MISAGH EBRAHIMPOUR

Dynamic and Steady State Modeling of VRD Column in Equation-Oriented Environment

São Paulo

2015

MISAGH EBRAHIMPOUR

Dynamic and Steady State Modeling of VRD Column in Equation-Oriented Environment

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia

Área de Concentração: Engenharia Química

Orientador: Prof. Dr. Galo Antonio Carrillo Le Roux

São Paulo

2015

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catalogação-na-publicação

Ebrahimpour, Misagh

Dynamic and Steady State Modeling of VRD Column in Equation Oriented Environment / M. Ebrahimpour -- versão corr. -- São Paulo, 2015. 128 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Química.

1.Vapor Recompression Distillation 2.Modeling 3.Simulation 4.Equation Oriented Environment 5.Reduced Index Formulation I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Química II.t.

ACKNOWLEDGMENTS

First of all, I acknowledge my family who has supported me during this time away from my country by their positive energies and prayers.

I should be grateful to my advisor Professor Galo Antonio Carrillo Le Roux for his help, and the opportunity of studying in his research group.

Special thanks go out to RTO research group members, without whom this dissertation would not have been completed: José Eduardo Alves Graciano, José Otávio Matias and Alvaro Marcelo Acevedo Peña.

Thanks to Petrobras members for their information and consults, especially, Dr. Antonio Carlos Zanin (CETAI).

I also want to thank my friends at the chemical engineering department: Bruno Santoro, Bruno Capron, Andre Yamashita, Aldo Hinojosa, Marion Villacampa, Leandro Goulart and Beatriz Sanabria Arenas.

RESUMO

O propósito desta dissertação é realizar a modelagem, simulação e validação de um modelo, tanto em estado estacionário quanto dinâmico, de uma unidade despropanizadora da Petrobras. Para o estudo foi utilizado o ambiente orientado a equações, EMSO (Environment for Modeling, Simulation and Optimization).

A coluna despropanizadora é uma torre de destilação de alta pureza que possui um comportamento altamente não-linear devido às fortes interações causadas pelo sistema de recompressão do vapor. A modelagem desse processo é um desafio devido às características que apresenta.

Inicialmente, foi desenvolvido nesse ambiente um modelo em estado estacionário, robusto, rápido e preciso, com a finalidade de prover as predições em estado estacionário necessárias para a implementação efetiva de uma rotina de otimização em tempo real, Real Time Optimization (RTO).

A modelagem dinâmica de processos em equilíbrio resulta, frequentemente, em sistemas de equações algébrico diferenciais de índice superior. Para solução do problema de índice normalmente são utilizadas relações fenomenológicas, as quais introduzem novas fontes de erros provenientes de parâmetros e detalhes de projeto desconhecidos. Considerando que a resposta da coluna em relação às mudanças na composição, em geral, ocorre em uma escala de tempo de ordem de grandeza duas vezes mais lenta que as respostas às mudanças nas vazões, foi proposta uma abordagem similar a um controlador proporcional com ganho elevado para substituir as relações fenomenológicas e assim resolver o problema de índice. A estrutura do modelo dinâmico é baseada na do modelo estacionário e contém mais de nove mil equações. A validação dos resultados da simulação com dados reais da planta mostra que a abordagem proposta consegue prever satisfatoriamente o comportamento do sistema.

Palavras-chave: Destilação com recompressão de vapor, modelagem, simulação, ambiente orientado a equações, Formulação de índice reduzido

ABSTRACT

Dynamic and steady state modeling and simulation validation of an industrial depropanizer owned by Petrobras are carried out in the Equation-Oriented environment using EMSO (Environment for Modeling, Simulation and Optimization).

The depropanizer is a high purity distillation column with high nonlinear behavior because of the strong interactions due to the vapor recompression. Furthermore, the difference between internal and external material/energy flows causes a complex multi-time-scale dynamics. Modeling such process is a challenging problem due to these characteristics.
Initially, a steady state model, robust, fast and precise, able to provide steady state predictions that are necessary for effective implementation of Real Time Optimization (RTO) was developed in EMSO.

In addition, the modeling of dynamic equilibrium processes often results in higher index DAE systems. Usually, phenomenological relationships are used to solve the index problem, but this approach gives rise to errors as a result of unknown parameters and project details that are assumed. Considering that the column's response to composition changes, in general, takes place over a timescale one or two order of magnitude slower than those of flow rate changes, an approach similar to a proportional loop with arbitrarily large gain is used as an alternative to solve the index problem. The dynamic model structure is based on the steady state model and contains more than nine thousand equations. Validation simulation results from comparison between real plant data and dynamic model have shown that the proposed approach is able to predict the dynamic behavior of the column properly.

Keywords: Vapor Recompression Distillation, Modeling, Simulation, Equation-Oriented Environment, Reduced Index Formulation

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

ACRONYMS AND ABBREVIATIONS

EMSO	Environment for Modeling, Simulation and Optimization
RTO	Real Time Optimization
VRD	Vapor Recompression Distillation
EO	Equation Oriented
DAE	Differential-Algebraic Equations
ODE	Ordinary Differential Equations
SM	Sequential Modular
BFD	Backward Differentiation Formula
PI	Proportional Integral Controller
MHE	Moving Horizon Estimation

ROMAN SYMBOL

A_a	Tray Active Area
a	Coefficients of Backward Differentiation Formula
B	Steady State Detection Index
B	Bottom flow rate (eq. 4.22)
c	Number of Components
C_t^L	Liquid Total Molar Concentration
C_t^V	Vapor Total Molar Concentration
D	Top Flow rate
d	Series of Data
E	Energy
E_j^L	Transfer Rate Energy per Tray (Liquid Phase)
E_j^V	Transfer Rate Energy per Tray (Vapor Phase)
E^{MV}	Vapor Murphree Efficiency
$\hat{f}_i^{\mathbf{L}}$	Fugacity of Species *i* (liquid Phase)
$\hat{f}_i^{\mathbf{V}}$	Fugacity of Species *i* (Vapor Phase)
F^L	Feed Flow Rate (liquid Phase)
F^V	Feed Flow Rate (Vapor Phase)

G	Parameter (eq. 4.28 and 4.29)
h	Step Size of Backward Differentiation Formula
h^L	Liquid Enthalpy
h^V	Vapor Enthalpy
hv	Vapor Heat Transfer Coefficient
hl	Liquid Heat Transfer Coefficient
hf	Feed Enthalpy
J_{ij}^L	Diffusive Molar Flux of Component *i* in the j^{th} Stage (Liquid Phase)
J_{ij}^V	Diffusive molar Flux of Component *i* in the j^{th} Stage (Vapor Phase)
K	Equilibrium Ratio
k	Controller gain
K^L	Binary Mass Transfer Coefficient (Liquid Phase)
K^V	Binary Mass Transfer Coefficient (Vapor Phase)
L	Liquid Flow rate
$l_k(z)$	Lagrange Polynomial
L_{liq}	Liquid Level
M_{liq}	Liquid Molar Hold-up
M_{vap}	Vapor Molar Hold-up
M	Total Molar Hold-up

N	Number of Trays
n	Degree of Polynomial
η_{cp}	Compressor Efficiency
N_{ij}^{L}	Molar Transfer Rate of Component *i* in the Liquid Phase, j^{th} Stage
N_{ij}^{V}	Molar Transfer Rate of Component i in the Vapor Phase, j^{th} *Stage*
N_{tj}	Total Mass transfer Rate in the j^{th} *Stage*
NE	Number of Measurements
NY	Number of Variables
P	Pressure
P^{L}	Liquid Pressure
P^{V}	Vapor Pressure
Q	Heat
S	Entropy
T	Temperature
T^{L}	Liquid Temperature
T^{V}	Vapor Temperature
T_s	Threshold value
T_w	Threshold value
T_u	Threshold value

u^L	Liquid Internal Energy
u^V	Vapor Internal Energy
V	Vapor Flow rate
V_{tray}	Tray Volume
v^L	Specific Molar volume (Liquid Phase)
v^V	Specific Molar volume (Vapor Phase)
X	Differential Variables
x	Mole Fraction of Components (Liquid Phase)
Y	Variable of Data Reconciliation Objective Function
y	Mole Fraction of Components (Vapor Phase)
y^*	Vapor Ideal Composition
Z	Algebraic Variables

GREEK SYMBOLS

α	Pressure drop coefficient
αI	Total Interfacial Area
β	Coefficients of Backward Differentiation Formula
γ	Parameters of Steady State detection Algorithm
ΔP	Pressure Drop
ΔT	Temperature Difference

Δz	Tray Spacing
θ	Parameters of Steady State detection Algorithm
λs	Adjustment Parameter
ξ	Parameters of Steady state detection Algorithm
ρ	Specific Mass
ρ^L	Liquid Specific Mass
ρ^V	Vapor Specific Mass
σ^2	Variance
τ	Integral time
$\widehat{\Phi}_i^L$	Fugacity Coefficient of Species *i* (Liquid Phase)
$\widehat{\Phi}_i^V$	Fugacity Coefficient of Species *i* (Vapor Phase)

SUBSCRIPTS AND SUPERSCRIPTS

C	Cold Stream
cp	Compressor
eq	Equilibrium
H	Hot Stream
in	Inlet Stream
L	Liquid Property
m	Measured Data

out	Outlet Stream
r	Rectification Zone
re	Reconciled Data
s	Stripping Zone
S	Entropy
sat	Saturated
sub	Subcooled
V	Vapor property
*	Ideal Condition

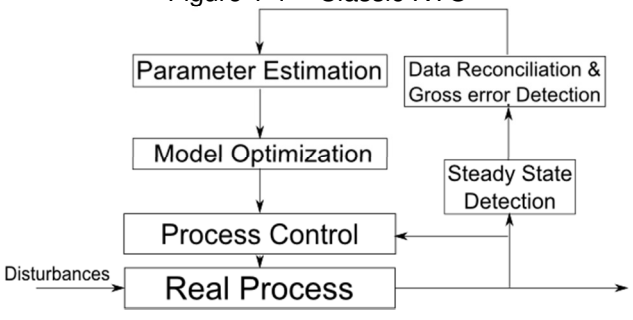
TABLE OF CONTENTS

1. INTRODUCTION

In the chemical process industry, real-time optimization (RTO) is a crucial tool to increase process profitability in scenarios where feedstock composition and market price variations demand rapid adaptation of the process conditions. The benefits achieved from RTO implementation in a process plant depend on the value of increased processing capacity, differences among product prices, specific energy consumption, number of independent variables and constraints, and the ability to accurately model process responses (WHITE, 1997). To date, there are around 300 RTO applications (DARBY et al., 2011) spanning a wide variety of chemical and petrochemical processes (MERCANGOZ; DOYLE, 2008).

The main steps of a classical RTO framework (Figure 1.1) are steady state detection, parameter estimation, and optimization (SEQUEIRA; GRAELLS; PUIGIANER, 2002; NAYSMITH; DOUGLAS, 2008). In the first step, online plant data are analyzed (including data reconciliation and gross error detection) to determine, by some criterion, if the process is (reasonably) steady. Then, the steady state data is used to update some (relevant) parameters of the model such that it can represent the actual plant conditions as closely as possible. In the third step, the updated (adjusted) model is used to optimize the economic objective function, obtaining a new optimal operating point, which becomes the new target for the control system.

Figure 1-1 – Classic RTO



source: NAYSMITH AND DOUGLAS(2008)

The success of an RTO strategy largely depends on the mathematical steady state model of the process and its solution. The steady state model must be flexible to adjust the real behavior within a wide interval of process conditions (CHACHUAT et al., 2009) and accurate enough to guarantee that the calculated optimum is close to the real one (KRISHNAN; BARTON; PERKINS, 1992; FORBES; MARLIN; MACGRECOR, 1994). The solution strategy must be able to handle large plant disturbances in a fast and robust way to guarantee a new optimal point at every RTO cycle. These requirements make mechanistic models and equation-oriented (EO) simulation environments especially suitable for RTO applications (MEIXELL; GOCHENOUR; CHEN, 2000; YIP; MARLIN, 2004).

The present work focuses on generating a steady state model suitable to the RTO, and a dynamic model for an industrial-scale C3 splitter column (propane-propylene splitter) owned by Petrobras. The studied C3 splitter column in this work is a vapor recompression distillation (VRD) column which is a system for energy integration systems in process and energy industries. Energy integration is largely used in industry due to the high cost of energy and the corresponding need to minimize utility usage by pairing energy generation and consumption within the same plant. In vapor recompression distillation this energy integration is performed by heating the reboiler with the compressed stream coming from top of the column (NULL HL, 1976).

Although the mathematical modeling of the equipment involved in a VRD process is well-known, the tight material and energy integration (complex dynamics) in this system due to Reboiler-Condenser combination and large reflux flow, added to process nonlinearities, and the large number of equations used to describe this superfractionator (around 8500) makes this simulation particularly difficult to converge, especially in sequential modular (SM) simulators (AYDIN; BENALI, 2009; HEYEN; LENDENT; KALITVENTZEFF, 1994) and partially explains why the simplified and pseudo stream approaches have been previously proposed to simulate and optimize this process (CHOUDHARI; GUNE; DIVEY, 2012; KONINCKX 1988). These simplifications, together with neglecting the presence of minor components (<1 mol %) in the feed stream, might

affect the outcome of the optimization. The implementation of a rigorous model of the VRD process in an equation-oriented (EO) simulation environment mitigates the convergence difficulties generated by the recycle streams (MEIXELL et al., 2010), which avoids the previously mentioned simplifications and allows a better representation of the process behavior.

In addition to steady state modeling, the present work also considers the development of a dynamic model, which can be applied either as a virtual plant or for Dynamic Real Time Optimization (DRTO).

First proposed methods for solving separation systems models involving tray by tray calculation began in the 30's decade. From the 50's decade, with the advent of digital computers, new algorithms and simulators were developed. In the 70's decade, first commercial simulators were applied in industrial plants and development of rigorous models was considered.

Gani (1986) presented a generic and rigorous dynamic model considering thermodynamic equilibrium between the phases. Hydrodynamic behavior was included with the possibility of the increase of internal liquid and vapor phases and important events such as flooding and tray drying. For proving the efficiency and generality of the model, various industrial simulations were carried out, validating the model with plant data.

Next, Cameron (1986) discussed the numerical aspects of the previous work. As in that time solvers for Differential- Algebraic system of equations (DAE) were not available, dynamic simulation of distillation columns required system separation. The set of model equations was divided into a subsystem of ordinary differential equations (ODE) and a subsystem of algebraic ones. In this manner, the complete solution was obtained by sequential solution of two blocks in each integration step.

Considering the two latest works, Ruzi (1988) proposed developments of start-up policies for distillation column based on rigorous dynamic simulations. The results of their work led to characterizing the start-up condition of a plant.

Gani (1989) modified the dynamic model in order to use it as a new method for solving steady state simulations. The objective of this work was to obtaining robustness in solving steady state simulations. The exchange between steady state and dynamic models depended on the equations residual. If, with the initial estimates, the set of algebraic equations could not be solved, the problem switched to the dynamic mode. In the dynamic mode, the states were estimated by integration with the initial estimates corresponding to the initial condition. As soon as some integration criteria were achieved, the problem switched to the steady state mode which was solved without major difficulties.

The works developed by Gani and his co-workers were able to properly predict the vapor and liquid phases hydraulic behaviors inside the column. However, more details could be considered by adding the thermodynamic non-equilibrium condition between the phases. Considering non-equilibrium condition, the mass and heat transfer resistance between the phases could be considered, leading to more complex system of equations. In the work of Biardi and Grotolli (1989), transient diffusion of components was modeled by Maxwell-Stephan equations. The importance of this study is in the comparison between the ideal models (with thermodynamic equilibrium) and real ones (where the equilibrium condition is considered only in liquid-vapor interphase). Comparison was carried out with experimental data of laboratory scale and industrial columns. These models were able to provide accurate results, but they need an additional set of parameters related to the mass and heat transfer between the phases. Obtaining this set of parameters is difficult and includes a high level of uncertainty. Additionally, computational cost is increased as the complexity of the generated system of equations increases. Developments of more powerful computers and more robust integrators have led to faster solution of dynamic models. For instance, in the work of Olsen et al. (1997) three distillation columns of a methane purification unit were modeled

based on models developed by Gani et al. (1986). The model consisted 266 variables and 97 parameters and was simulated two times faster than real process.

In this work the EMSO environment (Soares et al., 2003) is used, which is able to solve Algebraic and DAE systems. However, rigorous dynamic tray-by-tray modeling of distillation column with the fixed pressure profile leads to high index system of differential and algebraic equations that demands high-order derivatives of thermodynamic properties. As thermodynamic properties calculation are not able to evaluate such derivations, for solving such a system an index one formulation should be proposed. One suggested formulation is adding algebraic equations to the system of equations that determine the vapor and liquid flow rates in the outlet of the trays without fixed pressure profile. Usually for distillation column, this is done by correlating vapor flow to the pressure drop and physical properties of the fluid (Wang et al. 2003), and liquid flow to the liquid level of each tray (Wang et al. 2003). The drawback related to these equations is the uncertainty in the parameters values that involve geometrical characteristics and coefficients, affecting the model response.

Considering that compositions dynamic is much slower compared to flow dynamics and tight pressure control in the case study, the present work proposes a formulation similar to the proportional controller in which for calculating vapor and liquid flow rates the pressure profile and the liquid level are kept approximately constant. The proposed correlations contain the initial condition and only one parameter for each correlation. These parameters are set in order to keep pressure drop and liquid level approximately constant on the trays.

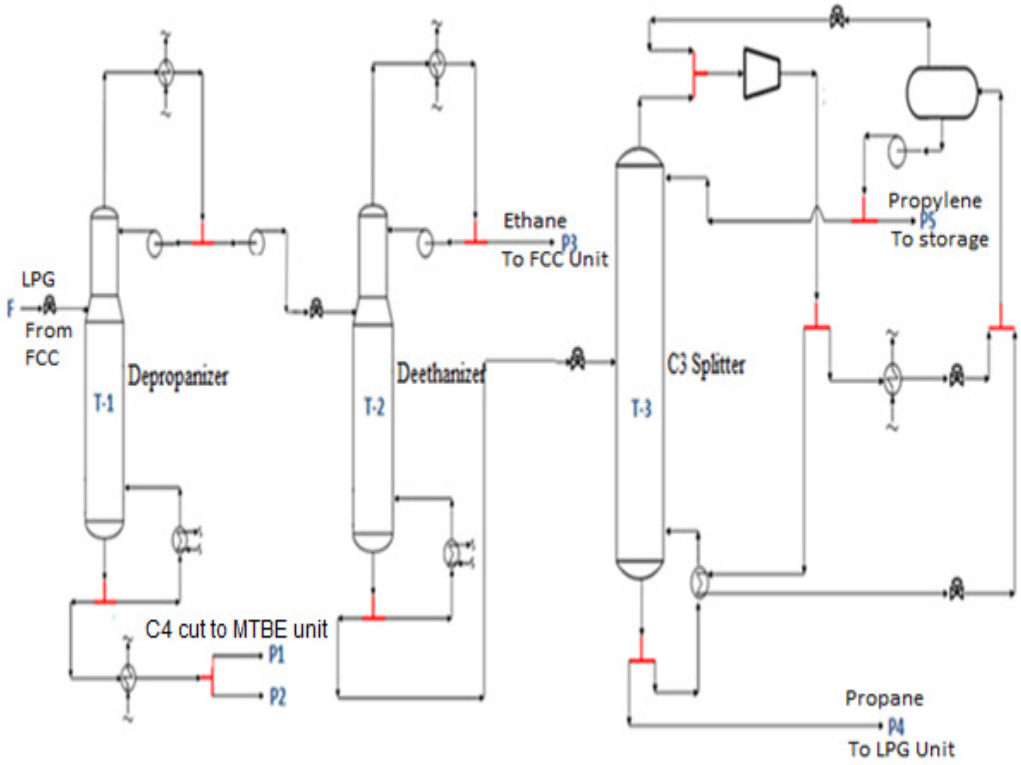
This dissertation is structured as follow: in chapter 2, process is presented and its details are explained. In chapter 3, thermodynamic model, steady state detection, data reconciliation and parameter estimation methods are studied. Also, analysis of steady state model and its validation are presented. Chapter 4 discusses different approaches for dynamic modeling and explains theirs mathematical difficulties. Then, the dynamic model is presented and is analyzed in design condition. Additionally, dynamic model

performance is compared to the real plant data. Finally, general conclusions and future works are given in section 6.

2. PROCESS DESCRIPTION

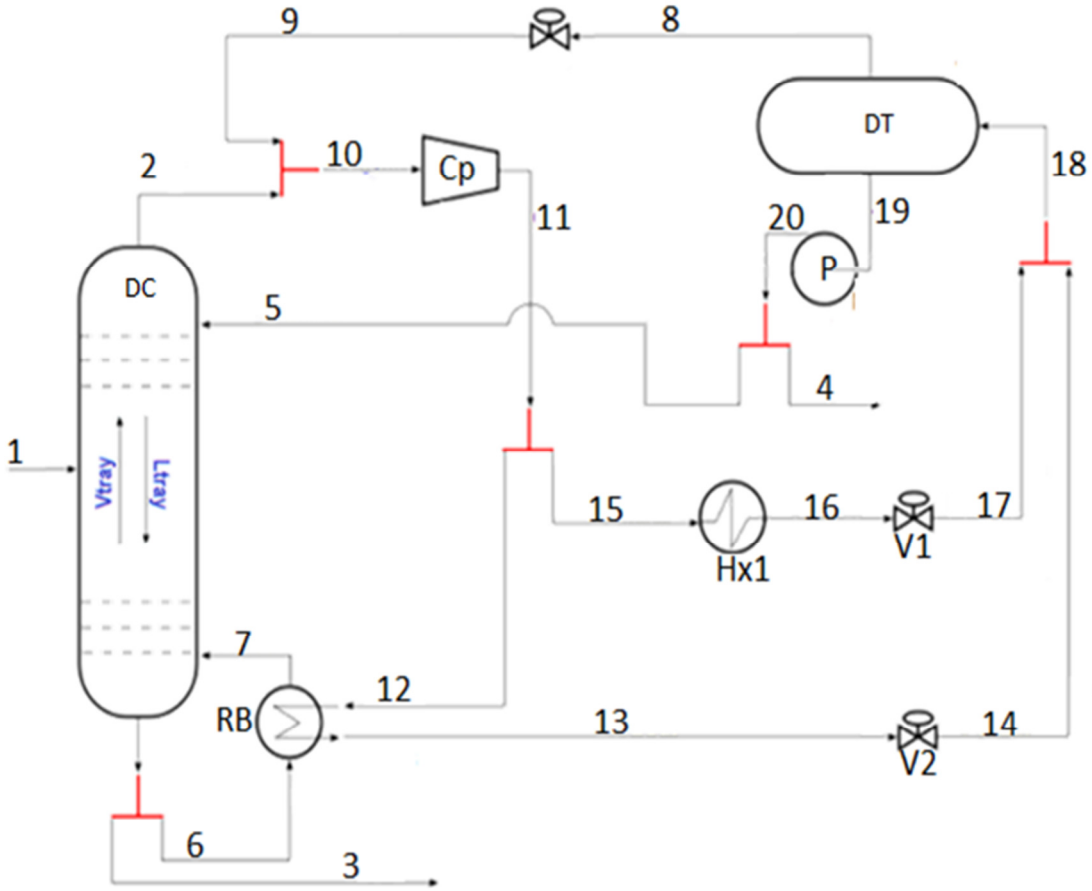
The REPLAN's Propylene Production Unit was designed to produce 265,000 t/y of propylene polymer grade with high purity (99.5% molar at minimum). The Propylene Production Unit at REPLAN consists of three different distillation towers (Depropanizer, Deethanizer and C3 Splitter). The Unit is fed by the LPG (Liquefied Petroleum Gas) from the FCC Units (Fluid Catalytic Cracking) and produces two product streams: a C4 Cut that is sent to MTBE (Methyl Tertiary Butyl Ether) Unit and Propylene (Figure 2.1).

Figure 2-1- Unit flow sheet



The RTO implementation is desired for the vapor recompression distillation column, C3 splitter (Figure 2.2).

Figure 2-2 - C3 Splitter flow sheet



Vapor recompression assisted distillation (VRD) is a heat integration technique, widely used in the chemical industry to economically separate close-boiling mixtures, thus its most practical applications occur in binary systems where the separation is difficult (low relative volatility) (HARWARDT; MARQUARDT, 2012; ANNAKOU; MIZSEY, 1995).

The smaller the temperature difference between the top and bottom is, the more profitable the heat pump application will be, for the following reasons: small temperature differences indicate difficult separations with high reflux ratios, and consequently, high steam and cooling water consumptions, and the pressure ratio and the compression power which are required increase with the increase of temperature difference.

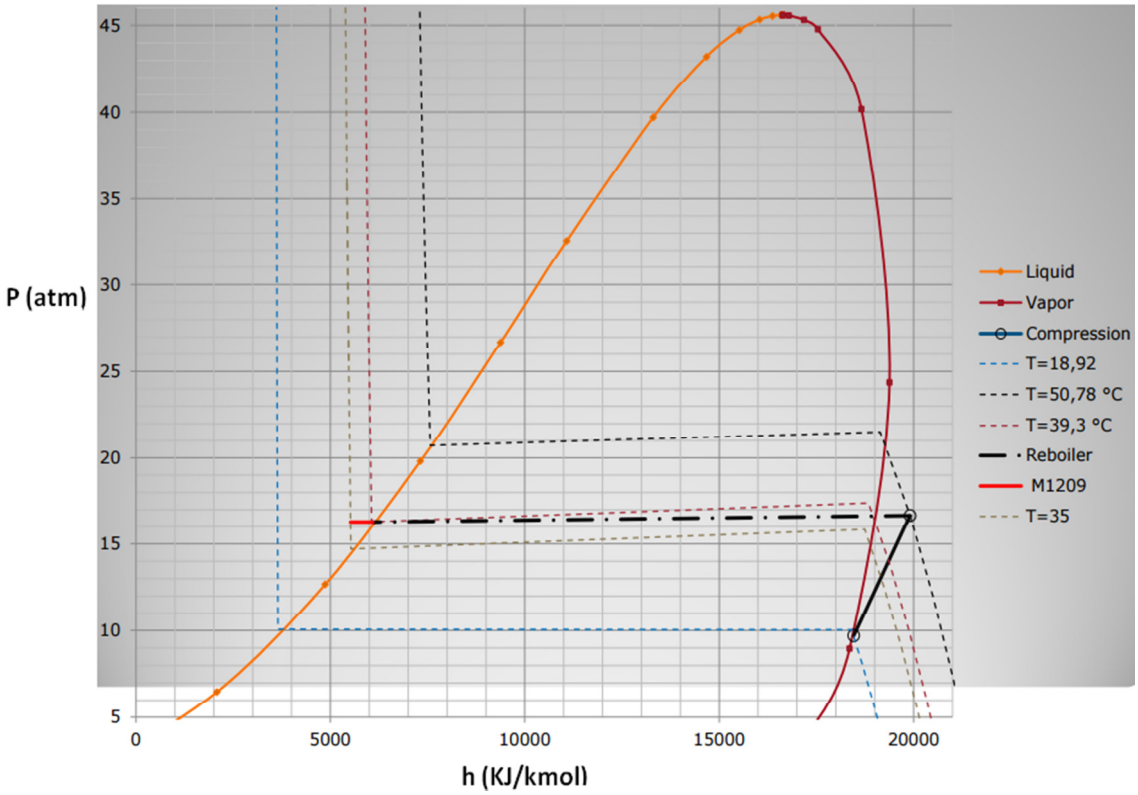
The flow sheet shown in Figure 2.2 illustrates the main features of the vapor recompression distillation (VRD) process. A low molecular weight hydrocarbon mixture

expands through valve and enters the distillation column (DC), where high-purity propylene is obtained as the overhead product, stream 4, and propane is obtained at the bottom, stream 3. The overhead product, mixed with the vapor coming from distribution tank (DT), stream 9, and is compressed to increment its condensing temperature in compressor (CP). One part of the outlet flow from (CP) is sent to condenser (HX1), stream 15, and is used to control column pressure. The rest condenses in reboiler (RB), stream 12. Outlet streams from HX1 and RB, expands through valves V1, V2 respectively, returning to DT where a portion of the liquid is sent to DC as reflux, stream 5, and the other part is sent to storage as main product, stream 4.

Actually, for using conventional distillation column, the operating pressure should be increased so that cooling water can be utilized for condensing the top vapor. However, relative volatility decreases with pressure and separation gets more difficult. By applying Vapor Recompression Distillation (VRD) technology, the column pressure can be kept lower, but the top vapor is compressed in the compressor up to the necessary pressure in such a way that its condensing temperature is greater than the boiling temperature of the column bottom product.

The compressing process through the compressor and its effect on thermodynamic properties of the top vapor (pure propylene) is shown in Figure 2.3. First the top vapor (saturated vapor, blue line) is compressed up to 16.2 atm reaching the temperature T = 50.78ºC in superheat condition. Then, Outlet of the compressor is condensed in the reboiler-condenser (RB). As the condenser side of the reboiler-condenser (RB) is a submerged flooding condenser, the heat load to the other side is adjusted by changing the liquid level, and consequently the surface area. Finally, accumulated liquid inside the condenser side is subcooled (red line).

Figure 2-3 - Top vapor temperature changes through compression and condensing



3. STEADY STATE MODELI

The steady state model for this process was developed in an Equation Oriented (EO) environment using software EMSO (Environment for Modeling Simulation and Optimization).

In the Sequential Modular approach, unit operations are presented by modules (procedural representation) with defined input and output streams. The output streams are calculated from input streams within each module. When solving a steady state flowsheet, the solution from one module is used as input to the following module in the sequence.

Equation Oriented (EO) modeling is an alternative strategy for solving flowsheet simulations. Instead of solving each block in sequence, EO gathers all the model equations together and solves them at the same time. For the steady state case this is a set of nonlinear algebraic equations (NL), represented by a sparse occurrence matrix with unsymmetric blocks along the diagonal that can be solved by Newton method or other approaches.

The EO strategy can be very effective in the situations where SM struggles, such as:

- highly heat-integrated processes
- highly recycled processes
- processes with many specifications
- process optimization

Normally, these types of problems are very difficult to solve with SM strategy, because they require many successive solutions to the flowsheet and may contain many nested convergence loops.

Although the number of variables and equations can be very large, EO solves the flowsheet simultaneously without nested convergence loops and usually utilizes

analytical first order derivations. As a result, EO strategy can solve much larger problems using the same computational effort.

The main details of the developed model in this work are given in Appendix A and the model code in EMSO language is available in Appendix D. Table 3.1 represents numerical characteristics of the developed steady state model.

Table 3-1 - Model details

Information	Value
Number of Variables	8480
Number of Equations	8463
Number of Specifications	17
Degrees of Freedom	0

The performance of the RTO depends on the plant model accuracy. It should be flexible enough to cover the wide interval of process conditions and accurate enough to guarantee the precision of the optimal point.

Beside this, in this work for initializing the dynamic model, steady state simulation results were applied. More explanations about the dynamic model and necessity of consistent initial conditions can be seen in section 4.3.

In the following sections of this chapter, first thermodynamic considerations are studied. In section 3.2, the applied method for detecting the steady state points form plant data is presented. Data reconciliation and parameter estimation routines are discussed in section 3.3. Sections 3.4 and 3.5 consist analysis of steady state model and its validation respectively.

3.1. THERMODYNAMIC EQUILIBRIUM

Thermodynamic equilibrium is provided when thermal, mechanical and chemical conditions are satisfied:

$$T^L = T^V \tag{3-1}$$

$$P^L = P^V \tag{3-2}$$

$$\hat{f}_i^{\,L} = \hat{f}_i^{\,V} \quad (i = 1,2,3...c) \tag{3-3}$$

T is temperature, P is pressure , $\hat{f}_i^{\,L}$ is the fugacity of the *ith* component in liquid phase , $\hat{f}_i^{\,V}$ is the fugacity of the components in vapor phase, L and V are liquid and vapor molar flow rates respectively.

The fugacity equality is often represented using fugacity coefficient. Fugacity coefficient of component *i* in the gas solution is defined as:

$$\widehat{\Phi}_i^{\,V} \equiv \frac{\hat{f}_i^{\,V}}{P y_i} \quad (i = 1,2,3...c) \tag{3-4}$$

And for liquid solution:

$$\widehat{\Phi}_i^{\,L} \equiv \frac{\hat{f}_i^{\,L}}{P x_i} \quad (i = 1,2,3...c) \tag{3-5}$$

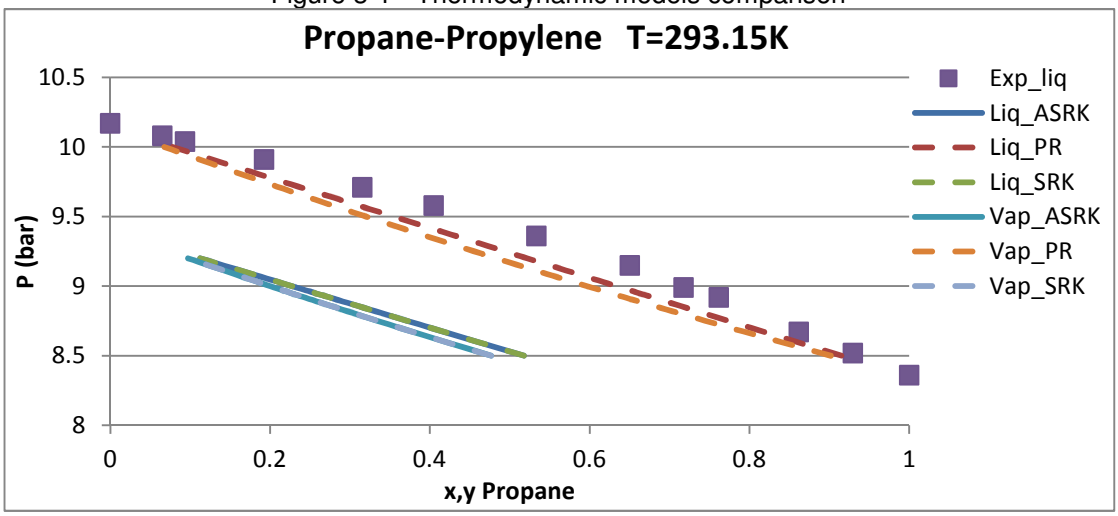
This allows to write the fugacity equality as follows:

$$\widehat{\Phi}_i^{\,L} x_i = \widehat{\Phi}_i^{\,V} y_i \quad (i = 1,2,3...c) \tag{3-6}$$

Propylene-propane mixture shows non-ideal behavior especially in vapor phase due to high operating pressure (MUHLER et al. 1990). Therefore, the dynamic and steady state modeling of the unit was implemented in EMSO (Environment for Modeling Simulation and Optimization) using PR (Peng-Robinson) equation of state of the VRTHERM thermodynamic package.

Thermodynamic model was selected based on analysis considering different available options. Figure3.1 shows analysis for mixture of propane and propylene at T=293.15K at different pressures in which experimental data was obtained from Ho et al. 2006. As shown, PR equation of state calculates vapor and liquid compositions better comparing to SRK and ASRK.

Figure 3-1 - Thermodynamic models comparison



Besides this, as mass transfer limitations prevent the vapor leaving a tray from being in precise equilibrium within the liquid on the tray; consequently, the assumption of ideal stages is only an approximation. For representing the real situation, Murphree efficiency can be adjusted. Murphree efficiency is the ratio of actual change in the average vapor composition to the change that would occur, if the vapor leaving the tray was in with the liquid leaving the tray.

The Murphree vapor phase tray efficiency is the most commonly used efficiency definition, and is the only one with practical importance (eq.(3.7)).

$$E_i^{MV} = \frac{y_i^{out} - y_i^{in}}{y_i^{*} - y_i^{in}} \tag{3-7}$$

Thus, the worse the contact between the phases of the system, the worse the mass transfer between them and more significant will be the non-idealities inherent in the process, leading to lower values of Murphree efficiency.

3.2. STEADY STATE DETECTION

Steady state detection is an important subject in Real Time Optimization, identification of steady state models, analysis of processes, dada reconciliation and other applications. These applications need data in steady state or close to that, and for this, one efficient technique is necessary.

