การจำลองและออกแบบตัวควบคุมข่ายงานนิวรัลของกระบวนการผลิตเมทิลเมทาคริเลตสำหรับปฏิกิริยาเอสเทอริฟิเคชันในเครื่องปฏิกรณ์แบบแบตช์

นางสาวธนัชพร เจริญนิยม

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเคมี ภาควิชาวิศวกรรมเคมี
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2554

NEURAL NETWORK MODELING AND CONTROLLER DESIGN OF THE METHYL METHACRYLATE PRODUCTION PROCESS FOR ESTERIFICATION REACTION IN A BATCH REACTOR

Miss Thanutchaporn Charoenniyom

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 2011

Thesis Title	NEURAL NETWORK MODELING AND CONTROLLER DESIGN OF THE METHYL METHACRYLATE PRODUCTION PROCESS FOR ESTERIFICATION REACTION IN A BATCH REACTOR
By	Miss. Thanutchaporn Charoenniyom
Field of Study	Chemical Engineering
Thesis Advisor	Professor Paisan Kittisupakorn, 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

……………………………………………….. Chairman
(Associate Professor Tharathon Mongkhonsi, Ph.D.)

……………………………………….……. Thesis Advisor
(Professor Paisan Kittisupakorn, Ph.D.)

……………………………………………….. Examiner
(Assistant Professor Soorathep Kheawhom, Ph.D.)

……………………………………………….. External Examiner
(Wachira Daosud, D. Eng.)

ธนัชพร เจริญนิยม : การจำลองและออกแบบตัวควบคุมข่ายงานนิวรัลของกระบวนการผลิตเมทิลเมทาคริเลตสำหรับปฏิกิริยาเอสเทอริฟิเคชันในเครื่องปฏิกรณ์แบบแบตช์ (NEURAL NETWORK MODELING AND CONTROLLER DESIGN OF THE METHYL METHACRYLATE PRODUCTION PROCESS FOR ESTERIFICATION REACTION IN A BATCH REACTOR) อ. ที่ปรึกษาวิทยานิพนธ์หลัก: ศ. ดร. ไพศาล กิตติศุภกร, 115 หน้า

งานวิจัยนี้เสนอการจำลองข่ายงานนิวรัลสำหรับทำนายโปรไฟล์ความเข้มข้นและอุณหภูมิในเครื่องปฏิกรณ์แบบแบตช์สำหรับการผลิตเมทิลเมทาคริเลตและการจำลองข่ายงานนิวรัลแบบผกผันเพื่อทำนายค่าอุณหภูมิแจ๊กเก็ตเป้าหมายของระบบให้ความร้อน/ทำความเย็น การฝึกข่ายงานนิวรัลใช้เลเวนเบิร์ก-มาร์ควอร์ทเป็นอัลกอริทึม โครงสร้างที่เหมาะสมของข่ายงานนิวรัลที่ได้รับใช้ทำนายค่าตัวแปรเสตทภายในอัลกอริทึมของการควบคุมทำนายแบบจำลอง เพื่อใช้หาค่าตัวแปรปรับกระบวนการที่เหมาะสมโดยใช้วิธีการออปติไมซ์แบบซักเซสซีฟควอดราติกโปรแกรมมิ่ง (SQP) เพื่อควบคุมอุณหภูมิในเครื่องปฏิกรณ์แบบแบตช์ การควบคุมที่อาศัยข่ายงานนิวรัลที่ใช้ในงานวิจัยนี้ประกอบด้วยการควบคุมข่ายงานนิวรัลแบบผกผัน และการควบคุมทำนายแบบจำลองร่วมกับข่ายงานนิวรัล อีกทั้งงานวิจัยนี้ได้ศึกษาวิธีการออปติไมซ์เพื่อหาโปรไฟล์ของอุณหภูมิในเครื่องปฏิกรณ์เพื่อให้ได้ผลิตภัณฑ์มากที่สุดที่เวลาสุดท้าย อุณหภูมิออปติมอลที่ได้จากการออปติไมซ์แบบพลวัติกำหนดให้เป็นค่าเป้าหมายสำหรับการออกแบบตัวควบคุม การทดสอบสมรรถนะของการควบคุมที่นำเสนอศึกษาโดยการเปลี่ยนค่าพารามิเตอร์ปฏิบัติการ ผลการจำลองพบว่าตัวควบคุมทำนายแบบจำลองร่วมกับข่ายงานนิวรัลทนทานกว่าตัวการควบคุมแบบสัดส่วนปริพันธ์อนุพันธ์ และตัวควบคุมข่ายงานนิวรัลแบบผกผันทนทานกว่าตัวควบคุมแบบสัดส่วนปริพันธ์อนุพันธ์ ดังนั้น ตัวควบคุมทำนายแบบจำลองร่วมกับข่ายงานนิวรัลให้ผลลัพธ์ดีที่สุดเมื่อเปรียบเทียบกับตัวการควบคุมแบบสัดส่วนปริพันธ์อนุพันธ์ และตัวควบคุมข่ายงานนิวรัลแบบผกผันในกรณีปกติ (แบบจำลองถูกต้อง) และกรณีแบบจำลองผิดพลาด

ภาควิชา วิศวกรรมเคมี ลายมือชื่อนิสิต ____________

สาขาวิชา วิศวกรรมเคมี ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก ____________

ปีการศึกษา 2554

527 03244 21 : MAJOR CHEMICAL ENGINEERING
KEYWORDS : METHYL METHACRYLATE / NEURAL NETWORK BASED MODEL PREDICTIVE CONTROL / OPTIMIZATION

THANUTCHAPORN CHAROENNIYOM : NEURAL NETWORK MODELING AND CONTROLLER DESIGN OF THE METHYL METHACRYLATE PRODUCTION PROCESS FOR ESTERIFICATION REACTION IN A BATCH REACTOR. ADVISOR : PROF. PAISAN KITTISUPAKORN, Ph.D., 115 pp.

This work presents a neural network forward model to predict a concentration and temperature profiles in a batch reactor for a MMA production, and a neural network inverse model to predict a jacket temperature set point of a heating/cooling system. The neural network forward and inverse models have been developed based on the Lenvenberg-Marquardt training algorithm. An obtained optimal neural network structure for the forward model has been employed to predict state variables over a predictive horizon within a model predictive control (MPC) algorithm for searching optimal control actions via successive quadratic programming (SQP). To control the temperature in the batch reactor, the neural network based control approaches studied in this work consisting of a neural network direct inverse control (NNDIC) and a neural network based model predictive control (NNMPC) have been formulated. In addition, a dynamic optimization approach has been applied to find out an optimal operating temperature to achieve maximizing the MMA product at specified final time. An obtained optimal temperature is then applied as a set point for the controller design. Robustness tests of the proposed controllers have been studied with respect to the changes in operating parameters. Simulation results have indicated that the NNMPC controller is more robust than the PID and NNDIC controllers and the NNDIC controller is more robust than the PID controller. Therefore, the NNMPC controller gives the best control results among the PID and NNDIC controllers in the nominal and plant/model mismatch cases.

Department : Chemical Engineering Student's Signature ....

Field of Study : Chemical Engineering Advisor's Signature ....

Academic Year : 2011

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my thesis advisor, Professor Paisan Kittisupaakorn for the continuous support of my thesis, for his patience, motivation, enthusiasm and immense knowledge. His wide knowledge and his logical way of thinking have been of great value for me. I could not have imagined having a better advisor for my Master Degree study.

I am deeply grateful Dr. Wachira Daosud, Department of Chemical Engineering, Bhurapa University, for her great guidance, encouragement and support. Under her supervision, I learned a lot of neural network and controller design as well as the process control and techniques.

I wish to especially thank to the other members of my thesis committee, Associate Professor Tharathon Mongkhonsi, Assistant Professor Soorathep Kheawhom for their time and the useful comments on this thesis.

I would like to thank my friends at the Control and Systems Engineering Laboratory for their good spirit shared, supports and wonderful times, we have had over the years. Their friendships have encouraged me to overcome many difficulties.

Finally, I would like to express my deepest gratitude to my beloved parents for their endless love, inspiration understanding and encouragement.

CONTENTS

Page

Abstract in Thai iv
Abstract in English v
Acknowledgements vi
Contents vii
List of Tables x
List of Figures xi
Nomenclature xiv

Chapter I INTRODUCTION 1
1.1 Research Objective 3
1.2 Scopes of Research 3
1.3 Contributions of Research 3
1.4 Methodology of Research 4

Chapter II LITERATURE REVIEWS 5
2.1 Methyl methacrylate Production Process 5
2.2 Neural Network Modeling and Controller Design 6
2.3 Neural Network base Model Predictive Control 9

Chapter III THEORY 11
3.1 Methyl methacrylate 11
3.2 Methyl methacrylate Manufacturing 12
3.2.1 C2 routes 12
3.2.2 C3 routes 14
3.2.3 C4 routes 15
3.3 The Mathematical Model for Methyl Methacrylate Process 17
3.3.1 Mass balance 18
3.3.2 Energy balance 18
3.4 Neural Network Introduction 19
3.5 Components of Neural Network 20

Page

3.5.1 Summation function 20
3.5.2 Transfer function 21
3.5.2.1 Linear transfer function 21
3.5.2.2 Log-Sigmoid transfer function 21
3.5.2.3 Tan-Sigmoid transfer function 23
3.5.3 Error function 23
3.5.3.1 Sum square error 23
3.5.3.2 Mean square error 24
3.5.4 Learning Function 24
3.5.4.1 Supervised learning 24
3.5.4.2 Unsupervised learning 25
3.6 Neural Network Architecture 26
3.6.1 Network Structure 26
3.6.1.1 Feedforward neural networks 26
3.6.1.2 Feedback neural networks 26
3.6.2 Network Layers 27
3.7 Training Algorithm 30
3.7.1 Back propagation Algorithm 30
3.7.2 Levenberg-Marquardt method 33
3.8 Model Predictive Control 34
3.9 Optimization 37

Chapter IV NEURAL NETWORK FORWARD MODEL AND NEURAL NETWORK INVERSE MODEL FOR THE PROCESS 40
4.1 Neural network modeling 40
4.2 Neural Network Inverse Model 48
4.3 Simulation Results 48
4.4 Dynamic Optimization 56

Page

Chapter V THE CONTROLLER DESIGN BASED NEURAL NETWORK FOR THE TEMPERATURE CONTROL OF THE PROCESS 59

5.1 NNMPC strategy 59

5.2 Neural network direct inverse control (NNDIC) strategy 60

5.3 Simulation Results 61

Chapter VI CONCLUSIONS 79

6.1 Neural Network Forward Model and Inverse Model 79

6.2 Neural Network based Model Predictive Control 80

REFERENCE 81

APPENDICES 87

Appendix A Mean Square Error of Neural Network Modeling 88

Appendix B Process Parameters 94

Appendix C Proportional Integral Derivative Control (PID) Strategy 96

Appendix D Controller Performance Indices 97

Appendix E Euler's Method 99

VITA 100

LIST OF TABLES

Page

Table 3.1 MMA property 11

Table 3.2 Learning functions 24

Table 4.1 Optimization results 56

Table 5.1 Performance index of NNMPC strategy, PID control strategy and DIC strategy for nominal and model mismatch cases 64

Table 5.2 Performance indices of the NNMPC and the PID control for the best case 71

Table A.1 Mean squared error value of the neural network forward model 1 layer 87

Table A.2 Mean squared error value of the neural network forward model 2 layer 87

Table A.3 Mean squared error value of the neural network inverse model 1 layer 90

Table A.4 Mean squared error value of the neural network inverse model 2 layer 90

Table B.1 Esterification and physical properties of MMA system 93

Table B.2 Initial condition of MMA system 94

LIST OF FIGURES

		Page
Figure 3.1	Molecular structure of MMA	11
Figure 3.2	Routes to MMA	13
Figure 3.3	Axons, dendrites and synapse in a biological neuron	20
Figure 3.4	Linear transfer function	22
Figure 3.5	Log-Sigmoid transfer function	22
Figure 3.6	Tan-Sigmoid transfer function	23
Figure 3.7	Supervised learning	25
Figure 3.8	Unsupervised learning	25
Figure 3.9	Feedforward neural networks	26
Figure 3.10	Feedback neural networks	27
Figure 3.11	Single-layers neural networks	28
Figure 3.12	Multi-layers neural networks	29
Figure 3.13	Back Propagation of the Error in a Two-Layer Network	31
Figure 3.14	Forward Propagation and Backward Propagation in Training Phase	31
Figure 3.15	MPC Strategy	35
Figure 3.16	Neural network model predictive flowchart	36
Figure 4.1	Procedure for obtaining forward and inverse neural network model	43
Figure 4.2	Set point of the jacket temperature (T_{jsp}) for the training data set 1 of the forward model	44
Figure 4.3	The jacket temperature (T_{jsp}) for the training data set 1 of the forward model	44
Figure 4.4	Concentration profile of MMA for the training data set 1 of the forward model	45
Figure 4.5	Reactor temperature profile (Tr) for the training data set 1 of the forward model	45
Figure 4.6	Data summation of set point of jacket temperature for forward model	46
Figure 4.7	Data summation of the jacket temperature for the forward model	46
Figure 4.8	Data summation of concentration profile of MMA for forward model	47

Page

Figure 4.9 Data summation of reactor temperature for forward model 47

Figure 4.10 The neural network forward of the process 49

Figure 4.11 The neural network inverse model of the process 50

Figure 4.12 The testing set 1 result of concentration of MMA for neural network model (structure 8-6-8-3) 50

Figure 4.13 The testing set 1 result of reactor temperature for neural network model (structure 8-6-8-3) 51

Figure 4.14 The testing set 1 result of jacket temperature for the neural network forward model (structure 8-6-8-3) 51

Figure 4.15 The testing set 2 result of concentration of MMA for neural network model (structure 8-6-8-3) 52

Figure 4.16 The testing set 2 result of reactor temperature for neural network model (structure 8-6-8-3) 52

Figure 4.17 The testing set 2 result of jacket temperature for neural network Forward model (structure 8-6-8-3) 53

Figure 4.18 The validating set result of concentration of MMA for neural network forward model (structure 8-6-8-3) 53

Figure 4.19 The validating set result of reactor temperature for neural network forward model (structure 8-6-8-3) 54

Figure 4.20 The validating set result of jacket temperature for neural network Forward model (structure 8-6-8-3) 54

Figure 4.21 The testing set 1 result of manipulated variable for neural network inverse model (structure 8-4-8-1) 55

Figure 4.22 The testing set 2 result of manipulated variable for neural network inverse model (structure 8-4-8-1) 55

Figure 4.23 The validating set result of manipulated variable for neural network inverse model (structure 8-4-8-1) 56

Figure 4.24 Temperature profile for the optimization problem 58

Figure 5.1 The NNMPC strategy 60

Figure 5.2 The structure of NNDIC strategy 61

Figure 5.3 The temperature control using NNMPC under the nominal case 62

Page

Figure 5.4 The temperature control using PID control under the nominal case 63

Figure 5.5 The temperature control using NNDIC under the nominal case 64

Figure 5.6 The temperature control using NNMPC under the parameter mismatch case (-30% U) 66

Figure 5.7 The temperature control using PID control under the parameter mismatch case (-30% U) 67

Figure 5.8 The NNDIC strategy for controlling of reactor temperature for increasing case (-30% U) 68

Figure 5.9 The temperature control using NNMPC under the parameter mismatch case (+30% k_1, -30% k_2, +30% ΔH and -30% U) 69

Figure 5.10 The temperature control using PID control under the parameter mismatch case (+30% k_1, -30% k_2, +30% ΔH and -30% U) 70

Figure 5.11 The temperature control using NNDIC under the parameter mismatch case (+30% k_1, -30% k_2, +30% ΔH and -30%U) 71

Figure 5.12 The temperature control using the best case of NNMPC under the nominal case 73

Figure 5.13 The temperature control using the best case of PID control under the nominal case 74

Figure 5.14 The temperature control using the best case of NNMPC under the parameter mismatch case (-30% of U) 75

Figure 5.15 The temperature control using the best case of PID control under the parameter mismatch case (-30% of U) 76

Figure 5.16 The temperature control using the best case of NNMPC under the parameter mismatch case (+30%k_1, -30% k_2, +30% ΔH and -30% U) 77

Figure 5.17 The temperature control using the best case of PID control under the parameter mismatch case (+30% k_1, -30% k_2, +30% ΔH and -30% U) 78

Figure C.1 Closed loop control of PID 95

Figure D.1 Performance characteristics for the step response of an under damped process 98

Page

Figure E.1 Euler's method for a range 99

NOMENCLATURE

A, B, C, D	Correlation constants for each compound
C_i	Concentration of component *i* (mol/m^3)
C_{pi}	Liquid heat capacity at 293.15 K (*J/*(*mol K*))
E_a	Activation energy (*J/mol*)
k_0	Frequency factor
k_1, k_2	Reaction rate constants ($m^{5.1}$/($mol^{1.7}$ *min*))
M	Control horizon
MW_i	Molecular weight of component *i* (*kg/kmol*)
P	Prediction horizon
R_i	Reaction rate of component *i* (*mol/*(m^3min))
T_r	Reactor temperature (*K*)
T_j	Jacket temperature (*K*)
T_{jsp}	Set point of jacket temperature (*K*)
V_j	Volume of jacket (m^3)
w_j	Weight of neural network
W_j	Weight for tuning of NNMPC
ΔH	Heat of reaction (*J/mol*)
ρ_i	Density of component *i* (*kmol/m^3*)

CHAPTER I

INTRODUCTION

Methyl methacrylate (MMA), which is carried out for producing polymethyl methacrylate (acrylic plastics) and polymer dispersions, is an important chemical polymer intermediate. The world production capacity has been double increased in past 15 years and the MMA demand is still expected growth in the future (Spivey et al., 1997). The MMA can be manufactured by many routes (Nagai, 2001). An esterification of mathacrylic acid with methanol in a batch reactor is a successful route for producing MMA on account of providing the maximum yield (Witczak and Skrzypek, 2010). In a batch reactor with exothermic reactions, the heat-released of reactions in heating period may become very large very quickly and the heat-generated exceeds the cooling capacity of the reactor (Konakom, Kittisupakorn and Muujtaba, 2008; Mujtaba, Aziz and Hussain, 2006). As a consequence, the temperature reaches to runaway. To overcome this problem, conventional control strategies have been carried out to solve this problem (Aziz, Hussain and Mujtaba, 2000; Cho, Edgar and Lee, 2008; Babu and Jyotsna, 2001; Szeifert, Chovan and Nagy, 1999). In contrast, the conventional model based control strategies can be applied to control the systems, the controller performance be dependent on the accuracy of the mathematical model.

To improve the control performance, neural networks is a new interested approach that can be carried out successfully to capture the dynamics of nonlinear and complex systems (Alippi and Piuri, 1999; Mujtaba, Aziz and Hussain, 2006; Kim et al., 2004; Loh and Fong, 1995; Murray, Neumerkel and Sbarbaro, 1992; Yu and Gomm, 2003). Neural networks have the advantages of distributed information processing and the inherent potential for parallel computation. The potential for the processing and approximation relates to operating data without the prior knowledge of the process. It can learn adequately accurate models and give good non-linear control when model equations are not know or only partial state information is available. Neural networks can be employed to be a mathematical model, an estimator and a controller. Kittisupakorn et al. (2005) demonstrated dynamic neural network modeling for hydrochloric acid recovery acid process to predict the concentration profile of a hydrochloric acid recovery process consisting of double fixed-bed ion exchange columns. Rusinowski and Stanek (2007) presented a method and example results of calculations of neural modeling of steam boilers. Charoenniyom et al. (2011) applied neural network to be a modeling for the methyl methacrylate production process in a batch reactor and Thampasato et al. (2011) proposed neural network modeling for a batch crystallizer. For the process control, Nueaklong et al. (2011) investigated neural network modeling for a hard chrome electroplating process to predict the plating solution temperature in a hard chrome electroplating bath and applied the neural network inverse model as a controller for controlling plating solution temperature to the desired temperature range. Daosud et al. (2005) presented the neural network for inverse model to be a controller for a steel pickling process. Kittisupakorn et al. (2009) presented a multi-layer feedforward neural network based model predictive control for a steel pickling process. The neural network for forward model is applied as mathematical model to predict the state variables in the model predictive control algorithm. For the use of a neural networks as an estimator, Arpornwichanop and Shomchoam (2009) applied neural network as an estimator to estimate the unmeasured state variables for fed-batch bioreactors.

The goal of this work is to improve a control technique to control a MMA production process. To achieve this, this work is applied a neural network forward model to predict a dynamics behavior and a neural network inverse model to control the process integrated with the dynamic optimization. Both neural networks are trained based on Levenberg-Marquardt algorithm. Optimal structures of the neural network are chose based on mean square error (MSE). An obtained optimal structure for forward model is carried out to predict a reactor temperature over a prediction horizon within model predictive control algorithm for searching optimal control actions via successive quadratic programming (SQP). The neural network controls consist of a neural network direct inverse control (NNDIC) and a neural network based model predictive control (NNMPC). Robustness of the proposed controls is investigated with respect to parameters mismatch. In addition, this work presents a dynamic optimization to find out an optimal operating temperature to achieve maximizing the MMA product. An optimal temperature result is represented a set point for the control design.

1.1 Research Objective

The objective of this research is aimed at carrying out process modeling for the prediction of the concentration and the temperature profiles of the MMA and the controller design of a based neural network in a batch reactor for the MMA production.

1.2 Scopes of Research

The scopes of this research are as follows.

1.2.1 The MMA production process for esterification of a methacrylic acid with methanol in a batch reactor is studied in this research.

1.2.2 A neural network forward model is applied to predict a concentration of methyl methacrylate and a reactor temperature, and a neural network inverse model to predict a set point of a jacket temperature. The neural network forward and inverse models have been developed based on Lenvenberg-Marquardt training algorithm.

1.2.3 The neural network base model predictive control (NNMPC) and the neural network direct inverse control (NNDIC) are developed to be a controller for the temperature control in the batch reactor of the production.

1.3 Contributions of Research

The main contributions of this research are as follows.

1.3.1 The neural network for the prediction temperature profile in a batch reactor of the MMA production process has been used to represent the process.

1.3.2 The NNDIC controller and NNMPC controller has been developed to control the MMA production process to a target value desired.

1.4 Methodology of Research

The methodologies of this research are as follows.

1.4.1 Literature review and plan research is studied.

1.4.2 Mathematical model of a MMA production and neural network modeling of the MMA production is created.

1.4.3 An optimal neural network forward model for a prediction of a concentration and temperature profiles, and an optimal neural network inverse model for a prediction of a set point of jacket temperature profile of the MMA production is determine.

1.4.4 A dynamic optimization is applied to find an optimal operating temperature to achieve maximizing the desired product.

1.4.5 NNDIC controller design to control temperature in the batch reactor is created.

1.4.6 NNMPC is created

1.4.7 All simulation results are collected and summarized.

1.4.8 Data analysis and writing a thesis is prepared

CHAPTER II

LITERATURE REVIEWS

This chapter presents literature reviews of a MMA production process, a neural network for modeling and a neural network based control.

2.1. Methyl methacrylate Production Process

Methyl methacrylate can be produced in different ways on C_2-C_4 hydrocarbon feed stocks. In the present review, the recent commercialized and expected MMA technologies will be described and a comparison of these production routes. While Rohm and Hass Co. began to industrially produce a methacrylic ester (ethyl methacrylate) first in 1933, ICI reformed the Rohm's method and commercialized MMA in 1937 by acetone cyanohydrine (ACH) process. Recently, MMA can be produced in many routes. Most of these new processes have been developed for an environmentally friendly production of MMA and thereby the development of catalysts was the key technology.

Koichi Nagai (2001) studied routes of a MMA production in different ways base on C_2, C_3 and C4 hydrocarbon and demonstrated the recent commercialized and expected MMA technologies. In addition, this paper presents the comparison between advantages and disadvantage of each production processes.

Hai-feng et al. (2006) presented intermetallic Pb-Pb catalysts to produce MMA base on direct oxidative esterification of methacrolein with methanol in a slurry reactor. The reaction was operated at 80 °C, 3.8% (w) of catalyst, 2 hour. Resulting, the selectivity and the yield of MMA were 90% and 76.5%, respectively.

Witczak et al. (2010) studied the esterification kinetics between methacrylic acid and methanol with heteropolyacids catalysts in a batch reactor. The reaction was operated at 313 K to 348 K and initial molar ratios of reactants from 3 to 10. In addition, this paper shows an effect of catalyst, temperature and molar ratio. Resulting, these catalysts can be used instead homogeneous catalysts.

2.2. Neural Network Modeling and Controller Design

Recently, neural networks have been successfully applied for controlling and modeling in many chemical processes which are complexity and non-linearity. Neural networks only require an input and an output data from plant and have the advantages of distributed information processing. In many cases, when sufficiently rich data are available, it can provide fairly accurate models for nonlinear controls when model equations are not known or only partial state information is available (Kittisupakorn, 2009).

Limpornchaijaroen (1996) demonstrated recurrent feedforward neural network for modeling of gravity flow tank process and continuous stirrer tank reactor. Neural network inverse models for process control are studied composing of adaptive neural network controller (1), adaptive neural network controller without error (2), adaptive neural network controller with error (3), nonlinear internal model controller (4) and simple feedback neural network controller (5). In addition, this paper presents the comparison of their controller performance of set point changing, load changing and parameter changing. In the simulation results for set point changing, the controllers number 2, 3, 4 and 5 have better performance than the PID controller. For load changing, the controllers number 4 and 5 have better performance than the PID controller. For parameter changing, the controllers number 1, 2, 3 and PID controller are overshoot less than controller number 4 and 5.

Nanmjaruskochakorn (1997) studied model temperature changing of a liquid steel in a BOF's process during tapping and adding some additive using neural networks. This study shows that an optimal architecture of neural network for process modeling consists of 11 nodes in an input layer, 4 nodes in a hidden layer and a node in an output layer. A learning rate and momentum values of the neural network prediction are 0.01 and 0.5, respectively. An error of the forecast model to the real values was found to be 7 ºC.

Somsiri (1997) applied neural network to determine temperature profile of a liquid steel in a BOF's process. In addition, this paper presents neural network parameters and process parameters which effect to a process temperature. The simulation results show that an optimal neural network architecture consist of 11 input nodes in an input layer, 4 nodes in a hidden layer and 1 node in an output layer. A learning rate and momentum of neural network prediction are 0.01 and 0.5, respectively.

Tangteerasunun (2004) proposed neural network models for a prediction of a concentration profile of hydrochloric acid in a pickling process that is complexity and highly nonlinear. The network is trained using backpropagation and Lenvenberg-Marquaedt techniques and MSE minimum technique is used to find an optimal neural network architecture. Resulting, the optimal neural network architecture of concentration of cation resin from 0 to 3,000 ppm, 3,000 to 6,000 ppm and concentration of anion resin from 0 to 2,000 ppm were [5-11-13-2], [5-8-9-2] and [5-13-13-2] respectively. The simulation result shows that the multilayer feedforward neural network models with two hidden layers provide sufficiently accurate prediction of the process.

Daosud (2005) investigated the use of a neural network direct inverse model-based control strategy (NNDIC) to control a steel pickling process. An optimal neural network architectures are determined by the mean squared error (MSE) minimization technique. The robustness of the proposed inverse model neural network control strategy is investigated with respect to change in disturbances, model mismatch and noise effects. Simulation results show the superiority of the NNDIC controller in the cases involving disturbance, model mismatch and noise while the conventional controller gives better results in the nominal case.

Saeyang (2005) described a neural network model to predict a solid percentage and viscosity of shampoo in a shampoo production tank. The neural network is trained using backpropagation and Lenvenberg-Marquardt algorithm. Accuracy evaluation of the receiving model is determined base on Root Mean Square Percent (RMSP) Error and Maximum Percent (MP) Error. The simulation results show that the multiplayer feed forward neural network model with 5 nodes in first hidden layer and 9 nodes in second hidden layer provides the best prediction of solid percentage and viscosity. The RMSP and MP of an optimal neural network structure for the prediction of solid percentage and viscosity are 9.71%, 8.83%, 15.73% and 42.83%, respectively.

Daosud (2006) proposed a neural network based model predictive control for multivariable system in a steel pickling process to control hydrochloric acid concentrations in acid baths. The neural network is trained using Lenvenberg-Marquardt algorithm. The achieving algorithm is tested to control in many cases consisting of set point changing, disturbance, model mismatch and presence of noise. In addition, this paper shows the comparison between two different control strategies consisting of neural network direct inverse control (NNDIC) and Dual mode (DM). Resulting, the DM control gives good control for the steel pickling process and removes the offset when compared to NNDIC and PI controller.

I. M. Mujtaba, N. Aziz and M. A. Hussain studied three different types of nonlinear control strategies composing of a generic model control, a direct inverse model control and a internal model control and implemented in batch reactors using neural networks techniques. In addition, a dynamic optimization problem with a simple model are solved a priori to obtain optimal operation policy in term of reactor temperature with an objective to maximize the desired product in a given batch time. The simulations show that all types of controller perform well the tracking the optimal temperature profile and achieving target conversion to the desired product.

Shomchom (2006) demonstrated an on-line optimal control with a neural network estimator of an ethanol production in a fed-batch reactor to modify an optimal feed profile. The neural network is applied to estimate unmeasured state variable. The simulation results show that the on-line optimal control with the neural network estimator gives a better performance than an off-line optimal control.

Neueaklong et al. (2011) presented a neural network modeling for a hard chrome electroplating process to predict a plating solution temperature in a hard chrome electroplating bath. The aim was to apply the neural network inverse model-based control (NNDIC) strategy for controlling plating temperature to desired temperature range. For the performance comparison between the NNDIC and a conventional PI control under nominal case and mismatch cases, it is found that the conventional PI controller gives better results than NNDIC in both cases.

2.3 Neural Network base Model Predictive Control

MPC is an advanced method of process control that has been in use in the process industries. The model used in MPC is generally intended to represent the behavior of complex dynamic systems. However, the performance for behavior prediction of MPC depended on model accuracy. Recently, neural network modeling is achieved to solve this problem for prediction the process outputs. Neural network based model predictive control (NNMPC) are used in many control configurations and can be applied in plant uncertainties (Kittisupakorn et al., 2009; Ławryńczuk, 2008; Damour et al.; 2010).

Wei et al. (2002) proposed MPC strategy based on a feedforward neural network model for polypropylene process. In order to infer on line product properties, a dynamic process model was developed. A recursive prediction error method was used to update the model parameters when there is a significant model prediction error. To obtain an optimal control strategy during grade transitions, a nonlinear MPC controller was applied based on a neural network model which is trained using an inputs and an outputs data of a process model. Performance of the nonlinear controller was compared with a conventional PID controller. The results indicate that the MPC controller can obtain satisfactory performance and consequently results in significant reduction in transition time and product variability.

Daosud (2006) presented a neural network based model predictive control strategy (NNMPC) for improvement of a steel pickling process. The controlled variables are the hydrochloric acid concentrations in the acid baths. In the training step for modeling, multiple-input single-output multilayer feedforward neural network models are developed using input-output data sets which obtain from mathematical model simulation. In the control algorithm, the neural network models are applied for prediction of the future process response in a model predictive control (MPC) algorithm to determine an optimal control actions using the successive quadratic programming (SQP). The algorithm is tested to control in the steel pickling process. The simulation results show that the NNMPC is better performance than conventional PI controller.

Kiran and Jana (2009) demonstrated cell growth and metabolite production greatly depending on the feeding of the nutrients in fed-batch fermentations. A strategy for controlling the glucose feed rate in fed-batch baker's yeast fermentation and a novel controller was studied. The different between the specific carbon dioxide evolution rate and oxygen uptake rate was applied as controller variable. The neural network based model predictive is developed to be a controller. The performance of the controller was evaluated by the set point tracking. The result shows that the controller is a good performance for control.

Kittisupakorn et al. (2009) developed a multi-layer feedforward neural network model based predictive control. In the acid baths three variables under controlled are the hydrochloric acid concentrations. In the modeling, multiple input, single output recurrent neural network subsystem models are developed using input–output data sets obtaining from mathematical model simulation. In the control (MPC) algorithm, the feedforward neural network models are used to predict the state variables over a prediction horizon within the model predictive control algorithm for searching the optimal control actions via sequential quadratic programming. The proposed algorithm is tested for control of a steel pickling process in several cases such as for a set point tracking, a disturbance, a model mismatch and presence of noise. The results for the neural network model predictive control (NNMPC) overall show better performance in the control of the system over the conventional PI controller in all cases.

Konakom et al. (2010) proposed neural network-based model predictive control (NNMPC) for definition optimal policy tracking and determined by dynamic optimization of a batch reactive distillation column. Multi-layer feedforward neural network model and estimator are developed and used in the model predictive control algorithm. The simulation results show that the NNMPC provides satisfactory control performance for set point tracking problems. The robustness of the NNMPC is investigated with respect to plant/model mismatches and is compared with a conventional proportional controller (P). It has been found that the NNMPC provides better control performance than the P controller does in all cases.

CHAPTER III

THEORY

3.1 Methyl methacrylate

Methyl methacrylate (CAS No. 80-62-6) is a clear liquid state, colorless, volatile and flammable liquid with ester-like odors. Its vapor is irritating to the eyes, nose and throat. Fire should be extinguished with carbon dioxide, dry chemical or foam. The general property following the table 3.1

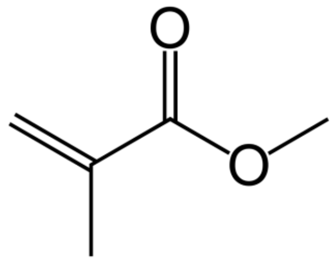


Figure 3.1 Molecular structure of MMA

Table 3.1 MMA property

Property	MMA
Molecular formula	$C_5H_8O_2$
Molecular weight	100.12 g/mol
Appearance	Colorless liquid
Density	0.94 g/cm^3
Melting point	-48 °C
Boiling point	101 °C
Water solubility	1.5g/100ml (25 °C)
Viscosity	0.6 cP at 20 °C

The most of MMA is polymerized to produce homopolymers and copolymer with the largest application being the casting, molding or extrusion of polymethyl methacrylate (PMMA) or modified polymers. A major application of MMA polymers and copolymers is in surface coatings and impregnation resins to give color fastness and weather-resistance properties to latex paints and lacquer resins.

3.2 Methyl methacrylate Manufacturing

The present and proposed manufacturing routes for MMA based on natural gas or crude oil raw materials. It is convenient to further categorize these various manufacturing routes according to the specific raw materials. Specifically, Figures 3.2 shows outline general routes based on propylene, ethylene and isobutane/isobutylene. Methanol is a common raw material for all process to produce MMA. The commercial viability of a process is determined by the aggregate of raw material cost and utilization, operating costs with particular attention to energy related charges, waste disposal costs, environmental impact and plant capital investment.

