            INTEGER*4 INDEX(11,2)
ISN 0006            DO 108 I=1,N
ISN 0007        108 INDEX(I,1)=0
ISN 0008            II=0
ISN 0009        109 AMAX=-1.DO
ISN 0010            DO 110 I=1,N
ISN 0011            IF(INDEX(I,1))110,111,110
ISN 0012        111 DO 112 J=1,N
ISN 0013            IF(INDEX(J,1))112,113,112
ISN 0014        113 TEMP=DABS(A(I,J))
ISN 0015            IF(TEMP-AMAX)112,112,114
ISN 0016        114 IROW=I
ISN 0017            ICOL=J
ISN 0018            AMAX=TEMP
ISN 0019        112 CONTINUE
ISN 0020        110 CONTINUE
ISN 0021            IF(AMAX)225,115,116
ISN 0022        116 INDEX(ICOL,1)=IROW
ISN 0023            IF(IROW-ICOL)119,118,119
ISN 0024        119 DO 120 J=1,N
ISN 0025            TEMP=A(IROW,J)
ISN 0026            A(IROW,J)=A(ICOL,J)
ISN 0027        120 A(ICOL,J)=TEMP
ISN 0028            II=II+1
ISN 0029            INDEX(II,2)=ICOL
ISN 0030        118 PIVOT=A(ICOL,ICOL)
ISN 0031            A(ICOL,ICOL)=1.DO
ISN 0032            PIVOT=1.DO/PIVOT
ISN 0033            DO 121 J=1,N
ISN 0034        121 A(ICOL,J)=A(ICOL,J)*PIVOT
ISN 0035            DO 122 I=1,N
ISN 0036            IF(I-ICOL)123,122,123
ISN 0037        123 TEMP=A(I,ICOL)
ISN 0038            A(I,ICOL)=0.
ISN 0039            DO 124 J=1,N
ISN 0040        124 A(I,J)=A(I,J)-A(ICOL,J)*TEMP
ISN 0041        122 CONTINUE
ISN 0042            GO TO 109
ISN 0043        125 ICOL=INDEX(II,2)
ISN 0044            DO 126 I=1,N
ISN 0045            IROW=INDEX(ICOL,1)
ISN 0046            TEMP=A(I,IROW)
ISN 0047            A(I,IROW)=A(I,ICOL)
ISN 0048        126 A(I,ICOL)=TEMP
ISN 0049            II=II-1
ISN 0050        225 IF(II)125,127,125
ISN 0051        115 WRITE(6,1001)
ISN 0052            NXK=1
ISN 0053       1001 FORMAT(1HC,2X,11H ZERO PIVOT)
ISN 0054        127 RETURN
ISN 0055            END
```

```
ISN 0040            FLOW(I)=DFL*W(I)
ISN 0041         27 CONTINUE
ISN 0042         37 CONTINUE
ISN 0043            MAB=1
ISN 0044            HB=HL(1)
ISN 0045            HD=HL(12)
ISN 0046            DHB=DHL(1)
ISN 0047            DHD=DHL(12)
ISN 0048            IF(IJK)1,2,1
ISN 0049          2 DO 26 I=2,NN
ISN 0050            DQQ(I)=CQ(I)
ISN 0051         26 CONTINUE
ISN 0052            DQC=AQC
ISN 0053            IF(QTIME.EQ.1)  GO TO 1
ISN 0055            DQR=QR
ISN 0056          1 QXG=1.0D-QXF
ISN 0057            AG=1.0D-XF
ISN 0058            A11=FLOW(12)*(HVW(11)*X2(12)+HVM(11)*XTRAY(12)-HL(12))
ISN 0059            A121=HVW(11)*X2(12)+HVM(11)*XTRAY(12)-HD
ISN 0060            A12=B=A121
ISN 0061            A13=DFLOW(12)*(HVW(11)*DX2(12)+HVM(11)*DXTRAY(12)-DHL(12))
ISN 0062            A14=CQW*(HVW(11)*DX2(12)+HVM(11)*DXTRAY(12)-DHD)
ISN 0063            A15=EVAPOR(11)*(HVW(11)*DY2(11)+HVM(11)*DYTRAY(11)-DHV(11))
ISN 0064            A16=HOUP(12)*(HVW(11)*X2(12)+HVM(11)*XTRAY(12)-HL(12))/
                   1(DMUE*TINCT)
ISN 0065            A17=CHOUP(12)*(HVW(11)*DX2(12)+HVM(11)*DXTRAY(12)-DHL(12))/
                   1(DMUE*TINCT)
ISN 0066            A122=A11+SIGMA*(A13+A14-A15-A16)*2.0D-A17
ISN 0067            DC=A122*S12
ISN 0068            AQC=QC
ISN 0069            IF((IDIST.EQ.5).OR.(IDIST.EQ.6))  GO TO 100
ISN 0071            A=2.0D-0
ISN 0072            SA=0.0D
ISN 0073            DO 12 I=1,N
ISN 0074            A3=A3+HMOP(I)*HL(I)-CHMOP(I)*DHL(I)
ISN 0075            A=2.0D+A-HMOP(I)-CHMOP(I)
ISN 0076         12 CONTINUE
ISN 0077            A3=A3*A/(DMUE*TINCT)
ISN 0078            A=2.0D-A/(DMUE*TINCT)
ISN 0079            A42=SIGMA*(DF-QD-TB)-A42
ISN 0080            DV1=QR-F-HB-A-A3-HD*F-HF-A122+SIGMA*(DF*DHF+DQR-HD*DHD-DB*DHB-
                   1DQC)-A3
ISN 0081            DQ1=HL*A121-HB

      COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SIZE=0000K,
                        SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF
ISN 0002            SUBROUTINE ETCMB(DMUE,TINCT,SIGMA,HV,HL,TF,TEMPB,IFF,IJK,IDIST,
                   1IIK)
                C   SUBROUTINE ETCMB
                C   SUBROUTINE FOR TOTAL ENTHALPY AND MATERIAL BALANCE
                C   THE SUBROUTINE CALCULATES THE SET OF NEW FLOW RATES
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
ISN 0004            REAL*8 FLOW(12)
ISN 0005            REAL*8 FLMIN(12),FLMAX(12),QQ(12),CQQ(12)
ISN 0006            REAL*8 VAPOR(11),FLOW(12),HOUP(12),AYTRAY(12),X2(12),Y1(12)
ISN 0007            REAL*8 Y2(12),HV(12),HL(12),TEMPB(12),DTEMPB(12)
ISN 0008            REAL*8 EVAPOR(12),DFLOW(12),CHOUP(12)
ISN 0009            REAL*8 XY1(12),XY2(12),X1(12),AXTRAY(12),CHOUP(12)
ISN 0010            REAL*8 CHV(12),CHL(12),HVW(12),HVM(12),HLW(12),HLM(12)
ISN 0011            COMMON /ULL/ DHL,DHV,DTEMPB
ISN 0012            COMMON /XF/ XF,F,L
ISN 0013            COMMON /RATE/ VAPOR,FLOW,QQ
ISN 0014            COMMON /VAP/ DVAPOR,DFLOW
ISN 0015            COMMON /HP/ HOUP,CHOUP,HLW,HLM
ISN 0016            COMMON /ULCON/ XTRAY,X1,XY1,XY2
ISN 0017            COMMON  /COMP/ X2,HV,XL,Y1,Y2
ISN 0018            COMMON  /ULIS/ DQQ,DFLOW,X2
ISN 0019            COMMON /FXFD/XF,F,D
ISN 0020            COMMON /VLM/HVW,HL,HVM,HLM
ISN 0021            COMMON /VLCON/XTRAY,DXTRAY
ISN 0022            COMMON /MATB/ MTI,MAB
ISN 0023            DIMENSION
ISN 0024            REAL*8
ISN 0025            DELTA=0.0D
ISN 0026            IF(MAB.EQ.0)  GO TO 37
ISN 0028            DO 51 I=1,12
ISN 0029            DQ(I)=0.0D
ISN 0030            DO 27 I=1,N
ISN 0031            DF=DF+FLOW(I)-HV(I)-DFLOW(I)*HL(I)-QQ*HL(I)
ISN 0032            AQC=(DMUE*TINCT*SIGMA*DF)/(1.0D+SIGMA*(HLW(12)*DFLOW(12)-QQ*HL(12)
ISN 0033            IJK=0
ISN 0035            DO 27 I=1,N
ISN 0036            DQ(I)=AQC
ISN 0037            DQ(I)=DQ(I)*HL(I)/(HV(I)-HL(I))
ISN 0038            FLOW(I)=FLOW(I)+DQ(I)
```

```
ISN 0082          DE=DX*(1./(...
ISN 0083          SF=DE*...
ISN 0084      100 CONTINUE
ISN 0085          A35=0.00
ISN 0086          A36=0.00
ISN 0087          DO 5 K=1,NJ
ISN 0088          I=N-K
ISN 0089          A1=HVW(I-1)+A2(I)+HVW(I-1)*XT-AY(I)-HL(I)
ISN 0090          A21=1.00/A19
ISN 0091          A22=DX(HVW(I-1)+X(I))+HVW(I-1)*XT-AY(I))-HO)
ISN 0092          A23=DVAPL*(I-1)*(HVW(I-1)-DY(I-1)+HVW(I-1)*DY(I-1)-DHV(I-1))
ISN 0093          A24=DFLOW(I)*(HVW(I-1)+X2(I)+HVW(I-1)*DXT-AY(I)-DHL(I))
ISN 0094          A25=DL*(HVW(I-1)+X2(I))+HVW(I-1)*DXT-AY(I))-DHO)
ISN 0095          A26=0.00
ISN 0096          A27=0.00
ISN 0097          DO 6 J=1,N
ISN 0098          A26=A26+(HVW(I-1)-X2(J)+HVW(I-1)*XT-AY(J)-HL(J))
ISN 0099          A27=A27+(HOUP(J)*(HVW(I-1)-AX(J)+HVW(I-1)*XT-AY(J)-HL(J))
ISN 0100      6 CONTINUE
ISN 0101          A28=(A27-A26)/(DPUL*FNCT)
ISN 0102          IF(I.GT.IFF) GO TO 5
ISN 0103          A29=F*(HVW(I-1)+X...+HVW(I-1)*XF-HF)
ISN 0104          A30=DF*(HVW(I-1)+XO+HVW(I-1)*DXF-DHF)
ISN 0105          GO TO 6
ISN 0106      5 A29=0.00
ISN 0107          A30=0.00
ISN 0108      6 FLOW(I)=A21*(-A22+GC+SIGMA*(A23-A24-A25+DGC)+A28+A29+SIGMA*
                 1A30)+A36
ISN 0109          IF(MAZ.NE.0) GO TO 7
ISN 0110          RFLOW=FLOW(I)/(1.00+DELTA)
ISN 0111          IF((FLOW(I).GT.RFLOW).AND.(FLOW(I).GT.FLMIN(I))) GO TO 24
ISN 0112          IF((FLOW(I).LT.RFLOW).AND.(FLOW(I).GT.FLMIN(I))) GO TO 21
ISN 0113          IF((FLOW(I).LT.RFLOW).AND.(RFLOW.LT.FLMIN(I))) GO TO 21
ISN 0114      24  FLOWM=FLOW(I)*(1.00+DELTA)
ISN 0115          IF((FLOW(I).LT.FLOWM).AND.(FLOWM.LT.FLMAX(I))) GO TO 22
ISN 0116          IF((FLOW(I).LT.FLOWM).AND.(FLOWM.LE.FLMAX(I))) GO TO 22
ISN 0117          FLOW(I)=FLMAX(I)
ISN 0118          GO TO 23
ISN 0119      22  FLOW(I)=FLOWM
ISN 0120          GO TO 23
ISN 0121      20  FLOW(I)=FFLOW
ISN 0122          GO TO 23
ISN 0123      21  FLOW(I)=FLMIN(I)
```

```
ISN 0124      23  GO(I)=FLOW(I)*(1.0-GC+A22-SIGMA*(A23-A24-A25+DGC)-
                 1A26-SIGMA*A30-A28-A35
ISN 0125          A35=0.00
ISN 0126          A36=0.00
ISN 0127          A25=0.00
ISN 0128          DO 25 J=1,N
ISN 0129          A35=A35+HOUP(J)*SIGMA*DGC(J)
ISN 0130          A25=GC(J)*SIGMA*DGC(J)
ISN 0131          A36=A36+A25*A21
ISN 0132      25  CONTINUE
ISN 0133          GO TO 4
ISN 0134      31  GO(I)=0.00
ISN 0135      4   CONTINUE
ISN 0136          DO 6 J=1,N
ISN 0137          A38=0.00
ISN 0138          DO 31 J=1,N
ISN 0139          A38=A38+HOUP(J)-DHOUP(J)
ISN 0140      31  CONTINUE
ISN 0141          A7=A38/(HOUP(1)*TIMCT)
ISN 0142          IF(I.GT.IFF) GO TO 7
ISN 0143          ZF=F
ISN 0144          DZF=DF
ISN 0145          GO TO 151
ISN 0146      7   ZF=0.00
ISN 0147          DZF=0.00
ISN 0148      151 CONTINUE
ISN 0149          B=VAPL*(I-1)*FLOW(I)*D-ZF-SIGMA*(HVAP(I-1)*GO-FLOW(I)-GO*DZF)*A33
ISN 0150          A39=0.00
ISN 0151          A41=0.00
ISN 0152          DO 39 J=1,NC
ISN 0153          A41=A41+A(J,I)*GO(J)
ISN 0154          A39=A39+A(J,I)*DGO(J)
ISN 0155      39  CONTINUE
ISN 0156          FLOW(I)=FLOW(I)
ISN 0157          DO 40 J=1,NC
ISN 0158          A(J,I)=A(J,I)+DA(J,I)
ISN 0159      40  CONTINUE
ISN 0160          B=B
ISN 0161          GO TO 5
```

