DYNAMIC SIMULATION OF BINARY DISTILLATION PROCESS

A DISSERTATION

Submitted in partial fulfilment of the
requirements for the award of the degree
of
MASTER OF ENGINEERING
in
ELECTRICAL ENGINEERING
(With Specialization in System Engineering and Operations Research)

By

VENGALA CHANDRAMOULY

DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE–247 667 (INDIA)

JANUARY, 2000

Gratis.

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation entitled "***DYNAMIC SIMULATION OF BINARY DISTILLATION PROCESS***" in the partial fulfillment of the requirements for the award of **Master of Engineering** in the **System Engineering & Operation Research**, submitted in the Department of **Electrical Engineering, University of Roorkee**, Roorkee, is an authentic record of my own work carried out under the guidance of **Prof. M.K. Vasantha**, Professor and **Dr. (Ms.) Indra Gupta**, Assistant Professor, Department of Electrical Engineering, University of Roorkee, Roorkee.

I have not submitted the matter embodied in this dissertation for the award of any other degree.

V. Chandramouly

Date : 31st January 2000 **(VENGALA CHANDRAMOULY)**

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of our knowledge and belief.

Indra
31.1.2000

Dr. (Ms.) Indra Gupta
Assistant Professor,
Department of Electrical Engineering,
University of Roorkee,
Roorkee-247667 (India)

M Vasantha
31.1.2000

Prof. M.K. Vasantha
Professor,
Department of Electrical Engineering,
University of Roorkee,
Roorkee-247667 (India)

ACKNOWLEDGEMENTS

I take this opportunity to express my sincere gratitude to Prof. M.K. Vasantha, Professor, and Dr. (Ms) Indra Gupta, Assistant Professor, Electrical Engineering Department, University of Roorkee, for encouraging me to undertake this dissertation as well as providing me all the necessary guidance and inspirational support throughout this work, without which this work would not have been in its present shape.

Special thanks to Dr. A.K. Pant and Dr. H.O. Gupta, Professors, Electrical Engineering Department, University of Roorkee, for encouraging, giving the moral support to do the work, and providing the excellent facilities of the Lab, at all times, whole-heartedly, without which it was impossible for me to complete the work.

I also take this opportunity to thank Mr. Sanjay Kumar Jain and Mr. Belachew B.G., Research Scholars, Electrical Engineering Department, for helping me in handling my computer with their experience and knowledge which gave momentum and made the dissertation work roll fast.

I am indebted to my colleagues Mr. P.C. Pant, Mr. Vinay Kumar Mangla, Mr. Vijander Singh and Ms. Asha Rani for taking interest in discussing the problem and encouraging.

I also thank Mr. Kalyan Singh and Mr. Rajendra Singh, Lab Technicians, Microprocessor & Computer Lab, for helping me in my work.

Special sincere heartfelt gratitude to my mother whose sincere prayer, best wishes, concern, support, unflinching encouragement and thoughtfulness has been a constant source of strength and upliftment to me during the entire work.

Last, but not the least, I thank all the members of Electrical Engineering Department for their help and encouragement at the hour of need.

VENGALA CHANDRAMOULY

ABSTRACT

The available models of Distillation Column of different complexities are studied. A multi-stage binary column and a multi-component multi-stage distillation column have been simulated separately. The simulation of binary distillation column assumes constant Murphree vapour efficiency, however the simulation of multi-component distillation column expects it as an input variable. Both the simulations are amalgamated in a menu driven user friendly software developed in C++ as the part of this dissertation. Also facility is provided in the above-developed software to observe various desired characteristics like concentration of distillate with respect to time, sample tray concentration with respect to time and tray temperature with respect to time for selected tray etc. For the simulation of distillation process rigorous model is considered. The simulation model has been tested for the binary mixture and the multi-component mixture reported in Luyben [15]. Results obtained by above simulation coincide with the results reported.

CONTENTS

Page No

NOMENCLATURE

A_{net}	Net area of the tray, m^2
A_l	Correction term in algebraic enthalpy balance of rigorous dynamic model
B	Bottom product rate, kmols/h
B_i	Correction term in enthalpy balance of rigorous dynamic model
c_i	Correction term in algebraic enthalpy balance of rigorous dynamic model
D	Distillate product rate kmols/h
Di	Correction term in algebraic enthalpy balance of rigorous dynamic model
Ei	Discrepancy function in enthalpy balance
F_i	Total feed flow rate into i^{th} tray kmols/h
f_{ij}	Component feed flow rate kmols/h
$f_j(T_i)$	Relationship between temperature and enthalpy
h_i	Total molar enthalpy of liquid mixture kj/kmol
hf_{ij}	Component feed enthalpy kj/kmol
hl_{ij}	Component liquid enthalpy kj/kmol
h_F	Total molar enthalpy of feed kj/kmol
h_v	Hght of the weir, m
hi	Total molar enthalpy of vapour kj/kmol
Hv_{ij}	Component vapour enthalpy kj/kmol
HCAPV	Heat capacity of vapour
HCAPL	Heat capacity of liquid
VP1	Vapour pressure at temperature T1
VP2	Vapour pressure at temperature T2
i_j	J^{th} colocation point
I	Objective function
K_{ij}	Equilibrium constant
Li	Total liquid flow rate leaving the tray, kmols/h

l_{ij}	Component liquid flow rate leaving the i^{th} tray kmols/h
l_w	Length of the weir, m
M_B	Liquid molar holdup in reboiler kmols
M_D	Liquid molar in reflux drum
Mi	Liquid molar holdup on i^{th} tray kmols
MD_i	Average molar density of liquid kmols/m^3
M_{il}	Liquid molar hold up at initial time kmols
M_{ij}	Discrepancy function in material balances
DENS	Density of component
NC	Number of components
NT	Total number of trays in distillation column
P_i	Total pressure on i^{th} tray, mm Hg
P_{ij}^{sat}	Pure component vapour pressure, mm Hg
Q_C	Condenser heat duty KJ/h
Q_B	Reboiler heat duty KJ/h
Q_{ij}	Discrepancy function in efficiency equilibrium relationship
R	Reflux ratio
t	Time, h
T_C	Condenser temperature in, °c
T_{Ei}	Experimentally measured temperature, °c
T_F	Temperature of the feed, °c
t_f	Time taken to reach final steady state, h
t_0	Initial time, h
T_{si}	Simulation temperature, °c
v_{ij}	Component vapour flow rate from the tray kmols/h
V_i	Total vapour flow rate from the tray kmols/h
x	Liquid composition of more volatile component, mole fraction
xij	Liquid composition if j^{th} component on i^{th} tray , mole fraction
x_{Fij}	Component liquid composition of j^{th} component in feed, mole fraction
y	Vapour composition of more volatile component, mole fraction
yij	Vapour composition of j^{th} component on i^{th} tray, mole fraction
y*	Equilibrium vapour composition of more volatile component, mole fraction

y_{ij}^*	Equilibrium vapour composition of j^{th} component on i^{th} tray, mole fraction
η_{ij}	Murphree stage efficiency
α	Relative volatility

Subscripts

i	Stage number
j	Component number
B,b	Related to reboiler

Superscripts

Sat	Saturation vapour efficiency

CHAPTER-1

INTRODUCTION TO DISTILLATION PROCESS

Distillation is one of the widely used energy intensive separation process in which a liquid or vapour mixture of two or more substances is separated into its component fractions of desired purity by the application and removal of heat. It can be classified into two ways namely Binary distillation and Multi-component distillation.

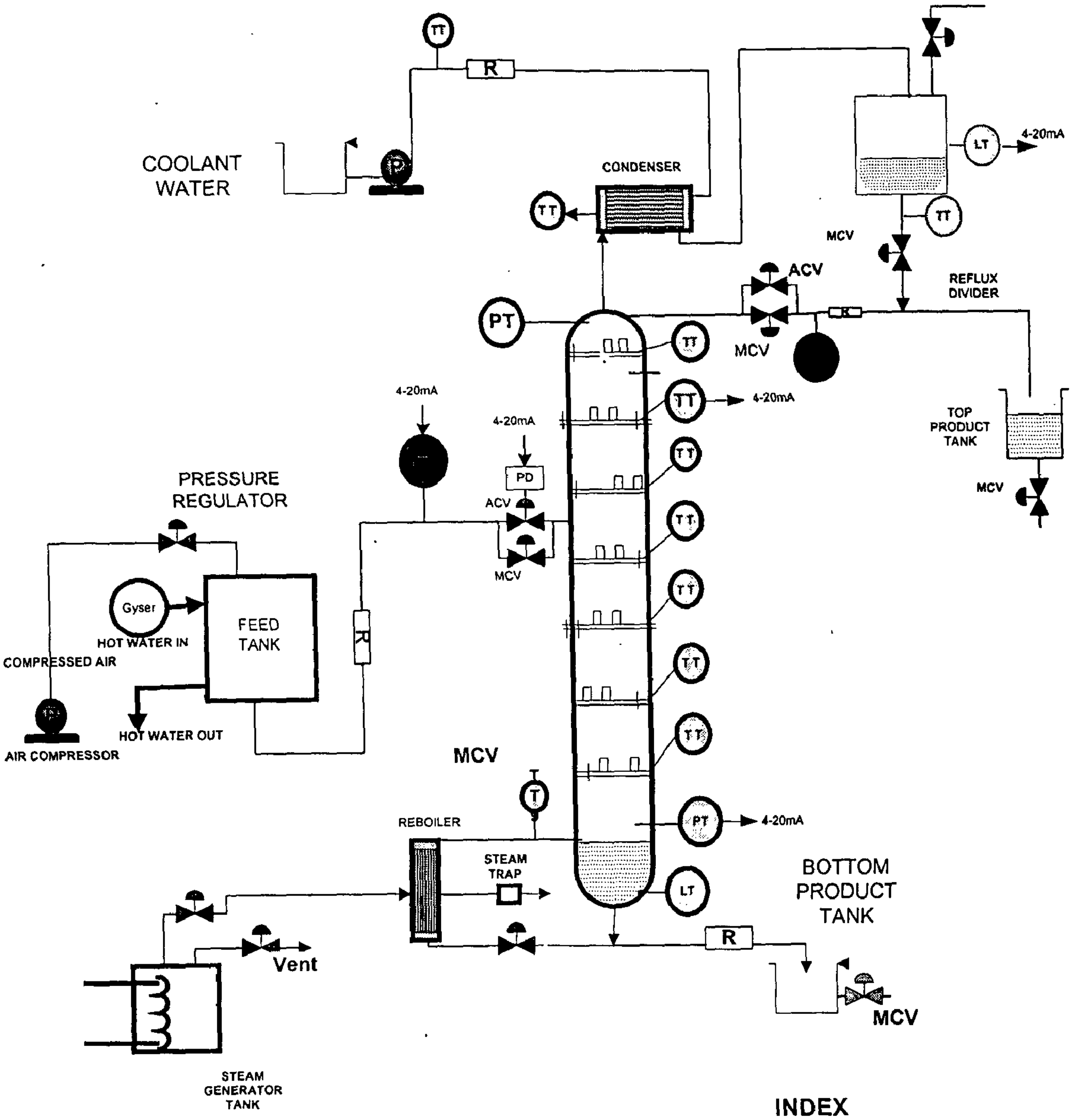
- **Binary distillation** It refers to separation of two substances
- **Multi-component distillation** - It involves the separation of more than two substances.

1.1 PRINCIPLE OF OPERATION OF DISTILLATION PROCESS

Pure liquids have different vapour pressures at a given temperature. On applying heat, vapours formed will have substances with large vapour pressure. These vapours are condensed and a pure liquid is obtained.

Basic steps in distillation operation consist of repeated contact of vapour and liquid phases on trays. The mixing of these phases result in heat and mass transfer. This transfer results in vapour, which is richer in the more volatile substances – those having higher pressures. If this vapour is condensed, certain amount of purification is achieved.

Distillation may be carried out as a batch operation or as a continuous operation. It may be affected, in continuous contact equipment (packed towers) or in stage wise contact towers. We are concerned with continuous distillation of binary mixtures in stage type distillation towers.



INDEX

TT	Temperature tramsmitter
FT	Flow Transmitter
DT	Density Transmitter
PT	Pressure Transmitter
LT	Level Transmitter
P	Pump
PD	Power Driver
ACV	Automatic control valve
MCV	Manual control valve
R	Rotometer

Fig 1. 1 Schematic Diagram of Distillation Column with Instrumentation and Control Component

The schematic diagram of a typical stage type distillation column is shown in Fig.1.1 and it consists of the following components:

- **A vertical shell** with a number of equally spaced trays mounted inside it.
- **Tray** contains two conduits, one on each side, called down comers. Liquid flows through these down comers by gravity from each tray to the one below. There are different types of trays :
  - *Sieve tray* - is a sheet of metal containing a number of perforations for vapour flow. Pressure should be high enough to prevent weeping of liquid through holes.
  - *Bubble cap tray* - has a riser or chimney fitted over each port and a cap covering riser. Vapours rise in riser, moved downwards by cap and out through slots in cap and finally bubbles out through liquid on tray.
  - *Valve tray* - has perforations covered with a lift able cap. Rising vapours lift the cap and the flow gets directed to the liquid horizontally thus allowing perfect mixing. It is highly efficient with wide operating range and cost factors.
- **A weir** on one side maintains the liquid level at a suitable height on that tray .
- **Reboiler** - The vertical shell is connected by suitable piping to a heating device called reboiler. It provides necessary vaporisation for the distillation operation.
- **Condenser** - There is another piping to a condenser, which condenses the overhead vapours.

For the instrumentation and control of parameters of Distillation Column various additional components are shown in Fig 1.1. These are :

- *Flow Transmitters (FT):* Four flow transmitters are employed for sensing the flow of feed and distillate.
- *Pressure Transmitters (PT)*: To sense the vapour pressure at the bottom and at the top of the distillation column.
- *Level transmitter (LT):* It is used to sense the level of the reflux drum.
- *Temperature transmitters (TT):* These are used for sensing the temperature at various places.
- *Manual control valves (MCV):* These are used for fixing up and controlling flow at various places in the column.
- *Rotometers (R):* These are used for measuring the velocity of drained fluid.
- *Automatic control valves (ACV):* These are used for automatic control of the process along with parallel to the manual control valves.

1.2 DISTILLATION CONTROL CONCEPT

In typical industrial situation, the disturbance in column occurs in one of many processing units in the plant and, therefore, it will be subjected to disturbances from the upstream processing units. Automatic controls are needed to counteract the effect of these disturbances.

For design of efficient control schemes which minimise energy consumption and also maintain good product quality, is a very crucial step because of the presence of complex interactions of the process variables in distillation columns. In order to control distillation process the number of controlled variables have to be determined. Intuitions suggests that variables as mentioned below -

- Product compositions
- Column pressure
- Base level and
- Reflux accumulator level

Should be controlled for the following reasons.

- Production objectives normally require the delivery of product of acceptable quality. This necessitates that product compositions be held constant.
- Column pressure has a direct bearing on the separation capability of the tower. Therefore, it is desirable to maintain constant column pressure.
- An adequate inventory in the column base is desirable because without it the reboiler tubes would be exposed to excessively high temperature if the flow of feed were to be interrupted.
- Finally, a suitable liquid level in the reflux drum dampens out the effect of unexpected surges in vapour boiler upon reflux and serves as a liquid seal.

Strategy

In order to control one or both product compositions in a distillation process, Feed F is separated into two products D and B. The overall material balance for this column may be expressed as

$$F = D + B \tag{1.1}$$

The component material balance for the more volatile substance, which we denote as substance A, is

$$F X_F = D X_d + B X_b \tag{1.2}$$

The term B in equation (1.2) may be replaced by F-D in accordance with equation (1.1) to give

$$F X_F = DX_d + (F - D) X_b$$

Or

$$\frac{D}{F} = \frac{X_F - X_b}{X_d - X_b} \qquad (1.3)$$

$$\text{and} \frac{B}{F} = \frac{X_d - X_F}{X_d - X_b} \qquad (1.4)$$

Equation (1.3) gives the unique steady-state relationship between D/F and X_d, X_b, X_F, while equation (1.4) gives the unique steady-state relationship between B/F and X_d, X_b, X_F. Typically the control objectives in a distillation operation are to maintain X_d and/or X_b at set point in the presence of disturbance. These disturbances may be characterised as

(1) Process loads

These disturbances consist of changes in flow rate, composition, and thermal condition of the feed to the disturbance column.

(2) Changes in heating and cooling medium supply conditions

Distillation process equipped with reboilers and condenser are subjected to disturbances arising from the changes in the supply conditions of the heating and cooling medium used in these heat-exchange devices.

(3) Equipment fouling

Heat exchange equipment fouls with extensive use. Since controller parameters are typically functions of process parameters, and since fouling can cause these parameters to change, the performance of the automatic control system associate with these devices can deteriorate with time if the controller parameters are not updated to account for the effect of fouling.

In distillation process the main control objectives are to maintain X_d and/or X_b constant in the presence of changes in F and X_F. Equation (1.3) and (1.4) completely describe the effect of changing D and/or V upon X_d and X_b. For a given upset in F or X_F, these equations will tell us how to control X_d and/or X_b, as in the following cases

(1) control of X_d or X_b for upsets in F

This is a single composition-control problem. It applies to those cases in which tight control of one of the two product compositions is deemed sufficient.

(2) control of X_d or X_b for upsets in X_F:

The control system for control of X_d for upsets in F must also be able to handle upsets in X_F or else a single feedback controller will not be able to hold the composition at set point for both types of disturbances.

On the basis of above description, we can classify the control schemes for distillation process. Even with the assumption that pressure is constant, there are four controlled variables (X_d, X_b, l_d, l_b, where h_d and h_b denote liquid level in the reflux drum and in the column base, respectively) and four manipulated variables (L_{N+1}, D, B, V_b). Thus there are four factorials or, 24 combinations, of these controlled and manipulated variables. In control jargon, D and B are called material-balance stream while L_{N+1} or V_b are called an energy-balance streams. A distillation control scheme that utilises either D or B for composition control is called a *material-balance control scheme* and one that uses L_{N+1} or V_b for composition control is called an *energy-balance control scheme*. The schemes in which D and B control liquid level in the reflux drum and in the column base fall into the category of *indirect material-balance control schemes,* whereas the others have been labelled *mixed control schemes.*

1.2.1 Pairing and Interactions in Distillation Column

In distillation process determining the correct Paring of controlled and manipulated variables in single composition control and dual composition control is one of the important aspect. In dual composition control, interaction among control loops can be a significant problem even when the variables are paired correctly. There are four controlled variables X_d, X_b, l_d, and l_b and four manipulated variables D, B, L_{NT+1}, and V. Thus there are 4! Or 24 possible ways of pairing these variables. In single-composition control, three of these controlled variables are connected to three manipulated variables and the fourth manipulated variable is not adjusted. In dual composition control, all four controlled variables are connected to the four manipulated variables. Table 1 shows all possible pairing for controlled and manipulated variables for single composition control.

The 24 possible combinations are shown in Table- 1 out of which eight of the pairings shown utilise material-balance streams D or B as the manipulated variable in the composition-control loop, therefore, these eight schemes are referred to as material-balance schemes. Eight of the schemes utilise energy-balance variables L_b or V and, therefore, are called energy-balance schemes. Four of the schemes are labelled indirect material-balance schemes in which he liquid levels h_d and h_b are controlled by D or B, which results in automatic control of inventory in the column. Finally, the remaining four schemes are called mixed schemes in which the composition is controlled by a material-balance stream while the other material-balance stream is not adjusted. The purpose of the exercise in this section is to assess which of the 24 schemes are suitable. Two factors that guide in this analysis are:

1. Steady-state gain, which is measure of how sensitive the output is to changes in the selected input.

2. Dynamic responsiveness, which is a measure of how quickly the output responds to changes in a given input.

TABLE 1.1 Possible Pairings for Controlled and Manipulated Variables Single Composition Control

Case No	Control Reflux accumulator	Control column base Level By	Control composition By	Variable Not manipulated	Classification of control schemes
1	D	L	B	V	Material balance
2	D	V	B	L	Material balance
3	L	D	B	V	Material balance
4	L	B	D	V	Material balance
5	B	L	D	V	Material balance
6	B	V	D	L	Material balance
7	V	D	B	L	Material balance
8	V	B	D	L	Material balance
9	D	L	V	B	Energy balance
10	D	V	L	B	Energy balance
11	L	D	V	B	Energy balance
12	L	B	V	D	Energy balance
13	B	L	V	D	Energy balance
14	B	V	L	D	Energy balance
15	V	D	L	B	Energy balance
16	V	B	L	D	Energy balance
17	D	B	V	L	Indirect material balance
18	D	B	L	V	Indirect material balance
19	B	D	L	V	Indirect material balance
20	B	D	V	L	Indirect material balance
21	L	V	D	B	Mixed
22	L	V	B	D	Mixed
23	V	L	B	D	Mixed
24	V	L	D	B	Mixed

1.2.2 Feed forward Control in Distillation Process

Normally, numerous types of disturbances, including those in process loads affect feedback control system. The notion of feed forward control is to measure the disturbance in process loads as they occur and to take corrective actions in the manipulated variables so that the controlled variable remains unaffected. Thus, to implement feed forward control, the disturbance variables must be measured.

Let $G_L(s)$, $G_p(s)$ and $G_c(S)$ are load, process and feedback transfer functions [Fig. 1.2(a)], then system response due to load disturbance L(s) can be written in mathematical form as

$$C(s) = \frac{G_L(s)}{1+G_c(s)G_p(s)} L(s) \qquad (1.5)$$

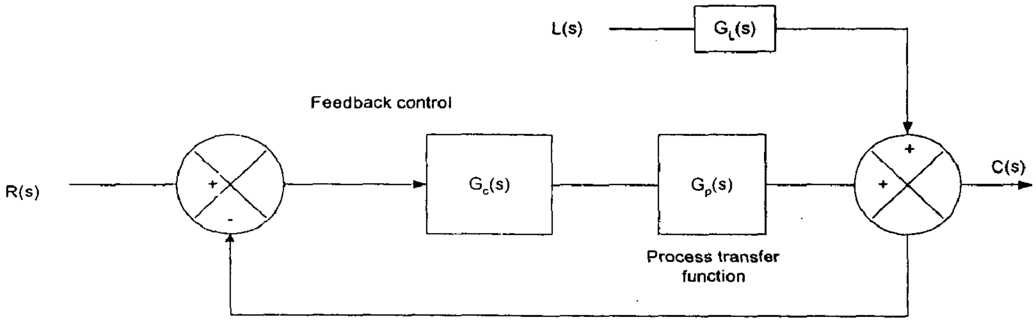


Fig 1.2 (a) feedback control system

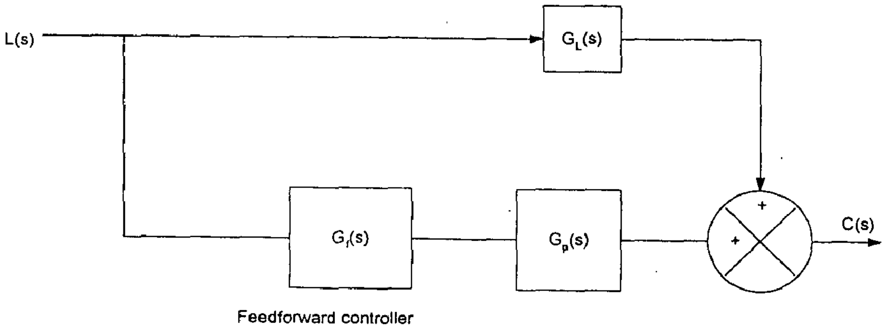


Fig 1.2 (b) feedforward control system

In Eq. (1.5), Gp(s) includes valve dynamics, both GL(s) and Gp(s) include the dynamics of the measuring element, and C(s), L(s) are in deviation form. Ideally C(s) should be zero for any L(s). The magnitude of C(s) is reduced by large controller gain term in Gc(s) but there is a limit on how large gain can be made because of physical limitations and system stability considerations. Thus a disturbance in L(s) upsets C(s), often significantly, if GL(s) and Gp(s) contain dead time elements and long legs.

The concept of feedforward control is shown in Fig. 1.2(b). The idea is to measure the disturbances as they occur and feed them to a controller that adjusts the manipulated variable in such a way that the controlled variable is unaffected. System response for feedforward can be written as

$$C(s) = [G_f(s)\,G_p(s) + G_L(s)]L(s) \tag{1.6}$$

If the quantity in the brackets is set equal to zero, then C will be zero for all L. This gives

$$G_f(s) = -\frac{G_L(s)}{G_p(s)} \tag{1.7}$$

In industrial practice feedforward control is seldom used alone, but rather in combination with feedback control, for the following reasons.

