Mechanistic, Neural Network, and Intelligent Hybrid Models for a Three-Phase Fluidised-Bed Biofilm Reactor

A thesis presented in partial
fulfillment of the requirements
for the degree of
Doctor of Philosophy
in
Environmental Engineering ,at
Institute of Technology and Engineering
Massey University

Yoon-Seok Hong

1998

This thesis is dedicated to
my grandmother 이용녀.

너는 마음을 다하여 여호와를 의뢰하고
네 명철을 의뢰하지 말라.
너는 범사에 그를 인정하라
그리하면 네 길을 지도하시리라 (잠언 3장 5-6절)

Trust in the Lord with all your heart,
And do not lean on your own understanding.
In all our ways acknowledge Him,
And He will make your paths straight.
(Proverbs 3:5-6)

ABSTRACT

Over the past three decades, considerable amount of research efforts have been undertaken in order to develop a mathematical model for a three-phase fluidised-bed biofilm reactor (TPFBBR). Although biofilm properties such as biofilm thickness and its density are allowed to vary with biofilm growth in the model to simulate the real TPFBBR system, they are assumed to be constant in the majority of models developed for a TPFBBR. The main goal of this thesis is to develop mathematical models incorporating dynamic biofilm growth for a TPFBBR using three different modelling approaches such as a mechanistic model, a neural network model, and an intelligent hybrid model with a neurofuzzy model.

This thesis consists of three parts. Firstly, a dynamic biofilm growth model, which reflects the variation of biofilm thickness and its density in time, is developed. This model is derived from a biomass balance equation and is solved by the method of characteristics. The biofilm detachment model is proposed and incorporated within the dynamic biofilm growth model. The dynamic biofilm growth model with detachment is then combined with a reaction-diffusion model and reactor model to form an integrated model of a TPFBBR. Simulation method of integrated model incorporating the dynamic biofilm growth model is developed. It is observed that results predicted are in good agreement with experimental data and the integrated model proposed provides a valuable tool to predict performance of a TPFBBR.

Secondly, the sequential neural network model, which is composed of two parts, namely, the neural process estimator and the neural process predictor, is developed to describe the task of process estimation and prediction for a TPFBBR. In order to implement the sequential neural network model, multilayer feedforward neural network (MFNN) with cascaded-correlation (C-C) learning and extended Kalman filtering (EKF) learning, and generalized regression neural network (GRNN) are used. Results shows that the sequential neural network model has the feasibility as

intelligent estimators and dynamic predictors and gives considerably good results in process estimation and prediction for a TPFBBR.

Finally, this thesis shows how a combination of both mechanistic and empirical modelling approaches, called a hybrid model, can be implemented and utilised for modelling a TPFBBR. The neurofuzzy model as an empirical part of hybrid model is used to estimate the variation of the biofilm thickness and biofilm density, and is combined with mechanistic model-based reaction-diffusion and axial-dispersion models to predict the dynamic behavior and performance of a TPFBBR according to the variation of biofilm density and biofilm thickness. This hybrid modelling approach due to its flexibility shows a unified framework through incorporation of strong points of both mechanistic and empirical models, and provides a new modelling framework with a great potential to be applied to other types of biofilm reactors.

ACKNOWLEDGEMENT

I would like to express sincere appreciation to Professor Rao Bhamidimarri for his support, encouragement, and outstanding guidance throughout the course of my Ph.D. research. In addition, his trust in me has left a deep impression on me.
I also thank my adviser Dr. Aroon Parshotam for advice, encouragement, and support.

Gratitude is expressed to Massey University for the financial support with Doctoral Scholarship.

The assistance of John Alger in building experimental systems is gratefully acknowledged.

I am greatly indebted to my grandmother, my mother, my farther and my sister for the unconditioned love and support. I also wish to thank parent-in-law for their love and encouragement. I would like to thank my wife Yun-Ah, my son Min-Woo, and my daughter Sue-Ji, who gave me new meaning to my life. In many ways, my wife Yun-Ah is a co-author of this thesis. Her patience, understanding, unshakable faith in me, and help had played a major role in the course of my research work.

Finally, my ultimate thanks go to my Lord God and Savior Jesus Christ. Without the wisdom from Jesus Christ, this work would not have been possible.

Praise and Glorify the Lord Jesus Christ

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 INTRODUCTION

Biofilms, which are a matrix of cells and cellular products attached to a solid surface, are well known to have a great potential to remove organic matter from wastewater. Biofilm systems are complex as a results of a combination of factors, such as bacterial growth, substrate consumption, attachment, external-internal mass transfer of substrate, cell death, detachment, the structure of the support particle, and competition between bacterial species: all of which play a significant role in the overall capacity of the biofilm process. A three-phase fluidized-bed biofilm reactor (TPFBBR) has received considerable attention for use in aerobic wastewater treatment. Generally, a TPFBBR has a number of advantages over suspended-growth systems such as the activated-sludge process. The most important feature of a TPFBBR is that high biomass concentration can be retained in the reactor as biofilms on the support particles, which leads to high reaction rates at low hydraulic retention times. In addition, a TPFBBR has an improved resistance to the change of environmental conditions, more resistance to toxic chemicals and heavy metals, and better process stability relative to suspended-growth systems.

Because of these advantages, a TPFBBR has been demonstrated to outperform other reactor configurations used in wastewater treatment (Lee et al., 1979).

It is necessary to characterise the biofilm properties in order to enhance understanding of the complexities of biofilm processes and to properly design and control a TPFBBR. The biofilm thickness and biofilm density are commonly used to help characterise the biofilm growth on the support particles, and are widely applied for macro-scale biofilm modelling and design purposes in a TPFBBR. Several biofilm models have been developed in which substrate transport to consumption by the biofilm is described (Atkinson, et al., 1967; Atkinson & Daoud, 1970; Lamotta, 1976; Williamson & McCarthy, 1976; Rittmann & McCarty, 1981). In the majority of those models, the biofilm properties such as biofilm thickness and its density were assumed to be constant. Hence, the development of mathematical models incorporating dynamic biofilm growth which reflects the variability of biofilm thickness and biofilm density as a function of time is necessary for design, optimisation, and control of a TPFBBR. This thesis is mainly concerned with the development of a mathematical model incorporating the dynamic biofilm growth for a TPFBBR.

Several modelling approaches are available for deriving the desired process model. There are fundamentally two different modelling approaches that form the basis of process models, namely a mechanistic approach and an empirical approach. *Mechanistic models*, usually expressed in the form of differential equations, are based on the physical and (bio)chemical phenomena occurring within a process. In other words, the development of a mechanistic model is mainly driven by *a priori* knowledge of the relevant mechanism and from first principles (chemical and physical laws, mass balances, and so on). Mechanistic models of wastewater treatment processes are generally developed from application of reactor engineering principles, i.e., they combine expressions representing the intrinsic kinetics and transport events with mass

balance equations describing the characteristics of the particular physical system under consideration (Grady, 1983).

In contrast, *empirical models* simply relate operating input and output variables to each other and do not require a priori knowledge of the physical and (bio)chemical taking place within the process. In the present context, they are characterised by fewer parameters for the description of data compared to the mechanistic model and are derived applying statistical techniques to fit empirical functions to fit the input-output data. Such black box descriptions from pilot plant data are quite useful for design. Recently, artificial neural network (ANN) techniques as black box modelling tools have widely been used for many applications in robotics, electronic processes, and chemical and biochemical processes. The main advantages of using ANNs in process modelling are: (1) it has the ability to learn complex nonlinear relationships with limited *prior* knowledge of the process structure (2) it can perform inferences for an unknown combination of input variables (Hong et al., 1998).

As an alternative to overcoming the weak points of both mechanistic and empirical models, a so-called *hybrid model* (grey box model), which is a combination of mechanistic and empirical models, has recently been introduced (Psichogious & Ungar, 1992; Tompson & Kramer, 1994). In a hybrid model, part of the poorly and inaccurately known processes are modelled by the empirical model, and apart of the behaviour of the known processes are modelled mechanistically. ANNs are particularly attractive in a hybrid model to obtain the best possible description of processes.

The main goal of this thesis is to develop mathematical models incorporating dynamic biofilm growth for a three-phase fluidised-bed biofilm reactor (TPFBBR), based on these different modelling approaches outlined above:

1) a mechanistic model incorporating dynamic biofilm growth model with detachment,
2) a sequential neural network model, and
3) an intelligent hybrid model, combined with a neurofuzzy process estimator.

Three different modelling approaches are applied to estimate time progressions of biofilm thickness and biofilm density and to predict the dynamic changes in performance of a TPFBBR.

1.2 THESIS OVERVIEW

This thesis consists of seven chapters.

Chapter 2: This chapter serves as a general overviews of the three-phase fluidised-bed biofilm reactor (TPFBBR). First, I give the some background of a TPFBBR. Second, hydrodynamics of a TPFBBR including flow regime, phase holdups, mixing characteristics, mass transfer processes are described. Finally, a brief description of biofilm formation and biofilm characteristics taking place in a TPFBBR is presented.

Chapter 3: This chapter presents the experimental system and experiment methods.

Chapter 4: The main aim of this chapter is to develop the dynamic biofilm growth model which is able to reflect the simultaneous variability of biofilm thickness and biofilm density as a function of time, based on the mechanistic modelling approach. The biofilm detachment model is also proposed. This dynamic biofilm

growth model with detachment is combined with a reaction-diffusion model and reactor model to form an integrated model of a TPFBBR.

Chapter 5: The purpose of this chapter is to describe the application of neural network process modelling approach for modelling of the dynamic change of the biofilm thickness and biofilm density and the prediction of dynamic performance of a TPFBBR. The following three-different types of neural network are implemented:

1. multilayer feedforward neural network (MFNN) with cascaded-correlation (C-C) learning algorithm
2. multilayer feedforward neural network (MFNN) with extended Kalman filtering (EKF) learning algorithm
3. generalized regression neural network (GRNN).

Using the above neural network frameworks, the sequential neural network model is developed to describe the task of process estimation and prediction for a TPFBBR.

Chapter 6: As an alternative to the mechanistic model presented in chapter 4 and the neural network model presented in chapter 5, the main objective of this chapter is to develop the intelligent hybrid model, which is a combination of the mechanistic and empirical models of a TPFBBR. The neurofuzzy model is developed to work as *process estimators* to estimate variations of the biofilm thickness and biofilm density based on the available measurement variables. This neurofuzzy model is combined with a reaction-diffusion model and axial-dispersion model to explore the intelligent hybrid model which can predict the dynamic behavior and performance of a TPFBBR.

Chapter 7: The conclusions and summary of the main issues considered in this thesis are drawn in chapter 7. Applicability of models developed in this thesis is discussed.

1.3 CONTRIBUTIONS OF THIS THESIS

The main contributions of this thesis to the filed of mathematical modelling of a TPFBBR can be summarised as follows:

Derivation of new dynamic biofilm growth model: Based on a mechanistic modelling approach, the dynamic biofilm growth model incorporating biofilm detachment, which is derived from biomass balance equation and is able to reflect the simultaneous variability of biofilm thickness and biofilm density as a function of time, is developed.

Development of integrated model: By combining the newly developed dynamic biofilm growth model with a reaction-diffusion model and reactor model, the integrated model for a TPFBBR is developed.

Development of sequential neural network model: Using a neural network modelling approach, the sequential neural network model having the neural process estimator and the neural process predictor is developed to estimate the dynamic change of the biofilm thickness and biofilm density and to predict the dynamic performance of a TPFBBR.

Development of intelligent hybrid model: In order to provide a new comprehensive model for modelling the dynamics of a TPFBBR, the intelligent hybrid model is developed. It consists of two parts including the neurofuzzy model, which serves

as *a process estimator* of difficult-to-model process variables, and the mechanistic models, which represent the known mechanistic knowledge of TPFBBR processes.

Development of computer programs: To numerically implement the modelling approaches proposed in this thesis, the simulation software written in FORTRAN for chapter 4, the sequential neural network programs written in C^{++} for chapter 5, and the computer program written in MATLAB^{TM} with Matlab external interface engine for FORTRAN for chapter 6 are developed.

Chapter 2

Three-Phase Fluidised-Bed Biofilm Reactor (TPFBBR) - Background

2.1 THREE-PHASE FLUIDISED-BED BIOFILM REACTOR (TPFBBR)

Three-phase fluidization is an operation used to bring into contact gas, liquid, and solid particles. The solid particles are fluidised by upflow liquid, which is the continuous phase, and cocurrent gas bubbles. This three-phase fluidised-bed reactor (TPFBR) has received great attention in the past three decades because of its effectiveness in chemical processes such as hydrodesulfurfization of oil, Fisher-Tropsch synthesis, catalytic oxidation, and cracking of hydrocarbons.

An illustration of a TPFBBR is given in Fig. 2.1. The liquid and gas phases pass through the reactor upward, and keep the solid particles in suspension, consisting of

particles generally ranging from micrometers and millimeters. The fluid phases leave at the top of the reactor while the solid phase remains in the reactor in the forms of a fluidised layer. The characteristics of a three-phase fluidised-bed reactor (TPFBR) have been reviewed by Ostergaard (1968), Epstein (1981), and Muroyama & Fan (1985).

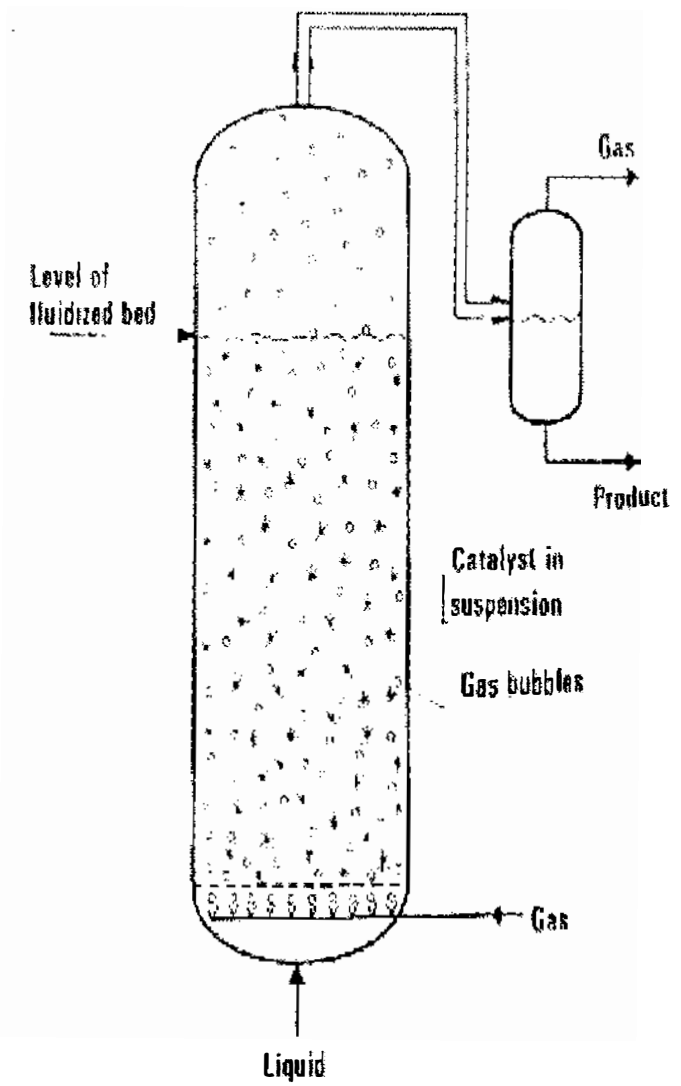


Figure 2.1. The three-phase fluidised-bed reactor (TPFBR).

Recently fluidised-bed biofilm reactors (FBBR), either the two-phase (bioparticle-liquid) or the three-phase (gas-liquid-bioparticle), have been considered as one of the most efficient fixed film-type bioreactors for wastewater treatment.

A schematic of a TPFBBR is shown in Fig. 2.2. The influent wastewater enters the reactor through a liquid distributor and air is sparged through a porous metal disk at the bottom of the column reactor.

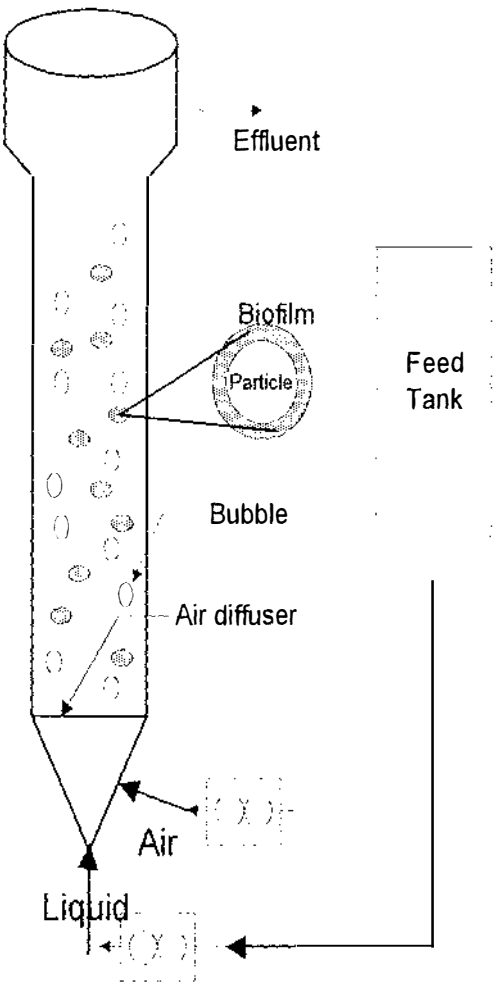


Figure 2.2. The three-phase fluidised-bed biofilm reactor (TPFBBR) system.

Solid support materials, such as sand, activated carbon or synthetic materials which provide for biofilm growth, are placed in the reactor through which wastewater is passed upflow with sufficient liquid velocity or air velocity to fluidise the particles. The particles fluidise when the velocity of a fluid stream upward through a bed of particles is sufficiently high, and the drag force on a particle can thus overcome the gravitational force of a particle. The biofilms grow on support particles as attached biomass surrounding each of the support particles. As the wastewater contaminants pass by the bioparticles covered by biofilms, they are removed from the wastewater through adsorptive and biochemical mechanisms.

The first application of fluidized bed biofilm reactor was in the area of nitrification and denitrification of wastewater treatment or organic wastewater treatment (Jeris & Owens, 1975; Scott & Hancher, 1976; Mulcahy et al., 1980; Shieh, 1980; Shieh et al., 1981). Jeris & Owens (1975) reported a successful practice of a pilot-scale denitrification fluidised bed bioreactor. They stated that the pilot-scale FBBR consistently produced greater than 99 percent removal of the influent nitrogen in less than 6.5 min at a flux rate of 8.16 $m^3/d/m^2$. Andrew & Tien (1981) investigated the FBBR involving simultaneous biological and activated carbon treatment of organic wastewater treatment. The advantages of fluidized bed biofilm reactors for denitrification in comparison with packed-bed biofilm reactors or other suspended growth treatment processes such as activated-sludge process include superior performance and no clogging.

In comparison to packed-bed biofilm reactor consisting of immobilised cells, the fluidised bed reactors have the advantage of good solid-fluid mixing and minimal pressure drop. In real application of a FBBR, the very high rates of volumetric loading rates for wastewater treatment have been obtained since the high biologically active surface area is available and the high biomass concentration can be maintained. It is known that the average biomass concentration in a FBBR ranges between 10000 *mg/l* and 50000 *mg/l* and is 10 or 40 times greater than that in conventional suspended growth systems (Table 2.1). Because of the high biomass

concentration, a FBBR gives a 200-500 percent reduction in reactor volume when compared to other fixed film processes such as tricking filters and rotating biological contractors (RBC) and suspended growth systems. Another benefit of FBBR is upgrading an existing wastewater treatment plant can be accomplished simply by adding additional fluidised-bed biofilm reactor units.

Recently, a three-phase fluidised-bed biofilm reactor has been for biological degradation of phenol (Holladay et al., 1978; Lee et al., 1979; Wisecarver & Fan, 1989; Fan et al., 1987). Holladay et al. (1978) reported high phenol degradation rates for a synthetic phenol wastewater. They compared phenol degradation in stirred-tank, packed-bed and fluidized-bed rectors. Their results showed that although a high phenol-bearing wastewater could be treated in the stirred-tank reactor, this treatment method required the largest reactor volume because of the long retention times. The highest degradation rate and shortest retention time was observed for the fluidized-bed bioreactor.

The advantages of a three-phase fluidised-bed biofilm reactor can be summarised as follows (Ryhiner et al., 1988):

1. fluidisation is created by the gas velocity;
2. liquid recycle is not needed;
3. oxygen transfer occurs throughout the reaction zone.

Some disadvantages of the FBBR have been reported (Lee et al., 1979; Tzeng, 1991):

1. difficulty in obtaining good liquid-solid disengagement,
2. not suitable for treating wastewater containing compounds requiring long retention times,
3. exhibiting relatively fluctuating operation at high bed expansion conditions.

In a TPFBBR, the decrease in biofilm density with increasing biofilm thickness is significant in the operation of a TFBBR. This is because stratification takes place in the reactor and less dense bioparticles with thicker biofilms tend to concentrate at the top of the reactor. This can lead to an increase in bed expansion (the volume of voids increases). As the bed expansion continues, the less dense particles with thicker biofilms may be washed out and it causes the reduction of TPFBBR performance. Thus it is necessary to implement control mechanisms on the biofilm thickness of bioparticles in order to maintain a satisfactorily treatment efficiency for a long-term operation (Tzeng, 1991).

Table 2.1. Comparisons of biomass concentration among biological processes (Perry, 1996).

Process	MLSS (*mg/l*)	Surface area (m^2/m^3)
Activated sludge	1500-3000	-
Pure oxygen activated sludge	2000-5000	-
suspended growth nitrification	1000-2000	-
Trickling filter	-	3.5-15
RBC	-	9-13
FBBR-CBOD removal	12000-20000	260-400
FBBR-nitrification	8000-12000	250-380
FBBR-denitrification	25000-40000	250-380

2.2 HYDRODYNAMICS OF TPFBBR

2.2.1 Flow Regimes

As a first approximation, a TPFBR could be described as a bed fluidised by the liquid which the gas phase flows as in a bubble column. This is substantially true if the liquid velocity is relatively high and the gas velocity remains low. By contrast, if the gas velocity is high and the liquid velocity is low, this situation is totally

different, with the appearance of pulsed flow. For intermediate gas and liquid velocities, a progressive passage can be observed from the fixed bed to the completely fluidised bed. These situations are shown in Fig. 2.3.

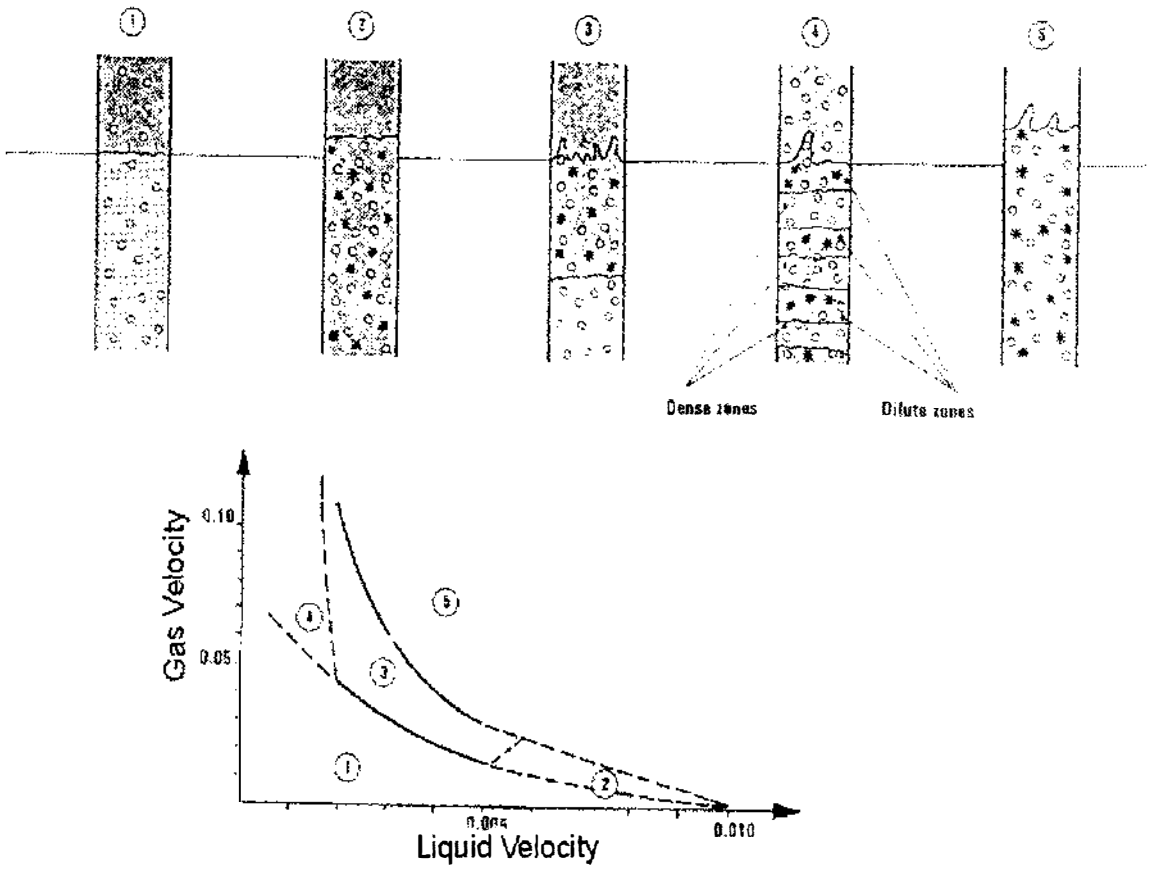


Figure 2.3. Flow pattern in a three-phase fluidized-bed reactor. (1) Fixed bed (2) Pseudo liquid/solid fluidised bed (3) Partially fluidised bed (4) Pulsed bed (5) Fluidised bed (Trambouze et al., 1988).

2.2.2 Minimum Fluidization Velocity

The minimum fluidization velocity (U_{lmf}) is the basic design parameter in a TPFBBR. There are a number of empirical correlations reported in the literature for estimating U_{lmf}. Some of them are listed in Table 2.2.

Table 2.2. Summary of empirical equations for M.F. velocity.

Authors	Correlation (SI units)
Ermakova et al. (1970)	$\frac{U_{lmf}}{U''_{lmf}} = 1 - \varepsilon_g - 0.5U_g^{0.075}$
Begovich and Watson (1978), 1	$\mathrm{Re}_{lmf} = 5.121 \times 10^{-1} Ar_l^{0.662} Fr_g^{-0.118}$
Begovich and Watson (1978), 2	$\frac{U_{lmf}}{U''_{lmf}} = 1 - 1622U_g^{0.436} u_l^{0.227} d_p^{0.58} (\rho_s - \rho_l)^{-0.305}$
Costa et al. (1986)	$U_{lmf} = 6.969 \times 10^{-4} U_g^{0.328} (\phi d_p)^{1.086} (\rho_s - \rho_l)^{0.865} D_T^{0.042} u_l^{-0.355}$
Song et al. (1989)	$\frac{U_{lmf}}{U''_{lmf}} = 1 - 375U_g^{-0.327} u_l^{0.227} d_p^{1.213} (\rho_s - \rho_l)^{-0.423}$

(U''_{lmf} = liquid velocity at minimum liquid-solid fluidisation)

Recently, Zang et al. (1995) found that especially for low gas velocity, the Gas-Perturbed Liquid model, together with the approximate equation for α_{mf} (Eq. 2.1), showed almost a good agreement with the experimental data as the best available empirical equation for U_{lmf}, and the advantage of correctly reducing to the Wen-Yu correlation (1966) for minimum two-phase fluidization as the gas velocity goes to zero. The minimum fluidization velocity (U_{lmf}) based on Gas-Perturbed Liquid model is given by

$$\mathrm{Re}_{lmf} = \frac{U_{lmf} d_{bp} \rho_L}{\mu_L} = \sqrt{\left(42.86\frac{(1-\varepsilon_{mf})}{\phi}\right)^2 + 0.5715\phi\varepsilon_{mf}^3(1-\alpha_{mf})^3 Ar_l} - 42.86\frac{(1-\varepsilon_{mf})}{\phi} \quad (2.1)$$

where Re_{lmf} is liquid Reynolds number at minimum three-phase fluidization, μ_L is the viscosity of liquid, ρ_L is the density of liquid, d_{bp} is the particle diameter, ε_{mf} is the bed porosity at minimum fluidization velocity, α_{mf} is the gas holdup on solid-

free basis, ϕ is the particle shape factor (0.906), and Ar_l is the liquid Archimedes number. α_{mf} can be computed from the following correlation (Yang et al., 1993)

$$\alpha_{mf} = \frac{0.16}{\varepsilon_{mf}} \frac{U_G}{U_G + U_L} \tag{2.2}$$

where U_L, U_G are superficial liquid velocity and gas velocity, respectively.

2.2.3 Pressure Drop and Phase Holdups

The pressure drop across the TPFBR is important because pumping costs could be a significant part of the total operating cost. Various transport variables such as gas-liquid and liquid-solid mass transfer coefficients can be correlated to the pressure drop using the analogy between mass and momentum transfer processes. The pressure drop due to the bed alone, ΔP is calculated simply on the basis of the static pressure:

$$(-\Delta P) = g(\varepsilon_l \rho_L + \varepsilon_g \rho_G + \varepsilon_s \rho_S) H_e \tag{2.3}$$

where ρ_G, ρ_L, ρ_S are the density of gas, liquid, and solid, respectively and $\varepsilon_g, \varepsilon_l, \varepsilon_s$ are the gas, liquid, and solid holdup, respectively. H_e is the fluidised bed height and g is the gravitational acceleration.

The performance of a TPFBR is highly influenced by the hydrodynamic properties such as the phase holdups. For example, the design of a TPFBR depends on the expansion or contraction of the fluidised bed. The bubble size, gas residence time, and consequently the gas-liquid mass transfer are influenced by the phase holdups.
The overall phase holdups in a TPFBR can be obtained through the following equations:

$$\varepsilon_l + \varepsilon_g + \varepsilon_s = 1 \tag{2.4}$$

$$\varepsilon_s = \frac{M_s}{(H_e\, a_p\, \rho_S)} \tag{2.5}$$

where a_p is the surface area per unit volume of solids and M_s is total weight of solid particles.

The individual phase holdups in a TPFBR have been investigated by Kim et al. (1975), Armstrong et al. (1976), and Bhatia et al. (1972). In the conventional studies on phase holdups and bed porosity, overall values have been obtained through the pressure profile along the fluidised bed, the total amount of solid particles, and the continuity of the three phases. In order to obtain information on local phase holdups, *in situ* measuring probes have been used in various studies (Begovich & Watsons, 1978a). It is inevitable that the hydrodynamics of the fluidised bed is somewhat disrupted by an *in situ* measuring device. Therefore, one has to be cautious in the design of *in situ* probes so that the disturbance to the hydrodynamics can be minimised (Lee & de Lasa, 1987). In any case, the results obtained from *in situ* probe measurements indicate that there are variations of phase holdups in both the axial and radial directions in a TPFBR, particularly in the top section of the bed.

2.2.4 Gas-Liquid Mass Transfer

The mass transfer can play an important role in the determination of overall reaction rate in a TPFBR, particularly for biological reaction. The rates of mass transfer steps are dependent on the hydrodynamic properties of a TPFBR.

Measurements of gas-liquid mass transfer have usually assumed a plug flow model as a basis. With these conditions, several correlations have been established to calculate the volumetric gas-liquid mass transfer coefficient (Ostergaard & Fosbel, 1972; Lee & Worthington, 1974; Robinson & Wilke, 1974). Ostergaard & Fosbel

(1972) reported that the gas-liquid volumetric mass transfer coefficient ($K_l a$) in a TPFBR varies considerably with the axial distance from the gas distributor. In addition, $K_l a$ varies with the particle size. For example, shallow beds containing large particles exhibit particularly high gas-liquid mass transfer rates. Alvarez-Cuenca et al. (1983) obtained that at high gas velocity, $K_l a$ was the highest in the region between the dense bed region and the dilute bed region.

Studies of gas-liquid mass transfer may involve the measurement of the interfacial area (a). Lee & Worthington (1974) measured the volumetric gas-liquid transfer $K_l a$ and the interfacial a, separately. In their experiment, the mass transfer was found to increase with an increase in the bubble size. They showed that $K_l a$ varies linearly with ε_g. The volumetric mass transfer coefficient $K_l a$ depends on the gas velocity, sparger design and is sensitive to the physico-chemical properties, particularly, those which promote or prevent coalescence. In addition, the column diameter has some influence if it is small. In a TPFBR, $K_l a$ can be affected by the presence of solids. Some authors (Kato et al., 1972; Nuguen-Tien & Deckwer, 1981) indicate that the degree of influence of suspended particles on $K_l a$ depends on the particle concentration, the particle size, the liquid-solid density difference, the geometrical sizes and the operating conditions of the reactor (i.e., gas and liquid velocity). At high liquid velocities and low gas velocities, the $K_l a$ values are slightly higher than those without the presence of solids (Nuguen-Tien & Deckwer, 1981). Kato et al. (1972) showed that for higher solid concentration a steep decrease in $K_l a$ was found which was caused by a decrease in a. Dhahuka & Stepanek (1980a) reported that with an increase in particle size, $K_l a$ decreased because of a decrease in a.

Most reported studies for the estimation of $K_l a$ in a three-phase fluidized-bed have used particles with densities ranging over 2000 kg/m^3 while most TPFBBR use particles with densities ranging up to 1300 kg/m^3. So the direct application of the

correlations for $K_l a$ developed for dense particles requires the care if used for a TPFBBR which normally uses low density particles. Recently, Nore et al. (1992) studied hydrodynamics, gas-solid and liquid-solid mass transfer with in a TPFBBR with the low density ranging from 1300 to $1700 kg/m^3$. They found that with the particle density ranging from 1300 to $3000 kg/m^3$ in a TPFBBR, $K_l a$ increased strongly when the gas velocity was increased and less strongly when the liquid velocity was increased. They suggested the following correlation for $K_l a$:

$$K_l a = 4766\left(\frac{U_L}{\varepsilon_l}\right)^{0.58}(U_G)^{0.50}\left(\frac{\rho_{sp}}{\rho_L}\right)^{-0.9} \tag{2.6}$$

where ρ_{sp} is the density of the particle and ρ_L is the density of liquid.

2.2.5 Liquid-Solid Mass Transfer

Just as in the case of gas-liquid mass transfer, the liquid-sold mass transfer step may play an important role in the performance of a TPFBR for chemical or biological reactions. The structure of the biofilm tends to slow the transport of substrate through the biofilm and therefore, the substrate concentration surrounding the microorganisms within the biofilm is less than that in bulk liquid. Thus the mass transport properties of the biofilm are of critical importance in assessing the overall performance of a FBBR.

Substrate conversion in a FBBR can be described by the following steps, as shown in Fig. 2.4, (La Motta, 1976):

1) Transport of substrate from the bulk liquid to the liquid-biofilm interface (external mass transfer);
2) Transport of substrate with the biofilm (internal mass transfer), and
3) Substrate conversion reactions with the biofilm.

Step 2) and 3) take place simultaneously and thus neither can be said to control while step 1) occurs in series with steps 2) and 3). For intrinsic reaction rates with positive dependence on substrate concentration (i.e., Monod kinetic), the gradients established by step 1) and 2) decrease the observed reaction rate by decreasing intrabiofilm substrate concentration (Shieh & Keenan, 1986).

In order to describe mass transport from the bulk liquid to the surface of the support particle and reaction at that position, the *Nersnt diffusion layer* and a *stagnant film theory* have widely been used and lead to the following equation for the flux F_s of substrate from the bulk liquid to the interface, as shown in Fig. 2.4:

$$F_s = k_{ls}(C_s^l - C_s^s) \tag{2.7}$$

where C_s^l and C_s^s are the substrate concentrations at the interface and in the liquid, respectively, and k_{ls} is the liquid-solid mass transfer coefficient.

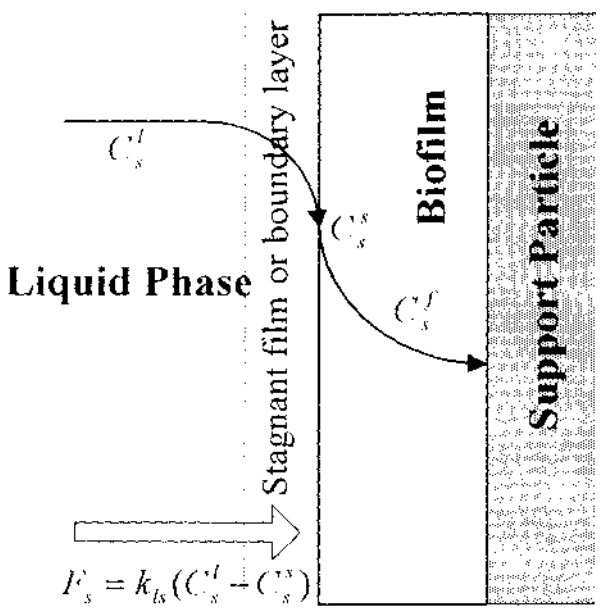


Figure 2.4. Illustration of mass transfer processes in a FBBR.

An analytical solution for k_{ls} is possible for the ideal case of a single sphere at rest in an infinite stagnant fluid. k_{ls} is then given by:

$$k_{ls} = \frac{2D_m}{d_{sp}} \tag{2.8}$$

where D_m is the molecular diffusivity in liquid.

For the general case of mass transfer between a moving fluid and a spherical particle, the Sherwood number, *Sh*, Schmidt number, *Sc*, Stanton number, *St*, and Froude number, *Fr*, relate the physical properties of the system to the mass-transfer coefficient and are more often used (Brodkey & Hershey, 1988). These correlations are most often expressed in terms of dimensionless numbers, often in the form of a power series.

$$Sh = \frac{k_{ls}d_{sp}}{D_m} \tag{2.9}$$

$$Sc = \frac{\mu_L}{\rho_L D_m} \tag{2.10}$$

$$St = \frac{k_{ls}\varepsilon}{U_L} \tag{2.11}$$

$$Fr = \frac{U_G}{\sqrt{gD_c}} \tag{2.12}$$

In general, steady-state theories for the liquid-solid mass transfer are largely classified into two categories; those based on the terminal velocity-slip velocity approach and the others based on Kolmogoroff's theory. In the terminal velocity-slip velocity approach, the steady slip velocity between solid and liquid is used in the correlation for the Sherwood number. Based on this theory, the experimental data for the liquid-solid mass transfer coefficient (k_{ls}) are often correlated by a dimensionless equation of the form,

$$Sh = 2.0 + \alpha\, Sc^{1/3}\, \mathrm{Re}^{1/2} \tag{2.13}$$

The value of the constant α reported in the literature lie between 0.03 and 1.0 (Shah, 1979). A review of the data of Rowe & Claxton (1966) on the Reynolds number range 20 through 2000 indicates that α=0.76 for liquids.

Beek (1971) developed a more general correlation of liquid-solid mass transfer coefficient within a fluidised-bed based on the data of several researchers. The correlation of Beek is

$$St\, Sc^{2/3} = (0.81 \pm 0.05)\, \mathrm{Re}^{-0.5} \tag{2.14}$$

Kolmogoroff's theory is based on the length scale of the micro-scale eddies, which is defined as

$$\eta = \left(\frac{\nu^3}{E}\right)^{1/4} \tag{2.15}$$

and the velocity scale is defined as

$$v_1 = (\nu E)^{1/4} \tag{2.16}$$

where E is the local energy dissipation rate per unit mass. From the stochastic behavior of the fluid flow around the suspended particle and Kolmogoroff's theory of isotropic turbulence the following relationship for the Reynolds number can be derived (Shah et al., 1982)

$$\begin{aligned} H_t > \eta > d_{sp} \qquad & \mathrm{Re} = c^* \left(\frac{E d_{sp}^4}{\nu^3}\right)^{1/2} \\ H_t > d_{sp} > \eta \qquad & \mathrm{Re} = c^* \left(\frac{E d_{sp}^4}{\nu^3}\right)^{1/3} \end{aligned} \tag{2.17}$$

where c^* represents a dimensionless constant and H_i is a characteristic length, for instance, the suspension height. By using Kolmogoroff's theory, the energy dissipation rate in TPFBR can be calculated from the pressure drop experienced by the gas flow rate. The energy input P' is approximated by

$$P' = U_G \Delta p = U_G \rho_L g \varepsilon_l H_i \tag{2.18}$$

where Δp is the pressure drop in the bed. Therefore the specific energy dissipation rate, E, can be calculated as,

$$E = U_G g \tag{2.19}$$

Information in the literature pertaining to liquid-solid mass transfer in related systems, such as two-phase FBBR is fairly comprehensive. Little is known, however, about liquid-solid mass transfer in a TPFBBR. Arters & Fan (1984) developed the liquid-solid particle mass transfer coefficient in a TPFBR. They employed cylindrical particles of benzoic acid which were fluidised with water and air. Their results showed that liquid-solid mass transfer in a TPFBR is higher than that in a two-phase fluidised bed at a given liquid velocity. Furthermore, the Sherwood number (Sh) for k_{ls} increase with increasing gas velocity. Liquid-solid mass transfer in a TPFBR appears to be relatively independent of the liquid velocity, as has been noted for a two-phase fluidised bed reactor. The correlation of Arters & Fan for k_{ls} is given as

$$Sh = \frac{k_{ls} d_{sp}}{D_m} = 0.228(1 + 0.0826 \mathrm{Re}_g^{0.623}) Ga^{0.323} Sc^{0.4} \left(\frac{\rho_{sp} - \rho_L}{\rho_L} \right)^{0.3} \tag{2.20}$$

and the Gallileo number (Ga) is defined as

$$Ga = \frac{d_{sp}^3 \rho_L^2 g}{\mu_L^2} \tag{2.21}$$

where D_m is molecular diffusivity in liquid and ρ_{sp} is density of particle.

Recently, Nore et al. (1992) studied hydrodynamics, gas-solid and liquid-solid mass transfer with in a TPFBBR with the low density ranging from 1300 to 1700 kg/m^3. In their studies, increasing the gas velocity increased k_{ls}, especially at low gas velocities for low particle densities and the liquid velocity had almost no effect on liquid-solid mass transfer coefficient. Nore et al. correlation for estimation of k_{ls} with good prediction of the 250 values measured are

$$k_{ls} = 1.10\left(\frac{U_L}{\varepsilon_l}\right)^{0.43} d_{bp}^{\,0.24} \quad \text{at } U_{Gi} = 0$$
$$k_{ls} = 0.197\left(\frac{U_L}{\varepsilon_l}\right)^{0.051} d_{bp}^{\,0.17} \quad \text{at } U_{Gi} \neq 0 \tag{2.22}$$

Typical range for k_{ls} with $d_{sp} \approx 1130\,kg/m^3$ were varying from about 0.0003 to 0.0013 cm/s.

2.2.6 Mixing

Gas Mixing

Mixing of the gas phase is due to the fact that the gas bubbles have different velocities associated with their size. It appears that backmixing increases with rising gas flow rates, and with decreasing particle size. However, this could be ignored as a first approximation (Muroyama & Fan, 1985). Michelsen & Ostergaard (1970) reported that the determination of axial dispersion was difficult since the axial-dispersion model was proven to be unsuccessful in accounting for the gas phase mixing in system which rapid coalesce. In such a system, a negative dispersion coefficient was observed. Furthermore, accurate determination of the axial dispersion coefficient for the gas phase is difficult for beds with large particles. In TPFBR or

TPFBBR, plug flow can often be assumed for the gas phase. Appreciable backmixing, however, may occur in the liquid phase, especially for beds with fine particles in cocurrent TPFBR or TPFBBR. The backmixing of the liquid and solid particles in such a bed is primarily caused by the rising motion of coalesced large bubbles (Muroyama & Fan, 1985).

Liquid Mixing

The axial-dispersion model with a single value of the axial-dispersion coefficient to characterize the whole bed has been most commonly used to describe liquid mixing in TPFBR (Ostergaard, 1968; Ermakova et al., 1973; Shah, 1979) or TPFBBR (Wisecarver & Fan, 1987; Petersen & Davison, 1995).

The various correlations for calculating E_L have been developed by some authors (Joshi, 1980; Muroyama et al., 1978; Kim et al., 1992). The reported data on the axial liquid dispersion coefficient E_L indicate that E_L depends on the gas velocity, the liquid properties, and bed diameter. The influence of the liquid phase properties is not clearly understood (Shah, et al., 1982). Davison et al. (1977) showed that, depending on the liquid flow rate, the axial dispersion coefficient (E_L) of the liquid retains a value between bout 40 and 120 cm^2/sec in the dispersed bubble flow regime; in the coalesced bubble flow regime, E_L retains very high values, varying from 200 to 400 cm^2/sec.

Joshi (1980) proposed a unified correlation of the axial liquid mixing in gas-liquid two phase columns and three-phase fluidized beds. His correlation has the form:

$$E_L = 0.29(U_L + V_c)D_c \tag{2.23}$$

where E_L is in m^2/sec, D_c in m, and V_c in m/sec. V_c can be calculated by:

$$V_c = 1.31\left\{gD_c\left[U_G + U_L - \frac{\rho_L U_L}{\varepsilon_s\rho_S + \varepsilon_l\rho_L} - \varepsilon_s\left(\frac{\rho_S}{\varepsilon_s\rho_S + \varepsilon_l\rho_L} - 1\right)U_t - \varepsilon_g V_{b\infty}\right]\right\}^{1/3} \quad (2.24)$$

Here, U_G, U_L, U_t and $V_{b\infty}$ are in $m/\sec$, and ρ_L and ρ_S are in kg/m.

Kim et al. (1992) found that E_L increases with increasing gas and liquid velocities in three-phase fluidized beds of small particles whereas the effect of liquid surface tension and liquid viscosity on E_L are found to be small. They also showed that E_L decreases with increasing particle size and it sharply increases with increasing column size. They proposed the correlation

$$Pe_L = \frac{d_{bp}U_L}{E_L} = 20.19\left(\frac{d_{bp}}{D_c}\right)^{1.66}\left(\frac{U_L}{U_L + U_G}\right)^{1.03} \quad (2.25)$$

Studies on the backmixing characteristics of liquid have been reviewed by Fan & Muroyama (1985). They recommend the equation of Kim et al. (1992) for the estimation of E_L in a TPFBR since their correlations cover a wide range of literature data.

Solid mixing

One of the characteristics of a TPFBR or TPFBBR of low-density particles which most distinguish them from those of high-density particles is the axial nonhomogeneity of the holdup of the phases. This nonhomogeneity of the axial phase holdups is also true in TPFBBR. In a TPFBBR, the decrease in bioparticle density with increasing biofilm thickness on support particles is significant in the operation of a TPFBBR. Thus with the bed, nonhomogeneous distribution take place and the less dense bioparticles with thicker biofilms tend to locate at the top of the reactor. Fig 2.5 shows the typical solid holdup distribution in a TPFBR containing low-density particles.

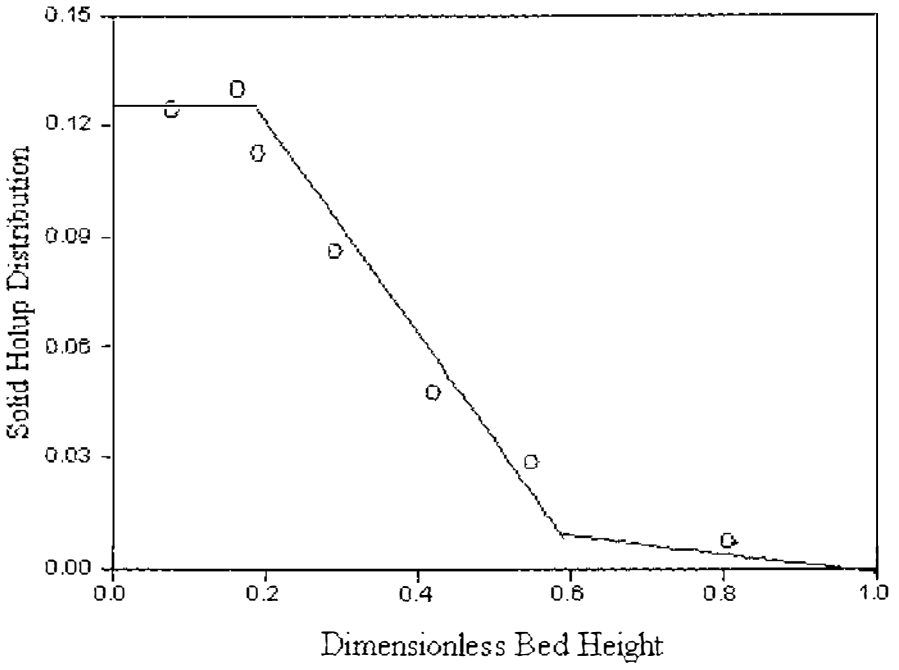


Figure 2.5. Axial solid distribution in a TPFBR (Tang & Fan, 1989).

An investigation of solids mixing in a TPFBR was conducted by Fan et al. (1982), Fan et al. (1984), and Tang & Fan (1989). Fan et al. (1982) showed that the degree of particle segregation decreases sharply with an increase in the gas velocity. The degree of the solid mixing, however, would be expected to be lower than that of the liquid mixing in a TPFBR (Tang & Fan, 1989). Tang & Fan (1989) examined the solid mixing in a TPFBR containing low-density particles and developed a mechanistic model with axial solid dispersion coefficient E_s. They found that in the dispersed bubble regime (high gas velocity), gas velocity has only a slight effect on the axial solid holdup distribution. The solid dispersion increases significantly with a decrease in the liquid velocity, corresponding to a transition from the dispersed to the coalesced bubble regime. Tang & Fan (1989) proposed the following correlation for estimating axial solid dispersion coefficient E_s, which is expressed as a function of particle terminal velocity and gas velocity:

$$E_S = 0.080 U_G^{1.34} U_{t_0}^{-1.22} \tag{2.26}$$

where U_{t_0} is the terminal falling velocity of a single particle in an infinite liquid medium. The Eq. (2.26) can be used to estimate E_S in a TPFBR of particles with U_{t_0} ranging from 0.025 to 0.055 m/s and operate in the dispersed bubble regime.

2.3 BIOFILM CHARACTERISTICS IN A TPFBBR

2.3.1 Biofilm Formation

Microorganisms, primarily bacteria, exhibit a tendency for adsorbing to and colonising surfaces submerged in aquatic environments. The immobilized cells grow, reproduce, and produce extracellular polymeric substance (EPS) in the development of a biofilm. Bryers & Characklis (1982) suggested the process of biofilm development on the support particles occurring in a biofilm reactor and described the net results of the following transport and biological processes:

1. Adsorption of dissolved organics at the surface of particles
2. Transport of microbial particles to the surface
3. Microorganisms adhesion to the surface biofilm production
4. Biofilm growth
5. Biofilm detachment

In step 4, biofilm production is the net accumulation of attached material due to cellular reproduction and microbial production of extracellular polymers. During the biofilm development, portion of biofilm peels away from the particle surface and is entrained in the liquid flow. There exist two mechanisms: 1) detachment 2) sloughing. Detachment is a process of continuous biofilm removal and is highly dependent on hydrodynamic conditions (Peyton & Characklis, 1993). Sloughing,

however, appears to be a random, massive removal of biofilm attributed to oxygen/nutrient limitations deep within the biofilm (Bryers & Characklis, 1982).

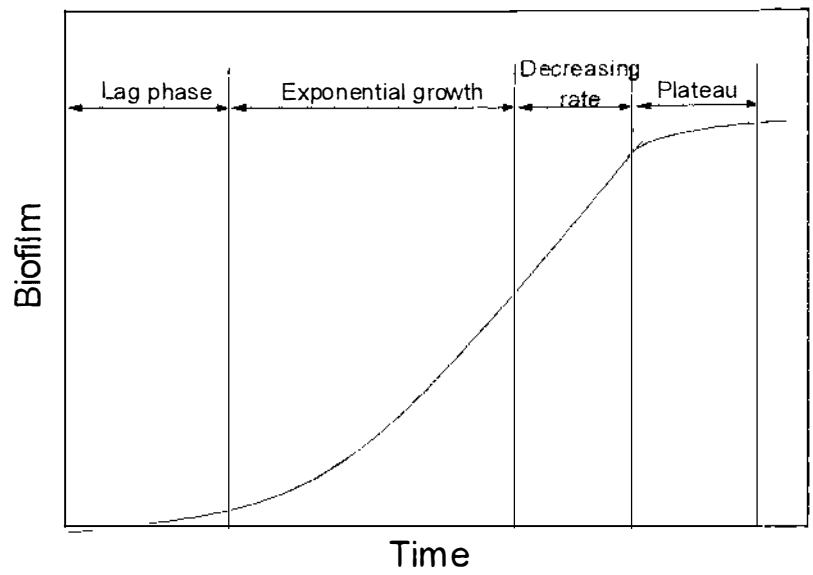


Figure 2.6. Biofilm deveopment processes.

In summary, biofilm development on the support particles is a dynamic process influenced by microbial attachment and detachment processes and growth. Many researches (Rittmann, 1982; Chang et al., 1991; Peyton & Characklis, 1993; Trinet et al., 1991; Tijhuis et al., 1994; Gjaltena et al., 1995) have focused on the understanding of biofilm detachment during biofilm formation in fluidised bed biofilm reactor or biofilm airlift suspension (BAS) reactor because the performance of biofilm process is related to the biomass amount on support particles.

It has been known that many factors affect biofilm formation in a fluidised-bed biofilm Reactor (FBBR). Shieh & Keenan (1986) summarise the following factors affecting biofilm formation in a FBBR:

1. Microbial type and Species
2. Support particle surface characteristics
3. Polysaccharide materials
4. Hydrodynamic conditions
5. Substrate characteristics and concentration
6. Environmental conditions.

2.3.2 Biofilm Characteristics

Substrate removal processes occurring in the biofilm reactor is the result of simultaneous interaction between the transport process of substrate within biofilm and the substrate conversion processes to biomass by biochemical reaction. Generally, biochemical reactions in a biofilm reactor are expressed on the basis of a unit of biomass. The removal rate of substrate, for example, is expressed as the mass of substrate removed per unit time per unit of biomass while the rate of biomass accumulation by the substrate conversion process to biomass is expressed as the amount of biomass accumulated per unit time per unit of biomass present. In order to form these processes mathematically, the biofilm thickness and biofilm density are widely used for modelling the physical characteristics of biofilm reactor because the mass of biofilm present in biofilm rector can be expressed as a function of the biofilm thickness and biofilm density.

Biofilm Thickness

Biofilm thickness is one of the most important parameters in biofilm reactor because it represents the total accumulation of biomass in the reactor. Biofilm thickness, the perpendicular distance from the substratum to the biofilm-bulk liquid interface, has been used to determine the distance through which substrates and nutrients must diffuse to fully penetrate a biofilm.

When considering biofilm thickness, it is important that a distinction be made between the total biofilm thickness and the active biofilm thickness (Grady Jr., 1983). It is meant that a thick biofilm does not guarantee a greater substrate removal rate compared to that of a thin biofilm. It has been known that the substrate removal rate increases with increasing biofilm thickness up to a certain thickness, beyond which it remains constant. When the biofilm thickness is less than a critical value, commonly referred as optimum biofilm thickness, the entire biofilm is active. As the biofilm thickness exceeds the optimum biofilm thickness, only that portion of the biofilm is active. As a result, increase in biofilm thickness beyond the optimum biofilm thickness will not induce a corresponding increase in observed substrate removal rate (Grady Jr., 1983; Shieh & Keenan, 1986; Truelar & Characklis, 1982). Truelar & Characklis (1982) also found that the value of active biofilm thickness increased as the substrate concentration in the liquid phase increased.

In a TPFBBR, large biofilm thickness by excessive accumulation of biomass on support particles is not desirable and it leads to a greater bed height increase which reduces the bed stability at a given operating conditions. In extreme cases, washout of bioparticles from the reactor could occur and it causes the loss of performance of a TPFBBR. In order to properly operate a TPFBBR and maximise the performance of a TPFBBR, control of biofilm thickness is required. The biofilm growth in a TPFBBR is generally slow since 70-90% of biomass is continuously being detached from the support particles during the biofilm formation (Hong et al, 1998; Gjaitema et al, 1995). The biofilm detachment is very significant because this affects directly the amount of biomass and ultimately the performance of a TPFBBR.

Biofilm Density

The biofilm density is an another important parameter in a biofilm reactor because the substrate removal rates are related to the biomass present. The biofilm density must be coupled with the biofilm thickness and included in any mathematical model when calculating the accurate reaction rate.

The biofilm density is generally expressed in terms of the dry weight of biomass of the biofilm per unit volume of wet biofilm. The biofilm density obtained in a TPFBBR generally had higher values to those from a two-phase FBBR or from other biofilm reactors because turbulence and particle-particle interactions enable the biofilms to be denser. Table 2.3 summarises the steady-state biofilm thickness and biofilm density data determined from several biofilm reactors.

Table 2.3. Steady-state biofilm density and biofilm thickness values from different types of biofilm reactors.

Biofilm Density & Biofilm Thickness	Reactor Type	Substrate	Reference
X_a^f=50 mg/cm^3 L_f=100 μm	Tubular reactor	Municipal wastewater	Tomlinson et al. (1966)
X_a^f=50 mg/cm^3 L_f=1100-3800 μm	RBC	Industrial wastewater	Paolini et al. (1979)
X_a^f=48 mg/cm^3 L_f=113 μm	two-phase FBBR	Municipal wastewater	Shieh et al. (1981)
X_a^f=81 mg/cm^3 L_f=59.8 μm	TPFBBR	Phenol	Wisecarver & Fan (1989)
X_a^f=141 mg/cm^3 L_f=23 μm	TPFBBR	Phenol	Livingston & Chase (1989)
X_a^f=30 mg/cm^3 L_f=181 μm	TPFBBR	Phenol	Hirata et al. (1982)

(X_a^f = biofilm density, L_f = biofilm thickness)

Hoehn & Ray (1973) first reported that the biofilm density varies with the biofilm thickness and reaches a maximum value at a thickness consistent with the active biofilm thickness. Shieh et al. (1981) observed a profound dependence of biofilm density on biofilm thickness in a two-phase FBBR for denitrification of municipal wastewater. Fan et al. (1987) and Tanyolac & Beynenal (1997) found that as the biofilm thickness increased, the biofilm density increased up to a maximum value,

started to decrease with increasing biofilm thickness, and finally reached a minimum value. Therefore, in order to evaluate a true performance of a TPFBBR or a biofilm reactor, a mathematical model is required to include the relationship between biofilm thickness and biofilm density since the performance of a TPFBBR or a biofilm reactor depends on the biofilm thickness and biofilm density.

