ผลกระทบของการเกิดฟาวลิ่งของเครื่องแลกเปลี่ยนความร้อน
ในหน่วยไฮโดรทรีทเตอร์ต่อการปฏิบัติการและการบำรุงรักษาที่เหมาะสม

นางสาวประพิศพรรณ เตชะวานิชชัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเคมี ภาควิชาวิศวกรรมเคมี
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2551

THE IMPACT OF HEAT EXCHANGER FOULING IN HYDROTREATING UNIT ON THE OPTIMUM OPERATION AND MAINTENANCE

Miss Prapitpan Techawanitchai

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Chemical Engineering
Department of Chemical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2008

Thesis Title	THE IMPACT OF HEAT EXCHANGER FOULING IN HYDROTREATING UNIT ON THE OPTIMUM OPERATION AND MAINTENANCE
By	Miss Prapitpan Techawanitchai
Field of Study	Chemical Engineering
Thesis Principal Advisor	Soorathep Kheawhom, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

....Dean of the Faculty of Engineering

(Associate Professor Boonsom Lerdhirunwong, Dr.Ing.)

THESIS COMMITTEE

Paisan Kittisupakorn .... Chairman

(Associate Professor Paisan Kittisupakorn, Ph.D.)

Soorathep Kheawhom .... Thesis Principal Advisor

(Soorathep Kheawkhom, Ph.D.)

.... Member

(Kasidit Noothong, Ph.D.)

Weerawat .... Member

(Assistant Professor Weerawat Pattaweekhongkha, Ph.D.)

ประพิศพรรณ เดชะวานิชชัย : ผลกระทบของการเกิดฟาวลิ่งของเครื่องแลกเปลี่ยนความร้อนในหน่วยไฮโดรทรีทเตอร์ต่อการปฏิบัติการและการบำรุงรักษาที่เหมาะสม (THE IMPACT OF HEAT EXCHANGER FOULING IN HYDROTREATING UNIT ON THE OPTIMUM OPERATION AND MAINTENANCE) อ.ที่ปรึกษา : อ.ดร.สุรเทพ เขียวหอม, 93 หน้า.

การเกิดฟาวลิ่งของเครื่องแลกเปลี่ยนความร้อนในหน่วยกำจัดซัลเฟอร์นั้น ส่งผลกระทบโดยตรงต่อการลดลงของประสิทธิภาพในการแลกเปลี่ยนความร้อน ส่งผลให้ต้องการพลังงานทดแทนมากขึ้นและการดำเนินการผลิตต้องหยุดลง เพื่อนำเครื่องแลกเปลี่ยนความร้อนนั้น มาทำการบำรุงรักษา งานวิจัยนี้ได้นำแบบจำลองของข่ายงานนิวรัล มาทำนายโพรไฟล์ของการเกิดฟาวลิ่งของเครื่องแลกเปลี่ยนความร้อนในหน่วยกำจัดซัลเฟอร์ เพื่อหาระยะเวลาที่เหมาะสม ที่จะนำเครื่องแลกเปลี่ยนความร้อนมาทำการบำรุงรักษา กระบวนการเกิดฟาวลิ่งเป็นกระบวนการที่มีความซับซ้อน และมีความไม่เป็นเชิงเส้นสูง การหาแบบจำลองทางคณิตศาสตร์ค่อนข้างยุ่งยาก ทางเลือกหนึ่งที่น่าสนใจ คือการประยุกต์ใช้ข่ายงานนิวรัลมาเป็นตัวแทนกระบวนการเกิดฟาวลิ่ง ซึ่งข่ายงานนิวรัลสามารถหาแบบจำลองที่มีความซับซ้อนและไม่เป็นเชิงเส้นได้ดี โดยเฉพาะในกรณีที่มีรายละเอียดของกระบวนการจำกัด

ภาควิชา: วิศวกรรมเคมี
สาขาวิชา: วิศวกรรมเคมี
ปีการศึกษา: 2551

ลายมือชื่อนิสิต: Prapitpam T.
ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก: สุรเทพ เขียวหอม

##4871424021 : MAJOR CHEMICAL ENGINEERING

KEY WORD: HEAT EXCHANGER FOULING / ARTIFICIAL NEURAL NETWORK / OPTIMUM OPERATION

PRAPITPAN TECHAWANITCHAI: THE IMPACT OF HEAT EXCHANGER FOULING IN HYDROTREATING UNIT ON THE OPTIMUM OPERATION AND MAINTENANCE THESIS PRINCIPAL ADVISOR: SOORATHEP KHEAWKHOM, Ph.D., 93 pp.

Fouling of feed/effluent heat exchanger in Hydrotreating unit have a significant impact on the operation efficiency. Hydrotreater fouling presents loss of heat transfer, enlargement of energy consumption, increasing in maintenance cost and throughput limitation. Thus, to solve these problems, heat exchangers are cleaned during shutdown. In this research, an artificial neural network has been used to develop the fouling model of feed/effluent heat exchanger in Hydrotreating unit. Developed algorithm is employed to determine the optimal operation and cleaning interval of the feed/effluent heat exchanger surfaces. The artificial neural network is widely applied in modeling and optimization of chemical process especially nonlinear systems. The main advantage of the use of artificial neural network is obtaining a highly accurate mathematical model of the system without the detail of the system. The process modeling applications use the artificial neural network to approximate the relationship between the input and output variables

Department: Chemical Engineering Student's signature: Prapitpan T.

Field of study: Chemical Engineering Thesis Principal Advisor's signature: Soorathep Kheawkhom

Academic year: 2008

ACKNOWLEDGEMENTS

No work of authorship is ever the accomplishment of one individual, and this thesis is no exception. The support is not limited only in my academic life, which the research work and writing this thesis were carried out, it also includes the support some way, encouragement, or personal involvement during my private life. Thus, it is virtually impossible to mention all those people who have contributed to me and consequently made my life more enjoy and happy than normal. Hence, in the first people, I would like to thank all these un-named individuals.

I am very grateful to my parents and I for their unconditional love and persistent support. I owe them a giant debt of gratitude. Thus, I am most of all my grateful to my parents, also my very first teacher, who has taught me to know a character. Without this support it would not be possible to prepare a work such as this.

I gratefully acknowledge my advisor Dr. Soorathep Kheawkhom for his critical guidance and his firm support during years.

I would also thank all of the members of the Master's degree in Chulalongkorn University. It has been nice to be here and work with them.

Finally, my appreciation goes to all those individuals who are (and also those who are not) mentioned and who contributed to my knowledge and understanding, and as such find their contribution reflected in my thesis.

CONTENTS

Page

ABSTRACT (THAI)..iv

ABSTRACT (ENGLISH)..v

ACKNOWLEDGEMENT...vi

CONTENTS...vii

LIST OF TABLES...x

LIST OF FIGURES..xi

LIST OF ABBREVIATIONS..xii

CHAPTER I INTRODUCTION..1

1.1 Important and reasons for research...1

1.2 Research objectives..2

1.3 Scope of research..2

1.4 Contribution of research..3

1.5 Methodology..3

CHAPTER II LITERATURE REVIEW...5

2.1 Mathematic model of fouling...4

2.2 Optimization of cleaning schedule...5

CHAPTER III THEORIES AND PRINCIPLES...6

3.1 Catalytic cracking...6

3.1.1 FCC feed pretreating...8

3.1.2 Hydrotreating..9

3.1.3 Hydrotreating catalyst...11

3.1.4 Aromatics reduction..12

3.1.5 Reactions...13

3.2 Fouling..15

3.2.1 Impact of fouling...15

3.2.2 Causes of fouling...16

3.2.3 Fouling mechanisms..18

3.3 Basic knowledge of heat exchanger...20

Page

3.3.1 Overall heat transfer coefficient....21
3.3.2 Fouling factor....22
3.3.3 The log mean temperature different....23
3.4 Artificial neural network....24
3.4.1 Introduction of an artificial neural network....24
3.4.2 Components of an artificial neural network....26
3.4.2.1 Weighting factors....26
3.4.2.2 Summation function or basis function....26
3.4.2.3 Transfer function or activation function....27
3.4.2.4 Scaling....28
3.4.2.5 Output function....29
3.4.2.6 Error function....29
3.4.2.7 Learning function....30
3.4.3 Architecture of an artificial neural network....30
3.4.3.1 Network structures....30
3.4.3.2 Connection structures....31
3.4.3.3 Network layer....31
3.4.4 Learning algorithm of an artificial neural network....32
3.4.4.1 Supervised learning....32
3.4.4.2 Unsupervised learning....33
3.4.5 Multilayer feed-forward neural network....33
3.4.6 Design of artificial neural network....34
3.4.6.1 Structure and size of network....34
3.4.6.2 Data collection....34
3.4.6.3 Data processing....35
3.4.6.4 Weight initialization....35
3.4.6.5 Training the network....35
3.4.6.6 Model validation....36

Page

3.4.6.7 Basic step of an artificial neural network design..............................37

3.4.7 Application of an artificial neural network.......................................38

CHAPTER IV SYSTEM MODELING APPLICATION..39

4.1 Description of the process...39

4.2 Mathematical fouling model..40

4.2.1 Polley fouling model...40

4.2.2 Mohammad fouling model...41

4.2.3 Evaluation of the proposed model...41

4.3 Preliminary study of single heat exchanger...44

4.3.1 Heat exchanger model description...44

4.3.2 Hot and cold streams outlet temperature..44

4.3.3 Overall heat transfer coefficient...46

4.4 Neural network estimator...46

4.4.1 Neural network training...47

4.4.2 Design of a neural network estimator...49

4.4.3 Results and discussions...55

CHAPTER V OPTIMIZATION OF CLEANING SCHEDULE FOR FEED/EFFLUENT EXCHANGER..56

5.1 Optimization of cleaning schedule for single heat exchanger.............................58

5.1.1 The influence of number of cleaning period on the optimal of cleaning schedule...59

5.1.1.1 The influence of number of cleaning period on the optimal of cleaning schedule with 2 years of operation time............................59

5.1.1.2 The influence of number of cleaning period on the optimal of cleaning schedule with 3 years of operation time............................61

5.1.2 The influence of cost of furnace extra fuel on the optimal of cleaning schedule...63

5.1.2.1 The influence of cost of furnace extra fuel on the optimal of

Page

cleaning schedule with 2 years of operation time.................................63

5.1.2.2 The influence of cost of furnace extra fuel on the optimal of

cleaning schedule with 3 years of operation time................................65

5.2 Summary results for optimal of cleaning schedule...67

CHAPTER VI CONCLUSION AND RECOMMENDATION.......................................69

6.1 Conclusion...69

6.2 Recommendation...70

REFERENCES...71

APPENDICES...73

APPENDIX A...74

APPENDIX B...78

APPENDIX C...83

VITA..93

LIST OF TABLES

Page

Table 3.1 Thermal versus catalytic cracking yields on similar topped crude feed..........7

Table 3.2 Polymerization Mechanisms..........18

Table 3.3 Transfer functions of an artificial neural network..........28

Table 3.4 Learning algorithms of an artificial neural network..........32

Table 4.1 Polley fouling model constants comparison with actual process..........41

Table 4.2 Mohammad fouling model constants comparison with actual process..........42

Table 4.3 Summation of square errors between proposed model and actual process..42

Table 4.4 The transfer functions in each layer of the neural network estimator..........47

Table 4.5 The neuron network model with ten hidden nodes..........49

Table 4.6 The neuron network model with fifty hidden nodes..........50

Table 4.7 The neuron network model with one hundred hidden nodes..........51

Table 4.8 The neuron network model with five hundred hidden nodes..........52

Table 4.9 The mean square error for the various architectures of the neural network.55

Table 5.1 Model parameter for single heat exchanger case..........59

Table 5.2 The comparison of number of cleaning and operating cost
within 2 years of operation time..........60

Table 5.3 Model parameter for single heat exchanger case..........61

Table 5.4 The comparison of number of cleaning and operating cost
within 3 years of operation time..........61

Table 5.5 Model parameter for single heat exchanger case..........63

Table 5.6 Optimal cleaning schedule with different cost of furnace extra fuel
within 2 years of operation time..........64

Table 5.7 Model parameter for single heat exchanger case..........65

Table 5.8 Optimal cleaning schedule with different cost of furnace extra fuel
within 3 years of operation time..........65

Table 5.9 Summary result for 2 years optimal cleaning schedule
with 340 Baht/(MKJ/h)..........68

Page

Table 5.10 Summary result for 3 years optimal cleaning schedule with 340 Baht/(MKJ/h)....................68

Table 5.11 Summary result for 2 years optimal cleaning schedule with 410 Baht/(MKJ/h)....................68

Table 5.12 Summary result for 3 years optimal cleaning schedule with 410 Baht/(MKJ/h)....................68

LIST OF FIGURES

Page

Figure 3.1 Catalytic hydrodesulfurization unit..10

Figure 3.2 Hydrotreater Fouling - Cause / Effect Diagram..16

Figure 3.3 Components of biological neural network..25

Figure 3.4 The structure of multilayer feed-forward neural network............................34

Figure 3.5 Basic steps of an artificial neural network design...37

Figure 4.1 Process flow diagram for feed/effluent exchanger.....................................39

Figure 4.2 Comparison of fouling rate of the Polley model and actual process.............42

Figure 4.3 Comparison of fouling rate of the Mohammad model and actual process.....43

Figure 4.4 The diagram of heat exchanger...44

Figure 4.5 Neural network configurations for modeling...48

Figure 4.6 The validation performance with ten hidden nodes...49

Figure 4.7 The validation performance with fifty hidden nodes.......................................50

Figure 4.8 The validation performance with one hundred hidden nodes........................51

Figure 4.9 The validation performance with five hundred hidden nodes 52

Figure 4.10 The difference value with ten hidden nodes...53

Figure 4.11 The difference value with fifty hidden nodes..53

Figure 4.12 The difference value with one hundred hidden nodes 54

Figure 4.13 The difference value with five hundred hidden nodes.................................54

Figure 5.1 Process flow diagram for feed/effluent exchanger.....................................56

Figure 5.2 The comparison of number of cleaning
and operating cost within 2 years...60

Figure 5.3 The comparison of number of cleaning and operating cost
within 3 years of operation time..62

Figure 5.4 The comparison of operating cost in the different cost
of furnace extra fuel within 2 years of operation time...................................64

Figure 5.5 The comparison of operating cost in the different cost
of furnace extra fuel within 3 years of operation time...................................66

LIST OF ABBREVIATIONS

U	Overall heat transfer coefficient, KJ/h m^2 ^{o}C
A	Total area available for heat transfer, m^2
R	Thermal resistance, m^2 h ^{o}C/KJ
ΔT_{lm}	Log mean temperature, ^{o}C
$T_{c,in}$	Inlet temperature for cold stream, ^{o}C
$T_{c,out}$	Outlet temperature for cold stream, ^{o}C
$T_{h,in}$	Inlet temperature for hot stream, ^{o}C
$T_{h,out}$	Outlet temperature for hot stream, ^{o}C
T_w	Tube wall temperature, ^{o}C
T_f	Tube side surface temperature, ^{o}C
Re	Reynold number
Pr	Prandtl number
E	Activation Energy, KJ/mol
Q_c	Heat transfer rate for cold stream, KJ/h
Q_h	Heat transfer rate for hot stream, KJ/h
$C_{p,c}$	Specific heat for cold stream, KJ/(Kg/K)
$C_{p,h}$	Specific heat for hot stream, KJ/(Kg/K)
NP	The number of cleaning period
C_E	The cost of furnace extra fuel, Baht/(KJ/h)
C_{cl}	The cost of heat exchanger cleaning, Baht/unit

CHAPTER I

INTRODUCTION

1.1 Important and reasons for research

In an integrated refinery, hydrotreating unit helps insure maximum yield of useful products from every barrel of crude oil processed. Hydrotreaters cover a broad class of units, which have two main functions. The first is to stabilize refined products by adding hydrogen to unsaturated compounds. Unsaturated or olefinic compounds are very reactive and can cause undesired polymerization products in downstream processes or product streams. The second function is to remove impurities such as metal contaminants and sulfur, oxygen, nitrogen, and halide compounds from process streams, which could be detrimental to downstream processes.

Fouling can be described as the process in which entrained particulates, or those formed in the process stream, deposit onto process equipment. This deposition can have a significant, negative impact on the operational efficiency of the unit. In hydrotreater, loss of heat transfer and/or increased pressure drop are the most obvious results of fouling in the preheat exchangers, while fouling of the reactor bed results in increased pressure drop. Hydrotreater fouling presents refiners with four major concerns: increased energy costs, increased maintenance costs, throughput limitations, and restrictions to operating flexibility. Thus, to solve these problems, heat exchangers are cleaned between shutdowns or during operations by isolating the exchanger through bypasses.

Planning of heat exchanger cleaning during operations is very important. On the one hand, cleaning results in less energy costs over the time horizon after it is cleaned, but it also implies that the exchanger needs to be put offline during cleaning and therefore in this period of time the cost actually increases. Thus, while cleaning is advantageous, doing it too often may not be economically advisable after all.

The artificial neural network is widely applied in modeling and optimization of chemical process especially nonlinear systems. The main advantage of the use of artificial neural network is obtaining a highly accurate mathematical model of the system without the detail of the system. The process modeling applications use the artificial neural network to approximate the relationship between the input and output variables. During the process modeling, a number of candidate models are considered and only one model, which is expected to the best prediction of the process outputs with the given process inputs, is selected. The selected model is the one that is expected to have the least prediction error in the future. In addition, artificial neural network is also the universal function approximator that typically works better than the traditional function approximation method for the application of any arbitrary system.

In this research, an artificial neural network has been used to develop the model of feed/effluent heat exchanger fouling in Hydrotreating unit. Developed algorithm is employed to determine the optimal operation and cleaning interval of the feed/effluent heat exchanger surfaces.

1.2 Research objectives

The objectives of this research are:

1. Develop the empirical models of feed/effluent heat exchanger fouling in Hydrotreating unit by using neural network software.
2. To determine the optimal operation and cleaning schedule of the feed/effluent heat exchanger.

1.3 Scope of research

The scope of this research are presented as following:

1. Hydrotreating unit in Fluid Catalytic Cracking plant (FCC) are studied.
2. Neural networks software is used to develop the empirical model.
3. The optimal operation and maintenance will be determined for feed/effluent heat exchanger in Hydrotreating unit under fouling condition.

1.4 Contribution of research

The contribution of this research are as following:

1. We proposed an appropriate model of feed/effluent heat exchanger in Hydrotreating unit under fouling condition.
2. We developed algorithm to determine the optimal operation and cleaning schedule of the feed/effluent heat exchanger.

1.5 Methodology

Methodology of this research can be described as following:

1. The first plan, various literatures related to mathematical model for heat exchanger fouling and optimal model for cleaning and maintenance were studied.
2. Thoroughly study process description and operating condition of Hydrotreating unit.
3. We collected the operating data of Hydrotreating unit in FCC plant.
4. We applied the mathematical models and developed empirical models of feed/effluent heat exchanger in Hydrotreating unit under fouling condition.
5. We verified the developed models with plant operating data.
6. We determined the optimal operation and maintenance by optimizing the developed models.
7. We summarized and concluded our research.

CHAPTER II

LITERATURE REVIEW

In recent years, the fouling is the major problem in the heat exchanger. It reduces heat transfer rate causing in the reduction of heat exchanger performance. There are many effective fouling mitigations techniques such as using antifouling chemical, increasing the heat transfer surfaces and regular cleaning. The combination of these techniques applies in order to minimize the overall operating cost. In general, it is necessary to determine the optimum cleaning schedule for heat exchanger in order to improved heat exchanger performance. This chapter provides a review of the optimization of cleaning schedule of heat exchanger network.

2.1 Mathematic model of fouling

The simulation of fouling in heat exchanger for predicting the variation of overall heat transfer coefficient and variations of outlet temperature of hot and cold streams is important. These variations have a significant effect on production rate and operating cost. The main goal is to find the variation of overall heat transfer coefficient with time to plan the optimum cleaning schedule of heat exchanger that provide the minimum operating cost.

In the past, G.T. Polley *et al.* (2002) presented a correlation study of the data set of fouling thresholds in crude oil. The fouling threshold concept was first proposed by Ebert and Panchal, using fouling data obtained from pilot plant and refinery side-stream monitoring tests. Knudsen et al. showed the existence of a true threshold for Alaskan crude oil tested in a series of laboratory pilot plant experiments. This paper discusses the application of the original ‘threshold model’ forms to the important data set of Kundsen et al., which is almost unique in reporting data below the fouling threshold. A logical, semi-empirical formulation is developed for use in further studies. Comparison between predictions of the Ebert-Panchal model and experimental data is difficult in the absence of information on the physical properties of the oils. Not only do the physical properties vary widely between individual crude, but also their temperature dependencies and reactivity being strongly determined by composition can differ noticeably. Consequently, the use of a threshold model to predict fouling rates may not be appropriate. Therefore, the experiment results of Knudsen et al. clearly demonstrate the existence of a fouling threshold, postulated by Ebert and Panchal, for the crude oil tested. The Ebert-Panchal model does not provide a good prediction of the conditions for the onset of fouling observed in these tests, mainly because it features a much greater sensitivity to velocity than was observed. However, this model does provide reasonable predictions of subsequent fouling rates for data presented by Knudsen et al. By adjusting the activation energy parameter, the model also provides good predictions for data from other studies. In additional, Mohammad Reza Jafari Nasr, Mehdi Majidi Givi (2006), proposed a new model for crude oil fouling in preheat exchangers of crude distillation units. Their experimental results of Australian light crude oil with the tube side surface temperature and fluid velocity were used. The amount of activation energy depends on the surface temperature has been calculated. A new model including a term for fouling formation and a term for fouling removal due to chemical and tube wall shear stress was proposed, respectively. To propose a new model the experimental results reported by Seleh et al. were used. The model for crude oil fouling proposed by Seleh et al. can only able to predict fouling without considering the effect of fluid velocity on the fouling removal. They used portable fouling research unit (PFRU) fitted with an annular HTRI heat transfer probe, which was operated at constant heat flux with time and all runs were conducted in the transition region. For comparing the proposed model, the constants of models were calculated based on the experimental results. It can be seen that the deviation of the proposed model from the experimental results is lower than the others. The proposed model results a fair agreement in comparison with the experimental data. It is revealed than among the models to predict the fouling formation, one considered formation and removal of fouling layer has a great importance. Fortunately, the proposed model takes this advantage. To use the model and also to calculate the activation energy the fouling rate should be available. As shown the model can predict fouling rate for various crude and in different conditions better than other models.