In this work, steady state detection was applied in order to identify the steady state experimental points. These points were used for evaluating the steady state model and initializing the dynamic model.

There are several approaches for steady state detection. NARASIMHAN et al. (1986) presented the *Composite Statistical Test* - CST and the *Mathematical Test of Evidence* - MTE. In CST, successive time periods are defined and evaluated according to covariance matrix and sample mean, and in MTE, differences in mean value are compared with the variability between periods. CAO and RHINEHART(1995) created a method based on moving average or first order filter.

In this work the approach created by JIANG et al. (2003) was applied which is based on wavelet transform and determines one steady state index with values between 0 and 1. In the period of times in which the index value is equal to one, the process is at steady state. Moreover, for multivariable processes one global steady state index that is stand on same degree of importance for each variable is considered. Appendix B presents

more clarifications on the methodology. Obtained results for some variables of the unit can be seen in Fig (3.2-3.7) in which the first diagram shows steady state index value ($B_i(t)$) for the variable and the second diagram shows its actual value. Fig (3.8) presents the results for steady state multivariable index $B_m(t)$.

Figure 3-2 **-** Steady state detection for flow rate of stream 1 (t/h)

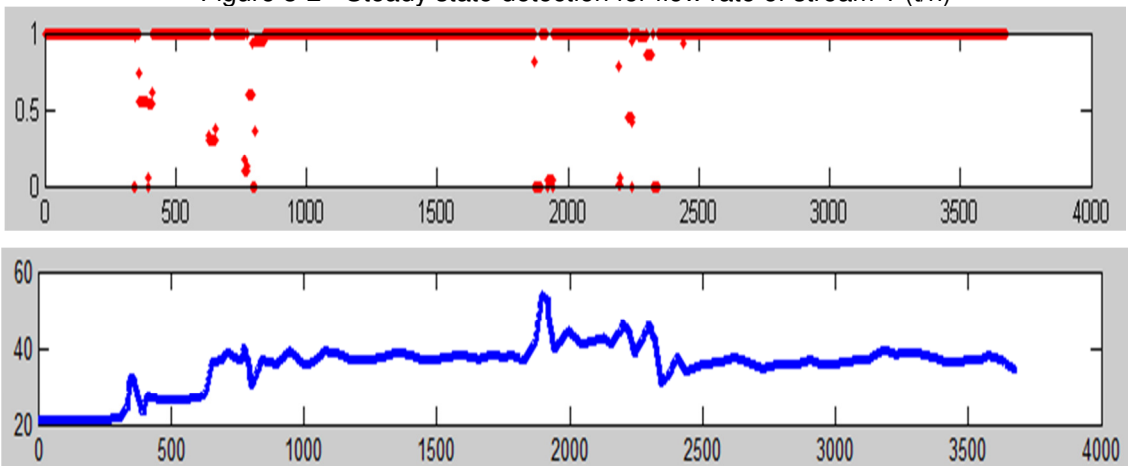


Figure 3-3 **-** Steady state detection for temperature of stream 1 (c)

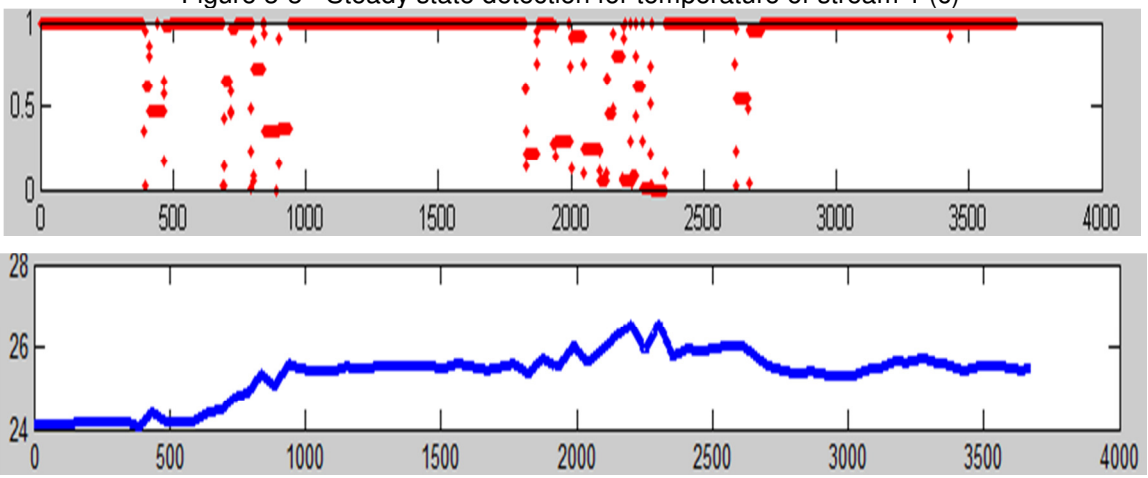


Figure 3-4 -Steady state detection for flow rate of stream 4 (t/h)

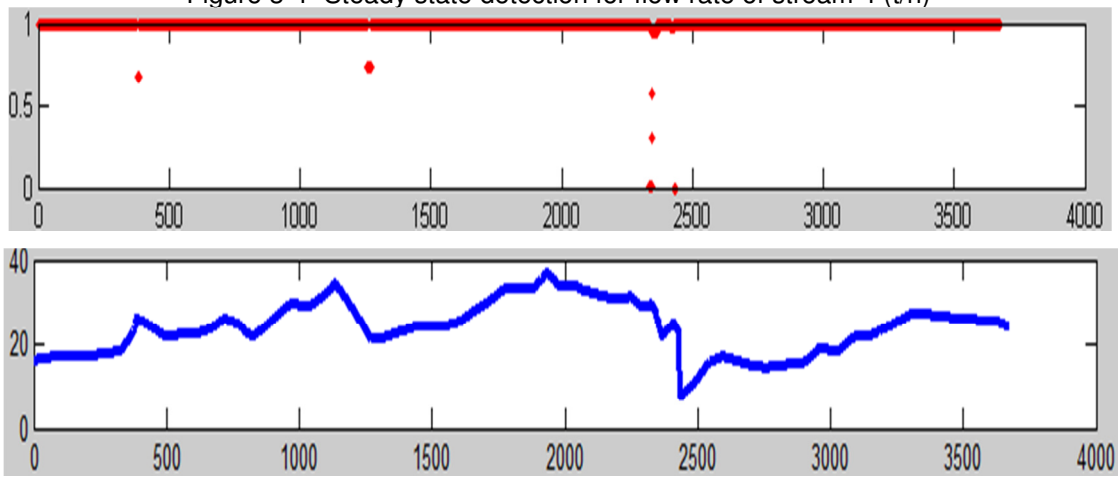


Figure 3-5 **-** Steady state detection for temperature of stream 4 (c)

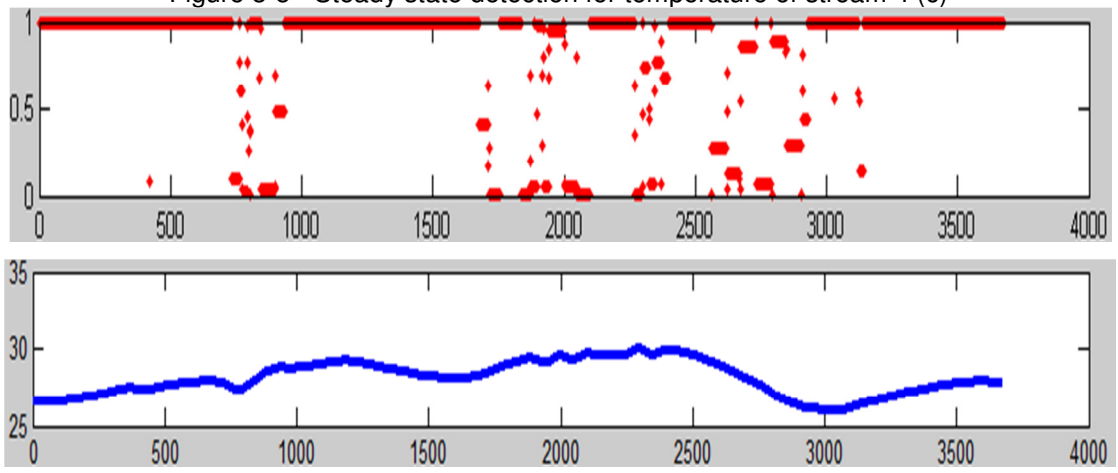


Figure 3-6 - Steady state detection for flow rate of stream 3 (t/h)

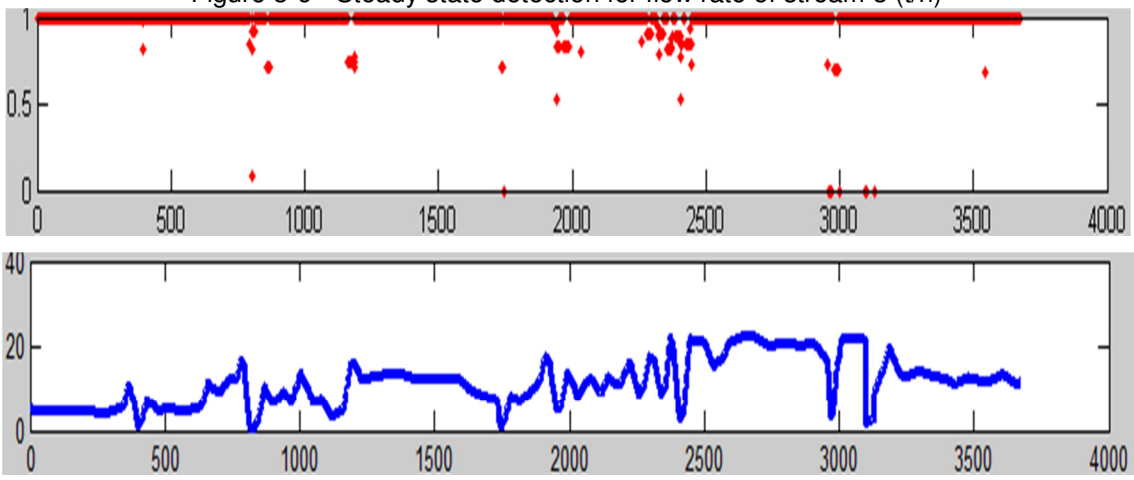


Figure 3-7 -Steady state detection for temperature of stream 3 (c)

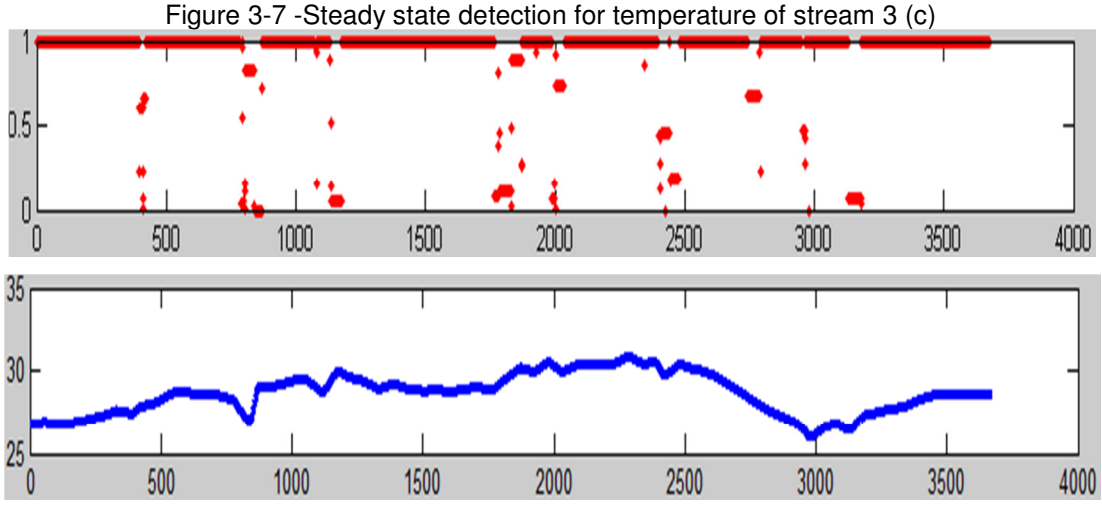
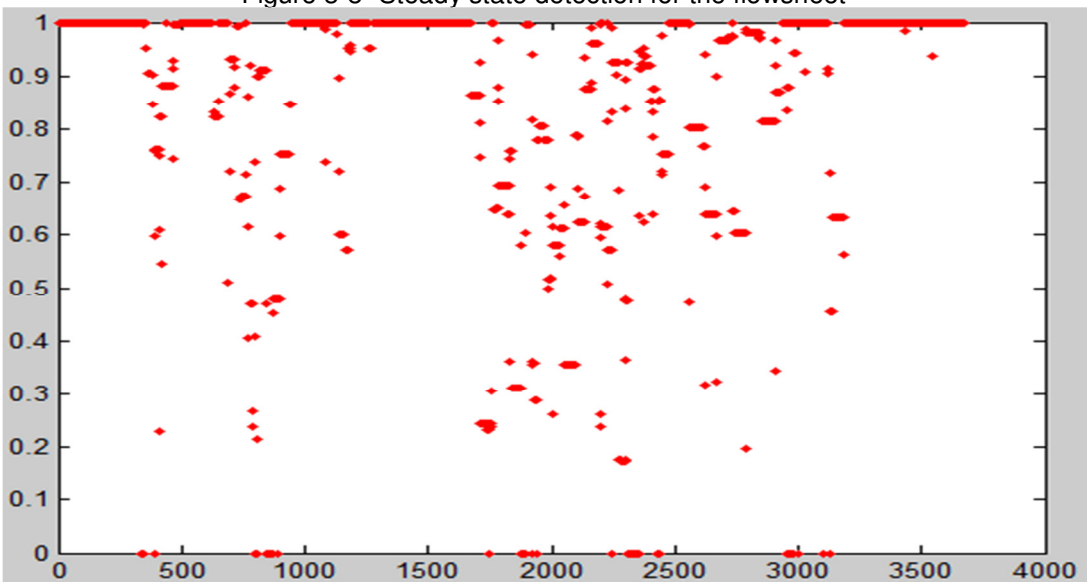


Figure 3-8 -Steady state detection for the flowsheet



3.3. DATA RECONCILLATION AND PARAMETER ESTIMATION

In order to experimental measurements represent the precise information and reliably describe the process, reconciliation is carried out. Reconciliation is the procedure of fitting the measurements with respect to conservation laws and physical constraints of the system that results in reducing the effect of random errors in the data.

KUEHAN (1961) for the first time presented the data reconciliation in steady state condition. He considered the reconciliation as an optimization problem subject to constraints (mass and energy balances), aiming to minimize the weighted least square objective function:

$$F_{obj} = \sum_{i=1}^{NE} \sum_{j=1}^{NY} \frac{\left(Y_{ij}^{re} - Y_{ij}^{m}\right)^2}{\sigma_{ij}^2} \quad (3\text{-}8)$$

In the above equation, *NE* is the number of measurements, *NY* is the number of variables, Y_{ij}^{re} is the reconciled variable, Y_{ij}^{m} is the measured variable and σ^2 is the variance of the measured variables.

The weighted least square objective function leads to precise statistics, considering that errors have normal distribution and are not correlated. Beside this, this chosen objective function deals with various set of data, provided that measurement errors are know or that they can be at least estimated. The variance of the measured variables is the normalization factor of variables.

In this work, variables of the objective function (eq. 3.8) are as follows:

- Stream 3 (Bottom product flow rate)
- Stream 5 (Reflux flow rate)
- Stream 12 (Reboiler inlet flow rate)
- Stream 1 (Feed flow rate)
- Propane concentration in stream 4 (Distillate product)
- Propylene concentration in stream 3 (Bottom product)

Besides this, parameters influence on the process response will be analyzed in section 3.4. Estimating the proper value for these parameters is pursued in order to update the model in real time applications.

A simplex optimization routine was applied in both cases for solving data reconciliation and parameter estimation problems in EMSO environment.

3.4. MODEL ANALYSIS

The analysis of the model allows a better understanding of the process and the establishment of boundaries to process specifications and the parameters updated in the RTO cycle. All this information is necessary for the RTO scheme not only because parameter estimation and optimization are model dependent tasks, but also because convergence must be ensured when process conditions and model parameters change.

The results shown below correspond to simulations in EMSO with the base specifications presented in Table 3.2.

Table 3-2 **-** Process specifications for the project case

Stream 1	Flow Rate(t/h)	45.7343	
	Molar Composition	ethane	1.21e-4
		Isobutene	3.26e-4
		Trans-2Butene	9.3e-6
		N-Butane	9.3e-6
		Propylene	0.731
		1-Butene	1.3e-4
		1,3Butadiene	9.3e-6
		Propane	0.267
		Cis-2-Butene	9.3e-6
		Isobutane	6.1e-4
	Pressure (atm)	11.27	
	Temperature (K)	345.35	
Stream 3	Flow Rate(t/h)	13.2796	
Stream 5	Flow Rate(t/h)	392.204	
Stream 16	Temperature (K)	308.15	
stream 13	Temperature (K)	308.15	

3.4.1. FEED COMPOSITION

The influence of minor components on the performance of the process is evaluated choosing light (ethane) and heavy (isobutane) components as representatives of this group. The mole fractions of ethane and isobutane in the feed stream were varied in 1 order of magnitude regarding the base case, 1.21×10^{-4}_1×10^{-3} (ethane) and 6.1×10^{-4} _6.1×10^{-3} (isobutane); this interval was chosen based on the expected feed composition variations in the VRD process at REPLAN.

The temperature profile of distillation column does not show significant change. In other words, the process is not sensitive to composition changes of ethane and isobutane (Figure 3.9, 3.10).

Figure 3-9 - Temperature profile for different compositions of Ethane

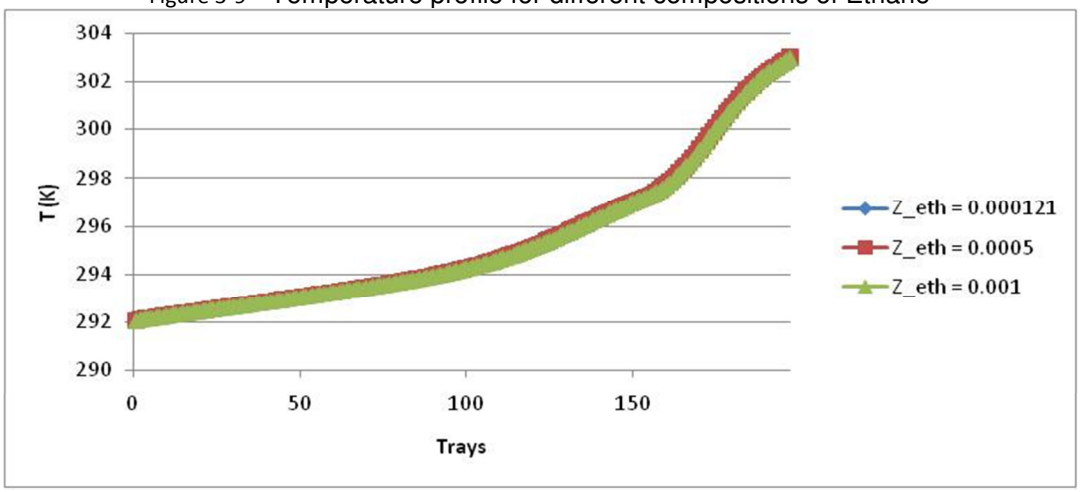
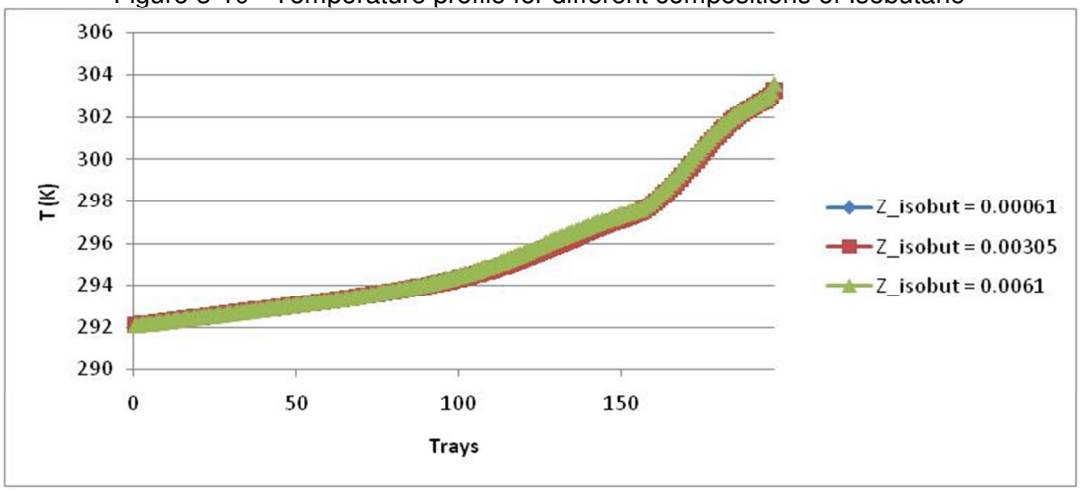


Figure 3-10 - Temperature profile for different compositions of Isobutane



3.4.2. VAPOR MURPHREE EFFICIENCY

Two vapor Murphree efficiencies, one for the rectification zone (E_r^{MV}) and one for the stripping zone (E_r^{MV}), were defined in order to take into account non-ideality of the mass transfer phenomena in the distillation column.

These efficiencies are varied between 0.1 and 1 in the rectification and stripping zones. The temperature profiles, obtained for different efficiency values are shown in Figure 3.11 and 3.12. The results indicate that the tray temperatures are more affected by efficiencies of stripping zone than rectification zone.

Figure 3-11 - Influence of vapor efficiency of rectification zone

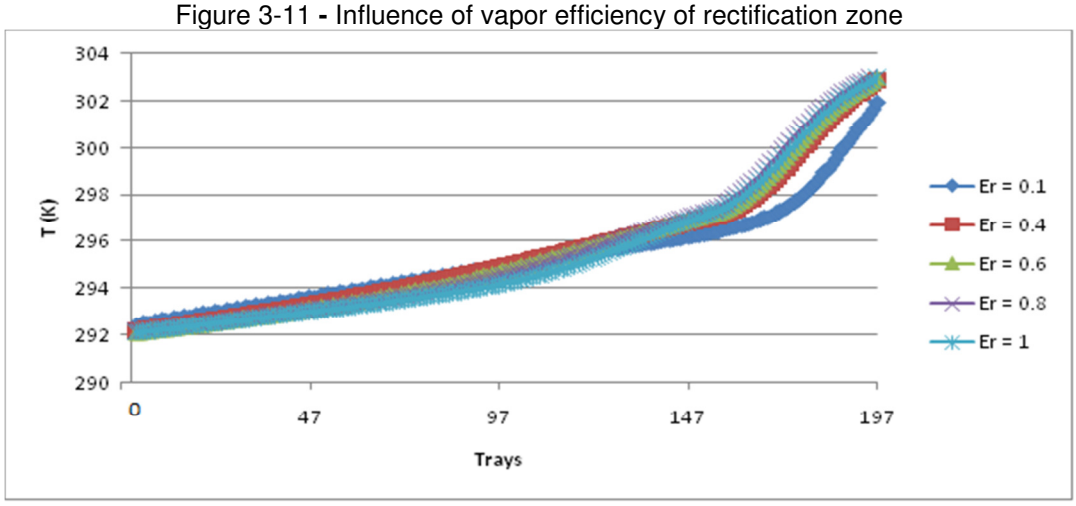
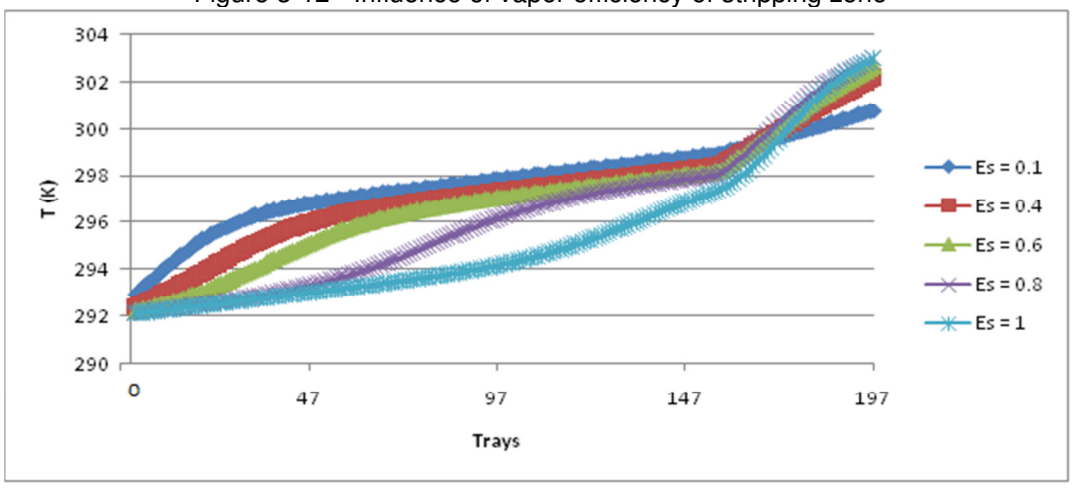


Figure 3-12 - Influence of vapor efficiency of stripping zone



The reduction in mass transfer performance in the rectification zone (low values of E_r^{MV}) causes an increase of the temperatures of the rectification zone and a decrease of the temperatures of the stripping zone. A similar behavior is obtained with respect to the efficiencies of the stripping zone, E_s^{MV}.

3.4.3. REFLUX RATIO

The influence of reflux ratio on product purity, as well as on energy input to the process was studied. The product purity achieved in the distillation increases proportionally to

the reflux ratio up to a RR around 12.8, little improvements in purity are achieved for refluxes larger than this value, and the minimum reflux ratio that meets product specifications (mol % of propylene ≥ 99.5 in stream 2) is about 12 (Figure 3.15).

The energy demand of the compressor increases linearly with the reflux ratio (Figure 3.13) and the contribution of the energy supplied to the process by the compressor decreases asymptotically as the reflux ratio increases; this energy represents between 9 and 12% of the heat exchanged in the reboiler (Figure 3.14).

Figure 3-13 -Influence of reflux ratio on work input to the compressor

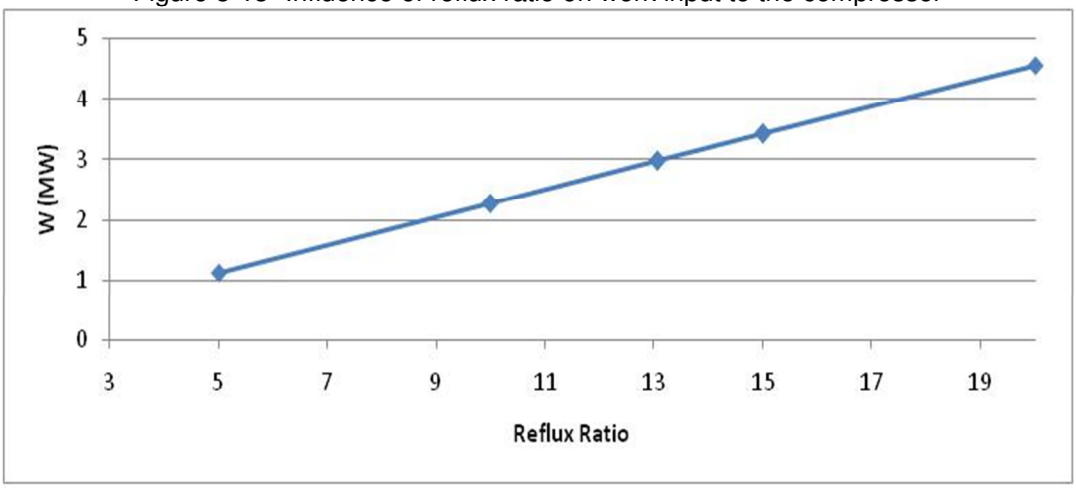


Figure 3-14 -Influence of reflux ratio on the ratio of work input to heat transfer in reboiler

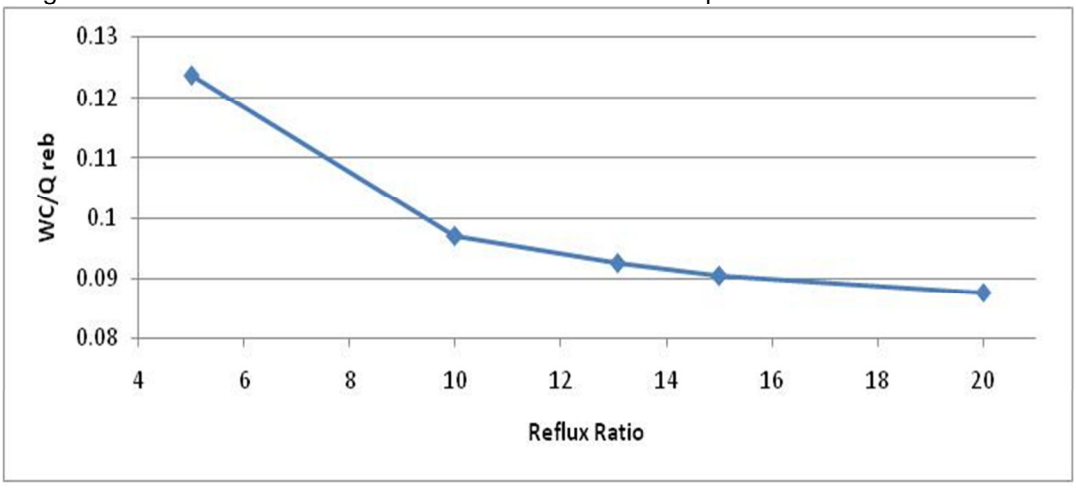
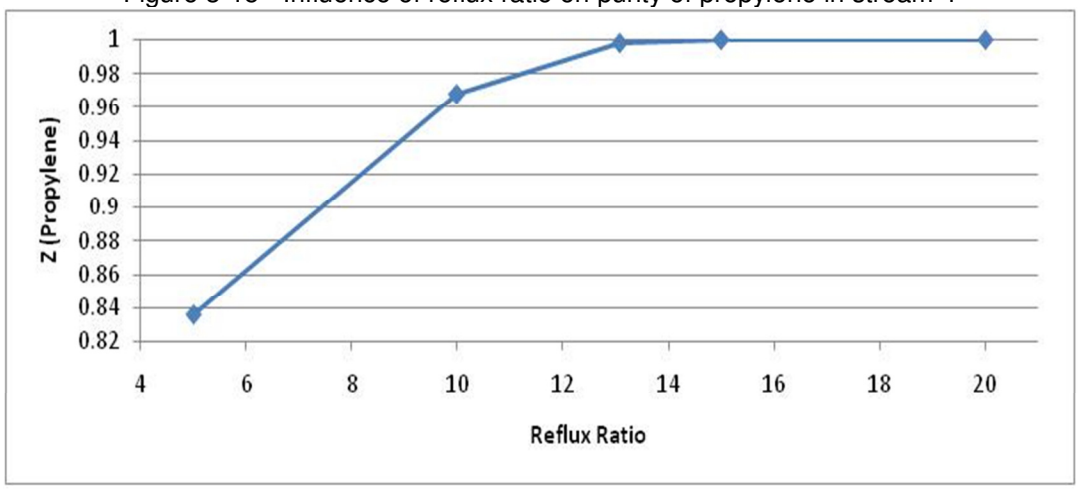


Figure 3-15 - Influence of reflux ratio on purity of propylene in stream 4



3.4.4 PRESSURE DROP IN DISTILLATION COLUM

The influence of column pressure drop was evaluated in the interval of 0–1.25 atm, assuming a linear profile, i.e., constant pressure drop per tray. The pressure drop in the distillation column considerably influences the temperature profile of the column (Figure 3.16), especially in the stripping zone. Temperature differences between the first tray and the last tray change from 7.17 K (ΔP = 0 atm) to 11.98 K (ΔP = 1.25 atm). The heat exchange in the reboiler slightly increases with the pressure drop of the column, 0.2% larger for ΔP = 1.25 atm than when ΔP = 0 atm, while the mole fractions of propylene in tray 1 and propane in tray 197 changed less than 0.1% and 0.5% respectively when compared to the simulation with neglected pressure drop

Figure 3-16 - Influence of pressure drop on distillation column temperature profile

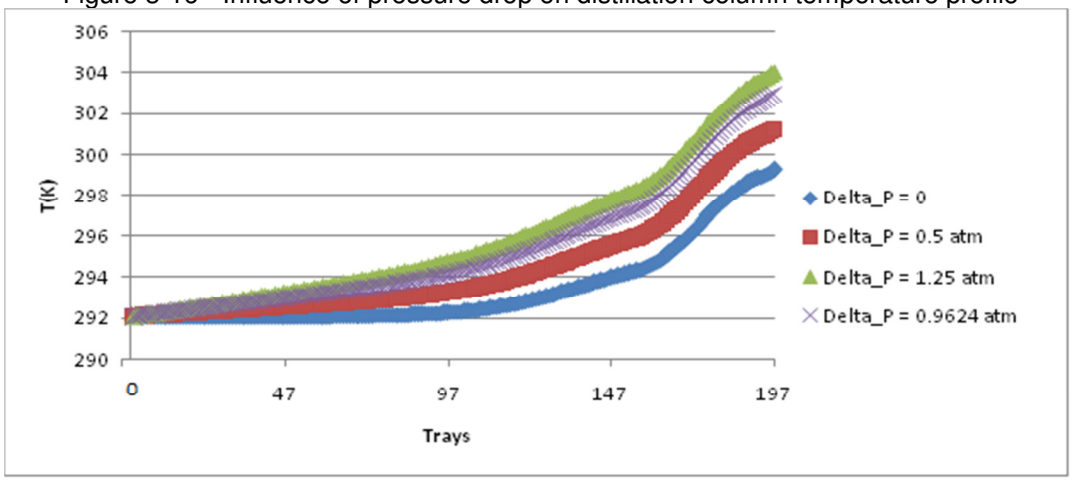
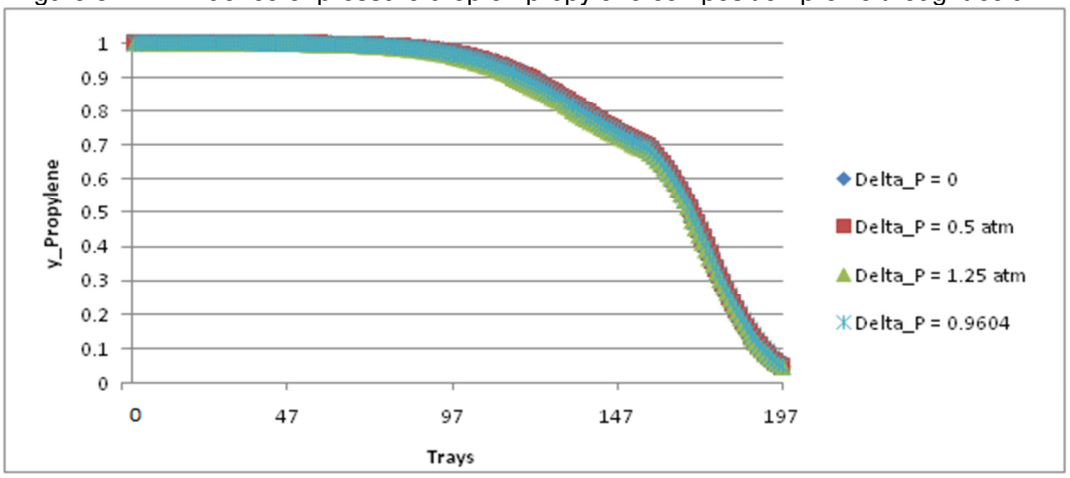


Figure 3-17 - Influence of pressure drop on propylene composition profile thought column



3.4.5 COMPRESSOR EFFICIENCY

The influence of the isentropic compressor efficiency, η_{cp}, was analyzed by changing it in the interval 0.4–1. The main changes in the process associated with this variable are in the recycled vapor, stream 8, as well as the outlet temperature and energy consumption in the compressor (Figure 3.18, 3.19). The outlet temperature of the compressor decreases as the efficiency increases; these changes are greater for efficiencies below 0.6. A similar tendency is shown by energy demand.