Recently researches have investigated through micro-scale observation technique (e.g., the microelectrode or micro-slicing techniques) to enhance the understanding of the complexities of biofilm structure, that is the influence of biofilm structure on transport and transformation processes in biofilms (Zang & Bishop, 1994a; Zang & Bishop, 1994b; Bishop et al., 1995). Their results from direct experimental evidence showed that the biofilms have a non-uniform spatial distribution of biofilm properties, which conflicts with the *a priori* assumption of many biofilm growth models, that is a uniform spatial distribution of biofilm properties. The structure of biofilms can be highly stratified, and therefore, biofilm density, porosity, pore structure, and the composition of the microbial community can be quiet different in each layer. Bishop et al. (1995) showed that the biofilm density of a top layer and a bottom layer can be 5-10 times different and the rate of biofilm density increase with biofilm depth is greater for thin biofilms than for thick biofilms. It is known that the density with biofilm depth is the result of competition and drainage. As a result of the change of density in biofilm, the porosity also has a spatial distribution within the biofilm. Because of these distributions of biofilm density and porosity, both tortuosity and effective diffusivity change with biofilm depth. Zang & Bishop (1994b) showed that for biofilm with porosity of 0.84-0.93 in the top layers and 0.58-0.67 in the bottom layer, tortuosity factors increase approximately from 1.115 in the top layer to 1.6 in the bottom layer. Although the information from the micro-scale-based observations is quite useful and practical, it is still difficult to incorporate it into macro-scale-based mathematical models. Averages values of biofilm thickness, biofilm density, and effective diffusivity are still widely used for macro-scale-based modelling and design purposes.

Chapter 3

Experimental Methods

3.1 INTRODUCTION

The basic objectives of the experimental work are:

- To obtain the value of variables needed as input to the mathematical model presented in chapter 4, 5, and 6.
- To obtain dynamic performance data of a TPFBBR which can be used to evaluate the mathematical model proposed in this thesis.

3.2 EXPERIMENTAL

3.2.1 Reactor

Three laboratory-scale three-phase fluidized-bed biofilm reactors with identical dimensions were constructed and operated under different operating conditions to obtain the dynamics of the biofilm growth after startup. A diagram of the TPFBBR is illustrated in Fig. 3.1. The major components of each TPFBBR include reactor, gas-liquid separator, air and liquid distributor, feed pumps, and feed and effluent tanks.

The reactor was made of plexiglass and was of 4.4 cm ID and, with six sampling ports installed along the bed height to obtain both liquid and bioparticle samples. The bottom of the bed was conical in shape which was employed to promote even distribution of the incoming wastewater. The enlarged section with 6.6 cm ID and 25 cm height was fitted at the top of the bed to provide better liquid-bioparticle separation. The total reactor volume was 1200 mL. Air sparged through a porous metal disk, which covered the bottom cross section, at rates which caused fluidization of the support particles without any liquid flow. The average pore size of a porous metal disk is 13 μm. The pH was controlled between 6.7 and 6.9, and the temperature held at 21° C. The dissolved oxygen was monitored with an electrode.

3.2.2 Startup and Operation of a TPFBBR

Initially, 15% of reactor volume was filled with clean activated carbon particles (settled volume %). Activated mixed culture corresponding to 5% of reactor volume was then added to the TPFBBR containing synthetic wastewater with the phenol concentration of 200 *mg/l*. The TPFBBR was fluidized by upward air flow and was operated in a batch mode for a few days to allow the buildup of active microbial cells on activated carbon particles. The operation was switched to a continuous mode. The inlet phenol concentration was maintained between 150 *mg/l* and 200 *mg/l* during all experimental runs but the liquid and air velocity were varied to obtain the information for dynamic growth of biofilm under different runs. The pH was controlled to between 6.7 and 6.9, and the temperature was held at 21• C. The dissolved oxygen was monitored with an electrode. All measurements of biofilm growth on support particles were conducted at 12 hours intervals. The operating conditions of the experimental runs performed in this study are summarised in Table 3.1.

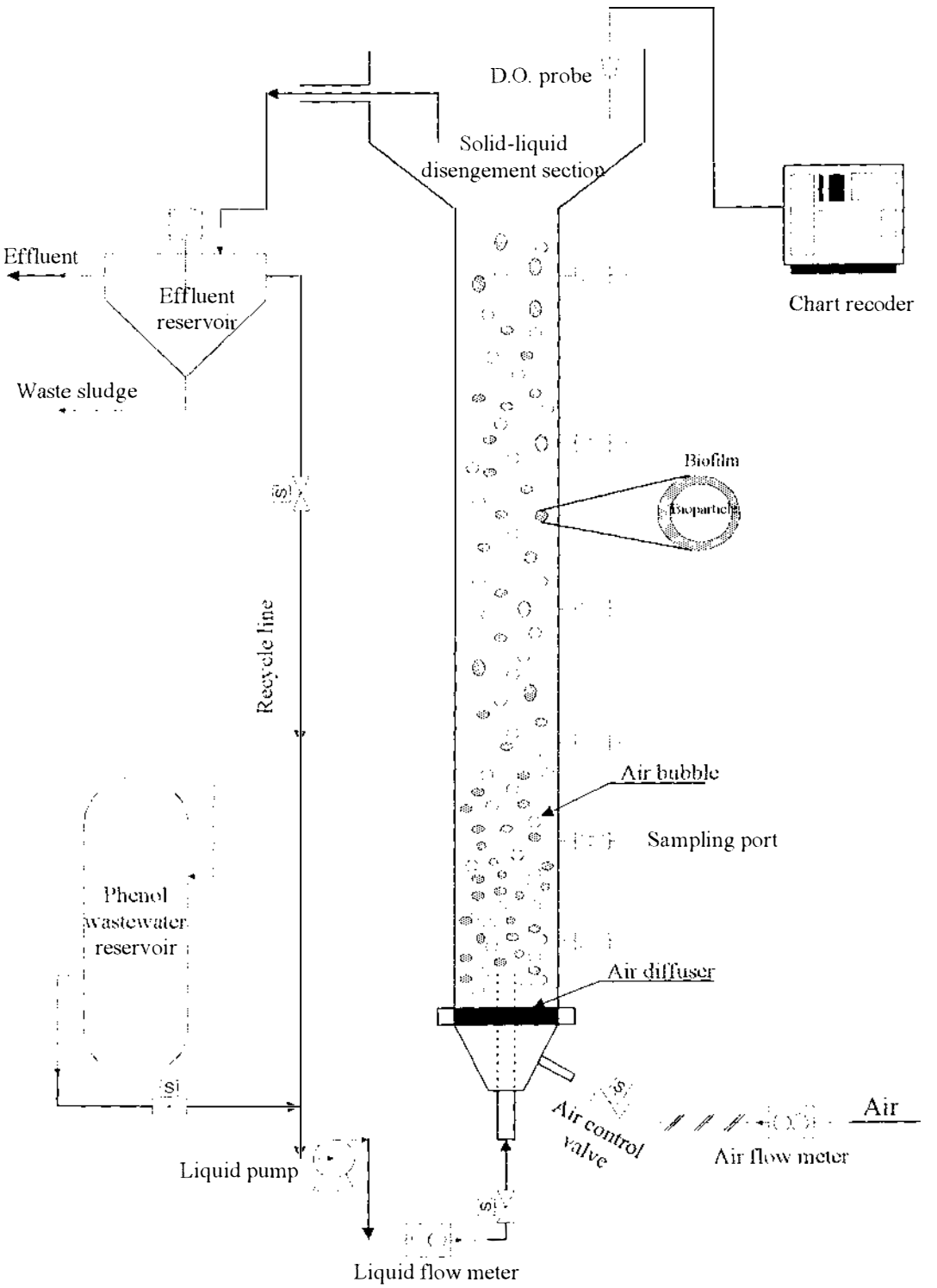


Figure 3.1. Schematic diagram of experimental apparatus.

Table 3.1. Experimental conditions.

Run	Particle size (μm)	Air velocity (cm/sec)	Liquid velocity (cm/sec)	Inlet phenol concentration (mg/cm^3)
1	275	0.36	0.062	150-200
2	400	0.36	0.023	155-185
3	570	0.842	0.078	147-183
4	570	0.48	0.066	151-196
5	600	0.73	0.08	151-190
6	600	0.42-0.7	0.057	143-176
7	710	0.9	0.03	148-184
8	710	1.2	0.03	155-193

3.2.3 Microorganisms and Culture Medium

The biofilm formed on the support particles was a mixed culture which is predominately *Pseudomonas putida* and *Alcaligenes eutrophus* (Bhamidimarri, 1987). This mixed culture was then conditioned to a synthetic growth medium containing phenol as the sole carbon and energy sources. The composition of the synthetic growth medium is shown in Table 3.2. The synthetic growth medium was diluted to yield the desired phenol concentration with fresh distilled water for each experimental run.

Cell suspensions for inoculations were made by mixing the freeze dried stock culture (about a quarter teaspoon) with 50 mL of the synthetic growth medium at a defined phenol concentration. It was incubated at 21° C in 250-mL shake flasks on a rotary shaker at 250 rpm for 24 h. To ensure sufficient oxygen supply the flasks were fitted with lids with holes. After this period, the mixture was left for the filler material to settle. 10 mL of the supernatant liquid was transferred into 40 mL fresh medium and

incubated as earlier. The last step was repeated twice before the culture was regarded as well established.

Table 3.2. Composition of synthetic growth medium.

Component	Concentration (mg/l)
KH_2PO_4	420
K_2HPO_4	375
$(NH_4)_2SO_4$	244
NaCl	30
$CaCl_2 2H_2O$	30
$MgSO_4\ 7H_2O$	30
$FeCl_2H_2O$	4.6
C_6H_5OH	Varying

3.2.4 Analytical Methods

Phenol Assay

Phenol concentration was measured spectro-photometrically by monitoring the optical density 274 *nm* with a spectrophotometer (Phillips PU 8625 UV/VIS, Cambridge, England) equipped with quartz sample cells. This direct method, without protracted sample preparation, could be employed because the aromatic ring of the phenol molecule absorbs UV light. The data collected were then converted from the optical density to concentration units using a previously prepared calibration curve.

Suspended Biomass Concentration

Suspended biomass concentration was measured by determining the optical density of the culture broth. This was done using a spectrophotometer (Phillips PU 8625

UV/VIS, Cambridge, England) at a wavelength of 620 *nm* equipped with quartz sample cells.

Biofilm Thickness

The biofilm thickness was measured with the help of microscopy (Olympus CHA, Olympus Corp., USA) with a previously calibrated eyepiece with ocular scale. For a representative value, the biofilm thickness of each bioparticle was measured at four locations around the bioparticles per sample. To obtain the average biofilm thickness through the height of reactor, at least total 200 bioparticles were taken from 6 sampling ports and measured at 12 hour intervals.

Biofilm Density

The biofilm density is expressed in terms of the dry weight of biomass of the biofilm per unit volume of wet biofilm. The biofilm density was measured in accord with a dry weight measurement based on Tang et al. (1987). The weight of the attached dried biomass plus that of the support particles was measured by drying the bioparticles (at 100° C for 24 hours) taken from the each sampling port. The biofilms were then removed by $NaOH_2$ solution from the support particles. The clean support particles were then washed, dried at 100° C for 24 hours, and weighted. The dry biomass weight was obtained by subtracting the weight of dried support particles from the weight of the dried bioparticle. Finally, the biofilm density was estimated using the following equation:

$$X_a^f = \frac{W_b}{(\pi/6)\left[\left(d_{sp} + 2L_f\right)^3 - d_{sp}^3\right] \cdot N} \tag{3.1}$$

where X_a^f is the biofilm dry density, d_{sp}, the diameter of support particles, N, the total number of support particles in the sample, L_f, the biofilm thickness, and W_b, the dried biomass.

Chapter 4

Mechanistic Model for a TPFBBR Incorporating Dynamic Biofilm Growth

4.1 INTRODUCTION

Several mechanistic model-based biofilm models have been developed in which the substrate transport to consumption by the biofilm is described. In the majority of these models, biofilm properties such as biofilm thickness and biofilm density were assumed to be constant (Atkinson, et al., 1967; Atkinson & Daoud, 1970; Lamotta, 1976; Williamson & McCarthy, 1976; Rittmann & McCarty, 1981). Although steady-state biofilm models for growing biofilms can be quite satisfactory for design purposes, they may insufficient to describe the start-up phases and the dynamic responses to the changing conditions in the systems. These models describe neither the variation in biofilm thickness nor its density in time. Thus a dynamic biofilm growth model reflecting the variability of biofilm thickness and its density as a function of time is necessary for design, optimisation, and control of a TPFBBR. Net biofilm growth rate depends highly on biofilm detachment rate. Therefore, information on the biofilm detachment is required because the performance of biofilm

process is dependent on the quantity of biomass amount on support particles. However, biofilm detachment processes are the least studied and understood.

The aim of this chapter is to develop an integrated model incorporating dynamic biofilm growth for a TPFBBR by taking a mechanistic modelling approach. First, the dynamic biofilm growth model is developed which is able to reflect the simultaneous variability of biofilm thickness and biofilm density as a function of time, derived from a biomass mass balance equation. A biofilm detachment model is also developed as a function of shear loss and attrition. Using the method of characteristics to arrive at a solution for propagation of biofilm thickness and biofilm density in time, the dynamic biofilm growth model with detachment is proposed. In order to form the integrated model, the dynamic biofilm growth model with detachment is combined with a reaction-diffusion model and reactor model to predict the dynamic change of performance in a TPFBBR according to the variation of biofilm thickness and biofilm density. The experimental data are compared with simulation results to show the effectiveness of integrated model and dynamic biofilm growth model developed.

4.2 OVERVIEW OF BIOFILM GROWTH MODEL

4.2.1 Steady-State Biofilm Growth Model

Many steady-state biofilm models describing substrate concentration profiles over biofilms have been developed. In many of these models, zero or first-order kinetics were assumed so that analytical solutions were available (Atkinson, et al., 1967; Atkinson & Daoud, 1970; Williamson & McCarty, 1976; La Motta, 1976; Dalli & Chau, 1987; Rittmann & McCarty, 1981; Harremoes, 1982). However, numerical solutions were obtained (Wanner & Gujer, 1982).

Rittmann & McCarty (1981) modelled biofilm thickness by assuming that a steady-state biofilm model is one in which growth would just be balanced by cellular decay. Later, Rittmann (1982) extended his model to incorporate biofilm detachment which depended on the biofilm thickness and mass as well as upon the shear stress. Andrew

& Tien (1981) developed a biofilm growth model which is similar in concept to the biofilm model of Rittmann. Howell & Atkinson (1974) proposed a biofilm model in which the biofilm thickness increased over time by assuming that no continual detachment occurred so substrate removal would result in accumulate cell mass. In the steady-state biofilm models, the biofilm thickness is assumed to be constant and the biomass distribution over the entire biofilm is also assumed to be homogeneous. In reality, the biofilm thickness and biofilm density are not constant over time, and the biofilm density is not constant over the entire biofilm (Masuda et al., 1992; Zang & Bishop, 1994a; Zang & Bishop, 1994b; Bishop et al., 1995). Gradients develop because the growth rate is dependent on the local substrate concentrations.

The major limitation of the steady- state biofilm models are: (1) that the biofilm thickness of the steady-state biofilm model must be known; and (2) only the steady-state is described, which means that it does not give insight into the response of the processes to change (Wijffels & Tramper, 1995). However, steady-state models provide descriptions of biofilm processes in the case of thick biofilms, in which the penetration depth of substrate is smaller than the biofilm thickness. In this case, accurate information about the total thickness of the biofilm for determination of the capacity is not important. In the case of thinner biofilms formed in a TPFBBR, however, the biofilm thickness is a key parameter to contribute to the reactor performance.

4.2.2 Dynamic Biofilm Growth Model

In the dynamic biofilm growth models, the biofilm thickness and biofilm density will increase or decrease as a function of time. The consideration of the varying biofilm thickness and its density in the biofilm is essential to study the dynamics of biofilm reactors. Modelling of the dynamic biofilm growth has been attempted by several investigators (Benefield & Molz, 1985; Wanner & Gujer, 1986; Jones et al., 1993; Wijffels et al., 1989; Wijffels et al., 1991).

In dynamic biofilm growth models reported in the literature, the biofilm thickness or biofilm density was restricted to a maximum or minimum. If growth is not assumed to be restricted, or if no biofilm is removed, the biofilm will become infinitely thick. As this is not realised in practice, some modellers introduce restrictions such as no growth to a maximum biofilm thickness (Toda & Sato, 1985; Jones et al., 1993) or the assumption that there is biofilm loss at the maximum thickness (Benefield & Molz, 1985; Wanner & Gujer, 1986). In practice, even though the biofilm thickness and biofilm density can vary simultaneously by several orders of magnitudes and exhibit temporal variations quite different from their long-time or steady-state patterns, most of the above models do not describe a simultaneous change of biofilm thickness and biofilm density in time.

4.3 DYNAMIC BIOFILM GROWTH MODEL

The dynamic biofilm growth model presented here is based on the following assumptions:

- Growth is limited by a single substrate and all other nutrients are present in excess.
- Various species of microorganisms present in the biofilm can be described as homogeneous biomass.
- Properties of biofilm change only in the direction perpendicular to the biofilm-support particle interface.
- Axial gradients of biofilm thickness are not significant.
- Growth mechanisms of suspended cells and attached cells are assumed to be identical.
- Biomass within the biofilm is uniformly distributed.

This model is based on a one-dimensional conservation law for the biomass in the biofilm. A mass balance may be written for a differential volume element $A \cdot \Delta L_f$ of the biofilm (Fig. 4.1).

$$A\Delta L_f \frac{\partial X_a^f(L_f,t)}{\partial t} = A\left[F_X(L_f,t)\Big|_{L_f} - F_X(L_f,t)\Big|_{L_f+\Delta L_f}\right] + A\Delta L_f R_X \tag{4.1}$$

where $X_a^f(L_f,t)$ is the biofilm density based on dry weight of biomass as function of time t and biofilm thickness ΔL_f. $F_X(L_f,t)$ is the flux of biomass, and R_X is the rate of change of biofilm density. By dividing Eq. (4.1) by $A \cdot \Delta L_f$ and taking the limit as these dimensions go to zero, we have

$$\frac{\partial X_a^f(L_f,t)}{\partial t} + \frac{\partial F_X(L_f,t)}{\partial L_f} = R_X \tag{4.2}$$

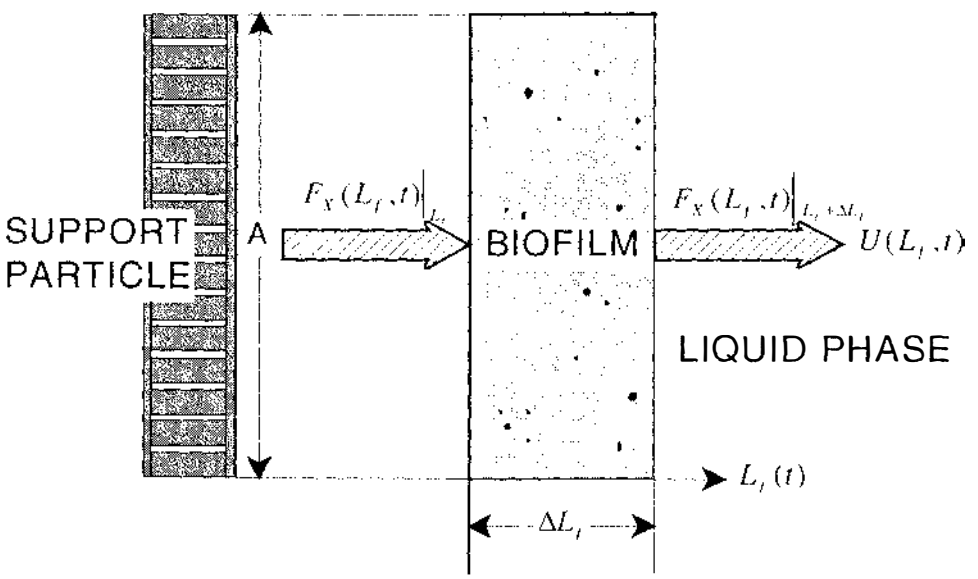


Figure 4.1. Schematic of biofilm growth with a volume element of biofilm.

The biofilm is assumed to consist of a liquid phase and a solid phase. Biofilm liquid phases generally amounts to about 80% of the biofilm volume. It is assumed that as

the metabolic reaction in the biofilm increases, there is an increase in the amount of biomass at the end of a finite time interval which causes an increase in the biofilm solid phase. Biofilm thickness can be considered to change to accommodate the change in the biofilm solid phase and in the biofilm liquid phase. It is also assumed that although the biomass accumulates on the support particles, the biofilm density decreases since the volume expansion of the biofilm results in the decrease in the biofilm solid phase. The flux of biomass that leads to the change of the biofilm thickness can be expressed as

$$F_X(L_f,t) = U(L_f,t)X_a^f(L_f,t) \tag{4.3}$$

Substitution of Eq. (4.3) into Eq. (4.2) leads to

$$\frac{\partial X_a^f(L_f,t)}{\partial t} + \frac{\partial}{\partial L_f}\left[U(L_f,t)\,X_a^f(L_f,t)\right] = R_X \tag{4.4}$$

From the assumption of uniform distribution of biomass, the one-dimensional mass balance equation, which models the biofilm growth as function of time and biofilm thickness, can be expressed as

$$\frac{\partial X_a^f(L_f,t)}{\partial t} + U(L_f,t)\frac{\partial X_a^f(L_f,t)}{\partial L_f} = R_X \tag{4.5}$$

The growth kinetics of biofilm is assumed to follow Monod kinetic. The expression for the specific growth rate is thus:

$$u(C_s^f) = \frac{u_{\max} C_s^f}{K_s + C_s^f} \tag{4.6}$$

where u is the specific growth rate, $u_{\max}$, the maximum specific growth rate, K_s, the Monod constant for substrate. By assuming Monod kinetics, the rate of change of biofilm density can be expressed as

$$R_X = -(u \cdot X_a^f(L_f, t) - R_{de}) \tag{4.7}$$

where R_{de} is the biofilm detachment rate which considers the transfer of biomass from the biofilm phase to the liquid phase.

4.3.1 Detachment Model

4.3.1.1 General Overview of Biofilm Detachment Model

The biofilm detachment is the entrainment of cells from an existing biofilm into the bulk liquid and is the primary process that balances cell growth in a biofilm (Peyton & Characklis, 1993). The biofilm detachment is a complicated function involving several variables, which take into account the hydrodynamics of the liquid and gas flow at the biofilm surface and the biofilm morphology and heterogeneity.

In principle, the biofilm detachment coefficient (b_{de}) can be expressed in terms of the characteristics of the biofilm, the support particle, and the hydrodynamic variables likely to affect detachment

$$b_{de} = f(X_a^f, L_f, C_p, U_L, U_G) \tag{4.8}$$

where C_p is the particle concentration in the reactor.

The biological process performances depend on the biomass amount in a reactor. This is especially true for biofilm process, in which high volumetric loading rates or low effluent concentrations are associated with the ability to accumulate a large biofilm mass in the reactor. The biofilm mass accumulating on support particles is dependent on substrate utilisation, biofilm growth, decay, and biofilm detachment. For design and operation purposes, a model must predict the amount of biomass in a biofilm reactor. Therefore, among the mechanisms controlling biofilm reactor performance, detachment process is one of the least studied and understood. One commonly applied

biofilm detachment model assumes a first-order dependence on biofilm thickness and mass (Chang & Rittmann, 1988).

$$R_{de} = b_{de} X_a^f L_f \tag{4.9}$$

Others have postulated that shear loss rate is a power law (Bakke et al., 1984) or second-order function of biomass (Bryers, 1984). For example,

$$R_{de} = b_{de} (X_a^f L_f)^2 \tag{4.10}$$

Wanner and Gujer (1986) have used that biofilm shear loss rate was a second-order function of biofilm thickness in multispecies population dynamics. Based on an analysis of limited data, Rittmann (1982) suggested

$$R_{de} = b_{de} X_a^f L_f \tau^{0.58} \tag{4.11}$$

where τ is the shear stress.

A first-order dependence on shear stress of the form (Bakke et al., 1990)

$$R_{de} = b_{de} X_a^f \tau \tag{4.12}$$

has also been proposed. Speitel and DiGiano (1987) suggested that growth rates in the biofilm influence shear loss rates and have proposed as expression of the form

$$R_{de} = (b_s + b_s' u) L_f \tag{4.13}$$

where b_s is the biofilm shearing coefficient and b_s' is the dimensionless parameter describing the biological aspects of shearing.

Trinet et al. (1991) suggested the correlation of factors affecting biofilm detachment rate by using multiple regression analysis in a TPFBBR. Chang et al. (1991) also showed the statistical correlation of factors affecting detachment rate in liquid-fluidized bed. Peyton and Characklis (1992) proposed that the detachment rate should

be proportional to the product of the substrate utilization, biomass yield, and biofilm thickness and have proposed as expression of the form

$$R_{de} = b_{de} \frac{Q}{A} (C_s^{in} - C_s^l) Y_s L_f \tag{4.14}$$

where Q is the flow rate, A is the surface area of the reactor, C_s^{in} is the inlet substrate concentration, C_s^l is the substrate concentration in the reactor, and Y_s is the yield coefficient.

Recently, Peyton and Characklis (1993) also showed that the detachment rate is independent of shear stress, but is dependent on the cellular production rate and have proposed as expression of the form:

$$R_{de} = b_{de} u X_a^f L_f \tag{4.15}$$

4.3.1.2 Important Parameters Influencing Detachment in a TPFBBR.

Rittmann (1982) suggested that there is a strong relation of b_{de} to $\tau^{0.58}$ in an annular reactor. But for a two-phase FBBR, Chang et al. (1991) showed that the effect on detachment rate of liquid shear stress (τ) was not significant and for a three-phase FBBR. Trinet et al. (1991) also founded that the range of τ (approximately 9.5 $dyne/cm^2$) was too small to distinguish any effects of τ on biofilm detachment. Bhamidimarri and See (1990) showed that shear stress required for balancing detachment with growth for a phenol degrading biofilm was 19.2 N/m^2 in a rotating cylinder bioreactor. Duddrige et al. (1982) on the other hand reported that a significant biofilm detachment attached to stainless steel occurred at shear stress above 100-120 $dyne/cm^2$. Therefore, as the liquid shear stress, τ present in a TPFBBR is too small to detach the biofilm on the support particles, the effect on detachment of τ may be thought to be not significant. Gjaltema et al. (1995) presented a preliminary survey of

factors affecting biofilm detachment in three-phase biofilm airlift loop reactors. These factors are summarised in Table 4.1. The study of Gjaltema et al. (1995) demonstrated a strong positive relationship between the biofilm detachment rate and the concentration of clean support particles, and between the biofilm detachment rate and the roughness of clean support particles. This study also showed that the biofilm detachment at lower superficial air velocity was not significantly different, but the detachment at higher superficial air velocity was considerably higher. A similar relationship has been observed by Tijhuis et al. (1994) for developing biofilms in a BAS reactor. In a recent study on the effect on biofilm accumulation due to air velocity, Tavares et al. (1996) showed that biofilm accumulation decreases when gas velocity increases. The air velocity therefore is demonstrated to be an important operational variable for biofilm characteristics and biomass accumulation in a TPFBBR.

Table 4.1. Summary of qualitative effects of tested parameters on detachment (Gjaltema et al., 1995).

Parameter	Effect
Biomass decay	negligible
Biofilm pellet batch	not clear
Biofilm storage time	not clear
Pellet morphology	no effect observed, data limited
Pellet diameter	no effect observed, data limited
Bare carrier roughness	important
Bare carrier concentration	important
Flow regime	important
Superficial air velocity	within one flow regime on effect observed, data limited
Reactor geometry	important
Bottom clearance	no effect

In the research into the damage due to liquid shear, turbulence, and particle-particle interactions, Croughan et al. (1988) showed that damage due to the interactions between particles and turbulent eddies was most important at lower support particle concentration; particle-particle interactions alone dominated at higher particle concentration. In a study of investigating biofilm detachment in three-phase fluidized-bed biofilm reactor, Trinet et al. (1991) also observed that increased particle-particle attrition, which is proportional to particle concentration, and increased air turbulence described by Re (Reynolds number), caused the biofilms to be denser and thinner. The detachment rate increased as particle concentration and Re increased. It has also been known that the detachment strongly increased with increase in the liquid velocity. The liquid turbulence may influence the erosion process, in which biofilm is continuously removed from the surface of biofilm in addition to attrition process. Therefore, as mentioned in Tijhuis et al. (1994) the detachment in the biofilm airlift system (BAS) reactor and the two- or three-phase FBBR are most probably governed by particle-particle interactions because the support particle concentration is generally high, whereas in the rotating drum reactor other mechanisms, notably liquid shear stress, are dominating detachment. The biofilm density obtained in a TPFBBR generally had higher values to those from two-phase FBBR because turbulence and particle-particle interactions render the biofilms to be thinner and denser. This is probably due to the fact that cells in the biofilms developed in a TPFBBR pack more tightly and due to considerably higher erosion force attrition forces. In conclusion, the attrition effect due to particle-particle interactions and erosion forces caused by superficial air and liquid velocities must be taken into account in developing a model for biofilm detachment of a TPFBBR.

4.3.1.3 Detachment Model Formulation

The detachment rate in a TPFBBR will be expected to be a positive function of erosion force caused by liquid turbulence and gas turbulence. This erosion effect at the biofilm surface affects the biofilm accumulation and therefore is related to the solid holdup and the bed expansion characteristics because the change of biofilm accumulation (such as biofilm thickness) influences strongly the solid holdup and bed

expansion. Furthermore, different solids holdup (or bed expansion) may affect the detachment rate. Meanwhile the solid holdup is strongly related to the attrition effect due to particle-particle interactions. Generally the effect of attrition effect on the detachment was described as the support particle concentration present in the reactor (Chang et al., 1989, Trinet et al., 1991). A large particle concentration indicates greater particle-particle attrition and it also corresponds to less liquid turbulence. This support particle concentration is closely dependent on the solids holdup, which decreases with increase in the superficial liquid velocity and on the bed turbulence, which causes random movement of the support particles and influences particle-to-particle contacts. The attrition effect on the detachment is proportional to the liquid velocity resulting in turbulence and inversely proportional to the solid holdup in a TPFBBR.

Traditionally the solid holdup (or bed expansion characteristic) has not been considered in the detachment models. In this chapter, the particle concentration resulting in the attrition is defined as the solids holdup. Therefore it may be suggested that major variables to be considered for the description for biofilm detachment in a TPFBBR are the particle concentration as function of the solid holdup, and turbulence due to superficial gas and liquid velocity. Since the shear stress is thought to be poorly suited for studies on the erosion effect for the detachment in a TPFBBR (Tijhuis et al., 1994; Huang et al., 1996), the concept of a velocity gradient (V_G) employed by Amirtharajah (1978) is introduced to describe the turbulence resulting in the erosion force and the attrition. Amirtharajah introduced the concept of a velocity gradient in a fluidized bed first suggested by Camp (1964) and it is defined as

$$V_G = 0.5\left(\frac{E_P}{\mu_L Q_L}\right) \tag{4.16}$$

where V_G is the velocity gradient, E_P, the energy dissipation rate, μ_L, the liquid dynamic viscosity, and Q_L, the liquid volume.

The frictional pressure gradient (ΔP) in a fluidized bed, which is equal to the weight of solids per unit volume of the bed corrected for the pseudo-homogeneous fluid buoyancy, is given by the Ergun equation (1952) applied to the liquid-solid system. Hence,

$$\Delta P = \left(-\frac{dP}{dZ} \right)_f = \varepsilon_s (\rho_{bp} - \rho_L) g = \frac{150 U_L \mu_L \varepsilon_s^2}{d_{bp}^2 (1-\varepsilon_s)^3} + \frac{1.75 U_L^2 \rho_L \varepsilon_s}{d_{bp} (1-\varepsilon_s)^3} \tag{4.17}$$

The Ergun equation assumes that the gas and the liquid together behave as a homogeneous fluid with no relative motion between them. Such an assumption is at best valid only for very small gas/liquid velocity ratios and correspondingly small gas bubbles. The energy dissipation rate (E_p) can be computed from multiplication of the frictional pressure gradient by the liquid flow rate.

$$E_p = \varepsilon_s H_e (\rho_{bp} - \rho_L) g A U_L \tag{4.18}$$

where H_e is the expanded height, A , the surface area of bed.

In summary the biofilm detachment coefficient can be formulated by

$$b_{de} = \psi \frac{V_G \mathrm{Re}_g}{\varepsilon_s} \tag{4.19}$$

where b_{de} is the biofilm detachment coefficient, ψ , a constant, Re_g , Reynolds number as a function of the superficial gas velocity, ε_s , the particle concentration as a function of the solids holdup. The biofilm detachment rate (R_{de}) can be described as

$$R_{de} = b_{de} X_a^f (L_f, t) = \psi \frac{V_G \mathrm{Re}_g}{\varepsilon_s} X_a^f (L_f, t) \tag{4.20}$$

4.3.2 Biofilm Growth Model Incorporating Detachment

Wanner and Gujer (1986) proposed that the expansion velocity of biofilm thickness may be expressed as a linear function of the specific growth rate and biofilm thickness. It follows from this that the shrinkage velocity of biofilm thickness due to detachment is proportional to the increase in biofilm thickness and is a function of biofilm detachment coefficient. Therefore, the rate of change of biofilm thickness can be described as

$$\begin{aligned} U(L_f,t) = \frac{dL_f}{dt} &= uL_f - b_{dc}L_f \\ &= \bar{u}L_f \end{aligned} \tag{4.21}$$

where $\bar{u}$ is the net specific growth rate.

The net change rate of biofilm density can therefore be formulated as

$$\begin{aligned} R_X &= -(uX_a^f(L_f,t) - b_{dc}X_a^f(L_f,t)) \\ &= -\bar{u}X_a^f(L_f,t) \end{aligned} \tag{4.22}$$

Substituting Eq. (4.21) and (4.22) into Eq. (4.5) leads to the resulting mass balance equation for the biofilm growth.

$$\begin{aligned} \frac{\partial X_a^f(L_f,t)}{\partial t} + \bar{u}L_f \frac{\partial X_a^f(L_f,t)}{\partial L_f} \\ = -\bar{u}X_a^f(L_f,t) \end{aligned} \tag{4.23}$$

Let us introduce a new time variable, T by taking

$$\frac{dt}{dT} = \frac{1}{\bar{u}} \tag{4.24}$$

This transformation converts Monod kinetic to first-order kinetic in the new time frame and with this substitution gives

$$\frac{\partial X_a^f}{\partial T} + L_f \frac{\partial X_a^f}{\partial L_f} = -X_a^f \tag{4.25}$$

This first-order partial differential equation can be solved by the method of characteristics (Aris & Amundson, 1973; Hunter & Asenjo, 1990), by parameterizing T, L_f, and X_a^f as arbitrary function of a dummy variable λ. Let us choose

$$\frac{dT}{d\lambda} = 1 \text{ and } \frac{dL_f}{d\lambda} = L_f \tag{4.26}$$

which are the coefficients of $\frac{\partial X_a^f}{\partial T}$ and $\frac{\partial X_a^f}{\partial L_f}$, respectively. Expanding $\frac{\partial X_a^f}{\partial \lambda}$ by the chain rule in terms of $T(\lambda)$ and $L_f(\lambda)$ gives

$$\begin{aligned} \frac{dX_a^f}{d\lambda} &= \frac{\partial X_a^f}{\partial T}\frac{dT}{d\lambda} + \frac{\partial X_a^f}{\partial L_f}\frac{dL_f}{d\lambda} \\ &= \frac{\partial X_a^f}{\partial T} + L_f \frac{\partial X_a^f}{\partial L_f} = -X_a^f \end{aligned} \tag{4.27}$$

Solving for L_f, T, and X_a^f as functions of λ, gives

$$T = T_0 + \lambda \tag{4.27a}$$

$$L_f = L_{f_0} \exp(\lambda) \tag{4.27b}$$

$$X_a^f = X_{a_{\max}}^f \exp(-\lambda) \tag{4.27c}$$

The new time variable T denotes time, so $T_0 = 0$. Application of Eq. (4.27) could lead to an infinitely low biofilm density and high biofilm thickness. As this is not realized in practice, the following constraints are added to this equation.

$$L_{f_0} \le L_f \le L_{f_{\max}} \tag{4.28a}$$

$$X^f_{a_{\max}} \le X^f_a \le X^f_{a_{final}} \tag{4.28b}$$

where $X^f_{a_{final}}$ and $L_{f_{\max}}$ are the observable final biofilm density and maximum biofilm thickness. Finally the solution X^f_a and L_f may be recast in terms of real time t by substituting equation (4.24).

$$L_f = L_{f_0} \exp((u - \psi \frac{V_G Re_g}{\varepsilon_s}) \cdot t) \qquad L_{f_0} \le L_f \le L_{f_{\max}} \tag{4.29}$$

$$X^f_a = X^f_{a_{\max}} \exp(-(u - \psi \frac{V_G Re_g}{\varepsilon_s}) \cdot t) \quad X^f_{a_{\max}} \le X^f_a \le X^f_{a_{final}} \tag{4.30}$$

Eq. (4.29) and Eq. (4.30) shows that the biofilm thickness between L_f and $L_f + \Delta L_f$ increases exponentially, so that biofilm density at each biofilm thickness decreases exponentially toward its final value. The dynamic biofilm growth model presented here can describe simultaneously the change of biofilm thickness and biofilm density in time.

4.4 REVEIW OF MATHEMATICAL MODELLING OF A TPFBBR

The modelling of a liquid-solid, two-phase FBBR has been attempted by several investigators (Mulcahy, 1978; Ying and Weber, 1979; Shieh, 1980; Mulcahy et al., 1981). The microbial growth kinetics considered in the above models were assumed to be either zero-order or first-order, or Monod kinetics.

In the case of a TPFBBR, Park et al. (1984) presented a model for a TPFBBR for penicillin production which utilized substrate inhibition kinetics. However, they considered only the limiting cases of complete mixing and plug flow, and the liquid-

solid mass transfer resistance was neglected. Tang & Fan (1987) developed a model for a draft-tube TPFBBR for phenol degradation which considered doubled - substrate limiting kinetics, with Haldane type substrate inhibition kinetics for the phenol and Monod kinetics for oxygen. In their model, the liquid and solid phases in the reactor were assumed to be completely mixed due to the large degree of internal circulation within the draft-tube TPFBBR. Wisecarver & Fan (1989) presented a model for a conventional TPFBBR for a phenol degradation that used double-substrate-limiting kinetics. The model they presented included the effect of gas-liquid and liquid-solid mass transfer, axial dispersion of the liquid phase, and simultaneous diffusion and reaction with the biofilm. However, they assumed that the solids were well mixed and steady-state growth conditions were maintained, implying that the concentration on the surface of the biofilm was constant throughout the reactor. These assumption, together with the assumption that the axial dispersion coefficient was constant throughout the reactor, allowed them to develop an analytical solution for the bulk fluid concentration profile though the biofilm to determine the surface concentration. Then, using a material balance, they determined the reactor exit concentration. Livingston and Chase (1989) also presented a model for draft-tube TPFBBR for phenol degradation which considered double-substrate limiting kinetics, but, did not include gas phase material balance in their model. Recently, Petersen and Davison (1995) developed a model of tapered-bed TPFBBR to convert glucose to ethanol. In their model, the concentration profiles in the bulk fluid were determined as a function of the axial-bed position. To do so, a dispersed-plug flow model was employed. They did not account for the fact that the dispersion coefficient was a function of the flow rate of the gas through a cross section of the bed and that the bed might not have a constant cross section but rather be tapered.

4.5 INTEGRATED MODEL DEVELOPMENT

The integrated model for TPFBBR presented here is divided into three submodels:

- The **Biofilm Growth Model**, which considers accumulation of biomass and expansion or shrinking of the biofilm thickness through metabolic

reaction by microorganisms attached to support particles and biofilm detachment.

- The **Reaction-Diffusion Model**, which considers a simultaneous intra-biofilm reaction-diffusion.
- The **Reactor Model**, which considers axially dispersed transport of substrate and suspended biomass as a function of bed height.

4.5.1 Reaction-Diffusion Model

The concentration profiles of substrate describing the simultaneous transport and removal within the biofilm, C_s^f, lead to the following equations.

$$\frac{\partial C_s^f}{\partial t} = D_{sf}\left[\frac{\partial^2 C_s^f}{\partial L^2} + \frac{2}{L}\left(\frac{\partial C_s^f}{\partial L}\right)\right] - \frac{1}{Y_{x/s}}\frac{u_{\max} C_s^f}{K_s + C_s^f} X_a^f(L_f, t) \tag{4.31}$$

where L is the radial position within the bioparticle, D_{sf} is the effective diffusion coefficient in the biofilm, $Y_{x/s}$ is the growth yield coefficient of substrate, K_s is the saturation constant of substrate, $u_{\max}$ is the maximum specific growth rate, L_f is the biofilm thickness, and X_a^f is the biofilm density.

Eq. (4.31) is subject to boundary conditions specifying a zero intraparicle substrate concentration at the start of the experiment, substrate transfer from the liquid phase to the biofilm based on a liquid-solid mass transfer coefficient (under conditions of time-dependent substrate concentration in the liquid phase) and no substrate flux at the centre of the support particle. Thus,

$$\begin{aligned} &\frac{\partial C_s^f}{\partial L} = 0 && \text{at } L = 0 \\ &D_{sf}\frac{\partial C_s^f}{\partial L} = k_{l,s}(C_s^l - C_s^f) && \text{at } L = L_f(t) \\ &C_s^f(L_f, 0) = 0 && \text{at } t = 0 \end{aligned} \tag{4.32}$$

where k_{ls} is the liquid-solid mass transfer coefficient and C_s^l is the phenol concentration in the liquid phase.

4.5.2 Reactor Model

In a three-phase fluidised-beds, the gas phase may often be assumed to be plug flow. Appreciable backmixing, however, may occur in the liquid phase, especially for beds of small particles in concurrent three-phase fluidization. The backmixing of the liquid and solid particles in such a bed is primarily caused by the rising motion of coalesced large gas bubbles. An axial dispersion model has been most commonly used to describe the backmixing behaviour of the liquid phase and to simulate substrate removal in a TPFBBR (Wisecarver & Fan, 1989; Petersen & Davison, 1995). A mass balance for substrate in the liquid phase, C_s^l, yields the following equations.

$$\frac{\partial C_s^l(Z,t)}{\partial t} = E_L \frac{\partial^2 C_s^l}{\partial Z^2} - \frac{U_L}{\varepsilon_l}\frac{\partial C_s^l}{\partial Z} - \frac{k_{ls} a_p \varepsilon_s}{\varepsilon_l}(C_s^l - C_s^s) \tag{4.33}$$

The corresponding boundary and initial conditions for C_s^l are

$$\begin{aligned} U_L C_s^l(0,t) - \varepsilon_l E_L \frac{\partial C_s^l(0,t)}{\partial Z} &= U_L C_s^{in} \quad \text{at } Z = 0 \\ \frac{\partial C_s^l(H,t)}{\partial Z} &= 0 \quad \text{at } Z = H \\ C_s^l(Z,0) &= C_{s0}^l \quad \text{at } t = 0 \end{aligned} \tag{4.34}$$

where C_s^l is the phenol concentration in liquid phase, Z is the axial distance from the bottom of bed, E_L is the axial dispersion coefficient, U_L is the superficial liquid velocity, ε_l is the liquid holdup, ε_s is the solid holdup, a_p is the biofilm surface area per unit volume of solids, C_s^{in} is the inlet phenol concentration, and k_{ls} is the liquid-solid mass transfer coefficient.

The Danckwerts conditions express the phenomenon that, at the entrance, the rate by which the substrate is fed in the reactor is equal to the rate at which it crosses the $Z=0$ plane through combined convection and diffusion. The zero gradient condition at the exit expresses the fact that no substrate conversion can take place outside of a TPFBBR. While the biofilm growth model and reaction-diffusion model are linked through biofilm thickness and biofilm density, the reaction-diffusion model and reactor model are also coupled through C_s^l.

4.5.3 Hydrodynamic Parameters

In order to simulate the performance of a TPFBBR, estimation of the phase holdups, gas-liquid and liquid-solid mass transfer coefficients, the liquid phase axial dispersion coefficient, the minimum fluidization velocity, and the expanded bed height under various conditions are essential. These hydrodynamic parameters are closely linked to the characteristics of biofilm (such as the biofilm density and its thickness). As the biofilm grows on the support particles, the characteristics of biofilm are changed and influence the hydrodynamic conditions of the bed. Once the biofilm thickness is predicted in the dynamic biofilm growth model with the biofilm detachment, the diameter of the biofilm-covered particle is given by

$$d_{bp} = d_{sp} + 2\,L_f \tag{4.35}$$

where d_{bp} and d_{sp} are the diameter of the clean and biofilm-covered particle, respectively. Then the density of the biofilm-covered particle can be estimated from the following equation

$$\rho_{bp} = \rho_{bw} + (\rho_{sp} - \rho_{bw})\left(\frac{d_{sp}}{d_{bp}}\right)^3 \tag{4.36}$$

where ρ_{bp} is the biofilm-covered particle and ρ_{sp} is the density of the clean particle.

The wet density of biofilm (ρ_{bw}) can also be calculated from the following equation (Mulcahy, 1978)

$$\rho_{bw} = \rho_L + \left(\frac{\rho_b}{1000 \cdot 0.8} \right) \tag{4.37}$$

where ρ_L is the density of liquid and ρ_b is the density of the biofilm. In Eq. (4.37), 0.8 means the ratio of volatile to total biofilm solid.

4.5.3.1 Axial-Dispersion Coefficient

The axial-dispersion model used for estimation of substrate gradient as function of bed height can be applied for Reactor Model in a TPFBBR. In the axial-dispersion model, the axial-dispersion coefficient (E_L) describing the backmixing behaviour of the liquid phase is very important and many correlations for E_L have been suggested (Kato et al., 1972; Kim et al., 1975; Kim et al., 1992). To predict E_L, the correlation given by Kim et al. (1992) is used in this work since their correlations cover a wide range of literature data and can be applied for a TPFBBR with small, low density particles. Generally, E_L increases with increasing gas and liquid velocities in three-phase fluidized beds of small particles whereas the effect of liquid surface tension and liquid viscosity on E_L are found to be small. E_L decreases with increasing particle size and it sharply increases with increasing column size (Kim et al., 1992)

$$Pe_L = \frac{d_{bp} \cdot U_L}{E_L} = 20.19 \left(\frac{d_{bp}}{D_c} \right)^{1.66} \left(\frac{U_L}{U_L + U_G} \right)^{1.03} \tag{4.38}$$

where Pe_L is the Peclet number.

4.5.3.2 Liquid-Solid Mass Transfer Coefficient

The liquid-solid mass transfer resistance between the liquid phase and the surfaces of biofilms has been regarded as negligible in most of the mathematical modelling for a FBBR (Ying & Weber, 1979; Mulcahy et al., 1981; Park et al., 1984). However, neglecting this resistance may significantly affect the accuracy of the evaluation of the rate of overall substrate removal rate when the substrate loading is low. Tang & Fan (1989) found that larger than 15% error in phenol biodegradation rate was obtained if the liquid-solid mass transfer coefficient (k_{ls}) was neglected. Most reported studies for the estimation of k_{ls} in a three-phase fluidized bed have used particles with densities ranging from 2000 to 2800kg/m^3 while most TPFBBR use particles with densities ranging up to 1300kg/m^3. The results of Arters & Fan (1986) showed that k_{ls} in a three-phase fluidized bed increased with increasing gas velocity and was independent of liquid velocity. It was also independent of particle size at low gas velocities, but positively dependent on particle size at high gas velocities. Recently, Nore et al. (1992) studied hydrodynamics, gas-solid and liquid-solid mass transfer with in a TPFBBR with the low density ranging from 1300 to 1700kg/m^3. In their studies, increasing the gas velocity increased k_{ls}, especially at low gas velocities for low particle densities and the liquid velocity had almost no effect on the liquid-solid mass transfer coefficient. Nore et al. correlation for estimation of k_{ls} with good prediction of the 250 values measured are

$$k_{ls} = 1.10\left(\frac{U_L}{\varepsilon_l}\right)^{0.43} d_{bp}{}^{0.24} \quad \text{at} \quad U_G = 0$$

$$k_{ls} = 0.197\left(\frac{U_L}{\varepsilon_l}\right)^{0.051} d_{bp}{}^{0.17} \quad \text{at} \quad U_G \neq 0 \qquad (4.39)$$

Typical range for k_{ls} with d_{sp} value of 1130kg/m^3 varied between 0.0003 and 0.0013 cm/s.

4.5.3.3 Phase Holdups

When injecting gas into liquid-solid fluidized bed, rising bubbles within the gas phase changes the hydrodynamic behaviour and two important variables affected by the gas injection which are the gas holdup and the liquid backmixing. The accurate estimation of phase holdups in a TPFFBR is very important because the solids holdup reflects the bioparticle concentration, and also the gas holdup and liquid holdup influence mass transfer. Thus, the performance of a TPFBBR is highly influenced by the phase holdups. The phase holdup behavior in a TPFBBR is dependent on the following factors:

- Biofilm properties, such as thickness, density, and surface roughness.
- Particle properties, such as size, density, and wettability.
- Fluid properties, such as gas and liquid velocities, surface tension, and viscosities.
- Reactor properties, such as bed geometry

The overall phase holdups in a three-phase fluidised-bed can be obtained through the following equations

$$\Delta P = H_e (\varepsilon_l \rho_L + \varepsilon_g \rho_G + \varepsilon_s \rho_{bp}) g$$
$$\varepsilon_l + \varepsilon_g + \varepsilon_s = 1 \qquad (4.40a,b)$$

where ΔP is the pressure drop across the bed, H_e is the expanded bed height, ρ_G is the density of gas, and g is the gravitational acceleration.

The method of phase holdup measurement is based on the assumption of a homogeneous fluidized bed. In other words, it is assumed that there is no axial or radial variation of phase holdup. In reality, there are significant variations of phase holdups in a TPFBBR, particularly at high fluidizing velocities (Lee & de Lasa, 1987; Tang & Fan, 1989). In addition, when high fluidizing velocities are used the surface of the fluidized bed becomes ambiguous and fluctuant. In this case a higher level of error may result in the use of Eq. (4.40).

The hydrodynamic models so far presented to assess the holdup behaviour can generally be described by the wake model, which takes into account the role of the wake behind the bubble (Ostergaard, 1968). In the wake model, the role of the wakes behind the gas bubbles in the liquid flow is mathematically formulated to elucidate the solid holdup (or bed expansion) and the liquid holdup behaviour. The wake concept considers the three-phase fluidized bed to be composed of: (1) the gas bubble region, (2) the wake region, and (3) the liquid-solid fluidization region. The various correlations for phase holdups based on the wake model have been developed (Ostergaard, 1968; Bhatia & Epstein, 1974a; Darton & Harrison, 1976). The differences among these correlations lie in the assumptions made in the model for the solids concentration in the wake region, the correlations for the bed porosity in the liquid-solid fluidization region, and the correlations for the gas holdup or bubble velocity. There are some uncertainties, however, regarding the wake structures and the values of the parameters required in this approach. On the other hand, purely empirical correlations for the individual phase holdups has also been developed by many authors (Ostergaard & Michelsen, 1969; Dakshinamurty et al., 1971; Kim et al., 1972, 1975; Soung, 1978; Begovich & Watson, 1978a; Kato et al., 1981; Lee & de Lasa, 1987). Since the gas and liquid velocities can be considered constant in present work, average values for the phase holdups may also be used.

In order to estimate the phase holdups in this study, the purely empirical correlations suggested by Begovich & Watson (1978a) are chosen because of the large data base from a wide variety of particles which they used for their correlation. The gas and solids holdup are estimated from the following correlations, respectively.

$$\varepsilon_g = (0.048 \pm 0.010) U_L^{0.72 \pm 0.028} d_p^{0.168 \pm 0.061} D_c^{-0.125 \pm 0.088} \tag{4.41}$$

$$\varepsilon_s = 1 - 0.371 U_L^{0.271} U_G^{0.041} (\rho_{bp} - \rho_L)^{-0.316} d_{bp}^{-0.268} \mu_L^{0.055} D_c^{-0.033} \tag{4.42}$$

The liquid holdup and bed porosity can also be calculated from

$$\varepsilon_l = 1 - \varepsilon_g - \varepsilon_s$$
$$\varepsilon = 1 - \varepsilon_s \qquad (4.43a,b)$$

The solids holdup influences the biofilm specific surface area (a_p), and it can be estimated as

$$a_p = \frac{6\,\varepsilon_s}{d_{bp}} \qquad (4.44)$$

4.6 RESULTS AND DISCUSSION

4.6.1 Computer Program

The coupled biofilm growth, reaction-diffusion, and reactor models represent two ordinary differential equations (ODEs) and a set of second-order partial differential equations (PDEs). These equations can not be solved analytically because the above models consist of PDEs with nonlinear reaction terms. Therefore, numerical techniques must be employed. Typical numerical techniques are available for solving these types of PDEs dependent on spatial variables and a time variable, including finite difference method (FDM) and the method of weighted residuals. Typically, FDM has been extensively applied to obtain numerical approximation for the biofilm processes (Ying & Weber, 1979; Wang & Tien, 1984; Tang & Fan, 1987). Among several methods of weighted residuals, the orthogonal collocation technique developed by Villadsen & Stewart (1967) has also used to solve PDEs for biofilm-reactor models (Kim et al., 1978; Speitel et al., 1987; Livingston & Chase, 1989; Huchinson & Robinson, 1990). It is difficult to determine which of these techniques is best suited. This is because the numerical stability and convergency properties not only depend on the nature of the equations, but also are strongly influenced by the parameter values. For example, in a practical application for a TPFBBR, if the solution to the problem has a steep gradient such as the substrate gradients within the biofilm, FDM is more appropriate since functions with steep gradients are better approximated by FDM than by the orthogonal collocation method. On the other hand,

if the solution does not have a steep gradient like substrate gradient in the liquid phase, the orthogonal collocation method is preferred for spatial approximation. It can also be computationally faster and more accurate than FDM.

In this work, a moving-grid method developed by Furzeland et al. (1990) is used to solve nonlinear time-dependent PDEs having solutions with steep gradients in space and time. This method is based on a Lagrangian description of the PDEs and a smoothed equidistribution principle to define the grid positions at each time level. It has been coupled with the method of lines to form a semi-discrete approximation of the original PDEs by providing centered differencing in the spatial variables. The original PDEs by using a spatial discretization method, are then transformed into a system of the differential algebraic equations (DAEs). The resulting nonlinear systems of DAEs are solved to obtain a numerical solution to the original PDEs by DASSL solver (Petzold, 1983), which is an excellent algorithm for all stiff problems. In this study, the numerical solution is developed by Fortran language using Microsoft FORTRAN Powerstation™ 4.0.

4.6.2 Computer Simulation Scheme of Integrated Model

Once the biofilm thickness and biofilm density at a given time are known through Eq. (4.29) and Eq. (4.30), substrate distribution inside the biofilm can be calculated and the substrate concentration in the liquid phase can also be calculated within a single time step. Starting with initial values for X_a^f and L_f, these results in substrate concentration profiles in the biofilm and substrate concentration gradients in the liquid phase. The basic design methods of a TPFBBR have been proposed as shown in Fig. 4.2 and all simulations carried out by this method. All parameters used in the simulation were obtained from the literature and are summarized in Table 4.2.

Table 4.2. Values of parameters used in simulation.

Parameters	Value	Reference
u_{max}	$7.2\times10^{-6}\,s^{-1}$	The present work
$Y_{x/s}$	0.38	The present work
K_s	0.0091 mg/cm^3	The present work
ρ_{sp}	1.4 mg/cm^3	-
d_{sp}	Varying	-
D_{sf}	$0.275\times10^{-5}\,cm^2/s$	Worden & Donaldson (1987)
H	74 cm	-
D_c	4.4 cm	-
U_L	0.08 cm/s	-
U_G	0.73 cm/s	-
C_s^{in}	0.2 mg/cm^3	-

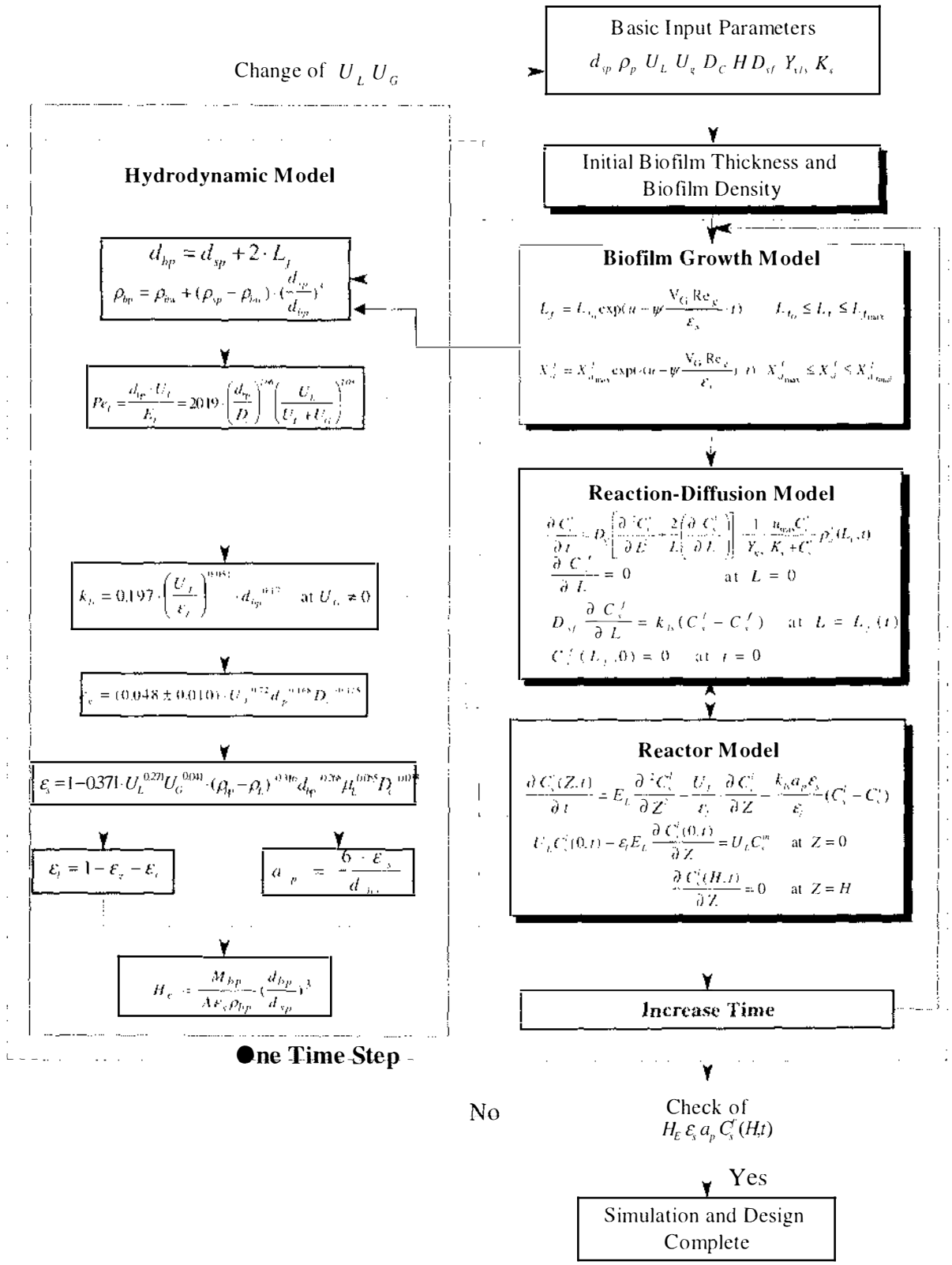


Figure 4.2. Design and simulation method for a TPFBBR.