```
EVEL 18 ( SEPT 69 )                    OS/360  FORTRAN H                    DATE  70.344/22

      COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002            SUBROUTINE FUNT(ZETA,GFUNT,SIGMA,DMUE,TINCT,GRAD)
                    C
                    C     SUBROUTINE FUNT
                    C     THIS SUBROUTINE IS FOR CALCULATING THE G-FUNCTION
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
ISN 0004            REAL*8 GFUNT(12),UDM(12),UDW(12),ZETA(12),HOUP(12)
ISN 0005            REAL*8 CHCUPT(12),HCUPW(12),HCUPM(12),DHOUPW(12),DHOUPM(12)
ISN 0006            REAL*8 CY1(12),CY2(12),CXTRAY(12),CX2(12),FHOUP(12)
ISN 0007            REAL*8 GRAD(11,11)
ISN 0008            COMMON /DWM/ DWCC,DMCC,UDW,UDM,BDW,BDM
ISN 0009            COMMON /NC/N,NN,L
ISN 0010            COMMON /CLOS/ CC,CB,CF,CXF
ISN 0011            COMMON /HP/ HCUP,CHCUP,FHCUP
ISN 0012            COMMON /FXFD/XF,F,D,B
ISN 0013            COMMON /CLDCOM/CXTRAY,CX2,CY1,CY2
ISN 0014            ABW=0.DO
ISN 0015            ABM=0.DO
ISN 0016            ACW=0.DO
ISN 0017            ACM=0.DO
ISN 0018            XG=1.DO-XF
ISN 0019            CXG=1.DO-CXF
ISN 0020            DMUET=1.DO/(DMUE*TINCT)
ISN 0021            DO 1 I=1,N
ISN 0022            DHOUPM(I)=CHOUP(I)*CXTRAY(I)
ISN 0023            DHCUPW(I)=CHCUP(I)*CX2(I)
ISN 0024            ABM=ABM+DHCUPM(I)
ISN 0025            ABW=ABW+DHCUPW(I)
ISN 0026          1 CONTINUE
ISN 0027            DO 2 I=2,NN
ISN 0028            ACM=ACM+ZETA(I)*UDM(I)
ISN 0029            ACW=ACW+ZETA(I)*UDW(I)
ISN 0030          2 CONTINUE
ISN 0031            UDM=CXTRAY(12)*DC
ISN 0032            UDW=DD-CDM
ISN 0033            BBM=CXTRAY(1)*CB
ISN 0034            BBW=CB-CBM
ISN 0035            DWCCN=F*XF+SIGMA*(CF*CXF-CDP-CBM)+DMUET*ABM
ISN 0036            DMCDD=1.DO+ZETA(12)*BDM*(1.DO+HOUP(1)*DMUET/B)+DMUET*ACM+
                   1DMUET*(FCUP(12)/C)
ISN 0037            DWCCN=F*XG+SIGMA*(CF*CXG-CDW-CBWT+DMUET*ABW
ISN 0038            DWCDD=1.DO+ZETA(12)*BDW*(1.DO+HOUP(1)*DMUET/B)+DMUET*ACW+

                    IDMUET=(HCUP(12)/D).
ISN 0039            DMCC=DMCCN/DMCCD
ISN 0040            DWCC=DWCCN/DWCCD
ISN 0041          5 GFUNT(11)=DMCC+DWCC-D
ISN 0042            DO 3 I=2,NN
ISN 0043            HOUPM(I)=ZETA(I)*UDM(I)*DMCC
ISN 0044            HOUPW(I)=ZETA(I)*UDW(I)*DWCC
ISN 0045            GFUNT(I-1)=HOUPW(I)+HCUPM(I)-FHOUP(I)
ISN 0046          3 CONTINUE
ISN 0047          4 RETURN
ISN 0048            END
```

```
EVEL 18 ( SEPT 69 )          OS/360  FORTRAN H                    DATE  70.344/2
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002            SUBROUTINE CCEFT(ABFA,TAU,DMUE,COF8,COF,GCOF,FCOF,N,NN)
                    C
                    C     SUBROUTINE CCEFT
                    C
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
ISN 0004            REAL*8 BCOF(12),ACOF(12),CCOF(12),CCOF(12),COF8(12)
ISN 0005            REAL*8 ABFA(12),TAU(12),COF(12,12),GCOF(12),FCOF(12)
ISN 0006            DO 2 I=1,N
ISN 0007            BCOF(I)=0.D0
ISN 0008            DCOF(I)=0.D0
ISN 0009            ACOF(I)=0.D0
ISN 0010            CCOF(I)=0.D0
ISN 0011            FCOF(I)=0.D0
ISN 0012            GCOF(I)=0.D0
ISN 0013          2 CONTINUE
ISN 0014            DO 1 I=1,N
ISN 0015            DO 1 J=1,N
ISN 0016            COF(I,J)=0.D0
ISN 0017          1 CONTINUE
ISN 0018            DO 6 I=1,N
ISN 0019            COF(I,I)=-(1.D0+ABFA(I)*(1.D0+TAU(I)/DMUE))
ISN 0020          6 CONTINUE
ISN 0021            DO 7 I=2,N
ISN 0022            COF(I,I-1)=1.D0
ISN 0023          7 CONTINUE
ISN 0024            DO 8 I=1,NN
ISN 0025            COF(I,I+1)=ABFA(I+1)
ISN 0026          8 CONTINUE
ISN 0027            DO 40 I=1,N
ISN 0028            BCOF(I)=CCF(I,I)
ISN 0029            DCOF(I)=COF8(I)
ISN 0030         40 CONTINUE
ISN 0031            DO 41 I=2,N
ISN 0032            ACOF(I)=1.D0
ISN 0033         41 CONTINUE
ISN 0034            DO 42 I=1,NN
ISN 0035            CCOF(I)=CCF(I,I+1)
ISN 0036         42 CONTINUE
ISN 0037            FCOF(1)=CCCF(1)/BCCF(1)
ISN 0038            DO 43 I=2,NN
ISN 0039            FCOF(I)=CCCF(I)/(BCOF(I)-ACOF(I)*FCOF(I-1))

ISN 0040         43 CONTINUE
ISN 0041            GCOF(1)=DCCF(1)/BCCF(1)
ISN 0042            DO 44 I=2,N
ISN 0043            GCOF(I)=(CCOF(I)-ACOF(I)*GCOF(I-1))/(BCOF(I)-ACOF(I)*FCOF(I-1))
ISN 0044         44 CONTINUE
ISN 0045            RETURN
ISN 0046            END
```

COMPUTER PROGRAM FOR OPTIMAL CONTROL CALCULATION

```
(VCL 21.7  4 JAN 73 )          OS/360  FORTRAN H,                    DATE  73.271/1

      COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SIZE=0000K,
                         SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                C     MAIN PROGRAM                                          00000010
                C     SEQUENTIAL DISTURBANCE
                C     OPTIMAL TWO-POINT CONTROL POLICY FOR MEOH-WATER BINARY DIST.   00000020
                C     THE SYSTEM--LINEAR AND STATIONARY                     00000030
                C     THE PERFORMANCE INDEX--QUADRATIC                      00000040
                C     THE CONTROL------FEEDBACK                             00000050
                C     THE TECHNIQUE USE IN SOLVING THIS PROBLEM IS BASED ON THE      00000060
                C     MINIMUM PRINCIPLE AND A MODIFIED RICCATI TRANSFORMATION        00000070
                C     THE VARIABLES UNDER CONTROLLED--XD AND XB             00000080
                C     THE MANIPULATED VARIABLES--D AND V(1)                 00000090
                C     THE CONTROL AND MANIPULATED VARIABLES ARE DEVIATION VARIABLES  00000100
                C     THE SYSTEM OF DIFFERENTIAL EQUATIONS IS SOLVED BY RUNGE-KUTTA
                C     FOURTH-ORDER METHOD
                C     THE TWO CO-STATE VARIABLES ZC ARE LOCATED BY THE PATTERN       00000130
                C     SEARCH TECHNIQUE                                      00000140
                C     M1=1    FEED DISTURBANCE
                C     M1=2 CHANGE IN OPERATING CONDITION  F(I)=0.0    FF=0.0
                C     STATE VARIABLES X=X(INST)-X(DISIRED)
                C     U=U(INST)-U(DISIRED)
                C     NT-----TERMINAL TIME                                  00000150
                C     M------NUMBER OF MANIPULATED VARIABLES                 00000160
ISN 0002              IMPLICIT REAL*8(A-H,O-Z)                              00000170
ISN 0003              REAL MESAGE(80)
ISN 0004              REAL*8 A1(12,12),AT1(12,12),B1(12,2),BT1(2,12)
ISN 0005              REAL*8 U(2),HOUP(12),AZE1(12),AZE2(12),F(12),AT(12,12),BT(2,12)   00000220
ISN 0006              REAL*8 CQ(12,12),AR(2,2),A(12,12),B(12,2),SS(12,12),RR1(2,2)     00000190
ISN 0007              REAL*8 CC(2),DD(2),FS(2),UL(2),XL(12),XTRAY(12),ZC(2)          00000200
ISN 0008              REAL*8 AK1(121,12),AK2(121,12),AL5(121,12),XTRY(121,12)
ISN 0009              REAL*8 AJ(121,12,12)
ISN 0010              COMMON /ABC/ RR1,RR                                   00000250
ISN 0011              COMMON /BAY/ U,XTRAY
ISN 0012              COMMON /ABB/ BT,AT,A,B,QQ
ISN 0013              COMMON /ABF/ AJ                                       00000270
ISN 0014              COMMON /ABX/ F,FF,ERROR                               00000280
ISN 0015              COMMON /ABM/ AZE1,AZE2                                00000290
ISN 0016              COMMON /SEAR1/ ZC,X1,X2,NOCAL
ISN 0017              COMMON /BAX/ SR2,HOUP
ISN 0018              COMMON /ABD/ AK1,AK2,AL5                              00000320
ISN 0019              COMMON /SEAR/ TLIM
ISN 0020              COMMON /ABI/ M2,XTRY,XL
ISN 0021              COMMON /MAB/ NT,M                                     00000350


ISN 0022              COMMON /NAC/ IB,MK,KK
ISN 0023              READ(1,99) MESAGE
ISN 0024              READ(1,26) TN,H3,M1,M2,MK,KK,KEY                      00000380
ISN 0025              READ(1,1) N,NT,M,D,BOT,FF,ERROR                       00000390
ISN 0026              READ(1,1000) NOCAL,TLIM                               00000400
ISN 0027              READ(1,2)(ZC(I),I=1,M)                                00000410
ISN 0028              READ(1,1001)(CC(I),I=1,M)                             00000420
ISN 0029              READ(1,1001)(DD(I),I=1,M)                             00000430
ISN 0030              READ(1,1001)(FS(I),I=1,M)                             00000440
ISN 0031              READ(1,1001)(UL(I),I=1,M)                             00000450
ISN 0032              DO 54 I=1,M                                           00000460
ISN 0033           54 READ(1,1001)(RR(I,J),J=1,M)
ISN 0034              DO 90 I=1,M
ISN 0035           90 READ(1,1001)(RR1(I,J),J=1,M)                          00000470
ISN 0036              DO 49 I=1,N                                           00000480
ISN 0037           49 READ(1,1001)(B(I,J),J=1,M)
ISN 0038              READ(1,2)(HOUP(I),I=1,N)                              00000490
ISN 0039              READ(1,2) (F(I),I=1,N)                                00000500
ISN 0040              READ(1,2) (XTRAY(I),I=1,N)                            00000510
ISN 0041              READ(1,2) (XL(I),I=1,N)                               00000520
ISN 0042              DO 55 I=1,N                                           00000530
ISN 0043           55 READ(1,2)(A(I,J),J=1,N)                               00000540
ISN 0044              DO 52 I=1,M                                           00000550
ISN 0045           52 READ(1,2)(SS(I,J),J=1,N)                              00000560
ISN 0046              DO 53 I=1,M                                           00000570
ISN 0047           53 READ(1,2)(QQ(I,J),J=1,N)
ISN 0048              WRITE(3,94) MESAGE
ISN 0049           71 WRITE(3,59) TN,H3,M1,M2,MK,KK,KEY                     00000590
ISN 0050              WRITE(3,1) N,NT,M,D,BOT,FF                            00000600
ISN 0051              WRITE(3,1009) NOCAL,TLIM                              00000610
ISN 0052                 WRITE (3, 1005 )  ( CC(J), J = 1, M)               00000620
ISN 0053                 WRITE (3, 1006 )  ( DD(J), J = 1, M)               00000630
ISN 0054                 WRITE (3, 1007 )  ( FS(J), J = 1, M)               00000640
ISN 0055              WRITE(3,42)(ZC(I),I=1,M)                              00000650
ISN 0056              WRITE(3,3)(UL(I),I=1,M)                               00000660
ISN 0057              WRITE(3,9)                                            00000670
ISN 0058              WRITE(3,43)((RR(I,J),J=1,M),I=1,M)                    00000680
ISN 0059              WRITE(3,10)                                           00000690
ISN 0060              WRITE(3,48)((B(I,J),J=1,M),I=1,N)
ISN 0061              WRITE(3,70)(HOUP(I),I=1,N)                            00000720
ISN 0062              WRITE(3,50)(F(I),I=1,N)                               00000730
ISN 0063              WRITE(3,50)(XTRAY(I),I=1,N)                           00000740
ISN 0064              WRITE(3,51)(XL(I),I=1,N)
```

```
ISN 0065          WRITE(3,6)                                                00000720
ISN 0066          WRITE(3,5)((A(I,J),J=1,NY),I=1,N)                         00000740
ISN 0067          WRITE(3,7)                                                00000750
ISN 0068          WRITE(3,5)((SS(I,J),J=1,N),I=1,N)                         00000760
ISN 0069          WRITE(3,8)                                                00000770
ISN 0070          WRITE(3,5)((QQ(I,J),J=1,N),I=1,N)                         00000780
ISN 0071          X1=0.D0
ISN 0072          X2=0.D0
ISN 0073          SR2=DSQRT(2.D0)
ISN 0074          M1=M                                                      00000790
ISN 0075          NTN=NT+1                                                  00000800
ISN 0076          IW=NOCAL                                                  00000810
ISN 0077          BESTE=0.D0                                                00000820
ISN 0078          DO 11 K=1,NTN                                             00000830
ISN 0079          DO 11 I=1,N                                               00000840
ISN 0080          DO 11 J=1,N                                               00000850
ISN 0081       11 AJ(K,I,J)=0.D0                                            00000860
ISN 0082          DO 12 I=1,N                                               00000870
ISN 0083          DO 12 J=1,N                                               00000880
ISN 0084       12 AJ(NTN,I,J)=SS(I,J)                                       00000890
ISN 0085          DO 13 I=1,NTN                                             00000900
ISN 0086          DO 13 J=1,N                                               00000910
ISN 0087          XTRAY(I,J)=0.D0                                           00000920
ISN 0088          AK1(I,J)=0.D0                                             00000930
ISN 0089          AK2(I,J)=0.D0                                             00000940
ISN 0090       13 AL5(I,J)=0.D0                                             00000950
ISN 0091          DO 18 I=1,N                                               00000960
ISN 0092       18 XTRY(1,I)=XTRAY(I)                                        00000970
ISN 0093          DO 14 I=1,N                                               00000980
ISN 0094          AZE1(I)=0.D0                                              00000990
ISN 0095       14 AZE2(I)=0.D0                                              00001000
ISN 0096          AZE1(1)=1.D0                                              00001010
ISN 0097          AZE2(12)=1.D0                                             00001020
ISN 0098          DO 100 I=1,N                                              00001030
ISN 0099          DO 100 J=1,N                                              00001040
ISN 0100          A1(I,J)=A(I,J)                                            00001050
ISN 0101          AT1(I,J)=0.D0                                             00001060
ISN 0102      100 AT(I,J)=0.D0                                              00001070
ISN 0103          DO 47 I=1,N                                               00001080
ISN 0104          DO 47 J=1,M                                               00001090
ISN 0105       47 B1(I,J)=B(I,J)                                            00001100
ISN 0106          DO 101 I=1,M                                              00001110
ISN 0107          DO 101 J=1,N                                              00001120