1. Control systems are often subjected to many disturbances. Thus if a feedforward control system is based on a particular disturbance and some other disturbance occurs, the control system will not be able to maintain the controlled variable at set point.

2. The feedforward controller design requires the knowledge of the process transfer function $G_p(s)$ and the load transfer function $G_L(s)$. If modelling is accurate, as is often the case, the term in the brackets in Eq. (1.6) will not vanish and the controlled variable will show a deviation from set point.

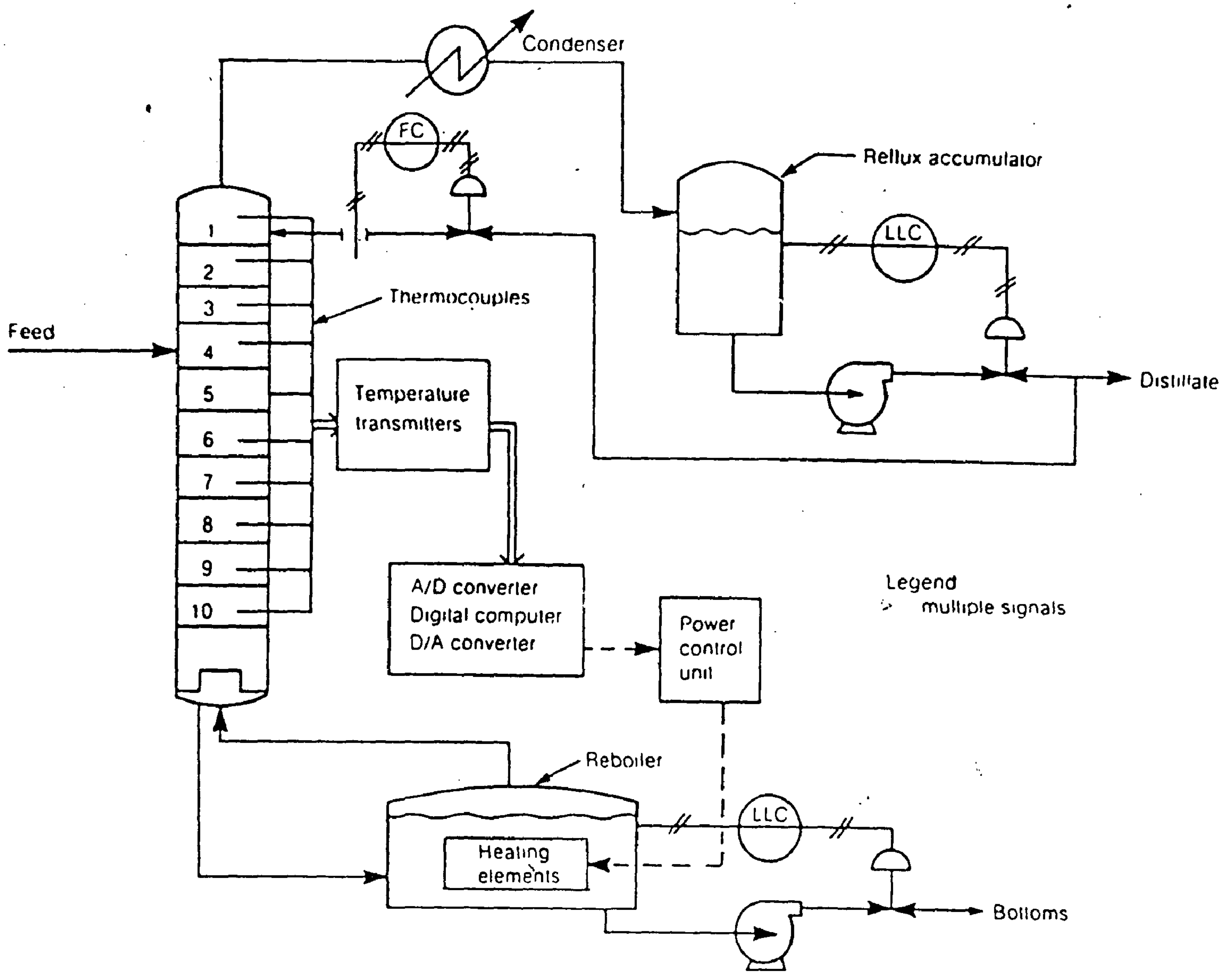


Fig 1.3 Schematic Diagram for Distillation Process Control

1.2.3 Inferential Control in Distillation Process

Typical Production Objectives in Distillation Process require the delivery of products whose compositions meet certain specifications. The distillation control system, therefore, must hold product compositions as near the set points as possible in the face of upsets. Distillation column is generally subjected to disturbances in the feed and the control of product quality is often achieved by maintaining a suitable tray temperature near its set point. A single-temperature feedback control is not always effective since maintaining a constant tray temperature does not necessarily result in constant product composition. The use of composition analyses often overcomes this problem. An alternative to measuring the composition is to estimate it from other process elements. The estimated composition then may be used in a control scheme to determine valve position directly, or it may be used to manipulate the set point of a temperature controller as in parallel cascade control. This is the notation behind inferential control developed Brosilow and coworkers (1978). The inferential control scheme uses measurements of secondary outputs, in this instance, selected tray temperatures, and manipulated variables to estimate the effect of unmeasured disturbances in the feed on product quality. The estimated product compositions are then used in a scheme to achieve improved composition control. Patke (1979) has applied the inferential control technique to single composition control of a laboratory-scale distillation column separating a mixture of n-propanol and methanol. A schematic of the column and the associated control system is shown in Figure (1.3). The objective of the control efforts is to control the methanol concentration in the bottom product by manipulating reboiler heat input. This is accomplished by measuring a set of suitable tray temperatures from which the bottom methanol composition is estimated and the controller output computed.

1.3 LITERATURE REVIEW

The importance of the study of distillation column dynamics was recognised a long time ago. Though the mathematical models required to simulate distillation processes have been known since late 1940' s early 1950' s several simplifications had to be built in these models in order to obtain solution because of the non availability of high speed digital computers. Pigford and Marshal (1947) were the first authors to solve linearized version of differential equations using Laplace transform technique. Rosenbrock (1958) modelled a general binary distillation column with several simplifications and compared the results obtained from various methods including (1) Laplace transforms (2) peturbation method (3) graphical method (4) anolog computer. He concluded that the solution obtained using the digital computer is to be preferred over the other methods because of the ease with which similar problems can be solved by simply changing a few parameters in the programme code.

Huckaba et al. (1963, 1965) removed most of the restrictive assumptions made by the previous authors. They modelled a general binary distillation system and used an algebraic energy balance to account for non-equimolal overflow. A simplified equilibrium relationship was used and the resulting set of equations was integrated on a digital computer using the Runge-Kutta technique. They simultaneously carried out experiments with a 12-plate column and found that the model satisfactorily represented the column operation when the assumptions were applicable.

Mah et al. (1962) and Sargent (1963) used a transition matrix method for solving problems related to general multistage process. This method used linearized equations and the linearized parameters in these equations were assumed to remain constant throughout the time step. Based on the condition at the next time step, the parameters were re-evaluated and the process was repeated.

Renfroe (1965) carried out experimental work by distilling mixture of benzene, toluene and n-heptene in a semi-commercial scale column. He made several step and pulse tests of fed composition to the column, operating under steady state conditions. He developed first order with dead time transfer function models to represent the experimental column response. Simulation on a small computer showed the same trends as the actual test but could not duplicate the concentration values observed in the column. Distefano (1968) studied many numerical methods for use in simulation multi-component distillation. He concluded that a modified predictor-corrector (The Adams-Moultan-shell procedure) was best suited for the solution of his distillation model.

Howard (1970) made an important contribution to dynamic simulation, by proposing a very general model, which allowed a wide choice of the variables specifications, required to define the column operation. Variable plate holds-ups, and efficiencies, heat losses from the column, and finite time liquid flow dynamics are some of the specifications considered. Howard also showed how to convert differential enthalpy balances to algebraic enthalpy balances. He used Kutta-Merson method to integrate ODEs of his model. He also carried out experiments on distillation of tenary mixer of benzene, toluene and ethyl benzene and tried to measure efficiencies and heat losses during the transient period. He concluded that efficiencies and heat losses obtained during the transient period were not satisfactory because of measurement errors. He observed that the inaccuracies in the computer simulation are due to inadequate information about the characteristics of the real column than to deficiencies of the computer model.

After 1970, most of the efforts have been to improve the efficiency of integration of the stiff differential equations since Runge-Kutta type methods require very small step size. Efforts have also been directed to build more complexities in the model so that computer simulation closely matches with the real time processes. Raemarker et al. (1975) made use of certain system parameters as indices that quantity the extent of the transient interactions. One important factor introduced is the Rijinsdrop factor, λ_1 defined as

$$\lambda_1 = \left[\frac{\partial L}{\partial V}\right]_{M_L} \tag{1.8}$$

Which is the partial derivative of liquid flow with respect to the vapour on a tray with constant liquid hold up, M_L. It has been found to accurately predict the vapour flow interactions in the column.

Luyben et al. (1974) made stiffness analysis of the dynamic model during the unsteady period as a prelude to develop efficient integration scheme. Ballard and Brosilow (1978) came up with an efficient integration scheme, called semi - implicit Runge-Kutta method with step size control. Holland et al. (1980), Sourisseau and Doherty (1980) and Bosten et al. (1981) proposed a modification of Gear's predictor-corrector method (1971). Seider and Prokopakis (1981, 1983) proposed adaptive Runge-Kutta (ASTRK) method, and compared it with semi-implicit and Gear methods. They concluded that their method allowed relatively larger step sizes and required less storage.THey also pointed out that ASIRK type methods are A-stable whereas the family of backward difference type formulate in the Gear method are just stiffly stable.

Gallun et al. (1982) and Hlland and Liapias (1983) discussed the generalised distillation problem. They used non-linear hold-up equations in both the liquid and the vapour phases, allowed pressure variation throughout the column. Murphree vapour phase efficiencies and non-ideal thermodynamic equilibrium relationships like Wilson equations were used in formulating their model. They solved resulting differential algebraic equations by modified semi-implicit Runge-Kutta and Gear methods. While they found Gear method to be faster they concluded that exact comparison could not be made because the two methods used different approaches. They also tested their model against plant data and found that it satisfactory represented the plant.

1.4 ORGANIZATIN OF DISSERTATION

Chapter 1 outlines the problem and describes the brief literature review of the distillation process. The dynamic models of distillation process have been described in chapter 2. Chapter 3 deals with the simulation of distillation process while chapter 4 presents the development of user friendly software package for distillation process. Simulation case studies involving binary distillation and multi component distillation process are discussed in chapter 5. Finally, chapter 6 outlines the main conclusions and suggests the scope for further work.

CHAPTER-2

DYNAMIC MODELS OF DISTILLATION PROCESS

Dynamic simulation of distillation is characterized by stiff set of differential algebraic equations (DAEs) of material and efficiency equilibrium relationships. The dynamic model to be developed in this work is expected to meet the following objectives:

(1) The model should be easy to use for the design of distillation column control schemes, online identification and optimization purposes.

(2) It is an open loop dynamic simulation and the response to any changes in the input should be obtained from this model in a reasonable amount of computer time.

In order to integrate dynamic model equations in time, a steady state solution is required. This can be obtained from Napthali-Sandholm steady state method or from parameter estimation by optimization. The dynamic model uses the following information directly from steady state model:

(1) Liquid composition on each tray
(2) Liquid flow rates from each tray
(3) Tray temperatures from each tray
(4) Condenser and reboiler duties
(5) Efficiencies

In Distillation column shown in fig. 1.1 the trays are numbered from bottom to top, first one being the reboiler and the last one being the condenser.

2.1 DYNAMIC MODEL

In developing a dynamic model of distillation process, the following assumptions have been made.

(1) Good mixing of each phase on each stage.

(2) No entrainment or dispersion in between the stages.

(3) Vapor and liquid streams leave each tray at equilibrium.

(4) Vapor holdup on the trays is negligible.

(5) Transportation delay of liquid and vapor between two trays is negligible.

(6) The column operates at atmospheric pressure and under adiabatic conditions.

(7) Murphree stage efficiency based on vapor phase is valid.

(8) Metal walls and the trays have negligible heat capacity, that means enthalpy of metal is not taken into consideration.

(9) For the present study, a total condenser and a partial reboiler are assumed to be used. The reflux drum and reboiler holdups are treated as well mixed pools.

(10) Finally there is a definite relationship between the concentration of a component in the vapor and liquid leaving each tray and it is given by bubble point calculation.

The mass and energy balance equations are obtained by applying conservation laws to each tray, condenser and reboiler.

2.1.1 Component Material Balance Equations

(a) Concentration for condenser

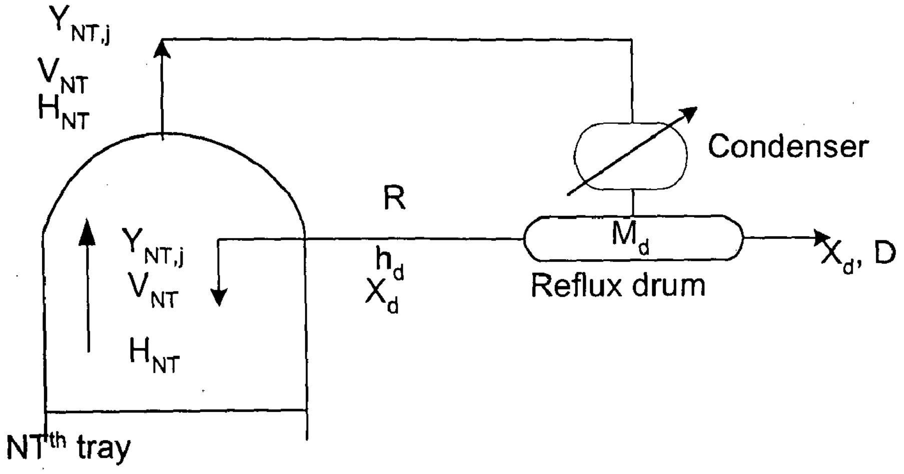


Fig 2.1 Modelling of Condenser

Where

M_D = liquid molar holdup in the reflux drum, kmols

D = distillate rate, kmols/h (=V_{NT}-R)

$X_{D,j}$ = liquid composition of jth component in reflux drum

$Y_{NT,j}$ = vapor composition of jth component leaving NTth tray

R = Total liquid flow rate into the NTth tray from reflux drum, kmols/h

V_{NT}= Total vapor flow rate leaving the NTth tray, kmols/h

The change in reflux drum level with time i.e. $\frac{dM_D}{dt} = 0$ because of the presence of reflux drum level controller. This means at any time

$$D=V_{NT}-R$$

Component material balance around condenser is given by

$$M_D \frac{dx_{D,j}}{dt} = V_{NT} y_{NT,j} - V_{NT} x_{D,j} \qquad .....2.1$$

for j=1,...........,NC

where, NC = number of components.

b) For general ith tray

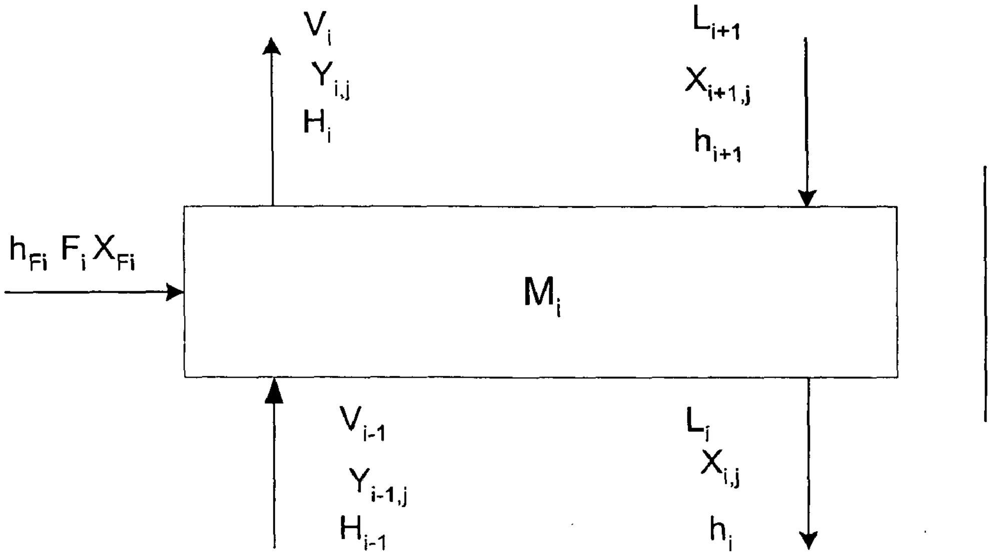


Fig 2.2 Modelling of general ith tray

where

M_i = molar liquid holdup on ith tray,kmols
x_{ij} = liquid composition of jth component leaving the ith tray
L_i = total liquid flow rate leaving ith tray, kmols/h
V_i = total vapor flow rate leaving Ith tray, kmols/h
F_i = total feed flow rate into ith tray, kmols/h
x_{Fij} = liquid composition of jth component in feed, kmols

Component material balance equation for ith tray is given by

$$\frac{d(M_i x_{ij})}{dt} = L_{i+1} x_{i+1,j} - L_i x_{ij} - V_i y_{ij} + V_{i-1} y_{i-1,j} + F_i x_{Fij} \qquad \text{......2.2}$$

for j= 1,..........,NC;   i= 1,...........NT

Where

$$y_{ij} = \eta_{ij}(y_{ij}^{*} - y_{i-1,j}) + y_{i-1,j} \qquad \text{.....2.3}$$

η_{ij} =murphree stage efficiency based on vapor phase of jth component on ith tray.

y_{ij}^{*} =equilibrium vapor composition

L_i is an additional variable and it is related to M_i through

$$L_i = 3.33 l_w [M_i/(A_{net}.M_{Di}) - h_w] \frac{3600}{2.204} M_{Di} \qquad \text{....2.4}$$

where

l_w=length of the weir,ft

A_{net}=net area of the tray,ft^2

h_w= height of the weir,ft

M_{Di}=average molar density of liquid on ith tray, kmols/ft^3.

L_i= liquid flow rate from ith tray, kmols/h.

c) Component Material balance for reboiler:

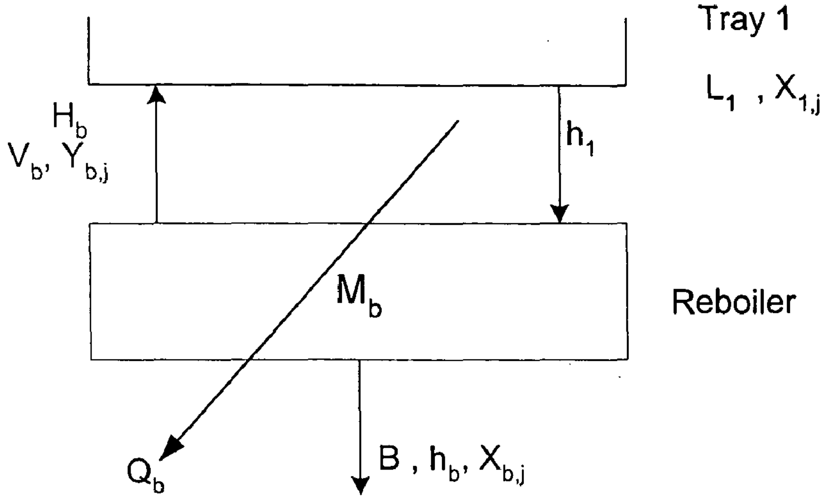


Fig 2.3 Modelling of Reboiler

where

M_B= liquid molar holdup in reboiler, kmols

L_1= total liquid flow rate entering reboiler, kmols/h

$X_{B,j}$= liquid composition of jth component in bottom product.

V_B= total vapor flow rate leaving reboiler, kmols/h

$Y_{B,j}$=vapor composition of jth component leaving reboiler

B= total bottom product rate, kmols/h

The change of reboiler holdup with time I.e $\frac{dM_B}{dt} = 0$,This means at any time,t

$B = L_1 - V_B$

Component material balance around reboiler is given by

$$M_B \frac{dx_{B,j}}{dt} = L_1 x_{1,j} - V_B y_{B,j} - (L_1 - V_B) x_{B,j} \qquad \text{....2.5}$$

$$y_{B,j} = \eta^{v}_{1,j} . k_{1,j} x_{B,j} \qquad ....2.6$$

$\eta^{v}_{1,j}$ = vaporization efficiency of j^{th} component in reboiler

$k_{1,j}$ = equilibrium constant of j^{th} component in reboiler

2.1.2 Total Material Balance Equation

For ith general tray

$$\frac{dM_i}{dt} = L_{i+1} - L_i - V_i + V_{i-1} + F_i \qquad ....2.7$$

for I=1,....................NT

2.1.3 Total Enthalpy Balance Equations:

(a) Enthalpy Balance for Condenser

$$M_D \frac{dh_D}{dt} = V_{NT} H_{NT} - V_{NT} h_D - Q_c \qquad ....2.8$$

where

H_{NT} = total molar enthalpy of vapor leaving the last tray kJ/kmol

h_D = total molar enthalpy of liquid leaving the reflux drum, kJ/kmol

Q_c = condenser duty, kJ/h

(b) Enthalpy balance for i^{th} general tray

$$\frac{d(M_i h_i)}{dt} = L_{i+1} h_{i+1} - L_i h_i - V_i H_i + V_{i-1} H_{i-1} + F_i h_{Fi} \qquad .....2.9$$

for i=1,...........,NT

where

h_i = total molar enthalpy of liquid leaving i^{th} tray, kJ/kmol

H_i = total molar enthalpy of vapor leaving i^{th} tray, kJ/kmol

(c) Enthalpy balance for reboiler

$$M_{B.}\frac{dh_B}{dt} = L_1 h_1 - V_B h_B - (L_1 - V_B)h_B + Q_B \qquad ....2.10$$

where

Q_B = reboiler duty , KJ/h

h_B = total molar enthalpy of liquid leaving reboiler

H_B = total molar enthalpy of vapor leaving reboiler, kJ/kmol

Enthalpies on any tray are calculated by mixing rule as given by

$$h_i = \sum_{j=1}^{NC} hl_{ij} x_{ij} \, , \qquad .....2.11$$

NC

(2) **Semi-Rigorous Model:**

In this model it is assumed that energy dynamics is so rapid that the enthalpy balance reduces to an algebraic equation. This means thermal equilibrium is achieved much faster than phase equilibrium.

(3) **Simple Model:**

In this model one more assumption of constant total holdup is made ($M_{si}=M_i$ moles on each tray, where M_{si} is the initial steady state holdup). This means that $\frac{dM_i}{dt}=0$. Then all the total material balance equations of the model become algebraic in addition to enthalpy equations, which are already algebraic.

2.3 DEVELOPMENT OF RIGOROUS DYNAMIC MODEL:

Differential equations comprising of components and total material balances, total enthalpy balances and efficiency equilibrium relationships represent the rigorous model. But the enthalpy balances are not independent differential equations and they can be reduced to algebraic equations.