4.6.3 Model Simulation Results

To investigate the validity of an integrated model incorporating dynamic biofilm growth model, simulation results were compared to the experimental data. Several sets of operating conditions were examined as shown in Table 3.1 having a variety of air velocity, liquid velocity, inlet phenol concentration, and particle size.

Table 4.3. Operating conditions for model simulations.

	Run 1	**Run 2**	**Run 3**	**Run 4**
Particle Size (μm)	570	570	710	275
Air velocity ($cm/\sec$)	0.48	0.842	1.2	0.36
Liquid Velocity ($cm/\sec$)	0.066	0.078	0.03	0.062

First, the dynamic biofilm growth model with detachment was evaluated by comparing simulations with the experimental data. Fig. 4.3-4.6 compared the simulation results of the dynamic biofilm growth model and experimental data for biofilm thickness and biofilm density. In our dynamic biofilm growth model, the average biofilm thickness needs to be restricted to a maximum value while the average biofilm density has to be limited in a minimum value (Eq. (3.29) and Eq. (3.30)). Table 4.4 shows the boundary conditions used in dynamic biofilm growth model for each simulation. In Table 4.4, the values of biofilm detachment coefficient are estimated by the parameter estimation technique using the Levenberg-Marquardt optimisation algorithm.

Through Fig. 4.3-4.6, we describe the case of Fig. 4.3 since the time variation of biofilm thickness and biofilm density for each set of operating conditions has nearly the same pattern, but maximum biofilm thickness and minimum biofilm density are

different. Fig. 4.3. depicts a typical time progression of the average biofilm thickness and biofilm density with the simulation result of dynamic biofilm growth model.

Table 4.4. Boundary conditions for each biofilm growth model simulation.

		Run 1	Run 2	Run 3	Run 4
Biofilm Thickness (μm)	Initial value	13	11	17	7.8
	Finial Value	165	150	230	100.5
Biofilm Density (mg/cm^3)	Initial Value	133	174	285	198
	Finial Value	25	20	32.9	31.6
Detachment Coefficient		0.0719	0.063	0.068	0.065

Data for Fig. 4.3 are taken from operating run 1 (see Table 4.3) at an air velocity of 0.48 $cm/\sec$, a liquid velocity of 0.066 $cm/\sec$, and a particle size of 570 μm. From Fig. 4.3, the lag phase (within 50 hours after startup) of the growth of biofilm on the support particles after startup of the TPFBBR is observed. This lag time could be due to adsorption of dissolved organics and microbial cells onto the surface of the support particle, and the irreversible attachment of microorganisms to the support surface and growth of the biofilm to a measurable thickness. In the initial stage, a small and uniform biofilm of detectable thickness could be observed on the edges of support particles and the biofilm grows as dense, thin, uniformly distributed biomass matrix which lacks the voids present in the thick biofilm. After the lag phase, the biofilm thickness increased rapidly and then levelled off at approximately 150 μm from 380 hours of operation. During the initial stage of biofilm formation (startup - 70 hours), the increased biofilm thickness observed causes an increase of biofilm density upto certain value, although this phenomena was not observed during early stages of the

biofilm. From 70 hours after startup, as the biofilm grows on the support particle, the biofilm density decreased with the increasing biofilm thickness and levelled off at approximately 25 mg/cm^3, corresponding to 150 μm of the biofilm thickness.

It is evident from Fig. 4.3-4.6, the biofilm density is greatly influenced by the biofilm thickness. The biofilm density decreased rapidly during the period of increasing biofilm thickness ranging from approximately 30 μm to 110 μm and stabilised in the range between 20 and 50 mg/cm^3. It is clear that the observed decrease in biofilm density with increasing biofilm thickness was significant and there exists the inverse relationship between biofilm thickness and its density during the formation of biofilm growth. Our experimental results are in accordance with several other literature reports (Hoehn & Ray, 1973; Fan et al., 1987; Zang et al., 1995; Tanyolac & Beyenal, 1997).

From Fig. 4.3-4.6, the biofilm formation in a TPFBBR is slow since the 60 - 90% of biomass was detached continuously from the support particles. Table 4.4 shows biofilm detachment coefficient values used in dynamic biofilm growth model. Dynamic biofilm growth model is able to predict the variation of biofilm density with increasing biofilm thickness. Using Eq. (4.29) and Eq. (4.30) of dynamic biofilm growth model, high biofilm density for thin biofilms and low biofilm density for thicker biofilms was predicted. Fig. 4.3-4.6 show that dynamic biofilm growth model gives good predictions of biofilm density for biofilm thicknesses greater than 15-20 μm and describes the inverse significance of relationship between biofilm thickness and biofilm density during the biofilm formation stages.

Despite the relative mathematical simplicity of the proposed dynamic biofilm growth model in a TPFBBR, the model predictions agree well with experimental data for biofilm thickness and biofilm density. The ability of the dynamic biofilm growth model to describe the time variation of biofilm thickness and biofilm density indicates the soundness of the dynamic biofilm growth model and its usefulness for micro-scale-based modelling of a biofilm reactor.

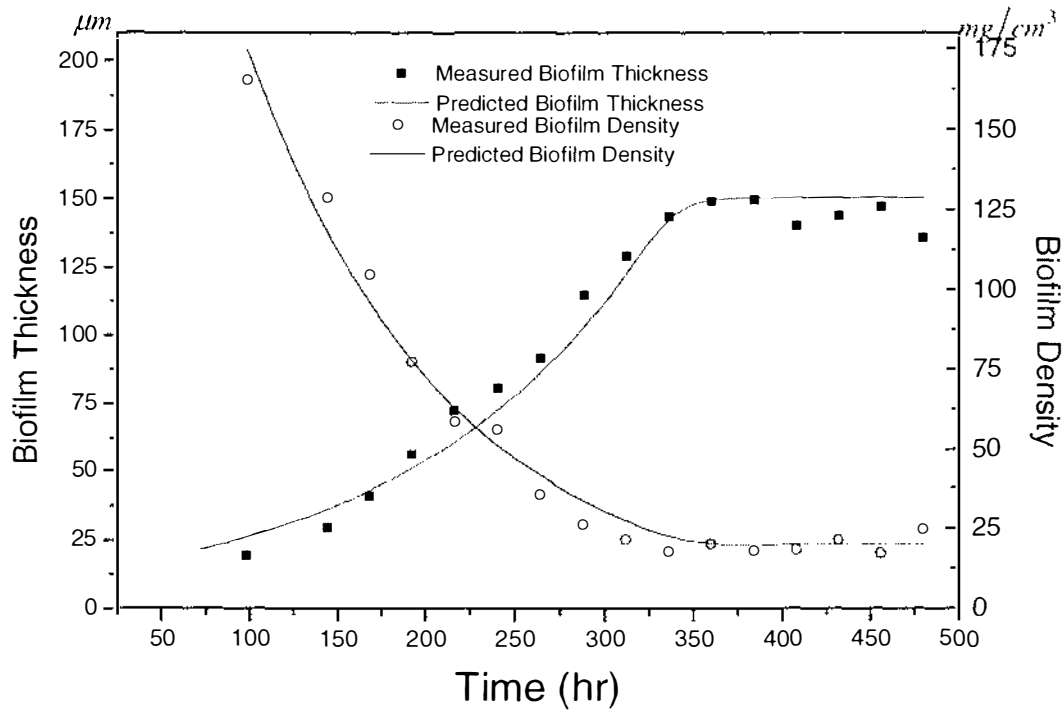


Figure. 4.3. Time progression of the average biofilm thickness and biofilm density ($U_L = 0.066\ cm/\sec$, $U_G = 0.48\ cm/\sec$, $d_{sp} = 570\ \mu m$).

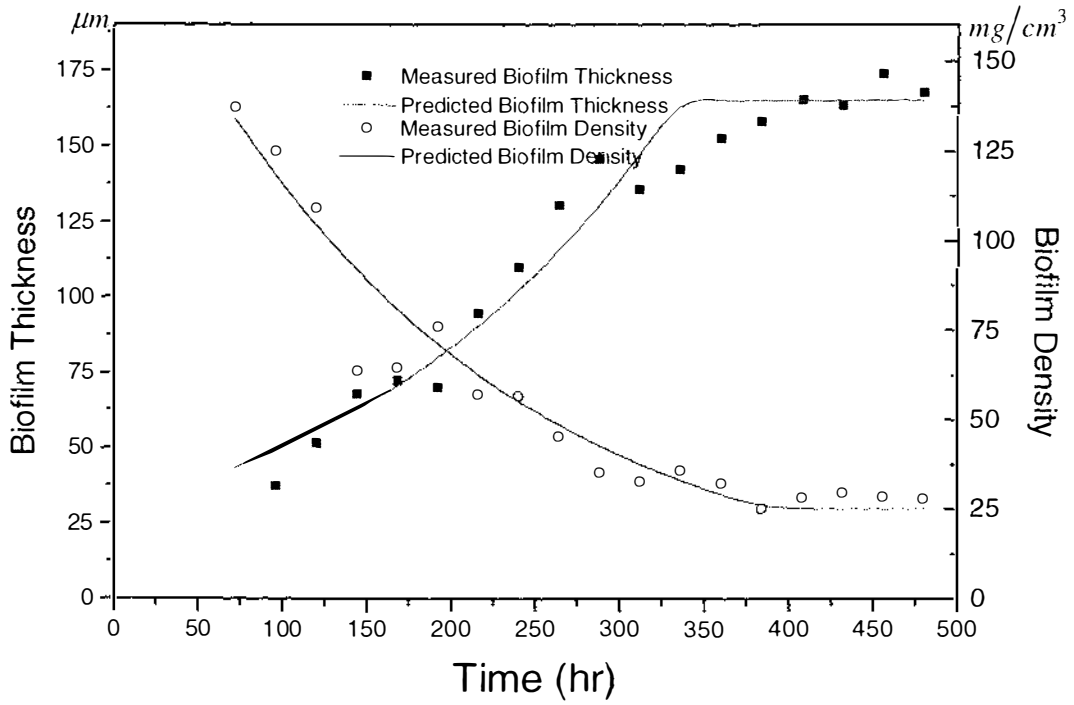


Figure. 4.4. Time progression of the average biofilm thickness and biofilm density ($U_L = 0.078\ cm/\sec$, $U_G = 0.842\ cm/\sec$, $d_{sp} = 570\ \mu m$).

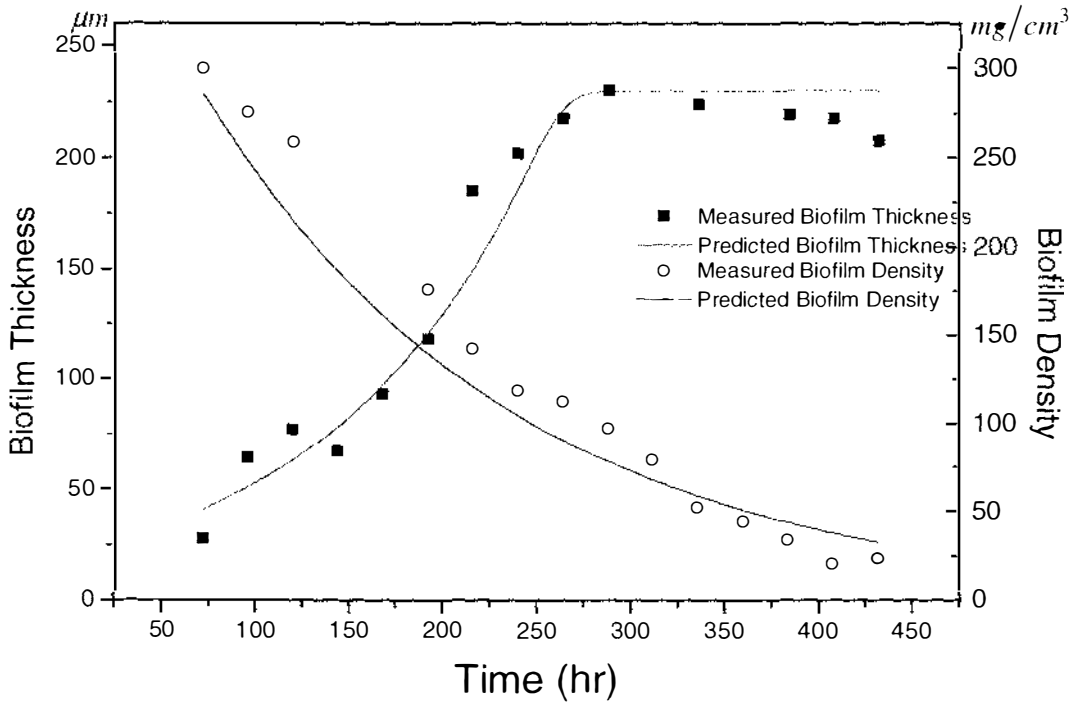


Figure 4.5. Time progression of the average biofilm thickness and biofilm density ($U_L = 0.03\ cm/\sec$, $U_G = 1.2\ cm/\sec$, $d_{sp} = 710\ \mu m$).

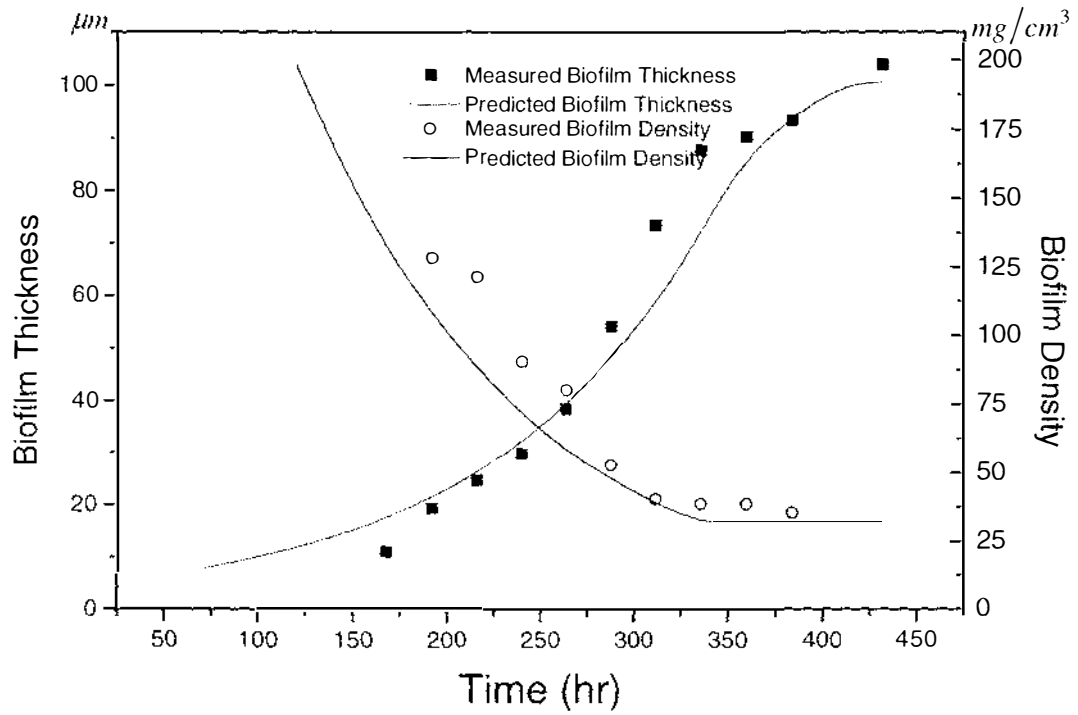


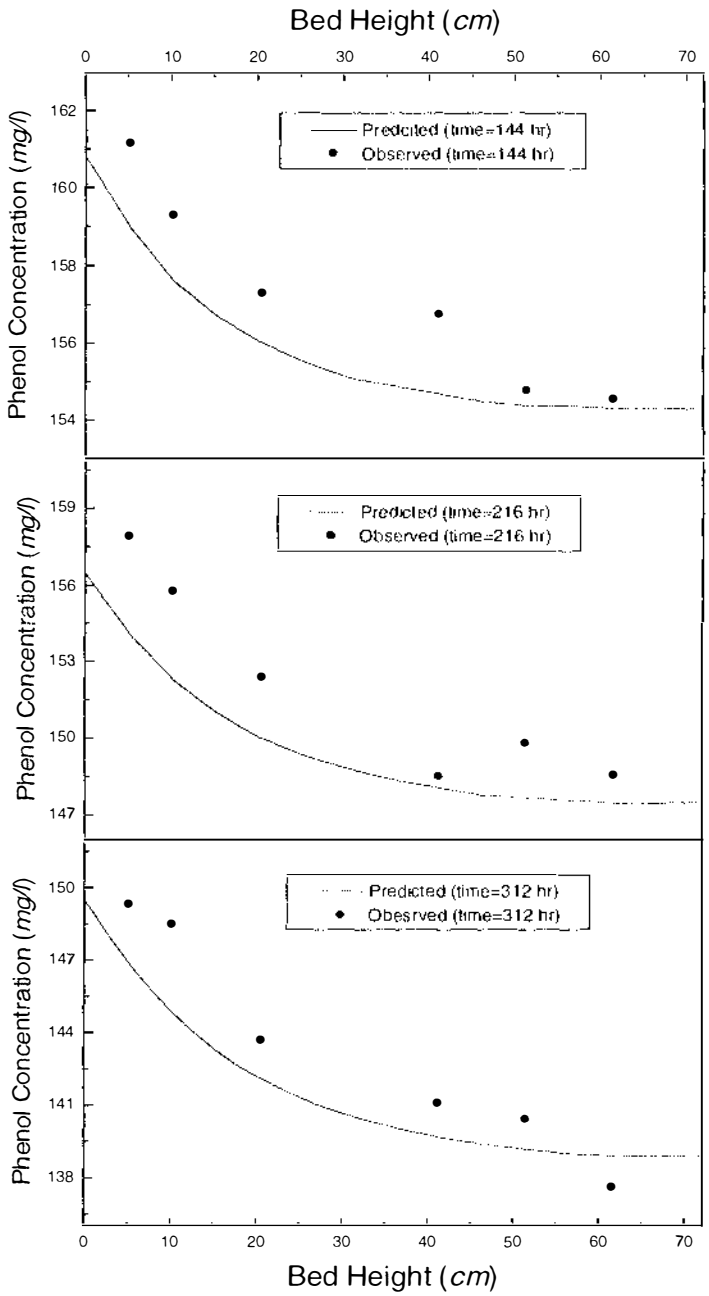
Figure 4.6. Time progression of the average biofilm thickness and biofilm density ($U_L = 0.062\ cm/\sec$, $U_G = 0.36\ cm/\sec$, $d_{sp} = 275\ \mu m$).

The biofilm thickness and its density predicted by the dynamic biofilm growth model are coupled with the reactor model and reaction-diffusion model as parameters of theirs (see Fig. 4.2). These biofilm thickness and biofilm density are also used for the estimation of some parameters such as the axial dispersion coefficient (Eq. (4.38)), the liquid-solid mass transfer coefficient (Eq. (4.39)), and the phase holdups (Eq. (4.41)-Eq. (4.43)).

The phenol concentration profiles in the liquid phase as a function of bed height at different times under two different sets of operating conditions, are predicted from the reactor model. These are shown in Fig. 4.7 and Fig. 4.8. It can be observed that the prediction of phenol concentration profiles as a function of bed height at the different times agrees well with the experimental data. As shown in Fig. 4.7-4.8, during the biofilm growth after startup, the phenol removal rate and phenol concentration profiles as a function of the bed height vary over wide ranges along with corresponding variations in biofilm thickness and biofilm density. When the axial-dispersion model-based reactor model is applied to predict the axial concentration profile in the liquid phase, it predicts that the concentration at the entrance of the reactor is much lower than the inlet concentration because of boundary condition given in Eq. (4.34) which describe a large degree of backmixing in the liquid phase. The actual axial phenol concentration profile drops off sharply from the bottom of the reactor to 40 *cm* of the bed height. Since near the bottom of the reactor where there is strong mixing caused by the air and liquid flow, this potion of the reactor acts as a completed-mixed type reactor and much of the bioparticle is located in this area. Thus, a significant phenol removal takes place on approximately 50 % area of the total bed height from the bottom of the reactor. The axial phenol concentration profiles did not change significantly from 50 *cm* of the bed height to the top of the bed height and were essentially constant. Because near the top of the reactor where there is little mixing, this portion of the reactor can be considered as plug-flow type reactor. The phenol removal in this portion of the reactor is limited.

The integrated model proposed performs well in predicting the phenol concentration profiles with dynamic biofilm growth model and an understanding the dynamic

behavior of TPFBBR process. It is demonstrated that the integrated model with dynamic biofilm growth model provides a new comprehensive model for modelling the dynamics of a TPFBBR.



continued on next page…

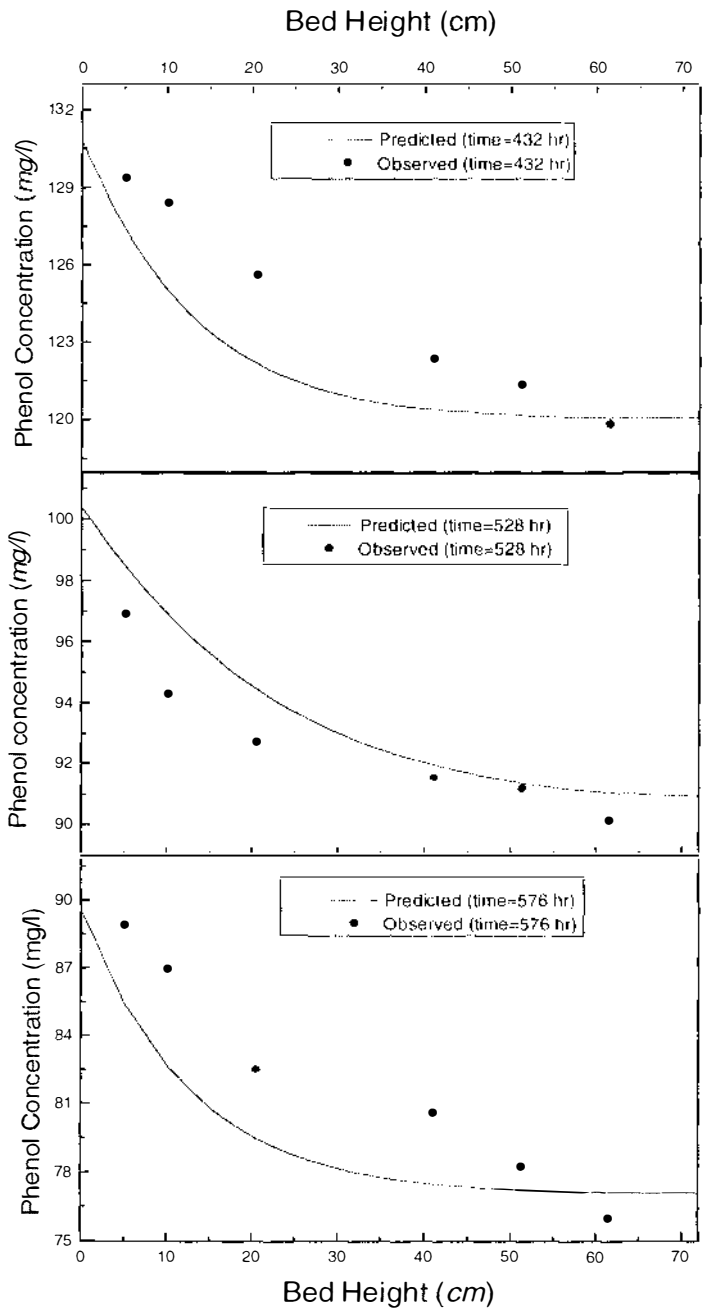
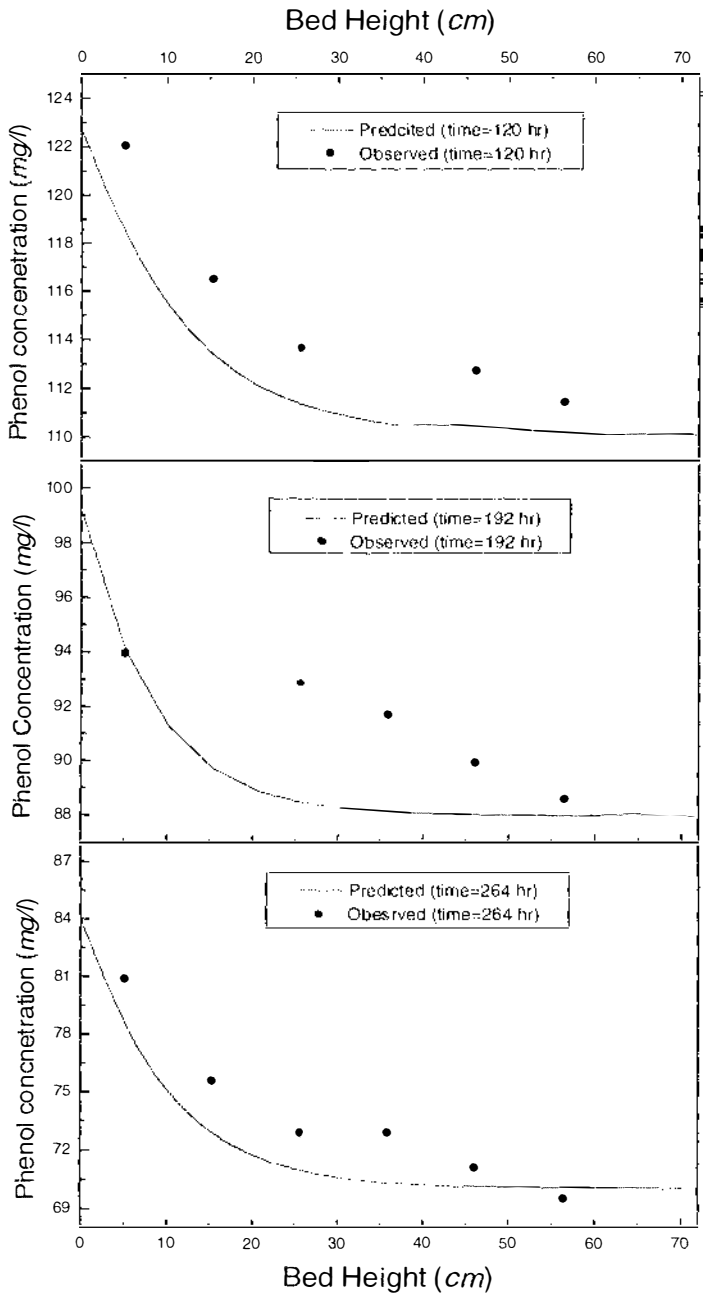


Figure 4.7. The substrate concentration profiles in the liquid phase as a function of bed height at the different times (U_L = 0.078 $cm/\sec$, U_G = 0.842 $cm/\sec$, $d_{sp} = 570\,\mu m$).



continued on next page…

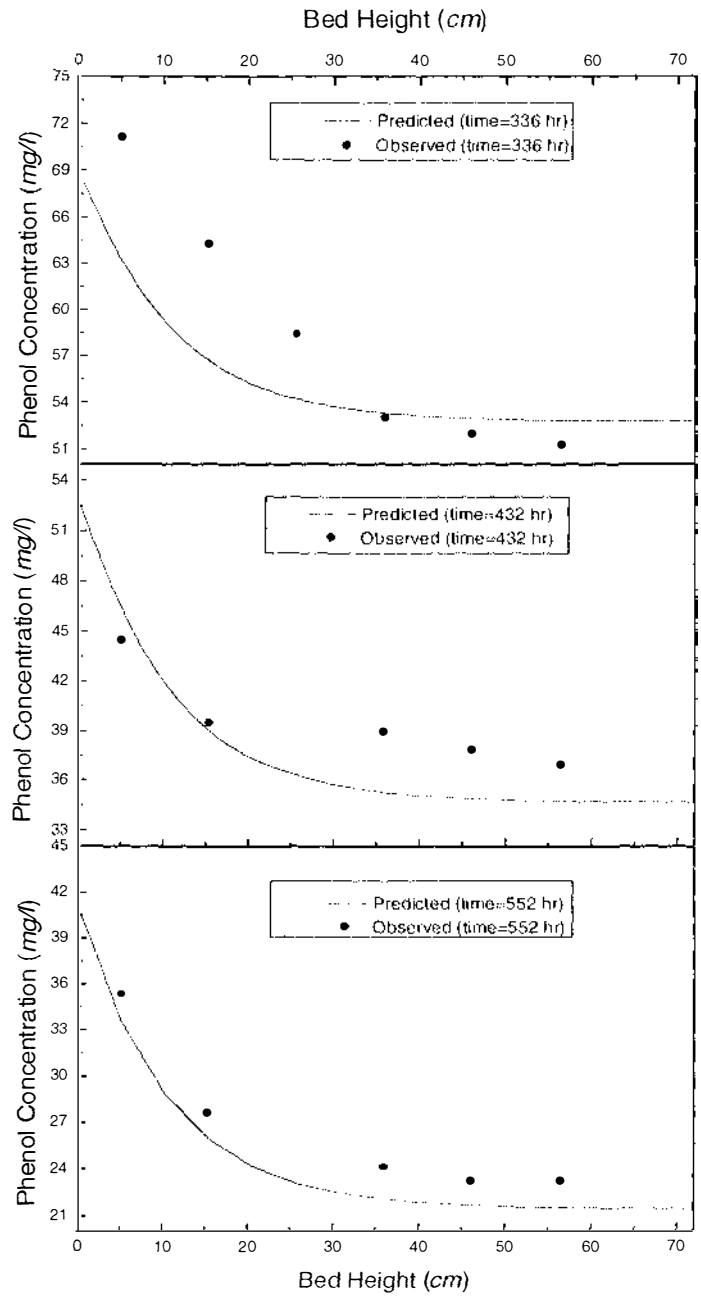


Figure 4.8. The substrate concentration profiles in the liquid phase as a function of bed height at the different times. (U_L = 0.03 $cm/\sec$, U_G = 1.2 $cm/\sec$, d_{sp} = 710 μm).

In order to illustrate the dynamic behavior of TPFBBR predicted by the simulation, 3-dimensional plots of the phenol concentration profile as a function of bed height and time are presented in Fig. 4.9-4.10.

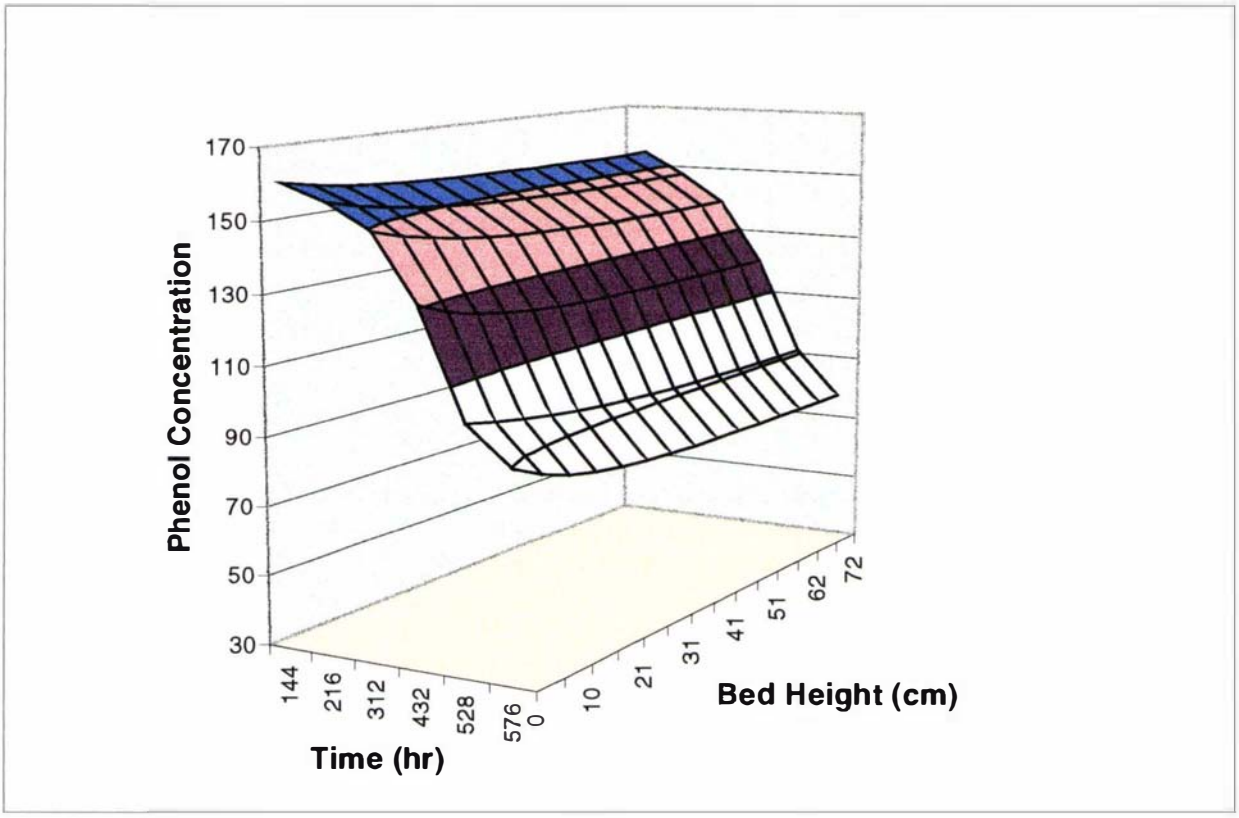


Figure 4.9. 3-D plots of phenol concentration profile as function of bed height and time ($U_L = 0.078\ cm/\sec$, $U_G = 0.842\ cm/\sec$, $d_{sp} = 570\ \mu m$).

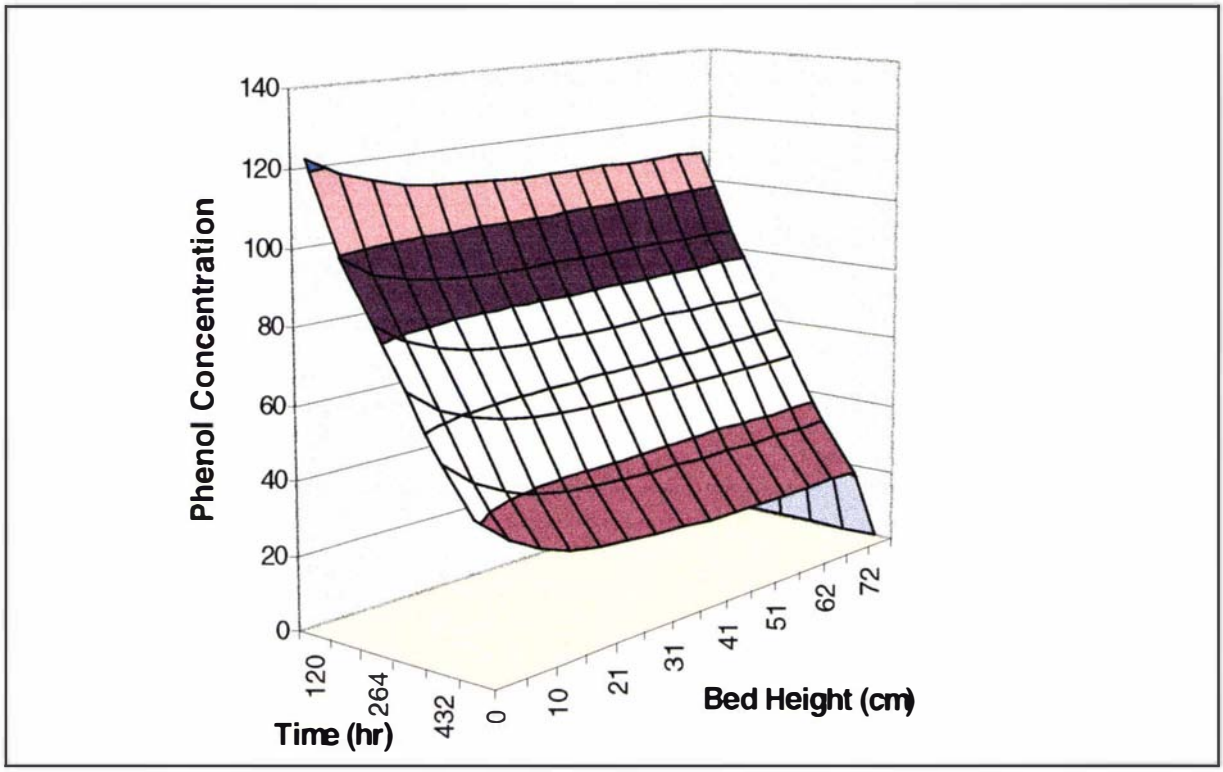


Figure 4.10. 3-D plots of phenol concentration profile as function of bed height and time ($U_L = 0.03\ cm/\sec$, $U_G = 1.2\ cm/\sec$, $d_{sp} = 710\ \mu m$).

4.7 CONCLUSIONS

The dynamic biofilm growth model, which reflects the variation of biofilm thickness and its density in time, is derived from a biomass balance equation. The biofilm detachment model is also proposed. In order to arrive at a solution of a dynamic biofilm growth model, the method of characteristic is used. In our biofilm growth model with detachment, the biofilm thickness and its density can be predicted at a given time. It can be seen that the results predicted using this dynamic biofilm growth model, which accounts for varying biofilm thickness and its density in time, are in good agreement with experimental data.

The proposed biofilm growth model with detachment was coupled to a reactor model and a reaction-diffusion model to form an integrated model of a TPFBBR. An integrated model is developed to describe the dynamic behaviour of a TPFBBR during biofilm formation and an integrated design method for a TPFBBR is projected based on the model presented. Simulation results of integrated model incorporating the dynamic biofilm growth model indicate that biofilm density and biofilm thickness are the main design parameters in a TPFFBR. The proposed integrated model provides a valuable tool to predict performance in a TPFBBR and to develop the optimal control strategy.

Chapter 5

Sequential Neural Network Model for a TPFBBR

5.1 INTRODUCTION

Even though the integrated model incorporating dynamic biofilm growth developed in Chapter 4 is biochemically significant and can be satisfactory for design purposes, there are weak points in applying this to real-world process modelling and control. These weaknesses include: (1) dynamic biofilm growth on the support particles includes a large number of complex and highly interacting biochemical, transport and hydrodynamic phenomena, and the knowledge to give mechanistic description for biofilm growth is still limited and poorly known, (2) this model requires the specification of a large number of parameters, many of which are difficult to measure, (3) a large number of parameters included in this model need to respecify parameter values for different operational conditions, and (4) this model requires time-consuming computation procedures.

As an alternative to mechanistic models, there has been a major research interest in artificial neural network (ANN), a powerful tool for nonlinear modelling and process control. ANN offers the distinctive ability to learn complex relationships without requiring the mechanistic knowledge about processes. Therefore, it has a great potential in areas such as biological processes where complex and poorly known mechanisms have to be treated. ANN has been successfully used as a process variable estimator for unknown (or unmeasured) variables and a process prediction tool in bioreactor modelling applications (Thibault et al., 1990; Linko & Zhu, 1991; Breusegem, 1991; Simutis et al., 1993; Morris et al., 1994).

The purpose of this chapter is to describe the application of neural network process modelling approach for modelling the dynamic change of the biofilm thickness and biofilm density and for predicting the dynamic performance of a TPFBBR. First, the basic concepts of neural networks are introduced for nonlinear process modelling based on input-output data. Second, the most common *multilayer feedforward neural network* (MFNN) is described with the standard *backpropagation* learning algorithms. Third, the different leanings such as *cascaded-correlation* (C-C) learning and *extended Kalman filtering* (EKF) learning for MFNN and *generalized regression neural network* (GRNN) are briefly discussed. Fourth, the sequential neural network model, which is composed of two parts, namely, the neural process estimator and the neural process predictor, is developed to describe the task of process estimation and prediction for a TPFBBR. Finally, experimental data taken from a laboratory-scale TPFBBR is used to demonstrate the power of the proposed sequential neural network model. This shows the feasibility of using sequential neural network model as intelligent estimators and dynamic predictors. It is demonstrated that the sequential neural network model gives considerably good results in process estimation and prediction for a TPFBBR.

5.2 BASICS OF ARTIFICIAL NEURAL NETWORKS

5.2.1 What is an Artificial Neural Network (ANN)?

An *artificial neural networ*k is a parallel, distributed information processing structure consisting of *processing elements* (which can process a local memory and can carry out localised information processing operations) interconnected via unidirectional signal channel called *connections*. Each processing element had a single output connection that branches ("fan out") into as many collateral connections as desired; each carries the same signal - the *processing element output signal*. The processing element output signal can be of any mathematical type desired. The information processing that goes on within each processing element can be defined arbitrarily with the restriction that it must be completely local; that is must depend only on the current values of the input signals arriving at the processing element via imprinting connections and on values stored in the processing element's local memory (Heicht-Nielsen, 1988).

The neural networks are used for two main tasks in engineering applications: 1) *function approximation* and 2) *pattern classification*. In function approximation, the neural network is trained to approximate a mapping of its inputs and outputs. Many neural network models have been proved as universal approximators, *i.e.*, the network can approximate any continuous function arbitrary well. The pattern classification application can be regarded as a specific case of the function approximation. The mapping is done from the input space to a finite number of output classes.

Currently, there have been a wide variety of neural networks that are being studied. Based on characteristics, such as the class of inputs, the method of training, and weight updating procedures, these networks can be classified as shown in Table 5.1. Kohonen (1990) classified neural network architectures into three categories depending on model of the nervous system (Fig. 5.1). In Fig. 5.1, *feedforward*

networks transforms sets of input signals into sets of output signals. The desired input-output transformation is usually determined by external, *supervised* adjustment of the system parameters. In *feedback networks* (recurrent), the input information defines the initial activity state of a feedback system. After state transitions, the asymptotic final state is identified as the outcome of the computation. In competitive, *unsupervised* or self-organizing category cells (neurons), the neighbouring cells (neurons) in the network complete in their activities and develop iteratively specific detectors for different input signal patterns.

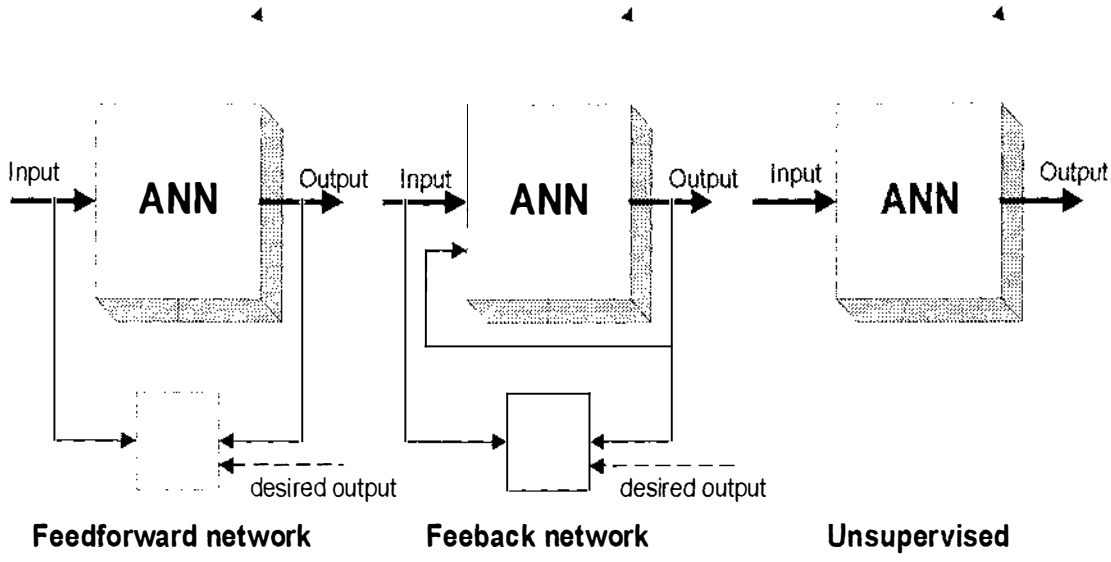


Figure 5.1. Neural network models (dotted line illustrates the training scheme).

Of all the available neural networks, the multilayer feedforward neural network is widely used for the application of chemical and bioprocess engineering. This thesis focuses on the application of the type of the multilayer feedforward neural network (MFNN) to a TPFBBR. Table 5.1 summarises the categorisation of some neural networks used widely in many engineering applications.

Table 5.1 Categorisation of some neural networks.

NEURAL NETWORK	LEARNING	STRUCTURE	USAGE
Multilayer perceptron	supervised	feedforward	classification function approximation
Radial basis function network	supervised or unsupervised	feedforward	classification function approximation
Self-organizing map	unsupervised	feedforward	classification
Learning Vector Quantization (LVQ)	supervised	feedforward	classification

5.2.2 An Artificial Neural Network as a Process Modelling Tool

In order to properly optimize and control a process, it is necessary to develop a good mathematical model. Since most of the advanced control approaches are based on a mathematical model of the processes under consideration and the optimal operating strategies can be by simulating using the process model under different conditions.

For process modelling, the best possible model is a mechanistic model which consists of a set of differential equations which define the relationship between input variables and output variables. A mechanistic model, however, is either too difficult to formulate or too difficult to solve the resulting set of equations in many cases. To be more specific, models in mathematical modelling of biofilm growth are formulated involving either Monod's kinetics or one of its modified expressions such as Haldane expression reflecting product inhibition or substrate inhibition or both. If,

however, kinetic expressions for on-line interpretation and simulation of biological processes are desired, these mechanistic models require too detail information about the process to be applicable. Moreover, it is likely that model parameters should be specified and updated to obtain an agreement between actual and predicted value. Other drawbacks are that the development of good kinetic or process model usually requires very time-and money-consuming tasks because the necessary knowledge to give a mechanistic description for a specific (bio)chemical process is usually limited and are still poorly understood (Saxen & Saxen, 1996).

As an alternative to a mechanistic model, there has been major research interest in artificial neural networks (ANNs), a powerful tool for nonlinear modelling and process control. The main advantages of using ANNs in process modelling are: (1) it has the ability to learn complex nonlinear relationship with limited prior knowledge of the process structure and (2) it can perform inferences for an unknown combination of input variables (Hong et al., 1998). So ANNs are prime candidates for use in dynamic process modelling for the representation of nonlinear processes. Due to the advantages of a neural network, a number of researchers have successfully applied a neural network based modelling approach to wastewater treatment processes. Capodaglio et al. (1991) identified an ANN model for the simulation and forecasting of the sludge bulking based on sludge volume index (SVI) in full-scale activated sludge process. Collins & Ellis (1992) applied a neural network model to the prediction of a required chemical dosage in a wastewater treatment plant. Tyagi & Du (1992) used a neural network to predict the effect of heavy metals in the performance of the activated sludge process. Zhao et al. (1997) demonstrated a hybrid model, which consists of a simplified process model and an ANN, for developing a dynamic model of a sequencing batch reactor. In their hybrid model, the outputs of the trained ANN compensated for the output errors of the simplified process model. Karim and Rivera (1992) reviewed the application of ANN in bioprocess state estimation.

The multilayer feedforward neural network (MFNN) seems to be a very attractive choice when neural networks are used for process modelling and control purposes.

This is because it has been theoretically proven that the MFNN can approximate any continuous function arbitrarily well provided that enough neurons are used (Cybenko, 1989). However, in order to obtain a valid model of the process, neural networks, in general, require a large number of training and test data, even for a moderate number of model parameters (the weight and biases). A more detailed discussion of this issue can be found in Baum & Haussler (1989). Another disadvantage is that neural networks are non-parametric models. In a non-parametric model, the model parameters (the weights and biases in the MFNN) usually have no interpretation in relation to the process to be modelled.

5.2.3 Multilayer Feedforward Neural Networks (MFNN)

The multilayer (3-layer) feedforward neural network consists of one input layer, one or more hidden layers, and one output layer. The general structure of multilayer (3-layer) feedforward neural network is given in Fig. 5.2. The first layer, the input layer, is strictly a preprocessing layer that simply distributes the input to the next layer. It does not perform, as subsequent layers do, a nonlinear transformation of its input data. An output layer delivers the output from the neural network. In between these two layers, there could be several layers called "hidden layers". Input and out data vectors are scaled from 0 to 1 and scaled data are fed into the neural network at the input layer. Each of these layers consists of *neurons* (or *processing elements*), which are represented by the circle in Fig. 5.2. All the neurons in one layer are connected to all neurons in the following layer by a set of unidirectional weights (represented by the lines in Fig. 5.2). In addition to the regular neurons, there are bias neurons which provide a constant input of unity. Bias neurons are connected to all the neurons in the hidden and output layers through a set of bias weight. Since there is only one forward path for the flow of information, these networks are called *feedforward neural network*.

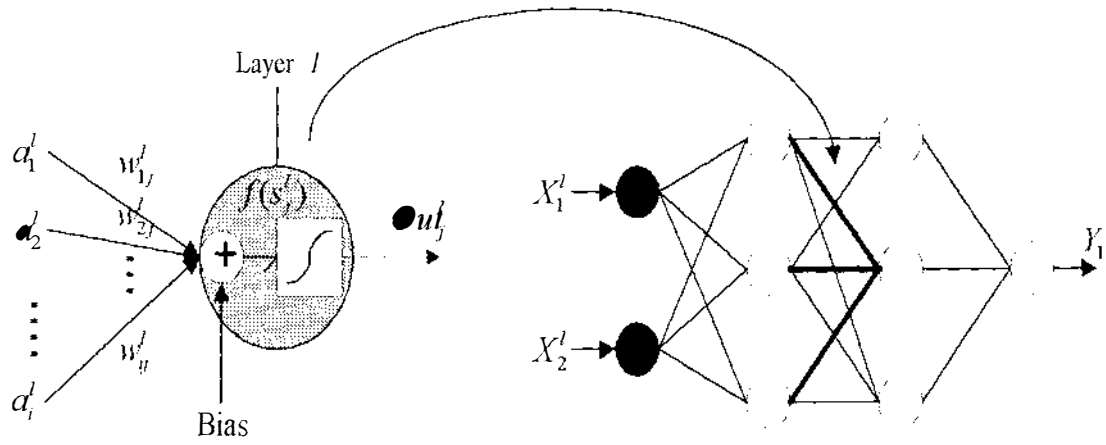


Figure 5.2. Architecture of the multilayer feedforward neural network (MFNN).

Typical neuron performs two functions: a weighted linear combination of its input component (activity) and a nonlinear transformation of this activity value. A single neuron extracted from the l^{th} layer is also depicted in Fig. 5.2. The input to this neuron consists of the N-dimensional vector X_i and a unit bias. Each input is multiplied by a weight which denotes connections between neuron i in the previous layer and neuron j in the layer l. The products are summed up to give the *activation potential* (or *activation state*) s_j^l:

$$s_j^l = \sum_i w_{ij}^l x_i^{l-1} + w_{bj}^l \tag{5.1}$$

The output of the j^{th} neuron in layer l, Out_j^l is then calculated as the nonlinear activation functions such as sigmoid function.

$$Out_j^l = f(s_j^l) = \frac{1}{1+e^{-s_j^l}} \tag{5.2}$$

There are several types on non-linear activation functions. Differentiable, non-linear activation functions can be used in networks trained with backpropagation. The most common are the logistic function and the hyperbolic tangent function (Fig. 5.3).

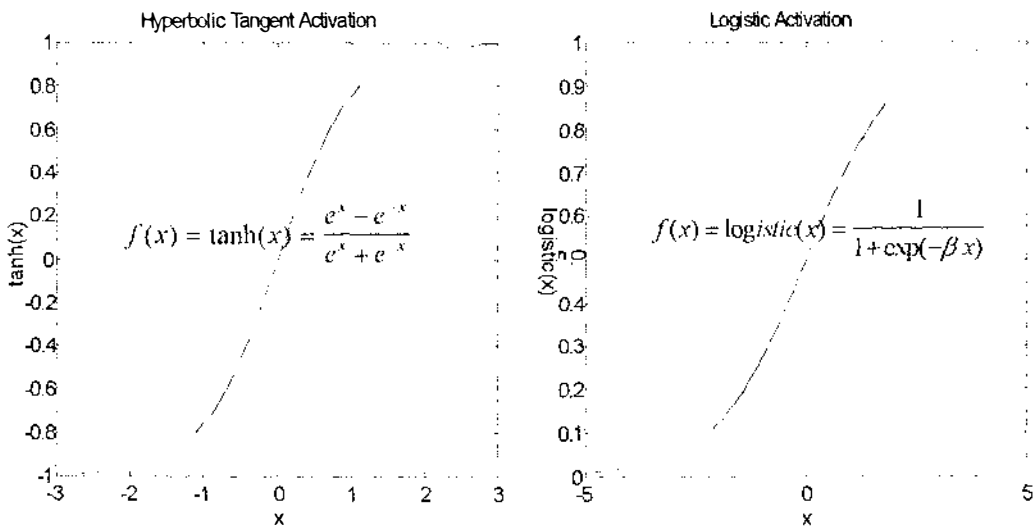


Figure 5.3. Common activation functions used in neural network.

Generally, the number of neurons in the input and output layers is determined by the number of input and output variables involved in the problem. The number of neurons in the hidden layer(s) is related to the converging performance of the output error function during the process. The optimal number of neurons was determined by trial and error. Too few hidden neurons limits the ability of the neural network to model the process, and too many hidden elements may allow too much freedom of the weights to adjust and results in the noise present as the data base used as the training. The determination of the architecture of neural networks is a time-consuming task when applying ANN to a new problem. In order to overcome these problems, genetic algorithms (GAs) have been used to develop the architecture of an ANN (Goldberg, 1989; Miller et al., 1989; Whitely et al., 199●; Maniezzo, 1994).

5.2.4 Backpropagation (BP) Training Algorithm

The knowledge required to map input into an appropriate output is embodied by the weights. Initially, the weights appropriate to a given problem domain are unknown. Until a set of applicable weights is found the network has no ability to deal with the problem to be solved. The process of finding a useful set of weights is called *training*. Training begins with a training set consisting of specimen input with associated outputs that represents a correct prediction. The existence of a specific desired output for each of the training set vectors at least to within some defined error limit. Training the network involves moving from the training set has to teach it. If the training set is good and the training algorithm is effective, the network should then be able to correctly predict inputs not belonging to the training set. This phenomenon is termed *generalisation*. Thus we see that the application of neural network to a specific problem involves two distinct phases. During the training phase, the network weights are adapted to reflect the problem domain as shown in Fig. 5.4(a) at the left. In the second phase (prediction phase), the weights have been frozen and the network when presented with the test data calculates a predicted value. This is illustrated in Fig. 5.4(b).

The neural network is provided with a training set of input vectors each with a desired output vector, $\{(x_1, d_1), (x_2, d_2), \ldots, (x_n, d_n)\}$. For given X, the difference between the desired output and the actual output of the network is the error

$$e = d(n) - y(n) \tag{5.3}$$

The total squared error over the training set is thus given by

$$J = \sum_{n=1}^{N} e_n^T e_n \tag{5.4}$$

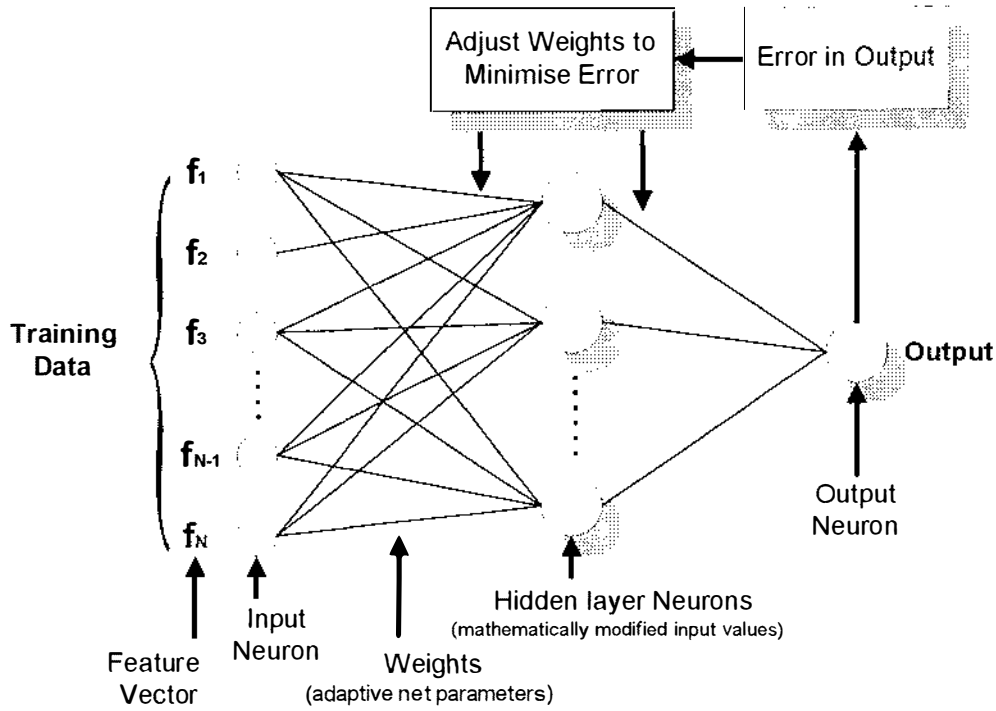


Figure 5.4(a). The training phase of MFNN.

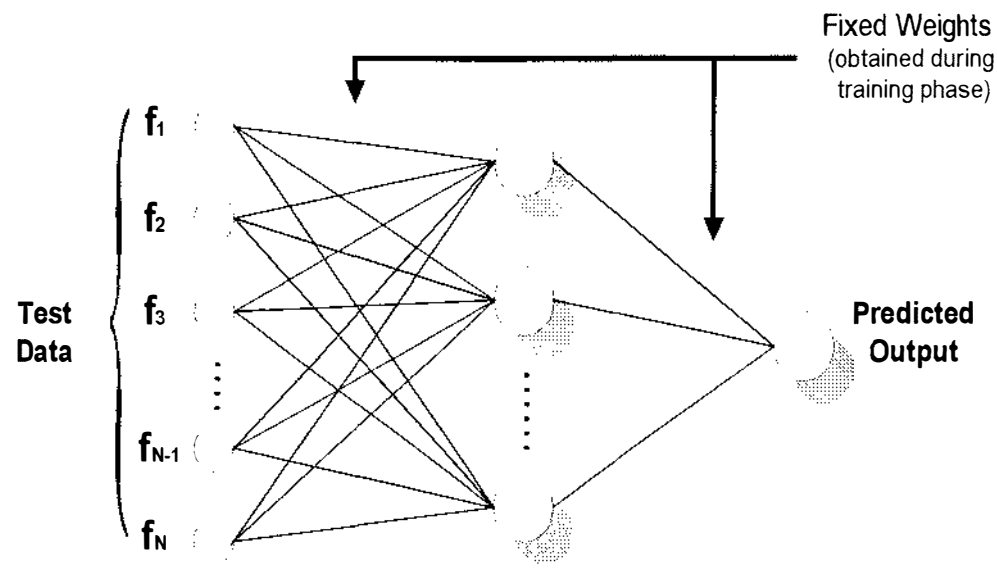


Figure 5.4(b). The prediction phase of MFNN.

