SLOVAK UNIVERSITY OF TECHNOLOGY
IN BRATISLAVA
FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

Reg. No.: FCHPT-5414-44247

Optimal Control of Processes in Chemical Technology

Diploma Thesis

2015 **Bc. Daniela Pakšiová**

SLOVAK UNIVERSITY OF TECHNOLOGY
IN BRATISLAVA
FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

Reg. No.: FCHPT-5414-44247

Optimal Control of Processes in Chemical Technology

Diploma Thesis

Study program:	Automation and Information Engineering in Chemical and Food Technologies
Study field number:	2621
Study field:	5.2.14 Automation
Workplace:	Trondheim, Bratislava
Supervisor:	Prof. Ing. Miroslav Fikar, DrSc.
Consultant:	Prof. Sigurd Skogestad

Bratislava, 2015 **Bc. Daniela Pakšiová**

Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation and Mathematics

Faculty of Chemical and Food Technology
Academic year: 2014/2015
Reg. No.: FCHPT-5414-44247

DIPLOMA THESIS TOPIC

Student:	**Bc. Daniela Pakšiová**
Student's ID:	44247
Study programme:	Automation and Information Engineering in Chemistry and Food Industry
Study field:	5.2.14. automation
Thesis supervisor:	prof. Ing. Miroslav Fikar, DrSc.
Consultant:	prof. S. Skogestad
Workplace:	Trondheim, Bratislava

Topic: **Optimal control of processes in chemical technology**

Specification of Assignment:

The work will be devoted to optimal control of typical processes in chemical and food technologies. The first part will deal with membrane filtration processes, their modelling, simulations, and properties. Optimal control of this processes will be studied. The second part will deal with a more general problem of optimal control with emphasis on control of multiple units.

The work will be done in cooperation with NTNU Trondheim.

Tasks:
- Modelling of membrane processes
- Optimal control design
- Control of multiple membrane units
- Testing and determination of performance using simulations

Length of thesis: 40

Selected bibliography:

1. Lewis, F L. *Applied optimal control & estimation : Digital design & implementation.* Englewood Cliffs: Prentice Hall, 1992. 624 s. ISBN 0-13-140361-X.
2. Bryson, A E. – Ho, Y. *Applied optimal control : Optimization, estimation, and control.* New York: Taylor & Francis, 1975. 481 s. ISBN 0-89116-228-3.
3. Bertsekas, D P. *Dynamic Programming and Optimal Control : Volume I.* Belmont: Athena Scientific, 2005. 543 s. ISBN 1-886529-26-4.

Assignment procedure from: 16. 02. 2015

Date of thesis submission: 24. 05. 2015

L. S.

Bc. Daniela Pakšiová
Student

prof. Ing. Miroslav Fikar, DrSc.
Head of department

prof. Ing. Miroslav Fikar, DrSc.
Study programme supervisor

Acknowledgements

I am grateful to several people for their help during the course of writing my diploma thesis. First of all, I would like to express my gratitude to Professor Miroslav Fikar, who gave me the opportunity to participate in the Erasmus program and to work on my diploma thesis at NTNU in Trondheim, Norway.

Moreover, I would like to thank my supervisor in Trondheim, Professor Sigurd Skogestad, for all of his patient guidance, support, encouragement and sessions of discussions, which were of great help to me for compliting this work. My other thanks goes to Johannes Jäschke for his patience, advice, and guidance.

Special thanks goes to my family and my friends who were there for me at all times whenever I needed them, who supported me and encouraged me.

Daniela Pakšiová
Bratislava, 2015

Abstract

This work deals with membrane separation processes, their modelling and optimal control. A self-optimizing control is introduced, a method proposed by Johannes Jäschke (NTNU, Norway) to control heat exchanger network by controlling "Jäschke temperatures". This type of self-optimizing control is applied to control membrane processes, specifically membrane contactor network. This network consists of a stream split into several branches where each branch passes through one or more membrane contactors. The Jäschke temperatures are defined for each branch, and the optimal controlled variable is the difference between Jäschke temperatures of each branch. By controlling difference of Jäschke temperatures a near-optimal condition is achieved, where there is no need for optimization to find the optimal set-points for the controlled variables.

The aim of this work is modelling a membrane contactor network and its optimal control to achieve maximum separation and to confront the results obtained from using the Jäschke temperature to the ones from optimization and NTU method.

Keywords: Self-optimizing Control, Membrane Process Network, Optimal Control, Membrane Contactor Network

Abstrakt

Táto práca sa zaoberá membránovými separačnými procesmi, ich modelovaním a optimálnym riadením. Pojem samo-optimalizačné riadenie sa zanáša, ako metóda navrhnutá Johannesom Jäschke (NTNU, Nórsko) na riadenie sústavy výmenníkov tepla pomocou riadením "Jäschkého teplôt". Tento typ samo-optimalizačného riadenia sa aplikoval na riadenie membránových procesov, obzvlášť na riadenie sústavy membránových kontaktorov. Táto sústava membránových kontaktorov sa skladá z prúdu suroviny, ktorý je rozdelený do viacerých vetví, na ktorých sa nachádza jeden alebo viacero membránových kontaktorov. Jäschkého teploty sú definované pre každú vetvu, pričom optimálna riadená veličina je rozdiel týchto Jäschkého teplôt na každej vetve. Riadením rozdielu Jäschkého teplôt sa získa takmer optimálna prevádzka, pričom netreba riešiť optimalizačný problém hľadania optimálnej hodnoty žiadanej veličiny.

Cieľom tejto práce je modelovanie do siete zapojených membránových kontaktorov a ich optimálne riadenie na dosiahnutie maximálnej separácie a porovnanie výsledkov získaných za použitia Jäschkého teploty s výsledkami dosiahnutými z optimalizácie a NTU metódy.

Kľúčové slová: Samo-optimalizačné riadenie, sústava membránových procesov, optimálne riadenie, sústava membránových kontaktorov

Contents

List of Figures

List of Tables

List of abbreviations

HDA	Hydrodealkylation of toluene
HEN	Heat exchange network
MC	Membrane contactor
MEN	Mass exchange network
MSA	Mass separation- agent
NTU	Number of transfer units

Chapter 1

Introduction

Optimization is a process of identification of the best solution from among a set of possible solutions. Those solutions are obtained by minimization or maximization of an objective function subject to some constraints. Thus, optimization is very effective and powerful tool when it comes to solve process problems (Chong and Zak, 2001). Nowadays, optimization methods are widely used in business, industry, government, engineering and computer science. Optimization techniques are used when it comes to questions of industrial planning, resource allocation, scheduling, shortest route, optimal (minimum time) trajectories for space missions, etc. An edge in maximizing profits or minimizing costs can often mean the difference between success and failure in business. There is no single method available for solving all optimization problems. Therefore, a number of methods have been developed for solving different types of optimization problems (Chinneck, 2009).

The concept of self-optimizing control is introduced, a method for finding optimal controlled variables and by controlling these variables a near-optimal operation is achieved. This approach of optimizing plant performance uses a process model off-line to find a "self-optimizing control" structure. The idea of this approach is to find a function c of the process variables which when held constant by implementing simple control structures, e.g. PID controllers or by more complex model predictive control, leads to the optimal adjustments of the manipulated variables (Jäschke and Skogestad, 2012).

This work deals with optimization of membrane separation processes. There has not been done much work regarding membrane optimization since membrane separation processes are relatively new processes. Nowadays, they are becoming increasingly important in the process industries used both in production and downstream processing due to their energy demands and higher efficiency of the achieved separation or processing goals. Most industrial scale separation processes are based on energy intensive methods such as distillation, evaporation and freeze crystallization (Kujawski, 1989). Membrane technologies can offer significant advantages over existing separation processes such as energy savings, low-cost modular construction, high selectivity of separated materials and processing of temperature-sensitive products, they are eco-friendly since the membrane approach requires the use of relatively simple and non-harmful materials, modules can be added and optimized in a process design to achieve desired separation, they have low specific power consumption, which reduce the production cost, and the process is continuous and is generally carried out at atmospheric conditions (Ravanchi et al., 2009).

The material in this work is organized into several chapters. The first chapter contains an overview of membrane processes and their applications in industry. The second chapter focuses on membrane contactors and their modelling. This is followed by chapters regarding optimization and self-optimizing control. Following chapters offer few case studies solved for membrane contactors. The procedure of optimization and self-optimizing control are applied to three case studies of membrane contactors which are connected in a network with the goal to maximize the separation process. Such networks consisting of several membrane contactors are solved with NTU method, in MATLAB solver `fmincon` and by self-optimizing control using Jäschke temperatures as to prove the applicability of self-optimizing control where Jäschke temperatures are used to determine the control variables and thus self-optimize the problem of multiple membrane contactors connected in a network. The last chapters discuss mass exchanger networks and a special case of application of membrane contactors.

Chapter 2

Membrane Processes

Membrane processes cover a group of separation processes in which the characteristics of a membrane are used to separate the components of a solution or a suspension. The membrane acts as a semipermeable barrier which separates two phases and restricts transport of various chemicals in selective manner. This separation may occurs between two liquid phases, two gas phases or between a liquid and a gas phase (Geankoplis, 2008). The transport through the membrane occurs when a driving force is applied to the components in the feed stream. In these processes the feed stream is separated into two: the fraction that permeates through the membrane, called the permeate, and the fraction containing the components that have not been transported through the membrane, usually called the retentate (Fikar and Paulen, 2015). A simple representation of a membrane is shown in Fig. 2.1. The size of the components to be separated and the nature and magnitude of the driving force provide criteria for a classification of the membrane separation processes.

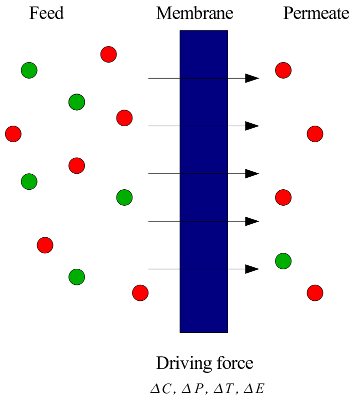

Figure 2.1: A representation of a membrane process

Based on their driving forces membrane processes may be divided in four main groups (Anonymous, 1999; Ravanchi et al., 2009):

- pressure driven processes - microfiltration, ultrafiltration , nanofiltration , reverse osmosis, membrane contactors(6)
- concentration driven processes - dialysis, pervaporation, forward osmosis, artificial lung, gas separation
- electrical potential gradient processes - electrodialysis, fuel cells, electrofiltration, electrodeionization, membrane electrolysis
- temperature gradient processes - membrane distillation

In a membrane separation processes there are different ways of filtering a feed solution through the membrane. The most common operative designs which can be used are as follows:

- cross-flow mode, where the feed stream flows parallel to the membrane and is separated into the retentate and the permeate stream.
- dead-end mode, where the feed stream is forced perpendicularly to the membrane leading to the concentrate phase and a permeate.
- counter-current mode, where two feed streams flow inside membrane module at the opposite ends. This is mostly used in case of dialysis and membrane contactors (Baker, 2004; Geankoplis, 2008)

The cross-flow filtration mode is preferred over dead-end mode since it provides higher and more sustainable fluxes. By manipulating the out-flow of the solution, the pressure increases and a higher level of separation is achieved (Fikar and Paulen, 2015).

In dead-end filtration suspended particles are retained on a filter creating so called filter cake which grows and the resistance of the filter cake increases causing the filter to plug. If the flux has been reduced to unacceptably low levels the membrane has to be cleaned or replaced (Wilson, 2000).

In the following sections a more detailed description is presented for several membrane processes. The membrane processes discussed here are microfiltration, ultrafiltration, nanofiltration, reverse osmosis, dialysis, pervaporation and gas separation followed by a section dedicated to application uses of such processes in industry. Membrane contactors as membrane processes are covered in following chapter. An overview of membrane processes is shown in Tab. 2.1.

Table 2.1: Overview of membrane separation processes

Process	Retentate	Permeate	Driving force
Microfiltration	suspended solids bacteria	dissolved solutes and water	pressure
Ultrafiltration	macromolecules and colloids	dissolved salts and water	pressure
Nanofiltration	dissociated acids, dissolved salts and sugars	undissociated acids, and water	pressure
Reverse osmosis	dissolved salts	water	pressure
Dialysis	dissolved salts and gases	blood	concentration
Gas separation	impermeable gases and vapours	permeable gases and vapours	concentration
Pervaporation	impermeable microsolutes	permeable microsolutes	vapour pressure

2.1 Microfiltration

Microfiltration represents the most widely spread type of membrane processing industrial market within the membrane field. It is used for removal of micron-sized particles, like bacteria, yeast cells, sand, salt, colloids from suspensions or gases. Nowadays, microfiltration processes are operated in different fields such as biotechnological, automobile, electronics and food industry.

In this membrane process the driving force is the pressure gradient between the retentate and permeate streams. Microfiltration uses membranes filters with pore size bigger than $0.1\mu m$, which are permeable to the fluid, but retain particles with bigger particle size than the membrane. Thus, such membranes filtrate macromolecules and species with smaller particle sizes from suspended particles and bacteria bodies (Fikar and Paulen, 2015; Wilson, 2000).

Microfiltration can be carried out in two different operation modes: dead-end and cross-flow mode. Dead-end filtration is preferred when the concentration of particles to be removed from the fluid is very low, as in the case of gas cleaning and sterile filtration in the pharmaceutical industry (Wilson, 2000).

2.2 Ultrafiltration

Ultrafiltration needs, in general, relatively low differential pressures, starting from 0.1 of atmosphere up to a few atmospheres. The membrane retains on the basic of molecular size, shape or chemical structure. This type of membrane process retains macromolecules, typically proteins, polymers, microbiological contaminants, minerals and species with larger particle sizes then the pore size of the membrane, which is approximately 0.05 to 0.002μm.

By filtration through ultafiltration membrane the permeate stream may consist of dissociated acids, divalent salts, sugars and species with smaller particle sizes (Fikar and Paulen, 2015). It also allows most ionic inorganic species to pass through the membrane (Anonymous, 1999).

2.3 Nanofiltration

Nanofiltration is relatively a recent technology compared to other pressure driven filtration processes. It was turned into an industrial practice in 1980s. The membrane pore size varies between 0.002 and 0.001μm, making this membrane more selective, but at the same time the process requires higher operating pressure than microfiltration or ultrafiltration. The operating pressure is from 5 to 20 atm (Fikar and Paulen, 2015). The properties of nanofiltration membranes lie between phenomenons used in reverse osmosis, where transport is governed by a solute-diffusion mechanism, and ultrafiltration, where separation is based on sieving effect or, in some cases, charge effects (Shon et al., 2013).