Figure 3-18 - Influence of compressor efficiency on work input of compressor

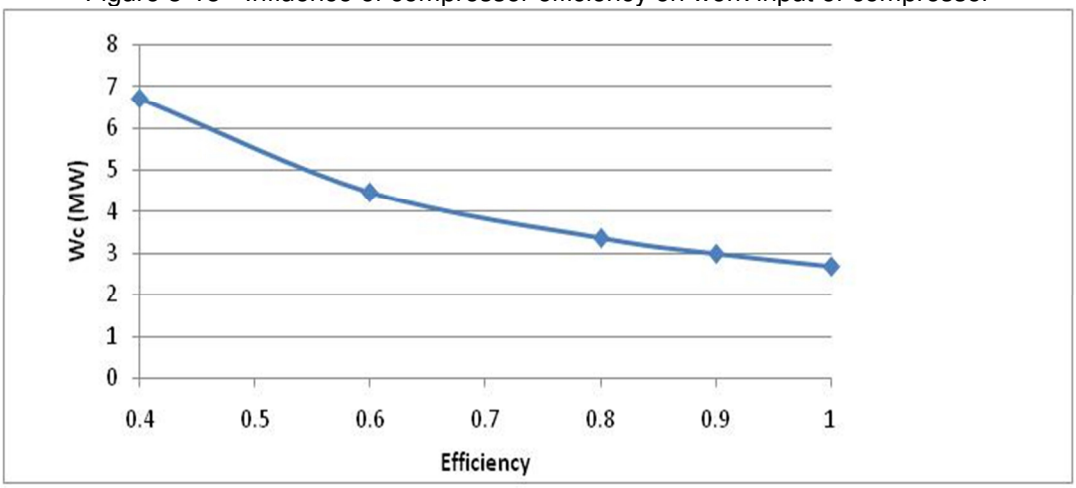
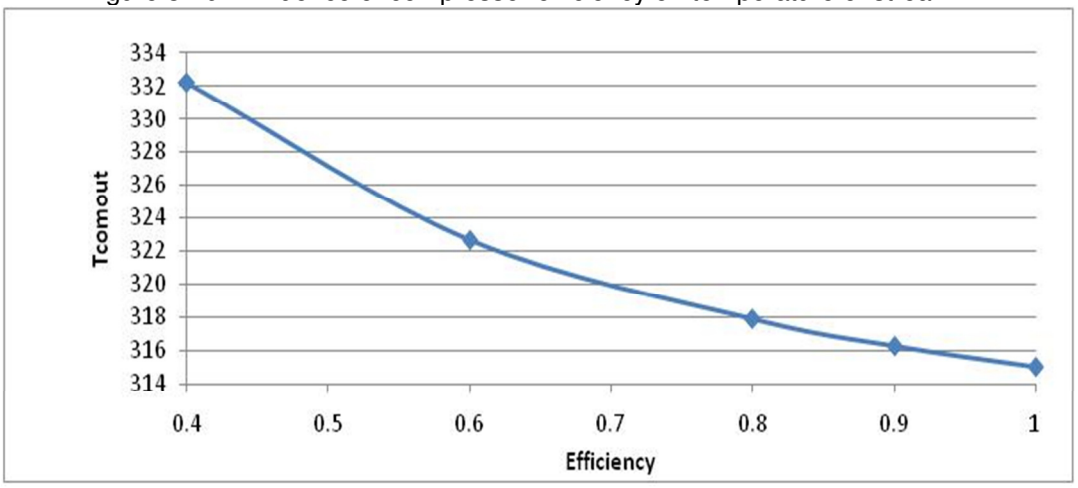


Figure 3-19 - Influence of compressor efficiency on temperature of stream 11



3.4.6 BOTTOM FLOW RATE

The bottom stream flow rate is changed in the interval between 250 and 350 kmol/h, where 299.92 kmol/h is the specification of the base case. Simulations for flow rates lower than 230 kmol/h did not converge, while it is possible to increase the bottom flow rate until it matches the feed flow rate, stream 1. The latter feature is important because the model is able to represent the real situation that takes place when the product is off specification and the distillated product valve is closed. The increase in the bottom flow reduces the heat load in the reboiler and vapor flow in the column as well as the energy consumption in the compressor (Figure 3.20-3.22). A flow rate about 289.413 kmol/h in the bottom stream is the operational point that meets process specifications (mol % propylene ≥ 99.5 in distillate product) with the highest recovery of propylene (91.08 mol %) and the minimum energy consumption in the compressor (2.97 MW).

Figure 3-20 - Influence of bottom flow rate on mole fraction of propylene in stream 4 and its recovery

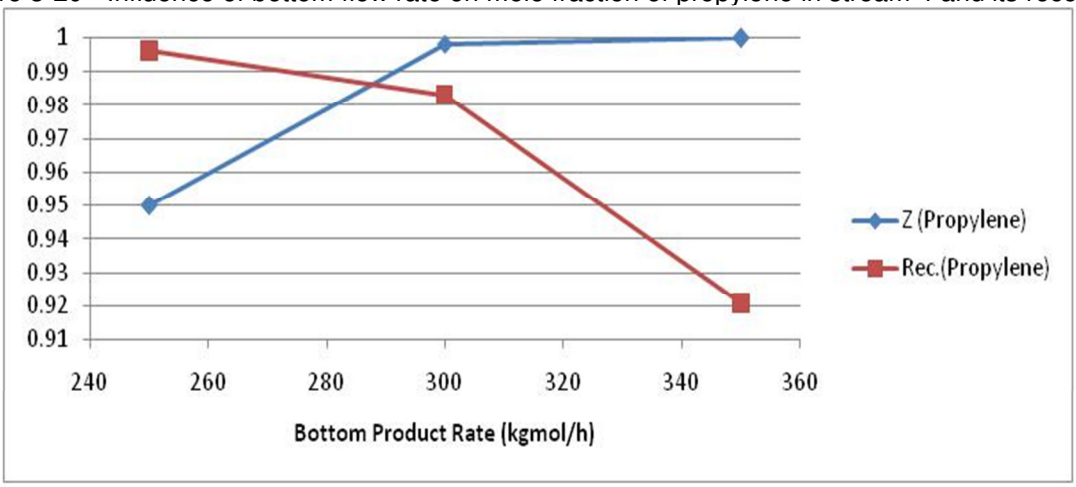


Figure 3-21 -Influence of bottom flow rate on flow rate of stream 2

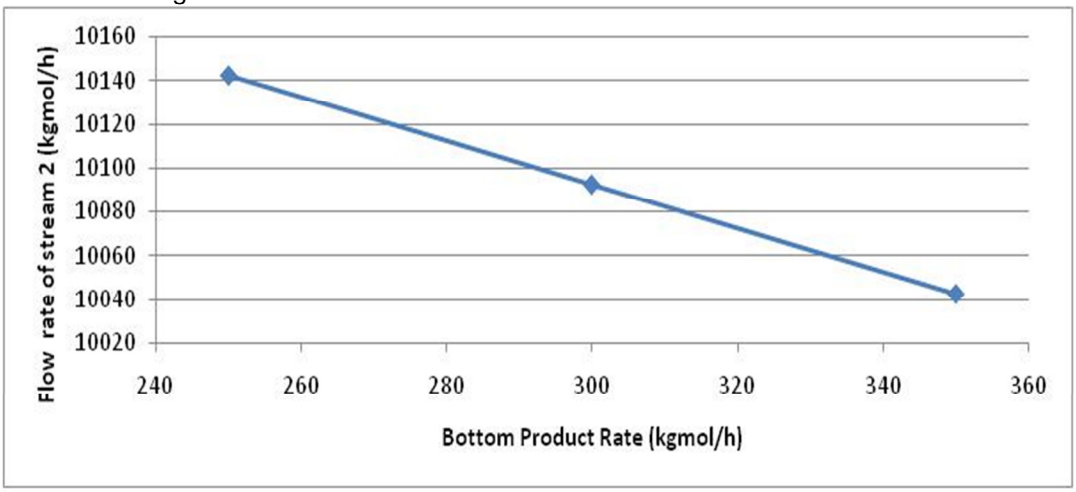
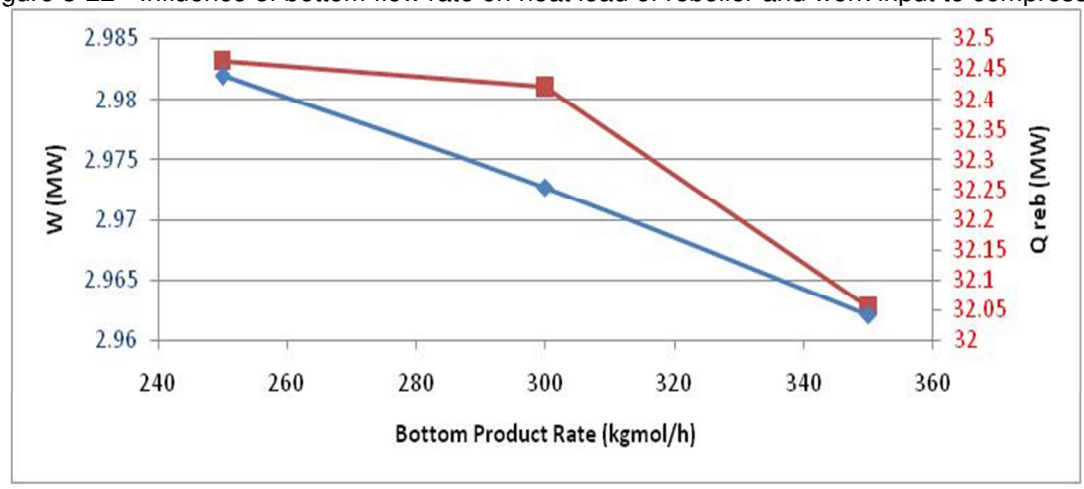


Figure 3-22 - Influence of bottom flow rate on heat load of reboiler and work input to compressor



3.5. VALIDATION OF STEADY STATE MODEL

For inferring the quality of the developed steady state model, validations were performed in two cases. In the first case, validation was carried out considering the process design condition.

Table 3.3 contains specification for the case 1 (project condition). Table 3.4 shows the simulated data and its relative and absolute error considering Murphree efficiencies of rectification zone, stripping zone and compressor efficiency equal to one.

In order to reproduce bottom and top product specifications and compressor discharge temperature, estimation routine presented in section 3.3 was applied to adjust vapor Murphree efficiency of stripping and rectification zones and compressor efficiency (Table 3.5)

Table 3-3- Specification for case_1 (design condition)

Stream 1	Flow Rate(t/h)	45.7343	
	Molar Composition	ethane	1.21e-4
		Isobutene	3.26e-4
		Trans-2Butene	9.3e-6
		N-Butane	9.3e-6
		Propylene	0.731
		1-Butene	1.3e-4
		1,3Butadiene	9.3e-6
		Propane	0.267
		Cis-2-Butene	9.3e-6
		Isobutane	6.1e-4
	Pressure (atm)	11.27	
	Temperature (K)	345.35	
Stream 3	Flow Rate(t/h)	13.2796	
Stream 5	Flow Rate(t/h)	392.204	
Stream 16	Temperature (K)	308.15	
stream 13	Temperature (K)	308.15	

Table 3-4 -Comparison for case_1(considering $\mathbf{E_T^{MV}}$=1, $\mathbf{E_S^{MV}}$ =1, η_{cp}= 1)

Case_1	Simulated data	Project condition	Relative Error_%	Absolute Error
Mole Fraction of Ethane + Propane in stream 4	1.95e-4	0.00401	95.137157	0.003815
Mole Fraction of Propylene in stream 3	0.04018	0.050013	19.660888	0.009833
Flow rate of stream 12 (t/h)	390.244	389.055	0.3056	1.189
Flow rate of stream 4 (t/h)	32.5183	36.0912	9.89	3.5729
Temperature of stream 12 (k)	314.992	317.550	0.8055	2.558

Table 3-5 -Comparison for case_1 (considering $\mathbf{E_T^{MV}}$=0.7329, $\mathbf{E_S^{MV}}$=0.7592, η_{cp}= 0.8)

Case_1	Simulated data	Project condition	Relative Error_%	Absolute Error
Mole Fraction of Ethane + Propane in stream 4	0.004141	0.00401	3.266832918	0.000131
Mole Fraction of Propylene in stream 3	0.05037	0.0500133	0.713210286	0.0003567
Flow rate of stream 12 (ton/h)	385.768	389.055	0.8448	3.287
Flow rate of stream 4 (ton/h)	32.5246	36.0912	9.88	3.566
Temperature of stream 12 (k)	317.863	317.550	0.098	0.313

In case 2 (Table 3.6), simulated data was compared to steady state plant data detected from steady state detection part (section 3.2). Steady state plant data then was reconciled applying the developed model and the method described in section 3.3.

Table 3.7 shows the specifications value before and after reconciliation of data. Table 3.8 illustrates the specifications and simulated results considering Murphree efficiencies of rectification and stripping zones efficiencies and compressor efficiency equal to one. By adjusting the efficiency of both zones and compressor efficiency, top and bottom products specifications and compressor discharge temperature were approached (Table 3.9).

Table 3-6 - Specifications for case_2 (operating condition)

Stream 1	Flow Rate(t/h)	36.212	
	Molar composition	Propylene	0.7643
		Propane	0.2362
	Pressure (atm)	11.32	
	Temperature (K)	298.67	
Stream 3	Flow Rate(t/h)	11.267	
Stream 5	Flow Rate(t/h)	364.945	
Stream 16	Temperature (K)	295.265	
stream 13	Temperature (K)	303.538	

Table 3-7 **-** Reconciled and plant data

		Plant data	Reconciled data
Stream 1	Flow Rate (t/h)	36.212	34.401
Stream 3	Flow Rate (t/h)	11.267	10.711
Stream 4	Flow Rate (t/h)	24.872	23.689
Stream 5	Flow Rate (t/h)	364.945	364.260
Stream 12	Flow Rate (t/h)	350.849	360.386

Table 3-8 - Comparison for case_2 (considering E_r^{MV}=1, E_s^{MV}=1, η_{cp} = 1)

Case_2	Simulated data	Plant Data	Relative Error_%	Absolute Error
Mass Fraction of Propane in stream 4	1e-7	0.0027	99.996	0.0026999
Mole Fraction of Propylene in stream 3	0.2227	0.21438	3.8809	0.00832
Temperature of stream 12 (k)	316.772	318.736	0.6161	1.964

Table 3-9 - Comparison for case_2 (considering $\mathbf{E_r^{MV}}$=0.3, $\mathbf{E_s^{MV}}$=1, η_{cp}= 0.87)

Case_2	Simulated data	Plant Data	Relative Error_%	Absolute Error
Mass Fraction of Propane in stream 4	0.0027	0.0027	0	0
Mole Fraction of Propylene in stream 3	0.2291	0.21438	6.8663	0.01472
Temperature of stream 12 (k)	318.536	318.736	0.0627	0.2

Mass transfer characteristics in distillation processes are commonly taken into account using vapor Murphree efficiencies. These efficiencies depend on flow regime, tray layout, and physical properties of the mixture (Lockett, 1986).

Vapor Murphree efficiencies in multicomponent systems can be predicted from tray layout, mass transfer models, including flow pattern, and correlations for binary systems using the Maxwell–Stefan approach (MUHLER;SEGURA,2000; CHAN;FAIR, 1984) or treated as a parameter to be estimated in the model. Both approaches have been used for modeling industrial-scale distillation columns (KLEMOLA;ILME, 1996; RAO et al. 2001; DAVE et al. 2003). The most common practice in RTO, adopted in the model proposed here, is treating efficiencies as adjustable parameters updated from steady state plant data (DAVE et al. 2003; LUO et al. 2012). Other factors that justify the use of efficiencies as adjustable parameters in our model are the risk of convergence difficulties due to the unbounded behavior of vapor efficiency in multicomponent systems (RAO et al. 2001; DAVE et al. 2003; LUO et al. 2012; KRISHNA et al. 1977),

the uncertainty of available correlations used to predict efficiencies (±25%), and the fact that tray efficiencies will be used to represent model imperfections due to mass transfer, phase equilibrium, and pressure drop in the column.

4. DYNAMIC MODEL

Dynamic modeling of distillation column can be carried out applying different approaches involving different levels of complexity in interaction between liquid and vapor and mathematical formulation.

In this chapter, first various types of dynamic modeling approaches are studied. The theory behind these approaches including: complex transfer rate modeling, more simplified models (reduced models) and rigorous models are introduced.

In addition, mathematical issues related to solving higher index system of differential-algebraic equations caused by dynamic modeling of vapor-liquid equilibrium is discussed. Finally, the rigorous tray by tray dynamic model of Vapor Recompression Distillation (VRD) and its validation considering experimental data of plant is presented.

4.1. TRANSFER RATE MODELING

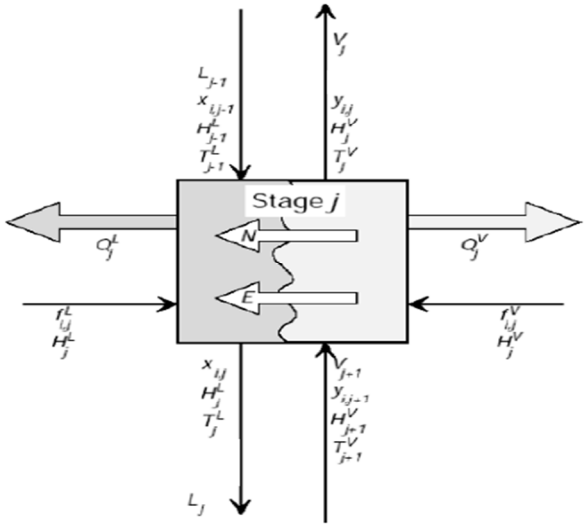
Generally, a stage of a distillation column is modeled with the hypothesis of thermodynamic equilibrium between vapor and liquid, however in reality, this equilibrium is not reachable. The first effort for considering this non-ideality, or non-equilibrium condition, is the efficiency consideration for a equilibrium stage.

To explain the deviation in equilibrium between phases, models based on transfer rates or non-equilibrium models were created. Based on this method, the distillation is modeled by transfer rates that are driven by deviation from the equilibrium and non-equilibrium condition between the phases. In the first works, transfer rates are modeled using Maxwell-Stephan (Maxwell-Stephan models).

Maxwell-Stephan model is based on the existence of liquid and vapor film with temperature and concentration gradients. These films are in contact through an interface, where the thermodynamic equilibrium between vapor and liquid phases is assumed (KOEIJER; KJELSTRUP, 2004).

A global schematic representation for a stage, where thermodynamic equilibrium does not exist, can be shown as Figure 4.1.

Figure 4-1 -Schematic diagram of a stage without equilibrium consideration (KOOJIMAN, 1995)



The wavy line in the middle of the diagram represents the interface between phases, that can be liquid and vapor (in the distillation stage case) or two liquid phases (in the extraction case).

These two phases exchange mass and energy with rates *N* and *E*. The driving force of these rates is only due to non-equilibrium consideration between the phases.

Generally, the model of a real stage or the non-equilibrium stage can be divided in three parts. First, the energy and mass balance for the liquid phase, second the balances for the vapor phase, and the third part, interface equations that represent the mass and energy transfer rates and the equilibrium relation.

In mass transfer, liquid and vapor molar rates contain a diffusive (eq.(4.1)) and a convective contribution (eq.(4.2)):

$$N_{ij}^{V} - N_{ij}^{L} = 0 \qquad (4\text{-}1)$$

$$N_{ij}^{V} = J_{ij}^{V} a_j^{I} + y_{ij} N_{tj} \quad (4\text{-}2)$$

$$N_{ij}^{L} = J_{ij}^{L} a_j^{I} + x_{ij} N_{tj} \quad (4\text{-}3)$$

where a_j^I is the total interfacial area in the stage *j* and N_{tj} total mass transfer of the stage *j* with :

$$N_{tj} = \sum_{i=1}^{c} N_{ij} \quad (4\text{-}4)$$

The diffusive flux *J*, in the matrix form is:

$$(J^V) = C_t^V \, [K^V] \, (\overline{y^V - y^I}) \quad (4\text{-}5)$$

$$(J^L) = C_t^L \, [K^L] \, (\overline{x^I - x^L}) \quad (4\text{-}6)$$

C_t^V is the total molar concentration of the vapor phase, C_t^L is the total molar concentration of the liquid phase, k^V and k^L are mass transfer coefficients, $(\overline{y^V - y^I})$ and $(\overline{x^I - x^L})$ are the medium difference of the molar fractions between the interface and the vapor and liquid phase respectively. To calculate these differences, there are many theories and correlations. Generally, these calculations depend on tray geometry and hydraulic conditions.

Similar to the mass transfer, transfer rate of the energy through the interface is written as:

$$E_j^V - E_j^L = 0 \quad (4\text{-}7)$$

The transfer rate of energy for the liquid and vapor phases is defined as:

$$E_j^V = a_j^I hv \, (T^V - T^I) + \sum_{i=1}^{c} N_{ij}^V \overline{H_{ij}} V \quad (4\text{-}8)$$

$$E_j^L = a_j^I hl \, (T^I - T^L) + \sum_{i=1}^{c} N_{ij}^L \overline{H_{ij}} L \quad (4\text{-}9)$$

hv and hl are convection heat transfer coefficients of the liquid and vapor phases T^V, T^I and T^L are temperatures of the vapor, interface and liquid and $\overline{H_{ij}}$ is the partial molar enthalpy of the components *i* in the j^{th} stage.

In addition to mass and energy transfers, the thermodynamic equilibrium through the interface should be considered.

The disadvantage of non-equilibrium model using the transfer rate is the necessity of the mass transfer coefficients matrix which needs experimental data. Unfortunately, this type of data is not easily available in the literature rendering the implementation of this model difficult. Another negative aspect is the necessity of calculating the properties of the additional mixtures, comparing to the equilibrium models. These additional properties are basically the surface tension, viscosity, thermal conductivity, in addition to mass transfer coefficients.

Despite these difficulties, non-equilibrium models can potentially represent distillation systems more accurately than equilibrium models (BIARDI; GROTTOLI, 1989; KOOJIMAN, 1995) but depend on the availability of transfer-related information.

4.2. REDUCED MODELS

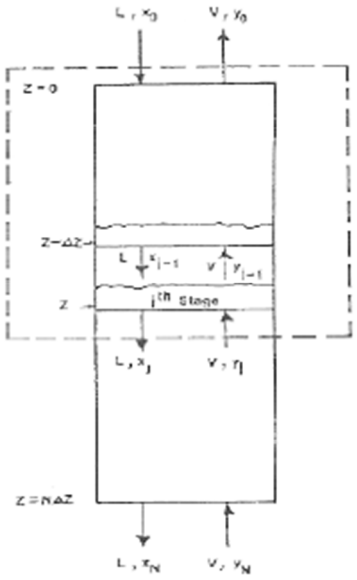
Typical rigorous model of a distillation column consists of equations that describe the components concentration, liquid and vapor flow rates, temperature, pressure drop and the equilibrium relation (or non-equilibrium) between liquid and vapor phases. For large distillation columns, due to computational cost, reduced models are recommended (MUSCH; STEINER, 1993). Additionally, many times the generated models are high index DAE. The index shows the degree of difficulty for solving DAE system. DAE systems with index equal to zero are ordinary differential equations. DAE systems with index higher than one are named higher index problems and cannot be solved directly by conventional integrators code (BRENNAN; CAMPBELL; PETZOLD, 1989).

During the decade of 1980, the first principal works about model reduction were published. In that time only integrators for ODE (ordinary differential equations) were available. DAE systems were solved sequentially, thus the simulations were so slow.

Mathematical aspects of dynamic modeling and related index concept will be discussed in more details in the section 4.3.

In 1983, series of works that applied polynomial approximation for model reduction were developed. Orthogonal collocation generally is associated to a technique for discretization of partial differential equations. Using collocation for the simulation of distillation columns is a extension of this technique (HUSS; WESTERBERG, 1996). In the work of CHO and JOSEPH (1984), the procedure for reduction of model variables by orthogonal collocation is presented, providing the basis of the explanation that will be presented in the next paragraph. The procedure of counting the trays for the author is the same as Figure 4.2.

Figure 4-2 -Trays counting (CHOE and JOSEPH ,1984)



Considering material balance for one tray of distillation column, and assuming constant molar accumulation $M_j$and molar flow rate *L* and *V*, after simplification leads to

$$M_j \frac{dx}{dt} = (Lx - Vy)_{j-1} - (Lx - Vy)_j \quad j = 1,2,\ldots N \tag{4-10}$$

Where *N* is the number of distillation trays. If the concentration profile, *x,* of a distillation column could be considered continuous, it could be written as:

$$x(z) = \sum_{k=1}^{n+2} l_k(z) x_k \tag{4-11}$$

Where *z* is the distance along the column, $l_k(z)$ are the Lagrange polynomials, x_k is the value of *x* in the arbitrary point *z* and *n* is the degree of polynomial.

Writing eq. 4.10 in terms of z:

$$M \frac{dx}{dt}\Big|_{z_j} = (Lx - Vy)_{z_j} - \Delta z - (Lx - Vy)_{z_j} \tag{4-12}$$

Where Δz is the tray spacing. n+2 equations are necessary for calculating n+2 xk , 2 corresponds to the boundary condition.

Substituting eq. (4.11) in eq.(4.12), gives :

$$M \frac{dx_j}{dt} = \sum_{k=1}^{n+2} [\, l_k(z_j - \Delta z) - l_k(z_j)\,](Lx_k - Vy_k) \quad j = 2,3, \ldots n+2 \tag{4-13}$$

Defining :

$$A_{jk} = l_k(z_j - \Delta z) - l_k(z_j) \tag{4-14}$$

Where A_{jk} is the determined constants by the degree of the selected polynomial, eq.(4.15) presents the final form as follows:

$$M \frac{dx_j}{dt} = \sum_{k=1}^{n+2} A_{jk} (Lx_k - Vy_k) \quad j = 2,3, \ldots n+2 \tag{4-15}$$

It should be noted that *N* balance equations eq.(4.10) were substituted by *n* reduced equations eq.(4.15).

Choosing a value for *n* that is much smaller than *N*, leads to enormous reduction in order (CHO; JOSEPH, 1984).

This is the basis of polynomial approximation. For reduction of complete model, this procedure should be carried out with the other equations and variables of the rigorous model.

In 1985, Stewart et al. (1985) developed another form of orthogonal collocation where state variables of each stage are approximated by the Hahn polynomial. As the number of collocation points approximates the number of real trays of the system, the response of the reduced system that approximates the response of the complete system gets better with the higher order of approximation allowed by the method (one point of collocation per stage).

STEWARD et al. (1985) and PINTO et al.(1988) tested new strategies for model reduction and analyzed different kinds of orthogonal polynomials. The principal aim of the previous works is the fact that feed tray and the ends of the column (condenser and reboiler) should be considered as the points of collocation and interpolation in order to model reduction gives the best results. Beside this, different orthogonal polynomials than the usual ones (Lagrange, Jacobi and Hahn) were created and applied for the successful model reduction.

BENALLOU et al. (cited MUSCH; STEINER, 1993) proposed one important method of the model reduction based on compartmental models. These models are based on the fact that adjacent trays have small flow rate and temperature differences, so can be considered as a group in “compartments”. Especial energy and mass balances and hydrodynamic relations were developed for these blocks, where only one tray is modeled, that is the reference tray of the block. Various uniform temperature, pressure and flow rates are considered inside the compartments, thus the algebraic equations between the compositions are omitted from each block.

To improve the method of BENALLOU et al. (1986) and mitigate some simplifications disadvantages, MUSCH and STEINER (1993) carried out some modifications in the original model, involving substitution of vaporization efficiency by the Murphree

efficiency and adding components balance differential equations instead of algebraic concentration relations. The principal simplifications of this model, compared to complete rigorous model of distillation column are:

- All of the trays inside a block have the same accumulation;
- The liquid and vapor flow rates inside a compartment are uniform;
- Temperature profile between the adjacent trays until the reference tray is linear;
- Pressure drop of each tray inside a block is equal to pressure drop of the reference tray of the same compartments

Considering theses simplifications, the reduced model contain the following equations:

Per tray:

- (number of components -1) differential mass balance equations ;
- (number of components) equations of equilibrium relation between the phases;

Per block (per compartment):

- One global mass balance differential equation;
- One energy balance equation;
- One equation for calculating the liquid and one for calculating the vapor flow rate;
- One equation for calculating pressure drop;
- One equation of temperature profile interpolation;

The accuracy of model response depends on the number of blocks in which the column was reduced to.

Another form of model reduction was proposed by OSORIO et al. (2004), with substitution of differential- algebraic (DAE) equations by ordinary differential equations (ODE). The algebraic equations, mainly those that contain phase equilibrium, are substituted by a neural network that is perfectly adjusted using experimental data. For this adjustment, a large quantity of experimental data is necessary. This combination of rigorous models with empirical reduced models reduces the simulation time to 40%. Despite its great efficiency, it cannot be applied for predictions outside of the zone that was not involved the experimental data, limiting its capacity of extrapolation.

Additionally, linear low-order models which are obtained from linearizing nonlinear model, are suitable for controller design. Front or wave approach introduced by GILLS and RETZBACH (1980), is not as accurate and general, although it may be useful for some separations.

Many other works with different applications and methodologies were found in the literature, among them, BETLEM (2000) and DIWEKAR et al.(1987) by applying reduced models in batch distillation and a series of works of SIRIVASTAVA and JOSEPH (1985) showed that the degree of model reduction depends on the behavior of temperature profile and presented new methodologies for removing this limitation.

Even thought the reduced models have good precision, for dynamic of system and steady state results, they cannot substitute rigorous models.

4.3. RIGOROUS MODELS

A typical rigorous model of a distillation column consist of equations that describe the components concentration, liquid and vapor flow rates, temperature, pressure drop and the relation between liquid and vapor phases (Appendix C). However, even in this "rigorous" model a number of model simplifications are included. These typically include prefect mixing in both phases on all the trays, thermal and thermodynamic equilibrium between the phases (100% tray efficiency or possibly some simple Murphree relationships for the efficiency of each component).

As for solving steady state modeling, in the dynamic case, there are sequential and modular approaches.

In dynamic sequential modular simulators the states and input stream values in each module are known at the beginning of each time step. The values at the following time step might be calculated explicitly from the states and input values at the present time step. Another possibility is to calculate the values at the following time step implicitly by sequential calculations. Here the output values at the present time step from the module earlier in the sequence, are used as input values to calculate both the states and the

output of the actual module. For both schemes each module might contain its own integration routine, and the flowsheet calculation might be a combination of explicit and implicit calculations.

When the equation oriented approach is applied to dynamic cases the equation system is a set of ordinary differential and algebraic equations (DAE's), which is usually stiff, sparse and nonlinear. When solving stiff equations, the time step is determined from stability considerations not from accuracy considerations. Hence the ideal integration method for stiff equations is an implicit integration method.
Considering the semi-explicit DAE system as follows:

$X' = f(X,Z,t)$ and $g(X,Z,t) = 0$

Where:

X- differential variables

Z- algebraic variables

$y^T = [X^T Z^T]$

It is solved using an extension of ODE solvers. There are two approaches for solving these systems.

1. Nested Approach:
    - given X_n, solve $g(X_n,Z_n) = 0 ===> Z_n(X_n)$
    - using ODE method evolve $X_{n+1} = \Phi(X_n,Z_n(X_n),t_n)$

This is the most common approach:

- requires $Z = Z(X)$ (implicit function)
- required if only an explicit method is available (e.g. explicit Euler or Runge-Kutta)
- can be expensive due to inner iterations

2. Simultaneous Approach (GEAR, 1971):

Solve $X' = f(X,Z,t),\ g(X,Z,t) = 0$ simultaneously using an implicit solver to evolve both X and Z in time.

- requires an implicit solver
- much more efficient
- provides for more flexible problem specifications

For instance, consider a Backward Differentiation Formula (BDF) solver. For a semi-explicit system, we can write:

$$X_{n+1} = h\,\beta_{-1}\,f(X_{n+1},Z_{n+1},t_n) + \sum_{j=0,k} a_j X_{n-j} \quad (4\text{-}16)$$

$$g(X_{n+1}, Z_{n+1}, t_{n+1}) = 0 \quad (4\text{-}17)$$

and this system can be solved for X_{n+1}, Z_{n+1} using Newton's method.

$$\begin{bmatrix} I - h\beta_{-1}\frac{\partial f}{\partial X} & -h\beta_{-1}\frac{\partial f}{\partial Z} \\ \frac{\partial g}{\partial X} & \frac{\partial g}{\partial Z} \end{bmatrix} \begin{bmatrix} \Delta X_{n+1} \\ \Delta Z_{n+1} \end{bmatrix} = - \begin{bmatrix} X^l_{n+1} - \sum_{j=0}^{k} X_{n-j}\alpha_j - h\beta_{-1} f(X^l_{n+1},\ Z^l_{n+1}, t_{n+1}) \\ g\ (X^l_{n+1}, Z^l_{n+1}, t_{n+1}) \end{bmatrix} \quad (4\text{-}18)$$

Note that the Jacobian matrix is nonsingular at h=0 as long as $\frac{\partial g}{\partial z}$ is nonsingular (a necessary condition for the implicit function Z(X)). Thus if $\frac{\partial g}{\partial z}$ is nonsingular, both the Nested and Simultaneous approaches should work.

The degree of difficulty in solving DAE system depends on the singularity condition of $\frac{\partial g}{\partial z}$ which relates to the system index (PETZOLD, 1982).

For semi-explicit DAE system, the index is the minimum number of times that g(X,Z,t) must be differentiated with respect to time in order to yield a pure ODE system:

$$X' = f(X,Z,t) \quad (4\text{-}19)$$
$$Z' = s(X,Z,t) \quad (4\text{-}20)$$

A DAE system of index 1 contains a structurally non-singular subset of algebraic equations ($\frac{\partial g}{\partial z}$ is nonsingular). Therefore, this can be solved together with the subset of ODEs or separately at any instant of time, given the initial values of differential variables.

Besides this, as in a DAE with index 1, $\frac{\partial g}{\partial z}$ is nonsingular, one can specify X(0) freely and solve for Z(0) from g(X(0),Z(0),0) = 0. But for high index problems (index > 1), as $\frac{\partial g}{\partial z}$ is singular, X(0) can not be specified freely and we need to analyze and reformulate the DAE to get consistent initial conditions. In particular, consistent initial conditions for such systems must satisfy not only the original equations themselves, but also the first or higher-order differentials with respect to time of some of these equations. Consequently, the number of variables for which arbitrary initial values may be specified is less than the number of differential equations in the system (PANTELIDES, 1988). Soares and Secchi in 2012, studied an alternative approach that can be used to initialize DAE systems. In their work, an algorithm was developed that determines which variables can be specified for both the low and high-index systems so as to obtain a consistent initial values.

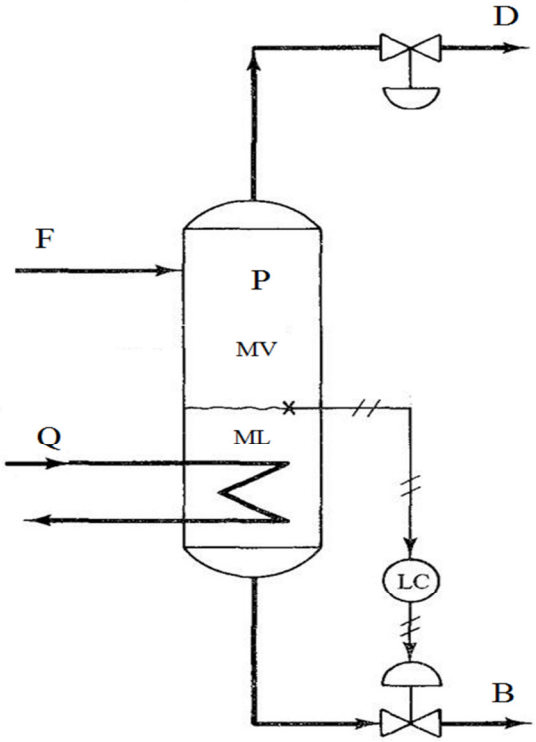
For high index DAEs, solvers that are capable of solving this kind of system can be applied or system index can be reduced to one (Index reduction). Index reduction can be implemented manually by adding algebraic equations or algorithms such as Pantelides algorithm and Dummy Derivative algorithm.