A backpropagation algorithm, which is a typical supervised learning algorithm, is one of the simpler members of a family of training algorithms, collectively termed *gradient decent*. This algorithms was first described by Werbos (1974), and introduced by Rumelhart et al. (1986) as a useful and versatile training algorithm for multilayer feedforward neural network. The objective of training is to determine the set of weights W which minimises the cost J subject to constraint of the network topology.

After each presentation of a sample from the training set, the weights are adapted according to

$$\Delta W = -\mu \hat{\nabla} \simeq -\mu \frac{\partial e^T e}{\partial W} \tag{5.5}$$

where $\frac{\partial e^T e}{\partial W}$ is the instantaneous error gradient corresponding to the current input pattern, and μ controls learning rate which governs the distance traveled in the direction of the negative gradient when the step in weight space taken. Eq. (5.5) states that the change in each weight w_{ij}^l will along the negative gradient leading to a steep descent along the local error surface.

The task now is to convert Eq. (5.5) into a difference equation suitable for use in a computer implementation. To accomplish this, the partial derivative $\frac{\partial e^T e}{\partial W}$ is evaluated corresponding to each weight in the network. The chain rule is applied to calculate the gradient with respect to a weight w_{ij}^l:

$$\frac{\partial e^T e}{\partial w_{ij}^l} = \frac{\partial e^T e}{\partial s_j^l} \frac{\partial s_j^l}{\partial w_{ij}^l} = \delta_j^l Out_i^{l-1} \tag{5.6}$$

where the error signal δ_j^l is defined as:

$$\delta_j^l \triangleq \frac{\partial e^T e}{\partial s_j^l} \tag{5.7}$$

Recall s_j^l, which is defined in Eq. (5.1), is the summing junction which the weight interest feeds, and Out_i^{l-1}, which is defined in Eq. (5.2), represents the change in neural activation state s_j^l due to change in w_{ij}^l. This leads to the weight update equation:

$$\Delta w_{ij}^l = -\mu \delta_j^l Out_i^{l-1} \tag{5.8}$$

The backpropagation algorithm provides two rules for calculating the error signal δ_j^l of a neuron, depending on whether the neurons is in the output layer or in a hidden layer.

$$\begin{aligned} \delta_j^L &\triangleq \frac{\partial e^T e}{\partial s_j^L} \\ &= \frac{\partial e^T e}{\partial Out_j^L} \frac{\partial Out_j^L}{\partial s_j^L} \\ &= \frac{\partial e^T e}{\partial Y_i} f'(s_j^L) \end{aligned} \tag{5.9}$$

To evaluate $\partial e^T e / \partial Out_j^L$ in Eq. (5.9), the two cases must be considered individually:

1. The destination is an output neuron.
2. The destination is a hidden neuron.

For a destination neuron in the output layer we have direct access to the error $e^T e$ as a function of Out_j^L. Therefore we write:

$$\begin{aligned}\frac{\partial e^T e}{\partial Y_i} &= \frac{\partial}{\partial Y_i}\left(\sum (d_j - Y_i)\right)^2 \\ &= 2e_j \frac{\partial (d_j - Y_i)}{\partial Y_i} \\ &= -2e_j\end{aligned} \tag{5.10}$$

Substituting into Eq. (5.9) yields

$$\delta_j^l = -2e_j f'(s_j^l) \tag{5.11}$$

Calculating δ_j^l for destination neurons in hidden layers is made a bit more complicated by the fact that there is no explicit desired response for each neuron. So we cannot differentiate the error function directly. We note, however, that $e^T e$ is influenced through s_j^l indirectly through all node values s_j^{l+1} in the next layer. We must once again apply the chain rule to obtain:

$$\begin{aligned}\delta_j^l &\triangleq \frac{\partial e^T e}{\partial s_j^l} \\ &= \sum_i \frac{\partial e^T e}{\partial s_j^{l+1}} \frac{\partial s_j^{l+1}}{\partial s_j^l}\end{aligned} \tag{5.12}$$

The summation is taken over all branches that neuron *i* in layer *l* feeds. Note that

$$\begin{aligned}\frac{\partial s_j^{l+1}}{\partial s_j^l} &= \frac{\partial s_j^{l+1}}{\partial Out_i^l} \frac{\partial Out_i^l}{\partial s_j^l} \\ &= w_{ij}^{l+1} f'(s_j^l)\end{aligned} \tag{5.13}$$

Next make the critical observation that $\partial e^T e / \partial s_j^{l+1}$. Making these substitutions leads to the relation

$$\delta_j^l \approx f(s_j^l)\sum_j \delta_j^{l+1} w_{ij}^{l+1} \tag{5.14}$$

This is a recursive formula in which previously calculated δ terms are backpropagated through connecting synapses. The equations for backpropagation are summarised as follows:

$$\Delta w_{ij}^l = -\mu\, \delta_j^l\, Out_i^{l-1} \tag{5.15}$$

$$\delta_j^l = \begin{cases} -2\, e_i f'(s_i^L) & l = L \\ f'(s_i^l)\sum_j \delta_j^{l+1} w_{ij}^{l+1} & 1 \le l \le L-1 \end{cases} \tag{5.16}$$

For the bias weight w_b^l, note that $a_i^{l-1} = 1$ in Eq. (5.15)

The overall process of backpropagation learning including both the forward and backward pass is presented in Fig. 5.5. To apply the backpropagation algorithm the network weights must first be initialised to small random values. It is important to make the initial weights "small". Choosing initial weights too large will make the network untrainable. After initialisation, training set vectors are then applied to the network. Running the network forward will yield a set of actual values. The backpropagation can then be utilised to establish a new set of weights. The total error should decrease over the course of much such iteration. If it does not, an adjustment to the training parameter μ may be required. One full presentation of all the vectors in the training set is termed an *epoch*. When the weights approach values such that the total network error, over a full epoch, falls below a pre-established threshold, the network is said to have converged.

The *backpropagation* (BP) algorithm is a general method for iteratively solving a multilayer perceptrons' weights and biases. It uses a steepest descent technique which is very stable when a small learning rate is used, but has slow convergence properties. Several methods for speeding up BP have been used including

momentum (Rumelhart et al., 1986) and variable step-size control methods (Franzini, 1987; Fahlman, 1988). Clearly, many practical and theoretical issues must be addressed to fully understand the training process and neural networks. A detailed coverage of all topics is beyond the scope of this chapter. Excellent reviews on topology of neural networks and learning algorithms are presented by Haykin (1994).

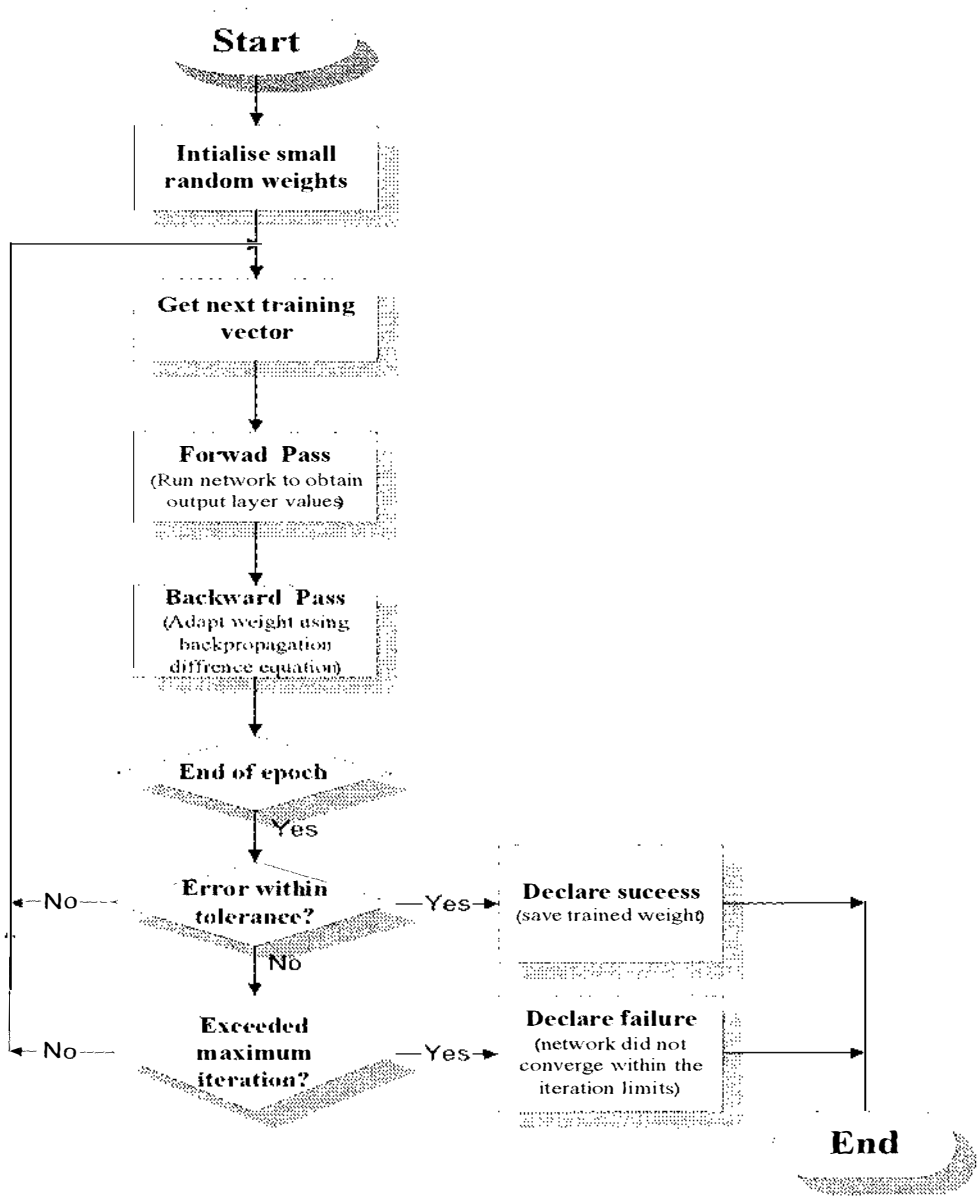


Figure 5.5. Backpropagation flow chart.

5.2.5 Cascade-Correlation (C-C) Algorithms

Two problems are associated with the standard BP algorithm. Firstly, appropriate learning parameters need to be chosen. Their tuning is not trivial. Secondly, the convergence speed of the training process is slow, which is the weakest point of BP algorithm. In order to overcome these problems, the cascade-correlation algorithm (C-C) has been introduced by Fahlman & Lebiere (1991) in order to find suited MFNN structures and to decrease the training effort for complex learning tasks by splitting them into a number of independent subtasks.

C-C algorithm is characterised as a constructive learning rule. C-C algorithm begins with minimal network, consisting only of an input and an output layer, then automatically trains and adds new hidden units one by one, creating a multilayer structure. Once a new hidden unit has been added to the network, its input-side weights are frozen. This unit then becomes a permanent feature-detector in the network, available for producing outputs or for creating other, more complex feature detectors (Fahlman & Lebiere, 1991).

The C-C algorithm is demonstrated in the following way:

1. C-C algorithm starts with a minimal network consisting only of input and an output layer. Both layers are fully connected.
2. Train all the connections ending at an output unit with a usual learning algorithm until the error of the network no longer decreases.
3. Generate the so-called candidate units. Every candidate unit is connected with all input units and with all existing hidden units.
4. Try to maximise the correlation between the activation of the candidate units and residual error of the network by training all the links leading to a candidate unit. The training is stopped when the correlation scores no longer improve.
5. Choose the candidate unit with the maximum correlation, freeze its incoming weights and add it to the network. Loop back to step 2.

6. This algorithm is repeated until the overall error of the neural network falls below a given value.

The form of the cascaded-correlation network is shown in Fig. 5.6. The C-C algorithm has several advantage over existing algorithms: it learns very quickly, the network determines its own size and topology, it retains the structures it has built even if the training set changes, and it requires no backpropagation of error signals through the connections of the network (Fahlman & Lebiere, 1991).

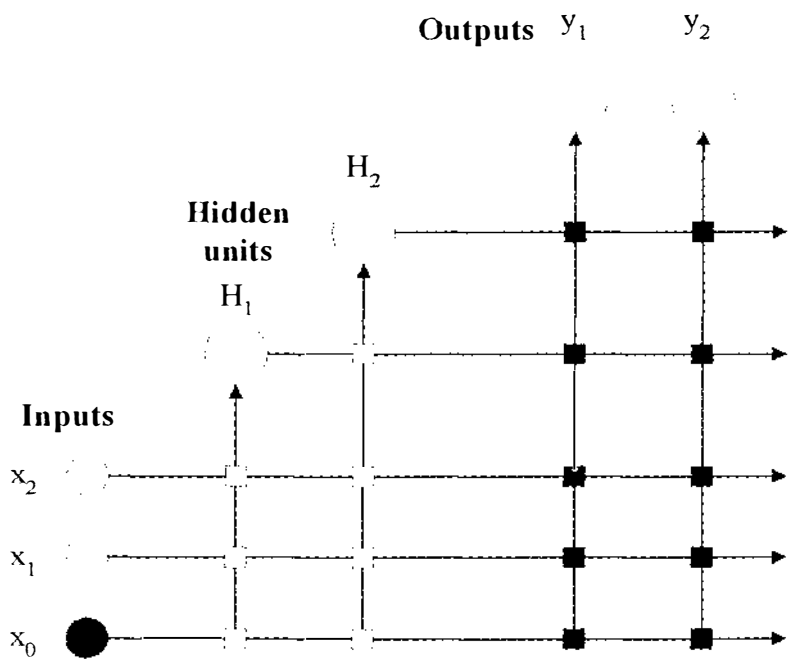


Figure 5.6. The architecture of the neural network trained with C-C after 2 hidden units have been added. White squares represent weights which are trained and then frozen, while the black squares show weights which are retrained after the addition of each hidden unit. Hidden unit H_1 is added first, and then hidden unit H_2, and so on (Fahlman & Lebiere, 1991).

5.2.6 Extended Kalman Filtering (EKF) Learning Algorithm

In order to overcome weak points of backpropagation learning algorithms, the extended Kalman filtering (EKF) algorithm has also been introduced as a learning method for training the weights in a MFNN (Singhal & Wu, 1989; Palmieri et al., 1991; Lange & Hirzinger, 1995) or a recurrent MFNN (Puskorius & Feldkamp, 1994).

Basically, a Kalman filter is a set of mathematical equations that provide an efficient recursive solution of the least-squares method. This is very powerful in several aspects: it supports estimation of past, present, and even future states, and it can do so even when the nature of the modelled system is unknown. The extended Kalman filtering (EKF) algorithm applies the standard (linear) Kalman filter to nonlinear systems with additive white noise by continually updating a linearisation around the previous state estimate. The EKF learning algorithm to training a MFNN considers the weights of the MFNN to be states, and the desired outputs of the neural network to be the observations within a discrete state space transition framework. To use an EKF learning algorithm, a state space representation of the neural network is formulated and the resulting state space model is augmented by the weight vector W. Therefore,

$$W(k+1) = \phi(W(k),k) + \varpi(\varsigma,k) : \text{System model} \tag{5.17}$$

$$Y(k) = \psi(W(k),k) + v(\varsigma,k) : \text{Measurement model} \tag{5.18}$$

where W is a vector consisting of all the weights and biases in the neural network, Y the output of the neural network, and the iteration k corresponds to the presentation of the kth pattern. $\varpi(\varsigma,k)$ and $v(\varsigma,k)$ represent the process and measurement noise , respectively. In Eq. (5.17), the nonlinear function, $\phi(\bullet)$, relates the state at step k to the state at step $k+1$. The nonlinear function, $\psi(\bullet)$, in Eq. (5.18) relates the state, $W(k)$, to the measurement, $Y(k)$.

The equation for the EKF learning algorithm fall into two steps: (1) time update equations which are responsible for projecting forward the current weight (state) and error covariance estimates to obtain a *priori* estimates for the next time step, and (2) measurement update equations which are responsible for the feedback - i.e. for incorporating a new measurement into a *priori* estimate to obtain a *posteriori* estimate.

Using the preceding models for the system and the measurement, methods for updating the estimate of the weights (system state) can be derived as followings:

$$G(k) = P^{-}(k)H^{T}(k)[H(k)P^{-}(k)H^{T}(k) + \sigma^{2}(k)] \tag{5.19}$$

$$W^{+}(k) = W^{-}(k) + G(k)[Y(k) - \psi(W^{-}(k), k)] \tag{5.20}$$

$$P^{+}(k) = P^{-}(k) - G(k)H(k)P^{-}(k) \tag{5.21}$$

where

$$H(k) = \psi^{T}(\hat{W}^{-}(k), k) = \frac{\partial \psi(W(k), k)}{\partial W(k)} \tag{5.23}$$

where $\hat{W}(k)$ is the Kaman filter estimate of the weight vector W at step k, $P(k)$ is an error covariance matrix which is used to model the correlation or interaction between each pair of weights in the neural network, $G(k)$ is called the Kalman gain matrix which is computed at each step and is used to update the weight vector W and error covariance matrix P, and $H(k)$ is the gradient matrix resulting from linearising the neural network.

In Eq. (5.19)-(5.23), the symbol, +, represents the step just prior to performing the updates, and the symbol, -, represents the time just after the updates.

In Eq. (5.19), σ^{2} is the variance of the measurement noise and is given as followings:

$$\sigma^2 = E\{v \cdot v\} \tag{5.24}$$

The propagation equations of the system state from k to $k+1$ are given by

$$\hat{W}^-(k+1) = W^+(k) \tag{5.25}$$

$$P^-(k+1) = P^+(k) + Q(k) \tag{5.26}$$

$Q(k)$ is the covariance matrix of the stochastic changes of the parameters and is described by

$$Q(k) = E\{\varpi \cdot \varpi^T\} = I\, q(k) \tag{5.27}$$

where I is identity matrix and q is slowly changing in relation to the propagation time.

In order to implement the EKF learning algorithm, the initial conditions at step $k=0$ must be specified, namely, $W(0)$ and $P(0)$. The initialisation of weight vector is set randomly, except for the output layer where the parameters are set to zero, and the matrix $P(0)$ is initialised as a diagonal matrix with the large diagonal components such as 500. The learning parameters q and σ^2 of the EKF have to be adapted in each epoch, and are estimated by the method which is proposed by Lange & Hirzinger (1995). σ^2 is estimated by the mean error as followings:

$$\sigma^2 \approx 0.1 \frac{1}{N} \sum_{n=1}^{N} \left(Y(n) - \psi(\hat{W}(k), n) \right)^2 \tag{5.28}$$

The change of the optimal values $W(k)$ due to changing linearisation of Eq. (5.23) is calculated by

$$q = \frac{0.01}{k}(\hat{W} - \hat{W}_{old})^T (\hat{W} - \hat{W}_{old}) \tag{5.29}$$

This change is assumed to occur from one epoch to another and thus weights the differences of the estimated value between epochs. So Eq. (5.23) is executed only at the end of every epoch, otherwise replaced by q=0.

5.2.7 Generalized Regression Neural Network (GRNN)

The Generalized Regression Neural Network (Specht, 1991) is a feedforward neural network best suited to function approximation tasks such as system modelling and prediction. The GRNN is composed of four layers. The first layer is the input layer and is fully connected to the pattern layer. The second layer is the pattern layer and has one neuron for each input pattern. This layer performs the same function as the first layer radial basis function (RBF) neurons: its output is a measure of the distance the input is from the stored patterns. The third layer is the summation layer and is composed of two types of neurons: S-summation neurons and a single D-summation neuron (division). The S-summation neuron computes the sum of the weighted outputs of the pattern layer while the D-summation neuron computes the sum of the unweighted outputs of the pattern neurons. There is one S-summation neuron for each output neuron and a single D-summation neuron. The last layer is the output layer and divides the output of each S-summation neuron by the output of the D-summation neuron. A general diagram of a GRNN is shown in Fig. 5.7.

The GRNN is based on nonlinear regression theory, a well-established statistical technique for function estimation. By definition, the regression of a dependant variable Y on an independent variable X estimates the most probable value for X, each with a corresponding value for Y (X and Y are, in general, vectors). Note that Y may be corrupted by additive noise. Despite this the regression method will produce the estimated value of Y which minimises the mean-squared error.

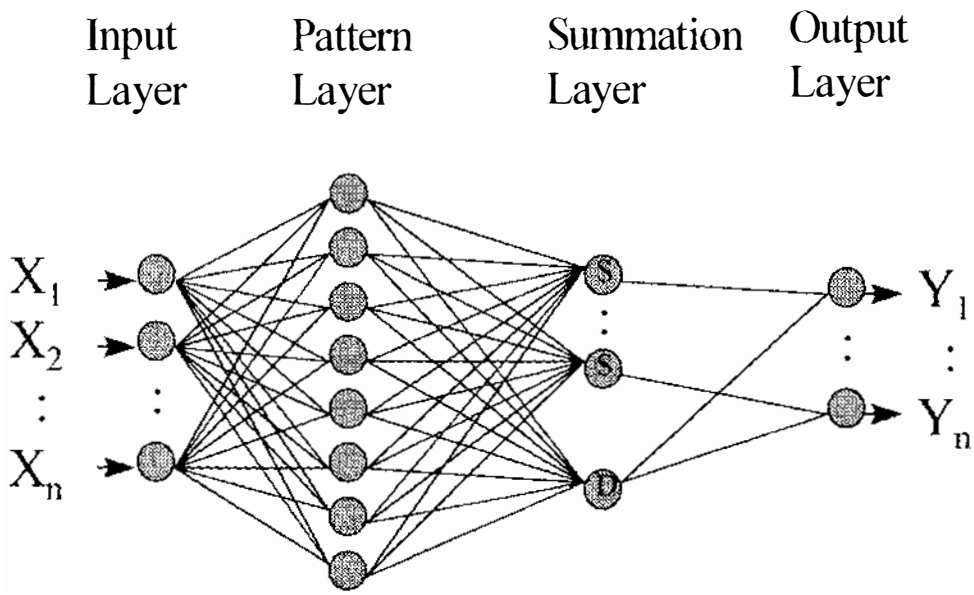


Figure 5.7. Generalized Regression Neural Network.

GRNN is based upon the following formula:

$$E[Y \mid X] = \frac{\int_{-\infty}^{\infty} Y \cdot f(X,Y)dY}{\int_{-\infty}^{\infty} f(X,Y)dY} \tag{5.30}$$

where Y is the output of the estimator, X is the estimator input vector, $E[Y \mid X]$ is the expected value of output, given the input vector X, and $f(X,Y)$ is the joint probability density function (PDF) of X and Y.

In essence, GRNN is a method for estimating $f(X,Y)$, given only a training set. Because the PDF is derived from the data with no preconceptions about its form, the system is perfectly general. There is no problem if the functions are composed of multiple disjoint non-Gaussian regions in any number of dimensions, as well as those

of simpler distributions. The probability estimator $\hat{f}(X,Y)$ is based upon the following equation:

$$\hat{f}(X,Y) = \frac{1}{(2\pi)^{(p+1)/2}\sigma^{(p+1)}} \frac{1}{n} \sum_{i=1}^{n} \exp\left[-\frac{(X-u^i)^T (X-u^i)}{2\sigma^2} \right] \exp\left[-\frac{(Y-Y^i)}{2\sigma^2} \right] \quad (5.31)$$

where u^i is the input training vector, Y^i is the desired output corresponding to u^i, p is the dimension of the input vector X, and σ is a constant controlling the size of the receptive region. Substituting the joint probability estimator $\hat{f}(X,Y)$ in Eq. (5.31) into the conditional mean Eq. (5.30), and interchanging the order of integration and summation yields the desired conditional mean, designated $\hat{Y}(X)$. The output of a GRNN is the conditional mean given by:

$$\hat{Y}(X) = \frac{\sum_{i=1}^{n} Y^i \exp\left(-\frac{D_i^2}{2\sigma^2} \right)}{\sum_{i=1}^{n} \exp\left(-\frac{D_i^2}{2\sigma^2} \right)} \quad (5.32)$$

where $D_i^2 = (X-u_i)^T (X-u_i)$ which represents the squared distance between the input vector X and the training vector u. When the smoothing parameter σ is made large, the estimated density is forced to be smooth and in the limit becomes a multivariate Gaussian with covariance $\sigma^2 I$. On the other hand, a smaller value of σ allows the estimated density to assume non-Gaussian shapes.

Note that Eq. (5.32) is identical to the radial basis function (RBF) with normalisation, except that the desired values are used for the weight the output network. Also, the resulting neural network topology is identical to the normalised RBF neural network. The major difference lies in the way that in GRNN, instead of training the weights, one simply assigns to the weight the desired value directly from the training set associated with input training vector u^i and its corresponding output

vector Y^i. Note that the calculation of the Gaussian is performed in the pattern layer, the multiplication of the weight vector and summations are performed in the summation layer, and the division is performed in the output layer.

The GRNN learning phase is similar to that of a probabilistic neural network (PNN). It does not learn iteratively as do most ANNs. Instead, it learns by storing each input pattern in the pattern layer and calculating the weights in the summation layer. The equations for the weight calculations are given below. The pattern layer weights are set to the input patterns.

$$W_p = u_i^T \tag{5.33}$$

The summation layer weights matrix is set using the training target outputs. Specifically, the matrix is the target output values appended with a vector of ones that connect the pattern layer to the D-summation neuron.

$$W_s = \left[Y^i \ \text{ones}\right] \tag{5.34}$$

5.3 SEQUENTIAL NEURAL NETWORK MODEL

To develop a neural network-based model describing the process dynamics in a TPFBBR, the sequential neural network as shown in Fig. 5.8 is developed. The sequential neural network presented here is composed of two parts: (1) *the neural process estimator* and (2) *the neural process predictor*. In Fig. 5.8, the first neural network serves as *the neural process estimator*, receives as inputs the measured variables, and provides the estimation of the one-step-ahead biofilm thickness and biofilm density. Because of the important of biofilm thickness and biofilm density as a process variable in a TPFBBR it is important that an accurate process estimator be developed. This neural process estimator is a typical example of a general technique in process control called inferential estimation, where an estimate of primary

variables (biofilm thickness and biofilm density), which is difficult or expensive to measure, can be inferred from the cheap and more readily available secondary variables such as suspended biomass concentration and dissolved oxygen concentration.

To be more specific, on-line monitoring and estimation in biological processes have attracted considerable interest during the past decade (Bastin & Dochain, 1990). However, on-line monitoring and estimation of a wide range of biomass concentration in a suspended growth system, or biofilm thickness and biofilm density in a biofilm reactor are considered a difficult task due to lack of reliable techniques. Thus the control policies of most bioreactors including a biofilm reactor are based on the use of off-line analysis for process supervision. In the case of a fluidised-bed biofilm reactor, this off-line method requires the removal of bioparticles as a sample from the reactor, and a number of samples to obtain the desired measurement accuracy. Most significantly, the limitation of the sampling frequency to reduce loss of solid particles from the reactor or to minimise potential for contamination in case of pure culture systems causes the process engineer to react slowly to any undesirable condition and process disturbance. Therefore, through the off-line measurement, it may be impossible to have enough information to develop the monitoring strategies and high performance control system necessary for the reactor efficiency improvement. Recent years there has been development of the so-called *software sensor*, which can be defined as an algorithms for the on-line estimation of the state variables and the parameters which are not measurable or determinable in real time based on the related measurable data (Bastin & Dochain, 1990). The main role of the neural process estimator, which acts as an intelligent software sensor, is to estimate the future dynamic behavior of process variables such as the biofilm thickness and biofilm density.

The outputs of the neural process estimator subsequently form of part of the inputs for the second neural network. To be more specific, the second neural network can be regarded as *the neural process predictor* capable of predicting the output concentration using the estimated biofilm thickness and biofilm density. The main

role of the neural process predictor is to forecast the future dynamic behavior of a process. This is necessary for optimal process supervision and control.

5.3.1 Criteria for Validation of Neural Network Model

In order to evaluate prediction accuracy of the neural network model, it is necessary to use various model validation techniques. The neural network model can be evaluated only by comparing it's output sequence [$\hat{y}(n)$, $t = 1, 2, \ldots, N$] to the actual data [$y(n)$, $t = 1, 2, \ldots, N$], for the same set of inputs.

For a neural network model with a set of estimated parameters ($\hat{\theta}$), the most widely used criterion to evaluate the prediction accuracy of neural network model is the *Root-Mean- Squared Error (RMSE)* defined as

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N}\left(y(n) - \hat{y}(\hat{\theta}, n)\right)^2}{N}} \tag{5.35}$$

The *Mean Squared Error (MSE)* is also used, and defined as

$$MSE = \frac{\sum_{n=1}^{N}\left(y(n) - \hat{y}(\hat{\theta}, n)\right)^2}{N} \tag{5.36}$$

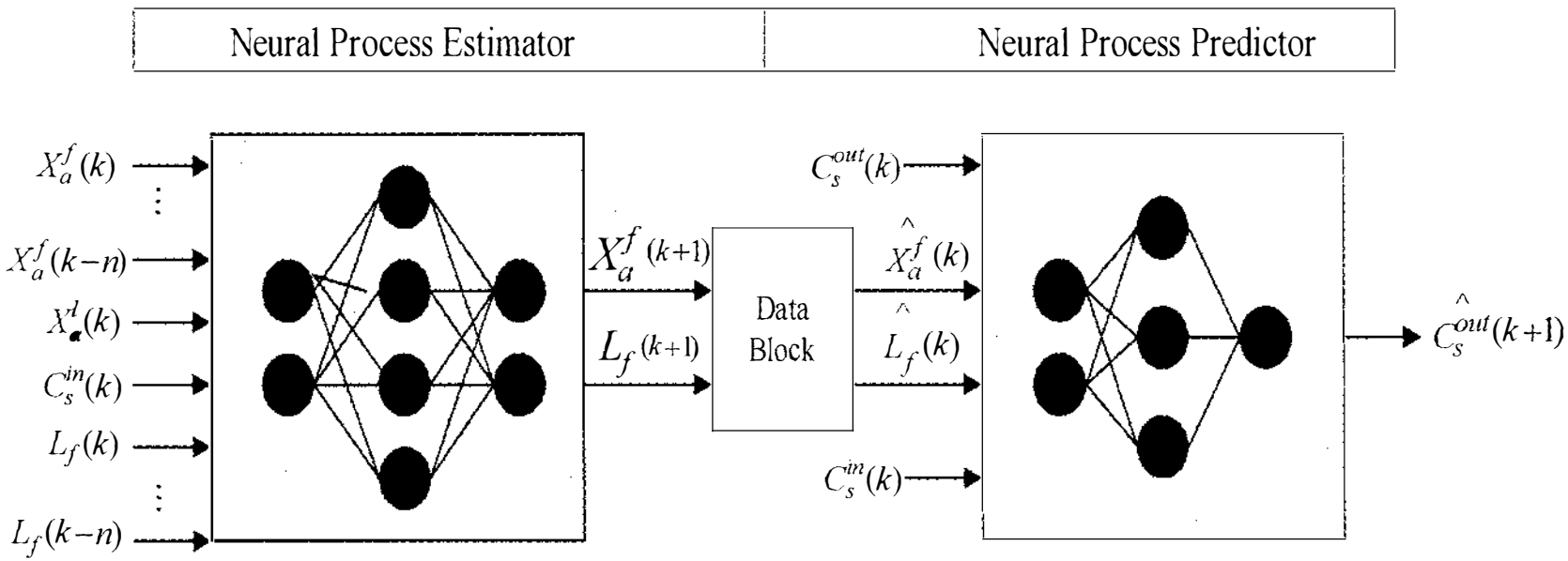


Figure 5.8. Architecture of the sequential neural network model

($X_a^f(k+1)$ = one-step-ahead biofilm density, $L_a(k+1)$ = one-step-ahead biofilm thickness, n = time delay).

An alternative criterion for model evaluation is the *coefficient of determination* R^2. R^2 coefficient is a ratio of the variation explained by the neural network model to the variation of the data. It is a relative measure defined as follows

$$R^2 = \frac{\sum_{n=1}^{N}(\hat{y}(n) - \bar{y}(n))^2}{\sum_{n=1}^{N}(y(n) - \bar{y}(n))^2}, \quad \text{where } \bar{y} = \frac{\sum_{n=1}^{N} y(n)}{N} \tag{5.37}$$

The value of R^2 varies between 0 and 1, where a value of 1 indicates a perfect model.

The *correlation coefficient* measures how well the network prediction trend with the targets in the training set. The range of the correlation coefficient is from -1 to 1. The closer the coefficient is to 1, the more accurate the predictions. The closer to 0 (or below), the less accurate and more random the predictions become. This plot often trends opposite the RMSE, the correlation increases as RMSE decreases. It can, however, be more informative because it uses an absolute scale to better quantifies the agreement (1 is perfect linear correlation, 0 is random). The extreme targets and predictions are the most heavily weighted in the calculation of the correlation coefficient (for binary output types all cases are at extremes).

Generally, the error history plot on the training set can be monitored to determine the rate of network learning and it can be used to determine when learning has reached its maximum level. Other interesting information can be derived from the training and test set error history. It is common to find long "plateaus" in the error level where no significant learning takes place. This behavior is particularly common when multiple hidden layers are being employed. This indicates that the network is trying to "figure out" certain input/output relationships. Plateaus are often followed by steep descents in the training error, yielding accelerated periods of learning. It is important that "plateau" conditions are not mistaken for a converged network.

The error history plot on the test set is used for overtraining analysis and helps determine how well the network generalises learned information. This error depicts how well the network predicts cases not used in the training process. Unfortunately, the difference between an error on the training set and an error on the test set can be due to various reasons (Ikonen, 1996):

1) the noise in the training data has been captured by the model
2) the statistical properties of training an data sets are not similar,
3) or the training data not contain enough information in order to determine all the parameters.

With many real industrial processes including the TPFBBR data presented here, it is not possible in practice, to collect two data sets of good quality (as it would become very expensive). That makes the separation of facts 1) and 2) difficult.

5.4 RESULTS AND DISCUSSIONS

5.4.1 Data Sets

In this chapter, sequential neural network modelling was tested using two data sets collected by experiments in chapter 3. Table 5.2 shows the details of two data sets used for the sequential neural network modelling. Historical data on biofilm thickness, biofilm density, suspended biomass concentration, dissolved oxygen (D.O.) concentration, and inlet phenol concentration were collected and split into two parts. A training set including 70% of the data was used to train a neural network. A test set including the remaining 30% was used to test the trained neural network in order to how well the network generalises or predicts on unseen data not used during training.

5.4.2 Data Transformation

First, the influence of the data transformation in neural network modelling was investigated. Raw data generated from a real process or simulation model consists generally of many variables such as concentration, pressure, temperature, flow rate, etc. Each of these variables is measured in different units having different magnitudes. If raw data are fed into a neural network, then variables having a larger magnitude are given unequal importance due to the nature of the weight update procedure.

Table 5.2. Operating conditions.

Data Set 1		**Data Set 2**	
No. data	41	No. data	45
Max. inlet phenol concentration	190.4 (*mg/l*)	Max. inlet phenol concentration	176.8 (*mg/l*)
Min. inlet phenol concentration	151.9 (*mg/l*)	Min. inlet phenol concentration	143.3 (*mg/l*)
Gas Velocity	0.73 (*cm/s*)	Gas Velocity	0.42-0.7 (*cm/s*)
Particle size	600 *μm*	Particle size	600 *μm*
Temperature	21°C	Temperature	21°C

In order to give equal weightage to all the variables, representative input and output data used to teach the neural network are scaled into a range of 0 to 1 (in case of the sigmoid activation function, between 0.1 to 0.9 due to the limitation of the sigmoid activation function). The simplest method is to scale all the data between 0 and 1 by

using the maximum and minimum values of the variables as scaling constants. For example, if $X(n)$ [n=1, 2, 3, ···., N] is an input sequence, then scaled values, $X_s(t)$ are

$$X_s(n) = \frac{X(n) - X_{\min}}{X_{\max} - X_{\min}}, \quad n = 1, 2, 3, \ldots, N \tag{5.38}$$

On the other hand, sometimes transforming data other than the data scaling method reduces the ability of the system to learn. Some trail and error may be appropriate, or perhaps you might provide a number of alternative forms of the same variable to the neural network.

In this work, a continuous transformation was employed. The general form of a continuous transformation is:

$$y = s_0 \cdot f(s_i x + O_i) + O_o \tag{5.39}$$

where x is a raw data, y is a transformed data, f is a continuous function, s_i, O_i implement an inner scaling of the raw data to map it to an optimal sub-domain of f, and s_0, O_o implement an outer scaling so that y lies within a suitable range for the neural network. Each transformation is identified by its continuous function f, which can be any one of the following: (1)' Identify function', (2) 'Natural logarithm function' ($log(x)$), (3) 'Log of Log' ($log(log(x))$, (4) 'Exponential function' ($exp(x)$), (5) 'Exponential of Exponential' ($exp(exp(x))$, (6) 'Square function' (x^2), (7) 'Fourth Power function' (x^4), (8) 'Square root function' ($\sqrt{x}$), (9) 'Fourth root function' ($\sqrt[4]{x}$), (10) 'Inverse function' ($1/x$), (11) '1/(Square function)', (12) '1/(Fourth Power function)', (13) '1/(Square root function)', (14) '1/(Fourth root function)', (15) 'Hyperbolic tangent function' ($tanh(x)$), (16) Log ($x/(1-x)$).

In order to find the influence of the data transformation for the performance of the neural network, the process estimator and process predictor were performed separately using data set 1. The results of transformation functions for each input

used in neural process estimator are shown in Table 5.3. Fig. 5.9(a)-(b) shows the comparisons between estimation result in the case with a data transformation used as inputs to the neural network and estimation results in the case with raw data as input. From Table 5.4, comparing these RMSE function values, it can be seen that the estimation error is less than 30 % as much in the case where the raw data set was used. As it can be seen from Fig. 5.9(a)-(b), the coupling of the MFNN with data transformation yields an improvement of simulation results. This is true particularly in the case where there is noise in the data set.

Table 5.3. Continuous data transformation function used in the neural process estimator.

Variables	Transformation function used
$L_f(k)$ (output)	Natural logarithm function
$X_a^f(k)$ (output)	1/(Square root function)
$L_f(k\text{-}1)$ (input)	Log $(x/(1-x))$
$L_f(k\text{-}2)$ (input)	Log $(x/(1-x))$
$X_a^f(k\text{-}1)$ (input)	1/(Square root function)
$X_a^f(k\text{-}2)$ (input)	Log of Log
$X_a^l(k\text{-}1)$ (input)	1/(Square function)
$C_s^{in}(k\text{-}1)$ (input)	Square function

When the simulation is carried out with the raw data and the data scaling method only, the topology of the MFNN is 6-7-2. But in case of the simulation with continuous data transformation, the MFNN with 12-9-2 is employed. Transformation functions for each input used in process predictor shown in Fig. 5.9(c), is also shown in Table 5.5.

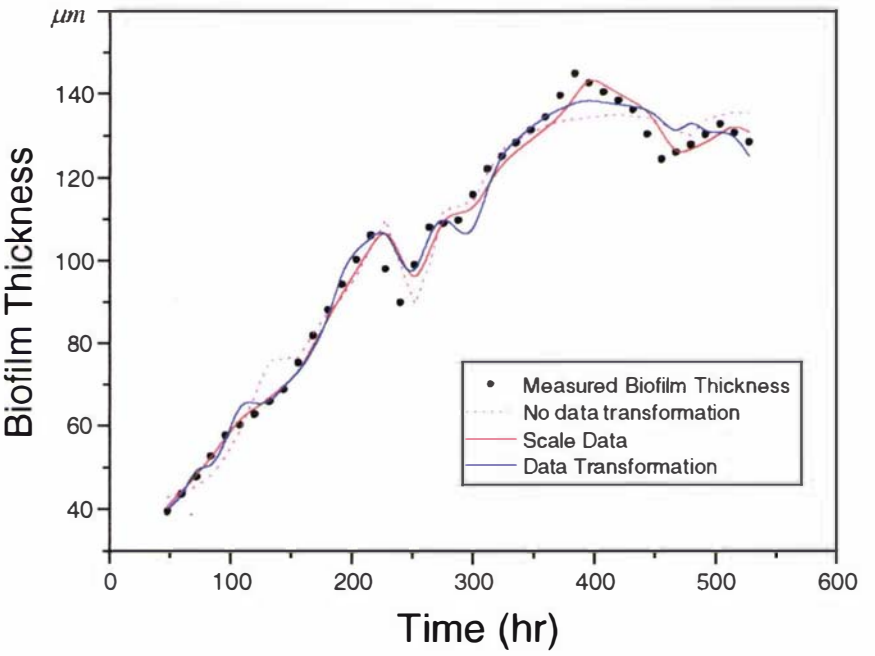
From Fig. 5.9(a)-(c), comparing simulation results, it can be seen that the prediction accuracy where the data scaling method only was used is as much as same in the case where the continuous transformation was used. The continuous transformation method does not provide a significant advantage over the data scaling method for the data set used here. Thus for the sake of simplicity of the neural network, the entire neural network modelling presented here has incorporated only the scaling data method for data transformation since the continuous transformation causes the large topology of the neural network over the data scaling method.

Table 5.4. RMSE comparisons between data transformation and raw data for neural process estimator and predictor.

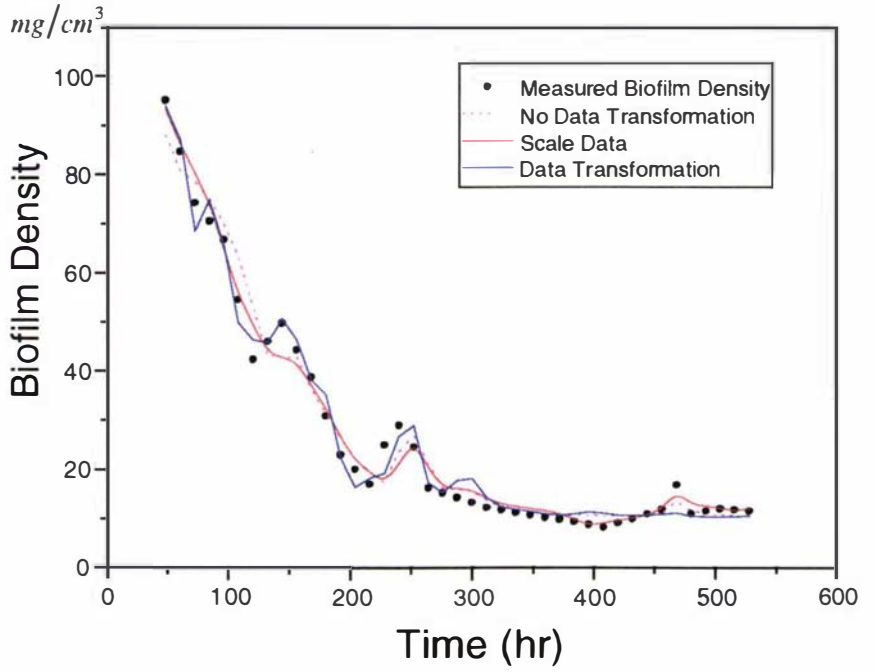
	Neural Estimator		**Neural Predictor**
	RMSE (biofilm thickness)	RMSE (biofilm density)	RMSE (effluent phenol concentration)
MFNN with no data transformation	5.41	4.66	4.88
MFNN with data transformation	4.09	3.10	3.041
MFNN with scale data only	3.76	3.66	2.934

Table 5.5. Continuous data transformation function used in the neural process predictor.

Variables	Transformation function used
$C_s^{out}(k)$ (output)	Square function
$C_s^{out}(k\text{-}1)$ (input)	Square function
$C_s^{out}(k\text{-}2)$ (input)	Hyperbolic tangent function
$C_s^{out}(k\text{-}3)$ (input)	Square function
$L_f(k\text{-}1)$ (input)	Log $(x/(1\text{-}x))$.
$L_f(k\text{-}2)$ (input)	Log $(x/(1\text{-}x))$.
$X_a^f(k\text{-}1)$ (input)	1/(Square root function)
$X_a^f(k\text{-}2)$ (input)	Natural logarithm function
$X_s^f(k\text{-}1)$ (input)	Hyperbolic tangent function
$X_s^f(k\text{-}2)$ (input)	Hyperbolic tangent function
$C_s^{in}(k\text{-}1)$ (input)	1/(Square function)
$C_s^{in}(k\text{-}2)$ (input)	1/(Square function)



(a)



(b)

Figure 5.9. Comparison of prediction results between modelling with data transformation and raw data for neural process estimator (data set 1): (a) biofilm thickness (b) biofilm density.

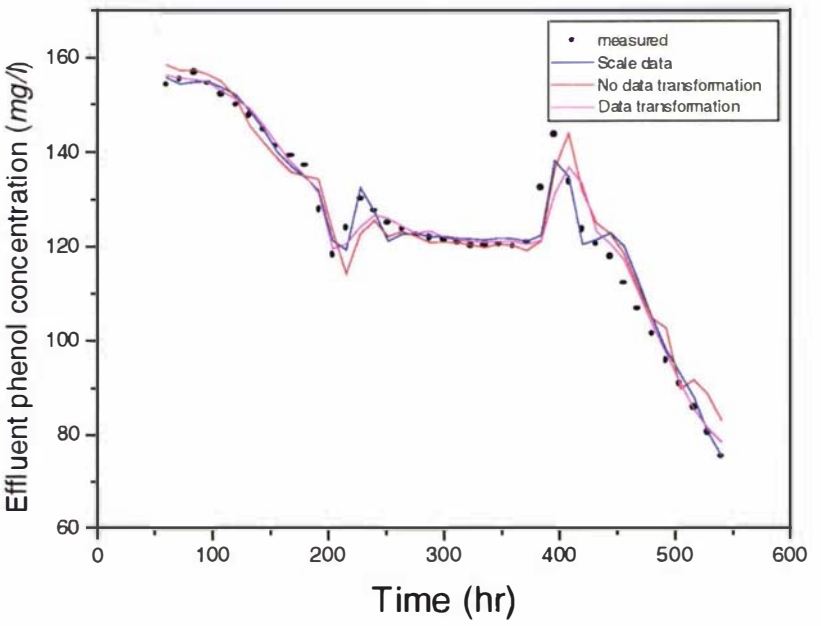


Figure 5.9(c). Comparison of prediction results between modelling with data transformation and raw data for neural process predictor (data set 1).

5.4.3 Neural Process Estimator

When the neural process estimator was applied to data sets 1 and 2, the past biofilm thickness, biofilm density, suspended biofilm density, inlet phenol concentration, and dissolved oxygen concentration are used as inputs to the process estimator.

The data set 1 are written in { L_f(k), X_a^f(k), L_f(k-1), L_f(k-2), X_a^f(k-1), X_a^f(k-2), X_a^l(k-1), C_s^{in}(k-1)}. Output L_f(k) and X_a^f(k) are the one-step ahead biofilm thickness and biofilm density, respectively. The past L_f(k-1), L_f(k-2), X_a^f(k-1), and X_a^f(k-2) are used as inputs to the network. X_a^l(k-1) the suspended biomass concentration, C_s^{in}(k-1) the inlet substrate concentration are also used as inputs.

Even though the superficial liquid velocity and gas velocity affect the variation of the biofilm thickness and biofilm density in a TPFBBR, they were not used as inputs since they were essentially constant at any point of each operating condition in data set 1. Thus, the information they contain would not contribute towards estimating biofilm thickness and biofilm density and prediction for effluent phenol concentration. Unlike the data set 1, the superficial gas velocity was changed from 0.42 *cm/s* to 0.7 *cm/s* at 336 hours after startup. For modelling using data set 2, the gas velocity was considered to be the input.

The output from the neural network process estimator for data set 1 is shown in Fig. 5.10 (a)-(b). For MFNN with cascaded-correlation (C-C) learning with data set 1, the neural network employed 6 inputs and 1 hidden layer with 7 logistic neurons. And there were 2 output neurons using the logistic transfer function. This process estimator closely follows the measured patterns, giving an RMSE of 3.761 on the estimation of biofilm thickness for data set 1, as well, the estimation on the estimation of biofilm density is acceptable (RMSE= 3.662).

For MFNN with extended Kalman filtering (EKF) learning, the topology of neural network was the same as that of MFNN with C-C learning. The RMSE results are 3.692 on the estimation of biofilm thickness and 2.899 on the estimation of biofilm density, respectively. For GRNN as the process estimator, the RMSE's for the biofilm thickness and the biofilm density were 3.601 and 2.982, respectively. GRNN used as neural process estimator employs 6 inputs, 2 hidden layers, and 1 output. The first hidden layer, the pattern layer has 30 neurons. The second hidden layer, the summation layer, has 3 neurons.

The same procedure as that adopted for data set 1 was applied to data set 2. The neural networks were tested on the overall 45 data points. The results of an one-step ahead estimation are shown in Fig. 5.11(a)-(b). The topology of MFNN with C-C learning and EKF learning was 7-10-2, i.e., 7 input neurons and one hidden layer with 10 logistic neurons, and two output neurons using the hyperbolic tangent transfer function. For the topology of the GRNN, GRNN employs 7 inputs, 2 hidden

layers, and 1 output. The first hidden layer, the pattern layer has 33 neurons. The second hidden layer, the summation layer, has 3 neurons.

Table 5.6 shows the comparisons of RMSE for each neural network used for neural process estimator. From Table 5.6, RMSE with data set 1 was a little lower than that with data set 2. From Table 5.6, the performance of MFNN with C-C learning, MFNN with EKF learning, and GRNN is similar but the GRNN estimate relatively more accurate than that by MFNN. The estimated biofilm thickness and biofilm density with three different types of neural networks have proved to be consistent with the experimental data.

Table 5.6. RMSE results of the neural process estimator for data set 1 and 2.

	RMSE of Data Set 1		**RMSE of Data Set 2**	
	Biofilm Thickness	Biofilm Density	Biofilm Thickness	Biofilm Density
MFNN C-C	3.761	3.662	5.111	5.231
MFNN EKF	3.692	2.899	4.539	5.135
GRNN	3.601	2.982	4.400	4.755

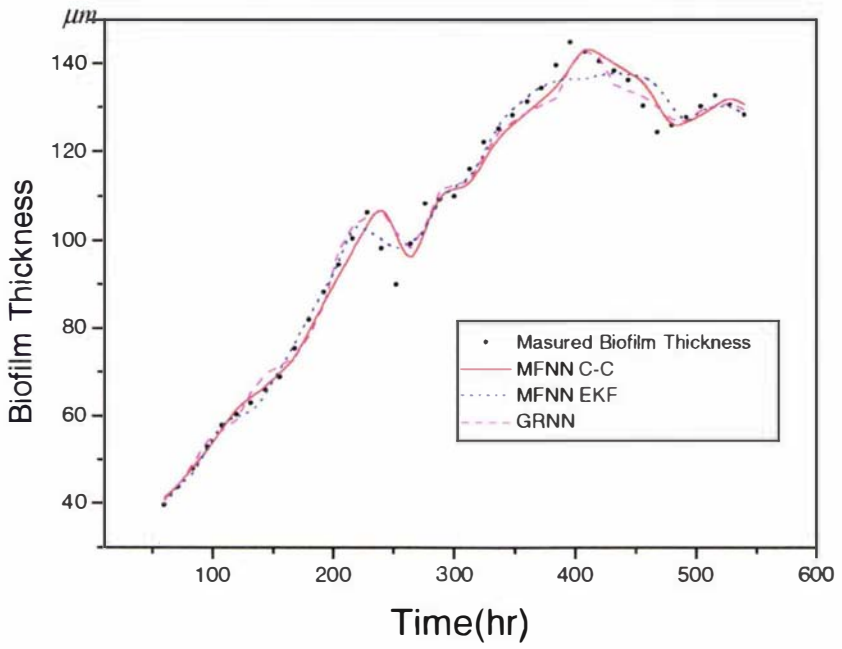


Figure 5.10(a). Results of the neural process estimator for the biofilm thickness using data set 1.

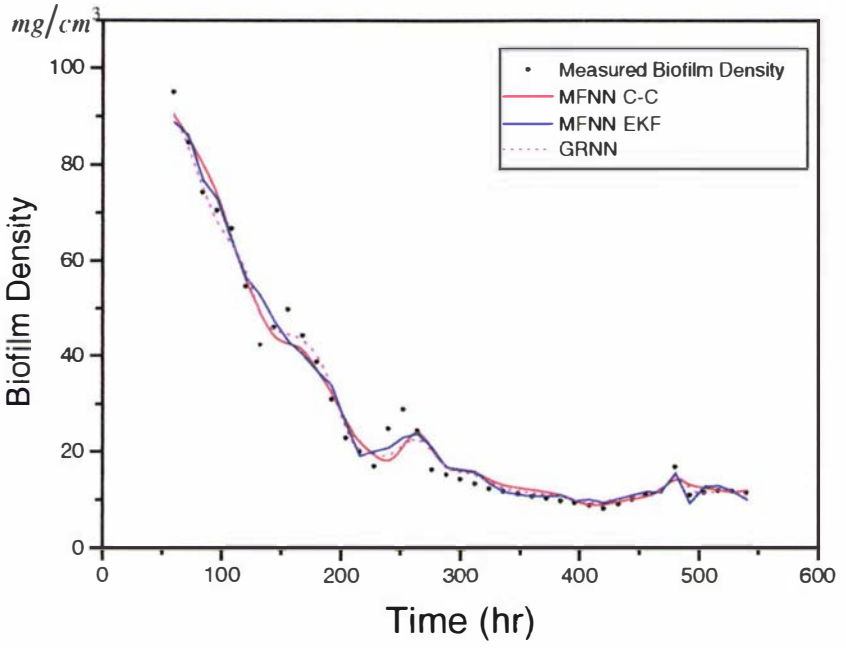


Figure 5.10(b). Results of the neural process estimator for the biofilm density using data set 1.

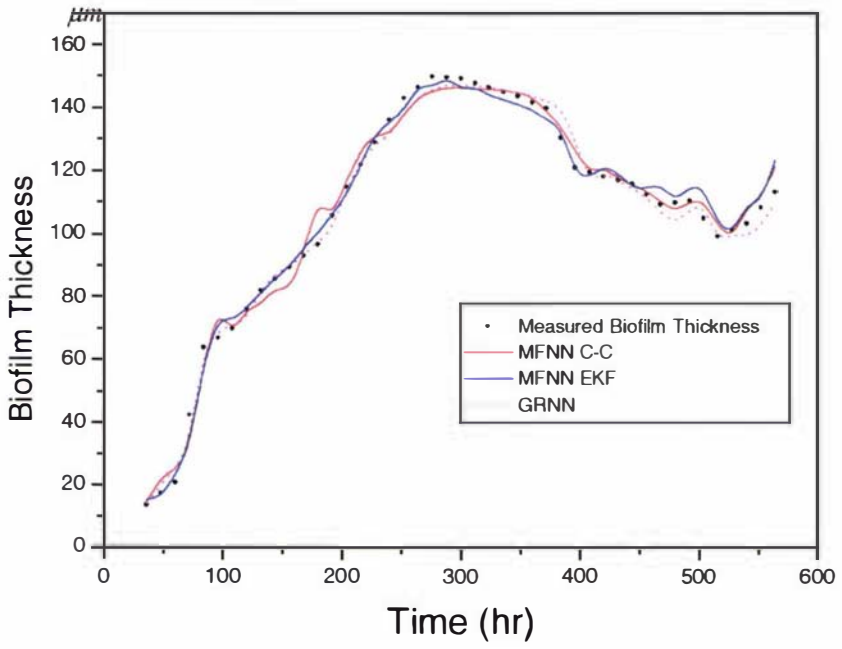


Figure 5.11(a). Results of the neural process estimator for the biofilm thickness using data set 2.

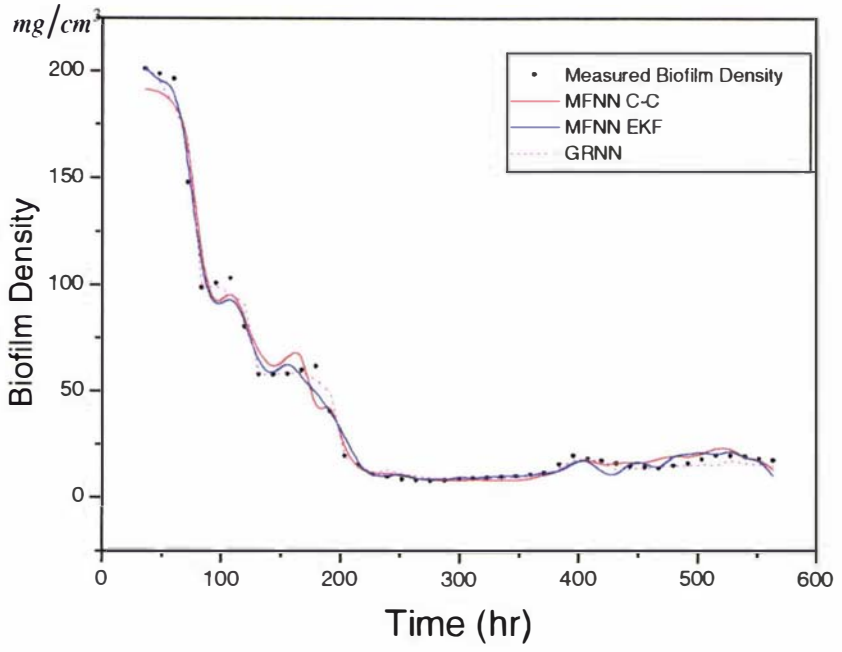


Figure 5.11(b). Results of the neural process estimator for the biofilm density using data set 2.

5.4.4 Neural Process Predictor

The biofilm thickness and biofilm density, which are outputs of the neural process estimator, are subsequently fed into the neural process predictor to form of part of the inputs. Past effluent phenol concentration, suspended biomass concentration, and inlet phenol concentration are also served as inputs for the neural process predictor.

The data set 1 and 2 are written in $\{C_s^{out}(k), C_s^{out}(k-1), C_s^{out}(k-2), C_s^{out}(k-3), L_f(k-1), L_f(k-2), X_a^f(k-1), X_a^f(k-2), X_s^f(k-1), X_s^f(k-2), C_s^{in}(k-1), C_s^{in}(k-2)\}$. Output $C_s^{out}(k)$ is the one-step ahead effluent phenol concentration. The past $C_s^{out}(k-1)$, $C_s^{out}(k-2)$, and $C_s^{out}(k-3)$ are used as inputs to the network. L_f biofilm thickness, X_a^f biofilm density, X_s^f the suspended biomass concentration, and C_s^{in} the inlet substrate concentration are also used as inputs. The topologies of each type of neural network for the neural network predictor are given in Table 5.7.

Table 5.7. Topologies of each neural network.

		Process Estimator	P ocess Predictor
MFNN C-C	data set 1	6-7-2	11-11-1
	data set 2	7-10-2	11-9-1
MFNN EKF	data set 1	6-7-2	11-11-1
	data set 2	7-10-2	11-9-1
GRNN	data set 1	6-30-3-2	11-36-5-1
	data set 2	7-33-3-2	11-38-4-1

Table 5.8 also collects the RMSE results of the MFNN, GRNN for data set 1-2. Fig. 5.12 (a)-(d) show the plots of measured values as well as the curves predicted by the process predictor. Good correlation exists between the measured and predicted for

both MFNN with EKF learning and GRNN. It is demonstrated that the neural process predictor serve as good predictors and at the same time provide a valuable information on dynamic performance of a TPFBBR which is difficult to be modelled and to be predicted.

