Cape Peninsula
University of Technology

INVESTIGATION AND DEVELOPMENT OF METHODS FOR OPTIMAL CONTROL
OF THE ACTIVATED SLUDGE PROCESS

by

KOKETSO PORTIA KUJANE

Thesis submitted in fulfilment of the requirements for the degree

Master of Technology: Electrical Engineering

in the Faculty of Engineering

at the Cape Peninsula University of Technology

Supervisor: Professor Raynitchka Tzoneva

Bellville
December 2009

DECLARATION

I, Koketso Portia Kujane, declare that the contents of this thesis represent my own unaided work, and that the thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

_____   _____
Signed   DEC 2009
                                          Date

ABSTRACT

This project was started as a result of strict environmental and health regulations together with a demand for cost effective operation of wastewater treatment plants (WWTPs). The main aim of this project is how to keep effluent concentration below a prescribed limit at the lowest possible cost. Due to large fluctuations in the quality and quantity of the influent concentrations, traditional control methods are not adequate to achieve this aim. The major drawback with these methods is that the disturbances affect the process before the controller has time to correct the error (Olsson and Newell, 1999: 454). This problem is addressed through the use of modern control systems.

Modern control systems are model based predictive algorithms arranged as feed-forward controllers (Olsson and Newell, 1999: 454). Normally a controller is equipped with a constant set point; the goal in this project is to calculate an optimal DO trajectory that may be sampled to provide a varying optimal set-point for the Activated Sludge Process. In this project an optimal control problem is formulated using DO concentration as a control variable. This requires a model of the process to be controlled, a mathematical expressions of the limitations on the process input and output variables and finally the objective functional which consists of the objectives of the control.

The structures of the Benchmark plant (developed within the COST 682 working group) and the Athlone WWTPs are used to implement this optimal control strategy in MATLAB. The plant's full models are developed based on the mass balance principle incorporating the activated sludge biological models; ASM1, ASM2, ASM2d and ASM3 (developed by the IWA working groups). To be able to develop a method that may later on be used for online control, the full models are reduced based on the technique in Lukasse (1996). To ensure that the reduced models keep the same prediction capabilities as the full models, parameters of the reduced models are calculated based on the Least Squares principle. The formulated optimal control problem is solved based on the decomposition-coordination method that involves time decomposition in a two layer structure.

MATLAB software is developed to solve the problems for parameter estimation, full and reduced model simulation, and optimal control calculation for the considered different cases of plant structures and biological models. The obtained optimal DO trajectories produced the effluent state trajectories within prescribed requirements. These DO trajectories may be implemented in different SCADA systems to be tracked as set points or desired trajectories by different types of controllers.

ACKNOWLEDGEMENTS

I wish to thank:

- The Lord God almighty, who gave me the internal strength through out the research work
- My parents Agisanyang Lawrence and Dorah Kujane, sisters Bonolo Ramaisa, Tshwanelo Kujane, brother Khumo Kujane and nephew Warona Kujane for their patience and understanding during the study.
- Professor Raynitchka Tzoneva for her technical and moral support through all the years of study and during the research.
- The Electrical Engineering Controls Group at the Cape Peninsula University of Technology

DEDICATION

For my brother Khumo Kujane

TABLE OF CONTENTS

CHAPTER FOUR: DERIVATIONS OF THE MODELS OF THE WASTEWATER TREATMENT PLANTS

CHAPTER FIVE: PARAMETER ESTIMATION AND SIMULATIONS OF THE MODELS

CHAPTER SIX: DEVELOPMENT OF METHOD FOR OPTIMAL CONTROL CALCULATION FOR THE ACTIVATED SLUDGE PROCESSES

CHAPTER SEVEN: CONCLUSION AND FUTURE RECOMMENDATIONS

LIST OF TABLES

GLOSSARY

Terms/Acronyms/Abbreviations	Definition/Explanation
ASM	Activated sludge model
ASM1	Activated sludge model No. 1
ASM2	Activated sludge model No. 2
ASM2d	Activated sludge model No. 2d
ASM3	Activated sludge model No. 3
ASP	Activated sludge process
BOD	Biolochemical oxygen demands
C	Carbon
CO2	Carbon dioxide
COD	Chemical oxygen demand
COST	European Cooporation in the field of Scientific and Technical Research
Denitrification	The process of converting nitrate-nitrogen to gaseous nitrogen through anoxic heterotrophic cell growth
DO	Dissolved oxygen
Dynamic model	Set of differential equations which predict time varying performance of a process
Dynamics	Transient or unstable conditions that changes with time
IAWQ	International Association on Water Quality
Input/output model	Model that describes system behaviour as being a function of only the present and past inputs / outputs
IWA	International Water Association
Least squares method	A method of fitting a curve to given points such that the sum of the squares of the deviations between the curve and the points is minimal
LGQ	Linear Quadratic Gaussian
MATLAB	Matrix Laboratory
Model	Set of equations used to describe a process
MPC	Model predictive control
N	Nitrogen
Nitrification	Conversion of ammonia-nitrogen to nitrite and nitrate-nitrogen through autotrophic cell growth
Online estimation	Estimation of model parameters during the operation of the system
P	Phosphorus
PAO	Phosphorus accumulating organisms
Parameter	A variable whose different values result in different values of the observed outputs
PID	Proportional Integral Derivative
PLC	Programmable logic controller
S	Substrate
Set point	The desired value for a control system at which a process should operate
Simulation	Using computer to predict performance of a process under different conditions
TKN	Total Kjeldahl Nitrogen; sum of ammonia and organic nitrogen
Trajectory	A sequence of values collected at equal time intervals
UCT	University of Cape Town
WWTP	Wastewater Treatment Plant

NOMENCLATURE

Symbols / letters	Definition/Explanation
A	Matrix with coefficients of the wastewater components concentrations in the system
B	Matrix with coefficients of the input wastewater concentrations
b_A	Correction factor for anoxic growth of heterotrophs
$b_{A,NOX}$	Anoxic endogenous respiration rate of XA
$b_{A,O2}$	Aerobic endogenous respiration rate of XA
b_H	Heterotrophic decay rate
$b_{H,NOX}$	Anoxic endogenous respiration rate of XH
$b_{H,O2}$	Aerobic endogenous respiration rate of XH
$b_{STO,NOX}$	Anoxic respiration rate of XSTO
$b_{STO,O2}$	Aerobic respiration rate of XSTO
b_{PAO}	Rate constant for lysis of XPAO
b_{PHA}	Rate constant for lysis of XPHA
b_{PP}	Rate constant for lysis of XPP
C	Matrix of the yield coefficients
e	Error between the measured and true values of variables
e_x, e_u, e_u	Gradients
$f = f_1, f_2, ..., f_l$	I-Dimensional vector of functions
f_P	Fraction of biomass yielding particulate products
f_{SI}	Production of SI in hydrolysis
f_{XI}	*Production of XI in endogenous respiration*
i_{XB}	COD mass in biomass
i_{XP}	COD mass in products from biomass
J	Cost function
k	Time instant (discrete)
K	Final time of the prediction horizon
k_a	Ammonification rate
$K_{A,ALK}$	Bicarbonate saturation for nitrifiers
K_{ALK}	Saturation coefficient for alkalinity
$K_{A,NH4}$	Ammonium substrate saturation for XA
$K_{A,O2}$	Oxygen saturation for nitrifiers
K_F	Saturation coefficient for growth on SF
K_{fe}	Saturation coefficient for fermentation of SF
K_h	Hydrolysis rate constant
K_{IPP}	Inhibition coefficient for PP storage
K_{La}	Oxygen mass transfer rate coefficient
K_{MAX}	Maximum ration or XPP/XPAO
K_{NH}	Ammonia hsc for autotrophs