2.2 Optimization of cleaning schedule

The optimization of cleaning schedule in heat exchanger network has been considered until recently. In the past, Casado *et al.* (1990) proposed a model based on the cost of cleaning the fouled exchangers. The asymptotic fouling model for counter current flow exchangers and thermal analysis of the hot and cold streams are implemented in this work. This work explained the costs of fouling and proposed a time dependent objective and cost function in the process operation. The cleaning plan based on minimization of the process operation cost. Next, Michael C. Georgiadis, Lazaros G. Papageorgiou (2001) proposed a mathematical programming framework for introduction of fouling consideration during the heat integration of batch plant operation. A short-term scheduling problem is considered that seeks to determine the optimal utilization of the available plant resourced over a given time horizon. A characteristic of this problem is that the performance of each heat-integrated unit, which decreases with time due to fouling, can be restored to its initial state by performing cleaning operations. The overall problem is formulated as a mixed integer non-linear programming (MINLP) model. It is proved that fouling considerations can significantly affect the production schedule as well as heat integration opportunities. This work has been concentrated on the introduction of fouling considerations in the heat integration to a short-term scheduling formulation, the approach presented is equally applicable to periodic scheduling. In additional, Javier H. Lavaja, Miguel J. Bagajewicz (2004), presented a new mixed-integer linear model for the planning of heat exchanger cleaning in chemical plants. The model maximizes the net present value based on the cost of cleaning and the cost of energy and takes into account changes in production rates and even changes in the properties and flows of the different streams throughout time. This is important for the case of petroleum distillation in refineries that process different types of crude. The model is multi-period and uses two different fouling models. Consequently, Markowski, M. *et al.* (2005) illustrated the optimization of cleaning schedule for heat exchanger network. Heat exchanger cleaning is postulated to maximize the avoid loss understood as the value of energy recovered of cleaning the heat exchanger network, minus the value of energy recovered without heat exchanger network cleaning, minus the cost of heat exchanger network cleaning. The result shown that the value of energy recovered is affected by the specific cost of energy. The cost of heat exchanger network cleaning depends on the cost of cleaning intervention on a specific heat exchanger. The formulation is both integer and continuous decision variables and the function is mixed integer nonlinear programming problem. For a large heat exchanger network may require a prohibitively large computational effort. But an approximation solution can be obtained by maximizing a nonlinear function in many integer variables. Moreover, M Kuosa *et al.* (2007) studied the effect of heat exchanger fouling on the performance of Stirling engine and determine the optimal cleaning. The optimal cleaning can be determined by adding the costs for one period that includes the time for the fouling and time for cleaning. The optimization is based on determining the fouling time that yields the minimum cost flow. Cleaning time is assumed independent of the fouling time. The cleaning is performed for both heat exchangers at the same time.

CHAPTER III

THEORIES AND PRINCIPLES

3.1 Catalytic Cracking

Catalytic cracking is the most important and widely used refinery process for converting heavy oils into more valuable gasoline and lighter products. Originally cracking was accomplished thermally but the catalytic process has almost completely replaced thermal cracking because more gasoline having a higher octane and less heavy fuel oils and light gases are produced. The light gases produced by catalytic cracking contain more olefins than those produced by thermal cracking (Table 3.1).

Table 3.1 Thermal Versus Catalytic Cracking Yields on Similar Topped Crude Feed

	Thermal cracking		Catalytic cracking	
	wt%	vol%	wt%	vol%
Fresh feed	100.0	100.0	100.0	100.0
Gas	6.6		4.5	
Propane	2.1	3.7	1.3	2.2
Propylene	1.0	1.8	2.0	3.4
Isobutane	0.8	1.3	2.6	4.0
n-Butane	1.9	2.9	0.9	1.4
Butylene	1.8	2.6	2.6	3.8
C_5 + gasoline	26.9	32.1	40.2	46.7
Light cycle oil	1.9	1.9	33.2	32.0
Decant oil			7.7	8.7
Residual oil	57.0	50.2		
Coke	0		5.0	
Total	100.0	96.5	100.0	102.2

The cracking process produces carbon (coke), which remains on the catalyst particle and rapidly lowers its activity. To maintain the catalyst activity at a useful level, it is necessary to regenerate the catalyst by burning off this coke with air. As a result, the catalyst is continuously moved from reactor to regenerator and back to reactor. The cracking reaction is endothermic and the regeneration reaction exothermic.

Some units are designed to use the regeneration heat to supply that needed for the reaction and to heat the feed up to reaction temperature. These are known as “heat balance” units.

Average riser reactor temperatures are in the range 900 to 1000^{o}F (480-540^{o}C), with oil feed temperatures from 500 to 800^{o}F (260-425^{o}C) and regenerator exit temperatures for catalyst from 1200 to 1500^{o}F (650-815^{o}C).

The catalytic-cracking processes in use today can all be classified as either moving-bed or fluidized-bed units. There are several modifications under each of the classes depending upon the designer or builder. The Thermafor catalytic cracking process (TCC) is representative of the moving-bed units and the fluid catalytic cracking (FCC) of the fluidized-bed units. There are very few TCC units in operation today and the FCC unit has taken over the field. The FCC units can be classified as either bed or riser (transfer line) cracking units depending upon where the major fraction of the cracking reaction occurs.

The process flows of both types of processes are similar. The hot oil feed is contacted with the catalyst in either the feed riser line or the reactor. As the cracking reaction progresses, the catalyst is progressively deactivated by the formation of coke on the surface of the catalyst. The catalyst and hydrocarbon vapors are separated mechanically, and oil remaining on the catalyst is removed by steam stripping before the catalyst enters the regenerator. The oil vapors are taken overhead to a fractionation tower for separation into streams having the desired boiling ranges.

The spent catalyst flows into the regenerator and is reactivated by burning off the coke deposits with air. Regenerator temperatures are carefully controlled to prevent catalyst deactivation by overheating and to provide the desired amount of carbon burn0off. This is done by controlling the air flow to five a desired CO_2/CO ratio in the exit flue gases or the desired temperature in the regenerator. The flue gas and catalyst are separated by cyclone separators and electrostatic precipitators. The catalyst in some units is steam-stripped as it leaved the regenerator to remove adsorbed oxygen before the catalyst is contacted with the oil feed.

3.1.1 FCC Feed Pretreating

The trend toward low sulfur and nitrogen contents in gasolines and diesel fuels requires that either the FCC unit feed or products be treated to reduce sulfur and nitrogen.

Treating feed to the FCC unit offers the advantages that the sulfur and nitrogen in the gasoline and diesel fuel products are reduced and, by adding hydrogen to the feed, naphtha and LCO yields are increased without lowering the olefins content and octanes of the naphtha fraction. For refineries that do not hydrotreat the FCC feed or naphtha products, over 95% of the sulfur in the gasoline blending pool is from the FCC naphtha.

The hydrotreating unit can be operated in several ways: as a hydrodesulfurization (HDS) unit, a mild hydrocracking (MHC) unit, or a partial-conversion hydrocracking unit. In all cases the product sulfur content has to be less that 135 wppm to produce a refinery gasoline blending pool with less than 50 wppm sulfur and less than 85 wppm to produce a refinery gasoling blending pool of less than 30 wppm.

3.1.2 Hydrotreating

The terms hydrotreating, hydroprocessing, hydrocracking, and hydrodesulfurization are used rather loosely in the industry because, in the processes hydrodesulfurization and hydrocracking, cracking and desulfurization operations occur simultaneously and it is relative as to which predominates. In this paper, hydrotreating refers to a relatively mild operation whose primary purpose is to saturate olefins and/or reduce the sulfur and/or nitrogen content (and not to change the boiling range) of the feed. Hydrocracking refers to processes whose primary purpose is to reduce the boiling range and in which most of the feed is converted to products with boiling range lower than that of the feed. Hydrotreating and hydrocracking set the two ends of the spectrum and those processes with a substantial amount of sulfur and/or nitrogen removal and a significant change in boiling rage of the products versus the feed are called hydroprocessing in this paper.

Hydrotreating is a process to catalytically stabilize petroleum products and/or remove objectionable elements from products or feedstocks by reacting them with hydrogen. Stabilization usually involves converting unsaturated hydrocarbons such as olefins and gum-forming unstable diolefins to paraffins. Objectionable elements removed by hydrotreating include sulfur, nitrogen, oxygen, halides, and trace metals. Hydrotreating is applied to a wide range of feedstocks, from naphtha to reduce crude. When the process is employed specifically for sulfur removal it is usually called hydrodesulfurization, or HDS. To meet environmental objectives it also may be necessary to hydrogenate aromatic rings to reduce aromatic content by converting aromatics to paraffins.

Although there are about 30 hydrotreating processes available for licensing, most of them have essentially the same process flow for a given application. Figure 3.1 illustrates a typical hydrotreating unit.

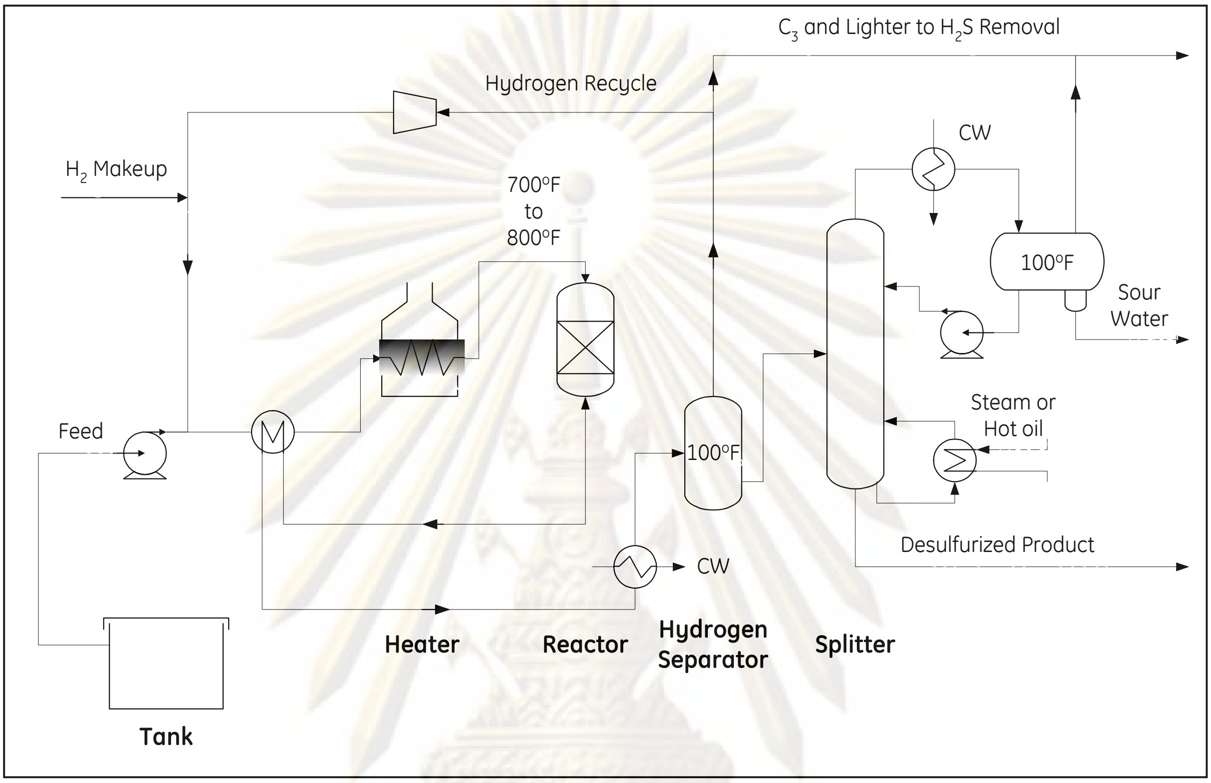

Figure 3.1 Catalytic hydrodesulfurization unit

The oil feed is mixed with hydrogen-rich gas either before or after it is preheated to the proper reactor inlet temperature. Most hydrotreating reactions are carried out below 800°F (427°C) to minimize cracking, and the feed is usually heated to between 500 and 800°F (260-427°C). The oil feed combined with the hydrogen-rich gas enters the top of the fixed-bed reactor. In the presence of the metal-oxide catalyst, the hydrogen reacts with the oil to produce hydrogen sulfide, ammonia, saturated hydrocarbons, and free metals. The metals remain on the surface of the catalyst and other products leave the reactor with the oil-hydrogen stream. The reactor effluent is cooled before separating the oil from the hydrogen-rich gas. The oil is stripped of any remaining hydrogen sulfide and light ends in a stripper. The gas may be treated to remove hydrogen sulfide and recycled to the reactor.

3.1.3 Hydrotreating Catalyst

Catalyst developed for hydrotreating include cobalt and molybdenum oxides on alumina, nickel oxide, nickel thiomolybdate, tungsten and nickel sulfides, and vanadium oxide. The cobalt and molybdenum oxides on alumina catalysts are in most general use today because they have proven to be highly selective, easy to regenerate, and resistant to poisons. They must be activated by converting the hydrogenation metals from the oxide to the sulfide form.

If, however, the removal of nitrogen is a significant consideration, catalysts composed of nickel-cobalt-molybdenum or nickel-molybdenum compounds supported on alumina are more efficient. Nitrogen is usually more difficult to remove than sulfur from hydrocarbon streams, and any treatment which reduces excess nitrogen concentration to a satisfactory level usually will effectively remove excess sulfur. Nickel-containing catalysts generally require activation by presulfiding with carbon disulfide, mercaptans, or dimethyl sulfide before bringing up to reaction temperature; however, some refiners activate these cobalt-molybdenum catalysts bu injecting the sulfiding chemical into the oil feed during startup. The sulfiding reaction is highly exothermic and care must be taken to prevent excessive temperatures during activation.

Cobalt-molybdenum catalysts are selective for sulfur removal and nickel-molybdenum catalysts are selective for nitrogen removal, although both catalysts will remove both sulfur and nitrogen. Nickel-molybdenum catalysts have a higher hydrogenation activity than cobalt-molybdenum, which results, at the same operating conditions, in a greater saturation of aromatic rings. Simply stated, if sulfur reduction is the primary objective, then a cobalt-molybdenum catalyst will reduce the sulfur a given amount at less severe operating conditions with a lower hydrogen consumption than nickel-molybdenum catalyst. If nitrogen reduction or aromatic ring saturation is desired, nickel-molybdenum catalyst is the preferred catalyst.

The ability to adjust pore size to concentrate pores around a particular diameter has a great impact on the hydrotreating activity both at start-of-run (SOR) and as the catalyst ages. Reactions taking place in the hydrotreating of gas oils [400-1050^{o}F (200-566^{o}C)] generally require a minimum pore size to overcome most diffusional restrictions. Pores that are larger than necessary lend little to improving diffusional characteristics and as the pore diameters of the catalyst increase the surface area decreases (at constant pore volume).

Activity generally decreases with surface area and loss in pore volume occurs in the smallest diameter pores first. Highest activity retention is maintained if pore volume is concentrated in a very narrow range of pore diameters.

At the hydrotreating severity to reduce sulfur in LCO to 0.05 %wt, the performance of high-activity NiMo and CoMo catalysts appears to be equivalent.

Catalyst consumption varies from 0.001 to 0.007 Ib/bbl (0.003 to 0.02 kg/m^3) feed depending upon the severity of operation and the gravity and metals content of the feed.

3.1.4 Aromatics Reduction

Hydrogen partial pressure is the most important parameter controlling aromatic saturation. Depending on type of feedstock, the required hydrogen partial pressure to reduce aromatic content to 10 vol% may vary as much as 40%. Several investigators have shown that a LHSVs if 2.0, aromatics in diesel fuel blending stocks can be reduced to <10 vol% only at pressure of 1500 psig (10.4 MPa) or greater.

Hydrogenation is an exothermic reaction and equilibrium yields are favored by low temperatures. Reaction rated increase with temperature, and hydrogenation of aromatic ring compounds is a compromise between using low reactor temperatures to achieve maximum reduction of aromatic content and a high temperature to give high reaction rates and a minimum amount of catalyst charge per barrel of feed. Maximum aromatic reduction is achieved between 700-750^{o}F (370-400^{o}C) [usually between 705-725^{o}F (375-385^{o}C)] because of the interrelation between thermodynamic equilibrium and reaction rates.

High-pressure single-stage hydrotreating of only the front end [400-550^{o}F (205-288^{o}C)] of a LCO reduced hydrogen consumption and extended catalyst life. Usually this fraction originally contains about 11.1 wt% mono-aromatics and 17.5 wt% di-aromatics. Hydrogenation at 1200 psig (8.2 MPa) reduces the di-aromatic content to 0.4 wt% and increases the mono-aromatic content to 18.3 wt%. Saturation of the final aromatic ring is difficult because of the resonance stabilization of the mono-aromatic ring. Hydrogenation at 1500 psig (10.3 MPa) is required to reduce the aromatic content to 10 wt%, but only about 1/3 as much hydrogen is required as compared to reducing the aromatic content of the full-range [400-650^{o}F (205-345^{o}C)] LCO. This is because the back end of the LCO contains only di- and tri$^+$ -aromatics and the front end contains almost all of the mono-aromatics, about 1/3 of the di-aromatics, and none of the tri-aromatics in the LCO.

Hydrotreating the feed to the FCC unit reduces the sulfur contents of the FCC products but also increases the aromatic content of the LCO (probably because the percentage of mono-aromatic compounds in the feed is increased). Hydrotreating the FCC feed also makes it more difficult to reduce the aromatics content of the LCO to <20 vol%.

3.1.5 Reactions

The main hydrotreating reaction is that of desulfurization but many others take place to a degree proportional to the severity of the operation. Typical reactions are:

- *Desulfurization*

  a. Mercaptans:

  $$RSH + H_2 \longrightarrow RH + H_2S \qquad (3.1)$$

  b. Sulfides:

  $$R_2S + 2H_2 \longrightarrow 2RH + H_2S \qquad (3.2)$$

  c. Disulfides:

  $$(RS)_2 + 3H_2 \longrightarrow 2RH + H_2S \qquad (3.3)$$

- *Denitrogenation*

  a. Pyrrole:

  $$C_4H_4NH + 4H_2 \longrightarrow C_4H_{10} + NH_3 \qquad (3.4)$$

  b. Pyridine:

  $$C_5H_5N + 5H_2 \longrightarrow C_5H_{12} + NH_3 \qquad (3.5)$$

- *Deoxidation*

  a. Phenol:

  $$C_6H_5OH + H_2 \longrightarrow C_6H_6 + H_2O \qquad (3.6)$$

b. Peroxides:

$$C_7H_{13}OOH + 3H_2 \longrightarrow C_7H_{16} + 2H_2O \qquad (3.7)$$

- *Dehalogenation*

  Chlorides:

$$RCl + H_2 \longrightarrow RH + HCl \qquad (3.8)$$

- *Hydrogenation*

  Pentene:

$$C_5H_{10} + H_2 \longrightarrow C_5H_{12} \qquad (3.9)$$

- *Hydrocracking*

$$C_{10}H_{22} + H_2 \longrightarrow C_4H_{10} + C_6H_{14} \qquad (3.10)$$

The case of desulfurization is dependent upon the type of compound. Lowering-boiling compounds are desulfurized more easily than higher-boiling ones. The difficulty of sulfur removal increases in the order paraffins, naphthenes, and aromatics.

Nitrogen removal requires more severe operating conditions than does desulfurization. For middle distillate fractions from crude oils containing high concentrations of nitrogen compounds, more efficient nitrogen reduction is achieved by using a catalyst charge of 90% nickel-molybdenum and 10% nickel-tungsten.

All reactions are exothermic and, depending on the specific conditions, a temperature rise through the reactor of 5 to 20^{o}F (3 to 11^{o}C) is usually observed.

3.2 Fouling

3.2.1 Impact of Fouling

Fouling can be described as the process in which entrained particulates, or those formed in the process stream, deposit onto process equipment. This deposition can have a significant, negative impact on the operational efficiency of the unit. In hydrotreater, loss of heat transfer and/or increased pressure drop are the most obvious results of fouling in the preheat exchangers, while fouling of the reactor bed results in increased pressure drop. Hydrotreater fouling presents refiners with four major concerns: increased energy costs, increased maintenance costs, throughput limitations, and restrictions to operating flexibility.

One direct economic penalty of preheat exchanger fouling is the cost of additional furnace fuel gas, required to bring the feed up to reactor temperature. Hydrotreater fouling, in either the preheat exchangers or the reactor can significantly increase maintenance costs. Ideally, maintenance should only be required when the unit is shut down to replace spent catalyst. The penalty of replacing catalyst prematurely is additional maintenance costs and the cost of the catalyst. In today's economy, shutting down a unit for the sole purpose of unplugging reactor beds or cleaning exchangers is a process interruption few refiners can afford. Fouling in the preheat exchangers and reactor bed can impact unit throughput. The reactor inlet temperature is critical to the reaction of hydrogen with the unsaturated compounds and contaminants in the feed. After the preheat exchangers transfer as much heat as possible to the feed, the furnace provides the additional heat needed to achieve the reactor inlet temperature. If the preheat exchangers cannot transfer sufficient heat to the feed, the furnace may lack the capacity to heat the feed to the necessary temperature. If this happens, charge rates will have to be reduced in order to effectively hydrotreat the feed; otherwise, product quality may be impacted. Fouling can also increase the pressure drop across the preheat exchangers and/or reactor bed. This increased pressure drop may result in charge rates being reduced due to a hydraulic limit. Finally, efficient hydrotreater operation allows the refiner to capitalize on market opportunities by upgrading lower quality feeds to higher profitability products. Seasonal demands for these upgraded products can dramatically influence unit and refinery profit margins. The operating flexibility achieved by reducing fouling in a hydrotreater becomes an important consideration to today's refiner.