3.2.1 C_2 routes

Ethylene can be used to produce MMA via propionaldehyde, propionic acid or methyl propionate. The key step of this route is the condensation reaction of such an intermediate with formaldehyde to make methacrolein, methacrylic acid and MMA.

1) BASF's process: In this route, MMA is made from ethylene via propionaldehyde, methacrolein, methacrylic acid as follows:

$$CH_2{=}CH_2 + CO + H_2 \longrightarrow CH_3CH_2CHO \quad (3.1)$$

$$CH3CH2CHO + HCHO \xrightarrow{(CH_3)_2NH} CH2{=}CH(CH3)\text{-}CHO{+}H2O \quad (3.2)$$

$$CH2{=}CH(CH3)\text{-}CHO \xrightarrow{O_2} CH2\text{-}C(CH3)\text{-}COOH \quad (3.3)$$

$$CH2\text{-}C(CH3)\text{-}COOH \xrightarrow{CH_3OH} MMA \quad (3.4)$$

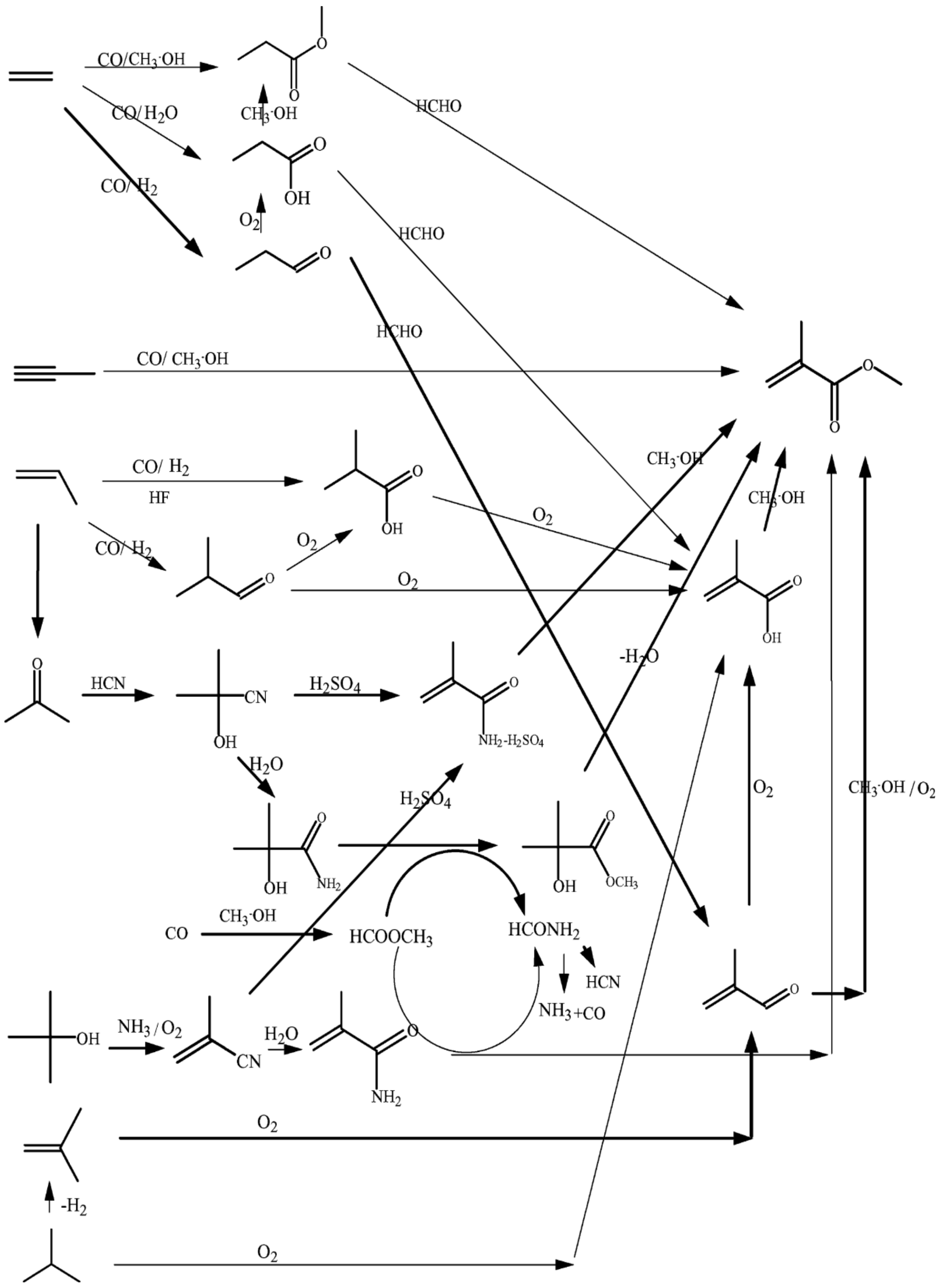


Figure 3.2 Routes to MMA

2) Other C_2 route

Propionic acid or methyl propionate process have been examined in many enterprises for a long time. Propionic acid is produced by hydrocarbonylation reaction of ethylene for which metal carbonyl is used as a catalyst, or by oxidation of propionaldehyde and methacrylic acid. If one of these methods would be industrialized, it could become a simple and excellent process because of its fewer steps.

$$CH3CH2COOH+HCHO \longrightarrow CH2=C(CH3)\text{-}COOH + H2O \quad (3.5)$$

$$CH3CH2COOCH3+HCHO \longrightarrow CH2=C(CH3)\text{-}COOCH3+H2O \quad (3.6)$$

3.2.2. C_3 routes

1) New Acetone Cyanohydrins (ACH) process

The conventional ACH method is based on the C3 route. In this process, ACH is hydrated to α-hydroxy isobutylamide with the manganese oxide catalyst in absence of sulfuric acid and then is esterified by methylformate to provide methyl-hydroxy isobutylate. Methylformate becomes formamide at this time. After that, the formamide is dehydrated to give hydrogen cyanide (HCN) which is recycled in the ACH preparation process. Methyl α-hydroxy isobutylate is converted to MMA in dehydration reactions.

$$(CH_3)_2\text{–}C(OH)CN+H_2O \longrightarrow (CH_3)_2\text{–}C(OH)CONH_2 \quad (3.7)$$

$$(CH3)2\text{–}C(OH)CONH2+HCOOCH3 \rightarrow (CH3)2C(OH)COOCH3+HCONH2 \quad (3.8)$$

$$(CH3)2\text{–}C(OH)COOCH \longrightarrow CH2=C(CH3)COOCH3 + H2O \quad (3.9)$$

$$HCONH2 \rightarrow HCN + H2O \quad (3.10)$$

2) Isobutyric acid process

In this route, methacrylic acid is made from propylene and isobutyric acid and this route is researched by Atochem and Rohm. Propylene, carbon monoxide, water and large quantities of hydrogen fluoride are mixed at low temperature. After that, the methyl methacrylate is produced by the oxidation of isobutyric acid. Main problems of this route are dangerous of hydrogen fluoride and corrosion.

$$CH_2{=}CHCH_3 + CO + H_2O \xrightarrow{HF} (CH_3)_2{-}CHCOOH \qquad (3.11)$$

$$(CH3)2{-}CHCOOH + \frac{1}{2} O2 \rightarrow CH2{=}C(CH3)COOH + H2O \qquad (3.12)$$

3.2.3 C_4 routes

1) Escambia process

Isobutylene is oxidized to provide α-hydroxy isobutyric acid using N_2O_4 and nitric acid at a low temperature of 5–10 ºC in liquid phase. After esterification and dehydration, MMA is obtained. However, this process does not seem to be economical by favorable because the yield is not sufficient and there are some problems such that large amounts of nitric acid and NO_x should be handled.

2) Methacrylonitrile (MAN) process

Asahi Chemical Co. developed and commercialized a route via methacrylonitrile (MAN) for MMA production.

$$(CH_3)_3C{-}OH + NH_3 + 32O_2 \rightarrow CH_2{=}C(CH_3)CN + 4H_2O \qquad (3.13)$$

$$CH2{=}C(CH3)CN + H2SO4 + H2O \rightarrow CH2{=}C(CH3){-}CONH2{\cdot}H2SO4 \qquad (3.14)$$

$$CH2{=}C(CH3){-}CONH{\cdot}H2SO4 + CH3OH \rightarrow CH2{=}C(CH3)COOCH3 + NH4HSO4 \qquad (3.15)$$

MAN process can be produced by ammoxidation almost in the same step as acrylonitrile which is produced in large quantities. Afterwards, MAN process has been hydrated by sulflic acid to methacrylamide. The same process as in conventional ACH was adopted because the yield of MAN process can be equal to that of acrylonitrile and the total yield is superior to the above-described two-step oxidation method.

3) Direct oxidative esterification process

Asahi Chemical established and started in 1998 a new process based on direct oxidative esterification of methacrolein which does not produce by-products such as ammonium bisulfate. The raw material is the same tert-butanol as in the direct oxidation method. In contrast with the direct oxidation process, this direct oxidative esterification process has only two steps. Methacrolein is produced in the same way as in the direct oxidation process by gas phase catalytic oxidation, is simultaneously oxidized and is esterified in liquid methanol to get MMA directly.

$$CH2{=}C(CH_3){-}CHO + CH_3OH + 12O_2 \rightarrow CH_2{=}C(CH_3){-}COOCH_3 + H_2O \quad (3.16)$$

4) C_4 direct oxidation process

The reactions by the direct oxidation method consist of two-step: Firstly, the oxidation of isobutylene or TBA with air to produce methaacrolein and then oxidation of methaacrolein to produce methacrylic acid. Secondly, the esterification by methanol to produce MMA

$$CH2{=}C{-}(CH3)2 \text{ (or } (CH3)3C{-}OH)+O2 \rightarrow CH2{=}C(CH3){-}CHO+H2O \quad (3.17)$$

$$CH2{=}C(CH3){-}CHO+\frac{1}{2}O2 \rightarrow CH2{=}C(CH3){-}COOH \quad (3.18)$$

$$CH2{=}C(CH3){-}COOH+CH3OH \rightarrow CH2{=}C(CH3){-}COOCH3+H2O \quad (3.19)$$

In this research, MMA is produced by esterification between methacrylic acid and methanol with dodecatungstophoric acid catalyst and reaction carried out in the liquid phase as follow:

CH2=C(CH3)-COOH + CH3OH $\Leftrightarrow$ CH2=C(CH3)-COOCH3+H2O (3.22)

Esterification is a chemical reaction used for making esters. The reaction in which a caboxylic acid combines with an alcohol in the presence of a catalyst (commonly concentrated sulphuric acid) to form an ester is called Esterificatin reaction. It is reversible reaction. The esters so formed are fruity in odors. They are sweet smelling compound.

3.3 The Mathematical Model for Methyl Methacrylate Process

The mathematical model for esterification reaction of a methyl methacrylate in a batch reactor is studied to describe in this simulation work.

CH2=C(CH3)-COOH + CH3OH $\Leftrightarrow$ CH2=C(CH3)-COOCH3+H2O (3.23)
(A) (B) (C) (D)

The kinetic equation of the esterification reaction of methacrylic acid with methanol obtained from Witczak et al. (2010) is assumed reversible reaction occur in liquid phase as show below:

$$R_A = k_1 C_{cat} \frac{C_A C_B}{C_C^{0.3}} - k_2 C_{cat} C_C^{0.7} C_D \quad (3.24)$$

where C_A, C_B, C_C, C_D and C_{cat} refer to concentration of methacrylic acid, methanol, methyl methacrylate, water and catalyst in mol/m^3, respectively.

The reaction rate constants (k_1, k_2) at several temperatures determined based on the basis of the experimental data, were used to establish the activation energy (E_a)

and the frequency factor (k_0). To do so, an Arrhenius-type temperature dependence of the reaction constants was used.

$$k = k_0 \exp\left(\frac{E_a}{RT_r} \right) \tag{3.25}$$

where R is the gas constant.

From Witczak et al. (2010), the reaction rate constants for this reaction are described by the following equations.

$$k_1 = 960 \exp\left(-\frac{63600}{RT_r}\right) \tag{3.26}$$

$$k_2 = 600 \exp\left(-\frac{67320}{RT_r}\right) \tag{3.27}$$

3.3.1 Mass balance

The mass balance describes the change of each components in a batch reactor which is expressed as follows:

$$\frac{dC_A}{dt} = -R_A \tag{3.28}$$

$$\frac{dC_B}{dt} = -R_A \tag{3.29}$$

$$\frac{dC_C}{dt} = R_A \tag{3.30}$$

$$\frac{dC_D}{dt} = R_A \tag{3.31}$$

3.3.2 Energy balance

The energy balance describes the change of temperature in a reactor and a jacket. Assuming the amount of heat retained in the walls of the rest of the system, an energy balance around the reactor give the following model.

$$\frac{dTr}{dt} = \frac{Qr + Qj}{V\rho Cpr} \tag{3.32}$$

$$\frac{dT_j}{dt} = \frac{F_j p_j Cp_r (T_{jsp} - T_j) - Q_j}{V_j p_j Cp_j} \tag{3.33}$$

$$Q_r = -\Delta H R_A V \tag{3.34}$$

$$Q_j = UA(T_j - T_r) \tag{3.35}$$

where Q, ΔH and U are heat released from reaction (kJ/min), heat of reaction (kJ/kmol) and heat transfer coefficient (kJ/(min m2 °C)), respectively.

3.4 Neural Networks Introduction

The original inspiration for the term Artificial Neural Network came from examination of central nervous systems and their neurons, axons (sends signals), dendrites (receives signals) and synapses (connects an axon to a dendrite) which constitute the processing elements of biological neural networks investigated by neuroscience. Axons are signals sender to dendrites which are signals receiver and the axons and dendrites are connected by synapses. In an artificial neural network simple artificial nodes are called variously such as neurons, processing elements (PEs) or units. These neurons are interconnected to each other in complex arrangements to transmit the information between the brain and receptors. Figure 3.3 shows a schematic sketch of the natural sets of neurons that consist of axons, dendrites, synapses and soma (cell body).

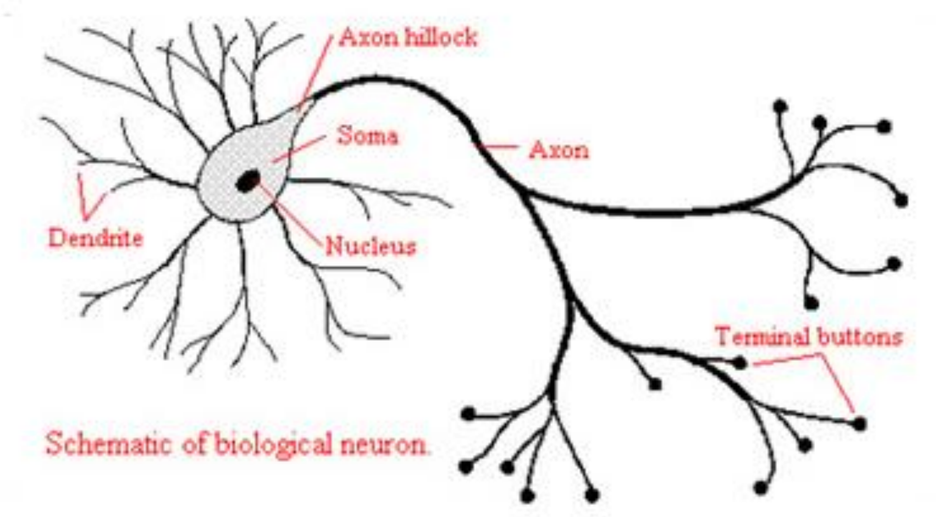


Figure 3.3 Axons, dendrites and synapse in a biological neuron

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.

3.5 Components of Neural Networks

The neural networks consist of many interconnected neurons or nodes and many functions for calculation of neural networks outputs. In each node, there are many components that are used to build a neural network. These components are described as the follow:

3.5.1 Summation function

Summation function, which is mathematical mapping with function $u(w,x)$ where w, x refer to metric weight and input vector, respectively, is used to combine inputs signal from each nodes. The simplistic summation function is found by multiplying each components of x vector by the corresponding component of w vector and then adding up all the products. Summation function is applied to transform those inputs signal to next function that is transfer function. The summation equation as show below:

$$u_i(w,x) = \sum_{j=1}^{n} w_{ij} x_{ji} \tag{3.36}$$

where x and w are the column vector of n inputs and row vector of n weights as follow:

$$x = [x_1, x_2, x_3, \ldots, x_n]^T \tag{3.37}$$

$$w = [w_{i,1}, w_{i,2}, w_{i,3}, \ldots, w_{i,n}] \tag{3.38}$$

3.5.2 Transfer function

Transfer function or activation function transformed summation or net that receive from outputs of summation function to neural network outputs. 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 outputs. The transfer functions as show below:

1) Linear transfer function

The linear transfer function is utilized in the output layer for output expansion purpose. The result that receives from this transfer function is a linear and the calculation can be expressed as equation (3.40) and figure 3.4.

$$f(\text{net}) = \begin{cases} 1, \textit{if}\ \text{net} \geq 1 \\ net \quad \textit{if}\ -1 < net < 1 \\ -1, \textit{if}\ \text{net} \leq -1 \end{cases} \tag{3.39}$$

2) Log-Sigmoid transfer function

The Log-Sigmoid transfer function is a subset of nonlinear transfer functions. This transfer function will convert high positive value into 1 and converted high negative value into 0. The calculations of the transfer function as show in equation (3.41) and figure 3.5.

$$f(\text{net}) = \frac{1}{1+e^{-net}} \tag{3.40}$$

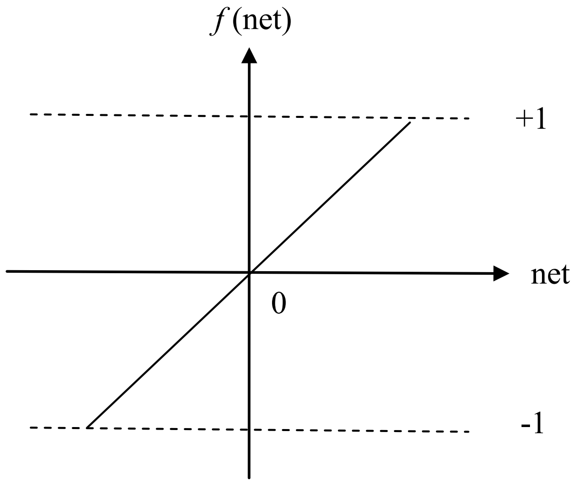


Figure 3.4 Linear transfer function

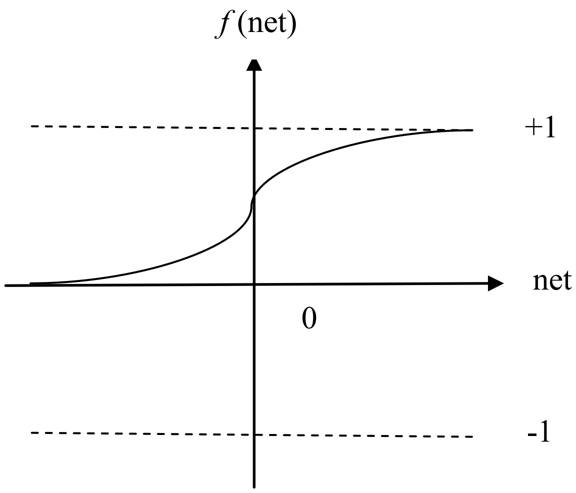


Figure 3.5 Log-Sigmoid transfer function

3) Tan-Sigmoid transfer function

One of nonlinear transfer functions is Tan-Sigmoid transfer function. The transfer function will transform high positive value into 1 and converted high negative value into -1. The equation and figure are shown below:

$$f(\text{net}) = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}} \tag{3.41}$$

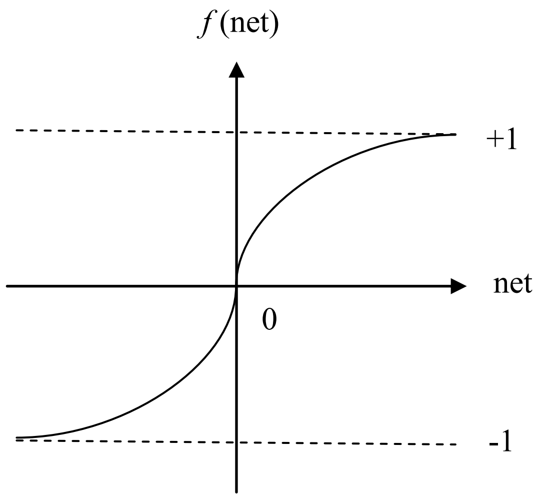


Figure 3.6 Tan-Sigmoid transfer function

3.5.3 Error function

An objective of a network training is to minimize difference values between target values and network output values until the difference values less than expect value. The difference values are transformed to an error function that has an effect to the network structure. Type of error function are given by

1) Sum square error

This error function is combination of square difference between network output values y and target values p.

$$SSE = \sum_{i=1}^{N} (y_i - p_i)^2 \tag{3.42}$$

2) Mean square error

This error function measures the average of the square error. The error is the amount of quantity difference between network output values y and target values p and N refer to numbers of data.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - p_i)^2 \tag{3.43}$$

3.5.4 Learning Function

The purpose of the learning function is adjusting of the connection weight values that connect the inputs of each processing elements to achieve the desired results. Information is stored and distributed throughout the network via the interconnection weights. There are classified into two types:

Table 3.2 Learning functions

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

1) Supervised learning

A supervised learning requires a teacher for determination network outputs that accord with teaching data. In the learning network, a training process consists of the input and output data. During the training, the neural network output is compared to the teaching data and the weight values are adjusted for the network outputs according with teaching data that is target value. This learning algorithm tries to minimize the difference between teaching data and the network output. This different values are called error. This error reduction is created over time by continuously modifying the input weights until acceptable network accuracy is reached.

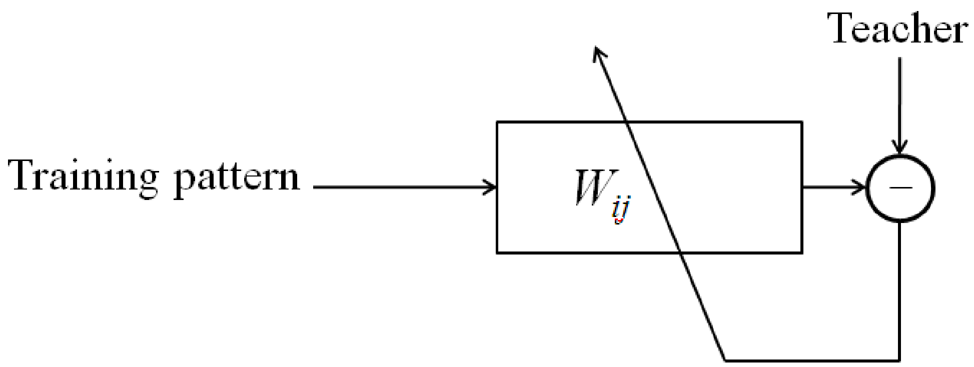


Figure 3.7 Supervised learning

2) Unsupervised learning

An unsupervised learning or self-supervised learning does not require a teacher for determination network outputs. The network determines using only inputs data. Sometimes this learning algorithm is called self-supervised learning. These networks look for regularity or trend in the input signals and make adaptations according to the function of the network.

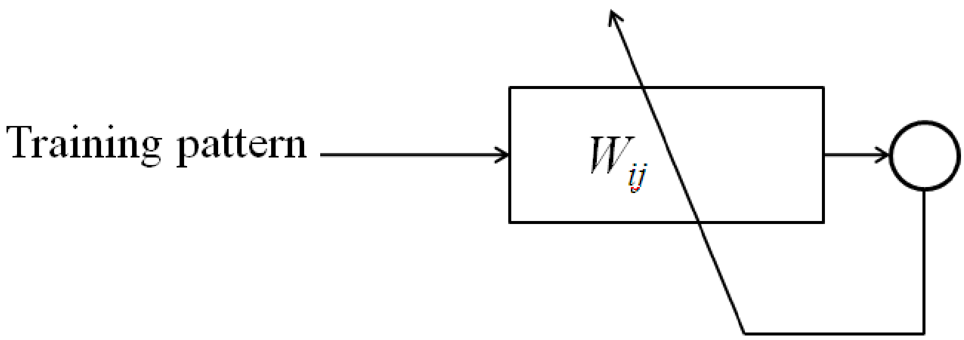


Figure 3.8 Unsupervised learning

3.6 Neural Networks Architecture

3.6.1 Network Structure

1) Feedforward neural networks allow the signals travel from an input to an output one way only. There is no feedback in the network such as the output of any layer. Feedforward neural networks tend to be straight forward networks that associate inputs with outputs.

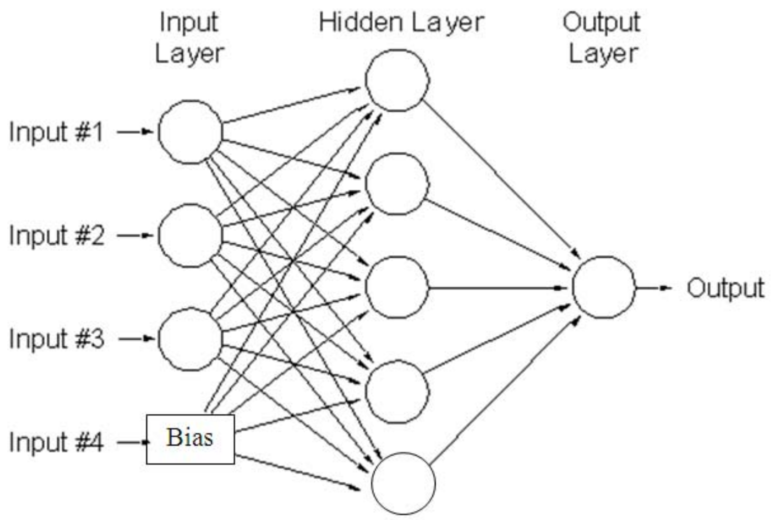


Figure 3.9 Feedforward neural networks

2) Feedback neural 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. These remain at the equilibrium point until the change and a new equilibrium need to be found.

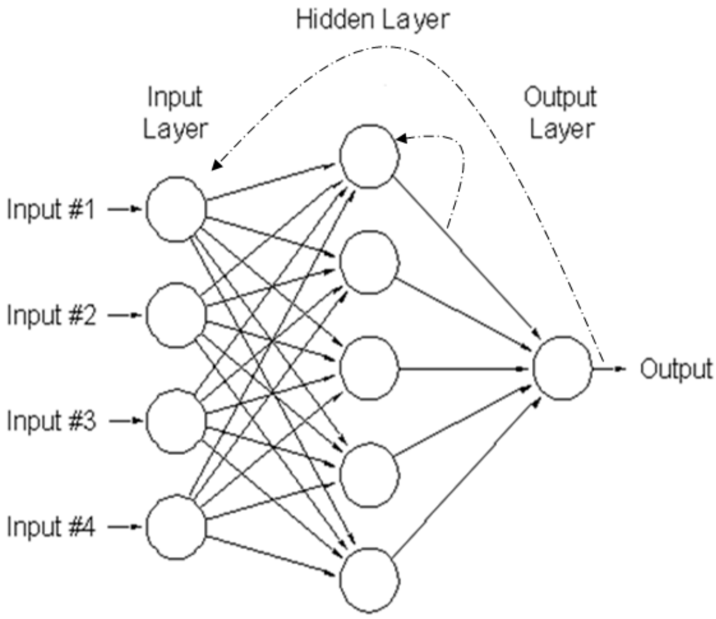


Figure 3.10 Feedback neural networks

3.6.2 Network Layers

The number of operating nodes in parallel are called layer. A layer of a neural network consists of the input vector p, a weight matrix w, the summation units, the bias vector b, the transfer function units and the output vector a. Each elements of the input vector is connected to each nodes in each layers though the weight matrix. The simply neural network structures compose of a layer that is an output layer. In a single-layer neural network, the structure does not have a hidden layer and shows in figure 3.9. The multi-layer neural network structure consists of hidden layer(s) and output layer and shows in figure 3.10. The input vector represents the raw information that is fed into the network. A hidden layer is between an input and an output layer. An 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

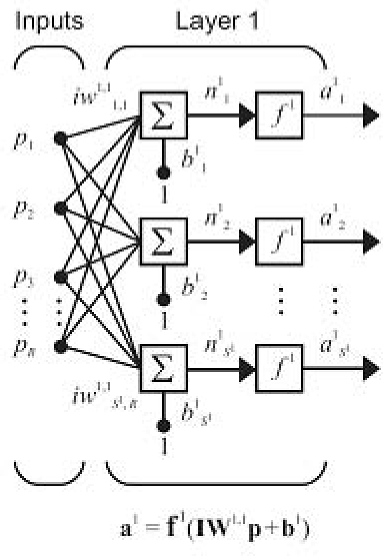


Figure 3.11 Single-layers neural networks

The matrix p, which is previously defined, consists of individual inputs p_1, p_2, p_3,…, p_R. There inputs are connected to each nodes though the weight matrix w which is now becoming an S×R matrix as defined below:

$$w = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{pmatrix}_{S\times R} \quad (3.44)$$

$$b = \left[\begin{bmatrix} b_1 & b_2 & \cdots & b_S \end{bmatrix}^T\right]_{S\times 1} \quad (3.45)$$

$$a = \left[\begin{bmatrix} a_1 & a_2 & \cdots & a_S \end{bmatrix}^T\right]_{S\times 1} \quad (3.46)$$

The network output can be expressed as follows:

$$a_{S\times 1} = f\left(\left[w_{S\times R} * p_{R\times 1}\right] + b_{S\times 1}\right) \quad (3.47)$$

Now consider a network with several layers. Each layer has its own weight matrix, its own bias vector, a net input vector and an output vector. Figure 3.12 introduces some additional notation to distinguish between these layers. This figure will use superscripts to identify the layers. Specifically, this figure is appended the number of the layer as a superscript to the names for each of these variables. Thus, the weight matrix for the first layer is written as w^1, and the weight matrix for the second layer is written as w^2. This notation is used in the three-layer network.

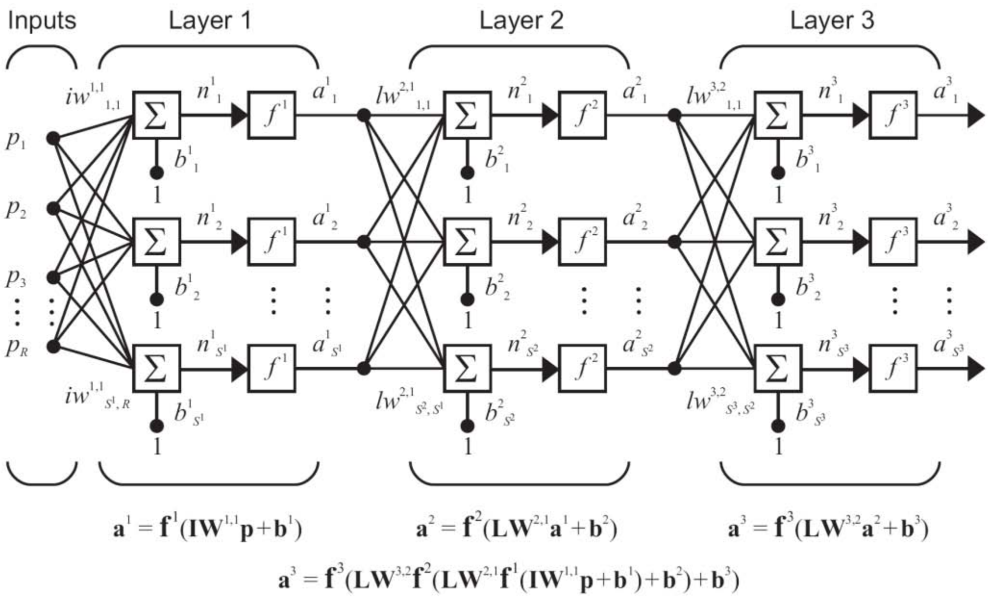


Figure 3.12 Multi-layers neural networks

As shown, there are R inputs, a^1 nodes in the first layer, a^2 nodes in the second layer, etc. As noted, different layers can have different numbers of nodes. The outputs of layers one and two are the inputs for layers two and three. A layer whose output is the network output that is called an output layer. The other layers are called hidden layers. The multi-layer neural network in Figure 3.12 has an output layer (layer 3) and two hidden layers (layers 1 and 2).

3.7 Training Algorithm

The aim of network training is to minimize error between targets and network outputs. Training is procedure to determine optimal values of the connection weights and biases. Training is begun by initially assigning arbitrary small random values to the weights. This proceeding is iterated until a satisfactory model is obtained. In this work, we concerned the feedforward neural network which utilizes the supervised training method.

3.7.1 Back propagation Algorithm (Zilouchian and Jamshidi, 2001)

Back propagation algorithm is one of the most popular algorithms for training a network due to its success from both simplicity and applicability viewpoints. The algorithm consists of two phases: Training phase. In the training phase, first, the weights of the network are randomly initialized. Then, the output of the network is calculated and compared to the desired value. In sequel, an error of the network is calculated and used to adjust the weights of the output layer. In a similar fashion, the network error is also propagated backward and used to update the weights of the previous layers. Figure 3.13 shows how the error values are generated and propagated for weights adjustments of the network. In the recall phase, only the feedforward computations using assigned weights from the training phase and input patterns take place. Figure 3.12 shows both the feedforward and back propagation paths. The feedforward process is used in both recall and training phases. On the other hand, as shown in Figure 3.12 (b), back propagation of error is only utilized in the training phase. In the training phase, the weight matrix is first randomly initialized. After that, the output of each layers are calculated starting from the input layer and moving forward toward the output layer. Thereafter, the error at the output layer is calculated by comparison between actual output and the desired value to update the weights of the output and hidden layers.