$S_{O,sat}$	Dissolved oxygen saturation
S_{PO4}	Phosphate
S_s	Readily biodegradable substrate
u	Vector of the control variables
$u = u_1, u_2, ..., u_m$	Independent variables
u_{min}	Constraints on the control variables
$u^\cdot$	Vector of the optimal control variables
V	Volume
V_1	Tank 1 volume
V_2	Tank 2 volume
V_3	Tank 3 volume
V_4	Tank 4 volume
V_5	Tank 5 volume
x	Vector of the process variables
X_e	Components concentrations in the effluent wastewater
X_{in}	Wastewater components concentration in the input
X_{min}	Constraints on the state variables
X_{out}	Wastewater components concentration in the output
X_r	Recycle concentrations of the components of the wastewater sludge
X_0	Components concentrations in the influent wastewater
X_1	Tank 1 wastewater components concentration
X_2	Tank 2 wastewater components concentration
X_3	Tank 3 wastewater components concentration
X_4	Tank 4 wastewater components concentration
X_5	Tank 5 wastewater components concentration
$x^\cdot$	Vector of the optimal state variables
x^{SP}	Wastewater effluent set points
$X_{B,A}$	Autotrophic biomass
$X_{B,H}$	Active heterotrophic biomass
X_I	Particulate non-biodegradable inert material
X_{MeOH}	Metal-hydroxides
X_{MeP}	Metal phosphates
X_{ND}	Particulate biodegradable organic nitrogen
X_P	Inert particulate products arising from biomass decay
X_{PAO}	Phosphate-accumulating organisms
X_{PHA}	A cell internal storage product of phosphorus-accumulating organisms
X_{PP}	poly phosphate
X_s	Slowly biodegradable substrate
X_{SS}	Suspended solids
X_{STO}	A cell internal storage product heterotrophic organisms
X_{TSS}	Total suspended solids

$Y_{H,NOX}$	Anoxic yield of heterotrophic biomass
$Y_{H,O2}$	Aerobic yield of heterotrophic biomass
Y_{PAO}	Yield coefficient for biomass
Y_{PHA}	Biomass requirement for PP storage
Y_{PO4}	PP requirement for biomass storage
$Y_{STO,NOX}$	Anoxic yield of stored product per SS
$Y_{STO,O2}$	Aerobic yield of stored product per SS
α_x	Steps of the gradient procedures
ε_0	Constraint on the norm of the gradient
φ	Termed regressor
λ	Vector of the Lagrange multipliers
θ	Parameter vector
$\hat{\theta}$	Estimate of Θ
Δt	Sampling time
μ_A	Autotrophic maximum growth rate
μ_H	Heterotrophic maximum growth rate
μ_{PAO}	Maximum growth rate for XPAO
ηfe	Anaerobic hydrolysis reduction factor
η_g	Correction factor for anoxic growth of heterotrophs
η_h	Correction factor for anoxic hydrolysis
$\eta_{NOX}\ \eta_{NOX}$	Reduction factor for denitrification

CHAPTER ONE
INTRODUCTION

1.1   Introduction

Clean water is a valuable resource lacking in many parts of the world especially in Africa where people have died from diseases related to water contamination (e.g. cholera). Every living thing on earth (microorganisms, plants, animals, humans and even our brain) consists mostly of water. More than 70% of the earth's surface is covered by water, only a small part of which is suitable for either human consumption or agricultural use (approximately 0.5% of all water in the world). Due to pollution from agriculture, households and industry reaching rivers, lakes and seas it is important for wastewater (sewage) to be properly treated in order to remove harmful substances before it reaches the environment.

Strict environmental and health regulations together with a demand for cost effective ways of wastewater treatment have made control technology in Wastewater Treatment Plants (WWTPs) an important priority. In WWTPs, the control system is used to supervise the inherently dynamic wastewater treatment processes from the influent wastewater through biological processes up until effluent dispersal. The introduction of a control system in these plants is expected to reduce the need for larger volumes, improve the effluent water quality, decrease the use of chemicals, and save energy and operational costs. Sensors for dissolved oxygen and nutrients such as ammonium, nitrate and phosphorus have been developed and improved in recent years, thus increasing the potential for effective monitoring, automation and control of wastewater treatment plants. This assists in reaching the objective of the wastewater treatment process operation, which is to provide reliable and stable process control with highly efficient results (low concentration of substrates in the effluent wastewater) at minimum cost.

In this thesis a method for solution of the problem for optimal control of the activated sludge process is developed. Generally the objective of optimal control is to make the output signals of the system behave in a desired way by manipulating certain input signals that affect the behaviour of the system. The control strategy is firstly to estimate the disturbance in the system by making use of the process model to predict the behaviour of the influent disturbances.

Secondly to use optimal control theory (process model, process constraints and the objective functional) to search for optimal control signal to be used as a reference trajectory for the controller for real time process control. In this thesis, optimal control problem is solved based on the reduced process models for the purpose of future control application in real time.

The focus of the project is to control dissolved oxygen needed by microorganism in the ASP to grow. The aim of the suggested method is to efficiently control and reduce unnecessary dissolved oxygen concentration, which in turn reduces the amount of energy, required for high airflow rate and thus minimises cost.

In this chapter, the biological process of a wastewater treatment plant is briefly described. The common strategy that is used when controlling this process is also described. The optimal control strategy that is used in this work is briefly introduced giving the reader a short explanation of the strategy. The strategy is fully described in Chapter three.

1.2 Introduction to wastewater treatment

Wastewater treatment plants comprise various processes used for the reduction of contaminants in wastewater. A typical wastewater treatment plant includes a mechanical system and a biological process. The mechanical system consists of grids and sand filters for removing large objects as well as primary sedimentation for removing suspended solids. If not removed these objects may cause system blocks and damage equipment. A biological system is used for the reduction of organic matter and harmful nutrients such as nitrogen and phosphorus. The mechanical and biological systems of a wastewater treatment plant are shown in Figure 1.1. (Samuelsson, 2001).
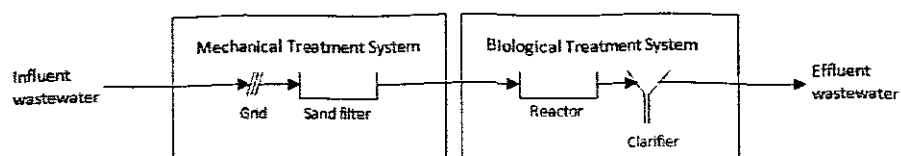
Figure 1.1: The mechanical and biological wastewater treatment systems

(Adapted from Samuelsson, 2001)

2

1.2.1 Activated sludge process

The activated sludge process (ASP) is one of the biological processes used at most wastewater treatment plants. This process is used to biologically reduce contents of organic matter, nitrogen and phosphorus in wastewater. If untreated the organic matter may cause oxygen depletion when discharged to the recipient water.

Nitrogen and phosphorus may stimulate undesired plant growth in the recipient water and thereby affect the ecosystem negatively. A typical layout of the ASP is shown in Figure 1.2 (Samuelsson, 2001).
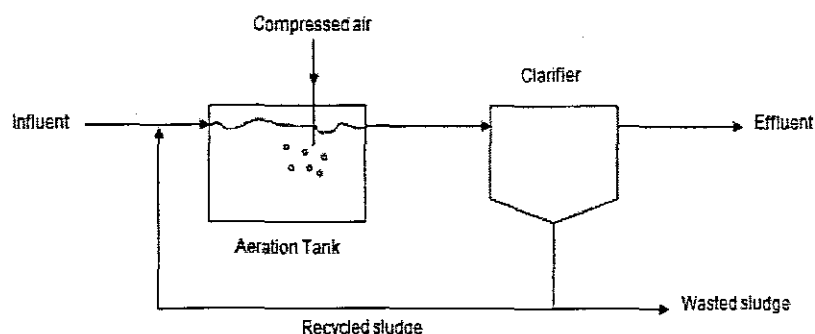
Figure 1.2: A typical layout of the activated sludge process

(Adapted from Samuelsson, 2001)

A typical ASP consists of an aeration tank and a clarifier as shown above. In the aeration tank, microorganisms utilize oxidized organic materials for their growth and energy. Microorganisms then form a biological floc which gets suspended due to aeration. After for 4 to 8 hours the floc is allowed to overflow into the clarifier. After being in the clarifier for 1 to 2 hours, gravity forces the floc to settle in the bottom.