3.2.2 Causes of Fouling

Many variables can impact a feed stream's fouling potential and severity in the hydrotreater pre-heat exchangers and reactor bed. Feed composition, feed storage and handling, unit design, and operating parameters can each have a significant influence on fouling. A thorough understanding of how these variables, individually and collectively, impact fouling is essential to identifying cost-effective solutions. A cause and effect diagram (Figure 2) presents an overview of many of the factors which influence hydrotreater fouling. An operational change, or a process design modification may often minimize a variable's influence on fouling; however, in many cases, a more cost-effective solution may be chemical treatment.

The way a stream is stored and handled prior to being charged to the hydrotreater can be an important key to understanding a feed's fouling potential. Unstable feeds sent to tankage have a tendency to foul more, primarily due to the potential for oxygen contamination. Unstable feeds readily react with oxygen to form peroxides, which can polymerize, resulting in fouling in the hydrotreater unit. Some integrated refineries have the ability to send the feed directly to the hydrotreater from another unit; therefore, by-passing tankage. Feeds that have bypassed tankage typically foul less than feeds from tankage or imported feeds, since there is less exposure to oxygen and a decreased time for polymerization reactions to occur.

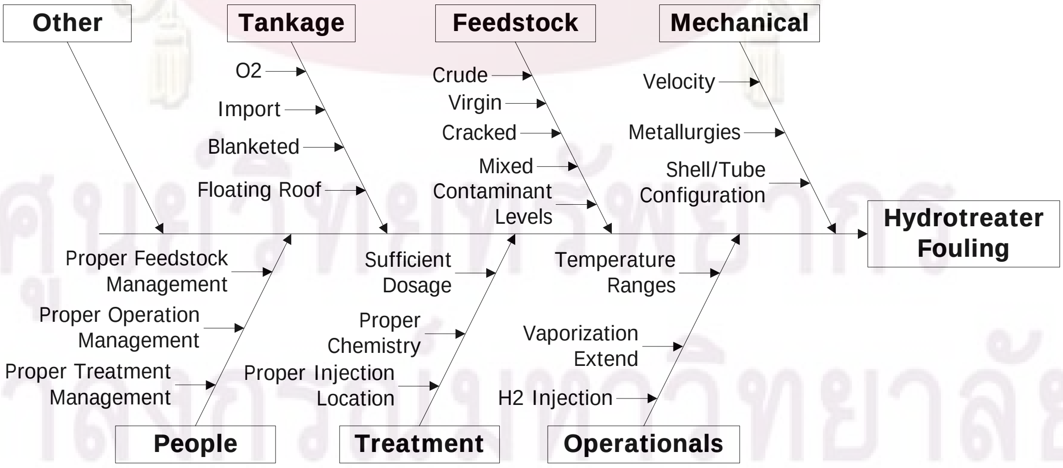

Figure 3.2 Hydrotreater Fouling - Cause / Effect Diagram

A wide range of streams, from naphtha to resids, can make up the feed to hydrotreaters. Cracked heavier gas oils and resids usually have a greater fouling tendency than a straight run naphtha or kerosene stream. The potential for a stream to foul is, in part, a function of the various trace compounds, which are part of the feed. Consequently, knowing the type and source of the various hydrotreater feeds is an important first step in identifying potential fouling precursors. The way a stream is stored and handled prior to being charged to the hydrotreater can be an important key to understanding a feed's fouling potential. Unstable feeds sent to tankage have a tendency to foul more, primarily due to the potential for oxygen contamination. Unstable feeds readily react with oxygen to form peroxides, which can polymerize, resulting in fouling in the hydrotreater unit.

Some integrated refineries have the ability to send the feed directly to the hydrotreater from another unit; therefore, by-passing tankage. Feeds, which have bypassed tankage typically, foul less than feeds from tankage or imported feeds, since there is less exposure to oxygen and a decreased time for polymerization reactions to occur. The mechanical design of the unit can also have a definite impact on the rate of fouling. However, a unit redesigns or modification may be too costly to be justified by just reduced fouling potential. Units which process feeds different from those for which they were designed may have an increased fouling tendency. This can specially be seen in units operating at lower than design capacity. Lower than design velocities will increase the deposition potential of any particulates present in the feed stream. Changes in the hydrogen feed rate or its impact on feed vaporization can also change the nature of fouling.

More severe operating conditions, designed to increase hydrotreating conversion, can also contribute to fouling by increasing the polymerization potential associated with higher operating temperatures. When operating at these elevated temperatures, the complete vaporization of the feed in the preheat exchangers may result in particulates depositing on the exchanger surfaces at that point.

An intangible in the operation of the unit and its potential for fouling is the management and control of the unit operations. It is essential that operations be aware of the various feed sources and the impact of process conditions on fouling. Feeds and operating conditions will change throughout a run, and can cause a unit to foul severely in a short period of time. Hydrotreaters have been known to foul in less than 60 days. Chemical treatment is being used to control hydrotreater fouling, it is important that the appropriate chemistry be selected to address the specific fouling mechanism. The chemical treatment should also be injected in the proper location and streams to maximize its benefit in reducing the overall fouling potential.

Table 3.2 Polymerization Mechanisms

<table>
<tr><th colspan="6">Free Radical Polymerization</th></tr>
<tr><td colspan="2">Radical Formation</td><td colspan="2">Propagation</td><td colspan="2">Termination</td></tr>
<tr><td colspan="2">RXH + RX$^-$ + H
X = N, S, O, C</td><td colspan="2">RRRRX$^-$ + H</td><td colspan="2">RX + CS → RXCS
CS = Chain Stopper</td></tr>
<tr><td colspan="2">Peroxy Radical Formation
RX + O^2 → ROOH
ROOH → ROO + H</td><td colspan="2">C = C + ROO → ROOCC</td><td colspan="2">ROO$^-$ + AO + ROOAO
AO = Antioxidant</td></tr>
<tr><td colspan="2">Metal Catalyzed
RXH + M → RX + H
M = (Fe, Cu, Ni, Va, Cr, Ca, Mg)</td><td colspan="2">RRRRX + M + H</td><td colspan="2">RXH + M + RXH = Metal Coordinator</td></tr>
<tr><th colspan="6">Non-Free Radical Polymerization (Condensation)</th></tr>
<tr><td colspan="3">RX + RY →
RXRYRXRYRXRY + H_2O
RX and RY = ROH, RNH, RCOOH, C = C</td><td colspan="3">RXRY + Cl → RXRY Cl + H_2O
Cl = Condensation Inhibitor</td></tr>
</table>

3.2.3 Fouling Mechanisms

Two predominant fouling mechanisms, which can occur in hydrotreater units, are deposition and polymerization. Deposition occurs when a particulate becomes too large to remain entrained in a liquid or gas flowing stream. The particulate then deposits on the heat transfer surface of the heat exchanger or becomes trapped on the top of the catalyst bed. Both inorganic and organic particulates can deposit and cause fouling in hydrotreaters.

Deposition of particulate matter is a function of several different operating variables, which include the size of a particle, the bulk fluid density, and the velocity of the bulk fluid through the mechanical equipment. The smaller the particle size, the less the tendency to settle out of a flowing stream. The bulk fluid density becomes lighter as the material is heated, and as it becomes lighter, the potential for particles to settle out increases, depending upon the weight and size of the particle. If the fluid is completely vaporized in the heat exchanger train, it is possible for the larger particles to immediately settle out and foul the heat transfer surfaces at that point. This phenomenon will depend a great deal on the vapor velocities at the point of vaporization. As particulates agglomerate, or get larger, their potential to deposit also increases.

Particulates in the hydrotreater feed can be organic or inorganic. Inorganic particulates are materials such as iron sulfide and other corrosion products, catalyst particles, or inorganic salts, which have become entrained in the hydrotreater, feed stream. Organic particulates may be formed in tankage or in transport by various polymerization mechanisms. The polymer may also be formed in the preheat exchangers and grow to such a large size that it drops out of solution and deposits on the process equipment. There is three major polymerization mechanisms, which can occur in hydrotreater, feed streams: free radical, metal catalyzed, and non-free radical (condensation) (Table 3.2).

Free radical polymerization occurs when a free radical is formed and continues to react with other molecules. The free radicals continue to propagate in the feed stream producing longer chain polymers. These longer chain polymers will continue to be produced as long as free radicals are being formed. Free radical polymerization is easily initiated in the presence of light and heat, and its rate of polymer formation increases exponentially with temperature. A general rule is that for every 10°C (25°F) increase in temperature, the rate of polymer formation doubles. Formation of sediment or organic polymers in tankage is possible by free radical polymerization.

One type of free radical polymerization is that of oxygen initiated polymerization. In this case, oxygen reacts with a compound to form oxygen or peroxy radicals, which react to form polymer. The oxygen source is typically from air in non-blanketed tankage. Another source of oxygen may be oxygenated compounds in the feed stream, which become more reactive as the feed stream is heated.

There are numerous types of free radicals, which can be formed from different trace compounds found in a feed stream. These include the breaking of a double bond, or unsaturated bonds, to produce an alkyl radical, or other types of polymerization precursors such as nitrogen and sulfur radicals, which are easily formed at the temperatures found in the preheat exchanger train. The formation of free radicals has been investigated extensively, and it is known that some compounds form free radicals more readily than others. A special case of the free radical polymerization mechanism is metal catalyzed polymerization. There are numerous metals, which in very low concentrations, can act as a catalyst and initiate polymerization reactions. The metals may be in the form of metal salts or metal complexes. Metal salts may themselves contribute to deposit formation, or disassociate at higher temperatures to catalyze some polymerization reactions. The metal ion has the ability to catalyze reactions by making it easier to form free radicals at lower temperatures. Some of these catalytically reactive metals are: iron, copper, nickel, vanadium, chromium, calcium, and magnesium.

Another type of polymerization reaction is anion free radical mechanism in which the formation of a polymer is not through the formation of a free radical, but results from the reaction of two different molecules together under the right conditions.

One of the reactive components may be a radical, or compound, from a free radical-initiated polymerization step. The condensation polymerization reaction is an example of non-free radical polymerization, in that two large radicals, or compounds, react together to form an even larger compound, but in their reaction also generate a smaller compound, such as water. This new, larger compound can continue to react with other reactive species in the feed stream to make higher molecular weight polymers. At some point, the polymer will either:

- Get so large in size that it is no longer able to stay entrained or soluble in the fluid stream and deposit.
- All the different compounds that can react with it are consumed, and no further polymer is formed.

3.3 Basic knowledge of Heat Exchanger

Heat exchangers are devices that facilitate hate transfer between two fluids at different temperatures without allowing them to mix. There are called indirect contact exchangers. However, if the fluids do not tend to mix naturally, a direct contact heat exchanger may be used, e.g. a water chiller.

3.3.1 Overall Heat Transfer Coefficient

Heat is being transferred from the fluid inside, through dirt or fouling film, through the tube wall, through another fouling film to the outside fluid at a local bulk temperature. The general form of this coefficient is written

$$U = \frac{1}{A_t R_t} \tag{3.11}$$

where *A_t* is the total area available for heat transfer and *R_t* is the effective (overall) thermal resistance. Calculating U is clearly a matter of first determining *R_t*. For a simple shell and tube heat exchanger, thermal resistance is the sum of three basic resistive components in series:

- convection resistance on the tube side
- conduction resistance of the wall between the two flow streams
- convection resistance on the shell side

So the overall resistance as

$$R_t = \frac{1}{A_i h_i} + R_c + \frac{1}{A_o h_o} \tag{3.12}$$

where subscripts i and o designate inner and outer, respectively, and R_c is the conductive resistance for a cylindrical surface derived as

$$R_c = \frac{\ln(r_o / r_i)}{2\pi Lk} \tag{3.13}$$

Notice that can use either A_i or A_o as the total area available for heat transfer A_t in equation (3.2). It makes no difference which one is used, as long as the specified. Using $A_o = 2\pi r_o L$ as the basis, we obtain

$$U_o = \frac{1}{A_o R_t} = \frac{1}{A_o\left(\frac{1}{A_i h_i} + \frac{\ln(r_o / r_i)}{2\pi Lk} + \frac{1}{A_o h_o}\right)} \tag{3.14}$$

which, after simplifying yields

$$U_o = \frac{1}{\left(\frac{r_o}{r_i h_i} + \frac{r_o \ln(r_o / r_i)}{k} + \frac{1}{h_o}\right)} \tag{3.15}$$

Similarly, for $A_i = 2\pi r_i L$

$$U_i = \frac{1}{\left(\frac{1}{h_i} + \frac{r_i \ln(r_o / r_i)}{k} + \frac{r_i}{r_o h_o}\right)} \tag{3.16}$$

Because of normal operation, inner surface of a heat exchanger can become coated with deposits that leach out of the working fluids and corrode due to reaction with the fluid. These factors present additional resistance to heat transfer that can be modeled via a fouling factor equation (3.3) can be modified as

$$R_t = \frac{F_i}{A_i} + \frac{1}{A_i h_i} + R_c + \frac{1}{A_o h_o} + \frac{F_o}{A_o} \tag{3.17}$$

to consider fouling, where F_o and F_i are the fouling factors for outer and inner surfaces, respectively. Generally, performance is gradually degraded over time and costs are increased because of maintenance requirements and down time. The fouling factor is generally a known quantity based on the working fluid.

3.3.2 Fouling Factor

After period of the operation the heat transfer surfaces for a heat exchanger may become coated with various deposits present in the flow systems, or the surfaces may become corroded as a result of the interaction between the fluids and the material used for construction of the heat exchanger. In either even, this coating represents an additional resistance to the heat flow, or thus results in decreased performance. The overall effect is usually represented by a fouling factor, or fouling resistance, R_f, which must be included along with the other thermal resistances making up the overall heat transfer coefficient.

Fouling factor must be obtained experimentally by determining the values of U for both clean and dirty conditions in the heat exchanger. The fouling factor is thus defined as

$$R_f = \frac{1}{U_{dirt}} - \frac{1}{U_c} \tag{3.18}$$

3.3.3 The Log Mean Temperature Different

The physical flow of heat exchanger units is simply too complex to obtain analytical solution, e.g. because turbulence, developing flow, separation, etc. Moreover, diverse geometry and architecture preclude generalizing results into a few relevant correlations. Instead, we will take what may be thought of as an approximate integral approach in which the analysis is only dependent upon temperatures at the inlets and outlets and the overall convection coefficient.

The form of Newton's Law of Cooling

$$Q = UA\Delta T_{lm} \tag{3.19}$$

where A_t and U_m are the total area available for heat transfer and the overall convection coefficient. Respectively, and ΔT_{lm} is a temperature difference between the hot and cold fluids is not a constant.

Here, we will introduce the method of the "Log Mean Temperature Difference "(LMTD) for solving heat exchanger problems. This procedure allows us to calculate ΔT_{lm}, which can be thought of as an appropriately averaged temperature difference between the two flow streams. It is probably no surprise that ΔT_{lm} depends upon the heat exchanger configuration, flow arrangement, etc. There are a number of additional assumptions we must make to implement this method

- The unit is insulated such that no heat is exchanged with its surroundings; heat transfer only takes place between the hot and cold streams within the unit
- specific heats of both fluids are constant
- overall heat transfer coefficient is constant
- potential and kinetic energy changes can be neglected

So the form of LMTD for co current flow are defined as

$$\Delta T_{lm} = \frac{(T_{h,out} - T_{c,out}) - (T_{h,in} - T_{c,in})}{\ln[(T_{h,out} - T_{c,out})/(T_{h,in} - T_{c,in})]} \qquad (3.20)$$

the form of LMTD for counter current flow are defined as

$$\Delta T_{m} = \frac{(T_{h,in} - T_{c,out}) - (T_{h,out} - T_{c,in})}{\ln[(T_{h,in} - T_{c,out})/(T_{h,out} - T_{c,in})]} \qquad (3.21)$$

where the subscripts *h* and *c* designate the hot and cold fluids, respectively.

Stated verbally, it is the temperature difference at one end of the heat exchanger less the temperature difference at the other end of the exchanger divided by the natural logarithm of the ratio of these two temperature differences.

3.4 Artificial Neural Network

Artificial neural networks are mathematical structures, which built from the attempt to emulate the human brain or biological network. These networks involve with the learning process of interesting systems. After artificial neural networks have learned, the trained network can be used to perform certain tasks depending on the particular application. In addition, the artificial neural networks have the ability to learn from their environment and adapt it in an interactive manner similar to the biological counterparts.

3.4.1 Introduction of an Artificial Neural Network

Since the artificial neural network paradigm emerged from the attempt to emulate and understand the working of the human brain or biological neural network. In the nervous system, the brain is the central element of the nervous system, which is connected to receptors that shuttle sensory information to it, and delivers action commands to effectors. In addition, the brain is a huge and complicated neural network, which consists of about 1011 neurons. Each neuron consists of three main components: dendrites, cell body and axon (as shown in Figure 3.3). Dendrites, which are branchlike nerve fiber around the neural cell body, receive signals from other neurons by the receiving zones, called synapse. The cell body or soma sums the incoming signals, which are received from dendrites, and sends them to an axon. Axon which is a long fiber-like extension from cell body is the transmit channel of impulses to the other neurons.

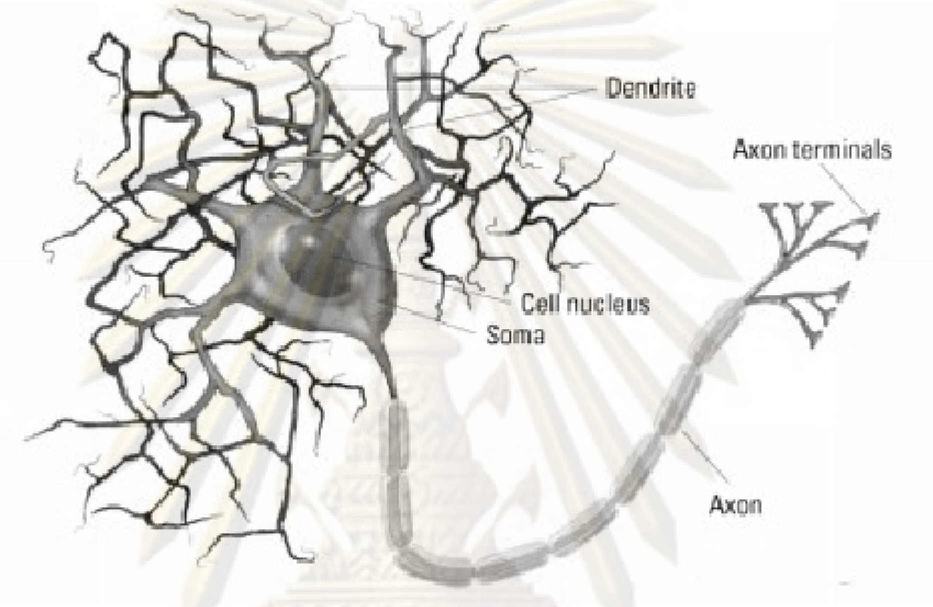

Figure 3.3 Components of biological neural network

The mentioned basic concept of biological neural network lead to research in the area of the mechanism and model of human brain including develop the model to solve complex problems in science and engineering. The first artificial neuron was created in 1943 by McCulloch and Pits. They proposed the model of a simple neuron, which seemed appropriate for modeling symbolic logic and its behavior. The McCulloch-Pitts neuron is a simple unit having a linear activation function with threshold value to produce an output. In 1959, Rosenblatt began work on the perceptron which consisted of neuron-like processing units with linear thresholds, and were arranged in layers similar to biological systems. The perceptron can learn and compute a weighted sum of the inputs, subtract a threshold, and pass one of two possible values out as the result. In addition, Widrow and Hoff developed the models, which are called MADALINE (Multiple Adaptive Linear). MADALINE was the first artificial neural network to be applied to a real world problem.

3.4.2 Components of an Artificial Neural Network

The artificial neural network consists of many interconnected artificial neurons or nodes. In each node, there are many components that are used to build an artificial neural network. These components are described as the follow.

3.4.2.1 Weighting Factors

Weighting factors are adaptive coefficients within the artificial neural network that determine the intensity of the input signal as registered by the artificial neuron. An artificial neuron usually receives many inputs for create the network. Each input has its own relative weight, which impacts the input on the summation function. Some inputs are made more important than others so that they have a greater effect on the processing element as they combine to produce a neural response. Thus, their weighting factors are greater than the others. These weighting factors can be modified in response to various training sets and according to a network's specific topology.

3.4.2.2 Summation Function or Basis Function

The first step in the operation is to compute the sum of all inputs. Mathematically, the inputs and the corresponding weights are vectors which can be represented as (x1, x2,..., xn) and (w1, w2,..., wn). The simplistic summation function is found by multiplying each component of the x vector by the corresponding component of the w vector and then adding up all the products. Moreover, the summation function can be more complex than the simplistic summation function. The inputs and the weighting factors can be combined in many different ways before passing to the transfer function.