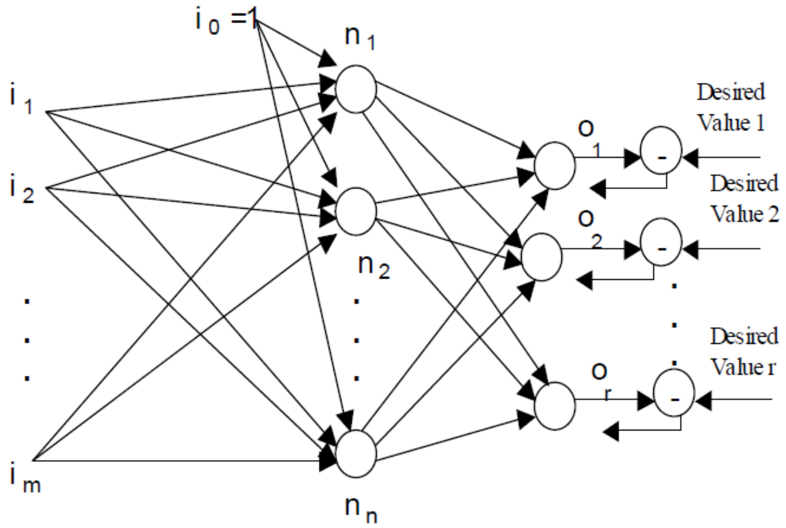
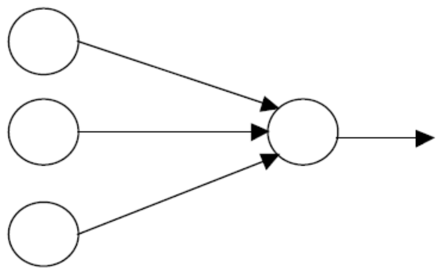
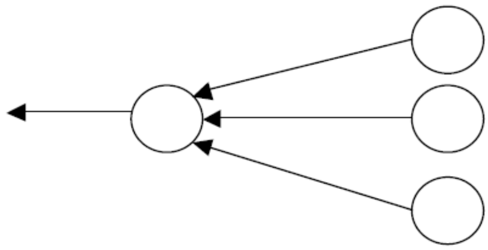


Figure 3.13 Back Propagation of the Error in a Two-Layer Network



a) Forward propagation



b) Backward propagation

Figure 3.14 Forward Propagation and Backward Propagation in Training Phase

There are two different methods of updating the weights. In the first method, weights are updated for each input patterns using an iteration method. In the second method, an overall error for all the input output patterns of training sets is calculated. In other words, either each of the input patterns or all of the patterns together can be used for updating the weights. The training phase will be terminated when the error value is less than the minimum set value provided by the designer. For the disadvantages of back propagation algorithm, the training phase is very time consuming. During the recall phase, the network with the final weights resulting from the training process is employed. Therefore, for every inputs pattern in this phase, the output will be calculated using both linear calculation and nonlinear activation functions. The process provides a very fast performance of the network in the recall phase. There is one of its important advantages. The methodology of the conventional backpropagation method is mentioned below (Hussain, 1994):

Inputs are summed and propagated to the hidden layer for a node *j* as:

$$net_j = \sum_{i=1}^{Ni} W_{ij} p_i^1 + b_j \tag{3.48}$$

Output from node j is given by

$$a_j^2 = f(net_j) \tag{3.49}$$

where f is the transfer function or activation function used in the hidden nodes

Hidden layer output is propagated to node k at the output layer given as:

$$net_k = \sum_{j=1}^{N_j} W_{kj} a_j^2 + b_k \tag{3.50}$$

Output from the node *k* is:

$$a_k^3 = f(net_k) \tag{3.51}$$

Error is calculated at the output layer as:

$$e = \frac{1}{2}\sum_{k=1}^{N_k}(p_k - a_k^3)^2 \tag{3.52}$$

Weights are adjusted along the negative gradient descent of the error given as:

$$\Delta w_{kj} = -\eta \frac{\partial e}{\partial w_{kj}} \tag{3.53}$$

Weights in the output and the hidden layers are then corrected using equations below:

$$\Delta w_{ji} = \eta a_j^2 (1 - a_j^2) a_i^1 \sum_{k=1}^{N_k} \delta_k^3 w_{kj} \tag{3.54}$$

and $$\Delta w_{kj} = \eta (p_k - a_k^3) a_k^3 (1 - a_k^3) a_j^2 \tag{3.55}$$

The constant η (called the learning rate, and nominally equal to one) is put in to speed up or slow down the learning if required.

The gradient descent is simply the technique where parameters, such as weights and biases, are moved in the opposite direction to the error gradient. Each step down, the gradient results in smaller errors until an error minimum is reached. The network can get a better performance using an approximation of Newton's method called Levenberg-Marquardt. This technique is more powerful than the gradient descent, but also requires more memory.

3.7.2 Levenberg-Marquardt method

Like the quasi-Newton methods, the Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feedforward networks), then the Hessian matrix can be approximated as

$$H = J^T J \tag{3.56}$$

and the gradient can be computed as

$$g = J^T e \tag{3.57}$$

where J is the Jacobian matrix that contains 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 back-propagation technique that is much less complex than computing the Hessian matrix.

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

$$w_{k+1} = w_k - [J^T J + \mu I]^{-1} J^T e \tag{3.58}$$

When the scalar μ is zero, this is just 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 toward 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 is always reduced at each iterations of the algorithms.

3.8 Model Predictive Control

MPC is a widely used meaning to deal with large multivariable constrained control issues in an industry. The main aim of a MPC is to minimize a performance criterion in the future that would possibly be subject to constraints on the manipulated inputs and outputs, where the future behavior is computed according to a model of the plant. MPC is a set of algorithms based on the models. MPC pays more attention to the function, than to the formulation, of the model. The function of a prediction model is based on the past information and the future inputs to predict the future output. Any collection of information, as long as it has the function of prediction, irrespective of the concrete form, can be the prediction model. (Seborg et al., 2004)

The basic ideas of model predictive control are follows:

1. Explicit use of a model to predict the process output at future time instants.
2. Calculation of a control sequence minimizing an objective function.
3. Receding strategy, so that at each instant the horizon is displaced towards the future that involves the application of the first control signal of the sequence calculated at each step.

The methodology of all the controllers belonging to the MPC family is characterized by the following strategy, represented in figure 3.15:

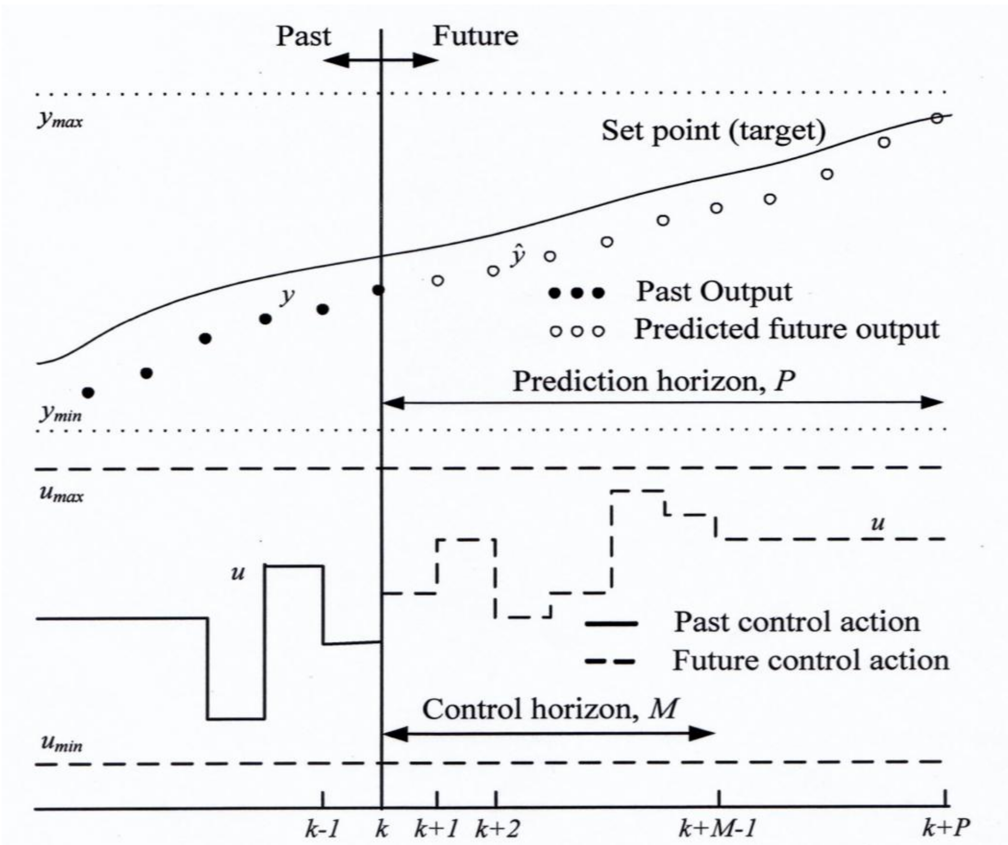


Figure 3.15 MPC Strategy

The set points for the control calculations, also called target, are calculated from an economic optimization based on a stead state model of the process, traditionally, a linear model. The control calculations are based on current measurements and predictions of the future values of the outputs. The predictions are made using a dynamics model, typically a linear empirical model such as a multivariable version of the step response of difference equation models. Alternatively, transfer function or state space models can be employed. For very nonlinear processes, both physical models and empirical models, such as neural networks, have been used in nonlinear MPC.

The objective of the MPC control calculations is to determine the sequence of control moves so that the predicted response moves to the set point in an optimal manner. The actual output y, predicted output y_p and manipulated input u are shown in figure 3.15. At the current sampling instant, denoted by *k*, the MPC strategy calculated the set of *M* values of the input *{u(k+i-1), I = 1,2,…,M}*. The set consist of the current input *u(k)* and *M-1* future in inputs. The input is held constant after the *M* control moves. The inputs are calculated so that a set of *P* predicted outputs $\{y_p(k+i),\ i = 1,2,\ldots,P\}$ reaches the set point in an optimal manner. The control calculations are based on optimizing an objective function. The number of predictions *P* is referred to as the prediction horizon while the number of control moves *M* is called the control horizon.

The idea of model predictive control is to utilize a model of the process in order to predict and optimize the future system behavior. The model form can be described by the following.

$$\dot{X} = f(X(t), U(t)) \tag{3.59}$$

The control law of the model predictive control is determined from the minimization of the controlled variable and manipulated variable. The optimization problem is as follows. (Kittisupakorn, 2008)

Objective function : $$\min \int_0^{t_f} (W_1 (X - X_{sp})^2 + W_2 (\Delta U)^2) dt \tag{3.60}$$

State space model : $\dot{X} = f(X(t), U(t))$ (3.61)

Manipulated variable constraint : $U_{\min} \leq U \leq U_{\max}$ (3.62)

State variable constraint : $X_{\min} \leq X \leq X_{\max}$ (3.63)

Terminal state constrain : $X(t + t_f) = X_{sp}$ (3.64)

where W_1 and W2 are the weighting factors on the controlled and manipulated variables, respectively, t_f is the terminal time, U_{min} and U_{max} are the minimum and maximum bounds of manipulated variables and X_{min} and X_{max} are the minimum and maximum bounds of state variables.

A distinguishing feature of MPC is its receding horizon approach. Although a sequence of *M* control moves is calculated at each sampling instant, only the first move is actually implemented. Then a new sequence is calculated at the next sampling instant, after new measurements become available; only the first input move is implemented. This procedure is repeated at each sampling instant.

3.9 Optimization

Optimization refers to the choosing of the best element from some set of available alternatives. This technique is one of the major quantitative tools in industrial decision making. A wide variety of problem in the design, construction, operation, and analysis of chemical plants (as well as many other industrial processes) can be resolved by optimization (Edgar et al., 2001). The objective of dynamic optimization problem is to compute an optimize condition for production processes.

The key elements of an optimization are

1. Objective Function
2. Constrains
3. Decision Variable

An objective function, which refers to equations created to determine the best values of decision variables (maximum or minimum values), is a mathematical function. There may be more than one objective function for a given optimization problem. There are different types of objective function depending on the needs and uses.

Constraints are values definition of feasible region of the process. There are classified into two types as follow:

1. Equality constraints are constraints that indicate the limits of the process or its product such as the purity of the products, mass and energy balance.

2. Inequality constraints are constraints that indicate the limit due to design and other limits

Decision variable refers to the change of parameters which effect to objective function. In optimization method, decision variable is changed to determine maximum or minimum of objective function and is used set point in process control system.

The optimization models represent problem choices as decision variables and seek values that maximize or minimize objective functions of the decision variables subject to constraints on variable values expressing the limits on possible decision choices. The optimization model description is stated as:

max/min $f(\mathrm{x})$ objective function

Subject to: $h(x) = 0$ equality constraints (3.65)

$g(x) \geq 0$ inequality constraints (3.66)

where x is a vector of n decision variables ($x_1, x_2, \ldots, x_n$),

h(x) is a vector of equations of dimension

g(x) is a vector of inequalities of dimension

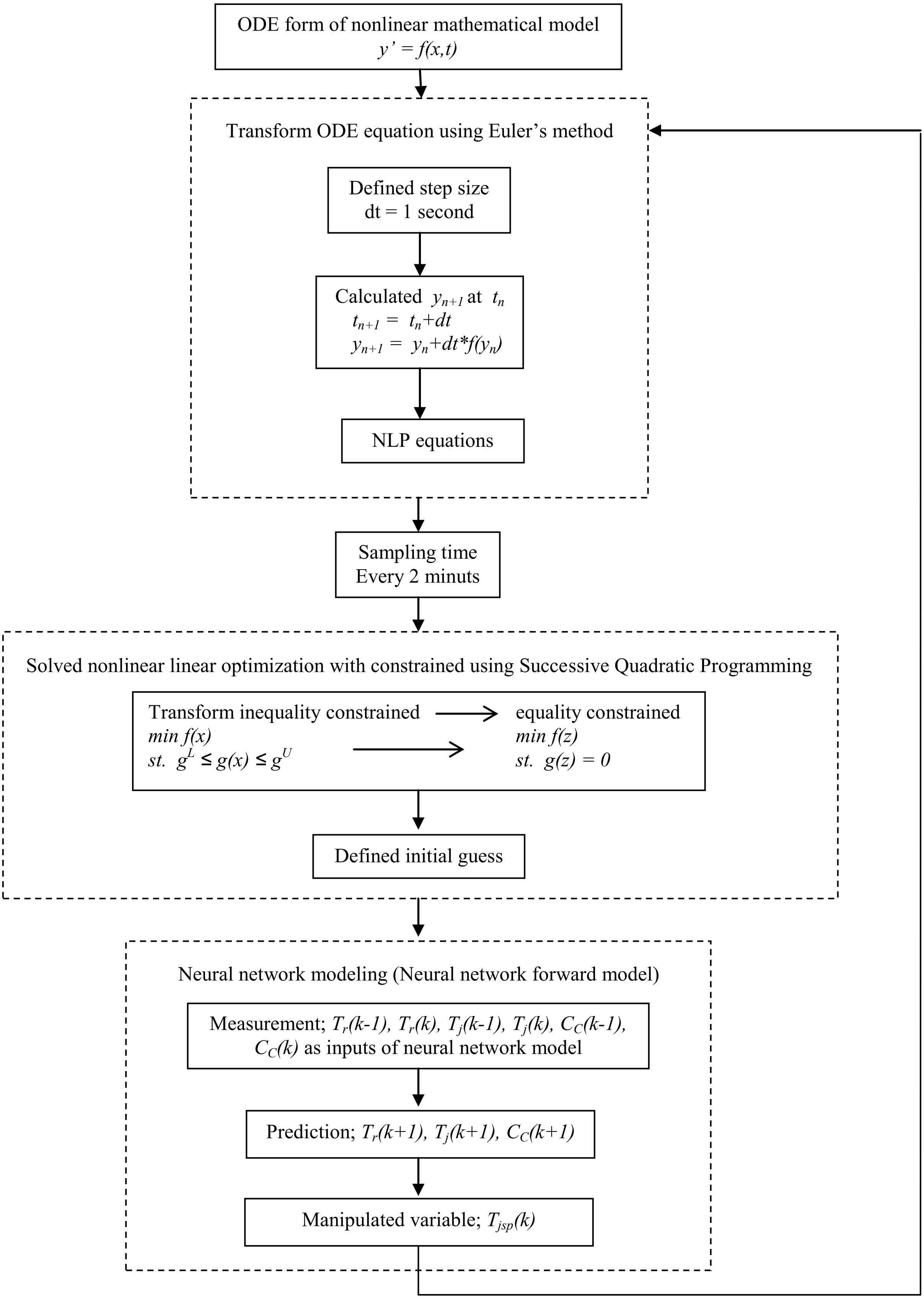


Figure 3.16 Neural network model predictive control flowchart

CHAPTER IV

NEURAL NETWORK FORWARD MODEL AND NEURAL NETWORK INVERSE MODEL FOR THE PROCESS

4.1 Neural network modeling

In this part, a neural network model is applied to predict a concentration of MMA and a temperature of the process. The training, testing and validating data sets for a neural network modeling are generated from the obtained mathematical model. In the generating data for the network training, the network is trained in many possible scenarios consisting of plant certainty and plant uncertainty cases composing of rate of reaction, heat of reaction and overall heat transfer coefficient. For the setting data to train, the manipulated variable is changed in the range 260-380 K as step change.

The generated data requires normalization for achieving a good performance neural network modeling. In the normalization step, all data were scaled in range of minimum and maximum value. The minimum and maximum data were compared to 0.05 and 0.95, respectively. An equations for data normalization and converting back can be expressed as follows:

$$x_{nor} = \frac{(x - x_{\min})(0.95 - 0.05)}{(x_{\max} - x_{\min})} + 0.05 \tag{4.1}$$

$$x = \frac{(x_{nor} - 0.05)(x_{\max} - x_{\min})}{(0.95 - 0.05)} + x_{\min} \tag{4.2}$$

where x and x_{nor} are the actual and normalized values, respectively.

The data were normalized between 0.05 and 0.95 because the output of network is calculated from sigmoid activation function. Therefore, data in each groups of variable were scaled according to the corresponding data range. In the training step, the network is trained using the Levenberg-Marquardt algorithm and is determined the optimal structure based on MSE. During the training, the neural network adjusts the weights and biases in each nodes connection but there does not adjust during the testing and validating to evaluate the neural network performance.

The structure for forward model consists of 8 nodes in the input layer and 3 nodes in the output layer. The nodes for the input layer compose of the past and present values of the reactor temperature T_r, the concentration of methyl methacrylate C_C, the jacket temperature T_j and the set point of the jacket temperature T_{jsp} which are a function of the reactor temperature, the jacket temperature and the concentration of methyl methacrylate. The nodes for the output layer are to predict the future values of the reactor temperature, the jacket temperature and the concentration of methyl methacrylate as follows:

$$C_C(k+1), T_r(k+1) = f\,(C_C(k-1), C_C(k), T_r(k-1), T_r(k), T_{jsp}(k-1), T_j(k), T_j(k-1), T_{jsp}(k)) \tag{4.3}$$

where k denotes the current time of the variables.

In the neural network design step, the appropriated neural network structure is defined by choosing the number of nodes in the hidden layer. The neural network forward model is trained using Levenberg-Marquardt algorithm. The sigmoid function is used as the activation function of the nodes in the hidden layers and linear function is used as the activation function in its output layer. The common objective of the neural network training is to minimize the error between the predicted neural network values and actual targeted values. The equation for MSE calculation is shown below:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(T_{aci} - T_{pi})^2 \tag{4.4}$$

where n is the number of data, T_{ac} is the actual targeted temperature values and T_p is the predicted neural network values.

After training step, the trained neural network is tested by testing data sets and validating data sets for the performance monitoring. If the MSE values of the testing data are not desired, the obtained neural network is not suitable and requires more training. It can be done by reinitialize the weights and biases and retain the neural network. On the other hand, if the MSE values of the testing data are desired, the obtained neural network is validated with validating data set. If the MSE value of validating set is not desired, the network struture is changed by changing the numbers of hidden layers and the number of nodes in the hidden layer. In this research, the number of hidden layer is varied from 1 node to 20 nodes.

Basic steps of the neural network designing are shown in figure 4.1. Many procedures of the neural network designing are summarized in this figure. The examples of training data sets for this process are shown in figures 4.2 to 4.9

After generating data, all of generating data sets are integrated, normalized and randomized, respectively. Figures 4.5-4.7 show the set point of the jacket temperature, the concentration of MMA and the reactor temperature obtaining. After that, the total obtained data are classified into 3 sets consisting of the training, testing and validating as 60%, 30% and 10% of the total obtained data, respectively. The numbers of samples for training, testing and validating data sets are 2,700, 1,400 and 400, respectively.

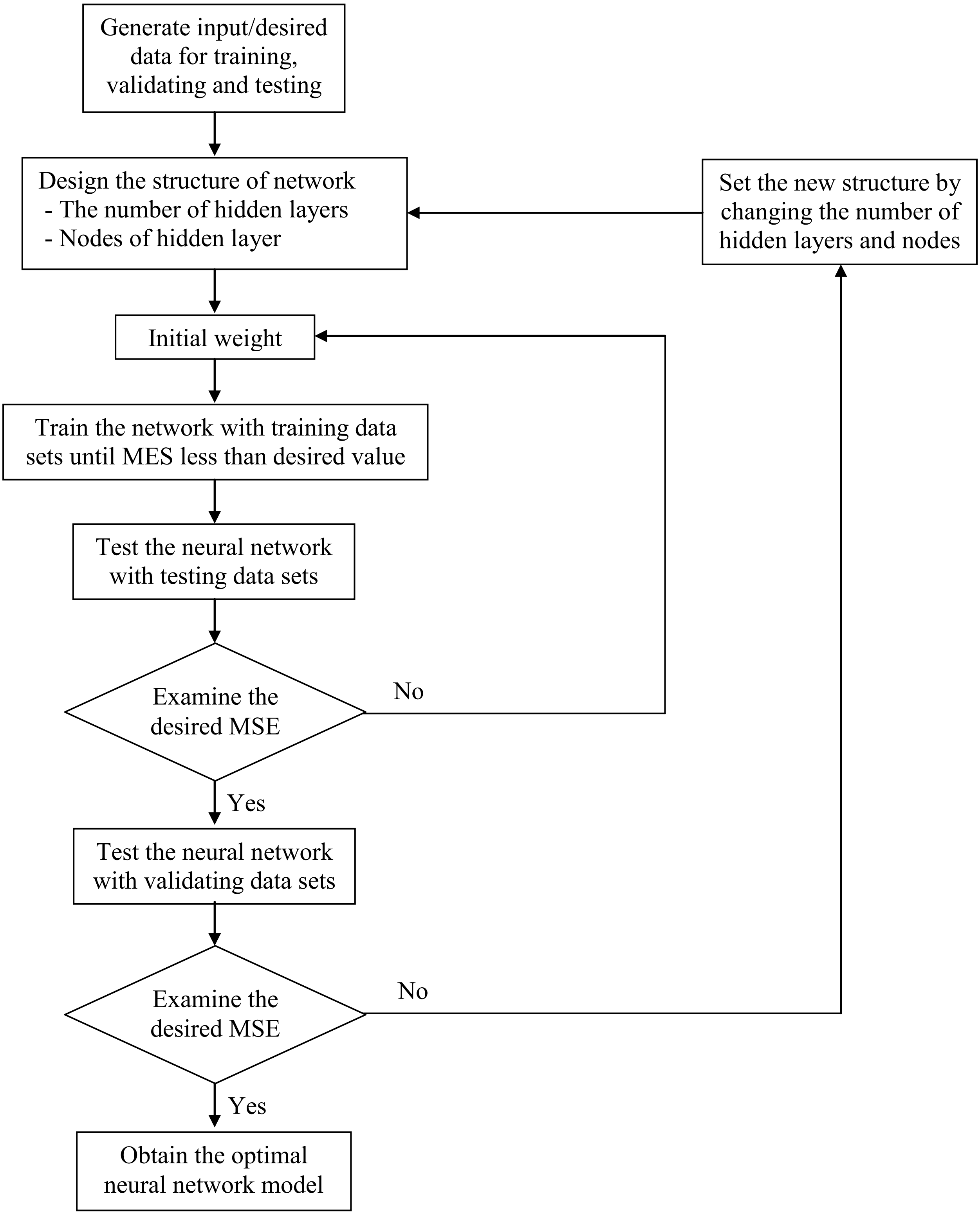


Figure 4.1 Procedure for the obtaining forward and inverse neural network model

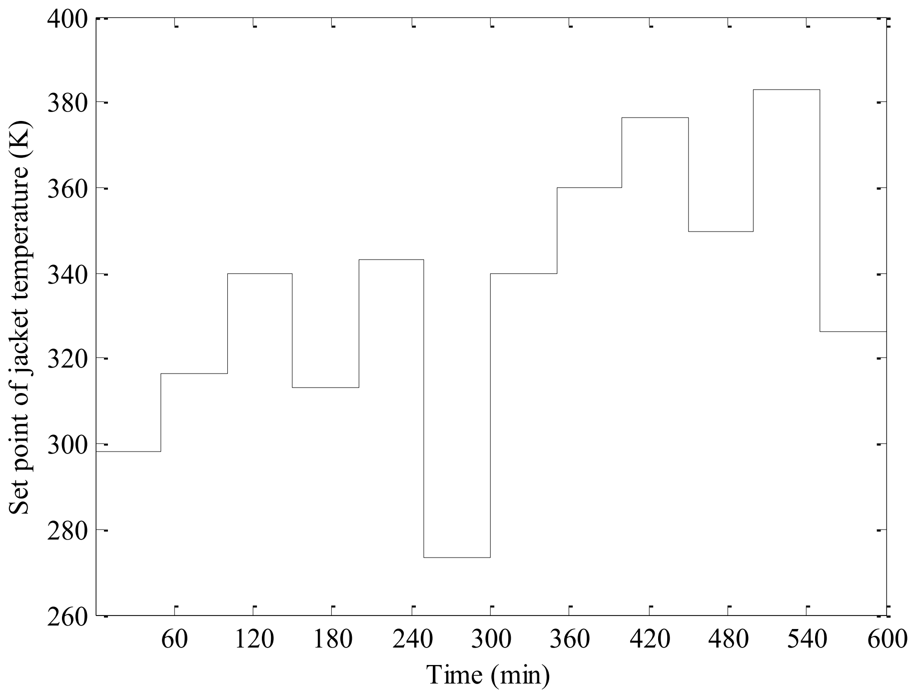


Figure 4.2 Set point of the jacket temperature (T_{jsp}) for the training data set 1 of the forward model

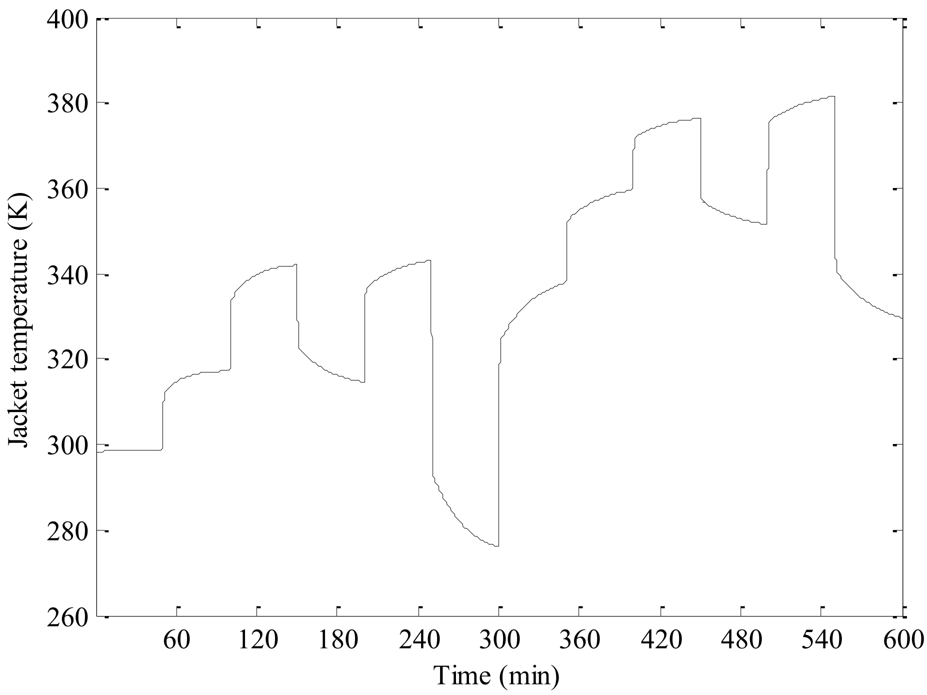


Figure 4.3 The jacket temperature (T_{jsp}) for the training data set 1 of the forward model

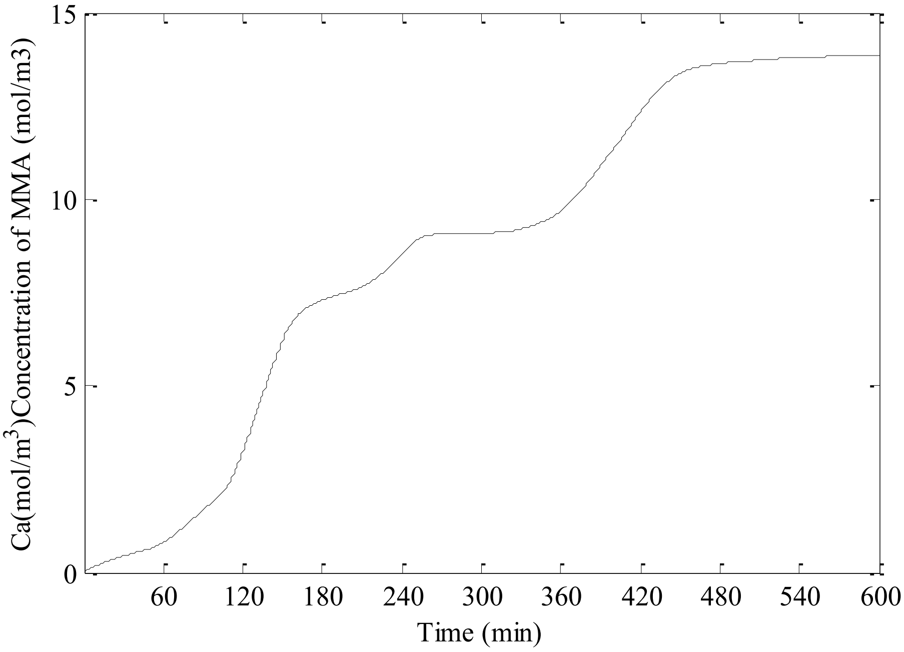


Figure 4.4 Concentration profile of MMA for the training data set 1 of the forward model

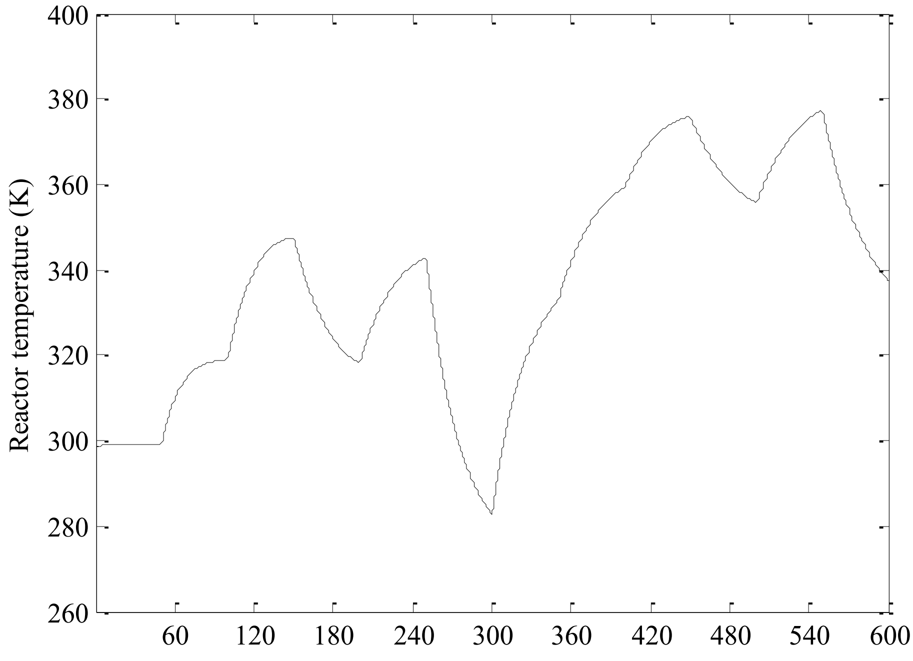


Figure 4.5 Reactor temperature profile (T_r) for the training data set 1 of the forward model

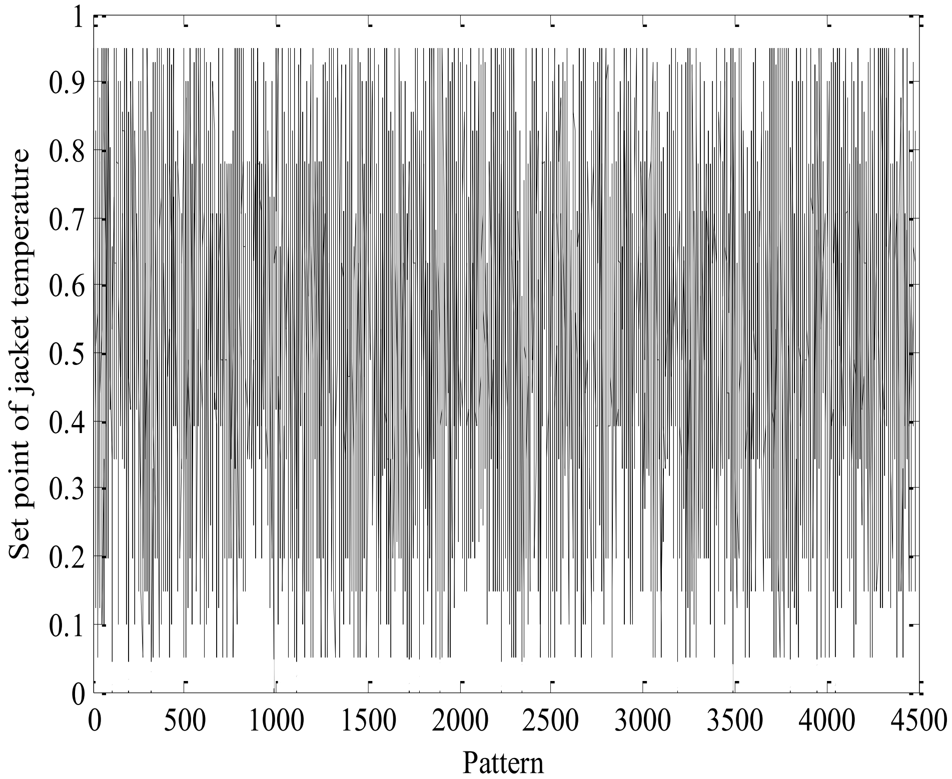


Figure 4.6 Data summation of the set point of the jacket temperature for the forward model