The settled material is then drained from the bottom while the treated wastewater is allowed to flow from the top of the clarifier. Organic matter leaves the system in three ways; as part of the settled material, as CO_2 or with the effluent (Matas, 2003). The quality of the effluent depends on biomass growth in the aeration tank therefore a part of the sludge (settled biomass) is recycled back to the aerated tank to keep a balance between biomass concentration and waste material in the tank.

A basic ASP may be extended to include biological removal of nitrogen. Nitrogen removal requires a two step configuration with aerobic and anoxic tanks. A basic ASP

3

designed to include the biological removal of nitrogen is shown in Figure 1.3. In the aerated tank, ammonia nitrogen is converted into nitrate through nitrification process while in the anoxic tank nitrate is converted into nitrogenous gas through denitrification process. This is how nitrogen leaves the system: it is converted into a gas that disperses into the atmosphere.
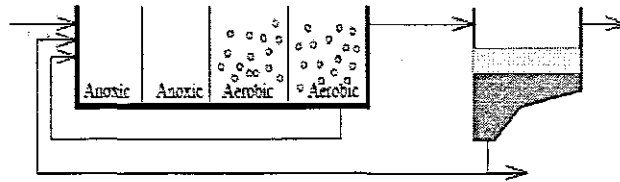
Figure 1.3: Typical layout of the ASP incorporating biological nitrogen removal

(Adapted from Lindberg, 1997)

A basic ASP may be further extended to include biological removal of phosphorus. Phosphorus in wastewater is in the form of polyphosphate granules. These polyphosphate granules are used as energy source by the polyphosphate-accumulating organisms (PAO) under anaerobic conditions. The University of Cape Town plant layout shown in Chapter four on Figure 4.5 is a typical example of an ASP configured to incorporate biological phosphorus removal.

1.2.2 Activated sludge process optimal control

Traditionally there has been very little emphasis on wastewater control systems; as a result wastewater treatment plants are only equipped with simple control systems. These systems are mainly used for *on and off* control and for monitoring purpose. Dissolved oxygen (DO) is the most commonly monitored and controlled variable of the activated sludge process (ASP). DO sensors are used to monitor and relay measurement of DO to a computer. Simultaneously, a programmable logic controller (PLC) controls the amount of DO supplied to the aeration tanks. Typically, a PLC is set to regulate DO value to 2 mg/l.

Cost minimization is an important factor in the operation of the wastewater treatment plant. In order to run a plant economically, operational costs such as pumping energy and aeration energy should be minimized. At the same time discharges to the water recipients should be kept at a low level. Minimizing the operational costs and at the same

4

time treat wastewater properly lead to conflict of interest that must be solved. This conflict can be seen as a multi-objective optimization problem where effluent discharges must be balanced against operational costs. The main question is how to keep the effluent discharges below a certain pre-scribed limit to the lowest possible cost.

The main purpose of this project is to solve this optimization problem. This is solved through automatic control of the process. The objectives of automatic control are making the output signals of the system behave in a desired way by manipulating certain input signals that affect the behaviour. The first step in the design of automatic control strategy is to determine which signals to control, which to measure or estimate and which control signal to use. The next step is to recognize the disturbances in the system. The last step is to derive the control signal to the system. Finally, the control strategy for the process may be developed.

1.2.2.1 Controllable parameters of the activated sludge process

The ASP can be considered as represented by chemical and biological variables as shown Figure 1.4. The chemical group consists of variables such as temperature, DO, dissolved carbon dioxide, pH and flow rate. The biological group consists of concentrations of substrate and biomass. As opposed to the chemical variables, the biological variables are not measurable online. The chemical group controls enzyme activity of the microorganisms and in this way controls the biological variables. This means that if the optimal conditions for the chemical variables are reached according to some criteria, then it will also be reached for the biological variables.
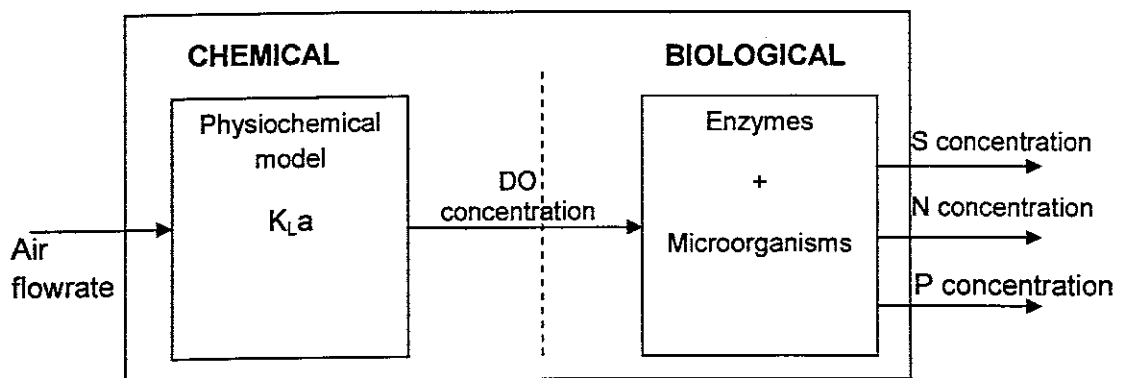
Figure 1.4: Dependence between chemical and biological variables

K_La is the oxygen transfer coefficient and it describes the rate at which oxygen is transferred into the system. DO is the Dissolved Oxygen concentration, S is Substrate, N is Nitrogen and P is Phosphorus. The figure shows how air an input to the chemical variables is controlled to give the DO output that is necessary to the biological variables. Figure 1.5 an example of a closed loop system control of DO concentration in the aerobic tank of the activated sludge process.
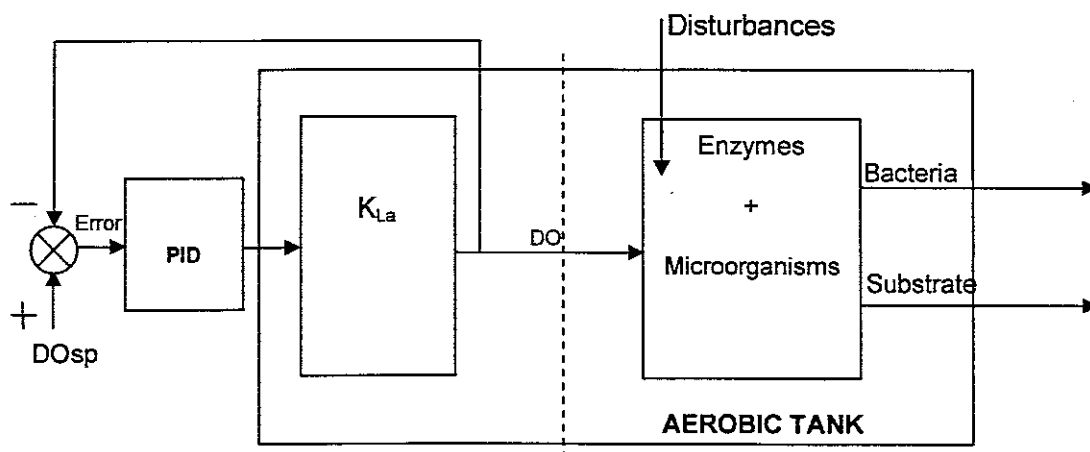
Figure 1.5: DO control of the Activated Sludge Control

1.2.2.2 Disturbance factors in the activated sludge process

The main disturbance of the ASP is the inflow load, which is characterized with different values of the flow and waste component concentrations. The inflow characteristics change with the change of temperature, wind and time. The load change leads to changes of the environment conditions for the microorganisms, which requires different values of the DO. The required DO values are not always clear, it has to be estimated or calculated in some way. The solution of this problem is to determine the required DO solving optimal control problem for the process. This problem is formulated and solved in the project.

1.3 Introduction to optimal control

Optimal control theory is based on the calculus of variations method; it is aimed at making functionals reach minimum or maximum values. Optimal control is a mathematical optimization method for deriving optimal control trajectories.

6

The objective of optimal control is to predict best values of the control variable that will make the process behave in an optimal way. This means that a system must achieve effluent requirements while satisfying physical constraints of the plant. This is achieved through minimization of a pre-specified optimality criterion (Katebi, 1998).