2.3.1 Reduction Of Differential Enthalpy Balance Equation To Algebraic Balance Equations:

(a) To reduce enthalpy balance for ith tray to algebraic equation. Consider component material balance for ith tray as given by equation (2.2).

$$\frac{d(M_i x_{ij})}{dt} = M_i \frac{dx_{ij}}{dt} + x_{ij}\frac{dM_i}{dt} = L_{i+1}x_{i+1j} - L_i x_{ij} - V_i y_{ij} + V_{i+1}y_{i+1j} + F_i x_{Fij} \quad ..2.13$$

Total material balance for ith tray is given by

$$\frac{dM_i}{dt} = L_{i+1} - L_i - V_i + V_{i-1} + F_i \qquad \text{.....2.14}$$

Multiply equation (2.14) by x_{ij} and substitute in equation (2.13) to get

$$M_i \frac{d(x_{ij})}{dt} = V_{i-1}\left(y_{i-1j} - x_{ij}\right) + L_{i+1}\left(x_{i+1j} - x_{ij}\right) - V_i\left(y_{ij} - x_{ij}\right) + F_i\left(x_{Fij} - x_{ij}\right) \qquad \text{.....2.15}$$

We know that

$$h_i = \sum_{j=1}^{NC} hl_{ij} x_{ij} = \sum_{j=1}^{NC} f_j(T_i) x_{ij} \qquad \text{.......2.16}$$

On differentiation of equation (2.16) with respect to t we get

$$\frac{dh_i}{dt} = \sum_{j=1}^{NC} x_{ij} \left[\frac{d\{f_j(T_i)\}}{dT_i}\right] + \sum_{j=1}^{NC} f_j(T_i) \frac{dx_{ij}}{dt} \qquad \text{........2.17}$$

$$\frac{d\{f_j(T_i)\}}{dt} = \left[\frac{d\{f_j(T_i)\}}{dT_i}\right]\left[\frac{dT_i}{dt}\right] = \frac{d\{f_j(T_i)\}}{dT_i} \sum_{j=1}^{NC} \left[\frac{\partial T_i}{\partial x_{ij}}\right] \frac{dx_{ij}}{dt} \qquad \text{........2.18}$$

$\dfrac{\partial T_i}{\partial x_{ij}}$ derivatives will be considered later

Multiply equation (2.17) with M_i and then substitute equation (2.18) to get

$$M_i \frac{dh_i}{dt} = \sum_{j=1}^{NC} x_{ij} \left[\frac{d\{f_j(T_i)\}}{dT_i}\right] \sum_{j=1}^{NC} \left[\frac{\partial T_i}{\partial x_{ij}}\right] M_i \frac{dx_{ij}}{dt} + \sum_{j=1}^{NC} f_j(T_i) M_i \frac{dx_{ij}}{dt} \qquad \text{.......2.19}$$

On substitution of equation (2.15) into equation (2.19) for $M_i \dfrac{dx_{ij}}{dt}$ and after rearrangement one can get

$$M_i \frac{dh_i}{dt} = V_{i+1}\left\{\sum_{j=1}^{NC} f_j(T_i)(y_{i-1j} - x_{ij}) + \sum_{j=1}^{NC} x_{ij}\left[\frac{d\{f_j(T_i)\}}{dT_i}\right]\sum_{j=1}^{NC}\left[\frac{\partial T_i}{\partial x_{ij}}\right](y_{i-1j} - x_{ij})\right\}$$

$$+ L_{i-1}\left\{\sum_{j=1}^{NC} f_j(T_i)(x_{i+1j} - x_{ij}) + \sum_{j=1}^{NC} x_{ij}\left[\frac{d\{f_j(T_i)\}}{dT_i}\right]\sum_{j=1}^{NC}\left[\frac{\partial T_i}{\partial x_{ij}}\right](x_{i+1j} - x_{ij})\right\}$$

$$- V_i\left\{\sum_{j=1}^{NC} f_j(T_i)(y_{ij} - x_{ij}) + \sum_{j=1}^{NC} x_{ij}\left[\frac{d\{f_j(T_i)\}}{dT_i}\right]\sum_{j=1}^{NC}\left[\frac{\partial T_i}{\partial x_{ij}}\right](y_{ij} - x_{ij})\right\}$$

$$+ F_i\left\{\sum_{j=1}^{NC} f_j(T_i)(x_{Fij} - x_{ij}) + \sum_{j=1}^{NC} x_{ij}\left[\frac{d\{f_j(T_i)\}}{dT_i}\right]\sum_{j=1}^{NC}\left[\frac{\partial T_i}{\partial x_{ij}}\right](x_{Fij} - x_{ij})\right\}$$

....(2.20)

Total enthalpy balance given by equation (2.9) is written as

$$\frac{d(M_i h_i)}{dt} = M_i \frac{dh_i}{dt} + h_i \frac{dM_i}{dt} = L_{i+1}h_{i+1} - L_i h_i - V_i H_i + V_{i-1}H_{i-1} + F_i h_{Fi} \qquad ....2.21$$

substitute equation (2.14) into equation (2.21) and rearrange to get

$$M_i \frac{dh_i}{dt} = L_{i+1}(h_{i+1} - h_i) - V_i(H_i - h_i)H_i + V_{i-1}(H_{i-1} - h_i) + F_i(h_{Fi} - h_i) \qquad .....2.22$$

statements in equations (2.20) and (2.22) are equivalent and may be combined to give the final equation for the energy balance as

$$V_{i-1}[(H_{i-1} - h_i) - A_i] - L_{i+1}[(h_{i+1} - h_i) - B_i] - V_i[(H_i - h_i) - C_i] + F_i[(h_{Fi} - h_i) - D_i] = 0$$

...2.23

in which A_i, B_i, C_i & D_i are coefficients of V_{i-1},L_{i+1},V_i and F_i respectively in equation (2.20).equation (2.20) is enthalpy balance equation of Ith tray in algebraic form.

(b) Reduction of differential enthalpy balance equation of condenser to algebraic equation:

Consider the total enthalpy balance developed in equation (2.8)

$$M_D \frac{dh_D}{dt} = V_{NT} H_{NT} - V_{NT} h_D - Q_c \qquad \text{.........2.24}$$

Component material balance equation of condenser

$$M_D \frac{dx_{D,j}}{dt} = V_{NT}\left(y_{NT,j} - x_{D,j}\right) \qquad \text{........2.25}$$

Also

$$h_D = \sum_{j=1}^{NC} hl_{D,j} x_{D,j} = \sum_{j=1}^{NC} f_j(T_D) x_{D,j} \qquad \text{....2.26}$$

Multiply equation (1.26) with M_D and then differentiate this equation with respect to t and follow the same procedure as shown from equation (2.17) to (2.18) to get

$$M_D \frac{dh_D}{dt} = \sum_{j=1}^{NC} x_{D,j} \frac{d\{f_j(T_D)\}}{dT_D} \sum_{j=1}^{NC}\left[\frac{\partial T_D}{\partial x_{D,j}}\right] M_D \frac{dx_{D,j}}{dt} + \sum_{j=1}^{NC} f_j(T_D) M_D \frac{dx_{D,j}}{dt} \qquad \text{...2.27}$$

On substitution of equation (2.25) into equation (2.27) and on rearrangement one can get

$$\begin{aligned} M_D \frac{dh_D}{dt} &= V_{NT}\left\{\sum_{j=1}^{NC} x_{D,j}\left[\frac{d\{f_j(T_D)\}}{dT_D}\right]\sum_{j=1}^{NC}\left[\frac{\partial T_D}{\partial x_{D,j}}\right]\left(y_{NT,j} - x_{D,j}\right)\right\} \\ &+ \sum_{j=1}^{NC} f_j(T_D)\left(y_{NT,j} - x_{D,j}\right) \end{aligned} \qquad \text{.........2.28}$$

Statements in equation (2.24) and (2.28) are equivalent and these may be combined to give

$$V_{NT}\left(H_{NT} - h_D - A_D\right) - Q_c = 0 \qquad \text{.........2.29}$$

In which A_D is coefficient of V_{NT} in equation (2.28). Equation (2.29) is enthalpy balance equation of condenser in algebraic form.

(C) Reduction of differential enthalpy balance equation of reboiler to an algebraic equation:

Consider the total enthalpy balance of reboiler in equation (2.10) once again

$$M_B \frac{dh_B}{dt} = L_1 h_1 - V_B H_B - \left(L_1 - V_B\right) + Q_B \qquad \text{......2.30}$$

Also consider component material balance equation (2.5) once again

$$M_B \frac{dx_{B,j}}{dt} = L_1 x_{1,j} - V_B y_{B,j} - \left(L_1 - V_B\right) x_{B,j} \qquad \text{..........2.31}$$

Differentiate $M_B h_B = \sum_{j=1}^{Nc} hl_B x_{B,j}$ with respect to t and follow the same procedure

explained from equation (2.17) to (2.18) to get

$$M_B \frac{dh_B}{dt} = \sum_{j=1}^{Nc} x_{B,j} \left[\frac{d\left\{f_j(T_B)\right\}}{dT_B}\right] M_B \frac{dx_{B,j}}{dt} + \sum_{j=1}^{NC} f_j\left(T_B\right) M_B \frac{dx_{B,j}}{dt} \qquad \text{........2.32}$$

On substitution of equation (2.31) to equation (2.32) and on rearrangement one can get

$$\begin{aligned} M_B \frac{dh_B}{dt} = {} & L_1 \left\{ \sum_{j=1}^{NC} x_{B,j} \left[\frac{d\left\{f_j(T_B)\right\}}{dT_B}\right] \sum_{j=1}^{NC} \left[\frac{\partial T_B}{\partial x_{B,j}}\right] \left(x_{1,j} - x_{B,j}\right) + \sum_{j=1}^{NC} f_j\left(T_B\right)\left(x_{1,j} - x_{B,j}\right) \right\} \\ & - V_B \left\{ \sum_{j=1}^{NC} x_{B,j} \left[\frac{d\left\{f_j(T_B)\right\}}{dT_B}\right] \sum_{j=1}^{NC} \left[\frac{\partial T_B}{\partial x_{B,j}}\right] \left(y_{B,j} - x_{B,j}\right) + \sum_{j=1}^{NC} f_j\left(T_B\right)\left(y_{B,j} - x_{B,j}\right) \right\} \end{aligned} \qquad \text{......2.33}$$

The equation (2.33) and equation (2.30) are equivalent and these may be combined to give

$$L_1\left[\left(h_1 - h_B\right) - B_B\right] - V_B\left[\left(H_B - h_B\right) - C_B\right] + Q_B = 0 \qquad \text{......2.34}$$

In equation (2.34) B_1 and C_1 are coefficients of L_1 and V_B respectively of equation (2.33). Equation (2.34) is enthalpy balance equation of reboiler in algebraic form.

2.3.2 To Determine $\frac{\partial T_i}{\partial x_{ij}}$:

The composition of the j^{th} component in liquid and vapour leaving the i^{th} tray are related through bubble point equation:

$$\sum_{j=1}^{NC} y_{i,j} = 1 \qquad \text{....2.35}$$

for i=1,........, NT

i.e.

$$\sum_{j=1}^{NC} k_{i,j}\left(x_{i,1},..............,x_{i,NC},T_i\right)x_{i,j} = 1 \qquad \text{..........2.36}$$

or

$$\sum_{j=1}^{NC} x_{i,j}\gamma_{i,j}p_{i,j}^{sat} = P_i \qquad \text{..........2.37}$$

where, $k_{i,j}$ = equilibrium constant of j^{th} component on i^{th} tray

$\gamma_{i,j}$ = activity coefficient of j^{th} component on i^{th} tray

$p_{i,j}^{sat}$ = pure component vapor pressure of j^{th} component on i^{th} tray

P_i = total pressure on i^{th} tray

Here $k_{i,j}$ is considered to be function of both liquid compositions $x_{i,1}$,.......,$x_{i,NC}$ and temperature T_i at the specified pressure P_i . Various partial derivatives of one variable with respect to another which are defined by an implicit function may be determined by the following chain rule method illustrated by Mickley ,Sherwood and Reed (1957)

By writing left side of equation (2.36) as a general function of z_i of all variables , the designed partial derivatives can be determined as :

$$\frac{\partial T_i}{\partial x_{i,j}} = -\left(\frac{\partial z_i}{\partial x_{i,j}}\right)\Bigg/\left(\frac{\partial z_i}{\partial T_i}\right) \qquad \text{......2.38}$$

This expression can be evaluated from bubble point relationship. Note that because the sum of the mole fraction must be equal to unity, only NC-1 compositions are independent. The bubble point relationship should be written as:

$$\sum_{j=1}^{NC-1} k_{i,j}(x_{i,1},\ldots\ldots,x_{i,NC},T_i) + k_{i,NC}(x_{i,1},\ldots\ldots\ldots,x_{i,NC-1},T_i)\left[1-\sum_{j=1}^{NC-1} x_{i,j}\right] = 1 \qquad \ldots.2.39$$

$$Z_i(x_{i,1},\ldots,x_{i,NC-1},T_i) = \left\{\sum_{j=1}^{NC-1}\left[k_{i,j}(x_{i,1},\ldots x_{i,NC-1},T_i) - k_{i,NC}(x_{i,1},\ldots,x_{i,NC-1},T_i)\, x_{i,j}\right]\right\} \qquad \ldots..2.40$$
$$+\{k_{i,NC}(x_{i,1},\ldots,x_{i,NC-1},T_i)\} - 1 = 0$$

Partial derivatives can be evaluated using equation (2.38) and (2.40)

2.3.3 Statement of Final Set of Equations for Rigorous Dynamic Simulation:

1) Component material balance equations:

a) for condenser:

$$M_D \frac{dx_{D,j}}{dt} = V_{NT} y_{NT,j} - V_{NT} x_{D,j} \qquad \ldots\ldots\ldots\ldots.2.41$$

for j=1,..........,NC

b) For ith general tray:

$$\frac{d(M_i x_{ij})}{dt} = L_{i+1} x_{i+1,j} - L_i x_{ij} - V_i y_{ij} + V_{i-1} y_{i-1,j} + F_i x_{Fij} \qquad \ldots..2.42$$

for j= 1,..........,NC; i= 1,...........NT

Where

$$y_{ij} = \eta_{ij}(y^*_{ij} - y_{i-1,j}) + y_{i-1,j}$$

η_{ij} = murphree stage efficiency based on vapor phase of jth component on ith tray.

y_{ij}^* = equilibrium vapor composition

c) For reboiler

$$M_B \frac{dx_{B,j}}{dt} = L_1 x_{1,j} - V_B y_{B,j} - (L_1 - V_B) x_{B,j} \qquad \text{..........2.43}$$

$$y_{B,j} = \eta_{1,j}^v . k_{1,j} x_{B,j} \qquad \text{..........2.44}$$

$\eta_{1,j}^v$ = Vaporization efficiency of jth component in reboiler

$k_{1,j}$ = Equilibrium constant of jth component in reboiler

Here $\eta_{1,j}^v$ denotes the vaporization efficiency, since Murphree tray efficiency can not be defined for reboiler in the usual sense. Often it is assumed to be one. For making material balance equations, one must start from reboiler upwards since in order to calculate $y_{i,j}$, it is necessary to know $y_{I-1,j}$. Vapor composition leaving first tray can be calculated from equation (2.43). since both $y_{B,j}$ and $\eta_{1,j}$ are known. The same calculation procedure can be used for finding $y_{i,j}$ when i=1,......,NT-1.

2) Total material balance for ith general tray

$$\frac{dM_i}{dt} = L_{i+1} - L_i - V_i + V_{i-1} + F_i \qquad \text{..........2.45}$$

3) Total enthalpy balance rearranged to express vapor flow rates in explicit form

a) for reboiler

$$V_B = \frac{[Q_B + L_1(h_1 - h_B - B_B)]}{[H_B - h_B - C_B]} \qquad \text{..........2.46}$$

b) For ith general tray

$$V_i = \frac{\{V_{i-1}[(H_{i-1} - h_i) - A_i] + L_{i-1}[(H_{i-1} - h_i) - B_i] + F_i[(h_{Fi-1} - h_i) - D_i]\}}{[(H_{i-1} - h_i) - A_i]} \quad \text{.......2.47}$$

for i=1,.....NT

4) Relationship for L_i in terms of M_i and M_{Di}

$$L_i = 3.33 l_w [M_i / (A_{net} . M_{Di}) - h_w] \frac{3600}{2.204} M_{Di} \quad \text{.............2.48}$$

for i= 1,......,NT

5) Bubble point relationship

$$\sum_{j=1}^{NC-1} k_{i,j}(x_{i,1}, \ldots\ldots, x_{i,NC}, T_i) + k_{i,NC}(x_{i,1}, \ldots\ldots\ldots, x_{i,NC-1}, T_i)\left[1 - \sum_{j=1}^{NC-1} x_{i,j}\right] = 1 \quad \text{......2.49}$$

Equation (2.41) to (2.49) form the rigorous dynamic model.

2.4 DEVELOPMENT OF SEMI-RIGOROUS DYNAMIC MODEL:

With the assumption that thermal equilibrium is achieved faster i.e. the enthalpy dynamics is very rapid, left hand side terms of equation (2.8) to (2.10) can be dropped and total enthalpy balance equations become algebraic. However component and total material balances remain the same as equation (2.41) to (2.45).

2.4.1 Total Enthalpy Balance Equations of Semi-rigorous Dynamic Model:

With the above simplification equations (2.47) and (2.48) are replaced by the following set of equations (2.50) and (2.51).

a) For reboiler:

$$V_B = \frac{[Q_B + L_1(h_1 - h_B)]}{[H_B - h_B]} \quad \text{......2.50}$$

b) For general i^th^ tray

$$V_i = \frac{(V_{i-1}H_{i-1} - L_i h_i + L_{i+1}h_{i+1} + F_i h_{Fi})}{H_i} \qquad ....2.51$$

for i=1,.......,NT

2.5 DEVELOPMENT OF SIMPLE DYNAMIC SIMULATION MODEL:

With the assumption that molar hold ups do not change with time and remain at steady state values , the total material balance equations become algebraic. Already time independent enthalpy assumption has made total enthalpy balance algebraic.

2.5.1 Component Material Balance Equations:

a) Condenser

$$M_D \frac{dx_{D,j}}{dt} = V_{NT} y_{NT,j} - V_{NT} x_{D,j} \qquad ....2.52$$

for j=1,......,NC

b) General i^th^ tray

$$\frac{d(M_i x_{ij})}{dt} = L_{i+1} x_{i+1,j} - L_i x_{ij} - V_i y_{ij} + V_{i-1} y_{i-1,j} + F_i x_{Fij} \qquad ....2.53$$

where

$$y_{ij} = \eta_{ij}(y^*_{ij} - y_{i-1,j}) + y_{i-1,j} \qquad ....2.54$$

for i=1,...,NT j=1,....,NC

b) for reboiler:

$$M_B \frac{dx_{B,j}}{dt} = L_1 x_{1,j} - V_B y_{B,j} - (L_1 - V_B) x_{B,j} \qquad ....2.55$$

for j=1,....,NC

2.5.2 Total Material And Energy Balances:

These are rearranged to get

$$V_{NT} = \frac{Q_c}{(h_{i+1} - H_i)} \qquad \text{..........2.56}$$

$$V_{i-1} = \frac{(h_{i+1}\Delta M - \Delta H)}{(h_{i+1} - H_i)} \qquad \text{...........2.57}$$

$$L_{i+1} = V_i - \Delta M \qquad \text{...........2.58}$$

for i=1,....,NT

where

$$\Delta M = V_i - \Delta M \qquad \text{...........2.59}$$

$$\Delta H = V_{NT} H_{NT} - Rh_D - F_i h_{Fi} \qquad \text{...........2.60}$$

V_{i-1} and L_{i+1} can be obtained by simultaneous solution of total material and energy balance equations which are algebraic in this simple model.

2.6 PHASE EQUILIBRIUM AND BUBBLE POINT CALCULATIONS:

The distribution of the components between the vapor and the liquid phases is given by the equilibrium constant k value:

$$k_{i,j} = \frac{\gamma_{i,j} p_{i,j}}{P_i} = \frac{y_{i,j}}{x_{i,j}} \qquad \text{...........2.61}$$

The pure component vapor pressure of the jth component $p_{i,j}^{sat}$ is given by Antoine equation where system pressure is supplied as an input parameter. Ideal thermodynamics was not chosen because of high non-ideality of the present system. Various types of non-ideal thermodynamics, such as SRK, PVR, UNIFAC have been tested and all of them have given almost similar results. Out of which, Wilson thermodynamics has been selected in the present study because of it's computational efforts. The activity coefficients is computed from Wilson equation. Equilibrium is said to exist on i^{th} tray. When the tray temperature is found such that the bubble point relation is satisfied as follows:

$$\sum_{j=1}^{NC} k_{i,j} x_{i,j} = 1 \qquad \text{........2.62}$$

The above relationship is satisfied by an iterative procedure, commonly known as bubble point calculation.

2.7 DEVELOPMENT OF STEADY STATE MODEL FOR MULTI COMPONENT DISTILLATION COLUMN:

The steady state model can easily be obtained from unsteady state model by setting $\frac{d}{dt} = 0$ in equation (2.1) to (2.10) however in the present study this model has not been used to obtain steady state solution. Instead Napthali-Sandholm method for steady state simulation (1971) method has been used because of it's better convergence characteristics and more efficient use of computation time.

2.7.1 Napthali-Sandholm method for steady state model.

(i) General i^{th} tray

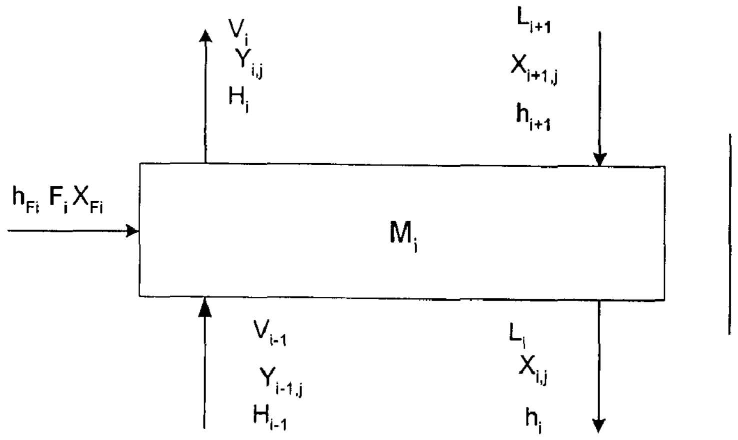


Fig 2.4 Modelling of general i^{th} tray

Where

$L_{I,j}$=liquid flow rate of j^{th} component leaving i^{th} tray kmols /h

$hl_{I,j}$ = liquid phase molar enthalpy of j^{th} component leaving i^{th} tray kJ/kmol

V_{ij}= Vapor flow rate of j^{th} component leaving i^{th} tray kmols/h

Hv_{ij}= vapor phase molar enthalpy of j^{th} component leaving i^{th} tray kJ/kmol

f_{ij} = feed flow rate of j^{th} component into i^{th} tray

hf_{ij} = feed enthalpy of j^{th} component into i^{th} tray kJ/kmol

a) Component material balance for i^{th} tray

$$M_{i,j} = L_{i+1,j} - L_{i,j} - V_{i,j} + V_{i-1,j} + f_{i,j} \qquad ....2.63$$

for I=1,....,NT; for j=1,.....,NC;

b) Efficiency equilibrium relationship

$$Q_{i,j} = \frac{\eta_{ij} k_{ij} V_i l_{ij}}{L_i} - V_{ij} + \frac{(1-\eta_{ij}) v_{i+1,j} V_i}{V_{i+1}} \qquad ....2.64$$

for I=1,....,NT ; for j=1,.....,NC;

With the following notations

$$L_i = \sum_{j=1}^{NC} l_{ij} \quad \text{and} \quad V_i = \sum_{j=1}^{NC} v_{ij} \qquad ....2.65$$

c) Total enthalpy balance for i^{th} tray

$$E_i = h_{i+1} - h_i - H_i + H_{i-1} + h_{Fi} \qquad .....2.66$$

With

$$h_i = \sum_{j=1}^{NC} l_{ij} hl_{ij} \quad H_i = \sum_{j=1}^{NC} v_{ij} Hv_{ij} \quad \text{and} \quad h_{Fi} = \sum_{j=1}^{NC} f_{ij} h_{fij} \qquad ......2.67$$

(ii) Condenser

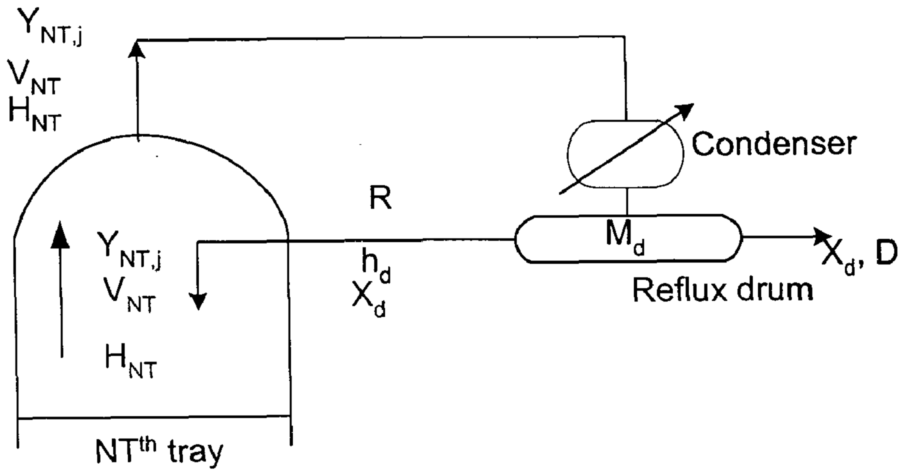


Fig 2.5 Modelling of Condenser

Where

T_c = condenser temperature

$V_{NT} = \sum_{j=1}^{NC} v_{NT,j}$ = Distillate rate kmols/h

$L_{NT} = \sum_{j=1}^{NC} l_{NT,j}$ = Total liquid flow rate into the NT-1th tray

$V_{NT-1} = \sum_{j=1}^{NC} v_{NT-1,j}$ = Total vapor flow rate entering the condenser

$R = \frac{L_{NT}}{V_{NT}}$ = Reflux ratio

a) Component material balance

$$E_{NT} = v_{NT-1,j} - l_{NT,j} - v_{NT,j} \qquad \text{.......2.68}$$

b) Efficiency equilibrium relationship is replaced by the following specification since the condenser is total as

$$L_{NT,j} - Rv_{NT,j} = 0 \qquad \text{......2.69}$$

c) Total energy balance for condenser is replaced by the following specification

$$T_{NT} - T_c = 0 \qquad \text{.......2.70}$$

T_c condenser temperature is supplied as an input specification.