The summation function can divide into two common forms:

- *Linear Basis Function (LBF)*

Linear basis function is a hyper plane-type function, which is a first order basis function. The net value is a linear combination of the inputs and the weighting factors, which is shown as the follow.

$$sum = \sum_{j=1}^{n} w_{ij} x_j \qquad (3.22)$$

- *Radial Basis Function (RBF)*

Radial basis function is a hyper sphere-type function, which involves with the second-order (nonlinear) basis function. The net value, which represents the distance to a reference pattern, is shown as the follow.

$$sum = \sqrt{\sum_{j=1}^{n} \left(x_i - w_{ij}\right)^2} \qquad (3.23)$$

3.4.2.3 Transfer Function or Activation Function

The results from the summation function, almost always the weighted sum, are transformed to a working output by the transfer function. In the transfer function, the total summation of the inputs and the weighting factors can be compared with some threshold to determine the neural network output. If the summation is greater than the threshold value, the processing element generates a signal. If the summation is less than the threshold, no signal is generated from the transfer function. The transfer function is generally non-linear function. Linear functions are not useful because the linear transfer functions are limited such as the output is simply proportional to the input. For example, the most common transfer functions, which are the Step function, Ramp function, Linear function, Log-Sigmoid function, Tangent-Sigmoid function, Gaussian function and Arc tangent function are shown in Table 3.3.

Table 3.3 Transfer functions of an artificial neural network

Function	Equation	Characteristic
Step function	$f(x) = \begin{cases} 1 & if \quad x > 0 \\ 0 & otherwise \end{cases}$	Linear
Ramp function	$f(x) = \begin{cases} 1 & if \ x \geq 1 \\ x & if \ x < 1 \\ -1 & if \ x < -1 \end{cases}$	Linear
Linear function	$f(x) = x$	Linear
Log-Sigmoid function	$f(x) = \dfrac{1}{1 + e^{-x}}$	Non-linear
Tangent-Sigmoid function	$f(x) = \dfrac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$	Non-linear
Gaussian function	$f(x) = ce^{-x^2}$	Non-linear
Arc tangent function	$f(x) = \arctan\left(\dfrac{x}{3.1416}\right) + 0.5$	Non-linear

3.4.2.4 Scaling

In the neural networks training, the networks can be made more efficient if scaling processing steps are carried out on the input pattern and target. For example, the back propagation algorithm is used to train a feed-forward perceptron, if a sigmoid function is used as a non-linear activation function, the saturation limit are 0 and 1. If the training patterns have large values compared to these limits, the non-linear activation functions could be operating almost exclusively in a saturated mode and not allow the network to train. Therefore, the training data should be range-scale to avoid this problem.

3.4.2.5 Output Function

Each processing element or neuron allows one output signal, which it may be a output to hundreds of inputs from other neurons. This is just like the biological neurons, which there are many inputs and only one output action. Normally, the output is directly equivalent to the transfer function's result.

3.4.2.6 Error Function

In the training of supervised networks, the training procedure requires a measure of the difference between the neural network output values and the target (desired output) values. The difference between the target and output values is so called the error. This error is transformed by the error function to match particular network architecture. For example, the most common error functions are sum square error, mean square errors and mean absolute error.

These error functions are described as the follow.

- *Sum Square Error*

$$SSE = \sum_{j=1}^{N} \left(y_i - p_i \right)^2 \qquad (3.24)$$

- *Mean Square Error*

$$MSE = \frac{1}{N} \sum_{j=1}^{N} \left(y_i - p_i \right)^2 \qquad (3.25)$$

- *Mean Absolute Error*

$$MAE = \frac{1}{N} \sum_{j=1}^{N} \left| y_i - p_i \right| \qquad (3.26)$$

which y_i is the network output

p_i is the network target

3.4.2.7 Learning Function

The purpose of the learning function is to modify the variable connection weights on the inputs of each processing element according to some neural based algorithm to achieve a desired result. There are two types of learning algorithm; supervised and unsupervised learning. Supervised learning requires a teacher, which may be a training set of data or an observer who grades the performance of the network results. For unsupervised learning, the system must organize itself by some internal criteria designed into the network.

3.4.3 Architecture of an Artificial Neural Network

3.4.3.1 Network Structures

Artificial neural network structure can be divided into common types such as feed-forward networks and feedback networks.

- *Feed-Forward Networks*

Feed-forward networks allow the signals travel from input to output one way only. There is no feedback in the network such as the output of any layer does not affect in the same layer. Feed-forward networks tend to be straight forward networks that associate inputs with outputs.

- *Feedback Networks*

Feedback networks, which allow the signals travel in both directions of the network, are very powerful networks. These networks are dynamic which their states change continuously until they reach an equilibrium point. They remain at the equilibrium point until the inputs change and a new equilibrium need to be found.

3.4.3.2 Connection Structures

An artificial neural network comprises the neuron and weight building blocks. The behavior of the network depends on the interaction between these building blocks. There are four common types of connections such as feed-forward, feedback, lateral and time-delayed connections.

- *Feed-Forward Connections*

For all the neural network models, the data from neurons of a lower layer are propagated forward to neurons of an upper layer via feed-forward connection networks.

- *Feedback Connections*

For all the neural network models, the feedback connections bring the data from neurons of an upper layer back to neurons of a lower layer.

- *Lateral Connections*

For all the neural network models, the lateral connections allow the neurons to interact in the same layer.

- *Time-Delayed Connections*

Delay elements may be incorporated into the connections to yield temporal dynamics models. They are more suitable for temporal pattern recognitions.

3.4.3.3 Network Layers

The layers of an artificial neural network are divided into three types such as input layer, hidden layer and output layer. The input layer represents the raw information that is fed into the network. The hidden layer is between the input and output layer. The output layer is the last layer of the networks that depends on the activity of the hidden layers and the weights between the hidden and output layers.

3.4.4 Learning Algorithm of an Artificial Neural Network

Training or Learning means the modifying values of the weighting factor in the interconnections to achieve some target criteria for the output layer. Information is stored and distributed throughout the network via the interconnection weights. Many of learning algorithms are proposed and divided into two types, which are shown in Table 3.4.

Table 3.4 Learning algorithms of an artificial neural network

Learning algorithm	
Supervised learning	Unsupervised learning
Perceptron	Additive Grossberg (AG)
Adaline	Adaptive Resonance Theory (ART)
Backpropagation	Continuous Hopfield (CH)
Bolzman Machine (BM)	Learning Matrix (LM)
Associate Reward Penalty (ARP)	Learning Vector Quantizer (LVQ)

3.4.4.1 Supervised Learning

In the supervised learning, training process consists of the input and output data. This data is often referred to as the training set. During the training, the actual output of an artificial neural network is compared to the desired output. Weighting factors, which are usually randomly set to begin, are then adjusted by the network. Thus, the network will produce a closer match between the desired and the actual output in the next iteration. The learning algorithm tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until acceptable network accuracy is reached.

3.4.4.2 Unsupervised Learning

Unsupervised learning, which is sometimes called self-supervised learning, is limited to networks known as self-organizing maps. These kinds of networks are not in widespread use. These networks use no external influences to adjust their weights. Instead, they internally monitor their performance. These networks look for regularities or trends in the input signals, and makes adaptations according to the function of the network. Even without being told whether it's right or wrong, the network still must have some information about how to organize itself. This information is built into the network topology and learning rules.

3.4.5 Multilayer Feed-Forward Neural Network

Multilayer feed-forward neural network is a one of the most popular artificial neural network architectures which is widely used in the function approximation or modeling any arbitrary system. This type of network is also sometimes called the multilayer perceptron because of its similarity to perceptron networks with more than one layer. Multilayer feed-forward neural network consists of an input layer, one or more internal layers and an output layer. The internal layers are called hidden layers because they only receive internal inputs (inputs from other processing units) and produce internal outputs (outputs to other processing units). The structure of multilayer feed-forward neural network is shown in Figure 3.4.

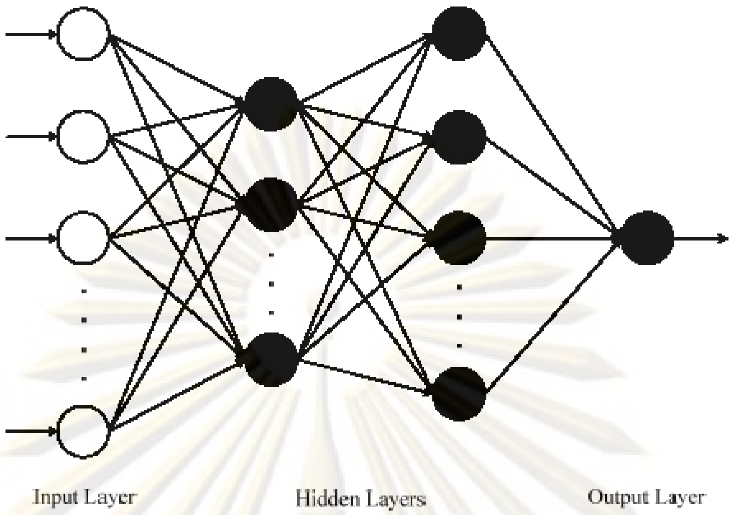

Figure 3.4 The structure of multilayer feed-forward neural network

3.4.6 Design of Artificial Neural Network

3.4.6.1 Structure and Size of Network

No standard procedure has been known to determine the structure and the number of neurons or nodes in the network for any particular application. However, the general procedure for selecting the hidden nodes is to fix an initial size and then check the error tolerance of this structure. If this error satisfies, the training process is stopped. If not, the size and the structure are revised and the whole procedure repeats until it satisfies the tolerance.

3.4.6.2 Data Collection

In utilizing of an artificial neural network, the data set collection is normally split into various sets. One is the training set which is used to train the network weights and normally span the operating region of the model. Later is the testing data set, which is used for final validation of the trained network. The selection of inputs data, which is fed into the networks, is an important consideration for any particular application. For steady state application, the selection of inputs to the networks basically depends on the relevant variables likely to have an effect on the predicted output variable. For modeling the dynamic behavior of a system, it would not only depend on these relevant variables but also the time history of these variables as well as the time history of the output variables. The knowledge of the system such as the model order is use as the initial guide to decide on the time history.

3.4.6.3 Data Processing

After the data collection, all data should be pre-processed using statistical procedure. Data in the training sets are pre-processed to have zero mean and unit variance. This is necessary to prevent input with large average values in certain dimension.

3.4.6.4 Weight Initialization

The initial weight specification has an effect on the speed and quality of neural network training. The small random number is normally used to initialize the weights of the network so that each connection responds slightly differ during training. If the final prediction does not satisfy the error tolerance during training, the weights are also re-initialized and the identification process is repeated.

3.4.6.5 Training the Network

Training is a procedure to determine the optimal values of the connection weights and bias weights. Training begins by initially assigning arbitrary small random values to the weights. Training proceeds iteratively until a satisfactory model is obtained. In each of iteration, called an epoch, the actual outputs corresponding to all the sets of inputs in the training set are predicted and the weights are adjusted in the direction of the output prediction error is decreased. The weights are incrementally adjusted for every pattern in every of iteration and they gradually converge on the optimal values. Different network architectures require the different training or learning algorithms. The training times can be significantly reduced by the use of suitable training algorithms. However, back propagation algorithm remains the mainstay of performing neural network learning.

3.4.6.6 Model Validation

Over-learning, which occurs when the network starts to learn the presented pattern in a point-wise fashion instead of learning the functionality, is a potential problem that can easily occur in process identification. During over-learning, the performance of the network training continues to improve on the learning data set but starts to degrade on the testing set. However, it can be dealt with proper training and validation.

3.4.6.7 Basic Steps of an Artificial Neural Network Design

There are many procedures of the artificial neural network design but the basic steps of the artificial neural design are summarized as the follow.

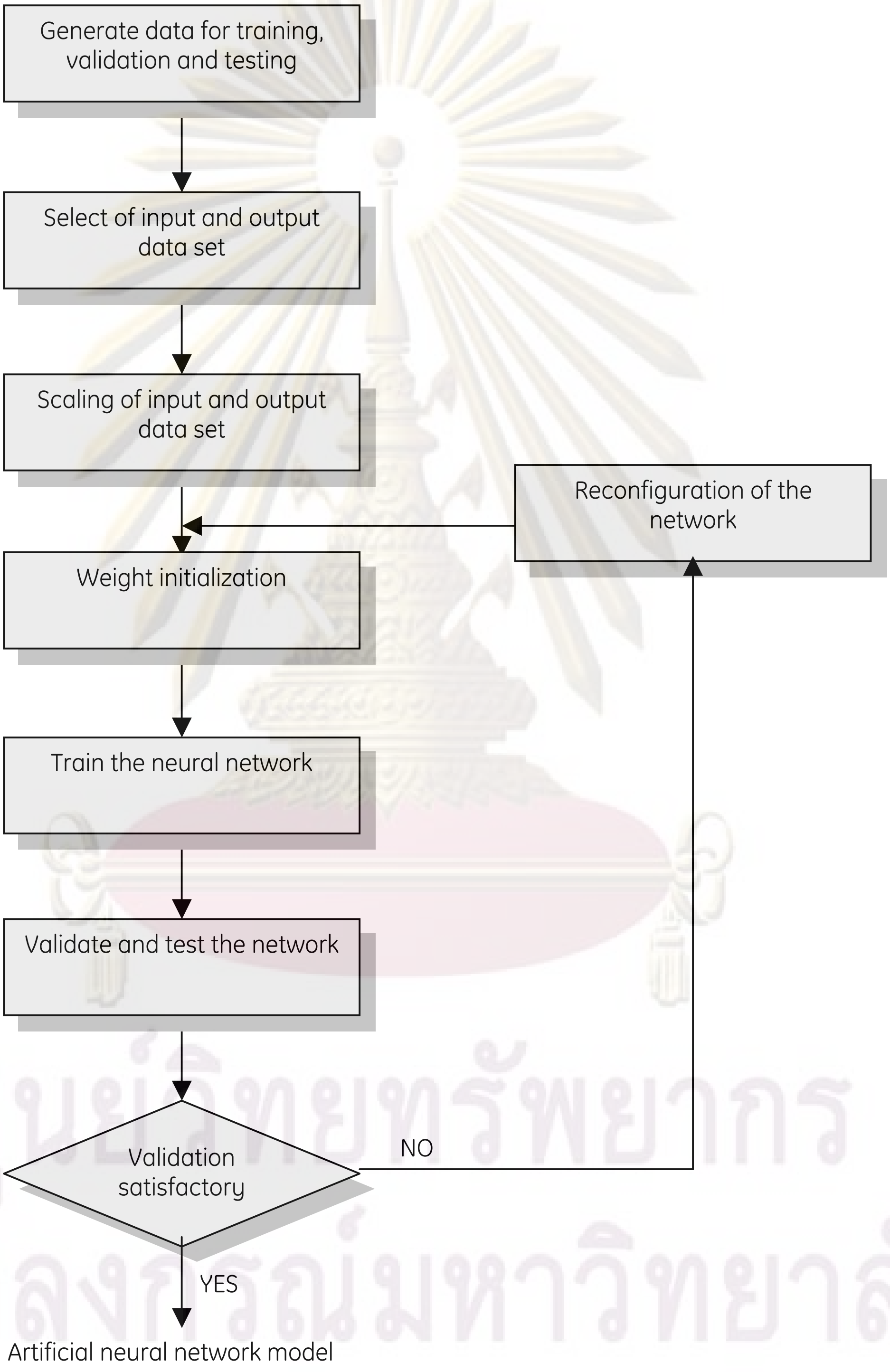
Generate data for training, validation and testing
Select of input and output data set
Scaling of input and output data set
Reconfiguration of the network
Weight initialization
Train the neural network
Validate and test the network
Validation satisfactory
NO
YES
Artificial neural network model

Figure 3.5 Basic steps of an artificial neural network design

3.4.7 Application of an Artificial Neural Network

The artificial neural network is widely applied in modeling of the unknown nonlinear systems. The main advantage of the use of artificial neural network is obtaining a highly accurate mathematical model of the system without the detail of the system. The process modeling applications use the artificial neural network to approximate the relationship between the input and output variables. During the process modeling, a number of candidate models are considered and only one model, which is expected to the best prediction of the process outputs with the given process inputs, is selected. The selected model is the one that is expected to have the least prediction error in the future. In addition, artificial neural network is also the universal function approximator that typically works better than the traditional function approximation method for the application of any arbitrary system.

CHAPTER IV

SYSTEM MODELING APPLICATION

In this chapter, the use of neuron network approach to model the overall heat transfer coefficient of feed/effluent exchanger, which has inherently complex and non-linear behavior, is presented. This chapter is divided into two sections; the first section is the actual process used the mathematical model, and the other section is the use of neuron network approach to represent the actual process.

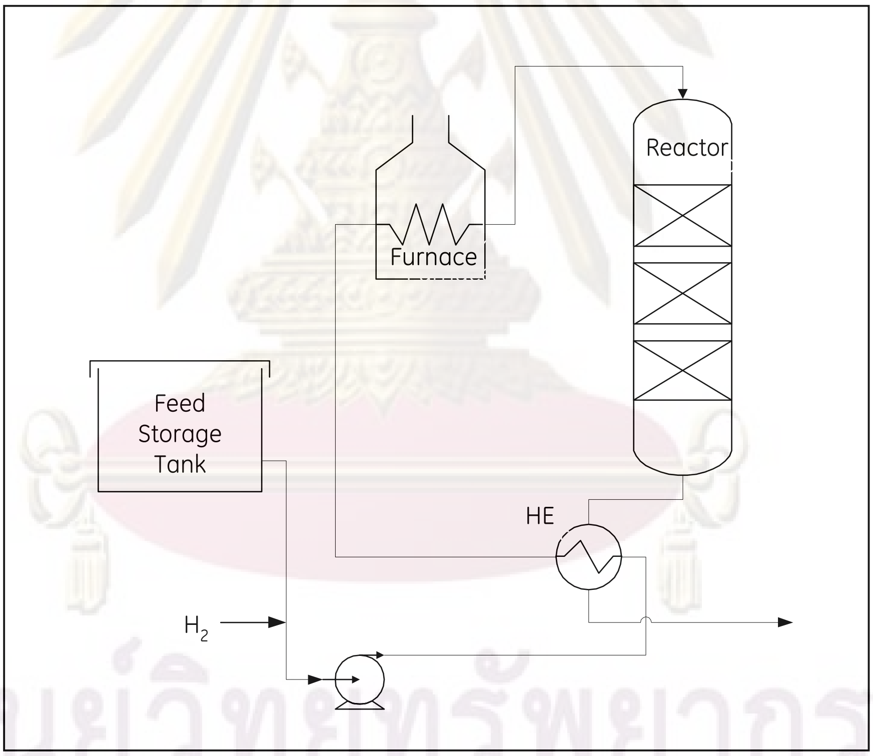

Figure 4.1 Process flow diagram for feed/effluent exchanger

4.1 Description of the process

Hydrotreating unit typically has two main functions. The first is to catalytically stabilize refined products by adding hydrogen to unsaturated compounds. Unsaturated or olefinic compounds are very reactive and can cause undesired polymerization products in downstream process. The second function is to remove impurities such as sulfur, oxygen, nitrogen and halide compounds from process stream, which could be detrimental to downstream processes.

From figure 4.1, the feed stream is mixed with hydrogen before heated in a set of feed/effluent exchanger prior sent to a furnace. The furnace then heats the feed stream to the required reactor temperature. Hydrogen gas, which is essential to the reaction, is relative inertly until attaining reactor temperature in the presence of a catalyst. The reaction consists of a packed bed filled with a catalyst over which the exothermic hydrotreating reaction takes place. After the reaction has taken place, the reactor product is cooled by transferring heat to the feed in the feed/effluent exchangers. The excess hydrogen, and reaction by-products such as ammonia, hydrogen sulfide are then separated from the saturated product stream. The metal contaminants in the feed are trapped on the catalyst surface.

4.2 Mathematical fouling model

A large number of models for heat exchanger fouling have been proposed. However, they are not able to predict the fouling formation by changing of the operating conditions and differing feed types. Some models can only able to predict fouling without considering the effect of fluid velocity on the fouling removal.