1.4   Problem identification

The South African constitution states that the provincial governments have to ensure effective performance by their municipalities in terms of wastewater and sewage disposal (South Africa. National Government, 1996). This includes adherence to local and international standards governing effluent wastewater disposal. In recent years, these standards have been made more severe in order to improve the state of the environment as well as cater for disposal problems due to growing urbanisation and industrialization. As a result of these developments, there has been a lot of support from the Water Research Commission towards South African research in wastewater treatment, with the main focus being on the designs of the wastewater treatment plants (WRC, 2002). Not much of the researches are directed towards wastewater treatment plants control systems.

Most wastewater treatment plants around the world and in South Africa use simple feedback control systems with *on and off* capabilities. This is despite the fact that wastewater treatment processes are known to be highly complex and nonlinear in their nature. The other issue surrounding wastewater treatment plants is the high operational cost incurred.

Much of the operational cost is related to the electrical energy utilized during aeration. It is believed that electricity consumption makes up to more than 40 percent of the total operational costs (Motor Management Success, 2001). It is also believed that the energy saving potential for wastewater treatment plants is very high. This potential can be reached through the use of advanced control and monitoring systems.

Without advanced control and monitoring systems, wastewater treatment plants cannot run efficiently and effectively. Figure 1.6 shows a decrease in concentrations of phosphorus in some of the rivers of the European Union after building and exploration of modern control systems in many European countries. Advanced control systems give more possibilities to achieve these trends at a lower energy costs.

7
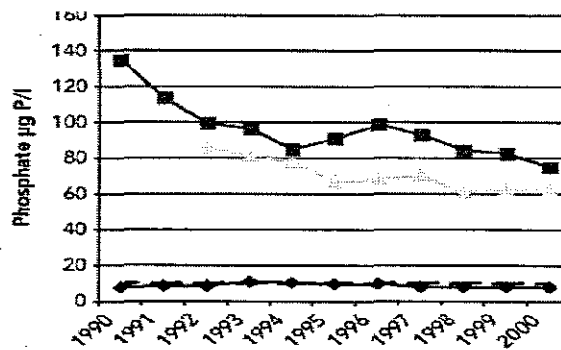
Figure 1.6: Phosphate concentrations in some of the rivers in Europe

(Adapted from EEA, nd)

The dynamic complexities of wastewater treatment processes, the related energy cost and the strict effluent requirements cannot be dealt with by means of simple feedback systems. They require the use of advanced state of the art control systems, such as optimal control. The availability of on-line sensors and mathematical models makes it possible to design advanced control systems for the activated sludge process. Several advanced control strategies have previously been suggested by Bastin and Dochain (1990). Application of modern control theory in combination with new on-line sensors and the appropriate parts of advanced models has great potential to improve the effluent water quality, decrease the use of chemicals and save energy and money.

1.5 Motivation

Control systems in wastewater have been recognized for almost 30 years but it has never been a major part of the wastewater treatment plant (WWTP) design. Many WWTPs have been designed and constructed without any attention given to the control systems.

According to Henze et al (2000) the application of control systems increases the capacity of the biological nutrient removal by 30%. In other cases, the application of control systems has significantly decreased operating costs.

Wastewater treatment plants are susceptible to a lot of disturbances; these disturbances are a reason for the plants control. The impact of the disturbances has to be compensated or eliminated before influencing the behaviour of the plant.

8

The wastewater disturbances are related to varying influent flow concentration and composition and varying influent flow rates with time scale ranging between hours to months. Storms and peak loads also make up the disturbances; they prohibit steady state behaviour of the plant. Traditionally, wastewater disturbances have been dampened by construction of bigger plants to attenuate large loads (Henze et al, 2000).

Wastewater systems are dynamic systems and any correction takes time before it can be noticed in the system. A wastewater treatment plant is always in transient state because of the varying influent flow rates, concentration and compositions. The control system has to recognize this and by online measurements and corrective actions bring the various process variables to their desired values.

Dissolved oxygen (DO) concentration is a key variable in the operation of the activated sludge process. From a biological point of view, a choice of a proper DO set point is crucial. Insufficient air supply will influence the growth of the organisms, the floc formation and the settling properties of the sludge. The controller decision is based on the current state of the process obtained by a sensor and feedback.

Simple feedback control system requires information on the current state of the process. Optimal control systems have capabilities to use predicted disturbances to predict plant behaviour and then calculate a proper DO control set point to attenuate the disturbance before it even reaches the plant. An example is the possibility for aeration to be increased before a load change hits the plant.

1.6   Statement of the problem

Main problems encountered during operation of an ASP are the stringent requirements on effluent quality and the high cost incurred. With the main contributor to the high operational cost being the aeration related energy cost.

The ASP requires oxygen for the minimization of carbon and nutrients in wastewater. Too much oxygen may decrease process efficiency and increase energy consumption. With too little oxygen, microorganisms may not perform well enough to remove harmful wastewater components. Finding adequate oxygen concentration in the aeration tank is the problem considered and solved in the thesis.

9

The aim is to determine the amount of DO concentration required for the process to produce the pre-specified effluent trajectories at a lower cost.

1.7 Research aim and objectives

The overall aim of the research project is to investigate and develop methods for calculation of the dissolved oxygen optimal trajectory that may later be used for real time optimal control of activated sludge processes. The idea is to use reduced ASP models to observe the behaviour of the ASP based on values of the predicted influent concentrations. Then to pre-calculate values of the dissolved oxygen (DO) concentration (in the aeration tanks) that forces the process to behave in a pre-specified manner. The calculated DO values are then treated as set points for the controller governing the rate of air supply to the real plant. By so doing, it is certain that the controller considers the effect of the influent disturbances of the ASP. The aim is achieved through the following objectives:

1. Literature review and analysis
   o Analysis of the need for optimal DO control
   o Literature review of the existing methods for optimal DO control
2. Design of models and methods for optimal control calculation
   o Analysis of the full models to investigate their possibilities to be reduced in size and complexity
   o Development of reduced models based on the following biological models: Activated Sludge Model No. 1 (ASM1), Activated Sludge Model No. 2 (ASM2), Activated Sludge Model No. 2d (ASM2d) and Activated Sludge Model No. 3 (ASM3)
   o Development of the parameter estimation procedure
   o Formulation of the problem for optimal control
   o Development of a decomposition coordination method for optimal control calculation
3. Development of software
   o Development of MATLAB software for simulation of the full models
   o Development of MATLAB software for parameter estimation of the reduced models
   o Development of MATLAB software for simulation of the reduced models
   o Development of MATLAB software for optimal control calculation
   o All program for simulation

1.8 Hypothesis

Disturbances to the ASP as well as the ASP nonlinear and complex dynamics make it difficult to use classical control systems for ASP control. These control systems require a priori knowledge of the process, which is very limited due to the lack of sensors for ASP components. The optimal control strategy is one of the best strategies to use because it utilizes the process model to predict the behaviour of the process and calculate optimal controls to change the behaviour of the process.

1.9 Delimitation

The research is based on the development of methods for solution of the problem for optimal control of the activated sludge process. For the control system dissolved oxygen is the chosen control variable.

The following reduced activated Sludge models are used in the formulation of the optimal control problem:

- Activated Sludge Model No. 1 (ASM1)
- Activated Sludge Model No. 2 (ASM2)
- Activated Sludge Model No. 2d (ASM2d)
- Activated Sludge Model No. 3 (ASM3)

There are a lot of other biological models in the literature; however these models are not considered here.

The algorithm for calculation of the optimal control problem is implemented on MATLAB. To solve the formulated optimal control problem, the Lagrangian and the decomposition methods are used.

The research is based on the Benchmark and the Athlone treatment plant structures. The research project does not include controller design.

1.10 Assumptions

The research project is based on the following assumptions:

- The mass balance equations accurately represent the actual processes.