This type of filtration can retain multivalent iones, sugars, dissociated acids, divalent salts and species with larger particle sizes (Fikar and Paulen, 2015). Nanofiltration can also remove alkalinity, thus, the product water may become corrosive. It also removes hardness from the water (Anonymous, 1999). It is mostly used in separation of organic and inorganic compounds, water desalination and softening.

2.4 Reverse Osmosis

Reverse osmosis can effectively remove almost all inorganic components from water. This process operates with the smallest membrane pore size and it requires the highest pressure, of approximately 10 to 100 atmospheres, in order for filtration to take place (Fikar and Paulen, 2015).

Reverse osmosis removes every macro-solute in the water letting only water to pass through the membrane. Hence, it can also retain radium, natural organic substances, pesticides, cysts, bacteria and viruses. This process can be operated at ambient temperature and without any phase change. It is also useful for the processing of thermally and chemically unstable products (Geankoplis, 2008).

In normal osmosis, a semipermeable membrane is placed between water and a salt solution. The membrane allows only water to pass through itself, but not the salt. The difference in water concentration causes water to flow into the salt solution until osmotic equilibrium is reached. (Ravanchi et al., 2009)

In reverse osmosis, a pressure, higher than the osmotic pressure, is applied to the salt solution. Thus, the water flows from the salt solution to the pure water. This phenomenon is called reverse osmosis and it is mostly applied in industry regarding desalination of seawater or brackish water.

2.5 Dialysis

In dialysis, small solutes in one liquid phase diffuse through a porous membrane to the second liquid or vapour phase. This type of process is used to separate species based on their different diffusion rates in the membrane. For this membrane process a counter-current flow is typical, where the feed with solutes to be separated flow on the opposite side of the membrane as the solvent stream.

In general, dialysis is used with aqueous solutions on both sides of the membrane. Solute fluxes depend on the concentration gradient in the membrane, thus, dialysis is characterized by low flux rates than other membrane processes such as nanofiltration and reverse osmosis, which depend on applied pressure (Geankoplis, 2008).

2.6 Membrane Gas Separations

The separation of mixtures of gases is performed to obtain one or more of the constituents in a pure form. Gas separation occurs through a membrane and depends on differences in the permeabilities of various gases through a given membrane. The driving force is the concentration difference.

Gas separation may take place with two types of membranes, porous and non-porous. The transport mechanisms through these two membranes are different. Based on the type of membrane used to separate the mixture of gases two main approaches are considered.

In a porous membranes a gas diffusion occurs based on Knudsen flow mechanism. The rates of molecular diffusion of various gas molecules depend on the pore size and the molecular weights (Ravanchi et al., 2009).

In nonporous membrane the solute gas first dissolves in the membrane and then diffuses in the solid to the other gas phase on the other side of the membrane. This separation involving nonporous membrane follows Fick's law of diffusion (Geankoplis, 2008).

2.7 Pervaporation

Pervaporation is a membrane separation process in which a liquid mixture is separated by partial vaporization through a dense non-porous membrane. The membrane acts as a barrier between the feed in liquid phase and the permeate in vapour phase. This process involves the liquid-vapour phase change to achieve the separation, which occurs due to the low partial pressure at the permeate side (Wilson, 2000).

The driving force of this process is the vapour pressure difference between the feed solution and the permeate. A vacuum is kept on the permeate side of the membrane while the feed side of the membrane is kept at atmospheric or elevated pressure to maintain the driving force for the process. The desired component in the feed stream is selectively permeated by the membrane and removed as a vapour from the other side of the membrane.

Pervaporation can effectively separate solutions, mixtures of components which have close boiling points or azeotropes that are difficult to separate by other means (Basile et al., 2015).

2.8 Industrial applications

Most of the pressure driven membrane processes such as microfiltration, ultrafiltration and reverse osmosis are well established processes in industrial applications. While other membrane separation processes including gas separation and pervaporation are in development, where only a number of plants have been installed so far, but their application is expected to expand in the future (Ravanchi et al., 2009). Main applications of membrane processes are listed below.

Microfiltration The main applications consist of disposable cartridges for sterile filtration of water for the pharmaceutical industry and final point-of-use polishing of ultrapure water for the electronics industry, cold sterilization of beer, wine and other beverages where microfiltration cartridge removes all yeast and bacteria from the filtrate (Wilson, 2000). Other typical applications involve filtration of microorganisms as heat-free sterilisation, and water and waste water treatment.

Ultrafiltration Major application is electrocoat paint mechanism, where ultrafiltration removes ionic impurities from the paint tank containing paint emulsion where metal parts are immersed. Other applications may involve concentration of milk whey in the food industry to recover milk proteins and to remove lactose and salts in the membrane filtrate, and the concentration of oil emulsions in the metal finishing industry. Ultrafiltration is not very common in industry for treatment of waste streams due to being expensive.

Nanofiltration It is mostly used in separation of organic and inorganic compounds, water desalination and softening, and production of potable water.

Reverse osmosis Among well-established applications are production of drinking water, desalination of seawater and brackish water to produce potable water, production of ultrapure water in electronics industry. Other applications may involve water softening, dewatering and water purification for medical use and food processing (Fikar and Paulen, 2015; Wilson, 2000).

Dialysis This process finds application mostly in pharmaceutical and biotechnology industry and it is primarily used as an artificial kidney, a replacement for body's waste disposal in people with kidney failure. It can also be used as artificial lungs in order to oxygenate blood to maintain patients during heart surgery. Recovery of sodium hydroxide in cellulose processing, recovery of acids from metallurgical liquors, removal of products from a culture solution in fermentation, desalting of cheese whey solids and reduction of alcohol concentration in beer are among other applications of dialysis (Geankoplis, 2008).

Pervaporation There are three major applications of pervaporation, and they are dehydration of solvants, water purification and organic/inorganic separations as an alternative to distillation. The only process installed on a large scale is dehydration of ethanol and isopropanol to produce dry alcohol. The other commercial application is the separation of dissolved VOCs from industrial wastewater streams and the recovery of volatile flavour and aroma components in the food processing industry.

Gas separation Applications of membrane gas separation are still in development. The largest gas separation process in use is the production of nitrogen from air. Another industrial applications involve separation of hydrogen from gases like nitrogen and methane, and natural gas separations and air dehydration and oxygen enrichment. A growing application is the removal of condensable organic vapours from air and other streams (Wilson, 2000).

Chapter 3

Membrane Contactors

Membrane contactors (MCs) are membrane systems, where two phases are separated by a membrane allowing their contact without any dispersion of one phase within another. Membrane contactors can be classified as a membrane system where a mass transport between two contacting phases is carried out by diffusion across the interface. Based on different types of membrane contactor phases and driving force the main membrane contactor systems are listed in Tab. 3.1, with their respective driving forces and phases (Drioli et al., 2005).

In all types of membrane contactors the species to be removed encounter resistance during its passage from one phase to another. However, it is not always important and it can be neglected. The membrane separation is integrated with a phase contacting operation like extraction for liquid-liquid phases or absorption/desorption for gas-liquid phases. Membrane contactors also provides higher contact areas and large flow ratio differences can be used for the two phases, thus avoiding flooding or foaming problem (Baker, 2004; Gabelman and Hwang, 1999). Membrane contactors are typically made as a shell-and-tube device with microporous capillary hollow-fibre membranes, which have sufficiently small pores to prevent direct mixing of the two phases.

This membrane processes have been of interest in chemical, petrochemical, pharmaceutical and galvanic industry. Industrial applications of membrane contactors consist of delivery or removal of gases from liquids, deoxygenation of ultrapure water for the electronics industry, treatment of boiler feed water for power plants, selective removal of heavy metals from a galvanic process bath and ammonia product recovery, adjustment of carbonation levels in beverages, and deoxygenation of beer to preserve flavour. New applications are in development such as removal of water and CO_2 from natural gas by using membrane gas absorption, or separation of gaseous olefin/paraffine mixtures by absorption.(Wilson, 2000)

In the next sections process model of dialysis is described as a special case of membrane contactors. Dialysis membrane module resemble in operation, in construction and in local variation in the driving force to the hollow fiber membrane contactors (Dindore and Versteeg, 2005; Skogestad, 2009). It can be considered a liquid – liquid membrane contactor using dialysis liquid to extract the urea.

Table 3.1: Overview of membrane contactors

Membrane Contactor	Phase 1	Phase 2	Driving force
membrane strippers/scrubbers and membrane extractors	gas/liquid	liquid	concentration
supported liquid membranes	gas/liquid	gas/liquid	partial pressure or concentration
membrane distillation	liquid	liquid	partial pressure
osmotic distillation	liquid	liquid	partial pressure
membrane emulsifiers	liquid	liquid	pressure
phase transfer catalysis	liquid	liquid	concentration

Dialysis is a well-spread process, where in a counter-current flow blood is being purified by introducing a dialysis liquid which removes all impurities from the blood. During the process urea fluid is transferred from the blood through a semi-permeable membrane to dialysis liquid, poor of urea liquid (Fig. 3.1). The salt content is balanced on each side to avoid any driving force due to concentration difference of these components.

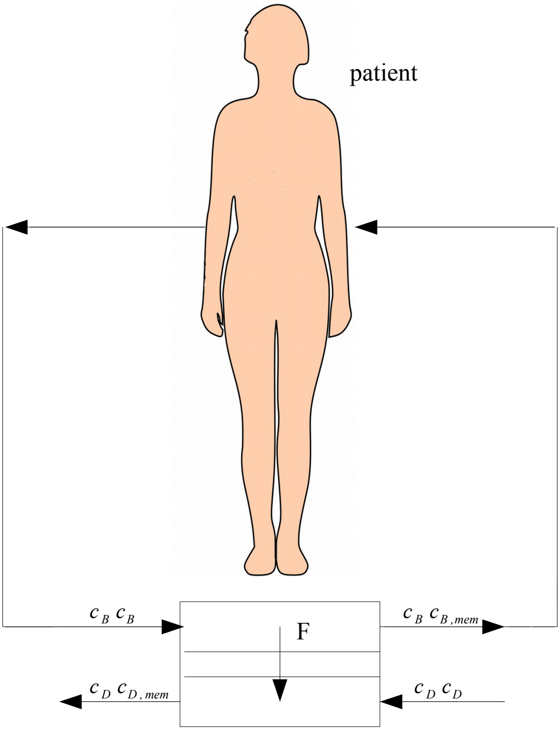

Figure 3.1: Dialysis process

3.1 Process Model

Basic equations for a counter-current dialysis process are obtained from mass balance based on mass change of streams. The process model consists of three equations in total, which describe mass exchange occurring in the patient and in the streams within the membrane module.

3.1.1 Mass Transfer

The mass transfer occurring through the membrane can be described using Ficks's law of diffusion:

$$F = kA \triangle \bar{c} \tag{3.1}$$

where F [mol/min] stands for amount of urea transferred from the blood, k [m/min] is the mass transfer coefficient, which is assumed to be constant, A [m^2] is the area of the membrane, and $\triangle\bar{c}$ [mol/m^3] is the concentration difference across the membrane which represents the driving force.

In general, the concentration difference varies with the position through the membrane module. Thus, in order to correctly describe the mass transfer the logarithmic mean difference is used (Skogestad, 2009). The log mean driving force in Eq. 3.1 is given by

$$\triangle\bar{c} = \frac{\triangle c_1 - \triangle c_2}{ln(\triangle c_1 / \triangle c_2)} \tag{3.2}$$

where the local concentration differences $\triangle c_1, \triangle c_2$ at the two ends of the membrane module are respectively given by

$$\triangle c_1 = c_{B,mem} - c_D \tag{3.3}$$

$$\triangle c_2 = c_B - c_{D,mem}. \tag{3.4}$$

where c_B [mol/m^3] denotes concentration of urea in blood, $c_{B,mem}$ [mol/m^3] is the urea concentration in the exiting blood, c_D [mol/m^3] stands for concentration of urea in dialysis fluid, $c_{D,mem}$ [mol/m^3] is the urea concentration in the exiting dialysis fluid.

3.1.2 Dynamic Model

The mass balance for the patient can be formulated using one differential equation to described the accumulation occurring in his body. In order to derive dynamic mass balance several assumption were taken into considerations.

There is no generation and no conversion of urea by chemical reactions in the body, volume V of body fluids is constant, and urea is evenly distributed in all body fluids. With these assumptions taken into account the mass balance for the patient is as follows

$$\frac{\mathrm{d}n}{\mathrm{d}t} = \dot{n}_{in} - \dot{n}_{out} \tag{3.5}$$

Every element in previous equation can be expressed as

$$n = Vc_B \ [mol] \tag{3.6}$$

$$\dot{n}_{in} = q_B c_B \ [mol/s] \tag{3.7}$$

$$\dot{n}_{out} = q_B c_{B,mem} \ [mol/s] \tag{3.8}$$

where q_B is the blood flow.

Thus, the mass balance for patient can be expressed as

$$V\frac{\mathrm{d}c_B}{\mathrm{d}t} = q_B(c_B - c_{B,mem}). \tag{3.9}$$

3.1.3 Steady-state Model

Under assumption of having small mass of both blood and dialysis fluid it is possible to neglect the accumulation of urea occurring in the membrane module. Thus, the mass balance for the membrane unit is valid for steady state, where there is no accumulation (Skogestad, 2009).

A mass balance around blood and dialysis fluid stream of the membrane module yields for blood stream

$$q_B c_B = F + q_B c_{B,mem} \tag{3.10}$$

and for dialysis fluid stream

$$F + q_D c_D = q_D c_{D,mem} \tag{3.11}$$

where q_D is the dialysis fluid flow.

Chapter 4

Optimization of Membrane Contactor Networks

In this chapter optimization of membrane contactor networks is discussed. As it was mentioned in previous chapter, the model of membrane contactor is based on dialysis process, as a special case of membrane contactor.

An optimization problem consists of maximizing or minimizing an objective function, which is subject to some equality and inequality constraints. The constraints are conditions described by some equations that must always be true no matter what the solution is. These constraints may include requirements for the product, constraints on manipulated variable as to have a nonzero flow or other operational limitations (Skogestad, 1999). The optimization problem can be defined as

$$\min_{u} \; J(u,x,d) \tag{4.1}$$

$$s.t. \; g(u,x,d) = 0 \tag{4.2}$$

$$h(u,x,d) \leq 0 \tag{4.3}$$

where J stands for the objective function, u denotes degrees of freedom, x represents the state variables and d stands for disturbances. Some of inequality constraints h may be active constraints, thus, they must be equal to zero. If these active constraints are controlled with a corresponding number of degrees of freedom and the states x are eliminated using the model equations g the problem becomes an unconstrained optimization problem

$$\min_{u} \; J(u,d) = J\big(u_{opt}(d), d\big) = J_{opt}(d) \tag{4.4}$$

where u_{opt} are degrees of freedom to be found and J_{opt} denotes the optimal value of the objective function. Then, the ideal controlled variable is the gradient, which must be controlled to zero for optimality (Jäschke and Skogestad, 2014),

$$c = J_u = \frac{\partial J}{\partial u} = 0. \tag{4.5}$$

4.1 Process Description

In this section a membrane contactor network is studied. In general, a membrane contactor network may consist of N parallel lines and on each line there are M number of membrane contactors, as it is illustrated in Fig. 4.1.