Pantelides algorithm determines the equations that must be differentiated in order to make the high-index problem solvable (PANTELIDES, 1988). Dummy Derivative algorithm makes a clever selection of the dummy derivatives in order to obtain a non-singular reduced-order system (MATTASON, 1993).

For showing the index problem and its numerical solution difficulties in our case, a flash tank example is chosen and its index analysis was carried out.

As shown in Figure 4.3, one component mixture enters the flash tank in which the liquid hold up (M_{liq}) and volume of vapor (V_v) are constant.

Figure 4-3- Flash Tank



For a one-component mixture the assumption of vapor-liquid equilibrium gives that the pressure (*P*) must equal the saturation pressure for the component

$$P = P^{sat}(T) \tag{4-21}$$

Consequently, specifying the tank pressure is equivalent to specifying its temperature *T*. Furthermore, because V_v is constant, M_{vap} is directly related to *T*.

Also in this case considering the "perfect control" assumption, M_{liq} can be considered as constant. Assume that *B* (outlet liquid flow) is used to keep the level constant (M_{liq}).

In addition to the feed the independent variables are then *D* and *Q*. Given these independent variables we may obtain the independent variables *T* (temperature) and *B* as functions of time using:

$$\frac{d(M_{vap})}{dt} = F - B - D \tag{4-22}$$

$$M_{Liq}\frac{du^L}{dt} + \frac{d\,(M_{vap}u^V)}{dt} = Fh^f - Bh^L - Dh^V + Q \tag{4-23}$$

Many numerical software packages will not be able to solve the mentioned problem. The reason is the "index problem" that arises in general whenever the equations involve differentiation of a variable which is not a state variable. State of a dynamical system is the minimum number of variables that must be specified at t=0 (in addition to the independent variables such as feed, *Q* and *B* in this case) in order to be able to solve the equations for t > 0.

In the present problem there is only one state variable, which may be chosen to be the hold-up M_{vap} (or equivalently *P* or *T* which are directly related to M_{vap}).

At a given time step the state variable M_{vap} (and consequently *P* and *T*) is known, and the user has to supply the value of $\frac{dM_{vap}}{dt}$ at this time step.

This derivation is given by eq.(4.22), but first the *B*, which is an dependent variable, must be obtained. B may be found from eq.(4.23), but this requires that time derivative of the energy in the tank is known. The term du^L/dt is approximately equal to dh^L/dt which appears on the left hand side of eq.(4.23) and may be written as:

$$\frac{dh^L}{dt} = \frac{\partial h^L}{dM_{vap}} \frac{dM_{vap}}{dt} \tag{4-24}$$

And we may substitute $\frac{dM_{vap}}{dt}$ from the material balance and change eq.(4.23) to an algebraic equation. However, evaluating the term:

$$\frac{\partial h^L}{dM_{vap}} = \frac{dh^L}{dT} \left(\frac{\partial T}{\partial P}\right)^{sat} \frac{\partial P}{\partial M_{vap}} \tag{4-25}$$

requires analytical expression for derivatives of the enthalpy equations, the saturation pressure equation and the equation of state. In general, these may be extremely tedious to obtain, in particular for multicomponent mixtures.

Although the high index problem is a numerical issue, it can be removed by not assuming M_{liq} constant, and introducing, $B = f(M_{liq})$ (the function may represent a self-regulating effect). By adding this algebraic equation, the material and energy balance give expressions for the time derivatives of the total mass $M = M_{liq} + M_{vap}$ and the total energy $U = M_{liq}u_{liq} + M_{vap}u_{vap}$ and *M* and *U* can be used as states. At the given time step,

M and *U* are known. A constant volume (tank volume) and constant internal energy flash yield the phase distribution (M_{liq} and M_{vap}) and the pressure/temperature. The derivatives of *M* and *U* may subsequently be calculated.

If one does this index analysis for a distillation tray (without simplified assumptions of the case study), would need to add two algebraic equations to calculate the vapor and liquid flow rates of the outlet streams (without fixed pressure) as mentioned in equations C15 and C16 of appendix C.

There is a variety of correlations for calculating the pressure drop (vapor flow rate) and liquid flow rate in the trays outlet. Usually, pressure drop is summarized in a equation considering liquid level, vapor flow rate and physical properties of the fluid. For instance, Wang et al. (2003) considered this correlation :

$$\Delta P = \left(\frac{V.\ \nu^V}{A_a}\right)^2 .\ \rho^V .\ \alpha + \rho^L .\ g\ .\ L_{liq} \qquad (4\text{-}26)$$

In which F^V is vapor flow rate, L_{liq} is the liquid level on the tray, α is the pressure drop coefficient in the dry tray, *A* is the active area per tray, and ν^V is the molar volume of the vapor phase. The majority of correlations to calculate the liquid flow rate are based on Francis classical equation, in which flow rate is related simply to liquid level on the tray. Wang et al. (2003) developed this correlation:

$$L = \alpha_w\ .l_w .\ \frac{\left(\frac{L_{liq}-\beta.h_w}{\beta}\right)^{\frac{3}{2}}}{\nu^L} \qquad (4\text{-}27)$$

Where L_{liq} is the liquid level on the tray, ν^L is the molar volume of the liquid phase, α_w and β are the parameters that should be adjusted.

4.3.1. DYNAMIC MODEL FOR VRD UNIT

Vapor recompression distillation (VRD) column is favored for separation involving close-boiling liquids. Such separations result into large reflux ratios and a small compressor duty is needed to facilitate the heat transfer in a combined reboiler-condenser. Reboiler-Condenser combination leads to a significant amount of energy recycle through the combined, which introduces strong interaction between different units in this system. Furthermore, there is also a large amount of material recycle owing to the large reflux flows. The tight material and energy integration in vapor recompression distillation shows a potential for complex dynamics (FITZMORRIS; MAH, 1979; FERRE; CESTELLS; FLORES, 1985).

The main difference between vapor recompression distillation and conventional distillation is in the way in which energy is added and removed from the column. In a conventional stream-heated, water-cooled distillation column, the dynamic changes in energy addition (reboiler-duty) or removal (condenser duty) can be varied independently. In addition, the dynamics of the reboiler and condenser are typically very fast compared to the dynamics of the column (MUHRER, 1990).

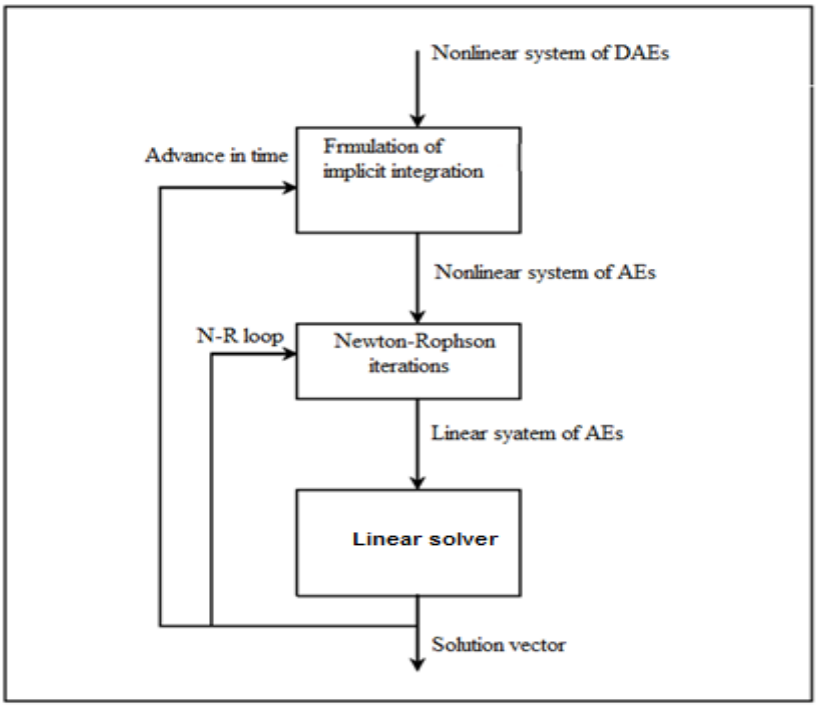
Most research on vapor recompression distillation has emphasized steady state economics (NULL, 1976), focusing on capital cost, operating costs and optimal steady state operating conditions. However, very few papers have focused on the dynamics of these columns ((QUADRI, 1981), (MUHRER et al. 1990)).

In this work, rigorous tray by tray modeling based on vapor and liquid thermodynamic equilibrium was developed (Appendix E). Distillation trays were modeled dynamically according to equations of Appendix C. Sump and accumulator dynamics follows the same equations, however, instead of equations C15 and C16 top and bottom products control the levels of accumulator and sump respectively. Dynamics of expansion valves, trim heat exchanger, compressor and pump are considered fast and are neglected (LUYBEN, 2004). Reboiler-condenser was modeled as the steady state model but heat is transferred from the reboiler side to the condenser side with a first order lag (GROSS, 1998).

In addition, vapor hold up must be considered due to high operating pressure. In columns operating at moderate pressures (less than 10 atm), ignoring vapor hold up is a good assumption, because vapor densities are much lower than liquid densities. This means that most of the material on the trays is in the liquid phase, despite the fact that the volume of vapor is typically a factor of 10 greater than the volume of liquid. However, as pressures increase toward the critical pressures of the components being distilled, the difference between liquid and vapor densities decreases, and vapor hold up becomes more important (CHOE and LUYBEN, 1987). In propylene-propane system, vapor hold up represents about 30% of the total hold up and has to be considered (CHOE and LUYBEN, 1987).

The steady state modeling, dynamic model was developed in Equation Oriented environment using EMSO. For solving DAE problems, EMSO applies the algorithm shown in Figure 4.4 using DASSL (PETZOLD, 1983) or SUNDIALS (HINDMARSH et al., 2005), but it is not able to solve higher index DAE problems with conventional thermodynamic packages. This is due to the incapability of EMSO's thermodynamic package in calculating the thermodynamic properties derivatives of order higher than one, thus DAE index must be reduced to one.

Figure 4-4 - EMSOs algorithm for solving DAE



Specifying the pressure profile (because it is valid for moderate to high pressure columns where the relative changes in pressures are typically small (CHOE and LUYBEN, 1987)) and calculating vapor flow from algebraic equation is a usual approach for dynamic modeling of distillation column, but this configuration leads to high index problem.

For achieving an index one formulation, we should add two algebraic equations to calculate liquid and vapor flow rates in which the pressure profile is not fixed..

Equations C15 and C16 of appendix C were considered as:

$$V = V^{ref} + G^*(\Delta P - \Delta P^{ref}) \quad (4\text{-}28)$$

$$L = L^{ref} + G^*(L_{liq} - L_{liq}^{ref}) \quad (4\text{-}29)$$

In which *ref* mentions the value of variables in the initial point.

These algebraic equations calculate liquid and vapor outlets which are function of liquid hold up and pressure (or pressure drop) respectively. For the accumulator, the inlet is a mixture of liquid and vapor. Additionally, in eq.(4.28) vapor outlet has linear relationship with pressure difference instead of delta pressure for accumulator and sump dynamic models.

The *G* values in these equations are carefully adjusted to reflect the real overall dynamic behavior of the system (GROSS, 1998b).

As can be seen, the common correlations mentioned in section 4.3 for calculating liquid and vapor flow rates were not applied here. This is due to the fact that, these correlations involve parameters that vary according to the process design and operation condition and inaccuracy in their values can affect the model response considerably.

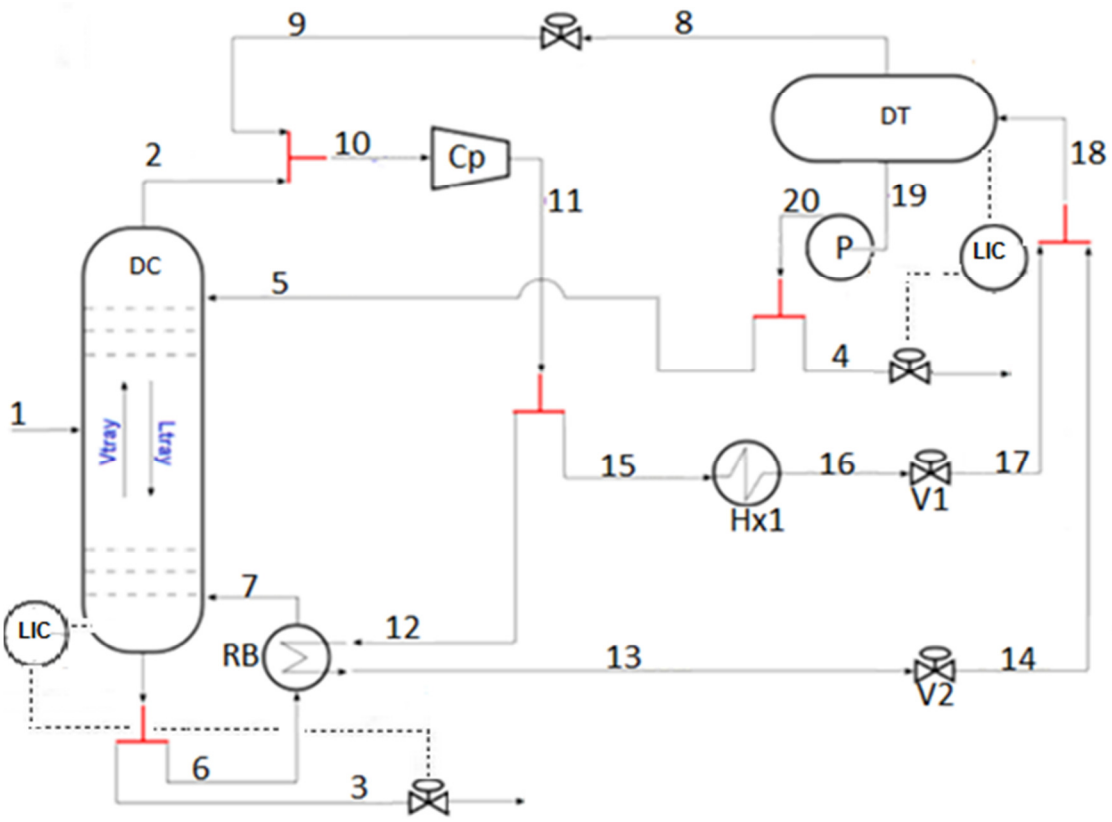
For distillation trays, choosing large values of G in eq.(4.28) and eq.(4.29) is equivalent to keeping the value of pressure and liquid hold up approximately constant. As in the studied system, pressure is tightly controlled by auxiliary cooler (HX1) and the dominant composition dynamics are much slower than the flow dynamics (nearly unaffected by the flow dynamics (LEVY; FOSS; GRENS, 1969)), equations 4-28 and 4-29 could be applied.

By applying equations (4.28) and (4.29), the problems that would arise due to errors in the parameters values of common equations like (4.26) and (4.27) were avoided and computational time was reduced considerably.

4.3.1.1. VALIDATION OF DYNAMIC MODEL

Dynamic validation of the model was performed in two cases. In the first case, design condition (Table 3.2) was studied and step changes in reflux flow rate (stream 5) and hot stream flow rate through the reboiler side of RB (stream 12) were considered. Step changes caused 100 kmol/h reduction in the flow rates of the streams. Distillate (stream 4) and bottom product (stream 3) flow rates were manipulated applying PI controllers to keep the sump and accumulator levels close to their set points values.

Figure 4-5 - C3 Splitter flow sheet with controllers



Tuning parameters of the controllers are based on plant data and are as follows:

Table 4-1 - Tuning parameters of controllers

Controller	K	τ	Bias
LIC (Top)	1.5	900 (s)	32.5246 (t/h)
LIC (Bottom)	0.8	10000 (s)	13.210 (t/h)

Figure 4-6 - Flow rate of stream 4 (kmol/h)

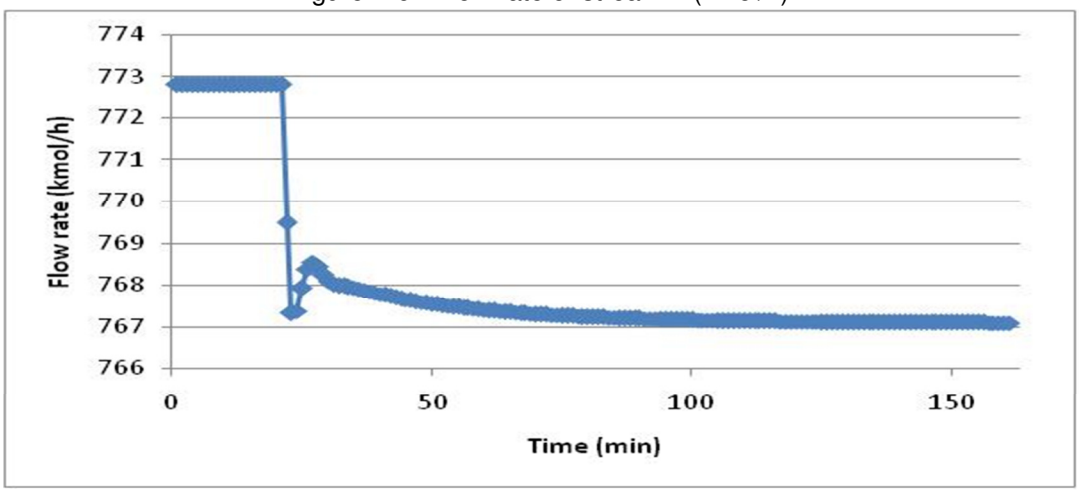


Figure 4-7 - Flow rate of stream 3 (kmol/h)

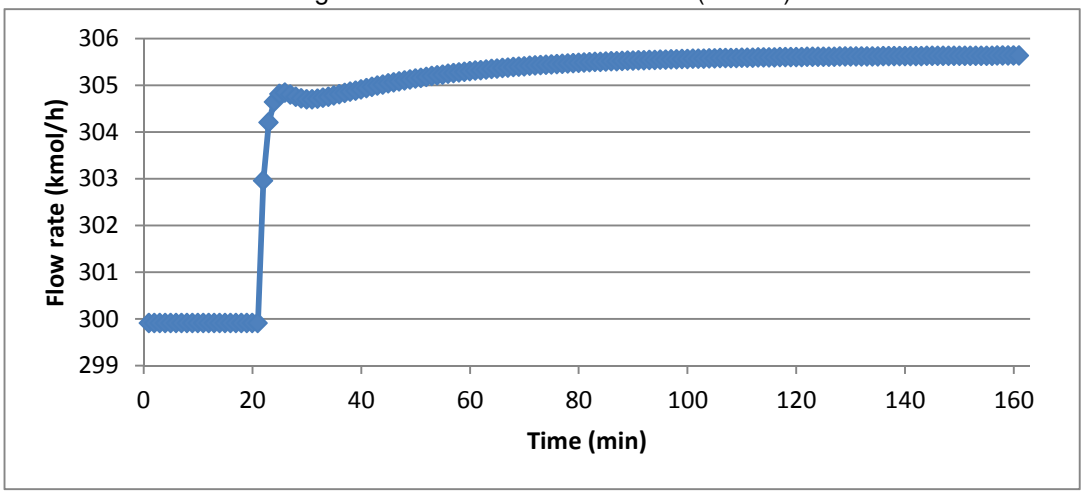


Figure 4-8 - Level of accumulator

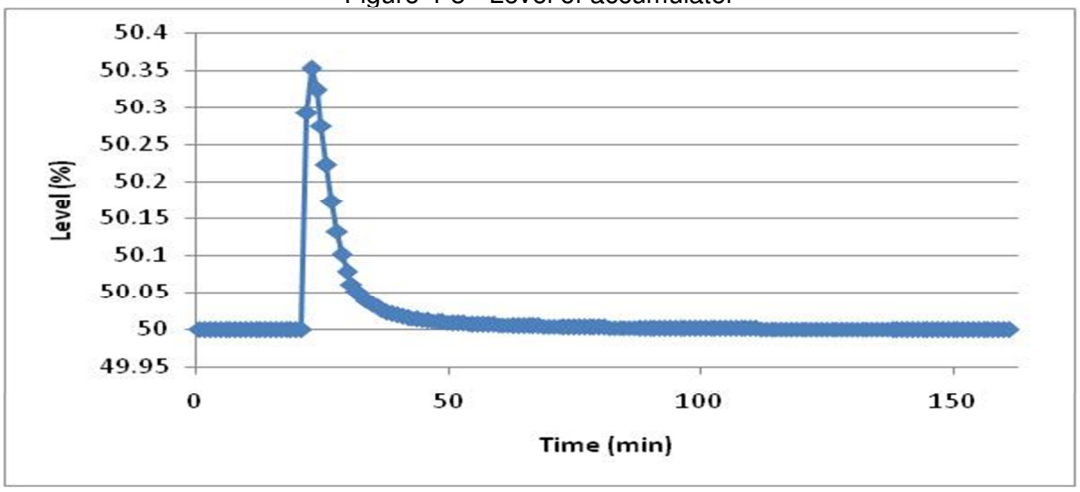


Figure 4-9 - Level of sump

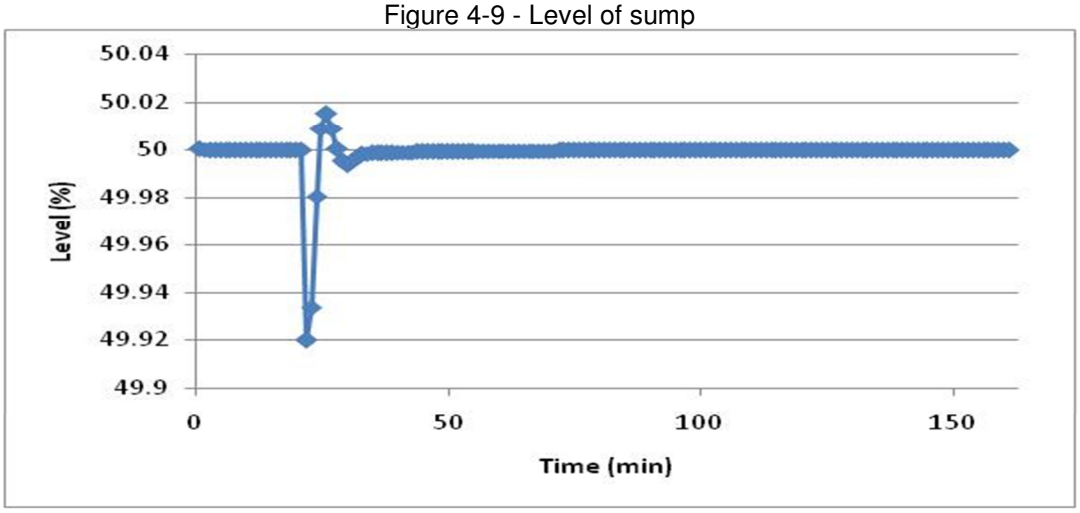


Figure 4-10 - Mole fraction of propylene in stream 4

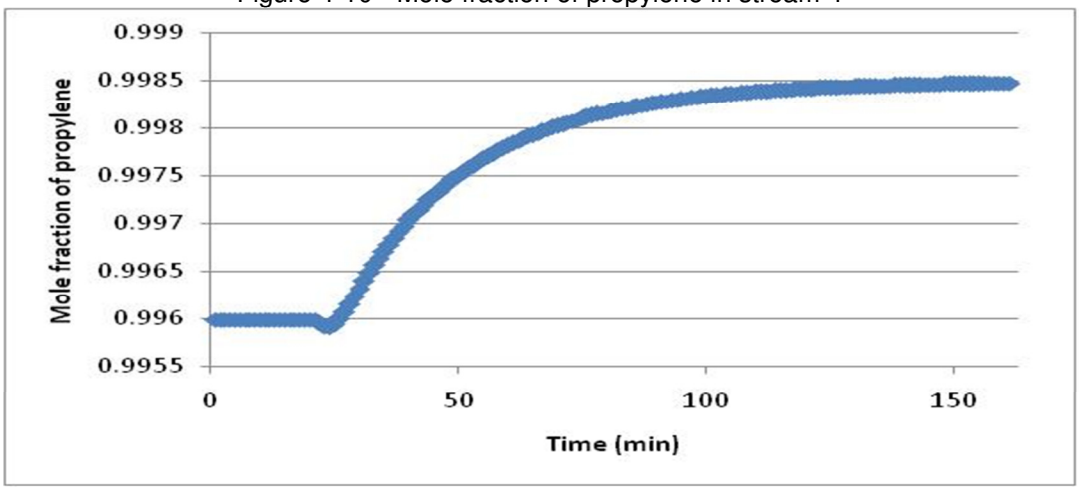


Figure 4-11 - Mole fraction of propane in stream 4

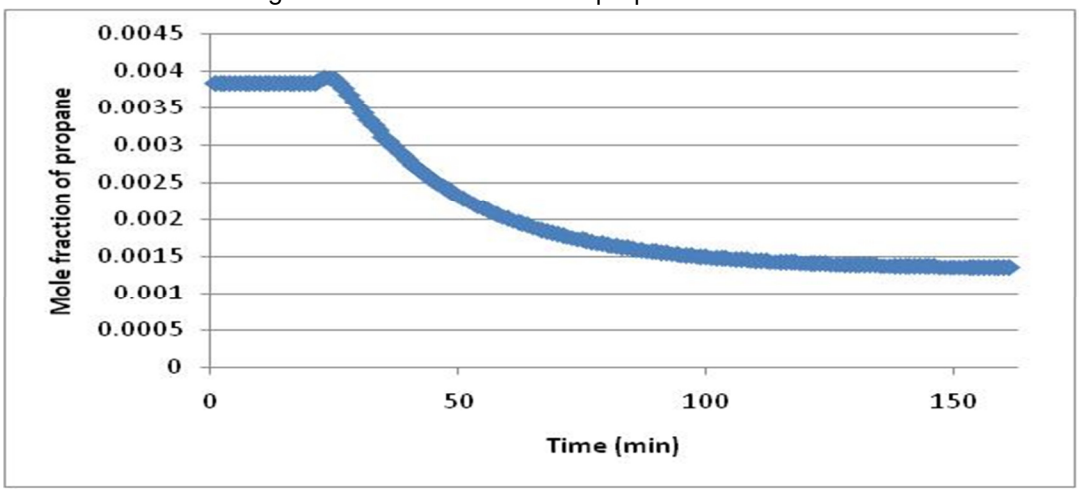


Figure 4-12 - Mole fraction of propylene in stream 3

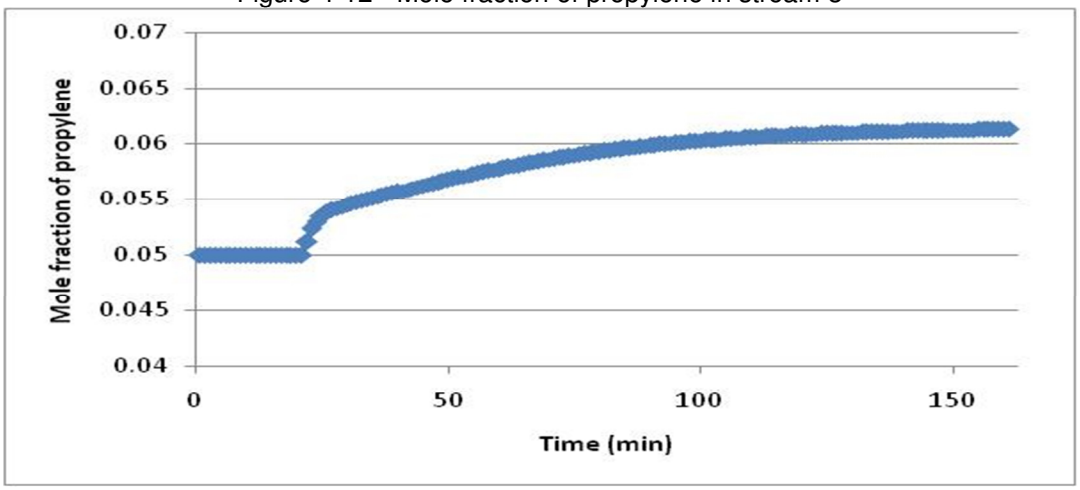


Figure 4-13 - Mole fraction of propane in stream 3

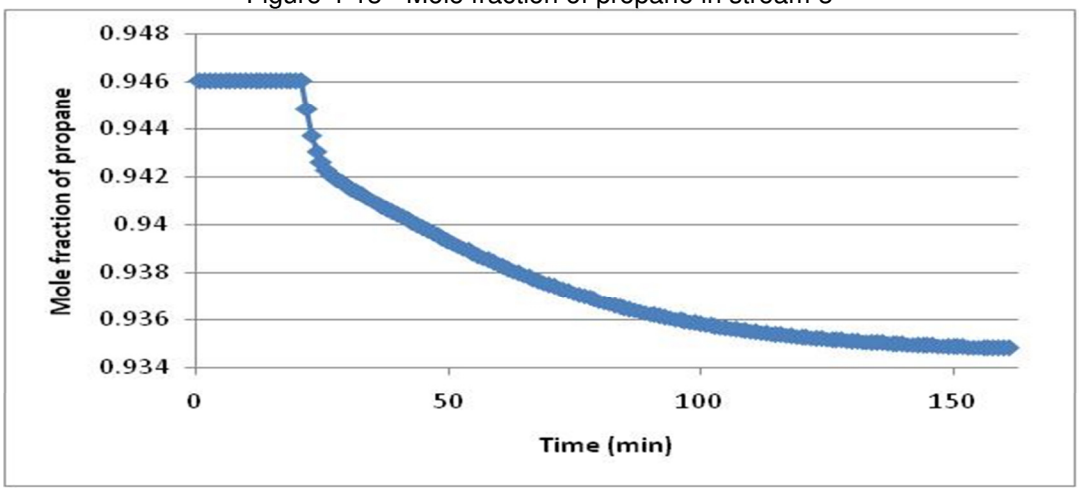


Figure 4-14 - Liquid flow rate of tray 100 (kmol/h)

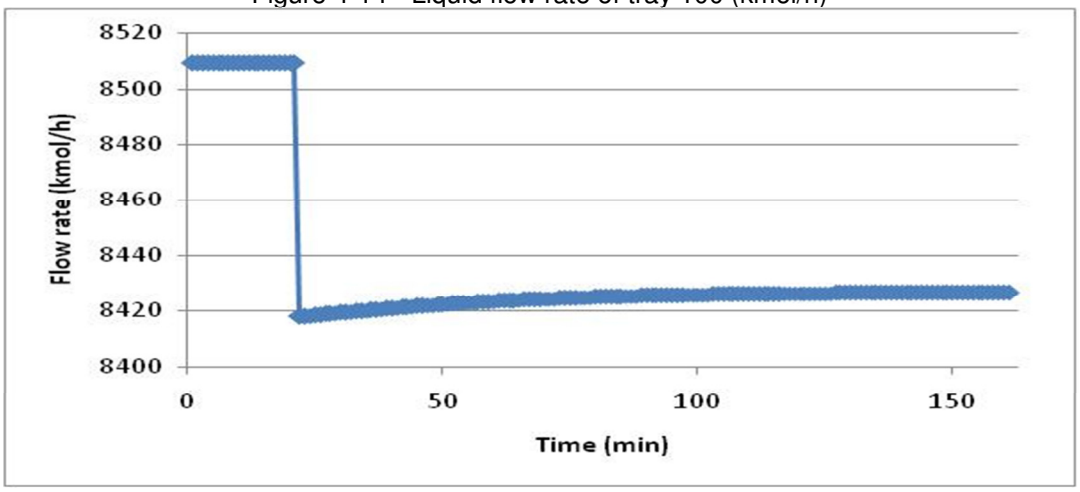
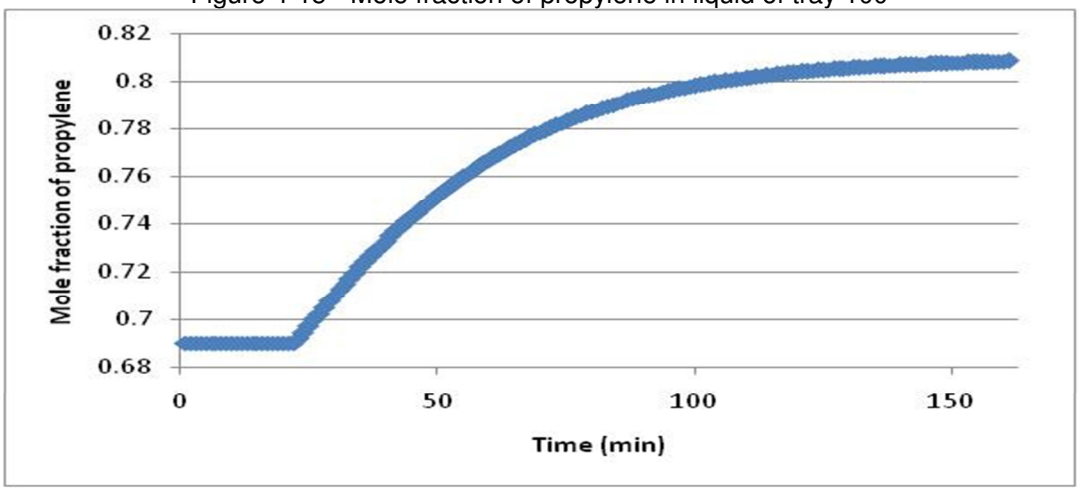


Figure 4-15 - Mole fraction of propylene in liquid of tray 100



Figures 4.6 - 4.9 show the response of the controllers to the applied steps. Figures 4.10 - 4.13 indicate the effects of reflux flow rate (stream 5) and flow rate of the hot stream through the reboiler side (stream 12) on the products composition. In the plant, flow rates of these two streams are manipulated based on the compositions of propane at the top product and propylene at the bottom product respectively.

Reduction of reflux flow rate increases the molar component flow rate of both components at the top product, although purity of propylene decreases. This control strategy could be applied when the top product purity is more than the minimum required specification (0.995 molar concentration of propylene).

Table 4-2 - Summary of reflux (stream 5) flow rate reduction influence

Reflux reduction	Molar rate of propane	Molar Rate of propene	Mole fraction of propane	Mole fraction of propene
Bottom product	↓	↓	↑	↓
Top product	↑	↑	↑	↓

On the other hand, the reduction of the flow rate of the hot stream through the reboiler side of RB increases the molar composition of propylene in both products and consequently decreases the molar composition of propane. However, molar flow rate of both propylene and propane at the bottom product increased and at the top product decreased.

Table 4-3 - Summary of reboiler inlet (stream 12) reduction influence

Reboiler inlet reduction	Molar rate of propane	Molar Rate of propene	Mole fraction of propane	Mole fraction of propene
Bottom product	↑	↑	↓	↑
Top product	↓	↓	↓	↑

Figures 4.10 - 4.13 show that the hot stream flow rate (stream 12) reduction dominated the reflux flow rate reduction, as can be understood from the dynamic behavior of the mole fractions and flows.

Figure 4.14 and 4.15 highlight the difference between column internal composition and internal flow rate dynamics. Composition curve shows a much slower dynamic comparing to the flow rate.

In the second case, the dynamic simulation was performed for more than two days of operation in order to verify the quality of model response compared to real plant data. In

this case study the proposed dynamic model uses the adjusted vapor efficiency values from steady state validation (see section 3.5).

In this validation simultaneous variations in reflux flow rate (stream5), flow rate of reboiler hot side (stream 12), feed flow rate (stream1) and composition of feed were considered (Figures 4.16 - 4.20). Flow rates of bottom product (stream 3) and distillate product were specified as the plant data, since currently they are manipulated manually in the plant so as to control the sump and accumulator levels. It is should be noted that flow rate of stream 12 (Figure 4.20) was modified based on the reconciled data in the initial point.