4.2.1 Polley fouling model

Polley et al. improved the threshold model of Ebert and Panchal. The fouling rate varies with the type of fluid. Tube wall temperature has a strong effect on fouling formation and sedimentation. Polley et al. proposed the fouling formation model that required the tube wall temperature as follows:

$$R_f(t) = \alpha \operatorname{Re}^{-0.8} \operatorname{Pr}^{-0.33} \exp\left(\frac{-E}{RT_w}\right) - \gamma \operatorname{Re}^{0.8} \tag{4.1}$$

Where α = 277.8 (m^{2}K/J), γ = 4.167 x 10^5 (m^{2}K/J), E = 48 (KJ/mol) and R = gas constant

4.2.2 Mohammad fouling model

Mohammad proposed a new model for crude oil fouling in preheat exchangers of crude distillation units. The experiment results of Australian light crude oil with the tube side surface temperature and the fluid velocity were used. A new model including a term of fouling formation and a term of fouling removal due to chemical and tube wall shear stress was proposed as follows:

$$R_f(t) = \alpha \mathrm{Re}^{\beta} \exp\left(\frac{-E}{RT_f}\right) - \gamma \mathrm{Re}^{0.4} \qquad (4.2)$$

where α = 10.98 x 10^{-3} (m^2K/J), β = -1.547, *E* = 22.618 (kJ/mol), γ = 9.60 x 10^{-14} (m^2K/J)

4.2.3 Evaluation of the proposed model

For comparing the proposed models, the constants of model should be calculated based on the actual process. The film temperature (T_f) and the tube wall temperature (T_w) are assumed close by inlet temperatures of the cold stream ($T_{c,in}$). Table 4.1 and 4.2 shows these constants. Table 4.3 shows the error in different models in comparison with actual process. In this table, summation of square of errors (equation 4.3) in Polley and Mohammad are calculated.

$$\sum_{j=1}^{n} \left(R_{f,actual}(j) - R_{f,\mathrm{mod}el}(j) \right)^2 \qquad (4.3)$$

Table 4.1 Polley fouling model constants comparison with actual process

Model	Actual process
E (J/mol)	47999.94
α (m^2K/J)	278.57
γ (m^2K/J)	5.024 x 10^{-4}

Table 4.2 Mohammad fouling model constants comparison with actual process

Model	Actual process
E (J/mol)	22618
α (m^2K/J)	0.01098
β	-1.547
γ (m^2K/J)	5.794×10^{-6}

Table 4.3 Summation of square errors between proposed model and actual process

Model	Polley	Mohammad
Summation of squares of errors	1.016×10^{-6}	5.875×10^{-11}

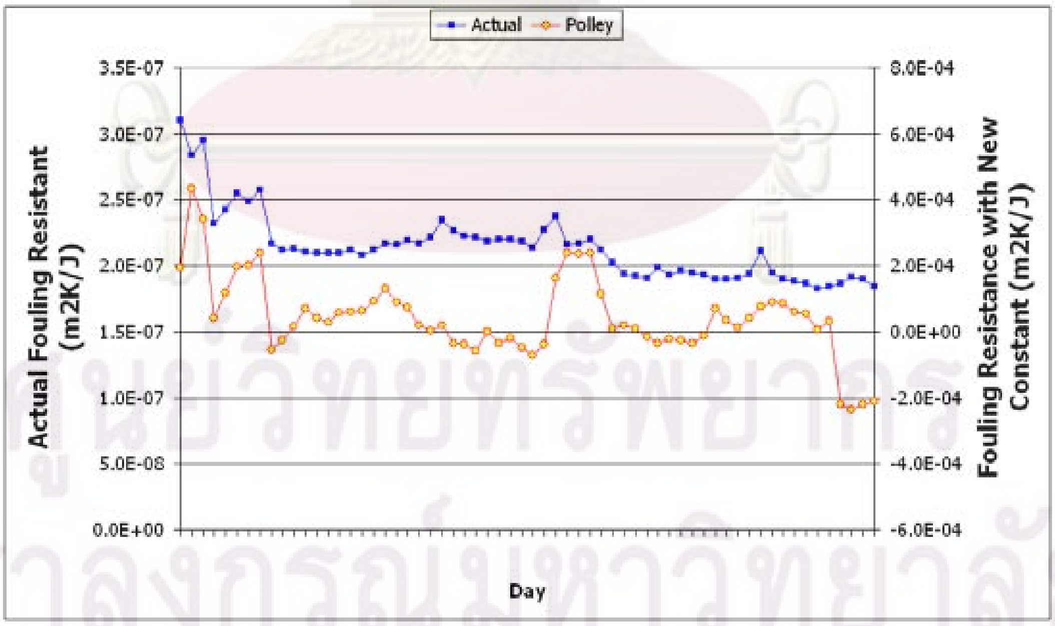

Figure 4.2 Comparison of fouling rate of the Polley model and actual process

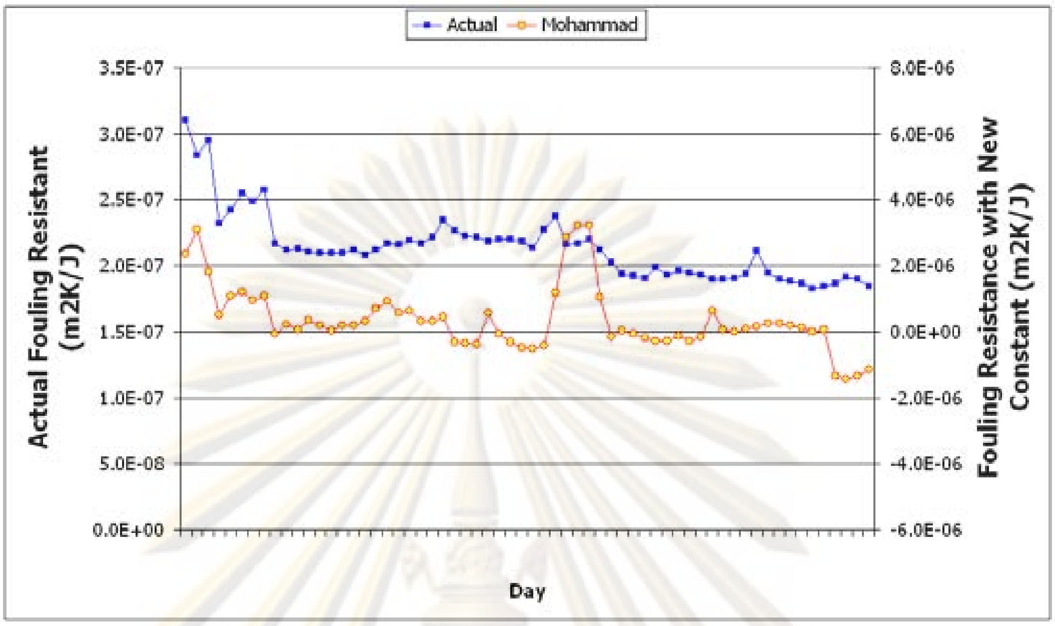

Figure 4.3 Comparison of fouling rate of the Mohammad model and actual process

As shown in table 4.1 and 4.2 the constant, γ of both proposed model are much different from the actual process. And the deviation of the Mohammad model from the actual process results is lower than Polley model. Figure 4.1 and 4.2 compares the fouling rate, which are calculated from the proposed model based on feed/effluent exchanger fouling data. It is revealing that there are many deviations among the proposed models to predict the fouling formation of actual process. One consideration is the fouling formation and removal of fouling layer has a great importance. To use the model and also to calculate the activation energy the fouling rate should be available. Moreover, it should be mentioned that to extend the proposed models for the other type of crude oil the constant value have to be recalculated correspondingly.

4.3 Preliminary study of single heat exchanger

In this case, we considered at a counter current shell and tube heat exchanger. The configuration of heat exchanger is shown in Figure 4.3.

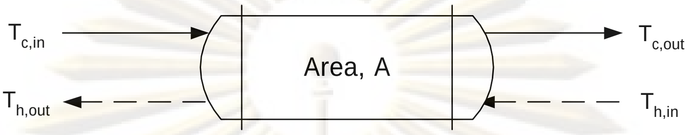

Figure 4.3 The diagram of heat exchanger

4.3.1 Heat exchanger model description

The formulation presented here is based on the assumptions below.

- The input mass flow rates are conducted when plant operating constant load. We thus consider here the situation where the heat exchanger is operating under maximum throughput.

- Constant physical parameters. The variation in fluid heat capacity with temperature is neglected.

- Film heat transfer coefficients are assumed to remain constant despite changes in Prandtl and Reynold numbers caused by different temperatures and fouling.

- No energy losses.

4.3.2 Hot and cold streams outlet temperature

The feed/effluent exchanger is formulated as counter current flow. Assuming no energy loss, an energy balance for the cold and hot streams of a heat exchanger gives

$$Q_c = m_c C_{p,c}(T_{c,out} - T_{c,in}) \tag{4.4}$$

$$Q_h = m_h C_{p,h}(T_{h,in} - T_{h,out}) \tag{4.5}$$

Where Q_c, Q_h are a heat transfer rate for cold and hot stream, KJ/hr, m is a mass flow rate, kg/hr and $C_{p,c}$, $C_{p,h}$ are a specific heat for cold and hot stream, KJ/(kg K)

The heat transfer rate in a heat exchanger is explained as

$$Q = UA\Delta T_{lm} \quad (4.6)$$

Where U is an overall heat transfer coefficient (KJ/hr m^2 °C), A is a heat transfer surface area (m^2) and ΔT_{lm} is logarithmic mean temperature difference (LMTD, °C)

The log mean temperature difference, ΔT_{lm} is given by

$$\Delta T_{lm} = \frac{(T_{h,in} - T_{c,out}) - (T_{h,out} - T_{c,in})}{\ln\left[\frac{(T_{h,in} - T_{c,out})}{(T_{h,out} - T_{c,in})}\right]} \quad (4.7)$$

Using the above equations, the cold and hot stream outlet temperature can be computed from

$$T_{c,out} = M_h T_{h,in} + M_c T_{c,in} \quad (4.8)$$

$$T_{h,out} = T_{h,in} - \frac{1}{k_1}(T_{c,out} - T_{c,in}) \quad (4.9)$$

Where k_1 and k_2 are

$$k_1 = \frac{m_h C_{p,h}}{m_c C_{p,c}} \quad (4.10)$$

$$k_2 = \frac{UA}{m_h C_{p,h}} \quad (4.11)$$

By defining M_c and M_h as

$$M_c = \frac{(1-k_1)\exp(-k_2(k_1-1))}{\exp(-k_2(k_1-1))-k_1} \tag{4.12}$$

$$M_h = \frac{k_1(\exp(-k_2(k_1-1))-1)}{\exp(-k_2(k_1-1))-k_1} \tag{4.13}$$

4.3.3 Overall heat transfer coefficient

The overall heat transfer coefficient, *U* (KJ/hr m^2 °C), for heat exchanger was found using as

$$U \ = \frac{m_c C_{p,c}(T_{c,out} - T_{c,in})}{A\Delta T_{lm}} \tag{4.14}$$

From equations (4.14), the inlet and outlet temperatures of the cold stream and hot stream ($T_{c,in}$, $T_{c,out}$, $T_{h,in}$ and $T_{h,out}$) , the mass flow rate of hot stream and cold stream (m_c and m_h), the heat transfer area (*A*), the overall heat transfer coefficient (U_c) and the specific heat capacity of cold stream and hot stream ($C_{p,c}$ and $C_{p,h}$) for the feed/effluent exchanger are known. The values of k_1, k_2, M_1, and M_2 were calculated. Thus, the value of outlet temperature of cold stream and hot stream, the overall heat transfer coefficient and the heat transfer rate are obtained.

4.4 Neural network estimator

In chemical industrial processes, some process variables cannot be directly measured or are difficult to measure. Due to this fact, a state estimator is proposed to cope with this problem. Artificial neural network is found to be the one of various tools that can be used to estimate the unmeasured process variables since it is considered as an universal approximator that can approximate any arbitrary function.

4.4.1 Neural network training

In the current work, a multilayer feed-forward neural network was applied as a state estimator to determine the overall heat transfer coefficient of feed/effluent exchanger in hydrotreating unit. For the design of neural network, the data sets for the network traingin are devided into three different sets; training sets, validation sets and testing sets. Generally, the data for modeling the system must cover the entire system in order to represent accurately the system behavior. For hydrotreating unit, the process variables, which consist of the Reynolds (*Re*) and Prandtl (*Pr*) number and overall heat transfer coefficient (*U*), are considered as the input data. In the neural network architecture, a multilayer feed forward network, which consists of an input layer, three hidden layers, and an output layer, is employed for modeling the process. The network architecture is shown in Figure 4.4.

For the data preparation, all data is normalized which it has mean of zero and standard deviation of one for achieving a good performance of neural network model. Levenberg–Marquardt Backpropagation algorithm (as shown in APPENDIX A) with the early stopping mechanism is used to train this multilayer feed-forward neural network. The Mean Square Error (MSE) is used as the criteria for the network selection and also for the stopping weights and biases adjustment.

Table 4.4 The transfer functions in each layer of the neural network estimator

Layer	Variables	Transfer Function	Equation
Layer #2	f_j^2	Log-Sigmoid	$f_j^2(x) = \dfrac{1}{1+e^{-x}}$
Layer #3	f_j^3	Log-Sigmoid	$f_j^3(x) = \dfrac{1}{1+e^{-x}}$
Layer #4	f_j^4	Linear	$f_j^4(x) = x$

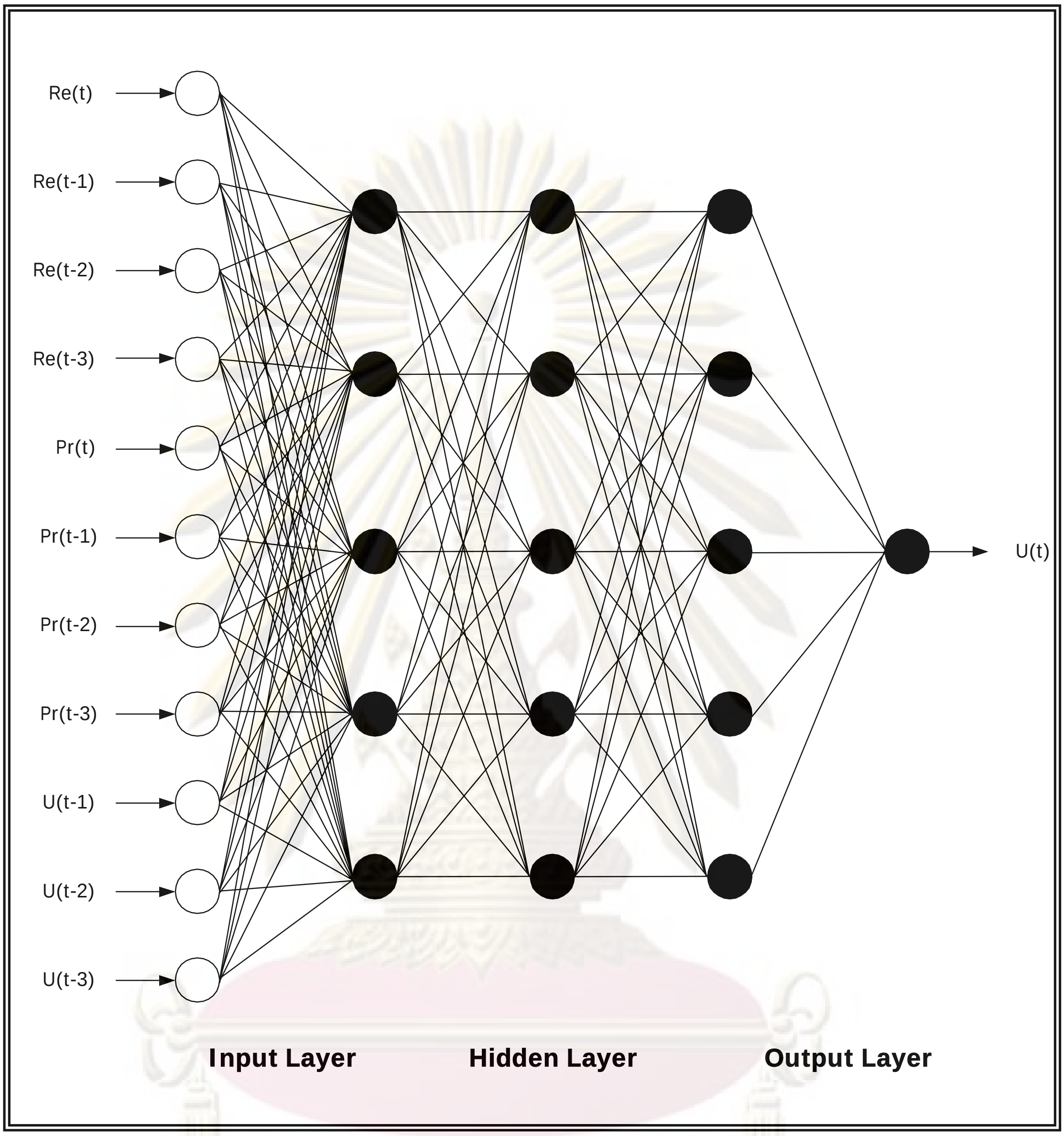

Figure 4.4 Neural network configurations for modeling

From figure 4.4, the input layer is composed of eleven nodes which are the Reynolds number (*Re*), the Prandtl number (*Pr*), and overall heat transfer coefficient in feed/effluent exchanger at time *t, t-1, t-2* and *t-3*. The three hidden layer are composed of ten, fifty, one hundred and five hundred hidden nodes with the log-sigmoid transfer function and one output node with a linear transfer function in the output layer, which is the estimating value of the overall heat transfer coefficient in feed/effluent at time *t*.

4.4.2 Design of a neural network estimator

As mention in the previous section, the appropriate configuration of neural network model can be determined by varying the hidden layer size, the desired error and the maximum epochs. The hidden layer size is varied from ten nodes to five hundred nodes, the desired error is varied from 1x10^{-3} to 1x10^{-5} and the maximum epochs are varied from 10000 to 100000.

The result of determination of network modeling are shown only variation of hidden nodes size as follows:

Table 4.5 The neuron network model with ten hidden nodes

Parameter	Value
Hidden Layers	3
Hidden Nodes	10
Desired Error	1x10^{-5}
Maximum Epochs	50000

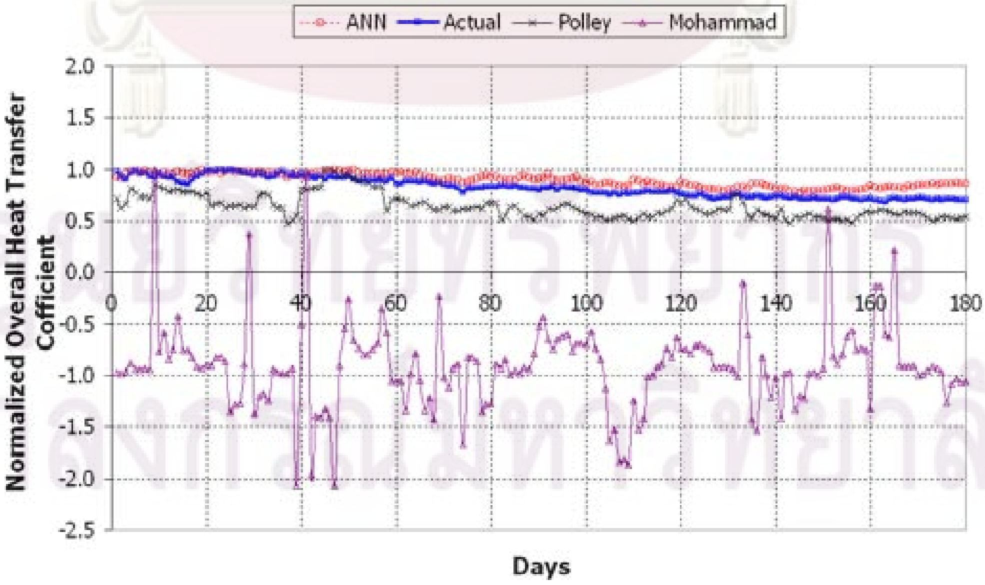

Figure 4.5 The validation performance with ten hidden nodes

Table 4.6 The neuron network model with fifty hidden nodes

Parameter	Value
Hidden Layers	3
Hidden Nodes	50
Desired Error	1x10^{-5}
Maximum Epochs	50000

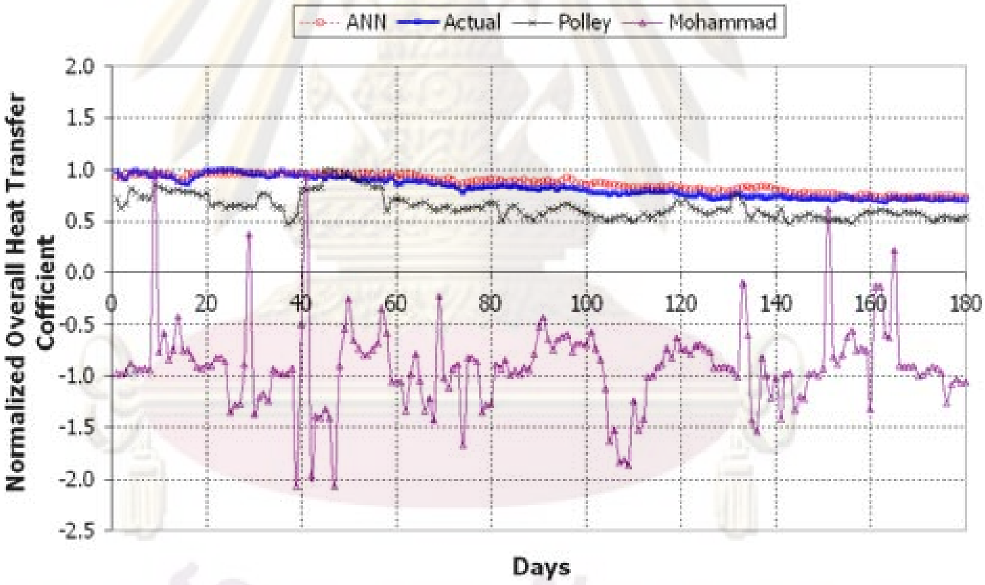

Figure 4.6 The validation performance with fifty hidden nodes

Table 4.7 The neuron network model with one hundred hidden nodes

Parameter	Value
Hidden Layers	3
Hidden Nodes	100
Desired Error	$1x10^{-5}$
Maximum Epochs	50000

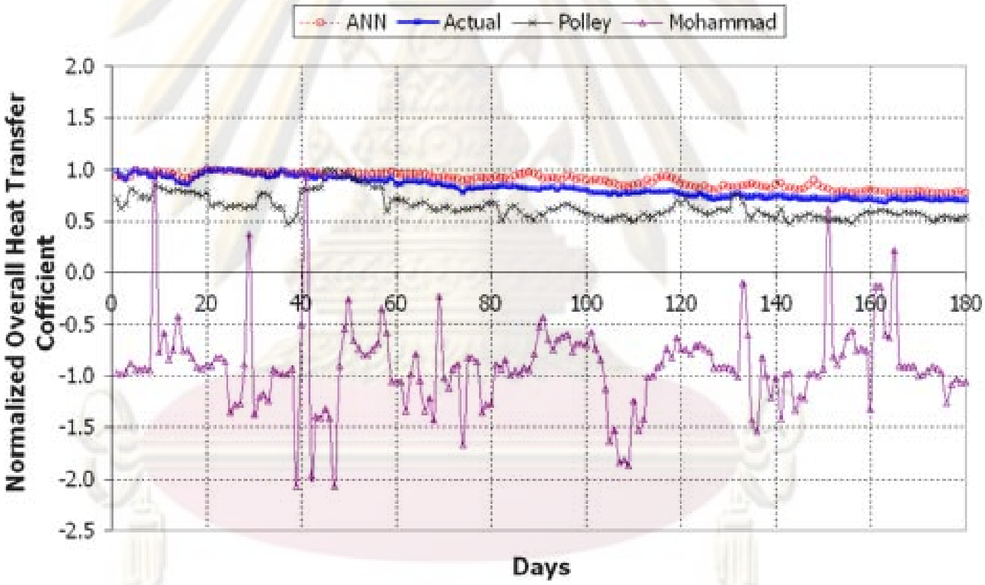

Figure 4.7 The validation performance with one hundred hidden nodes

Table 4.8 The neuron network model with five hundred hidden nodes

Parameter	Value
Hidden Layers	3
Hidden Nodes	500
Desired Error	$1x10^{-5}$
Maximum Epochs	50000