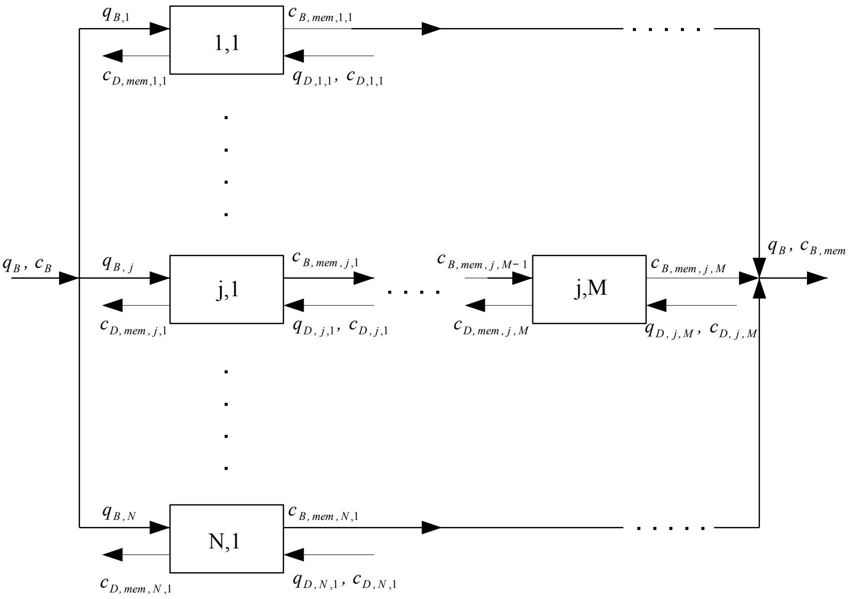

Figure 4.1: A membrane contactor network

A common feed stream q_B is split into several streams, which are filtrated by membrane contactors on each branch. The objective is to distribute the streams in such way that the total mass transfer is maximized (Jäschke and Skogestad, 2014).

For general membrane contactor network the membrane contactor equations in chapter 3 are used and the notation is adjusted, thereby for one given contactor the mass transfer equation is

$$F_{i,j} = kA_{i,j} \triangle c_{i,j} \tag{4.6}$$

where $kA_{i,j}$ is the respective kA value for i-th membrane contactor on branch j. Total mass balance of the system can be expressed as

$$q_B - \sum_{j=1}^{N} q_{B,j} = 0 \tag{4.7}$$

where the feed concentration remains always the same

$$c_B = c_{B,j} \text{ for all } j. \tag{4.8}$$

The $N-$th flow rate is given by the mass balance as

$$q_{B,N} = q_B - \sum_{j=1}^{N-1} q_{B,j} \tag{4.9}$$

obtaining only $N-1$ streams that can be adjusted independently. These $N-1$ streams are also the degrees of freedom, denoted as

$$u = [q_{B,1}, q_{B,2}, \dots q_{B,N-1}]^T \tag{4.10}$$

All the streams after passing through membrane contactors are merged together forming one exiting stream with the end concentration. In order to calculate the end concentration obtained from the network the overall mass balance is used. Thus, the end concentration can be calculated as weighted sum of concentrations of individual streams,

$$c_{B,mem} = \frac{1}{q_B} \sum_{j=1}^{N} q_{B,j} c_{B,j}. \tag{4.11}$$

Adjusting the notation for mass balances (Eq. 3.10, 3.11) the equality constraints for parallel membrane contactor network of N branches, illustrated in Fig. 4.2, consists of model equation derived for each unit and equations for mass balance of the system,

$$g = \begin{pmatrix} Fi,j - kA_j \triangle c_j \\ Fi,j - q_{B,j}(c_B - c_{B,mem,j}) \\ Fi,j - q_{D,j}(c_{D,mem,j} - c_{D,j}) \end{pmatrix} = 0 \tag{4.12}$$

To ensure that log mean difference is always positive and greater than zero inequality constraints h must be formulated. It is also important to ensure positive differences between the inlet blood concentration and the outlet blood concentration and positive difference for dialysis fluid exiting stream and inlet concentration. Therefore the inequality constraints are as follows

$$h = \begin{pmatrix} c_{B,j} - c_{B,mem,j} \\ c_{D,mem,j} - c_{D,j} \\ 1.01 - \frac{c_{B,mem,j} - c_{D,j}}{c_{B,j} - c_{D,mem,j}} \end{pmatrix} \geq 0. \tag{4.13}$$

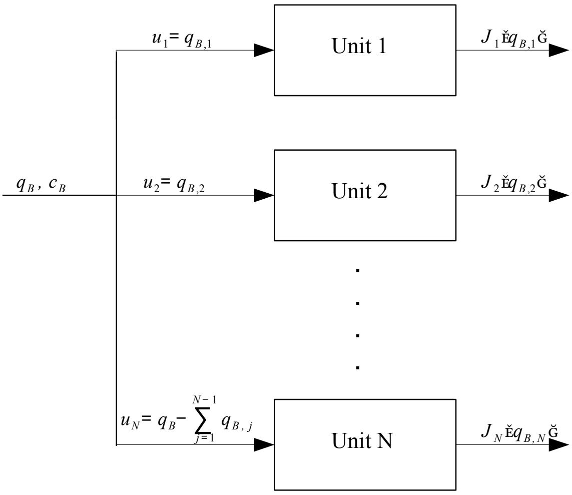

Figure 4.2: Parallel membrane contactor network

4.2 Optimization

A parallel membrane contactor network with common feed concentration c_B is shown in Fig. 4.2. The objective function for such network is to minimize total operating cost J, which can be expressed as a sum of the individual costs J_j of each branch,

$$J = \sum_{j=1}^{N} J_j(q_{B,j}) \tag{4.14}$$

This can be achieved by optimal distribution of the streams $q_{B,j}$ between the branches.

$$\min_u \; J = c_{B,mem} \tag{4.15}$$

$$s.t. \; g = 0 \tag{4.16}$$

$$h \leq 0 \tag{4.17}$$

In order to formulate the optimality condition for parallel membrane contactor network $N-1$ degrees of freedom are chosen as the flows in respective branches. A flow change in j branch is compensated by a change of flow in the last branch N (Jäschke and Skogestad, 2014),

$$\delta q_{B,N} = -\delta q_{B,j}, \quad \text{where } j \neq N \tag{4.18}$$

Thus, there is no flow change in other branches. and the change δJ in the cost for a variation in $\delta q_{B,j}$ can be described as

$$\frac{\delta J}{\delta q_{B,j}} = \frac{\delta(J_1 + \cdots + J_j + \cdots + J_N)}{\delta q_{B,j}} = \frac{\delta(J_j + J_N)}{\delta q_{B,j}} \tag{4.19}$$

By applying equation 4.18 the change of total operating cost becomes

$$\frac{\delta J}{\delta q_{B,j}} = \frac{\delta J_j}{\delta q_{B,j}} - \frac{\delta J_N}{\delta q_{B,N}} \tag{4.20}$$

The equation (Eq. 4.20) holds for all degrees of freedom, thus, the marginal costs for all branches $\delta J_j/\delta F_j$ are equal (Downs and Skogestad, 2011). By denoting the degrees of freedom as the streams (Eq. 4.10), this leads to formulation of the optimality condition and to the optimal controlled variables expressed as

$$\frac{\partial J_1}{\partial u_1} = \cdots \frac{\partial J_j}{\partial u_j} \cdots = \frac{\partial J_N}{\partial u_N}, \tag{4.21}$$

$$c = \begin{pmatrix} \frac{\partial J_1}{\partial u_1} - \frac{\partial J_N}{\partial u_N} \\ \vdots \\ \frac{\partial J_j}{\partial u_j} - \frac{\partial J_N}{\partial u_N} \\ \vdots \\ \frac{\partial J_{N-1}}{\partial u_{N-1}} - \frac{\partial J_N}{\partial u_N} \end{pmatrix}. \tag{4.22}$$

4.2.1 Objective function

The goal of membrane contactor network is the optimal adjustment of the manipulated variables u, which are the splits between j number of branches

$$u_j = \frac{q_{B,j}}{q_B} \tag{4.23}$$

By optimally controlling the splits the operating cost must be minimized.

$$J = -(\text{income} - \text{expenses}) \tag{4.24}$$

As different sources of streams may have different prices, Jäschke has proposed a cost function, which takes into consideration all the prices. Therefore, the objective function ca be rewritten as

$$J = -\sum_{j=1}^{N}\sum_{i=1}^{M} p_{i,j}F_{i,j} \tag{4.25}$$

where $p_{i,j} = p_{i,j}^{rev} - p_{i,j}^{cost}$ is the difference of prices for each unit in the network. If the prices of the streams are equal, the objective function is maximizing the total mass transfer. If all the streams are blended together, maximizing total mass transfer corresponds also to minimizing the end concentration of the final stream (Eq. 4.11)(Jäschke and Skogestad, 2014).

4.3 Self-Optimizing Control

Self-optimizing control is introduced as a simple method to optimize operation of a heat exchanger network. This is achieved by controlling the temperature difference to zero and obtaining a near-optimal operation for heat exchange network. This control structure was proposed by Johannes Jäschke, and it was primarily used with heat exchanger network for which the objective function was defined as maximization of the end temperature. The derivation and proof of this approach can be viewed in (Jäschke and Skogestad, 2014).

In this work the self-optimizing control is applied to network of membrane contactors where the objective function is minimization of the end concentration.

The goal is to find a set of controlled variables c,

$$c = J_u = \frac{\partial J}{\partial u} \tag{4.26}$$

which when kept constant at the optimal values c_{opt}. In the ideal case this would lead to optimal operation of the network. However, there is a loss associated with keeping constant the controlled variables at their optimal values. Thus, the operation achieved is close to optimum. The loss is expressed as

$$L(u,d) = J(u,d) - J(u_{opt}(d),d) \tag{4.27}$$

These controlled variables are functions of measurements that are easy to obtain, and are equivalent to controlling the gradient to zero. Johannes Jäschke introduced difference of Jäschke temperatures as such controlled variables, which are used for heat exchange network. Since membrane contactor network and heat exchange network are similar in design, such controlled variables can also be applied to membrane contactor network. By applying this control to the membrane contactor network it is possible to reach a near-optimal operation by only manipulating the split u in the network.

In order to be able to use self-optimizing control some assumptions must be taken into consideration. All prices are equal to 1, and the driving force for mass transfer is given by arithmetic mean difference. The Jäschke temperatures can be expressed as the marginal costs for each branch

$$T_{J,j} = \frac{\partial J}{\partial q_{B,j}}, \qquad j = 1, \ldots, N \tag{4.28}$$

The Jäschke temperature are defined as

$$T_{J,j} = \sum_{i=1}^{M_j} a_{i,j} \tag{4.29}$$

where the parameter $a_{i,j}$ is defined recursively as

$$a_{i,j} = \frac{(\theta_{i,j} - \theta_{i-1,j})(\theta_{i,j} - \theta_{i-1,j} - a_{i-1,j})}{\theta_{hi,j} - \theta_{i-1,j}}, \qquad a_{0,j} = 0 \tag{4.30}$$

The parameter $\theta_{i,j}$ stands for shifted concentration, obtained as follows

$$\theta_{i,j} = c_{B,i,j} - c_B \tag{4.31}$$

In the case of arbitrary prices $p_{i,j}$, the marginal costs for each branch j is defined as

$$\frac{\partial J}{\partial q_{B,j}} = T^e_{J,j} \tag{4.32}$$

where Economic Jäschke temperature $T^e_{J,j}$ can be expressed as

$$T^e_{J,j} = \sum_{i=1}^{M_j} p_{i,j} a_{i,j}. \tag{4.33}$$

This method offers some advantages over conventional optimization methods. It is exclusively based on simple concentration measurements and there is no need to know flow rates of the streams, nor are required mass transfer properties to calculate the Jäschke temperatures.

4.4 NTU method

The number of transfer units (NTU) method is based on calculation of efficiency for heat exchangers in steady state. It is mostly used when there is insufficient information to calculate the logarithmic mean temperature difference. Since there is analogy between the process model of membrane contactor defined in section 3.1 and a heat exchanger (Skogestad, 2009), this method can also be applied for this specific process model.

This method offers formulas to calculate the concentration of streams exiting from the membrane module. In order to derive these concentrations, efficiency for each membrane unit i must be calculated. By changing the notation used in (Skogestad, 2009) so it can be applied for membrane process, the efficiency for blood stream can be expressed as

$$\epsilon_{B,i} = \frac{1 - \exp(N_{tu,i}(C_i - 1))}{C - \exp(N_{tu,i}(C_i - 1))} \tag{4.34}$$

while the efficiency for dialysis liquid is

$$\epsilon_{D,i} = \epsilon_{B,i} C_i \tag{4.35}$$

where parameters N_{tu} and C are calculated from the following formulas

$$N_{tu,i} = \frac{kA}{q_{B,i}}, \tag{4.36}$$

$$C_i = \frac{q_{B,i}}{q_{D,i}} \tag{4.37}$$

Then the outlet stream concentrations of blood and dialysis liquid are given as follows

$$c_{D,i} = (1 - \epsilon_{D,i})c_{D,i} + \epsilon_{D,i}c_B \tag{4.38}$$

$$c_{B,i} = \epsilon_{B,i}c_{D,i} + (1 - \epsilon_{B,i})c_B \tag{4.39}$$

The NTU method yields a linear relationship between the inlet and outlet concentration.

In singular case when $C_i = 1$ the efficiency must be corrected by changing slightly the C value or by calculating its values as follows

$$\epsilon_{B,i} = \frac{N_{tu,i}}{1 + N_{tu,i}}. \tag{4.40}$$

Chapter 5

Case Study

The NTU method, optimization and self-optimization were applied to optimize membrane contactor network for three different configurations, which consist of membrane contactors on different branches, and where the streams after being purified are blended together. It is assumed that all the prices are equal on both branches, making this a minimization problem where the objective function is to minimize the end concentration.