ISN 0108          BT1(I,J)=0.D0                                             00001130
ISN 0109      101 BT(I,J)=0.D0                                              00001140
ISN 0110          CALL TRANP(A1,N,N,AT1)                                    00001150
ISN 0111          CALL TRANP(B1,N,M1,BT1)                                   00001160
ISN 0112          DO 46 I=1,M                                               00001170
ISN 0113          DO 46 J=1,N                                               00001180
ISN 0114       46 BT(I,J)=BT1(I,J)                                          00001190
ISN 0115          DO 17 I=1,N                                               00001200
ISN 0116          DO 17 J=1,N                                               00001210
ISN 0117          AT(I,J)=AT1(I,J)                                          00001220
ISN 0118       17 CONTINUE                                                  00001230
ISN 0119          WRITE(3,60)                                               00001240
ISN 0120          WRITE(3,66)((AT(I,J),J=1,N),I=1,N)                        00001250
ISN 0121          WRITE(3,61)                                               00001260
ISN 0122          WRITE(3,66)((BT(I,J),J=1,N),I=1,M)                        00001270
ISN 0123          WRITE(3,62)                                               00001280
ISN 0124          WRITE(3,59)((B1(I,J),J=1,M),I=1,M)                        00001290
ISN 0125          IF(KEY.GT.1)  GO TO 199
ISN 0127          CALL DKG(N,TN,H3)
ISN 0128          IB=0                                                      00001300
ISN 0129          CALL DRGV(N,TN,H3)
ISN 0130          IB=1                                                      00001310
ISN 0131          CALL DKGV(N,TN,H3)
ISN 0132          IF(MK.GT.1)  GO TO 25
ISN 0134          IB=2                                                      00001320
ISN 0135          CALL DKGV(N,TN,H3)
ISN 0136       25 IF(KEY.EQ.0)  GO TO 23
ISN 0138          WRITE(8) AJ,AK1,AK2,AL5
ISN 0139          RETURN
ISN 0140      199 READ(8) AJ,AK1,AK2,AL5
ISN 0141       23 IB=3
ISN 0142          CALL SFP(N,H1)                                            00001330
ISN 0143          RETURN                                                    00001340
ISN 0144     1000 FORMAT(I10,F20.10)                                        00001350
ISN 0145     1001 FORMAT(4F20.10)                                           00001360
ISN 0146        1 FORMAT(3I10,4F12.6)                                       00001370
ISN 0147       26 FORMAT(4F10.4,3I5)                                        00001380
ISN 0148        2 FORMAT(4F20.10)                                           00001390
ISN 0149     1004 FORMAT(1H0,5X,'NO. OF TRIALS=',I10,5X,'TOLERATED ERROR=',D12.6)  00001400
ISN 0150     1005    FORMAT ( ///  '  STEPSIZE          CC =', 2012.6)     00001410
ISN 0151     1006 FORMAT (' DECREMENTATION  DD =',2012.6)                   00001420
ISN 0152     1007 FORMAT (' STEPSIZE LIMIT    E =', 2D12.6)                00001430
ISN 0153       42 FORMAT(1H0,5X,'ZC1=',D20.10,'ZC2=',D20.10)                00001440
```

```
ISN 0154        51 FORMAT(1H0,10X,'FINAL   DEIVATION='/(1H0,2X,12D10.5))                00001850
ISN 0155        50 FORMAT(1H0,10X,'INITIAL DEIVATION='/(1H0,2X,12D10.4))               00001860
ISN 0156         3 FORMAT(1H0,5X,'FINAL CONTROL=',5X,2D20.10)                          00001870
ISN 0157        56 FORMAT(1H0,10X,'F(I)='/(1H0,2X,12D10.4))                            00001880
ISN 0158        59 FORMAT(1H0,5X,'N=',I2,5X,'M=',D12.6,5X,'M1=',D12.6,5X,'M2=',
                   1D12.6,5X,'MK=',I5,5X,'KK=',I5,5X,'KEY=',I5)
ISN 0159        94 FORMAT('1',80A1///)
ISN 0160        93 FORMAT(80A1)
ISN 0161         9 FORMAT(1H0,20X,' RR-MATRIX')                                        00001900
ISN 0162        48 FORMAT(1H0,5X,2D20.6)                                               00001910
ISN 0163        10 FORMAT(1H0,20X,'AE-MATRIX')                                         00001920
ISN 0164         6 FORMAT(1H0,20X,' A--MATRIX')                                        00001930
ISN 0165         8 FORMAT(1H0,20X,' CO--MATRIX')                                       00001940
ISN 0166         7 FORMAT(1H0,20X,' SS--MATRIX ')                                      00001950
ISN 0167         5 FORMAT(1H0,2X,12D10.4)                                              00001960
ISN 0168        60 FORMAT(1H0,20X,'A--TRANSPOSE  ')                                    00001970
ISN 0169        66 FORMAT(1H0,2X,12D10.4)                                              00001980
ISN 0170        61 FORMAT(1H0,20X,'B--TRANSPOSE  ')                                    00001990
ISN 0171        62 FORMAT(1H0,20X,'INVERSE OF RR-MATRIX')                              00002000
ISN 0172        58 FORMAT(1H0,5X,2D20.10)                                              00002010
ISN 0173        70 FORMAT(1H0,5X,'HOUP='/(1H0,2X,12D10.4))
ISN 0174           END                                                                 00002020
```

```
      COMPILER OPTIONS:- NAME=  MAIN,OPT=02,LINECNT=45,SIZE=0000K,
                         SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002              SUBROUTINE AMATM(A6,B6,R6,N)                                    00002030
          C           SUBROUTINE AMATM  (MATRIX MULTIPLICATION)                       00002040
          C                                                                           00002050
ISN 0003              IMPLICIT REAL*8(A-H,O-Z)                                        00002060
ISN 0004              DIMENSION A6(12,12),B6(12),R6(12)                               00002070
ISN 0005              DO 310 I=1,N                                                    00002080
ISN 0006          310 R6(I)=0.D0                                                      00002090
ISN 0007              DO 311 I=1,N                                                    00002100
ISN 0008              DO 311 J=1,N                                                    00002110
ISN 0009          311 R6(I)=R6(I)+A6(I,J)*B6(J)                                       00002120
ISN 0010              RETURN                                                          00002130
ISN 0011              END                                                             00002140
```

```
VEL 21.7 ( JAN 73 )          OS/360  FORTRAN H                                  DATE  73.271/1

      COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SIZE=0000K,
                         SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,IC,NOXREF
ISN 0002          SUBROUTINE FCT(KX,Y,DERY,NDIM)                                  00002150
ISN 0003          IMPLICIT REAL*8(A-H,O-Z)                                        00002160
ISN 0004          REAL*8 RR(2,2),AT(12,12),BT(2,12),AZE1(12),AZE2(12)             00002170
ISN 0005          REAL*8 Y(12),DERY(12),AJX(12,12),CY(12,12),CZ(2,12),CC(12,12)   00002180
ISN 0006          REAL*8 CH(12),CF(12),CC1(12),CM(12),CP(12),CII(12),ABJ(12,12)   00002190
ISN 0007          REAL*8 CO(12),ABX(12),CJ(12),CK(12),CN(12),CL(12,12)            00002200
ISN 0008          REAL*8 RR(2,2),F(12),ZC(2),B(12,2),  A(12,12),QQ(12,12),AT1(12,12)
ISN 0009          REAL*8 AK1(121,12),AK2(121,12),AL5(121,12),AJ(121,12,12)
ISN 0010          COMMON /ABC/  RR1,RR                                            00002230
ISN 0011          COMMON /ABB/  BT,AT,A,B,QQ
ISN 0012          COMMON /ABF/  AJ                                                00002250
ISN 0013          COMMON /SEAR1/ ZC                                               00002260
ISN 0014          COMMON /ABH/  AZE1,AZE2                                         00002270
ISN 0015          COMMON /ABX/  F,FF                                              00002280
ISN 0016          COMMON /ABD/  AK1,AK2,AL5                                       00002290
ISN 0017          COMMON /NAC/  IB,M1
ISN 0018          COMMON /NAB/NT,M                                                00002310
ISN 0019          NTT=NT+1                                                        00002320
ISN 0020          DO 1 I=1,NDIM                                                   00002330
ISN 0021          DO 1 J=1,NDIM                                                   00002340
ISN 0022        1 AJX(I,J)=AJ(KX,I,J)                                             00002350
ISN 0023          DO 3 I=1,M                                                      00002360
ISN 0024          DO 3 J=1,NDIM                                                   00002370
ISN 0025        3 CZ(I,J)=0.D0                                                    00002380
ISN 0026          DO 5 I=1,NDIM                                                   00002390
ISN 0027          DO 5 J=1,NDIM                                                   00002400
ISN 0028          CY(I,J)=0.D0                                                    00002410
ISN 0029          CL(I,J)=0.D0                                                    00002420
ISN 0030        5 CE(I,J)=0.D0                                                    00002430
ISN 0031          DO 12 I=1,NDIM                                                  00002440
ISN 0032          CJ(I)=0.D0                                                      00002450
ISN 0033          CN(I)=0.D0                                                      00002460
ISN 0034          CK(I)=0.D0                                                      00002470
ISN 0035          CO(I)=0.D0                                                      00002480
ISN 0036          CH(I)=0.D0                                                      00002490
ISN 0037          CF(I)=0.D0                                                      00002500
ISN 0038          CC1(I)=0.D0                                                     00002510
ISN 0039          CM(I)=0.D0                                                      00002520
ISN 0040       12 CP(I)=0.D0                                                      00002530
ISN 0041          DO 25 I=1,NDIM                                                  00002540
ISN 0042          DO 25 J=1,NDIM                                                  00002550

ISN 0043          ATI(I,J)=AT(I,J)                                                00002560
ISN 0044       25 CONTINUE                                                        00002570
ISN 0045          DO 2 K=1,NDIM                                                   00002580
ISN 0046          DO 2 I=1,M                                                      00002590
ISN 0047          DO 2 J=1,M                                                      00002600
ISN 0048        2 CZ(I,K)=CZ(I,K)+RR1(I,J)*BT(J,K)                                00002610
ISN 0049          DO 4 K=1,NDIM                                                   00002620
ISN 0050          DO 4 I=1,NDIM                                                   00002630
ISN 0051          DO 4 J=1,M                                                      00002640
ISN 0052        4 CY(I,K)=CY(I,K)+B(I,J)*CZ(J,K)                                  00002650
ISN 0053          DO 6 K=1,NDIM                                                   00002660
ISN 0054          DO 6 I=1,NDIM                                                   00002670
ISN 0055          DO 6 J=1,NDIM                                                   00002680
ISN 0056        6 CE(I,K)=CE(I,K)+AJX(I,J)*CY(J,K)                                00002690
ISN 0057        7 CALL AMATM(ATI,Y,CF,NDIM)                                       00002700
ISN 0058          CALL AMATM(CE,Y,CM,NDIM)                                        00002710
ISN 0059          IF(IB-1)13,14,9                                                 00002720
ISN 0060        8 IF(IB-2)10,10,11                                                00002730
ISN 0061       13 DO 23 I=1,NDIM                                                  00002740
ISN 0062       23 DERY(I)=CH(I)-CF(I)-AZE1(I)                                     00002750
ISN 0063          GO TO 15                                                        00002760
ISN 0064       14 DO 16 I=1,NDIM                                                  00002770
ISN 0065       16 DERY(I)=CH(I)-CF(I)-AZE2(I)                                     00002780
ISN 0066          GO TO 15                                                        00002790
ISN 0067       10 DO 17 I=1,NDIM                                                  00002800
ISN 0068       17 DERY(I)=CH(I)-CF(I)-AJX(I,5)                                    00002810
ISN 0069          GO TO 15                                                        00002820
ISN 0070       11 DO 18 K=1,NDIM                                                  00002830
ISN 0071          DO 18 I=1,NDIM                                                  00002840
ISN 0072          DO 18 J=1,NDIM                                                  00002850
ISN 0073       18 CL(I,K)=CL(I,K)+CY(I,J)*AJX(J,K)                                00002860
ISN 0074          DO 26 I=1,NDIM                                                  00002870
ISN 0075          DO 26 J=1,NDIM                                                  00002880
ISN 0076       26 ABJ(I,J)=A(I,J)-CL(I,J)                                         00002890
ISN 0077          CALL AMATM(ABJ,Y,ABY,NDIM)                                      00002900
ISN 0078          DO 19 I=1,NDIM                                                  00002910
ISN 0079          CJ(I)=AK1(KX,I)                                                 00002920
ISN 0080          CK(I)=AK2(KX,I)                                                 00002930
ISN 0081          IF(M1.GT.1)  GO TO 19
ISN 0083          CN(I)=AL5(KX,I)                                                 00002940
ISN 0084       19 CONTINUE                                                        00002950
ISN 0085          CALL AMATM(CY,CJ,CM, NDIM)                                      00002960
ISN 0086          CALL AMATM(CY,CK,CP, NDIM)                                      00002970
```

```
ISN 0087          IF(M1.GT.1)  GO TO 27                                          PA
ISN 0089          CALL AMATM(CY,CN,CO, NDIM)
ISN 0090       27 DO 24 I=1,NDIM                                                 00002980
ISN 0091       24 DERY(I)=ABY(I)-ZC(1)*CM(I)-ZC(2)*CP(I)-CO(I)*F(5)*F(I)         00002990
ISN 0092       15 RETURN
ISN 0093          END                                                            00003010
                                                                                 00003020
```