Table 5.8. The comparison of RMSE for MFNN with EKF and GRNN ($RMSE_{trn}$ = RMSE of training set, $RMSE_{tst}$ = RMSE of testing set).

	Data Set 1		Data Set 2	
	MFNN EKF	GRNN	MFNN EKF	GRNN
	Process Predictor	Process Predictor	Process Predictor	Process Predictor
$RMSE_{trn}$	2.738	2.79	1.538	1.344
$RMSE_{tst}$	3.131	3.64	2.495	1.877

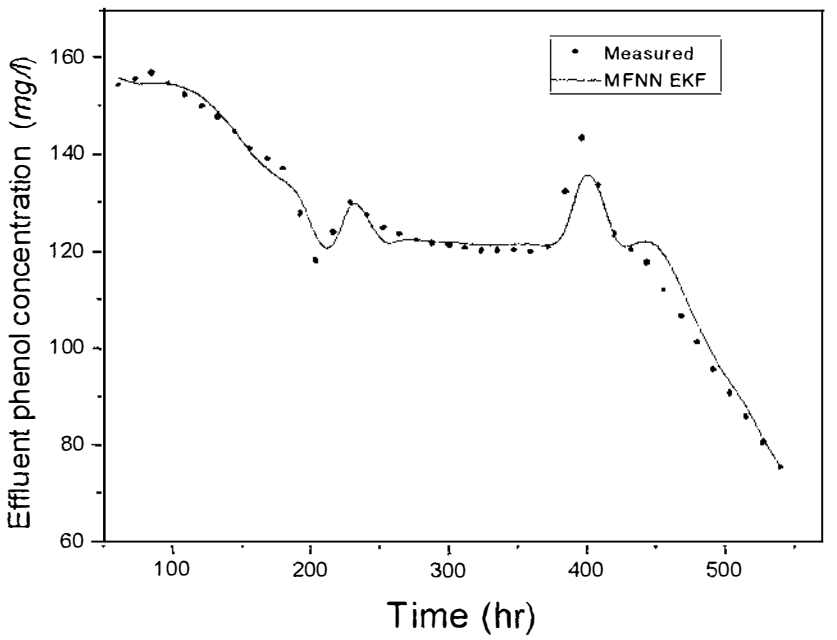


Figure 5.12(a). Results of the neural process predictor (MFNN with EKF) using data set 1.

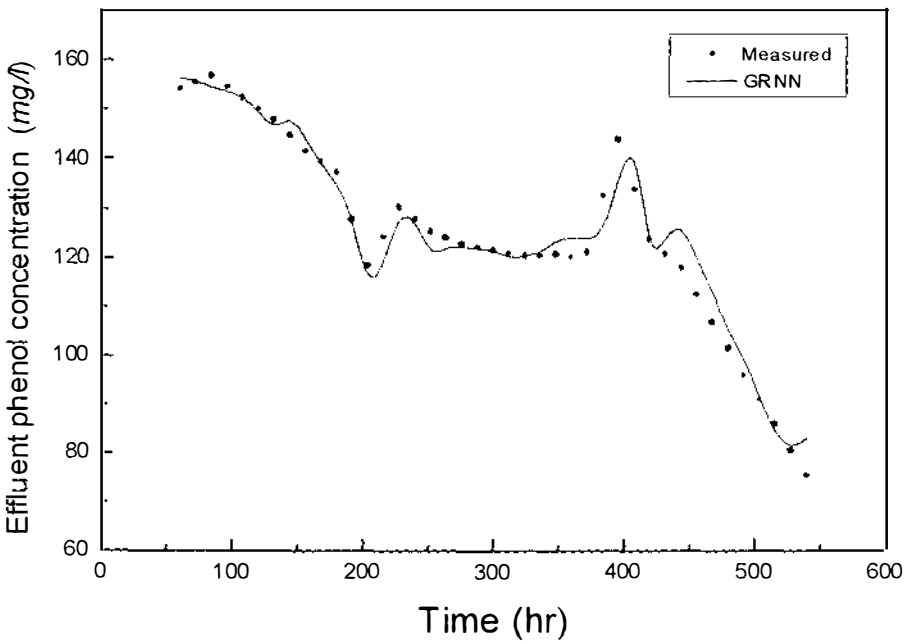


Figure 5.12(b). Results of the neural process predictor (GRNN) using data set 1.

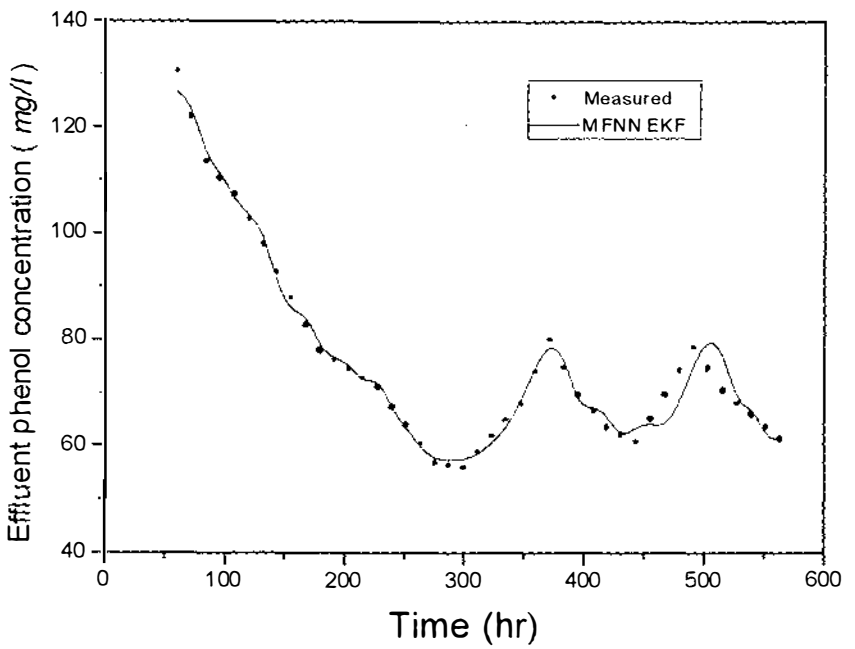


Figure 5.12(c). Results of the neural process predictor (MFNN with EKF) using data set 2.

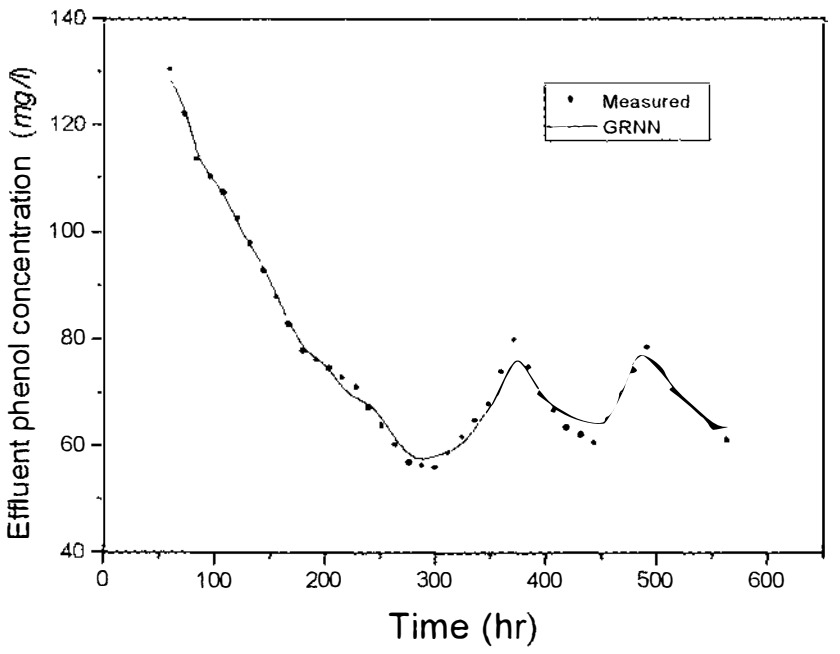


Figure 5.12(d). Results of the neural process predictor (GRNN) using data set 2.

During the startup phases in a TPFBBR, the modelling results using sequential neural network demonstrate that the modelling technique using a GRNN and MFNN provides a valuable tool for predicting the outputs with high levels of accuracy and for understanding the dynamic behavior of processes.

5.5 CONCLUSIONS

A process engineer, who is faced with characterisation or prediction of the process behavior, has to model the considered process. But the derivation of a proper mathematical model to describe a complex bioprocess is usually quite difficult, particularly for a biofilm reactor, such as a TPFBBR.

Neural networks offer an alternative to solve this problem since it does not require any *a priori* knowledge about the structure of the relationships that exist between

important variables. All that is required is to give the neural networks the necessary information and to let it learn by representative examples.

The sequential neural network model presented here is composed of two parts. The first part, *the neural process estimator*, can serve as a nonparametric approximator to estimate difficult-to-model process variables such as biofilm thickness and biofilm density with the available measured variables. The second part, *the neural process predictor*, can predict the dynamic change of performance of the TPFBBR based on the estimated biofilm thickness and biofilm density by the neural process estimator, and other measured variables.

We consider three types of the neural networks, which are MFNN with C-C learning, MFNN with EKF learning, and GRNN, to develop the sequential neural network model with two different operating data sets. It has been shown that the neural process estimator component of the sequential neural network models are capable of capturing the nonlinear relationship between process variables such as biofilm thickness and biofilm density and process input variables with no prior knowledge about the complex biofilm growth behavior occurring in a TPFBBR. The neural process predictor component of the sequential neural network can also predict the effluent phenol concentration with high level of accuracy, which is difficult to be captured by existing mechanical models. Thus, the sequential neural network model performs well in modelling the dynamics of the complex TPFBBR, not only in the training phase but also in the testing phases.

It can be concluded that instead of the complex TPFBBR model consisting of a dynamic biofilm growth model, reaction-diffusion model, and reactor model, the sequential neural network modelling approach proposed here provides a good alternative to describe the dynamic behavior of a TPFBBR and has the potential to be successfully implemented within a control strategy. Finally, the modeling approach presented here is readily applicable to a variety of other complex processes.

Chapter 6

Intelligent Hybrid Model with a Neurofuzzy Process Estimator for a TPFBBR

6.1 INTRODUCTION

The mechanistic model developed in Chapter 4 and the artificial neural network model developed in Chapter 5 have drawbacks. Real biological processes such as TPFBBR, are strongly characterized by nonlinear dynamics and are usually complex and poorly known. Therefore, formulating an accurate model based on a mechanistic approach requires very time- and money-consuming tasks. An artificial neural network modelling approach, on the other hand, is mainly data driven and the resulting model is not believed to have any exploration properties. Therefore, data used for neural network-based modelling should cover the whole domain of interest in order to avoid the danger of exploration when using the model. It is very difficult

to extract structural knowledge for the processes under consideration and the empirical approach highly is criticised as a scientific approach.

In order to overcome the weak points of both the mechanistic and neural network models, a so-called *hybrid model*, which is a combination of a mechanistic and empirical model has recently been introduced (Psichogious & Ungar, 1992; Su et al., 1992; Tompson & Kramer, 1994; van Can et al., 1997). In a hybrid model, a part of the poorly and inaccurately known processes are modelled by an empirical model such as ANN or neurofuzzy system, and the behaviour of the known processes is modelled mechanistically.

In a TPFBBR, processes associated with the diffusive transport of substrate with the biofilm from the liquid phase and processes associated with the axially-dispersive transport of substrate in the liquid phase are usually more accurately known than processes associated with dynamic biofilm growth on the support particle and the kinetics of conversion of substrate in the biofilm. The main goal of this chapter is to develop the intelligent hybrid model for a TPFBBR because the hybrid model based on a combination of both known mechanistic and empirical knowledge and empirical data in a TPFBBR, may be reliable, and offers potential advantages for practical application. In the intelligent hybrid model presented, the neurofuzzy model is used to model the unknown process of dynamic biofilm growth on the support particle, combined with the known processes such as the axial-dispersion and reaction-diffusion models to build the *intelligent hybrid model* of a TPFBBR. In this work, we first focus on developing the neurofuzzy model as *process estimators* to estimate the variation of the biofilm thickness and biofilm density based on the available measurement variables. Next, we aim to explore the intelligent hybrid model, which is the concept of combining the neurofuzzy model for estimating the biofilm thickness and biofilm density with mechanistic models to predict the performance of a TPFBBR. In our hybrid model, the neurofuzzy model is combined with a reaction-diffusion model and axial-dispersion model to show the dynamic behavior and performance of a TPFBBR according to the variation of biofilm density and biofilm

thickness. It is demonstrated that this intelligent hybrid model provides a valuable tool for understanding the dynamic behavior and a comprehensive dynamic model of a TPFBBR.

6.2. FUZZY SYSTEM

6.2.1. Fuzzy Sets

A fuzzy system is used to represent the imprecision found in natural language. To describe this, Zadeh (1973) introduced the concept of a fuzzy set. Fuzzy sets represent vague description of objects i.e. tall, small, cold, bright, etc.

For conventional sets, rigid membership requirements are imposed upon the objects within the set. An object is a member of a set to degree 0 (not in the set at all) or 1 (completely in the set). For example, the set of TALL men could be defined to be all men 6 feet or taller (Fig. 6.1). As shown in Fig. 6.1, the conventional set classifies a man as either TALL, or not TALL at all. There is no middle ground. In contrast, fuzzy sets have more flexible membership requirements that allow for partial membership in a set. A man 6 feet tall is a member the fuzzy set TALL to degree 0.5 (Fig. 6.1). A man 5 feet 6 inches tall is TALL to degree 0.25, a man 6 feet 6 inches tall is TALL to degree 0.75.

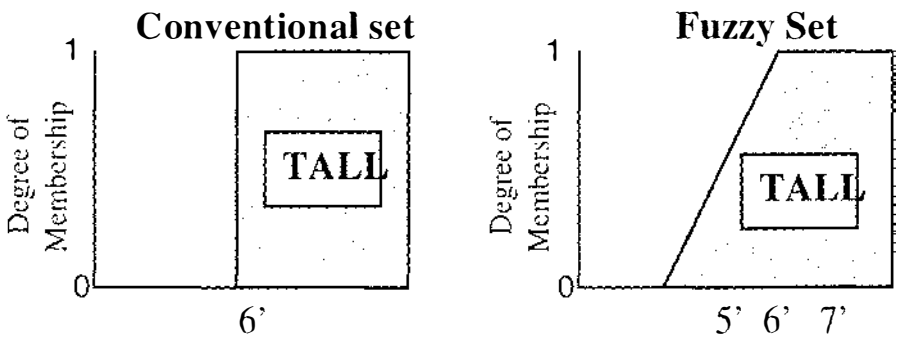


Figure 6.1. Conventional Sets vs. Fuzzy Sets.

Mathematically, a fuzzy set, A, is a function defined on the universe of discourse, X, given by:

$$u_A(x): X \rightarrow [0,1] \tag{6.1}$$

where A is the linguistic variable (or fuzzy label) describing the variable x. The universe of discourse of a variable is its range and can be either continuous or discrete. $u_A(x)$ represents the membership function, x belonging to the fuzzy set A. In general, the shape of a membership function depend on the application and can be trapezoidal, bell-shaped, triangular, or Gaussian, etc as shown in Fig. 6.2.

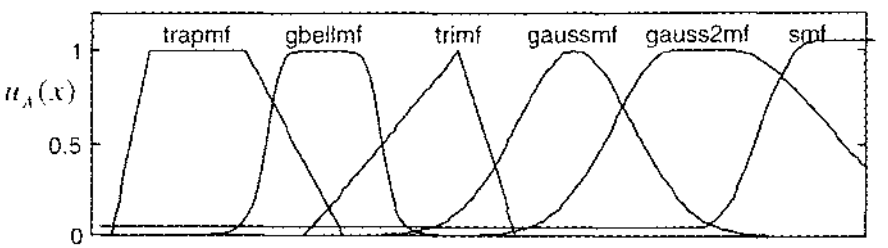


Figure 6.2. Different shapes of membership functions.

Fig. 6.3 illustrates one example of the discourse for the linguistic variable temperature. Linguistic values, which define these variables, are: Cold, Warm, and Hot.

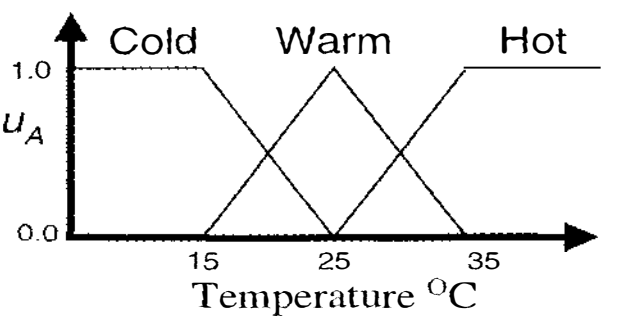


Figure 6.3. Typical fuzzy set for temperature.

6.2.2 Fuzzy Inference System

To construct a fuzzy system we have to describe mapping from one universe of discourse to another and this can be achieved using fuzzy algorithms. Fuzzy IF-THEN rules (or fuzzy *implication* statements) can be used to describe part of such a mapping and a collection of rules from an algorithms. Depending on the encoding method of a fuzzy algorithm, several fuzzy-rules-base systems can be distinguished: linguistic fuzzy model (Mamdani, 1977), fuzzy relational model (Pedrycz, 1983), Takagi-Sugeno model (Takagi & Sugeno, 1985).

In fuzzy inference system (or fuzzy-rules-base system), every fuzzy rule has a two parts:

- antecedent part(premise), expressed by: ***IF*** ···
- consequent part, expressed by: ***THEN*** ···

The antecedent part is the description of the state of the system which should turn on the rule, and the consequent is the action that the operator who controls the system must take. Consider the following example of dealing with a problem of a high effluent BOD based on linguistic fuzzy model:

$$\text{IF } \overbrace{(\text{EBOD is Large}) \text{ AND } (\text{ESS is Small})}^{\text{antecedent}} \text{ THEN } \overbrace{(\Delta\text{WSFR is Negative Large})}^{\text{consequence}} \qquad (6.2)$$

where EBOD is effluent BOD, ESS is effluent suspended solid concentration, and ΔWSFR is change in the waste sludge flow rate. In this example, EBOD, ESS, and ΔWSFR are linguistic variables, and Small, Large, and Negative Large are linguistic values (or fuzzy labels) that are characterised by appropriate membership functions. The linguistic values 'Small', 'Large', and 'Negative Large' have a certain degree of vagueness and fuzziness. This fuzziness can be described by membership functions which can assume different curves, e.g. straight lines, bell-shaped, Gaussian, and so forth (Fig. 6.2).

Most fuzzy inference system use production rules to represent the relation among the linguistic variables and derive actions from the inputs. The computation of fuzzy rules is called fuzzy rule inference that is a calculus consisting of two main steps:

(1) aggregation step - the antecedents are evaluated using membership functions to belief levels,
(2) composition step - the antecedents are then combined using fuzzy operator (such as T-norm and T-conorm) to produce the final output activation level.

The antecedent of the fuzzy rule is formed from the *intersection* (the fuzzy **and** operation) of the univariate linguistic statements, which can be represented by a multivariate fuzzy set. Via the fuzzy *implication* operator (**IF ··· THEN ···**) individual rules map the multivariate fuzzy set into the rules consequence. A complete rule base is produced from the *union* (performed by the fuzzy **or** operator) of all the fuzzy rules.

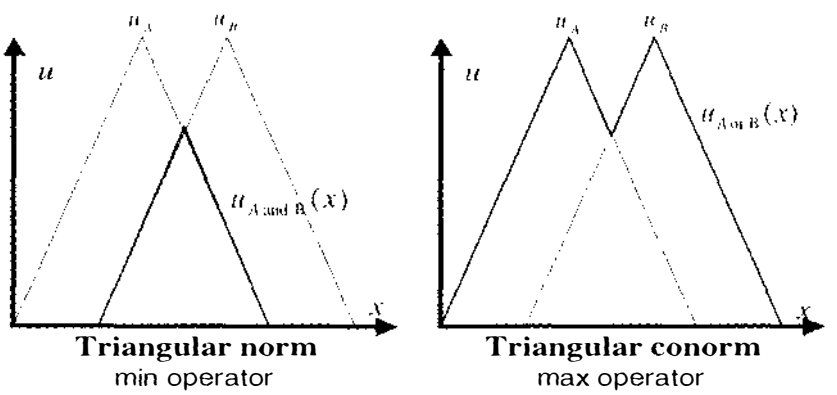
To implement a fuzzy system (or controller), the functions that performs the logical fuzzy operations, **and**, **IF ··· THEN ···**, and **or** have to be defined. There are many ways to define the fuzzy *implication* and the fuzzy operator **and**. Nearly, 40 distinct fuzzy implication functions have been described in the literature (Lee, 1990). Generally, *intersection* operators are called T-norms and *union* operator called T-conorms. These norms provide a wide range of suitable functions but the most often used operations are the ***min*** and ***product*** operator for **and** operator, and the ***max*** and ***sum*** operator for **or** operator. They are shown in Table 6.1.

For example, suppose that we defined two fuzzy sets by their memberships u_A and u_B which have triangular shape (dotted lines on Fig. 6.4a). The application of T-norm gives the fuzzy set A **and** B which is represented by its membership function

$u_{A \text{ and } B}(x)$ (solid line on Fig. 6.4(a)). The application of T-conorm on these fuzzy sets gives the fuzzy set represented with solid line on the Fig. 6.4.

Table 6.1. The most popular logical fuzzy operators.

T-norms	• ***min***: $u_A(x)$ **and** $u_B(x)$ =min $\{u_A(x), u_B(x)\}$ • ***algebraic product***: $u_A(x)$ **and** $u_B(x) = u_A(x) * u_B(x)$ ***bounded product***: $u_A(x)$ **and** $u_B(x)$ =max(0, $u_A(x)+u_B(x)-1$
T-conorms	• ***max***: $u_A(x)$ **or** $u_B(x)$=max $\{u_A(x), u_B(x)\}$ • ***algebraic sum***: $u_A(x)$ **or** $u_B(x) = u_A(x)+u_B(x)-u_A(x)*u_B(x)$ • ***disjoint sum***: $u_A(x)$ **or** $u_B(x)$ = max $\{\min(u_A(x), 1-u_B(x)), \min(1-u_A(x), u_B(x))\}$
Fuzzy Implication	• ***material implication***: $u_{A\to B}(x) = u_A(x) \to u_B(x) = not\,(u_A(x))$ or $u_B(x)$ • ***propositional calculus***: $u_{A\to B}(x) = u_A(x) \to u_B(x) = not\,(u_A(x))$ or $(u_A(x)$ and $u_B(x))$



(a) (b)

Figure 6.4. Graphical representation of a fuzzy operator.

Every fuzzy system (or fuzzy controller) is composed of four principal blocks as shown Fig. 6.5 (Jang, 1993):

- knowledge base (rules and parameters for membership functions)
- decision unit (inference operations on the rules)
- fuzzification interface (transformation of the crisp inputs into degrees of match with linguistic variables)
- defuzzification interface (transformation of the fuzzy results of the inference into a crisp output)

In many engineering applications, the inputs and outputs are numerical values, rather than fuzzy sets. To deal with this, the fuzzy system must be equipped with conversion interfaces, so-called *fuzzification* and *defuzzification* units, as shown in Fig. 6.5.

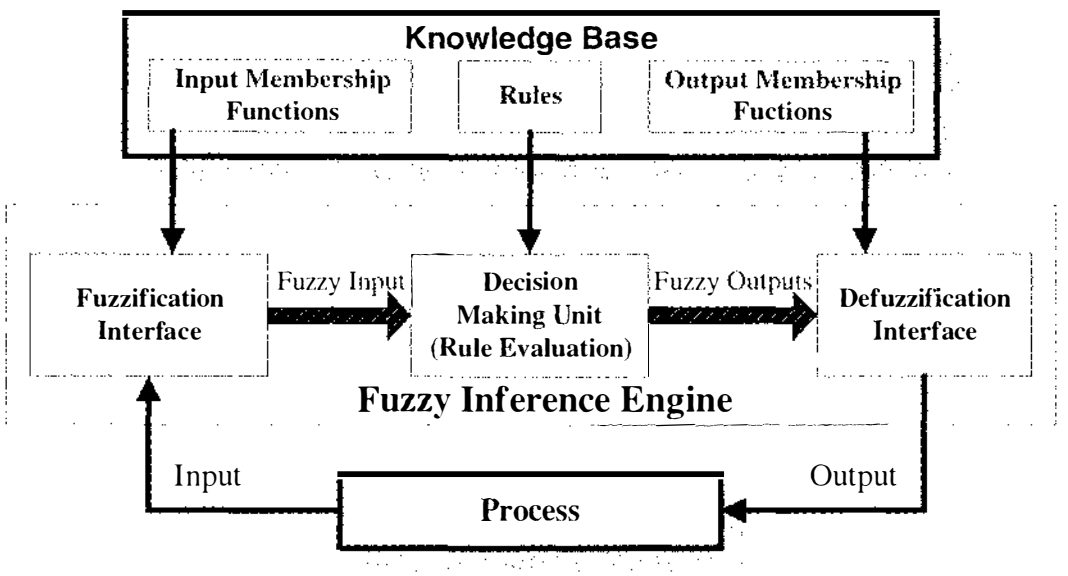


Figure 6.5. General structure of fuzzy inference system.

The fuzzy inference engine of the system from Fig. 6.5 is represented on the Fig. 6.6.

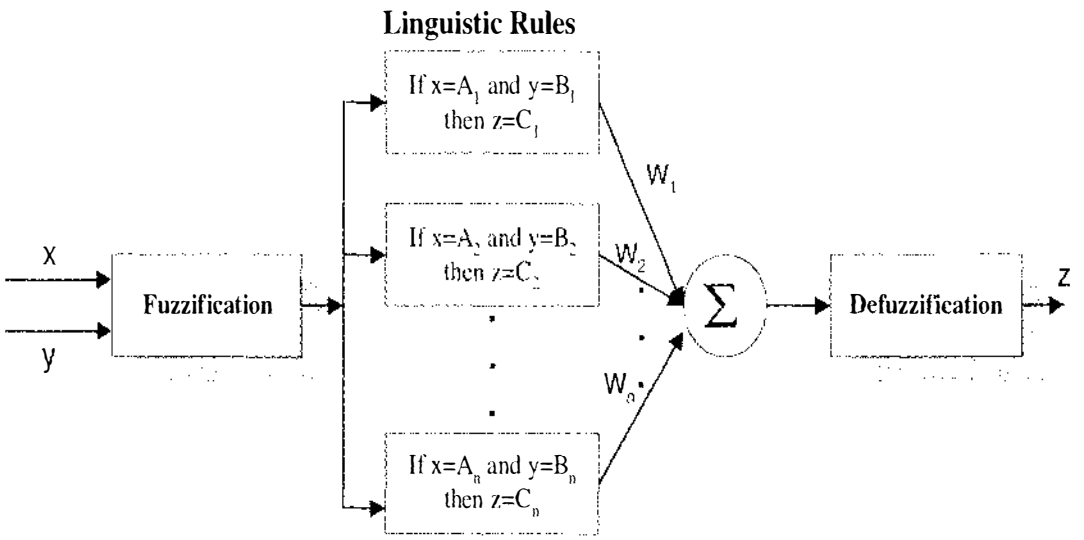


Figure 6.6. General structure of fuzzy inference engine.

In *fuzzification* unit, input values are considered as fuzzy singletons and membership grades of all fuzzy propositions in the rule antecedents are evaluated. Fuzzification means using the membership functions of linguistic variables to compute each term's degree of validity at a specific point of the process. When a fuzzy rule fires (activates), it fires to a certain degree of depending on the belief level in each antecedents are evaluated in the premise of the rule. The antecedents are evaluated using membership functions to belief levels, which are then combined using fuzzy operator (T-norm and T-conorm) to produce the final output activation level. Finally, the output activation level is used to either scale or clip the fuzzy output set. Clipping the output is called Max-Min inference, and scaling the output is called Max-Dot inference. There are a number of fuzzy inference engines, but the most cited in the literature (Jang, 1993) are:

1. **Max-Dot method** (type 1). The final output membership function for each output is the union of the fuzzy sets assigned to that output in a conclusion after scaling their degree of membership values to peak at the degree of

membership for the corresponding premise (modulation by clipping) (Zimmermann, 1990).

2. **Max-Min method** (type 2). The final output membership function is the union of the fuzzy sets assigned to that output in a conclusion after cutting their degree of membership values at the degree of the corresponding premise (linear modulation). The crisp value of output is, most usually, the center of gravity of resulting fuzzy set (Lee, 1990).

3. **Takagi and Sugeno's method** (type 3). Each rule's output is a linear combination of input variables. The crisp output is the weighted average of each rule's output (Takagi & Sugeno, 1985).

See Fig. 6.7 for an example of both Max-Min and Max-Dot methods of fuzzy inference engine. The higher the output activation level of for true, the more it will contribute to the combined output of all the rules. Once all of the fuzzy output sets have been computed, they are summed or unioned together to produce the combined fuzzy output set (Fig. 6.8).

How do fuzzy rules produce final output? The result produced from the evaluation of fuzzy rule is, of course, a fuzzy set, which may be the sum or union of many fuzzy sets, with each rule that fired contributing a piece of the final output set. This fuzzy set is then converted into a single output value by a process known as *defuzzification.* Several defuzzification methods have been developed: (1) The Center of Maximum (CoM), (2) The Mean-of-Maximum (MoM), and (3) The Center-of-Area (CoA) methods. The most common method of defuzzification is the center of gravity (or centriod) method. In centroid method, the output value is equal to the weighted average of the positions of the centroids of output membership functions weighted by their actual membership grade (Fuller, 1995)

$$y = \frac{\sum_{i=N}^{N} u_i C_i}{\sum_{i=1}^{N} u_i} \tag{6.3}$$

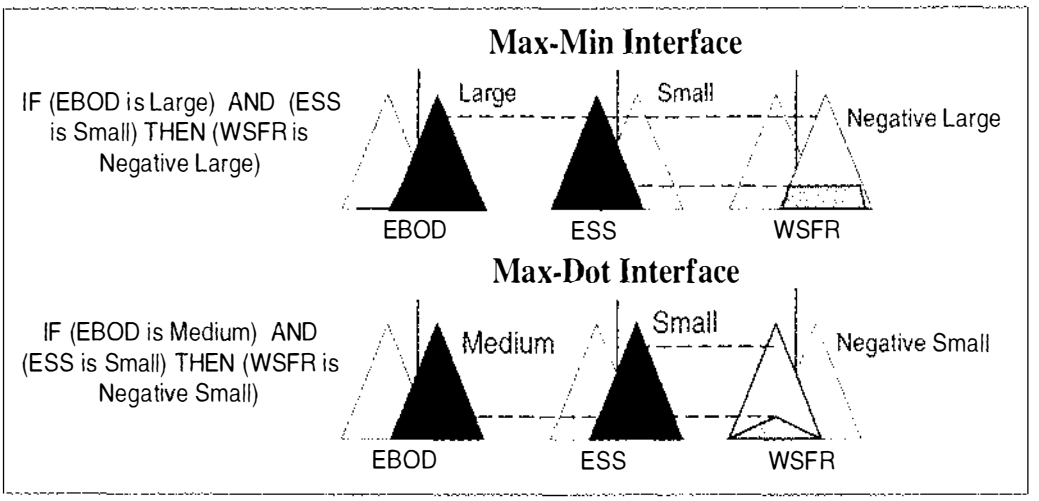


Figure 6.7. Max-Min and Max-Dot Interface.

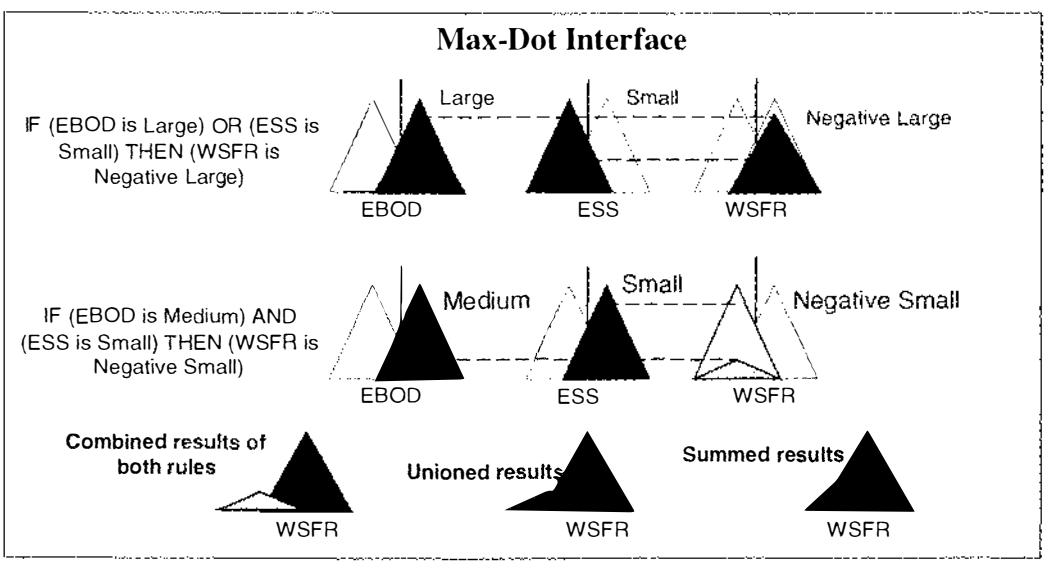


Figure 6.8. Max-Dot interface with unioned and summed results.

6.2.3 Takagi & Sugeno (TS) Fuzzy Model

Takagi and Sugeno (1985) developed a hybrid modelling technique designed to combine conventional and fuzzy modelling. The resulting model, called TS model is represented by a series of fuzzy rules of the form:

$$\text{IF } (X \text{ is } A^{i}) \quad \text{THEN} \quad (y = f_i(x_i)) \tag{6.4}$$

where $f_i(x_i)$, defined on $x_i \subset X$, is a local model used to approximate the response of the system in the region of the input space represented by the antecedent. The function $f_i(x_i)$ are often chosen as affine linear forms $y_i = a_i^T x + b_i$, where a_i is a parameter vector b_i is a scalar offset.

The overall output of TS model is calculated as a weighted average of the rule contributions:

$$y = \frac{\sum_{i=1}^{K} u_{A_i}(X) f_i(x_i)}{\sum_{i=1}^{K} u_{A_i}(X)} \tag{6.5}$$

where K is the number of rules and u_{A_i} is the membership degree of the i^{th} rule antecedent. For a=0, the TS model is equivalent to the linguistic model with singleton consequence. Adaptive-neural-network-based fuzzy inference system (ANFIS) developed by Jang (1993) is one of the most popular approaches to neurofuzzy modelling and is based on this type of fuzzy model structure.

6.3 NEUROFUZZY MODELLING

6.3.1 Contact Points of Fuzzy System and Artificial Neural Networks

The knowledge of artificial neural networks (ANNs) has been developed greatly in the recent years. Due to its strong nonlinear mapping and learning abilities, the application of ANN to modelling wastewater treatment processes has been successful (Capodaglio et al., 1991; Cote et al., 1995; Du et al., 1995; Zhao et al., 1997).

For dynamic process modelling, the neural networks-based model can be thought of as the nonlinear counterpart to ARMA (AutoRegressive Moving Average) models. The main advantages of using ANNs in process modelling are: (1) it has the ability to learn complex nonlinear relationship with limited *prior* knowledge of the process structure (2) it can perform inferences for an unknown combination of input variables (Hong et al., 1998). So ANNs are prime candidates for application in dynamic process modelling for the representation of nonlinear processes. However, analysis of the trained neural networks is difficult since these models appear as black-box models. Neither is it possible to extract structural knowledge for the process under consideration from the trained neural networks, nor is it easy to determine a suitable topology of the ANNs for a special problem and to set the parameters of the learning algorithms.

On the other hand, since Zadeh's first pioneering paper (Zadeh, 1965), there have been fuzzy modelling attempts to combine numerical and symbolic processing into one framework. Fuzzy modelling is knowledge-based system consisting of linguistic **IF-THEN** rules that can be constructed using the knowledge of human experts in the given field of interest. Fuzzy modelling also utilises universal approximators that can realise nonlinear mappings. These features allow qualitative knowledge to be combined with quantitative data in complementary ways (Babuska & Verbrunggen, 1996). Compared to other nonlinear approximation techniques (such as ANNs), fuzzy modelling provides a more transparent representation of the nonlinear systems and appears very useful when the responses to change in manipulated variables are nonlinear or when there is a lack of well-defined mathematical model. However, as system complexity increases, reliable fuzzy rules and membership functions used to describe the systems behaviour are difficult to determine. Furthermore, due to the

dynamic nature of biological process such as activated sludge process, fuzzy rules and membership functions must be adaptive to the changing environment in order to continue to be useful. The quality of fuzzy modelling can be significantly influenced by changing shapes of membership functions and fuzzy rules. Thus methods for performing the adjustment of membership functions and modification of fuzzy rules are necessary.

The advantages of ANN in compensating for the weak points of fuzzy system, and the advantages of fuzzy systems in compensating for weak points of ANN are shown in Fig. 6.9. This mutual improvement is achieved by combining fuzzy system and ANN (Fig. 6.9), and this new method is called *neurofuzzy modelling* (or *fuzzy neural network*). As shown in Fig. 6.9, the aim of neurofuzzy system is to combine collectively the benefits of both fuzzy system and ANN. Simply, the given system is expressed as linguistic fuzzy expressions and learning methods of ANN are used to learn the system. Furthermore, the neurofuzzy system can prevent the knowledge acquired through learning based on the fuzzy knowledge from being thrown into a black box. In addition, neurofuzzy system is also capable of extracting fuzzy knowledge from numerical data since they allow incorporation of both numerical and linguistic data into the system.

Generally, these neurofuzzy systems have the following features (Jin et al., 1995):

1. A fuzzy system is used to create a relevant perception perspective, which possesses very clear physical meanings.
2. All the fuzzy rules are expressed by a group of weights of an ANN and can be adjusted in a more effective way.
3. The nonlinear characteristic of the ANN endows the fuzzy model greater abilities to describe a given complex system.

6.3.2 Background of Neurofuzzy Systems

As described in the previous section, to overcome inherent drawbacks with fuzzy system (Fig. 6.9), namely, the choice of appropriate fuzzy **IF-THEN**-rules and membership function, and the lack of learning function in order to tune these in order to improve the quality of modelling, recently many researchers have focused on the so-called neurofuzzy modelling and control. A number of different schemes have been developed, such as the adaptive-neural-network-based fuzzy inference system (ANFIS) (Jang, 1993), neural networks with fuzzy weight (Buckley & Hayashi, 1994), neuro-fuzzy adaptive models (Brown & Harris, 1994), and fuzzy neural network (Nack and Kruse, 1996). Generally, the neurofuzzy systems can be divided into two groups:

1. Neural network based fuzzy inference system (NNFIS)
2. Fuzzy neural network (FNN)

The objective of NNFIS is to incorporate neural concepts, such as learning and parallelism, into fuzzy inference systems. The architecture of the systems is parallel, and they exploit the same learning algorithms, which are used with neural networks. In the FNN (Pedrycz, 1992; Gupta & Rao, 1994), the fuzzy ideas are incorporated into neural networks. The FNN consists of two components: a fuzzy system and an ANN. the fuzzy system can be either a fuzzy inference block which converts linguistic information for the neural network or the neural network can drive the fuzzy inference block. The only NNFIS is considered as the neurofuzzy system and studied in more detail in this thesis. Table 2 shows some examples of the neurofuzzy systems used in several recently introduced NNFIS.

Simutis et al. (1993) showed the application of fuzzy-aided neural network for real-time state estimation and process prediction in the alcohol formation step of production-scale beer brewing. Ye et al. (1994) applied the neurofuzzy system with a five layer neural network for the control of fed-batch cultivation of recombinant

Escherichia coli. In their application, the change in pH of the culture broth and the specific growth rate were used as inputs to neurofuzzy system to calculate the glucose feeding rate. They demonstrate that a feedforward-feed- back control strategy with neurofuzzy system is a promising control strategy for the control of high cell density cultivation and high expression of a target gene in fed-batch cultivation of a recombinant strain.

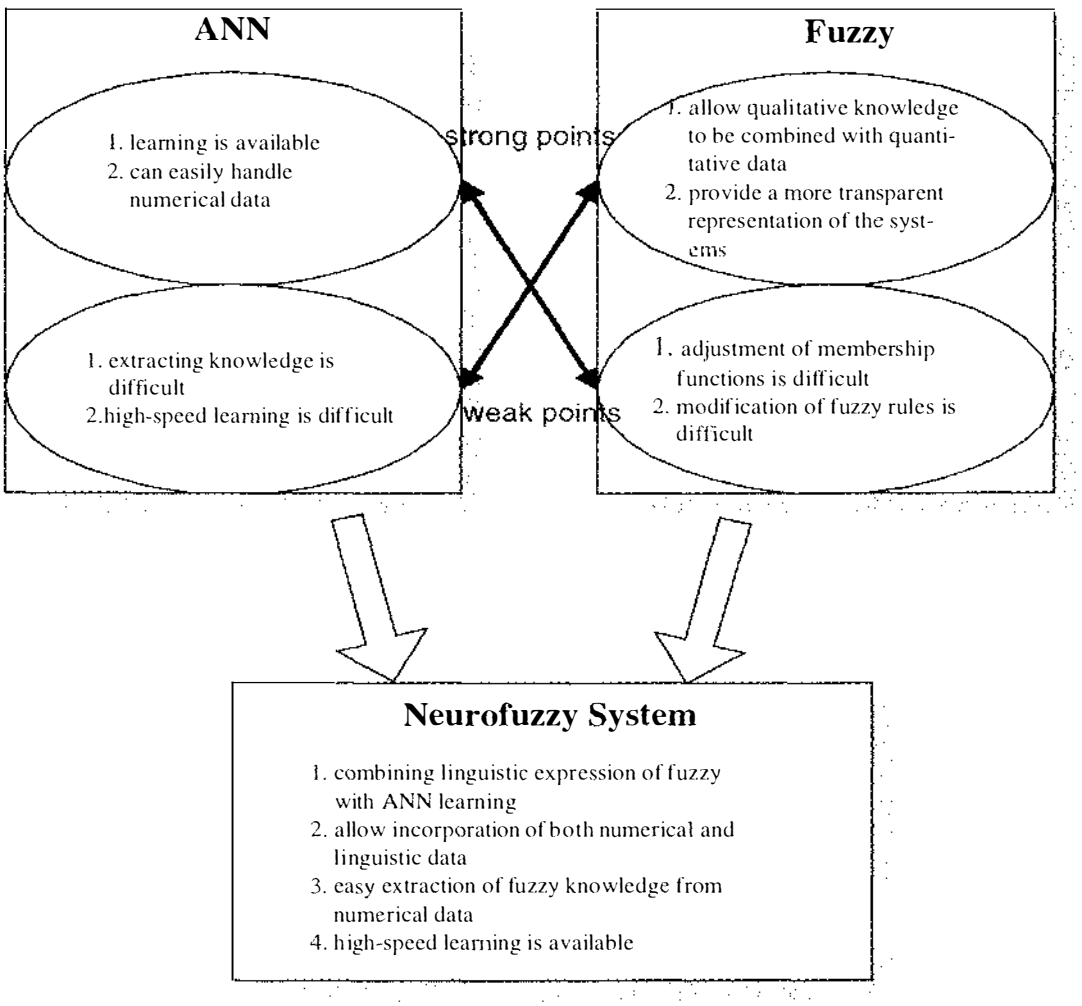


Figure 6.9. The main advantages of the neurofuzzy system.

Table 6.2. Some examples of the neurofuzzy systems.

Neurofuzzy system	Antecedent membership functions	Partitioning	Rule connectives	Consequent membership functions	Defuzzifi-cation
Horikawa et al. (1992)	bell-shaped	adaptive grid	and/product	singleton or monotonic	WA, Tsukamoto
Nack & Kruse (1993)	Tsukamooto's monotonic	adaptive grid	and/min	Tsukamooto's monotonic	Tsukamoto
Jang (1993)	bell-shaped	adaptive grid	and/product	singleton, or functional	WA
Lin & Lee (1991)	bell-shaped	adaptive grid	and/min or/max	bell-shaped	COA,WA
Nie & Linkens (1993)	bell-shaped	Cluster	and/product	Singleton	WA
Berenji & Khedhar (1993)	Triangular	adaptive grid	and/soft-min	triangular	MOM
Wang & Mendel (1992)	bell-shaped	radial	and/product	singleton	WA

(COA= Center of Area, WA= weighted average, MOM= Mean of Maximum)

6.3.3 The Architecture of the ANFIS Neurofuzzy System

One of the interesting architectures for a neurofuzzy system (Table 6.2) is ANFIS (Adaptive Neural Fuzzy Inference System) which is functionally equivalent to Takagi and Sugeno's method described in the previous section (Jang, 1993). To describe ANFIS, let us consider a system which has two inputs x and y and only one output z. In addition, the rule base contains only two fuzzy rules.

Rule 1: **IF** x is A_1 **and** y is B_1 **THEN** $z_1=a_1x+b_1y$

(6.6)

Rule2: **IF** x is A_2 **and** y is B_2 **THEN** $z_2=a_2x+b_2y$

The firing levels of the rules are computed by

$$\begin{aligned} \alpha_1 &= A_1(x_0) \times B_1(y_0) \\ \alpha_2 &= A_2(x_0) \times B_2(y_0) \end{aligned} \tag{6.7}$$

where can be modelled by any continuous T-norm (Table 1) , e.g. product T-norm

$$\begin{aligned} \alpha_1 &= A_1(x_0) \wedge B_1(y_0) \\ \alpha_2 &= A_2(x_0) \wedge B_2(y_0) \end{aligned} \tag{6.8}$$

then the individual rule outputs are derived from the relationships

$$z_1 = a_1x_0 + b_1y_0, \; z_2 = a_2x_0 + b_2y_0 \tag{6.9}$$

and the crisp control action is expressed as

$$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2} = \beta_1 z_1 + \beta_2 z_2 \tag{6.10}$$

where β_1 and β_2 are the normalised values of α_1 and α_2 with respect to the sum $(\alpha_1+\alpha_2)$, i.e.

$$\beta_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2}, \quad \beta_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2} \tag{6.11}$$

The ANFIS, a hybrid neural net computationally identical to Takagi & Sugeno type of fuzzy reasoning is shown in the Fig. 6.10.

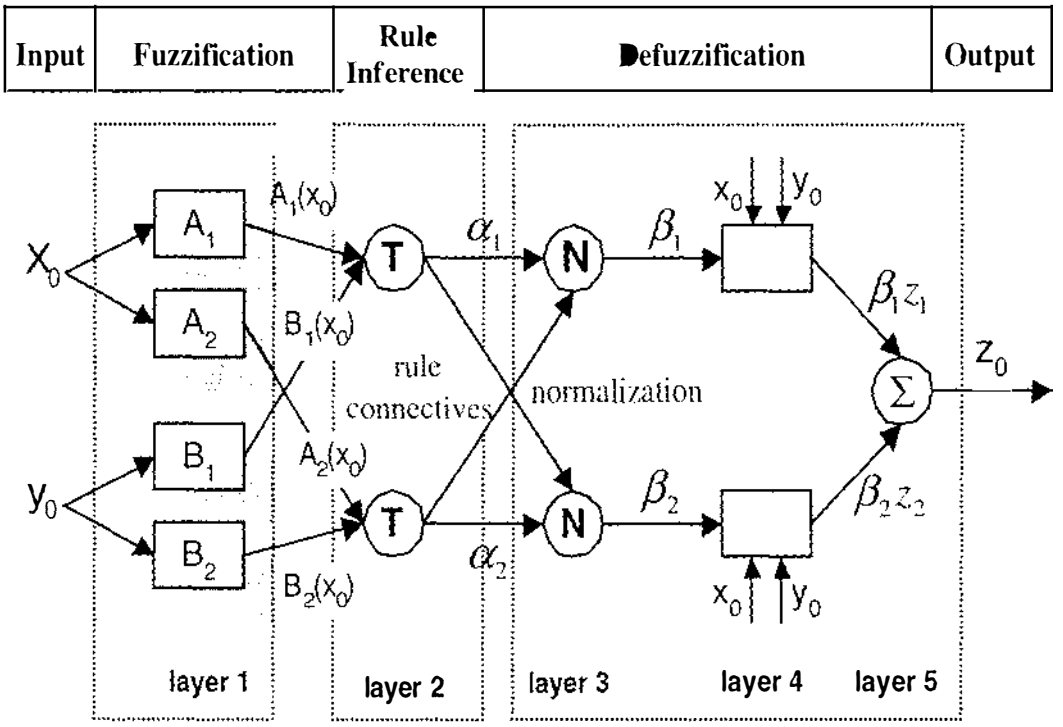


Figure 6.10. The architecture of the ANFIS neurofuzzy model.

- **Layer 1**. *Fuzzification*

  This layer consists of linguistic variables. Each neuron in this layer represents an input membership function of the antecedent of a fuzzy rule. The crisp inputs x_0 and y_0 are fuzzified by using membership functions of the linguistic variables A_1 and A_2. It is very important that node functions have to be differentiable, and we choose a bell-shaped membership function

$$A_i(u) = \exp\left[-\frac{1}{2}\left(\frac{u-a_{i1}}{b_{i1}}\right)^2\right]$$
$$B_i(u) = \exp\left[-\frac{1}{2}\left(\frac{u-a_{i2}}{b_{i2}}\right)^2\right] \qquad (6.12)$$

to represent the linguistic terms, where, $\{a_{i1}, a_{i2}, b_{i1}, b_{i2}\}$ is the parameter set. As the values of these parameters change, the bell-shaped functions vary accordingly, thus exhibiting various forms of membership functions on linguistic labels A_i and B_i.

- **Layer 2**. *Rule nodes*

  Second layer contains one node per each fuzzy IF-THEN rule. Each rule node performs connective operation between rule antecedents (IF-part). In other words, each node computes the firing strength of the associated rule. Usually, the **min** or the **product** (see Table 1.) is used as intersection **and**. The union **or** is usually done **max** operation. The output of the top node is

$$\alpha_1 = A_1(x_0) \times B_1(y_0) = A_1(x_0) \wedge B_1(y_0) \tag{6.13}$$

  and the output of the bottom node is

$$\alpha_2 = A_2(x_0) \times B_2(y_0) = \alpha_2 = A_2(x_0) \wedge B_2(y_0) \tag{6.14}$$

  Both nodes in this layer are labelled by T, because we can choose other T-norms for modelling the logical **and** operator. The nodes of this layer are called *rule nodes.*

- **Layer 3**. *Normalization*

  Every node in this layer is labelled by N to indicate the normalization of the firing levels. The output of top neuron is the normalized (with respect to the sum of the firing levels) firing level of the first rule

$$\beta_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2} \tag{6.15}$$

and the output of the bottom node is the normalised firing level of the second rule

$$\beta_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2} \tag{6.16}$$

- **Layer 4**. *Consequence layer*

  This layer gives the consequent part of the rule. In this study the output of top node is the product of the normalised firing level and the individual rule output of the first rule

$$\beta_1 z_1 = \beta_1 (a_1 x_0 + b_1 y_0) \tag{6.16}$$

  The output of top node is the product of the normalised firing level and the individual rule output of the first rule

$$\beta_2 z_2 = \beta_2 (a_2 x_0 + b_2 y_0) \tag{6.17}$$

- **Layer 5**. *Summation*

  Single node in this layer computes the overall system output as the sum of all incoming signals:

$$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2} = \beta_1 z_1 + \beta_2 z_2 \tag{6.18}$$

6.3.4 Hybrid Learning Algorithms of the ANFIS

Generally the purpose of the learning in the neurofuzzy system can be summarised as (Takagi & Lee, 1992):

1. deciding number of fuzzy rules,
2. deciding shape of the membership functions,
3. deciding consequent parameters,
4. deciding number of the input variables,
5. deciding fuzzy reasoning method.

(1) and (2) correspond to deciding how to cover the input space. They are highly dependent on each other. (3) corresponds to determining the coefficients of the linear equation in the case of the Takagi & Sugeno's type model or determining the consequent part membership functions in the case of Mamdani model (Mamdani, 1974). Table 6.3 shows some of learning schemes used in several recently introduced neurofuzzy systems. Their training methods differ very much from each other and no comparison of methods have been presented.

The ANFIS architecture consists of two trainable parameter sets:

1. The antecedent membership function parameters.
2. The polynomial parameters, also called the consequent parameters.

Each ANFIS training epoch, using the hybrid learning rule, consists of two passes. The consequent parameters are obtained during the forward pass using a least-squares optimisation algorithm and the premise parameters are updated using a gradient descent algorithm. During the forward pass all node outputs are calculated up to layer 4. At layer 4 the consequent parameters are calculated using a least-squares regression method. Next, the outputs are calculated using the new consequent parameters and the error signals are propagated back through the layers to determine the premise parameter updates. The consequent parameters are usually solved for at each epoch during the training phase, because as the output of the last hidden layer changes due the backpropagation phase, the consequent parameters are no longer optimal. Since the singular value decomposition (SVD) is computationally intensive, it may be most efficient to perform it every few epochs versus every epoch.

Table 6.3. Learning schemes of neurofuzzy system.

Neurofuzzy system	Premise learning	Consequent Learning	Adding of new fuzzy sets or fuzzy rules during learning
Horikawa et al. (1992)	gradient descent		No
Nack & Kruse (1993)	fuzzy gradient descent		No
Jang (1993)	gradient descent	least-square method	No
Lin & Lee (1991)	SOM, initial learning	gradient descent	Yes
Nie & Linkens (1993)	modified SOM	gradient descent	Yes
Berenji & Khedhar (1993)	gradient descent		No
Wang & Mendel (1992)	orthogonal least squares method (OLS)		OLS

6.4 INTELLIGENT HYBRID MODELLING FOR A TPFBBR

6.4.1 Reaction-Diffusion Model

The concentration profiles of substrate describing the simultaneous transport and removal within the biofilm are represented by the following equation

$$\frac{\partial C_s^f}{\partial t} = D_{sf}\left[\frac{\partial^2 C_s^f}{\partial L^2} + \frac{2}{L}\left(\frac{\partial C_s^f}{\partial L}\right)\right] - \frac{1}{Y_{x/s}}\frac{u_{\max} C_s^f}{K_s + C_s^f} X_a^f(L_f, t) \qquad (6.19)$$

where the boundary conditions of Eq. (6.19) are:

$$\begin{aligned} &\frac{\partial C_s^f}{\partial L} = 0 \qquad \text{at } L = 0 \\ &D_{sf}\frac{\partial C_s^f}{\partial L} = k_{ls}(C_s^l - C_s^f) \quad \text{at } L = L_f(t) \\ &C_s^f(L_f, 0) = 0 \quad \text{at } t = 0 \end{aligned} \qquad (6.20)$$

6.4.2 Axial-Dispersion Model

In three-phase fluidised-beds, the gas phase may often be assumed to be plug flow. Appreciable backmixing, however, may occur in the liquid phase, especially for beds of small particles in concurrent three-phase fluidization. The backmixing of the liquid and solid particles in such a bed is primarily caused by the rising motion of coalesced large gas bubbles. An axial dispersion model has been most commonly used to describe the backmixing behaviour of the liquid phase and to simulate substrate removal in a TPFBBR (Wisecarver & Fan, 1987; Petersen & Davison, 1995). A mass balance for substrate in the liquid phase yields the following equation

$$\frac{\partial C_s^l(Z,t)}{\partial t} = E_L \frac{\partial^2 C_s^l}{\partial Z^2} - \frac{U_L}{\varepsilon_l} \cdot \frac{\partial C_s^l}{\partial Z} - \frac{k_{ls} a_p \varepsilon_s}{\varepsilon_l} (C_s^l - C_s^s) \tag{6.21}$$

where corresponding boundary and initial conditions for above equation are

$$\begin{aligned} U_L C_s^l(0,t) - \varepsilon_l E_L \frac{\partial C_s^l(0,t)}{\partial Z} &= U_L C_s^{in} \quad \text{at } Z = 0 \\ \frac{\partial C_s^l(H,t)}{\partial Z} &= 0 \quad \text{at } Z = H \\ C_s^l(Z,0) &= C_{s0}^l \quad \text{at } t = 0 \end{aligned} \tag{6.22}$$

In order to simulate the performance of a TPFBBR using the axial-dispersion model, estimation of the phase holdups, gas-liquid and liquid-solid mass transfer coefficient, the liquid phase axial dispersion coefficient and the expanded bed height under various are required.

In the axial-dispersion model, the axial-dispersion coefficient (E_L) describing the backmixing behaviour of the liquid phase is very important and several correlations for E_L have been suggested. To predict E_L, the correlation of Kim et al. (1992) is used since their correlation covers a wide range of literature data and can be applied to a TPFBBR with small, low-density particles.

$$Pe_L = \frac{d_{bp} U_L}{E_L} = 20.19 \left(\frac{d_{bp}}{D_c} \right)^{1.66} \left(\frac{U_L}{U_L + U_G} \right)^{1.03} \tag{6.23}$$

Recently, Nore et al. (1992) studied hydrodynamics, gas-solid and liquid-solid mass transfer in a TPFBBR with particle densities ranging from 1300 to 1700 kg/m^3. In their studies, increasing the gas velocity increased k_{ls}, especially for low particle densities and the liquid velocity had almost no effect on liquid-solid mass transfer

coefficient. Nore et al. (1992) estimated k_{ls} satisfactorily using the following correlations

$$k_{ls} = 1.10\left(\frac{U_L}{\varepsilon_L}\right)^{0.43} d_{bp}^{0.24} \qquad \text{at } U_G = 0 \tag{6.24a}$$

$$k_{ls} = 0.197\left(\frac{U_L}{\varepsilon_L}\right)^{0.051} d_{bp}^{0.17} \qquad \text{at } U_G \neq 0 \tag{6.24b}$$

In order to estimate the phase holdups in this study, the purely empirical correlations suggested by Begovich & Watson (1978a) are chosen because of the large data base from a wide variety of particles which they used for their correlation.
The gas and solid holdup are estimated from the following correlations, respectively.

$$\varepsilon_g = (0.048 \pm 0.010) U_L^{0.72 \pm 0.028} d_p^{0.168 \pm 0.061} D_c^{-0.125 \pm 0.088} \tag{6.25}$$

$$\varepsilon_s = 1 - 0.371 U_L^{0.271} U_G^{0.041} (\rho_{bp} - \rho_L)^{-0.316} d_{bp}^{-0.268} \mu_L^{0.055} D_c^{-0.033} \tag{6.26}$$

The liquid holdup and bed porosity can also be calculated from

$$\begin{aligned} \varepsilon_l &= 1 - \varepsilon_g - \varepsilon_s \\ \varepsilon &= 1 - \varepsilon_s \end{aligned} \tag{6.27a,b}$$

The solids holdup influences the biofilm specific surface area (a_p), which can be estimated by

$$a_p = \frac{6\varepsilon_s}{d_{bp}} \tag{6.28}$$

6.4.3 Intelligent Hybrid Model for a TPFBBR

Generally, a hybrid model, as shown in Fig. 6.11, may be divided into two categories: (1) *serial hybrid model* (2) *parallel hybrid model.*

In serial hybrid models (Psichogious & Ungar, 1992; Tompson & Kramer, 1994; Schubert et al., 1994), the empirical model such as ANN and neurofuzzy model is placed in series with a mechanistic model, as illustrated in Fig. 6.11(a). Psichogious & Ungar (1992) and Schubert et al. (1994) proposed a serial approach in modelling a fermentation process. The neural network component of the hybrid model estimated the biomass concentration and the specific growth rate, which was input into the component mass balances.