This problem was solved in steady state without considering dynamic model of the patient, thus simplifying the whole model to only steady state model of the membrane module. In steady state three different methods were used to obtain the optimal value of the split, as the manipulated variable, in order to achieve the minimal value of the end concentration of urea in the exiting blood stream. The optimal solution of the end concentration was obtained by solving it as a nonlinear optimization problem, by calculating efficiencies for membrane contactors and by using concept of self-optimizing control.

5.1 Two Units in Parallel

A network consisting of two membrane contactors in parallel with a common feed concentration c_B is presented in Fig. 5.1.

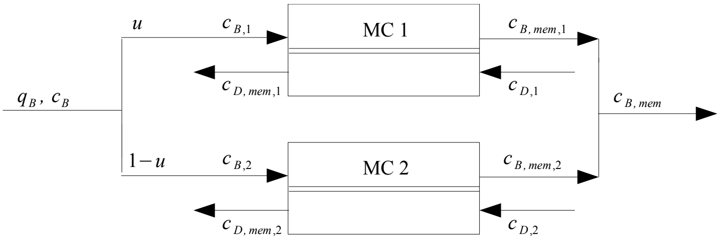

Figure 5.1: Two membrane contactors in parallel

The network parameters for such configuration are listed in Tab. 5.1. Instead of selecting the flows directly, the split is used as manipulated variable. Based on the configuration the the split is defined as ratio between the flow on the first branch and the feed flow

$$u = \frac{q_{B,1}}{q_B} \tag{5.1}$$

Consequently the streams for each branch are defined as

$$q_{B,1} = uq_B \tag{5.2}$$

$$q_{B,2} = (1-u)q_B \tag{5.3}$$

Table 5.1: Parameters of two parallel membrane contactors

Parameter	Value	Unit
c_B	50	[mol/m^3]
q_B	0.30	[l/min]
$c_{D,1}$	5	[mol/m^3]
$c_{D,2}$	3	[mol/m^3]
$q_{D,1}$	0.5	[l/min]
$q_{D,2}$	0.45	[l/min]
kA_1	0.1	[m^3/min]
kA_2	0.2	[m^3/min]

For a membrane contactor network of two parallel units only five measurements are needed to achieve self-optimizing control. These measurements are concentrations of the exiting blood and inlet concentration of dialysis fluid. For each unit of membrane contactor in the network a Jäschke temperature was defined. For this case study two Jäschke temperatures were calculated based on expressions presented in Section. 4.3 (Eq. 4.29, 4.30),

$$T_{J,1} = \frac{c_{B,mem,1} - c_B}{c_{D,1} - c_B} \tag{5.4}$$

$$T_{J,2} = \frac{c_{B,mem,2} - c_B}{c_{D,2} - c_B}. \tag{5.5}$$

Thus, the final self-optimizing controlled variable for this specific case is the difference of two Jäschke temperatures

$$c = T_{J,1} - T_{J,2} = 0 \tag{5.6}$$

Therefore, in the case of self-optimization the steady state performance is obtained by finding the split u at which the self-optimizing variables are all equal to each other.

Optimal operation of this network was determined by using build-in MATLAB function `fmincon`, NTU method and using the Jäschke temperatures. To calculate the optimal split using NTU method the efficiencies were calculated based on the formulas given in section 4.4 for all possible values of the split and then selecting the minimal value of the end concentration $c_{B,mem}$. In MATLAB environment the objective function was formulated as a minimizing problem of the end concentration subject to some equalities and inequalities, which are expressed by the process model, mass balance and positive values of all the variables and positive logarithmic mean difference. The results of optimal operation from optimization were compared to the operation achieved by NTU method and the Jäschke temperatures, and the results are given in Tab. 5.2.

Table 5.2: Results for optimal operation of the network with two parallel MC

Method	$c_{B,mem}$	u
Optimized	22.9906	0.3395
NTU method	22.9848	0.3530
Equal Jäschke Temp.	22.9866	0.3454

The results shown in Tab. 5.2 and in Fig. 5.2 indicates that a near-optimal operation was obtained with Jäschke temperatures, as the outlet concentration $c_{B,mem}$ and the split u from the Jäschke temperatures differ slightly from optimal outlet concentration and optimal split achieved by optimization and NTU method.

In Fig. 5.4 the concentration profiles are shown for both the optimal split and the self-optimizing split. As it is observable in the figure, there is only one point where $c = 0$. At this point near-optimal operation was achieved by using the Jäschke temperatures. For better illustration the difference of Jäschke temperatures was plotted as a function of the split in absolute value.

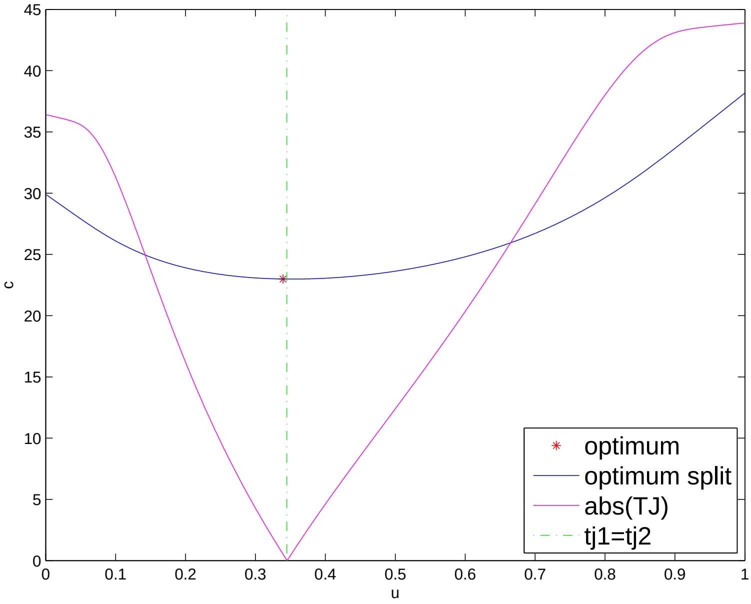

Figure 5.2: Optimal operation of parallel membrane contactor network

5.2 Two Units in Series and One in Parallel

A network with two membrane contactors in series and one in parallel is illustrated in Fig. 5.3 with network parameters given in Tab. 5.3.

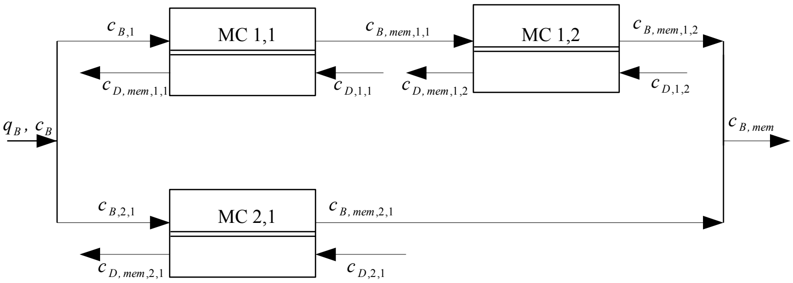

Figure 5.3: Two membrane contactors in series and one in parallel

Table 5.3: Parameters of three membrane contactors

Parameter	Value	Unit
c_B	60	[mol/m^3]
q_B	0.55	[l/min]
$c_{D,1,1}$	4	[mol/m^3]
$c_{D,1,2}$	2	[mol/m^3]
$c_{D,2,1}$	6	[mol/m^3]
$q_{D,1,1}$	0.4	[l/min]
$q_{D,1,2}$	0.5	[l/min]
$q_{D,2,1}$	0.45	[l/min]
$kA_{1,1}$	0.1	[m^3/min]
$kA_{1,2}$	0.2	[m^3/min]
$kA_{2,1}$	0.1	[m^3/min]

In this case too, the objective function is to minimize the end concentration. The same procedure was applied as in previous case to determine the optimal value of the end concentration. Regarding the approach of self-optimizing control new Jäschke temperatures have to be defined for each unit in the network, two for the first branch and one for the second branch,

$$T_{J,1,1} = \frac{c_{B,mem,1,1} - c_B}{c_{D,1,1} - c_B} \tag{5.7}$$

$$T_{J,1,2} = \frac{(c_{B,mem,1,2} - c_{B,mem,1,1})(c_{B,mem,1,2} + c_{B,mem,1,1} - 2c_B - T_{J,1,1})}{c_{D,1,2} - c_{B,mem,1,1}} \tag{5.8}$$

$$T_{J,2,1} = \frac{c_{B,mem,2,1} - c_B}{c_{D,2,1} - c_B} \tag{5.9}$$

Then, the self-optimizing controlled variable for this network is the difference of two Jäschke temperatures for the first branch and one Jäschke temperature for the second branch,

$$c = T_{J,1,1} + T_{J,1,2} - T_{J,2,1} = 0 \tag{5.10}$$

By keeping the controlled variable c constant and equal to zero the maximal end concentration is achieved.

Table 5.4: Results for network of three MC

Method	$c_{B,mem}$	u
Optimized	33.4567	0.7869
NTU method	33.3949	0.7440
Equal Jäschke Temp.	33.4049	0.7621

The simulation results are presented in Tab. 5.4, where the optimal end concentration is given together with the end concentration by using the NTU method and the approach of Jäschke temperatures. As in the previous case, the difference between these three values is very small. It can be observed that the split is also close to the optimum. The outlet concentration $c_{B,mem}$ and absolute value of the controlled variable c were plotted as a function of split $u$and it is illustrated in Fig. 5.4. The MATLAB code for both case studies can be found in Appendix A.

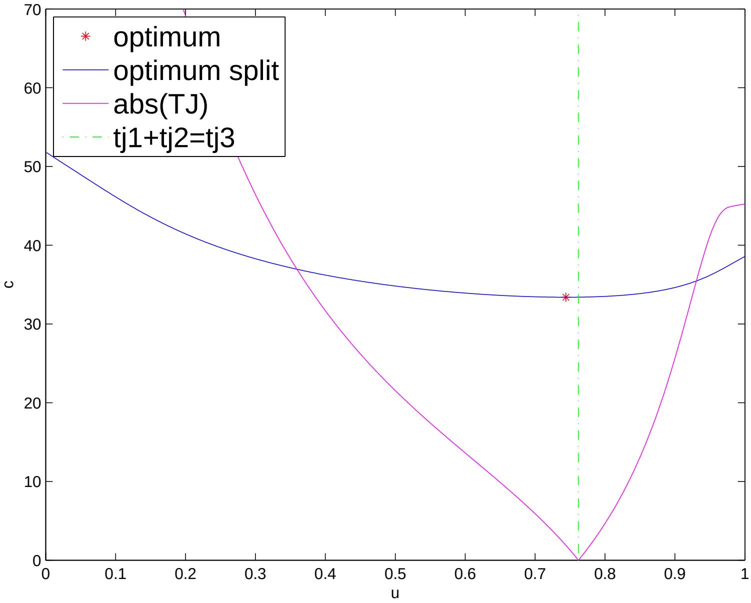

Figure 5.4: Optimal operation of three membrane contactors

5.3 Four Units in Series and Two in Parallel

A network with four membrane contactors in series and two in parallel is shown in Fig. 5.5 with network parameters given in Tab. 5.5.

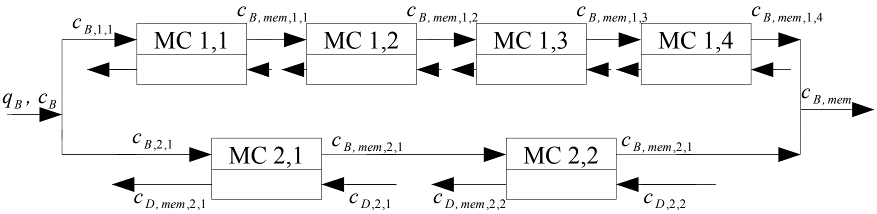

Figure 5.5: Four membrane contactors in series and two in parallel

Table 5.5: Parameters of six membrane contactors

Parameter	Value	Unit
c_B	80	[mol/m^3]
q_B	0.5	[l/min]
$c_{D,1,1}$	2	[mol/m^3]
$c_{D,1,2}$	5	[mol/m^3]
$c_{D,1,3}$	7	[mol/m^3]
$c_{D,1,4}$	3	[mol/m^3]
$c_{D,2,1}$	9	[mol/m^3]
$c_{D,2,2}$	1	[mol/m^3]
$q_{D,1,1}$	0.4	[l/min]
$q_{D,1,2}$	0.3	[l/min]
$q_{D,1,3}$	0.5	[l/min]
$q_{D,1,4}$	0.5	[l/min]
$q_{D,2,1}$	0.35	[l/min]
$q_{D,2,2}$	0.45	[l/min]
$kA_{1,1}$	0.4	[m^3/min]
$kA_{1,2}$	0.2	[m^3/min]
$kA_{1,3}$	0.3	[m^3/min]
$kA_{1,4}$	0.2	[m^3/min]
$kA_{2,1}$	0.1	[m^3/min]
$kA_{2,2}$	0.2	[m^3/min]

For this case study six Jäschke temperatures were calculated as follows,

$$T_{J,1,1} = \frac{c_{B,mem,1,1} - c_B}{c_{D,1,1} - c_B} \tag{5.11}$$

$$T_{J,1,2} = \frac{(c_{B,mem,1,2} - c_{B,mem,1,1})(c_{B,mem,1,2} + c_{B,mem,1,1} - 2c_B - T_{J,1,1})}{c_{D,1,2} - c_{B,mem,1,1}} \tag{5.12}$$

$$T_{J,1,3} = \frac{(c_{B,mem,1,3} - c_{B,mem,1,2})(c_{B,mem,1,3} + c_{B,mem,1,2} - 2c_B - T_{J,1,2})}{c_{D,1,3} - c_{B,mem,1,2}} \tag{5.13}$$

$$T_{J,1,4} = \frac{(c_{B,mem,1,4} - c_{B,mem,1,3})(c_{B,mem,1,4} + c_{B,mem,1,3} - 2c_B - T_{J,1,3})}{c_{D,1,4} - c_{B,mem,1,3}} \tag{5.14}$$

$$T_{J,2,1} = \frac{c_{B,mem,2,1} - c_B}{c_{D,2,1} - c_B} \tag{5.15}$$

$$T_{J,2,2} = \frac{(c_{B,mem,2,2} - c_{B,mem,2,1})(c_{B,mem,2,2} + c_{B,mem,2,1} - 2c_B - T_{J,2,1})}{c_{D,2,2} - c_{B,mem,2,1}} \tag{5.16}$$

and the final self-optimizing controlled variable for this network is

$$c = T_{J,1,1} + T_{J,1,2} + T_{J,1,3} + T_{J,1,4} - T_{J,2,1} - T_{J,2,2} = 0 \tag{5.17}$$

Therefore, it is enough to control this controlled variable to zero in order to achieve a near-optimal operation of the network.