```
      COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SIZE=0000K,
                         SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002          SUBROUTINE FUNT( X,Y,DERY,NDIM)                                00003030
ISN 0003          IMPLICIT REAL*8(A-H,O-Z)                                       00003040
ISN 0004          REAL*8 A(12,12),B(12,2),CM(12,12),RR(2,2),F(12)                00003050
ISN 0005          REAL*8 BBJ(12,12),BJ(2,12),RBJ(2,12)                           00003060
ISN 0006          REAL*8 Y(12,12),DERY(12,12),C(12,12),CO(12,12),CJ(12,12)       00003070
ISN 0007          REAL*8 RRI(2,2),AT(12,12),BT(2,12)                             00003080
ISN 0008          COMMON /ABCA/ RR,FP                                            00003090
ISN 0009          COMMON /ASB/ BT,AT,A,B,CC
ISN 0010          COMMON /NA/ NT,M                                               00003110
ISN 0011          DO 1 I=1,NDIM                                                  00003120
ISN 0012          DO 1 J=1,NDIM                                                  00003130
ISN 0013          C(I,J)=0.00                                                    00003140
ISN 0014          CO(I,J)=0.00                                                   00003150
ISN 0015          CJ(I,J)=0.00                                                   00003160
ISN 0016        1 BBJ(I,J)=0.00                                                  00003170
ISN 0017          DO 5 I=1,M                                                     00003180
ISN 0018          DO 5 J=1,NDIM                                                  00003190
ISN 0019          BJ(I,J)=0.00                                                   00003200
ISN 0020        5 RBJ(I,J)=0.00                                                  00003210
ISN 0021          DO 2 K=1,NDIM                                                  00003220
ISN 0022          DO 2 I=1,NDIM                                                  00003230
ISN 0023          DO 2 J=1,NDIM                                                  00003240
ISN 0024          C(I,K)=C(I,K)+AT(I,J)*Y(J,K)                                   00003250
ISN 0025        2 CO(I,K)=CO(I,K)+Y(I,J)*A(J,K)                                  00003260
ISN 0026          DO 3 K=1,NDIM                                                  00003270
ISN 0027          DO 3 I=1,M                                                     00003280
ISN 0028          DO 3 J=1,NDIM                                                  00003290
ISN 0029        3 BJ(I,K)=BJ(I,K)+BT(I,J)*Y(J,K)                                 00003300
ISN 0030          DO 4 K=1,NDIM                                                  00003310
ISN 0031          DO 4 I=1,M                                                     00003320
ISN 0032          DO 4 J=1,M                                                     00003330
ISN 0033        4 RBJ(I,K)=RBJ(I,K)+RRI(I,J)*BJ(J,K)                            00003340
ISN 0034          DO 7 K=1,NDIM                                                  00003350
ISN 0035          DO 7 I=1,NDIM                                                  00003360
ISN 0036          DO 7 J=1,M                                                     00003370
ISN 0037        7 BBJ(I,K)=BBJ(I,K)+B(I,J)*RBJ(J,K)                              00003380
ISN 0038          DO 8 K=1,NDIM                                                  00003390
ISN 0039          DO 8 I=1,NDIM                                                  00003400
ISN 0040          DO 8 J=1,NDIM                                                  00003410
ISN 0041        8 CJ(I,K)=CJ(I,K)+Y(I,J)*BBJ(J,K)                                00003420
ISN 0042          DO 6 I=1,NDIM                                                  00003430
```

```
ISN 0043          DO 6 J=1,NDIM                                          00003440
ISN 0044        6 DERY(I,J)=CJ(I,J)-C(I,J)-CD(I,J)-QQ(I,J)               00003450
ISN 0045          RETURN                                                 00003460
ISN 0046          END                                                    00003470
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=55,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LLAD,MAP,NOEDIT,ID,NOXREF
ISN 0002          SUBROUTINE DKG(NDIM,TN,H)                              00003480
ISN 0003          IMPLICIT REAL*8(A-H,O-Z)                               00003490
ISN 0004          REAL*8 X(12,12),YY(12,12),DERY(12,12),AJ(121,12,12)
ISN 0005          REAL*8 K1(12,12),K2(12,12),K3(12,12),K4(12,12)         00003510
ISN 0006          COMMON /ABF/AJ                                         00003520
ISN 0007          COMMON /BAX/ SK2
ISN 0008          COMMON /NAB/ NT,M                                      00003540
ISN 0009          X=TN                                                   00003550
ISN 0010          NDT=NT+1
ISN 0011          N=NT+1                                                 00003570
ISN 0012          DO 1 I=1,NDIM                                          00003580
ISN 0013          DO 1 J=1,NDIM                                          00003590
ISN 0014          DERY(I,J)=0.D0                                         00003600
ISN 0015        1 Y(I,J)=AJ(NDT,I,J)                                     00003610
ISN 0016          GO TO 13                                               00003620
ISN 0017        8 CALL FUNT(X,Y,DERY,NDIM)                               00003630
ISN 0018          DO 2 I=1,NDIM                                          00003640
ISN 0019          DO 2 J=1,NDIM                                          00003650
ISN 0020          K1(I,J)=H*DERY(I,J)                                    00003660
ISN 0021        2 YY(I,J)=Y(I,J)+.5D0*K1(I,J)                            00003670
ISN 0022          CALL FUNT(X,YY,DERY,NDIM)                              00003680
ISN 0023          DO 3 I=1,NDIM                                          00003690
ISN 0024          DO 3 J=1,NDIM                                          00003700
ISN 0025          K2(I,J)=H*DERY(I,J)                                    00003710
ISN 0026        3 YY(I,J)=Y(I,J)+(1.D0-1.D0/SR2)*K2(I,J)+(-.5D0+1.D0/SR2)*K1(I,J)  00003720
ISN 0027          CALL FUNT(X,YY,DERY,NDIM)                              00003730
ISN 0028          DO 5 I=1,NDIM                                          00003740
ISN 0029          DO 5 J=1,NDIM                                          00003750
ISN 0030          K3(I,J)=H*DERY(I,J)                                    00003760
ISN 0031        5 YY(I,J)=Y(I,J)-(1.D0/SR2)*K2(I,J)+(1.D0+1.D0/SR2)*K3(I,J)  00003770
ISN 0032          CALL FUNT(X,YY,DERY,NDIM)                              00003780
ISN 0033          DO 6 I=1,NDIM                                          00003790
ISN 0034          DO 6 J=1,NDIM                                          00003800
ISN 0035          K4(I,J)=H*DERY(I,J)                                    00003810
ISN 0036        6 Y(I,J)=Y(I,J)+K1(I,J)/6.D0+((1.D0-1.D0/SR2)/3.D0)*K2(I,J)+  00003820
                 1((1.D0+1.D0/SR2)/3.D0)*K3(I,J)+K4(I,J)/6.D0             00003830
ISN 0037          DO 7 I=1,NDIM                                          00003840
ISN 0038          DO 7 J=1,NDIM                                          00003850
ISN 0039        7 AJ(N,I,J)=Y(I,J)                                       00003860
ISN 0040       13 N=N-1                                                  00003870
ISN 0041          X=X+H
```

```
ISN 0042          IF(N.NE.0)  GO TO 8                                    00003880
ISN 0044          N=N+1                                                  00003890
ISN 0045          X=X-M                                                  00003900
ISN 0046          WRITE(3,11) X                                          00003910
ISN 0047          WRITE(3,10) N                                          00003920
ISN 0048          WRITE(3,9)((Y(I,J),J=1,NDIM),I=1,NDIM)                 00003930
ISN 0049          RETURN                                                 00003940
ISN 0050       11 FORMAT(1HO,5X,' X FROM FUNT =',D12.6)                  00003950
ISN 0051       10 FORMAT(1HO,5X,'N=',I5,10X,'AJ-MATRIX')                 00003960
ISN 0052        9 FORMAT(2X,12D10.4)                                     00003970
ISN 0053          END                                                    00003980
```

```
COMPILER OPTIONS - NAME= MAIN,OPT=02,LINECNT=45,SIZE=0000K,
                   SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002          SUBROUTINE TRANP(A,N,M,B)                              00003990
                C SUBROUTINE TRANP(A,N,M,B)                              00004000
                C CALCULATION OF THE TRANSPOSE OF A NBYM MATRIX          00004010
                C RESULT--------M BY N MATRIX                            00004020
ISN 0003          IMPLICIT REAL*8(A-H,O-Z)                               00004030
ISN 0004          DIMENSION A(N,M),B(M,N)                                00004040
ISN 0005          DO 1 I=1,M                                             00004050
ISN 0006          DO 1 J=1,N                                             00004060
ISN 0007        1 B(I,J)=0.D0                                            00004070
ISN 0008          DO 2 I=1,M                                             00004080
ISN 0009          DO 2 J=1,N                                             00004090
ISN 0010        2 B(I,J)=A(J,I)                                          00004100
ISN 0011          RETURN                                                 00004110
ISN 0012          END                                                    00004120
```

```
EVEL 21.7 ( JAN 73 )          OS/360 FORTRAN H                          DATE 73.271/1

      COMPILER OPTIONS - NAME= MAIN,OPT=02,LINECNT=65,SIZE=0000K,
                         SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002            SUBROUTINE DKGV(NDIM,TN,HO)
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
ISN 0004            REAL*8 HOUP(12), Y(12),YY(12),DERY(12),K1(12),K2(12),K3(12),K4(12) 00004700
ISN 0005            REAL*8 AK1(121,12),AK2(121,12),AL5(121,12),XTRY(121,12)
ISN 0006            COMMON /ABC/ AK1,AK2,AL5                                  00004720
ISN 0007            COMMON /BXX/ SR2,HOUP
ISN 0008            COMMON /ABI/ H2,XTRY
ISN 0009            COMMON /N4C/ I8,M1,KK
ISN 0010            COMMON /NAB/ NT,M                                         00004760
ISN 0011            M=MO
ISN 0012            IF(I8.EQ.3)  H=H2
ISN 0014            KN=1
ISN 0015            X=TN                                                      00004770
ISN 0016            NTT=NT+1                                                  00004780
ISN 0017            NT2=NTT+1                                                 00004790
ISN 0018            IF(X.NE.0.D0)  GO TO 12                                   00004800
ISN 0020            N1=1                                                      00004810
ISN 0021            N=0
ISN 0022            DO 21 I=1,NDIM                                            00004830
ISN 0023         21 Y(I)=XTRY(1,I)                                            00004840
ISN 0024            GO TO 14                                                  00004850
ISN 0025         12 N1=-1                                                     00004860
ISN 0026            N=NTT                                                     00004870
ISN 0027            DO 22 I=1,NDIM                                            00004880
ISN 0028               YY(I)=0.D0
ISN 0029         22 Y(I)=0.D0                                                 00004890
ISN 0030         14 DO 1 I=1,NDIM                                             00004900
ISN 0031          1 DERY(I)=0.D0                                              00004910
ISN 0032            GO TO 13                                                  00004920
ISN 0033          9 IF(I8-2) 39,39,23
ISN 0034         23 DO 24 I=1,NDIM
ISN 0035         24 YY(I)=Y(I)/HOUP(I)
ISN 0036         39 CALL FCT(N,YY,DERY,NDIM)
ISN 0037            DO 2 I=1,NDIM                                             00004940
ISN 0038            K1(I)=H*DERY(I)                                           00004950
ISN 0039          2 YY(I)=Y(I)+.500*K1(I)                                     00004960
ISN 0040            IF(I8-2) 25,25,26
ISN 0041         26 DO 31 I=1,NDIM
ISN 0042         31 YY(I)=YY(I)/HOUP(I)
ISN 0043         25 CALL FCT(N,YY,DERY,NDIM)                                  00004970
ISN 0044            DO 3 I=1,NDIM                                             00004980

ISN 0045            K2(I)=H*DERY(I)                                           00004990
ISN 0046          3 YY(I)=Y(I)+(1.D0-1.D0/SR2)*K2(I)+(-.500+1.D0/SR2)*K1(I)   00005000
ISN 0047            IF(I8-2) 27,27,28
ISN 0048         28 DO 30 I=1,NDIM
ISN 0049         30 YY(I)=YY(I)/HOUP(I)
ISN 0050         27 CALL FCT(N,YY,DERY,NDIM)                                  00005010
ISN 0051            DO 5 I=1,NDIM                                             00005020
ISN 0052            K3(I)=H*DERY(I)                                           00005030
ISN 0053          5 YY(I)=Y(I)-(1.D0/SR2)*K2(I)+(1.D0+1.D0/SR2)*K3(I)         00005040
ISN 0054            IF(I8-2) 29,29,32
ISN 0055         32 DO 33 I=1,NDIM
ISN 0056         33 YY(I)=YY(I)/HOUP(I)
ISN 0057         29 CALL FCT(N,YY,DERY,NDIM)                                  00005050
ISN 0058            DO 6 I=1,NDIM                                             00005060
ISN 0059            K4(I)=H*DERY(I)                                           00005070
ISN 0060          6 Y(I)=Y(I)+K1(I)/6.D0+((1.D0-1.D0/SR2)/3.D0)*K2(I)+        00005080
                    1((1.D0+1.D0/SR2)/3.D0)*K3(I)+K4(I)/6.D0                  00005090
ISN 0061            IF(I8-1) 17,10,20                                         00005100
ISN 0062         20 IF(I8-2) 15,15,4                                          00005110
ISN 0063         17 DO 7 I=1,NDIM                                             00005120
ISN 0064          7 AK1(N,I)=Y(I)                                             00005130
ISN 0065            GO TO 13                                                  00005140
ISN 0066         10 DO 18 I=1,NDIM                                            00005150
ISN 0067         18 AK2(N,I)=Y(I)                                             00005160
ISN 0068            GO TO 13                                                  00005170
ISN 0069         15 DO 19 I=1,NDIM                                            00005180
ISN 0070         19 AL5(N,I)=Y(I)                                             00005190
ISN 0071            GO TO 13                                                  00005200
ISN 0072          4 CONTINUE
ISN 0073            IF(KN-KK*(KN/KK)) 42,40,42
ISN 0074         42 KN=KN+1
ISN 0075            GO TO 8
ISN 0076         40 N=N+N1
ISN 0077            X=X+HO                                                    00005210
ISN 0078            DO 16 I=1,NDIM                                            00005220
ISN 0079         16 XTRY(N,I)=Y(I)
ISN 0080            KN=1
ISN 0081            GO TO 41                                                  00005230
ISN 0082         13 N=N+N1                                                    00005240
ISN 0083            X=X+H
ISN 0084         41 IF((N.NE.0).AND.(N.NE.NTT))  GO TO 9                      00005260
ISN 0086            N=N-N1                                                    00005270
ISN 0087            X=X-H
```

```
ISN 0088            WRITE(3,11) X,N                                          00005280
ISN 0089            WRITE(3,9)(Y(I),I=1,NDIM)                                00005290
ISN 0090            RETURN                                                   00005300
ISN 0091         11 FORMAT(1H0,5X,'X=',D12.6,'N=',I5,5X,'Y-VECTORS')         00005310
ISN 0092          9 FORMAT(2X,10D12.6)                                       00005320
ISN 0093            END                                                      00005330
```