11

- For optimal control problem solution, the predicted influent concentration characterised by the weather files is assumed to be a close approximation of the real influent concentration during the respective weather.
- The settler is assumed to be without biological reaction therefore only steady state mass balance equations are used to model activities in the settler.
- It is assumed that the activated sludge models give the correct representation of the actual processes.
- The reduced models do not change the main dynamic behaviour of the biological activity of the microorganisms and of the wastewater component concentrations.

1.11   Research methods

In this research a method for solution of the optimal control problem of the activated sludge process (ASP), specifically for the Athlone and Benchmark wastewater treatment plants is development. The objective is to determine a method for calculation of an optimal dissolved oxygen trajectory. The optimal dissolved oxygen trajectory is the one that gives minimum effluent concentration while consuming less control energy.

The research includes the development of the ASM1, ASM2, ASM2d and ASM3 reduced biological models of the ASP, Benchmark and Athlone plants mass balance models and method for the optimal control problem solution. The method is based on the optimal control theories, which require mathematical process modelling. The process modelling requires the implementation of mass balance theories. The method is evaluated on the basis of the benchmark environment that incorporates the usage of predicted influent conditions and the calculated effluent results based on those influent conditions. The achieved solutions are evaluated through simulation under the following conditions:

- With average values of the inflow and effluent concentrations
- With varying values of the inflow and influent concentrations
- With the obtained optimal control DO trajectory
- With a constant value of the control signal

As most of the wastewater treatment plants in South Africa are manually operated and therefore inefficient, the method for optimal control calculation should be flexible so as to be possible to be edited for use on different wastewater treatment plants in the country. This will improve the performance of wastewater treatment in South Africa.

Research methods used also include:

- Literature review on different advanced control systems for the ASP, optimal control theory and process modelling. This also includes a comparative analysis of the existing strategies and results from different publications.
- Mathematical modelling of the dynamic behaviours of the activated processes.
- Simulations for studying of the dynamic behaviours of the ASP
- Graphical user interface to show a life status of the process being controlled.
- Experimental method for studying of the various possibilities in terms of the wastewater treatment plant behaviour during different inputs.
- Description methods are used for the system description with all its specifications, characteristics and functions.
- A discussion of the results allows for explanation and comparison of the control strategy with respect to the research objectives.

1.12 Thesis outline

In this chapter, an introduction to the wastewater treatment processes is covered.
Also covered are the following: statement of the problem, aim and objectives, hypothesis, delimitation, assumptions, motivations and research methods. The rest of the chapters in the thesis are broken down as follows:

Chapter 2 presents a literature review. This includes biological model reduction methods, different activated sludge control strategies and possible control variables.
Chapter 3 presents detail information regarding optimal control theory as well as typical control strategies used in the activated sludge control studies.

In Chapter 4, a detailed description of the four activated sludge models implemented in this project is given. The models are reduced based on the technique in Lukasse (1996). The mass balance models of Athlone and Benchmark activated sludge processes are derived.

Chapter 5 includes a description and implementation of parameter estimation technique for the reduced order models. Then the algorithms for simulation of the models and parameter estimation are described. The results are given and discussed.

13

Chapter 6 describes methods used in developing the optimal control strategy for activated sludge processes. This strategy is then applied to the reduced Athlone and Benchmark mass balance models in MATLAB simulation environment. Simulation results are then discussed.

Chapter 7 presents the conclusion and significance of the strategy. This chapter also includes discussion regarding limitation of the strategy, possible areas of applications and future expansion of the project.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

The purpose of this chapter is to give an overview of different strategies used for advanced control of the activated sludge process (ASP). Focus is given to strategies used to control dissolved oxygen concentrations in the ASP. First, the most known model based advanced control strategies are described. Then a comparative analysis of the strategies is implemented to establish which the most convenient method is.

A survey of the ASP models is then done. Because of the complexities of these models, model reduction techniques are studied. To ensure that reduced models give the same prediction capabilities as the full models, parameter estimation techniques are also studied. Comparative analyses of both model reduction techniques and model parameter estimation techniques are done with the objective of determining the ideal techniques to use.

Finally a survey is taken on the specific application of advanced control methods to the ASP. The process can be controlled by varying certain inputs and outputs such as the sludge recycle and wastage rates and aeration. A brief retrospective on few key areas of development regarding aeration regulation is given.

2.2 Activated sludge process control

The function of a wastewater treatment plant (WWTP) is to speed up the natural process by which water purifies itself. The main objective of a WWTP is to produce an effluent meeting the required standards at the lowest possible cost. In this work, the wastewater treatment plant based on the activated sludge process (ASP) is considered.

The ASP is a widely used system for biological wastewater treatment. Traditionally the ASP mainly served to remove organics and ammonium from the wastewater. A basic design of the ASP consists of an aeration tank and a secondary settler. In the aeration tank the organics in the influent wastewater are transformed into settle-able biomass (activated sludge flocs) by biomass growth. In the secondary settler the sludge settles and is withdrawn from the bottom while the treated effluent flows over the weir at the top. The major part of the settled sludge is recycled back to the aeration tank and the rest is wasted. Sludge recycling prevents biomass washout in the reactor.

During the last 2 decades the interest in total nitrogen (N) and phosphorus (P) removal from wastewater has risen due to increasing effects of eutrophication in the aquatic environment. N removal involves the use of two biological processes; nitrification and denitrification. Nitrification takes place under aerobic conditions while denitrification requires anoxic conditions. The nitrate reduction depends on the level of DO concentration. Lower DO levels denitrification in the aerated zones possible. The naturally existing carbon in the aerated zones is then in a larger extent used for denitrification instead of being oxidized. Less oxygen is also re-circulated when using a low DO set-point, which improves denitrification in the anoxic zone.

To cater for N removal, the basic ASP structure is redesigned to include an anoxic reactor. Phosphate removal requires the presence of an anaerobic tank prior to the aeration tank. The ASP utilises bacteria within wastewater to remove organic matter, nitrogen and phosphorus. These bacteria require oxygen in order to oxidise and transform these wastewater contaminants into harmless compounds. In the ASP, oxygen is introduced by blowing air into the system. During oxidation, bacteria use energy to maintain their living function and to circulate.

From the above discussion, it is clear that the concentration of microorganisms in the aerated tank determines contaminants concentration in wastewater. It is also noted that microorganisms' concentration is affected by the sludge recirculation and the concentration of dissolved oxygen (DO) in the water. To help keep microorganisms at an optimal level, several control strategies are available. These include control of DO level, aeration volume control and internal recycling flow rate control. In this project, DO level control is considered as DO has direct influence on the effluent concentration and on the process operational cost.

2.3 Dissolved oxygen control

In the degrading process, large quantities of oxygen are consumed by the bacteria. This often results in a severe lack of oxygen. The dissolved oxygen (DO) concentration should be sufficiently high to cover the microorganisms' oxygen demand. On the other hand, a too high DO level may hamper the efficiency of processes that require anoxic conditions. Also, unused oxygen is wasted as it disperses into the atmosphere. This leads to high energy consumption due to the associated high air flow rate. For this reasons it is of interest to control the level of DO in the process. In most cases the DO level is kept at a

constant pre-specified value by adjusting the air flow rate. This is not ideal because the activated sludge process is subject to high variations in terms of influent wastewater concentration and flow rate. A feasible extension is to use a time varying set point for the DO level by applying cascade control. An advantage of using a variable DO set point is that only a minimal required amount is used resulting in energy savings. The disadvantage is that there may not be extra air to cater for mixing purpose. This drawback is solved by controlling the aeration volume; this part is not covered in this project.

Many South African wastewater treatment plants are equipped with classical control strategies that incorporate on/off or Proportional, Integral and Derivative (PID) control. Because of the influent wastewater variations there is a great need to consider the dynamic behaviour and the operational characteristics of the ASP in the control strategy. This requires the use of advanced model based control strategies. Doing so will ensure that an improved performance of WWTP can be obtained.

The drawback in applying advanced control techniques to the ASP is the lack of fully reliable online sensor for process monitoring. There are also lack of methods for measuring important variables required for modelling, parameter estimation, optimization and control calculation. The other challenges are due to the fact that the process is time varying, nonlinear and multivariable. The process's stiff dynamics (time constants which range from seconds to months) also make it hard to control the ASP. However this can be solved through the use of cascade control where fast controllers control faster sub-processes and slower controllers control the set point of the fast controllers.