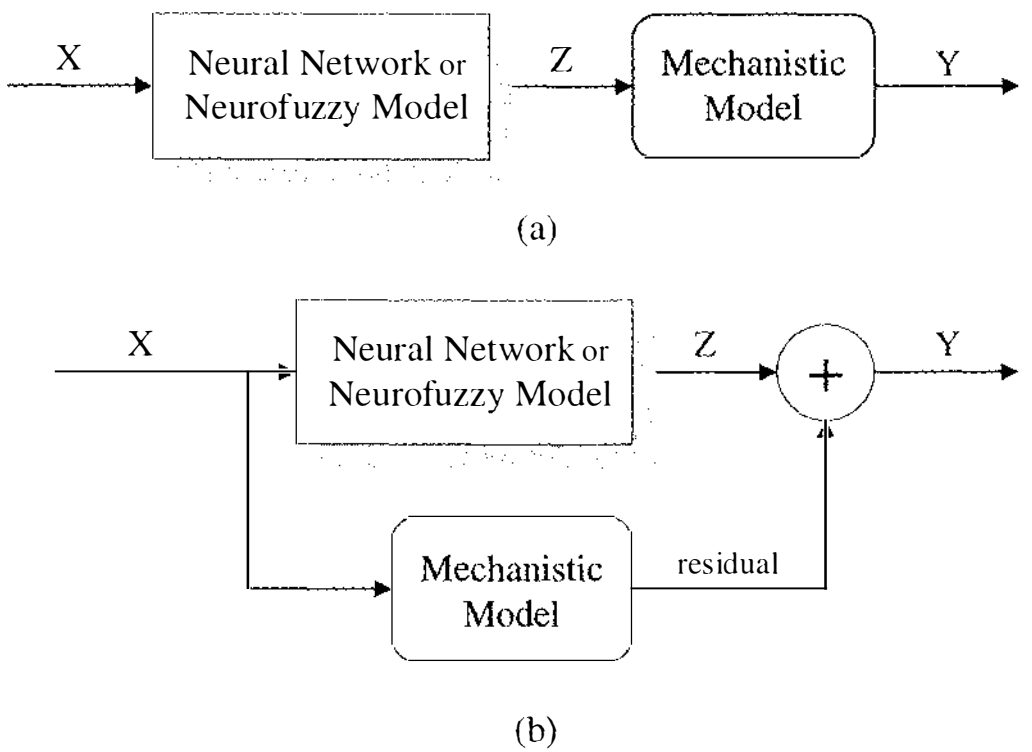


Figure 6.11. Type of the hybrid model; (a) Serial hybrid model (b) Parallel hybrid model (Tompson & Kramer, 1994).

In parallel hybrid models (Cote et al., 1995; Zhao et al., 1997), as illustrated in Fig. 6.11(b), the ANN or neurofuzzy model is placed in parallel with a mechanistic model. The ANN or neurofuzzy model is in fact an error model, which should model the difference between the output of a mechanistic model and real output of the process. In a parallel hybrid approach, the output of the ANN (or neurofuzzy model) and mechanistic model are combined to determine the total model output. The ANN or neurofuzzy component of the hybrid model is trained on the residual between the process data and the mechanistic model to compensate for any uncertainties that arise from the inherent process complexity (Tompson & Kramer, 1994).

Cote et al. (1995) showed the parallel approach on real-time data of activate sludge process. Zhao et al. (1997) demonstrated a hybrid model, which consists of a simplified process model and an ANN, for developing a dynamic model of a sequencing batch reactor (SBR). In their hybrid model, the outputs of the trained ANN compensated for the output errors of the simplified process model. The hybrid model output of the final predictions of the process states was obtained by summing of the outputs from the simplified process model and ANN. In these two cases, the hybrid model showed better interpolation than the black box ANN model.

In general, the biofilm growth mechanism in a TPFBBR is quite complex and often involves nonlinear expressions such as Monod or Haldane kinetics. The development of reliable biofilm growth model based on mechanistic model is extremely difficult and the application of hybrid model offers a logical alternative. The plot of the implementation of the serial hybrid model is given in Fig. 6.12. In our case, the hybrid model consists of two parts including a neurofuzzy model, which serves as a *process estimator* of difficult-to-model process variables (such as biofilm thickness and biofilm density), and a mechanistic model, which represents a mechanistic knowledge of process system by using differential equation.

The neurofuzzy model component of the hybrid model serves as a *process estimator* and receives as inputs certain measured variables and provides a prediction of the

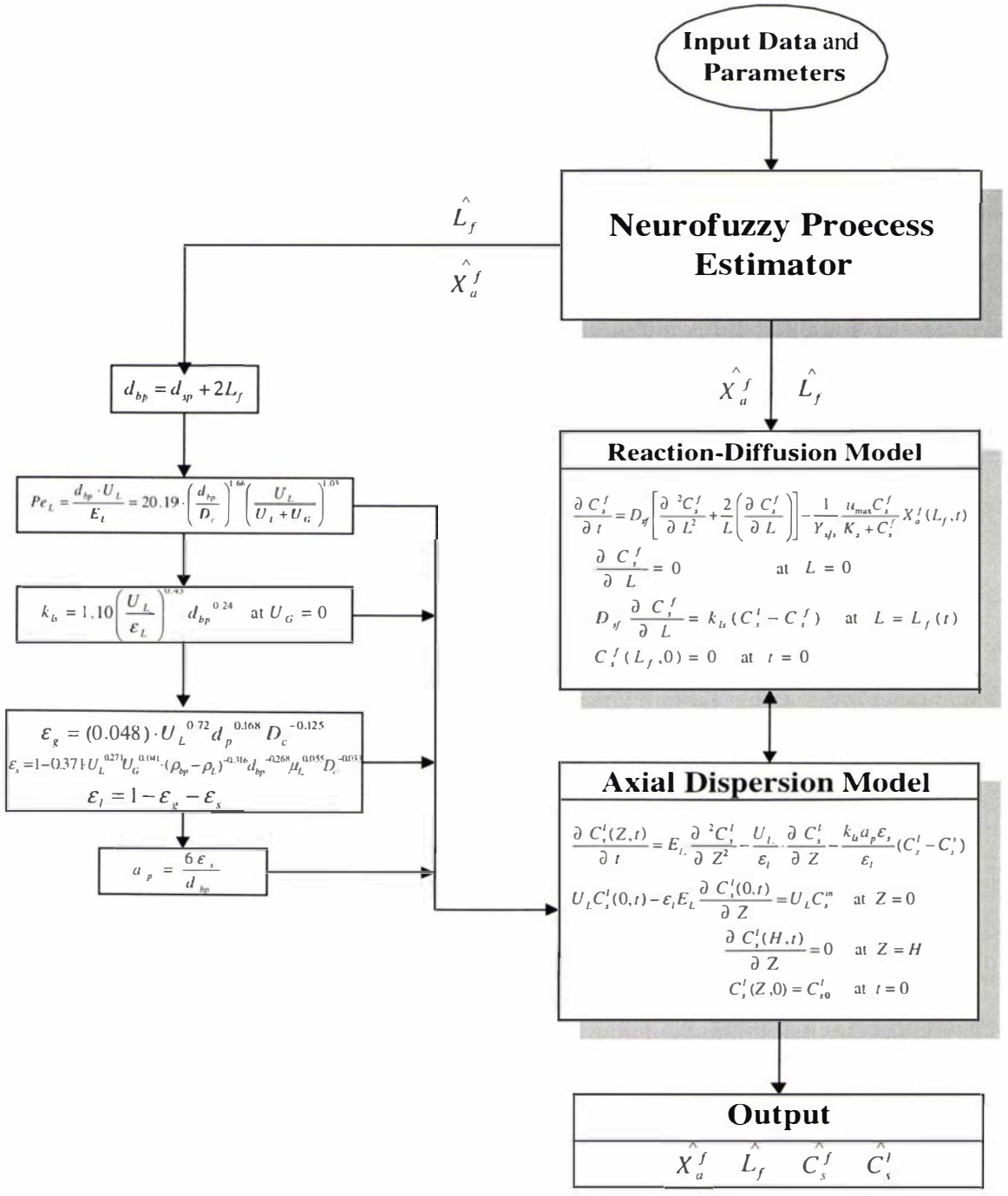


Figure 6.12. The structure of hybrid model presented for a TPFBBR ($\hat{L}_f$ = estimated biofilm thickness, $\hat{X}_a^f$ = estimated biofilm density).

one-step-ahead biofilm thickness and biofilm density which are difficult to measure. The main role of a neurofuzzy process estimator is that it can serve as good numerical estimators and at the same time can provide qualitative information about the complex biofilm growth which is difficult-to-model and difficult-to-measure. The outputs of neurofuzzy model serves as parameters for reaction-diffusion model and axial dispersion model, which produces the substrate concentration profiles within the biofilm phase and the substrate concentration profiles in the liquid phase as a function of bed height at the different times respectively. The combination of these different modelling approaches yields complete intelligent hybrid model for a TPFBBR.

The resulting hybrid model can be considered a structured neurofuzzy model which contains some known parts, in this case reaction-diffusion model and axial-dispersion model. Alternatively, There may be thought of as mechanistic models which contain process parameters whose dependence on process variables is modelled by a neurofuzzy model. This hybrid model has the advantage of the short development time of data-driven empirical model with extrapolation properties of knowledge-driven mechanistic model since they require less data than when a empirical model is used alone. Also less a prior knowledge about a complex biofilm growth mechanism is required than when a mechanistic model is used alone.

6.5 RESULTS AND DISCUSSION

6.5.1 Results of Neurofuzzy Estimator

Biofilm Thickness Estimation

A total of 41 data sets were compiled from a TPFBBR described in chapter 3. The data was split into two sets: (1) a training set including 70% of the data, and (2) a test set including the remaining 30%.

The data for biofilm thickness estimation are written in {Y(k), Y(k-1), Y(k-2), Y(k-3), Y(k-4), U1(k-1), U1(k-2), U2(k-1), U2(k-2), U3(k-1), U4(k-2), k=1, 41}. Output Y(k) represents the one-step ahead biofilm thickness. The current and past biofilm thickness (Y(k-1), Y(k-2), Y(k-3), Y(k-4)), and the current and past biofilm density (U1(k-1), U1(k-2)) are used as inputs to the neurofuzzy estimator. The current and past suspended biomass concentration (U2(k-1), U2(k-2)), and the current and past inlet substrate concentration (U3(k-1), U3(k-2)) are also used as inputs. Neurofuzzy process estimator has 1 output and 10 inputs for the estimation of biofilm thickness, as shown Fig. 6.13.

When the neurofuzzy model (or fuzzy model) is applied to a specific problem, the process of identifying a model is necessary. The identification process of a neurofuzzy model is generally divided into structure identification and parameter identification. The former means the combination of the input variables and the number of the membership functions in the premises and consequences. The latter identifies the parameters in both the premises and consequences. The characteristics of a neurofuzzy (or fuzzy) model depend heavily on the structures rather than on the parameters of the membership functions (Horikawa et al., 1992). So the selection of the structures is only done once in the process. There exists no a general rule of thumb for finding the best structure for the neurofuzzy model and determining the No. of the MF given in the input-output data. They are chosen empirically by trial and error method.

In order to eliminate the trial and error process for finding the best structure of the neurofuzzy model, the following heuristic searching algorithm is implemented in the neurofuzzy computer program:

Step 1: Determine the number of membership function (MF) being assigned to each input variables. Let us start with MF=2.

Step 2: Search possible input combinations from the considerable input variables (in our case, 10 variables). Let us start to search for 2 input combination.

Step 3: For each combination, the neurofuzzy model is tested with the data set until one epoch. Check the performance of model based on the RMSE (Eq. (5.32)) for each case.

Step 4: Then find a neurofuzzy model with the least RMSE value and save it.

Step 5: Increase the number of input variable combination. Repeat step 2 to step 4 until finishing 4 input variable combination.

Step 6: Change the number of MF. Repeat step 2 to step 5.

Step 7: The search is stopped if the number of MF reaches at 4.

Step 8: Finally comparing the each case saved in step 4, find the best structure of the neurofuzzy model, and run it.

The outline of the heuristic searching algorithm is shown in Fig. 6.13. Figure 6.14(a)-(b) represent some of result plots from the heuristic searching algorithm during the process of structure identification when MF=3. The some results of the process of structure identification using the above algorithms are shown Table 6.4. From Table 6.4, the number of membership functions assigned to each input of the neurofuzzy model for the estimation of the biofilm thickness (data set 1) was changed from 2 to 5.

In Table 6.4, the structures, which have marked (#), are selected as a best structure in each number of MF. The structure marked (*) is chosen as the best structure over all possible structures. Table 6.4 shows that the best structure of neurofuzzy model over biofilm thickness (data set 1) has Y(k-2) Y(k-4) U2(k-1) U3(k-1) when MF=3. Because the number of MF assigned to each 4 input variable is 3, so the rule number of is 81. The neurofuzzy model used here contains a total of 441 fitting parameters, of which 36 premise parameters and 405 are consequent parameters.

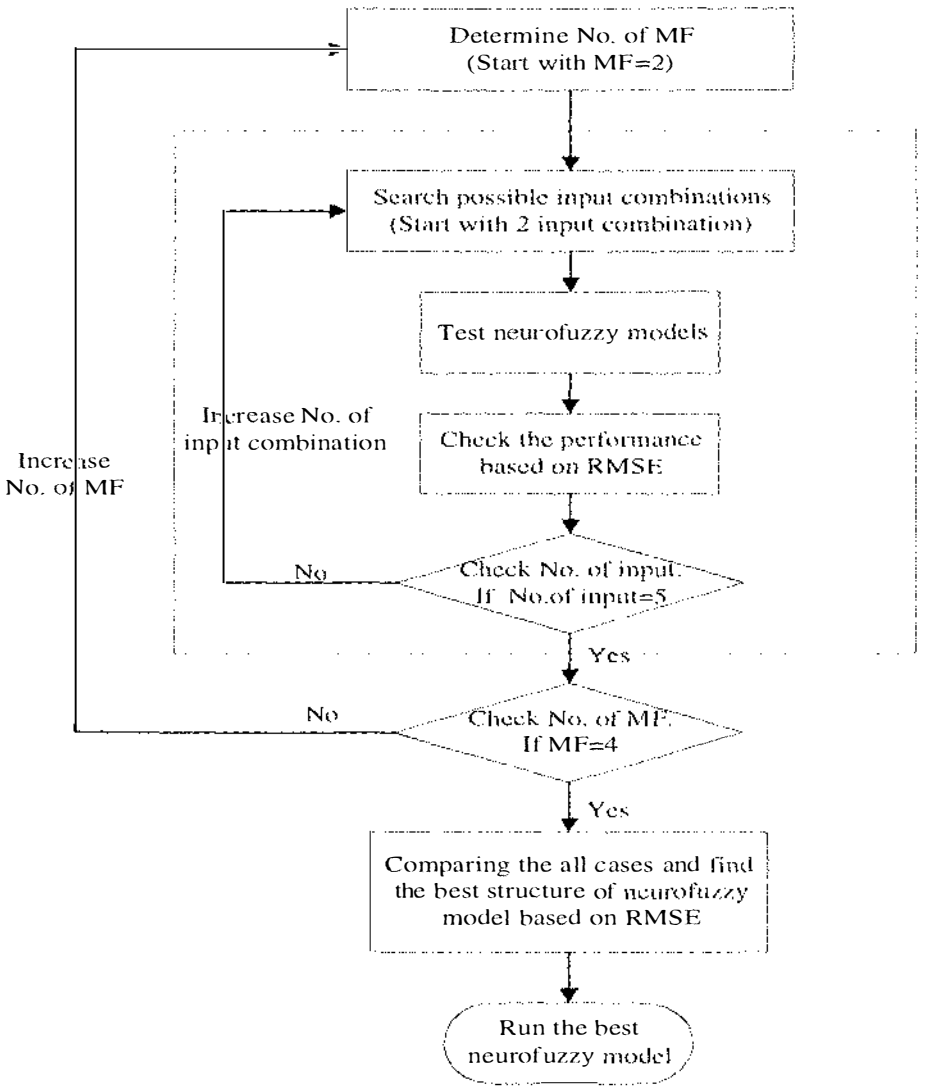
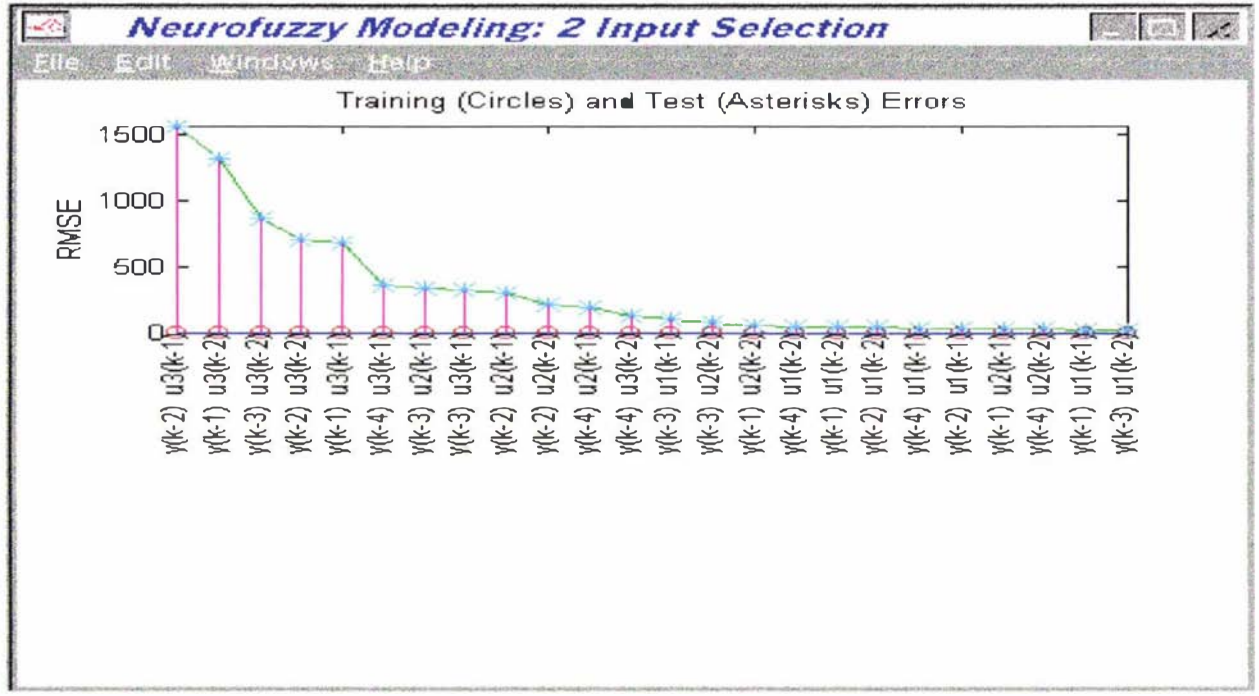
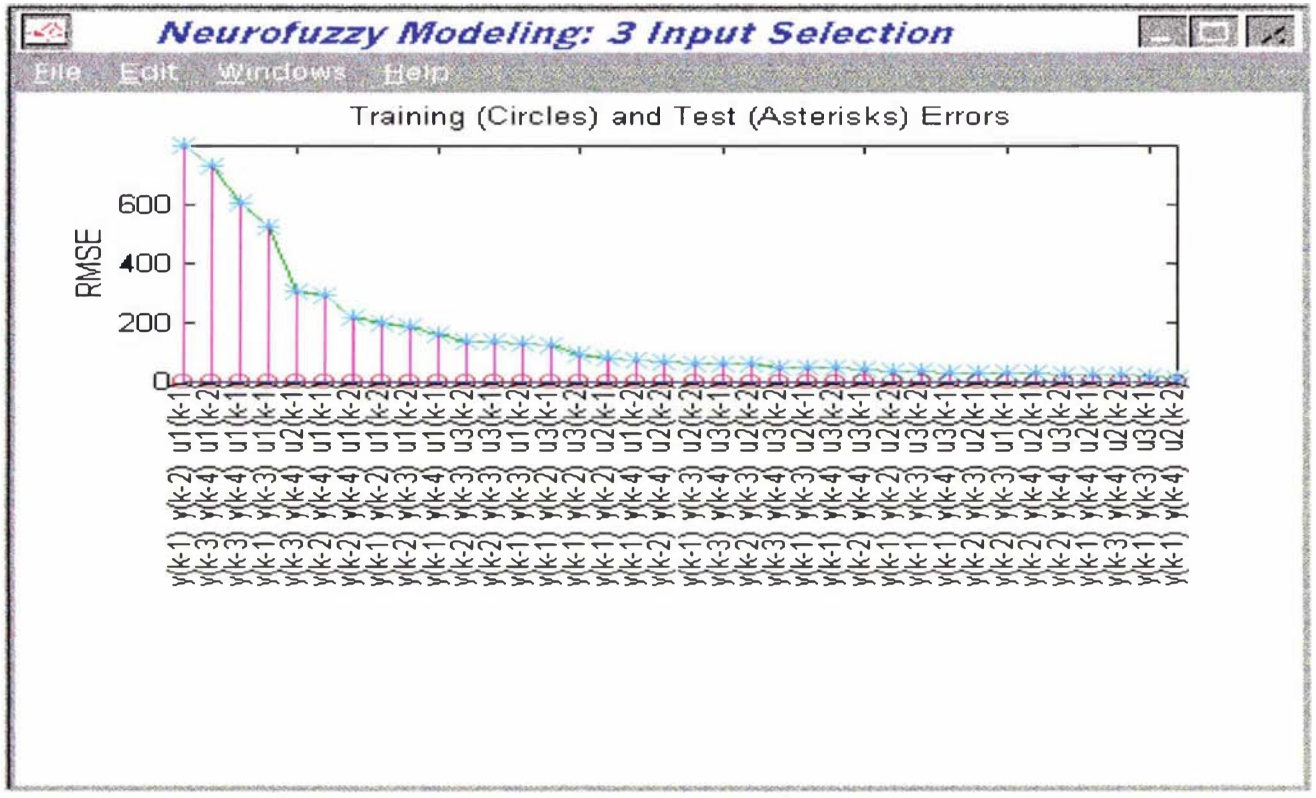


Figure 6.13. Outline of the heuristic searching algorithm for neurofuzzy system.



(a)

Figure 6.14(a). The result plots of the heuristic searching algorithm for structure indetification for biofilm thickness using data set 1(when MF=3): 2 input selection case.



(b)

Figure 6.14(b). The result plots of the heuristic searching algorithm for structure indetification for biofilm thickness using data set 1(when MF=3): 3 input selection cases.

Table 6.4. Identification results for structure of neurofuzzy models using the heuristic searching algorithm over biofilm thickness (data set 1).

No. of MF	No. of Input	Inputs Selected	RMSE
2	2	y(k-2) u1(k-2)	8.9811
		y(k-3) u1(k-1)	9.0401
		y(k-1) u2(k-2)	9.1169
		y(k-1) u2(k-1)	10.3131
		y(k-3) u1(k-2)	11.2706
	3	y(k-1) y(k-4) u1(k-1)(#)	7.5124
		y(k-2) y(k-3) u1(k-1)	8.4514
		y(k-1) y(k-2) u1(k-2)	11.4147
		y(k-3) y(k-4) u1(k-2)	18.7352
3	2	y(k-3) u1(k-2	11.6448
		y(k-1) u1(k-1)	13.4890
		y(k-4) u2(k-2)	17.6994
		y(k-1) u2(k-1)	18.0470
	3	y(k-1) y(k-4) u2(k-2)	6.9856
		y(k-1) y(k-3) u3(k-1)	8.5606
		y(k-3) y(k-4) u2(k-2)	12.3625
		y(k-1) y(k-4) u2(k-1)	12.5877
		y(k-2) y(k-4) u2(k-1)	14.4230
	4	y(k-2) y(k-4) u2(k-1) u3(k-1) (*)	4.6048
		y(k-2) y(k-4) u1(k-1) u2(k-2)	5.9731
		y(k-2) y(k-4) u1(k-2) u2(k-1)	6.4043
		y(k-1) y(k-4) u2(k-1) u3(k-1)	6.7329
		y(k-2) y(k-3) u1(k-2) u2(k-1)	6.7858
4	3	y(k-2) y(k-4) u3(k-1) (#)	6.0912
		y(k-1) y(k-3) u3(k-1)	6.6459
		y(k-1) y(k-2) u2(k-1)	9.3280
		y(k-1) y(k-2) u3(k-2)	11.8080
5	3	y(k-1) y(k-4) u2(k-2) (#)	7.2390
		y(k-1) y(k-2) u3(k-1)	8.3891
		y(k-2) y(k-4) u3(k-1)	8.2223
		y(k-1) y(k-3) u3(k-1)	11.2386

Fig. 6.15 represents the initial membership functions for each variable before learning, and Fig. 6.16 shows the membership functions for each variable after learning. After reaching 156 epochs, we have had RMSE for training = 3.961 and RMSE for testing = 4.604.

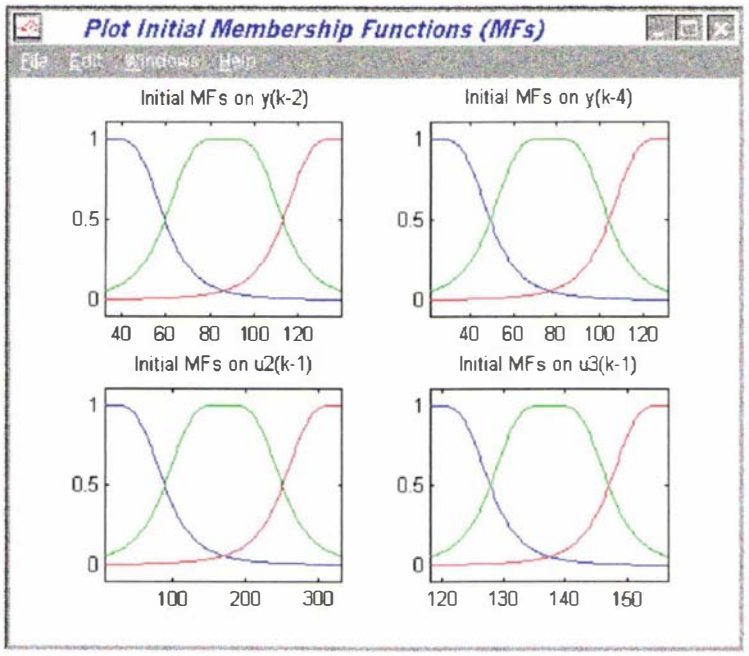


Figure 6.15. Initial membership functions for each variable (data set 1).

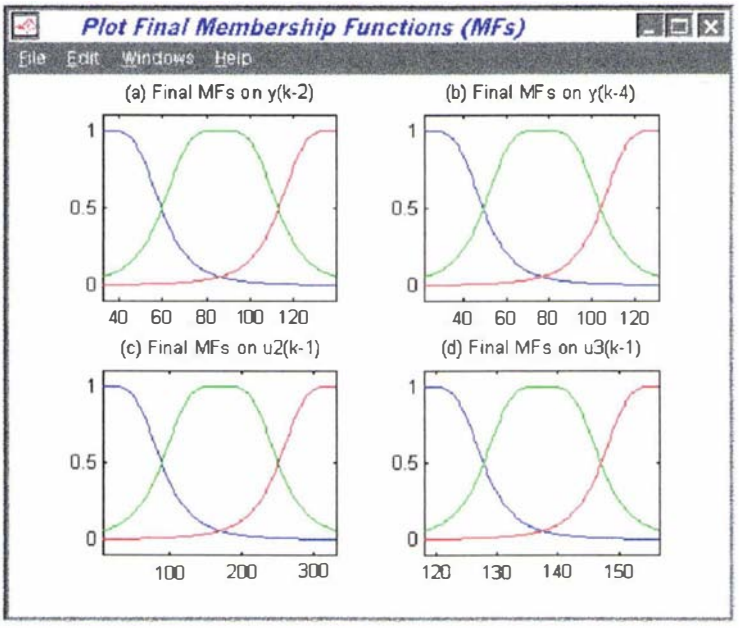


Figure 6.16. Membership functions after learning for each variable (data set 1).

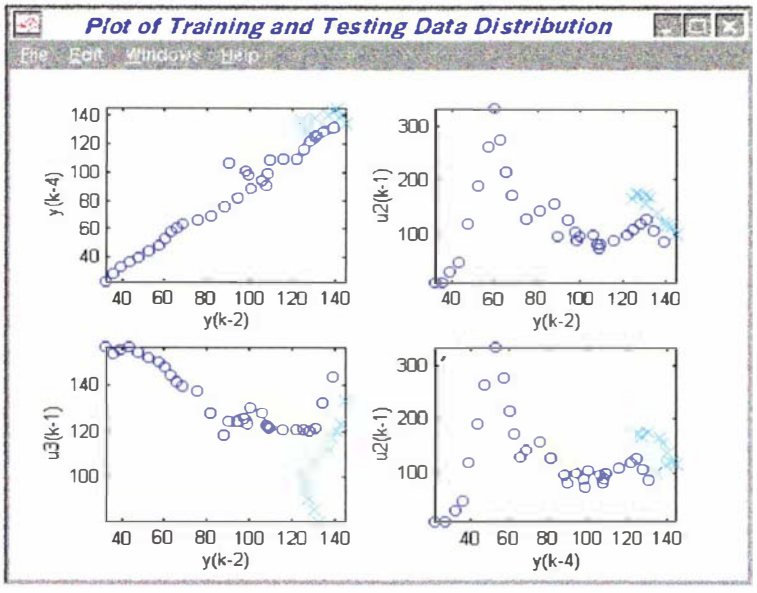
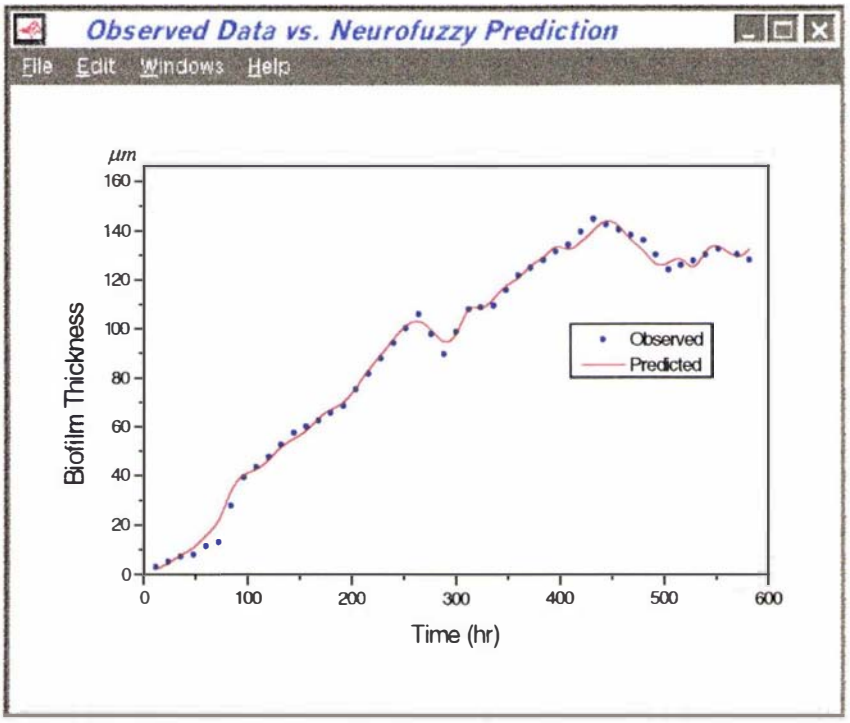
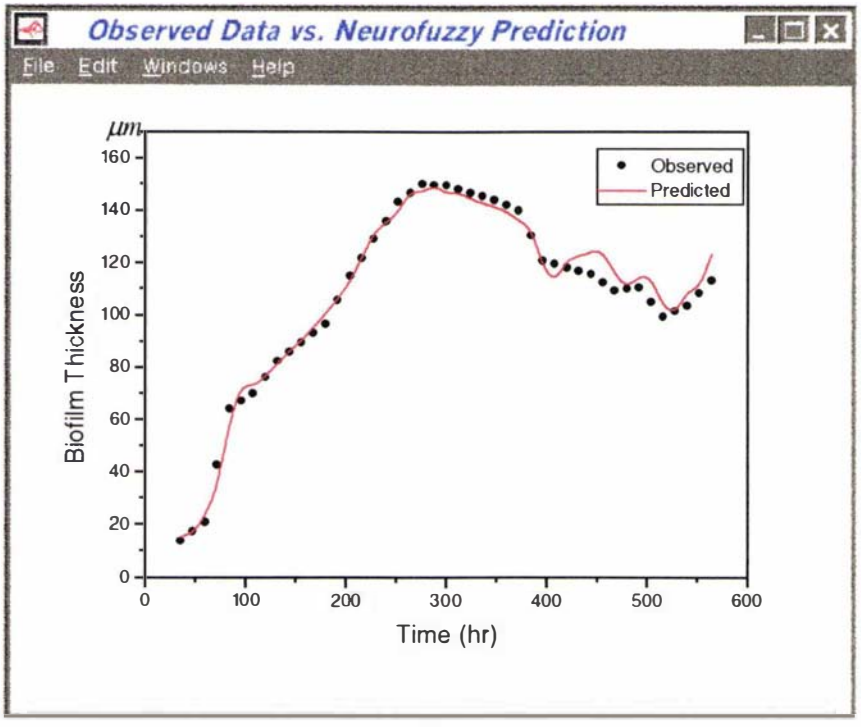


Figure 6.17. Plots of training and testing data distriubution on each input variable (data set1).



(a)



(b)

Figure 6.18. Neurofuzzy model prediction of biofilm thickness with observed value: (a) data set 1 (b) data set 2.

Fig. 6.18(a)-(b) shows the difference between the predicted value by a neurofuzzy model and the observed value, and demonstrates the ability of a neurofuzzy estimator to fit the rapidly growing biofilm thickness during the startup phase of a TPFBBR. Generally, from Fig. 6.18(a), lag phase (within 50 hours after startup) of the growth of biofilm on the support particles after startup of the TPFBBR is observed. This lag time could be due to adsorption of dissolved organics at the surface of the support particle, and the attachment of initial biofilm on the support particle. In this stage, a small and uniform biofilm of detectable thickness could be observed on the edges of support particles and the biofilm grows as dense, thin, uniformly distributed biomass matrix which lacks the voids present in the thick biofilm. After the lag phase, the biofilm thickness increased rapidly and then levelled off at approximately 135 μm from 600 hour. From the Fig. 6.18, the biofilm formation in a TPFBBR is slow since the 60-90% of biomass was detached continuously from the support particles (Tijhuis, et al., 1994). It is demonstrated that the neurofuzzy estimator produces the good simulation results and correctly identified even small change behavior in the biofilm thickness during the lag phase.

Biofilm Density Estimation

The structure identification for estimating the biofilm density model was done in the same manner as those used in the biofilm thickness prediction model. The data are written in {Y(k), Y(k-1), Y(k-2), Y(k-3), Y(k-4), U1(k-1), U1(k-2), U2(k-1), U2(k-2), U3(k-1), U4(k-1), k=1, 41}. Like the estimation of biofilm thickness, neurofuzzy process estimator has 1 output and 10 inputs for the estimation of biofilm density.

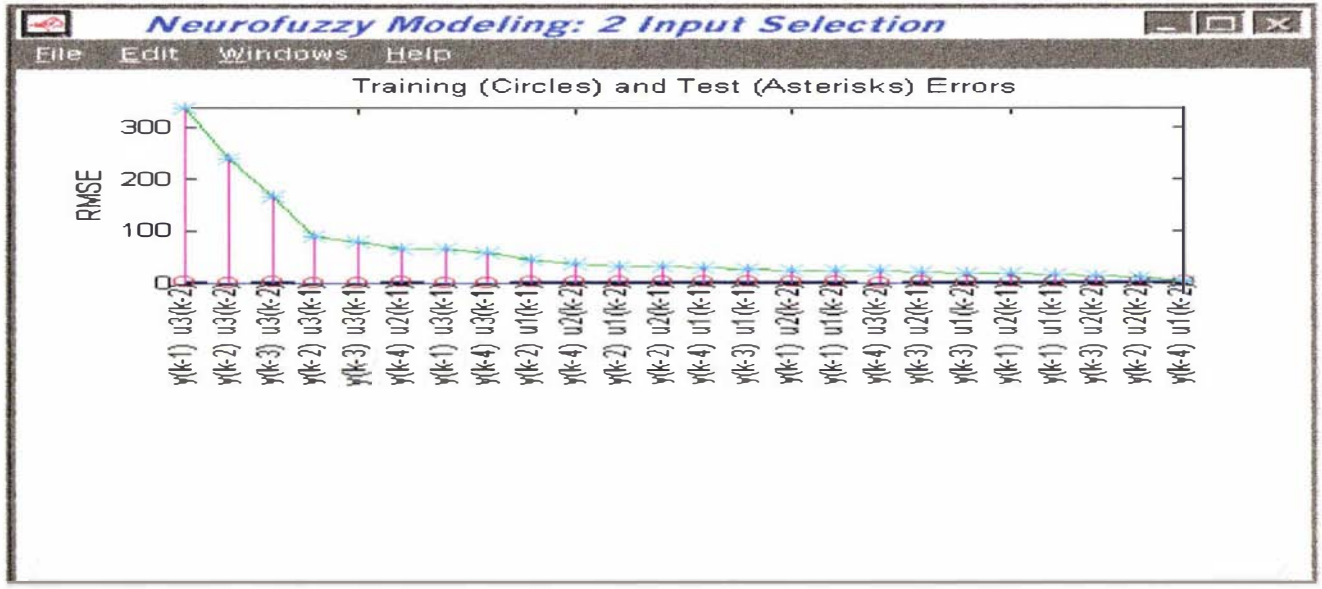
The same procedure for finding the best structure of neurofuzzy estimator as that adopted for the estimation of biofilm thickness was applied for the estimation of biofilm density. Figure 6.19(a)-(b) represent some result plots from the heuristic searching algorithm during the process of structure identification when MF=3. The some results extracted from the heuristic searching algorithm during the process of

structure identification are also shown Table 6.5. Table 6.5 shows that the best structure of neurofuzzy model over biofilm density (data set 1) has Y(k-1) Y(k-2) U2(k-2) U3(k-1). The neurofuzzy model used here contains 81 rules, with 3 MF being assigned to each 4 input variable and total number of fitting parameters is 441 which consist of 36 premise parameters and 405 consequent parameters. Fig. 6.20(a) represents the initial membership functions for each variable before learning, and Fig. 6.20(b) shows the membership functions for each variable after learning.

The plots in Fig. 6.22(a)-(b) show the comparisons between the observed data and neurofuzzy estimation of the biofilm density. Within 50 hours after startup (Fig. 6.22(a)), a small and uniform biofilm of detectable thickness could be observed on the edges of support particles. In the meantime, we observed that the increased biofilm thickness caused the increase of biofilm density to maximum value. This stage is not a dominant process during the formation of the biofilm. From 50-70 hours after startup, as the biofilm grows on the support particle, the biofilm density decreased in the increasing biofilm thickness and levelled off at approximately 12 mg/cm^3, corresponding to 135 μm of the biofilm thickness.

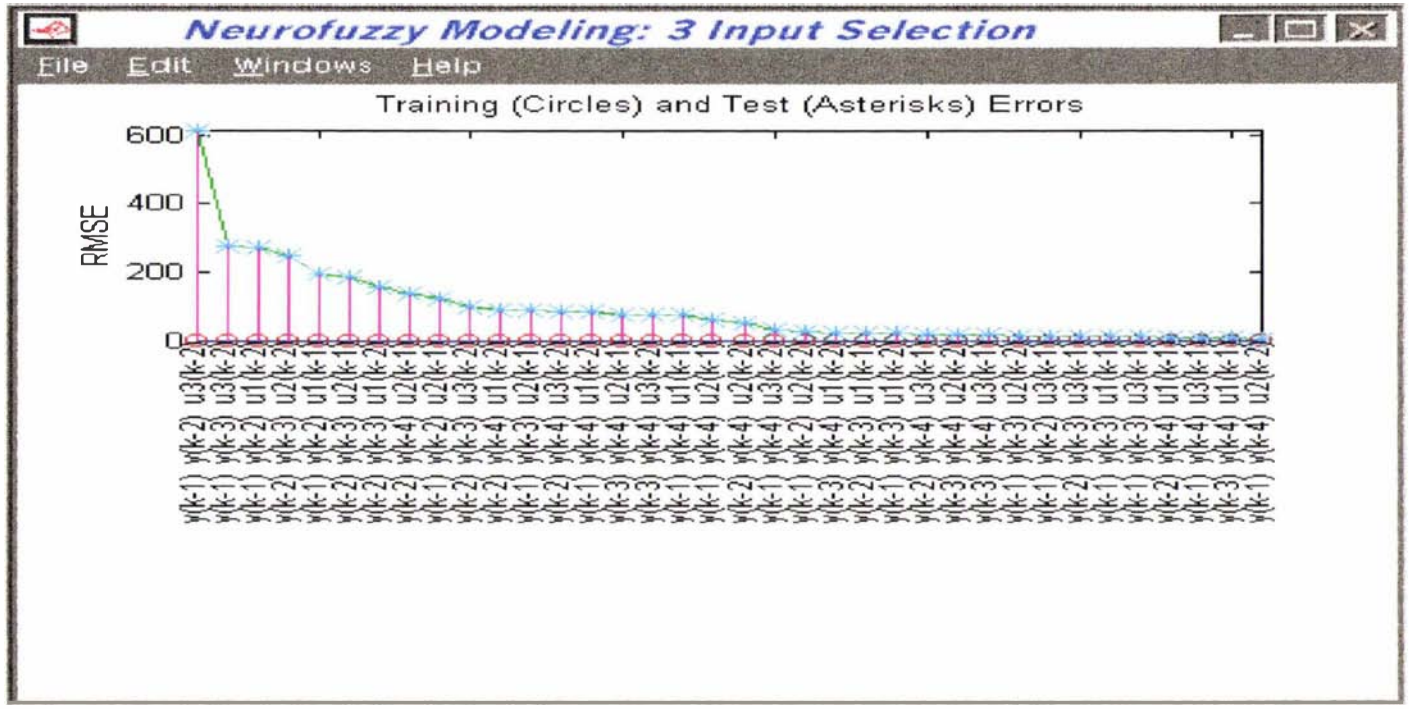
Fig. 6.23 shows 3-D surface response curve of the biofilm density with time and biofilm thickness. It is clear that there exists the inverse relationship between biofilm thickness and its density during the formation of biofilm growth.

It can be seen that the results predicted using neurofuzzy estimator, which accounts for varying biofilm thickness and its density in time, are in good agreement with the observed values. It is demonstrated that the neurofuzzy process estimator serves as a good numerical estimator and at the same time provides valuable information about the complex biofilm growth which is difficult-to-model and difficult-to-measure.



(a)

Figure 6.19(a). The result plots of the heuristic searching algorithm for structure indetification for biofilm density using data set 1(when MF=3): 2 input selection cases.

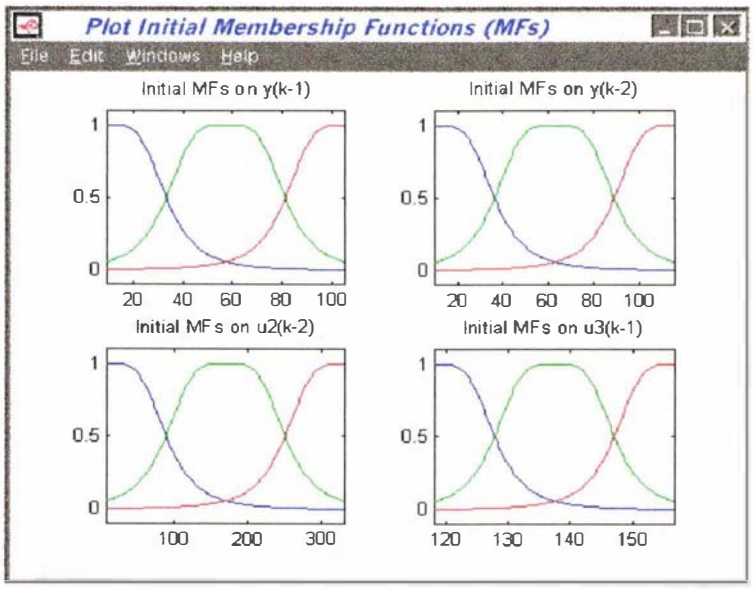


(b)

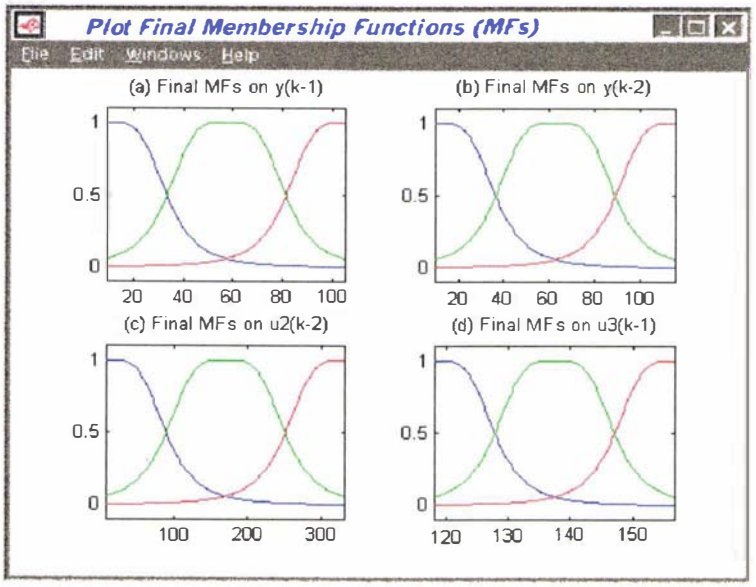
Figure 6.19(b). The result plots of the heuristic searching algorithm for structure indetification for biofilm density using data set 1(when MF=3): 3 input selection cases.

Table 6.5. Identification results for structure of neurofuzzy models using the heuristic searching algorithm over biofilm density (data set 1).

No. of MF	No. of input	Inputs Selected	RMSE
2	2	y(k-3) u1(k-1)	3.5247
		y(k-2) u2(k-2)	4.9354
		y(k-1) u3(k-2)	5.0637
		y(k-2) u1(k-1)	5.1189
		y(k-1) u2(k-1)	5.1495
	3	y(k-1) y(k-4) u1(k-2)	9.0538
		y(k-1) y(k-3) u2(k-1)	9.8118
		y(k-3) y(k-4) u1(k-1)	10.4218
		y(k-1) y(k-2) u3(k-2)	11.3029
		y(k-2) y(k-3) u1(k-1)	11.7064
	4	y(k-3) y(k-4) u1(k-1) u1(k-2) (#)	3.0513
		y(k-1) y(k-4) u1(k-2) u3(k-1)	3.7405
		y(k-1) y(k-3) u1(k-1) u3(k-1)	3.7977
		y(k-2) y(k-3) u1(k-1) u3(k-1)	4.1591
		y(k-1) y(k-3) u1(k-2) u3(k-1)	4.3565
3	2	y(k-4) u1(k-2)	3.7592
		y(k-2) u2(k-2)	9.4238
		y(k-3) u2(k-2)	14.3874
		y(k-1) u1(k-1)	15.0360
		y(k-1) u2(k-1)	17.7762
	3	y(k-1) y(k-4) u2(k-2)	2.3497
		y(k-3) y(k-4) u1(k-1)	3.9956
		y(k-1) y(k-4) u3(k-1)	4.5744
		y(k-2) y(k-4) u1(k-1)	4.8706
		y(k-1) y(k-3) u3(k-1)	6.1906
	4	y(k-1) y(k-2) u2(k-2) u3(k-1) (*)	1.9561
		y(k-2) y(k-4) u1(k-1) u2(k-2)	2.2748
		y(k-1) y(k-4) u1(k-2) u2(k-1)	2.3598
		y(k-1) y(k-3) u2(k-2) u3(k-1)	2.3815
		y(k-1) y(k-4) u2(k-1) u3(k-1)	2.3932
4	2	y(k-1) u2(k-2)	16.1288
		y(k-2) u1(k-1)	18.7777
		y(k-3) u1(k-2)	38.6724
		y(k-3) u1(k-1)	42.6744
		y(k-4) u1(k-1)	65.3808
	3	y(k-1) y(k-4) u3(k-1) (#)	2.2467
		y(k-2) y(k-4) u3(k-1)	2.9422
		y(k-1) y(k-4) u2(k-2)	3.1355
		y(k-3) y(k-4) u1(k-1)	3.7554
		y(k-1) y(k-4) u1(k-1)	3.8833



(a)



(b)

Figure 6.20. Membership functions on each variable (a) before learning (b) after learning (data set 1).

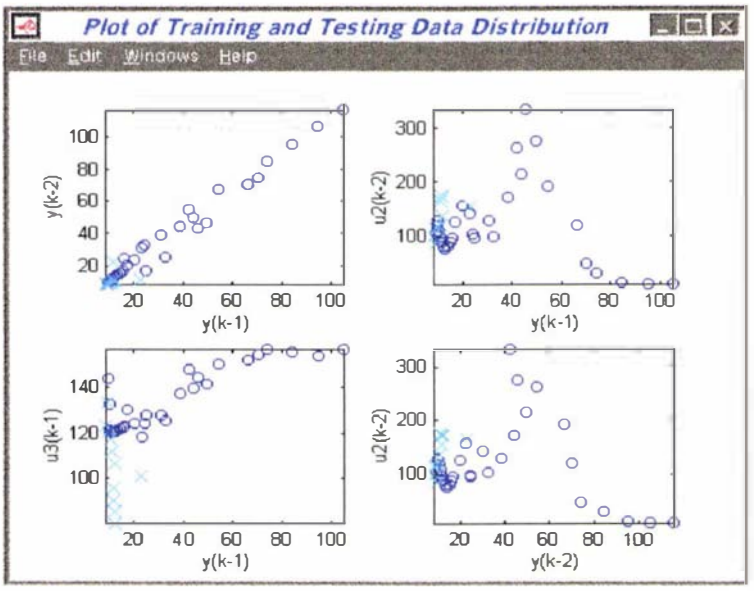
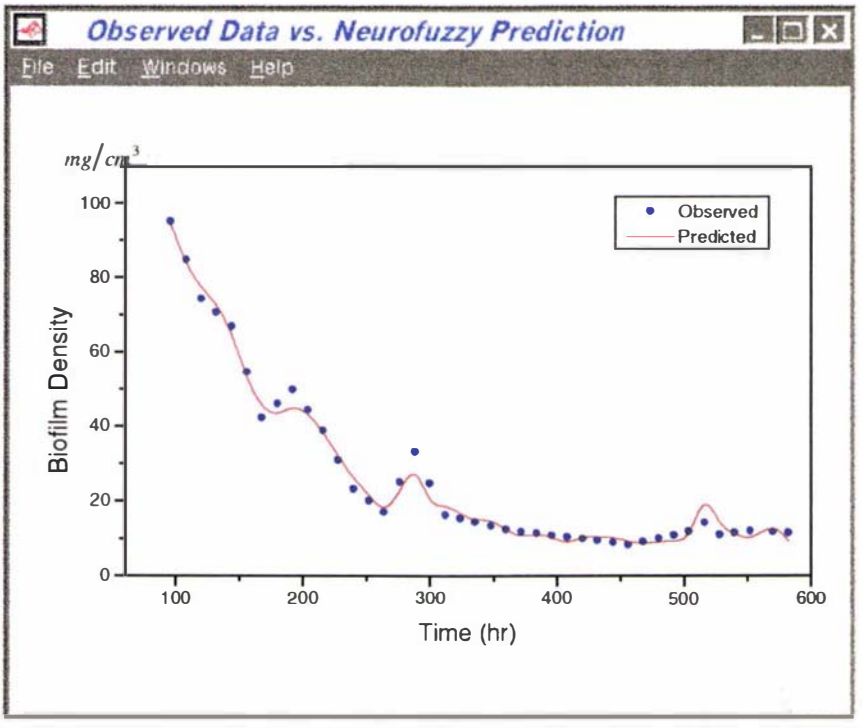
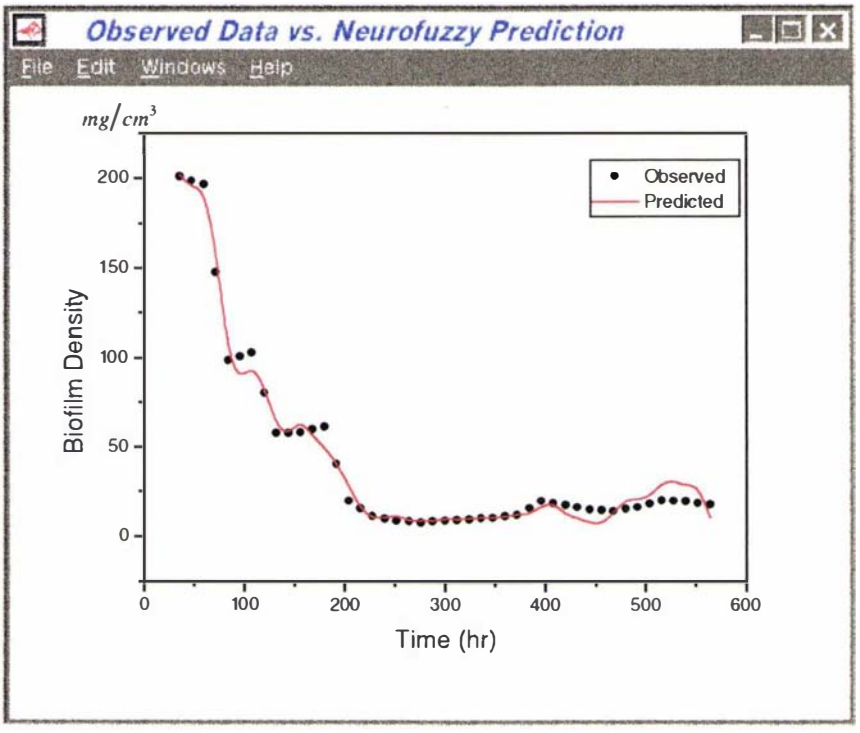


Figure 6.21. Plots of training and testing data distriubution on each input variable (data set1).



(a)



(b)

Figure 6.22. Neurofuzzy model prediction of biofilm density with observed value: (a) data set1 (b) data set 2.

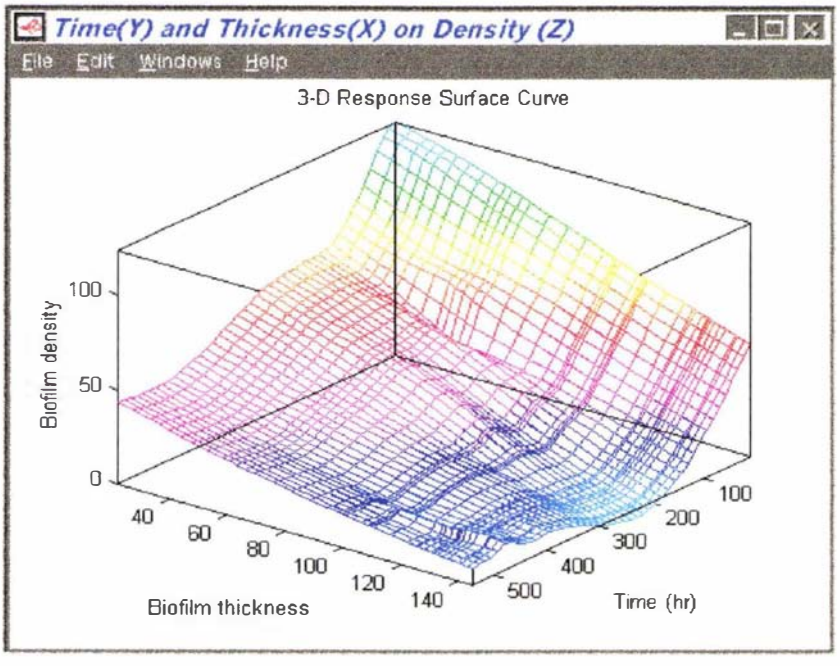


Figure 6.23. 3-D plot of time and biofilm thickness on biofilm density (data set 1).

6.5.2 Predictions of Dynamic Performance using Hybrid Model for a TPFBBR

The biofilm thickness and its density predicted by the neurofuzzy estimator are fed into axial-dispersion model and reaction-diffusion model as parameters. The biofilm thickness and biofilm density are also used for the estimation of some parameters such as the axial dispersion coefficient (Eq. (6.23)), the liquid-solid mass transfer coefficient (Eq. (6.24)), and the phase holdups (Eq. (6.25)-(6.27)).

The phenol concentration profiles in the liquid phase as a function of bed height at the different times are shown Fig. 6.24 (data set 1) and Fig. 6.25 (data set 2). It can be observed that the prediction of phenol concentration profiles as a function of bed height at the different times agrees quite well with off-line measurement data. From Fig. 6.25-6.26, during the biofilm growth after startup, the phenol removal rate and phenol concentration profiles as a function of the bed height vary along with corresponding variations in biofilm thickness and biofilm density. The axial phenol concentration profile drops off sharply from the bottom of the reactor to 30 *cm* of the bed height since much of the bioparticle is located in this area. Thus, significant of the substrate removal is taking place on approximately 35 % of the total bed height from the bottom of the reactor. In order to visualise the dynamic behavior of TPFBBR, the 3-dimensional plots of the phenol concentration profile as function of bed height and time are shown in Fig. 6.26(a)-(b). Fig. 6.27(a)-(b) also shows the 3-D plots of phenol concentration profiles within the biofilm.

The intelligent hybrid model used here performs well in predicting the phenol concentration profiles with a neurofuzzy process estimator and understanding the dynamic behavior of a TPFBBR process. It is demonstrated that the intelligent hybrid model used here provides a new comprehensive model for modelling the dynamics of a TPFBBR, which is the most complex reactor system to model.