Figure 4-16 - Flow rate of stream 1 (t/h)

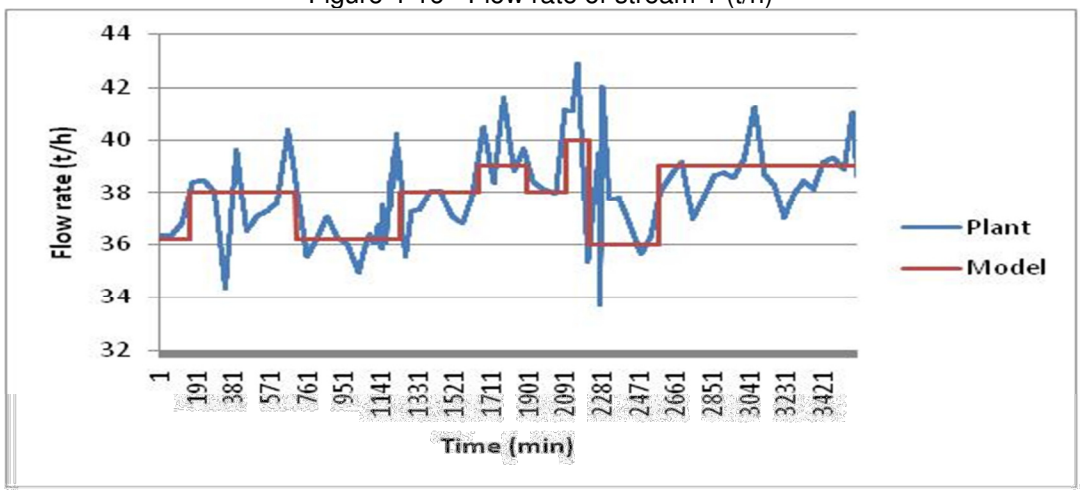


Figure 4-17 - Mole fraction of propylene in stream 1

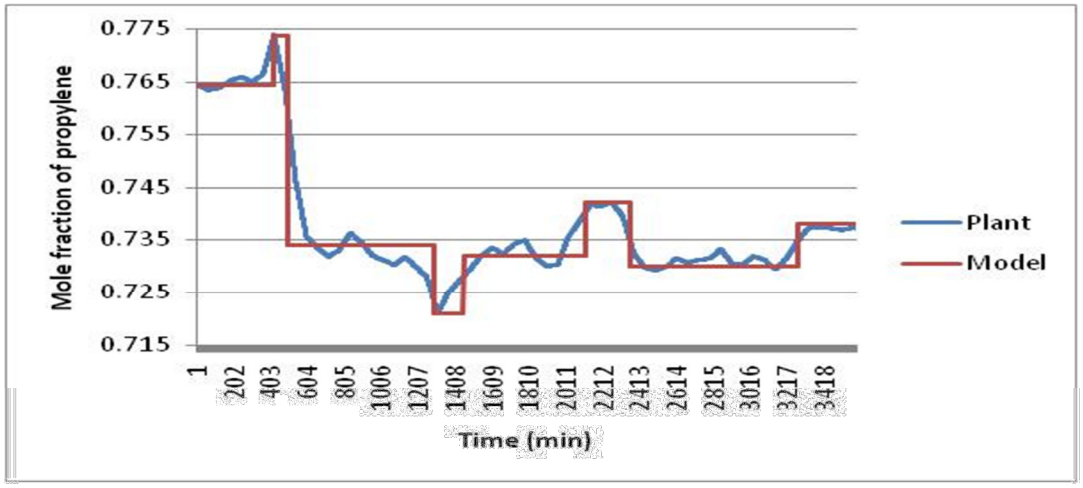


Figure 4-18 - Mole fraction of propane in stream 1

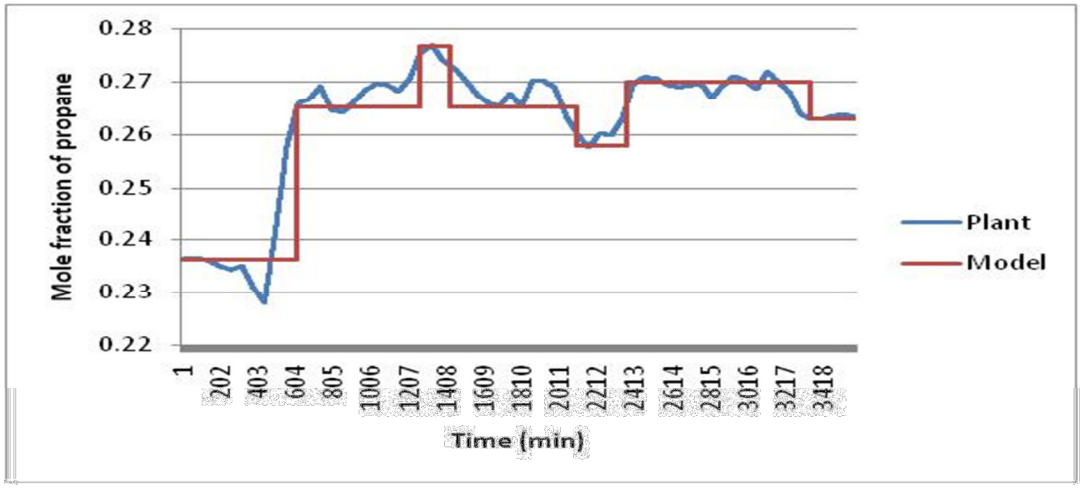


Figure 4-19 - Flow rate of stream 5 (t/h)

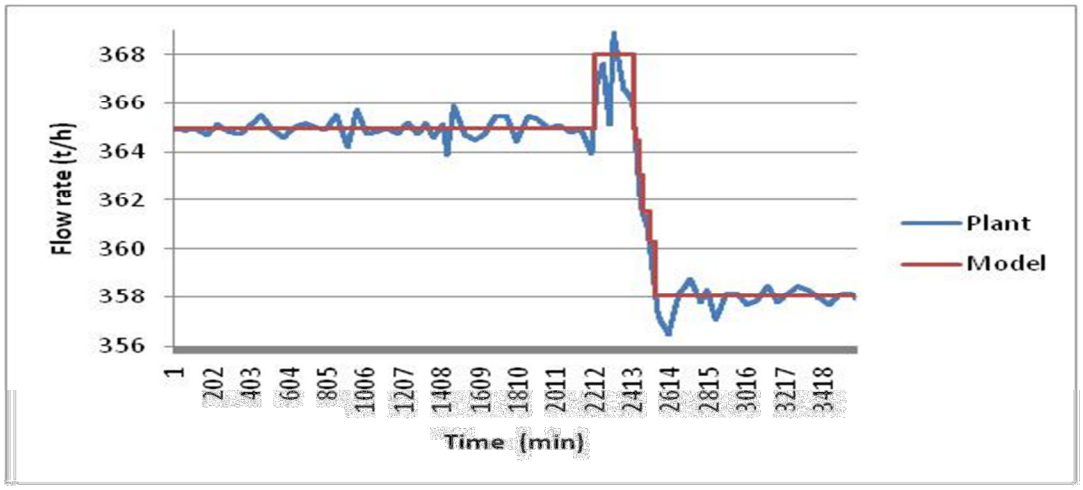


Figure 4-20 **-** Flow rate of stream 12 (t/h)

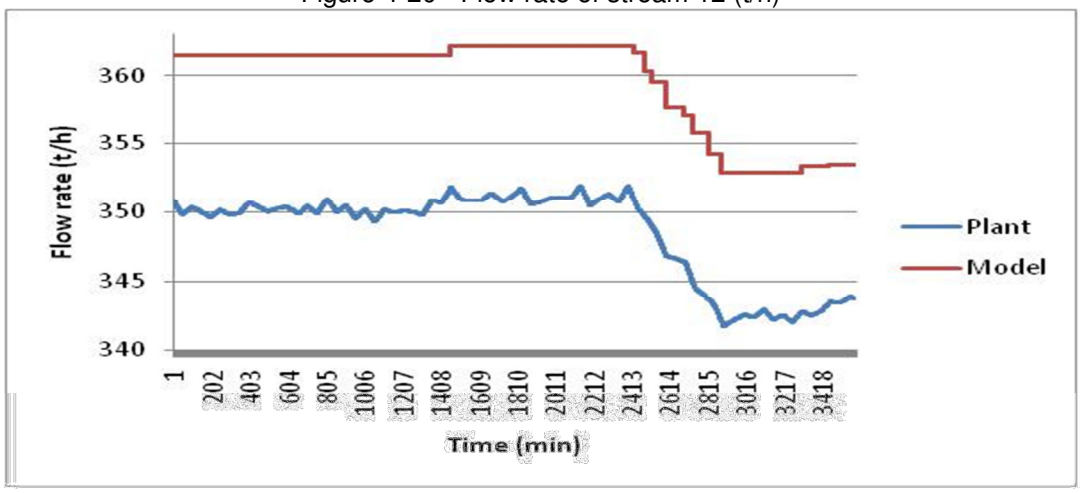


Figure 4-21 - Mole fraction of propylene in stream 3

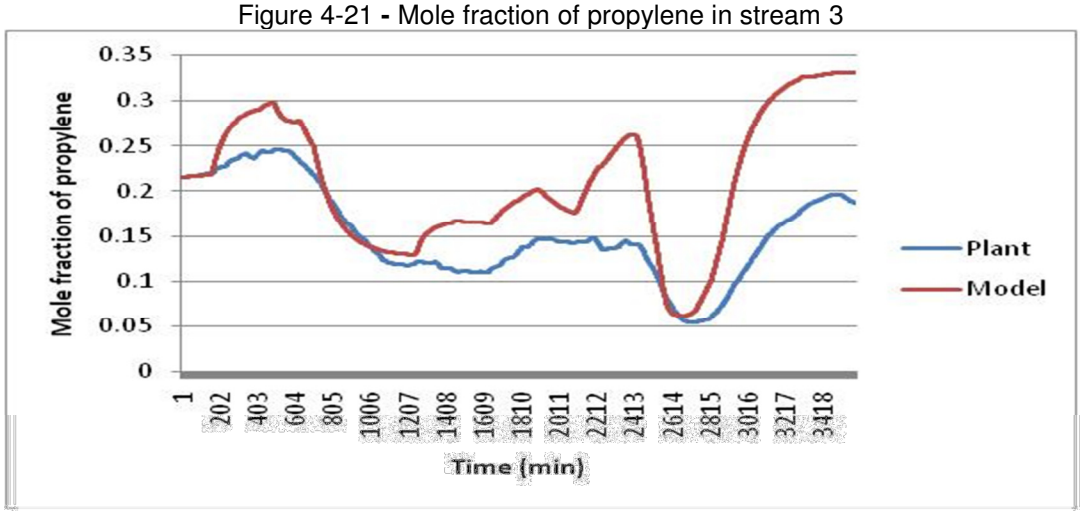


Figure 4-22 - Mole fraction of propylene in stream 4

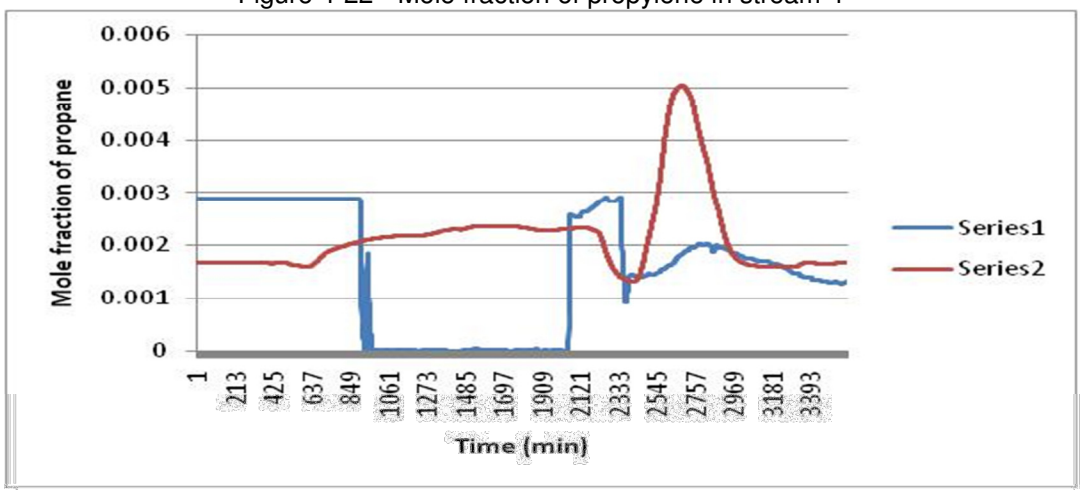


Figure 4-23 - Temperature of tray 85 (K)

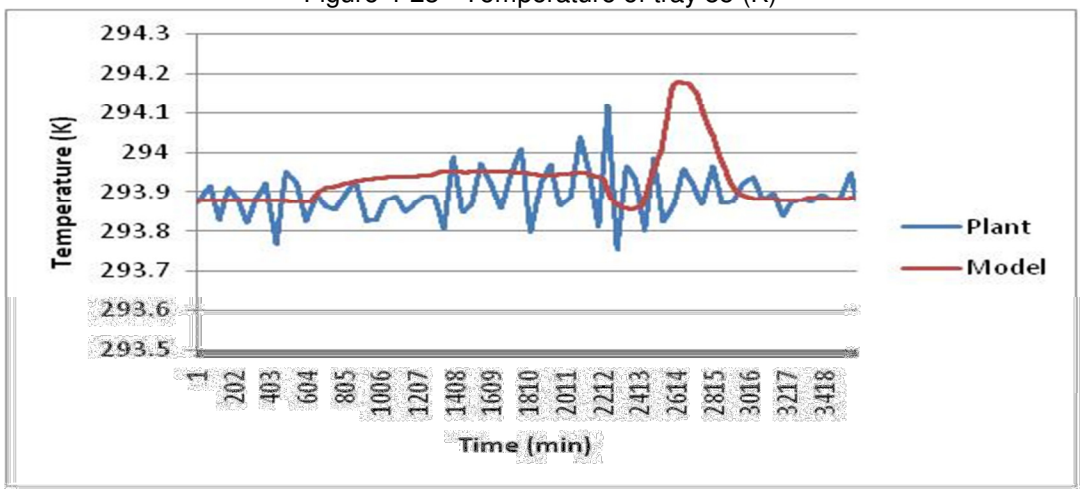


Figure 4-24 -Temperature of boil up (k)

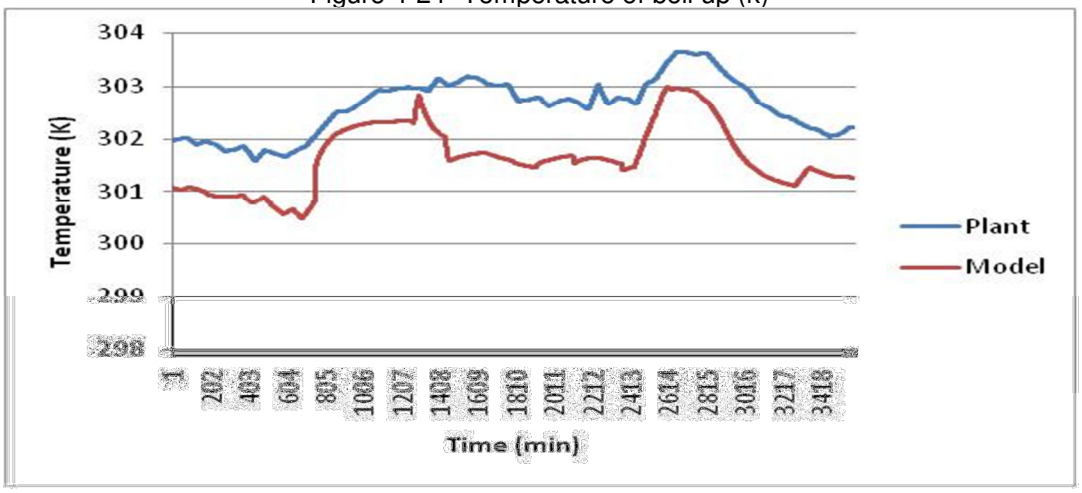


Figure 4-25 -Temperature of tray 17 (k)

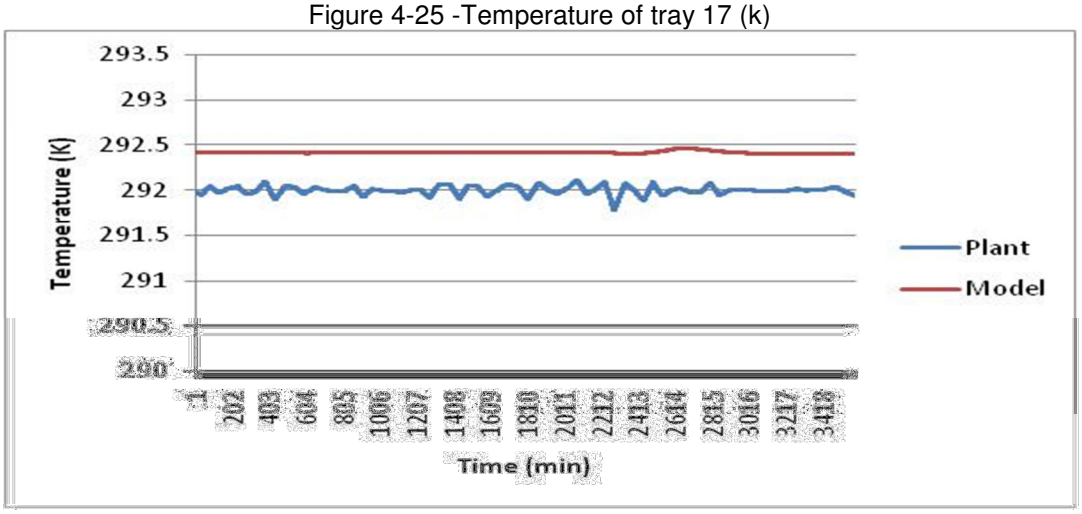
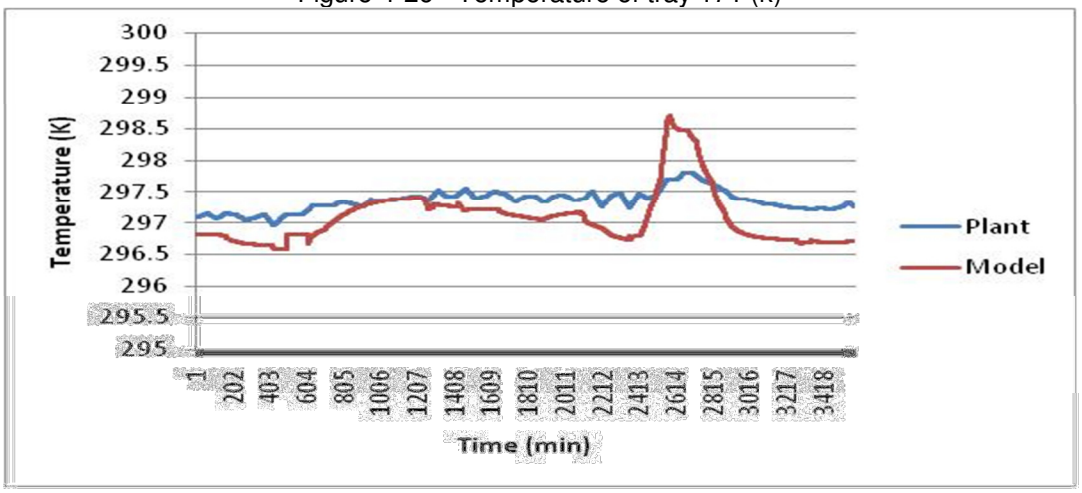


Figure 4-26 **-** Temperature of tray 171 (k)



Temperature variations of boil up, temperature of trays17, 85, 171, composition of propylene in distillate (stream 4) and propane in the bottom product (stream 3) were compared with real plant data in Figures 4.16 through 4.26.

Products quality control is a problem of making precise adjustment of heat addition and rate of heat removal from the tower. Heat removal determines the internal reflux flow rate and heat added determines the internal vapor rate. These internal vapor and liquid flow rates determine the circulation rate, which is the degree of separation between two key components.

Figure 4.21 shows that the propylene composition in the bottom product was highly affected by the feed flow rate and propylene composition variations in the feed stream before applying changes in the manipulated variables. It is due to the fact that feed enters the column on tray 157 that is very close to the bottom of the column.

After changing the manipulated variables, their effects on the dynamic responses dominated the effects of the entrances.

The dynamic response of temperature in the bottom and propylene concentration in two products can be explained according to the changes of reflux flow rate and thereafter flow rate of reboiler hot side. First, the decrease of reflux flow rate led to a slight increase in concentration of propane in the top product, however the decrease of reflux rate decreased the propylene amount (mole flow) inside the column and consequently at the bottom product. Next, as the flow rate through reboiler hot side continued to go down, its effect on the products composition appeared. It led to less heat entering the column that increased the propylene concentration at the bottom product and decreased the propane concentration at the top product.

With respect to dynamic responses of temperatures, their behaviors can also be explained according to the changes of manipulated variables as same as the compositions dynamic.

Trays temperatures were approximately constant during the studied period, which is natural in vapor recompression distillation columns as the difference between top and bottom temperature is very low.

As shown in figures, model can predict dynamic behavior of the plant properly. The main differences between model predicted values and plant data are in propane composition at the top product. This can occur due to uncertainties in composition measurements, in flow rate measurements as was illustrated in data reconciliation (section 3.1.2). Additionally, neglecting some minor components in the feed, and some inaccuracies in thermodynamic parameters are some of the probable causes. Ignoring the dynamics of the condenser side of the reboiler-condenser which is a flooding submerged condenser and considering the first order lag for exchanging the heat load provided by the condenser side can be another reason for differences between plant and model responses.

Besides this, there is the delay between the model and experimental data, this is due to the online analyzer (gas chromatography) delay. For columns like propylene-propane splitter (high reflux ratio), that have slow-responding composition dynamics, the composition dynamics for the primary product has a time constant of about 2 hours and analyzer design is less of an issue. In this case, when the cycle time for the analyzer increases from 5 minutes to 10 minutes, it does not significantly affect the feedback control performance.

5. CONCLUSION AND DIRECTIONS FOR FUTURE WORK

Dynamic and steady state modeling and simulation of highly interlinked vapor recompression distillation column were carried out. It was possible to simulate the process without using pseudo stream approach, differently from what a recent publication suggests (CHOUDHARI et al., 2012)

The proposed steady state model will be used to execute the optimization step in the RTO cycle, due to its superior characteristics (speed, robustness and flexibility).

The steady state model is a valuable tool to gain general understanding and to assess the robustness of the model, some conclusion drawn from its analysis are the following:

(1) Slight variations of temperature profile by changing the composition of lightest and highest components were obtained. These show the possibility of treating the multicomponent mixture as a propane/propylene binary system that undoubtedly leads to the considerable reduction in computational cost. However, for binary mixture simplification, monitoring the feed composition data of plant is necessary as the presence of minor components might reduce the product purity achieved by the process, and even generate an off-specification product (Friedman, 1995).

(2) Column pressure drop has significant effect on temperature profile of the column. As Figure 3.17 shows, pressure drops effect on the composition profile of the desired product is negligible; however, the main effect of pressure is modifying the saturation temperatures and not the equilibrium compositions.

(3) The isentropic efficiency of the compressor is an important factor for the process profitability. The obtained results demonstrated that the energy demand of the compressor is highly influenced by its isentropic efficiency.

(4) Reflux ratio and bottom flow rate are important decision variables in the optimization process, because they have a direct effect on product composition and recovery as well as on the energy demand of the process.

For data reconciliation and parameter estimation in steady state mode the least square method was applied. Vapor Murphree efficiency of stripping and rectification zones and compressor efficiency values were adjusted for approaching the product specifications. This shows how the parameter estimation in RTO cycle is able to provide the desired proper model for the optimization step.

Besides this, dynamic model was proposed based on steady state model. New index one formulation was applied instead of using common phenomenological correlations. Capability of this new formulation in representing the column dynamics was demonstrated by analyzing the dynamic behavior of the column in design condition by applying step changes in manipulated variables, and comparing the simulated data with the real plant data.

Future works could focus on implementation of the steady state model in the complete RTO cycle and analyzing RTOs performance by applying the dynamic model as a virtual plant. Also, Moving Horizon Estimation (MHE) could be implemented using the dynamic model for estimating the corrected system states.

6. REFERENCES

ANNAKOU, O.; MISZEY, P. Rigorous investigation of heat pump assisted distillation. Heat Recovery Systems and CHP, v. 15, p. 241-247, 1995.

AYDIN, B.; BENALI, M. Simulation and optimization of depropanizer, C3-splitter and debutanizer. In Spring Meeting & 5th Global Congress on Process Safety; *AICHE Journal*: New York, 2009.

BENALLOU, A.; SEBORG, D.E.; MELLICHAMP, D.A. Dynamic compartmental models for separation processes. AICHE Journal, v. 32, p. 1067-1078, 1986.

BETLEM, B.H. L. Batch distillation column low-order models for quality program control. Chemical Engineering Science, v. 55, p. 3187-3184, 2000.

BIARDI, G.; GROTOLLI, M.G. Development of new simulation model for real trays distillation column. Computers and Chemical Engineering, v. 13, n. 4/5, p. 441-449, 1989.

BRENNAN, K.E.; CAMPBELL, S.L.; PETZOLD, L.R. Numerical solution of initial value problems in Differential-Algebraic Equations. New York: North-Holland, 1989.

CAMERON, I.T.Solution of DAE systems using diagonally implicit Runge-Kutta methods. *Journal of Computational and Applied Mathematics*, v. 26, p. 255-266, 1995.

CAMERON, I.T.; RUIZ, C.A.;GANI, R.A. A generalized model for distillation columns -ii- numerical and computational aspects. *Computers and Chemical Engineering*, v. 10, p. 199-211, 1986.

CAO, S.; RHINEHART, R.R. An efficient method for on- line identification of steady state. *Journal of process control*, v. 5, p. 363-374, 1995.

CHACHUAT, B.; SRINIVASAN, B.; BONVIN, D. Adaptation strategies for real-time optimization. *Computers and Chemical Engineering*, v. 33, p. 1557-1567, 2009.

CHAN, H.; FAIR, J.R. Prediction of point efficiencies on sieve trays. 1. Binary systems. *Ind. Eng. Chem. Process Des. Dev.*, 23 (4), p 814–819, 1984.

CHOE, Y-S, LUYBEN, W.L. Rigorous Dynamic Models of Distillation Columns. *Ind. Eng. Chem. Res*. 26 (10), p. 2158–2161,1987.

CHO, Y-S.; JOSEPH, B. Reduced order models for separation columns - iii - application to columns with multiple feeds and side streams. Computers and Chemical Engineering, v. 8, n. 2, p. 81-90, 1984.

CHOUDHARI, A.; GUNE, P.; DIVEY, J. Distillation optimization by vapor recompression. Chem. Eng. 119, 43, 2012.

DARBY, M. L.; NIKOLAOU, M.; JONES, J.; NICHOLSON, D. RTO: An overview and assessment of current practice. *Journal of process control*, v. 21, p. 874-884, 2011.

Dave, D. J.; DABHIYA, M. Z.; SATYADEV, S. V. K.; GANGULY, S.; SARAF, D. N. Online tuning of a steady state crude distillation unit model for real time applications. *Journal of Process Control,* v. 13, p. 267-282, 2003.

DIWEKAR, U.M.; MALIK, R.K.; MADHAVAN, K.P. Optimum reflux rate policy determination for multicomponent batch distillation columns. Computers and Chemical Engineering, v. 11, n. 6, p. 629-637, 1987.

FERRE J.A.; CESTELLES F.; FLORES J. Optimization of a distillation column with a direct vapor recompression heat pump. *Ind. Eng. Chem. Process Des. Dev.*, 1985, 24 (1), pp 128–132, 1985.

FITZMORRIS, K.E.; MAH, R.S.H. Improving distillation column design using thermodynamic availability analysis. *AICHE Journal*, v. 26, p. 265-273, 1980.

FORBES, J. F.; MARLIN, T. E.; MACGREGOR, J. F. Model adequacy requirements for optimizing plant operations. *Computers and Chemical Engineering*, v. 18, p. 457-510, 2001.

GANI, R.; CAMERON, I.T. Extension of dynamic models of distillation columns to steady-state simulation. *Computers and Chemical Engineering,* v. 13, p. 271-280, 1989.

GANI, R.; RUIZ, C.A.CAMERON, I.T.A generalized model for distillation columns - i - model description and applications. *Computers and Chemical Engineering*, v. 10, p. 181-198, 1986.

GEAR, C.W. Simultaneous Numerical Solution of Differential-Algebraic Equations, *IEEE Transactions on Circuit Theory*, CT-18, pp.89-95 , 1971.

GILLES, E.D.; RETZBATC, B. Reduced models and control of distillation columns with sharp temperature profiles. Proc. 19th IEEE Conf. on Design and Control, 865-870, 1989.

GROSS, F. Modeling, Simulation and Controllability Analysis of an Industrial Heat-Integrated Distillation Process, *Computers and Chemical Engineering*, v. 22, p. 223-237, 1998.

GROSS . F. Energy Efficient Design and Operation of Complex Distillation Processes, PhD Thesis, SWISS FEDERAL INSTITUTE OF TECHNOLOGYZURICH, 1998.

HARWARDT, A.; MARQUARDT, W. Heat-integrated distillation columns: Vapor recompression or internal heat integration. *AICHE Journal*, v. 58, p. 3740-3750, 2012.

HEYEN, G.; LEDENT, T.; KALITRENTZENFT, B.A. Modular package for simultaneous calculation of complex interlinked separation processes. *Computers and Chemical Engineering*, v. 18, p. S69-S73, 1994.

HINDMARSH, A.C.; BROWN, P.N.; GRANT, K.E.; LEE S.L.; SERBAN, R.; SHUMAKER, D.E.; WOODWARD, C.S. SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers, ACM Transactions on Mathematical Software, 31(3), pp. 363-396, 2005. Also available as LLNL technical report UCRL-JP-200037.

HUSS, R.S.; WESTERBERG, A.W. Collocation methods for distillation design. 1. model description and testing. Ind. Eng. Chem. Res., v. 35, p. 1603-1610, 1996.

JIANG, T.; CHEN, B.; HE, X.; STUART, P. Application of steady-state detection method based on wavelet transform. *Computers and Chemical Engineering*, v. 27, p. 569-578, 2003.

JOGWAR, S. S.; DAOUTIDIS, P. Vapor Recompression Distillation: Multi-Scale dynamics and Control. In American Control Conference, 2009.ACC'09. (pp. 647-652). IEEE, 2009.

KADAM, J. V.; SCHLEGEL, M.; MARQUARDT, W.; TOUSAIN, R. L.; VAN HESSEM, D. H.; VAN DEN BERG, J.; BOSGRA, O. H. A two-level strategy of integrated optimization and control of industrial processes: a case study. *European Symposium on Computer Aided Process Engineering*-12, p. 511-516, Elsevier ,2002.

KLEMOLA, K. T.; ILME, J. K. Distillation efficiencies of an industrial-scale i-Butane/n Butane fractionators. Ind. Eng. Chem. Res., 35, 4579, 1996.

KOEIJER, G.M. de; KJELSTRUP, S. Application of irreversible thermodynamics to distillation. Int. J. of thermodynamics, v.7, n.3, p. 107-114, September 2004.

KONINCKX, J. On-line Optimization of Chemical Plants Using Steady State Models. Ph.D. Dissertation, University of Maryland, College Park, p 424.(23), 1988.

KOOJIMAN, H.A., Dynamic non equilibrium column simulation. PhD thesis, Clarkson University, 1995.

KRISHNA, R.; MARTINEZ, H. F.; SREEDHAR, R.; STANDART, G. L. Murphree point efficiencies in multicomponent systems. Trans. Inst. Chem. Eng. 1977, 55, 178.

KRISHNAN, S.; BARTON, G. W.; PERKINS, J.D. Robust parameter estimation in on-line optimization, Part I. Methodology and simulated case study. *Computers and Chemical Engineering*, v. 16, p. 545-562, 1992.

KUHEN, D. R., DAVIDSON, H., Computer control. II. Mathematics of control, *Chemical Engineering Progress*, v. 57, pp. 44-47, 1961.

LEVY, R.S.; FOSS, A.S.; GRENS, E.A. Response model of a binary distillation column. Ind. Eng. Chem. Fundamentals, 8, 765-776, 1969.

LOCKETT, M.J. Distillation tray fundamentals. New York: Press Syndicate of the University of Cambridge, 1986.

LUO, N.; QIAN, F.; YE, Z.-C.; CHENG, H.; ZHONG, W.M. Estimation of Mass-Transfer Efficiency for Industrial Distillation. Ind. Eng. Chem. Res. 2012, 51, 3023.

LUYBEN, W.L. Process Modeling, Simulation, and Control for Chemical Engineers. Book, 1996.

LUYBEN, W.L. Use of Dynamic Simulation to converge Complex Process Flowsheets. *CACHENEWS*, NO. 59, 2004.

MATTASON, S.; SODERLIND, G. Index reduction in differential algebraic systems using dummy derivatives, *SIAM J. Sci. Comput.*, v. 14, P. 677–692, 1993.

MEIXELL, M. D.; GOCHENOUR, B.; CHEN. Industrial Applications of Plant-Wide Equation-Oriented Process Modeling. In Advances in Chemical Engineering; Sundmache, K., Ed.; Elsevier: Burlington, MA, Chapter 3, p 152, 2010.

MERCANGOZ, M.; DOYLE, F. J. Real-time optimization of the pulp mill benchmark problem. *Computers and Chemical Engineering*, V. 32, p. 789-804, 2007.

MUHRER, C.A; COLLURA, M.A; Luyben, W.L. Control of Vapor Recompression Distillation Columns, *Ind. Eng. Chem. Res.*, v. 29, p. 59-71, 1990.

MULLER, N.P.; SEGURA, H. An overall rate-based stage model for cross flow distillation columns. *Chemical Engineering Science,* v. 55, p. 2515-2528, 2000.

MUSCH, H. E.; STEINER, M. Order reduction of rigorous dynamic models for distillation columns. Computers and Chemical Engineering, v. 17, n. Suppl., p. S311-S316, 1993.

NARASIMHAN, S. ; MAH, S. H.; TAMHANE, A.C. A Composite Statistical Test for Detecting Changes of Steady States. *AIChE Journal*, 46, p.1409-1418 , 1986.

NAYSMITH, M. R.; DOUGLAS, P. L. Review of Real Time Optimization in the Chemical Process Industries. Dev. Chem. Eng. Miner. Process. ,v. 3, p. 67-87, 2008.

NULL, H.R. Heat Pumps in Distillation. Chem. Eng. Progress 7(27):58-64, 1976.

OSORIO, D.; PPÉREZ-CORREA, R.; BELANCIC, A.; AGOSIN, E. Rigorous dynamic modeling and simulation of wine distillations. Food Control, v. 15, p. 515-521, 2004.

PANTELIDES, C.C.; GRISTIS D.; MORISON K.R.; SARGENT R.W.H. The mathematical modeling of transient systems using Differential-Algebraic equations, *Computers and Chemical Engineering*, v.12, p. 449-454, 1988.

PANTELIDES, C. C. The consistent initialization of differential-algebraic equations, *SIAM J. Sci. and Stat. Comput.*, 9(2), 213–231. (19 pages) , 1988.

PANTELIDES, C. C.; RENFRO, J. G. The online use of first-principles models in process operations: Review, current status and future needs. *Computers and Chemical Engineering*, V. 51, P. 136-148, 2013.

PETZOLD, L.R. Differential/Algebraic Equations are not ODE's, SIAM J. Sci. and Stat. Comput., 3(3), 367–384, 1982.

PETZOLD, L.R. A description of DASSL- a differential-algebraic equation system solver, in Scientific Computing (Edited by R.Stepleman). IMACS/North-Holland, New York, 1983.

PINTO, J.C.; BISCAIA Jr., E.C. Order reduction strategies for models of staged separation systems.CCE, v. 12, n. 3, p. 821-8318, 1988.

PONTON, J. W.; GAWTHROP, P. J. Systematic construction of dynamic models for phase equilibrium processes. In Computers & chemical engineering, 15(12), 803-808, 1991.

QUADRI, G.P. Use of heat pump in P-P splitter, part 1 : process design part 2 : process optimization, *Hydrocarbon Proc.*, v.60, p.119-126 and 147-151, 1981.

QUANG NHU HO; KYE SONG VOO; BYUNG GWON LEE; JONG SUNG LIM. Measurement of vapor – liquid equilibria for the binary mixture of propylene (R-1270) + propane (R-290). *Fluid Phase Equilibria*, v.245, p. 63-70, 2006.

RAO, D. P.; GOUTAMI, C. V.; JAIN, S. A direct method for incorporation of tray efficiency matrix in simulation of multicomponent separation processes. *Computers and Chemical Engineering*, V. 25, P. 1141-1152, 2001.