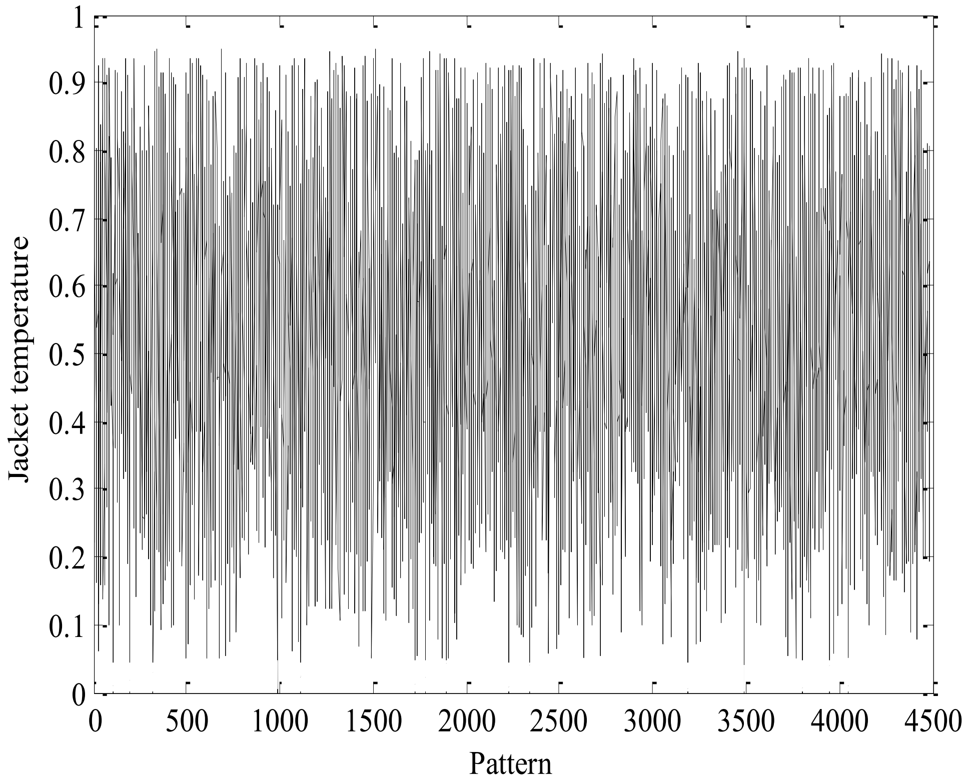


Figure 4.7 Data summation of the jacket temperature for the forward model

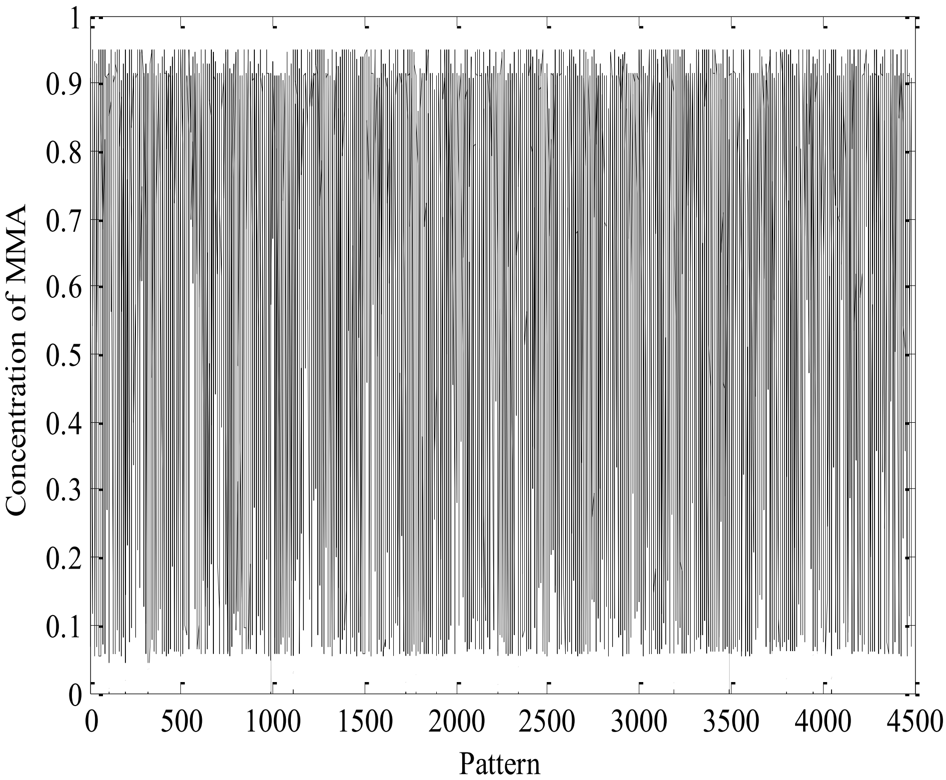


Figure 4.8 Data summation of the concentration profile of MMA for the forward model

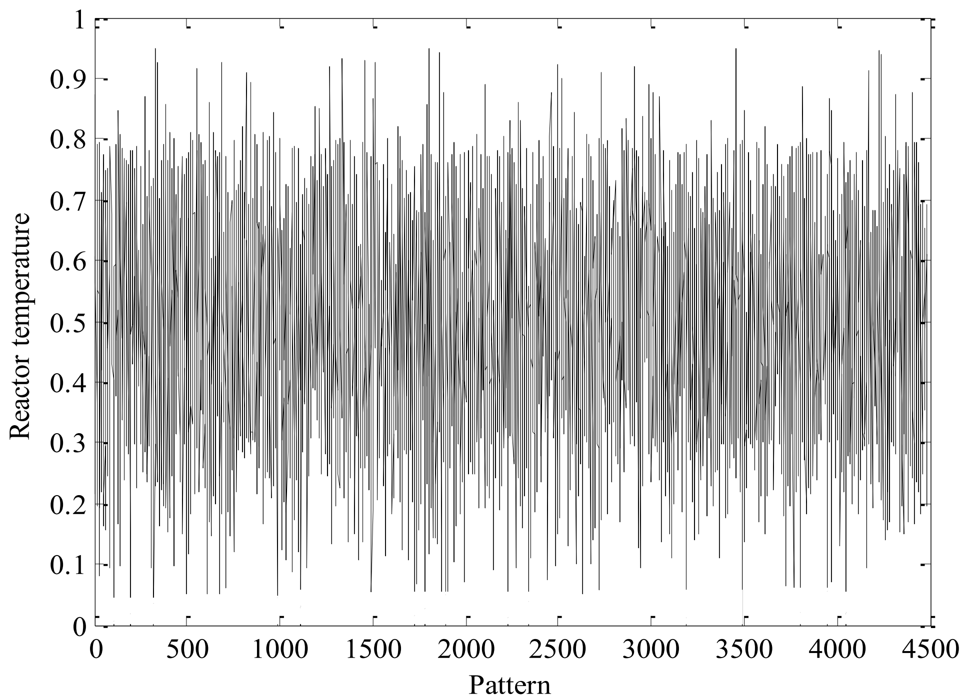


Figure 4.9 Data summation of the reactor temperature for the forward model

4.2 Neural Network Inverse Model

In this part, the neural network inverse model is applied to control the process. The detailed procedure to find the inverse neural network model is summarized in the figure 4.1. The structure for inverse model consists of 8 nodes in the input layer and a node in the output layer.

The optimal neural network inverse model is utilized to predict the manipulated variable (set point of jacket temperature). The prediction of manipulated variable for controller requires the past and present values of the process outputs and the past values of manipulated variable as well as it requires the future value of set point of reactor temperature. The input and output pattern for the inverse model is shown below.

$$T_{jsp}(k) = f^{-1} (C_C(k-1), C_C(k), T_r(k-1), T_r(k)), T_j(k-1), T_j(k), T_r(k+1), T_{jsp}(k-1)) \quad (4.5)$$

where k denotes the current time of the variables.

In this work, the optimal structures are selected by the minimum MSE method. The numbers of nodes in the hidden layers are varied from 1 to 20 nodes. Table A.1 and A.2 show the MSE values obtained from the neural network forward model and Table A.3 and A.4 show the MSE values obtained from the neural network inverse model for the process. Based on the minimizing MSE error values, it is found that numbers of nodes in hidden layer for the forward model and inverse model are 6 nodes in first hidden layer and 8 nodes in second hidden layer and 4 nodes in first hidden layer and 8 nodes in second hidden layer which the best to be applied respectively. The optimal neural network architectures for the forward model and inverse model are [8-6-8-3] and [8-4-8-1] as show in figures 4.10 and 4.11, respectively.

4.3 Simulation Results

After training process, the neural network for forward model and inverse model are validated by the sets of validating data for the performance monitoring. Figures

4.18-4.20 show the results of neural network model validating which is the optimal neural network structure (8-6-8-3 structure). The results in these figures indicated the forward neural network models for prediction of the concentration of MMA and the reactor temperature profiles. Figures 4.23 shows the results of the inverse neural network model validating which is the optimal structure (8-4-8-1 structure). The results in this figure represented the manipulated variable or control action for the process T_{jsp}.

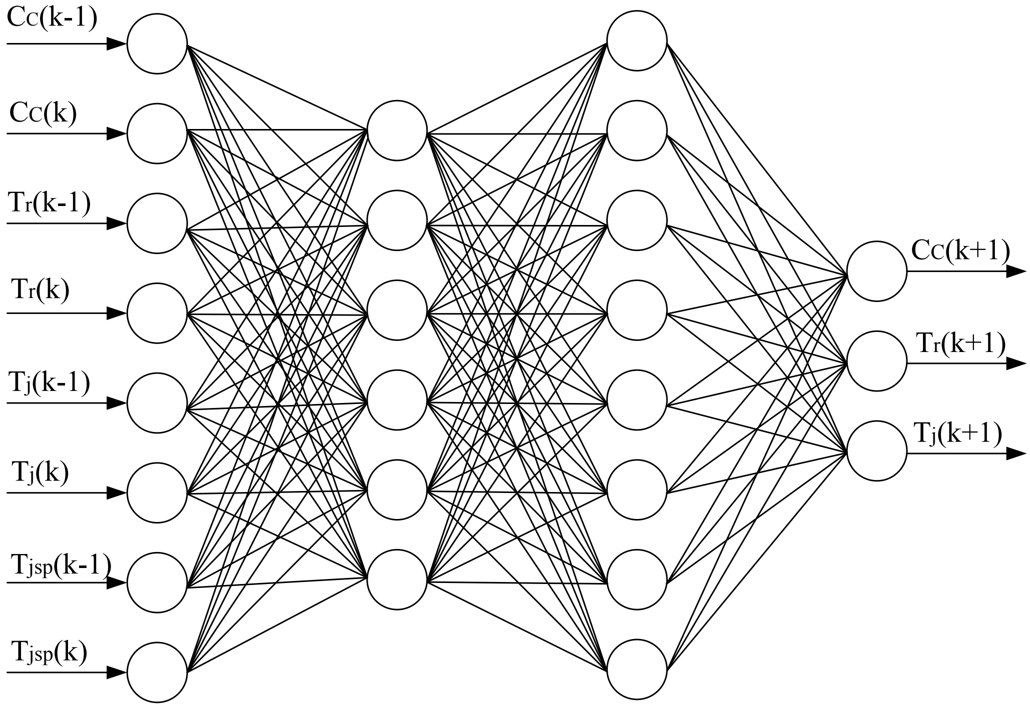


Figure 4.10 The neural network forward of the process (structure 8-6-8-3)

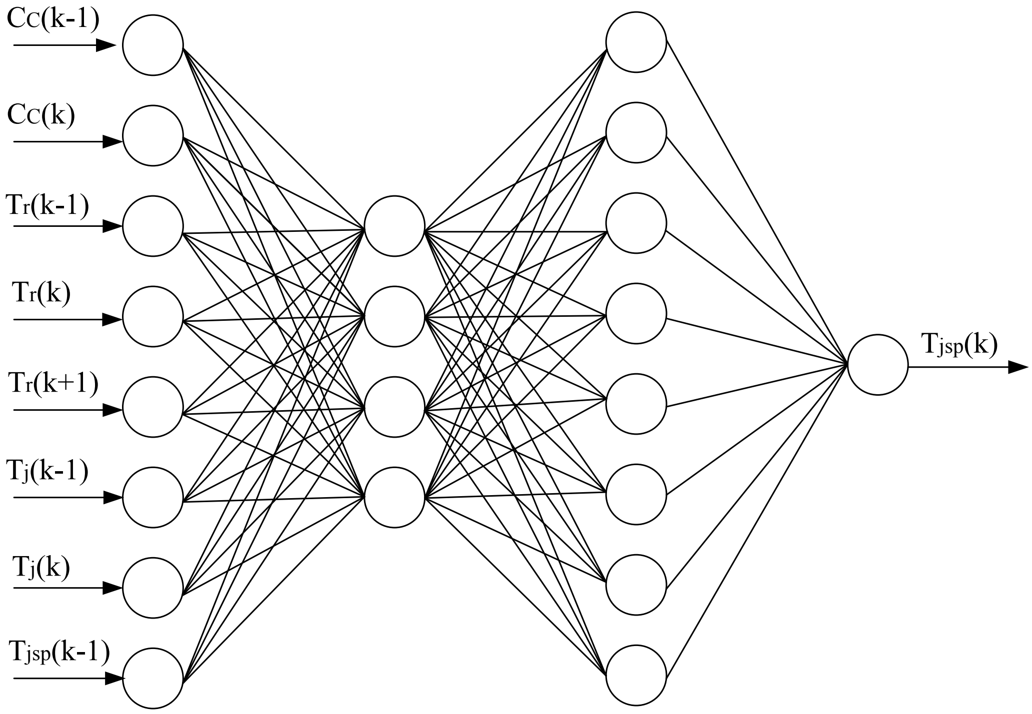


Figure 4.11 The neural network inverse model of the process (structure 8-4-8-1)

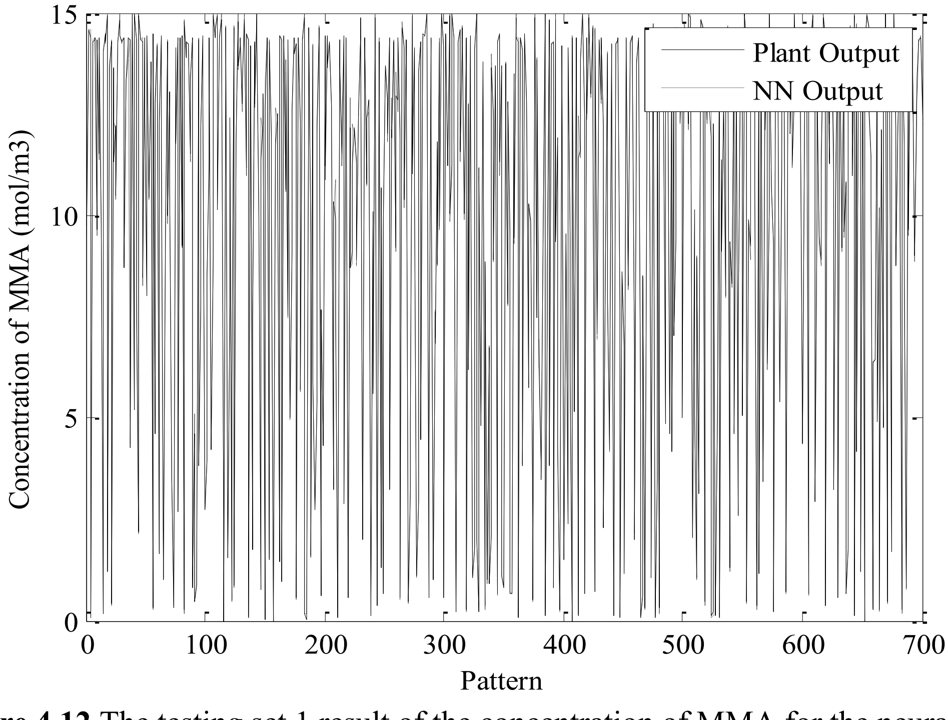


Figure 4.12 The testing set 1 result of the concentration of MMA for the neural forward network model (structure 8-6-8-3)

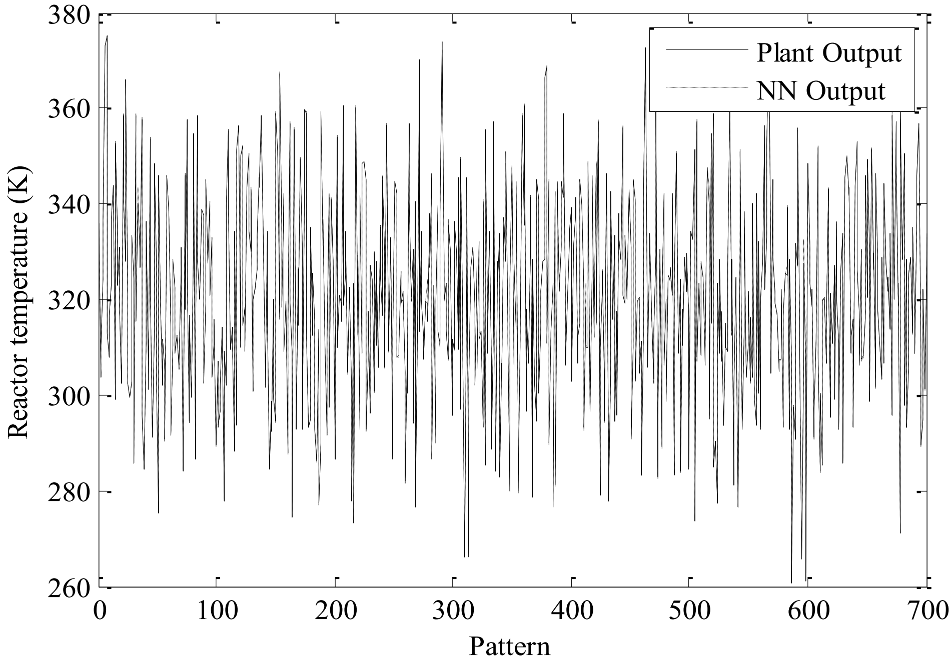


Figure 4.13 The testing set 1 result of the reactor temperature for the neural network forward model (structure 8-6-8-3)

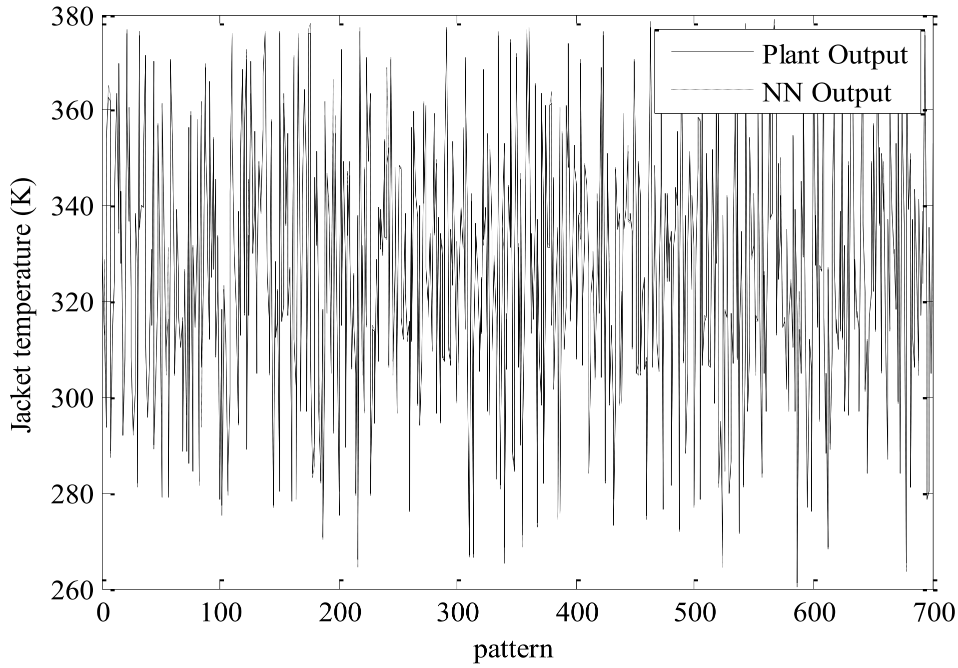


Figure 4.14 The testing set 1 result of the jacket temperature for the neural network forward model (structure 8-6-8-3)

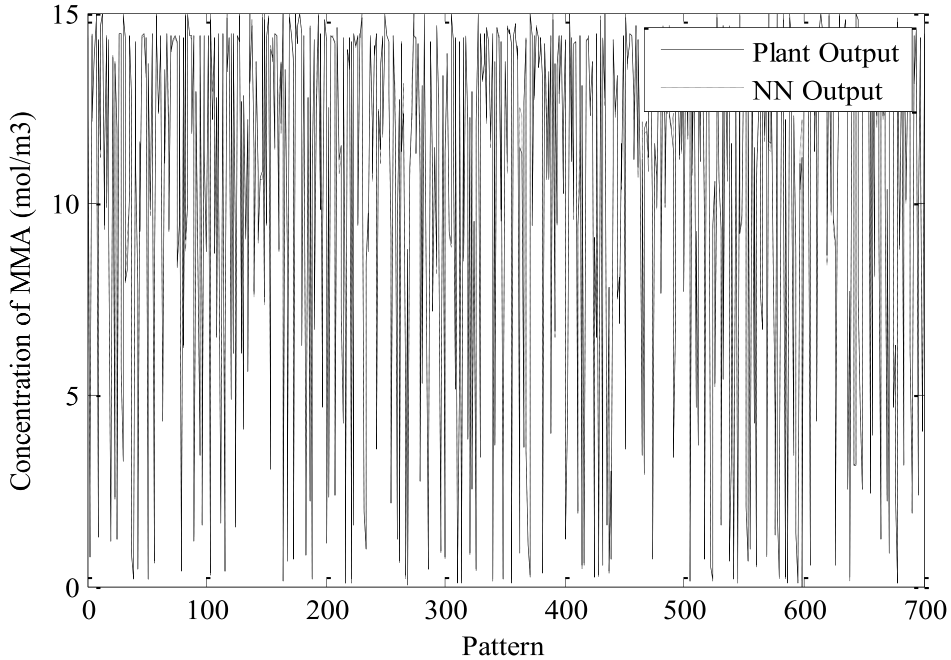


Figure 4.15 The testing set 2 result of the concentration of MMA for the neural forward network model (structure 8-6-8-3)

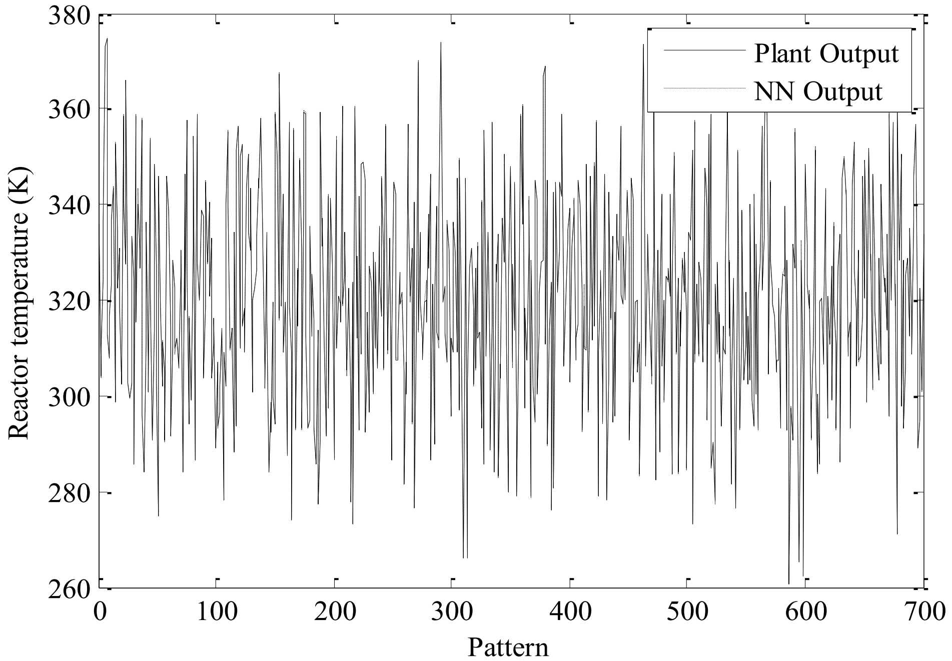


Figure 4.16 The testing set 2 result of the reactor temperature for the neural network forward model (structure 8-6-8-3)

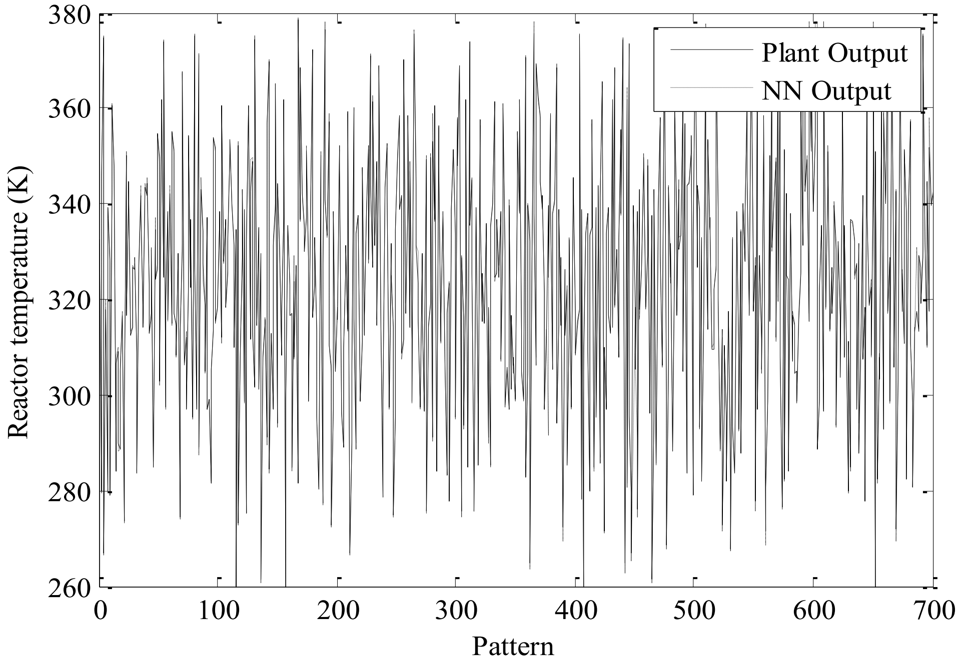


Figure 4.17 The testing set 2 result of the jacket temperature for the neural network forward model (structure 8-6-8-3)

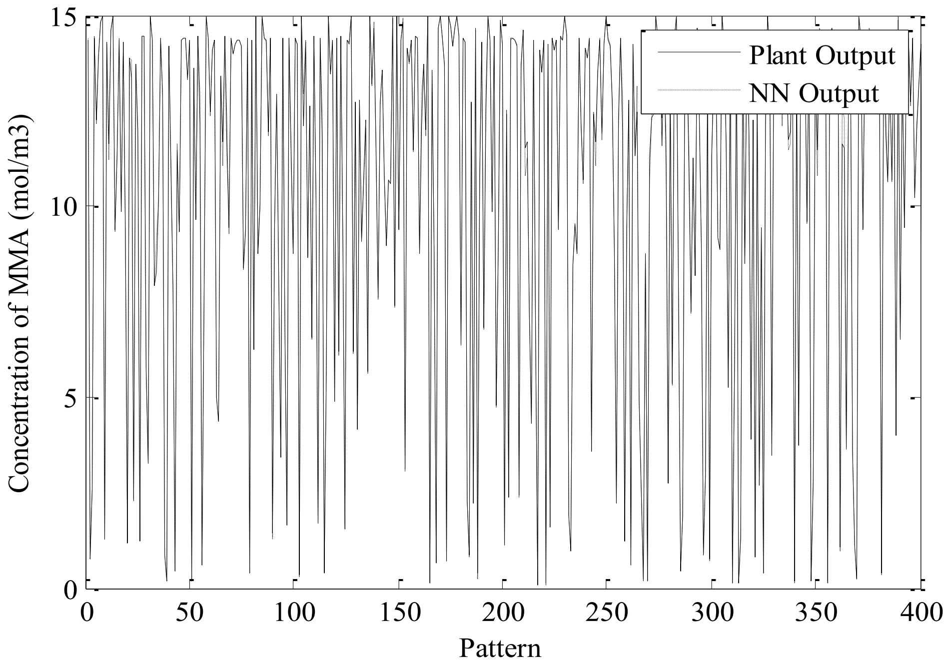


Figure 4.18 The validating set result of the concentration of MMA for the neural forward network forward model (structure 8-6-8-3)

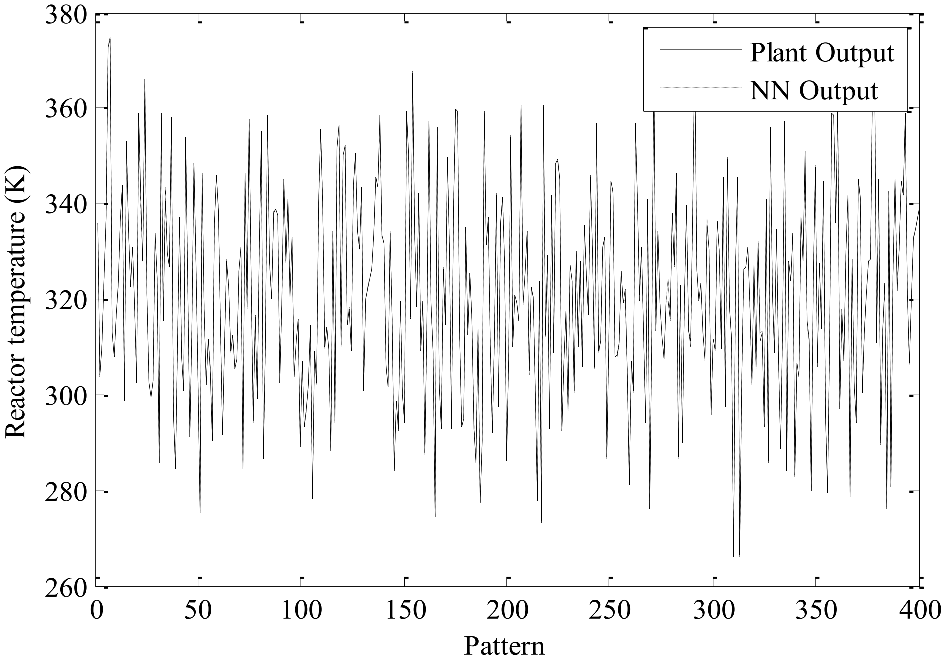


Figure 4.19 The validating set result of the reactor temperature for the neural network forward model (structure 8-6-8-3)

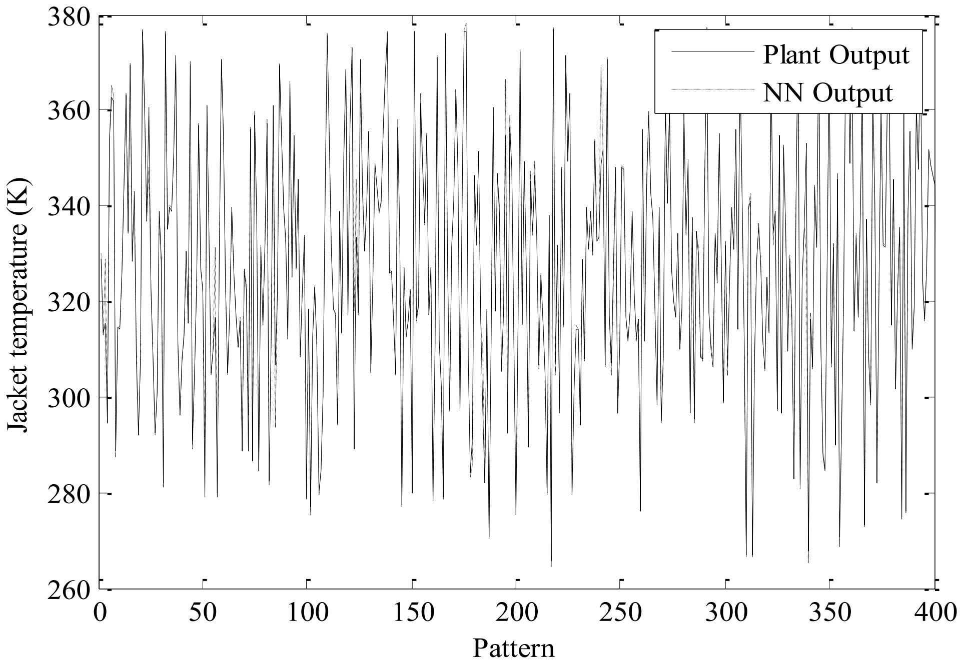


Figure 4.20 The validating set result of the jacket temperature for the neural network forward model (structure 8-6-8-3)

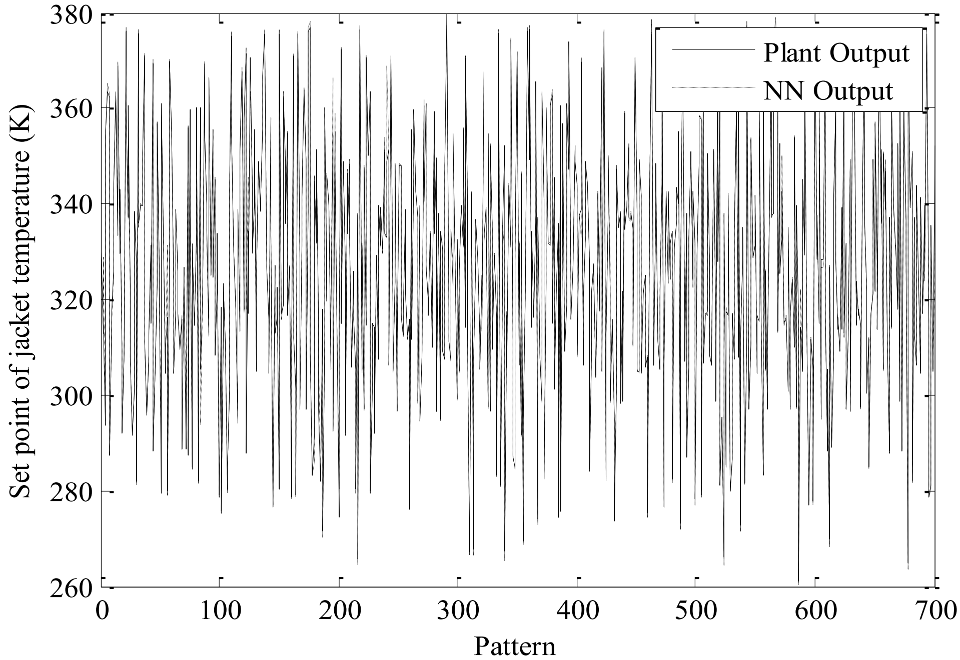


Figure 4.21 The testing set 1 result of the manipulated variable for the neural network inverse model (structure 8-4-8-1)

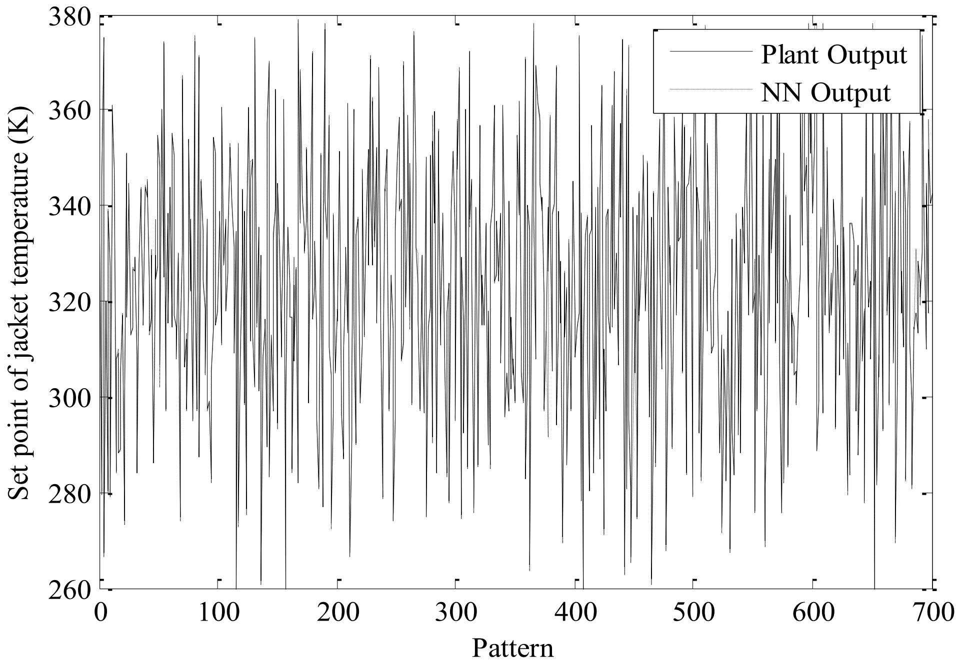


Figure 4.22 The testing set 2 result of the manipulated variable for the neural network inverse model (structure 8-4-8-1)

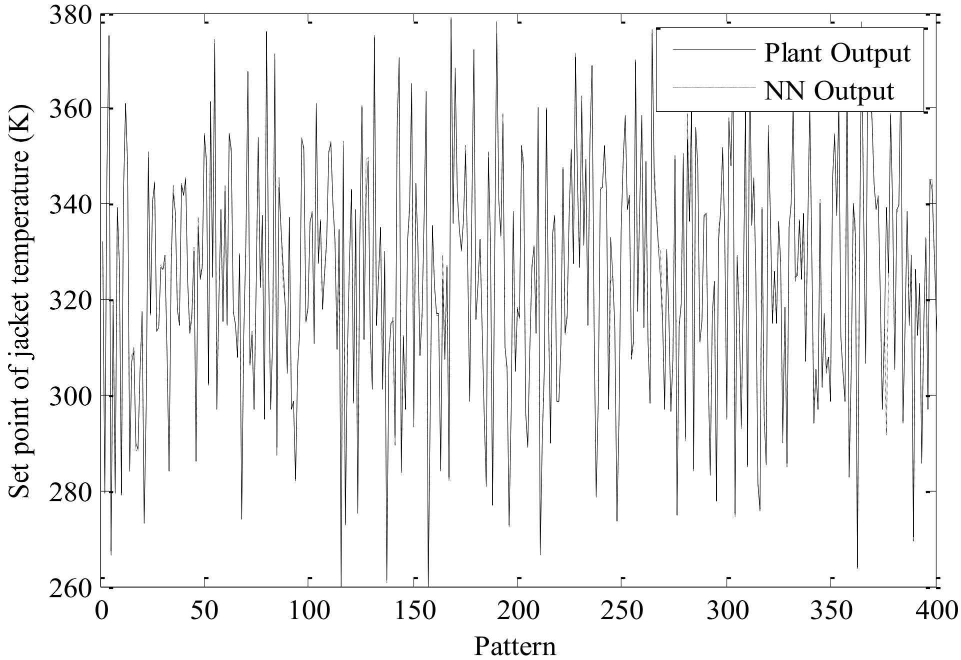


Figure 4.23 The validating set result of the manipulated variable for the neural network inverse model (structure 8-4-8-1)

4.4 Dynamic Optimization

In this work, a Matlab program is written to solve the optimization problem using a successive quadratic programming (SQP) algorithm in Matlab Optimization Toolbox. The written program is tested to determine an optimal temperature of the exothermic batch reactor studied by Aziz et al. (2000). The temperature results show that this program is effective and applicable to determine an optimal temperature of this work.