Because of these drawbacks there has been little interest in using advanced control strategies in wastewater treatment. However the development towards tighter effluent demands have contributed to an increasing interest in applying advanced control strategies for WWTPs. Furthermore there has been a development in the availability of sensors for wastewater treatment monitoring; an overview of the existing sensor technology was given in 2003 by Vanrolleghem and Lee.

The application of modern control systems has proved a great potential in dealing with the complex dynamics of the ASP and the disturbances affecting it.

Modern control systems have enabled for savings in energy and less use of chemicals. Such system is applied and reported by Carlsson (1997).

These control systems are able to deal with nonlinear multi input multi output processes with uncertainties. They are based on time domain, state space representation of the processes they control.

That is, a process is described using a set of nonlinear mathematical equations. Nonlinear high order processes such as the activated sludge processes are described by high order differential equations. Modern control systems have the capability to design and predict behaviours of the complex nonlinear processes. Some benefits of using advanced control strategies include possibilities to; meet effluent demands. achieve cost efficient solutions and reduce impact of the process disturbance.

Modern control systems are based upon the optimization of a cost function. This field is referred to as optimal control. In this field, a model of the process is often used to predict the effects of any control signal. Then based on the selected cost function the best control signal is chosen. The advantage of optimal control is that it provides necessary and sufficient conditions which must be satisfied for the design problem to have a solution. This means that in situations where the design requirements cannot be satisfied, the designer will not search in vain for solutions of the problem.

Process optimisation is classified as either offline or online. Offline process optimisation refers to applications where offline simulations with the calibrated model are used to determine optimal control signals. The result may then be implemented on and tested on the full-scale plant. This is required because of strict demands imposed on existing wastewater treatment plants whereby the initially required effluent quality cannot be obtained any longer. Hence simulations are often used to evaluate whether the pollutant removal efficiencies can be improved via improved process control. Online process optimization simulations with the calibrated model are applied in an online optimisation method, such as model predictive control (MPC) algorithm.

Model based control systems include:
- Model predictive control
- Model adaptive control
- Optimal control

In these methods, classical feedback control loop is sometimes used independently or together with a feed forward control loop. In feedback control, measurements or estimates of the control state are used to calculate suitable values of the control variables. In feed forward control, the process disturbance is included in the control law thereby improving the disturbance attenuation.

Modern control incorporates implementation of controllers in computers and uses advanced algorithms for calculation of the control signals.

2.4 Model based control systems

2.4.1 Model predictive control

Many of the ideas discussed in this section are from Camacho & Bordons (2004).

Model predictive control (MPC) is one of the advanced methods of process control used in complex multi input multi output systems. MPC uses a dynamic model of the plant to predict the future effect of the manipulated variables on the plant output. A dynamic model requires past process input and output knowledge to do predict future process output. The model predictive controller uses the predicted output and current plant measurements to calculate future optimal trajectory of the process input (control variables). Optimal control trajectory results in process operation that meets process input and output constraints. The model predictive controller then sends the calculated or optimised process input to the controller to be implemented on the process.

A basic structure of a MPC is presented in Figure 2.1. As mentioned above, the model takes current and future process measurement and predicts future process output. The predicted process output goes to the optimiser where it solves the optimization problem. The optimiser incorporates a criterion function, constraints and history of past variables. The optimiser checks and minimizes the difference between the predicted output and the desired output. The optimiser then generates and predicts the optimal process input values that are sent to the controller to be implemented on the process.

MPC uses iterative optimization approach to solve this optimization problem. At a particular time, the current process input and output are sampled and used to compute future optimal inputs and outputs. These optimization calculations are performed at each moment in time using the latest measurements. Only the first calculated values of the input variables are sent to the controller.

19

The prediction horizon is then shifted forward and the procedure is repeated every time. The new predicted input is also sent to the model to be used for prediction of future output. Because of this time shifting, MPC is often referred to as receding horizon control. As seen from the figure below, MPC is a feed forward kind of control system. The model predictive control is effective on systems with large number of constrained inputs and output. Its advantage is its ability to be applied online. The disadvantage is that it is more complex to design a model predictive controller; all the constraints have to be incorporated into a single objective function before it can be optimised.
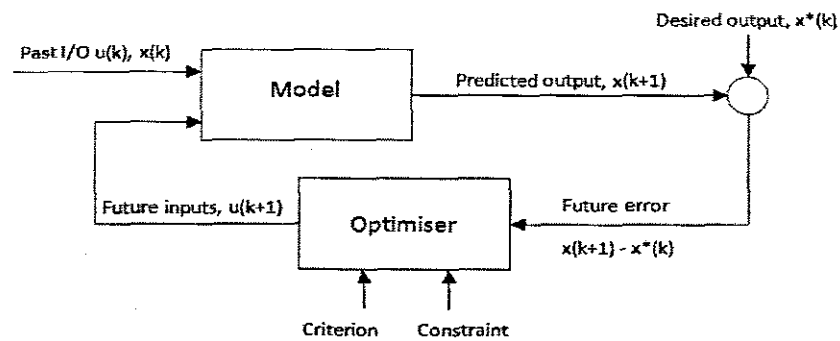
Figure 2.1: A basic structure of a model predictive control system

(Adapted from Camacho & Bordons, 2004:4)

The application of model predictive control on an activated sludge process is reported by Steffens and Lant (1999), Samuelsson (2001), Maciejowski, (2002), Qin and Badgwell (2003), Zhang et al. (2003), Chotkowski et al. (2005), Holenda et al. (2005a), Holenda et al. (2005b) and Stare et al. (2007). In most of these cases, simulations results show a rejection of influent disturbance. In cases where dissolved oxygen (DO) was chosen as the manipulated variable, the results show low aeration energy requirements thus low operational costs. The importance of MPC is that it allows the enclosure of constraints on the inputs and outputs.

The work of Steffens and Lant (1999) is based on the assumption of a multivariable control problem rather than focusing on DO control. However this method may also be applicable to oxygen control of wastewater treatment plants (Holenda, 2007).

2.4.2 Adaptive control

Many of the ideas discussed in this section are taken from Tao (2003). Adaptive control involves modification of controller control law to be able to deal with slowly time varying process parameters. In other words, the control law must adapt to time changing process conditions. The closed loop controller has parameters that are updated to change the response of the system. A reference model is used to capture the desired response, and then the desired response is compared to the system output. The controller parameters are updated based on the difference between the desired and the system outputs. The goal is for the parameters to meet the ideal values that cause the plant response to match the model response.

A typical structure of an adaptive control system is presented in Figure 2.2. A model takes a control input and predicts the process outputs. The predicted output is then compared to the actual process output and the error is used to calculate parameters of the controller. The adjusted parameters are sent to the controller to be implemented on the process in order to minimise the error. This control strategy boosts the advantage of being able to reduce controller response time while eliminating oscillations. Adaptive control systems provide quick adaptations for known inputs. A disadvantage is that it may have trouble adapting to unknown processes or random disturbances (Spooner et al., 2002).

The application of model adaptive control on an activated sludge process is reported by Lindberg and Carlsson (1995), Lindberg and Carlsson (1996a), Lindberg (1997) Olsson and Newell (1999), Chotkowski et al. (2005). The controllers gave good performance and adapted well to changes in the process. Lindberg (1997) evaluated the strategy in a pilot plant and the controller in this case managed to control the specified concentration very well. In 2004, Smets et al. gave an overview of optimal adaptive control of biochemical reactors. Smets et al. concluded that adaptive controller does not guarantee optimality of the results obtained.