(iii) Reboiler:

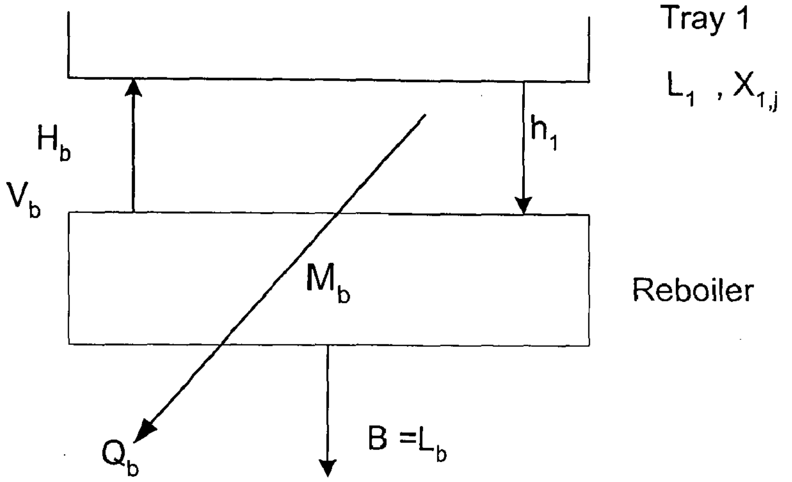


Fig 2.6 Modelling of Reboiler

Here reboiler is considered as an equilibrium stage with efficiency equal to 1.

$B = L_B = \sum_{j=1}^{NC} l_{B,j}$ = Bottoms product kmols

$V_B = \sum_{j=1}^{NC} v_{B,j}$ = Total vapor flow rate leaving reboiler

a) Component material balance:

$$M_{B,j} = l_{1,j} - v_{B,j} - l_{B,j} \qquad \text{.......2.71}$$

b) Equilibrium relationship:

$$Q_{B,j} = \frac{v_{B,j}}{V_B} - k_{B,j}\frac{l_{B,j}}{L_B} \qquad \text{........2.72}$$

c) Enthalpy balance to reboiler is replaced by the following specification:

$$L_B - B = 0$$

Here Q_{ij}, E_l and M_{ij} represent discrepancy functions for equilibrium energy and component material balance equations respectively. Actually these discrepancy functions should be zero, when the solution has been reached. However at the beginning of the Newton-Raphson method these are non zero as the convergence is achieved. When these functions are within certain predefined tolerance, the solution is said to be obtained.

2.8 STATEMENT OF DYNAMIC MODEL EQUATIONS FOR BINARY DISTILLATION:

For the sake of convenience in binary distillation, slightly different nomenclature is used. Here x_1 represents the liquid phase mole fraction of more volatile component. The reason for stating binary model in this manner is that this nomenclature is convenient since there is one component material balance to be considered. Hence the subscript j is dropped. This binary model is also considered for developing reduced order model and for parameter estimation.

Component material balance:

$$\frac{d(M_i x_i)}{dt} = L_{i+1}x_{i+1} - L_i x_i - V_i y_i + V_{i-1}y_{i-1} + F_i x_{Fi} \qquad \text{........2.73}$$

y_i = mole fraction of more volatile component in vapor phase

$$y_i = \eta_i\left(y_i^* - y_{i+1}\right) + y_{i+1} \qquad \text{...2.74}$$

η_i = Murphree stage efficiency based on vapor phase .

Total material balance remains same as these of rigorous and semi-rigorous models:

Bubble point relationship:

$$\begin{aligned} &x_i\gamma_{i1}p_{i1}^{sat} + (1-x_i)\gamma_{i2}p_{i2}^{sat} - P_i = 0 \ \ or \\ &k_{i1}x_1 + (1-x_i)k_{i2} - 1 = 0 \end{aligned} \qquad .....2.75$$

CHAPTER-3

SIMULATION OF DISTILLATION PROCESS USING RIGOROUS MODEL

In this chapter systematic algorithms and their flowcharts are given for the dynamic simulation of binary as well as multi-component distillation process.

3.1 BINARY DISTILLATION COLUMN

The digital simulation of the distillation column is fairly straightforward, in which large number of ordinary differential equations (ODEs) and algebraic equations are solved. A systematic procedure is given for simplified binary distillation column, which is described through equations (2.73-2.75) in Chapter-2. For the simplified binary distillation column; Equimolal overflow, constant relative volatility, and theoretical plates have been assumed. A distillation column consist of NT trays and a tray is represented by two ODEs corresponding to total continuity equation and light component continuity equation and two algebraic equations corresponding to vapor – liquid phase equilibrium relationship and liquid hydraulic relationship. For the ith tray these equations are -

$$\frac{dM_i}{dt} = L_{i+1} - L_i \qquad \text{......3.1}$$

$$\frac{d(M_iX_i)}{dt} = L_{i+1}X_{i+1} + VY_{i-1} - L_ix_i - VY_i \qquad \text{......3.2}$$

$$Y_i = \frac{aX_i}{1+(a-1)X_i} \qquad \text{.......3.3}$$

$$L_i = \bar{L}_i + \frac{M_i - \bar{M}_i}{\beta} \qquad \text{.......3.4}$$

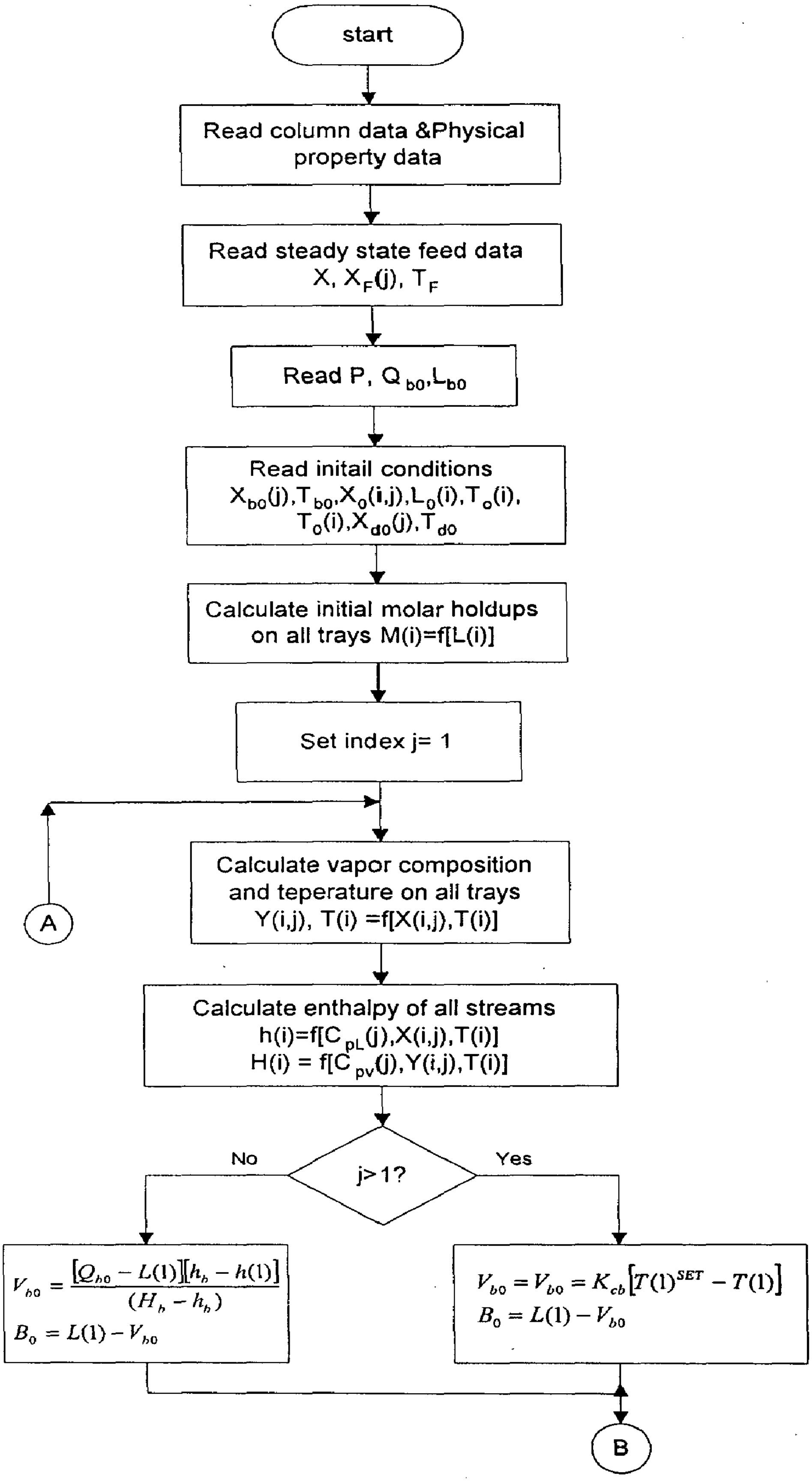


Fig 3.1 Flowchart for Dynamic Simulation of Distillation Process

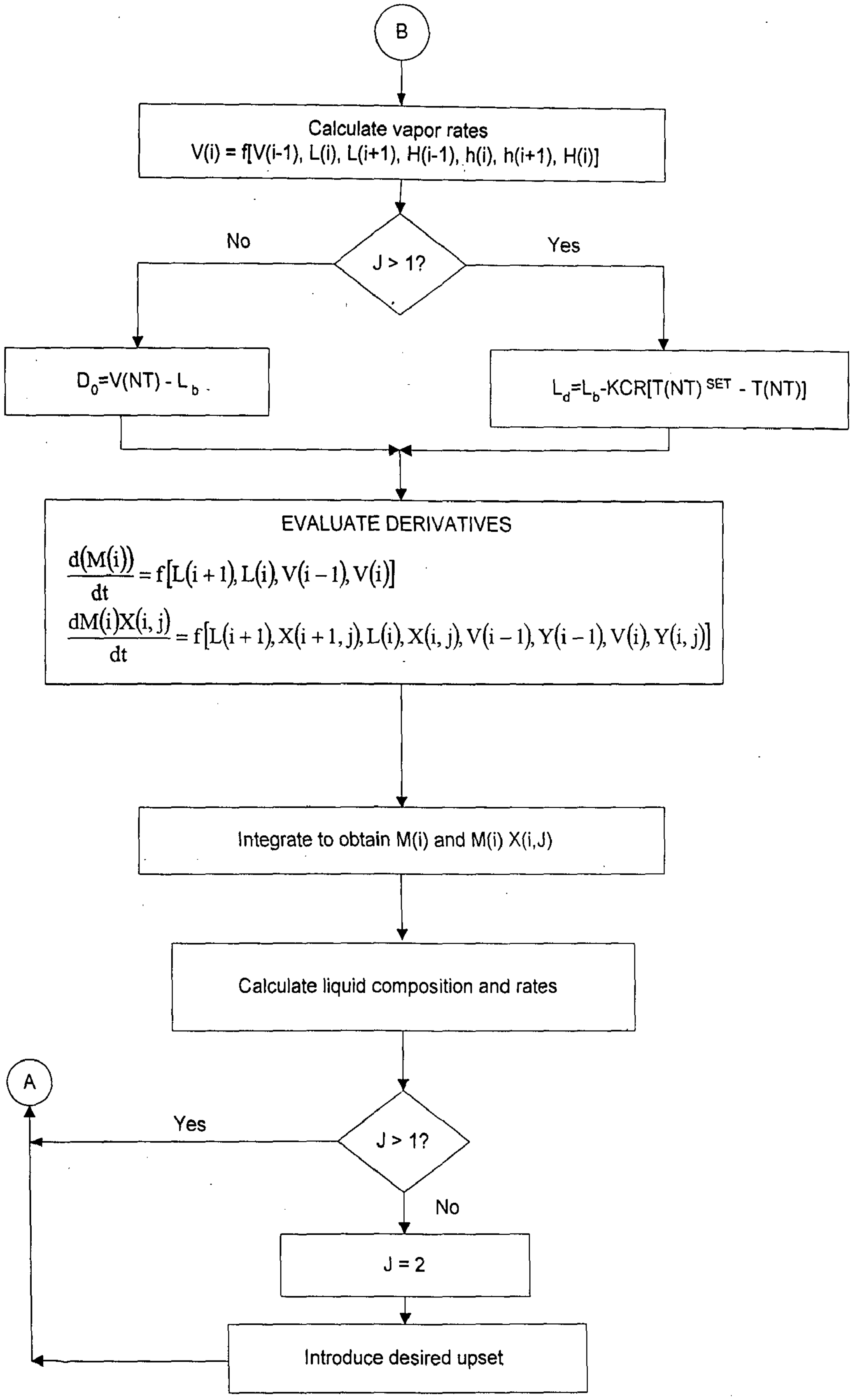

B
Calculate vapor rates
V(i) = f[V(i-1), L(i), L(i+1), H(i-1), h(i), h(i+1), H(i)]
J > 1?
No
Yes
D_0=V(NT) - L_b
L_d=L_b-KCR[T(NT)^{SET} - T(NT)]
EVALUATE DERIVATIVES
$\frac{d(M(i))}{dt} = f[L(i+1), L(i), V(i-1), V(i)]$
$\frac{dM(i)X(i,j)}{dt} = f[L(i+1), X(i+1,j), L(i), X(i,j), V(i-1), Y(i-1), V(i), Y(i,j)]$
Integrate to obtain M(i) and M(i) X(i,J)
Calculate liquid composition and rates
A
Yes
J > 1?
No
J = 2
Introduce desired upset

Fig 3.1 cntd..

Eq. (3.4) is a simple linear relationship between the liquid holdup on a tray, M_i, and the liquid flow rate leaving the tray, L_i. The parameter β is the hydraulic time constant and it typically lies in the range of 3 to 6 seconds per tray.

3.1.1 Algorithm for Binary Distillation Column

The ODEs are solved using Euler's method of integration and at each instant of time all holdups M_i and all liquid compositions X_l are obtained. For the known value of M_l and X_l the vapour concentration Y_l and liquid flow rate L_l are evaluated. The basic steps of the algorithm shown in fig 3.1 are as follows:

Step 1: Calculate vapor compositions on all trays from Eq. (3.3)

Step 2: Calculate all liquid flow rates from Eq. (3.4)

Step 3: Evaluate all derivatives from equation 3.1 and 3.2, which are called MDOT(N) and MXDOT(N) in the program.

Step 4: Integrate all ODEs with Euler method of integration and go to step 1.

3.2 MULTI-COMPONENT DISTILLATION COLUMN

The realistic distillation column consists of non-ideal column with NC components, non-equimolal overflow and inefficient trays. However the mathematical model is developed with following assumptions:

(1) Liquid on the tray is perfectly mixed and incompressible.

(2) Tray vapor holdups are negligible.

(3) Dynamics of the condenser and the reboiler is neglected.

(4) Vapor and liquid are in thermal equilibrium but not in thermal equilibrium. The departure from phase equilibrium is described by Murphree vapor efficiency .

The algorithm given in section 3.1.1 is only for dynamic simulation of binary distillation process with constant column data of the distillation process. In this section a generalized algorithm and flowchart have been given for the dynamic simulation studies of multi-component distillation process. The changes that have to be made to the basic structure of the solution algorithm are described as:

- More ODEs must be added per tray. We need one per component per tray. But this is easily programmed using doubly dimensioned variables X(N,J)., where N is the tray number and J is the component number.
- One energy balance per tray must be included if equimolal overflow cannot be assumed.
- An appropriate multi component bubble-point function must be used, which is more complex because of non-idealities, however it is provided with known liquid compositions and a known pressure to calculate the temperature and vapor compositions.

The specified multi component distillation column is assumed to have the following equipment configurations and conditions:

- There is one Feed plate onto which vapor feed and liquid feed are introduced.
- Pressure is constant and known on each tray. It decreases linearly with the height of the column from base value P_B to top value P_D.
- Coolant and steam dynamics are negligible in the condenser and reboiler.
- Vapor and liquid products D_V and D_L are taken off the reflux drum and are in equilibrium. Dynamics of the vapor space in the reflux drum and throughout the column are negligible.
- Liquid hydraulics are calculated from the Francis weir formula.

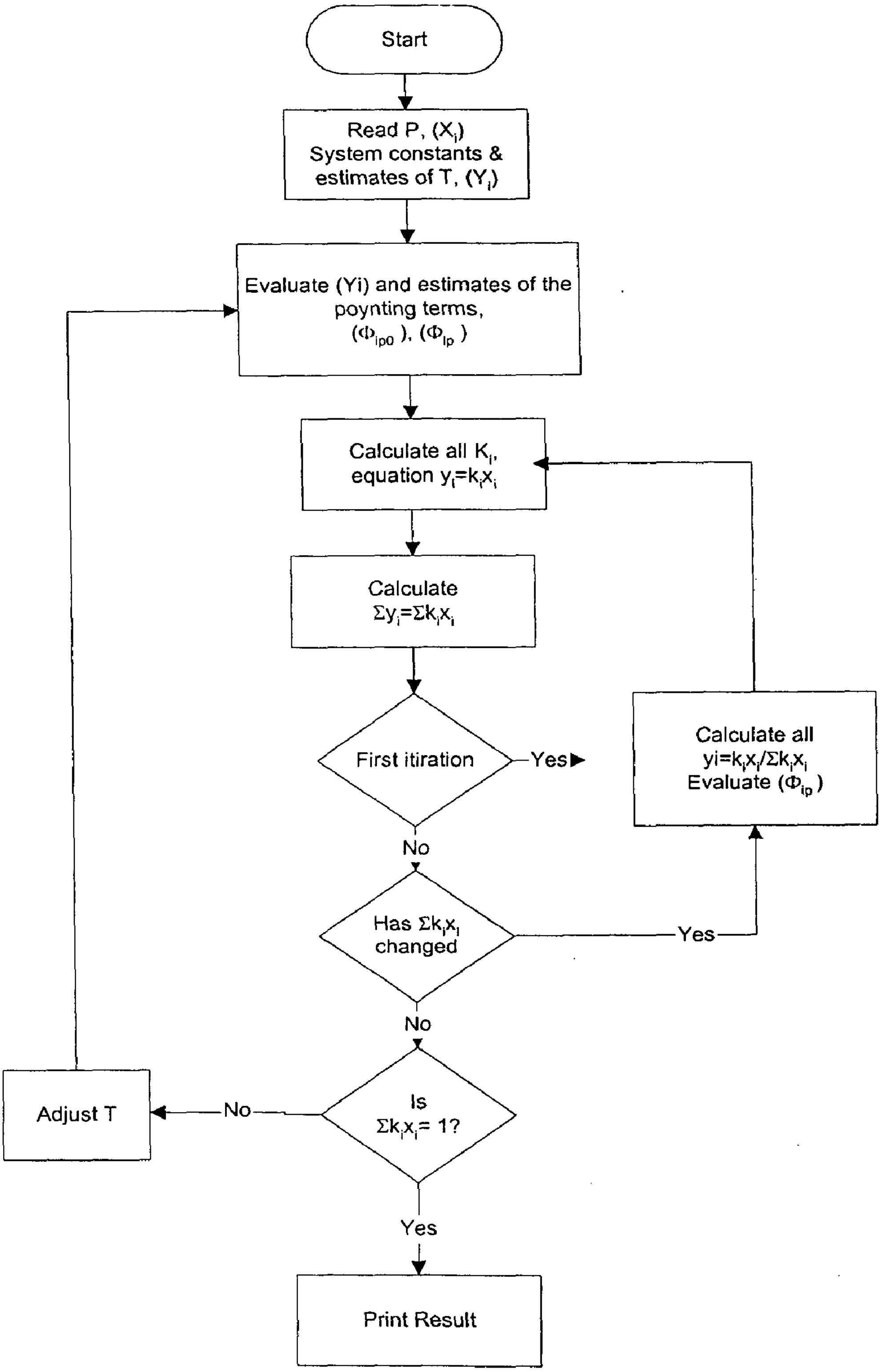


Fig. 3.2 Flowchart for Bubble point calculation (BUBPT)

G 10,009.

- Volumetric liquid holdups in the reflux drum and column base are held perfectly constant by changing the flow rates of bottoms product B and liquid distillate product D_L.

- Dynamic changes in internal energies on the trays are much faster than the composition or total holdup changes, so the energy equation on each tray is just algebraic.

- Reflux R and heat input to the reboiler Q_R are the manipulated variables. In the program given in appendix, they are simply held constant, thus giving the open loop response of the column. If the close loop response is desired, the program can be easily changed to use R to hold a temperature or a composition in the top of the column and to use Q_R to hold a temperature or a composition in the bottom of the column. There are two degrees of freedom, so two variables can be specified.

3.2.1 Algorithm for Multi-Component Distillation Process

The basic steps of the algorithm for the generalized program reflecting the above assumption and the change in the algorithm for the simplified binary distillation column are described as:

Step 1: Input data for column size, components, physical properties, feeds, and initial conditions (liquid compositions, liquid flow rates and temperatures on all trays).

Step 2: Calculate initial tray holdups and the pressure profile.

Step 3: Calculate the temperatures and vapor compositions from the vapor-liquid equilibrium data, using the function bubpt( ) (flowchart for bubble point calculation is shown in Fig 3.2.)

Step 4: Calculate liquid and vapor enthalpies, using function enth().

Step 5: Calculate vapor flow rates on all trays, starting in the column base, using the algebraic form of the energy equations.

Step 6: Evaluate all derivatives of the component continuity equations for all components on all trays plus the reflux drum and the column base.

Step 7: Integrate all ODEs (using Euler's method).

Step 8: Calculate new total liquid holdups from the sum of the component holdups. Then calculate the new liquid mole fraction from the component holdups and the total holdups.

Step 9: Calculate new liquid flow rates from the new total holdups for all trays, using function hydrau()

Step 10: Go to step 3 for the next step in time.

- The source code implementing the binary distillation process algorithm has been given in APPENDIX-A while source code implementing the multi – component distillation process has been given in APPENDIX-B.

CHAPTER-4

USER FRIENDLY SOFTWARE PACKAGE FOR DYNAMIC SIMULATION OF BINARY DISTILLATION PROCESS

4.1 DEVELOPMENT OF SOFTWARE PACKAGE

A graphical user interface (GUI) user friendly software package has been developed for the dynamic simulation study of binary distillation column. This is a pull down driven program. In this program there are three options and each option consists of two tasks.

1. Binary distillation process (BINARY CMP)
2. Multi component distillation process (MULTI CMP)
3. Help & Exit (EXIT &HELP)

In Binary distillation process the first task is for binary distillation process (BINARY CMP) which has been developed only for binary mixture. Selecting this task, program developed for binary distillation process gets executed and it takes data from "bin.dat" input file and it stores the results in "bin_result.dat" file. With the help of second task results and various characteristics of binary distillation process, which are obtained from the first task can be displayed by incorporating corresponding developed program.

In the second option multi-component (MULTI CMP) with the help of first task i.e. MI-Component, one can enter for generalized simulation of distillation process. Provision is given to accept data from user keyboard or pre-specified data file "mult.dat ". If the data is to be accepted from user keyboard further the program asks data interactively to help the user. A list of data expected from the program is given below.