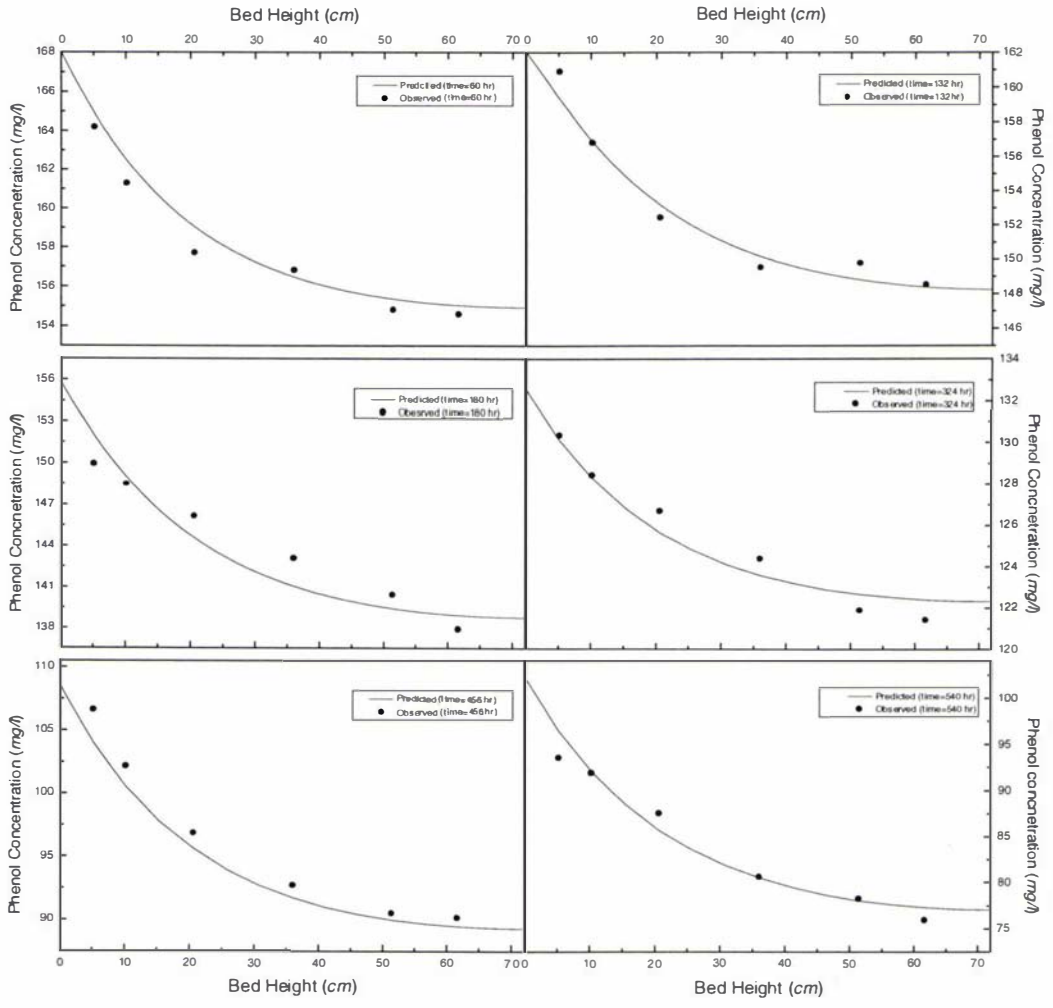


Figure 6.24. The substrate concentration profiles in the liquid phase as a function of bed height at the different times for data set 1.

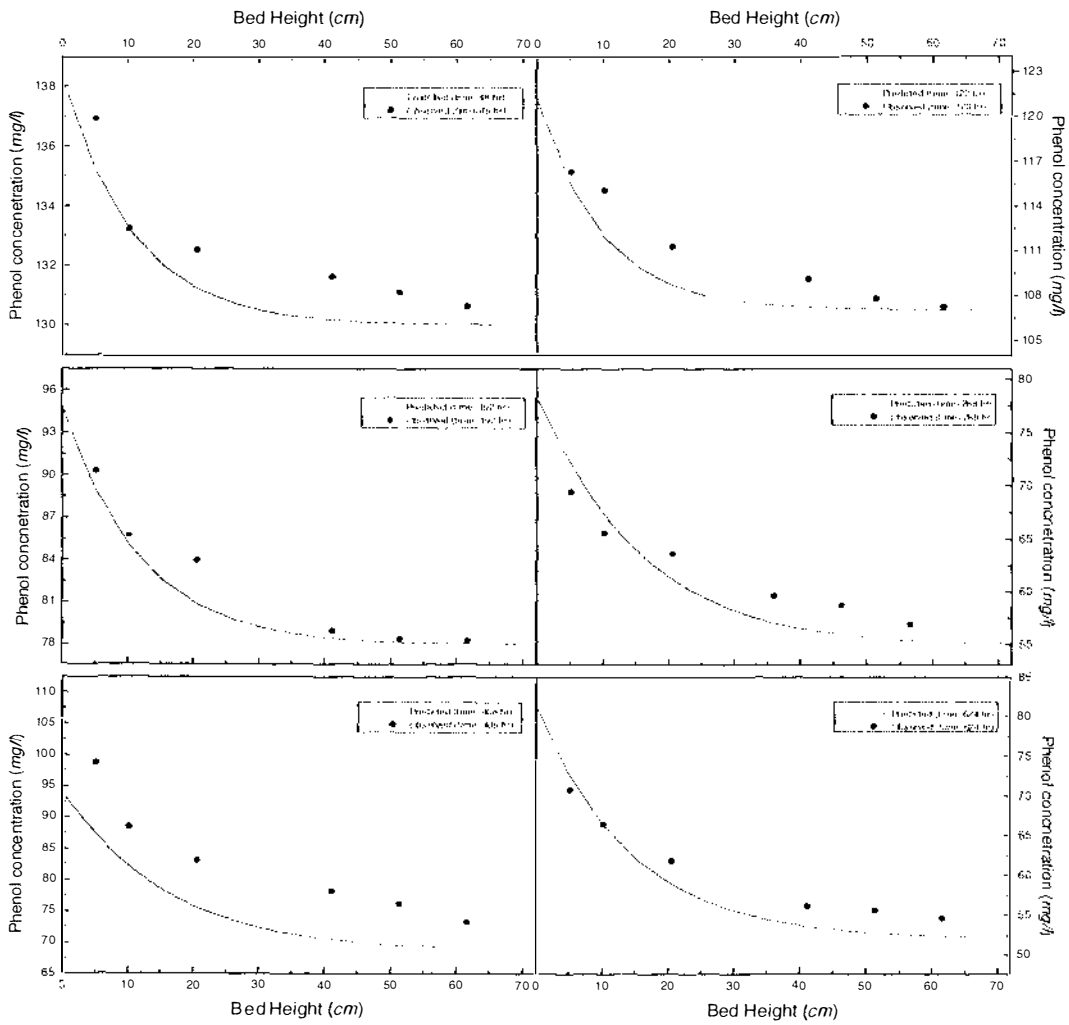
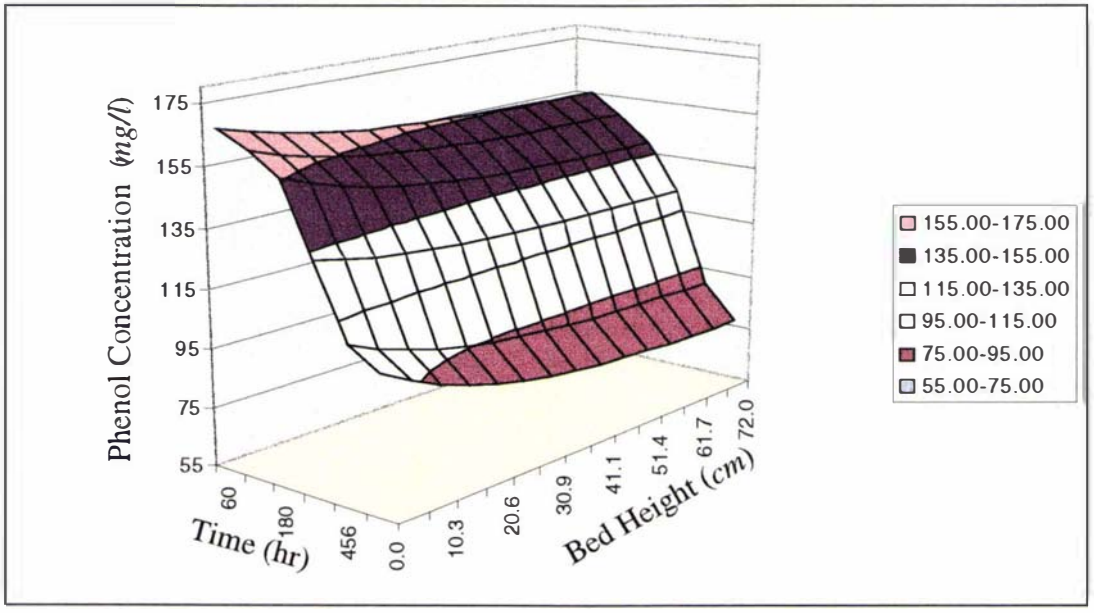
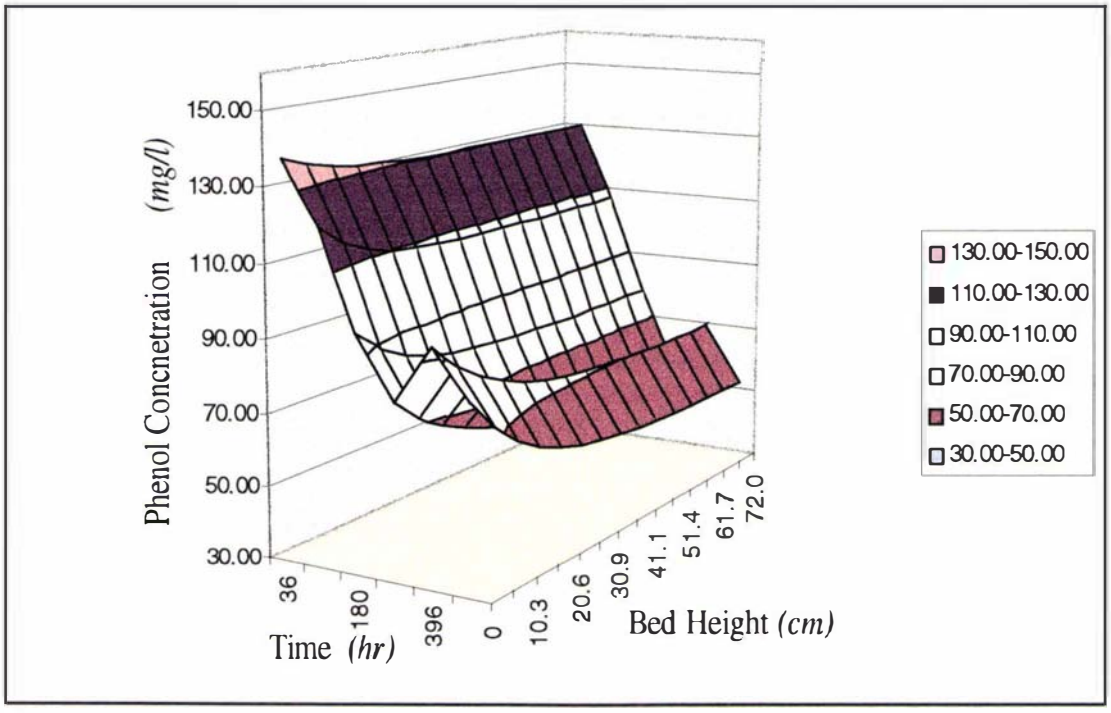


Figure 6.25. The substrate concentration profiles in the liquid phase as a function of bed height at the different times for data set 2.



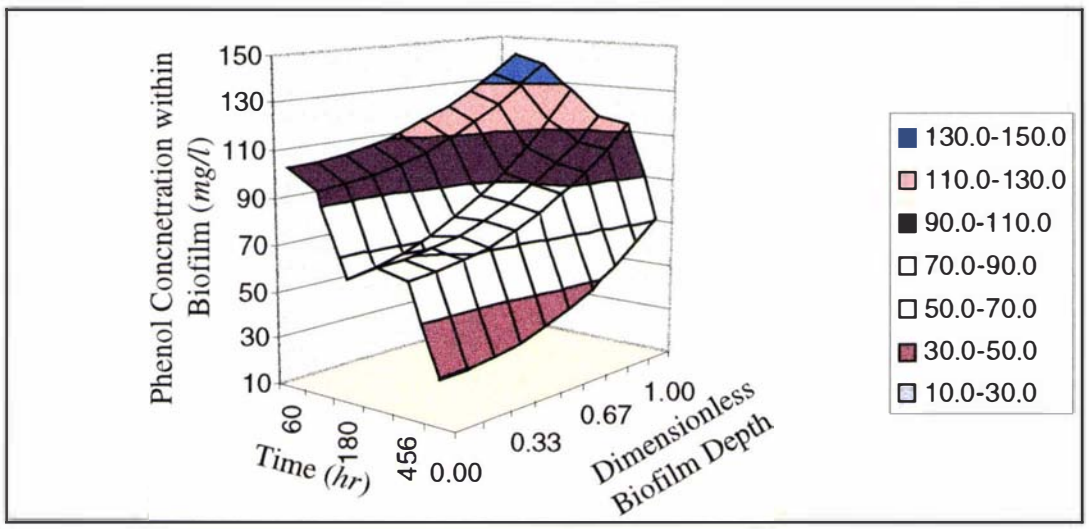
(a)

6.26(a). 3-D plots of phenol concentration profile as function of bed heights and time (data set 1).



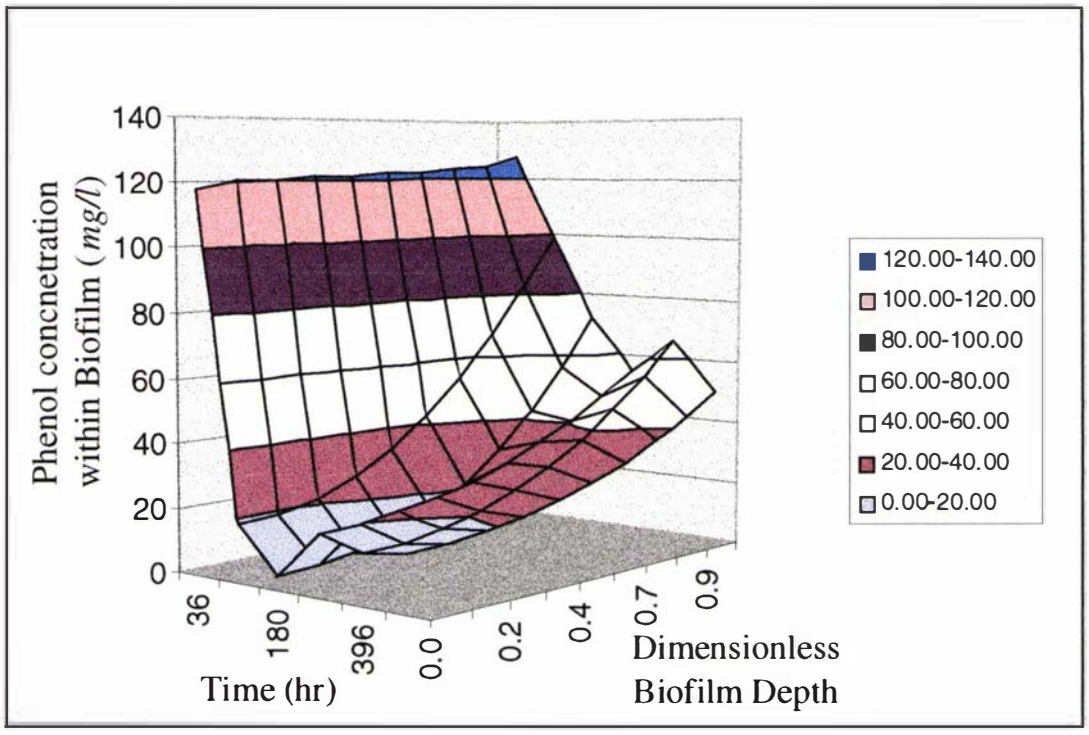
(b)

Figure 6.26(b). 3-D plots of phenol concentration profile as function of bed height and time (data set 2).



(a)

Figure 6.27(a). 3-D plot of phenol concentration profiles within the biofilm (data set 1).



(b)

Figure 6.27(b). 3-D plots of phenol concentration profiles within the biofilm: (b) data set 2.

6.6 CONCLUSIONS

We developed an intelligent hybrid modelling approach with a neurofuzzy process estimator to model a three-phase fluidised-bed biofilm reactor (TPFBBR). The intelligent hybrid model presented here consists of two parts including a neurofuzzy model, which serves as a *process estimator* of difficult-to-model process variables, and a mechanistic model, which represents a known mechanistic knowledge of the process system.

Here, the neurofuzzy part of the hybrid model was used to estimate the variation of the biofilm thickness and biofilm density in time. The outputs of the neurofuzzy process estimator subsequently form of part of the parameters for a mechanistic part of the hybrid model. The axial-dispersion model and reaction-diffusion model, which use the knowledge from the well-established general model, are used as the mechanistic model part of the hybrid model to describe the dynamic behavior and to predict the performance of TPFBBR system.

It is demonstrated that the intelligent hybrid model used provides a new comprehensive model for modelling the dynamics of a TPFBBR, which is known to be very difficult and complicated to model. The intelligent hybrid model performs well in modelling the dynamic behavior of a TPFBBR over the observed data. This intelligent hybrid model may be inexpensive, accurate, and reliable simultaneously, and should therefore be of potential interest for practical application.

It can be concluded that the hybrid modelling presented here provide a new framework scheme for modelling, estimation, and predicting other complex biological processes, such as a biofilm reactor whose dynamics of the biofilm growth are normally poorly known and can not easily be modelled. This approach therefore permits analysis and control of complex biological processes and reactors.

Chapter 7

Conclusions

7.1 SUMMARY

This thesis presents new models of a TPFBBR based on several modelling approaches. The issues we aimed to address include (a) developing the dynamic biofilm growth model, which reflects variation of biofilm thickness and its density in time, (b) implementing an integrated model incorporating dynamic biofilm growth to describe the dynamic behaviour of a TPFBBR, (c) developing the sequential neural network model as an alternative to tedious mechanistic models, and (d) developing the intelligent hybrid model with a neurofuzzy process estimator as a new modelling scheme.

Several conclusions can be drawn as a result of this research project. They are:

1. In Chapter 4, based on a mechanistic modelling approach, the dynamic biofilm growth model, which reflects variation of biofilm thickness and its density in time, is derived from a biomass balance equation using the method of characteristics. The biofilm detachment model is also proposed and incorporated

within the dynamic biofilm growth model. In our biofilm growth model incorporating detachment, the biofilm thickness and its density can be predicted in one model at a given time. It can be seen that the results predicted using a dynamic biofilm growth model, which accounts for varying biofilm thickness and its density in time, are in good agreement with experimental data.

2. In Chapter 4, the dynamic biofilm growth model with detachment is combined with a reaction-diffusion model and reactor model to from an integrated model of TPFBBR. The integrated model developed can describe the dynamic behaviour of TPFBBR. Simulation method of integrated model incorporating the dynamic biofilm growth model is developed. It is observed that the prediction of phenol concentration profiles as a function of bed height at the different times agrees well with experimental data.

3. Chapter 5 yields the application of a neural network model, as an alternative to complex mechanistic models of modelling of the dynamic change of the biofilm thickness and biofilm density and to predict the dynamic performance of a TPFBBR. To develop a model describing the process dynamics in a TPFBBR, the sequential neural network model is developed. The sequential neural network model presented here is composed of two parts. The first part, *the neural process estimator*, can serve as a nonparametric approximator of difficult-to-model process variables such as the biofilm thickness and biofilm density to estimate the biofilm thickness and biofilm density with the available measurement variables. The second part, *the neural process predictor*, can predict the dynamic change of performance of the TPFBBR based on the estimated biofilm thickness and biofilm density by the neural process estimator, and other measurement variables. It is demonstrated that the sequential neural network modelling approach proposed here provides a good alternative to describe the dynamic behavior of a TPFBBR and has the potential to be successfully implemented within a control scheme such as a nonlinear model predictive control (NMPC) scheme.

4. The results presented in Chapter 6 clearly show how the combination between two different modelling approaches, referred to as a hybrid model, can be coupled and utilised to model a TPFBBR. The intelligent hybrid model has a neurofuzzy process estimator to model the complicated TPFBBR. The results predicted using a neurofuzzy process estimator component of the hybrid model, which accounts for varying biofilm thickness and its density in time, are in good agreement with the observed values and thus the neurofuzzy process estimator serves as good numerical estimators. The mechanistic model components (reaction-diffusion model and axial-dispersion model) of the intelligent hybrid model also perform well in predicting the phenol concentration profiles with a neurofuzzy process estimator and an understanding the dynamic behavior of TPFBBR processes. This hybrid modelling approach can be of potential interest for practical applications and it suggests further work.

7.2 APPLICABILITY OF MODELS DEVELOPED IN THIS THESIS

The purpose of this section is to summarise and advice on the applicability of the modelling approaches implemented in this thesis. The different modelling approaches implemented in this thesis are compared in Fig. 7.1.

At the early stages in the model development, which is characterised by a lack of both empirical and mechanistic knowledge, an empirical modelling approach such as the neural network model presented in chapter 5 will often be useful as a starting point for more knowledge and eventually developing a mechanistic model. In process modelling, the transparency of the process model is important because model validation, model analysis, result interpretation, and application of the model are strongly dependent on a transparency of the model. From the transparency's point of view, the neural network modelling approach is often a black box that can not be directly interpreted in terms of the system mechanisms for a TPFBBR due to lack of

transparency. That is one major reason why the neural network model has only a limited applicability.

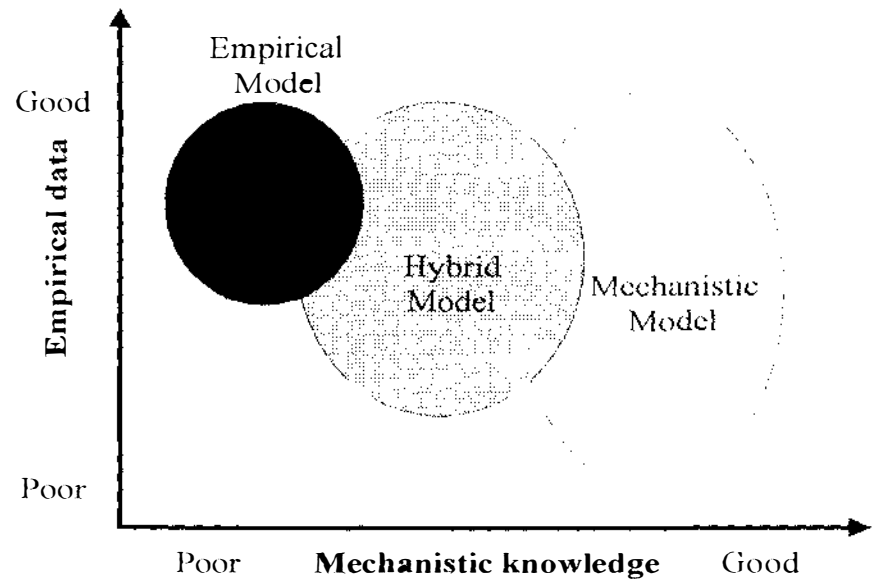


Figure 7.1. Different modelling paradigms.

In contrast, process engineer with the neural network model can collect more data, and end up in a state with more data and perhaps some improved mechanistic knowledge. In addition, the neural network model is often more accurate than a mechanistic model when the process is operating under similar conditions as when the data were collected. This result is also found in this thesis. Furthermore, the neural network model can be justified by the reduced time and effort required in building the models in real-world modelling problems. Linear empirical input/output models have been often used for model predictive control algorithms in some industries (Johansen, 1994). In the near future, the growing demand for improvement of process operation and supervision may require advanced controllers based on nonlinear models to become standard. Thus, the sequential neural network model

presented in Chapter 5 will have great advantages of applicability for the control of a TPFBBR.

In general, the mechanistic model has contributed greater understanding of the TPFBBR system, and is the ultimate goal of all model development of a TPFBBR. The mechanistic model is generally applicable for a wider class of problems than the empirical model due to the transparency of the model. Formulating an accurate model based on a mechanistic approach, however, requires very time-and money-consuming tasks since a deep understanding of the specific system mechanisms is not always available and certain aspects of the system are not sufficiently well understood.

Using the mechanistic modelling approach, the integrated model incorporating dynamic biofilm growth is developed in Chapter 4, but an open question remains:

> *Does the dynamic biofilm growth model presented in Chapter 4 describe the phenomena of biofilm growth completely?*

Although considerable amount of research has been undertaken to describe the biofilm growth on the support particles in a TPFBBR over the past three decades, knowledge of the system is still limited due to the complex and often poorly known nature of biofilm growth process. Most of mechanistic models for a TPFBBR are steady-state models that do not consider the dynamic biofilm growth, but the integrated model developed in Chapter 4 incorporates the biofilm growth as a dynamic process. Therefore, our mechanistic model provides a realistic description of a TPFBBR. As new phenomena for biofilm growth are discovered and more process data become available, incomplete knowledge can be updated into our model and our model can be improved.

Until recently, although both mechanistic and empirical models have strong points which can compensate weak points of both, the hybrid model (semi-mechanistic model, or semi-empirical model) has generally been avoided. A major reason for this

may be that software tools with a power of high-speed computation have been lacking. Along with the rapid improvement of computer technology, the hybrid modelling approach has recently attracted considerable attention in process modelling because of the increasing demand for nonlinear models to be applied in advanced controller, diagnosis systems, supervision, and optimal design. The hybrid modelling approach presented in Chapter 6 gives a model that is reasonably transparent, and supports both the empirical and mechanistic modelling approaches (Fig. 7.1). This is flexible in the sense that it may have different characteristics that depend on

- model representation
- amount and quality of relevant knowledge
- amount of relevant data available
- level of transparency
- type of empirical models used with it, like neural network, neurofuzzy, NARX (nonlinear autoregressive model with exogenous inputs), and NARMAX (nonlinear autoregressive moving-average model with exogenous inputs) etc.

For modelling complex problems such as a TPFBBR, the flexibility of a hybrid modelling approach with its transparency is the major advantage because it is characterised by a moderate combination of both mechanistic knowledge and empirical data. Compared to the neural network modelling approach presented in Chapter 5, the hybrid modelling approach of Chapter 6 has the same ability to fit data to an empirical model and provides greater transparency within the model. In contrast, the hybrid model may be thought of as a mechanistic model which contains process parameters whose dependence on process variables is modelled by an empirical model, in this case a neurofuzzy model. Our hybrid model provides the same level of model transparency as the existing steady-state models for a TPFBBR and demonstrates how the limitation of the existing steady-state models for a TPFBBR can be overcome.

The hybrid modelling approach due to its flexibility provides a unified framework through the incorporation of strong points for both mechanistic and empirical models. In addition, it plays an important role as a bridge because incomplete knowledge described by empirical models can be substituted with improved knowledge and it may lead to a completely mechanistic model for a TPFBBR as the mechanisms are better understood. The hybrid modelling approach presented in Chapter 6 provides a new modelling framework with a great potential of applicability to the other types of biofilm reactors.

This thesis represents a research effort on developing mathematical models incorporating biofilm growth using three different modelling approaches. The research findings bear importance to successful reactor design, operation, supervision, and control of a TPFBBR for future industrial application.

NOMENCLATURE

A	bed cross-sectional area (cm^2)
a_p	biofilm surface area per unit volume of solids (cm^{-1})
Ar_l	liquid Archimedes number
b_{de}	biofilm detachment coefficient ($\sec^{-1}$)
C_s^f	phenol concentration in the biofilm phase (mg/cm^3)
C_s^{in}	inlet phenol concentration (mg/cm^3)
C_s^l	phenol concentration in liquid phase (mg/cm^3)
d_{bp}	bioparticle diameter (cm)
d_{sp}	support particle diameter (cm)
D_c	reactor diameter (cm)
D_m	molecular diffusivity in liquid.
D_{sf}	effective diffusion coefficient in the biofilm ($cm^2/\sec$)
E	specific energy dissipation rate
E_L	axial-dispersion coefficient ($cm^2/\sec$)
E_P	energy dissipation rate
E_S	axial solid dispersion coefficient
Fr	Froude number
g	gravitational acceleration
Ga	Gallileo number
H	bed height (cm)
H_e	expanded bed height (cm)
K_i	inhibition constant of substrate
$K_l a$	gas-liquid mass transfer coefficient
k_{ls}	liquid-solid mass transfer coefficient ($cm/\sec$)

K_s saturation constant of substrate (mg/cm^3)

L radial position within the bioparticle

L_f biofilm thickness (cm)

L_{f_0} initial biofilm thickness (cm)

M_{bp} total mass of bioparticles (mg)

ΔP pressure drop across the bed

Pe_L Peclet number

R_d detachment rate

Re Reynolds number

Re_{lmf} liquid Reynolds number at minimum three-phase fluidization

Sc Schmidt number

Sh Sherwood number

St Stanton number

$U(L_f, t)$ velocity of the biofilm thickness change ($cm/\sec$)

$u(C_s^f)$ specific growth rate of biofilm ($\sec^{-1}$)

$\bar{u}$ net specific growth rate of the biofilm ($\sec^{-1}$)

$u_{\max}$ maximum specific growth rate ($\sec^{-1}$)

U_L superficial liquid velocity ($cm/\sec$)

U_{lmf} minimum fluidization velocity

U''_{lmf} liquid velocity at minimum liquid-slid fluidisation

U_G superficial gas velocity ($cm/\sec$)

U_{t_0} terminal falling velocity of a single particle in an infinite liquid medium

V_G velocity gradient

X_a^f biofilm density (mg/cm^3)

$X_{a_{\max}}^f$ maximum biofilm density (mg/cm^3)

$Y_{x/s}$ growth yield coefficient of substrate

Z axial distance from the bottom of bed

Greek letters

α_{mf} gas holdup on solid-free basis

ε_g gas holdup

ε_l liquid holdup

ε_{mf} bed porosity at minimum fluidization velocity

ε_s solid holdup

μ_L viscosity of liquid

ρ_{bp} density of bioparticle (mg/cm^3)

ρ_{bw} wet density of biofilm (mg/cm^3)

ρ_G density of gas

ρ_L density of liquid

ρ_{sp} density of support particle (mg/cm^3)

τ liquid shear stress ($dyne/cm^2$)

ϕ particle shape factor (0.906),

Superscrpts

f biofilm phase

l liquid phase

in inlet

s surface of biofilm

Subscripts

s substrate

REFERENCES

Alvarez-Cuenca, et al. (1983) Oxygen mass transfer in three-phase fluidized beds working with large flow rates. *Can. J. Chem. Eng.*, **61**, 58.

Amirtharajah, A. (1978) Optimum backwashing of sand filters. *J. Envir. Engng. Div, ASCE*, **104**, 917.

Andrews, G. F. and Tien, C. (1981) Bacterial film growth on adsorbent surface. *AICHE J.*, **27**, 396.

APHA (1989) *Standard Methods for the Examination of Water and Wastewater*. 17th editions, American Public Health Association, New York.

Armstrong, E. R., Baker, C. G. J., and Bergougnou, M. A. (1976) The effects of solids wettability on the characteristics of three-phase fluidization. In *Fluidization Technology*, D. L. Keairns, Ed., Hemisphere, Washington, D. C., 453.

Aris, R. and Amundson, N. (1973) *Mathematical Methods in Chemical Engineering, Volume II, First-order Partial Differential Equations and Applications.* Englewood Cliffs, N.J., Prentice-Hall.

Arters, D. C. and Fan, L.-S. (1986) Solid-liquid mass transfer in a gas-liquid-solid fluidized Bed. *Chem. Eng. Sci.*, **41**, 107.

Atkinson, B. and Daoud, I. S. (1970) Diffusion effects with microbial films. *Trans. IChemE.*, **48**, 245.

Atkinson, B., Swilley, E. L., Bush, A. W., and Williams D. A. (1967) Kinetics, mass transfer and organism growth in a biological film reactor. *Trans. IChemE.*, **45**, 257.

Alvarez-Quenca et al. (1983) Oxygen mass transfer in three-phase fluidized beds working at large flow rates. *Can. J. Chem. Eng.*, **61**, 58.

Bakke, R., Characklis, W. G., Turakhia, M. H., and Yeh, A.-I. (1990) Modeling a monopopulation biofilm system: *Psedomonas Aeruginosa.* In *Biofilm*, Characklis and Marshal. Eds., Wiley, New York.

Bakke, R., Trulear, M. G., Robinson, J. A., and Characklis, W. G. (1984) Activity of *Pseudomonas aeruginosa* in biofilms: steady state. *Biotechnol. Bioeng.*, **26**, 1418.

Bastin, G., and Dochain, D. (1990) *On-line Estimation and Adaptive Control of Bioreactors.* Elsevier, Amsterdam.

Baum, E. B. and Haussler, D. (1989) What size net gives valid generalization? *Neural Computation,* **1**, 151.

Beek, W. J. (1971) Mass transfer in fluidized beds. In *Fluidization*, Eds., J. F. Davison and D. Harrison, Academic Press, London & New York.

Begovich, J. M., and Watson, J. S. (1978a) Hydrodynamic characteristics of three-phase fluidized beds. In *Fluidization.*, D. F. Davison and D. L. Keairns, Eds., Cambridge Univ. Press.

Begovich, J. M and Watson. J. S. (1978b) Hydrodynamic characteristics of three-phase fluidized beds. *Fluidization, Proc. 2nd Eng. Found. Conf.,* Cambridge.

Benefield, L. and Molz, F. (1985) Mathematical simulation of a biofilm process. *Biotechnol. Bioeng.*, **27**, 921.

Berenji, H. R. and Khedkar, P. (1993) Learning and tuning fuzzy logic controller through reinforcements. *IEEE. Trans. Neural Networks*, **3,** 724.

Berthouex, P. M. and Box, G. E. (1996) Time series models for forecasting wastewater treatment plant performance. *Water. Res.*, **30**, 1865.

Bhamidimarri, S. M. R., Greenfield, P. F., and Bell, P. R. F. (1987) Oxidation of refractory organics in attached growth systems: Initial bacterial adsorption. *J. Ferment. Technol.*, **65**, 449.

Bhamidimarri, S. M. R. and See, T. T. (1992) Shear loss characteristics of an aerobic biofilm. *Wat. Sci. Tech.,* **26**, 595.

Bhatia, V. K., Evans, K. A, and Epstein, N. (1972) Effect of solids wettability on expansion of gas- liquid fluidized Beds. *Ind. Eng. Chem. Process Des. Dev.*, **11(1)**, 151.

Bishop, P. L., Zang, T. C., and Fu, Y.-C. (1995) Effects of biofilm structure, microbial distributions and mass transport on biodegradation processes. *Wat. Sci. Tech.*, **31,** 143.

Breusegem, V. V., Thibault, J., and Cheruy, A. (1991) Adaptive neural models for on-line prediction in fermentation. *Can. J. Chem. Eng.*, **69**, 481.

Brodkey, R. S. and Hershey, H. C. (1988) *Transport Phenomena: A Unified Approach.* McGraw-Hill, New York.

Brown, M. and Harris, C. (1994) *Neurofuzzy adaptive modelling and control.* Prentice Hall, New York.

Bryers, J. D. (1984) Biofilm formation and chemostat dynamics: pure and mixed culture considerations. *Biotechnol. Bioeng.*, **26**, 948.

Bryers J. D. and Characklis, W. K. (1982) Process governing primary biofilm formation. *Biotechnol. Bioeng.*, **24**, 2451.

Buckley, J. J. and Yoichi, H. (1994) Fuzzy neural networks. In *Fuzzy Sets, neural networks and Soft computing*, Ronal R. Yager and Lotfi A. Zadeh (Eds.) New York: van Nostrand Reinhold, 233.

Camp, T. R. (1964) Theory of water filtration. *J. San. Engng. Div., ASCE*, **90**, 1.

Capodaglio, A. G., Jones, H. V., Novotny, V., and Feng, X. (1991) Sludge bulking analysis and forecasting: application of system identification and artificial neural computing technologies. *Wat. Res.*, **25**, 1217.

Chang, H. T. and Rittman, B. E. (1988) A comparative study of biofilm on activated carbon. *J. Water Poll. Control Fed.*, **60**, 362.

Chang, H. T., Rittmann, B. E., Amar, D., Heim, R., Ehlinger, O., and Lesty, Y. (1991) Biofilm detachment mechanisms in a liquid-fluidized bed. *Biotechnol. Bioeng.*, **38**, 499.

Collins, A. G. and Ellis, G. W. (1992) Information processing coupled with expert systems for water treatment plants. *ISA Trans.*, **31**(1), 61.

Cote, M., Grandjean, B. P. A., Lessard, P., and Thibault, J. (1995) Dynamic modelling of the activated sludge process: improving prediction using neural networks. *Wat. Res.*, **29**, 995.

Croughan, M. S., Hamel, J.-F. P., and Wang, D. I. C. (1988) Effects of microcarrier concentration in animal cell culture. *Biotechnol. Bioeng.*, **32**, 975.

Cybenko, G. (1989) Approximation by superposition of a sigmoidal function. *Math. Control Signal Sys.*, **2**, 303.

Dakshinamurty, P., Sabrahmanyam, V., and Rao, J. N. (1971) Bed porosities in gas-liquid fluidization. *Ind. Eng. Chem. Process Des. Dev.*, **10**, 322.

Dalili, M. and Chau, P. C. (1987) Intraparticle diffusional effects in immobilized cell particles. *Appl. Microbiol. Biotechnol.*, **26**, 500.

Darton, R. C. and Harrison, D. (1976) Bubble wake structure in three-phase fluidization. In *Fluidization Technology*, Kearins, Ed., I 399, Hemisphere.

Davison, J. F., et al., (1977) Three-phase systems. In *Chemical reactor Theory: a review,* Lapidus and Amundson, Eds., Prentice-Hall, 655.

Debelak, K. A. and Sims, C. A. (1981) Stochastic modeling of an industrial activated sludge process. *Wat. Res.*, **15**, 1173.

Dhanuka, V. R., and Stepanek, R. (1978) Gas and liquid hold-up and pressure drop measurements in a three-phase fluidized bed. In *Fluidization*, Davison, J. F. & Keairns, D. L., Eds., Cambridge Univ. Press, 179.

El-Temtamy, S. A and Epstein, N. (1978) Bubble wake solids content in three-phase fluidized beds. *Int. J. Multiphase Flow*, **4**, 1978.

Epstein, N. (1981) Three-phase fluidization: Some knowledge gaps. *Can. J. Chem. Eng.*, **59**, 649.

Ergun, S. (1952) Fluid flow through packed column. *Chem. Eng. Pro.*, **48**, 89.

Ermakova, A., et al., (1973) A catalytic reactor with a three-phase gas-liquid-solid catalytic fluidized bed. *Theor. Found. Chem. Eng.*, **7**, 29.

Fahlman, S. E. (1988) Faster-Learning Variations on Back-Propagation: An Empirical Study. In *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann.

Fahlman, S. E. and Lebiere, C. (1990) The cascade-correlation learning architecture. In *Advance in neural Information processing Systems II*, D. S. Tourretzky, G Hinton, and T. Sejnowski, Eds. San Mateo, CA: Morgan Kaufmann.

Fan, L.-S., Chern, S. H., and Muroyama, K. (1982) Hydrodynamic characteristic of a gas-liquid-solid fluidized bed containing a binary mixture of particles. *AICHE Meeting*, LA.

Fan, L.-S., Fujie, K., Long, T.-R., and Tang, W.-T. (1987) Characteristics of draft tube gas-liquid-solid fluidized bed bioreactor with immobilized living cells for phenol degradation. *Biotechnol. Bioeng.*, **30**, 498.

Fan, L.-S., Leyva-Ramos, R., Wisecarver, K. D., and Zehner, B. J. (1990) Diffusion of phenol through a biofilm growth on activated carbon particles in a draft-tube three-phase fluidized-bed bioreactor. *Biotechnol. Bioeng.*, **35**, 279.

Fan, L.-S., Yamashita, T., and Jean, R.-H. (1984) Solid mixing and segregation a gas-liquid-solid fluidized bed. *AICHE Meeting,* San Francisco.

Franzini, M. A. (1987) Speech recognition with Back-Propagation. *Proceedings of 9th annual conference of IEEE Engineering* in medicine and Biology Society.

Fuller, R. (1995) *Neural Fuzzy System*, Lecture notes, Turku Center for Computer Science.

Furzeland, R. M., Verwer, J. G., and Zegeling, P. A. (1990) A numerical study of three moving grid methods for one-dimensional partial differential equations which are based on the method of lines. *J. Comput. Phys.*, **89**, 349.

Gjaitema, A., Tijhuis, L., Loosdrecht, M. C. M., and van Heijnen, J. J. (1995) Detachment of biomass from suspended nongrowing spherical biofilms in airlift reactors. *Biotechnol. Bioeng.*, **46**, 258.

Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.

Grady Jr., C. P. L. (1983) Modelling of biological fixed films - a state-of-the-art review. In *Fixed-Film Biological Process for Wastewater Treatment*, Wu, Y. C. and Smith, E. D., Eds., Noyes Data Corporation, New Jersey, USA.

Gupta, M. M. and Rap, D. H. (1994) On the principles of fuzzy neural networks. *Fuzzy Sets and Systems*, **61**, 1.

Haykin, S. (1994) *Neural Networks: A Comprehensive Foundation.* Prentice Hall, New Jersey, Upper Saddle River, USA.

Hecht-Nielsen, R. (1988) Neurocomputer Applications. In *Neural Computers*, Eckmiller et al., Eds., Springer-Verlag, New York.

Hoehn, R. C., and Ray, A. D. (1973) Effects of thickness on bacterial films. *J. Water Pollut. Control Fed.*, **45**, 2302.

Holladay, D. W., Hancher, C. W., Scott, C. D., and Chilite, D. D. (1978) Biodegradation of phenolic waste liquors in stirred-tank, packed-bed, and fluidized-bed bioreactor. *J. Water Poll. Contol Fed.*, **50**, 2573.

Hong, Y.-S., Bhamidimarri, S. M. R., and Parshotam, A. (1998) Modelling of biofilm growth and detachment in a three-phase fluidised-bed bio-reactor. *IPENZ Annual Conference*, Chemical Eng. Section, **5**, 43.

Hong, Y.-S., Bhamidimarri, S. M. R., and Parshotam, A. (1998) A genetic adapted neural network analysis of performance of the nutrient removal plant at Rotorua. *IPENZ Annual Conference*, Simulation & Control Section, **2**, 43.

Horikawa, S., Furahashi, T., and Uchikawa, Y. (1992) On fuzzy modelling using fuzzy neural networks with the backpropagation algorithms. *IEEE. Trans. Neural Networks*, **3**, 801.

Huang, J.-S. and Wu, C.-S. (1996) Specific energy dissipation rate got fluidized bed bioreactor. *Biotechnol. Bioeng.*, **50**, 643.

Hunter, J. B. and Asenjo, J. A. (1990) A population balance model of enzymatic lysis of microbial cells. *Biotechnol. Bioeng.*, **35**, 31.

Hutchinson, D. H. and Robinson, C. W. (1990) A microbial regeneration process for granular activated carbon- I. process modelling. *Wat. Res.*, **24**, 1209.

Ikonen, E. (1996) *Algorithms for process modelling using fuzzy neural networks.* Ph.D. Thesis, Department of Process Engineering, University of Oulu, Finland.

Jang, J.-S. R. (1993) ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Trans. Systems, Man & Cybernetics*, **23**, 665.

Jeris, J. S. and Owens, R. W. (1975) Pilot-scale, high-rate biological denitrification. *J. Water Poll. Control Fed.*, **47**, 2043.

Jeris, J. S., Owens, R. W., Hickey, R., and Flood, F. (1977) Biological fluidized-bed treatment for BOD and nitrogen removal. *J. Water Poll. Control Fed.*, **49**, 816.

Jin, Y., Jiang, J., and Zhu, J. (1995) Neural network based fuzzy identification and its application to modelling and control of complex systems. *IEEE Trans. Systems Man & Cybernetics*, **25**, 990.

Johansen, T. A. (1994) *Operating Regime based Process Modeling and Identification.* Ph.D. Thesis, The Norwegian Institute of Technology-University of Trondheim.

Jones, W. L., Dockery, J. D., Vogel, C.R., and Sturman, P.J. (1993) Diffusion and reaction within porous packing media: a phenomenological model. *Biotechnol. Bioeng.*, **41**, 947.

Joshi, J. B. (1980) Axial mixing in multiphase contractors-a unified correlation. *Trans. Inst. Chem Eng.*, **58**, 155.

Karim, M. N. and Rivera, S. L. (1992) Artificial neural networks in bioprocess state estimation. In *Advances in Biochemical Engineering/Biotechnology*, Edited by Fiecher, A., Springer-Verlag, 1.

Kato, Y., Morooka, S., Kago, T., Sarawatari, T., and Yang S.-Z. (1985) Axial holdup distribution of gas and solid particles in three-phase fluidized bed for gas-liquid(slurry)-solid system. *J. Chem. Eng. Japan*, **18**, 308.

Kato, Y., Nishiwaki, A., Kago, T., Fukuda, T., and Tanaka, S. (1972) Gas holdup and overall volumetric absorption coefficient in bubble column with suspended

solid particles: absorption rate of oxygen by an aqueous solution of sodium sulfite. *Int. Chem. Eng.*, **13**, 582.

Kim, B. R., Schimitz, R. A., Snoeyink, V. L., and Tauxe, G. W. (1978) Analysis of models for dichloramine removal by activated carbon in batch and packed-bed reactors using quasilinearlization and orthogonal collocation methods, *Water Res.*, **12**, 317.

Kim, S. D., Baker, C. G., and Bergougnou, M. A. (1975) Phase holdup characteristics of three-phase fluidized beds. *Can. J. Chem. Eng.*, **53**, 134.

Kim, S. D., Kim, H. S., and Han, J. H. (1992) Axial dispersion characteristics in three-phase fluidized beds. *Chem. Eng. Sci.*, **47**, 3419.

Kohonen, T. (1990) Self-organizing map. *Proc. IEEE*, **78**, No. 9, 1464.

Kong, S.-G. and Kosko, B. (1992) adaptive fuzzy systems for backing up a truck-and trailer. *IEEE Trans. Neural Networks*, **3**, 211.

La Motta, E. J. (1974) *Evaluation of diffusional resistances in substrate utilization by biological films.* Ph.D. Thesis, University of North Carolina.

La Motta, E. J. (1976) Internal diffusion and reaction in biological films. *J. Environ. Eng.*, **10**, 765.

Lange, F. and Hirzinger, G. (1995) Application of multilayer perceptrons to decouple the dynamical behaviour of robot links. In *Int. Conf. on Artificial Neural networks,* Paris, France.

Lee, C. C. (1990) Fuzzy logic in control systems: fuzzy logic controller-part 1 and part 2. *IEEE Trans. Systems, Man & Cybernetics*, **20**, 665.

Lee, S. L. P and deLasa, H. I. (1987) Phase Holdup in three-phase fluidized Bed. *AICHE J.*, **33**, 1359.

Lee, D. D., Scott, C. D., and Hancher, C. W. (1979) Fluidized-bed bioreactor for coal-conversion effluent. *J. Water Pollut. Control Fed.,* **51**, 974.

Lee, J. C. and Worthington, H. (1974) Gas-liquid mass transfer in three-phase fluidized beds. *Int. Chem. E. Symp. Ser.,* No. 38, London.

Lin, C.-T. and Lee, C. S. G. (1991) Neural-network-based fuzzy logic control and decision system. *IEEE Trans. Computers*, **40**, 1320.

Linko, P. and Zhu, Y. (1991) Neural network programming in bioprocess variable estimation and state prediction. *J. Biotechnol.*, **21**, 253.

Livingston, A. G. and Chase, H. A. (1989) Modeling phenol degradation in a fluidized-bed bioreactor. *AICHE J.*, **35**, 1980.

Maniezzo, V. (1994) Genetic evolution of the topology and weight distribution of neural networks. *IEEE Trans. Neural Networks*, **5**, 39.

Miller, G. F., Todd, P. M., and Hedge, S. U. (1989) Designing neural networks using genetic algorithms. *Proc. of the Third Int'l Conf. on Genetic Algorithms and Their Applications*, Morgan Kaufmann, San Mateo, CA, 379.

Mamdani, E. H. (1974) Application of fuzzy algorithms for control of simple dynamic plant. *Proc. of IEEE*, **121**, No 12, 1585.

Michelsen, M. L. and Ostergaard, K. (1970) Hold-up and fluid mixing in gas-liquid fluidized beds. *Chem. Eng. J.*, **1**, 37.

Morris, A. J., Montague, G. A., and Willis, M. J. (1994) Artificial neural networks studies in process modelling and control. *Trans. IChemE.*, **72A**, 3.

Mulcahy, L. T. (1978) *Mathematical model of the fluidized bed biofilm reactor.* Ph.D. Thesis, University of Massachusetts.

Mulcahy, L. T., Shieh, W. K., and Lamotta, E. J. (1981) Simplified mathematical models for a fluidized-bed biofilm reactor., *AICHE Symp. Ser, No.209*, **77**, 273.

Muroyama, K. and Fan, L.-S. (1985) Fundamentals of gas-liquid-solid fluidization. *AICHE J.*, **31**, 1.

Muroyama, K., et al. (1978) Axial liquid mixings in three-phase fluidized beds. *Kagaku Kogaku Ronbunshu*, **4**, 662.

Nauck, D and Rudolf Cruse, R. (1996) Designing neuro-fuzzy systems through backpropagation. In W. Pedrycz, Editor, *Fuzzy Modelling: Paradigms and Practice*, Kluwer, Boston, 203.

Nie, J. and Linkens, D. A. (1993) Learning control using fuzzified self-organizing radial basis function network. *IEEE Trans. Fuzzy Systems*, **1**, 280.

Livingston, A. G. and Chase, H. A. (1989) Modeling phenol degradation in a fluidized-bed bioreactor, *AICHE J.*, **35**, 1980.

Nore, O., Briens, C., Margaritis, A., and Wild, G. (1992) Hydrodynamics, gas-liquid mass transfer and particle-liquid heat and mass transfer in a three-phase fluidized bed for biochemical process application. *Chem. Eng. Sci.*, **47**, 3573.

Nuguen-Tien, K. and Deckwer, W.-D. (1981) Effect of liquid flow and particle concentration on gas-liquid mass transfer in a bubble column. *Chem. Eng. Sci.*

Ostergaard, K. (1968) Gas-liquid-particle operation in chemical reaction engineering. In *Adv. in Chem. Eng.*, T. B. Drew et al., Eds., Academic Press, 71.

Ostergaard, K. and Fosbol, P. (1972) Transfer of oxygen across the gas-liquid interface in gas-Liquid fluidized beds, *Chem. Eng. J.*, **3**, 105.

Ostergaard, K. and Michelsen, M. L. (1969) On the use of the imperfect tracer pulse method for determination of hold-up and axial mixing. *Can. J. Chem. Eng.*, **47**, 107.

Park, Y. M., Davis, M. E., and Wallis, D. A. (1984) Analysis of a continuous, aerobic, fixed-film bioreactor. I : steady state behavior. *Biotechnol. Bioeng.*, **26**, 457.

Pedrycz, W. (1992) Fuzzy neural networks with reference neurons as pattern classifiers. *IEEE. Trans. Neural Networks*, **3**, 770.

Perry, H. H. (1996) *Chemical Engineer's Handbook*, 7th Edition, McGraw Hill, New York.

Petersen, J. N. and Davison, B. H. (1995) Development of a predictive description of an immobilized-cell, three-phase, fluidized-bed bioreactor. *Biotechnol. Bioeng.*, **46**, 139.

Petzold, L. R. (1983) A description of DASSL: A differential/algebraic system solver. In *IMACS Transaction on Scientific Computation*, Stepleman, R. S., Ed. North-Holland, Amsterdam.

Peyton, B. M. and Characklis, W. G. (1992) Kinetics of biofilm detachment. *Wat. Sci. Tech.*, **26**, 595.

Peyton, B. M. and Characklis, W. G. (1993) A statistical analysis of the effect of substrate utilization and shear stress on the kinetics of biofilm detachment. *Biotechnol. Bioeng.*, **41**, 728.

Psichogious, D. C. and Ungar, L. H. (1991) A hybrid neural network first principles approach to process modelling. *AICHE J.*, **38**(10), 1499.

Puskorius, G. V. and Feldkamp, L. A. (1994) Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks. *IEEE Trans. Neural Networks*, **5**, No. 2, 279.

Rittmann, B. E. (1982) The effect of shear stress on biofilm loss rate. *Biotechnol. Bioeng.*, **24,** 501.

Rittmann, B. E. and McCarty, P. L. (1981) Substrate flux into biofilms of any thickness. *J. Environ. Eng.*, **107**, 831.

Robinson, C. W. and Wilke, C. R. (1974) Simultaneous measurement of interfacial area and mass transfer coefficient for a well-mixed gas dispersion in aqueous electrolyte solutions, *AICHE J.*, **20**, 285.

Rowe, P. N. and Claxton, K. T. (1966) Heat and mass transfer from a single sphere to fluid moving through an array. *Trans. IChemE.*, **43**, 321.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986) Learning Internal Representations by error propagation in Rumelhart, D. E and McClelland, J. L., *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, MIT Press.

Rumelhart, D. E., McClelland, J. L., and the PDP Research Group. (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Vol. **1**, Cambridge, MA: MIT Press.

Scott, D. S. and Hancher, C. W. (1976) Use of a tapered fluidized bed as a continuous bioreactor. *Biotechnol. Bioeng.*, **22**, 667.

Shah, Y. T. (1979) *Gas-liquid-solid reactor design.* McGraw-Hill.

Shah, Y. T., Kelkar, B. G., and Godbole, S. P. (1982) Design parameters estimation for bubble column reactors. *AICHE J.*, **28**, 353.

Saxen, B. and Saxen, H. (1996) A neural-network based model of bioreaction kinetics. *Can. J. Chem. Eng.*, **74**, 124.

Shieh, W. K. (1980) Suggested kinetics model for the fluidized-bed biofilm reactor. *Biotechnol. Bioeng.*, **22**, 667.

Shieh, W. K. and Keenan, J. D. (1986) Fluidized bed biofilm reactor for wastewater treatment. In *Advances in Biochemical/Biotechnology*, **33**, Springer-Verlag, Berlin Heidelberg, Germany.

Shieh, W. K., Sutton, P. M., and Kos, P. (1981) Predicting reactor biomass concentration in a fluidized bed system. *J. Water Pollut. Control Fed.*, **53**, 1574.

Simutis, R., Havlik, I., and Lubbert, A. (1993) Fuzzy-aided neural network for real-time state estimation and process prediction in the alcohol formation step of production-scale beer brewing. *J. Biotechnol.*, **27**, 203.

Schubert, J., Simutis, R., Dors, M., Havlik, I., and Lubert, A. (1994) Bioprocess optimisation and control: application of hybrid modeling. *J. Biotechnol.*, **35**, 51-68

Singhal, S. and Wu, L. (1989) Training multilayer perceptrons with the extended Kalman algorithm. in *Advance in neural Information processing Systems I*, D. S. Tourretzky Ed., San Mateo, CA: Morgan Kaufmann.

Soung, W. Y. (1978) Bed expansion in three-phase fluidization. *Ind. Eng. Chem. Process Des. Dev.*, **17**, 33.

Specht, D. A. (1991) General Regression Neural Network. *IEEE Trans. Neural Networks*, **2**, No.5, 568.

Speitel, G. E. and DiGiano, F. A. (1987) Biofilm Shearing under dynamic conditions. *J. Environ. Eng., ASCE*, **113**, 464.

Su, H. T., Bhat, N., Minderman, P. A., and McAvoy, T. J. (1992) Integrating neural networks with first principles models for dynamic modeling. *IFAC Symp. on Dynamics and Control of Chemical Reactors, Distillation Columns, and Batch Processes*, (DYCORD+).

Schubert, J., Simutis, R., Dors, M., Havlik, I., and Lubert, A. (1994) Bioprocess optimisation and control: application of hybrid modeling. *J. Biotechnol.*, **35**, 51.

Takagi, H. and Lee, M. (1992) Neural networks and genetic algorithm approaches to auto-design of fuzzy systems.

Takagi, T. and Sugeno, M. (1985) Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Syst., Man & Cybernet.*, **15**,116.

Tang, W.-T. and Fan, L.-S. (1989) Hydrodynamics of a three-phase fluidized beds containing low-density particles. *AICHE J.*, **35**, 355.

Tang, W.-T. and Fan, L.-S. (1987) Steady state phenol degradation in a draft-tube, gas-liquid-solid fluidized-bed bioreactor. *AICHE J.*, **33**, 239.

Tanyolac, A. and Beyenal, H. (1997) Prediction of average biofilm density and performance of a spherical bioparticle under substrate inhibition. *Biotechnol. Bioeng.*, **56**, 319.

Tavares, C. R. G., Santanna JR, G. L., and Capdeville, B. (1995) The effect of air superficial velocity on biofilm accumulation in a three-phase fluidized-bed reactor. *Wat. Res.*, **29**, 2293.

Thibault, J., Van Breusegem, V. and Cheruy, A. (1990) On-line prediction of fermentation variables using neural networks. *Biotechnol. Bioeng.*, **36**, 1041.

Tijhuis. L., van Loosdrecht, M. C. M., and Heijnen, J. J. (1994) Formation and growth of heterotrophic aerobic biofilms on small suspended particles in airlift reactors. *Biotechnol. Bioeng.*, **44**, 595.

Tompson, M. L. and Kramer, M. A. (1994) Modeling chemical process using prior knowledge and neural networks. *AICHE J.*, **40**, No.8, 1328.

Trambouze, P., Van Landeghem, H., and Wauquier, P. (1988) *Chemical Reactor design/engineering/operation*. Gulf Publishing Company, Paris.

Trinet, F., Heim, R., Amar, D., Chang, H. T., and Rittmann, B. E. (1991) Study of biofilm and fluidization of bioparticles in a three-phase liquid fluidized-bed reactor. *Wat. Sci. Tech.*, **33,** 1347.

Trulear, M. G. and Characklis, W. G. (1982) Dynamics of biofilm processes. *J. Water. Pollut. Control Fed.*, **54**, 1288.

Turakhia, M. H. and Characklis, W. G. (1989) Activity of *Pseudomonas aeruginosa* in biofilms: Effect of calcium. *Biotechnol. Bioeng.*, **33**, 406.

Tyagi, R. D. and Du, Y. G. (1992) Kinetic model for the effects of heavy metals on activated sludge process using neural networks. *Environ. Technol.*, **13**, 883.

Tzeng, J.-W. (1991) *Study of fluidized bed reactor: Fluid dynamics and bioreactor application*. Ph. D. Thesis, Ohio State University.

van Can, H .J. L., Hellinga, C., Luben, K. A. M., Heijnen, J. J., and Braake, H. A. B. (1997) Strategy for dynamic process modeling based on neural networks in macroscopic balances. *AICHE J.*, **42**, 3403.

Villadsen, J. V. and Stewart, W. E. (1967) Solution of boundary value problems by orthogonal collocation. *Chem. Engng. Sci.*, **22**, 1483.

Wang, L.-X. and Mendel, J. M. (1992) Fuzzy basis functions, universal approximation, and orthogonal least squares learning. *IEEE. Trans. Neural Networks*, **3**, 807.

Wang, S.-C. P. and Tien, C. (1984) Bilayer film model for the interaction between adsorption and bacterial activity in granular activated carbon columns, *AICHE J.*, **30**, 786.

Wanner, O. and Gujer, W. (1986) A multispecies biofilm model. *Biotechnol. Bioeng.*, **28**, 314.

Wen, C. Y. and Yu, Y. H. (1966) Mechanics of fluidization. *Chem. Prog. Symp. Ser.*, No. **62**, 100.