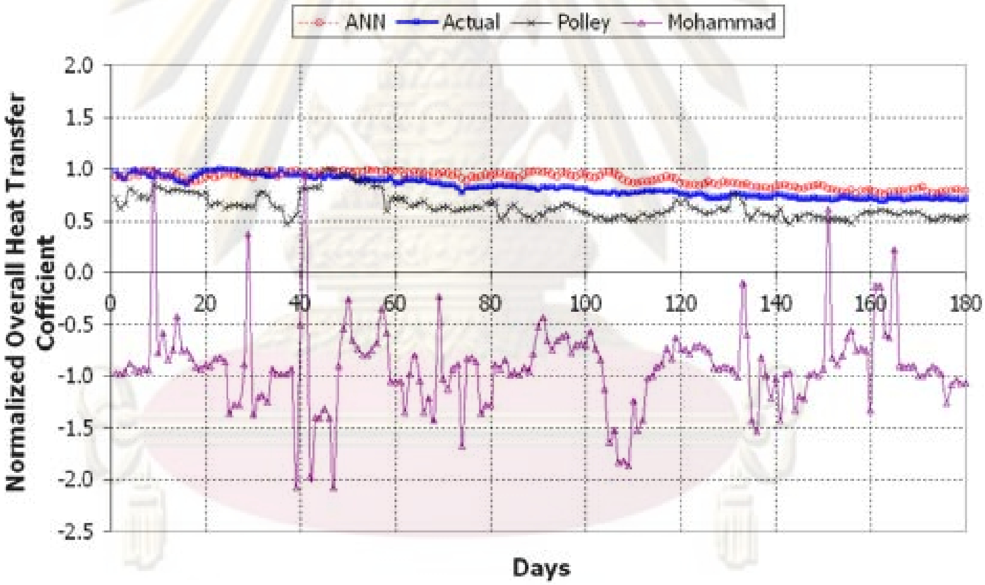

Figure 4.8 The validation performance with five hundred hidden nodes

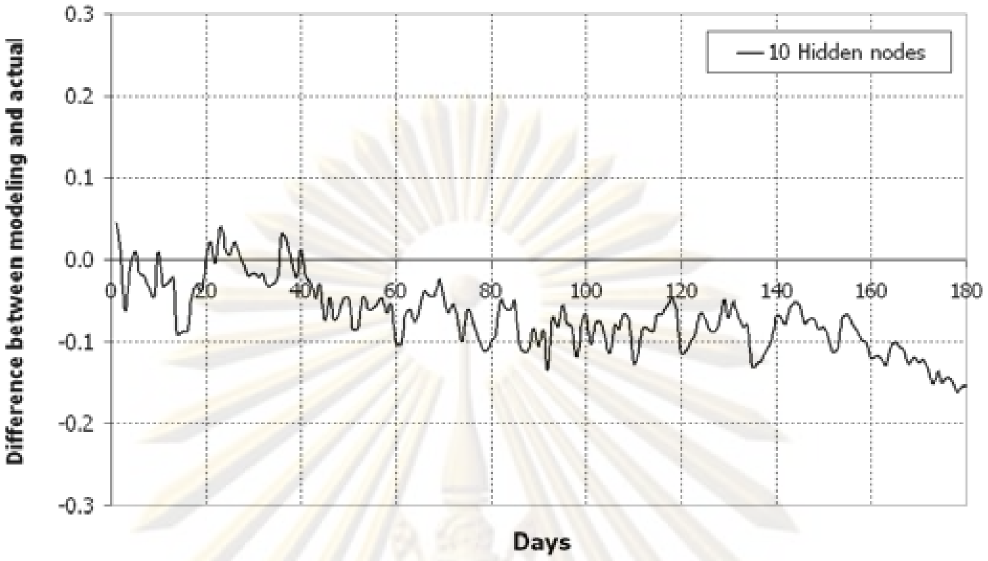

Figure 4.9 The difference value with ten hidden nodes

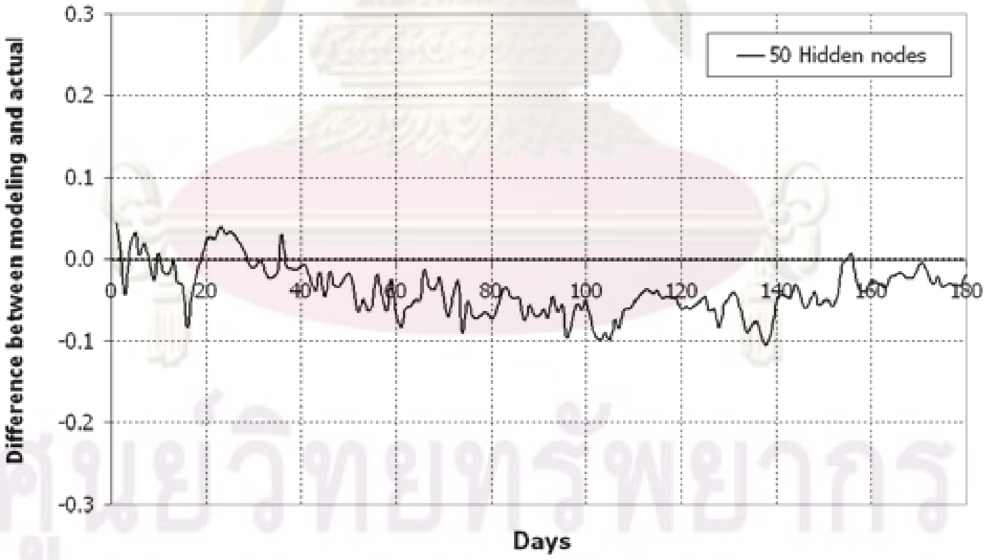

Figure 4.10 The difference value with fifty hidden nodes

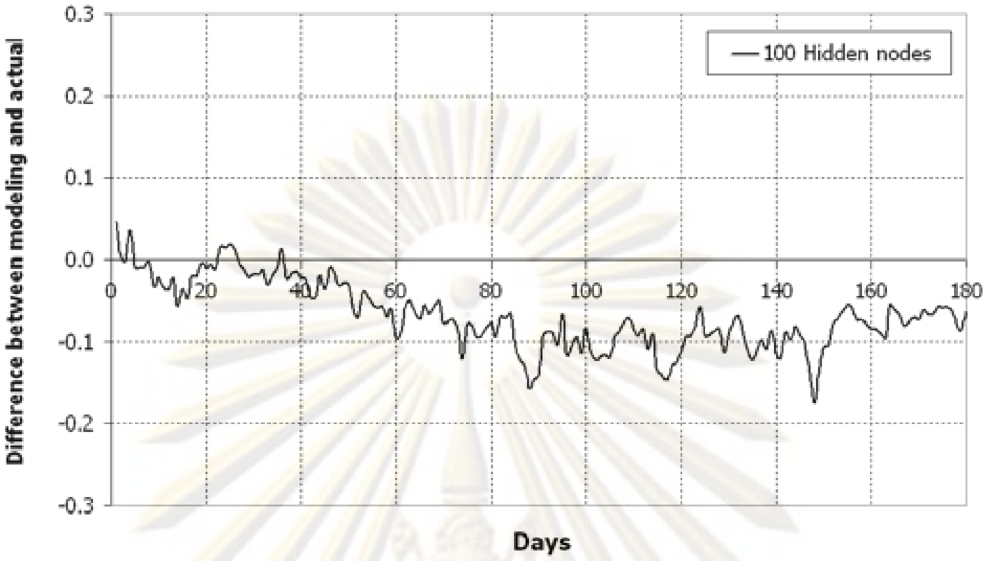

Figure 4.11 The difference value with one hundred hidden nodes

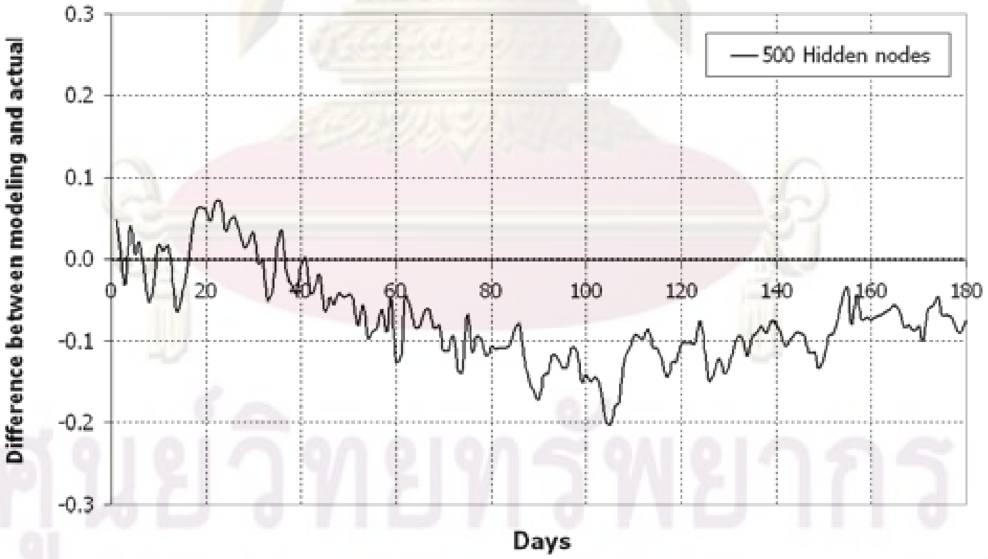

Figure 4.12 The difference value with five hundred hidden nodes

Table 4.9 The mean square error for the various architectures of the neural network

Number of hidden node	Mean square error (MSE)
10	0.00658
50	0.00241
100	0.00629
500	0.00849

4.4.3 Results and Discussions

The Mean Square Error for the various architectures of a neural network is presented in table 4.9. The architecture, which gives the minimum value of the Mean Square Error is considered for application as a neural network estimator. The optimum architecture of a neural network for the overall heat transfer coefficient estimation is shown in figure 4.6. It is indicated that network configuration quite fitting with actual data and after day-100, the validation set is remained closely to actual data. The other neural network estimators are illustrated in the figure 4.5-4.8. From figure 4.5 indicated that the network begins over fitting after day-40 and the validation set is further increased when the number of date increased. From figure 4.7 and 4.8 showed that the network configurations are similar and after day-40 the network begin over fitting.

From the results mentioned above, it can be concluded that the network with too few hidden nodes is incapable of complex process representation. On the other hand, if the network has too many hidden nodes, it will be poor generalization for untrained data.

Therefore, the fifty hidden nodes of network are selected as the appropriate network configuration employed to represent the overall heat transfer coefficient of feed/effluent exchanger in hydrotreating unit.

CHAPTER V

OPTIMIZATION OF CLEANING SCHEDULE FOR FEED / EFFLUENT EXCHANGER

This chapter presents the implementation of optimal cleaning schedule of feed/effluent exchanger by using neuron network estimator. The detail of the modeling of the overall heat transfer coefficient of exchanger can be seen in Chapter IV. The variation of inlet and outlet temperature of hot and cold streams is important. These variations have a significant effect on production rate and operating cost. This studied is divided into three cases: the first one focuses on the variation of number of cleaning heat exchanger. The second one focuses on the variation of the period of operating time whereas the third one deal with the cost of furnace extra fuel as shown in figure 5.1. The objective function for this problem is the minimization of total operating cost in a fixed operation time.

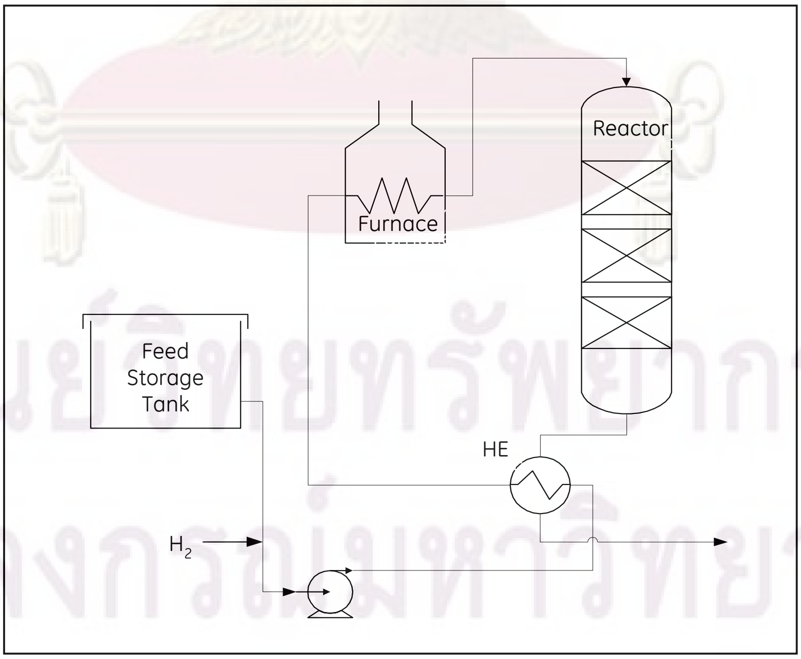

Figure 5.1 Process flow diagram for feed/effluent exchanger

The objective function of finding the optimum heat exchanger cleaning schedule is to be minimizing the operating costs in specific duration. Therefore, the objective function was computed for various time intervals between consecutive cleaning. The model refers to the tradeoff between furnace extra fuel costs due to fouling and heat exchanger cleaning costs.

$$Obj_{\cos t} = \sum_{p=0}^{NP-2} C_E(Q_{n,clean} - Q_{n,p}) + \sum_{NP-2}^{NP-1} C_E(Q_{n,clean} - Q_{n,p}) + \sum_{NP-1}^{NP} C_E(Q_{n,clean} - Q_{n,p}) + \sum_{p=1}^{NP}(NP * C_{cl}) \tag{5.1}$$

where

NP the number of cleaning period

C_E the cost of furnace extra fuel (Baht/(KJ/h))

$Q_{n,clean}$ the heat transfer rate between hot and cold fluids in unit *n* in clean condition (KJ/h)

$Q_{n,p}$ the heat transfer rate between hot and cold fluids in unit *n* at period *p* with fouling (KJ/h)

C_{cl} the cost of heat exchanger cleaning (Baht/unit)

The optimal cleaning schedule is obtained by minimizing equation (5.1) (minimizing the operation costs)

Applying the constraint in order to reduces the solution space as well as the amount of computations for finding the optimal cleaning schedule by minimizing equation (5.1), the constraint on the operating cost as follow, The outlet cold temperature of each heat exchanger must be greater than the inlet cold one and the outlet hot temperature of each heat exchanger must be lower than the inlet hot one:

Cold temperature:

$$T_{c,out} - T_{c,in} \geq 0 \tag{5.2}$$

Hot temperature:

$$T_{h,in} - T_{h,out} \geq 0 \tag{5.3}$$

There are also restrictions in minimum values that certain temperatures can reach:

$$T_{c,out} - T_{c,out}^{\min} \geq 0 \tag{5.4}$$

5.1 Optimization of cleaning schedule for single heat exchanger

Fouling decreases the overall heat transfer coefficient and heat transfer rate. This is resulting in the reduction of heat exchanger performance. Thus, the effect of fouling behavior on heat exchanger is important. In this section, we consider the cleaning schedule of single heat exchanger in a fixed operation time. The single heat exchanger is counter current flow for shell and tube heat exchanger. The input mass flow rate is fixed, and any variation in fluid heat capacity with temperature is neglected. This studied is divided into three cases: the first one focuses on the variation of number of cleaning period. The second one focuses on the variation of the period of operation time whereas the third one deals with the extra cost of furnace fuel.

5.1.1 The influence of number of cleaning period on the optimal cleaning schedule

The simulation of the variation of number of cleaning period on the optimal cleaning schedule and operating cost is presented in this section. It can be divided into two cases, first case focus on the 2 years operation time with the number of cleaning heat exchanger is 2, 3 and 4 times, respectively. The second case deals with 3 years operation with the same number of cleaning heat exchanger.

5.1.1.1 The influence of number of cleaning period on the optimal cleaning schedule with 2 years of operation time

The parameters for optimization of 2 years operation time are shown in table 5.1 and the optimal cleaning schedule is shown in table 5.2.

Table 5.1 Model parameter for single heat exchanger case

Parameter	**Value**
$T_{c,in}$ (^oC)	136.0
$T_{h,in}$ (^oC)	360.0
$T_{reactor,in}$ (^oC)	332.0
F_h (Kg/h)	88,040
F_c (Kg/h)	88,040
$C_{p,h}$ (KJ/Kg ^oC)	4.067
$C_{p,c}$ (KJ/Kg ^oC)	3.081
U_o (KJ/h m^2 ^oC)	2391.45
C_E (Baht/(MKJ/h)	340.0
Operation Time (Year)	2

Table 5.2 The comparison of number of cleaning and operating cost within 2 years of operation time

Month	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	No. of Cleaning	Cost (Baht)
								/											/						2	10,671,935
							/							/					/						3	7,447,957

The results of the influence of number of cleaning period on optimal cleaning schedule and operating cost are presented in figure 5.2.

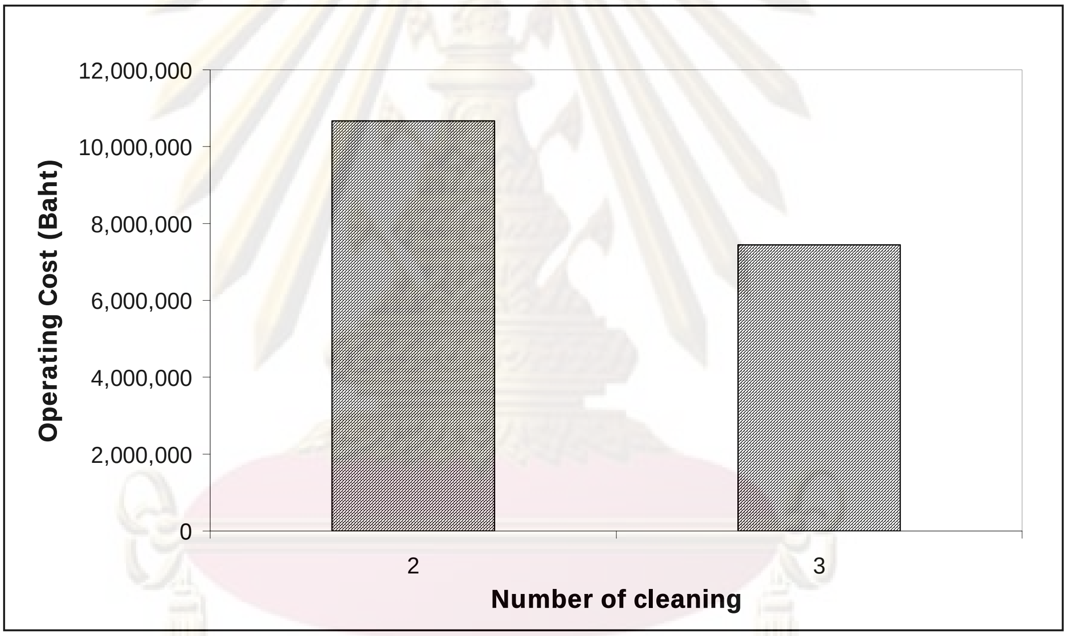

Figure 5.2 The comparison of number of cleaning and operating cost within 2 years of operation time

With 2 years of operation, it is observed that 2 number of cleaning period has 10,671,935 Baht for operating cost whereas more number of cleaning period has 7,447957 Baht, which is less operating cost as it can be seen from table 5.2 and figure 5.2. This is due to fouling occurred in heat exchanger will reduce overall heat transfer coefficient resulted in more energy consumed at furnace to maintain temperature inlet of reactor. Therefore, more number of cleaning periods will help reduction of operating cost.

5.1.1.2 The influence of number of cleaning period on the optimal cleaning schedule with 3 years of operation time

The parameters for optimization of 3 years operation time are shown in table 5.3 and the optimal cleaning schedule is shown in table 5.4.

Table 5.3 Model parameter for single heat exchanger case

Parameter	**Value**
$T_{c,in}$ (^{o}C)	136.0
$T_{h,in}$ (^{o}C)	360.0
$T_{reactor,in}$ (^{o}C)	332.0
F_h (Kg/h)	88,040
F_c (Kg/h)	88,040
$C_{p,h}$ (KJ/Kg ^{o}C)	4.067
$C_{p,c}$ (KJ/Kg ^{o}C)	3.081
U_o (KJ/h m^{2} ^{o}C)	2391.45
C_E (Baht/(MKJ/h)	340.0
Operation Time (Year)	3

Table 5.4 The comparison of number of cleaning and operating cost within 3 years of operation time

Month	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	No. of Cleaning	Cost (Baht)
									/										/																		2	16,334,616
									/									/							/												3	11,223,077

The results of the influence of number of cleaning period on optimal cleaning schedule and operating cost are presented in figure 5.3.