```
      COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SIZE=0000K,
                        SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002            SUBROUTINE DGSF(H,Y,Z,NDIM)                              00007020
                    C                                                        00007030
                    C     SUBROUTINE DGSF(H,Y,Z,NDIM)                        00007040
                    C                                                        00007050
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)                                 00007060
ISN 0004              REAL*8 Y(121),Z(121)
                    C                                                        00007080
ISN 0005            HT=.3333333333333D0*H                                    00007090
ISN 0006            IF(NDIM-5) 7,6,1                                         00007100
                    C                                                        00007110
                    C     NDIM IS GREATER THAN 5. PREPARATIONS OF INTEGRATION LOOP  00007120
ISN 0007          1 SUM1=Y(2)+Y(2)                                           00007130
ISN 0008            SUM1=SUM1+SUM1                                           00007140
ISN 0009            SUM1=HT*(Y(1)+SUM1+Y(3))                                 00007150
ISN 0010            AUX1=Y(4)+Y(4)                                           00007160
ISN 0011            AUX1=AUX1+AUX1                                           00007170
ISN 0012            AUX1=SUM1+HT*(Y(3)+AUX1+Y(5))                            00007180
ISN 0013            AUX2=HT*(Y(1)+3.875D0*(Y(2)+Y(5))+2.625D0*(Y(3)+Y(4))+Y(6))  00007190
ISN 0014            SUM2=Y(5)+Y(5)                                           00007200
ISN 0015            SUM2=SUM2+SUM2                                           00007210
ISN 0016            SUM2=AUX2-HT*(Y(4)+SUM2+Y(6))                            00007220
ISN 0017            Z(1)=0.D0                                                00007230
ISN 0018            AUX=Y(3)+Y(3)                                            00007240
ISN 0019            AUX=AUX+AUX                                              00007250
ISN 0020            Z(2)=SUM2-HT*(Y(2)+AUX+Y(4))                             00007260
ISN 0021            Z(3)=SUM1                                                00007270
ISN 0022            Z(4)=SUM2                                                00007280
ISN 0023            IF(NDIM-6) 5,5,2                                         00007290
                    C                                                        00007300
                    C     INTEGRATION LOOP                                   00007310
ISN 0024          2 DO 4 I=7,NDIM,2                                          00007320
ISN 0025            SUM1=AUX1                                                00007330
ISN 0026            SUM2=AUX2                                                00007340
ISN 0027            AUX1=Y(I-1)+Y(I-1)                                       00007350
ISN 0028            AUX1=AUX1+AUX1                                           00007360
ISN 0029            AUX1=SUM1+HT*(Y(I-2)+AUX1+Y(I))                          00007370
ISN 0030            Z(I-2)=SUM1                                              00007380
ISN 0031            IF(I-NDIM)3,3,4                                          00007390
ISN 0032          3 AUX2=Y(I)+Y(I)                                           00007400
ISN 0033            AUX2=AUX2+AUX2                                           00007410
ISN 0034            AUX2=SUM2+HT*(Y(I-1)+AUX2+Y(I+1))                        00007420
```

```
ISN 0035          4 Z(I-1)=SUM2                                                  00007430
ISN 0036          5 Z(NDIM-1)=AUX1                                               00007440
ISN 0037            Z(NDIM)=AUX2                                                 00007450
ISN 0038            RETURN                                                       00007460
ISN 0039          6 Z(NDIM-1)=SUM2                                               00007470
ISN 0040            Z(NDIM)=AUX1                                                 00007480
ISN 0041            RETURN                                                       00007490
                C   END OF INTEGRATION LOOP                                      00007500
                C                                                                00007510
ISN 0042          7 IF(NDIM-3) 12,11,8                                           00007520
                C                                                                00007530
                C   NDIM IS EQUAL TO 4 OR 5                                     00007540
ISN 0043          8 SUM2=1.1250*HT*(Y(1)+Y(2)+Y(2)+Y(2)+Y(3)+Y(3)+Y(3)+Y(4))     00007550
ISN 0044            SUM1=Y(2)+Y(2)                                               00007560
ISN 0045            SUM1=SUM1+SUM1                                               00007570
ISN 0046            SUM1=HT*(Y(1)+SUM1+Y(3))                                     00007580
ISN 0047            Z(1)=0.00                                                    00007590
ISN 0048            AUX1=Y(3)+Y(3)                                               00007600
ISN 0049            AUX1=AUX1+AUX1                                               00007610
ISN 0050            Z(2)=SUM2-HT*(Y(2)+AUX1+Y(4))                                00007620
ISN 0051            IF(NDIM-5)10,9,9                                             00007630
ISN 0052          9 AUX1=Y(4)+Y(4)                                               00007640
ISN 0053            AUX1=AUX1+AUX1                                               00007650
ISN 0054            Z(5)=SUM1+HT*(Y(3)+AUX1+Y(5))                                00007660
ISN 0055         10 Z(3)=SUM1                                                    00007670
ISN 0056            Z(4)=SUM2                                                    00007680
ISN 0057            RETURN                                                       00007690
                C                                                                00007700
                C   NDIM IS EQUAL TO 3                                           00007710
ISN 0058         11 SUM1=HT*(1.2500*Y(1)+Y(2)+Y(2)-2.500*Y(3))                   00007720
ISN 0059            SUM2=Y(2)+Y(2)                                               00007730
ISN 0060            SUM2=SUM2+SUM2                                               00007740
ISN 0061            Z(3)=HT*(Y(1)+SUM2+Y(3))                                     00007750
ISN 0062            Z(1)=0.00                                                    00007760
ISN 0063            Z(2)=SUM1                                                    00007770
ISN 0064         12 RETURN                                                       00007780
ISN 0065            END                                                          00007790
```

```
      COMPILER OPTIONS - NAME= MAIN,OPT=02,LINECNT=45,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002            SUBROUTINE ROOT(X0,H,NDIM)                                   00007800
                C   SUBROUTINE ROOT(X0,H,NDIM)                                   00007810
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)                                     00007820
ISN 0004            REAL*8 AK1(121,12),AK2(121,12),AL5(121,12),XTRY(121,12)
ISN 0005            REAL*8 AK11(121),AK22(121),AL55(121),AJ(121,12,12),AAJ1(12,12)
ISN 0006            REAL*8 XZ1(121),XZ12(121),XY1(121),XY12(121)
ISN 0007            REAL*8 BT(2,12),          U(2) ,XTRAY(12),ZJB(2),ZJR(2),XJ(12)
ISN 0008            REAL*8 ZC(2),XL(12),RPI(2,2),F(12),ZJ(12)                    00007840
ISN 0009            COMMON /ABC/ RPI
ISN 0010            COMMON /BAY/ U,XTRAY
ISN 0011            COMMON /ABB/ BT
ISN 0012            COMMON /ABD/ AK1,AK2,AL5
ISN 0013            COMMON /ABI/ H2,XTRY,XL
ISN 0014            COMMON /SEAR1/ ZC,X1,X2,NOCAL
ISN 0015            COMMON /ABE/ NT,M                                            00007870
ISN 0016            COMMON /ABX/ F,FF,ERROR
ISN 0017            COMMON /ABF/ AJ
ISN 0018              N=NDIM
ISN 0019            NOCAL=NOCAL-1
ISN 0020            TTN=0.00                                                     00007880
ISN 0021            NDIT=NT+1                                                    00007890
ISN 0022            DO 20 I=1,NDIM
ISN 0023            ZJ(I)=0.00
ISN 0024         20 XJ(I)=0.00
ISN 0025            CALL OKGV(NDIM,TTN,H)                                        00007900
ISN 0026            DO 3 I=1,NDIT                                                00007910
ISN 0027            XZ1(I)=0.00                                                  00007920
ISN 0028            XZ12(I)=0.00                                                 00007930
ISN 0029            XY1(I)=XTRY(I,1)                                             00007940
ISN 0030          3 XY12(I)=XTRY(I,12)                                           00007950
ISN 0031            WRITE(3,1) ZC(1),ZC(2)                                       00007960
ISN 0032            WRITE(3,4)                                                   00007970
ISN 0033            WRITE(3,5)(XY1(I),I=1,NDIT)                                  00007980
ISN 0034            WRITE(3,5)                                                   00007990
ISN 0035            WRITE(3,6)(XY12(I),I=1,NDIT)                                 00008000
ISN 0036            CALL DOSP(N,XY1,XZ1,NDIT)                                    00008010
ISN 0037            CALL DOSP(N,XY12,XZ12,NDIT)                                  00008020
ISN 0038            X1=DABS(XZ12(NDIT))
ISN 0039            X2=DABS(XZ1(NDIT))
ISN 0040            XC=X1+X2
ISN 0041            WRITE(3,7) XC                                                00008040
```

```
                                                                                PA
ISN 0042          WRITE(3,10) XX1(NOIT),XZ12(NOIT)
ISN 0043          IF((DABS(ZC(1)).LE.1.D-20).AND.(DABS(ZC(2)).LE.1.D-20)) GO TO 82
ISN 0045       83 IF(XO.GT.ERROR)  GO TO 2
ISN 0047       82 DO 44 K=1,NOIT
ISN 0048          DO 35 I=1,N                                                   00001490
ISN 0049           DO 35 J=1,N                                                  00001500
ISN 0050       35 AAJ1(I,J)=AJ(K,I,J)                                           00001510
ISN 0051          DO 34 I=1,N                                                   00001520
ISN 0052          AK11(I)=AK1(K,I)                                              00001530
ISN 0053          AK22(I)=AK2(K,I)                                              00001540
ISN 0054          AL55(I)=AL5(K,I)                                              00001550
ISN 0055       34 XTRAY(I)=XTRY(K,I)                                            00001560
ISN 0056          CALL XMATM(AAJ1,XTRAY,XJ,N)                                   00001570
ISN 0057          DO 36 I=1,N                                                   00001580
ISN 0058       36 ZJ(I)=XJ(I)+ZC(1)*AK11(I)+ZC(2)*AK22(I)+F(5)*AL55(I)
ISN 0059          DO 37 I=1,M                                                   00001600
ISN 0060          ZJB(I)=0.DO                                                   00001610
ISN 0061       37 ZJR(I)=0.DO                                                   00001620
ISN 0062          DO 38 I=1,M                                                   00001630
ISN 0063          DO 38 J=1,N                                                   00001640
ISN 0064       38 ZJB(I)=ZJB(I)+BT(I,J)*ZJ(J)                                   00001650
ISN 0065          DO 39 I=1,M                                                   00001660
ISN 0066          DO 39 J=1,M                                                   00001670
ISN 0067       39 ZJR(I)=ZJR(I)+RRI(I,J)*ZJB(J)                                 00001680
ISN 0068          DO 40 I=1,M                                                   00001690
ISN 0069       40 U(I)=-ZJR(I)                                                  00001700
ISN 0070          U1=U(1)
ISN 0071          U2=U(2)
ISN 0072          KK=K-1
ISN 0073          WRITE(3,41) KK,U1,U2
ISN 0074       44 CONTINUE                                                      00001730
ISN 0075        2 RETURN                                                        00008050
ISN 0076        1 FORMAT(1HO,5X,'ZC(1)=',D12.6,10X,'ZC(2)=',D12.6)              00008060
ISN 0077        7 FORMAT(1HO,5X,'XO=',D12.6)                                    00008070
ISN 0078        4 FORMAT(1HO,10X,' BOTTOM COMPOSITION TRAJECTORY')              00008080
ISN 0079        5 FORMAT(1HO,10X,'    TOP COMPOSITION TRAJECTORY')              00008090
ISN 0080        6 FORMAT(1HO,5X,1PD12.6)                                        00008100
ISN 0081       10 FORMAT(1HO,5X,'BOT. DEV.=',D12.6,5X,'TOP DEV.=',D12.6)
ISN 0082       41 FORMAT(1HO,5X,'INTERVAL=',I5,5X,'U(1)=',D12.6,5X,'U(2)=',D12.6)
ISN 0083          END                                                           00009110
```

```
      COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SIZE=0000K,
                        SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002          SUBROUTINE SEP(NDIM,M)
                  C SUBROUTINE SEP(NDIM,M)
                  C TWO DIMENSIONAL SEARCH---NEWTON-RAPHSON METHOD
ISN 0003          IMPLICIT REAL*8(A-H,O-Z)
ISN 0004          DIMENSION SS(2,2),ZC(2),ZCN(2),OZC(2),XN(2),CX(2),XM(2)
ISN 0005          COMMON /NAC/ NJ,NOPAR
ISN 0006          COMMON /SEAR1/ ZC,X1,X2,NOCAL
ISN 0007          COMMON /SEAR/ ERR
ISN 0008          MMK=0
ISN 0009          XO=0.DO
ISN 0010          NOCAL=NOCAL
ISN 0011          DELTA=.0000100
ISN 0012          CALL ROOT1(XO,M,NDIM)
ISN 0013       8  DO 1 I=1,NOPAR
ISN 0014          CX(1)=X1
ISN 0015          CX(2)=X2
ISN 0016          ZC(I)=ZC(I)+DELTA
ISN 0017          CALL ROOT1(XO,M,NDIM)
ISN 0018          XM(1)=X1
ISN 0019          XM(2)=X2
ISN 0020          DO 2 J=1,NOPAR
ISN 0021       2  SS(J,I)=(XN(J)-CX(J))/DELTA
ISN 0022          OZC(I)=ZC(I)
ISN 0023       1  ZC(I)=ZC(I)-DELTA
ISN 0024          DO 3 I=1,NOPAR
ISN 0025          XM(I)=-CX(I)
ISN 0026          ZCN(I)=0.DO
ISN 0027       3  ZC(I)=OZC(I)
ISN 0028          CALL DMATIN(SS,NOPAR,MMK)
ISN 0029          IF(MMK.GT.0)  GO TO 15
ISN 0031          DO 6 I=1,NOPAR
ISN 0032          DO 6 J=1,NOPAR
ISN 0033       6  ZCN(I)=ZCN(I)+SS(I,J)*XM(J)
ISN 0034          DO 7 I=1,NOPAR
ISN 0035       7  ZC(I)=ZC(I)+ZCN(I)
ISN 0036          CALL ROOT1(XO,M,NDIM)
ISN 0037          IF((X1.GT.ERR).OR.(X2.GT.ERR))  GO TO 8
ISN 0039          J=CAL=NOCAL-NOCAL
ISN 0040          WRITE(3,14) NOCAL
ISN 0041       14 FORMAT(1HO,5X,'THE MINIMUM HAS BEEN LOCATED IN',5X,I10,5X,
```