Werbos, P. (1974) *Beyond Regression: new tools for Prediction and Analysis in the Behavioral Sciences.* Ph.D. Thesis, Harvard University, Cambridge, MA.

Whitely, D., Starkweather, T., and Bogart, C. (1990) Genetic algorithms and neural networks: optimizing connections and connectivity. *Parallel Computing*, **14**, 347.

Wijffels, R. H., de Gooijer, C. D., Kortekass, S., and Tramper, J. (1991) Growth and substrate consumption of *Nitrobater agilis* cells immobilized in carrageenan: Part 1. dynamic modeling. *Biotechnol. Bioeng.*, **38**, 224.

Wijffels, R. H. and Tramper, J. (1989) Performance of growing *Nitrosomonas europaea* cells immobilized in k-carrageenan. *Appl. Microbiol. Biotechnol.*, **32**, 108.

Williamson, K. J. and McCarthy, P. L. (1976) A model of substrate utilization by bacterial films. *J. Water Pollut. Control Fed.*, **48**, 9.

Wisecarver, K. D. and Fan, L.- S. (1989) Biological phenol degradation in a gas-liquid-solid fluidized bed bioreactor. *Biotechnol. Bioeng.*, **33**, 1029.

Ye, K., Jin, S., and Shimizu, K. (1994) Fuzzy neural network for the control of high density cultivation of recombinant *Escherichia coli. J. Ferment. Bioeng.*, **77**, 663.

Ying, W. C. and Weber Jr, W. J. (1979) Biophysicochemical adsorption model system for wastewater treatment. *J. Water Pollut. Control Fed.*, **51**, 2661.

Zang, J.-P., Epstein, N., Grace, J. R., and Zhu, J. (1995) Minimum liquid fluidization velocity of gas-liquid fluiidzed beds. *Trans. IChem.*, **73**, 347.

Zang, T. C. and Bishop, P. L. (1994a) Structure, activity and composition of biofilms. *Wat. Sci. Tech.*, **29,** 335.

Zang, T. C. and Bishop, P. L. (1994b) Evaluation of tortuosity factors and effective diffusivities in biofilms. *Wat. Res.*, **28**, 2279.

Zhao, H., Hao, O. J., McAvoy, T. J., and Chang C.-H. (1997) Modeling nutrient dynamics in sequencing batch reactor. *J. Envir. Engng. ASCE*, **123**, 311.

Zimmermann, H.-J. (1990) *Fuzzy Sets Theory - and Its Application.* Kluwer Academic Publishers.

Appendix A

Main FORTRAN program for a TPFBBR used in Chapter 4.

```
C*********************************************************************C
C*                                                                   *C
C*     COMPUTER PROGRAM FOR THREE-PHASE FLUIDIZED BED                *C
C*     BIOFILM REACTOR TO SOLVE HYDRODYNAMIC MODEL,                  *C
C*     BIOFILM GROWTH MODEL,REACTION-DIFFUSION MODEL,                *C
C*     AND REACTOR MODEL HAVING NONLINEAR TIME-DEPENDENT             *C
C*     PARTIAL DIFFERENTIAL EQUATIONS                                *C
C*                                                                   *C
C*                                                                   *C
C*     THIS PROGRAM USE THE FOLLOWING SUBROUTINE NAMED               *C
C*                  1.INTODE (FOR STIFF PROBLEM)                     *C
C*                        STIFF                                      *C
C*                        JACOBN                                     *C
C*                        LUSKSB                                     *C
C*                        LUDCMP                                     *C
C*                  2.ONEPDE INTERFACE                               *C
C*                        EQUATION                                   *C
C*                        BOUNDARY                                   *C
C*                        DIFFUSION                                  *C
C*                                                                   *C
C*    ONEPDE IS AN INTERFACE SUBSROUTINE WHICH USES CENTERED         *C
C*    APPROXIMATIONS TO CONVERT ONE-DIMENSIONAL SYSTEMS OF           *C
C*    PARTIAL DIFFERENCIAL EQUATIONS INTO A SYSTEM OF ORDINARY       *C
C*    DIFFERENTIAL EQUATIONS. THIS ROUTINE IS INTENDED TO BE         *C
C*    USED WITH A ROUBST ODE INTEGRATOR, INTODE.                     *C
C*                                                                   *C
C*                                                                   *C
C*          PROGRAMMER : YOON-SEOK HONG                              *C
C*                  Institute of Technology & Engineering            *C
C*                        Massey University                          *C
C*                                                                   *C
C*********************************************************************C
C
C  MAIN PROGRAM FOR A THREE-PHASE FBBR.
C  THIS PROGRAM IMPLEMENTS A BIOFILM GROWTH,REACTOR, REACTION-
C DIFFUSION MODELS.
C
    INTEGER KMAXX,NMAX,NEQN
    PARAMETER (KMAXX=300,NMAX=500)
    INTEGER I,KMAX,KOUNT,NBAD,NOK,NPDE,NPTS,KODE,IK
    REAL EPS,H1,HMIN,X1,X2,X,Y
C
    COMMON /PROB/KODE
    COMMON /BIO1/DIASP,DIABP,DENBP,AP,SH,DM
    COMMON /HYDRO1/H,VOLR,DIAR,DENSP,DENL,DENG,VISL,VISG
    COMMON /HYDRO2/RGA,PE,UMF2,UMF3,FR,ARL
    COMMON /SIMUL1/UG,EL,RKL,ARGH,RSH,RLH,CIN
    COMMON /SIMUL2/UMAX,EKS,RKS,RKO,DIFS,YXS,YXO,EFB,RLF,DENB,AB,RKI
    COMMON /PATH/ KOUNT,X(KMAXX),Y(NMAX,KMAXX)
    COMMON NRHS
    COMMON /MESH/XX(15)
    COMMON /COORD/ICORD
    COMMON /SIZES/NPDE,NPTS
    DIMENSION YSTART1(15),YSTART2(15),YSTART3(15)
    EXTERNAL INFILE,HYDRODYNAMICS,ONEPDE,STIFF
C
    DO 1000 IK=1,3
    KODE=IK
    IF(KODE.EQ.1) GO TO 700
    IF(KODE.EQ.2) GO TO 800
    IF(KODE.EQ.3) GO TO 900
```

```
 700 CONTINUE
C
C    DEFINE INPUT PARAMETRS OF ODE INTEGRATOR FOR REACTOR MODEL
C    (INITIAL TIME,FINAL TIME, TIME STEPSIZE et,al.)
C
     NRHS=0
     X1=0.0
     X2=100.
     EPS=1.0E-4
     H1=0.05
     HMIN=0.0
     KMAX=100
     DXSAV1=(X2-X1)/50.0
C
C    DEFINE INPUT PARAMETRS OF ONEPDE FOR REACTOR MODEL
C
     ICORD=0
     NPDE=1
     NPTS=15
     NVAR=15
C
C    OPEN OUTPUT FILE
C
     CALL INFILE(X1,X2,EPS,H1,HMIN,KMAX)
C
C    EXECUTE THE HYDRODYNAMIC MODEL OF THREE-PHASE FBBR.
C
     CALL HYDRODYNAMICS
C
C    DEFINE MESH AND INITIAL CONDITION FOR REACTOR
C
     DX1=72./FLOAT(NPTS-1)
     DO 50 K=1,NPTS
       XX(K)=FLOAT(K-1)*DX1
       C0=0.15
  50 YSTART1(K)=C0
C
     CALL INTODE(YSTART1,NVAR,X1,X2,EPS,H1,HMIN,NOK,
    *          NBAD,ONEPDE,STIFF,DXSAV1,KMAX)
C
C    PRINT OUT OF RESULTS-TABLE HEADING FOR REACTOR MODEL.
C
     WRITE(6,'(/5X,A,T52,I6)') 'SUCCESSFUL STEPS:          ',NOK
     WRITE(6,'(5X,A,T52,I6)') 'BAD STEPS:                ',NBAD
     WRITE(6,'(5X,A,T52,I6)') 'FUNCTION EVALUATIONS:      ',NRHS
     WRITE(6,'(5X,A,T52,I6)') 'STORED INTERMEDIATE VALUES: ',KOUNT
     WRITE(6,38)UL,UG,CIN,C0,RLF,RSH
     DX1=72./FLOAT(NPTS-1)
     DO 64 K=1,NPTS
       XX(K)=FLOAT(K-1)*DX1
  64 CONTINUE
       WRITE(6,39)XX(1),XX(2),XX(3),XX(4),XX(5),XX(6),XX(7),XX(8)
     DO 65 I=1,KOUNT
       WRITE(6,40)X(I),Y(1,I),Y(2,I),Y(3,I),Y(4,I),Y(5,I),Y(6,I),
    *          Y(7,I),Y(8,I)
  65 CONTINUE
     WRITE(6,41)
     WRITE(6,42)
     WRITE(6,43)XX(9),XX(10),XX(11),XX(12),XX(13),XX(14),XX(15)
     DO 66 I=1,KOUNT
       WRITE(6,44)X(I),Y(9,I),Y(10,I),Y(11,I),Y(12,I),Y(13,I),Y(14,I),
    *          Y(15,I)
  66 CONTINUE
     WRITE(6,45)
C
  38 FORMAT(//18X,34H**** SIMULATION RESULTS TABLE ****,//
    *3X,40HTHE SUPERFICIAL LIQUID VELOCITY     UL=,F7.4/
    *3X,40HTHE SUPERFICIAL GAS VELOCITY        UG=,F7.4/
    *3X,40HTHE INLET SUBSTRATE CONCENTRATION   CIN=,F7.4/
    *3X,40HTHE INTIAL SUBSTRATE CONCENTRATION,  C0=,F7.4/
    *3X,40HTHE BIOFILM THICKNESS              RLF=,F7.4/
```

```
     *3X,40HTHE SOLID HOLDUP                RSH=,F7.4
     *//14X,51HCONCENTRATIONS AS A FUNCTION OF TIME AND BED HEIGHT/
     *2X,75(1H-)/4X,5H TIME,2X,1HI,1X,22H    DISTANCE FROM THE ,
     *1X,17HBOTTOM OF REACTOR)
  39 FORMAT(/10X,1HI,1X,8F8.2/2X,75(1H-))
  40 FORMAT(/2X,F7.3,1X,1HI,1X,8F8.4)
  41 FORMAT(/2X,75(1H-))
  42 FORMAT(//14X,51HCONCENTRATIONS AS A FUNCTION OF TIME AND BED HEIG
     *HT/2X,75(1H-)/4X,5H TIME,2X,1HI,1X,22H    DISTANCE FROM THE ,
     *1X,17HBOTTOM OF REACTOR)
  43 FORMAT(/11X,1HI,1X,8F8.4/2X,75(1H-))
  44 FORMAT(/2X,F8.3,1X,1HI,1X,8F8.5)
  45  FORMAT(/2X,75(1H-))
C*********************************************************************C
C
GO TO 1000
  800 CONTINUE
C
C    DEFINE INPUT PARAMETRS OF ONEPDE FOR REACTION-DIFFUSION MODEL
C
     ICORD=2
     NPDE=1
     NPTS=15
     NVAR=15
C
C    DEFINE INPUT PARAMETRS OF INTEGRATOR FOR REACTION-DIFFUSION MODEL
C    (INITIAL TIME,FINAL TIME, TIME STEPSIZE et al.)
C
     NRHS=0
     X12=0.0
     X22=50.
     EPS2=1.0E-4
     H12=0.003
     HMIN2=0.0
     KMAX=100
     DXSAV2=(X22-X12)/50.0
C
C    DEFINE MESH AND INITIAL CONDITION FOR REACTION-DIFFUSION MODEL
C
     DX2=0.00629/FLOAT(NPTS-1)
     DO 100 I=1,NPTS
       XX(I)=FLOAT(I-1)*DX2
       B0=0.0
  100 YSTART2(I)=B0
C
     CALL INTODE(YSTART2,NVAR,X12,X22,EPS2,H12,HMIN2,NOK,
    *          NBAD,ONEPDE,STIFF,DXSAV2,KMAX)
C
     WRITE(6,'(/5X,A,T52,I6)') 'SUCCESSFUL STEPS:          ',NOK
     WRITE(6,'(5X,A,T52,I6)') 'BAD STEPS:                ',NBAD
     WRITE(6,'(5X,A,T52,I6)') 'FUNCTION EVALUATIONS:      ',NRHS
     WRITE(6,'(5X,A,T52,I6)')'STORED INTERMEDIATE VALUES:  ',KOUNT
C
C    PRINT OUT OF RESULTS-TABLE HEADING FOR REACTION-DIFFUSION MODEL
C
     WRITE(6,140)
     DX2=0.00629/FLOAT(NPTS-1)
     DO 200 K=1,NPTS
       XX(K)=FLOAT(K-1)*DX2
  200 CONTINUE
     WRITE(6,150)XX(15),XX(14),XX(13),XX(12),XX(11),XX(10),XX(9)
     DO 210 I=1,KOUNT
     WRITE(6,350)X(I),Y(15,I),Y(14,I),Y(13,I),Y(12,I),Y(11,I),
    *           Y(10,I),Y(9,I)
  210 CONTINUE
     WRITE(6,400)
     WRITE(6,401)
     WRITE(6,402)XX(8),XX(7),XX(6),XX(5),XX(4),XX(3),XX(1)
     DO 220 I=1,KOUNT
     WRITE(6,403)X(I),Y(8,I),Y(7,I),Y(6,I),Y(5,I),Y(4,I),Y(3,I),Y(1,I)
  220 CONTINUE
```

```
      WRITE(6,404)
C
  140 FORMAT(//14X,48HCONCENTRATIONS AS A FUNCTION OF TIME AND BIOFILM/
     *2X,75(1H-)/4X,5H TIME,2X,1HI,1X,22H    DISTANCE FROM THE ,
     *1X,24HBIOFILM-LIQUID INTERFACE)
  150 FORMAT(/11X,1HI,1X,8F8.4/2X,75(1H-))
  350 FORMAT(/2X,F8.3,1X,1HI,1X,8F8.5)
  400 FORMAT(/2X,75(1H-))
  401 FORMAT(//14X,48HCONCENTRATIONS AS A FUNCTION OF TIME AND BIOFILM/
     *2X,75(1H-)/4X,5H TIME,2X,1HI,1X,22H    DISTANCE FROM THE ,
     *1X,24HBIOFILM-LIQUID INTERFACE)
  402 FORMAT(/11X,1HI,1X,8F8.4/2X,75(1H-))
  403 FORMAT(/2X,F8.3,1X,1HI,1X,8F8.5)
  404 FORMAT(/2X,75(1H-))
C*****************************************************************C
C
  900 CONTINUE
C
C     DEFINE INPUT PARAMETRS FOR BIOFILM GROWTH MODEL
C
      ICORD=2
      NPDE=1
      NPTS=15
      NVAR=15
C
C     DEFINE INPUT PARAMETRS OF INTEGRATOR FOR BIOFILM GROWTH MODEL
C     (INITIAL TIME,FINAL TIME, TIME STEPSIZE et al.)
C
      NRHS=0
      X13=0.0
      X23=15.
      EPS3=1.0E-4
      H13=0.1
      HMIN3=0.0
      KMAX=100
      DXSAV3=(X22-X12)/50.0
C
C     DEFINE MESH AND INITIAL CONDITION FOR BIOFILM GROWTH MODEL
C
      DX3=0.006/FLOAT(NPTS-1)
      DO 405 I=1,NPTS
        XX(I)=FLOAT(I-1)*DX3
        B10=0.001
  405 YSTART3(I)=B10
C
      CALL ODE(YSTART3,NVAR,X13,X23,EPS3,H13,HMIN3,NOK,
     *          NBAD,GROWTH,STIFF,DXSAV3,KMAX)
C
      WRITE(6,'(/5X,A,T52,I6)') 'SUCCESSFUL STEPS:          ',NOK
      WRITE(6,'(5X,A,T52,I6)') 'BAD STEPS:                ',NBAD
      WRITE(6,'(5X,A,T52,I6)') 'FUNCTION EVALUATIONS:      ',NRHS
      WRITE(6,'(5X,A,T52,I6)') 'STORED INTERMEDIATE VALUES: ',KOUNT
C
C     PRINT OUT OF RESULTS-TABLE HEADING FOR BIOFILM GROWTH MODEL
C
      WRITE(6,406)
      DX3=0.00629/FLOAT(NPTS-1)
      DO 407 K=1,NPTS
        XX(K)=FLOAT(K-1)*DX3
  407 CONTINUE
      WRITE(6,408)XX(1),XX(2),XX(3),XX(4),XX(5),XX(6),XX(7)
      DO 409 I=1,KOUNT
      WRITE(6,410)X(I),Y(1,I),Y(2,I),Y(3,I),Y(4,I),Y(5,I),
     *            Y(6,I),Y(7,I)
  409 CONTINUE
      WRITE(6,411)
      WRITE(6,412)
      WRITE(6,413)XX(8),XX(9),XX(10),XX(11),XX(12),XX(13),XX(14)
      DO 414 I=1,KOUNT
      WRITE(6,415)X(I),Y(8,I),Y(9,I),Y(10,I),Y(11,I),Y(12,I),Y(13,I),
     *         Y(14,I)
```

```
 414 CONTINUE
     WRITE(6,416)
C
 406 FORMAT(//14X,48HCONCENTRATIONS AS A FUNCTION OF TIME AND BIOFILM/
    *2X,75(1H-)/4X,5H TIME,2X,1HI,1X,22H     DISTANCE FROM THE ,
    *1X,24HBIOFILM-LIQUID INTERFACE)
 408 FORMAT(/11X,1HI,1X,8F8.4/2X,75(1H-))
 410 FORMAT(/2X,F8.3,1X,1HI,1X,8F8.5)
 411 FORMAT(/2X,75(1H-))
 412 FORMAT(//14X,48HCONCENTRATIONS AS A FUNCTION OF TIME AND BIOFILM/
    *2X,75(1H-)/4X,5H TIME,2X,1HI,1X,22H     DISTANCE FROM THE ,
    *1X,24HBIOFILM-LIQUID INTERFACE)
 413 FORMAT(/11X,1HI,1X,8F8.4/2X,75(1H-))
 415 FORMAT(/2X,F8.3,1X,1HI,1X,8F8.5)
 416 FORMAT(/2X,75(1H-))
C*********************************************************************C
C            END OF MAIN PROGRAM FOR A TPFBBR                         C
C*********************************************************************C
 1000 CONTINUE
     STOP
     END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C        OPEN INPUT AND OUTPUT FILES FOR PARAMETRS USED IN SIMULATION      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

     SUBROUTINE INFILE(X1,X2,EPS,H1,HMIN,KMAX)
     INTEGER KMAXX,NMAX,NEQN
     PARAMETER (KMAXX=300,NMAX=500)
     INTEGER KMAX,KOUNT,NRHS,NPDE,NPTS
     REAL H1,HMIN,X1,X2,X,Y
     COMMON /PATH/ KOUNT,X(KMAXX),Y(NMAX,KMAXX)
     COMMON NRHS
     COMMON /MESH/XX(15)
     COMMON /COORD/ICORD
     COMMON /SIZES/NPDE,NPTS
     EXTERNAL PDE,STIFF,SETPC
C
     CALL SETPC
C
C    HEADING PRINTOUT
C
     WRITE(6,10)
     WRITE(6,11)
     WRITE(6,13)
C
C    WRITING INPUT PARAMETRS OF ONEPDE FOR REACTOR MODEL
C
     WRITE(6,14)
     WRITE(6,15)NRHS,X1,X2
     WRITE(6,16)H1,HMIN,KMAX
     WRITE(6,17)ICORD,NPDE,NPTS

C
  10 FORMAT(65(1H*)/1H*,63X,1H*/
    *      1H*,18X,25HNUMERICAL SOLUTION OF THE,20X,1H*/
    *      1H*,13X,38HMATHEMATICAL MODELLING FOR THREE-PHASE,
    *      12X,1H*/1H*,17X,29HFLUIDIZED BED BIOFILM REACTOR,
    *      17X,1H*/1H*,63X,1H*/1H*,63X,1H*)
  11 FORMAT(1H*,63X,1H*/
    *      1H*,15X,33HTHIS PROGRAM USE ONEPDE INTERFACE,15X,1H*/
    *      1H*,12X,40HAND A ODE's INTEGRATOR FOR STIFF PROBLEM,
    *      11X,1H*/1H*,63X,1H*/1H*,63X,1H*)
  13 FORMAT(1H*,18X,27HPROGRAMMER : YOON-SEOK HONG,18X,1H*/
    *      1H*,63X,1H*/65(1H*))
  14 FORMAT(//5X,37H*** INPUT DATA FOR ODE INTEGRATOR ***//)
  15 FORMAT(5X,43HSTEPSIZE TO THE INITIAL TRIAL VALUE  NRHS=,I4/
    *      5X,43HINITIAL TIME STEP                  T1=,F10.3/
    *      5X,43HFINAL TIME STEP                    T2=,F10.3)
  16 FORMAT(5X,43HA GUSSED FIRST STEPSIZE             H1=,E13.5/
```

```
     *     5X,43HTHE MINIMUM ALLOWED STEPSIZE         HMIN=,E13.5/
     *     5X,43HTHE NUMBER OF STEPS TO BE STORED     KMAX=,I5)
  17 FORMAT(//5X,35H*** INPUT PARAMETERS FOR ONEPDE ***//
     *     5X,43HCOORDINATE SYSTEM INDICATOR          ICORD=,I4/
     *     5X,43HNUMBER OF PDE's                      NPDE=,I6/
     *     5X,43HNUMBER OF SPATIAL GRID POINTS        NPTS=,I6)
  25 FORMAT(2X,3(F6.3,2X))
  26 FORMAT(/2X,E11.4,2X,F6.3,2X,E11.4)
  27 FORMAT(/2X,3(E11.4,2X))
  28 FORMAT(/2X,2(E11.4))
  30 FORMAT(2X,2(E11.4,2X))
  31 FORMAT(2X,3(F7.4,3X))
  32 FORMAT(2X,E11.4,2X,F7.1,2X,F4.1)
     RETURN
     END
CC
CC


     SUBROUTINE SETPC
CC
     CHARACTER OUTFIL*12
     DATA OUTFIL/'           '/
     WRITE(*,'(/1X,A\)')' ENTER OUTPUT FILE NAME (e.g.,OUTFILE.DAT): '
     READ(*,'(BN,A)')OUTFIL
C
C    OPEN UNIT 6 FOR OUTPUT
C
     OPEN(6,FILE=OUTFIL)
     WRITE(*,504)
  504 FORMAT(///,10X,'*** PLEASE WAIT, RESULTS ARE DEVELOPING ***',/)
     RETURN
     END
CC
CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                  HYDRODYNAMIC FOR THREE-PHASE FBBBR                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     SUBROUTINE HYDRODYNAMICS
     COMMON /BIO1/DIASP,DIABP,DENBP,AP,SH,DM
     COMMON /HYDRO1/H,VOLR,DIAR,DENSP,DENL,DENG,VISL,VISG
     COMMON /HYDRO2/RGA,PE,UMF2,UMF3,FR,ARL
     COMMON /SIMUL1/UL,UG,EL,RKLA,RGH,RSH,RLH,CIN
     COMMON /SIMUL2/UMAX,EKS,RKS,RKO,DIFS,YXS,YXO,EFB,RLF,DENB,AB,RKI
C
C    INPUT PARAMETRS FOR REACTION-DIFFUSION MODEL
C
     UMAX=6.39E-5
     YXS=0.4
     YXO=0.354
     RKS=0.010948
     RKO=0.0001
     RKI=0.113
     EFB=0.064

     DIASP=0.0710
     RNP=224500.
     DIFS=2.75E-6
     DM=0.00000847

     RLF=0.023
     DENB=67.67
C
C    INPUT PARAMETERS FOR HYDRODYNAMICS
C
     UL=0.03
     UG=1.2

     H=72.
     VOLR=1144.
     DIAR=4.2
```

```
      DENSP=1.4
      DENL=1.
      DENG=0.0012
      VISL=0.01
      VISG=0.00018
      CIN=0.177
C
C     WRITE VALUE OF PARAMETERS FOR SIMULATION
C
      WRITE(6,40)
      WRITE(6,41)UMAX,YXS,YXO
      WRITE(6,42)RKS,RKO,RKI,EFB,RLF
      WRITE(6,43)DIASP,RNP
      WRITE(6,44)DIFS,DENB,DM
C
      WRITE(6,45)UL,UG
      WRITE(6,46)H,VOLR,DIAR
      WRITE(6,47)DENSP,DENL,DENG,VISL
C
C     DETERMINE BIOPARTICLE DENSITY
C
      DIABP=DIASP+RLF*2.
        DENBW=DENL+(DENB/(1000.*0.8))
      DENBP=DENBW+(DENSP-DENBW)*(DIASP/DIABP)*(DIASP/DIABP)*
     *     (DIASP/DIABP)
C
C     DETERMINE THE OVERALL GAS-LIQUID MASS TRANSFER COEFFICINT
C     USING THE CORRELATION OF SHAH et al.(AICHE. J. 28,353(1982)).
C
      RKLA=1.174*((UG/100.)**0.82)
C
C     CALCULATE GALLILEO NUMBER AND LIQUID ARCHIMEDES NUMBER.
C
      RGAP=(DIABP**3.)*(DENL**2.)
      RGA=(980.62*RGAP)/(VISL**2)
      ARL=((DIABP**3.)*DENL*(DENBP-DENL)*980.62)/VISL/VISL
C
C     CALCULATE FROUDE NUMBER
C
      FR=UG/SQRT(980.62*DIAR)
C
C     CALCULATE MINIMUM FLUIDIZATION VELOCITY OF TWO-PHASE FBBR
C     USING THE CORRELATION PRESENTED BY WEN & YU (1966).
C
      UMF2=(VISL/DIABP/DENL)*(SQRT((33.7*33.7)+(0.0408*ARL))-33.7)
C
C     CALCULATE MIN. FLUIDIZATION VELOCITY OF THREE-PHASE FBBR
C     USING GAS-PERTUBED LIQUID MODEL PRESENTED BY ZANG et,al.(1995).
C
c     PHIS=0.906
c     EMF=0.4287
c     ZETA=1.61*(UG**0.72)*(DIABP**0.168)*(DIAR**(-0.125))/EMF
c     AA=42.86*((1.-EMF)/PHIS)*((1.-EMF)/PHIS)
c     BB=0.5715*PHIS*EMF*EMF*EMF*(1.-ZETA)*(1.-ZETA)*(1.-ZETA)*ARL
c     CC=AA+BB
c     UMF3=(VISL/DIABP/DENL)*(SQRT(CC))-42.86*(1.-EMF)/PHIS
C
C     DETERMINE THE AXIA- DISPERSION COEFFICIENT USING KATO's et al.
C     CORRELATION FOR LOW LIQUID VELOCITY OR KIM & KIM's CORRELATION
C     FOR HIGH LIQUID VELOCITY.
C
      IF(UG.GE.5) GO TO 100
      WRITE(6,50)
      EL=UG*DIAR*(1.+8.*(FR**0.85))/(13.*FR)
      PE=UL*DIABP/EL
      IF(PE.GE.100)WRITE(6,51)
      WRITE(6,52)
   50 FORMAT(//2X,'KATOs CORREALATION IS CONSIDERED SINCE LIQUID',
     *' VELOCITY IS LOW.')
   51 FORMAT(/2X,'PECLET NUMBER IS HIGH, SO THE AXIAL DISPERSION',
     *' SHOULD NOT BE CONSIDERED.')
```

```
  52 FORMAT(//2X,'PECLET NUMBER IS LOW, SO THE AXIAL DISPERSION',
     *' SHOULD BE CONSIDERED.')
      GO TO 300
 100 WRITE(6,53)
  53 FORMAT(//5X,'Kim and Kim correlation could be considered',
     *' since liquid velocity is high.')
      PE=20.19*((DIABP/DIAR)**1.66)*((UL/(UL+UG))**1.03)
      EL=DIABP*UL/PE
      IF(PE.GE.1000)WRITE(6,51)
      WRITE(6,52)
C
C     DETERMINE LIQUID-SOLID MASS TRANSFER COEFFICIENT USING
C     THE CORRELATION DEVELOPED BY ARTERS & FAN (1986).
C
 300 REG=UG*DIABP*DENG/VISG
      SC=VISL/(DENL*DM)
      EKS=0.228*(DM/DIABP)*(1.+0.0826*(REG**0.623))*(RGA**0.323)*
     *    (SC**0.4)*(((DENBP-DENL)/DENL)**0.3)
      SH=EKS*0.5*DIABP/DIFS
      IF(SH.GE.300) WRITE(6,54)
      WRITE(6,55)
  54 FORMAT(//2X,'SHERWOOD NUMBER IS HIGH SUFFICIENT TO RENDER THE',
     *' EXTERNAL MASS TRANSFER RESISTANCE NEGLIBLE')
  55 FORMAT(//2X,'SHERWOOD NUMBER IS NOT HIGH SUFFICIENT TO RENDER ',
     *'THE EXTERNAL MASS TRANSFER RESISTANCE NEGLIBLE')
C
C     CALCULATE GAS HOLDUP USING THE CORRELATION PRESENTED BY
C     BEGOVICH & WATSON(1978).
C
      RGH=0.048*(UG**0.072)*(DIABP**0.168)*(DIAR**(-0.125))
C
C     DETERMINE SOLID HOLDUP AND LIQUID HOLDUP
C
      RSH=1.-0.371*(UL**0.271)*(UG**0.041)*((DENBP-DENL)**(-0.316))*
     *(DIABP**(-0.268))*(VISL**0.055)*(DIAR**(-0.033))
      RLH=1.-RGH-RSH
C
C     CALCULATE THE SPECIFIC SURFACE AREA OF BIOFILM-COVERED PARTICLE
C     AND OVERALL SURFACE AREA OF BIOPARTICLES IN REACTOR.

      AB=6.*RSH/DIABP
      AP=3.*VOLR*RSH/0.5/DIABP
C
C     WRITE RESULTS OF HYDRODYNAMIC MODELS
C
      WRITE(6,61)
      WRITE(6,62)DIABP,DENBP,RKLA,EKS,AB,AP
      WRITE(6,63)RGA,FR,ARL,UMF2,UMF3
      WRITE(6,64)SH,PE,EL,RGH,RSH,RLH
C
  40 FORMAT(//5X,44H*** VALUE OF PARAMETERS USED IN MATHEMATICAL
     *1X,9HMODEL ***,/3X,58(1H-))
  41 FORMAT(5X,30HMAXSIMUM SPECIFIC GROWTH
     */5X,45HRATE OF BIOMASS                    UMAX=,E10.3
     */5X,45HYIELD COEFFICIENT FOR SUBSTRATE    YXS=,F10.3
     */5X,45HYIELD COEFFICIENT FOR OXYGEN       YXO=,F10.3)
  42 FORMAT(5X,45HMONOD CONSTANT FOR SUBSTRATE       RKS=,E10.3
     */5X,45HMONOD CONSTANT FOR OXYGEN          RKO=,E10.3
     */5X,45HINHIBITION CONSTANT                RKI=,E10.3
     */5X,45HVOID FRACTION OF BIOFILM           EFB=,F10.3
     */5X,45HBIOFILM THICKNESS                  RLF=,F10.3)
  43 FORMAT(5X,45HDIAMETER OF SUPPORT PARTICLE      DIASP=,E10.3
     */5X,45HNUMBER OF BIOPARTICLES IN REACTOR     RNP=,E10.3)
  44 FORMAT(5X,24HDIFFUSION COEFFICIENT OF
     */5X,45HSUBSTRATE IN BIOFILM               DIFS=,E10.3
     */5X,45HBIOFILM DENSITY                    DENB=,E10.3
     */5X,45HMOLECULAR DIFFUSIVITY OF SUBSTRATE    DM=,E10.3)
  45 FORMAT(5X,45HSUPERFICIAL LIQUID VELOCITY       UL=,F10.3
     */5X,45HTHE SUPERFICIAL GAS VELOCITY       UG=,F10.3)
  46 FORMAT(5X,45HREACTOR HEIGHT                    H=,F10.3
     */5X,45HREACTOR VOLUME                     VOLR=,F10.3
```

```
     */5X,45HDIAMETER OF REACTOR                DIAR=,F10.3)
    47 FORMAT(5X,45HTHE SUPPORT PARTICLE DENSITY      DENSP=,F10.3
     */5X,45HTHE LIQUID DENSITY                DENL=,F10.3
     */5X,45HTHE GAS DENSITY                   DENG=,F10.3
     */5X,45HTHE LIQUID VISCOSITY              VISL=,F10.3
     */3X,58(1H-))
C
    61 FORMAT(//12X,38H*** RESULTS OF HYDRODYNAMIC MODELS ***
     */3X,58(1H-))
    62 FORMAT(5X,45HTHE DIAMETER OF BIOPARTICLES      DIABP=,F10.3
     */5X,45HTHE BIOPARTICLE DENSITY           DENBP=,F10.3
     */5X,45HTHE GAS-LIQUID MASS TRANSFER COEFICIENT RKLA=,F10.3
     */5X,45HTHE EXTERNAL MASS TRANSFER COEFFICIENT  EKS=,E10.3
     */5X,45HTHE SPECIFIC SURFACE AREA OF BIOPARTICLE  AB=,F10.3
     */5X,45HTOTAL SURFACE AREA OF BIOFILM PHASE     AP=,F10.3)
    63 FORMAT(5X,45HTHE GALLILEO NUMBER             RGA=,F10.3
     */5X,45HTHE FROUDE NUMBER                 FR=,F10.3
     */5X,45HTHE LIQUID ARCHIMEDES NUMBER      ARL=,F10.3
     */5X,45HTHE MIN. 2-PHASE FLUIDIZATION VELOCITY  UMF2=,F10.3
     */5X,45HTHE MIN. 3-PHASE FLUIDIZATION VELOCITY  UMF3=,F10.3)
    64 FORMAT(5X,45HTHE SHERWOOD NUMBER               SH=,F10.3
     */5X,45HTHE PECLET NUMBER                 PE=,E10.3
     */5X,45HTHE AXIAL DISPERSION COEFFICIENT     EL=,E10.3
     */5X,45HTHE GAS HOLDUP                    RGH=,F10.3
     */5X,45HTHE SOLID HOLDUP                  RSH=,F10.3
     */5X,45HTHE LIQUID HOLDUP                 RLH=,F10.3
     */3X,58(1H-))
      RETURN
      END
```

Appendix B

Some routines for multilayer feedforward neural network (MFNN) trained by extended Kalman filtering (EKF) learning algorithm used in Chapter 5.

/* File name: NNEKF.C */

```
/*
 Main Program for a multilayer feedforward neural network (MFNN)
 trained by extended Kalman filter(EKF) learning algorithm.

 Yoon-Seok Hong
 Institute of Technology & Engineering
 Massey University
 New Zealand
 E-Mail: Y.S.Hong@massey.ac.nz

*/

# include <stdio.h>                    /*FILE*/
# include <string.h>                   /*strcmp*/
# include <math.h>              /*sqrt*/
# include <time.h>                     /*clock(), CLK_TCK*/
# ifndef CLOCKS_PER_SEC
#  ifndef CLK_TCK
#   define CLOCKS_PER_SEC 1000000
#  else
#   define CLOCKS_PER_SEC CLK_TCK
#  endif
# endif

# include "comnn.h"
# define   scanf1(i0,il )      while(scanf(i0,il ) != 1)            \
                                                {                                \
                                                 printf("\7error in scanf, retry: ");\
                                                 scanf("%*s");                   \
                                                }

# define dprintf0(x,y)                { fprintf(x,y);      printf(y);      }
# define dprintf1(x,y,a)              { fprintf(x,y,a);    printf(y,a);    }
# define dprintf2(x,y,a,b)            { fprintf(x,y,a,b);  printf(y,a,b);  }
# define dprintf3(x,y,a,b,c)          { fprintf(x,y,a,b,c); printf(y,a,b,c); }
# define dprintf4(x,y,a,b,c,d)        { fprintf(x,y,a,b,c,d); printf(y,a,b,c,d); }
# define dprintf5(x,y,a,b,c,d,e) { fprintf(x,y,a,b,c,d,e); printf(y,a,b,c,d,e);}

# define abs(x)       ((x) >= 0 ? (x) : (-(x)))
# define min(x,y)     ( (x) <= (y) ? (x) : (y) )
# define max(x,y)     ( (x) >= (y) ? (x) : (y) )
# define pii(i)       (i) * ( 3 + i ) / 2

void main()
{
 COMNN nn[1];
 FILE *fp,*fp1;
 int i,j,j1,j2,k,jout;
 char method[10];
 clock_t anfang,zeit,maxzeit;
 float zeitsec,maxzeitsec;
 float sys_err,fehl;
 float eta,alpha;
 float minin,maxin,mintar,maxtar,mintest,maxtest;
 extern void net_read(FILE*,COMNN*);
 extern void bp(FILE*,COMNN*,clock_t,float,float);
 extern void bp_ekf(FILE*,COMNN*,clock_t);
```

```
extern void netz_out(float**,float***,int*,int);

fp = fopen("ekfnet.log","w");     /*Documentation file*/

dprintf0(fp,"                                    \n")
dprintf0(fp,"                                    \n")
dprintf0(fp,"  Neural Network trained by EKF Learning      \n")
dprintf0(fp,"                                    \n")
dprintf0(fp,"  March 10, 1998                    \n")
dprintf0(fp,"                                    \n")
dprintf0(fp,"  Intitute of Technology & Engineering      \n")
dprintf0(fp,"  Private Bag 11222                 \n")
dprintf0(fp,"  Palmerston North                  \n")
dprintf0(fp,"  New Zealand                       \n")
dprintf0(fp,"  E-Mail: Y.S.Hong@massey.ac.nz     \n")
dprintf0(fp,"                                    \n")
dprintf0(fp,"                                    \n")

/* Read network file and traing data, and dispaly*/

net_read(fp,nn);        /*auch Testdaten*/

dprintf1(fp,"\nTraining of %d patterns\n",nn->npatterns)
dprintf0(fp,"Net structure  ");
for(i=0; i<= nn->nlayers; i++)    dprintf1(fp,"%2d ",nn->layer_size[i])
dprintf1(fp," = %d weights\n",nn->nweights)

/* Normalisation of data*/

        for(k=0; k<nn->npatterns_test; k++)
        {           /* Normalisation of testing data */
         for (i=0; i<nn->layer_size[0]; i++)   /* normalise -> input */
            nn->in_test[k][i] =
                              ( nn->in_test[k][i] - nn->norm_in[i][0])/
                              ( nn->norm_in[i][1] - nn->norm_in[i][0]);
                                 /* normalise -> target */
         for (i=0; i<nn->layer_size[nn->nlayers]; i++)
            nn->target_test[k][i] = (nn->target_test[k][i]
                                          - nn->norm_tar[i][0]) /
                             (nn->norm_tar[i][1] - nn->norm_tar[i][0]);
        }

        for(k=0; k<nn->npatterns; k++)
        {           /* Normalisation of Training data */
         for (i=0; i<nn->layer_size[0]; i++)    /* normalise -> input */
            nn->in[k][i] = (nn->in[k][i]-nn->norm_in[i][0])/
                              ( nn->norm_in[i][1] - nn->norm_in[i][0]);
                           /* normalise -> target */
         for (i=0; i<nn->layer_size[nn->nlayers]; i++)
            nn->target[k][i] = (nn->target[k][i] - nn->norm_tar[i][0]) /
                              (nn->norm_tar[i][1] - nn->norm_tar[i][0]);

        }

/*Checking of normalisation*/

        maxin = -10.;
        minin= 10.;
        maxtest = -10.;
        mintest = 10.;

        for(k=0; k<nn->npatterns_test; k++)
        {
          for (i=0; i<nn->layer_size[0]; i++)
          {
                if(nn->in_test[k][i] < minin)
                                    minin = nn->in_test[k][i];
                if(nn->in_test[k][i] > maxin)
                                    maxin= nn->in_test[k][i];
          }
```

```
            for (i=0; i< nn->layer_size[nn->nlayers]; i++)
            {
                    if(nn->target_test[k][i] < mintest)
                                    mintest = nn->target_test[k][i];
                    if(nn->target_test[k][i] > maxtest)
                                    maxtest = nn->target_test[k][i];
            }
    }

    if(nn->npatterns_test)
        {
      if((minin< -1.5) || (maxin > 1.5) || (mintest< .05) || (maxtest> .95))
            dprintf0(fp,"\007")
          dprintf4(fp,
            "Min/max input %1.3f %1.3f;  target %1.3f %1.3f for test\n",
                                                    minin,maxin,mintest,maxtest)
        }

        maxin = -10.;
        minin = 10.;
        maxtar = -10.;
        mintar = 10.;

        for(k=0; k< nn->npatterns; k++)
        {
          for (i=0; i< nn->layer_size[0]; i++)
          {
                    if(nn->in[k][i] < minin) minin = nn->in[k][i];
                    if(nn->in[k][i] > maxin) maxin = nn->in[k][i];
          }

          for (i=0; i< nn->layer_size[nn->nlayers]; i++)
          {
                    if(nn->target[k][i] < mintar) mintar = nn->target[k][i];
                    if(nn->target[k][i] > maxtar) maxtar = nn->target[k][i];
          }
        }
    if((minin< -1.5) || (maxin > 1.5) || (mintar< .05) || (maxtar> .95))
            dprintf0(fp,"\007")
    dprintf4(fp,
            "Min/max input %1.3f %1.3f;  target %1.3f %1.3f for training\n",
                                                    minin,maxin,mintar,maxtar)

/* Calculation before training */

        sys_err = 0.0;
        for(k=0; k< nn->npatterns_test; k++)
        {
          for(j=0; j< nn->layer_size[0]; j++)
                        nn->state[0][j] = nn->in_test[k][j];
          for(jout=0; jout< nn->layer_size[nn->nlayers]; jout++)
          {
                    netz_out(nn->state,nn->weight,nn->layer_size,
                        nn->nlayers);
                    sys_err += (nn->target_test[k][jout]
                                    - nn->state[nn->nlayers][jout]) *
                      (nn->target_test[k][jout]
                                    - nn->state[nn->nlayers][jout]);
          }
        }
        if(nn->npatterns_test)
        {
          fehl = 2. * nn->norm_tar[0][1] *
                        sqrt(sys_err /
                          nn->npatterns_test /
                          nn->layer_size[nn->nlayers]);
          dprintf1(fp,
      "Rms-error before training          %9.6f for test data\n",fehl)
          dprintf1(fp,
            "Error according to PROBEN1 notation  %9.3f for test data\n",
                                    sys_err / nn->npatterns_test /
```

```
                                        nn->layer_size[nn->nlayers] *
                                        100 * (maxtest - mintest));
        }

        sys_err = 0.0;
        for(k=0; k< nn->npatterns; k++)
        {
          for(j=0; j<nn->layer_size[0]; j++)
                        nn->state[0][j] = nn->in[k][j];
          for(jout=0; jout< nn->layer_size[nn->nlayers]; jout++)
          {
                netz_out(nn->state,nn->weight,nn->layer_size,
                        nn->nlayers);
                sys_err += (nn->target[k][jout]
                                - nn->state[nn->nlayers][jout]) *
                          (nn->target[k][jout]
                                - nn->state[nn->nlayers][jout]);
          }
        }
        fehl = 2. * nn->norm_tar[0][1] *
                        sqrt(sys_err /
                          nn->npatterns /
                          nn->layer_size[nn->nlayers]);
        dprintf1(fp,
    "Rms-error before training      %9.6f for training data\n",fehl)
    dprintf1(fp,
    "Error according   %9.3f for training data\n",
                                        sys_err / nn->npatterns /
                                        nn->layer_size[nn->nlayers] *
                                        100 * (maxtar - mintar));


/* Training and error calculation */

        nn->i_iter = 0;                 /*Counting the No. of iteration*/
        nn->n_iter = 0;

        strcpy(method,"");

        zeit = 0.;

        while(1)
        {
         if(!strcmp(method,"bp") || !strcmp(method,"BP"))
         {
          dprintf0(fp,"Training with backpropagation\n");

          printf("eta alpha    ");
          scanf1("%f",&eta);
          scanf1("%f",&alpha);
          fprintf(fp,"eta %f alpha %f\n",eta,alpha);

          anfang = clock() - zeit;
          maxzeit = maxzeitsec * CLOCKS_PER_SEC + anfang + zeit;
          bp(fp,nn,maxzeit,eta,alpha);
          zeit = clock() - anfang;
         }
         if(!strcmp(method,"kf") || !strcmp(method,"KF"))
         {
          dprintf0(fp,"Training with EKF Learning Alogirithm\n");
          anfang = clock() - zeit;
          maxzeit = maxzeitsec * CLOCKS_PER_SEC + anfang + zeit;
          bp_ekf(fp,nn,maxzeit);
          zeit = clock() - anfang;
         }

/* Display training time */

        zeitsec = zeit / CLOCKS_PER_SEC;
        if(nn->i_iter)
          dprintf3(fp,"Cpu-time in h.min.sec %2d.%02d.%02d\n",
```

```
                    (int)zeitsec / 3600, ( (int)zeitsec % 3600) / 60,
                            (int)zeitsec % 60)


/* Display the training error after last iteration */

        sys_err = 0.0;
        for(k=0; k< nn->npatterns_test; k++)
        {
          for(j=0; j< nn->layer_size[0]; j++)
                        nn->state[0][j] = nn->in_test[k][j];
          for(jout=0; jout< nn->layer_size[nn->nlayers]; jout++)
          {
                netz_out(nn->state,nn->weight,nn->layer_size,nn->nlayers);
                sys_err +=  (nn->target_test[k][jout]
                                - nn->state[nn->nlayers][jout]) *
                          (nn->target_test[k][jout]
                                - nn->state[nn->nlayers][jout]);
          }
        }
        if(nn->npatterns_test && nn->i_iter)
        {
          fehl = 2. * nn->norm_tar[0][1] *
                        sqrt(sys_err /
                          nn->npatterns_test /
                          nn->layer_size[nn->nlayers]);
          dprintf2(fp,
      "Rms-error after %3d iterations      %9.6f for test data\n",
                                nn->i_iter,fehl)
      dprintf1(fp,
        "Error according                  %9.3f for test data\n",
                                sys_err / nn->npatterns_test /
                                nn->layer_size[nn->nlayers] *
                                100 * (maxtest - mintest));
        }

        sys_err = 0.0;
        for(k=0; k< nn->npatterns; k++)
         {
          for(j=0; j< nn->layer_size[0]; j++)          /*Eingangsdaten*/
                        nn->state[0][j] = nn->in[k][j];
          for(jout = 0; jout< nn->layer_size[nn->nlayers]; jout++)
          {
                netz_out(nn->state,nn->weight,nn->layer_size,nn->nlayers);
                sys_err += (nn->target[k][jout]
                                - nn->state[nn->nlayers][jout]) *
                          (nn->target[k][jout]
                                - nn->state[nn->nlayers][jout]);
          }
         }
         if(nn->i_iter)
         {
          fehl = 2. * nn->norm_tar[0][1] *
                        sqrt(sys_err /
                          nn->npatterns /
                          nn->layer_size[nn->nlayers]);
          dprintf2(fp,
        "Rms-error after %3d iterations      %9.6f for training data\n",
                                nn->i_iter,fehl)
      dprintf1(fp,
        "Error according to PROBEN1 notation  %9.3f for training data\n",
                                sys_err / nn->npatterns /
                                nn->layer_size[nn->nlayers] *
                                100 * (maxtar - mintar));
        }

/* Ask for further computation */

        fclose(fp);
        fp = fopen("ekfnet.log","a");
```

```
            printf("How many iterations shall be executed until next prompt?\n");
            printf("(0 = end of training, storage of weights)              ");
            scanf1("%d",&nn->n_iter);
            if(!nn->n_iter) break;
            printf("Max. cpu-time in s                                    ");
            scanf1("%f",&maxzeitsec);
            printf("Training method (BP for backpropagation learning or EKF learning) ");
            scanf1("%s",method);

          } /*while(1)*/

/* Save new weight and covariance matrix */

          fp1 = fopen("ekfnet.wei","w");    /* Weight value */
          fprintf(fp1,"weights\n");
            for (j=1; j <= nn->nlayers; j++)   /* for all layers and weights */
              for (j1=0; j1 < nn->layer_size[j]; j1++)
              {
                    for (j2=0; j2 < nn->layer_size[j-1]; j2++)
                      fprintf(fp1,"%13.10f\n", nn->weight[j][j1][j2+1]);
                fprintf(fp1,"%13.10f\n", nn->weight[j][j1][0]);
                    }
          fprintf(fp1,"end\n");
          dprintf0(fp,"Weights stored in nnekf.wei\n")
          fclose(fp1);

          fp1 = fopen("nnekf.ekf","wb");
          fprintf(fp1,"ekf\n");
          fwrite(nn->p,sizeof(float),nn->nweights*(nn->nweights+1)/2,fp1);
          fprintf(fp1,"\nend\n");
          dprintf0(fp,"Covariance matrix stored in nnekf.ekf\n")
          fclose(fp1);

}
```

/* File name: BPEKF.C */

```
/*
 Routine for extended Kalman filter(EKF) learning algorithm
 This routine is called by NNEKF.c

 Yoon-Seok Hong
 Institute of Technology & Engineering
 Massey University
 New Zealand
 E-Mail: Y.S.Hong@massey.ac.nz
*/


# include <stdio.h>
# include <stdlib.h>
# include <string.h>
# include <time.h>
# include <math.h>
# include "comnn.h"

# define dprintf0(x,y)          { fprintf(x,y);     printf(y);     }
# define dprintf1(x,y,a)        { fprintf(x,y,a);   printf(y,a);   }
# define dprintf2(x,y,a,b)      { fprintf(x,y,a,b); printf(y,a,b); }
# define dprintf3(x,y,a,b,c)    { fprintf(x,y,a,b,c); printf(y,a,b,c); }
# define pii(i)     (i) * ( 3 + i ) / 2
# define ACCURACY               .000001

void bp_ekf(FILE *fp, COMNN *nn, clock_t maxzeit)
{

    extern void netz_out(float**,float***,int*,int);
        extern void netz_back(float**,float***,float***,int*,int,int,int);
        extern int ekf(float*,float*,float,int,float*,float,float*);
```

```
        float res,diff,sig,q,test_err,sys_err,***jacobi,*weight_old,*pb;
        float sigmin,psum,jacobimax;
        time_t datum;
        int kit,i,i1,ii,j,j1,j2,jout,nsqrt;

        int status;

/********* Initialising ************************************/


        jacobi = (float***) malloc((nn->nlayers+1)*sizeof(float**));
        if(!jacobi) dprintf0(fp,"Error in malloc for 'jacobi'\007\n")

        jacobi[0] = (float**) malloc(nn->nstates*sizeof(float*));
        if(!jacobi[0]) dprintf0(fp,"Error in malloc for 'jacobi[0]'\007\n")
        jacobi[1] = jacobi[0];
        for(i=2; i<=nn->nlayers; i++)
          jacobi[i] = jacobi[i-1] + nn->layer_size[i-1];

        jacobi[0][0] = (float*) malloc(nn->nweights*sizeof(float));
        if(!jacobi[0][0])
                        dprintf0(fp,"Error in malloc for 'jacobi[0][0]'\007\n")

        for(i=2; i<=nn->nlayers; i++) jacobi[i][0] = jacobi[i-1][0]
            + nn->layer_size[i-1] * (nn->layer_size[i-2] + 1);
        for(i=1; i<=nn->nlayers; i++)
          for(j=1; j<nn->layer_size[i]; j++)
            jacobi[i][j] = jacobi[i][j-1] + nn->layer_size[i-1] + 1;

        pb = (float*) malloc(nn->nweights*sizeof(float));
        if(!pb) dprintf0(fp,"Errorr in malloc for 'pb'\007\n")
        weight_old = (float*) malloc(nn->nweights*sizeof(float));
        if(!weight_old) dprintf0(fp,"Error in malloc for 'weight_old'\007\n")


/* Training of neural network */
/*****************************************************************************/

        nsqrt = sqrt((double)nn->npatterns);
        q = .01;

        jacobimax = 0.;
        for(i=0; i<nn->npatterns; i++)
        {
          for(j=0; j<nn->layer_size[0]; j++)
                        nn->state[0][j] = nn->in[i][j];
          netz_out(nn->state,nn->weight,nn->layer_size,nn->nlayers);

          for(jout=0; jout<nn->layer_size[nn->nlayers]; jout++)
          {
                netz_back(nn->state,nn->weight,jacobi,
                        nn->layer_size,nn->nlayers,nn->nweights,jout);

                for(j=1; j<nn->nlayers; j++)
                  for(j1=0; j1<nn->layer_size[j]; j1++)
                    for(j2=0; j2<nn->layer_size[j-1]; j2++)
                      if(jacobi[j][j1][j2] * jacobi[j][j1][j2] > jacobimax)
                                jacobimax = jacobi[j][j1][j2] * jacobi[j][j1][j2];
          }
        }


        for(kit=0; kit<nn->n_iter; kit++)                       /*Iteration*/
        {

            sys_err = 0.0;
            for(i=0; i<nn->npatterns; i++)                      /*Training data*/
            {
                for(j=0; j<nn->layer_size[0]; j++)
                                nn->state[0][j] = nn->in[i][j];
                netz_out(nn->state,nn->weight,nn->layer_size,nn->nlayers);
```

```
        for(jout=0; jout< nn->layer_size[nn->nlayers]; jout++)
          sys_err += (nn->state[nn->nlayers][jout]
                                                  - nn->target[i][jout]) *
                     (nn->state[nn->nlayers][jout] - nn->target[i][jout]);
}
dprintf2(fp,"%4d  %12.9f",kit,sqrt((double)sys_err
                     / nn->layer_size[nn->nlayers] / nn->npatterns)
                                                  * 2. * nn->norm_tar[0][1]);

if(nn->npatterns_test)
{
        test_err = 0.0;
        for(i=0; i< nn->npatterns_test; i++)            /*Testing data*/
        {
         for(j=0; j< nn->layer_size[0]; j++)
                                nn->state[0][j] = nn->in_test[i][j];
         netz_out(nn->state,nn->weight,nn->layer_size,nn->nlayers);
         for(jout=0; jout< nn->layer_size[nn->nlayers]; jout++)
          test_err += (nn->state[nn->nlayers][jout]
                                                  - nn->target_test[i][jout]) *
          (nn->state[nn->nlayers][jout] - nn->target_test[i][jout]);
        }
        dprintf1(fp,"   %12.9f\n",sqrt((double)test_err
                     / nn->layer_size[nn->nlayers] / nn->npatterns_test)
                                                  * 2. * nn->norm_tar[0][1]);
}
else
        dprintf0(fp,"\n")


if(clock() > maxzeit) return;

sig = sys_err / nn->npatterns / nn->layer_size[nn->nlayers] /10.;

psum = 0.;
for(i=0; i< nn->nweights; i++) psum += sqrt((double)nn->p[pii(i)]);
sigmin = ACCURACY * jacobimax * psum * psum;
if(sig < sigmin)      sig = sigmin;

memcpy(weight_old,&nn->weight[1][0][0],nn->nweights*4);

ii = 0;
i1 = 0;

/*Training*/

for(i=0; i< nn->npatterns; i++)
          for(j=0; j< nn->layer_size[0]; j++)
                    nn->state[0][j] = nn->in[i1++];
          netz_out(nn->state,nn->weight,nn->layer_size,nn->nlayers);

          for(jout=0; jout< nn->layer_size[nn->nlayers]; jout++)
          {
              res = nn->target[i1][jout] -
                                                  nn->state[nn->nlayers][jout];
                    netz_back(nn->state,nn->weight,jacobi,
                              nn->layer_size,nn->nlayers,nn->nweights,jout);

                    status = ekf(&nn->weight[1][0][0],&jacobi[1][0][0],res,
                                        nn->nweights,nn->p,sig,pb);
                      /*ekf korrigiert weight aufgrund von jacobi und res*/
                    if(status)
                    {
                              fprintf(fp,"ekf singular  ");
                              dprintf3(fp,"sigma= %g, Psum= %g, Q= %g\n",
                              sig,psum,ACCURACY + q * diff / nn->nweights)
                    }

          }

          i1 += nsqrt;
```

```
                if(il   '' nn- npatterns)
                {
                        ii++;
                        i l = ii;
                }
            }

            diff = 0.;
            for(i=0; i nn- nweights; i++)
                        diff += (nn- weight[l][0][i] - weight_old[i]) *
                                (nn- weight[l][0][i] - weight_old[i]);
            for(j=0; j nn- nweights; j++)
                        nn- p[pii(j)] += ACCURACY + q * diff / nn- nweights;

            nn- i iter ! ;
        }

    free(jacobi[0][0]);
        free(jacobi[0]);
        free(jacobi);
        free(weight_old);
        free(pb);

}
```

/* File name: EKF.C */

```
/*
 Routine for extended Kalman filter
 This routine is called by NNEKF.C
*/

# include <stdio.h>          /*printf*/
int ekf(float *a, float *b, float res, int n, float *p, float s, float *pb)
{

/* Extended Kalman Filter (EKF)
/*
/*          a= estimation vector
/*          b= Gradient vector
/*          res= error of output
/*          n= length of vector
/*          p= P-Matrix
/*          s=Sigma**2
/*          pb =help vector of length n
/*
/*          In this program, the following declaration are required:
/*          float a[n],b[n],p[n*(n+1)/2],pb[n]
/*
/*    Examples:
/*
/*          n=100       !(declanation could be bigger)
/*          s=.01       !(not 0)
/*          a0=1.
/*          p0=100.

/*          for(i=0; i<(n+1)*n/2; i++)p[i] = 0.;
/*
/*          for(i=0; i<n; i++)
/*          {   a[i]=a0;
/*              p[(i+3)*i/2]=p0;
/*          }
/*
/*

    float nenner,pbk,pbn,bi,rn,*pp,*ppp;
    int i,j,k;
```

```
/*                                          pb = p * b
*/
        pp = p;
        for(k=0; k<n; k++)
        {
                pp += k;
                pbk = pp[0] * b[0];
                for(i=1; i<=k; i++)
                        pbk += pp[i] * b[i];
                ppp = &pp[k];
                for(i=k+1; i<n; i++)
                {
                        ppp = &ppp[i];
                        pbk += *ppp * b[i];
                }
                pb[k] = pbk;
        }
/*
        pp = p;
        for(i=1; i<n; i++)
        {
                bi = b[i];
                pp += i;
                for(k=0; k<i; k++)
                        pb[k] += pp[k] * bi;
        }
*//*                                                  nenner = s + bT * p * b
*/

        nenner = s;
        for(k=0; k<n; k++) nenner += b[k] * pb[k];

*/
*/
        if(nenner < .1 * s)
        {
                printf("EKF Learning singulaer: nenner= %10.3e %c\n",nenner,7);
                return(1);
        }


/* Correct the estimamted value and save b
/*                      a = a + p * b / nenner * res
*/
        rn = res / nenner;
        for(k=0; k<n; k++)
                a[k] += pb[k] * rn;


/*  Correction of covariance matrix
/*   p = p - p * b * bT * p / nenner
*/
        for(k=0; k<n; k++)
        {
                pbn = pb[k] / nenner;
                for(i=0; i<=k; i++)
                        p[i] -= pbn * pb[i];
                p += k+1;
        }

        return(0);
}
```