Table 5.6: Results for network of four MC in series and two in parallel

Method	$c_{B,mem}$	u
NTU method	12.7697	0.7530
Equal Jäschke Temp.	12.8472	0.8232

The simulation results are presented in Tab. 5.6, where the end concentration is compared between the NTU method and the approach of Jäschke temperatures. The end concentration obtained from NTU method is similar to the one obtained from using the Jäschke temperatures. However, split achieved with the Jäschke temperatures is distinctly different. This might be due to optimum being flat or arithmetic mean difference failed to approximate the logarithmic mean difference causing loss in the values greater than in the previous cases.

Chapter 6

Mass exchange networks

This chapter introduces the notion of synthesizing mass-exchange-networks (MEN's). MEN is a separation system which can separate the feed streams into several product streams of known conditions with a minimum operating costs. This chapter addresses the concept of MEN since a membrane separation process can also be viewed as a mass exchanger unit and an example of application of MEN is presented where a membrane process is used in hydrogen recovery from a purge stream.

The mass exchange network is a system consisting of several direct-contact mass transfer units operating in counter-current arrangement, where a mass separating agent, called MSA (mass-separation-agent) or a lean stream is used in order to selectively remove certain components from a rich stream. A rich stream is the stream from which the targeted components are removed, while the lean stream is the stream to which the targeted component is being transferred, also called MSA (El-Halwagi, 2006). The process is shown in Fig. 6.1

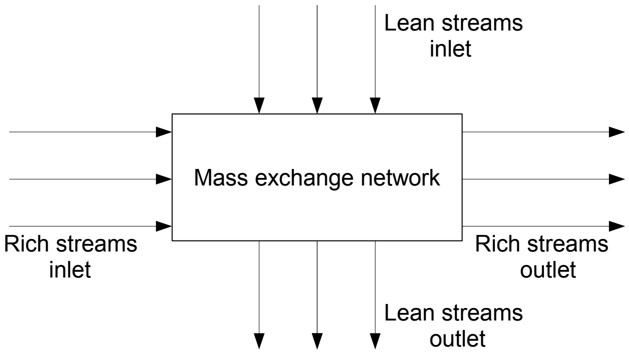

Figure 6.1: Mass Exchange Network

Examples of mass-separation-agent processes are distillation, extraction, leaching, evaporation, desorption, and more.

The goal of using mass exchange network is to minimize the network cost, where separation units are used to adjust the composition of streams in order to reach environmental or process requirements.

When synthesizing a MEN network design targets must be established. One of the targets is the minimum cost of the MSA's to be used, which influences the operating cost of the network. The other design target is minimum number of mass exchanger units. This target determinates the minimum number of mass exchanger units which are required in the network. Thus, this also affects the costs of the network and maintenance, instrumentation and pipework (El-Halwagi and Manousiouthakis, 1989).

In order to synthesize MEN information must be given with the list of streams to be integrated with their respective flows and inlet and outlet concentrations. The problem of synthesizing a MEN can be described as finding the optimal mass exchange network which will selectively transfer targeted species from the rich streams to the lean ones in order to meet environmental or process requirements. There has been a lot of work done in the field of solving the MEN synthesis.

The first procedure for optimal synthesis of MEN was introduced by (El-Halwagi and Manousiouthakis, 1989), which is based on the heat transfer pinch analysis. In pinch analysis the task is to find pinch point between mass transfer composite curves where is the smallest driving force.

This is analogous to the minimum allowable temperature difference in heat exchanger network. The pinch point defines the minimum driving force required for the unit. Pinch analysis can be used to predict targets for minimum number of units and minimum utility (MSA) which are required. Other methods for solving MEN synthesis are sequential optimization and simultaneous optimization methods. The pinch analysis is applicable to MEN because of analogy between heat and mass transfer. The analogy between these two networks, MEN and HEN is given in Tab. 6.1 (Szitkai, 2004).

Table 6.1: Analogy between heat and mass exchange networks

Concept	MEN	HEN
Transported material	mass	heat
Driving force	concentration difference	temperature
Feed	rich streams	hot streams
Sink	lean streams	cold streams

MEN is applicable in chemical process industry in the cases such as feed preparation, product separation, product finishing or in recovery of valuable materials. There are several industrial applications involving MEN, such as cleaning wastewater and flue gases from incinerators (Korovessi and Linninger, 2005).

6.1 Application of MEN in Hydrogen Recovery Process

One of possible applications of mass exchange networks is hydrogen recovery from the purge streams of an hydrodealkylation of toluene (HDA) . This process is used for production of benzene by using hydrodealkylation of toluene. The HDA process designed by J. Douglas (Douglas, 1988) contains several basic unit operations interconnected by process streams. These units are reactor, furnace, separator, compressor, heat exchangers, distillation columns.

There are two input substances for the chemical reactor, toluene and hydrogen which inside the reactor react together creating three products, benzene, diphenyl and methane. The outlet stream from the reactor contains except the three products also hydrogen and toluen which are subsequently recycled back and mixed with fresh toluen and hydrogen before entering the reactor.

It is necessary that hydrogen is fed to the reactor in excess in order to complete the reaction. Therefore, hydrogen is found in abundance in the reactor outlet stream together with the products and must be removed.

The goal of implementing MEN is to recover part of hydrogen available in the purge stream by resorting to a counter-current gas permeation units. Applying MEN in the design procedure leads to integrating streams entering and exiting of the process. Thus, the gas permeation units serves to exchange hydrogen between the purge stream exiting the reactor and the toluene inlet stream entering the reactor (Qiu et al., 2003). Fig. 6.2 illustrates implementation of MEN to recover part of the hydrogen in simplified schema of the reactor.

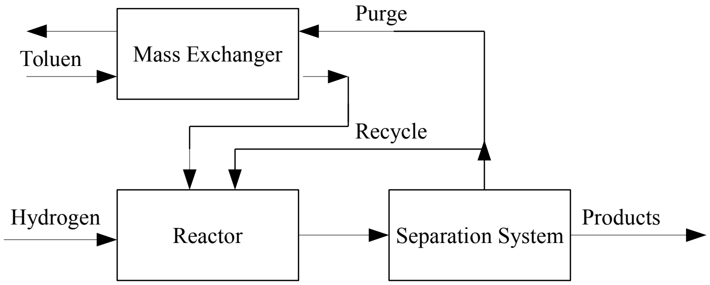

Figure 6.2: Implementation of MEN in the HDA process

The main advantage of implementing a counter-current mass integration into the HDA process is the reduction in the amount of hydrogen required by the process. HDA process with the mass integration does not need any recycle compressor to recycle the recovered hydrogen as it was the case of the original process design (Fischer and Iribarren, 2014).

Membrane technology such as gas permeation in the case of HDA process for hydrogen recovery is a very active area of research and development. More real-world processes, other than academic HDA process, could benefit from using the MEN heuristics in process design.

Chapter 7

Membrane Contactor for CO_2 and H_2S Removal from Natural Gas

In this chapter a possible case study is presented where self-optimizing control can be implemented for membrane contactor network. This case study is a system proposed for CO_2 and H_2S removal from natural gas.

Natural gas is an important and widely used energy source. It consists primarily of methane including some higher alkanes and a certain percentage of carbon dioxide, nitrogen, hydrogen sulfide and other sulfur components. Before it can be used as a energy source or as a fuel for cars it must be processed. By processing raw natural gas impurities are removed such as water, CO_2 and H_2S.

The removal of CO_2 is an important industrial application. It is essential to enhance the heating value of the gas while H_2S is very toxic and corrosive, thus, it must be removed from natural gas (Marzouk et al., 2012). These acid gases can be removed from gas mixtures by absorption technology, pressure swing adsorption and also by utilizing membrane technology which is frequently applied for this high-pressure separation. This process is conventionally done by using column absorption, but it has problems such as flooding and foaming and it is energy consuming and hard to operate. However, the use of membrane gas absorber process, or more specifically a gas-liquid membrane contactor, is a good alternative. They can absorb the by-products CO_2, H_2S and they can withstand high pressure without contaminating the gas products by the liquid phase and providing large contact area (Li and Chen, 2005).

CO_2 capture using membrane separation process can be accomplished by pre-combustion and post-combustion applications. In the study case, CO_2 removal is accomplished in the post-combustion stage using a gas-liquid membrane contactor which is in counter-current arrangement.

The separation itself occurs by semi-permeable selective membrane that separates gas from liquid phase and it let pass only CO_2 and H_2S and retains methane.

The gases flowing in one direction are absorbed by the liquid flowing in opposite direction on the other side of the membrane. This process is a concentration driven process which is directly related to the pressure of the gas inlet and permeates streams.

A salt in the liquid phase, also called ionic liquid, might be used as the absorption liquid in order to increase the selectivity of the system. Compared to standard amine solvents ionic liquids provide higher stability, lower flue gas losses, lower costs and lower environmental impact.

7.1 Process Description

The configuration proposed for removal of acid gases from natural gas is illustrated in Fig. 7.1.

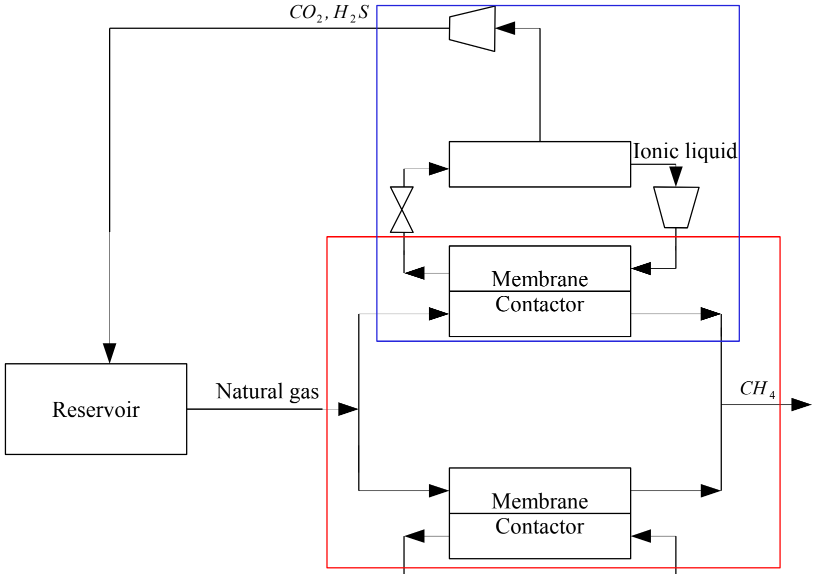

Figure 7.1: A scheme of membrane contactors for CO_2 and H_2S removal

The process introduced in this study case consists of two membrane contactors which are connected in a network to effectively remove acid gases from natural gas. After combustion natural gas enters membrane contactor network at high pressure where methane is purified from acid gases. The product streams from both membrane contactors are mixed together at the end of the process. Meanwhile, acid gases are removed by being absorbed into ionic liquid in gas-liquid membrane contactors. Afterwards, they are desorbed and reinjected back to the reservoir. Then, the ionic liquid is compressed and sent back to the membrane contactor to be reused in another separation. The process of absorption and desorption is based on pressure swing principle (Simons, 2010). At high pressure acid gases are absorbed and when the pressure is reduced the gases are released.

The red block corresponds to the case study consisting of two membrane contactors connected in parallel. These contactors are gas and liquid membrane contactors where ionic liquid is used to absorb gas from the stream based on pressure swing principle. Thus, the process model described in section 3.1 is not accurate. However, the process model for such process is still in development.

The blue block illustrated in the figure shows a another case study to be examined which consists of membrane contactor, compressor, and a valve. This case is interesting in commercial point of view where the goal is to optimize operating costs of the system.

Chapter 8

Discussion and Further Work

The concept of self-optimizing control was studied for membrane contactor network in steady state. The method of self-optimizing control is very simple and easy to implement. It also gives a performance close to optimum. In order to derive the Jäsche temperatures it is assumed that the arithmetic mean difference of concentrations serves as the driving force for the process. However, in cases when the arithmetic mean difference failed to approximate the logarithmic one the Jäschke temperature showed worse performance. The difference between the optimal value and the one achieved with the Jäschke temperature differs distinctly.

In this work only parallel networks were investigated, but when designing membrane networks it can be desirable to utilize each source to maximum to achieve best separation. This implies a crossover configuration of the network where the dialysis liquid streams should also be optimally distributed throughout the network. The parallel configuration of networks might not be the best, therefore more branches or and a bypass regulation might be desired. Thus, the Jäschke temperature control should be studied for more complex configurations.

Since this work is done only for steady state model of membrane contactor, performance should also be examined for other membrane separation processes and their different models which will include not only steady state analysis but also dynamic analysis.

Another issue which was not taken into consideration was study of different price constants. All prices in the case studies were assumed to be equal to one, consequently simplifying the problem from optimizing the operating cost of the network to minimizing the end concentration. Therefore, optimal operation with different prices needs to be included in the future work where self-optimizing control approach is applied to larger settings, and where the plant would be economically optimized in order to minimize the operating costs.

As for the future work, the case of self-optimizing control of membrane contactor network with disturbances might be interesting in order to examine its performance.

Another direction for future work is to investigate possible connections with MEN to established methods for membrane network design, such as pinch analysis used with MEN.

As for the possible case study described in section 7 where only two membrane contactors are considered to form the network for removal of acid gases from natural gas a proper model for a gas-liquid membrane contactor should be used and so the self-optimizing control could be performed. Another interesting issue to be studied can be the process itself of acid removal, compressing gases and the ionic liquid which is sent back to the membrane to perform another separation process. This problem should be optimize in order to minimize the operating costs.

Chapter 9

Conclusion

The aim of this work was to model and to control a membrane contactor network for which three different configurations were studied. All these network configurations were examined for steady state behaviour. To achieve maximum mass transfer between two fluids in the steady state model different methods were used: the NTU method, optimization and the Jäschke temperatures as a part of self-optimization control approach. The NTU method solved the problem by calculating efficiency for each membrane unit. For optimization the problem of a membrane network was stated as a nonlinear optimization problem and solved in MATLAB environment using `fmincon` solver. The last method uses the concept of self-optimization which was proposed by Johannes Jäschke, originally to control the heat exchanger network. The idea of self-optimization is to find a set of controlled variables c which when kept constant it leads to an operation close to optimum with acceptable loss.

In the studied cases the loss, related to Jäschke temperature results compared to the ones achieved by optimizing and NTU method, was different based on the configuration and parameters of the network. All of these methods used shows different results based on the algorithm they utilize. However, in two configurations a near-optimal operation was achieved while in the last configuration with six membrane contactors connected together there is a noticeable difference between the value obtained from the Jäschke temperature and the NTU method. This might have been caused by failure of arithmetic mean difference to approximate the logarithmic one.