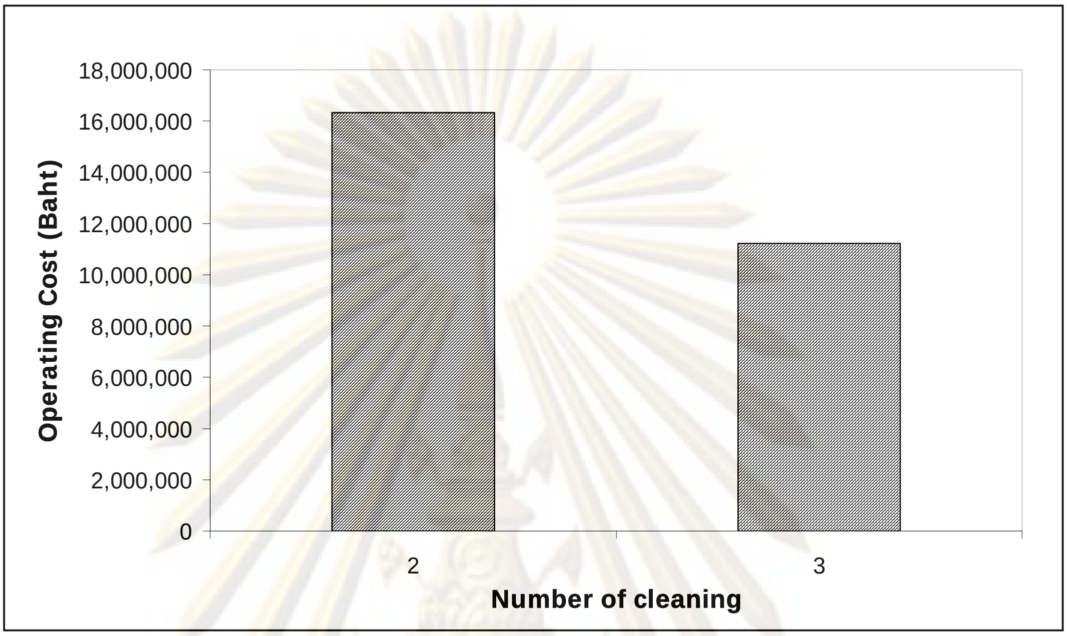

Figure 5.3 The comparison of number of cleaning and operating cost within 3 years of operation time

From table 5.4, the operating cost decreased when the number of cleaning period is increased since more cleaning actions help to increase overall heat transfer coefficient. This result is the main factor in objective function. Thus, the operating cost is reduced when the number of cleaning period is increased for both 2 years and 3 years of operation.

5.1.2 The influence of cost of furnace extra fuel on the optimal cleaning schedule

Nowadays, the major concerned for operating cost is furnace extra fuel cost and this cost tend to be increased every year. This section deals with cost of furnace extra fuel when it increase by 20% with 2 number of cleaning period. It can be also divided into two cases, first one deal with 2 years of operation time and the second one focus on 3 years of operation time with furnace extra fuel is 340 and 410 Baht/(MKJ/h), respectively.

5.1.2.1 The influence of cost of furnace extra fuel on the optimal cleaning schedule with 2 years of operation time

The parameters for optimization of 2 years operation time are shown in table 5.5 and the optimal cleaning schedule is shown in table 5.6.

Table 5.5 Model parameter for single heat exchanger case

Parameter	Value
$T_{c,in}$ (^{o}C)	136.0
$T_{h,in}$ (^{o}C)	360.0
$T_{reactor,in}$ (^{o}C)	332.0
F_h (Kg/h)	88,040
F_c (Kg/h)	88,040
$C_{p,h}$ (KJ/Kg ^{o}C)	4.067
$C_{p,c}$ (KJ/Kg ^{o}C)	3.081
U_o (KJ/h m^2 ^{o}C)	2391.45
C_E (Baht/(MKJ/h)	410.0
Operation Time (Year)	2

Table 5.6 Optimal cleaning schedule with different cost of furnace extra fuel within 2 years of operation time

Month	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Cost Fuel (Baht/ MKJ/h)	Cost (Baht)
								/											/						340	10,671,935
										/							/								410	12,786,745

The results of the different cost of furnace extra fuels on optimal cleaning schedule and operating cost are presented in figure 5.4.

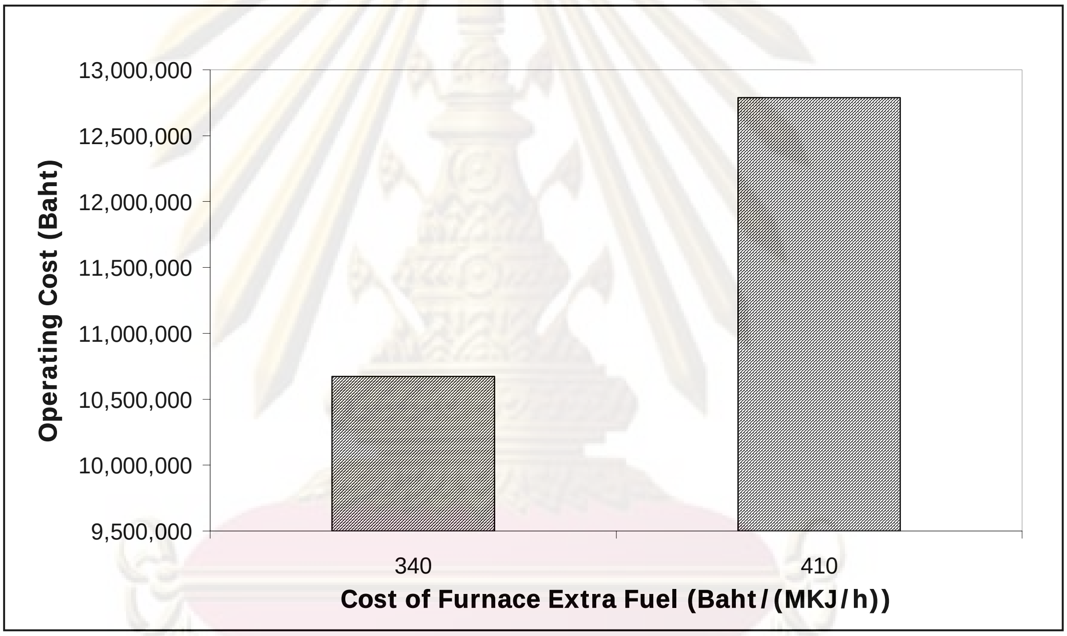

Figure 5.4 The comparison of operating cost in the different cost of furnace extra fuel within 2 years of operation time

From table 5.6 and figure 5.4, it is obviously seen that the rapid change of operating cost is occurred in case of large amount of cost of furnace extra fuel. The cost of furnace extra fuel is 340 Baht/(MKJ/h) with 10,671,935 Baht for the operating cost. Whereas, in the case of 410 Baht/(MKJ/h), the operating cost increase to 12,786,745 Baht.

5.1.2.2 The influence of cost of furnace extra fuel on the optimal cleaning schedule with 3 years of operation time

The parameters for optimization of 2 years operation time are shown in table 5.7 and the optimal cleaning schedule is shown in table 5.8.

Table 5.7 Model parameter for single heat exchanger case

Parameter	Value
$T_{c,in}$ (^{o}C)	136.0
$T_{h,in}$ (^{o}C)	360.0
$T_{reactor,in}$ (^{o}C)	332.0
F_{h} (Kg/h)	88,040
F_{c} (Kg/h)	88,040
$C_{p,h}$ (KJ/Kg ^{o}C)	4.067
$C_{p,c}$ (KJ/Kg ^{o}C)	3.081
U_{o} (KJ/h m^{2} ^{o}C)	2391.45
C_{E} (Baht/(MKJ/h)	410.0
Operation Time (Year)	3

Table 5.8 Optimal cleaning schedule with different cost of furnace extra fuel within 3 years of operation time

Month	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
									/										/			
												/									/	

Month	23	24	25	26	27	28	29	30	31	32	33	34	35	36	Cost Fuel (Baht/ MKJ/h)	Cost (Baht)
															340	16,334,616
															410	19,615,272

The results of the different cost of furnace extra fuels on optimal cleaning schedule and operating cost are presented in figure 5.5.

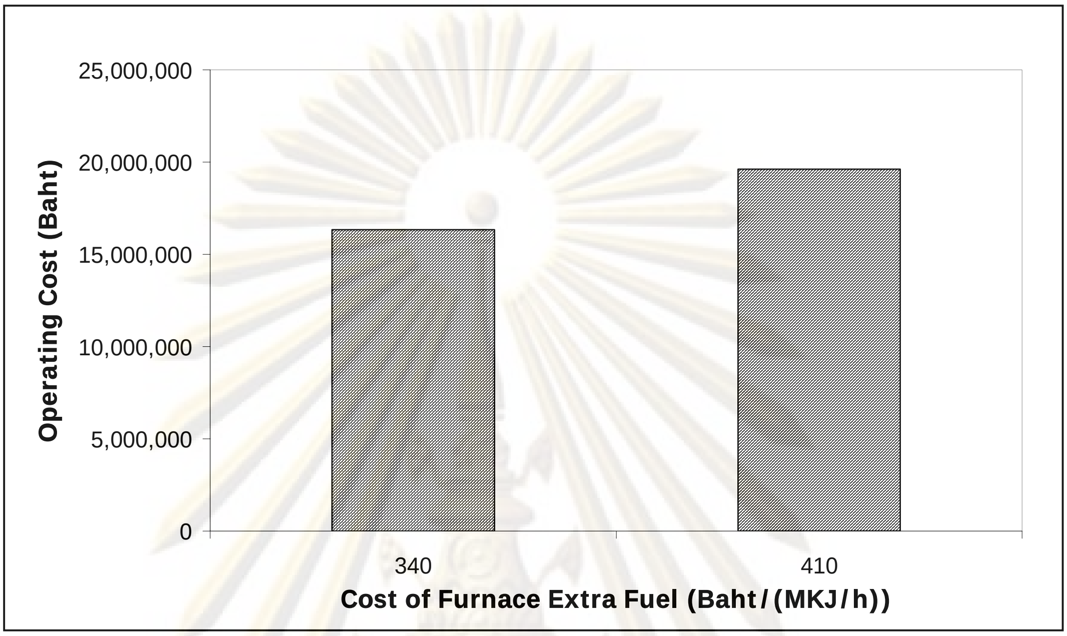

Figure 5.5 The comparison of operating cost in the different cost of furnace extra fuel within 3 years of operation time

It is observed that the major impact on operating cost is increasing of cost of furnace extra fuel as it can be seen from table 5.8 and figure 5.5. When cost of furnace extra fuel increase from 340 Baht/(MKJ/h) to 410 Baht/(MKJ/h), the operating cost is also increased from 16,334,616 Baht to 19,615,272 Baht.

5.2 Summary results for optimal cleaning schedule

Since fouling decreases the overall heat transfer coefficient and heat transfer rate resulted in the reduction of heat exchanger performance. The heat exchange needs to be shut down for cleaning and in this period of time the productivity lost actually increases. While cleaning is advantageous to recover heat transfer rate of exchanger, but doing too often may not be economically advisable. Therefore, the total lost due to fouling of heat exchanger is the combination of the operating cost and productivity lost as it indicated through equation (5.5)

$$TotalLost = OperatingCost + \Pr oductivityLost \qquad (5.5)$$

The equation (5.6) represents the cost saving of total lost in comparison of with and without plan of cleaning schedule of heat exchanger.

$$Saving = \frac{TotalLost(Unplanned) - TotalLost(Planned)}{TotalLost(Unplanned)} \times 100 \qquad (5.6)$$

The results of cost saving are shown in table 5.9 and 5.10 for 340 Baht/(MKJ/h) of furnace extra fuel and in table 5.11 and 5.12 present 410 Baht/(MKJ/h) of furnace extra fuel.

The productivity lost represents a major factor in total lost when heat exchanger has 3 number of cleaning period as shown in table 5.9-5.12.

For 2 number of cleaning period and 340 Baht/(MKJ/h) of furnace extra fuel, the % saving compared between unplanned cleaning and optimal cleaning schedule of this studied is 13.60% for 2 years and 16.31% for 3 years of operation as it can be seen from table 5.9 and 5.10. For the case of 410 Baht/(MKJ/h) of cost of furnace extra fuel (table 5.11 and 5.12), the % saving is 15.34% for 2 years and 17.98% for 3 years of operation. It is noted that the % saving of 3 years of operation is higher than 2 years of operation when cost of furnace extra fuel is 340 Baht/(MKJ/h). Similarly, 3 years of operation has more number of % saving than 2 years of operation if cost of furnace extra fuel increase to 410 Baht/(MKJ/h).

Table 5.9 Summary result for 2 years optimal cleaning schedule with 340 Baht/(MKJ/h)

	Operating Cost (Baht)	Productivity Lost (Baht)	Total Lost (Baht)	% Saving
Unplanned with 2 Cleaning Period	16,130,267	24,000,000	40,130,267	-
2 Cleaning Period	10,671,935	24,000,000	34,671,935	13.60%
3 Cleaning Period	7,447,957	36,000,000	43,447,957	-8.27%

Table 5.10 Summary result for 3 years optimal cleaning schedule with 340 Baht/(MKJ/h)

	Operating Cost (Baht)	Productivity Lost (Baht)	Total Lost (Baht)	% Saving
Unplanned with 2 Cleaning Period	24,195,400	24,000,000	48,195,400	-
2 Cleaning Period	16,334,616	24,000,000	40,334,616	16.31%
3 Cleaning Period	11,223,077	36,000,000	47,223,077	2.02%

Table 5.11 Summary result for 2 years optimal cleaning schedule with 410 Baht/(MKJ/h)

	Operating Cost (Baht)	Productivity Lost (Baht)	Total Lost (Baht)	% Saving
Unplanned with 2 Cleaning Period	19,451,204	24,000,000	43,451,204	-
2 Cleaning Period	12,786,745	24,000,000	36,786,745	15.34%
3 Cleaning Period	9,155,887	36,000,000	45,155,887	-3.92%

Table 5.12 Summary result for 3 years optimal cleaning schedule with 410 Baht/(MKJ/h)

	Operating Cost (Baht)	Productivity Lost (Baht)	Total Lost (Baht)	% Saving
Unplanned with 2 Cleaning Period	29,176,806	24,000,000	53,176,806	-
2 Cleaning Period	19,615,272	24,000,000	43,615,272	17.98%
3 Cleaning Period	13,043,071	36,000,000	49,043,071	7.77%

CHAPTER VI

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

The optimal cleaning schedules of feed/effluent exchanger in Hydrotreating unit subject to fouling are studied in this work. The aim of this work is to minimize the operating cost of heat exchanger. An artificial neural network has been approached to model the overall heat transfer coefficient of feed/effluent exchanger. It can be divided into two sections; the first section is the actual process used the mathematical model, and the other section is the use of neuron network approach to represent the actual process.

A large number of models of heat exchanger fouling have been proposed in the past. However, they are not able to predict the fouling formation by changing of the operating conditions and differing feed type. To comparing the proposed models, the constants of model should be recalculated based on the actual process. It is obviously seen that there are many deviations among the proposed model to predict the fouling formation of actual process. Therefore, an artificial neural network is applied to estimate the overall heat transfer coefficient, which is assumed to be an unmeasured variable of this process. A multilayer feed-forward network is trained by Lavenberg-Marquardt Backpropagation algorithm. The appropriation an artificial neural network, three number of hidden layers and fifty number of hidden nodes, is employed as a neural network estimator.

Next, the implementation of optimal cleaning schedule of feed/effluent exchanger using neuron network estimator is performed. The optimal solution solved by differential evolution method. This studied is divided into three cases: the first one focuses on the variation of number of cleaning heat exchanger. The second one focuses on the variation of the period of operating time whereas the third one deal with the cost of furnace extra fuel. The objective function is to minimize of total operating cost in a fixed operation time. The influences of number of cleaning period on optimal cleaning schedule are studied. The results shown that 2 number of cleaning period has operating cost more than 3 number of cleaning period. In addition, the influence of cost of furnace extra fuel on the optimal cleaning schedule is presented in this chapter. It is revealed that the operating cost increased when cost of furnace extra fuel increased. These parameters are sensitive to the optimal cleaning schedule. Moreover, the influences of total lost on the optimal cleaning schedule are studied. The results obtained that the total lost of planning of cleaning schedule is less than unplanned case.

6.2 Recommendation

For the future direction, the variations of flow rate for cold stream and hot stream and throughput losses due to fouling formation should be studied. Because of the disturbance on flow rate that the problem should be updated the flow rate at real time. Thus, the real time optimization is should be considered. In order to achieve the accurate planning, the objective function and constrain should be considered for the production plan and economic loss of the process.

REFERENCES

Bernard C.Grace. Fouling. Controlling Hydrotreating Fouling. Technical Paper of General Electric Company.

Casado, E., Model optimizes exchanger cleaning. Hydrocarbon Process 8 (1990): 71-78.

G.T. Polley, D.I. Wilson, B.L. Yeap, S.J. Pugh. Evaluation of laboratory crude oil threshold fouling data for application to refinery pre-heat trains. Applied Thermal Engineering 22 (2002): 777–788.

Hagan M., Demuth H. and Beale M. Neural Network Design. Boston: PWS, 1996.

James H. Gary, Glenn E. Handwerk. Catalytic Cracking. Petroleum Refining Technology and Economics. 4th edition. New York: Marcel Dekker (2001): 93-185.

Javier H. Lavaja and Miguel J. Bagajewicz. On a New MILP Model for the Planning of Heat Exchanger Network Cleaning. Ind. Eng. Chem. Res. 43 (2004): 3924-3938.

M. Kuosa, J. Kaikko, L. Koskelainen. The impact of heat exchanger fouling on the optimum operation and maintenance of the Stirling engine. Applied Thermal Engineering. (2006): 1-6.

Mariusz Markowski, Krzysztof Urbaniec. Optimal cleaning schedule for heat exchangers in a heat exchanger network. Applied Thermal Engineering 25 (2005): 1019–1032.

Micheal C. Georgiadis, Lazaros G. Papageorgiou. Optimal scheduling of heat-integrated multipurpose plants under fouling conditions. Applied Thermal Engineering 21 (2001): 1675-1697.

Mohammad Reza Jafari Nasr, Mehdi Majidi Givi. Modeling of crude oil fouling in preheat exchangers of refinery distillation units. Applied Thermal Engineering 26 (2006): 1572–1577.

APPENDICES

APPENDIX A

LEVENBERG-MARQUARDT BACKPROPAGATION ALGORITHM

A.1 Backpropagation Learning Algorithm

Backpropagation is the most widely used learning algorithm in an artificial neural network. In this algorithm, the error between neural network predicted output and the actual target is propagated backward from the output layer to the hidden layers and finally to the input layer. The weights and biases are changed in the direction of minimizing the prediction error.

For the multilayer feed-forward neural networks, the output of the first layer becomes the input of the following layer. The equations that describe of this operation are showed as the follows.

$$a^{m+1} = f^{m+1}\left(W^{m+1}a^{m} + b^{m+1}\right) \qquad \text{for } m = 0, 1, \ldots, M-1 \quad (A.1)$$

where *M* is the number of layers of the network. The neurons in the first layer receive external inputs:

$$a^{0} = p \qquad (A.2)$$

The outputs of the neurons in the last layer of the network are considered as the network outputs:

$$a = a^{m} \qquad (A.3)$$

A.1.1 Performance Index

The backpropagation algorithm for multilayer feed-forward neural networks use the mean square error as a criterion which is shown in equation (A.4).

$$
\begin{aligned}
F(x) = E\left[e^2\right] &= E\left[(t-a)^2\right] \\
&= E\left[(t-a)^T(t-a)\right] \qquad \text{(A.4)}
\end{aligned}
$$

where *x* is the vector of network weights and biases, *t* and *a* are the corresponding target output and actual output respectively. The steepest descent algorithm for the approximate mean square error is shown as the follows:

$$w_{i,j}^{m}(k+1) = w_{i,j}^{m}(k) - \alpha \frac{\partial \bar{F}}{\partial w_{i,j}^{m}} \qquad \text{(A.5)}$$

$$b_{i}^{m}(k+1) = b_{i}^{m}(k) - \alpha \frac{\partial \bar{F}}{\partial b_{i}^{m}} \qquad \text{(A.6)}$$

where α is the learning rate.

From equation (A.5) and (A.6), the error is an indirect function of the weights in the hidden layers. Therefore, the chain rule is used to determine the error gradient as the following:

$$\frac{\partial \bar{F}}{\partial w_{i,j}^{m}} = \frac{\partial \bar{F}}{\partial n_{i}^{m}} \times \frac{\partial n_{i}^{m}}{\partial w_{i,j}^{m}} \qquad \text{(A.7)}$$

$$\frac{\partial \bar{F}}{\partial b_{i}^{m}} = \frac{\partial \bar{F}}{\partial n_{i}^{m}} \times \frac{\partial n_{i}^{m}}{\partial b_{i}^{m}} \qquad \text{(A.8)}$$

The second term in each of these equations can be easily computed, since the net input to layer *m* is an explicit function of the weights and biases in that layer:

$$n_{i}^{m} = \sum_{j=1}^{S^{(m-1)}} w_{i,j}^{m} a_{j}^{m-1} + b_{i}^{m} \qquad \text{(A.9)}$$

Therefore,

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1}, \frac{\partial n_i^m}{\partial b_i^m} = 1 \tag{A.10}$$

Define

$$s_i^m = \frac{\partial \bar{F}}{\partial n_i^m} \tag{A.11}$$

which is the sensitivity of $\bar{F}$ to changes in the *ith* element of the net input at layer *m*, then equation (A.7) and (A.8) can be simplified to

$$\frac{\partial \bar{F}}{\partial w_{i,j}^m} = s_i^m a_j^{m-1} 1 \tag{A.12}$$

$$\frac{\partial \bar{F}}{\partial b_i^m} = s_i^m \tag{A.13}$$

Therefore, the approximate steepest descent algorithm is expressed as the follows.

$$w_{i,j}^m (k+1) = w_{i,j}^m (k) - \alpha s_i^m a_j^{m-1} \tag{A.14}$$

$$b_i^m (k+1) = b_i^m (k) - \alpha s_i^m \tag{A.15}$$

A.2 Training Function

A.2.1 Levenberg-Marquardt Method

The Levenberg-Marquardt method was designed to approach second order training speed without having the computing of the Hessian matrix. When the performance function has the form of a sum of squares which is typical in training feed-forward networks, then the Hessian matrix can be approximated as

$$H = J' J \tag{A.21}$$

and the gradient can be computed as:

$$\nabla f = J'e \tag{A.22}$$

where J is the Jacobian matrix, which contain the first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than the computing of the Hessian matrix.