```
ISN 0042        15 RETURN
ISN 0043           END
```

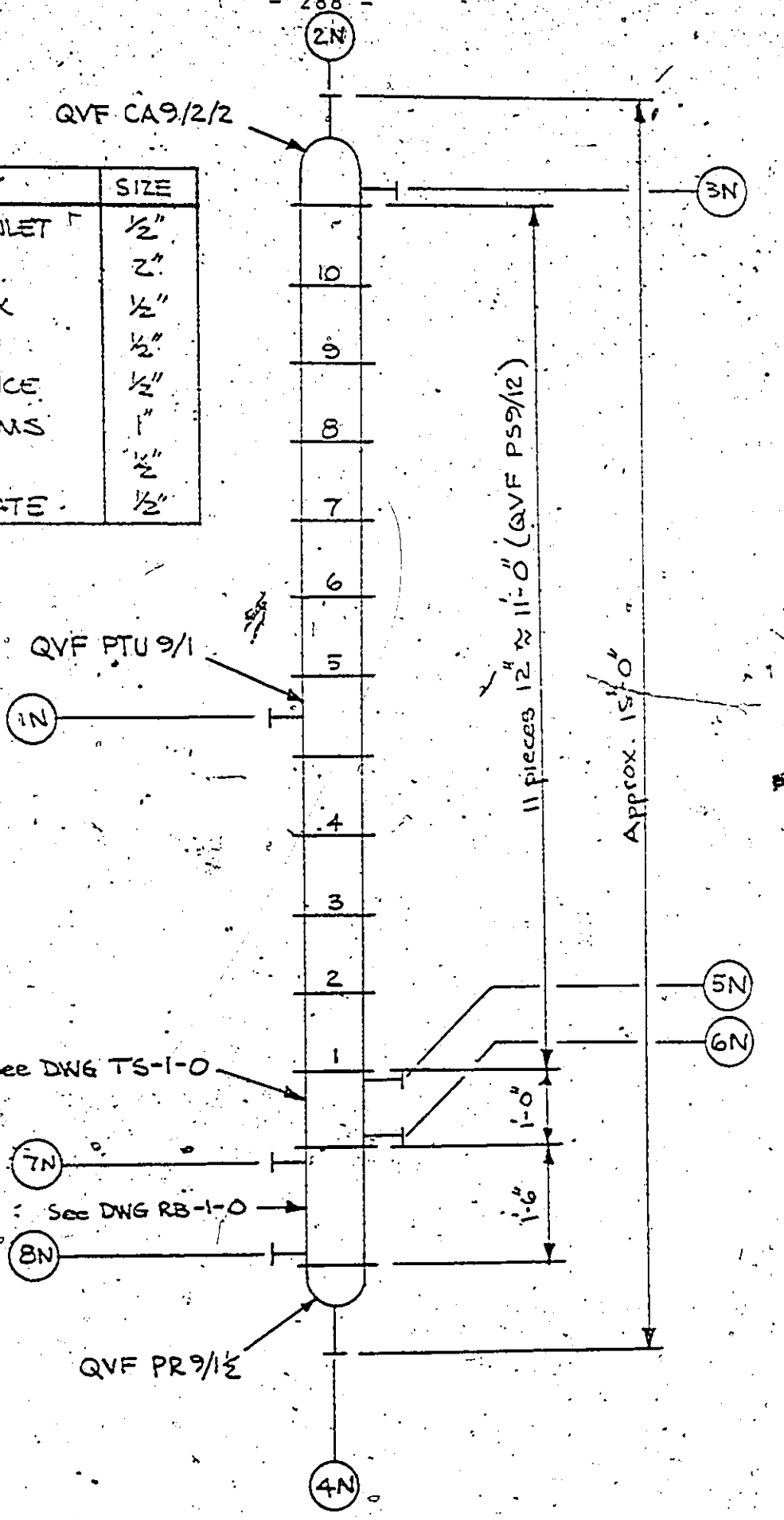
```
      COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=45,SIZE=0000K,
                         SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002              SUBROUTINE DMATIN(A,N,MMK)
                    C SUBROUTINE CMATIN(A,N,MMK)                (MATRIX INVERSION)
ISN 0003              IMPLICIT REAL*8(A-H,O-Z)
ISN 0004              REAL*8 A(2,2)
ISN 0005              INTEGER*4 INDEX(2,2)
ISN 0006              DO 109 I=1,N
ISN 0007           108 INDEX(I,1)=0
ISN 0008              II=0
ISN 0009           109 AMAX=-1.00
ISN 0010              DO 110 I=1,N
ISN 0011              IF(INDEX(I,1))110,111,110
ISN 0012           111 DO 112 J=1,N
ISN 0013              IF(INDEX(J,1))112,113,112
ISN 0014           113 TEMP=DABS(A(I,J))
ISN 0015              IF(TEMP-AMAX)112,112,114
ISN 0016           114 IROW=I
ISN 0017              ICOL=J
ISN 0018              AMAX=TEMP
ISN 0019           112 CONTINUE
ISN 0020           110 CONTINUE
ISN 0021              IF(AMAX)225,115,116
ISN 0022           116 INDEX(ICOL,1)=IROW
ISN 0023              IF(IROW-ICOL)119,118,119
ISN 0024           119 DO 120 J=1,N
ISN 0025              TEMP=A(IROW,J)
ISN 0026              A(IROW,J)=A(ICOL,J)
ISN 0027           120 A(ICOL,J)=TEMP
ISN 0028              II=II+1
ISN 0029              INDEX(II,2)=ICOL
ISN 0030           118 PIVOT=A(ICOL,ICOL)
ISN 0031              A(ICOL,ICOL)=1.00
ISN 0032              PIVOT=1.00/PIVOT
ISN 0033              DO 121 J=1,N
ISN 0034           121 A(ICOL,J)=A(ICOL,J)*PIVOT
ISN 0035              DO 122 I=1,N
ISN 0036              IF(I-ICOL)123,122,123
ISN 0037           123 TEMP=A(I,ICOL)
ISN 0038              A(I,ICOL)=0.
ISN 0039              DO 124 J=1,N
ISN 0040           124 A(I,J)=A(I,J)-A(ICOL,J)*TEMP
ISN 0041           122 CONTINUE
```