Youssef and Dahhou (1996) and Nejjari et al. (1999) presented a study on the application of adaptive control laws to waste treatment plants. Dochain (1992), Polihronnakis et al. (1993), Monroy et al. (1996), Dochain et al. (1997), Emmanouilides and Petro (1997) reported adaptive control in waste treatment plants that uses anaerobic digester processes.
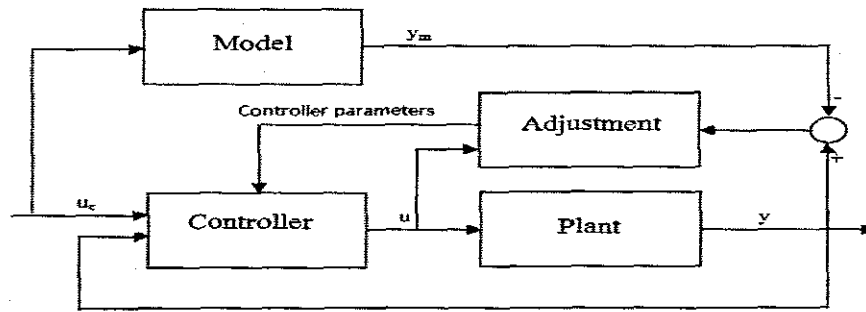
Figure 2.2: A basic structure of an adaptive control system

(Adapted from Mosca, 1995:237)

2.4.3 Optimal control

Many of the ideas discussed in this section are taken from Kirk (2004) and Bernard (1999). Optimal control system are systems in which control variables are determined to maximize or minimize some measure of the performance of a system while satisfying specified constraints (Christophersen, 2007). Optimality is achieved by making use of state space representation of the model. In the state space approach, to each state variable a feedback parameter is assigned. These parameters are solved in order to achieve the desired trajectory of the system's response. With optimal control, the parameters are often chosen in order to minimise the selected performance index.

Different approaches to achieve optimal behaviour are available. These approaches require solution to a set of partial differential equations. The objective is to find a trajectory that optimises the selected performance index. This means that there may be different solutions to the problem. As a result this process is iterative and often requires many successive attempts before the desired trajectory is found. This point is demonstrated in Figure 2.3.
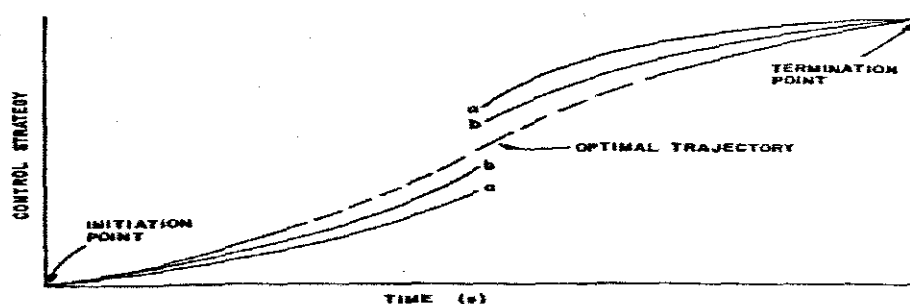
Figure 2.3: Successive attempts before an optimal path

(Adapted from Massachusetts institute of technology, nd:28)

22

Because of iterations mentioned above, solution to the optimal control problem is calculation intensive. This means that for complex problems, calculations may take longer than the time available during each sampling interval. Hence many optimal control problems are solved prior to the initiation of the control action. The solution to the optimal control problem is therefore usually not a feedback form but rather a sequence of signals that are applied at each sampling interval. The disadvantage to this approach is that this sequence will be valid only for the boundary conditions that were initially known. There is no provisions given for those conditions and there is no continuous feedback of measured variables. In other words this is a form of open loop control. If the process model is accurate and there is no change in the boundary conditions, then the resulting control action will be as desired. Otherwise an incorrect trajectory will be generated. The advantage of this approach is that it allows for achievement of a desired response without having to specify details of the trajectory. Another advantage is that this approach is applicable to nonlinear dynamic systems.

Lindberg (1997) determined that the one of the disadvantages of using optimal control strategy is that to calculate the control signal is very computationally intense. Simulation results by Nejjari et al. (1999) showed the controller efficiency in regulation, tracking and disturbance rejection. Lukasse et al. (1998 and 1999) determined that optimal results using this strategy were possible. Keesman et al. (2002) achieved good performance in both simulation and pilot plant application. Schütze et al. (2004) gave a review of optimal control in the context of integrated modelling which included research in single objective optimization.

Rauch and Harremoes (1999) considered such single objective optimization where two different manipulated signals were considered separately but then compared to find a trade off between them.

Schütze et al. (2002a) presented various objectives based on DO and ammonium concentration with a single objective chosen in the optimization of the process. In Vanrolleghem et al. (2005) ammonia concentration is selected as the objective to derive the optimal control strategy. In Schütze et al. (2002b) optimization of various objectives where finally introduced. Fu et al. (2008) also applied this strategy to multivariable objectives optimization.

This is relevant since, in practice a control strategy is required to meet multiple and possibly conflicting objectives, in order to meet the many needs placed on receiving waters. In this project the integrated wastewater process is not considered therefore the control strategy is limited to a single objective optimization.

2.4.4 Comparative analysis of modern control systems

In summary, all the control strategies that are described above produce good performance and disturbance attenuation. The optimal control strategy stands out due to the fact that it provides necessary and sufficient conditions which must be satisfied for the design problem to have an optimal solution. The other two strategies may produce results which are not optimal or may let the iteration procedure run forever because it may never reach a solution. There are available computer programming platforms which are to deal with the downfall of computational intensity required to solve the optimal control problem. The optimal control strategy is studied in details and applied to the activated sludge process in this project in Chapter 4 and 5 respectively.

2.5 Activated sludge models

As discussed in the previous sections, modern control systems are based on a mathematical representation of the process they control. Mathematical models have become a major tool to design closed loop control schemes. These mathematical models are useful in the design and tuning of such control strategies because the possibilities to evaluate a strategy in practice are usually very limited. This means that with mathematical models, different control laws can be easily tried and the parameters of the selected control law can be tuned without risk of disturbing the system (Vanrolleghem, 2002). With simulation, the current (and predicted) states of the effluent wastewater can be used to determine the control actions in the activated sludge process (ASP).

Most of the literature regarding the ASP is based on models developed by the Institute of Water Association (IWA). These models developed by the IWA task groups reflect the current understanding of the key biological and physicochemical processes that take place in such complex environments. Henze et al. (1987) produced the first IWA model called the activated sludge model no. 1 (ASM1) in 1983. A thorough description of the ASM1 may also be found in Jeppsson (1996) and Britta (2000).

24

ASM1 represents the biochemical processes occurring within the biological reactor at wastewater treatment plants that use the ASP. The model describes nitrogen and chemical oxygen demand within suspended-growth treatment processes, including mechanisms for nitrification and denitrification.

The second IWA model called the activated sludge model no. 2 (ASM2) was produced by Gujer et al. (1995). ASM2 is an extension of ASM1 incorporating mechanisms for biological phosphorus removal at a wastewater treatment plant. The third IWA model called the activated sludge model no. 2d (ASM2d) was produced in 1999 (Henze et al., 1999). ASM2d describes simultaneous phosphorus removal as well as nitrification-denitrification. In the ASM2d chemical phosphate removal is modelled and the behaviour of phosphate-accumulating organisms (PAO) is not describes. This is the difference between ASM2d and ASM2.

The latest IWA model called the activated sludge model no. 3 (ASM3) was also developed in 1999 by Gujer et al. (1999). ASM3 incorporates oxygen consumption, sludge production and nitrification-denitrification processes of the ASP. ASM3 is an extension of ASM1 where the main difference is the recognition of the importance of storage polymers in the heterotrophic conversion.

ASM1 is the most used of all the IWA models. According to Nelson and Sidhu (2008), the ASM1 have been found to give a good description of the ASP provided that the wastewater has been characterised in detail. All the four IWA models are considered in this thesis. The latest summary of all the IWA models can be found in Henze et al. (2000).

A few more models for the ASP have been developed based on the concepts used in the IWA models, these include ASM3_2N (Lacopozzi et al., 2006) and TU Delft model. TU Delft model is a phosphate removal model developed at the Technical University of Delft. This model is described in more basic terms than the ASM2d and it also has lower number of parameters.

Mathematical models and computer simulations are necessary to describe, predict and control the complex interactions of the processes. The number of reactions and microorganism that are mixed up in the process may be very big.