The Levenberg-Marquardt method uses this approximation to the Hessian matrix in the following Newton like update.

$$x^{k+1} = x^{k} - \left[J'J + \mu I\right]^{-1} J'e \tag{A.23}$$

When the scalar μ is zero, this is just a Newton's method using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each of iteration in the algorithm.

APPENDIX B

NEURAL NETWORK ESTIMATOR

B.1 Neural network source code for validation

```
void train_on_steepness_file     (struct fann *ann, char *filename,
                                  unsigned int max_epochs, unsigned int
                                  epochs_between_reports,
                                  float desired_error, float steepness_start,
                                  float steepness_step, float steepness_end)
{
        float error;
        unsigned int i;

        struct fann_train_data *data = fann_read_train_from_file(filename);

        if(epochs_between_reports)
        {
                printf("Max epochs %8d. Desired error: %.10f\n", max_epochs,
                desired_error);
        }

        fann_set_activation_steepness_hidden(ann, steepness_start);
        fann_set_activation_steepness_output(ann, steepness_start);
        for(i = 1; i <= max_epochs; i++)
        {
                error = fann_train_epoch(ann, data);

                if(epochs_between_reports &&
                    (i % epochs_between_reports == 0 || i == max_epochs || i == 1 ||
                    error < desired_error))
                {
                    printf("Epochs     %8d. Current error: %.10f\n", i, error);
                }
                if(error < desired_error)
                {
                    steepness_start += steepness_step;
                    if(steepness_start <= steepness_end)
                    {
                      printf("Steepness: %f\n", steepness_start);
                      fann_set_activation_steepness_hidden(ann, steepness_start);
                      fann_set_activation_steepness_output(ann, steepness_start);
                    }
                    else
                    {
                      break;
                    }
                }
        }
        fann_destroy_train(data);
}

int main()
{
        const unsigned int num_input = 11;
        const unsigned int num_output = 1;
        const unsigned int num_layers = 3;
        const unsigned int num_neurons_hidden = 50;
        const float desired_error = (const float) 0.00001;
        const unsigned int max_epochs = 50000;
        const unsigned int epochs_between_reports = 10000;
        unsigned int i;
        fann_type *calc_out;

        struct fann_train_data *data;

        struct fann *ann = fann_create_standard(num_layers,   num_input,
        num_neurons_hidden, num_output);

        data = fann_read_train_from_file("testdata20061.txt");
```

```
        fann_set_activation_function_hidden(ann, FANN_SIGMOID_STEPWISE);
        fann_set_activation_function_output(ann, FANN_SIGMOID_STEPWISE);
        fann_set_training_algorithm(ann, FANN_TRAIN_RPROP);
        fann_set_learning_momentum(ann, 0.1);

        fann_train_on_file(ann, "testdata20061.txt", max_epochs, epochs_between_reports,
        desired_error);

        for(i = 0; i != fann_length_train_data(data); i++)
        {
                calc_out = fann_run(ann, data->input[i]);
                printf("XOR test (%f, %f) -> %f, should be %f, difference=%f\n",
                        data->input[i][0], data->input[i][1], calc_out[0], data->output[i][0],
                        (float) fann_abs(calc_out[0] - data->output[i][0]));
        }

        fann_save(ann, "testdata2006-c.net");

        fann_destroy(ann);
        fann_destroy_train(data);

        return 0;
}
```

B.2 Neural network source code for testing

```
void train_on_steepness_file    (struct fann *ann, char *filename,
                                unsigned int max_epochs, unsigned int
                                epochs_between_reports,
                                float desired_error, float steepness_start,
                                float steepness_step, float steepness_end)
{
        float error;
        unsigned int i;

        struct fann_train_data *data = fann_read_train_from_file(filename);
        if(epochs_between_reports)
        {
                printf("Max epochs %8d. Desired error: %.10f\n", max_epochs,
                desired_error);
        }

        fann_set_activation_steepness_hidden(ann, steepness_start);
        fann_set_activation_steepness_output(ann, steepness_start);
        for(i = 1; i <= max_epochs; i++)
        {
                error = fann_train_epoch(ann, data);

                if(epochs_between_reports &&
                    (i % epochs_between_reports == 0 || i == max_epochs || i == 1 ||
                    error < desired_error))
                {
                    printf("Epochs     %8d. Current error: %.10f\n", i, error);
                }

                if(error < desired_error)
                {
                    steepness_start += steepness_step;
                    if(steepness_start <= steepness_end)
                    {
                        printf("Steepness: %f\n", steepness_start);
                        fann_set_activation_steepness_hidden(ann, steepness_start);
                        fann_set_activation_steepness_output(ann, steepness_start);
                    }
                    else
                    {
                        break;
                    }
                }
        }
        fann_destroy_train(data);
}
```

```
int main()
{
        const unsigned int num_input = 11;
        const unsigned int num_output = 1;
        const unsigned int num_layers = 3;
        const unsigned int num_neurons_hidden = 50;
        const float desired_error = (const float) 0.00001;
        const unsigned int max_epochs = 50000;
        const unsigned int epochs_between_reports = 10000;
        unsigned int i;
        fann_type *calc_out;

        struct fann_train_data *data;

        struct fann *ann ;

        ann = fann_create_from_file("testdata2006-c.net");
        data = fann_read_train_from_file("testdata2007.txt");

        for(i = 0; i != fann_length_train_data(data); i++)
        {
                calc_out = fann_run(ann, data->input[i]);
                printf("XOR test (%f, %f) -> %f, should be %f, difference=%f\n",
                        data->input[i][0], data->input[i][1], calc_out[0], data->output[i][0],
                        (float) fann_abs(calc_out[0] - data->output[i][0]));
        }



        fann_destroy(ann);
        fann_destroy_train(data);

        return 0;
}
```

APPENDIX C

NEURAL NETWORK FOR OPTIMIZATION

C.1 Source code for input data

```
9
1
1
0
0
3
20
20
10
0.00000000000000001
0.0001
0.0000000001
20
2
1
1
0.8
0.8
0.2
3.0 700.0
3.0 700.0
```

%1. int algorithm; 0:Random; 1:DE/best/1/exp; 2:DE/rand/1/exp; 3:DE/rand-to-best/1/exp; 4:DE/best/2/exp; 5:DE/rand/2/exp; 6:DE/best/1/bin; 7:DE/rand/1/bin; 8:DE/rand-to-best/1/bin; 9:DE/best/2/bin; 10:DE/rand/2/bin , recommend 6 & 9 with F=0.001 & CR=0

%2. int trigonometric mutation; 0: No; 1: Yes

%3. int fast_update algorithm; 0: No; 1: Yes

%4. int correction algorithm; 0: at bound; 1: non-bound

%5. int gradient correction; 0: No; 1: Yes

```
%6.  int sampling technique; 0: No; 1: Sobol; 2: Niederreiter 3: HSS
%7. int generation
%8. int population
%9. int stall generation limit
%10. double obj function tolerance limit
%11. double delta (for correction)
%12. double correction tolerance limit
%13. int correction_no
%14. int no. of variable
%15. int no. of ineq. constraints
%16. int no. of eq. constraints
%17. double F              (-1.0 for random)
%18. double CR(-1.0 for random)
%19. double mutation   (-1.0 for random)
%20. double LB UB
.
.
.

029918284
```

C.2 Source code for optimization

```
#include <stdio.h>
#include <stdlib.h>
#include <gsl/gsl_vector.h>
#include <gsl/gsl_matrix.h>
#include <gsl/gsl_errno.h>
#include <gsl/gsl_math.h>
#include "floatfann.h"

double mymax(double comval1, double comval2){
        if (comval1 > comval2 ) {
                return comval1;
        } else{
                return comval2;
```

```
    }
}

double mymin( double comval1, double comval2){
    if (comval1 < comval2 ) {
        return comval1;
    } else{
        return comval2;
    }
}

double objfunc(gsl_vector *variable, double *gsum, gsl_vector *g, double *hsum, gsl_vector
*h, int flag){
extern int fno, gno, hno, nog, noh;
double tolh = 1.0e-6, tolg = 1.0e-6;
    if (flag==0){
        fno = fno + 1;
            nog = nog + 1;
            noh = noh + 1;
        } else if (flag==1){
            fno = fno +1;
        } else if (flag==2){
            nog = nog +1;
        } else if (flag==3){
            noh = noh +1;
        }

    register int i;

    int j, time=1095;
    int t1i, t2i;
    double t1, t2, totalcost=0;

    t1i = (int) gsl_vector_get(variable, 0);;
    t1 = (double) t1i;

    t2i = (int) gsl_vector_get(variable, 1);    ;
```

```
t2 = (double) t2i;

////////////////////////g constraint/////////////////////
gsl_vector_set(g, 0, time-t1-t2);

for (i=0, *gsum=0.0; i<gno; i++){
    if (gsl_vector_get(g, i) <= -1.0*tolg) *gsum += -1.0 * gsl_vector_get(g, i);
}

///////////////////////////h constraint////////////////////
gsl_vector_set(h, 0, 0);

for (i=0, *hsum=0.0; i<hno; i++){
     if (fabs(gsl_vector_get(h, i)) >= tolh) *hsum +=  1.0*fabs(gsl_vector_get(h, i));
}

////////////////////////obj function///////////

   fann_type *calc_out;
   fann_type input[11];
   struct fann *ann = fann_create_from_file("testdata2006-c.net");
        double R=0.7139, P=0.9035;
        double U0=0.7221, U1=0.7211, U2=0.7104;
        double *U = malloc( sizeof(double)*time );
        double *UR = malloc( sizeof(double)*time );
        double *Q = malloc( sizeof(double)*time );
        double *cost_Q = malloc( sizeof(double)*time );
        double *Tc2 = malloc( sizeof(double)*time );
        double *Th2 = malloc( sizeof(double)*time );
        double A=0.0002851, Th1=360, Tc1=136, Fh=0.124, Fc=0.124, cph=960,
cpc=820, Tr=332, Qcostfactor=4.1, cleancostfactor=200000;
        double k1,k2,Mh,Mc;

if ((t1>0)&&(t2>0)&&(t1+t2 <= time)){
        for (j=0; j < t1; j++) {
                if (j==0){
                        U[j]=U0;
                        UR[j]=(1246889*(U[j]+1)/2);
                        k1 = (Fh*cph)/(Fc*cpc);
                        k2 = (UR[j]*A)/(Fh*cph);
                        Mh = k1*((exp(-k2*(k1-1)))-1)/(exp(-k2*(k1-1)) - k1);
                        Mc = (1-k1)*exp(-k2*(k1-1))/(exp(-k2*(k1-1)) - k1);
                        Tc2[j]=Mh*Th1+Mc*Tc1;
                        Th2[j]=Th1-(Tc2[j]-Tc1)/k1;
                        Q[j]=cpc*Fc*(Tr-Tc2[j]);
                        cost_Q[j]=Q[j]*Qcostfactor;
                } else if (j==1) {
                        U[j]=U1;
                        UR[j]=(1246889*(U[j]+1)/2);
                        k1 = (Fh*cph)/(Fc*cpc);
                        k2 = (UR[j]*A)/(Fh*cph);
                        Mh = k1*((exp(-k2*(k1-1)))-1)/(exp(-k2*(k1-1)) - k1);
                        Mc = (1-k1)*exp(-k2*(k1-1))/(exp(-k2*(k1-1)) - k1);
                        Tc2[j]=Mh*Th1+Mc*Tc1;
                        Th2[j]=Th1-(Tc2[j]-Tc1)/k1;
                        Q[j]=cpc*Fc*(Tr-Tc2[j]);
                        cost_Q[j]=Q[j]*Qcostfactor;

                } else if (j==2) {
                        U[j]=U2;
                        UR[j]=(1246889*(U[j]+1)/2);
                        k1 = (Fh*cph)/(Fc*cpc);
                        k2 = (UR[j]*A)/(Fh*cph);
                        Mh = k1*((exp(-k2*(k1-1)))-1)/(exp(-k2*(k1-1)) - k1);
                        Mc = (1-k1)*exp(-k2*(k1-1))/(exp(-k2*(k1-1)) - k1);
                        Tc2[j]=Mh*Th1+Mc*Tc1;
                        Th2[j]=Th1-(Tc2[j]-Tc1)/k1;
                        Q[j]=cpc*Fc*(Tr-Tc2[j]);
                        cost_Q[j]=Q[j]*Qcostfactor;
                } else{
                        input[0] = R; //R(i)
                        input[1] = R; //R(i-1)
                        input[2] = R; //R(i-2)
                        input[3] = R; //R(i-3)
                        input[4] = P; //P(i)
                        input[5] = P; //P(i-1)
                        input[6] = P; //P(i-2)
                        input[7] = P; //P(i-3)
                        input[8] = U[j-1]; //U(i-1)
                        input[9] = U[j-2]; //U(i-2)
                        input[10] = U[j-3]; //U(i-3)
                        calc_out = fann_run(ann, input);
                        U[j] = calc_out[0];
                        UR[j]=(1246889*(U[j]+1)/2);
                        k1 = (Fh*cph)/(Fc*cpc);
                        k2 = (UR[j]*A)/(Fh*cph);
                        Mh = k1*((exp(-k2*(k1-1)))-1)/(exp(-k2*(k1-1)) - k1);
                        Mc = (1-k1)*exp(-k2*(k1-1))/(exp(-k2*(k1-1)) - k1);
                        Tc2[j]=Mh*Th1+Mc*Tc1;
                        Th2[j]=Th1-(Tc2[j]-Tc1)/k1;
                        Q[j]=cpc*Fc*(Tr-Tc2[j]);
                        cost_Q[j]=Q[j]*Qcostfactor;
                    }
                }

                for (j=t1; j < t1+t2; j++){
                    if (j-t1==0){
                        U[j]=U0;
                        UR[j]=(1246889*(U[j]+1)/2);
                        k1 = (Fh*cph)/(Fc*cpc);
                        k2 = (UR[j]*A)/(Fh*cph);
                        Mh = k1*((exp(-k2*(k1-1)))-1)/(exp(-k2*(k1-1)) - k1);
                        Mc = (1-k1)*exp(-k2*(k1-1))/(exp(-k2*(k1-1)) - k1);
                        Tc2[j]=Mh*Th1+Mc*Tc1;
                        Th2[j]=Th1-(Tc2[j]-Tc1)/k1;
                        Q[j]=cpc*Fc*(Tr-Tc2[j]);
                        cost_Q[j]=Q[j]*Qcostfactor;
                    } else if (j-t1==1) {
                        U[j]=U1;
                        UR[j]=(1246889*(U[j]+1)/2);
                        k1 = (Fh*cph)/(Fc*cpc);
                        k2 = (UR[j]*A)/(Fh*cph);
                        Mh = k1*((exp(-k2*(k1-1)))-1)/(exp(-k2*(k1-1)) - k1);
                        Mc = (1-k1)*exp(-k2*(k1-1))/(exp(-k2*(k1-1)) - k1);
                        Tc2[j]=Mh*Th1+Mc*Tc1;
                        Th2[j]=Th1-(Tc2[j]-Tc1)/k1;
                        Q[j]=cpc*Fc*(Tr-Tc2[j]);
                        cost_Q[j]=Q[j]*Qcostfactor;
                } else if (j-t1==2) {
                        U[j]=U2;
                        UR[j]=(1246889*(U[j]+1)/2);
                        k1 = (Fh*cph)/(Fc*cpc);
                        k2 = (UR[j]*A)/(Fh*cph);
                        Mh = k1*((exp(-k2*(k1-1)))-1)/(exp(-k2*(k1-1)) - k1);
                        Mc = (1-k1)*exp(-k2*(k1-1))/(exp(-k2*(k1-1)) - k1);
                        Tc2[j]=Mh*Th1+Mc*Tc1;
                        Th2[j]=Th1-(Tc2[j]-Tc1)/k1;
                        Q[j]=cpc*Fc*(Tr-Tc2[j]);
                        cost_Q[j]=Q[j]*Qcostfactor;
                } else{
                        input[0] = R; //R(i)
                        input[1] = R; //R(i-1)
                        input[2] = R; //R(i-2)
                        input[3] = R; //R(i-3)
                        input[4] = P; //P(i)
                        input[5] = P; //P(i-1)
                        input[6] = P; //P(i-2)
                        input[7] = P; //P(i-3)
                        input[8] = U[j-1]; //U(i-1)
                        input[9] = U[j-2]; //U(i-2)
                        input[10] = U[j-3]; //U(i-3)
                        calc_out = fann_run(ann, input);
                        U[j] = calc_out[0];
                        UR[j]=(1246889*(U[j]+1)/2);
                        k1 = (Fh*cph)/(Fc*cpc);
                        k2 = (UR[j]*A)/(Fh*cph);
                        Mh = k1*((exp(-k2*(k1-1)))-1)/(exp(-k2*(k1-1)) - k1);
                        Mc = (1-k1)*exp(-k2*(k1-1))/(exp(-k2*(k1-1)) - k1);
```

```
                    Tc2[j]=Mh*Th1+Mc*Tc1;
                    Th2[j]=Th1-(Tc2[j]-Tc1)/k1;
                    Q[j]=cpc*Fc*(Tr-Tc2[j]);
                    cost_Q[j]=Q[j]*Qcostfactor;
          }
}

for (j=t1+t2; j < time; j++){
          if (j-t1-t2==0){
                    U[j]=U0;
                    UR[j]=(1246889*(U[j]+1)/2);
                    k1 = (Fh*cph)/(Fc*cpc);
                    k2 = (UR[j]*A)/(Fh*cph);
                    Mh = k1*((exp(-k2*(k1-1)))-1)/(exp(-k2*(k1-1)) - k1);
                    Mc = (1-k1)*exp(-k2*(k1-1))/(exp(-k2*(k1-1)) - k1);
                    Tc2[j]=Mh*Th1+Mc*Tc1;
                    Th2[j]=Th1-(Tc2[j]-Tc1)/k1;
                    Q[j]=cpc*Fc*(Tr-Tc2[j]);
                    cost_Q[j]=Q[j]*Qcostfactor;
          } else if (j-t1-t2==1) {
                    U[j]=U1;
                    UR[j]=(1246889*(U[j]+1)/2);
                    k1 = (Fh*cph)/(Fc*cpc);
                    k2 = (UR[j]*A)/(Fh*cph);
                    Mh = k1*((exp(-k2*(k1-1)))-1)/(exp(-k2*(k1-1)) - k1);
                    Mc = (1-k1)*exp(-k2*(k1-1))/(exp(-k2*(k1-1)) - k1);
                    Tc2[j]=Mh*Th1+Mc*Tc1;
                    Th2[j]=Th1-(Tc2[j]-Tc1)/k1;
                    Q[j]=cpc*Fc*(Tr-Tc2[j]);
                    cost_Q[j]=Q[j]*Qcostfactor;
          } else if (j-t1-t2==2) {
                    U[j]=U2;
                    UR[j]=(1246889*(U[j]+1)/2);
                    k1 = (Fh*cph)/(Fc*cpc);
                    k2 = (UR[j]*A)/(Fh*cph);
                    Mh = k1*((exp(-k2*(k1-1)))-1)/(exp(-k2*(k1-1)) - k1);
                    Mc = (1-k1)*exp(-k2*(k1-1))/(exp(-k2*(k1-1)) - k1);
```

```
        Tc2[j]=Mh*Th1+Mc*Tc1;
        Th2[j]=Th1-(Tc2[j]-Tc1)/k1;
        Q[j]=cpc*Fc*(Tr-Tc2[j]);
        cost_Q[j]=Q[j]*Qcostfactor;
    } else{
        input[0] = R; //R(i)
        input[1] = R; //R(i-1)
        input[2] = R; //R(i-2)
        input[3] = R; //R(i-3)
        input[4] = P; //P(i)
        input[5] = P; //P(i-1)
        input[6] = P; //P(i-2)
        input[7] = P; //P(i-3)
        input[8] = U[j-1]; //U(i-1)
        input[9] = U[j-2]; //U(i-2)
        input[10] = U[j-3]; //U(i-3)
        calc_out = fann_run(ann, input);
        U[j] = calc_out[0];
        UR[j]=(1246889*(U[j]+1)/2);
        k1 = (Fh*cph)/(Fc*cpc);
        k2 = (UR[j]*A)/(Fh*cph);
        Mh = k1*((exp(-k2*(k1-1)))-1)/(exp(-k2*(k1-1)) - k1);
        Mc = (1-k1)*exp(-k2*(k1-1))/(exp(-k2*(k1-1)) - k1);
        Tc2[j]=Mh*Th1+Mc*Tc1;
        Th2[j]=Th1-(Tc2[j]-Tc1)/k1;
        Q[j]=cpc*Fc*(Tr-Tc2[j]);
        cost_Q[j]=Q[j]*Qcostfactor;
    }
}

for (j=0; j < time; j++){

totalcost = totalcost + cost_Q[j];
}


totalcost = totalcost + 2*cleancostfactor;
```

```
		} else {
				totalcost=1e99;
		}


				fann_destroy(ann);
				free(U);
				free(UR);
				free(Q);
				free(cost_Q);
				free(Tc2);
				free(Th2);
		return totalcost;

	}
```

VITA

Miss Prapitpan Techawanitchai was born in Suphanburi, Thailand, on November 27, 1982. She graduated her bachelor Degree from Department of Chemical Technology in the Faculty of Science from Chulalongkorn University in 2004. In 2005, she entered the Graduated School of Chulalongkorn University to propose a Master of Engineering in Chemical Engineering and completed in 2008.