COLUMN DATA:

Number of trays(NT)
Number of feed trays (NF)
Number of components (NC)
Weir height (WHS)
Length (WLS)
Column diameter (DS) in stripping section.
Diameter (DR)
Weir height (WHR)
Length (WLR) in rectifying section.
Volumetric holdup in column base
Volumetric holdup in reflux drum

FEED DETAILS:

Name of Component
MW
DENSITY
HCAPV
HCAPL
VP1
T1
VP2
T2
Liquid feed rate (FL)
Liquid feed temperature (TFL)
Liquid feed concentration of each component (XF_J) J=1, NC;
Vapor feed rate (FV)
Vapor feed temperature (TFV)
Liquid flow concentration of each component (X_J) J=1, NC;

COLUMN CONDITIONS

The pressures in the top and base of the column (PD, PB)
Reboiler heat input (QR) and reflux flow rate(R)
Vapor and liquid distillate product flow rates (DV, DL)
Murphree phase efficiency (EFF)

INITIAL CONDITIONS

Initial base temperature (TB)
base concentration of each component (XB[J])
Initial temperature T(N) N=1, NT;
liquid flow rate (L0(N)) N=1, NT;
concentration of each component X(N,J) N=1, NT; J=1, NC;
Initial reflux drum temperature (TD)
base concentration of each component (XD[J]) j=1, NC;

With the help of second task the results of first tasks can be displayed. Also various desired characteristics, as mentioned below may be displayed.

(i) Concentration of distillate with respect to time
(ii) Sample tray concentration with respect to time
(iii) Tray temperature with respect to time for few selected trays.

In third option (Help & Exit) with the help of first task brief idea of operating program and definitions of variables in dynamic simulation models can be obtained. Second task is for exit to command prompt.

The basic flowchart for this software package is shown in Fig 4.1and Fig. 4.2 [(a)–(d)] shows the layout of main menu and various pull-down menus.

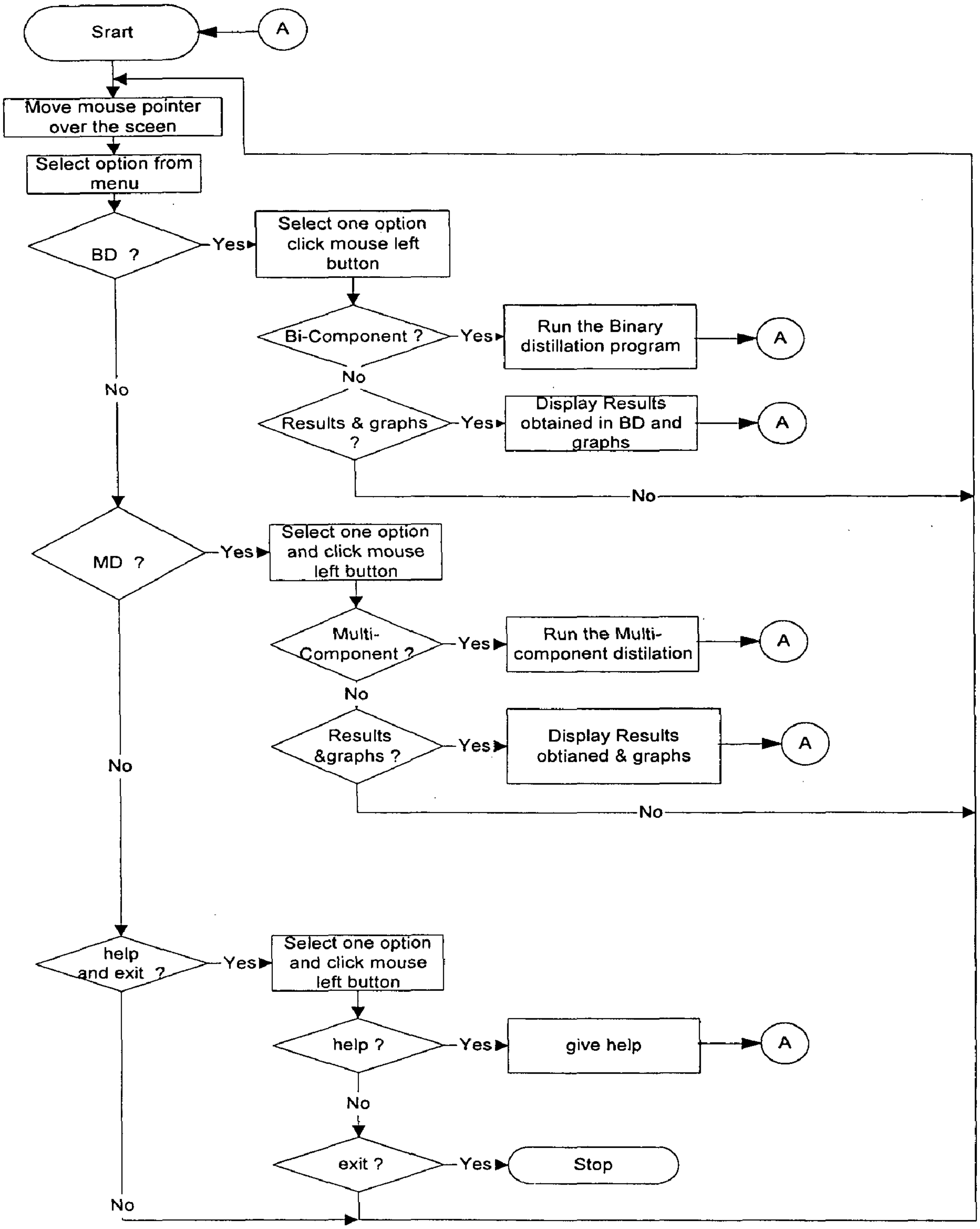
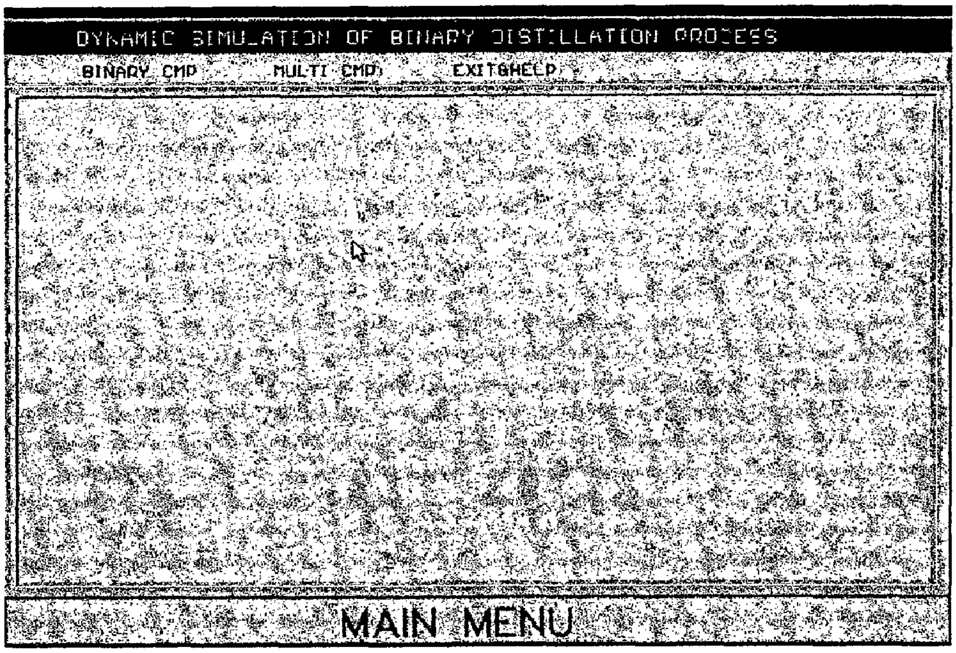
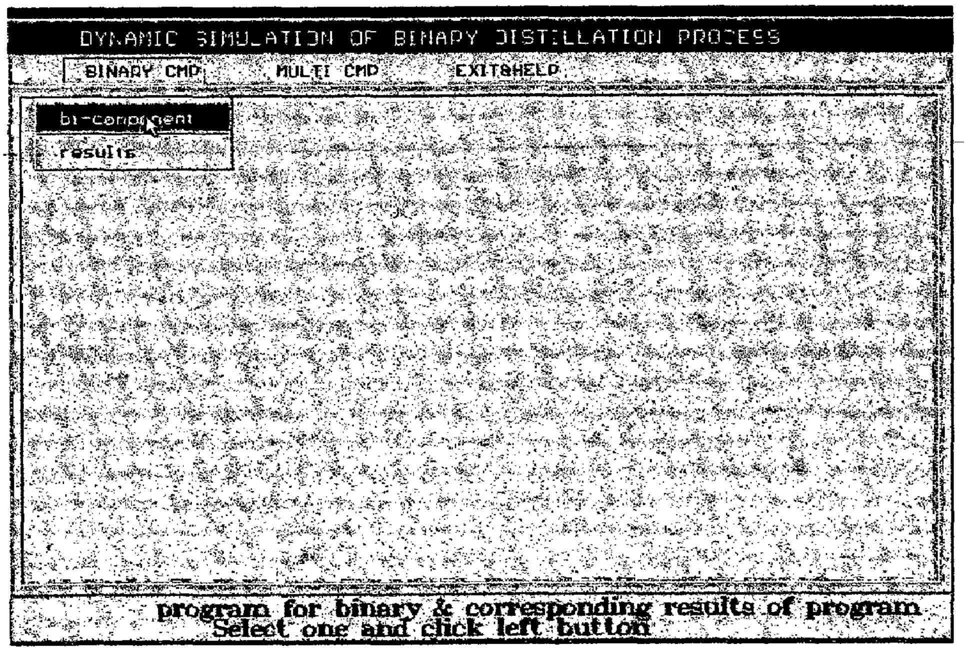


Fig 4.1 Flowchart for DIMOSIM-DP



MAIN MENU

Fig 4.2(a)



program for binary & corresponding results of program
Select one and click left button

Fig 4.2(b)

Fig 4.2 Layout of Menu and Pull-Down Menus

2) ***Name: maindow():***

Input: x, y co-ordinates and window color

(X1, Y1) starting coordinates

(X2, Y2) ending coordinates

C1 window color

Output: It displays window on the screen

Calls: none

Brief description: it displays window for specified coordinates and color.

3) ***Name: getkey():***

Input: *none*

Output: return pressed keystroke.

Calls: none

Brief description: it returns pressed keystroke

DERIVED CLASS "wind1"

1) ***Name: icons()***

Input: *none*

Output: displays icons on command menu.

Calls: icon ()

Brief Description: it displays normal icons on command menu.

2) ***Name:*** **store()**

Input: starting co-ordinates (X1, Y1)

Output: it restores background colour from starting co-ordinates

Calls: none

Brief description: it restores background colour at task box after selecting options and tasks.

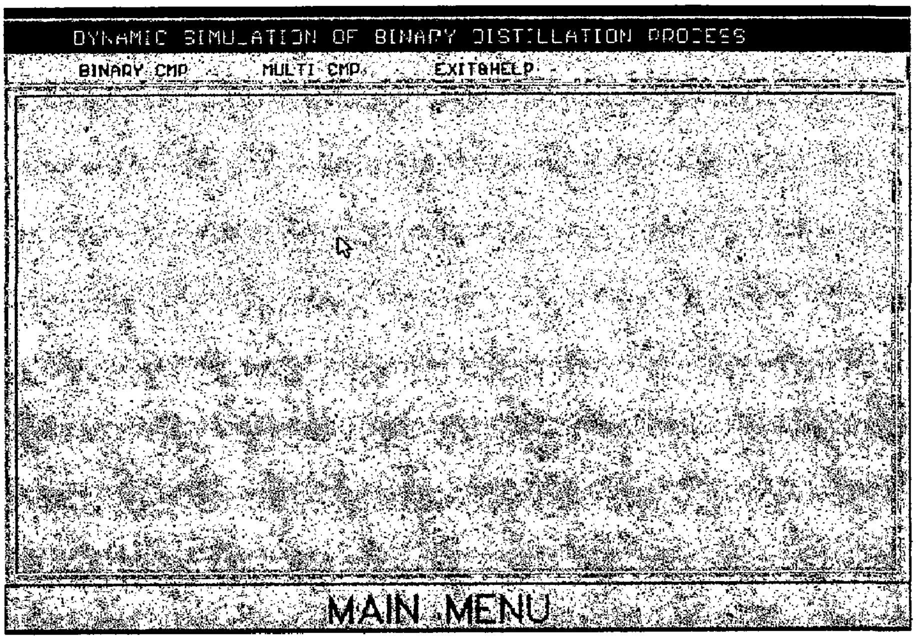
3) ***Name: restore()***

Input: starting coordinates

Output: it restores background colour of the entire screen.

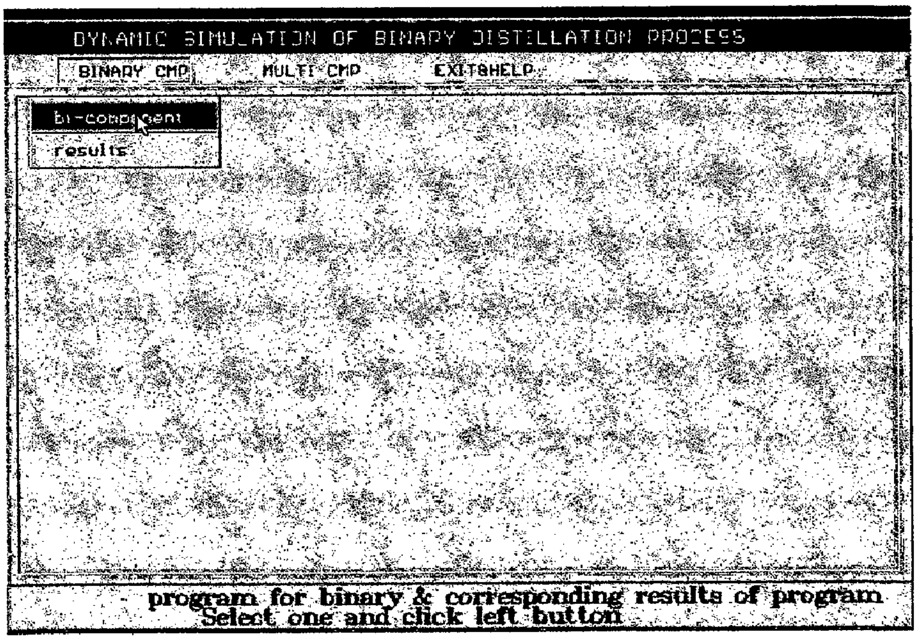
Calls: none

Brief Description: It restores the background colour after completing each option.



MAIN MENU

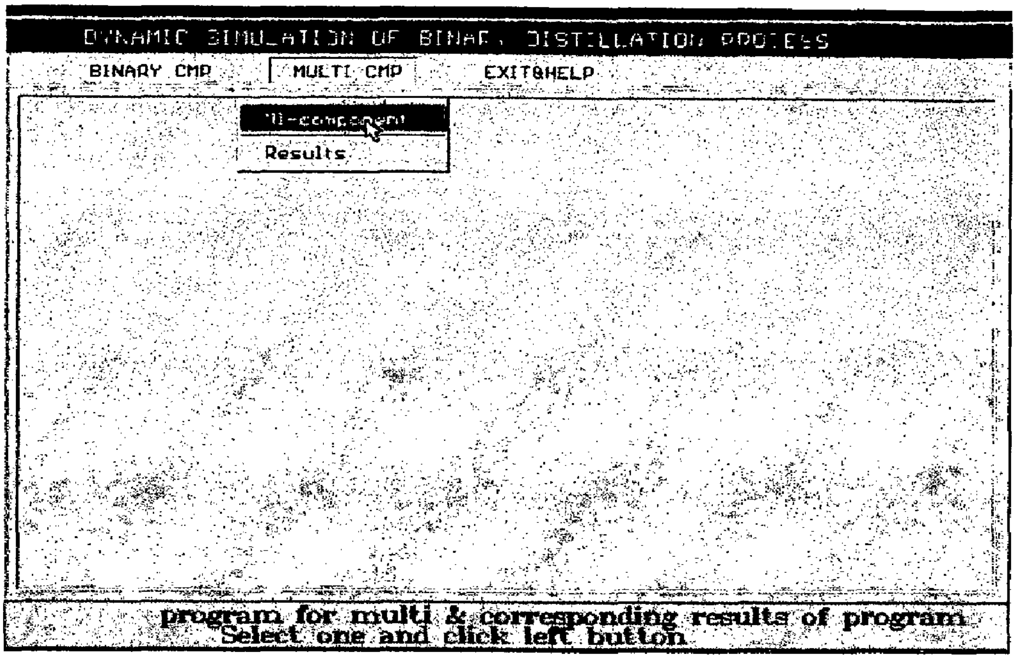
Fig 4.2(a)



program for binary & corresponding results of program
Select one and click left button

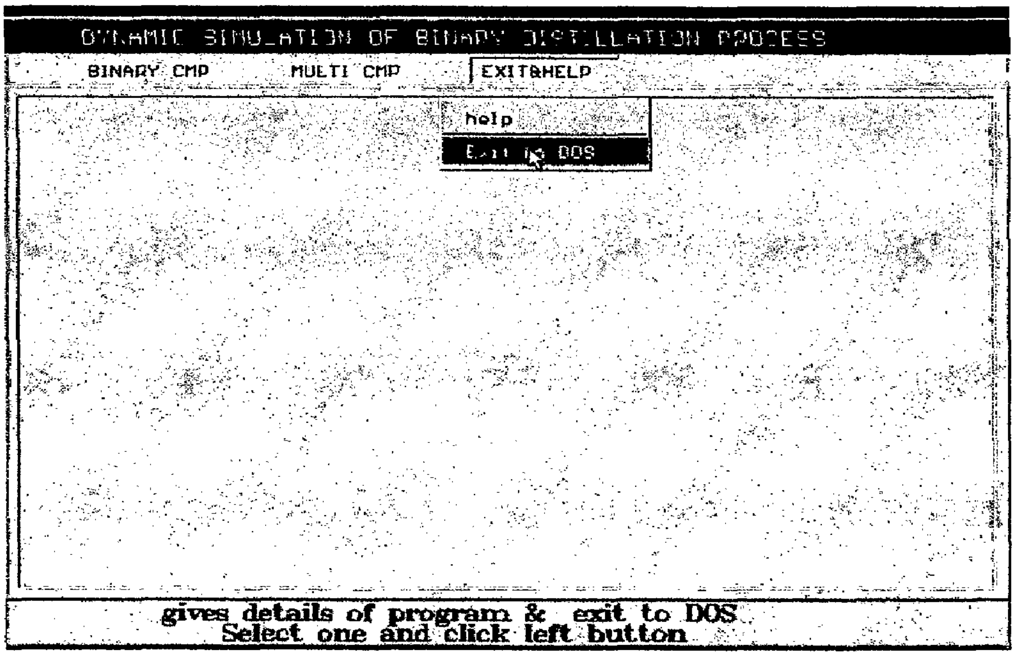
Fig 4.2(b)

Fig 4.2 Layout of Menu and Pull-Down Menus



program for multi & corresponding results of program
Select one and click left button

Fig 4.2(c)



gives details of program & exit to DOS
Select one and click left button

Fig 4.2(d)

Fig 4.2 Layout of Menu and Pull-Down Menus (cont..)

4.2 PROGRAM DETAILES:

A menu driven software DIMOSIM_DP(Dynamic simulation of Distillation Process) is developed in C++ on DOS environment. DIMSIM_DP invokes various functions developed, which are arranged in two classes.

1.) Basic Class "wind":

i Icon():
ii Maindow():
iii Getkey():

2.) Derived Class "wind1"

i icons():
ii store():
iii Restore():
iv Print():
v Plot():
vi mainbox():

Brief documentation of above functions is given in next section.

4.3 SOFTWARE MODULES DEVELOPED

BASIC CLASS "wind"

(i) *Name: icon():*

Input: co-ordinates of four corners of icon, left & top border colour, right & bottom colour and icon colour

(X1, Y1) starting coordinates

(X2, Y2) ending coordinates

C1, C2 and C3 left & top border colour, right & bottom colour and icon colour

Calls: *none*

Output: displays an icon according to co-ordinates and colours

Brief description: it displays an icon as specified in input

2) ***Name: maindow():***

Input: x, y co-ordinates and window color

(X1, Y1) starting coordinates

(X2, Y2) ending coordinates

C1 window color

Output: It displays window on the screen

Calls: none

Brief description: it displays window for specified coordinates and color.

3) ***Name: getkey():***

Input: *none*

Output: return pressed keystroke.

Calls: none

Brief description: it returns pressed keystroke

DERIVED CLASS "wind1"

1) ***Name: icons()***

Input: *none*

Output: displays icons on command menu.

Calls: icon ()

Brief Description: it displays normal icons on command menu.

2) ***Name:* store()**

Input: starting co-ordinates (X1, Y1)

Output: it restores background colour from starting co-ordinates

Calls: none

Brief description: it restores background colour at task box after selecting options and tasks.

3) ***Name: restore()***

Input: starting coordinates

Output: it restores background colour of the entire screen.

Calls: none

Brief Description: It restores the background colour after completing each option.

4) ***Name:*** **print()**

   Input: text file name

   Output: Displays text on the screen.

   Calls: none

   Brief Description: it displays text on the screen written in the mentioned text file.

5) ***Name:*** **plot()**

   Input: data file name containing two desired parameters.

   Output: Displays relationship between two specified parameters

   Brief Description: it displays relationship between two specified parameters on the screen according to data file.

6) ***Name:*** **mainbox()**

   Input: *none*

   Output: All above functions are called in this function

   Calls: All functions, both in base class and derived class.

   Brief Description: It performs various tasks according to status of mouse.

- The source code for the program DIMOSIM is given in APPENDIX-C.

CHAPTER-5

SIMULATION CASE STUDIES

In the present work the developed programs namely BINDS and MULTDS for binary distillation process and multi – component distillation process respectively are tested using the data available in Luyben [15]. BINDS program assumes constant Murphree vapor efficiencies for various plates of the distillation column, however the MULTDS program accepts the Murphree vapor efficiencies of various plates as an input variable. The results obtained from the developed programs are matching closely with the results given in [15].

5.1 BINARY DISTILLATION PROCESS

The program dynamic simulation of binary distillation process has been tested with data available in Luyben [15], and is listed in table 5.1.

Table 5.1 Input data for Binary Distillation Process

Number of Trays = 20

Feed rate = 100 kmols / hr.

Disturbance = 55%

Average molar weight per tray = 10

Feed tray = 10th

Tray No. (i)	Composition x(i)	Tray No. (i)	Composition x(i)
1	0.035	11	0.51526
2	0.05719	12	0.56295
3	0.08885	13	0.61896
4	0.1318	14	0.68052
5	0.18622	15	0.74345
6	0.24951	16	0.80319
7	0.31618	17	0.85603
8	0.37948	18	0.89995
9	0.43391	19	0.93458
10	0.47688	20	0.96079

The results obtained for 55% disturbance assuming zero initial feed are presented in table 5.2. Table 5.2 shows that steady state is reached in 9.5 hrs.

Table 5.2: Simulated Results for Binary Distillation Process

TIME	XB	X5	X10	X15	X20	Distillate XD	Reflux flow rate R	Vapor flow rate V
0	0.02	0.18622	0.47688	0.74345	0.96079	0.98	128.01	178.009995
0.5	0.020138	0.196699	0.513102	0.7494	0.960839	0.980004	128.0063	178.158417
1	0.02107	0.21174	0.524262	0.760487	0.961372	0.9801	127.9062	179.307281
1.505	0.02218	0.22043	0.530302	0.768526	0.962045	0.980347	127.6372	181.081696
2.005	0.022757	0.222079	0.532295	0.772189	0.962418	0.980608	127.3284	182.667221
2.505	0.022677	0.218762	0.531387	0.772209	0.962416	0.980762	127.105	183.694641
3.005	0.022111	0.212809	0.528756	0.769897	0.962125	0.980767	127.0217	184.099731
3.505	0.021309	0.206332	0.525568	0.766685	0.961701	0.98065	127.0671	183.986359
4.005	0.020503	0.20099	0.5228	0.763781	0.961303	0.980472	127.1888	183.540878
4.505	0.019868	0.197745	0.521076	0.761947	0.961043	0.980299	127.3232	182.973419
5.005	0.019495	0.196788	0.520571	0.761424	0.960961	0.980181	127.4178	182.465134
5.505	0.019393	0.197672	0.521064	0.761994	0.961032	0.980137	127.4467	182.132736
6.005	0.019502	0.199582	0.522106	0.763163	0.961189	0.980159	127.4106	182.014877
6.505	0.019724	0.201635	0.52321	0.764388	0.961358	0.980221	127.3302	182.079498
7.005	0.019956	0.20315	0.524011	0.765253	0.961478	0.98029	127.2347	182.248672
7.505	0.020122	0.203798	0.52434	0.765568	0.961523	0.980343	127.1506	182.43306
8.005	0.020187	0.203612	0.524221	0.765368	0.961493	0.980364	127.0938	182.563019
8.505	0.020158	0.202879	0.523809	0.764833	0.961412	0.980353	127.0686	182.605621
9	0.020071	0.201988	0.52331	0.764205	0.961315	0.98032	127.0683	182.565613
9.5	0.019969	0.201246	0.522892	0.763673	0.961228	0.980278	127.0801	182.471054

The variation of liquid concentration on 5th,10th,15th and 20th tray with respect to the time are shown in figs. 5.1[(a), (b), (c), (d)]. The time variation of concentration in reflux drum, concentration in reboiler and vapour flow rate are shown in fig. 5.2, 5.3, and 5.4 respectively.