The advantage of the Jäschke temperature approach is that the control variable is only dependent on concentration measurements, where the split u serves as the only manipulated variable. For this method arithmetic mean difference is assumed which in certain cases might fail to approximate the logarithmic mean difference and thus, causing the performance to degenerate.

Thanks to the analogy between the process model and the heat exchanger it was possible to apply the same Jäschke temperatures on the system to calculate a near optimal operation of the network. It shows that the concept is universal and may be applied for a wide range of systems.

As for the future work the Jäschke temperature can be applied to the possible case of membrane contactor network for removal of acid gases from natural gas and it might be interesting to further investigate implementation of membrane processes in MEN.

Chapter 10

Resumé

Táto práca sa zaoberá modelovaním a optimálnym riadením systémov membránových procesov, ktoré sú spolu zapojené do siete. Membránové procesy sa využívajú na separáciu zložiek za využitia membrány. V dnešnej dobe tieto membránové separačné procesy sa stavajú dôležitou časťou v priemysle pri výrobe a v druhotnom výrobnom odvetví vďaka ich vyššej výkonnosti separácie a nízkej spotrebe energie oproti klasickým separačným metódam ako je napr. destilácia.

Pri modelovaní a riadení sa ako typ membránového procesu zvolil membránový kontaktor, ktorý dovoľuje priamy kontakt medzi dvoma fázami. Pre špeciálny prípad tohto membránového procesu bol zvolený proces dialýzy, pre ktorý sa zostavil model procesu. Tento model pozostáva z algebraických rovníc vyjadrujúce prestup látky cez membránový modul v ustálenom stave a z diferenciálnej rovnice vyjadrujúcu zmenu objemu pacienta. Pri zanedbaní dynamického modelu pacienta sa model procesu použil na riadenie sústavy membránových kontaktorov.

V tejto práci sa riešilo niekoľko možných zapojení systémov. Každý systém pozostáva z membránových kontaktorov, pričom vstupujúci prúd suroviny sa rozdelil na niekoľko menších prúdov, ktoré sa následne očistili od nečistôt jedným či viacerými membránovými kontaktormi. Cieľom bolo optimalizovať konečnú koncentráciu, ako výsledok zmiešania prúdov po očistení membránovými kontaktormi.

Koncept samo-optimalizácie sa zavádza, ako spôsob riadenia systému, ktorý navrhol Johannes Jäschke (NTNU, Nórsko) na riadenie sústavy výmenníkov tepla. Táto metóda spočíva v nájdení optimálnych riadených premenných. Riadením týchto premenných je možné dosiahnuť takmer optimálnu prevádzku riešeného systému. Takáto premenná pre systém výmenníkov tepla je Jäschkého teplota, ktorá ak sa riadi na konštantnú hodnotu, sa získa optimálne nastavenie riadiacich premenných.

Na otestovanie riadenia za implementácie samo-optimalizácie s využitím Jäschkého teplôt bolo zvolených niekoľko simulačných scenárov. V každom z nich sa porovnávajú rôzne možnosti riadenia: použitie NTU metódy, použitie konceptu samo-optimalizácie s využitím Jäschkého teplôt a optimalizácia za využitia MATLAB solvera `fmincon`. Na základe výsledkov uvedených pre každý príklad sa porovnali výsledky dosiahnuté týmito metódami. Všetky tieto metódy spočívajú na rôznych algoritmoch pri výpočte konečnej koncentrácie, a z tohto dôvodu sa výsledky líšia medzi sebou. Na základe uvedených výsledkov možno skonštatovať, že pri využití metódy s Jäschkého teplotami sa dosiahlo takmer optimálne riešenie s minimálnou stratou pri prvých dvoch prípadoch. Posledný prípad ukázal zhoršenie pri výpočte optimálneho riešenia z dôvodu nepresnej aproximácie logaritmického stredného priemeru aritmetickým priemerom.

Na výsledkoch simulácie sa dokázalo, že túto formu riadenia je možne aplikovať aj pre iné typy systémov, ako to bolo pôvodne navrhnuté pre výmenníky tepla, pričom sa dosiahlo takmer optimálnej prevádzky systémov membránových kontaktorov. Z tohto dôvodu je možné túto stratégiu riadenia rozšíriť o zavedenie možných porúch a na ďalších typoch membránových procesov.

Bibliography

Anonymous. Membrane filtration. *Tech Brief*, 1999. URL http://www.nesc.wvu.edu/pdf/dw/publications/ontap/2009_tb/membrane_DWFSOM43.pdf. 19, 21

R. W. Baker. *Membrane Technology and Applications*. Wiley, 2004. ISBN 0-07-135440-9. 19, 25

A. Basile, A. Figoli, and M. Khayet. *Membrane Contactors: Fundamentals, Applications and Potentialities*. Elsevier, 2015. ISBN 978-1-78242-246-4. 23

J. W. Chinneck. *Practical Optimization: A Gentle Introduction*. Carleton University, 2009. URL http://www.sce.carleton.ca/faculty/chinneck/po.html. 16

E. K. P. Chong and S. H. Zak. *An Introduction to Optimization*. Wiley, 2001. ISBN 0-471-39126-3. 16

V. Y. Dindore and G. F. Versteeg. Gas-liquid mass transfer in a cross-flow hollow fiber module: Analytical model and experimental validation. *International Journal of Heat and Mass Transfer*, 48:3352–3362, 2005. 25

J. M. Douglas. *Conceptual Design of Chemical Processes*. McGraw Hili Book Company, 1988. ISBN 0070177627. 47

J. J. Downs and S. Skogestad. An industrial and academic perspective on plantwide control. *Journal of Process Control*, 35:99–110, 2011. 33

E. Drioli, A. Criscuoli, and E. Curcio. *Membrane Contactors: Fundamentals, Applications and Potentialities*. Elsevier, 2005. ISBN 9780080457017. 25

M. M. El-Halwagi. *Process Integration*. Elsevier, 2006. ISBN 0-12-370532-0. 45

M. M. El-Halwagi and Manousiouthakis. Synthesis of mass exchange networks. *AIChE Journal*, 35:1233–1244, 1989. 46

M. Fikar and R. Paulen. *Optimal Operation of Batch Membrane Processes*. Springer, 2015. ISBN 978-3-319-20475-8. 18, 19, 20, 21, 23

C. D. Fischer and O. A. Iribarren. Hydrogen recovery from the purge stream of a cyclohexane production process using a mass exchange heuristic. *International Journal of Hydrogen Energy*, 39:20094–20104, 2014. 47

A. Gabelman and S. T. Hwang. Hollow fiber membrane contactors. *Journal of Membrane Science*, 159:61–106, 1999. 25

Ch. J. Geankoplis. *Transport Processes and Separation Process Principles.* Prentice Hall, 2008. ISBN 0-13-101367-X. 18, 19, 21, 22, 23

J. Jäschke and S. Skogestad. Optimal controlled variables for polynomial systems. *Journal of Process Control*, 22:167–179, 2012. 16

J. Jäschke and S. Skogestad. Optimal operation of heat exchanger networks with stream split:only temperature measurements are required. *Computers and Chemical Engineering*, 70:35–49, 2014. 29, 30, 32, 33, 34

E. Korovessi and A. A. Linninger. *Batch Processes.* CRC Press, 2005. ISBN 9780824725228. 46

W. Kujawski. Application of pervaporation and vapor permeation in environmental protection. *Polish Journal of Environmental Studies*, 35:1233–1244, 1989. 16

J. L. Li and B. H. Chen. Review of co2 absorption using chemical solvents in hollow fiber membrane contactors. *Separation and Purification Technology*, 41:109–122, 2005. 49

S. A. M. Marzouk, M. H. Al-Marzouqi, M. Teramoto, N. Abdullatif, and Z. M. Ismail. Simultaneous removal of co2 and h2s from pressurized co2–h2s–ch4 gas mixture using hollow fiber membrane contactors. *Separation and Purification Technology*, 86:88–97, 2012. 49

Q. F. Qiu, G. P. Rangaiah, and P. R. Krisgnaswamy. Application of a plant-wide control design to the hda process. *Computers and Chemical Engineering*, 27: 73–94, 2003. 47

M. T. Ravanchi, T. Kaghazschi, A. Kargari, and H. Liu. *Application of membrane separation processes in petrochemical industry: a review*, volume 235. Elsevier, 2009. 17, 19, 21, 22, 23

H.K. Shon, S: Phuntsho, D. S. Chaudhary, S. Vigneswaran, and J. Cho. Nanofiltration for water and wastewater treatment - a mini review. *Drink. Water Eng. Sci.l*, 6:47–53, 2013. 21

K. Simons. *Membrane technology for* $\mathrm{CO_2}$ *capture.* dissertation, University of Twente, 2010. 51

S. Skogestad. *Chemical and energy process engineering.* CRC Press, 2009. ISBN 13-978-1-4200-8755-0. 25, 27, 28, 35

Sigurd Skogestad. Plantwide control: The search for the self-optimizing control structure. In *In: Preprints 14th IFAC World*, pages 325–330, 1999. 29

Z. Szitkai. *Synthesis of Mass Exchange Networks Using Mathematical Programming.* dissertation, Budapest University of Technology and Economics, 2004. 46

I. D. Wilson. *Encyclopedia of Separation Science.* Academic Press, 2000. ISBN 978-0-12-226770-3. 19, 20, 22, 23, 24, 25

Appendix A

Program Source Codes

Two Membrane Contactors in Parallel

1. The main program

```
% ==============================================================
%  ----------------- Dialysis process ---------------------------
%                     two in parallel
% ==============================================================
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %               1                %
                %        -------0-------         %
                %   ----|                |----    %
                %        -------0-------         %
                %               2                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all, clc
%% Parameters
par.cb=50;   %[mmol/l]
par.qb=0.3; %[l/min]
par.qd1=0.5; %[l/min]
par.qd2=0.45; %[l/min]
par.cd1=5; %[mmol/l]
par.cd2=3; %[mmol/l]
par.kA1=0.1; %[mmol/l]
par.kA2=.2; %[mmol/l]

cb=par.cb;
qb=par.qb;
qd1=par.qd1;
qd2=par.qd2;
cd1=par.cd1;
cd2=par.cd2;
kA1=par.kA1;
kA2=par.kA2;

% --------------------------------------------------------------
%% solving the problem with fmincon
```

```
A = []; b = []; Aeq = []; beq = [];
lb = 0*ones(5,1); ub = inf*ones(5,1);
int_u=[0.39;14;7;8.5;17.6];
options=optimset('display','iter','algorithm','active-set','tolcon',1
    e-8,'tolfun',1e-8);
[x,fval,exitflag]=fmincon(@obj_simpl,int_u,A,b,Aeq,beq,lb,ub,@(x)
    nonlcon(x, par),options);
c_Bmem = fval;u=x(1);
u_cBmem_exitflag=[u c_Bmem exitflag];

% % ======== fmincon with Jaschke temperatures ========
A = []; b = []; Aeq = []; beq = [];
lb = 0*ones(5,1); ub = inf*ones(5,1);
int_u=[0.3686;21.61;20.36;6.27;12.47];
options=optimset('display','iter','algorithm','active-set','tolcon',1
    e-8,'tolfun',1e-8);
[x,fval,exitflag]=fmincon(@obj_simpl,int_u,A,b,Aeq,beq,lb,ub,@(x)nnlc
    (x,par),options);
cTJ_Bmem = fval;u_TJ=x(1);
CTJ_u_cBmem=[u_TJ cTJ_Bmem exitflag];

% ---------------------------------------------------------------
%% solving problem using NTU method
N=1000;
for i=1:N
    u = i/N;
    U(i)=u;
    [c_B] = problem_iterations(cb,qb,kA1,kA2,cd1,cd2,qd1,qd2,u);
    cbmem1(i)=c_B(1); cbmem2(i)=c_B(2); cbmem(i)=c_B(3); cdmem1(i)=
        c_B(4); cdmem2(i)=c_B(5);
end
[c_Bmem,n]=min(cbmem);
u=U(n);
NTU_u_cbmem=[u,c_Bmem];
% % ======== NTU with Jaschke temperatures ========
Tj1=(cbmem1-cb).^2./(cd1-cb);
Tj2=(cbmem2-cb).^2./(cd2-cb);
Tj=abs(Tj1-Tj2);
[TJ,n]=min(Tj);
c_Bend=cbmem(n);
cJ_u_cbmem=[U(n) c_Bend];

figure,plot(U(n),c_Bmem,'r*',U,cbmem,'b',U,Tj,'m',[U(n) U(n)],[0 45],
    'g-.',[0 0],[0 1],'k'),
xlabel('u');ylabel('c');legend({'optimum   ','optimum split','abs(TJ)
    ','tj1=tj2'},'FontSize',16,'Location','SouthEast');
print(gcf,'-depsc2','MC2')

%% --------------- Results ---------------------------
% ----------------------------------------------------
clc,
disp(sprintf('\t THE RESULTS:\n      [split    c_bmem     exitflag]\n'))
    ;
disp(sprintf('\tNTU method:'));
disp(NTU_u_cbmem)
disp(sprintf('\t FMINCON:'));
disp(u_cBmem_exitflag)
disp(sprintf('\t NTU method with Jaschke temperature'));
disp(cJ_u_cbmem)
disp(sprintf('\t FMINCON with Jaschke temperature'));
disp(CTJ_u_cBmem)
```

2. The objective function

```
function obj_fun2=obj(x)
obj_fun2=(x(1)*x(2)+(1-x(1))*x(3));
```

3. Nonlinear constraints

```
function [c,ce]=nonlcon(x,par)
%% Defining
%           parameters
cb=par.cb;
qb=par.qb;
qd1=par.qd1;
qd2=par.qd2;
cd1=par.cd1;
cd2=par.cd2;
kA1=par.kA1;
kA2=par.kA2;

%           state variables
u=x(1);
cbmem1 = x(2); cbmem2 = x(3);
cdmem1 = x(4); cdmem2 = x(5);

qb1=u*qb;
qb2=(1-u)*qb;

%           Simplifying expressions
c1in=cbmem1-cd1;   c1out=cb-cdmem1;
c2in=cbmem2-cd2;   c2out=cb-cdmem2;
dt1=c1in/c1out;
dt2=c2in/c2out;
DeltaCb1=(c1in-c1out)/log(dt1);
DeltaCb2=(c2in-c2out)/log(dt2);
c1=cb-cbmem1;
c2=cb-cbmem2;
c_d1=cdmem1-cd1;
c_d2=cdmem2-cd2;
%% Inequality constraints
 c = [
%        1.01-dt1;
%        1.01-dt2;
      -c1;
      -c2;
      -c_d1;
      -c_d2;
       -cbmem1;-cbmem2;-DeltaCb1;-DeltaCb2;-qb2;-qb1;-dt1;-dt2;-c1out
           ;-c2out;-c1in;-c2in;
      ];
%% Equality constraints
ce = [
        qb-qb1-qb2;                       % Mass balance
        qb1*c1-qd1*c_d1;                   % mass balance, fluid
            stream, line 1
        qb2*c2-qd2*c_d2;                   % mass balance, fluid
            stream, line 2
        qb1*c1-kA1*DeltaCb1;             % mass balance, mass
            transfer, line 1
        qb2*c2-kA2*DeltaCb2;             % mass balance, mass
            transfer, line 2
        ];
end
```