In this type of problem, the objective is to determine the optimal temperature policy maximizing the amount of a desired product concentration for a given fixed batch time subject to bounds on the reactor temperature. The problem can be written mathematically as

$$\max_{T_r} C_C(t_f) \tag{4.6}$$

Subject to $\dot{x} = f(x(t), T_r, p, t)$ process model (4.7)

$x(t_0) = x(0)$ initial condition (4.8)

$$(T_r)_{min} \leq Tr \leq (T_r)_{max} \tag{4.9}$$

$$t_f = t_f^* \tag{4.10}$$

where x is the vector of state variables, $\dot{x}$ is the derivative of x with respect to time (t) and p is the process parameters. The batch time (t_f) is specified for 10 hours. The lower bound on the temperature is the initial temperature that operates at the ambient condition and the lower and upper bound is dictated by the maximum temperature of the experimental data used by Witczak et al. (2010) to build their models.

The dynamic optimization maximizing production concentration with respect to variations of time intervals: 1, 2, 4 and 8 intervals have been carried out. The simulation results with different time intervals are shown in figure 4.23. Table 4.1 reports the temperature and the concentration of MMA (desire product) of each time intervals. It has been found that at the final time, the maximum product achieved at the case of 8 intervals. The obtained temperature profile is represented the set point of the reactor temperature for controller design.

Table 4.1 Optimization results

Interval	Off-line optimal temperature (K)								$C_C(t_f)$
	Time (min)								
	0	75	150	225	300	375	450	525	
1	342.5								13.934
8	348.0	348.0	347.5	345.9	341.6	333.5	328.2	324.8	13.962

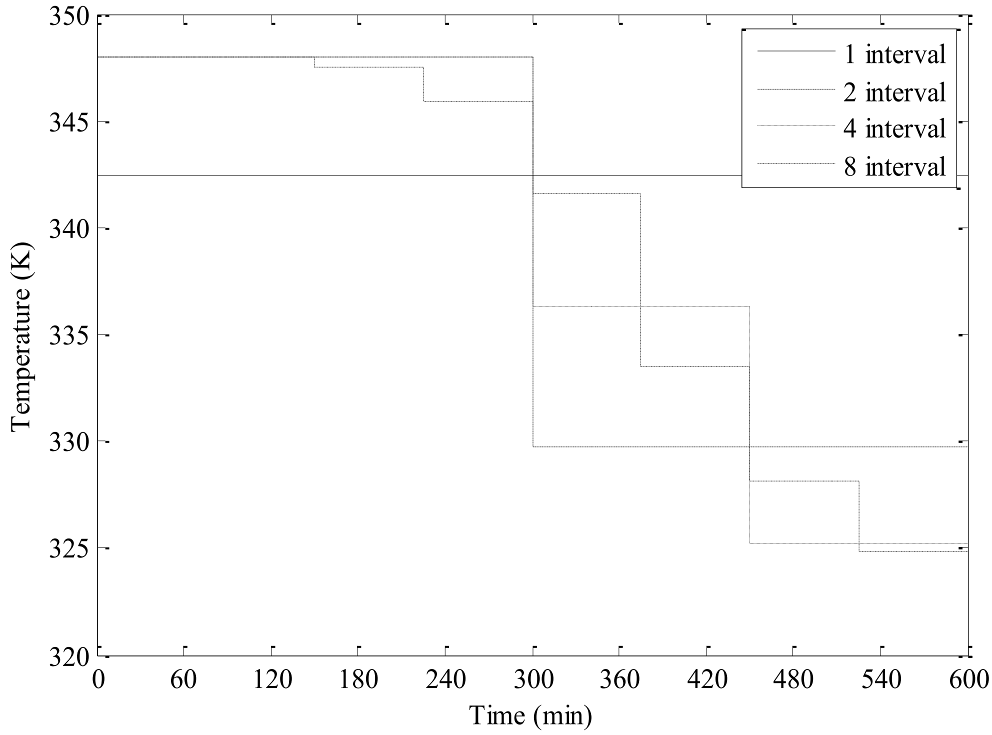


Figure 4.24 Temperature profile for the optimization problem

CHAPTER V

THE CONTROLLER DESIGN BASED NEURAL NETWORK FOR THE TEMPERATURE CONTROL OF THE PROCESS

This chapter describes a neural network based model predictive control (NNMPC), a PID control and a neural network direct inverse control (NNDIC) for temperature controlling in several cases such as a nominal case and parameter mismatch cases. In addition, this chapter demonstrates the robustness and performance of the proposed controllers.

5.1 Neural Network based Model Predictive Control (NNMPC)

An obtained neural network forward model in chapter IV is applied as a predictor to predict future values of outputs over a prediction horizon (*p*) within a model predictive control algorithm. An optimal manipulated variable (T_{jsp}) is determined by solving a optimization problem to minimize a specified objective function subject to the neural network model and lower and upper bound of the manipulated variable. To understand clearly in this concept, a structure for the NNMPC controller is shown in a figure 5.1. The Matlab program is used to solve a minimization problem using a successive quadratic programming (SQP) algorithm. The form of an objective function for the manipulated variable determining is shown below:

$$\min_{T_{jsp}} \sum_{i=1}^{p} \left[W_1 \{T_{rsp}(k+i) - T_r(k+i)\}^2 + W_2 \{\Delta T_{jsp}\}^2 \right] \tag{5.1}$$

Subject to

Neural network for forward model in chapter 4

$$(T_{jsp})_{min} \le T_{jsp}(k+i) \le (T_{jsp})_{max}, \qquad i = 1,2,3,\ldots,P \tag{5.2}$$

$$Tr(k+p) = Trsp(k+p) \tag{5.3}$$

where p is a parameter specifying the prediction horizon, M is the control horizon, W_i is the weighting parameter used to give different weights to different squared tracking error and T_{rsp} is the set point of the reactor temperature obtained off-line optimization.

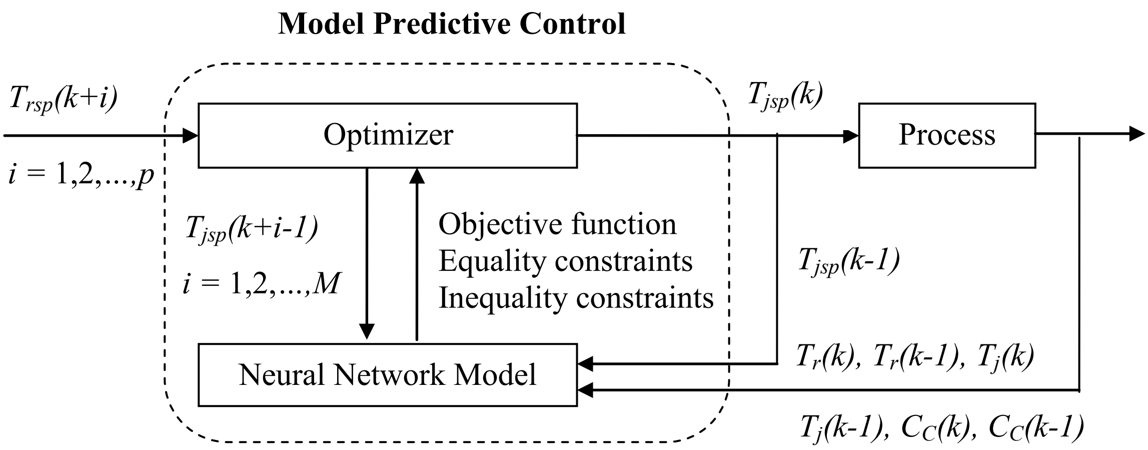


Figure 5.1 The NNMPC strategy

In order to tune parameter of the NNMPC, the decreasing of the prediction horizon effect to produce more rigorous control action, faster response and more overshoot**.** On the other hand, the increasing of control horizon tends to produce less rigorous control action, slowly response and less overshoot. From the reason, the applied prediction horizon and the used control horizon for the temperature control are 5 and 5 respectively.

5.2 Neural Network Direct Inverse Control (NNDIC)

In this part, a neural network inverse model is applied to control the process. The method of a neural network training as a controller is presented in chapter 4. An optimal neural network inverse model [8-4-8-1] is utilized to predict the manipulated variable (the set point of the jacket temperature). The prediction of the manipulated variable for controller requires the past and present values of the process outputs and the past values of the manipulated variable as well as it requires the future value of the set point of the reactor temperature. Figure 5.2 shows the structure of the NNDIC for controlling of reactor temperature.

The performance of the NNDIC controller is tested using uncertainty of the process. They consist of kinetic rate, heat of reactions and heat transfer coefficient.

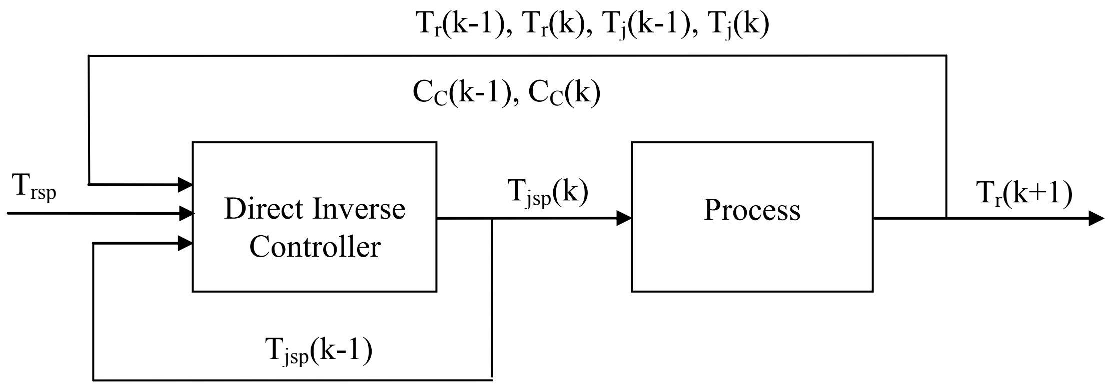
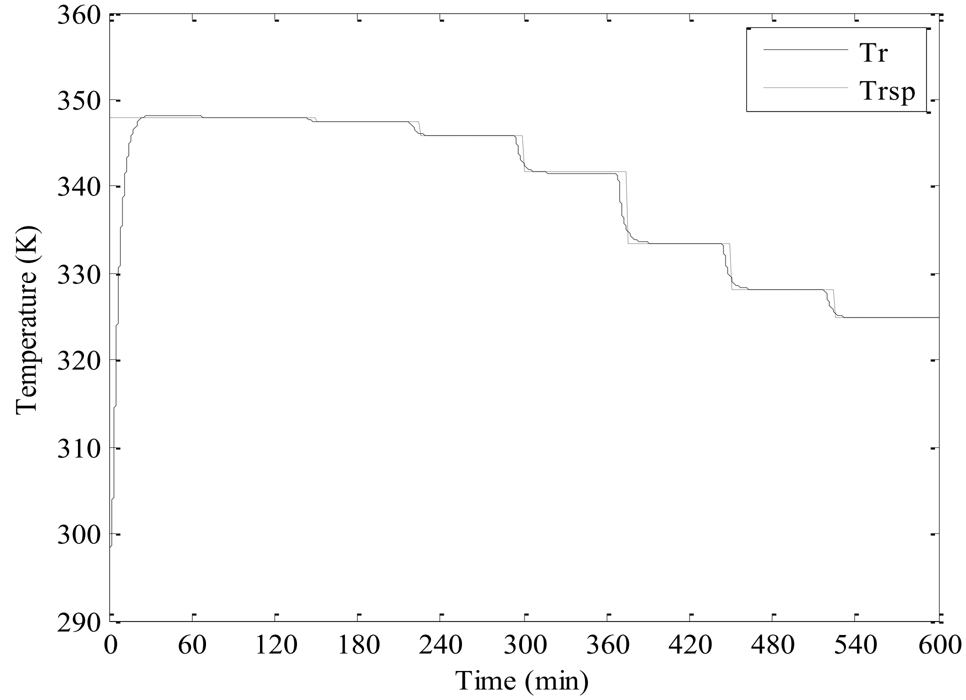


Figure 5.2 The structure of the NNDIC strategy

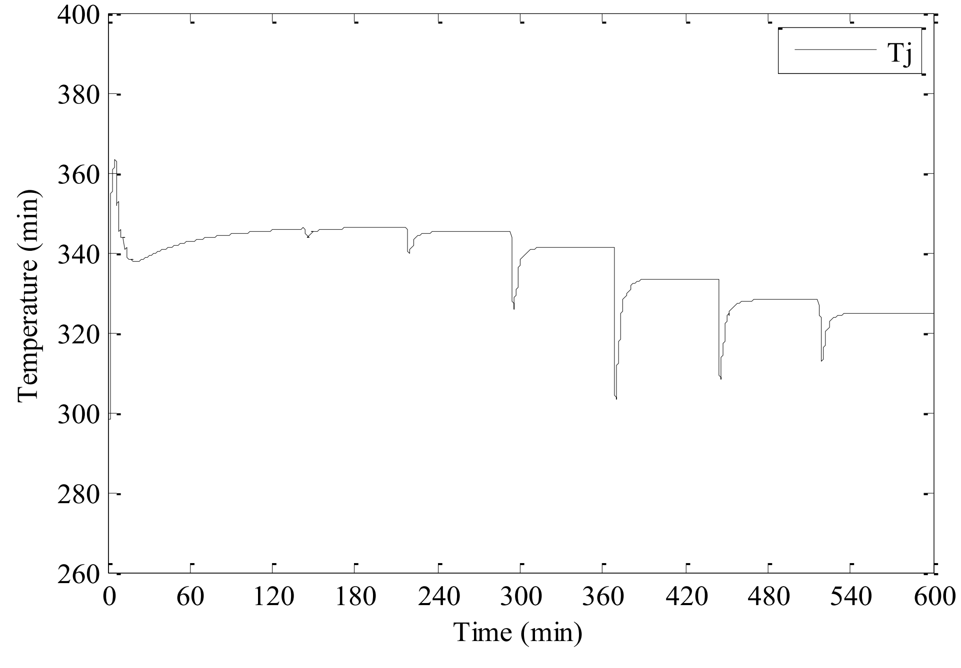
5.3 Simulation Results

In the simulation studies, the objective is to control the reactor temperature to the set point by adjusting the set point of the jacket temperature. They are divided into 2 cases which are the nominal case and parameter mismatch cases. The mismatch cases consist of increasing 30% of k_1, decreasing 30% of k_2, increasing 30% of ΔH and decreasing 30% of U from its nominal values. The closed-loop performance of the NNMPC, NNDIC and PID control are indicated by the integral of absolute value of the error (IAE).

For the nominal case, the process parameters are presented in Table A.1. Figure 5.3 shows the control of the reactor temperature using NNMPC and figure 5.4 and 5.5 show it with PID control and NNDIC, respectively. The results in these figures indicate that the NNMPC can bring the temperature closely to the set point without overshoot oscillations and offset. In contrast, the NNDIC and PID control case the overshoot and oscillation of the control variable. For the comparison between NNDIC and PID control in term of the offset, the NNDIC has the offset but the PID control has not offset. For the response of the manipulated variable, the manipulated variable adjustment of the NNMPC is smooth than the NNDIC and PID control.

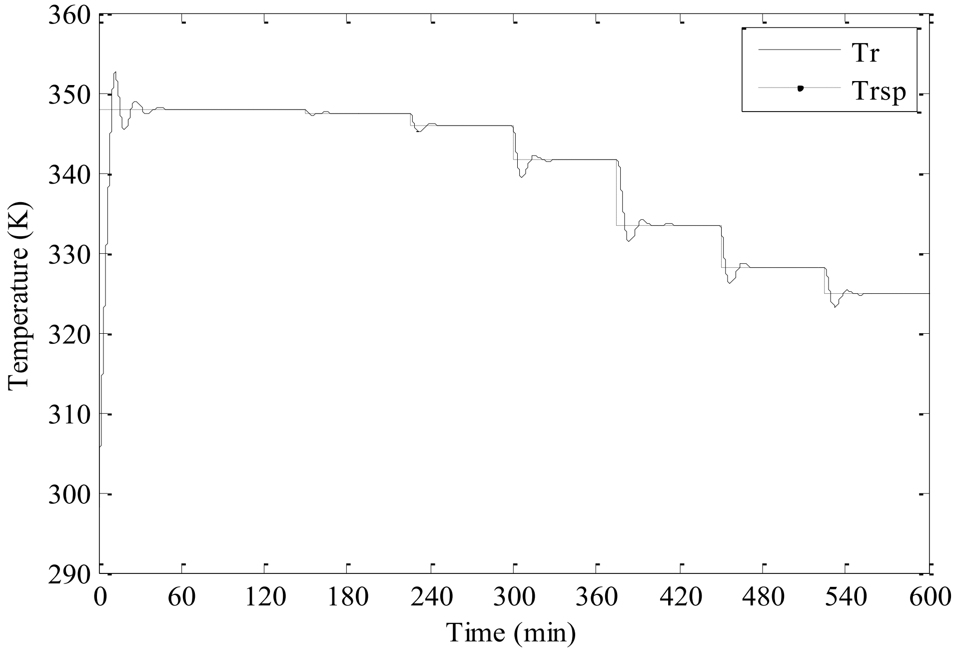


(a)

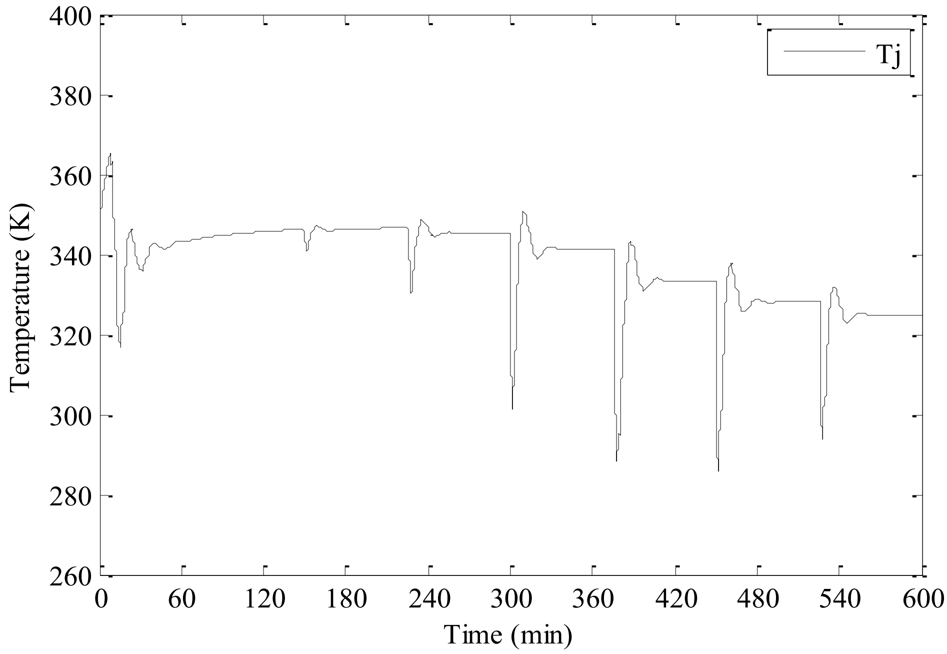


(b)

Figure 5.3 The temperature control using NNMPC under the nominal case: (a) the control variable (T_r) and (b) the manipulated variable (T_j)

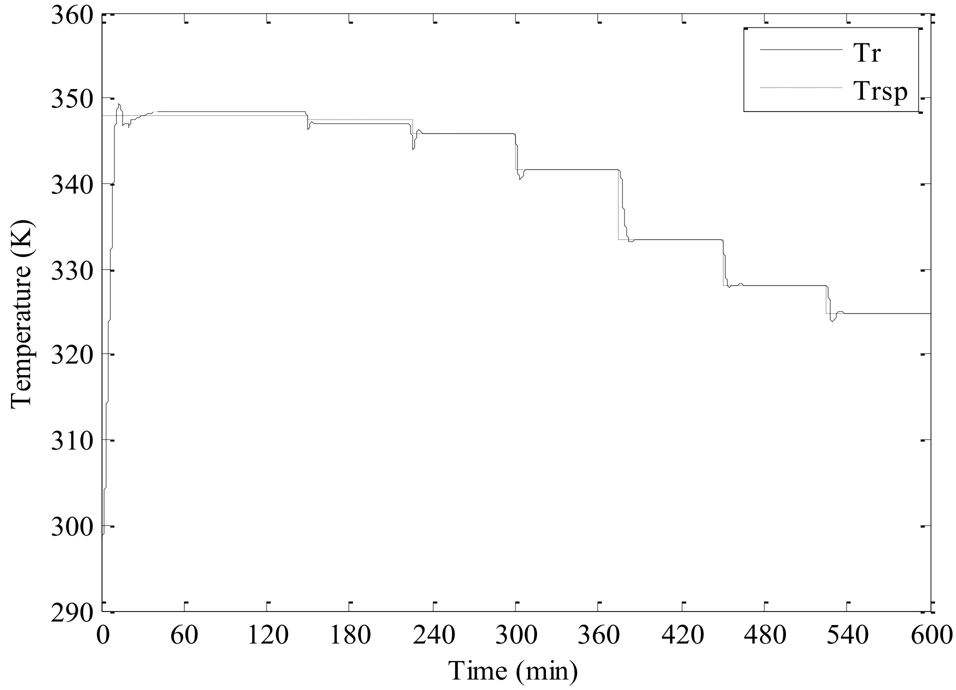


(a)

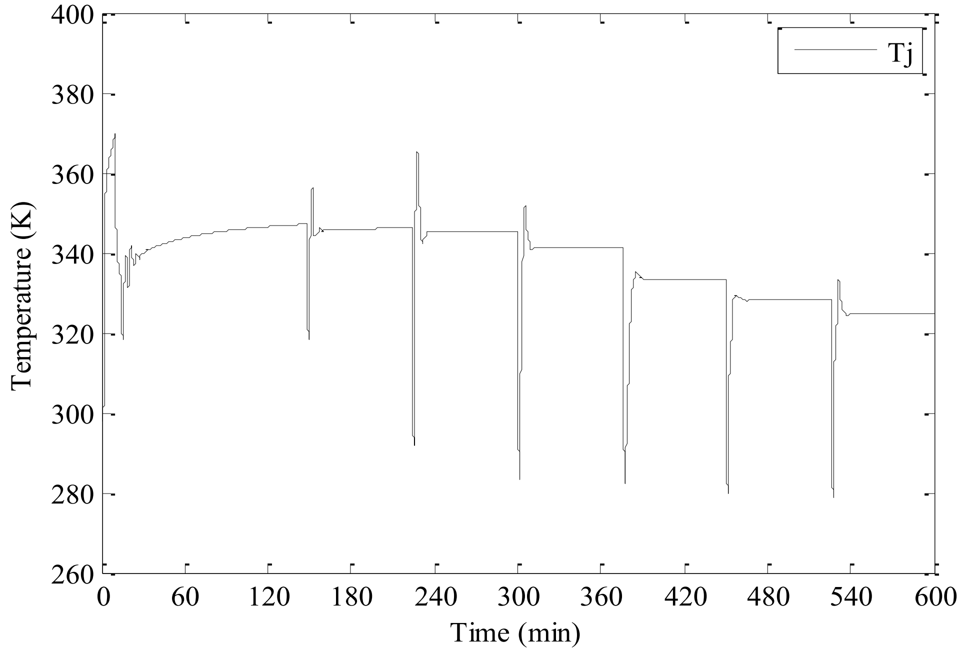


(b)

Figure 5.4 The temperature control using PID control under the nominal case: (a) the control variable (T_r) and (b) the manipulated variable (T_j)



(a)



(b)

Figure 5.5 The temperature control using NNDIC under the nominal case: (a) the control variable (T_r) and (b) the manipulated variable (T_j)

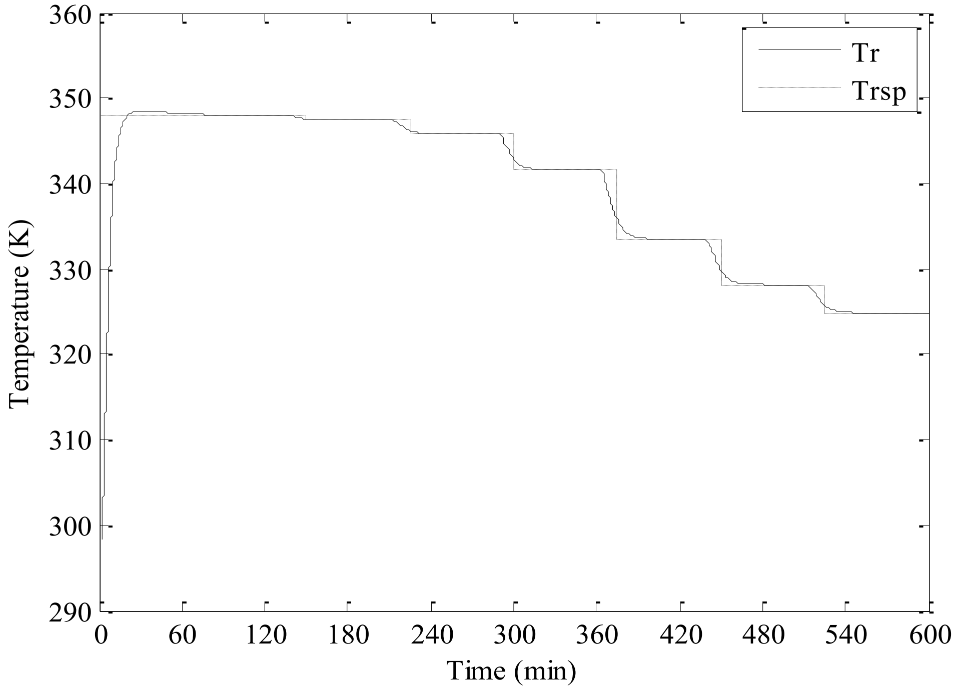
For the model mismatch cases, the rate of reaction for forward reaction, the rate reaction for reverse reaction, the heat of reaction and overall heat transfer coefficient are considered as the model mismatch parameters. The percent choosing for parameters mismatch is considered the percent changing from its nominal values that affect to the process response tending to more rigorous response. From this reason, the model mismatch parameters is divided into 6 cases consisting of increasing 30% k_1, decreasing 30% k_2, increasing 30% k_1 and decreasing 30% k_2, increasing 30% ΔH, decreasing 30% U and the last case increasing 30% k_1, decreasing 30% k_2, increasing 30% ΔH and decreasing 30% U from its nominal value.

The results of parameter mismatch cases for the NNMPC, PID control and NNDIC are shown in figure 5.6 to 5.11. These figures show that the proposed controls can control the process and bring the temperature to the set point. For the comparison, the NNMPC give the best control performance among all control in all parameter mismatch cases and the NNDIC is more robust than PID control.

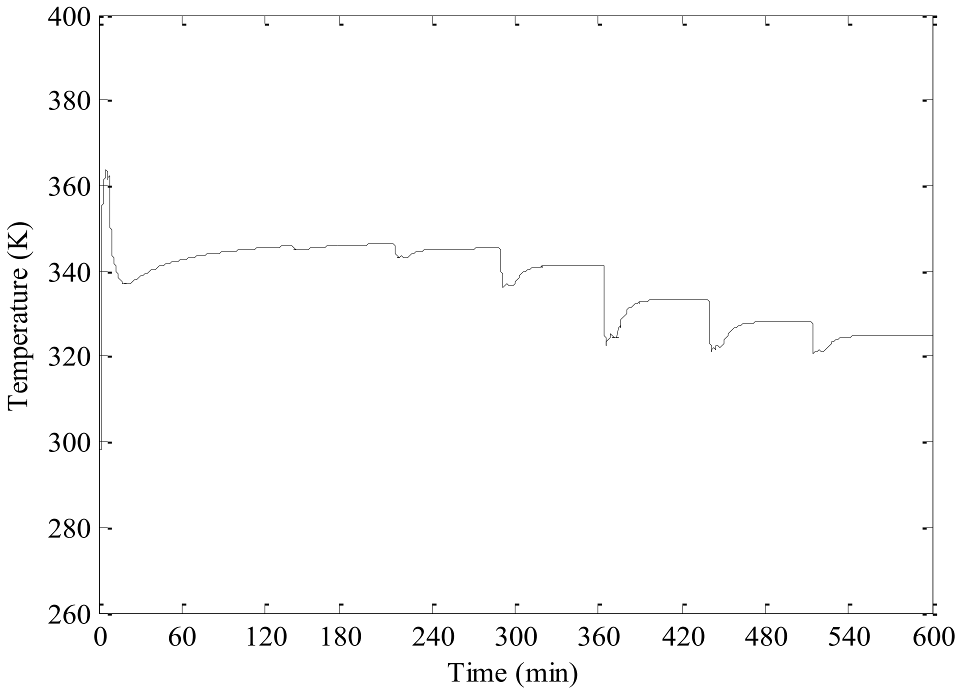
The performance index in terms of the absolute error (IAE) of three different controls consisting of the NNMPC, the PID control and the NNDIC in nominal case and parameter mismatch cases for performance testing are summarized in Table 5.1.