5.2 MULTI-COMPONENT DISTILLATION PROCESS

A five component mixture as given in [15] is selected for this study and the input data is given in table 5.3 and 5.4 containing column & mixture detail and initial conditions respectively.

Table 5.3: Column and mixture details for Multi component Distillation process

Number of trays NT = 15

Number of feed trays = 5

Weir hight in stripping section WHS = .75 in

Weir length in stripping section WLS = 1.25 in

Column diameter in stripping section DS = 72 in

Weir hight in rectifying section WHR = 48 in

Weir length in rectifying section WLR = 48 in

Column diameter in rectifying section DS = 72 in

Number of components = 5

Volumetric holdup in column base = 10.0 ft^3

Volumetric holdup in reflux drum = 10.0 ft^3

Liquid feed rate FL = 80 lb mol/hr

Liquid feed temperature TFL = 120 $^{\circ}$F

Pressure in the bottom = 21.20 psia

Pressure in the reflux drum = 19.7 psia

Reboiler heat input = 5.0 * 10^6 BTU

Reflux flow rate = 400.0 lb mol/hr

Vapour distillate product flow rate = 200.00 lb mol/hr

Murphree vapour efficiency = 0.5

Component	MW	DENS	HVAP	BPT	HCAPV	HCAPL	VP1	T1	VP2	T2
LLK	30.00	40.00	100.0	10.00	0.20	0.60	14.70	10.00	50.00	30.00
LK	50.00	40.00	90.00	90.00	0.40	0.60	14.70	90.00	500.0	200.0
INTER	90.00	60.00	70.00	150.0	0.30	0.50	14.70	150.0	150.0	200.0
HK	130.0	70.00	80.00	210.0	0.30	0.40	14.70	210.0	150.0	300.0
HHK	300.0	90.00	80.00	360.0	0.30	0.40	14.70	360.0	150.0	420.0

Table 5.4: Initial conditions for Multi component Distillation process

Tray No.	Initial Temp.	Liquid flow rate	Component Concentrations				
			1	2	3	4	5
Base-→ Component	201.58	0.00	0.00	7.25 e-03	4.88 e-02	8.36 e-1	1.08 e-1
1	154.90	740.10	9.99e-12	1.10e-01	2.40e-01	6.07e-01	4.33e-02
2	132.60	814.40	1.56e-09	2.86e-01	2.02e-01	4.73e-01	3.93e-02
3	120.20	892.00	1.82e-07	4.57e-01	1.31e-01	3.76e-01	3.59e-02
4	114.00	960.10	1.33e-05	5.72e-01	8.03e-02	3.14e-01	3.33e-02
5	108.40	986.00	7.60e-04	6.34e-01	4.96e-02	2.84e-01	3.25e-02
6	101.20	320.00	1.12e-03	8.18e-01	8.66e-02	9.42e-02	1.74e-06
7	98.20	381.90	1.29e-03	9.10e-01	4.46e-02	4.40e-02	7.76e-07
8	96.90	409.60	1.36e-03	9.53e-01	2.38e-02	2.18e-02	3.71e-07
9	96.20	423.70	1.40e-03	9.75e-01	1.28e-02	1.10e-02	1.81e-07
10	95.80	431.20	1.42e-03	9.86e-01	6.94e-03	5.63e-03	8.93e-08
11	95.50	435.20	1.43e-03	9.92e-01	3.74e-03	2.86e-03	4.40e-08
12	95.30	437.50	1.44e-03	9.95e-01	1.99e-03	1.45e-03	2.16e-08
13	95.10	438.70	1.44e-03	9.97e-01	1.04e-03	7.18e-04	1.04e-08
14	94.90	439.50	1.45e-03	9.98e-01	5.19e-04	3.42e-04	4.84e-09
15	94.20	438.60	1.75e-03	9.98e-01	2.36e-04	1.49e-04	2.05e-09
Reflux Drum R.D	77.26	400.00	1.74e-02	9.82 e-01	8.24 e-05	4.93 e-05	6.59 e-10

The five component mixture under consideration reaches steady state in 0.001 hrs. and takes three iterations. The results for the above mixture are presented in table 5.5 (a), 5.5 (b) and 5.5 (c).

Table 5.5 (a): Simulated results for 5 component mixture at time 0.00 hrs.

TIME = 0.00 Iteration No. 1

TRAY NO.	TB	X1	X2	X3	X4	X5	L
BASE →	201.596283	0	0.00725	0.0488	0.836	0.108	299.029968
1	154.856293	9.99e-12	0.11	0.24	0.607	0.0433	740.09997
2	132.61615	1.56e-09	0.286	0.202	0.473	0.0393	814.400024
3	120.233429	1.82e-07	0.457	0.131	0.376	0.0359	892
4	114.054855	1.33e-05	0.572	0.0803	0.314	0.0333	960.099976
5	108.400002	0.00076	0.634	0.0496	0.284	0.0325	986
6	101.178398	0.00112	0.818	0.0866	0.0942	1.74e-06	320
7	98.234787	0.00129	0.91	0.0446	0.044	7.76e-07	381.899994
8	96.915428	0.00136	0.953	0.0238	0.0218	3.71e-07	409.600006
9	96.191849	0.0014	0.975	0.0128	0.011	1.81e-07	423.700012
10	95.771614	0.00142	0.986	0.00694	0.00563	8.93e-08	431.200012
11	95.486832	0.00143	0.992	0.00374	0.00286	4.4e-08	435.200012
12	95.270561	0.00144	0.995	0.00199	0.00145	2.16e-08	437.5
13	95.095306	0.00144	0.997	0.00104	0.00072	1.04e-08	438.700012
14	94.923332	0.00145	0.998	0.000519	0.00034	4.84e-09	439.5
15	94.234634	0.00175	0.998	0.000236	0.00015	2.05e-09	438.600006
Liquid concentration in R.D	77.296219	0.0174	0.982	8.24e-05	4.93e-05	6.59e-10	400
Vapour concentration in R.D		0.55569	0.444301	9.704e-07	2.895e-07	8.07e-18	498.948364

Table 5.5 (b): Simulated results for 5 component mixture at time 0.0005 hrs.

TIME = 0.0005 hrs   Iteration No. 2

TRAY NO.	TB	X1	X2	X3	X4	X5	L
BASE →	201.60257	0	0.007259	0.048738	0.836078	0.107996	295.702759
1	154.874985	0	0.109901	0.239965	0.607135	0.043299	738.41156
2	132.618347	6.736e-09	0.285985	0.201882	0.473066	0.039304	812.782471
3	120.231911	2.079e-07	0.457019	0.130982	0.375973	0.035892	890.267639
4	114.053017	1.287e-05	0.572081	0.080267	0.314128	0.033315	958.120422
5	108.403831	0.00076	0.633917	0.049569	0.283976	0.032498	984.296082
6	101.184151	0.001118	0.818012	0.086566	0.094223	1.741e-06	318.71
7	98.247803	0.001284	0.910027	0.044578	0.044006	7.766e-07	380.4342
8	96.908325	0.001363	0.953019	0.023787	0.021804	3.712e-07	408.066
9	96.201591	0.001395	0.974989	0.012797	0.011006	1.812e-07	422.136
10	95.781815	0.001415	0.98601	0.006937	0.005631	8.936e-08	429.608154
11	95.492722	0.001427	0.991995	0.003738	0.002861	4.402e-08	433.612671
12	95.279732	0.001435	0.995025	0.001989	0.00145	2.162e-08	435.870239
13	95.093925	0.001441	0.997007	0.00104	0.000718	1.041e-08	437.089142
14	94.919853	0.001451	0.997987	0.000519	0.000342	4.842e-09	437.909637
15	94.240158	0.001748	0.99797	0.000236	0.000149	2.022e-09	437.208069
Liquid concentration in R.D	77.28476	0.017416	0.98201	8.235e-05	4.93e-05	0	400
Vapour concentration in R.D		0.555899	0.4441	9.692e-07	2.895e-07	0	504.591064

Table 5.5 (C):Simulated results for 5 component mixture at time 0.001 hrs.

TIME = 0.001 hrs Iteration No. 3

TRAY NO	TB	X1	X2	X3	X4	X5	L
BASE→	201.624878	0	0.007248	0.048665	0.836178	0.107999	295.583221
1	154.897766	0	0.109783	0.239906	0.607301	0.0433	738.225464
2	132.623642	1.35e-08	0.285921	0.201796	0.473156	0.039307	812.712708
3	120.234528	0	0.457001	0.130975	0.376	0.035888	890.184692
4	114.043587	1.46e-05	0.57211	0.080241	0.314247	0.033328	958.072327
5	108.418945	0.00076	0.633831	0.049543	0.283972	0.032498	984.449402
6	101.177071	0.00112	0.818008	0.086536	0.09425	1.743e-06	318.70697
7	98.246811	0.00128	0.910043	0.044558	0.044014	7.7715e-07	380.425598
8	96.915985	0.00136	0.953042	0.023777	0.021809	3.7150e-07	408.057678
9	96.200455	0.0014	0.974973	0.012795	0.011011	1.8129e-07	422.142548
10	95.780144	0.00142	0.986011	0.006935	0.005632	8.9426e-08	429.609528
11	95.493011	0.001427	0.991988	0.003736	0.002862	4.4070e-08	433.617249
12	95.282074	0.001433	0.995045	0.001989	0.00145	2.1630e-08	435.849762
13	95.100235	0.001437	0.997015	0.001039	0.000718	1.0417e-08	437.095581
14	94.927849	0.001448	0.997979	0.000519	0.000342	4.8434e-09	437.962769
15	94.247459	0.001744	0.997943	0.000236	0.000149	1.9881e-09	437.447327
Liquid concent ration in R.D	77.280777	0.017423	0.982027	8.23e-05	4.931e-05	0	400.000000
Vapour concent ration in R.D		0.556011	0.444037	9.68e-07	2.894e-07	0	504.577637

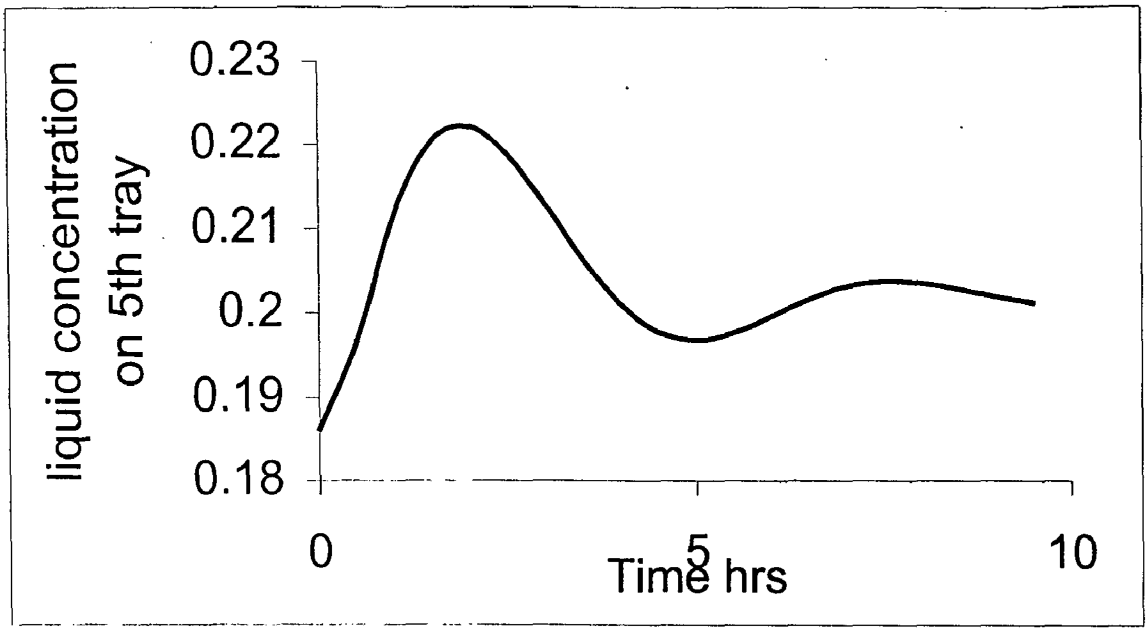


Fig 5.1 (a)

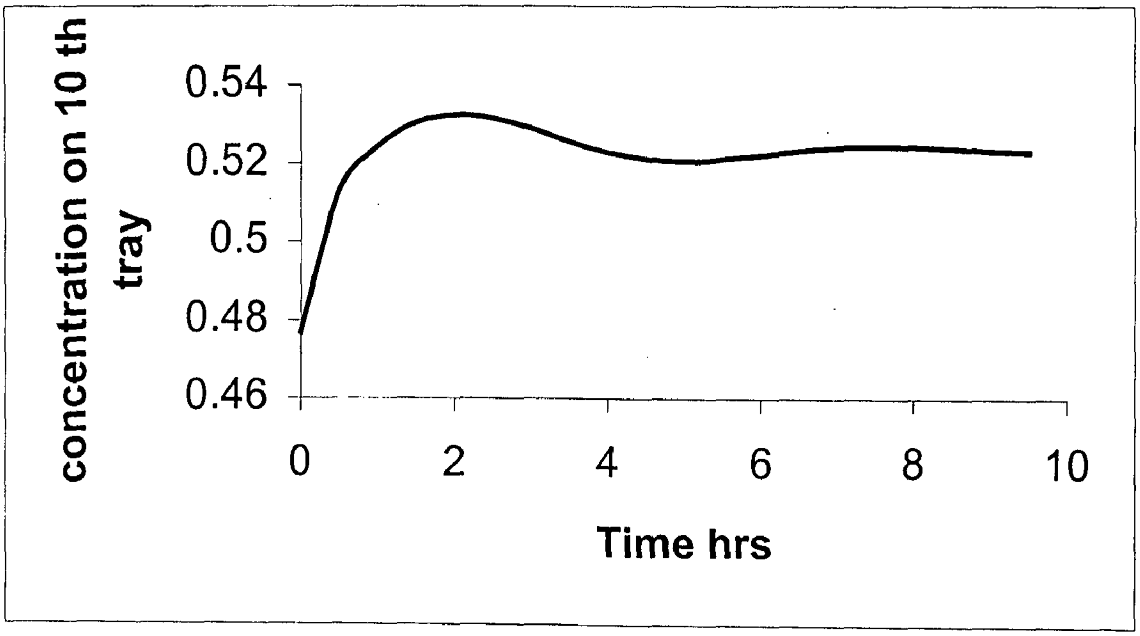


Fig. 5.1 (b)

Variation of liquid concentration on 5th and 10th tray with respect to time for binary distillation process.

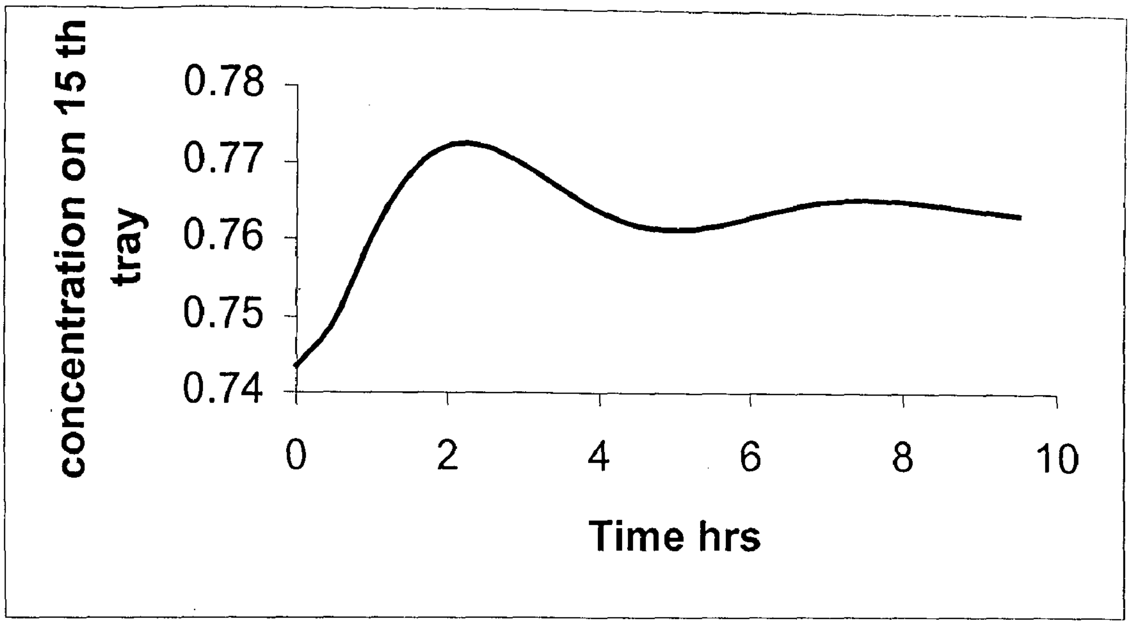


Fig. 5.2 (a)

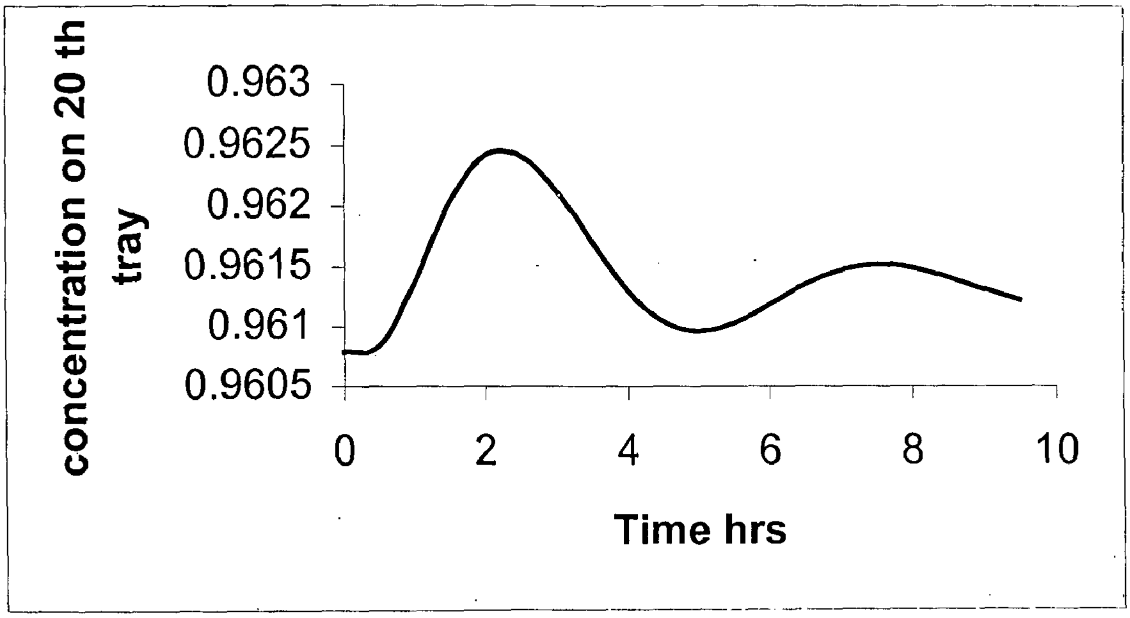


Fig. 5.2 (b)

Variation of liquid concentration on 15th and 20th tray with respect to time for binary distillation process.