```
ISN 0042          GO TO 109
ISN 0043      125 ICOL=INDEX(II,2)
ISN 0044          DO 126 I=1,9
ISN 0045          IROW=INDEX(ICOL,1)
ISN 0046          TEMP=A(I,IROW)
ISN 0047          A(I,IPO)=A(I,ICOL)
ISN 0048      126 A(I,ICOL)=TEMP
ISN 0049          II=II-1
ISN 0050      225 IF(II)125,127,125
ISN 0051      115 WRITE(6,1001)
ISN 0052              NPR=1
ISN 0053     1001 FORMAT(1HO,2X,21H ZERO PIVOT)
ISN 0054      127 RETURN
ISN 0055          END
```

APPENDIX I

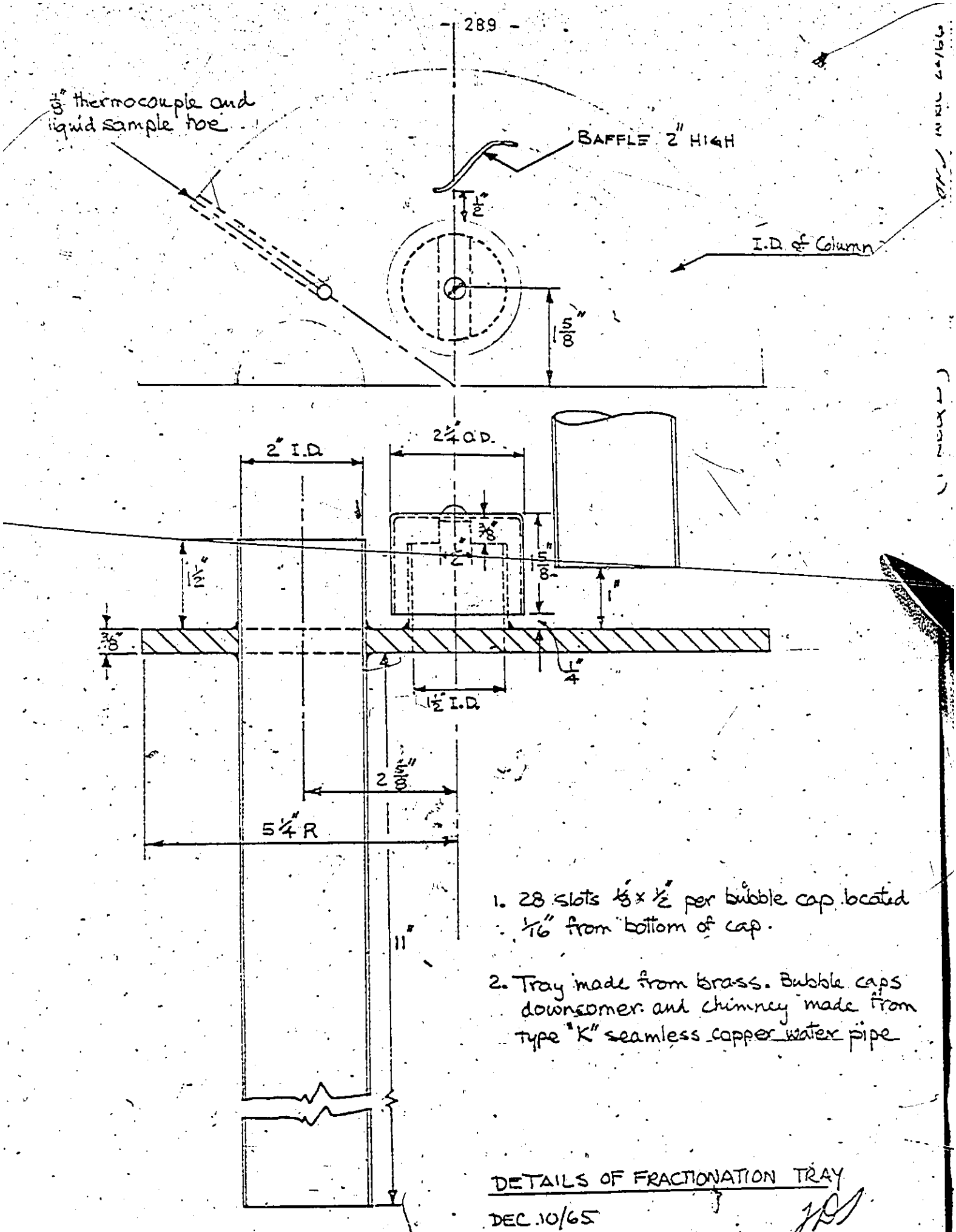
WORKING DRAWINGS OF THE DISTILLATION COLUMN

NO.	DUTY	SIZE
1N	FEED INLET	½"
2N	VAPOR	2"
3N	REFLUX	½"
4N	DRAIN	½"
5N	BALANCE	½"
6N	BOTTOMS	1"
7N	STEAM	½"
8N	CONDENSATE	½"



LABORATORY - COLUMN

AUG 25/65

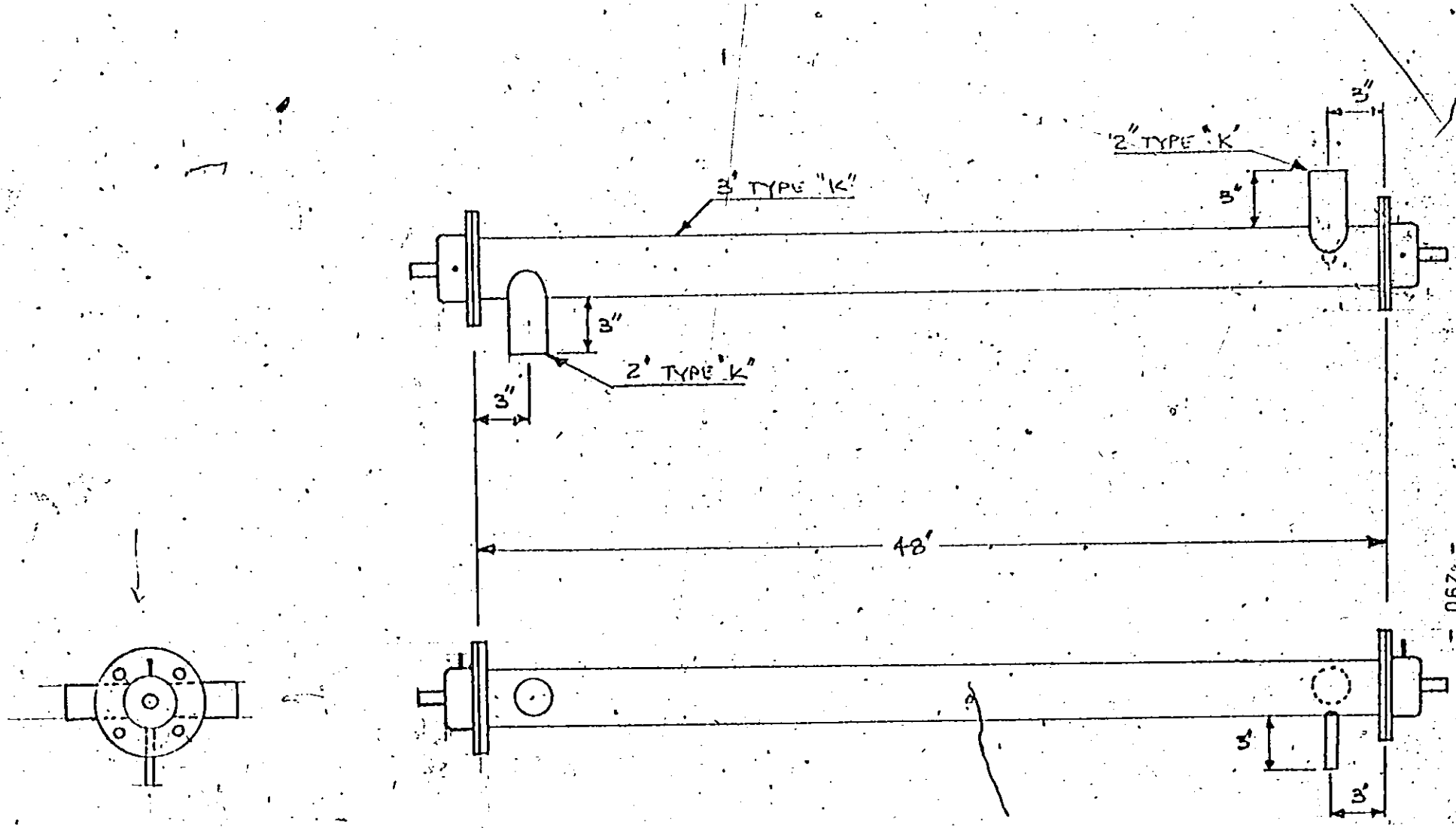


1. 28 slots 1/8 × 1/2 per bubble cap located 1/16" from bottom of cap.

2. Tray made from brass. Bubble caps downcomer and chimney made from type "K" seamless copper water pipe

DETAILS OF FRACTIONATION TRAY

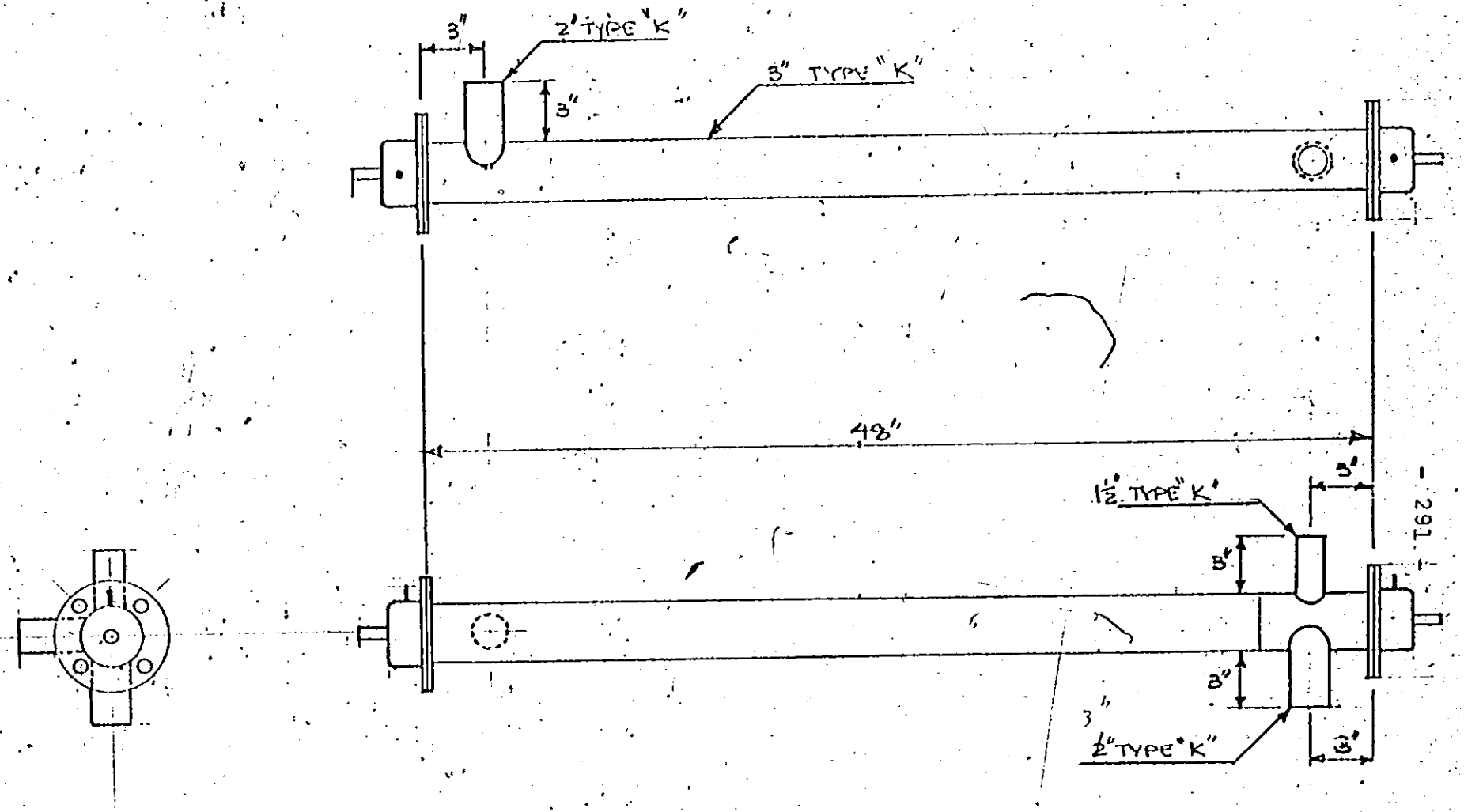
DEC. 10/65



Note: bottom cooler nozzles 3/4" type "K"
shell for feed preheater made from 3" sch 40 stl. pipe

DETAILS OF CONDENSER "A"
(1 REQ'D.)