Table 5.1 Performance indices of the NNMPC strategy, the PID control strategy and the DIC strategy for the nominal and model mismatch cases

Cases	IAE values		
	NNMPC	NNDIC	PID
Nominal	422.0569	422.4777	422.9892
+30%k_1	421.0025	421.2645	433.2399
-30%k_2	421.1529	422.0130	424.3498
+30%k_1, -30%k_2	421.4568	421.6259	434.5736
+30%ΔH	415.2102	419.5642	433.8296
-30%U	435.0598	456.2135	550.6485
+30%k_1, -30%k_2, +30%ΔH, -30%U	447.0252	462.7851	584.9196

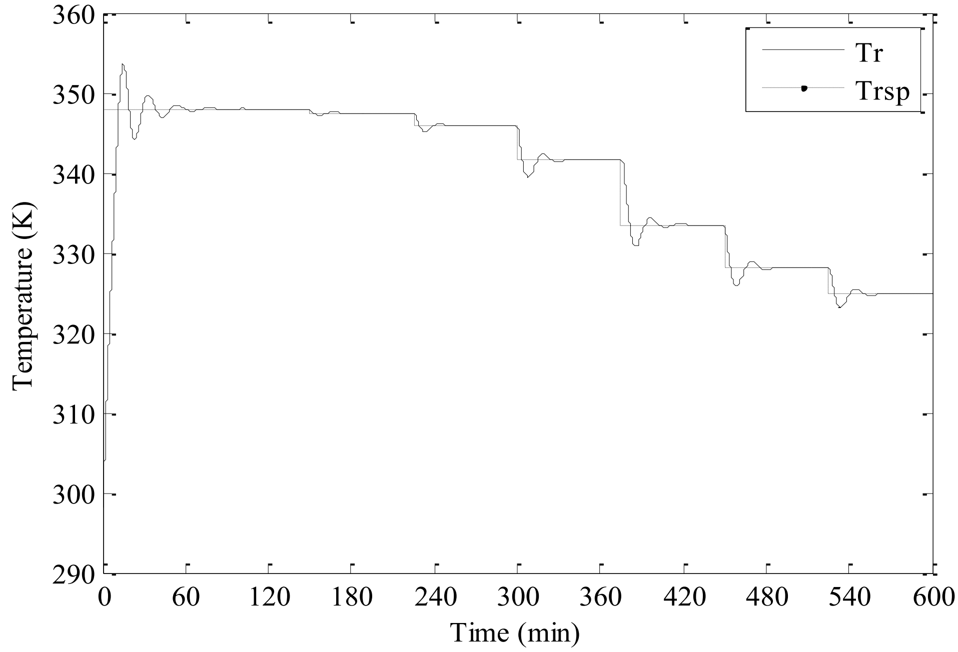


(a)

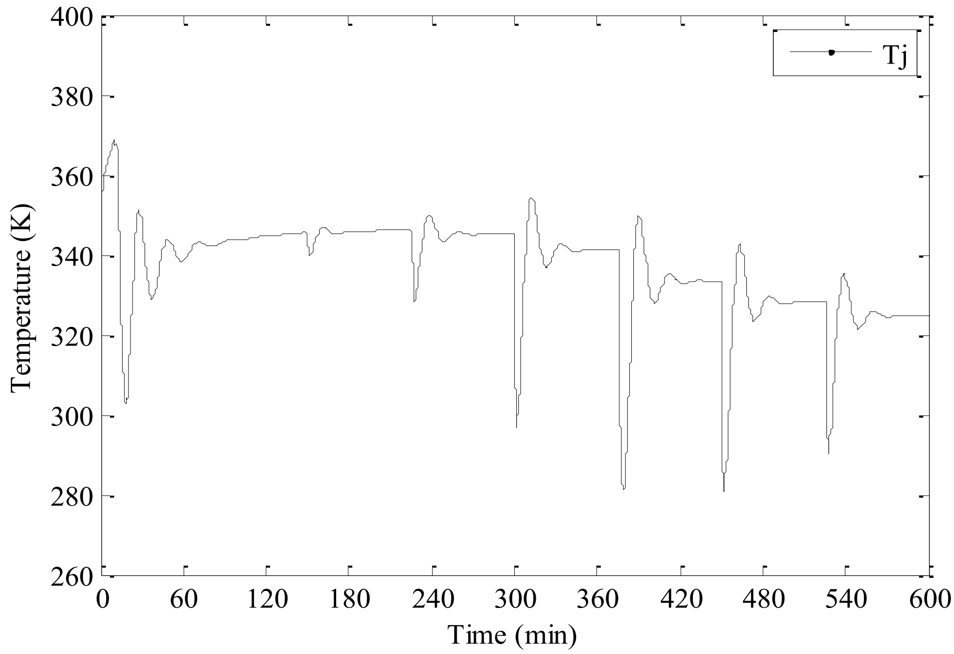


(b)

Figure 5.6 The temperature control using NNMPC under the parameter mismatch case (-30% U): (a) the control variable (T_r) and (b) the manipulated variable (T_{jsp})

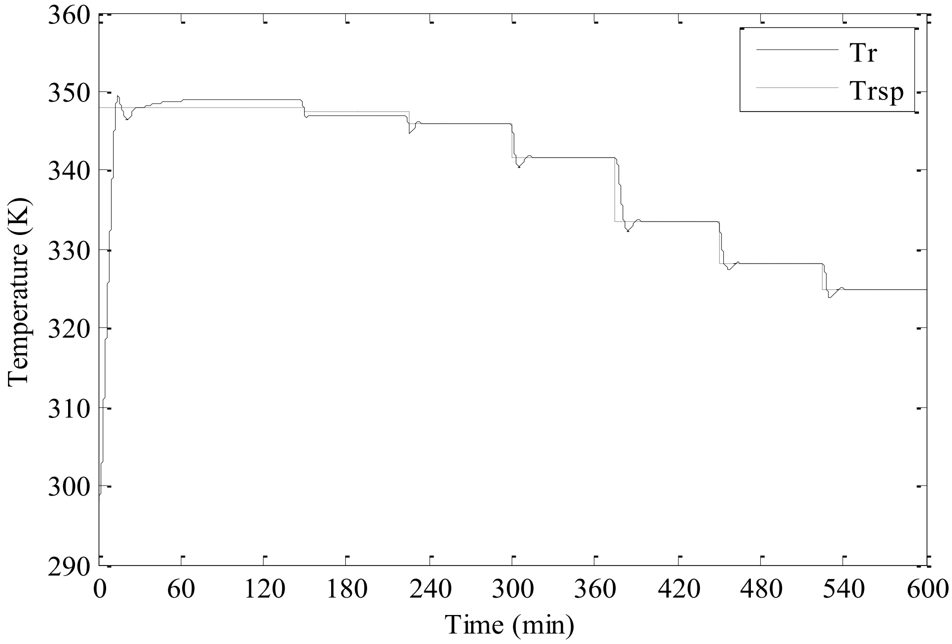


(a)

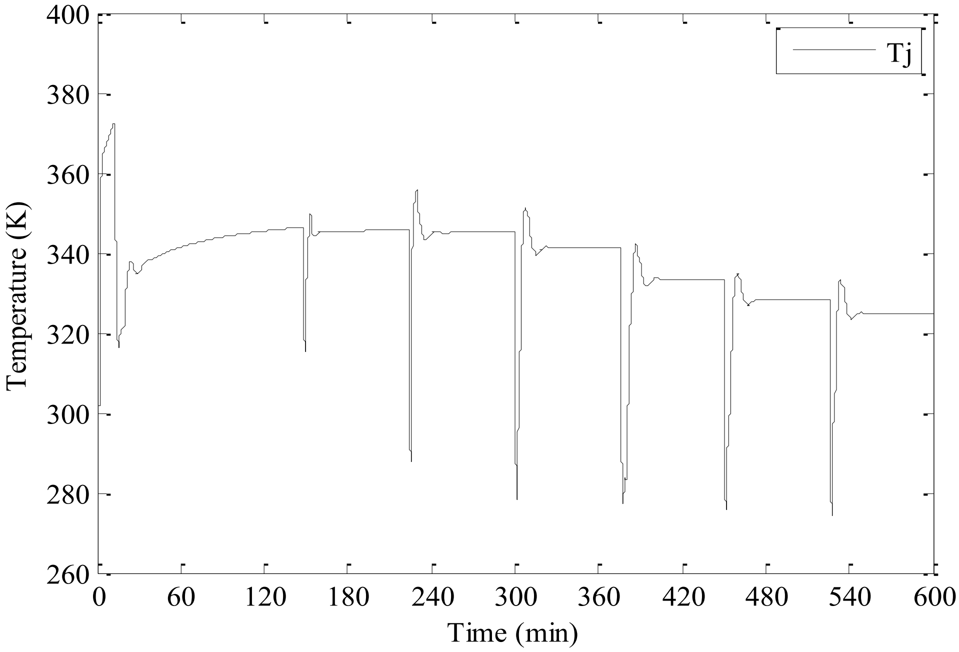


(b)

Figure 5.7 The temperature control using PID control under the parameter mismatch case (-30% U): (a) the control variable (T_r) and (b) the manipulated variable (T_{jsp})

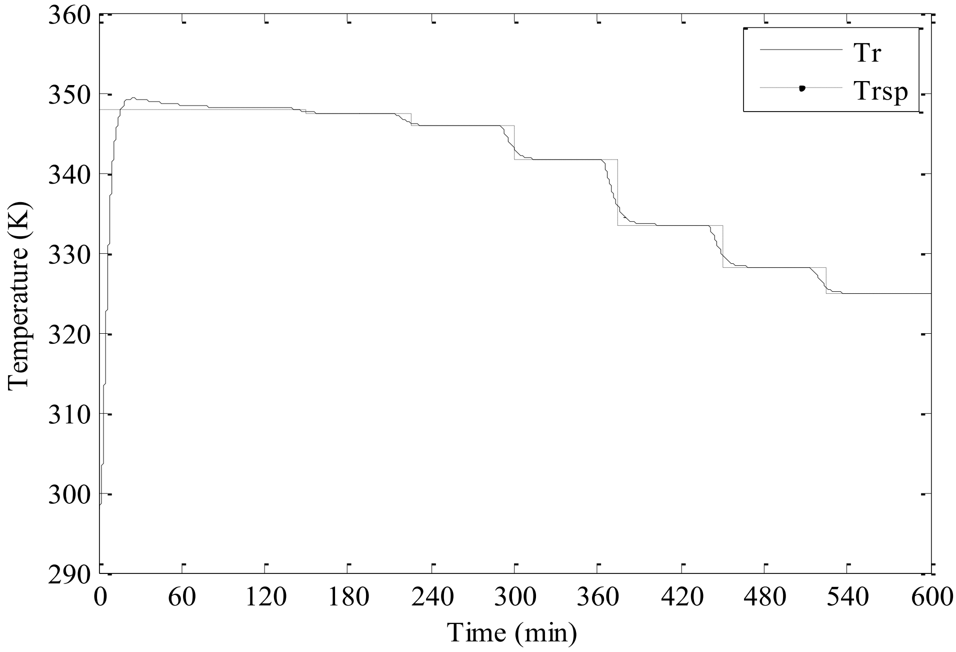


(a)

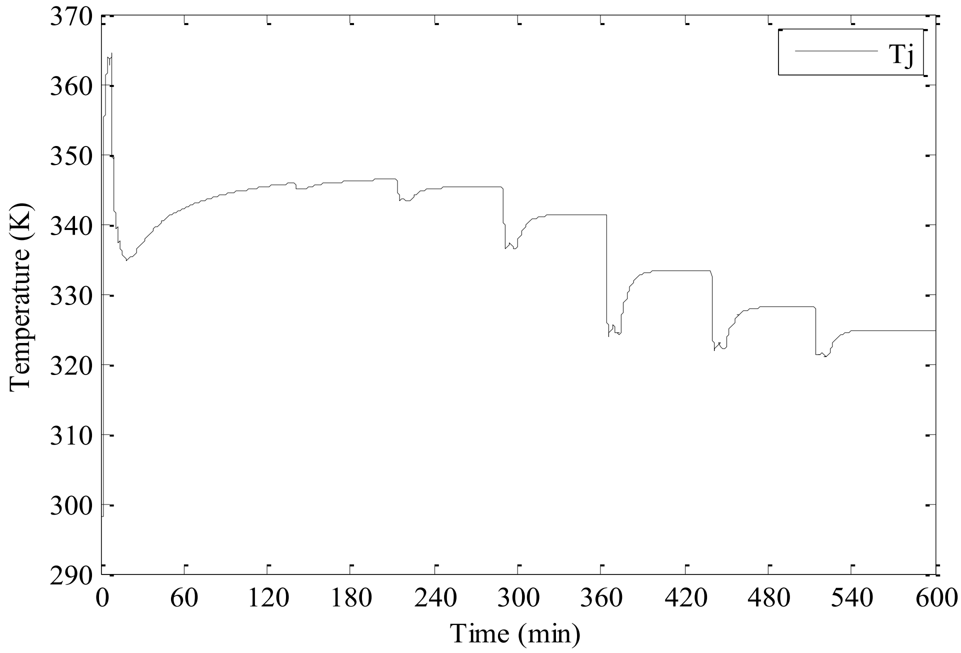


(b)

Figure 5.8 The temperature control using NNDIC under the parameter mismatch case (-30% U): (a) the control variable (T_r) and (b) the manipulated variable (T_{jsp})

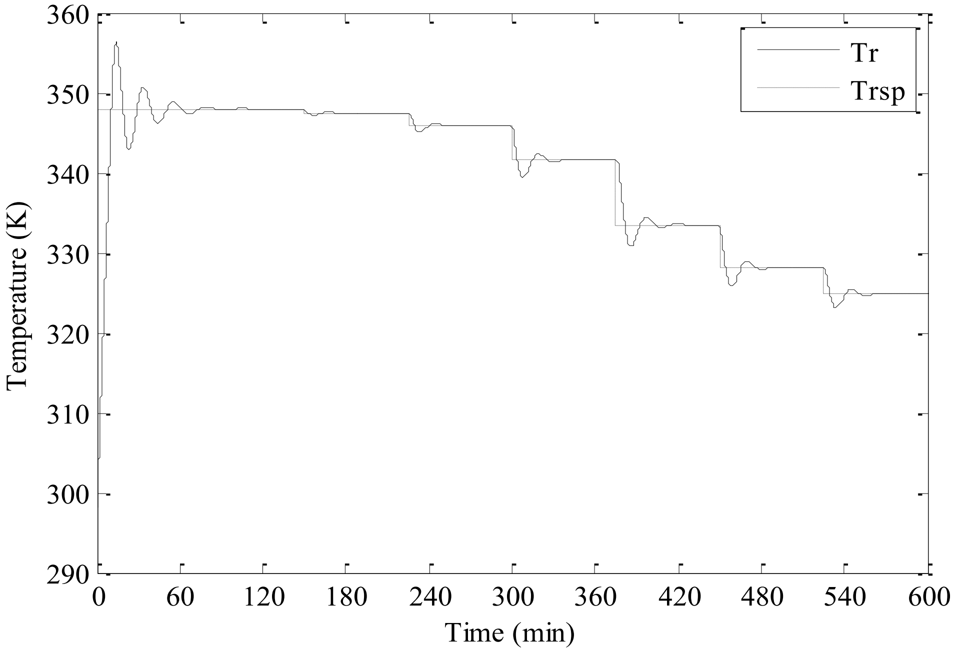


(a)

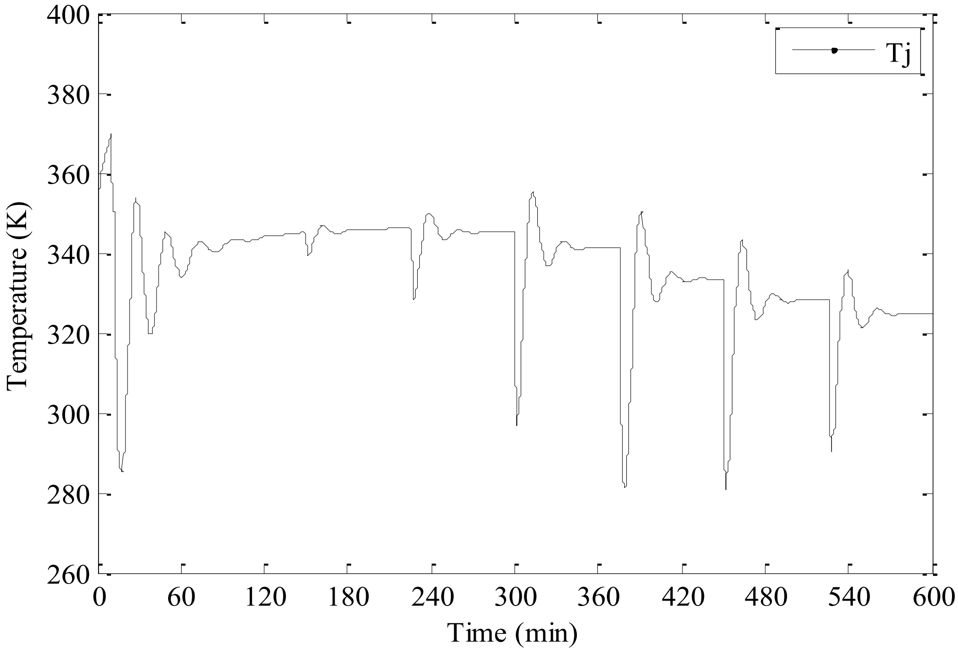


(b)

Figure 5.9 The temperature control using NNMPC under the parameter mismatch case (+30% k_1, -30% k_2, +30% ΔH and -30% U): (a) the control variable (T_r) and (b) the manipulated variable (T_{jsp})

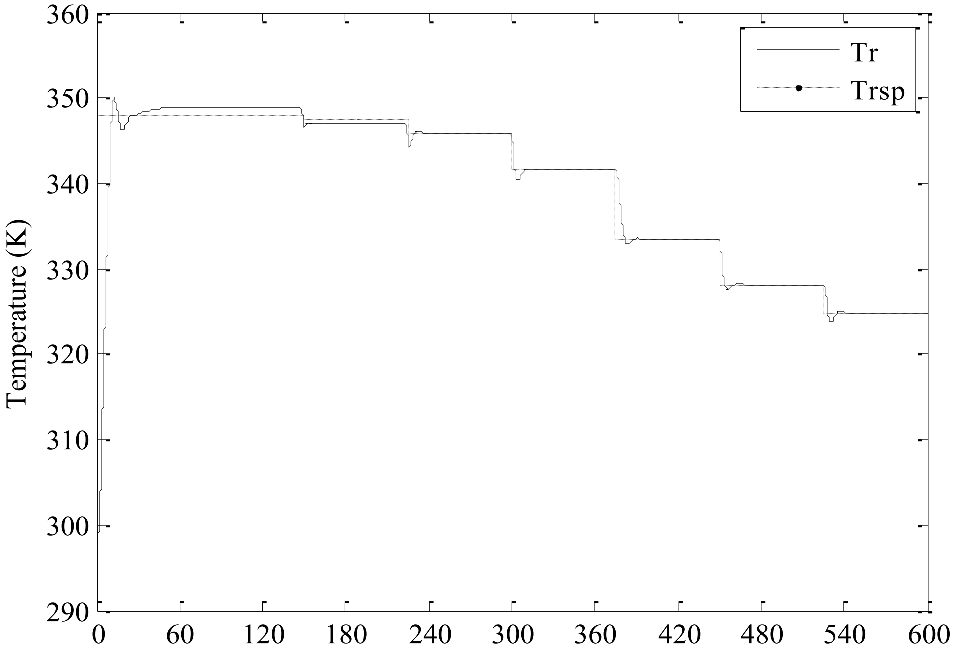


(a)

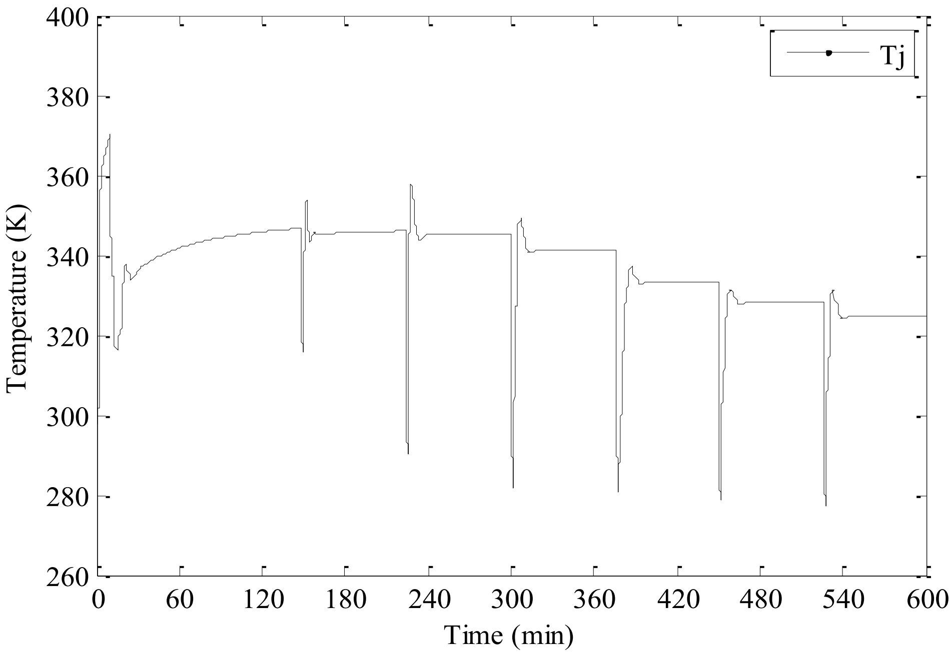


(b)

Figure 5.10 The temperature control using PID control under the parameter mismatch case (+30% k_1, -30% k_2, +30% ΔH and -30% U): (a) the control variable (T_r) and (b) the manipulated variable (T_{jsp})



(a)



(b)

Figure 5.11 The temperature control using NNDIC under the parameter mismatch case (+30% k_1, -30% k_2, +30% ΔH and -30% U): (a) the control variable (T_r) and (b) the manipulated variable (T_{jsp})

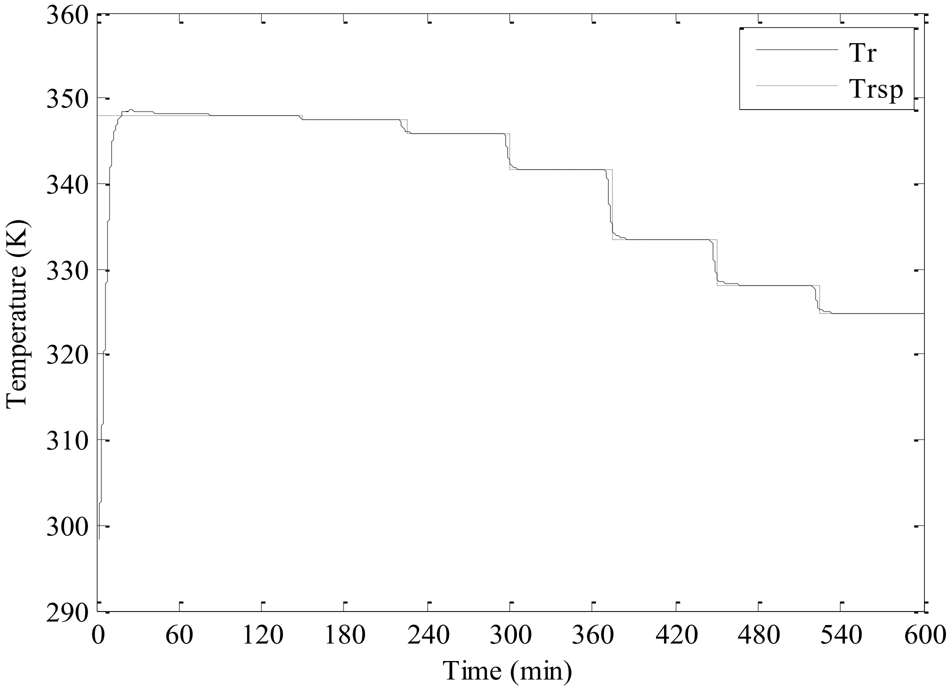
In this part, this research presents the performance of the proposed controllers in a best case. For the best case, the manipulated variable is extremely adjusted and the IAE is the least value. The proposed NNMPC and conventional PID controllers are new tuned to a best case. After that, the obtained NNMPC and PID controller are applied to control the process in certainty and uncertainty cases

First, the simulations are presented in a nominal case. Figure 5.12 and 5.13 show the control of the temperature using NNMPC and conventional PID controllers, respectively. These figures indicate that the both controls can bring the temperature closely to set point with an overshoot and an oscillation. Nevertheless, the conventional PID controller gives more an overshoot and an oscillation than the NNMPC does. For the comparison of the response of the manipulated variable, the manipulated variable adjustment of the NNMPC controller is smoother than the conventional PID controller.

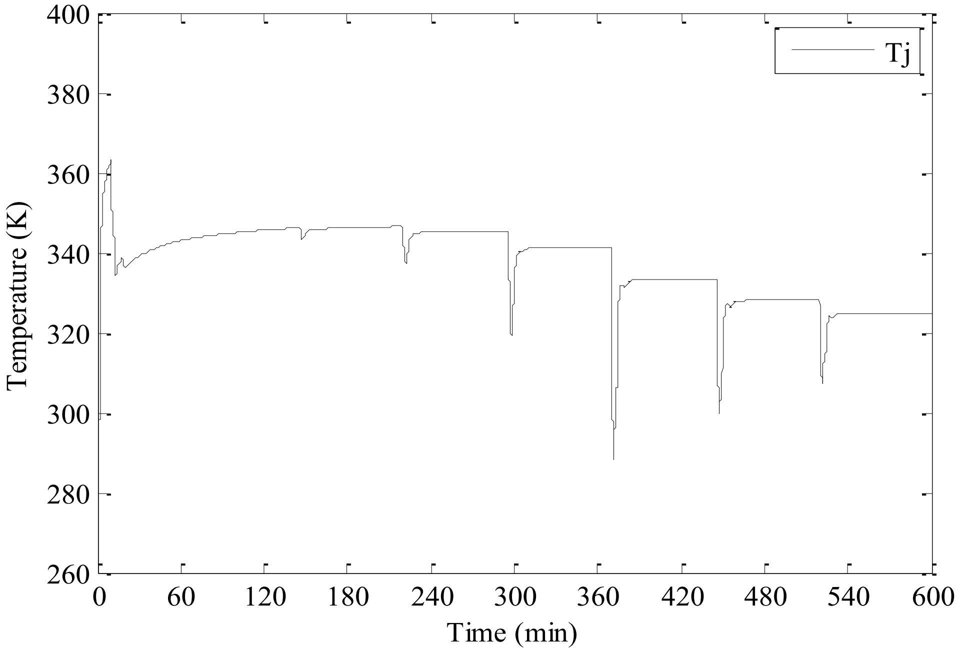
Next, the simulations are investigated in parameters mismatch cases. For the figure 5.14-5.17, they indicate that the NNMPC controller gives more robust and gives better control performance than the PID controller, similar to the nominal case study. The robustness of the NNMPC controller can be explained by the fact that the obtained neural network forward model for the use in the NNMPC controller is trained with the wide range of operating conditions whereas the PID control cannot handle the parameter mismatch as it is based on a nominal condition. Table 5.2 summarizes the control performances of the NNMPC and conventional PID control

Table 5.2 Performances indices of proposed controllers for the best cases

Cases	IAE values		
	NNMPC	PID	NNDIC
Nominal	324.6958	337.0042	422.4777
+30%k_1	323.2163	335.0225	421.2645
-30%k_2	324.1069	336.8489	422.0130.
+30%k_1, -30%k_2	323.1168	335.1431	421.6259
+30%ΔH	322.5920	333.4170	419.5642
-30%U	342.8060	405.4955	456.2135
+30%k_1, -30%k_2, +30%ΔH, -30%U	348.3206	407.4132	462.7851

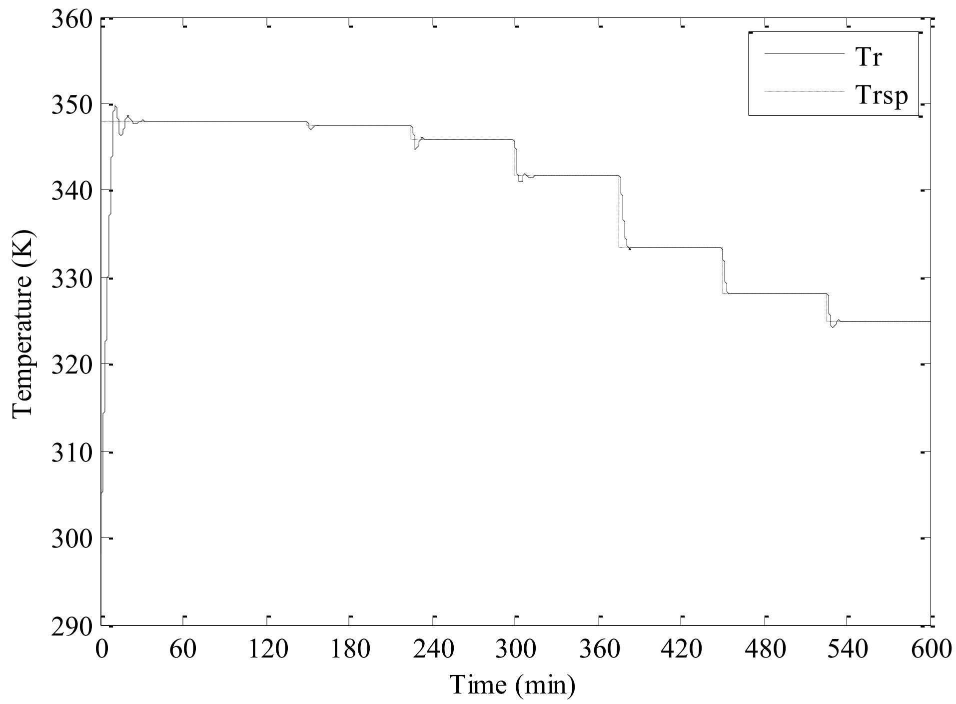


(a)

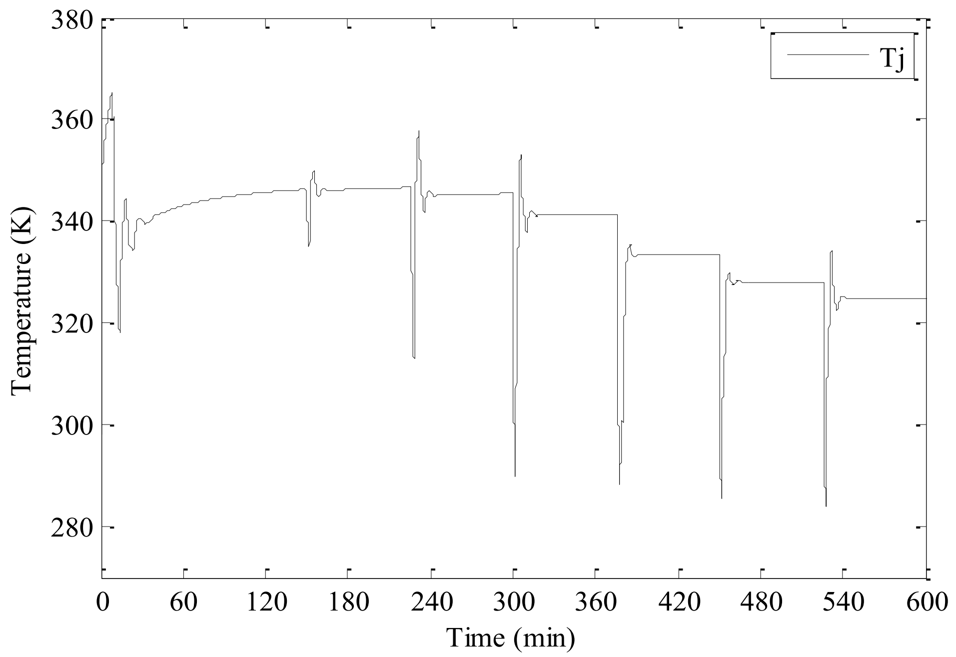


(b)

Figure 5.12 The temperature control using the best case of NNMPC under the nominal case: (a) the control variable (T_r) and (b) the manipulated variable (T_{jsp})

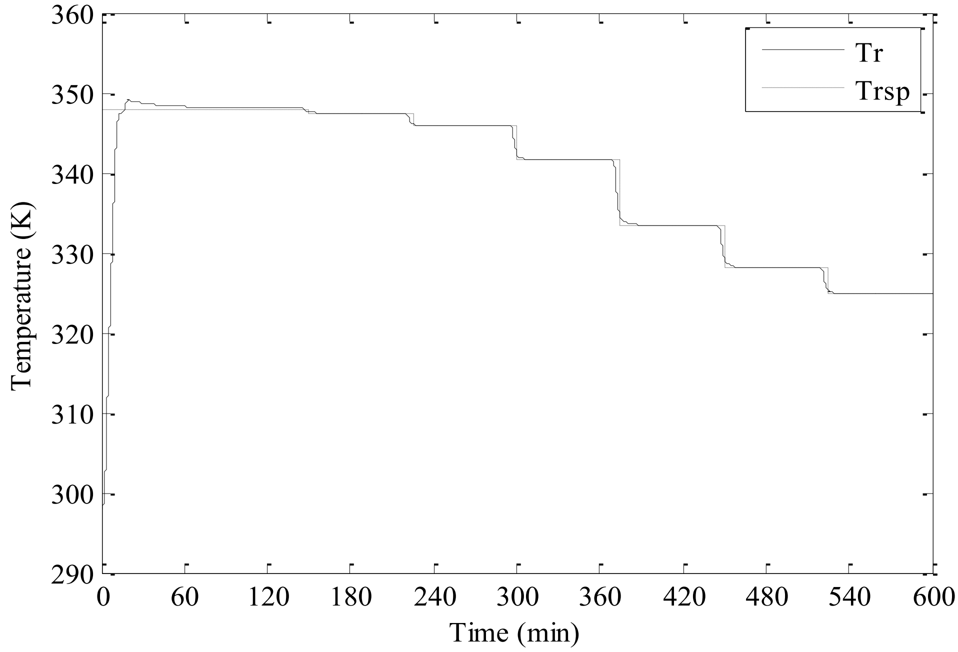


(a)

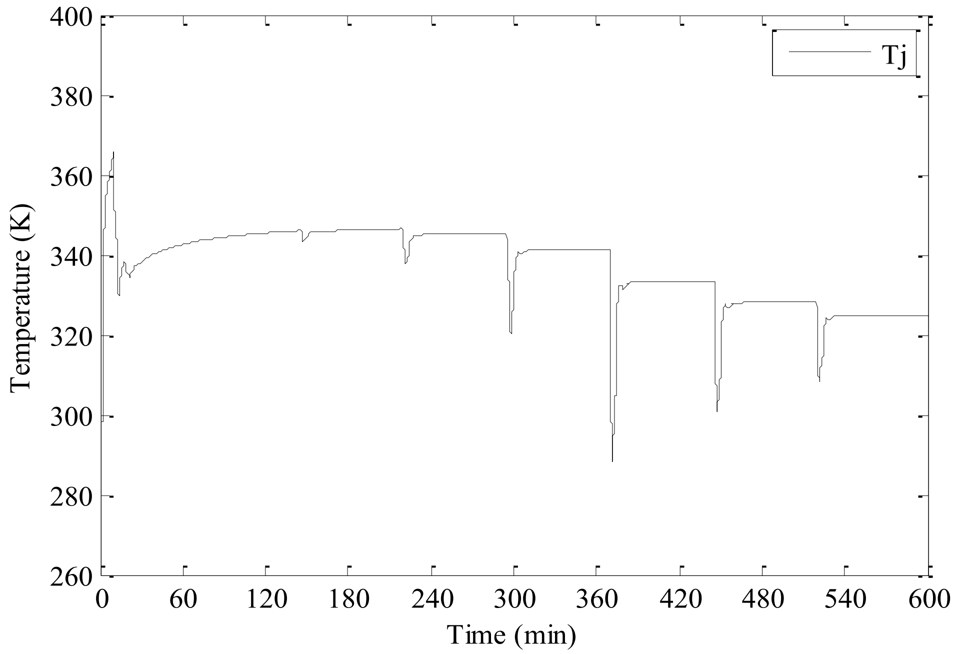


(b)