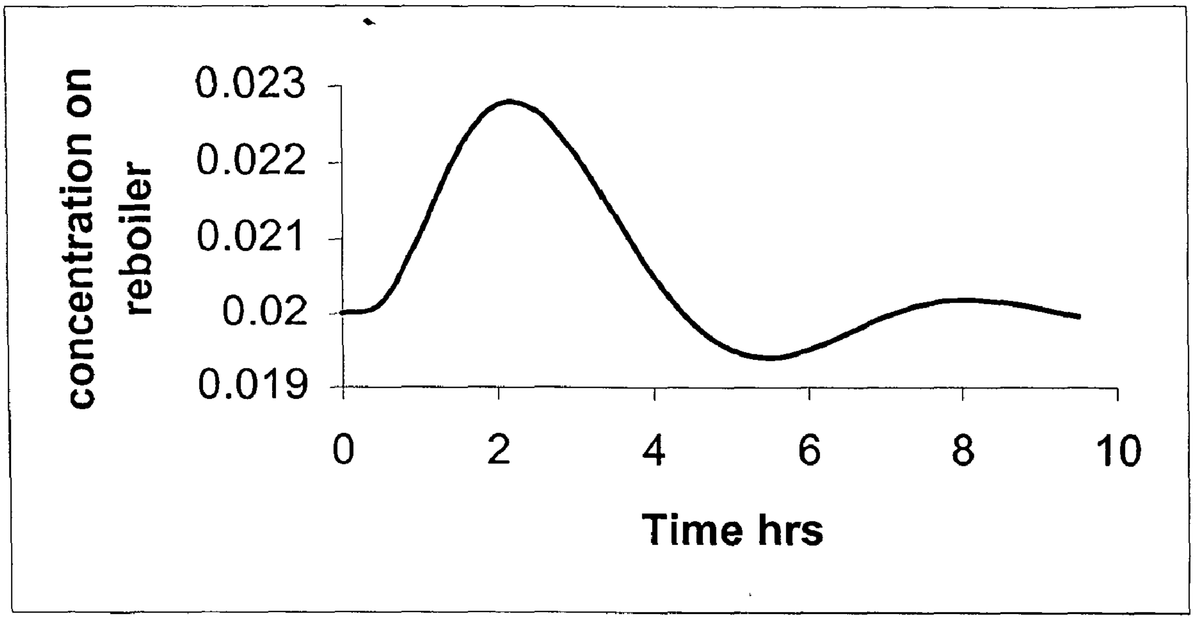


Fig. 5.3 variation of reboiler concentration

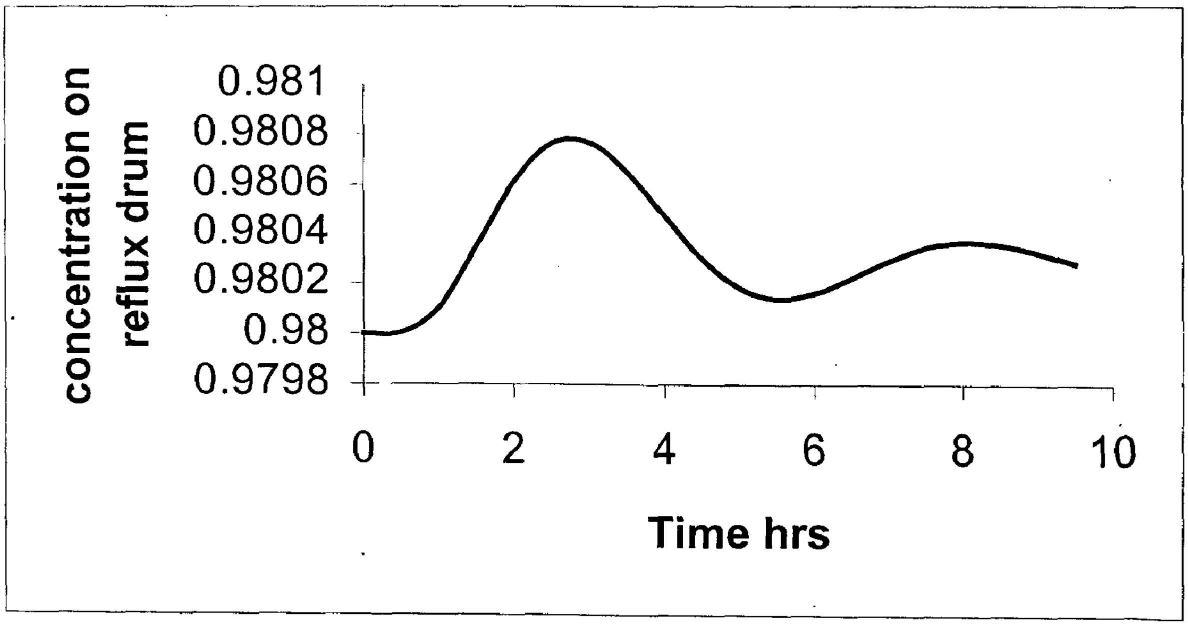


Fig. 5.4 variation of reflux concentration

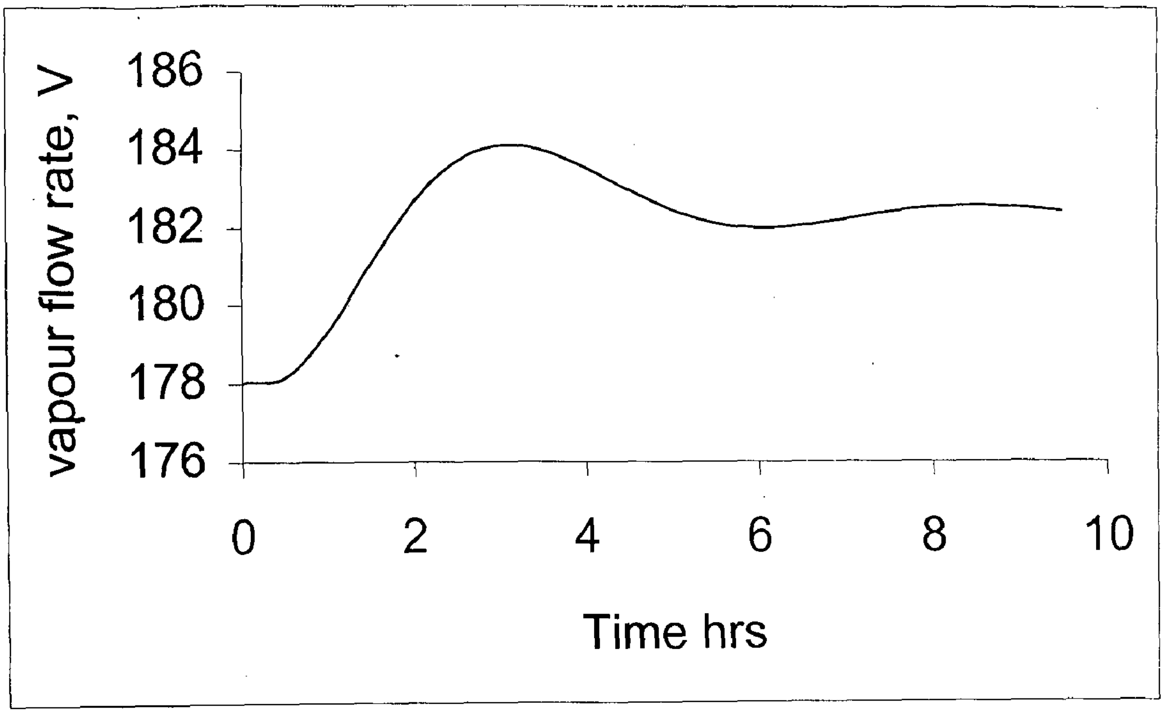


Fig 5.5 flow rate of vapour in the reflux drum

CHAPTER-6

CONCLUSIONS AND SCOPE FOR FUTURE WORK

Distillation is one of the widely used energy intensive separation processes of the chemical industry. Dynamic simulation of distillation is characterised by stiff set of differential algebraic equations (DAEs) of material and energy balances and efficiency equilibrium relationships. In present work different available models, which are derived into three categories namely Rigorous- where both material and enthalpy balances are represented by differential equations, Semi-Rigorous -where enthalpy balances are represented by algebraic equations and simple where enthalpy and total material balances are algebraic. All the three above models are studied and presented in detail. There is another category namely reduced order model; in this, effort is to develop a model, which lead to fast computation and less memory space requirement. The work can be extended for the study of Reduced order model of distillation process.

For the dynamic simulation of distillation process the rigorous model is selected and interactive software is developed to simulate continuous binary distillation process. Two case studies corresponding to the data for unknown mixture available in Luyben [15] have been performed and satisfactory results have been obtained. Variation of liquid concentration on selected trays with respect to time is also presented for binary distillation case where a disturbance of 55% is given in feed at 10th tray which is also presented by Fig. 5.1(b).

The developed user-friendly software also facilitates simulation of multi-component distillation (continuous). Another five component mixtures whose data have been reported in Luyben [15] is considered to test multiple component distillation simulation. Luyben has not reported the name of the mixture therefore the symbolic names have been used for the components constituting the mixture. Further the work can be extended to perform the study for the distillation process to obtain distillate responses and various component concentrations for the step change in input variables such as feed rate, reflux flow rate and reboiler duty cycle.

The *user-friendly software* developed here has its own limitations such as it does not support on-line analysis, parameter estimation, closed loop schemes and so on. Therefore, more features are to be added in order to make it commercially viable.

REFERENCES

1. Ballard D.M. and C.B. Brosilow, Dynamic Simulation Of Multi-component Distillation Columns, Annual Meeting Of Aiche, Miami (Nov.1978)
2. Benallou A, D. E. Seborg and D.A.Mellichamp, Low Order Physically Lumped Dynamic Models For Distillation Column Control, Annual Meeting Of Aiche, Losangeles (1982)
3. Boston J.F., and S.I.Sullivan, Jr., A New Class Of Solution Methods Of Multicomponent Multistage Separation Processes, Can.J. Chem. Engg, 52, 1(1974)
4. Boston,J.F.,H.I.Britt, S.Jirapongphan And V.B.Shah ,"An Advanced System For The Simulation Of Batch Distillation Operations, " Foundation Of Computer Aided Chemical Process Design, Eds., R.S.H.Mah.Seider, Aiche(1981)
5. Brosilow, C., R.Tanner And H.Tureff, Optimization Of Staged Counter-Current Processes, Aiche National Meeting , St.Louis(1968)
6. David M. Himmeblau, Basic Principles And Calculations In Chemical Engineering, Sixth Edition, Prentice-Hall of India Pvt ltd. New Delhi, 1997.
7. Distefano,G.P., Mathematical Modelling And Numerical Integration Of Multicomponent Batch Distillation Equations, Aiche.J.,14,946(1968)
8. Eykhoff, P., System Identification, Wiley, London(1974)
9. Finalayson,B.A.,The Method Of Weighted Residuals And Variational Principles , Academic Press, New York(1972)
10. Gallun,S.E. And C.D.Holland,Gear's Procedure For The Simultaneous Solutions Of Differential And Algebraic Equations With Application To Unsteady Distillation Problems, Comput. Chem.Engg.,6,231(1982)
11. Gear, C.W., Numerical Initial Value Problems In Ordinary Differential Equations, Prentice Hall, NJ (1971)
12. Graupe, D., Identification Of Systems, R.E.Krigger, Huntinton, New York (1972)

13. Holland, C.D., Unsteady State Process with Applications to Multicomponent Distillation, Englewood Cliffs , NJ, Prontice-Hall (1966).
14. Pradeep B. Deshpande, Distillation Dynamics and Control, Instrument society of America, McGraw-Hill Publishing company Limited
15. Willium L. LUYBEN, Process Modelling, Simulation And Control For Chemical Engineering Second Edition, McGRAW-HILL INTERNATIONAL EDITION Chemical Engineering Series

APPENDIX – A

```
///****************************************************************************************///
///******************************PROGRAM FOR *************************************///
///******DYNAMIC SIMULATION OF BINARY DISTILLATION PROCESS******///
///****************************************************************************************///

#include<iostream.h>
#include<fstream.h>
#include<conio.h>
#include<math.h>
#include<process.h>
#include<graphics.h>
#define N 30

int  nt = 20, nf = 10;
float x[N] = {0.0,0.035,0.05719,0.08885,0.1318,0.18622,0.24951,0.31618,
          0.37948,0.43391,0.47688,0.51526,0.56295,0.61896,0.68052,0.74345,
          0.80319,0.85603,0.89995,0.93458,0.96079 };

float y[N],l[N],lo[N],m[N],alpha = 2.0, beta = 0.1;
float mx[N],mdot[N],mxdot[N],xbdot,xddot,xb,yb,v,b,r,d;
float z,mo,mbo,mdo,kcd,kcb,ro,vo,f,taub,taud,xd;
float erintd, erintb,errd,errb;
double time, tprint,delta;
class befc
{
          private:
           int n;
     public:
            befc() { }
            void init();                        // initial conditons
            float equil(float);       // Vapour presure calculations
            void deriv();              // evaluating derivatives
            void integ();              // Euler integration
            void maincal();
            ~befc() {   }
};

          void befc::init()
          {
                    mdo = 100.; mbo = 100.; mo = 10.; ro = 128.01;
                    vo = 178.01; f = 100.;
                    kcd = 1000.; kcb = 1000.; taud = 5.; taub = 1.25; delta = .0050;
                    time = tprint = erintd = erintb = 0.0;
                    xb = 0.02; xd = 0.98;
                    ofstream tout("concent.dat");
                    tout<<"\n intail composition % ";
                    for(n=1;n<=nt;n++)
```

```
          {
            tout<<"\n"<<"x["<<n<<"] = "<<x[n]<<"%";
          }
          cout<<"\n";
          tout<<"\n % \n %";
          tout<<"\n no of tryas are 20 %";
          tout<<"\n feed rate is 100. %";
          tout<<"\n disturbance is 0.55 %";
          tout<<"\n average molal weight is  "<<mo<<"  %";

          //DISTURBANCE
          z = 0.55;

          //INITIAL CONDITIONS
          for(n=1;n<=nt;n++)
          {
                  m[n] = mo;
                  mx[n] = m[n]*x[n];
                  if(n>nf)
                          lo[n] = ro;
                  else
                  lo[n]  = ro+f;

          }


  }

  void befc::deriv()
  {
          xbdot = (l[1]*x[1]-v*yb-b*xb)/mbo;
          mdot[1] = l[2]-l[1];
          mxdot[1] = v*(yb-y[1]) + l[2]*x[2] - l[1]*x[1];

           // this is for trays below feed tray
          for(n=2;n<nf;n++)
          {
                  mdot[n] = l[n+1] - l[n];
                  mxdot[n] = v*(y[n-1]-y[n])+l[n+1]*x[n+1]-l[n]*x[n];
          }

           // this is for feed tray
          mdot[nf] = l[nf+1] - l[nf]+f;
          mxdot[nf] = v*(y[nf-1] - y[nf])+l[nf+1]*x[nf+1]-l[nf]*x[nf]+f*z;

           //this for trays above feed tray
          for(n = nf+1; n<nt; n++)
          {
                  mdot[n] = l[n+1] - l[n];
                  mxdot[n] = v*(y[n-1]-y[n])+l[n+1]*x[n+1]-l[n]*x[n];
          }
```

```
                //this for nth tray
                mdot[nt] = r - l[nt];
                mxdot[nt] = v*(y[nt-1]-y[nt])+r*xd-l[nt]*x[nt];
                xddot = v*(y[nt]-xd)/mdo;


        }

        //Euler integration
        void befc::integ()
        {
                time += delta;
                xb += xbdot*delta;
                for(n=1;n<=nt;n++)
                {
                        m[n] += mdot[n]*delta;
                        mx[n] += mxdot[n]*delta;
                        x[n] = mx[n]/m[n];
                        if(x[n]>1.0||x[n]<0.0)
                        {
                                cout<<"\n level too low or
                                        composotion unreal1->"<<x[n]<<" "<<n;
                                getch();
                                exit(0);
                        }
                }
                xd += xddot*delta;
                erintd += errd*delta;
                erintb += errb*delta;

        }

        float befc::equil(float xx)
        {
                return(alpha*xx/(1.+(alpha-1.)*xx));

}

        void befc::maincal()
        {
                ofstream fout("outbin.dat");
                ofstream vcm("graf7.dat");
                fout<<"\n RESULTS FROM Bi_DC % \n %";
                init();
                cout<<"  PRESS ANY KEY TO GO TO MAIN MENU";
                fout<<" TIME \t XB \t  X10 \t XD \t  r \t V \n %";


                // Tray liquid hydralics and vle
                while(time<10)
                {
                  for(n = 1; n<=nt; n++)
```

```
                  {
                               y[n] = equil(x[n]);
                               l[n] = lo[n] + (m[n] - mo)/beta;
                  }
                  yb = equil(xb);

                  //***Two PI feedback controllers ***///
                        errb = 0.02 - xb;
                        errd = .98 - xd;
                        v = vo - kcb * (errb + erintb/taub);
                        r = ro + kcd * (errd + erintd/taud) ;

                        //***Perfect level controllers in reflux***//
                        //***drum and column base****//
                        d = v - r;
                        b = l[1] - v;
                        if(r>0.&&d>0.&&b>0.&&v>0.)
                        {
                          deriv();
                          if(time>=tprint)
                          {
                                        cout<<"\n "<<time<<"  "<<xb<<"  "<<x[10];
                                        cout<<" "<<xd<<" "<<r<<"  "<<v;
                                        fout<<"\n "<<time<<"  "<<xb<<"  "<<x[10];
                                        fout<<" "<<xd<<" "<<r<<"  "<<v<<"%";
                                        tprint += 0.50;
                                        vcm<<"\n "<<time<<"  "<<v;
                          }
                          integ();
                        }
                        else
                        {
                                cout<<"\n level too low or composotion unreal";
                        getch();
                                exit(0);
                        }
                }
                fout<<"\n % \n  %";
        }

void main(void)
{
        clrscr();
        convert();
        setbkcolor(9);
        class befc binc;
        binc.maincal();
        getch();
        setbkcolor(0);
}
```