4. Nonlinear constraints with Jäschke temperatures

```
function [c,ce]=nnlc(x,par)
%% Defining
%           parameters
cb=par.cb;
qb=par.qb;
qd1=par.qd1;
qd2=par.qd2;
cd1=par.cd1;
cd2=par.cd2;
kA1=par.kA1;
kA2=par.kA2;

%           state variables
u=x(1);
cbmem1 = x(2); cbmem2 = x(3);
cdmem1 = x(4); cdmem2 = x(5);

qb1=u*qb;
qb2=(1-u)*qb;

%           Simplifying expressions
c1in=cbmem1-cd1;   c1out=cb-cdmem1;
c2in=cbmem2-cd2;   c2out=cb-cdmem2;
dt1=c1in/c1out;
dt2=c2in/c2out;
DeltaCb1=(c1in-c1out)/log(dt1);
DeltaCb2=(c2in-c2out)/log(dt2);
c1=cb-cbmem1;
c2=cb-cbmem2;
c_d1=cdmem1-cd1;
c_d2=cdmem2-cd2;


%% Inequality constraints
 c = [
      -c1;-c2;
      -c_d1;-c_d2;
      -cbmem1;-cbmem2;
      -DeltaCb1;-DeltaCb2;
      -qb2;-qb1;
      -dt1;-dt2;
      -c1out;-c2out;
      -c1in;-c2in;
      ];


%% Equality constraints
ce = [
        qb-qb1-qb2;                             % Mass balance
        qb1*c1-qd1*c_d1;                    % mass balance, fluid
            stream, line 1
        qb2*c2-qd2*c_d2;                    % mass balance, fluid
            stream, line 2
        qb1*c1-kA1*DeltaCb1;              % mass balance, mass
            transfer, line 1
        qb2*c2-kA2*DeltaCb2;              % mass balance, mass
            transfer, line 2
        Tj1-Tj2;
        ];
end
```

5. NTU method

```
function [c_B] =problem_iterations2(cb,qb,kA1,kA2,cd1,cd2,qd1,qd2,u)

    qb1=u*qb;
    qb2=(1-u)*qb;

    N1 = kA1/qb1;
    N2 = kA2/qb2;
    C1 = qb1/qd1;
    C2 = qb2/qd2;

%     e1 = (1-exp(-N1*(C1-1)))/(C1-exp(-N1*(C1-1)));
%     e2 = (1-exp(-N2*(C2-1)))/(C2-exp(-N2*(C2-1)));

    e1 = (1-exp(-N1*(1-C1)))/(1-C1*exp(-N1*(1-C1)));
    e2 = (1-exp(-N2*(1-C2)))/(1-C2*exp(-N2*(1-C2)));

    if(C1>0.999 && C1<1.001)
        e1=N1/(N1+1);
    end
    if(C2>0.999 && C2<1.001)
        e2=N2/(N2+1);
    end

    Cb1  = e1*cd1 + (1-e1)*cb;
    Cb2  = e2*cd2 + (1-e2)*cb;
    Cend=u*Cb1+(1-u)*Cb2;
    Cdmem1=(1-C1*e1)*cd1+C1*e1*cb;
    Cdmem2=(1-C2*e2)*cd2+C2*e2*cb;
    c_B = [Cb1 Cb2 Cend Cdmem1 Cdmem2];
%     c_B=[cbmem1  cbmem2  cbmem Cbmem1 Cbmem2 Cbmem ];
end
```

Two Membrane Contactors in Series and one in Parallel

1. The main program

```
% ==============================================================
%  ---------------- Dialysis process ---------------------------
%          two in series and one in parallel
% ==============================================================
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %              11     12              %
                %       -------0------0-------          %
                %   ----|                       |----    %
                %       -----------0-----------         %
                %                   21                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all, clc
par.cb=60;   %[mmol/l]
par.qb=0.55; %[l/min]
par.qd11=0.4; %[l/min]
par.qd12=0.5; %[l/min]
par.qd21=0.45; %[l/min]
par.cd11=4; %[mmol/l]
par.cd12=2; %[mmol/l]
par.cd21=6; %[mmol/l]
par.kA11=0.1; %[mmol/l]
par.kA12=0.2; %[mmol/l]
par.kA21=0.1; %[mmol/l]

cb=par.cb;
qb=par.qb;
qd11=par.qd11;
qd12=par.qd12;
qd21=par.qd21;
cd11=par.cd11;
cd12=par.cd12;
cd21=par.cd21;
kA11=par.kA11;
kA12=par.kA12;
kA21=par.kA21;

% ----------------------------------------------------------
%%  solving the problem with fmincon
A = []; b = []; Aeq = []; beq = [];
lb = 0*ones(7,1); ub = inf*ones(7,1);
% int_u=[0.5;34;12;14;7;12;2;];
% int_u=[0.738;34.0695;16.3627;12.2017;9.3194;8.2868;5.5437];
% int_u=[0.738;41.7589;20.9016;10.0712;11.2867;14.8342;5.825];
options=optimset('display','iter','algorithm','active-set','tolcon',1
    e-8,'tolfun',1e-8);
[x,fval,exitflag]=fmincon(@obj3,int_u,A,b,Aeq,beq,lb,ub,@(x)nonlcon2(
    x,par),options);
c_Bmem = fval;u=x(1);
u_cBmem_exitflag=[u c_Bmem exitflag];
```

```
% % ======== fmincon with Jaschke temperatures ========
A = []; b = []; Aeq = []; beq = [];
lb = 0*ones(7,1); ub = inf*ones(7,1);
int_u=[0.58;34;16;12;9;8;5;];
options=optimset('display','iter','algorithm','active-set','tolcon',1
    e-8,'tolfun',1e-8);
[x,fval,exitflag]=fmincon(@obj3,int_u,A,b,Aeq,beq,lb,ub,@(x)
    nonlcon2Tj(x,par),options);
cTJ_Bmem = fval;u_TJ=x(1);
CTJ_u_cBmem=[u_TJ cTJ_Bmem exitflag];

% -----------------------------------------------------------
%% solving problem using NTU method
N=1000;
for i=1:N
    u = i/N;
    U(i)=u;
    [c_B] = problem_iterations3(cb,qb,kA11,kA12,kA21,cd11,cd12,cd21,
        qd11,qd12,qd21,u);
    cbmem11(i)=c_B(1); cbmem12(i)=c_B(2); cbmem21(i)=c_B(3); cbmem (i
        )=c_B(4);
    cdmem11(i)=c_B(5); cdmem12(i)=c_B(6);cdmem21(i)=c_B(7);
end
[c_Bmem,n]=min(cbmem);c_Bmem;
u=U(n);
NTU_u_cbmem=[u,c_Bmem];
[cbmem11(n) cbmem12(n) cbmem21(n) cdmem11(n) cdmem12(n) cdmem21(n)];
figure,plot(u,c_Bmem,'r*')

% % ======== NTU with Jaschke temperatures ========
Tj11=(cbmem11-cb).^2./(cd11-cb);
Tj12=(cbmem12-cbmem11).*(cbmem12+cbmem11-2*cb-Tj11)./(cd12-cbmem11);
Tj21=(cbmem21-cb).^2./(cd21-cb);
Tj=abs(Tj11+Tj12-Tj21);
[TJ,n]=min(Tj);
c_Bend=cbmem(n);
cJ_u_cbmem=[U(n) c_Bend];
x0=[cbmem11(n) cbmem12(n) cbmem21(n) cdmem11(n) cdmem12(n) cdmem21(n)
    ];

hold on,plot(U,cbmem,'b',U,Tj,'m',[U(n) U(n)],[0 70],'g-.',[0 0],[0
    1],'k'),axis([0 1 0 70]);
xlabel('u');ylabel('c');legend({'optimum   ','optimum split','abs(TJ)
    ','tj1+tj2=tj3'},'FontSize',18,'Location','NorthWest');
% print(gcf,'-depsc2','MC3')
%% --------------- Results ---------------------------
clc,
disp(sprintf('\t THE RESULTS:\n     [split    c_bmem     exitflag]\n'))
    ;
disp(sprintf('\tNTU method:'));
disp(NTU_u_cbmem)
disp(sprintf('\t FMINCON:'));
disp(u_cBmem_exitflag)
disp(sprintf('\t NTU method with Jaschke temperature'));
disp(cJ_u_cbmem)
disp(sprintf('\t FMINCON with Jaschke temperature'));
disp(CTJ_u_cBmem)
```

2. The objective function

```
function obj_fun3=obj3(x)
obj_fun3=( x(1)*x(3)+ (1-x(1)) *x(4) );
```

3. Nonlinear constraints

```
function [c,ce]=nonlcon2(x,par)
%% Defining
%           parameters
cb=par.cb;
qb=par.qb;
qd11=par.qd11;
qd12=par.qd12;
qd21=par.qd21;
cd11=par.cd11;
cd12=par.cd12;
cd21=par.cd21;
kA11=par.kA11;
kA12=par.kA12;
kA21=par.kA21;
%           state variables
u=x(1);
cbmem11 = x(2); cbmem12 = x(3);cbmem21 = x(4);
cdmem11 = x(5); cdmem12 = x(6);cdmem21 = x(7);
% cbmem = x(8);
qb1=u*qb;
qb2=(1-u)*qb;

%           Simplifying expressions
c11in=cbmem11-cd11;   c11out=cb-cdmem11;
c12in=cbmem12-cd12;   c12out=cbmem11-cdmem12;
c21in=cbmem21-cd21;   c21out=cb-cdmem21;
dt11=c11in/c11out;
dt12=c12in/c12out;
dt21=c21in/c21out;
DeltaCb11=(c11in-c11out)/log(dt11);
DeltaCb12=(c12in-c12out)/log(dt12);
DeltaCb21=(c21in-c21out)/log(dt21);
c11=cb-cbmem11;
c12=cbmem11-cbmem12;
c21=cb-cbmem21;
c_d11=cdmem11-cd11;
c_d12=cdmem12-cd12;
c_d21=cdmem21-cd21;


%% Inequality constraints
c = [
      -c11;-c12;-c21;
      -c_d11;-c_d12;-c_d21;
      -cbmem11;-cbmem12;-cbmem21;
      -DeltaCb11;-DeltaCb12;-DeltaCb21;
      -qb1;-qb2;
      -dt11;-dt12;-dt21;
      -c11out;-c12out;-c21out;
      -c11in;-c12in;-c21in;
      ];


%% Equality constraints
ce = [
        % Mass balance
        qb-qb1-qb2;
        % Mass balance of membrane unit 11
        qb1*c11-qd11*c_d11;                   % mass balance, fluid
            stream, line 1
        qb1*c11-kA11*DeltaCb11;             % mass balance, mass
            transfer, line 1
        % Mass balance of membrane unit 12
        qb1*c12-qd12*c_d12;                   % mass balance, fluid
            stream, line 2
        qb1*c12-kA12*DeltaCb12;             % mass balance, mass
            transfer, line 2
        % Mass balance of membrane unit 21
        qb2*c21-qd21*c_d21;                   % mass balance, fluid
            stream, line 2
        qb2*c21-kA21*DeltaCb21;             % mass balance, mass
            transfer, line 2
        ];
 [cdmem21, c21out,dt21];
end
```

4. Nonlinear constraints with Jäschke temperatures

```
function [c,ce]=nonlcon2Tj(x,par)
%% Defining
%           parameters
cb=par.cb;
qb=par.qb;
qd11=par.qd11;
qd12=par.qd12;
qd21=par.qd21;
cd11=par.cd11;
cd12=par.cd12;
cd21=par.cd21;
kA11=par.kA11;
kA12=par.kA12;
kA21=par.kA21;
%           state variables
u=x(1);
cbmem11 = x(2); cbmem12 = x(3);cbmem21 = x(4);
cdmem11 = x(5); cdmem12 = x(6);cdmem21 = x(7);
qb1=u*qb;
qb2=(1-u)*qb;

%           Simplifying expressions
c11in=cbmem11-cd11;   c11out=cb-cdmem11;
c12in=cbmem12-cd12;   c12out=cbmem11-cdmem12;
c21in=cbmem21-cd21;   c21out=cb-cdmem21;
```

```
dt11=c11in/c11out;
dt12=c12in/c12out;
dt21=c21in/c21out;
DeltaCb11=(c11in-c11out)/log(dt11);
DeltaCb12=(c12in-c12out)/log(dt12);
DeltaCb21=(c21in-c21out)/log(dt21);
c11=cb-cbmem11;
c12=cbmem11-cbmem12;
c21=cb-cbmem21;
c_d11=cdmem11-cd11;
c_d12=cdmem12-cd12;
c_d21=cdmem21-cd21;

% Jaeschke temperatures
Tj11=(cbmem11-cb).^2./(cd11-cb);
Tj12=(cbmem12-cbmem11).*(cbmem12+cbmem11-2*cb-Tj11)./(cd12-cbmem11);
Tj21=(cbmem21-cb).^2./(cd21-cb);
%% Inequality constraints
c = [
      -c11;-c12;-c21;
      -c_d11;-c_d12;-c_d21;
      -cbmem11;-cbmem12;-cbmem21;
      -DeltaCb11;-DeltaCb12;-DeltaCb21;
      -qb1;-qb2;
      -dt11;-dt12;-dt21;
      -c11out;-c12out;-c21out;
      -c11in;-c12in;-c21in;
      ];
%% Equality constraints
ce = [
        Tj11+Tj12-Tj21;
        % Mass balance
        qb-qb1-qb2;
        % Mass balance of membrane unit 11
        qb1*c11-qd11*c_d11;                   % mass balance, fluid
            stream, line 1
        qb1*c11-kA11*DeltaCb11;             % mass balance, mass
            transfer, line 1
        % Mass balance of membrane unit 12
        qb1*c12-qd12*c_d12;                   % mass balance, fluid
            stream, line 2
        qb1*c12-kA12*DeltaCb12;             % mass balance, mass
            transfer, line 2
        % Mass balance of membrane unit 21
        qb2*c21-qd21*c_d21;                   % mass balance, fluid
            stream, line 2
        qb2*c21-kA21*DeltaCb21;             % mass balance, mass
            transfer, line 2
        ];
end
```