Figure 5.13 The temperature control using the best case of PID control under the nominal case: (a) the control variable (T_r) and (b) the manipulated variable (T_{jsp})

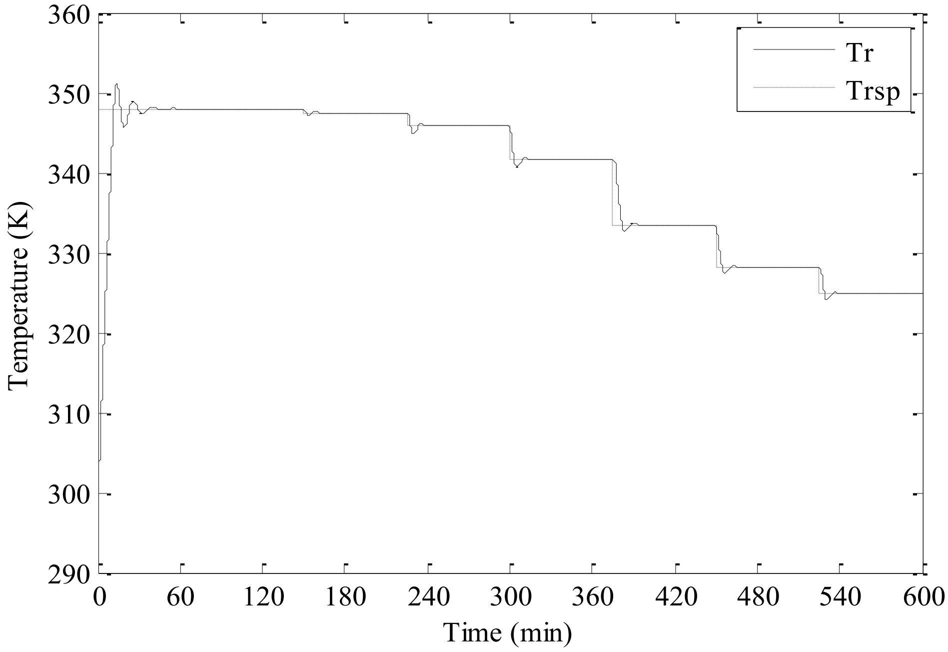


(a)

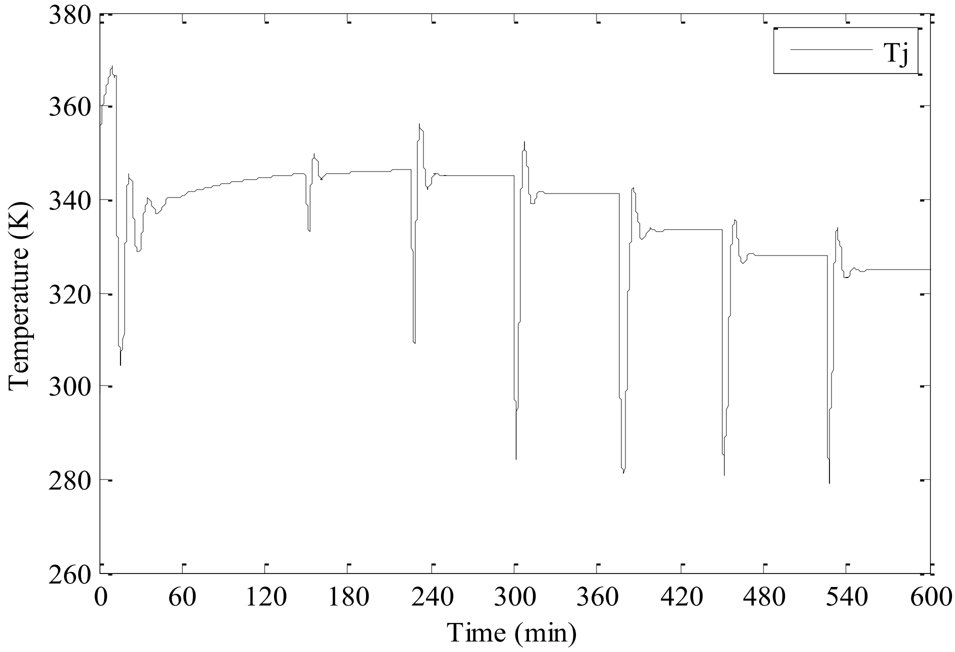


(b)

Figure 5.14 The temperature control using the best case of NNMPC under the parameter mismatch case (-30% U): (a) the control variable (T_r) and (b) the manipulated variable (T_{jsp})

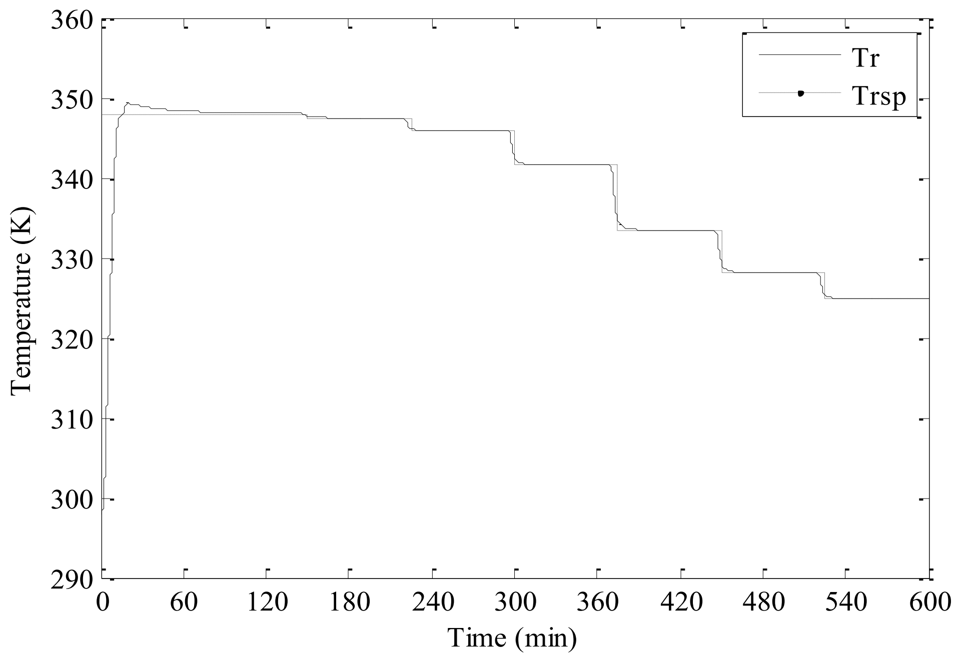


(a)

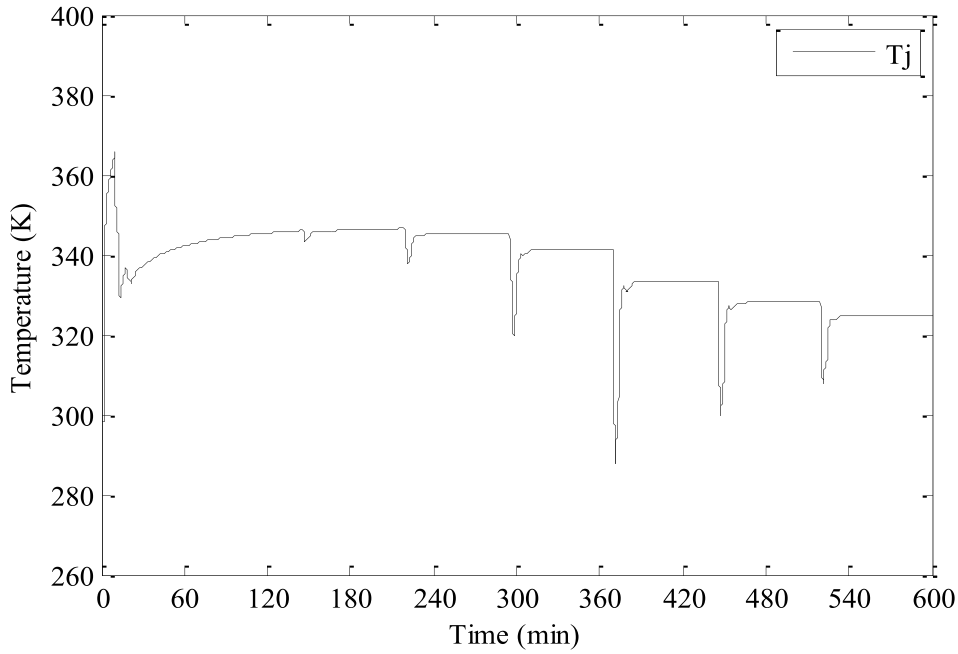


(b)

Figure 5.15 The temperature control using the best case of PID control under the parameter mismatch case (-30% U): (a) the control variable (T_r) and (b) the manipulated variable (T_{jsp})

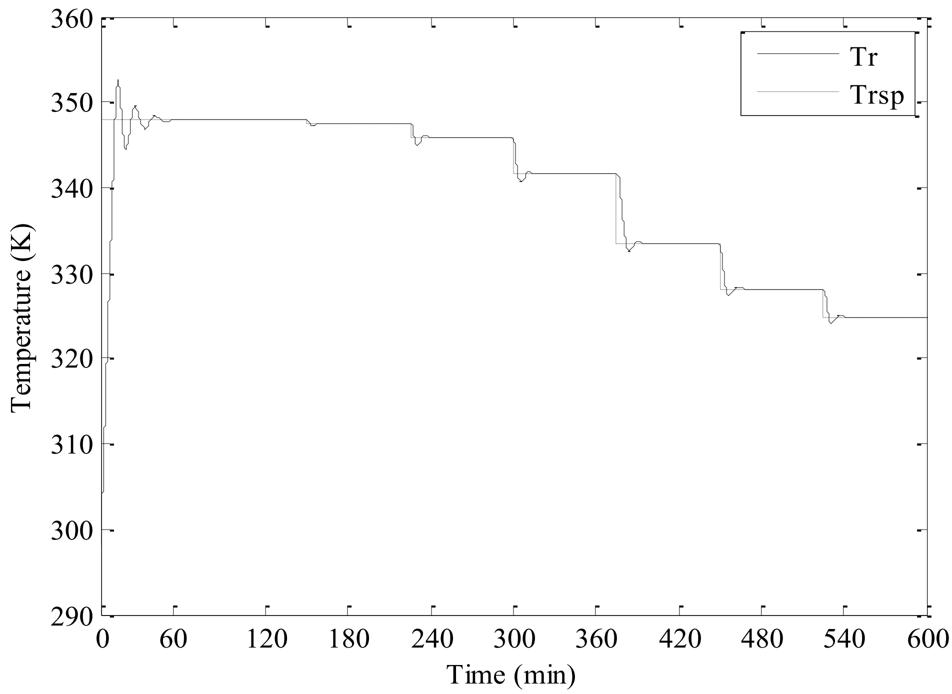


(a)

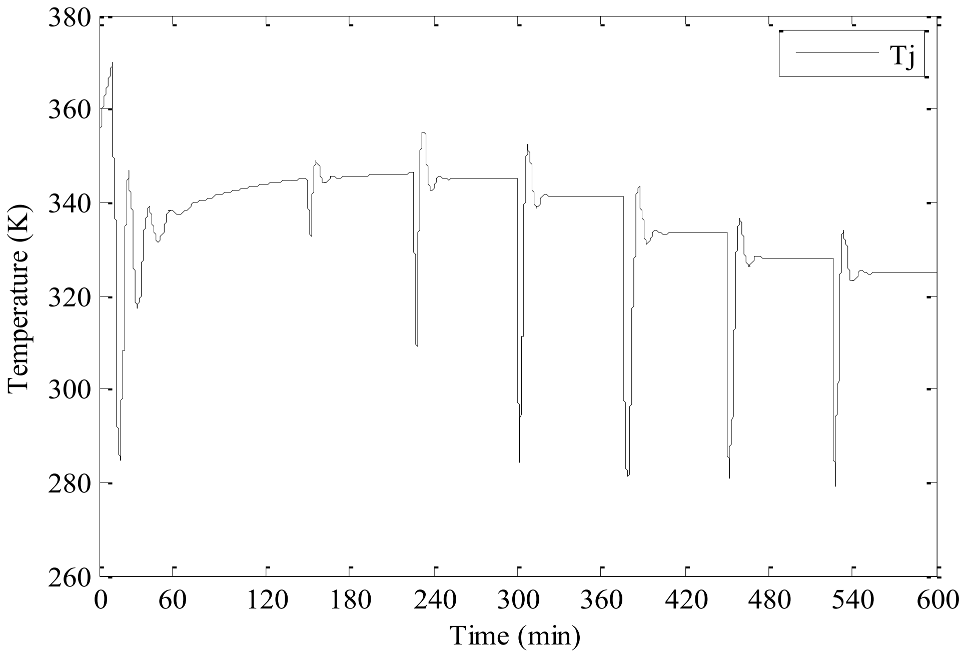


(b)

Figure 5.16 The temperature control using the best case of NNMPC under the parameter mismatch case (-30% k_1, -30% k_2, +30% ΔH and -30% U): (a) the control variable (T_r) and (b) the manipulated variable (T_{jsp})



(a)



(b)

Figure 5.17 The temperature control using the best case of PID control under the parameter mismatch case (+30% k_1, -30% k_2, +30% ΔH and -30% U): (a) the control variable (T_r) and (b) the manipulated variable (T_{jsp})

CHAPTER VI

CONCLUSIONS

An objective of this research is aimed at carrying out process modeling for the prediction of the process outputs and controller design of a based neural network in a batch reactor for a MMA production.

This work presents a neural network modeling for the prediction of a concentration profile and a temperature in the batch reactor. An obtained neural network modeling is applied to predict state variables over a predictive horizon within a model predictive control algorithm for searching the optimal control actions via successive quadratic programming (SQP). Two different types of nonlinear controller based presented neural network are the neural network direct inverse control (NNDIC) and a neural network based model predictive control (NNMPC). Robustness tests of the proposed controllers are studied with respect to the change of operating parameters. In addition, an off-line dynamic optimization approach is applied to find out an optimal operating temperature to achieve maximizing the MMA product at specified final time.

6.1 Neural Network Forward and Inverse Models

For the neural network models, the neural network models have been developed based on the Lenvenberg-Marquardt training algorithm with tan-sigmoid and linear functions as the activation function in the hidden layer and the output layer, respectively. The optimal structure of the neural network is based on MSE between testing and validating data. In the neural network modeling (the neural network forward model), the optimal structure consists of 8 nodes in the input layer, 6 nodes in the first hidden layer, 8 nodes in the second hidden layer and 3 nodes in the output layer. This structure gives the best structure network for the prediction. In the neural network inverse model, the network is applied to be a controller for controlling the reactor temperature by the prediction of the manipulated variable (the set point of the jacket temperature, T_{jsp}). The optimal structure consists of 8 nodes in the input layer, 4 nodes in the first hidden layer, 8 nodes in the second hidden layer and a node in the output layer. The structure gives the best structure for the prediction of the manipulated variable.

6.2 Neural Network based Model Predictive Control (NNMPC)

The neural network based model predictive control is tested with respect to changes in process parameters. It has been found that the NNMPC can bring the controlled variable to about its set point without an oscillations and an offset in all cases studies. The performance of the NNMPC controller is compared to the performance of the conventional PID controller and the performance of the NNDIC controller (another neural network model based control). For the comparison, the NNMPC are more robust than the conventional PID controller and the NNDIC controller and give better results in cases studies. For the comparison between NNDIC and PID controllers in term of an offset, the NNDIC controller give small offset but the PID controller has not an offset. For the response of the manipulated variable, the manipulated variable adjustment of the NNMPC controller is the smoothest among the NNDIC and PID controllers in all cases studies.

REFERNCES

Agarwal, M. A systemematic classification of neural network based control. IEEE Control Systems. 17.2(1997): 75-93

Alexandridis, A. P., Siettos, C. l., Sarimveis, H. K.,Boudouvis, A. G., and Bafas, G. V. Modeling of nonlinear process dynamics using Kohonen's neural networks fuzzy systems and Chebyshev series. Comp. Chem. Eng. 26(2002): 479-486.

Alippi, C., and Piuri, V. Neural modeling of dynamic systems with nonmeasurable state variables. IEEE. Trans. Instr. Meas. 6(1999): 1073-1080.

Arrpornwichanop., P. Kittisupakorn., and Mohamed Azian Hussaun. Model Based Control Strategies for a Chemical Batch Reactor with Exothermic Reactions. Korean Journal Chemical Engineering., 19(2) (2002): 221-226.

Bequette, B. W. Nonlinear control of chemical processes. Ind. Eng. Chem. Res. 30(1991): 1391-1413.

Bloch, G., and Denoeux, T. Neural networks for process control and optimization; Two industrial applications. ISA Trans. 42(2003).

Bonvin, D., 1998. Optimal operation of batch reactors. Journal of Process Control 8(5–6), 355–368.

Breeman, A. J. N. van, and Veelenturf, L. P. J. Neural adaptive feedback linearization control. J. A. 37(1996): 65-71.

Bryson, A., Ho, Y., 1981. Applied Optimal Control: Optimization, Estimation, and Control, (revised Edition). Taylor & Francis, London.

Bruckner, S., and Rudolph, S. Neural networks applied to smart structure control. Stuttgart: SPIE Aerosense (2002).

Carrasco, E., Banga, J., 1997. Dynamic optimization of batch reactors using adaptive stochastic algorithms. Industrial and Engineering Chemical Research 36: 2252–2261.

Chang, J. S., and Huang, K. L. Preformance study of control strategies for trajectory tracking problem of batch reactors. Can. J. Chem. Eng. 72(1994): 906-919.

Charoenniyom, P.kittisupakorn and W. Daosud, Neural Network Modeling and Optimization of the Methyl Methacrylate Production Process for Esterification Reaction in a Batch Reactor, Pure and Applied Chemistry International Conference, January 5-7, 2011.

Chawat Siwaranon. A Neural Network Modelling and Control for Etherification Hybrid Process. Master of Engineering Thesis, Chulalongkorn University. (2005).

Dadebo, S., McAuley, K., 1995. Dynamic optimization of constrained chemical engineering problems using dynamic programming. Computers and Chemical Engineering 19(5): 513–525.

Dale E. Seborg, Thomas F. Edgar and Duncan A. Mellichamp, Process dynamics and control, 2nd edition. United States of America: John Wiley & Sons, 2004.

Daosud. Neural Network based Model Predictive Control for A Steel Pickling Process. Doctor of Engineering Thesis, Chulalongkorn University. (2006).

Demuth, H., and Beale, M. Neural network toolbox for use with MATLAB: User's guide, version 3.0: MathWorks (1998).

Dias, F. M., and Mota, A. M. Comparison between different control strategies using neural networks. Protugal (2002)

Dong Hai-fang, Li Zeng-xi, Wang Zheng-ping, Zhao Wei, Diao Yan-yan, Zhang Suo-jiang. The Intermetallic Catalysts for Oxidative Esterification of Methacrolein to Methyl Methacrylate. The Chinese Journal of Process Engineering. 1009-606X(2006)04-0661-05.

Eaton, J., Rawlings, J., 1990. Feedback control of chemical processes using on-line optimization techniques. Computers and Chemical Engineering 14(4/5): 469–479.

Feng, X., and Huang, M. Studies of a membrane reactor: esterification facikitated by pervaporation. Chem. Eng. Sci. 51(1996): 4673-4679.

Fink, A., Topfer, S., and lsermann, R. Nonlinear model based control with local linear neuro-fuzzy models. App. Mech. 72(2003): 911-922.

Frangu, L., Caraman, S., and Ceanga, E. Model based predictive control using neural network for bioreactor process control. Romania, Dunarea de Joe University, Department of Automatic Control and Electronics (2000).

Gill, P., Henriques, J., Dourado, A., and Rammos, H. D. Non-linear predictive control based on a recurrent neural network. Protugal, Universidade de Coimbra, Informatics Engineering Department, CISUC (1999).

Grosman, B., and Lewin, D. R. Automated nonlinear model predictive control using genetic programming. Comp. Chem. Eng. (2001).

Gupta, M. M., Jin, L., and Homma, N. Static and dynamic neural networks: fundamentals to advanced theory. Wiley (2003).

Helbig, A., Abel, O., Marquardt, W., 1998. Model predictive control for on-line optimization of semi-batch reactors. Proceedings of the 1998 American Control Conference, Philadelphia, PA, pp. 1695–1699.

Henson, M. A., and Seborg, D. E. Nonlinear Process Control. NJ Prentice Hall. 1997.

Hornik, K. M., Stinchcombe, M., and While, H. Multilayer feedforward networks are universal approximators. Neu. Net. 2.5(1989): 359-366.

I. M. Mujtaba., N. Aziz., and M. A. Hussain. Neural Network based Modeling and Control in Batch Reactor. Chemical Engineering Research and Design. 84(2006): 635-644.

James. J. Spivey, Makarand R. Gogate, Joseph R. Zoeller, and Richard D. Colberg. Novel Catalysts for the Environmentally Friendly Synthesis of Methyl Methacrylate. Ind. Eng. Chem. Res. 36(1997): 4600-4608.

Juuso, E. K. Modeling and simulation in advanced control. Finland: Sim-Serv. (2006)

Kershenbaum, L. S., and Kittisupakorn, P. The use of a partially simulated exothermic (PARSEX) reactor for experimental testing of control algorithms. Trans IChemE. 72(1994): 55-63.

Kim, I. H., Fregene, K., Lee, D. H., Oh, T. S., and Wang, D. W. L. Neural network based system identification and controller synthesis for an industrial sewing machine. J. Cont. Aut. Sys. 2.1(2004): 81-91.

Kittisupakorn et al., Neural Network Based Model Predictive Control for a Steel Pickling Process, Journal of Process Control, 19 (2009) 579-590.

Kittisupakorn, P. Tangteerasunun and P. Thitiyasook, Dynamic Neural Network Modeling for Hydrochloric Acid Recovery Process, Korean J. Chem. Eng., 22(2005): 813-821.

Kocijan, J., and Smith, R. M. Nonlinear predictive control with a Gaussian process model. Springer LNCS. 3355(2005): 185-200.

Koichi Nagai. New developments in the production of methyl methacrylate. Journal of Applied Catalysis A: General 221(2001): 367-377.

K. Konakom, P. Kittisupakorn, and I. M. Mujtaba, Batch control improvement by model predictive control based on multiple reduced-models, Chemical Engineering Journal, 145(2008): 129-134.

Lee, K., Chin, I.-S., Lee, H., Lee, J., 1999. Model predictive control technique combined with iterative learning for batch processes. A.I.Ch.E. Journal 45(10): 2175–2187.

Lim, S. Y., Park, B., Hung, F., Sahimi, M., and Tsotsis, T. T. Design issues of pervaporation membrane reactors for esterification. Chem. Eng. Sci. 57(2002): 4933-4946.

Linda Thanasinthana. Neural Network based Optimal Control Strategy for A Batch Crystallization. Master of Engineering Thesis, Chulalongkorn University. (2007).

Loh, A. P., and Fong, K. F. Neural network modeling and control strategies for a pH process. Elsevier: J. Proc. Cont. 5. 6(1995): 355-362.

Matouq, M., Tagawa, T., and Goto, S. Comboned process for production of methyl tert-butyl ether from tert-butyl alcohol and methanol. J. Chem. Eng. Jpn. 27(1994): 302-306.

Murray, R., Neumerkel, D., and Sbarbaro, D. Neural network for modeling and control of non-linear dynamic system. Proceeding of 1992 IEEE international symposium on intelligent control. (1992): 404-409.

Ngamjaruskochakorn. Modeling of Temperature change of Liquid Steel in BOF by Neural Network. Master of Engineering Thesis, Chulalongkorn University. (1997).

Nueaklong, P. Kittisupakorn and W. Daosud, Dynamic and Control of Heat Exchanger System in Hard Chrome Electroplating Using Neural Network, International Conference on Computing and Information Technology, 11-12 May 2011.

Omidvar, O. M., and Elliott, D. L., eds. Neural systems for control. Elsevier: ISBN: 0125264305. (1997).

Perry, R. H., and Green, D. W. Perry's Chemical Engineers' Handbook. NY: 7th edition, McGraw-Hill (1997).

Tangteerasunun. Neural Network Model for The Prediction of Acid Contration and Steel Ion obtained from Hydrochloric Acid Recovery System by Ion Exchange. Master of Engineering Thesis, Chulalongkorn University. (2004).

Thampasato, P. Kittisupakorn and W. daosud, Neural Network Modeling and Optomization of a Batch Heating/Cooling Evaporative Crystallization, International Conference on Computing and Information Technology, 11-12 May 2011.

Petcherdsak, J. Use of multilayered feedforward networks for system identification, function approximation, and advance control. Master of engineering Thesis, Department of Chemical Engineering, Faculty of Engineering, Chulalongkorn University. (1999).

Riggs, J. B., and Rhinehart, R. R. Comparison between two nonlinear process model based controllers. Comp. Chem. Eng. 14(1998): 1075-108.

Limpornchajaroen. Use of Neural Network as Nonlinear Controllers. Master of Engineering Thesis, Chulalongkorn University. (1996).

Scott, M. G., and Ray, W. H. Neural network process models based on linear model structure. Neu. Comp. 6(1994): 718-738.

Silvia Curteanu. Direct and Inverst Neural Network Modeling in Free Radical Polymerization. Central European Science Journal CEJC 2(2004): 113-140.

Teresa Witczak, Jerzy Skrzypek. Liquid-Phase Esterification of Methacrylic Acid with Mathanol Catalyzed by Heteropolyacid. Internation journal of Chemical Reactor Engineering (2010).

Wahab, A. K. A., and Sulaiman, M. Z. Temperature reactors in chemical production processes and the application to the pervaporation assisted esterification. Chem. Eng. Technol. 19(1996): 117-126.

White, D. A., and Sofge, D. A., eds, The handbook of intelligent control. NY:Van Nostrand Reinhold (1992).

Widrow, B., and Rumelhart, D. E., and Lehr, M. A. Neural network: Application in industry. Business and science. J. A. 35.2(1994): 17-27.

Windrow, B., and Walach, E. Adaptive inverse control. NJ: Prentice-Hall (1996).

Wilarat Saeyang. Neural Network based Model for Prediction of Solid Percentge of Shampoo. Doctor of Engineering Thesis, Chulalongkorn University. (2005).

Yu, D. L., and Gomm, J. B. Implementation of neural network predictive control to a multivariable chemical reactor. Elsevier: Cont. Eng. Prac. 11(2003): 1315-1323.

Yu, D. L., and Gomm, J. B., and Williams, D. Neural model input selection for a MIMO chemical process. Elsevier, Eng. App. Art. Int. 13(2003): 15-23.

Zilouchian and M. Jamshidi, Intelligent control systems using soft computing methodologies. United States of America: CRC Press LLC, 2001.

Appendices

APPENDIX A

Mean Square Error of Neural Network Modeling

Table A.1 Mean squared error value of the neural network forward model: 1 layer

Number of Nodes in Hidden Layer	Mean square error (MSE)			
	Training	Testing 1	Testing 2	Validating
1	9.5786 $\times10^{-2}$	9. 7416 $\times10^{-2}$	7.7134 $\times10^{-2}$	8.6567 $\times10^{-2}$
2	2.3435 $\times10^{-4}$	2. 7492 $\times10^{-4}$	1.3454 $\times10^{-4}$	2.0988 $\times10^{-4}$
3	2.2113 $\times10^{-4}$	4.4761 $\times10^{-4}$	2.9788 $\times10^{-4}$	4.3245 $\times10^{-4}$
4	3.6755 $\times10^{-4}$	4.0032 $\times10^{-4}$	4.1234 $\times10^{-4}$	4.3464 $\times10^{-4}$
5	6.9030 $\times10^{-4}$	6.2344 $\times10^{-4}$	5.4443 $\times10^{-4}$	5.6432 $\times10^{-4}$
6	2.3431 $\times10^{-4}$	3.5566 $\times10^{-4}$	4.3254 $\times10^{-4}$	4.5687 $\times10^{-4}$
7	1.1090 $\times10^{-4}$	3.0250 $\times10^{-4}$	3.2354 $\times10^{-4}$	4.4554 $\times10^{-4}$
8	1.4403 $\times10^{-4}$	1.3455 $\times10^{-4}$	2.2356 $\times10^{-4}$	1.7678 $\times10^{-4}$
9	1.9090 $\times10^{-4}$	5.6609 $\times10^{-4}$	3.5797 $\times10^{-4}$	4.9745 $\times10^{-4}$
10	3.6500 $\times10^{-4}$	3.9032 $\times10^{-4}$	2.4577 $\times10^{-4}$	4.3651 $\times10^{-4}$
11	3.3562 $\times10^{-4}$	3.9475 $\times10^{-4}$	3.5600 $\times10^{-4}$	3.0120 $\times10^{-4}$
12	3.4565 $\times10^{-4}$	3.6992 $\times10^{-4}$	3.2349 $\times10^{-4}$	3.0887 $\times10^{-4}$
13	4.2134 $\times10^{-4}$	4.8825 $\times10^{-4}$	5.7756 $\times10^{-4}$	4.0013 $\times10^{-4}$
14	1.0650 $\times10^{-4}$	1.5998 $\times10^{-4}$	2.7809 $\times10^{-4}$	2.9889 $\times10^{-4}$
15	4.9618 $\times10^{-4}$	5.3233 $\times10^{-4}$	2.6568 $\times10^{-4}$	3.9807 $\times10^{-4}$
16	5.8334 $\times10^{-4}$	5.8896 $\times10^{-4}$	4.6981 $\times10^{-4}$	5.2526 $\times10^{-4}$
17	3.2134 $\times10^{-4}$	3.3085 $\times10^{-4}$	4.3687 $\times10^{-4}$	2.2786 $\times10^{-4}$
18	6.6576 $\times10^{-4}$	5.6358 $\times10^{-4}$	5.1120 $\times10^{-4}$	4.5189 $\times10^{-4}$
19	1.9032 $\times10^{-4}$	1.5475 $\times10^{-4}$	4.2656 $\times10^{-4}$	3.4345 $\times10^{-4}$
20	3.4355 $\times10^{-4}$	3.5677 $\times10^{-4}$	2.6546 $\times10^{-4}$	2.4099 $\times10^{-4}$

Table A.2 Mean squared error value of the neural network forward model: 2 layers

Nodes in 1st Hidden Layer	Nodes in 2nd Hidden Layer	Mean square error (MSE)			
		Training	Testing 1	Testing 2	Validating
2	2	6.2529 $\times10^{-4}$	5.4073 $\times10^{-4}$	7.7144 $\times10^{-4}$	8.2185 $\times10^{-4}$
2	4	4.0820 $\times10^{-4}$	5.9280 $\times10^{-4}$	3.0967 $\times10^{-4}$	2.3346 $\times10^{-4}$
2	6	7.0933 $\times10^{-4}$	8.4642 $\times10^{-4}$	7.4567 $\times10^{-4}$	5.0784 $\times10^{-4}$
2	8	6.8520 $\times10^{-4}$	5.7090 $\times10^{-4}$	3.9443 $\times10^{-4}$	4.5382 $\times10^{-4}$
2	10	2.7087 $\times10^{-4}$	1.2706 $\times10^{-4}$	1.0884 $\times10^{-4}$	1.5852 $\times10^{-4}$
2	12	6.8413 $\times10^{-4}$	4.4850 $\times10^{-4}$	6.5733 $\times10^{-4}$	3.2184 $\times10^{-4}$
2	14	4.5995 $\times10^{-4}$	3.9011 $\times10^{-4}$	2.6334 $\times10^{-4}$	1.0231 $\times10^{-4}$
2	16	7.5828 $\times10^{-4}$	6.0665 $\times10^{-4}$	7.5013 $\times10^{-4}$	6.6334 $\times10^{-4}$
2	18	5.6457 $\times10^{-4}$	4.1259 $\times10^{-4}$	6.4480 $\times10^{-4}$	4.8006 $\times10^{-4}$