RUIZ, C.A.; CAMERON, I.T.; GANI, R.A generalized model for distillation columns -iii- study of start-up operations. *Computers and Chemical Engineering*, v. 12, p. 1-14, 1988.

SEQUEIRA, S. E.; GRAELLS, M.; PUIGJANER, L. Real-Time Evolution for On-line Optimization of Continuous Processes. Ind. Eng. Chem. Res, 41, 181, 2002.

SOARES, R. D. P.; SECCHI, A. R. EMSO: A new environment for modeling, simulation and optimization. *Computer Aided Chemical Engineering*, 14, 947-952, 2003.

SOARES, R. D. P.; SECCHI, A. R. Direct initialization and solution of high-index DAE systems. *Computer Aided Chemical Engineering*, v 20, p. 157-162, 2005.

SRIVISTAVA, R.K.; JOSEPH, B. Reduced-order models for separation columns - v. selections of collocation points. Computers and Chemical Engineering, v. 9, n. 6, p.601-613, 1985.

STEWART, W.E.; LEVIEN, K.L; MORARI, M. Simulation of fractionation by orthogonal collocation. Chemical Engineering Science, v. 40, n. 3, p. 409-421, 1985.

WANG, L.; LI, P.; WOZNY, G.; WANG,S. A start-up model for simulation of batch distillation starting from a cold state. Computers and Chemical Engineering, v. 27 , p.1485-1497, 2003.

WHITE, D. C. Online optimization: What, where and estimating ROI. Hydrocarbon Process. 1997.

YIP, W. S.; MARLIN, T. E. The effect of model fidelity on real-time optimization performance. *Computers and Chemical Engineering*, V. 28, P. 268-280, 2004

APPENDIX A: STEADY STATE MODEL IMPLEMENTED IN EQUATION-ORIENTED ENVIRONMENT

This appendix summarizes the equations used to model the Vapor Recompression Distillation (VRD) process. They correspond to the distillation column, expansion valves, reboiler, and compressor. The adiabatic distillation column was modeled as a collection of individual trays (numbered from top to bottom). The equations used to model a tray *j* of the column (Table A.1) comprise 3C + 6 equations, the component and energy balances (A1 and A2), efficiency (A3–A5), hydraulic (A6), summation (A7), and total flow (A8) equations, and the thermal and mechanical equilibrium of the streams leaving the stage (A9 and A10). *F*, *L*, and *V* denote the molar flow rates of the feed, the internal liquid and vapor streams respectively. *x* and *y* are the mole fractions of the liquid and vapor phases, h_V and h_L are the vapor and liquid molar enthalpies, and P_j and ΔP_j are the total pressure and pressure drop in tray *j* (*j*=1,...N), respectively.

Table A-1 - Summary of equations for modeling an adiabatic distillation tray

$$F_j^{V} y_{iF_j^{V}} + F_j^{L} x_{iF_j^{L}} + V_{j+1} y_{ij+1} + L_{j-1} x_{ij-1} - V_j y_{ij} - L_j x_{ij} = 0 \qquad (i = 1, \ldots, C) \tag{A1}$$

$$F_j^{V} h_{F_j^{V}}^{V} + F_j^{L} h_{F_j^{L}}^{L} + V_{j+1} h_{j+1}^{V} + L_{j-1} h_{j-1}^{L} - V_j h_j^{V} - L_j h_j^{L} = 0 \tag{A2}$$

$$E_j^{MV}(y_{ij}^{*} - y_{ij+1}) - (y_{ij} - y_{ij+1}) = 0 \qquad (i = 1, \ldots, C-1) \tag{A3}$$

$$K_{ij} x_{ij} - y_{ij}^{*} = 0 \qquad (i = 1, \ldots, C) \tag{A4}$$

$$1 - \sum_{i=1}^{C} y_{ij}^{*} = 0 \tag{A5}$$

$$P_{j-1} + \Delta P_j - P_j^{V} = 0 \tag{A6}$$

$$1 - \sum_{i=1}^{C} y_{ij} = 0 \tag{A7}$$

$$F_j^{V} + F_j^{L} + V_{j+1} + L_{j-1} - V_j - L_j = 0 \tag{A8}$$

$$T_j^{L} = T_j^{V} \tag{A9}$$

$$P_j^{L} = P_j^{V} \tag{A10}$$

The expansion valves are modeled as an adiabatic process in which the outlet stream is in vapor–liquid equilibrium. The 2C + 5 equations involved in the modeling of the expansion are shown in Table A.2; these equations come from the component and energy balances (A11 and A12), the phase equilibrium condition (A13–A15), the sum of composition, and the total mole balance.

Table A-2 - Summary of equations for modeling the expansion valve

$$Fx_{iF} - Vy_i - Lx_i = 0 \qquad (i = 1, ..., C) \tag{A11}$$

$$Fh_F - Vh^V - Lh^L = 0 \tag{A12}$$

$$y_i - K_i x_i = 0 \qquad (i = 1, ..., C) \tag{A13}$$

$$T^L - T^V = 0 \tag{A14}$$

$$p^L - p^V = 0 \tag{A15}$$

$$1 - \sum_{i=1}^{C} y_i = 0 \tag{A16}$$

$$F - V - L = 0 \tag{A17}$$

The equations used to model the total reboiler are shown in Table A.3. The cold side of the total reboiler, identified by subscript "*C*", is the liquid coming from the last tray of the distillation column while the hot side, denoted by subscript "*H*", is the overheated vapor coming from the compressor. The cold-side fluid leaves the reboiler as saturated vapor while the hot-side fluid can exit the reboiler as saturated or subcooled liquid. Equations A18–A20 correspond to the material balances (total and by component) and the dew point condition. Equations A21–A31 describe the hot side of the reboiler and include the material balances, the boiling-point condition which allows the calculation of the outlet temperature using a subcooling degree, ΔT_H^{sub}, and the energy balance around the reboiler. The equation set derived for the reboiler can be applied to the other heat exchange devices of the flow sheet.

Table A-3 - Summary of equations for modeling a total reboiler

$$F_{\mathrm{C}}^{\mathrm{in}} - F_{\mathrm{C}}^{\mathrm{out}} = 0 \tag{A18}$$

$$x_{i\mathrm{C}}^{\mathrm{in}} - y_{i\mathrm{C}}^{\mathrm{out}} = 0 \qquad (i = 1, \ldots, C) \tag{A19}$$

$$y_{i\mathrm{C}}^{\mathrm{out}} - K_{i\mathrm{C}}^{\mathrm{out}} x_{i\mathrm{C}}^{\mathrm{out}} = 0 \qquad (i = 1, \ldots, C) \tag{A20}$$

$$T_{\mathrm{C}}^{\mathrm{L,out}} - T_{\mathrm{C}}^{\mathrm{V,out}} = 0 \tag{A21}$$

$$P_{\mathrm{C}}^{\mathrm{L,out}} - P_{\mathrm{C}}^{\mathrm{V,out}} = 0 \tag{A22}$$

$$1 - \sum_{i=1}^{C} x_{i\mathrm{C}}^{\mathrm{out}} = 0 \tag{A23}$$

$$F_{\mathrm{H}}^{\mathrm{in}} - F_{\mathrm{H}}^{\mathrm{out}} = 0 \tag{A24}$$

$$y_{i\mathrm{H}}^{\mathrm{in}} - x_{i\mathrm{H}}^{\mathrm{out}} = 0 \qquad (i = 1, \ldots, C) \tag{A25}$$

$$y_{i\mathrm{H}}^{\mathrm{out}} - K_{i\mathrm{H}}^{\mathrm{out}} x_{i\mathrm{H}}^{\mathrm{out}} = 0 \qquad (i = 1, \ldots, C) \tag{A26}$$

$$T_{\mathrm{H}}^{\mathrm{L,eq}} - T_{\mathrm{H}}^{\mathrm{V,eq}} = 0 \tag{A27}$$

$$P_{\mathrm{H}}^{\mathrm{L,out}} - P_{\mathrm{H}}^{\mathrm{V,out}} = 0 \tag{A28}$$

$$1 - \sum_{i=1}^{C} y_{i\mathrm{H}}^{\mathrm{out}} = 0 \tag{A29}$$

$$T_{\mathrm{H}}^{\mathrm{out}} - T_{\mathrm{H}}^{\mathrm{L,eq}} + \Delta T_{\mathrm{H}}^{\mathrm{sub}} = 0 \tag{A30}$$

$$F_{\mathrm{C}}^{\mathrm{in}} h_{F_{\mathrm{C}}}^{\mathrm{L,in}} + F_{\mathrm{H}}^{\mathrm{in}} h_{F_{\mathrm{H}}}^{\mathrm{V,in}} - F_{\mathrm{C}}^{\mathrm{out}} h_{F_{\mathrm{C}}}^{\mathrm{V,out}} - F_{\mathrm{H}}^{\mathrm{out}} h_{F_{\mathrm{H}}}^{\mathrm{L,out}} = 0 \tag{A31}$$

The compressor was modeled using isentropic efficiency, η_{CP}, to find the outlet temperature of the compressor; the equations involved in the compressor model (Table A.4) are material balances (A32 and A33), the pressure change through the compressor (A34), the isentropic condition (A35), and the isentropic efficiency expression (A36).

Table A-4 - Summary of equations for modeling the isentropic compressor

$$F^{\mathrm{in}} - F^{\mathrm{in}} = 0 \tag{A32}$$

$$y_i^{\mathrm{in}} - y_i^{\mathrm{out}} = 0 \qquad (i = 1, \ldots, C) \tag{A33}$$

$$P^{\mathrm{out}} - P^{\mathrm{in}} - \Delta P_{\mathrm{CP}} = 0 \tag{A34}$$

$$s^{\mathrm{in}} - s^{\mathrm{out}} = 0 \tag{A35}$$

$$\eta_{\mathrm{CP}}(h^{\mathrm{out}} - h^{\mathrm{in}}) - (h^{\mathrm{isen}} - h^{\mathrm{in}}) = 0 \tag{A36}$$

APPENDIX B: DETAILS OF IMPLEMENTED STEADY STATE DETECTION APPROACH

i. Calculation of first and second derivations for, *d1* and *d2* for series of data.

ii. Determination of the threshold value for the first derivation, *Ts*, that is equal to the standard derivation of the first derivative.

$$Ts = \sigma_{d1} \quad \text{(B1)}$$

iii. Determination of the threshold value for the second derivative, T_w, which is equal to the medium of the second derivative of data series.

$$Tw = \overline{d2} \quad \text{(B2)}$$

iv. Choosing the threshold value *Tu*. This value can be estimated based on process knowledge or by equation (B4), where *λs* is an adjustment parameter chosen by the user,

$$Tu = 3.Ts.\ \lambda s \quad \text{(B3)}$$

v. Calculating γ, $\theta(t)$ and $\xi(\theta(t))$ (Equations B4 to B6):

$$\gamma = \begin{cases} 0 & if\ |d2| \leq Tw \\ \frac{|d2| - Tw}{2}.Tw & if\ Tw < |d2| < 3.Tw \\ 1 & if\ |d2| \geq 3.Tw \end{cases} \quad \text{(B4)}$$

$$\Theta(t) = |d1| + \gamma.|d2| \quad \text{(B5)}$$

$$\xi(\theta(t)) = \frac{1}{2}\left[\cos\left(\frac{\Theta(t) - Ts}{Tu - Ts}\right).\pi\right) + 1 \right] \quad \text{(B6)}$$

vii. Calculation of steady state index for each variable:

$$B_i(t) = \begin{cases} 0 & if\ \Theta(t) \geq Tu \\ \xi(\Theta(t)) & if\ Ts < \Theta(t) < Tu \\ 1 & if\ \Theta(t) \leq Ts \end{cases} \quad \text{(B7)}$$

viii. Calculation of multivariable index:

$B_m(t) = \prod_{i=1}^{N}[Bi(t)]^{1.5}$ (B8)

Considering all of the variables means that they have the same importance. The system will be at steady state if all of the variables are stationary at the instance of time that evaluation is being done or $B_M(t) = 1$.

APPENDIX C: IMPLEMENTED DYNAMIC RIGOROUS MODEL IN EQUATION-ORIENTED ENVIRONMENT

Table C-1 represents the equations used for dynamic modeling of the distillation trays (Figure C.1).

Figure C- 1 - Distillation tray

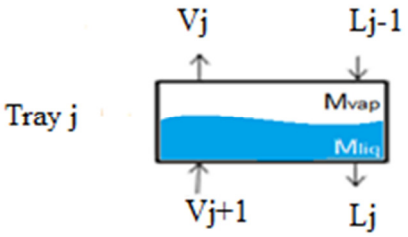


Table C-1 - Summary of equations for modeling adiabatic distillation tray

$$\frac{\partial M_j}{\partial t} = F_j^V + F_j^L + L_{j-1} + V_{j+1} - V_j - L \qquad j\ (j = 1,2,3...N) \qquad (C1)$$

$$M = M_{liq} + M_{vap} \qquad (C2)$$

$$\frac{\partial M_{i,j}}{\partial t} = F_j^V * y^V_{F^V_{i,j}} + F_j^L * x^L_{F^L_{i,j}} + L_{j-1} * x_{i,j-1} + V_{j+1} * y_{i,j+1} - V_j * y_{i,j} - L_j * x_{i,j}$$

$(i = 1,2,3...c)$ (C3)

$$M_{i,j} = M_{liq} * x_{i,j} + M_{vap} * y \qquad i,j(i = 1,2,3...c) \qquad (C4)$$

$$\frac{\partial E_j}{\partial t} = F_j^V * h^V_{F^V_{i,j}} + F_j^L * h^L_{F^L_{i,j}} + L_{j-1} * h^L_{j-1} + V_{j+1} * h^V_{j+1} - V_j * h^V_j - L_j * h^L_j \qquad (C5)$$

$E_j = M_{liq} * h_j^L + M_{vap} * h_j^V - P_j^V * Vtray$ (C6)

$x_{i,j} * K_{i,j} = y_{i,j}^*$ (i = 1,2,3...c) (C7)

$E_{i,j}^{MV} = \frac{y_{i,j} - y_{i,j+1}}{y_{i,j}^* - y_{i,j+1}}$ (i = 1,2,3...c-1) (C8)

$P_j^L = P_j^V$ (C9)

$T_j^L = T_j^V$ (C10)

$sum\ (x_{i,j}) = 1$ (C11)

$v_{liq} = f\ (T_j^L, P_j^L, x_{i,j})$ (C12)

$v_{vap} = f\ (T_j^V, P_j^V, y_{i,j})$ (C13)

$Vtray = M_{liq} * v_{liq} + M_{vap} * v_{vap}$ (C14)

$V_j = f\ (\Delta P, ...)$ (C15)

$L_j = f\ (L_{liq}, ...)$ (C16)

C1 is the overall molar balance considering liquid and vapor hold up. C3 is the component molar balance for each component and C11 is the summation of liquid molar components. C5 declares the energy balance for each tray. Non-ideal behavior of propylene-propane vapor mixture was declared in C8. C9 and C10 consist of mechanical and thermal equilibrium respectively. Specific volume of vapor, specific volume of liquid and total volume of tray were described in C12, C13 and C14 respectively. Outlet vapor and liquid flow rates are calculated by C15 and C16.

APPENDIX D: STEADY STAE MODEL CODE IN EMSO

```
using "types","streams";

FlowSheet ColumnFlowSheet

    PARAMETERS
    PP  as Plugin ( Brief         =  "Physical Properties" ,
                    Type          =  "PP"                  ,
                    Project       =  "ProjectPR_v2.vrtherm",
                    FlashTolerance =  1e-9 );

    NComp        as Integer    (Brief="Number of Components");
    DeltaPump    as pressure   (Brief="Pump Delta Pressure - B-9708 A/B");
    DPjr         as pressure   (Brief="Tray Delta Pressure, Rectification Zone");
    DPjs         as pressure   (Brief="Tray Delta Pressure, Stripping Zone");
    Eir          as efficiency (Brief="Individual Murphree Efficiency, Rectification");
    Eis          as efficiency (Brief="Individual Murphree Efficiency, Stripping");
    Er           as efficiency (Brief="Murphree Efficiency, Rectification");
    Es           as efficiency (Brief="Murphree Efficiency, Stripping");
    eta          as efficiency (Brief="Compressor Adiabatic Efficiency");
    Nt           as Integer    (Brief="Number of Distillation Trays");
    Nf           as Integer    (Brief="Number of Feed Tray");
    PcompOut     as pressure   (Brief="Compressor Outlet Pressure - C-9701");
    PdropB       as pressure   (Brief="Reboiler/Condenser Pressure Drop, tube side - P-9706 A/B");
    PdropC       as pressure   (Brief="Auxiliary Cooler Pressure Drop - P-9707 A/B");
    Pv           as pressure   (Brief="Pressure of Accumulator");
    P1           as pressure   (Brief="Column Top pressure");
    Qbl          as Real       (Brief="Reboiler/Condenser Heat Loss (minus)",final Unit = 'm^2*kg/s^3');
    Qcl          as Real       (Brief="Auxiliary Cooler Heat Loss (minus)"  ,final Unit = 'm^2*kg/s^3');
    TsubB        as temperature(Brief="Reboiler/Condenser Subcooling Degree, tube side, P-9706 A/B");
    TsubC        as temperature(Brief="Auxiliary Cooler Subcooling Degree, P-9707 A/B");
    Pi           as constant   (Brief="Pi Number",Default=3.14159265);
    SumpSpacing  as length     (Brief="Sump Spacing");
    TraySpacing1 as length     (Brief="Tray Spacing from 1:156");
    TraySpacing2 as length     (Brief="Tray Spacing from 157:197");
    TrayDiameter as length     (Brief="Tray Diameter");

    SET

    NComp               = PP.NumberOfComponents;
    FS.Mw               = PP.MolecularWeight();
    DeltaPump           = 2.62*'atm';
    DPjr                = 0.0049*'atm';
    DPjs                = 0.0049*'atm';
    Eir                 = 0.35;
    Eis                 = 1;
    Er                  = 1;
    Es                  = 1;
    eta                 = 1;
    Nt                  = 197;
    Nf                  = 157;
    PdropB              = 0*'atm';
    PdropC              = 0*'atm';
    PcompOut            = 15.3269*'atm';
    Pv                  = 13.0965*'atm';
    P1                  = 9.71034*'atm';
    Qbl                 = 0*'m^2*kg/s^3';
    Qcl                 = 0*'m^2*kg/s^3';

    DEVICES
    FS as CompleteFlowsheet;
```

```
    SPECIFY

    # COLUMN'S FEED

    FS.F.F = 1072.73*'kmol/h';
    FS.F.T = 345.35*'K';
    FS.F.P = 31*'atm';
    FS.F.z = [0.000121,0.73151,0.26726,0.000326,0.00013,0.0000093,0.0000093,0.0000093,0.00061,0.0000093];

    # Specifications (Two Degree of freedom)
    FS.Lreflux.F = 9319.58*'kmol/h';
    FS.PB.F = 299.92*'kmol/h';


    # Boil Up Vapor Fraction
    FS.vfB = 1;

    # Temperature of Streams 13 and 16
    FS.FBOut.T = 308.15*'K';
    FS.FCOut.T = 308.15*'K';



    Dynamic = false;
    GuessFile = "ColumnFlowSheet.rlt";

    NLASolver(
        File = "sundials",
        RelativeAccuracy        = 1E-3,
        AbsoluteAccuracy        = 1E-6,
        MaxIterations           = 150
    );
end
```

```
Model CompleteFlowsheet

    PARAMETERS

    outer DeltaPump    as pressure;
    outer DPjr         as pressure;
    outer DPjs         as pressure;
    outer Eir          as efficiency;
    outer Eis          as efficiency;
    outer Er           as efficiency;
    outer Es           as efficiency;
    outer eta          as efficiency;
    outer G            as Real;
    Mw(NComp)          as molweight;
    outer NComp        as Integer;
    outer Nt           as Integer;
    outer Nf           as Integer;
    outer PcompOut     as pressure;
    outer PdropB       as pressure;
    outer PdropC       as pressure;
    outer PP           as Plugin;
    outer Pv           as pressure;
    outer Pl           as pressure;
    outer Qbl          as Real;
    outer Qcl          as Real;
    outer TsubB        as temperature;
    outer TsubC        as temperature;
    outer Pi           as constant;
    outer TraySpacing1 as length;
    outer TraySpacing2 as length;
    outer TrayDiameter as length;
    outer SumpSpacing  as length;

    #Parameters for dynamic simulation
    #Column
    TrayArea                    as area(Brief = "Tray Area");
    TrayVt(Nt)                  as volume(Brief = "Tray Total Volume");

    #Accumulator
    AccumulatorL                as length(Brief = "Accumulator Length");
    AccumulatorR                as length(Brief = "Accumulator Radius");
    AccumulatorVt               as volume(Brief = "Accumulator Total Volume");
    SumpV                       as volume(Brief = "Accumulator Total Volume");

    VARIABLES

    AccumulatorAfilled as  area         (Brief="Occupied Area of Accumulator by the Liquid");
    AccumulatorLevel   as  length       (Brief="Level of Liquid Level in Accumulator");;
    AccumulatorLevelP  as  percent      (Brief="Level of Liquid Level in Accumulator - percentage");
    AccumulatorVfilled as  volume       (Brief="Occupied Volume of Accumulator by the Liquid");
    CompIn             as  vapour_stream(Brief="Compressor Intlet");
    CompOut            as  vapour_stream(Brief="Compressor Outlet");
    D                  as  liquid_stream(Brief="Distillate - Stream 4");
    D_mass             as  flow_mass    (Brief="Distillate Mass Flow Rate - Stream 4");
    D_mm               as  molweight    (Brief="Distillate Molar Weight - Stream 4");
    F                  as  streamPH     (Brief="Feed - Stream 1");
    FBIn               as  stream       (Brief="Reboiler/Condenser Inlet Stream - Stream 12, Tube Side");
    FBIn_mass          as  flow_mass    (Brief="Reboiler/Condenser Inlet Stream Mass Flow Rate - Stream 12, Tube Side");
    FBIn_mm            as  molweight    (Brief="Reboiler/Condenser Inlet Stream Molar Weight - Stream 12, Tube Side");
    FBOut              as  liquid_stream(Brief="Reboiler/Condenser Outlet Stream - Stream 13");
```

```
    FCIn            as  stream       (Brief="Auxiliary Cooler Inlet Stream - Stream 15");
    FCOut           as  liquid_stream(Brief="Auxiliary Cooler Outlet Stream - Stream 16");
    FCOut_mass      as  flow_mass    (Brief="Reboiler/Condenser Outlet Stream Mass Flow rate - Stream 16");
    FCOut_mm        as  molweight    (Brief="Reboiler/Condenser Outlet Stream Molar Weight - Stream 16");
    Fl              as  liquid_stream(Brief="Liquid Portion of the Feed Stream - Stream 1");
    Fv              as  vapour_stream(Brief="Vapor Porion of the Feed Stream - Stream 1");
    FV1VapourOut    as  vapour_stream(Brief="Outlet of Expansion Valve, Auxiliary Cooler");
    FV1LiquidOut    as  liquid_stream(Brief="Outlet of Expansion Valve, Auxiliary Cooler");
    FV2VapourOut    as  vapour_stream(Brief="Outlet of Expansion Valve, Reboiler/Condenser Cooler");
    FV2LiquidOut    as  liquid_stream(Brief="Outlet of Expansion Valve, Reboiler/Condenser Cooler");
    F_mass          as  flow_mass    (Brief="Feed Stream Mass Flow Rate - Stream 1");
    F_mm            as  molweight    (Brief="Feed Stream Molar Weight - Stream 1");
    LinB            as  stream       (Brief="Reboiler/Condenser Inlet Stream - Stream 6, Shell Side");
    LoutB           as  liquid_stream(Brief="Reboiler/Condenser Outlet Stream - Stream 7, Liquid Part, Shell Side");
    Lreflux         as  liquid_stream(Brief="Reflux - Stream 5");
    Lreflux_mass    as  flow_mass    (Brief="Reflux Mass Flow Rate - Stream 5");
    Lreflux_mm      as  molweight    (Brief="Reflux Molar weight - Stream 5");
    Lsump           as  liquid_stream(Brief="Sump Liquid Outlet Stream");
    Ltray(Nt)       as  liquid_stream(Brief="Liquid Internal Flows Through Column");
    MLA             as  mol          (Brief="Reboiler Liquid Molar Holdup");
    MLC(Nt)         as  mol          (Brief="Trays Molar Liquid Holdup");
    MLS             as  mol          (Brief="Sump Molar Vapour Holdup");
    MVA             as  mol          (Brief="Reboiler Vapor Molar Holdup");
    MVC(Nt)         as  mol          (Brief="Trays Molar vapour holdup on Trays");
    MVLiquidOut     as  liquid_stream(Brief="Expansion Valve to V-9709");
    MVS             as  mol          (Brief="Sump Molar vapour holdup on Trays");
    MVVapourOut     as vapour_stream (Brief="Expansion Valve to V-9709");
    PB              as  stream       (Brief="Bottom Product - Stream 3");
    PB_mass         as  flow_mass    (Brief="Bottom Product Mass Flow - Stream 3");
    PB_mm           as  molweight    (Brief="Bottom Product Molar Weight - Stream 3");
    Pj(Nt)          as  pressure     (Brief="Trays Pressure");
    Qbin            as  Real         (Brief="Reboiler/Condenser Heat Duty, Shell Side",final Unit = 'm^2*kg/s^3');
    Qbout           as  Real         (Brief="Reboiler/Condenser Heat Duty, Tube Side",final Unit = 'm^2*kg/s^3');
    Qc              as  Real         (Brief="Auxiliary Cooler Heat Duty",final Unit = 'm^2*kg/s^3');
    SC              as  fraction     (Brief="Split Range after Compressor; SC=FCIn.F/CompOut.F");
    SCb             as  fraction     (Brief="Split Range at Column Bottom; SCb=LinB.F/Ltray(Nt).F");
    SR              as  fraction     (Brief="Split Range after Accumulator = refluxo + produto; SR=D.F/TankL.F");
    SumpLevel       as  length       (Brief="Level of Liquid Level in Sump");
    SumpLevelP      as  percent      (Brief="Level of Liquid Level in Sump - percentage");
    SumpVfilled     as  volume       (Brief="Occupied Volume of Sump by the Liquid");
    TankL           as  liquid_stream(Brief="Accumulator Liquid Outlet - Stream 19");
    TankPump        as  liquid_stream(Brief="Accumulator Liquid Outlet after Pump, Stream 20");
    TankV           as  vapour_stream(Brief="Accumulator Vapor Outlet - Stream 8");
    Tiso            as  temperature  (Brief="Isentropic Efficiency");
    Vboiler         as  vapour_stream(Brief="Boil up Stream");
    vfB             as  fraction     (Brief="Vapor Fraction in Boil Up Stream");
    vLA             as  volume_mol   (Brief="Accumulator Liquid Molar Volume");
    vLC(Nt)         as  volume_mol   (Brief="Column Molar Specific Volume");
    vLS             as  volume_mol   (Brief="Accumulator Liquid Molar Specific Volume");
    VoutB           as  vapour_stream(Brief="Reboiler/Condenser Outlet Stream, shell side");
    Vsump           as  vapour_stream(Brief="Sump Vapour Outlet Stream");
    Vtank           as  vapour_stream(Brief="Recycle, V-9709");
    Vtray(Nt)       as  vapour_stream(Brief="Vapor Internal Flows Through Column");
    vVA             as  volume_mol   (Brief="Accumulator Vapour Molar Specific Volume");
    vVC(Nt)         as  volume_mol   (Brief="Column Molar Specific Volume");
    vVS             as  volume_mol   (Brief="Accumulator Liquid Molar Specific Volume");
    Win             as  Real         (Brief="Compressor Energy",final Unit = 'm^2*kg/s^3');
    yeq(NComp*Nt)   as  fraction     (Brief="Tray Equilibrium Composition");