APPENDIX – B

```
///*****************************************************************************************///
///*****************MULTI-DISTILATION COLUMN DYNAMICS ******************///
///********************INPUTS  R , QR and DV are fixed ***************************///
///*****************************************************************************************///
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
#include<math.h>
#include<graphics.h>
#include<process.h>
#define N 30
#define N1 10
int nt,nf,nc;
float c2[N1] ,c3[N1], c1[N1],lo[N],whs, wls, ds, whr, wlr, dr, mvb, mvd,r,dl;
float mw[N1], dnst[N1],hvap,bpt[N1],hcapv, hcapl, vp1, t1, vp2, t2,b,hlf,fl;
double avp[N],bvp[N];
float tfl,tfv;
float dv,eff,tb,td,mb,densa,mwa,hfow,md,fv,vb,qr,hlb,hvb,hv[N],hvd;
float xf[N1],x[N][N1],xb[N1],xd[N1],t[N],xx[N1],lv[N];
float l[N],mv[N],m[N],xm[N][N1],p[N],pb,pd,y[N][N1],yb[N1];
float yd[N1],v[N],yy[N1],yav[N1],yf[N1],hl[N],dm[N],dxb[N1];
float dxm[N][N1],dxd[N1],hvf,hld;
double delta,time, tprint;
char chr,ch;
class mult_dist
{

        private:
                        int loop,n,i,j,num;
                        float f,sumy,fslope,cnstnt;
                        float tsq,ps[N1];

                        char *name[N1];

        public:
                    mult_dist() {     }
                void readdata1();
                void readdata2();
                void calc();
                void mwdens(float x1[],float*, float *);
                void enth(float *,float x1[],float y1[],float *, float *);
                void hydrau(float *, float *, float x1[],float *, float *
                                                    , float *,float *);
                void bubpt(float*,float *,float *,float *);
                        ~mult_dist()  {   }

 };
```

```
void mult_dist::readdata1()
 {
        cout<<"\n READ NO OF TRAYS, FEED TRAY and NO OF COMPONENTS \n";
        cin>>nt>>nf>>nc;
        cout<<"\n\t READ WEIR HEIGHT, LENGTH and COLUMN DIAMETER IN
              STRIPING SECTION(in)\n\t";
        cin>>whs>>whr>>ds;
        cin>>dr>>wls>>wlr;
        cout<<dr<<" "<<wls<<" "<<wlr<<" ";

        cin>>mvb>>mvd;

        ///**************READ PHYSICAL PROPERTY DATA ***************///
        cout<<"\n\n   ENTER NAME, M-WEIGHT, DENSITY, HVAP, BPT,HCAPV,
              HCAPL, VP1, T1 VP2, T2 \n";

        for(j=1;j<=nc;j++)
        {
                cin>>mw[j]>>dnst[j]>>hvap>>bpt[j]
                >>hcapv>>hcapl>>vp1>>t1>>vp2>>t2;
                avp[j] = (t1+460.)*(t2+460.)*log(vp2/vp1)/(t1-t2);
                bvp[j] = log(vp2)-avp[j]/(t2+460.);
                c2[j] = hcapv*mw[j];
                c3[j] = hcapl*mw[j];
                c1[j] = hvap*mw[j]+(c3[j]-c2[j])*bpt[j];
                yy[j] = 0.0;
        }
        ////********READ FEED************////
        cout<<"\n\n READ FEED ";
        //float tfl;
        cin>>fl>>tfl;
        for(j=1;j<=nc;j++)
        {
                cin>>xf[j];
        }
        cin>>fv>>tfv;
        for(j=1;j<=nc;j++)
        {
                cin>>yf[j];
        }
         enth(&tfl,xf,yf,&hlf,&hvf);

        cout<<"\n\n READ    pd \t  pb \t  qr \t r \t dv \t eff \n";
        cin>>pd>>pb>>qr>>r>>dv>>eff;

 ///*****************************************************************************************///
 ///****************************READ INITIAL CODITIONS******************************///
 //*****READ INITIAL BASE TEMPERATURE & ITS CONCENTRATION ********///
 ///*****************************************************************************************///
```

```
        cout<<"\n\n READ INITIAL BASE TEMPERATURE \n\n";
        float blank;
        cin>>tb>>blank;
        for(j=1;j<=nc;j++)
        {
                cin>>xb[j];
        }
        getch();

    //*READ INITIAL TEMPERATURE, FLOW RATE & CONCENTRATION FOR //**ALL
            TRAYS**//

        cout<<"\n READ INITIAL TEMPERATURE, FLOW RATE & CONCENTRATION
             FOR ALL TRAYS \n ";
        for(n=1;n<=nt;n++)
        {
                cin>>num>>t[n]>>lo[n];
                for(j=1;j<=nc;j++)
                {
                        cin>>x[n][j];
                }
        }

//*READ INITIAL REFLUX DRUM TEMPERATURE & ITS CONCENTRATION*//
        cout<<"\n READ INITIAL REFLUX DRUM TEMPERATURE \n";
        cin>>td>>r;
        for(j=1;j<=nc;j++)
        {
                cout<<"\n base concentration of '"
           <<n<<"'component xb["<<n<<"]= ";
                cin>>xd[j];

        }
        getch();
}


void mult_dist::readdata2()
{
        ///************* READ COLUMN DATA ***************///
        clrscr();
        ifstream fin("mult.dat");
        fin>>nt>>nf>>nc;
        cout<<"\n "<<nt<<"\t"<<nf<<"\t"<<nc<<"\t";
        cout<<"\n\t READ WEIR HEIGHT, LENGTH and COLUMN DIAMETER IN
              STRIPING SECTION(in)\n\t";
        fin>>whs>>whr>>ds;
        cout<<whs<<" "<<whr<<" "<<ds<<" ";
        fin>>dr>>wls>>wlr;
        cout<<dr<<" "<<wls<<" "<<wlr<<" ";
        fin>>mvb>>mvd;
        cout<<mvb<<" "<<mvd;
```

```
    ///*************READ PHYSICAL PROPERTY DATA *************///
    cout<<"\n\n   ENTER NAME, M-WEIGHT, DENSITY, HVAP, BPT,HCAPV,
         HCAPL, VP1, T1 VP2, T2 \n";

    for(j=1;j<=nc;j++)
    {

         fin>>mw[j]>>dnst[j]>>hvap>>bpt[j]>>hcapv
           >>hcapl>>vp1>>t1>>vp2>>t2;

     cout<<"\n"<<mw[j]<<"  "<<dnst[j]<<"  "<<hvap<<"  "<<bpt[j]<<"  "<<hcapv<<"
              "<<hcapl<<" "<<vp1<<"  "<<t1<<" "<<vp2<<"  "<<t2;

          avp[j] = (t1+460.)*(t2+460.)*log(vp2/vp1)/(t1-t2);
          bvp[j] = log(vp2)-avp[j]/(t2+460.);
          c2[j] = hcapv*mw[j];
          c3[j] = hcapl*mw[j];
          c1[j] = hvap*mw[j]+(c3[j]-c2[j])*bpt[j];
          yy[j] = 0.0;
    }
    ////********READ FEED************////
    cout<<"\n\n READ FEED ";
    //float tfl;
    fin>>fl>>tfl;
    cout<<fl<<"\t"<<tfl;
    for(j=1;j<=nc;j++)
    {
           fin>>xf[j];
           cout<<"\t"<<xf[j];
    }
    //float tfv;
    fin>>fv>>tfv;
    cout<<"\n"<<tfv<<"\t"<<fv;
    for(j=1;j<=nc;j++)
    {
           fin>>yf[j];
           cout<<"\t"<<yf[j];
    }
     enth(&tfl,xf,yf,&hlf,&hvf);

     cout<<"\n\n READ    pd \t  pb \t  qr \t r \t dv \t eff \n";
     fin>>pd>>pb>>qr>>r>>dv>>eff;
     cout<<"\t"<<pd<<"  "<<pb<<"  "<<qr<<"  "<<r<<"  "<<dv<<"  "<<eff;


///*********************************************************************************///
///***************************READ INITIAL CODITIONS**************************///
//*****READ INITIAL BASE TEMPERATURE & ITS CONCENTRATION *******///
///*********************************************************************************///

     cout<<"\n\n READ INITIAL BASE TEMPERATURE \n\n";
```

```
        float blank;
        fin>>tb>>blank;
        cout<<tb<<"\t"<<blank;
        for(j=1;j<=nc;j++)
        {
                fin>>xb[j];
                cout<<"\t"<<xb[j];
        }
        getch();

//*READ INITIAL TEMPERATURE, FLOW RATE & CONCENTRATION FOR**//
                //**ALL TRAYS*//

    cout<<"\n READ INITIAL TEMPERATURE, FLOW RATE &
              CONCENTRATION FOR ALL TRAYS \n ";
        for(n=1;n<=nt;n++)
        {
                fin>>num>>t[n]>>lo[n];
                cout<<"\n"<<num<<" "<<t[n]<<"  "<<lo[n];
                for(j=1;j<=nc;j++)
                {
                        fin>>x[n][j];
                        cout<<"  "<<x[n][j];
                }
        }

//*READ INITIAL REFLUX DRUM TEMPERATURE & ITS CONCENTRATION**//


        cout<<"\n READ INITIAL REFLUX DRUM TEMPERATURE \n";
        fin>>td>>r;
        cout<<td<<"  "<<r;
        for(j=1;j<=nc;j++)
        {
                fin>>xd[j];
                cout<<"\t"<<xd[j];

        }
        getch();


}


void mult_dist::calc()
{
        ///***** main loop for each time step*****///
        gotoxy(2,5);
        cout<<"\n U want to give your own data otherwise it will take its own data ";
        cin>>chr;
        if(chr=='y')
        {
```

```
        readdata1();
}
else
{
        readdata2();
}
////**********CALCULATE INTIAL HOLDSUPS*******************////
mwdens(xb,&mwa,&densa);
mb = mvb*densa/mwa;
for(n=1;n<=nf;n++)
{
        for(j=1;j<=nc;j++)
        {
                xx[j] = x[n][j];
        }
        mwdens(xx,&mwa,&densa);
        lv[n] = lo[n]*mwa/densa;
        l[n] = lo[n];
        hfow = pow((lv[n]/(999.*wls)),0.66667);
        mv[n] = (hfow+whs/12.)*3.1416*ds*ds/(4.*144.);
        m[n] = mv[n]*densa/mwa;
}

for(n=nf+1;n<=nt;n++)
{
        for(j=1;j<=nc;j++)
        {
                xx[j] = x[n][j];
        }
        mwdens(xx,&mwa,&densa);
        lv[n] = lo[n]*mwa/densa;
        l[n] = lo[n];
        hfow = pow((lv[n]/(999.*wlr)),0.66667);
        mv[n] = (hfow+whr/12.)*3.1416*dr*dr/(4.*144.);
        m[n] = mv[n]*densa/mwa;
}

for(n=1;n<=nt;n++)
{
        for(j=1;j<=nc;j++)
        {
                xm[n][j] = m[n]*x[n][j];
        }
}

mwdens(xd,&mwa,&densa);
md = mvd*densa/mwa;

///*************CALCULATE PRESSURE PROFILE *******************///
for(n=1;n<=nt;n++)
{
        p[n] = pb - n*(pb - pd)/nt;
```

```
        }

        delta = 0.0001;
        time = tprint = 0.0;

///*****INITIAL GUESS OF V(5) FOR FIRST EFFICIENCY CALC.********///
        v[5] = 822.;
        while((time<0.0011))
        {

                bubpt(&tb,xb,yb,&pb);
                enth(&tb,xb,yb,&hlb,&hvb);

                for(j=1;j<=nc;j++)
                {
                        xx[j]=x[1][j];
                }
                bubpt(&t[1],xx,yy,&p[1]);
                for(j=1;j<=nc;j++)
                {
                        y[1][j] = yb[j] + eff*(yy[j]-yb[j]);
                        yy[j] = y[1][j];
                }
                enth(&t[1],xx,yy,&hl[1],&hv[1]);
                for(n=2;n<=nf;n++)
                {
                        for(j=1;j<=nc;j++)
                        {
                                xx[j] = x[n][j];
                        }
                        bubpt(&t[n],xx,yy,&p[n]);
                        for(j=1;j<=nc;j++)
                        {
                                y[n][j] = (yy[j] - y[n-1][j])*eff + y[n-1][j];
                                yy[j] = y[n][j];
                        }
                        enth(&t[n],xx,yy,&hl[n],&hv[n]);
                }

                for(j=1;j<=nc;j++)
                {
                        xx[j] = x[nf+1][j];
                }
                bubpt(&t[nf+1],xx,yy,&p[nf+1]);
                for(j=1;j<=nc;j++)
                {
                        yav[j] = ( yf[j]*fv + y[nf][j]*v[nf] ) / (v[nf]+fv);
                        y[nf+1][j] = (yy[j] - yav[j])*eff + yav[j];
                        yy[j] = y[nf+1][j];
                }
                enth(&t[nf+1],xx,yy,&hl[nf+1],&hv[nf+1]);
                for(n=nf+2;n<=nt;n++)
```

```
        {
                for(j=1;j<=nc;j++)
                {
                        xx[j] = x[n][j];
                }
                bubpt(&t[n],xx,yy,&p[n]);

                for(j=1;j<=nc;j++)
                {
                        y[n][j] = (yy[j]-y[n-1][j])*eff+y[n-1][j];
                        yy[j] = y[n][j];
                }
                enth(&t[n],xx,yy,&hl[n],&hv[n]);
        }
        bubpt(&td,xd,yd,&pd);
        enth(&td,xd,yd,&hld,&hvd);

        ///**************************************************************************///
        ///*************CALCULATE VAPOUR RATES**********************///
///**************************************************************************///

        vb = (qr*1000000.0 - l[1]*(hlb - hl[1]))/(hvb - hlb);
        b = l[1] - vb;

        if(b < 0.0)
        {
                break;
        }
          //cout<<"\n b "<<b;

        v[1] = ( hl[2]*l[2] + hvb*vb - hl[1]*l[1] )/ hv[1];
        for(n=2;n<nf;n++)
        {
                v[n] = (hl[n+1]*l[n+1] + hv[n-1]*v[n-1] - hl[n]*l[n])/hv[n];
        }

        v[nf] = (hl[nf+1]*l[nf+1] + hv[nf-1]*v[nf-1] - hl[nf]*l[nf] + hlf*fl) /hv[nf];
        v[nf+1]=(hl[nf+2]*l[nf+2] + hv[nf]*v[nf] + hvf*fv - hl[nf+1]*l[nf+1])
                        /hv[nf+1];

        for(n=nf+2;n<nt;n++)
        {
              v[n] = (hl[n+1]*l[n+1] + hv[n-1]*v[n-1] - hl[n]*l[n])/hv[n];
        }
        v[nt] = (hld*r + hv[nt-1]*v[nt-1] - hl[nt]*l[nt])/hv[nt];

        dl = v[nt] - dv - r;

        ///*********************************************************************///
        ///*************EVALUATE DERIVATIVES**********************///
///*********************************************************************///
        dm[1] = l[2] + vb - v[1] - l[1];
```

```
for(n=2;n<nf;n++)
{
        dm[n] = l[n+1] + v[n-1] - l[n] - v[n];
}

dm[nf] = l[nf+1] + fl + v[nf-1] - l[nf]-v[nf];
dm[nf+1] = l[nf+2] + fv + v[nf] - l[nf+1] - v[nf+1];

for(n=nf+2;n<nt;n++)
{
        dm[n] = l[n+1] + v[n-1]-l[n]-v[n];
}
dm[nt] = r + v[nt-1] - l[nt]-v[nt];
for(j=1;j<=nc;j++)
{
        dxb[j] = (x[1][j]*l[1] - yb[j]*vb - xb[j]*b)/mb;
        dxm[1][j] = x[2][j]*l[2] + yb[j]*vb - x[1][j]*l[1] - y[1][j]*v[1];
        for(n=2;n<nf;n++)
        {
                dxm[n][j] = x[n+1][j]*l[n+1] + y[n-1][j]*v[n-1]
                                             -x[n][j]*l[n] - y[n][j]*v[n];
        }

        dxm[nf][j] = x[nf+1][j]*l[nf+1] + y[nf-1][j]*v[nf-1]
                          -x[nf][j]*l[nf] - y[nf][j]*v[nf] + fl*xf[j];

        dxm[nf+1][j] = x[nf+2][j]*l[nf+2] + y[nf][j]*v[nf]

                            -x[nf+1][j]*l[nf+1] - y[nf+1][j]*v[nf+1] + fv*yf[j];

        for(n=nf+2;n<nt;n++)
        {
                dxm[n][j] = x[n+1][j]*l[n+1] + y[n-1][j]*v[n-1]
                                  -x[n][j]*l[n] - y[n][j]*v[n];

        }
        dxm[nt][j] = xd[j]*r + y[nt-1][j]*v[nt-1]
                         -x[nt][j]*l[nt] - y[nt][j]*v[nt];
        dxd[j] = (v[nt]*y[nt][j] - dv*yd[j] - (r+dl)*xd[j])/md;
}
if(time>=tprint)
{
        getch();
        clrscr();
        cout<<"\n TIME\tTB\tX1\tX2\tX3\tX4\tX5\tL\n";
        cout<<"\n"<<time<<" "<<tb<<" ";
        for(j=1;j<=nc;j++)
         cout<<xb[j]<<"  ";
        cout<<b;
        for(n=1;n<=nt;n++)
        {
```

```
                cout<<"\n"<<n<<" "<<t[n]<<" ";
                for(j=1;j<=nc;j++)
                 cout<<x[n][j]<<" ";
                cout<<l[n];
        }
        cout<<"\n"<<td<<" ";
        for(j=1;j<=nc;j++)
         cout<<xd[j]<<" ";
        cout<<r<<"\n  ";

        for(j=1;j<=nc;j++)
         cout<<yd[j]<<" ";
        cout<<dl;
        tprint += 0.0005;
}


///******************************************************************///
////*****INTEGRATION USING EULER METHOD    ********///
///******************************************************************///
time = time + delta;
for(n=1;n<=nt;n++)
 m[n] = m[n] + dm[n]*delta;
for(j=1;j<=nc;j++)
{
        xb[j] += dxb[j]*delta;
        if(xb[j]<=1.0e-9) xb[j] = 0.;
        if(xb[j]>1.0) xb[j] = 1.;
        for(n=1;n<=nt;n++)
        {
                xm[n][j] += dxm[n][j]*delta;
                x[n][j] = xm[n][j]/m[n];
                if(x[n][j]<=1.0e-9) x[n][j] = 0.;
                if(x[n][j]>1.) x[n][j] = 1.;
        }
        xd[j] += dxd[j]*delta;
        if(xd[j]<=1.0e-9) xd[j] = 0.;
        if(xd[j]>1.) xd[j] = 1.;
}

///********CALCULATE NEWLIQUID RATES**************///
for(n=1;n<=nf;n++)
{
        for(j=1;j<=nc;j++)
        {
                xx[j] = x[n][j];
        }
        hydrau(&m[n],&t[n],xx,&l[n],&whs,&wls,&ds);
}

for(n=nf+1;n<=nt;n++)
{
```

```
                        for(j=1;j<=nc;j++)
                        {
                                xx[j] = x[n][j];
                        }
                        hydrau(&m[n],&t[n],xx,&l[n],&whr,&wlr,&dr);
                }
        }

} ///****function calcu ends here ****///


void mult_dist::hydrau(float *mn,float *t1,float x1[],float *ln, float *wh,
                                                float *wl, float *dcol)
{

        mwdens(x1,&mwa,&densa);
        cnstnt = 183.2 * (*mn) * mwa/( densa * (*dcol) * (*dcol) )-(*wh)/12.;
        if(cnstnt > 0.0)
        {

                (*ln) =  densa*(*wl)*999.0 * pow(cnstnt,1.5)/mwa;
        }
        else
         (*ln) = 0.0;
        return;

}


void mult_dist::enth(float *t1,float x1[],float y1[],float *hl1,float *hv1)
{
        *hl1=0.0;
        *hv1=0.0;
        for(i=1;i<=nc;i++)
        {
                *hl1 += (x1[i])*c3[i]*(*t1);
                *hv1 += (y1[i])*(c1[i] + c2[i]*(*t1));
        }
        return;
}

void mult_dist::mwdens(float x1[],float *mwta,float *dnsta)
{
        *dnsta = 0.0;
        *mwta = 0.0;
        for(i=1;i<=nc;i++)
        {
                *mwta += x1[i]*mw[i];
                *dnsta += x1[i]*dnst[i];
        }
        return;
```

```
}

void mult_dist::bubpt(float *t1,float x1[], float y1[], float *p1)
{

        for(loop=1;loop<51;loop++)
        {
                sumy = 0.0;
                for(i=1;i<=nc;i++)
                {
                        ps[i] = exp(bvp[i] + avp[i]/((*t1) + 460.0));
                        y1[i] = ps[i]*x1[i]/(*p1);
                        sumy += y1[i];
                }
                if(fabs(sumy - 1.) < .00005)
                 return;
                f = (sumy - 1.)*(*p1);

                fslope = 0.0;

                tsq = pow(((*t1) + 460.0),2);

                for(i=1; i<=nc; i++)
                {
                        fslope -= avp[i]*x1[i]*ps[i]/tsq;
                }
                (*t1) -= f/fslope;
        }
        return;
}


void main(void)
{
        class mult_dist ml;
        clrscr();
        ch = 'y';
        while(ch == 'y')
        {
                ml.calc();
                getch();
                clrscr();
                gotoxy(3,5);
                cout<<"\n You want to simulate with an other data if no it return ";
                cout<<"\n\t\t to main menu \n";
                gotoxy(30,10);
                cin>>ch;
                clrscr();
        }

}
```

APPENDIX – C

```
///***************************************************************************************///
///USER FRIENDLY SOFTWARE PROGRAM FOR DYNAMIC SIMULATION///
///********************** OF DISTILLATION PROCESS **************************///
///**************************************************************************************///

#include<dos.h>
#include<bios.h>
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<process.h>
#include<fstream.h>
#include<io.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>
#include"mouse.c"
int x,y,button,bd[5];
int ch,e,ep,op1,p[10] = {0,0,0,0,0},en;
long int rd;
int pd[3][2],pq,l;
struct time q;


static char* icn[] = {" BINARY CMP"," MULTI CMP","EXIT&HELP"};
static char *elements[][2] = { "  bi-component","  results ",
                            "  Ml-component", "  Results","  help", "  Exit to DOS"};

static char *direct[] =  { "program for binary & corresponding results of program",
                           "program for multi & corresponding results of program",
                           "gives details of program &  exit to DOS" };

char* str ;
char *to;
char *word = "    ";
char ft[80];

class wind {
  private:
    int cr1,cr2,size,block,hi,lo,i,j,CR,ystart,yend;

  public:
         wind() { }
    void icon(int,int,int,int,int,int,int);
    void maindow(int,int,int,int,int,int);
    int getkey();
```

```
  void wel();
  void end();
  ~wind() {  }
};


void wind::icon(int x1,int y1,int x2,int y2,int c1,int c2,int c3)
{

           setcolor(c3);
           rectangle(x1-1,y1-1,x2+1,y2+1);
           setfillstyle(1,c3);

           floodfill((x1+x2)/2,(y1+y2)/2,c3);
           setcolor(c1);
           line(x1-2,y2+2,x1-2,y1-2);
           line(x1-2,y1-2,x2+2,y1-2);
           setcolor(c2);
           line(x1-2,y2+2,x2+2,y2+2);
           line(x2+2,y2+2,x2+2,y1-2);

       }

int wind::getkey()
{
                if(kbhit())
                {
                  ch = bioskey(0);
                  hi = ch >> 8;
                  lo = ch &  0x00FF;
                  if(lo == 0)
                   { return(300 + hi); }
                  else
                   { return(lo);   }
                }
}


void wind::wel()
{

       button=2;
       i = 1;
       setfillstyle(9,7);
       bar(10,10,630,470);
       icon(70,200,560,280,15,5,1);
       line(10,10,70,200);
       line(10,470,70,280);
       line(630,10,560,200);
       line(630,470,560,280);
       gettime(&q);
       pq = q.ti_hour;
```

```
        if(pq > 3 && pq <= 12)
           to =  "GOOD MORNING";
          if(pq>12&&pq<=15)
             to = "GOOD AFTER NOON ";
          if(pq>15&&pq<=21)
              to = "GOOD EVENING ";
           if(pq >21|| pq<5)
                  to = "GOOD NIGHT ";
        setcolor(15);
        outtextxy(250,450,"CLICK LEFT BUTTON");
        while(button!=1)
        {
          setcolor(i);
          settextstyle(7,0,5);
          //outtextxy(165,200,"WELCOME");
          outtextxy(70+i*5,210, to);
          delay(60);
          setcolor(1);
          outtextxy(70+i*5,210,to);
          i++;
          if(i>22)
          i = 1;
          if(button==1)
          break;
          getmousepos(&button,&x,&y);
        }
}


void wind::end()
{
        hidemouseptr();
        cleardevice();
        button=2;
        setfillstyle(9,7);
        bar(10,10,630,470);
        icon(100,200,520,280,15,5,1);
        line(10,10,100,200);
        line(10,470,100,280);
        line(630,10,520,200);
        line(630,470,520,280);
        setcolor(15);
        i = 1;
        outtextxy(250,450,"CLICK LEFT BUTTON");
        while(button!=1)
        {
          setcolor(i);
          settextstyle(8,0,6);
          outtextxy(100+i*5,200, "THANK YOU");
          delay(60);
          setcolor(1);
```

```
          outtextxy(100+i*5,200,"THANK YOU");
          i++;
          if(i>15)
          i = 1;
          if(button==1)
          break;
          getmousepos(&button,&x,&y);
         }
 }
```

```
class wind1: public wind {
    private:
     int i,j,k,area,d,l,adc,adc1,adc2;
     char gtch;
    public:
     wind1() {  }

     void icons()
     {
              for(j=1;j<4;j++)
              {
                wind::icon(130*j-90,40,130*j,55,7,7,7);
                setcolor(BLACK);
                settextstyle(2,0,5);
                outtextxy(130*j-84,40,icn[j-1]);

              }
        }



        void print(char* str)
        {
              ifstream pin(str);
              d =0;
              while(d<10)
              {
                      l = 0;
                      setcolor(15);
                for(j = 1;j<16;j++)
                {
                              pin>>word;
                              i = strlen(word);
                              if(word[0] == '&')
                              {
                                      l = 2;
                              }
```

```
                        for(k=0;k<i;k++)
                        {
                                ft[l+k] = word[k];
                        }
                        ft[l+i] = ' ';
                        if(word[0] == '&')
                        {
                          for(k=0;k<=2;k++)
                                 ft[k] = ' ';
                        }
                        if(word[i-1] == '%')
                        {
                                for(k=l+i-1;k<61;k++)
                                ft[k] = ' ';
                                break;
                        }
                        l += i+1;
          }
          outtextxy(100,106+(d)*30,ft);
          d++;
        }
}

void plot(char *str,int k1, int k2)
{

        ifstream tin(str);
        setcolor(59);
        icon(60,106,570,390,0,7,0);
        setcolor(WHITE);
        line(80, 370, 500, 370);
        line(100, 107, 100, 393);
        moveto(100,375);
        ftsgnl = 0;
        ftanlg = 0;
        settextstyle(2,0,4);
        setcolor(GREEN);
        for(j=0;j<19;j++)
        {
                d = 100 + int (ftsgnl*k1);
                k = 370 - int (ftanlg*k2);
                lineto(d,k);
                tin>>ftsgnl;
                tin>>ftanlg;
        }
        d = (400) /10;
        k = (300)/10;
        setcolor(15);
        for(i=100; i<500;i += d)
        {
          line(i,367,i,373);
```

```
          }
          settextstyle(2,1,4);
          for(i=370;i>100;i-=k)
          {
            j = 370-i;
            line(97,i,103,i);
          }


}

void store(int x1,int y1)
{

          setfillstyle(1,7);
          bar(x1+ 6,y1+35,620,y1+150);

}

void restore()
{

          setfillstyle(1,7);
          bar(9,103,637,400);

}

void  clean()
{
          setfillstyle(1,7);
          bar(2,445,638,473);
}

void mainbox()
{
          char chp;
          chp = 'r';
          while(chp == 'r')
          {
          chp = 'l';
          for(i=1;i<4;i++)
           for(j=0;j<2;j++)
                    pd[i][j] = 0;
          wind::wel();
          cleardevice();
          hidemouseptr();
          bd[0] = bd[1] = 1;
          wind::icon(0,10,638,440,7,7,7);
          wind::icon(1,61,637,440,15,0,7);    //\\main window//\\
          icon(10,69,628,432,0,15,7);
          wind::icon(2,40,638,55,7,7,7);           //\\command window//\\
          wind::icon(2,445,637,475,4,4,7);          //\\direction window//\\
```

```
setcolor(1);
settextstyle(3,0,4);
outtextxy(230,440,"MAIN MENU");
icon(2,12,640,32,9,9,9);
icons();
setcolor(15);
settextstyle(2,0,6);
outtextxy(50,13,"DYNAMIC SIMULATION OF BINARY DISTILLATION
PROCESS");
showmouseptr();
moveto(200,200);
e = 0;
rd= 0;
en = 0;
bd[0] = 10;
bd[1] = 10;
while(1)
{
        rd++;
        if(rd==1500)
        {
          en++;
          hidemouseptr();
          setfillstyle(1,1);
          bar(580,10,640,34);
          setcolor(3);
          setfillstyle(1,10);
          fillellipse(590+en*5,20, 10, 5);
          showmouseptr();
        }

   if(rd==3000)
   {
           hidemouseptr();
           setfillstyle(1,1);
           bar(580,10,640,34);
           /*settextstyle(1,1,1);
           setcolor(10);
           outtextxy(610,10+en*2,"V");*/
           setcolor(9);
           setfillstyle(1,9);
           fillellipse(600+en*5,20,3,10);
           rd =0;
           nosound();
           showmouseptr();
         }
         if(en>4)
          en = 0;
         getmousepos(&button,&x,&y);
         if(y>30&&y<55)
         for(i=1;i<4;i++)
          if(x>130*i-90&&x<130*i&&y>30&&y<55)
```

```
{
        p[i] += 1;
        e = i;
        bd[0] = bd[1] = 10;
        break;
}
if(p[e] ==500) p[e] = 10;
if(e>4) e = 0;

for(j=1;j<=4;j++)
 if(j!=e) p[j] = 0;
switch(e)
{
case 1:
case 2:
case 3:
  {
        if(p[i]!=1) break;
        hidemouseptr();
        icons();
        store(10,33);
        outtextxy(130*e-84,40,icn[e-1]);
        wind::icon(130*i-90,40,130*i,55,0,15,7);          //\\command
icon//\\
        wind::icon(130*i-110,70,130*i+20,120,15,0,7);
//\\options//\\
        icon(130*e-108,97,130*e+18,94,0,15,7);
        setcolor(0);
        settextstyle(2,0,5);
        for(k=0;k<2;k++)
        {
                outtextxy(130*e-110,75+k*25,elements[e-1][k]);
                pd[e][k] = 0;
        }
        clean();
        wind::icon(2,445,637,477,4,4,7);                    //\\direction
window//\\
        setcolor(1);
        settextstyle(1,0,1);
        outtextxy(130,455," Select one and click left button");
        outtextxy(100,440,direct[e-1]);
        showmouseptr();
  }
  break;
}

getmousepos(&button,&x,&y);
for(k=0;k<2;k++)
{
      if((x>130*e-110)&&x<(130*e+20)
        &&y>(75+k*25)&&y<(95+k*25))
  {
```

```
          op1 = k;
          pd[e][op1] = 2;
          break;
  }
}

for(j=1;j<4;j++)
{
          for(ep=0;ep<2;ep++)
           if(j!=e&&ep!=op1)
                    pd[j][ep] = 0;
}

getmousepos(&button,&x,&y);
if((e==1||e==2||e==3||e==4)&&pd[e][op1]>1)
if((x>130*e-110)&&x<(130*e+20)
  &&y>(75+op1*25)&&y<(95+op1*25))
 {
           settextstyle(2,0,5);
           if(bd[op1]>2)
           {
                    hidemouseptr();
                    bd[op1] = 0;
                    setfillstyle(1,6);
                    bar(130*e-110,73+op1*25, 130*e+20,92+op1*25);
                    setcolor(15);
                    outtextxy(130*e-110,75+op1*25,
                              elements[e-1][op1]);
                    if(op1 == 0) k = 1;
                    if(op1 == 1) k = 0;
                    bd[k] = 10;
                    setfillstyle(1,7);
                    bar(130*e-110,73+k*25,130*e+20,92+k*25);
                    setcolor(0);
                    outtextxy(130*e-110,75+k*25,elements[e-1][k]);
                    showmouseptr();
           }
 }


 showmouseptr();
 nosound();
 /////********** selecting options ends here ****************///
 /////////Bellows are for performing various functions////
 ////according to selection options////
 switch(e)
 {
  case 1:
  case 2:
           getmousepos(&button,&x,&y);
           if(op1==1)
           {
```

```
                if(pd[e][op1]>0)
                if(button == 1 )
                {
                        hidemouseptr();
                        if(e==1)
                                system("outbin.exe");
                        else
                                system("outmult.exe");
                        p[e] = 10;
                        e = 0;
                        chp = 'r';
                }
        }
        else
        if(op1==0)
        {
                if(pd[e][op1]>0)
                if(button == 1 )
                {
                        hidemouseptr();
                        cleardevice();
                        if(e==1)
                                system("BINDS.EXE");
                        else
                                system("MULTDS.EXE");
                        p[e] = 10;
                        e = 0;
                        chp = 'r';
                }
        }
        break;

case 3:
                getmousepos(&button,&x,&y);
                if(op1==0)
                {
                        if(pd[e][op1]>0)
                          if(button == 1 )
                          {
                                  hidemouseptr();
                                  system("HELP.EXE");
                                  p[e] = 10;
                                  e = 0;
                                  chp = 'r';
                          }
                }
                else
                if(op1==1)
                {
                        if(pd[e][op1]>0)
                        if(button == 1 )
                        {
```

```
                                        wind::end();
                                        closegraph();
                                        exit(1);
                                    }
                                }
                                break;

                        } //SWITCH STAPTEMENT ENDS HERE
                        if(chp =='r')
                        break;

                        if(y>94&&(e==1||e==2||e==3||e==4))
                        {
                                pd[e][op1]=0;
                        }
                        getmousepos(&button,&x,&y);
                        if(y>94&&(e==1||e==2||e==3||e==4)&&button==1)
                        {
                                icons();
                                wind::icon(130*e-90,40,130*e,55,7,7,7);

                                store(10,33);
                                op1=10;
                                e = 0;
                                p[i] = 0;
                        }

                }        //inner loop
            }            //outer loop
        }

        ~wind1() {  }

}; // derived class ends


void main(void)
{
        class wind1 *pin;
        class wind *pin2;
        int gd, gm;
        initmouse(&gd, &gm, "C:\\TC\\BGI");
        cleardevice();
        restrictmouseptr(5,3,635,400);
        pin2->wel();
        pin->mainbox();
        getch();
        closegraph();

}
```