5. NTU method

```
function [c_B] =problem_iterations_3(cb,qb,kA1,kA2,kA3,cd1,cd2,cd3,
    qd1,qd2,qd3,u1,u2)
    qb1=u1*qb;
    qb2=u2*qb;
    qb3=(1-u1-u2)*qb;

    N1 = kA1/qb1;
    N2 = kA2/qb2;
    N3 = kA3/qb3;
    C1 = qb1/qd1;
    C2 = qb2/qd2;
    C3 = qb3/qd3;

    e1 = (1-exp(-N1*(1-C1)))/(1-C1*exp(-N1*(1-C1)));
    e2 = (1-exp(-N2*(1-C2)))/(1-C2*exp(-N2*(1-C2)));
    e3 = (1-exp(-N3*(1-C3)))/(1-C3*exp(-N3*(1-C3)));

    if(C1>0.999 && C1<1.001)
        e1=N1/(N1+1);
    end
    if(C2>0.999 && C2<1.001)
        e2=N2/(N2+1);
    end
    if(C3>0.999 && C3<1.001)
        e3=N3/(N3+1);
    end

    Cb1  = e1*cd1 + (1-e1)*cb;
    Cb2  = e2*cd2 + (1-e2)*cb;
    Cb3  = e3*cd3 + (1-e3)*cb;
    Cend=u1*Cb1+u2*Cb2+(1-u1-u2)*Cb3;
    Cdmem1=(1-C1*e1)*cd1+C1*e1*cb;
    Cdmem2=(1-C2*e2)*cd2+C2*e2*cb;
    Cdmem3=(1-C3*e3)*cd3+C3*e3*cb;
    c_B = [Cb1 Cb2 Cb3 Cend Cdmem1 Cdmem2 Cdmem3];
end
```

Four Membrane Contactors in Series and Two in Parallel

1. The main program

```
% ===============================================================
%  ---------------- Dialysis process ---------------------------
%          4 in series and 2 in parallel
% ===============================================================
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %               11    12   13   14             %
                %        -------0------0-----0-----0------         %
                %   ----|                                     |----
                        %
                %        -----------0------0---------------          %
                %                   21    22                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all, clc
```

```
par.cb=80;    %[mmol/l]
par.qb=0.5; %[l/min]

par.qd11=0.4; %[l/min]
par.qd12=0.3; %[l/min]
par.qd13=0.5; %[l/min]
par.qd14=0.5; %[l/min]
par.qd21=0.35; %[l/min]
par.qd22=0.45; %[l/min]

par.cd11=2; %[mmol/l]
par.cd12=5; %[mmol/l]
par.cd13=7; %[mmol/l]
par.cd14=3; %[mmol/l]
par.cd21=9; %[mmol/l]
par.cd22=1; %[mmol/l]

par.kA11=0.4; %[mmol/l]
par.kA12=0.2; %[mmol/l]
par.kA13=0.3; %[mmol/l]
par.kA14=0.2; %[mmol/l]
par.kA21=0.1; %[mmol/l]
par.kA22=0.2; %[mmol/l]

cb=par.cb;
qb=par.qb;
qd11=par.qd11;
qd12=par.qd12;
qd13=par.qd13;
qd14=par.qd14;
qd21=par.qd21;
qd22=par.qd22;
cd11=par.cd11;
cd12=par.cd12;
cd13=par.cd13;
cd14=par.cd14;
cd21=par.cd21;
cd22=par.cd22;
kA11=par.kA11;
kA12=par.kA12;
kA13=par.kA13;
kA14=par.kA14;
kA21=par.kA21;
kA22=par.kA22;

%%           solving problem with fmincon
% % ======== fmincon with Jaschke temperatures ========
A = []; b = []; Aeq = []; beq = [];
lb = 0*ones(13,1); ub = inf*ones(13,1);
int_u=[0.5;65;46;26;19;46;16;16;34;29;19;15;18];
options=optimset('display','iter','algorithm','active-set','tolcon',1
    e-8,'tolfun',1e-8);
[x,fval,exitflag]=fmincon(@obj6,int_u,A,b,Aeq,beq,lb,ub,@(x)
    nonlcon6Tj(x,par),options);
cTJ_Bmem = fval;u_TJ=x(1);
CTJ_u_cBmem=[u_TJ cTJ_Bmem exitflag];
```

```
%% solving problem using NTU method
N=1000;
for i=1:N
    u = i/N;
    U(i)=u;
    [c_B] = problem_iterations4(cb,qb,kA11,kA12,kA13,kA14,kA21,kA22,
        cd11,cd12,cd13,cd14,cd21,cd22,qd11,qd12,qd13,qd14,qd21,qd22,u
        );
    cbmem11(i)=c_B(1); cbmem12(i)=c_B(2); cbmem13(i)=c_B(3);cbmem14(i
        )=c_B(4); cbmem21(i)=c_B(5); cbmem22(i)=c_B(6); cbmem(i)=c_B
        (7);
    cdmem11(i)=c_B(8); cdmem12(i)=c_B(9);cdmem13(i)=c_B(10);cdmem14(i
        )=c_B(11);cdmem21(i)=c_B(12); cdmem22(i)=c_B(13);
end
[c_Bmem,n]=min(cbmem);c_Bmem;
u=U(n);
NTU_u_cbmem=[u,c_Bmem];

%% --------------- Results ---------------------------
clc,
disp(sprintf('\t THE RESULTS: 6 units\n      [split   c_bmem
    exitflag]\n'));
disp(sprintf('\tNTU method:'));
disp(NTU_u_cbmem)
disp(sprintf('\t FMINCON with Jaschke temperature'));
disp(CTJ_u_cBmem)
```

2. The objective function

```
function obj_fun6=obj6(x)
obj_fun6=( x(1)*x(5)+ (1-x(1)) *x(7) );
```

3. Nonlinear constraints with Jäschke temperatures

```
function [c,ce]=nonlcon6Tj(x,par)
%% Defining
%           parameters
cb=par.cb;
qb=par.qb;
qd11=par.qd11;
qd12=par.qd12;
qd13=par.qd13;
qd14=par.qd14;
qd21=par.qd21;
qd22=par.qd22;
cd11=par.cd11;
cd12=par.cd12;
cd13=par.cd13;
cd14=par.cd14;
cd21=par.cd21;
cd22=par.cd22;
kA11=par.kA11;
kA12=par.kA12;
```

```
kA13=par.kA13;
kA14=par.kA14;
kA21=par.kA21;
kA22=par.kA22;
%           state variables
u=x(1);
cbmem11 = x(2); cbmem12 = x(3);cbmem13 = x(4);
cbmem14 = x(5); cbmem21 = x(6);cbmem22 = x(7);
cdmem11 = x(8); cdmem12 = x(9);cdmem13 = x(10);
cdmem14 = x(11); cdmem21 = x(12);cdmem22 = x(13);
% cbmem = x(8);
qb1=u*qb;
qb2=(1-u)*qb;

%           Simplifying expressions
c11in=cbmem11-cd11;   c11out=cb-cdmem11;
c12in=cbmem12-cd12;   c12out=cbmem11-cdmem12;
c13in=cbmem13-cd13;   c13out=cbmem12-cdmem13;
c14in=cbmem14-cd14;   c14out=cbmem13-cdmem14;
c21in=cbmem21-cd21;   c21out=cb-cdmem21;
c22in=cbmem22-cd22;   c22out=cbmem21-cdmem22;
dt11=c11in/c11out;
dt12=c12in/c12out;
dt13=c13in/c13out;
dt14=c14in/c14out;
dt21=c21in/c21out;
dt22=c22in/c22out;
DeltaCb11=(c11in-c11out)/log(dt11);
DeltaCb12=(c12in-c12out)/log(dt12);
DeltaCb13=(c13in-c13out)/log(dt13);
DeltaCb14=(c14in-c14out)/log(dt14);
DeltaCb21=(c21in-c21out)/log(dt21);
DeltaCb22=(c22in-c22out)/log(dt22);
c11=cb-cbmem11;
c12=cbmem11-cbmem12;
c13=cbmem12-cbmem13;
c14=cbmem13-cbmem14;
c21=cb-cbmem21;
c22=cbmem21-cbmem22;
c_d11=cdmem11-cd11;
c_d12=cdmem12-cd12;
c_d13=cdmem13-cd13;
c_d14=cdmem14-cd14;
c_d21=cdmem21-cd21;
c_d22=cdmem22-cd22;

% Jaeschke temperatures
Tj11=(cbmem11-cb).^2./(cd11-cb);
Tj12=(cbmem12-cbmem11).*(cbmem12+cbmem11-2*cb-Tj11)./(cd12-cbmem11);
Tj13=(cbmem13-cbmem12).*(cbmem13+cbmem12-2*cb-Tj12)./(cd13-cbmem12);
Tj14=(cbmem14-cbmem13).*(cbmem14+cbmem13-2*cb-Tj13)./(cd14-cbmem13);
Tj21=(cbmem21-cb).^2./(cd21-cb);
Tj22=(cbmem22-cbmem21).*(cbmem22+cbmem21-2*cb-Tj21)./(cd22-cbmem21);

%% Inequality constraints
c = [
      -c11;-c12;-c13;-c14;-c21;-c22;
      -c_d11;-c_d12;-c_d13;-c_d14;-c_d21;-c_d22;
      -cbmem11;-cbmem12; -cbmem13; -cbmem14;-cbmem21; -cbmem22;
      -DeltaCb11;-DeltaCb12;-DeltaCb13;-DeltaCb14;-DeltaCb21;-
          DeltaCb22;
      -qb1;-qb2;
      -dt11;-dt12;-dt13;-dt14;-dt21;-dt22;
      -c11out;-c12out;-c13out;-c14out;-c21out;-c22out;
      -c11in;-c12in;-c13in;-c14in;-c21in;-c22in;
      ];

%% Equality constraints
ce = [
         Tj11+Tj12+Tj13+Tj14-Tj21-Tj22;
        % Mass balance
        qb-qb1-qb2;
        % Mass balance of membrane unit 11
        qb1*c11-qd11*c_d11;                           % mass balance, fluid
            stream, line 1
        qb1*c11-kA11*DeltaCb11;                     % mass balance, mass
            transfer, line 1
        % Mass balance of membrane unit 12
        qb1*c12-qd12*c_d12;                           % mass balance, fluid
            stream, line 1
        qb1*c12-kA12*DeltaCb12;                     % mass balance, mass
            transfer, line 1
        % Mass balance of membrane unit 13
        qb1*c13-qd13*c_d13;                           % mass balance, fluid
            stream, line 1
        qb1*c13-kA13*DeltaCb13;                     % mass balance, mass
            transfer, line 1
        % Mass balance of membrane unit 14
        qb1*c14-qd14*c_d14;                           % mass balance, fluid
            stream, line 1
        qb1*c14-kA14*DeltaCb14;                     % mass balance, mass
            transfer, line 1
        % Mass balance of membrane unit 21
        qb2*c21-qd21*c_d21;                           % mass balance, fluid
            stream, line 2
        qb2*c21-kA21*DeltaCb21;                     % mass balance, mass
            transfer, line 2
        % Mass balance of membrane unit 22
        qb2*c22-qd22*c_d22;                           % mass balance, fluid
            stream, line 2
        qb2*c22-kA22*DeltaCb22;                     % mass balance, mass
            transfer, line 2
        ];
end
```

4. NTU method

```
function [c_B] =problem_iterations4(cb,qb,kA11,kA12,kA13,kA14,kA21,
    kA22,...
    cd11,cd12,cd13,cd14,cd21,cd22,qd11,qd12,qd13,qd14,qd21,qd22,u)

    qb1=u*qb;
    qb2=(1-u)*qb;
    N11 = kA11/qb1;
    N12 = kA12/qb1;
    N13 = kA13/qb1;
    N14 = kA14/qb1;
    N21 = kA21/qb2;
    N22 = kA22/qb2;
    C11 = qb1/qd11;
    C12 = qb1/qd12;
    C13 = qb1/qd13;
    C14 = qb1/qd14;
    C21 = qb2/qd21;
    C22 = qb2/qd22;

    e11 = (1-exp(-N11*(1-C11)))/(1-C11*exp(-N11*(1-C11)));
    e12 = (1-exp(-N12*(1-C12)))/(1-C12*exp(-N12*(1-C12)));
    e13 = (1-exp(-N13*(1-C13)))/(1-C13*exp(-N13*(1-C13)));
    e14 = (1-exp(-N14*(1-C14)))/(1-C14*exp(-N13*(1-C14)));
    e21 = (1-exp(-N21*(1-C21)))/(1-C21*exp(-N21*(1-C21)));
    e22 = (1-exp(-N22*(1-C22)))/(1-C22*exp(-N22*(1-C22)));

    if(C11>0.999 && C11<1.001)
        e11=N11/(N11+1);
    end
    if(C12>0.999 && C12<1.001)
        e12=N12/(N12+1);
    end
    if(C13>0.999 && C13<1.001)
        e13=N13/(N13+1);
    end
    if(C14>0.999 && C14<1.001)
        e14=N14/(N14+1);
    end
    if(C21>0.999 && C21<1.001)
        e21=N21/(N21+1);
    end
    if(C22>0.999 && C22<1.001)
        e22=N22/(N22+1);
    end

    Cb11  = e11*cd11 + (1-e11)*cb;
    Cb12  = e12*cd12 + (1-e12)*Cb11;
    Cb13  = e13*cd13 + (1-e13)*Cb12;
    Cb14  = e14*cd14 + (1-e14)*Cb13;
    Cb21  = e21*cd21 + (1-e21)*cb;
    Cb22  = e22*cd22 + (1-e22)*Cb21;
    Cend=u*Cb14+(1-u)*Cb22;
    Cdmem11=(1-C11*e11)*cd11+C11*e11*cb;
    Cdmem12=(1-C12*e12)*cd12+C12*e12*cb;
    Cdmem13=(1-C13*e13)*cd13+C13*e13*cb;
    Cdmem14=(1-C14*e14)*cd14+C14*e14*cb;
    Cdmem21=(1-C21*e21)*cd21+C21*e21*cb;
    Cdmem22=(1-C22*e22)*cd22+C22*e22*cb;
    c_B = [Cb11 Cb12 Cb13 Cb14 Cb21 Cb22 Cend Cdmem11 Cdmem12 Cdmem13
         Cdmem14 Cdmem21 Cdmem22];
end
```