```

```
196     SumpVfilled         as  volume       (Brief="Occupied Volume of Sump by the Liquid");
197     TankL               as  liquid_stream(Brief="Accumulator Liquid Outlet - Stream 19");
198     TankPump            as  liquid_stream(Brief="Accumulator Liquid Outlet after Pump, Stream 20");
199     TankV               as  vapour_stream(Brief="Accumulator Vapor Outlet - Stream 8");
200     Tiso                as  temperature  (Brief="Isentropic Efficiency");
201     Vboiler             as  vapour_stream(Brief="Boil up Stream");
202     vfB                 as  fraction     (Brief="Vapor Fraction in Boil Up Stream");
203     vLA                 as  volume_mol   (Brief="Accumulator Liquid Molar Volume");
204     vLC(Nt)             as  volume_mol   (Brief="Column Molar Specific Volume");
205     vLS                 as  volume_mol   (Brief="Accumulator Liquid Molar Specific Volume");
206     VoutB               as  vapour_stream(Brief="Reboiler/Condenser Outlet Stream, shell side");
207     Vsump               as  vapour_stream(Brief="Sump Vapour Outlet Stream");
208     Vtank               as  vapour_stream(Brief="Recycle, V-9709");
209     Vtray(Nt)           as  vapour_stream(Brief="Vapor Internal Flows Through Column");
210     vVA                 as  volume_mol   (Brief="Accumulator Vapour Molar Specific Volume");
211     vVC(Nt)             as  volume_mol   (Brief="Column Molar Specific Volume");
212     vVS                 as  volume_mol   (Brief="Accumulator Liquid Molar Specific Volume");
213     Win                 as  Real         (Brief="Compressor Energy",final Unit = 'm^2*kg/s^3');
214     yeq(NComp*Nt)       as  fraction     (Brief="Tray Equilibrium Composition");
215
216
217     EQUATIONS
218
219     # Units Conversion - Average Molecular Weight
220
221     D_mm            = sum(Mw*D.z);
222
223     FBIn_mm         = sum(Mw*FBIn.z);
224
225     FCOut_mm        = sum(Mw*FCOut.z);
226
227     F_mm            = sum(Mw*F.z);
228
229     Lreflux_mm      = sum(Mw*Lreflux.z);
230
231     PB_mm           = sum(Mw*PB.z);
232
233     # Mass Flow Rates
234
235     D_mass          = D.F*D_mm;
236
237     FBIn_mass       = FBIn.F*FBIn_mm;
238
239     FCOut_mass      = FCOut.F*FCOut_mm;
240
241     F_mass          = F.F*F_mm;
242
243     Lreflux_mass    = Lreflux.F*Lreflux_mm;
244
245     PB_mass         = PB.F*PB_mm;
246
247
248     #------------------------- DISTILLATION COLUMN: FEED CONDITION (ADIABATIC EXPANSION)-------------------------------
249
250     # Total Molar balance
251     F.F = Fl.F + Fv.F;
252
253     # Component molar balance
```

```
F.F*F.z = Fl.F*Fl.z + Fv.F*Fv.z;

# Energy balance
F.F*F.h = Fl.F*Fl.h + Fv.F*Fv.h;

# Thermodynamic Equilibrium
Fl.z*PP.LiquidFugacityCoefficient(Fl.T,Fl.P,Fl.z) = Fv.z*PP.VapourFugacityCoefficient(Fv.T,Fv.P,Fv.z);

#Thermal Equilibrium
Fl.T = Fv.T;

# Mechanical Equilibrium
(Fl.P) = 10.48*'atm';
(Fv.P) = (Fl.P);

#Summation
sum(Fv.z)=1;


#"DISTILLATION COLUMN: TRAYED SECTION"
#*
   |---Top Tray--------|
   |-------------------|
   |----Rectification--|
   |-------------------|
   |----Feed Vapor-----|
-->|----Feed Liquid----|
   |-------------------|
   |----Stripping------|
   |-------------------|
   |---Bottom tray-----|

*#

#"DISTILLATION COLUMN: TRAYED SECTION"
#------------------------------------------- COLUMN ---------------------------------------------------------------

#------------------------------------- TOP TRAY: TRAY 1 ----------------------------------------------------------

# Total Molar balance
Vtray(2).F + Lreflux.F = Vtray(1).F + Ltray(1).F;

# Component molar balance
Vtray(2).F*Vtray(2).z + Lreflux.F*Lreflux.z = Vtray(1).F*Vtray(1).z + Ltray(1).F*Ltray(1).z;

# Energy balance
1E-6*(Vtray(2).F*Vtray(2).h + Lreflux.F*Lreflux.h) = 1E-6*(Vtray(1).F*Vtray(1).h + Ltray(1).F*Ltray(1).h);

# Thermodynamic Equilibrium
yeq([1:NComp])*PP.VapourFugacityCoefficient(Ltray(1).T,Ltray(1).P,yeq([1:NComp])) = Ltray(1).z*PP.LiquidFugacityCoeff
Vtray(1).z([1:NComp-1])-Vtray(2).z([1:NComp-1])= Eir*(yeq([1:NComp-1])-Vtray(2).z([1:NComp-1]));

# Thermal Equilibrium
Vtray(1).T = Ltray(1).T;

# Mechanical Equilibrium
1E-4*(Pj(1))      = 1E-4*(P1);
1E-4*(Vtray(1).P) = 1E-4*(Pj(1));
1E-4*(Vtray(1).P) = 1E-4*(Ltray(1).P);
```

```
313
314     #Summation
315     sum(Vtray(1).z)=1;
316     sum(yeq([1:10]))=1;
317
318     #Liquid Specific Volume
319     vLC(1) = PP.LiquidVolume(Ltray(1).T, Ltray(1).P, Ltray(1).z);
320     # Vapour Specific Volume
321     vVC(1)= PP.VapourVolume(Vtray(1).T, Vtray(1).P, Vtray(1).z);
322     # Geometric Constraint
323     TrayVt(1) = MLC(1) * vLC(1) + MVC(1) * vVC(1);
324
325
326
327     #---------------------------------------- UPPER SECTION: TRAYS 2 TO Nf-2----------------------------------------
328
329     for stage in 2:Nf-2 do
330
331     # Total Molar balance
332     Vtray(stage+1).F + Ltray(stage-1).F = Vtray(stage).F + Ltray(stage).F;
333
334     # Component molar balance
335     Vtray(stage+1).F*Vtray(stage+1).z + Ltray(stage-1).F*Ltray(stage-1).z = Vtray(stage).F*Vtray(stage).z + Ltray(stage).
336
337     # Energy balance
338     1E-6*(Vtray(stage+1).F*Vtray(stage+1).h + Ltray(stage-1).F*Ltray(stage-1).h) = 1E-6*(Vtray(stage).F*Vtray(stage).h +
339
340     # Thermal Equilibrium
341     yeq([NComp*(stage-1)+1:NComp*stage])*PP.VapourFugacityCoefficient(Vtray(stage).T,Vtray(stage).P,yeq([NComp*(stage-1)+
342     Vtray(stage).z([1:NComp-1])-Vtray(stage+1).z([1:NComp-1]) = Eir*(yeq([NComp*(stage-1)+1:NComp*stage-1])-Vtray(stage+1
343
344     # Thermodynamic Equilibrium
345     Vtray(stage).T = Ltray(stage).T;
346
347     # Mechanical Equilibrium
348     1E-4*(Pj(stage))      = 1E-4*(Pj(stage-1) + DPjr);
349     1E-4*(Vtray(stage).P) = 1E-4*(Pj(stage));
350     1E-4*(Vtray(stage).P) = 1E-4*(Ltray(stage).P);
351
352     # Summation
353     sum(Vtray(stage).z)=1;
354     sum(yeq([NComp*(stage-1)+1:NComp*stage]))=1;
355
356     # Liquid Specific Volume
357     vLC(stage) = PP.LiquidVolume(Ltray(stage).T, Ltray(stage).P, Ltray(stage).z);
358     # Vapour Specific Volume
359     vVC(stage)= PP.VapourVolume(Vtray(stage).T, Vtray(stage).P, Vtray(stage).z);
360     # Geometric Constraint
361     TrayVt(stage) = MLC(stage) * vLC(stage) + MVC(stage) * vVC(stage);
362
363     end
364
365     #-----------------------------------------------------STAGE Nf-1-----------------------------------
366
367     # Total Molar balance
368     Vtray(Nf).F + Ltray(Nf-2).F + Fv.F = Vtray(Nf-1).F + Ltray(Nf-1).F;
369
370     # Component molar balance
371     Vtray(Nf).F*Vtray(Nf).z + Ltray(Nf-2).F*Ltray(Nf-2).z + Fv.F* Fv.z = Vtray(Nf-1).F*Vtray(Nf-1).z + Ltray(Nf-1).F*Ltra-
```

```
372
373     # Energy balance
374     1E-6*(Vtray(Nf).F*Vtray(Nf).h + Ltray(Nf-2).F*Ltray(Nf-2).h + Fv.F* Fv.h) = 1E-6*(Vtray(Nf-1).F*Vtray(Nf-1).h + Ltray
375
376     yeq([NComp*(Nf-2)+1:NComp*(Nf-1)])*PP.VapourFugacityCoefficient(Vtray(Nf-1).T,Vtray(Nf-1).P,yeq([NComp*(Nf-2)+1:NComp
377     Vtray(Nf-1).z([1:NComp-1])-Vtray(Nf).z([1:NComp-1])= Eir*(yeq([NComp*(Nf-2)+1:NComp*(Nf-1)-1])-Vtray(Nf).z([1:NComp-1
378
379     # Thermal Equilibrium
380     Vtray(Nf-1).T = Ltray(Nf-1).T;
381
382     # Mechanical Equilibrium
383     1E-4*(Pj(Nf-1)) = 1E-4*(Pj(Nf-2) + DPjr);
384     1E-4*(Vtray(Nf-1).P) = 1E-4*(Pj(Nf-1));
385     1E-4*(Vtray(Nf-1).P) = 1E-4*(Ltray(Nf-1).P);
386
387     # Summation
388     sum(Vtray(Nf-1).z)=1;
389     sum(yeq([NComp*(Nf-2)+1:NComp*(Nf-1)]))=1;
390
391     # Liquid Specific Volume
392     vLC(Nf-1) = PP.LiquidVolume(Ltray(Nf-1).T, Ltray(Nf-1).P, Ltray(Nf-1).z);
393     # Vapour Specific Volume
394     vVC(Nf-1)= PP.VapourVolume(Vtray(Nf-1).T, Vtray(Nf-1).P, Vtray(Nf-1).z);
395     # Geometric Constraint
396     TrayVt(Nf-1) = MLC(Nf-1) * vLC(Nf-1) + MVC(Nf-1) * vVC(Nf-1);
397
398     #-----------------------------------------------------Feed tray-----------------------------------------
399
400     # Total Molar balance
401     Vtray(Nf+1).F + Ltray(Nf-1).F + Fl.F = Vtray(Nf).F + Ltray(Nf).F;
402
403     # Component molar balance
404     Vtray(Nf+1).F*Vtray(Nf+1).z + Ltray(Nf-1).F*Ltray(Nf-1).z + Fl.F* Fl.z = Vtray(Nf).F*Vtray(Nf).z + Ltray(Nf).F*Ltray(
405
406     # Energy balance
407     1E-6*(Vtray(Nf+1).F*Vtray(Nf+1).h + Ltray(Nf-1).F*Ltray(Nf-1).h + Fl.F* Fl.h) = 1E-6*(Vtray(Nf).F*Vtray(Nf).h + Ltray
408
409     yeq([NComp*(Nf-1)+1:NComp*(Nf)])*PP.VapourFugacityCoefficient(Vtray(Nf).T,Vtray(Nf).P,yeq([NComp*(Nf-1)+1:NComp*(Nf)]
410     Vtray(Nf).z([1:NComp-1])-Vtray(Nf+1).z([1:NComp-1])= Eis*(yeq([NComp*(Nf-1)+1:NComp*(Nf)-1])-Vtray(Nf+1).z([1:NComp-1
411
412     # Thermal Equilibrium
413     Vtray(Nf).T = Ltray(Nf).T;
414
415     # Mechanical Equilibrium
416     1E-4*(Pj(Nf)) = 1E-4*(Pj(Nf-1) + DPjs);
417     1E-4*(Vtray(Nf).P) = 1E-4*(Pj(Nf));
418     1E-4*(Vtray(Nf).P) = 1E-4*(Ltray(Nf).P);
419
420     # Summation
421     sum(Vtray(Nf).z)=1;
422     sum(yeq([NComp*(Nf-1)+1:NComp*(Nf)]))=1;
423
424     # Liquid Specific Volume
425     vLC(Nf) = PP.LiquidVolume(Ltray(Nf).T, Ltray(Nf).P, Ltray(Nf).z);
426     # Vapour Specific Volume
427     vVC(Nf)= PP.VapourVolume(Vtray(Nf).T, Vtray(Nf).P, Vtray(Nf).z);
428     # Geometric Constraint
429     TrayVt(Nf) = MLC(Nf) * vLC(Nf) + MVC(Nf) * vVC(Nf);
```

```
430
431     #------------------------------------ LOWER SECTION: TRAYS Nf+1 TO Nt-1-------------------------------------
432
433     for stage in Nf+1:Nt-1 do
434
435     # Total Molar balance
436     Vtray(stage+1).F + Ltray(stage-1).F = Vtray(stage).F + Ltray(stage).F;
437
438     # Component molar balance
439     Vtray(stage+1).F*Vtray(stage+1).z + Ltray(stage-1).F*Ltray(stage-1).z = Vtray(stage).F*Vtray(stage).z + Ltray(stage).
440
441     # Energy balance
442     1E-6*(Vtray(stage+1).F*Vtray(stage+1).h + Ltray(stage-1).F*Ltray(stage-1).h) = 1E-6*(Vtray(stage).F*Vtray(stage).h +
443
444     yeq([NComp*(stage-1)+1:NComp*stage])*PP.VapourFugacityCoefficient(Vtray(stage).T,Vtray(stage).P,yeq([NComp*(stage-1)+
445     Vtray(stage).z([1:NComp-1])-Vtray(stage+1).z([1:NComp-1])= Eis*(yeq([NComp*(stage-1)+1:NComp*(stage)-1])-Vtray(stage+
446
447     # Thermal Equilibrium
448     Vtray(stage).T = Ltray(stage).T;
449
450     # Mechanical Equilibrium
451     1E-4*(Pj(stage)) = 1E-4*(Pj(stage-1) + DPjs);
452     1E-4*(Vtray(stage).P) = 1E-4*(Pj(stage));
453     1E-4*(Vtray(stage).P) = 1E-4*(Ltray(stage).P);
454
455     # Summation
456     sum(Vtray(stage).z)=1;
457     sum(yeq([NComp*(stage-1)+1:NComp*stage]))=1;
458
459     # Liquid Specific Volume
460     vLC(stage) = PP.LiquidVolume(Ltray(stage).T, Ltray(stage).P, Ltray(stage).z);
461     # Vapour Specific Volume
462     vVC(stage)= PP.VapourVolume(Vtray(stage).T, Vtray(stage).P, Vtray(stage).z);
463     # Geometric Constraint
464     TrayVt(stage) = MLC(stage) * vLC(stage) + MVC(stage) * vVC(stage);
465
466     end
467
468     #------------------------------------------------ LAST STAGE: TRAY Nt-----------------------------------------
469
470     # Total Molar balance
471     Vboiler.F + Ltray(Nt-1).F = Vtray(Nt).F + Ltray(Nt).F;
472
473     # Component molar balance
474     Vboiler.F*Vboiler.z + Ltray(Nt-1).F*Ltray(Nt-1).z = Vtray(Nt).F*Vtray(Nt).z + Ltray(Nt).F*Ltray(Nt).z;
475
476     # Energy balance
477     1E-6*(Vboiler.F*Vboiler.h + Ltray(Nt-1).F*Ltray(Nt-1).h) = 1E-6*(Vtray(Nt).F*Vtray(Nt).h + Ltray(Nt).F*Ltray(Nt).h);
478
479     # Thermodynamic Equilibrium
480     yeq([NComp*(Nt-1)+1:NComp*Nt])*PP.VapourFugacityCoefficient(Vtray(Nt).T,Vtray(Nt).P,yeq([NComp*(Nt-1)+1:NComp*Nt])) =
481     Vtray(Nt).z([1:NComp-1]) - VoutB.z([1:NComp-1]) = Eis*(yeq([NComp*(Nt-1)+1:NComp*(Nt)-1])-VoutB.z([1:NComp-1]));
482
483     # Thermal Equilibrium
484     Vtray(Nt).T = Ltray(Nt).T;
485
486     # Mechanical Equilibrium
487     1E-4*(Pj(Nt)) = 1E-4*(Pj(Nt-1) + DPjs);
488     1E-4*(Vtray(Nt).P) = 1E-4*(Pj(Nt));
489     1E-4*(Vtray(Nt).P) = 1E-4*(Ltray(Nt).P);
490
```

```
#------------------------------------------------ LAST STAGE: TRAY Nt----------------------------------------

# Total Molar balance
Vboiler.F + Ltray(Nt-1).F = Vtray(Nt).F + Ltray(Nt).F;

# Component molar balance
Vboiler.F*Vboiler.z + Ltray(Nt-1).F*Ltray(Nt-1).z = Vtray(Nt).F*Vtray(Nt).z + Ltray(Nt).F*Ltray(Nt).z;

# Energy balance
1E-6*(Vboiler.F*Vboiler.h + Ltray(Nt-1).F*Ltray(Nt-1).h) = 1E-6*(Vtray(Nt).F*Vtray(Nt).h + Ltray(Nt).F*Ltray(Nt).h);

# Thermodynamic Equilibrium
yeq([NComp*(Nt-1)+1:NComp*Nt])*PP.VapourFugacityCoefficient(Vtray(Nt).T,Vtray(Nt).P,yeq([NComp*(Nt-1)+1:NComp*Nt])) =
Vtray(Nt).z([1:NComp-1]) - VoutB.z([1:NComp-1]) = Eis*(yeq([NComp*(Nt-1)+1:NComp*(Nt)-1])-VoutB.z([1:NComp-1]));

# Thermal Equilibrium
Vtray(Nt).T = Ltray(Nt).T;

# Mechanical Equilibrium
1E-4*(Pj(Nt)) = 1E-4*(Pj(Nt-1) + DPjs);
1E-4*(Vtray(Nt).P) = 1E-4*(Pj(Nt));
1E-4*(Vtray(Nt).P) = 1E-4*(Ltray(Nt).P);

# Summation
sum(Vtray(Nt).z)=1;
sum(yeq([NComp*(Nt-1)+1:NComp*Nt]))=1;

# Liquid Specific Volume
vLC(Nt) = PP.LiquidVolume(Ltray(Nt).T, Ltray(Nt).P, Ltray(Nt).z);
# Vapour Specific Volume
vVC(Nt)= PP.VapourVolume(Vtray(Nt).T, Vtray(Nt).P, Vtray(Nt).z);
# Geometric Constraint
TrayVt(Nt) = MLC(Nt) * vLC(Nt) + MVC(Nt) * vVC(Nt);

#------------------------------------------------ SUMP ----------------------------------------

# Total Molar balance
LoutB.F + Ltray(Nt).F = Vsump.F + Lsump.F;

# Component molar balance
LoutB.F*LoutB.z + Ltray(Nt).F*Ltray(Nt).z = Vsump.F*Vsump.z + Lsump.F*Lsump.z;

# Energy balance
1E-6*(LoutB.F*LoutB.h + Ltray(Nt).F*Ltray(Nt).h) = 1E-6*(Vsump.F*Vsump.h + Lsump.F*Lsump.h);

# Thermodynamic Equilibrium
Vsump.z*PP.VapourFugacityCoefficient(Vsump.T,Vsump.P,Vsump.z) = Lsump.z*PP.LiquidFugacityCoefficient(Lsump.T,Lsump.P,

# Thermal Equilibrium
Vsump.T = Lsump.T;

# Mechanical Equilibrium
1E-4*(Vsump.P) = 1E-4*(Pj(Nt));
1E-4*(Vsump.P) = 1E-4*(Lsump.P);

# Summation
sum(Vsump.z)=1;
```

```
525
526     # Liquid Specific Volume
527     vLS = PP.LiquidVolume(Lsump.T, Lsump.P, Lsump.z);
528     # Vapour Specific Volume
529     vVS= PP.VapourVolume(Vsump.T, Vsump.P, Vsump.z);
530     # Geometric Constraint
531     SumpV = MLS * vLS + MVS * vVS;
532
533     SumpVfilled = MLS*vLS;
534     SumpVfilled = TrayArea*SumpLevel;
535
536     #"Tank Level in Percentage
537     SumpLevel = SumpSpacing*SumpLevelP/100;
538
539     #------------------------------------------------ BOTTOM ------------------------------------------------------
540
541     #"DISTILLATION COLUMN: BOTTOM FLOW SPLITTER"
542     PB.F   = SCb*Lsump.F;
543     LinB.F = (1-SCb)*Lsump.F;
544
545     PB.T   = Lsump.T;
546     LinB.T = Lsump.T;
547
548     1E-4*(PB.P)   = 1E-4*(Lsump.P);
549     1E-4*(LinB.P) = 1E-4*(Lsump.P);
550
551     PB.z   = Lsump.z;
552     LinB.z = Lsump.z;
553
554     1E-6*(PB.h)   = 1E-6*(Lsump.h);
555     1E-6*(LinB.h) = 1E-6*(Lsump.h);
556
557     PB.v   = Lsump.v;
558     LinB.v = Lsump.v;
559
560     #------------------------------------------- REBOILER/Condenser-------------------------------------------------
561
562     # Shell Side
563     vfB*LinB.F = VoutB.F;
564
565     (1-vfB)*LinB.F = LoutB.F;
566
567     LinB.F*LinB.z= VoutB.F*VoutB.z + LoutB.F*LoutB.z;
568
569     VoutB.z*PP.VapourFugacityCoefficient(VoutB.T,VoutB.P,VoutB.z) = LoutB.z*PP.LiquidFugacityCoefficient(LoutB.T,LoutB.P,
570
571     LoutB.T = VoutB.T;
572
573     1E-4*(LoutB.P) = 1E-4*(VoutB.P);
574     1E-4*(VoutB.P) = 1E-4*(LinB.P);
575
576     sum(LoutB.z)=1;
577
578     # Tube Side
579     FBIn.F = FBOut.F;
580
581     FBIn.z = FBOut.z;
582
```

```
1E-4*(FBIn.P) = 1E-4*(FBOut.P + PdropB);

# Energy Balance in Reboiler/Condenser

1E-6*Qbout = 1E-6*(FBIn.F*FBIn.h - FBOut.F*FBOut.h + Qbl);

1E-6*Qbin = 1E-6*(LinB.F*LinB.h - VoutB.F*VoutB.h - LoutB.F*LoutB.h);

1E-6*(6*diff(Qbin) + Qbin/'s') = 1E-6*(-1/'s'*Qbout);

#----------------------------------------MIXER in the Column Bottom-------------------------------------------------

Vboiler.F = VoutB.F + Vsump.F;

Vboiler.F*Vboiler.z = VoutB.F*VoutB.z + Vsump.F*Vsump.z;

1E-6*(Vboiler.F*Vboiler.h) = 1E-6*(VoutB.F*VoutB.h + Vsump.F*Vsump.h);

1E-4*(Vboiler.P) = 1E-4*(VoutB.P);

#------------------------------------------------ TOP ------------------------------------------------------------

#--------------------MIXER COMPRESSOR: ADIABATIC MIXING OF THE STREAMS ENTERING THE COMPRESSOR-----------------------

CompIn.F = Vtank.F + Vtray(1).F;

CompIn.F*CompIn.z = Vtank.F*Vtank.z + Vtray(1).F*Vtray(1).z;

1E-6*(CompIn.F*CompIn.h) = 1E-6*(Vtank.F*Vtank.h + Vtray(1).F*Vtray(1).h);

1E-4*(CompIn.P) = 1E-4*(Vtray(1).P);

#------------------------------- COMPRESSOR: CENTRIFUGAL USING ADIABATIC EFFICIENCY-------------------------------
CompIn.F=CompOut.F;

CompIn.z=CompOut.z;

1E-4*(CompOut.P) = 1E-4*(PcompOut);

PP.VapourEntropy(CompIn.T,CompIn.P,CompIn.z)=PP.VapourEntropy(Tiso,CompOut.P,CompOut.z);

1E-6*(eta*(CompOut.h-CompIn.h)) = 1E-6*(PP.VapourEnthalpy(Tiso,CompOut.P,CompOut.z)-CompIn.h);

Win = 1E-6*(CompOut.F*CompOut.h-CompIn.F*CompIn.h);

# ---------------------------------------- COMPRESSOR: FLOW SPLITTER ------------------------------------------
FCIn.F=SC*CompOut.F;
FBIn.F=(1-SC)*CompOut.F;

FCIn.T=CompOut.T;
FBIn.T=CompOut.T;

1E-4*(FCIn.P) = 1E-4*(CompOut.P);
1E-4*(FBIn.P) = 1E-4*(CompOut.P);

FCIn.z=CompOut.z;
FBIn.z=CompOut.z;

1E-6*(FCIn.h) = 1E-6*(CompOut.h);
1E-6*(FBIn.h) = 1E-6*(CompOut.h);

FCIn.v= CompOut.v;
FBIn.v= CompOut.v;
```

```
#------------------------------------------------- COOLER-------------------------------------------------------------
FCIn.F = FCOut.F;
FCIn.z = FCOut.z;

1E-4*(FCIn.P) = 1E-4*(FCOut.P + PdropC);

1E-6*(Qc + Qcl + FCIn.F*FCIn.h) = 1E-6*(FCOut.F*FCOut.h);

#------------------------------------------------ Expansion Valves-------------------------------------------------

    #"VALVE 1: ADIABATIC EXPANSION VALVE - After Cooler

    # Total Molar balance
    FCOut.F = FV1LiquidOut.F + FV1VapourOut.F;

    # Component Molar Balance
    FCOut.z*FCOut.F = FV1VapourOut.z*FV1VapourOut.F + FV1LiquidOut.z*FV1LiquidOut.F;

    # Energy Balance
    1E-6*(FCOut.h*FCOut.F) = 1E-6*(FV1VapourOut.h*FV1VapourOut.F + FV1LiquidOut.h*FV1LiquidOut.F);


    # Thermodynamic Equilibrium
    FV1VapourOut.z*PP.VapourFugacityCoefficient(FV1VapourOut.T,FV1VapourOut.P,FV1VapourOut.z) = FV1LiquidOut.z*PP.Liq

    # Thermal Equilibrium
    FV1VapourOut.T = FV1LiquidOut.T;

    # Mechanical Equilibrium
    1E-4*(FV1VapourOut.P) = 1E-4*(Pv);
    1E-4*(FV1VapourOut.P) = 1E-4*(FV1LiquidOut.P);

    # Summation
    sum(FV1VapourOut.z)=1;

    #"VALVE 2: ADIABATIC EXPANSION VALVE - After Reboiler/Condenser

    # Total Molar balance
    FBOut.F = FV2LiquidOut.F + FV2VapourOut.F;

    # Component Molar Balance
    FBOut.z*FBOut.F = FV2VapourOut.z*FV2VapourOut.F + FV2LiquidOut.z*FV2LiquidOut.F;

    # Energy Balance
    1E-6*(FBOut.h*FBOut.F) = 1E-6*(FV2VapourOut.h*FV2VapourOut.F + FV2LiquidOut.h*FV2LiquidOut.F);

    # Thermodynamic Equilibrium
    FV2VapourOut.z*PP.VapourFugacityCoefficient(FV2VapourOut.T,FV2VapourOut.P,FV2VapourOut.z) = FV2LiquidOut.z*PP.Liq

    # Thermal Equilibrium
    FV2VapourOut.T = FV2LiquidOut.T;

    # Mechanical Equilibrium
    1E-4*(FV2VapourOut.P) = 1E-4*(Pv);

    1E-4*(FV2VapourOut.P) = 1E-4*(FV2LiquidOut.P);

    # Summation
    sum(FV2VapourOut.z)=1;

    # VALVE 3: ADIABATIC EXPANSION VALVE - Recycle Stream

    Vtank.F = TankV.F;

    1E-6*(Vtank.h) = 1E-6*(TankV.h);
```

```
712
713         1E-4*(Vtank.P) = 1E-4*(P1);
714
715         Vtank.z = TankV.z;
716     #-------------------------------------MIXER: FROM VALVES 1 2 AND 3 TO TANK ---------------------------------------
717
718
719         # Total Molar balance
720         FV1LiquidOut.F + FV1VapourOut.F + FV2LiquidOut.F + FV2VapourOut.F = MVLiquidOut.F + MVVapourOut.F;
721
722         # Component Molar Balance
723         FV1LiquidOut.F*FV1LiquidOut.z + FV1VapourOut.F*FV1VapourOut.z + FV2LiquidOut.F*FV2LiquidOut.z + FV2VapourOut.F*FV
724
725         #"Energy Balance"
726         1E-6*(FV1LiquidOut.F*FV1LiquidOut.h + FV1VapourOut.F*FV1VapourOut.h + FV2LiquidOut.F*FV2LiquidOut.h + FV2VapourOu
727
728         # Thermodynamic Equilibrium
729         MVVapourOut.z*PP.VapourFugacityCoefficient(MVVapourOut.T,MVVapourOut.P,MVVapourOut.z) = MVLiquidOut.z*PP.LiquidFu
730
731         #Thermal Equilibrium
732         MVVapourOut.T = MVLiquidOut.T;
733
734         #Mechanical Equilibrium
735         1E-4*(MVVapourOut.P) = 1E-4*(min([FV1LiquidOut.P,FV1VapourOut.P,FV2LiquidOut.P,FV2VapourOut.P]));
736         1E-4*(MVVapourOut.P) = 1E-4*(MVLiquidOut.P);
737
738         # Summation
739         sum(MVVapourOut.z)=1;
740
741     #------------------------------------------------ Accumulator-----------------------------------------------------
742         TankV.F= MVVapourOut.F;
743         TankV.T= MVVapourOut.T;
744         1E-4*(TankV.P) = 1E-4*(MVVapourOut.P);
745         TankV.z= MVVapourOut.z;
746
747         TankL.F=MVLiquidOut.F;
748         TankL.T=MVLiquidOut.T;
749         1E-4*(TankL.P) = 1E-4*(MVLiquidOut.P);
750         TankL.z=MVLiquidOut.z;
751
752         #"Liquid Volume"
753         vLA = PP.LiquidVolume(TankL.T, TankL.P, TankL.z);
754         #"Vapour Volume"
755         vVA= PP.VapourVolume(TankV.T, TankV.P, TankV.z);
756
757         #"Geometry Constraint"
758         AccumulatorVt = MLA*vLA + MVA*vVA;
759         AccumulatorVfilled = MLA*vLA;
760         AccumulatorVfilled = AccumulatorAfilled*AccumulatorL;
761
762         #"Tank Level in Percentage
763         AccumulatorLevel = AccumulatorR*2*AccumulatorLevelP/100;
764
765         #"Vessel Cross Section Area"
766         AccumulatorAfilled = (Pi/2)*AccumulatorR^2 - AccumulatorR^2*asin((AccumulatorR - AccumulatorLevel)/AccumulatorR)
767                            + (AccumulatorLevel - AccumulatorR)*sqrt(AccumulatorR^2 - (AccumulatorLevel - AccumulatorR)^2);
768
769
```

```
    # ----------------------------------------PUMP AFTER ACCUMULATOR ------------------------------------------------
    TankL.F = TankPump.F;
    TankL.z = TankPump.z;
    TankL.T = TankPump.T;
    1E-4*(TankPump.P) = 1E-4*(TankL.P + DeltaPump);

    # ----------------------------------------SPLITTER after Accumulator---------------------------------------------
    D.F= SR*TankPump.F;
    Lreflux.F= (1-SR)*TankPump.F;

    D.T= TankPump.T;
    Lreflux.T= TankPump.T;

    1E-4*(D.P)= 1E-4*(TankPump.P);
    1E-4*(Lreflux.P) = 1E-4*(TankPump.P);

    D.z= TankPump.z;
    Lreflux.z= TankPump.z;

end
```