2	20	7.6637×10^{-4}	7.2852×10^{-4}	5.7268×10^{-4}	8.6440×10^{-4}
4	2	6.2294×10^{-4}	7.8241×10^{-4}	8.3036×10^{-4}	4.6949×10^{-4}
4	4	1.7404×10^{-4}	1.8776×10^{-4}	1.2378×10^{-4}	3.9644×10^{-4}
4	6	2.3457×10^{-4}	4.1450×10^{-4}	3.5493×10^{-4}	6.8922×10^{-4}
4	8	3.1740×10^{-4}	5.0300×10^{-4}	6.0152×10^{-4}	8.4624×10^{-4}
4	10	3.0933×10^{-4}	2.6248×10^{-4}	1.3111×10^{-4}	1.3115×10^{-4}
4	12	1.8288×10^{-4}	1.4148×10^{-4}	3.3854×10^{-4}	1.4021×10^{-4}
4	14	1.9206×10^{-4}	1.4267×10^{-4}	9.7454×10^{-5}	2.0360×10^{-4}
4	16	2.2262×10^{-4}	2.7477×10^{-4}	1.6185×10^{-4}	7.1939×10^{-4}
4	18	6.4585×10^{-5}	6.9090×10^{-5}	6.1910×10^{-5}	1.9805×10^{-4}
4	20	1.6621×10^{-4}	1.5215×10^{-4}	1.7694×10^{-4}	9.3966×10^{-5}
6	2	5.4745×10^{-5}	4.1042×10^{-5}	6.9301×10^{-4}	9.8691×10^{-5}
6	4	6.8196×10^{-5}	8.5523×10^{-5}	8.8480×10^{-5}	1.2185×10^{-4}
6	6	1.3913×10^{-4}	6.5683×10^{-4}	3.2410×10^{-4}	2.3346×10^{-4}
6	8	4.6068×10^{-5}	3.5618×10^{-5}	3.1400×10^{-5}	5.0784×10^{-5}
6	10	7.1280×10^{-5}	5.7535×10^{-5}	6.8735×10^{-5}	1.5382×10^{-4}
6	12	6.5745×10^{-4}	5.8256×10^{-5}	5.3498×10^{-5}	1.5852×10^{-4}
6	14	1.9816×10^{-4}	1.3638×10^{-4}	9.6655×10^{-5}	1.2184×10^{-4}
6	16	1.9718×10^{-4}	1.4605×10^{-4}	9.0586×10^{-5}	1.0231×10^{-4}
6	18	6.2529×10^{-5}	8.4073×10^{-5}	7.7144×10^{-5}	1.5140×10^{-4}
6	20	1.0820×10^{-4}	1.9280×10^{-5}	1.0967×10^{-5}	1.8006×10^{-4}
8	2	5.0933×10^{-4}	6.4642×10^{-4}	8.1634×10^{-4}	1.6440×10^{-4}
8	4	6.8520×10^{-5}	7.7090×10^{-5}	6.9443×10^{-5}	1.2502×10^{-4}
8	6	8.7087×10^{-5}	9.2706×10^{-5}	9.0884×10^{-5}	8.9644×10^{-5}
8	8	8.8413×10^{-5}	8.4850×10^{-5}	7.5733×10^{-5}	8.8922×10^{-5}
8	10	1.5995×10^{-4}	1.9011×10^{-4}	2.6334×10^{-5}	8.4624×10^{-5}
8	12	7.5828×10^{-5}	4.0665×10^{-5}	6.1739×10^{-5}	9.3115×10^{-5}
8	14	5.6457×10^{-5}	5.5119×10^{-5}	4.0453×10^{5}	7.4021×10^{-5}
8	16	1.6637×10^{-4}	1.6459×10^{-4}	9.2069×10^{-5}	9.0360×10^{-5}
8	18	2.2294×10^{-4}	2.8420×10^{-4}	8.3679×10^{-5}	7.1939×10^{-5}
8	20	1.7404×10^{-4}	2.5738×10^{-4}	3.0733×10^{-5}	6.9805×10^{-5}
10	2	5.3457×10^{-4}	4.0052×10^{-4}	1.2175×10^{-4}	5.2174×10^{-4}
10	4	3.1740×10^{-4}	5.2554×10^{-4}	1.2121×10^{-4}	3.5744×10^{-4}
10	6	3.0933×10^{-4}	7.2229×10^{-5}	1.1650×10^{-4}	4.6949×10^{-4}
10	8	4.9800×10^{-4}	4.2100×10^{-4}	9.7891×10^{-5}	3.7615×10^{-4}
10	10	4.2628×10^{-4}	4.3351×10^{-4}	1.1502×10^{-4}	4.2451×10^{-4}
10	12	1.5844×10^{-4}	3.6389×10^{-4}	9.3881×10^{-5}	6.8659×10^{-5}
10	14	1.2502×10^{-4}	4.1491×10^{-4}	7.7340×10^{-5}	3.0287×10^{-4}
10	16	1.1500×10^{-4}	4.6842×10^{-4}	1.5140×10^{-4}	3.2381×10^{-4}
10	18	1.4129×10^{-4}	3.2787×10^{-4}	1.1203×10^{-4}	9.4260×10^{-5}
10	20	9.4605×10^{-4}	3.7843×10^{-4}	1.2363×10^{-4}	5.0285×10^{-4}
12	2	4.8624×10^{-4}	3.9988×10^{-4}	9.4797×10^{-5}	3.3344×10^{-4}
12	4	9.2037×10^{-4}	6.4067×10^{-4}	1.0254×10^{-4}	3.8528×10^{-4}
12	6	3.9684×10^{-4}	5.1423×10^{-4}	1.0826×10^{-4}	7.8522×10^{-5}
12	8	4.9800×10^{-4}	3.1400×10^{-5}	1.6654×10^{-4}	5.6357×10^{-5}
12	10	1.2729×10^{-4}	4.4733×10^{-4}	9.8436×10^{-5}	4.7591×10^{-4}

12	12	2.0818×10^{-4}	5.0714×10^{-4}	9.1454×10^{-5}	4.0753×10^{-4}
12	14	2.9318×10^{-4}	3.9812×10^{-4}	1.1694×10^{-4}	1.0542×10^{-4}
12	16	5.1542×10^{-5}	3.1046×10^{-4}	1.2650×10^{-4}	5.5441×10^{-5}
12	18	4.8622×10^{-4}	3.8621×10^{-4}	8.2770×10^{-5}	4.6587×10^{-4}
12	20	1.7672×10^{-4}	9.7370×10^{-5}	1.0771×10^{-4}	2.0799×10^{-4}
14	2	8.7652×10^{-5}	7.7552×10^{-5}	1.1302×10^{-4}	3.9844×10^{-4}
14	4	1.8385×10^{-4}	8.8658×10^{-5}	1.0265×10^{-4}	9.7967×10^{-5}
14	6	4.8467×10^{-4}	4.8709×10^{-4}	8.0784×10^{-5}	4.4031×10^{-4}
14	8	2.1629×10^{-4}	3.1720×10^{-4}	9.2247×10^{-5}	3.2905×10^{-4}
14	10	1.0408×10^{-4}	4.0403×10^{-4}	1.1828×10^{-4}	3.8770×10^{-4}
14	12	3.8339×10^{-4}	3.8233×10^{-4}	7.8397×10^{-4}	2.9575×10^{-4}
14	14	3.4309×10^{-4}	9.4203×10^{-5}	8.9855×10^{-5}	3.6253×10^{-4}
14	16	3.9727×10^{-4}	3.9122×10^{-4}	1.0248×10^{-4}	3.8485×10^{-4}
14	18	1.8947×10^{-4}	2.8247×10^{-4}	9.0845×10^{-5}	2.3393×10^{-4}
14	20	1.6016×10^{-4}	2.6112×10^{-4}	1.0359×10^{-4}	6.8459×10^{-5}
16	2	9.5653×10^{-5}	8.5453×10^{-5}	1.0343×10^{-4}	6.5517×10^{-5}
16	4	3.2983×10^{-4}	4.2780×10^{-4}	1.0431×10^{-4}	4.8618×10^{-4}
16	6	3.6352×10^{-4}	3.6851×10^{-4}	1.5301×10^{-4}	7.5842×10^{-5}
16	8	5.4652×10^{-5}	8.4954×10^{-5}	1.0281×10^{-4}	3.2267×10^{-4}
16	10	4.0352×10^{-4}	4.0056×10^{-4}	1.0272×10^{-4}	3.6646×10^{-4}
16	12	6.1652×10^{-5}	6.1050×10^{-5}	9.1581×10^{-5}	5.6664×10^{-4}
16	14	6.9886×10^{-5}	9.9089×10^{-4}	9.4660×10^{-5}	3.5383×10^{-4}
16	16	4.5997×10^{-4}	4.5972×10^{-4}	8.6754×10^{-5}	8.8580×10^{-5}
16	18	4.4308×10^{-4}	4.4371×10^{-4}	1.5856×10^{-4}	7.2281×10^{-5}
16	20	3.9436×10^{-4}	4.9536×10^{-4}	9.2633×10^{-5}	6.3591×10^{-5}
18	2	3.9737×10^{-4}	3.9533×10^{-4}	1.5382×10^{-4}	4.1911×10^{-4}
18	4	4.5727×10^{-4}	4.5726×10^{-4}	1.0616×10^{-4}	5.0000×10^{-4}
18	6	3.4625×10^{-4}	4.4829×10^{-4}	9.6690×10^{-5}	5.0074×10^{-4}
18	8	4.2473×10^{-4}	3.2970×10^{-4}	1.0355×10^{-4}	2.4854×10^{-4}
18	10	4.1223×10^{-4}	4.1924×10^{-4}	1.2687×10^{-4}	4.2735×10^{-4}
18	12	4.8253×10^{-4}	3.8354×10^{-4}	1.4835×10^{-4}	7.5669×10^{-5}
18	14	2.5462×10^{-4}	3.5263×10^{-4}	8.3397×10^{-4}	2.9933×10^{-4}
18	16	2.2082×10^{-4}	4.2381×10^{-4}	1.3474×10^{-4}	2.4766×10^{-4}
18	18	1.4812×10^{-4}	5.4318×10^{-4}	1.6494×10^{-4}	3.8136×10^{-4}
18	20	6.4304×10^{-5}	7.4501×10^{-4}	1.8411×10^{-4}	2.6666×10^{-4}
20	2	6.2370×10^{-5}	6.2677×10^{-5}	1.8718×10^{-4}	3.3733×10^{-4}
20	4	6.2599×10^{-5}	6.2793×10^{-5}	1.9158×10^{-4}	4.2568×10^{-4}
20	6	2.4720×10^{-4}	3.4826×10^{-4}	1.7104×10^{-4}	3.6774×10^{-4}
20	8	4.4108×10^{-4}	8.4898×10^{-4}	1.5034×10^{-4}	6.2389×10^{-5}
20	10	4.2184×10^{-4}	3.2682×10^{-4}	9.4437×10^{-5}	2.8097×10^{-4}
20	12	2.0365×10^{-4}	5.0465×10^{-4}	1.5928×10^{-4}	4.5992×10^{-4}
20	14	1.5559×10^{-4}	3.5459×10^{-4}	9.6201×10^{-5}	4.4297×10^{-4}
20	16	3.2816×10^{-4}	3.2316×10^{-4}	1.5887×10^{-4}	4.4891×10^{-4}
20	18	2.8948×10^{-4}	4.8248×10^{-4}	1.5846×10^{-4}	5.1564×10^{-4}
20	20	1.7004×10^{-4}	3.7204×10^{-4}	1.6605×10^{-4}	3.4227×10^{-4}

Table A.3 Mean squared error value of the neural network inverse model: 1 layer

Number of Nodes in Hidden Layer	Mean square error (MSE)			
	Training	Testing 1	Testing 2	Validating
1	9.4761 x10^{-3}	7.1000 x10^{-3}	9.1000 x10^{-3}	8.5602 x10^{-3}
2	8.9042 x10^{-3}	3.7000 x10^{-3}	7.8000 x10^{-3}	6.6003 x10^{-3}
3	9.4761 x10^{-5}	8.3969 x10^{-5}	8.6630 x10^{-5}	7.5356 x10^{-5}
4	4.0194 x10^{-4}	6.2261 x10^{-5}	7.1207 x10^{-5}	7.6768 x10^{-5}
5	5.9030 x10^{-5}	3.3392 x10^{-5}	1.0475 x10^{-5}	2.3419 x10^{-5}
6	4.5889 x10^{-5}	2.1934 x10^{-5}	5.5400 x10^{-5}	4.0243 x10^{-5}
7	5.4250 x10^{-5}	2.9000 x10^{-4}	3.1309 x10^{-4}	1.6003 x10^{-4}
8	1.4403 x10^{-4}	1.0000 x10^{-4}	2.1019 x10^{-4}	2.8950 x10^{-4}
9	7.1729 x10^{-5}	8.5006 x10^{-5}	7.6124 x10^{-5}	8.5906 x10^{-5}
10	8.2035 x10^{-5}	7.6441 x10^{-5}	6.6324 x10^{-5}	7.9535 x10^{-5}
11	3.9475 x10^{-5}	3.4129 x10^{-5}	1.3714 x10^{-5}	2.4572 x10^{-5}
12	7.7896 x10^{-5}	9.0778 x10^{-5}	6.4565 x10^{-5}	7.4689 x10^{-5}
13	7.8825 x10^{-5}	9.8954 x10^{-5}	7.1012 x10^{-5}	8. 9839 x10^{-5}
14	7.0650 x10^{-5}	7.8000 x10^{-5}	6.9000 x10^{-5}	8.3590 x10^{-5}
15	4.9613 x10^{-5}	1.4000 x10^{-4}	3.1967 x10^{-4}	2.2909 x10^{-4}
16	5.8896 x10^{-4}	1.3000 x10^{-4}	4.1967 x10^{-4}	2.8374 x10^{-4}
17	3.3085 x10^{-4}	2.3000 x10^{-4}	5.1955 x10^{-4}	3.7747 x10^{-4}
18	5.6358 x10^{-4}	4.6014 x10^{-4}	3.1034 x10^{-4}	3.2485 x10^{-4}
19	8.9375 x10^{-4}	2.9000 x10^{-4}	1.6000 x10^{-4}	2.2709 x10^{-4}
20	3.2787 x10^{-4}	1.5810 x10^{-4}	8.6938 x10^{-5}	1.2352 x10^{-4}

Table A.4 Mean squared error value of the neural network inverse model: 2 layers

Nodes in 1^{st} Hidden Layer	Nodes in 2^{nd} Hidden Layer	Mean square error (MSE)			
		Training	Testing 1	Testing 2	Validating
2	2	7.0040 x10^{-4}	4.2024 x10^{-4}	2.6232 x10^{-4}	1.5595 x10^{-4}
2	4	4.7799 x10^{-4}	2.8679 x10^{-4}	2.7789 x10^{-4}	1.4778 x10^{-4}
2	6	2.0386 x10^{-4}	1.2232 x10^{-4}	2.7350 x10^{-4}	1.5320 x10^{-4}
2	8	2.4496 x10^{-4}	1.4697 x10^{-4}	2.7809 x10^{-4}	1.4651 x10^{-4}
2	10	1.9686 x10^{-4}	1.1812 x10^{-4}	3.2515 x10^{-4}	1.4278 x10^{-4}
2	12	6.8753 x10^{-4}	4.1252 x10^{-4}	2.7508 x10^{-4}	1.4363 x10^{-4}
2	14	9.9904 x10^{-4}	5.9942 x10^{-4}	2.5380 x10^{-4}	1.5039 x10^{-4}
2	16	3.4325 x10^{-4}	2.0595 x10^{-4}	2.7002 x10^{-4}	1.4774 x10^{-4}
2	18	2.2607 x10^{-4}	1.3564 x10^{-4}	2.9417 x10^{-4}	1.4395 x10^{-4}
2	20	6.6468 x10^{-4}	3.9881 x10^{-4}	2.5076 x10^{-4}	1.5963 x10^{-4}
4	2	9.3216 x10^{-4}	5.5930 x10^{-4}	1.3571 x10^{-4}	9.1772 x10^{-5}
4	4	4.9567 x10^{-4}	2.9740 x10^{-4}	1.4068 x10^{-4}	1.0103 x10^{-4}
4	6	6.1516 x10^{-4}	3.6909 x10^{-4}	1.0923 x10^{-4}	4.9637 x10^{-4}
4	8	3.6992 x10^{-5}	2.0184 x10^{-5}	1.0795 x10^{-5}	2.0489 x10^{-5}
4	10	3.5701 x10^{-4}	2.1421 x10^{-4}	1.2490 x10^{-4}	5.0353 x10^{-4}

4	12	1.7585×10^{-4}	1.0551×10^{-4}	1.1716×10^{-4}	5.8241×10^{-5}
4	14	1.0601×10^{-4}	6.3607×10^{-5}	1.0884×10^{-4}	7.3651×10^{-5}
4	16	1.5750×10^{-4}	9.4502×10^{-5}	1.0402×10^{-4}	4.1231×10^{-4}
4	18	3.1696×10^{-4}	1.9018×10^{-4}	8.3286×10^{-5}	2.9449×10^{-4}
4	20	3.8094×10^{-4}	2.2857×10^{-4}	1.3153×10^{-4}	2.7635×10^{-4}
6	2	6.9794×10^{-4}	4.1877×10^{-4}	1.2505×10^{-4}	4.4112×10^{-4}
6	4	1.4605×10^{-4}	8.7632×10^{-4}	9.2063×10^{-5}	5.9253×10^{-5}
6	6	4.9258×10^{-5}	2.9555×10^{-5}	9.9169×10^{-5}	3.5626×10^{-4}
6	8	3.4007×10^{-4}	2.0404×10^{-4}	8.3594×10^{-5}	4.1469×10^{-4}
6	10	4.4512×10^{-4}	2.6707×10^{-4}	9.9698×10^{-4}	7.6653×10^{-5}
6	12	1.8181×10^{-4}	1.0909×10^{-4}	1.1250×10^{-4}	3.1634×10^{-4}
6	14	2.2022×10^{-4}	1.3213×10^{-4}	9.5463×10^{-5}	3.2907×10^{-4}
6	16	5.6656×10^{-5}	3.3994×10^{-4}	9.7872×10^{-5}	3.5789×10^{-4}
6	18	8.2262×10^{-5}	4.9357×10^{-4}	1.0377×10^{-4}	4.4307×10^{-4}
6	20	2.2339×10^{-4}	1.3403×10^{-4}	8.2289×10^{-5}	4.1111×10^{-4}
8	2	6.0140×10^{-4}	3.6084×10^{-4}	1.0927×10^{-4}	3.5943×10^{-4}
8	4	2.6536×10^{-4}	1.5922×10^{-4}	1.0778×10^{-4}	5.6431×10^{-4}
8	6	7.4695×10^{-5}	4.4817×10^{-5}	1.1331×10^{-4}	6.1708×10^{-5}
8	8	6.5543×10^{-5}	3.9326×10^{-4}	1.0578×10^{-4}	3.9125×10^{-4}
8	10	2.1522×10^{-4}	1.2913×10^{-4}	9.9558×10^{-5}	8.6510×10^{-5}
8	12	1.0943×10^{-4}	6.5657×10^{-5}	9.1739×10^{-5}	3.4965×10^{-4}
8	14	5.4714×10^{-5}	3.2828×10^{-4}	1.0453×10^{-4}	3.3431×10^{-4}
8	16	1.2714×10^{-4}	7.6286×10^{-5}	9.2069×10^{-5}	6.4207×10^{-5}
8	18	9.7238×10^{-5}	5.8343×10^{-5}	1.3679×10^{-4}	4.7914×10^{-4}
8	20	8.6199×10^{-5}	5.1720×10^{-5}	1.0733×10^{-4}	6.6855×10^{-4}
10	2	7.5147×10^{-4}	4.5088×10^{-4}	1.2175×10^{-4}	5.2174×10^{-4}
10	4	5.8674×10^{-5}	3.5204×10^{-4}	1.2121×10^{-4}	3.5744×10^{-4}
10	6	1.5831×10^{-5}	9.4984×10^{-5}	1.1650×10^{-4}	4.6949×10^{-4}
10	8	5.2911×10^{-4}	3.3431×10^{-4}	9.7891×10^{-5}	3.7615×10^{-4}
10	10	1.2988×10^{-4}	7.7926×10^{-5}	1.1502×10^{-5}	4.2451×10^{-4}
10	12	6.6408×10^{-4}	3.9845×10^{-4}	9.3881×10^{-5}	6.8659×10^{-5}
10	14	1.9041×10^{-4}	1.1425×10^{-4}	7.7340×10^{-5}	3.0287×10^{-4}
10	16	1.5285×10^{-4}	9.1708×10^{-5}	1.5140×10^{-4}	3.2381×10^{-4}
10	18	5.4845×10^{-5}	3.2907×10^{-4}	1.1203×10^{-4}	9.4260×10^{-5}
10	20	1.4987×10^{-4}	8.9920×10^{-5}	1.2363×10^{-5}	5.0285×10^{-5}
12	2	5.5831×10^{-4}	3.3499×10^{-4}	1.9243×10^{-4}	4.1261×10^{-5}
12	4	1.2269×10^{-4}	7.3614×10^{-5}	6.8226×10^{-5}	1.0376×10^{-4}
12	6	8.0852×10^{-5}	4.8511×10^{-4}	8.7085×10^{-5}	2.1585×10^{-4}
12	8	1.3576×10^{-4}	8.1458×10^{-5}	5.6357×10^{-5}	8.6791×10^{-5}
12	10	3.2778×10^{-4}	1.9667×10^{-4}	7.1408×10^{-5}	2.7600×10^{-4}
12	12	1.0752×10^{-4}	6.4511×10^{-5}	4.0973×10^{-5}	7.8445×10^{-5}
12	14	8.3939×10^{-5}	5.0363×10^{-5}	9.4262×10^{-5}	4.4522×10^{-4}
12	16	1.7434×10^{-4}	1.0460×10^{-4}	4.8182×10^{-4}	1.6717×10^{-4}
12	18	1.3559×10^{-4}	8.1357×10^{-5}	5.2265×10^{-4}	1.5445×10^{-4}
12	20	1.1010×10^{-4}	6.6062×10^{-5}	9.7875×10^{-5}	1.3012×10^{-4}
14	2	8.7770×10^{-5}	5.2662×10^{-5}	2.5176×10^{-4}	9.1259×10^{-5}

14	4	6.4977×10^{-5}	3.8986×10^{-4}	3.1106×10^{-4}	3.8470×10^{-4}
14	6	7.9484×10^{-5}	4.7690×10^{-4}	3.2230×10^{-4}	1.2840×10^{-4}
14	8	1.1857×10^{-4}	7.1141×10^{-5}	3.2905×10^{-4}	2.9475×10^{-4}
14	10	7.0411×10^{-5}	4.2247×10^{-4}	4.4680×10^{-4}	3.8470×10^{-4}
14	12	1.3309×10^{-4}	7.9855×10^{-5}	7.7704×10^{-5}	3.2672×10^{-4}
14	14	3.3316×10^{-4}	1.9990×10^{-4}	2.7174×10^{-4}	2.4611×10^{-4}
14	16	1.4286×10^{-4}	8.5719×10^{-5}	3.8685×10^{-4}	4.9419×10^{-4}
14	18	1.1457×10^{-4}	6.8743×10^{-5}	6.5082×10^{-5}	4.4843×10^{-4}
14	20	1.2199×10^{-4}	7.3194×10^{-5}	6.8959×10^{-5}	1.9251×10^{-4}
16	2	2.0068×10^{-4}	1.2041×10^{-4}	8.9700×10^{-5}	2.0225×10^{-4}
16	4	2.2099×10^{-4}	1.3259×10^{-4}	3.1902×10^{-4}	7.6555×10^{-5}
16	6	2.2251×10^{-4}	1.3351×10^{-4}	2.5010×10^{-4}	4.6206×10^{-4}
16	8	4.7112×10^{-5}	2.8267×10^{-4}	7.7276×10^{-5}	1.6010×10^{-4}
16	10	9.5744×10^{-5}	5.7446×10^{-5}	2.4378×10^{-4}	8.8669×10^{-5}
16	12	7.3360×10^{-5}	4.4016×10^{-4}	8.4838×10^{-5}	4.6245×10^{-5}
16	14	2.3066×10^{-4}	7.5842×10^{-5}	1.5242×10^{-4}	1.8156×10^{-4}
16	16	1.1306×10^{-4}	6.7839×10^{-5}	1.1610×10^{-4}	3.3682×10^{-4}
16	18	1.3701×10^{-4}	8.2208×10^{-5}	2.3051×10^{-4}	3.2672×10^{-4}
16	20	5.8961×10^{-5}	3.5377×10^{-4}	4.9112×10^{-4}	2.4611×10^{-4}
18	2	4.3475×10^{-5}	2.6085×10^{-4}	1.3251×10^{-4}	1.6551×10^{-4}
18	4	3.4474×10^{-4}	8.8500×10^{-5}	3.3856×10^{-4}	6.4376×10^{-5}
18	6	9.4707×10^{-5}	5.6824×10^{-5}	1.3290×10^{-4}	2.5140×10^{-4}
18	8	6.8695×10^{-5}	4.1217×10^{-4}	7.3321×10^{-5}	1.7390×10^{-4}
18	10	4.2930×10^{-4}	2.5758×10^{-4}	1.6045×10^{-4}	4.7970×10^{-4}
18	12	1.3277×10^{-4}	7.9661×10^{-5}	1.5242×10^{-4}	1.6551×10^{-4}
18	14	3.3128×10^{-4}	1.9877×10^{-4}	4.7351×10^{-4}	1.9690×10^{-4}
18	16	2.1691×10^{-4}	2.6646×10^{-4}	4.6681×10^{-4}	3.7357×10^{-4}
18	18	1.1144×10^{-4}	6.6861×10^{-5}	1.1963×10^{-4}	3.4361×10^{-4}
18	20	1.0398×10^{-4}	6.2389×10^{-5}	1.8634×10^{-4}	1.5868×10^{-4}
20	2	1.1308×10^{-4}	6.7845×10^{-5}	9.5603×10^{-5}	5.5780×10^{-5}
20	4	1.1452×10^{-4}	6.8715×10^{-5}	2.3247×10^{-4}	1.8530×10^{-4}
20	6	3.3860×10^{-4}	3.6754×10^{-4}	5.3702×10^{-5}	9.1618×10^{-5}
20	8	1.7229×10^{-4}	6.2329×10^{-5}	4.1503×10^{-4}	1.9690×10^{-4}
20	10	5.9858×10^{-5}	2.8067×10^{-4}	5.1230×10^{-5}	1.2935×10^{-4}
20	12	1.2713×10^{-4}	7.6277×10^{-4}	2.1521×10^{-4}	1.8777×10^{-4}
20	14	1.3418×10^{-4}	8.0510×10^{-4}	1.9281×10^{-4}	8.9676×10^{-5}
20	16	9.1245×10^{-5}	4.4881×10^{-4}	3.1820×10^{-4}	5.6219×10^{-5}
20	18	4.8430×10^{-4}	5.1564×10^{-5}	8.2275×10^{-5}	3.0956×10^{-4}
20	20	1.0529×10^{-4}	3.4227×10^{-4}	4.7887×10^{-4}	5.0071×10^{-4}

j

APPENDIX B
Process Parameters

The reactions kinetics and physical properties foe the esterification reaction for methyl methacrylate in a batch reactor are shown in Table B.1. Those parameters are referred from Witczak et al. (2010) as well as William L. Luyben (2007)

Table B.1 Esterification and physical properties of MMA system

Symbol	Value	Unit
C_{pA}	167.817	(J/mol K)
C_{pB}	81.080	(J/mol K)
C_{pC}	191.202	(J/mol K)
C_{pD}	1000	(J/mol K)
ρ_A	1015	kg/m^3
ρ_B	791.8	kg/m^3
ρ_C	940	kg/m^3
ρ_D	1000	kg/m^3
ρ_j	1000	kg/m^3
ΔH	-57500	J/mol
MW_A	86.08	kg/kmol
MW_B	32.04	kg/kmol
MW_C	100.12	kg/kmol
MW_D	18.00	kg/kmol
V	0.0025	m
U	274.42	J/sec m^2 K
A	0.05	m^2
V_j	0.001	m^3
R	8.314	kJ/kmol K
C_{cat}	28.9	mol/m^3
F_j	3.5×10^{-5}	m^3/sec

Table B.2 Initial condition of MMA system

Symbol	Value	Unit
$C_A(0)$	15	mol/m^3
$C_B(0)$	45	mol/m^3
$C_C(0)$	0	mol/m^3
$C_D(0)$	0	mol/m^3
$T_r(0)$	298	K
$T_j(0)$	298	K

APPENDIX C

Proportional Integral Derivative Control (PID) Strategy

The PID control strategy is a generic is a generic control loop feedback mechanism (controller) and wildly used in industrial control system. The block diagram of PID controller is shown in figure C.1.

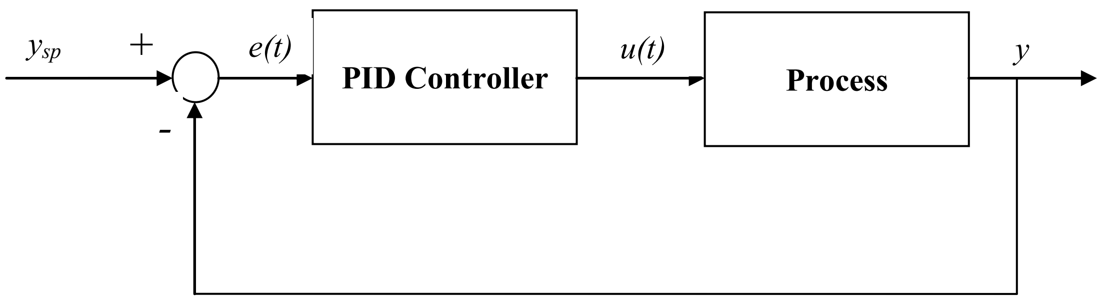


Figure C.1 Closed loop control of PID

Proportional, integral and derivative action can be combined by each of the modes operate in parallel. The parallel form of the PID control algorithm is given by

$$p(t) = p + K_C[e(t) + \frac{1}{\tau_I}\int e(t)dt + \tau_D \frac{de(t)}{dt}] \tag{C.1}$$

where e = ysp – y (C.2)

K_C = controller grain

τ_I = integral time

τ_D = derivative time

u(t) = controller output or manipulated variable

y = process output

y_{sp} = set point

p = bias value, this constant value is the output of the controller when the error equal to zero. Bias value is very often initially set at mid-scale (50% of controller output)

APPENDIX D

Controller Performance Index

For the process control, the performance indices are a very important design tools available to Engineers. One can quantitatively specify a desired system performance and from it mathematically calculate a set of system parameter(s). Formally a performance index is defined as a quantitative measure of the system so that a set of parameters in the system can be adjusted to meet the required specification optimally. There are at least four types of popular indices consisting of ISE, IAE, ITAE and ITSE. They are defined as follows:

1) Integral of the square error (ISE)

$$ISE = \int_0^\infty e^2(t)dt \tag{D.1}$$

2) Integral of the absolute error (IAE)

$$IAE = \int_0^\infty |e(t)|dt \tag{D.2}$$

3) Integral of the time absolute error (ITAE)

$$ITAE = \int_0^\infty t|e(t)|dt \tag{D.3}$$

4) Integral of the time square error (ITSE)

$$ITSE = \int_0^\infty te^2(t)dt \tag{D.4}$$

where *e(t)* is the deviation of the response from the set point.

If the error between output and set point of the system is large, ISE is better than IAE because the errors are squared and thus contribute more to the value of the integral. On the other hand, if the error is small, IAE is better than ISE because when the small errors (less than one) are squared, they become even smaller.

Figure D.1 is shown the characteristics of the step response of a second order underdamped process. The following terms are used to describe the dynamic of underdamped process.

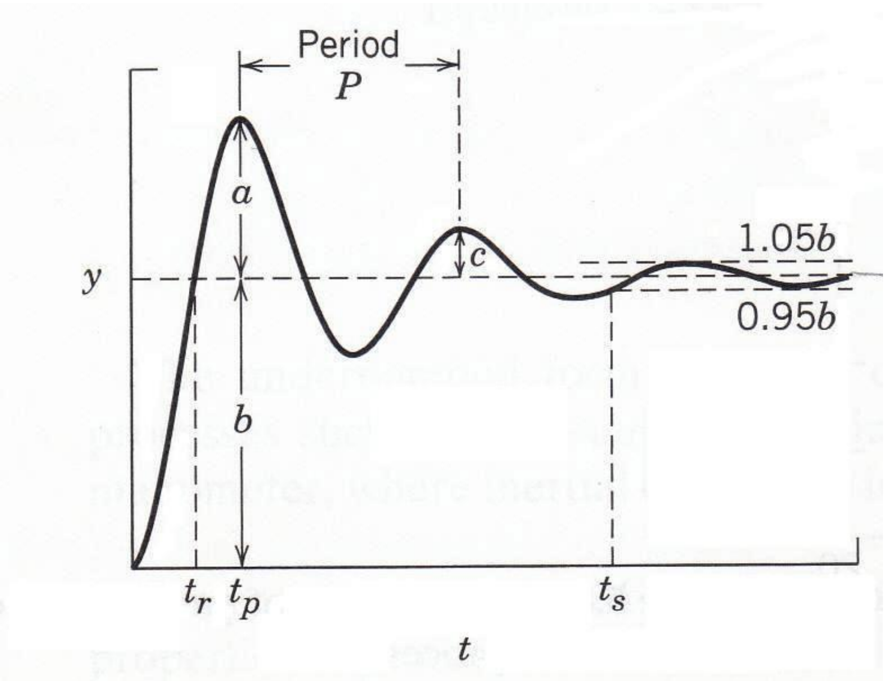


Figure D.1 Performance characteristics for the step response of an underdamped process

1) Rise Time, t_r is the time process output takes to first reach the new steady state value.

2) Time to First Peak, t_p is the time required for the output to reach its first maximum value.

3) Settling Time, t_s is the time required for process output to reach and remain inside a band whose width is equal to $\pm$ 5% of the total change in y for 95% response time.

4) Overshoot, $OS = a/b$ (% overshoot is $100a/b$).

5) Decay Ratio, $DR = c/a$ (where c is the height of second peak).

6) Period of Oscillation, P is the time between two successive peaks or two successive valleys of the response.

APPENDIX E
Euler's Methods

In order to use Euler's Method to generate a numerical solution to an initial value problem of the form:

$$\frac{dy}{dx} = f(x, y) \tag{E.1}$$

$$y(x_0) = y_0 \tag{E.2}$$

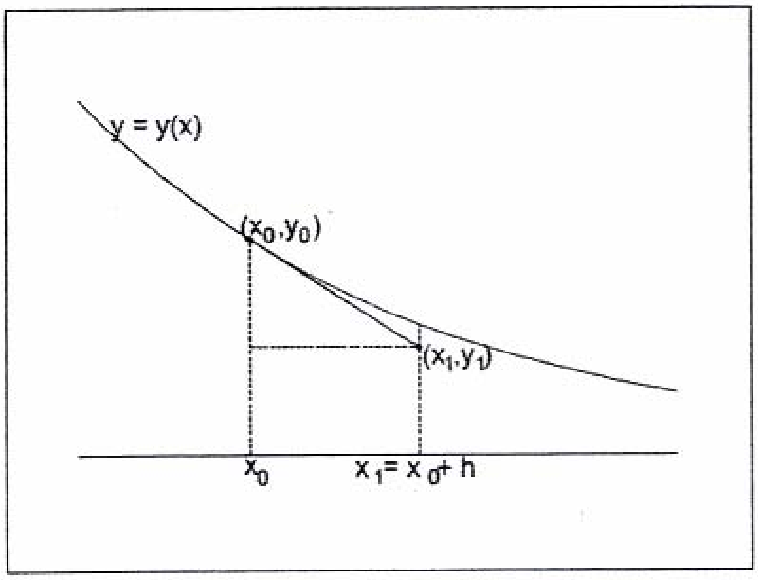


Figure E.1 Euler's method for a range

We decide upon what interval, starting at the initial condition, we desire to find the solution. We chop this interval into small subdivisions of length *h*. Then, using the initial condition as our starting point, we generate the rest of the solution by using the iterative formulas:

$$x_{n+1} = x_n + h \tag{E.3}$$

$$y_{n+1} = y_n f(x_n, y_n) \qquad n = 0,1,2,... \tag{E.4}$$

To find the coordinates of the points in our numerical solution, we terminate this process when we have reached the right end of the desired interval.

VITA

Miss Thanutchaporn Charoenniyom was born in Rayong, on November 24, 1986. After completing high school from Klaeng "Wittayasathaworn" school in 2007, she graduated her Bachelor Degree of Engineering in Chemical Engineering from Rajamangala University of Technology Thanyaburi in 2010. She has continued her study in Master Degree in Control and Systems Engineering, Department of Chemical Engineering, Faculty of Engineering, Chulalongkorn University since 2010 and finished her study in 2012.