APRIL 22/66

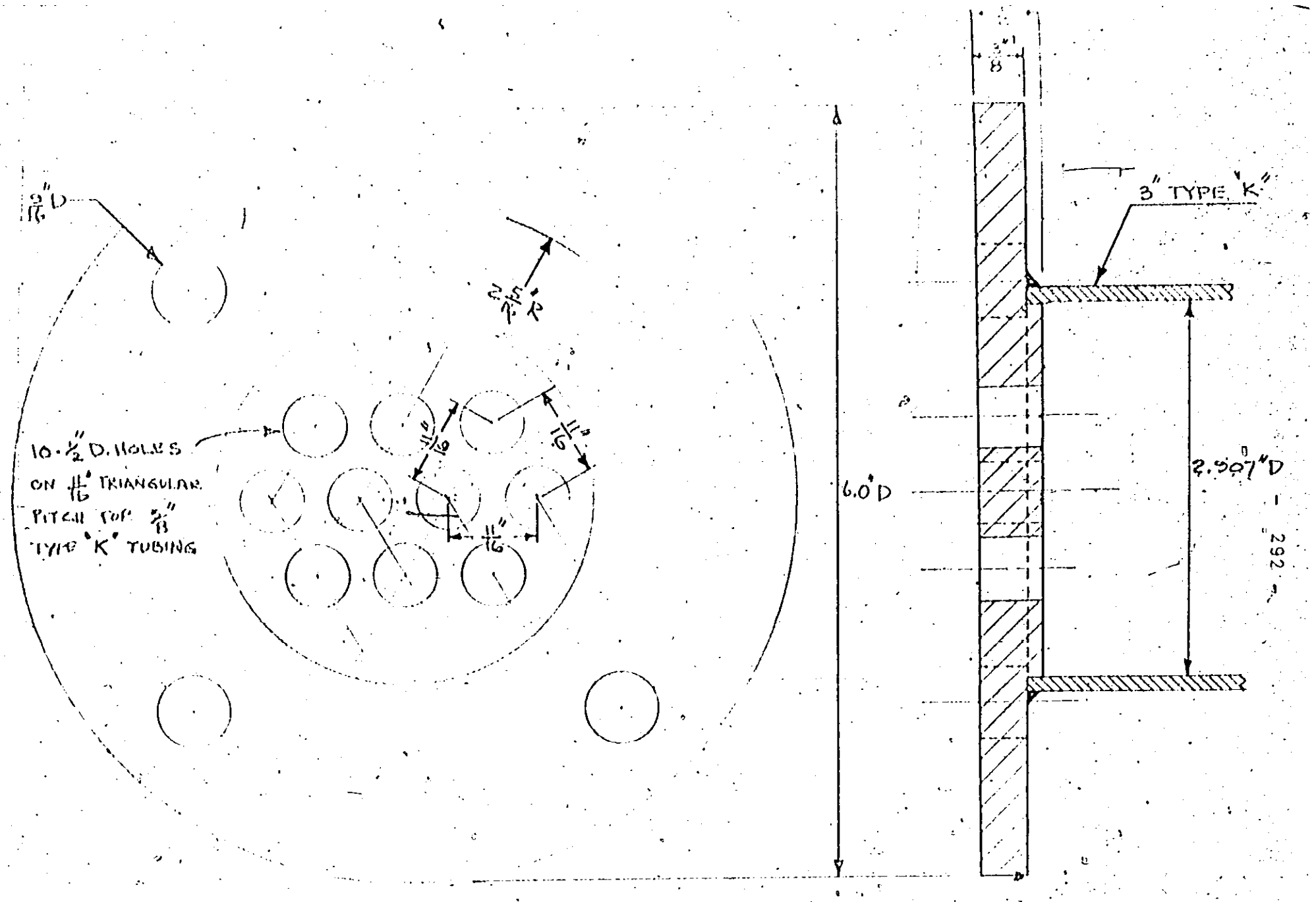


DETAILS OF CONDENSER "B"

(1 REQ'D)

APRIL 22/66

APRIL 22/66



* Note: Same tubesheet design used for bottom cooler and feed preheater.

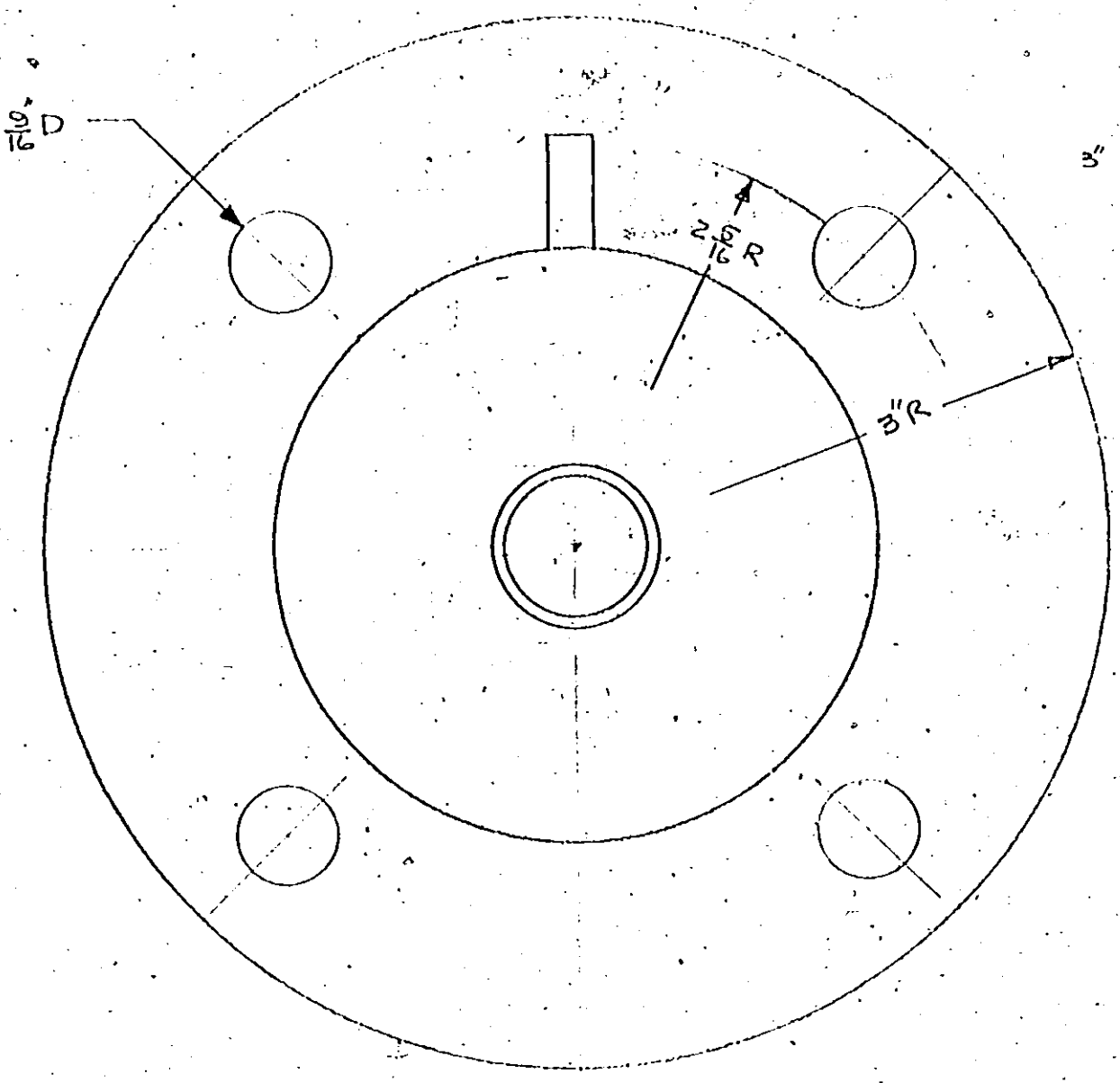
DETAILS OF CONDENSER TUBE SHEET

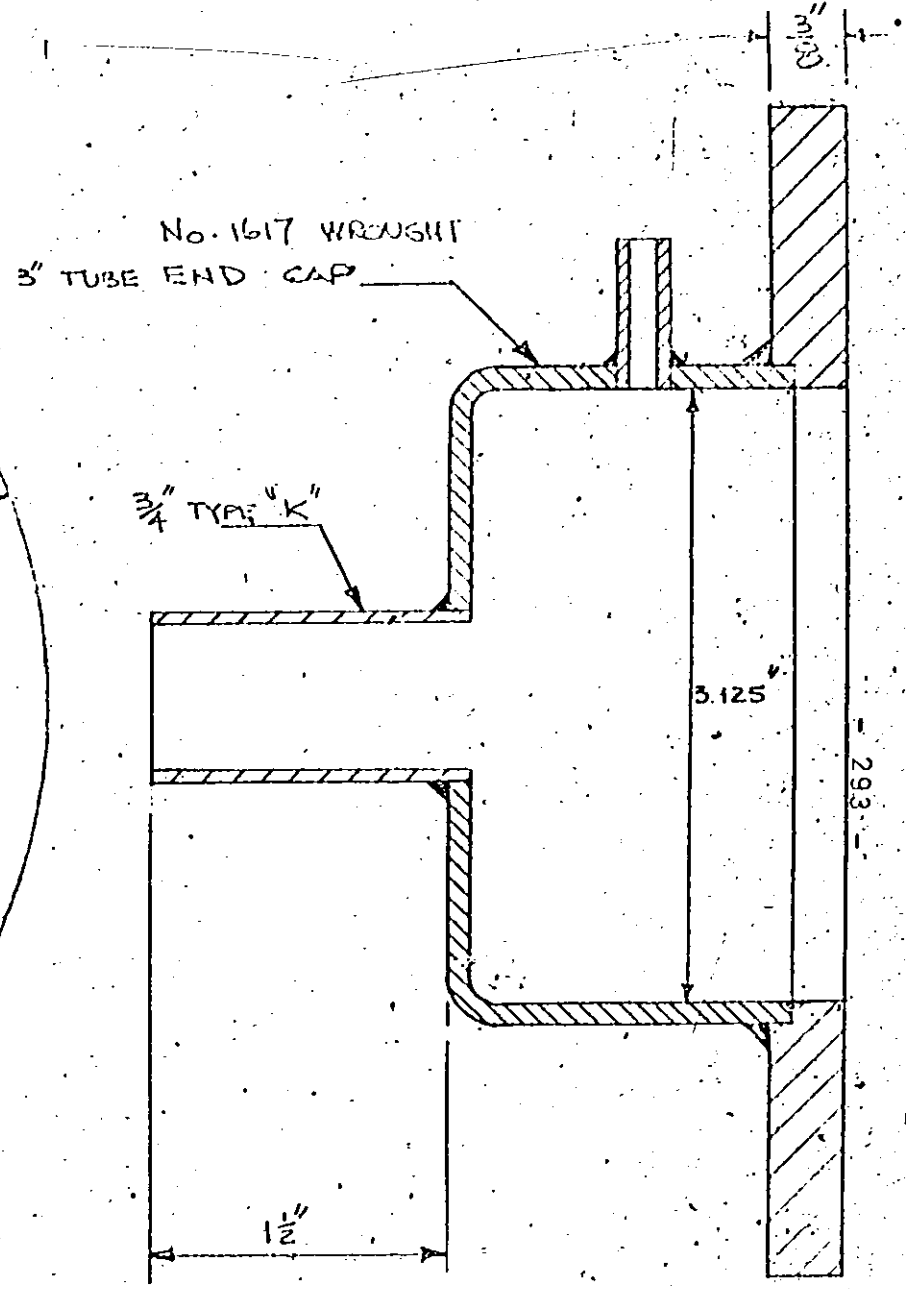
(4 REQ'D)

(4 REQ'D)

APRIL 22/66

APRIL 22/66



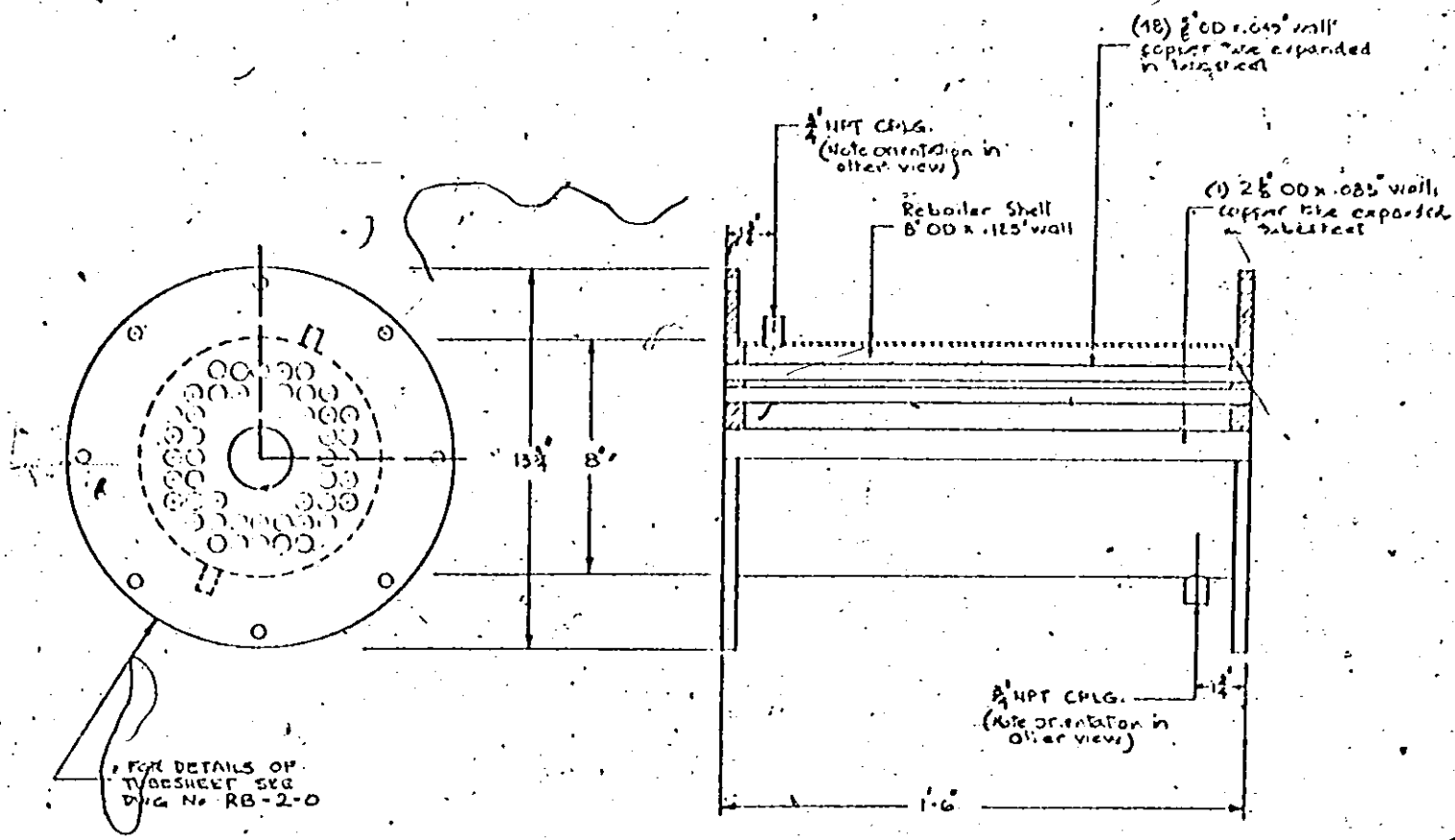


Note: Same head design used for bottoms cooler. Head for feed preheater made from steel.

DETAILS OF CONDENSER HEAD

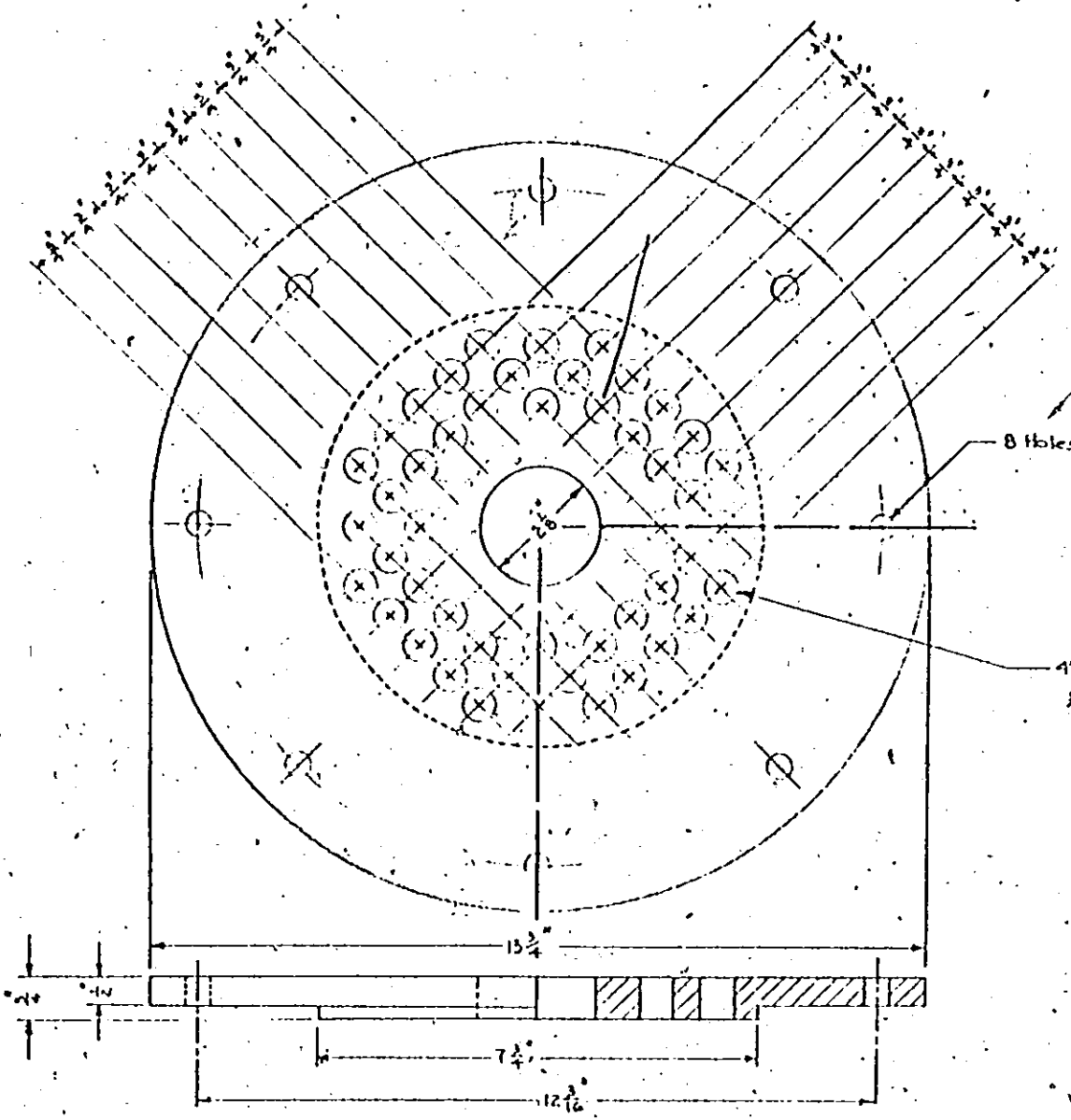
(4. REQ'D)

APRIL 22/64



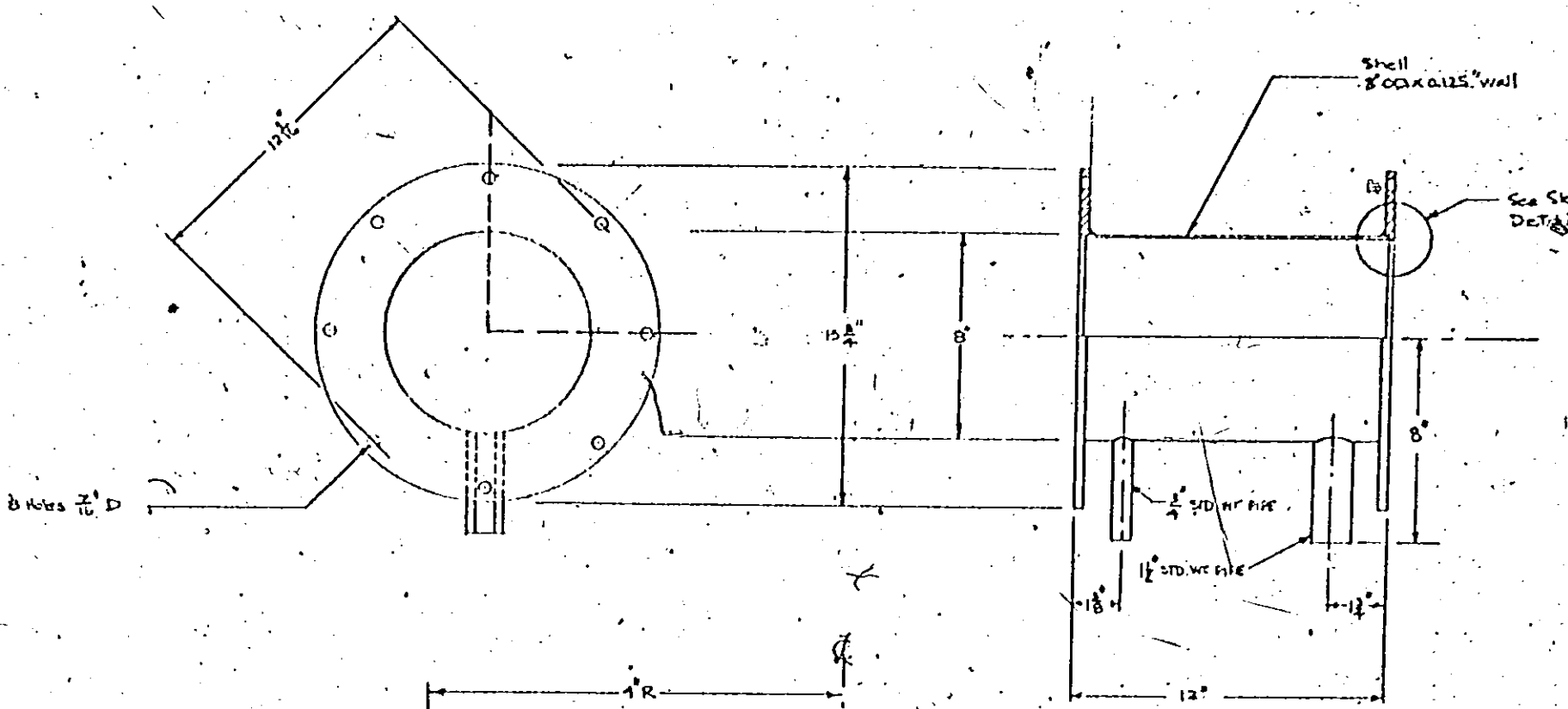
NOTES.
1. Tubesheet to be made of brass, all other parts to be copper.
2. Reboiler Shell Design 100 PSIG @ 250°F
Tube Design 25 PSIG @ 300°F

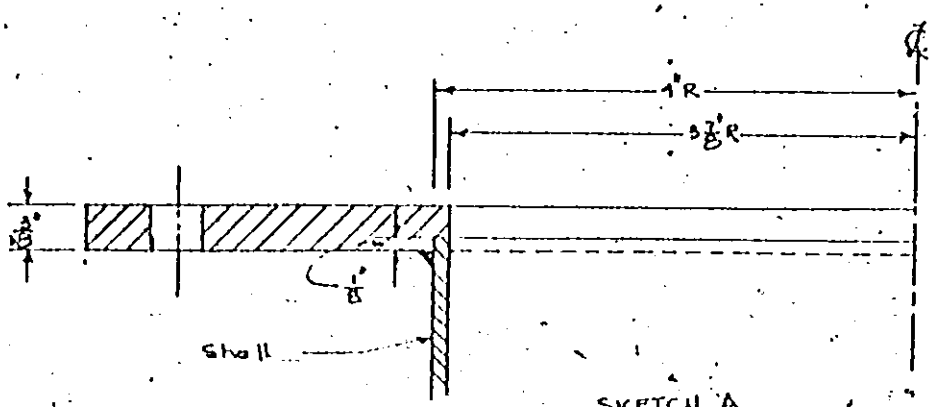
DETAILS OF TOWER REBOILER	
UNIVERSITY OF TEXAS	DEPT. CHEM. ENG.
SCALE 3/1	RB-1-0
DATE [illegible]	



DETAILS OF
REBOILER TUBESHEET

UNIVERSITY OF OTTAWA	DEPT. CHEM. ENG.
	PE-2-3





SKETCH 'A'

NOTES
1. Flanges to be made of brass, all other parts to be copper

DETAILS OF TOWER SECTION	
[illegible]	DRAWING NO.
[illegible]	TS-1-0

- 296 -