APPENDIX E: DYNAMIC MODEL CODE IN EMSO

```
using "types","streams";

FlowSheet ColumnFlowSheet
    PARAMETERS
    PP  as Plugin(Brief="Physical Properties",
                  Type="PP",
                  Project = "ProjectPR_v2.vrtherm",
                  FlashTolerance = 1e-9);
    NComp           as Integer      (Brief="Number of Components");
    DeltaPump       as pressure     (Brief="Pump Delta Pressure - B-9708 A/B");
    DPjr            as pressure     (Brief="Tray Delta Pressure, Rectification Zone");
    DPjs            as pressure     (Brief="Tray Delta Pressure, Stripping Zone");
    Eir             as efficiency   (Brief="Individual Murphree Efficiency, Rectification");
    Eis             as efficiency   (Brief="Individual Murphree Efficiency, Stripping");
    Er              as efficiency   (Brief="Murphree Efficiency, Rectification");
    Es              as efficiency   (Brief="Murphree Efficiency, Stripping");
    eta             as efficiency   (Brief="Compressor Adiabatic Efficiency");
    Nt              as Integer      (Brief="Number of Distillation Trays");
    Nf              as Integer      (Brief="Number of Feed Tray");
    PcompOut        as pressure     (Brief="Compressor Outlet Pressure - C-9701");
    PdropB          as pressure     (Brief="Reboiler/Condenser Pressure Drop, tube side - P-9706 A/B");
    PdropC          as pressure     (Brief="Auxiliary Cooler Pressure Drop - P-9707 A/B");
    Pv              as pressure     (Brief="Pressure of Accumulator");
    P1              as pressure     (Brief="Column Top pressure");
    Qbl             as Real         (Brief="Reboiler/Condenser Heat Loss (minus)",final Unit = 'm^2*kg/s^3');
    Qcl             as Real         (Brief="Auxiliary Cooler Heat Loss (minus)",final Unit = 'm^2*kg/s^3');
    TsubB           as temperature  (Brief="Reboiler/Condenser Subcooling Degree, tube side, P-9706 A/B");
    TsubC           as temperature  (Brief="Auxiliary Cooler Subcooling Degree, P-9707 A/B");
    Pi              as constant     (Brief="Pi Number",Default=3.14159265);
    SumpSpacing     as length       (Brief="Sump Spacing");
    TraySpacing1    as length       (Brief="Tray Spacing from 1:156");
    TraySpacing2    as length       (Brief="Tray Spacing from 157:197");
    TrayDiameter    as length       (Brief="Tray Diameter");
    Kcolp           as Real         (Brief="Gain of Column - Pressure Equation");
    Kcolh           as Real         (Brief="Gain of Column - Heigth Equation");
    Kacump          as Real         (Brief="Gain of Accumulator - Pressure Equation");
    Kacumh          as Real         (Brief="Gain of Accumulator - Heigth Equation");
    Ksumpp          as Real         (Brief="Gain of Sump - Pressure Equation");
    Ksumph          as Real         (Brief="Gain of Sump - Heigth Equation");

    NComp               = PP.NumberOfComponents;
    FS.Mw               = PP.MolecularWeight();
    DeltaPump           = 2.62*'atm';
    DPjr                = 0.0049*'atm';
    DPjs                = 0.0049*'atm';
    Eir                 = 0.35;
    Eis                 = 1;
    Er                  = 1;
    Es                  = 1;
    eta                 = 1;
    Nt                  = 197;
    Nf                  = 157;
    PdropB              = 0*'atm';
    PdropC              = 0*'atm';
    PcompOut            = 15.3269*'atm';
    Pv                  = 13.0965*'atm';
    P1                  = 9.71034*'atm';
    Qbl                 = 0*'m^2*kg/s^3';
    Qcl                 = 0*'m^2*kg/s^3';
```

```
61
62      # Trays Geometry:
63      TraySpacing1       = 0.45*'m';
64      TraySpacing2       = 0.475*'m';
65      TrayDiameter       = 4.9*'m';
66      SumpSpacing        = 10.5*'m';
67      FS.TrayArea        = 0.25*Pi*(TrayDiameter^2);
68      FS.TrayVt(1:156)   = FS.TrayArea*TraySpacing1;
69      FS.TrayVt(157:197) = FS.TrayArea*TraySpacing2;
70
71
72      # Accumulator Geometry:
73      FS.AccumulatorR    = 1.65*'m';
74      FS.AccumulatorL    = 9.5*'m';
75      FS.AccumulatorVt   = FS.AccumulatorR^2*Pi*FS.AccumulatorL;
76
77      # Sump Geometry:
78      FS.SumpV           = FS.TrayArea*SumpSpacing;
79      Kcolp  = 1e5; Kcolh = 1e7;
80      Kacump = 1e5; Kacumh = 1e7;
81      Ksumpp = 1e-9;
82      Ksumph = 1e5;
83      # Accumulator Controller Prameters
84      ControlTop.Kp      = 1.5;
85      ControlTop.Ki      = 900*'s';
86      ControlTop.Bias    = 32.546;
87
88      # Range of Accumulator Level Variations
89      ControlTop.MinInput = 5;
90      ControlTop.MaxInput = 95;
91
92      # Range of Distillate flow Rate Variations - Stream 4
93      ControlTop.MinOutput = 0;
94      ControlTop.MaxOutput = 64;
95
96
97      # Sump Controller Parametrs
98      ControlBottom.Kp     = 0.8;
99      ControlBottom.Ki     = 10000*'s';
100     ControlBottom.Bias   = 22.4935;
101
102     # Range of Sump Level Variations
103     ControlBottom.MinInput = 5;
104     ControlBottom.MaxInput = 95;
105
106     # Range of Bottom Product Flow Rate Variations - Stream 3
107     ControlBottom.MinOutput = 0;
108     ControlBottom.MaxOutput = 45;
109
110
111     DEVICES
112
113     FS            as CompleteFlowsheet;
114     Feed          as InputModel;
115     ControlTop    as PI_Controller;
116     ControlBottom as PI_Controller;
117
```

```
SET

FS.LtraySS          = [8012.60000000000,8013.38000000000,8014.16000000000,8014.94000000000,8015.72000000000,8016.49000
FS.VtraySS          = [9267.99000000000,8608.02000000000,8608.80000000000,8609.58000000000,8610.36000000000,8611.14000

FS.VsumpSS          = 7.751204709000000e-08*'kmol/h';
FS.LsumpSS          = 8952.114556000000*'kmol/h';

FS.TankVSS          = 0*'kmol/h';
FS.TankLSS          = 9267.991512000000*'kmol/h';

FS.PSS              = P1;
FS.VsumpPSS         = 10.68*'atm';
FS.TankVPSS         = Pv;
FS.PBSS             = 255.0020883000000*'kmol/h';

# HoldUps Values of Trays, Accumulator and Sump
FS.MLASS            = 450*'kmol';
FS.MLCSS            = 1*'kmol';
FS.MLSSS            = 100*'kmol';

SPECIFY

#COLUMN'S FEED

FS.F.F = 1072.73 *'kmol/h';
FS.F.T = 345.35*'K';
FS.F.P = 10.5*'atm';
FS.F.z = [0.000121,0.73151,0.26726,0.000326,0.00013,0.0000093,0.0000093,0.0000093,0.00061,0.0000093];

# Specifications (Two Degree of freedom)
FS.Lreflux.F = 9319.58*'kmol/h';# + Feed.F(1)*'kmol/h'
FS.FBIn.F    = 9175.5*'kmol/h'+ Feed.F(1)*'kmol/h';


# Boil Up Vapor Fraction
FS.vfB = 1;


# Temperature of Streams 13 and 16
FS.FBOut.T = 308.15*'K';
FS.FCOut.T = 308.15*'K';


# Accumulator Controller Specifications
ControlTop.SetPoint =   50;
ControlTop.Input    =   FS.AccumulatorLevelP;
ControlTop.Output   =   FS.D_mass/'t/h';



# Sump Controller Specifications
ControlBottom.SetPoint = 50;
ControlBottom.Input    = FS.SumpLevelP;
ControlBottom.Output   = FS.PB_mass/'t/h';
```

```
175
176     INITIAL
177
178
179     # Column
180     diff(FS.MLC + FS.MVC) = 0*'mol/s';
181     diff(FS.MLC*transp(FS.Ltray.z) + FS.MVC*transp(FS.Vtray.z)) = 0*'mol/s';
182     diff(FS.MLC*FS.Ltray.h + FS.MVC*FS.Vtray.h - FS.Vtray.P*FS.TrayVt) = 0*'m^2*kg/s^3';
183
184     #Reboiler/Condenser
185     diff(FS.Qbin) = 0*'m^2*kg/s^4';
186
187     #Accumulator
188     diff(FS.MLA + FS.MVA) = 0*'mol/s';
189     diff(FS.MLA*FS.TankL.z + FS.MVA*FS.TankV.z) = 0*'mol/s';
190     diff(FS.MLA*FS.TankL.h + FS.MVA*FS.TankV.h - (FS.AccumulatorVt - FS.AccumulatorVfilled)*FS.TankL.P) = 0*'m^2*kg/s^3';
191
192     #initial Sump
193     diff(FS.MLS + FS.MVS) = 0*'mol/s';
194     diff(FS.MLS*FS.Lsump.z + FS.MVS*FS.Vsump.z) = 0*'mol/s';
195     diff(FS.MLS*FS.Lsump.h + FS.MVS*FS.Vsump.h) = 0*'m^2*kg/s^3';
196
197
198     OPTIONS
199
200     Dynamic       = true;
201     GuessFile     = "ColumnFlowSheet.rlt";
202     TimeStep      = 1;
203     TimeEnd       = 3600;
204     TimeUnit      = 'min';
205     NLASolver(File="sundials",
206         RelativeAccuracy = 1e-3,AbsoluteAccuracy = 1e-6, MaxIterations = 1000);
207     DAESolver(File="sundials",
208         RelativeAccuracy = 1e-3,AbsoluteAccuracy = 1e-6, EventAccuracy = 1e-2);
209     SparseAlgebra = true;
210
211
212 end
213
214 Model CompleteFlowsheet
215
216     PARAMETERS
217
218     outer DeltaPump    as pressure;
219     outer DPjr         as pressure;
220     outer DPjs         as pressure;
221     outer Eir          as efficiency;
222     outer Eis          as efficiency;
223     outer Er           as efficiency;
224     outer Es           as efficiency;
225     outer eta          as efficiency;
226     outer G            as Real;
227     Mw(NComp)          as molweight;
228     outer NComp        as Integer;
```

```
    outer Nt           as Integer;
    outer Nf           as Integer;
    outer PcompOut     as pressure;
    outer PdropB       as pressure;
    outer PdropC       as pressure;
    outer PP           as Plugin;
    outer Pv           as pressure;
    outer Pl           as pressure;
    outer Qbl          as Real;
    outer Qcl          as Real;
    outer TsubB        as temperature;
    outer TsubC        as temperature;
    outer Pi           as constant;
    outer TraySpacing1 as length;
    outer TraySpacing2 as length;
    outer TrayDiameter as length;
    outer SumpSpacing  as length;



    # Parameters of Dynamic Simulation

    # Column:
    LtraySS(Nt)        as flow_mol     (Brief = "Steady State Liquid flows");
    MLCSS(Nt)          as mol          (Brief = "Steady State Tray Molar Liquid Holdup");
    PSS                as pressure     (Brief = "Steady state Tray Pressures");
    TrayArea           as area         (Brief = "Tray Area");
    TrayVt(Nt)         as volume       (Brief = "Tray Total Volume");
    VtraySS(Nt)        as flow_mol     (Brief = "Steady State Vapor Flow");

    # Accumulator:
    AccumulatorVt      as volume       (Brief = "Accumulator Total Volume");
    AccumulatorL       as length       (Brief = "Accumulator Length");
    AccumulatorR       as length       (Brief = "Accumulator Radius");
    MLASS              as mol          (Brief = "Steady State Accumulator Holdup");
    TankLSS            as flow_mol     (Brief = "Steady State Accumulator Liquid Flow");
    TankVPSS           as pressure     (Brief = "Steady State Accumulator Pressure");
    TankVSS            as flow_mol     (Brief = "Steady State Accumulator Vapor Flow");

    # Sump:
    SumpV              as volume       (Brief = "Accumulator Total Volume");
    MLSSS              as mol          (Brief = "Steady State Accumulator Holdup");
    LsumpSS            as flow_mol     (Brief = "Steady State Accumulator Liquid Flow");
    VsumpPSS           as pressure     (Brief = "Steady State Accumulator Pressure");
    VsumpSS            as flow_mol     (Brief = "Steady State Accumulator Vapor Flow");
    PBSS               as flow_mol     (Brief = "Steady State bottom product");


    VARIABLES

    AccumulatorAfilled as  area        (Brief="Occupied Area of Accumulator by the Liquid");
```

```
AccumulatorLevel   as  length       (Brief="Level of Liquid Level in Accumulator");;
AccumulatorLevelP  as  percent      (Brief="Level of Liquid Level in Accumulator - percentage");
AccumulatorVfilled as  volume       (Brief="Occupied Volume of Accumulator by the Liquid");
CompIn             as  vapour_stream(Brief="Compressor Intlet");
CompOut            as  vapour_stream(Brief="Compressor Outlet");
D                  as  liquid_stream(Brief="Distillate - Stream 4");
D_mass             as  flow_mass    (Brief="Distillate Mass Flow Rate - Stream 4");
D_mm               as  molweight    (Brief="Distillate Molar Weight - Stream 4");
F                  as  streamPH     (Brief="Feed - Stream 1");
FBIn               as  stream       (Brief="Reboiler/Condenser Inlet Stream - Stream 12, Tube Side");
FBIn_mass          as  flow_mass    (Brief="Reboiler/Condenser Inlet Stream Mass Flow Rate - Stream 12, Tube Side");
FBIn_mm            as  molweight    (Brief="Reboiler/Condenser Inlet Stream Molar Weight - Stream 12, Tube Side");
FBOut              as  liquid_stream(Brief="Reboiler/Condenser Outlet Stream - Stream 13");
FCIn               as  stream       (Brief="Auxiliary Cooler Inlet Stream - Stream 15");
FCOut              as  liquid_stream(Brief="Auxiliary Cooler Outlet Stream - Stream 16");
FCOut_mass         as  flow_mass    (Brief="Reboiler/Condenser Outlet Stream Mass Flow rate - Stream 16");
FCOut_mm           as  molweight    (Brief="Reboiler/Condenser Outlet Stream Molar Weight - Stream 16");
Fl                 as  liquid_stream(Brief="Liquid Portion of the Feed Stream - Stream 1");
Fv                 as  vapour_stream(Brief="Vapor Porion of the Feed Stream - Stream 1");
F_mass             as  flow_mass    (Brief="Feed Stream Mass Flow Rate - Stream 1");
F_mm               as  molweight    (Brief="Feed Stream Molar Weight - Stream 1");
LinB               as  stream       (Brief="Reboiler/Condenser Inlet Stream - Stream 6, Shell Side");
LoutB              as  liquid_stream(Brief="Reboiler/Condenser Outlet Stream - Stream 7, Liquid Part, Shell Side");
Lreflux            as  liquid_stream(Brief="Reflux - Stream 5");
Lreflux_mass       as  flow_mass    (Brief="Reflux Mass Flow Rate - Stream 5");
Lreflux_mm         as  molweight    (Brief="Reflux Molar weight - Stream 5");
Lsump              as  liquid_stream(Brief="Sump Liquid Outlet Stream");
Ltray(Nt)          as  liquid_stream(Brief="Liquid Internal Flows Through Column");
MLA                as  mol          (Brief="Reboiler Liquid Molar Holdup");
MLC(Nt)            as  mol          (Brief="Trays Molar Liquid Holdup");
MLS                as  mol          (Brief="Sump Molar Vapour Holdup");
MVA                as  mol          (Brief="Reboiler Vapor Molar Holdup");
MVC(Nt)            as  mol          (Brief="Trays Molar vapour holdup on Trays");
MVS                as  mol          (Brief="Sump Molar vapour holdup on Trays");
PB                 as  stream       (Brief="Bottom Product - Stream 3");
PB_mass            as  flow_mass    (Brief="Bottom Product Mass Flow - Stream 3");
PB_mm              as  molweight    (Brief="Bottom Product Molar Weight - Stream 3");
Qbin               as  Real         (Brief="Reboiler/Condenser Heat Duty, Shell Side",final Unit = 'm^2*kg/s^3');
Qbout              as  Real         (Brief="Reboiler/Condenser Heat Duty, Tube Side",final Unit = 'm^2*kg/s^3');
Qc                 as  Real         (Brief="Auxiliary Cooler Heat Duty",final Unit = 'm^2*kg/s^3');
SC                 as  fraction     (Brief="Split Range after Compressor; SC=FCIn.F/CompOut.F");
SCb                as  fraction     (Brief="Split Range at Column Bottom; SCb=LinB.F/Ltray(Nt).F");
SR                 as  fraction     (Brief="Split Range after Accumulator = refluxo + produto; SR=D.F/TankL.F");
SumpLevel          as  length       (Brief="Level of Liquid Level in Sump");
SumpLevelP         as  percent      (Brief="Level of Liquid Level in Sump - percentage");
SumpVfilled        as  volume       (Brief="Occupied Volume of Sump by the Liquid");
TankL              as  liquid_stream(Brief="Accumulator Liquid Outlet - Stream 19");
TankPump           as  liquid_stream(Brief="Accumulator Liquid Outlet after Pump, Stream 20");
TankV              as  vapour_stream(Brief="Accumulator Vapor Outlet - Stream 8");
Tiso               as  temperature  (Brief="Isentropic Efficiency");
Vboiler            as  vapour_stream(Brief="Boil up Stream");
vfB                as  fraction     (Brief="Vapor Fraction in Boil Up Stream");
vLA                as  volume_mol   (Brief="Accumulator Liquid Molar Volume");
vLC(Nt)            as  volume_mol   (Brief="Column Molar Specific Volume");
vLS                as  volume_mol   (Brief="Accumulator Liquid Molar Specific Volume");
VoutB              as  vapour_stream(Brief="Reboiler/Condenser Outlet Stream, shell side");
Vsump              as  vapour_stream(Brief="Sump Vapour Outlet Stream");
```

```
    Vtank             as  vapour_stream(Brief="Recycle, V-9709");
    Vtray(Nt)         as  vapour_stream(Brief="Vapor Internal Flows Through Column");
    vVA               as  volume_mol   (Brief="Accumulator Vapour Molar Specific Volume");
    vVC(Nt)           as  volume_mol   (Brief="Column Molar Specific Volume");
    vVS               as  volume_mol   (Brief="Accumulator Liquid Molar Specific Volume");
    Win               as  Real         (Brief="Compressor Energy",final Unit = 'm^2*kg/s^3');
    yeq(NComp*Nt)     as  fraction     (Brief="Tray Equilibrium Composition");


    EQUATIONS

    # Units Conversion - Average Molecular Weight

    D_mm            = sum(Mw*D.z);

    FBIn_mm         = sum(Mw*FBIn.z);

    FCOut_mm        = sum(Mw*FCOut.z);

    F_mm            = sum(Mw*F.z);

    Lreflux_mm      = sum(Mw*Lreflux.z);

    PB_mm           = sum(Mw*PB.z);

    # Mass Flow Rates

    D_mass          = D.F*D_mm;

    FBIn_mass       = FBIn.F*FBIn_mm;

    FCOut_mass      = FCOut.F*FCOut_mm;

    F_mass          = F.F*F_mm;

    Lreflux_mass    = Lreflux.F*Lreflux_mm;

    PB_mass         = PB.F*PB_mm;

    #----------------------------DISTILLATION COLUMN: FEED CONDITION (ADIABATIC EXPANSION)-----------------------------

    # Total Molar balance
    F.F = Fl.F + Fv.F;

    # Component molar balance
    F.F*F.z = Fl.F*Fl.z + Fv.F*Fv.z;

    # Energy balance
    F.F*F.h = Fl.F*Fl.h + Fv.F*Fv.h;

    # Thermodynamic Equilibrium
    Fl.z*PP.LiquidFugacityCoefficient(Fl.T,Fl.P,Fl.z) = Fv.z*PP.VapourFugacityCoefficient(Fv.T,Fv.P,Fv.z);

    #Thermal Equilibrium
    Fl.T = Fv.T;
```

```
    # Mechanical Equilibrium
    (Fl.P) = 10.48*'atm';
    (Fv.P) = (Fl.P);

    #Summation
    sum(Fv.z)=1;


    #"DISTILLATION COLUMN: TRAYED SECTION"
    #*
       |---Top Tray--------|
       |-------------------|
       |----Rectification--|
       |-------------------|
       |----Feed Vapor-----|
    -->|----Feed Liquid----|
       |-------------------|
       |----Stripping------|
       |-------------------|
       |---Bottom tray-----|

    *#


    #"DISTILLATION COLUMN: TRAYED SECTION"
    #------------------------------------------ COLUMN ------------------------------------------------------------

    #------------------------------------ TOP TRAY: TRAY 1 ------------------------------------------------------

    # Total Molar balance
    diff(MLC(1) + MVC(1)) = Vtray(2).F + Lreflux.F - Vtray(1).F - Ltray(1).F;

    # Component molar balance
    diff(MLC(1)*Ltray(1).z + MVC(1)*Vtray(1).z)
    = Vtray(2).F*Vtray(2).z + Lreflux.F*Lreflux.z - Vtray(1).F*Vtray(1).z - Ltray(1).F*Ltray(1).z;

    # Energy balance
    1E-6*(diff(MLC(1)*Ltray(1).h + MVC(1)*Vtray(1).h - Vtray(1).P*TrayVt(1))) = 1E-6*(Vtray(2).F*Vtray(2).h + Lreflux.F*L

    # Thermodynamic Equilibrium
    yeq([1:NComp])*PP.VapourFugacityCoefficient(Ltray(1).T,Ltray(1).P,yeq([1:NComp])) = Ltray(1).z*PP.LiquidFugacityCoeff
    Vtray(1).z([1:NComp-1])-Vtray(2).z([1:NComp-1])= Er*(yeq([1:NComp-1])-Vtray(2).z([1:NComp-1]));

    # Thermal Equilibrium
    Vtray(1).T = Ltray(1).T;

    # Mechanical Equilibrium
    10E-5*(Vtray(1).P) = 10E-5*(Ltray(1).P);

    #Summation
    sum(Vtray(1).z) = 1;
    sum(yeq([1:10]))=1;
```

```
    #Liquid Specific Volume
    vLC(1) = PP.LiquidVolume(Ltray(1).T, Ltray(1).P, Ltray(1).z);
    # Vapour Specific Volume
    vVC(1)= PP.VapourVolume(Vtray(1).T, Vtray(1).P, Vtray(1).z);
    # Geometric Constraint
    TrayVt(1) = MLC(1) * vLC(1) + MVC(1) * vVC(1);

    # Index Reduction Equations
    Vtray(1).F/'kmol/h' = VtraySS(1)/'kmol/h' + Kcolp*((Vtray(1).P/'kPa' - PSS/'kPa'));# -   DeltaP/'kPa');
    Ltray(1).F/'kmol/h' = LtraySS(1)/'kmol/h' + Kcolh*(MLC(1)*vLC(1)/TrayArea/'m' - MLCSS(1)*vLC(1)/TrayArea/'m');


    #---------------------------------------- UPPER SECTION: TRAYS 2 TO Nf-2-----------------------------------------

    for stage in 2:Nf-2 do

    # Total Molar balance
    diff(MLC(stage)+MVC(stage)) = Vtray(stage + 1).F + Ltray(stage - 1).F - Ltray(stage).F - Vtray(stage).F;

    # Component molar balance
    diff(MLC(stage)*Ltray(stage).z+MVC(stage)*Vtray(stage).z) = Vtray(stage + 1).F*Vtray(stage + 1).z + Ltray(stage - 1).

    # Energy balance
    1E-6*(diff(MLC(stage)*Ltray(stage).h + MVC(stage)*Vtray(stage).h - Vtray(stage).P*TrayVt(stage))) = 1E-6*(Vtray(stage

    # Thermal Equilibrium
    yeq([NComp*(stage-1)+1:NComp*stage])*PP.VapourFugacityCoefficient(Vtray(stage).T,Vtray(stage).P,yeq([NComp*(stage-1)+
    Vtray(stage).z([1:NComp-1])-Vtray(stage+1).z([1:NComp-1]) = Er*(yeq([NComp*(stage-1)+1:NComp*stage-1])-Vtray(stage+1)


    # Thermodynamic Equilibrium
    Vtray(stage).T = Ltray(stage).T;

    # Mechanical Equilibrium
    10E-5*(Vtray(stage).P) = 10E-5*(Ltray(stage).P);

    # Summation
    sum(Vtray(stage).z) = 1;
    sum(yeq([NComp*(stage-1)+1:NComp*stage]))=1;

    # Liquid Specific Volume
    vLC(stage) = PP.LiquidVolume(Ltray(stage).T, Ltray(stage).P, Ltray(stage).z);
    # Vapour Specific Volume
    vVC(stage)= PP.VapourVolume(Vtray(stage).T, Vtray(stage).P, Vtray(stage).z);
    # Geometric Constraint
    TrayVt(stage) = MLC(stage) * vLC(stage) + MVC(stage) * vVC(stage);

    # Index Reduction Equations
    Vtray(stage).F/'kmol/h' = VtraySS(stage)/'kmol/h' + Kcolp*((Vtray(stage).P/'kPa' - Vtray(stage - 1).P/'kPa') -   DPjr
    Ltray(stage).F/'kmol/h' = LtraySS(stage)/'kmol/h' + Kcolh*(MLC(stage)*vLC(stage)/TrayArea/'m' - MLCSS(stage)*vLC(stag

    end

    #-----------------------------------------------------STAGE Nf-1-------------------------------------

    # Total Molar balance
    diff(MLC(Nf-1)+MVC(Nf-1)) = Vtray(Nf).F + Ltray(Nf - 2).F + Fv.F - Vtray(Nf-1).F - Ltray(Nf-1).F;
```

```
    # Component molar balance
    diff(MLC(Nf-1)*Ltray(Nf-1).z+MVC(Nf-1)*Vtray(Nf-1).z) = Vtray(Nf).F*Vtray(Nf).z + Ltray(Nf - 2).F*Ltray(Nf - 2).z + F

    # Energy balance
    1E-6*(diff(MLC(Nf-1)*Ltray(Nf-1).h + MVC(Nf-1)*Vtray(Nf-1).h - Vtray(Nf-1).P*TrayVt(Nf-1))) = 1E-6*(Vtray(Nf).F*Vtray

    # Thermodynamic Equilibrium
    yeq([NComp*(Nf-2)+1:NComp*(Nf-1)])*PP.VapourFugacityCoefficient(Vtray(Nf-1).T,Vtray(Nf-1).P,yeq([NComp*(Nf-2)+1:NComp
    Vtray(Nf-1).z([1:NComp-1])-Vtray(Nf).z([1:NComp-1])= Er*(yeq([NComp*(Nf-2)+1:NComp*(Nf-1)-1])-Vtray(Nf).z([1:NComp-1]

    # Thermal Equilibrium
    Vtray(Nf-1).T = Ltray(Nf-1).T;

    # Mechanical Equilibrium
    10E-5*(Vtray(Nf-1).P) = 10E-5*(Ltray(Nf-1).P);

    # Summation
    sum(Vtray(Nf-1).z) = 1;
    sum(yeq([NComp*(Nf-2)+1:NComp*(Nf-1)]))=1;

    # Liquid Specific Volume
    vLC(Nf-1) = PP.LiquidVolume(Ltray(Nf-1).T, Ltray(Nf-1).P, Ltray(Nf-1).z);
    # Vapour Specific Volume
    vVC(Nf-1)= PP.VapourVolume(Vtray(Nf-1).T, Vtray(Nf-1).P, Vtray(Nf-1).z);
    # Geometric Constraint
    TrayVt(Nf-1) = MLC(Nf-1) * vLC(Nf-1) + MVC(Nf-1) * vVC(Nf-1);

    # Index Reduction Equations
    Vtray(Nf-1).F/'kmol/h' = VtraySS(Nf-1)/'kmol/h' + Kcolp*((Vtray(Nf-1).P/'kPa' - Vtray(Nf - 2).P/'kPa') - DPjr/'kPa');
    Ltray(Nf-1).F/'kmol/h' = LtraySS(Nf-1)/'kmol/h' + Kcolh*(MLC(Nf-1)*vLC(Nf-1)/TrayArea/'m' - MLCSS(Nf-1)*vLC(Nf-1)/Tra

    #-----------------------------------------------------Feed tray-------------------------------------------

    # Total Molar balance
    diff(MLC(Nf)  + MVC(Nf)) = Vtray(Nf + 1).F + Ltray(Nf - 1).F + Fl.F  - Vtray(Nf).F - Ltray(Nf).F;

    # Component molar balance
    diff(MLC(Nf)*Ltray(Nf).z+MVC(Nf)*Vtray(Nf).z) = Vtray(Nf + 1).F*Vtray(Nf + 1).z + Ltray(Nf - 1).F*Ltray(Nf - 1).z + F

    # Energy balance
    1E-6*(diff(MLC(Nf)*Ltray(Nf).h + MVC(Nf)*Vtray(Nf).h - Vtray(Nf).P*TrayVt(Nf))) = 1E-6*(Vtray(Nf + 1).F*Vtray(Nf + 1)

    # Thermodynamic Equilibrium
    yeq([NComp*(Nf-1)+1:NComp*(Nf)])*PP.VapourFugacityCoefficient(Vtray(Nf).T,Vtray(Nf).P,yeq([NComp*(Nf-1)+1:NComp*(Nf)]
    Vtray(Nf).z([1:NComp-1])-Vtray(Nf+1).z([1:NComp-1])= Es*(yeq([NComp*(Nf-1)+1:NComp*(Nf)-1])-Vtray(Nf+1).z([1:NComp-1]

    # Thermal Equilibrium
    Vtray(Nf).T = Ltray(Nf).T;

    # Mechanical Equilibrium
    10E-5*(Vtray(Nf).P) = 10E-5*(Ltray(Nf).P);

    # Summation
    sum(Vtray(Nf).z) = 1;
    sum(yeq([NComp*(Nf-1)+1:NComp*(Nf)]))=1;
```

```
559
560     # Liquid Specific Volume
561     vLC(Nf) = PP.LiquidVolume(Ltray(Nf).T, Ltray(Nf).P, Ltray(Nf).z);
562     # Vapour Specific Volume
563     vVC(Nf)= PP.VapourVolume(Vtray(Nf).T, Vtray(Nf).P, Vtray(Nf).z);
564     # Geometric Constraint
565     TrayVt(Nf) = MLC(Nf) * vLC(Nf) + MVC(Nf) * vVC(Nf);
566
567     # Index Reduction Equations
568     Vtray(Nf).F/'kmol/h' = VtraySS(Nf)/'kmol/h' + Kcolp*((Vtray(Nf).P/'kPa' - Vtray(Nf - 1).P/'kPa') -   DPjs/'kPa');
569     Ltray(Nf).F/'kmol/h' = LtraySS(Nf)/'kmol/h' + Kcolh*(MLC(Nf)*vLC(Nf)/TrayArea/'m' - MLCSS(Nf)*vLC(Nf)/TrayArea/'m');
570
571     #------------------------------------- LOWER SECTION: TRAYS Nf+1 TO Nt-1-------------------------------------
572
573     for stage in Nf+1:Nt-1 do
574
575     # Total Molar balance
576     diff(MLC(stage)+MVC(stage)) = Vtray(stage + 1).F + Ltray(stage - 1).F - Ltray(stage).F - Vtray(stage).F;
577
578     # Component molar balance
579     diff(MLC(stage)*Ltray(stage).z+MVC(stage)*Vtray(stage).z) = Vtray(stage + 1).F*Vtray(stage + 1).z + Ltray(stage - 1).
580
581     # Energy balance
582     1E-6*(diff(MLC(stage)*Ltray(stage).h + MVC(stage)*Vtray(stage).h - Vtray(stage).P*TrayVt(stage))) = 1E-6*(Vtray(stage
583
584     # Thermodynamic Equilibrium
585     yeq([NComp*(stage-1)+1:NComp*stage])*PP.VapourFugacityCoefficient(Vtray(stage).T,Vtray(stage).P,yeq([NComp*(stage-1)+
586     Vtray(stage).z([1:NComp-1])-Vtray(stage+1).z([1:NComp-1])= Es*(yeq([NComp*(stage-1)+1:NComp*(stage)-1])-Vtray(stage+1
587
588     # Thermal Equilibrium
589     Vtray(stage).T = Ltray(stage).T;
590
591     # Mechanical Equilibrium
592     10E-5*(Vtray(stage).P) = 10E-5*(Ltray(stage).P);
593
594     # Summation
595     sum(Vtray(stage).z) = 1;
596     sum(yeq([NComp*(stage-1)+1:NComp*stage]))=1;
597
598     # Liquid Specific Volume
599     vLC(stage) = PP.LiquidVolume(Ltray(stage).T, Ltray(stage).P, Ltray(stage).z);
600     # Vapour Specific Volume
601     vVC(stage)= PP.VapourVolume(Vtray(stage).T, Vtray(stage).P, Vtray(stage).z);
602     # Geometric Constraint
603     TrayVt(stage) = MLC(stage) * vLC(stage) + MVC(stage) * vVC(stage);
604
605     # Index Reduction Equations
606     Vtray(stage).F/'kmol/h' = VtraySS(stage)/'kmol/h' + Kcolp*((Vtray(stage).P/'kPa' - Vtray(stage - 1).P/'kPa') -   DPjs
607     Ltray(stage).F/'kmol/h' = LtraySS(stage)/'kmol/h' + Kcolh*(MLC(stage)*vLC(stage)/TrayArea/'m' - MLCSS(stage)*vLC(stag
608
609     end
610
611     #------------------------------------------------ LAST STAGE: TRAY Nt-----------------------------------------
612
613     # Total Molar balance
614     diff(MLC(Nt)+MVC(Nt)) = VoutB.F + Ltray(Nt - 1).F - Ltray(Nt).F - Vtray(Nt).F;
615
```

```
    # Component molar balance
    diff(MLC(Nt)*Ltray(Nt).z+MVC(Nt)*Vtray(Nt).z) = VoutB.F*VoutB.z + Ltray(Nt - 1).F*Ltray(Nt - 1).z - Ltray(Nt).F*Ltray

    # Energy balance
    1E-6*(diff(MLC(Nt)*Ltray(Nt).h + MVC(Nt)*Vtray(Nt).h - Vtray(Nt).P*TrayVt(Nt))) = 1E-6*(VoutB.F*VoutB.h + Ltray(Nt -

    # Thermodynamic Equilibrium
    yeq([NComp*(Nt-1)+1:NComp*Nt])*PP.VapourFugacityCoefficient(Vtray(Nt).T,Vtray(Nt).P,yeq([NComp*(Nt-1)+1:NComp*Nt])) =
    Vtray(Nt).z([1:NComp-1]) - VoutB.z([1:NComp-1]) = Es*(yeq([NComp*(Nt-1)+1:NComp*(Nt)-1])-VoutB.z([1:NComp-1]));

    # Thermal Equilibrium
    Vtray(Nt).T = Ltray(Nt).T;

    # Mechanical Equilibrium
    10E-5*(Vtray(Nt).P) = 10E-5*(Ltray(Nt).P);

    # Summation
    sum(Vtray(Nt).z) = 1;
    sum(yeq([NComp*(Nt-1)+1:NComp*Nt]))=1;

    # Liquid Specific Volume
    vLC(Nt) = PP.LiquidVolume(Ltray(Nt).T, Ltray(Nt).P, Ltray(Nt).z);
    # Vapour Specific Volume
    vVC(Nt)= PP.VapourVolume(Vtray(Nt).T, Vtray(Nt).P, Vtray(Nt).z);
    # Geometric Constraint
    TrayVt(Nt) = MLC(Nt) * vLC(Nt) + MVC(Nt) * vVC(Nt);

    # Index Reduction Equations
    Vtray(Nt).F/'kmol/h' = VtraySS(Nt)/'kmol/h' + Kcolp*((Vtray(Nt).P/'kPa' - Vtray(Nt - 1).P/'kPa') -   DPjs/'kPa');
    Ltray(Nt).F/'kmol/h' = LtraySS(Nt)/'kmol/h' + Kcolh*(MLC(Nt)*vLC(Nt)/TrayArea/'m' - MLCSS(Nt)*vLC(Nt)/TrayArea/'m');

    #------------------------------------------------ SUMP ----------------------------------------

    # Total Molar balance
    diff(MLS + MVS) = LoutB.F + Ltray(Nt).F - Vsump.F - Lsump.F;

    # Component molar balance
    diff(MLS*Lsump.z + MVS*Vsump.z) = LoutB.F*LoutB.z + Ltray(Nt).F*Ltray(Nt).z - Vsump.F*Vsump.z - Lsump.F*Lsump.z;

    # Energy balance
    1E-6*(diff(MLS*Lsump.h + MVS*Vsump.h - Vsump.P*SumpV)) = 1E-6*(LoutB.F*LoutB.h + Ltray(Nt).F*Ltray(Nt).h - Vsump.F*Vs

    # Thermodynamic Equilibrium
    Vsump.z*PP.VapourFugacityCoefficient(Vsump.T,Vsump.P,Vsump.z) = Lsump.z*PP.LiquidFugacityCoefficient(Lsump.T,Lsump.P,

    # Thermal Equilibrium
    Vsump.T = Lsump.T;

    # Mechanical Equilibrium
    1E-4*(Vsump.P) = 1E-4*(Lsump.P);

    # Summation
    sum(Vsump.z)=1;

    # Liquid Specific Volume
    vLS = PP.LiquidVolume(Lsump.T, Lsump.P, Lsump.z);
```

```
672     # Vapour Specific Volume
673     vVS = PP.VapourVolume(Vsump.T, Vsump.P, Vsump.z);
674     # Geometric Constraint
675     SumpV = MLS * vLS + MVS * vVS;
676
677     SumpVfilled = MLS*vLS;
678     SumpVfilled = TrayArea*SumpLevel;
679
680     #"Tank Level in Percentage
681     SumpLevel = SumpSpacing*SumpLevelP/100;
682
683     # Index Reduction Equations
684     Vsump.F/'kmol/h' = VsumpSS/'kmol/h' + Ksumpp*(Vsump.P/'kPa' - VsumpPSS/'kPa');
685
686     #------------------------------------------- BOTTOM ------------------------------------------------------------
687
688     #"DISTILLATION COLUMN: BOTTOM FLOW SPLITTER"
689     PB.F   = SCb*Lsump.F;
690     LinB.F = (1-SCb)*Lsump.F;
691
692     PB.T   = Lsump.T;
693     LinB.T = Lsump.T;
694
695     1E-4*(PB.P)   = 1E-4*(Lsump.P);
696     1E-4*(LinB.P) = 1E-4*(Lsump.P);
697
698     PB.z   = Lsump.z;
699     LinB.z = Lsump.z;
700
701     1E-6*(PB.h)   = 1E-6*(Lsump.h);
702     1E-6*(LinB.h) = 1E-6*(Lsump.h);
703
704     PB.v   = Lsump.v;
705     LinB.v = Lsump.v;
706
707     #------------------------------------------- REBOILER/Condenser-------------------------------------------------
708
709     # Shell Side
710     vfB*LinB.F = VoutB.F;
711
712     (1-vfB)*LinB.F = LoutB.F;
713
714     LinB.F*LinB.z= VoutB.F*VoutB.z + LoutB.F*LoutB.z;
715
716     VoutB.z*PP.VapourFugacityCoefficient(VoutB.T,VoutB.P,VoutB.z) = LoutB.z*PP.LiquidFugacityCoefficient(LoutB.T,LoutB.P,
717
718     LoutB.T = VoutB.T;
719
720     1E-4*(LoutB.P) = 1E-4*(VoutB.P);
721     1E-4*(VoutB.P) = 1E-4*(LinB.P);
722
723     sum(LoutB.z)=1;
724
725     # Tube Side
726     FBIn.F = FBOut.F;
727
```

```
728     FBIn.z = FBOut.z;
729
730     1E-4*(FBIn.P) = 1E-4*(FBOut.P + PdropB);
731
732     # Energy Balance in Reboiler/Condenser
733
734     1E-6*Qbout = 1E-6*(FBIn.F*FBIn.h - FBOut.F*FBOut.h + Qbl);
735
736     1E-6*Qbin = 1E-6*(LinB.F*LinB.h - VoutB.F*VoutB.h - LoutB.F*LoutB.h);
737
738     1E-6*(6*diff(Qbin) + Qbin/'s') = 1E-6*(-1/'s'*Qbout);
739
740     #---------------------------------------MIXER in the Column Bottom-------------------------------------------------
741
742     Vboiler.F = VoutB.F + Vsump.F;
743
744     Vboiler.F*Vboiler.z = VoutB.F*VoutB.z + Vsump.F*Vsump.z;
745
746     1E-6*(Vboiler.F*Vboiler.h) = 1E-6*(VoutB.F*VoutB.h + Vsump.F*Vsump.h);
747
748     1E-4*(Vboiler.P) = 1E-4*(VoutB.P);
749
750     #------------------------------------------- TOP ---------------------------------------------------------------
751
752     #------------------------ MIXER COMPRESSOR: ADIABATIC MIXING OF THE STREAMS ENTERING THE COMPRESSOR----------------
753
754     CompIn.F = Vtank.F + Vtray(1).F;
755
756     CompIn.F*CompIn.z = Vtank.F*Vtank.z + Vtray(1).F*Vtray(1).z;
757
758     1E-6*(CompIn.F*CompIn.h) = 1E-6*(Vtank.F*Vtank.h + Vtray(1).F*Vtray(1).h);
759
760     1E-4*(CompIn.P) = 1E-4*(Vtray(1).P);
761
762     #------------------------------- COMPRESSOR: CENTRIFUGAL USING ADIABATIC EFFICIENCY-------------------------------
763     CompIn.F=CompOut.F;
764
765     CompIn.z=CompOut.z;
766
767     1E-4*(CompOut.P) = 1E-4*(PcompOut);
768
769     PP.VapourEntropy(CompIn.T,CompIn.P,CompIn.z)=PP.VapourEntropy(Tiso,CompOut.P,CompOut.z);
770
771     1E-6*(eta*(CompOut.h-CompIn.h)) = 1E-6*(PP.VapourEnthalpy(Tiso,CompOut.P,CompOut.z)-CompIn.h);
772
773     Win = 1E-6*(CompOut.F*CompOut.h-CompIn.F*CompIn.h);
774
775     # ------------------------------------- COMPRESSOR: FLOW SPLITTER ------------------------------------------------
776     FCIn.F=SC*CompOut.F;
777     FBIn.F=(1-SC)*CompOut.F;
778
779     FCIn.T=CompOut.T;
780     FBIn.T=CompOut.T;
781
```

```
    1E-4*(FCIn.P) = 1E-4*(CompOut.P);
    1E-4*(FBIn.P) = 1E-4*(CompOut.P);

    FCIn.z=CompOut.z;
    FBIn.z=CompOut.z;

    1E-6*(FCIn.h) = 1E-6*(CompOut.h);
    1E-6*(FBIn.h) = 1E-6*(CompOut.h);

    FCIn.v= CompOut.v;
    FBIn.v= CompOut.v;

    #------------------------------------------------- COOLER-------------------------------------------------------
    FCIn.F = FCOut.F;
    FCIn.z = FCOut.z;

    1E-4*(FCIn.P) = 1E-4*(FCOut.P + PdropC);

    1E-6*(Qc + Qcl + FCIn.F*FCIn.h) = 1E-6*(FCOut.F*FCOut.h);

    #------------------------------------------------ Accumulator-------------------------------------------------------

    # Total Molar balance
    diff(MLA + MVA) = FBOut.F + FCOut.F - TankL.F - TankV.F;

    #"Component Molar Balance"
    diff(MLA*TankL.z + MVA*TankV.z) = FBOut.F*FBOut.z + FCOut.F*FCOut.z - TankL.F*TankL.z - TankV.F*TankV.z;

    # Energy Balance
    1E-6*(diff(MLA*TankL.h + MVA*TankV.h - (AccumulatorVt - AccumulatorVfilled)*TankL.P)) = 1E-6*(FBOut.F*FBOut.h + FCOut

    # Thermodynamic Equilibrium
    TankL.z*PP.LiquidFugacityCoefficient(TankL.T, TankL.P, TankL.z) = TankV.z*PP.VapourFugacityCoefficient(TankV.T, TankV

    # Summation
    sum(TankV.z) = 1;


    # Thermal Equilibrium
    TankL.T = TankV.T;

    # Mechanical Equilibrium
    1E-4*(TankL.P) = 1E-4*(TankV.P);

    # Liquid Specific Volume
    vLA = PP.LiquidVolume(TankL.T, TankL.P, TankL.z);
    # Vapour Specific Volume
    vVA= PP.VapourVolume(TankV.T, TankV.P, TankV.z);
    # Geometric Constraint
    AccumulatorVt = MLA*vLA + MVA*vVA;
    AccumulatorVfilled = MLA*vLA;

    # Index Reduction Equations
    TankV.F/'kmol/h' = TankVSS/'kmol/h' + Kacump*(TankV.P/'kPa' - TankVPSS/'kPa');
    TankL.F/'kmol/h' = TankLSS/'kmol/h' + Kacumh*(MLA*vLA/'m^3' - MLASS*vLA/'m^3');
```

```
    # ---------------------------------ADIABATIC EXPANSION VALVE in Recycle Stream----------------------------------
    Vtank.F = TankV.F;
    1E-6*(Vtank.h) = 1E-6*(TankV.h);
    1E-4*(Vtank.P) = 1E-4*(Vtray(1).P);
    Vtank.z = TankV.z;

    # ------------------------------------------PUMP AFTER ACCUMULATOR ----------------------------------------------
    TankL.F = TankPump.F;
    TankL.z = TankPump.z;
    TankL.T = TankPump.T;
    1E-4*(TankPump.P) = 1E-4*(TankL.P + DeltaPump);

    # ------------------------------------------SPLITTER after Accumulator-------------------------------------------
    D.F= SR*TankPump.F;
    Lreflux.F= (1-SR)*TankPump.F;

    D.T= TankPump.T;
    Lreflux.T= TankPump.T;

    1E-4*(D.P)= 1E-4*(TankPump.P);
    1E-4*(Lreflux.P) = 1E-4*(TankPump.P);

    D.z= TankPump.z;
    Lreflux.z= TankPump.z;

end
 #------------------------------------------------- Disturbances ---------------------------------------------------
Model InputModel

    PARAMETERS
    eps as Real;

    VARIABLES
    x as Real;
    n(1) as Real;
    u(1) as Real;
    F(1) as Real;

    SET
    eps = 1e-5;

    EQUATIONS

    diff(x)*'h' = -0.001*x;

    n(1)        = 0.5*(1 + tanh(( 0.99- x)/eps));
    u(1)        = -100*n(1);
    F(1)        = u(1);

    end
 #------------------------------------------- PI level Controller ----------------------------------------------------
Model PI_Controller
```

```
    PARAMETERS
    Kp            as Real   (Brief="Controller gain");
    Ki            as Real   (Brief="Integral time constant", Unit='s');
    Bias          as Real   (Brief="Previous scaled bias - Nominal Vlaue");
    MinInput      as Real   (Brief="Lower limit for Input Variable Value");;
    MaxInput      as Real   (Brief="Upper Limit for Input Variable Value");;
    MinOutput     as Real   (Brief="Lower limit for Output Variable value");;
    MaxOutput     as Real   (Brief="Upper Limit for Output Variable Value");;


    VARIABLES
    Input         as Real   (Brief="Process input");
    Output        as Real   (Brief="Process output");
    SetPoint      as Real   (Brief="Scaled setPoint");
    input         as Real   (Brief="Scaled input variable");
    setPoint      as Real   (Brief="Scaled set point");
    intTerm       as Real   (Brief="Integral term");
    outps         as Real   (Brief="Variable outp scaled between 0 and 1");
    bias          as Real   (Brief="Previous scaled bias - Normalized Vlaue");;

    EQUATIONS

    "Calculate integral term"
    Ki*diff(intTerm) =  input - setPoint;

    "Sum of proportional, integral and derivative terms"
    outps = bias + Kp*(input - setPoint) + intTerm;

     input*(MaxInput-MinInput) = Input-MinInput;
     setPoint*(MaxInput-MinInput) = SetPoint-MinInput;
     bias*(MaxOutput-MinOutput) = Bias-MinOutput;


    if outps > 1 then
        Output = MaxOutput;
    else
        if  outps < 0 then
        Output = MinOutput;
        else
        Output = outps*(MaxOutput - MinOutput) + MinOutput;

        end
    end

    INITIAL
    intTerm = 0;

end
```