As a resut an accurate description of such systems can result in highly complex models, which may not be very useful from a realistic point of view. For this reason these models are often reduced to an order that is more applicable from the control point of view.

2.5.1 Activated sludge model reductions

Some of the ideas discussed in this section are taken from Jeppsson (1996). Activated sludge models have been used mainly for process simulation and optimization of new plant design. They are useful tools for operator training, teaching and research purposes. These models describe the nonlinear and complex behaviour of the activated sludge processes. The complex and highly non-linear nature of these models are due to the Monod type expressions they use (Janssen et al., 2000). Due to these complexities, the IWA models are hard to calibrate and to use directly for control purposes.

When they are used for control purposes, problems arise with respect to the time needed for tuning the controllers, since the optimized settings for the controller are found by performing multiple simulations.

According to Jeppsson (1993), the complexity of the models causes a severe problem of identifiability and verifiability. This is due to the fact that several parameter combinations can often explain the same dynamic behaviour. For example, a change in the quality of the influent wastewater composition can quite often translate to kinetic parameter changes. In other words the model may be adjusted to measurement data using different combinations of parameter values and influent wastewater quality.

Major problems of the models come from their complex structure and the large number of states and parameters to be uniquely identified. Instrumentation problems also amplify these difficulties.

Previously there were not enough available on-line instrumentation and laboratory procedures to verify the details of such complex models. Another problem is that results from experiments on full scale plants are rarely reproducible and the process inputs cannot be manipulated and controlled in the way that would be needed.

These models are also highly dimensional and this is another reason they are not appropriate for online implementation (Chachuat at el., 2002, David et al., 2006).

For these reasons the models are often reduced and simplified before use. Most of the literature covers reduction of ASM1. In this thesis all 4 IWA models are reduced before being used for calculation of the optimal control strategy.

2.5.1.1 Model reduction techniques

The IWA models are widely used models for simulation of the activated sludge process. However, these models are quite complex and may therefore be hard to use for control design. In this section, some model reduction techniques which give more feasible models for model state and parameter estimation and control designs are reviewed.

In previous years, model order reduction was often done based on physical understanding and intuition (Limebeer, 2007). Jeppsson and Olsson (1993) used such simplified and qualitative physical assumptions to reduce the ASM1 for the purpose of online parameter identification of the ASP. In the work of Jeppsson and Olsson, dynamics of states assumed to be measurable online were neglected together with the state variable chosen as a control variable. From the eight basic processes represented in the IAWQ model only four remained in the reduced model. Simplified model structures based on such assumptions were also presented by Vanrolleghem (1994) and Jeppsson (1996).

Janssen et al. (2000) presented a reduced ASM1 model based on similar assumptions used by Jeppsson and Olsson. The objective was to develop a model that increases calculation speed of the simulations required for dissolved oxygen concentration controller tuning. Evaluation of this model showed a good estimation of the full model. In work, simulation time of the full model took 3.5 minutes for a dynamic simulation of 3 months and 13.5 minutes when the controller was implemented. The resulting reduced model decreased simulation time by a factor of 3. A dynamic simulation of 3 months took only a minute to complete and took 4.3 minutes when the controller was implemented. As a result it can be seen that a DO controller tuning can be done more efficiently with a reduced model.

The other approach used in complex model reduction is the singular perturbation technique (Agathos et al., 2003). The singular perturbation technique enables to reduce state differential equations which are faster to algebraic equations and those which are slower to constants.

27

This is based on the fact that states with quick dynamics reach steady state quickly. To obtain algebraic equations, the derivative term on the differential equations is neglected or equated to zero. This method applied on bioreactor models is discussed by Weijers (2000).

Steffens et al. (1996) applied this method to a carbon and nitrogen removing ASP based on the ASM1. In this paper a homotopy method is first used to establish relationships between the states and the dynamics of the system via eigenvalue decomposition. Using singular perturbation analysis, components with the slowest dynamics were assumed constant while components with the fastest dynamics were reduced to algebraic equations. The result produced significant reduction of the integration time. The 13 states from the ASM1 were reduced to 4.

Chachuat et al. (2002) developed a reduced ASM1 model based on similar method to the one used by Steffens et al. (1996). The model was used to describe the behaviour of alternating activated sludge treatment plants, with the aim of applying it to the optimal control of an aeration system. A combination of the reduction method with process specifications reduced the model to 5 state space dimensions. Comparison between the dynamic behaviour of both reduced and full ASM1 models shows accurate predictions over 8 hours.

Limebeer (2007) developed a procedure to automate model reduction process. In this procedure, a model reduction problem is formulated. The problem is to find a low order approximation of a high order time invariant model by minimising the error between the two models. The model reduction technique is called state space truncation. This is similar to singular perturbation technique where unimportant states (states corresponding to fast modes) from state space models are removed. States that correspond to fast modes are related by eigenvalues with negative real part.

Kalman's minimal realization (Shutter, 2000) is another well known model reduction technique. Kalman's minimal realization theory is shown to be an important tool to understand the system's structure. The Kalman's algorithm is based on the theory of controllability and observability and reduction of a non-minimal state-space realization until it became minimal.

Hann and Edgar (2002) presented a balancing approach to determine a reduced order model of a nonlinear system. The procedure introduces the concept of covariance matrices for local controllability and observability analysis of a nonlinear system. These covariance matrices are an extension of gramians of a linear system and were used to determine unobservable and uncontrollable parts of the system. Then an algorithm that eliminates these parts of the model further reducing the model was introduced. This model reduction procedure is simple to implement and can be applied to any system.

Aghaee et al. (2002) used a similar balancing technique on linear time-invariant systems. The technique is based on a conceptual viewpoint regarding the balancing of the controllability and observability gramians of a multivariable system in a given range of frequency. According to Aghaee et al, from a real-time applicability point of view, the frequency-domain balanced structure provides a good approach to the model reduction of large scaled systems.

The simulation results establish the usefulness of this method compared to the other techniques. On the other hand, optimal model approximation based on the balancing method has shown to be a useful tool for model reduction. Singular-value-based balancing, Linear Quadratic Gaussian (LQG) balancing, co prime factorization and H-infinity balancing are important practical tools for linear model reduction. For nonlinear systems, there has been important progress with the nonlinear extensions of the systematic methods of balancing and co-prime factorizations, mainly based on energy functions. These methods yield adequate procedures for nonlinear model reduction.

A state of the art nonlinear balancing theory is discussed in the Control systems Engineering Research report (2002). In this report a link between accuracy of function approximation with a neural network and the stability of the system is established, leading to a robust model-based control scheme using a nonlinear (neural network) model.

2.5.1.2 Comparative analysis of model reduction techniques

The optimal control problem is often computationally intensive; therefore the main objective is to obtain a model that increases calculation speed of the simulations. After a review of the reduction techniques, it is clear that it is much simpler and effective to just rely on intuition and physical understanding of the process.

29

The other reduction techniques are very involved and require a lot of time and effort to grasp.

The main objective of this project is to develop a method for optimal control of the process; model reduction is a secondary task. For this reason care was taken to put more effort and time on the main objective, and to choose one of the already proven model reduction techniques. Lukasse (2001) have done a lot of work and produced reduced models of the activated sludge model no. 1. Lukasse's model reduction technique is used in this project for the reduction of ASM2, ASM2d and ASM3, see Chapter 5.

Once a reduced model is obtained, it is necessary to validate that it accurately gives the same prediction capabilities as the full model. Model validation requires an optimization procedure in which the difference between the full and the reduced models are minimised by calculating parameters of the reduced models. In the next section, parameter estimation techniques are reviewed.

2.5.2 Parameter estimation techniques for model validation

To check that the reduced model accurately represents the full model, the models are supplied with same inputs and default parameters. If the model outputs are different then the next step is to apply model identification and then model validation procedures. In model identification; the reduced model is calibrated using full model output data and then used to calculate the kinetic and stoichiometric parameters. The full model and the reduced model with new parameters are then compared to validate that they produce similar outputs. Simulation using the set of parameters obtained should accurately reproduce full model results.

A simple procedure for model identification and validation is shown in Figure 2.4. As mentioned before model identifiability deals with the problem of determining unique model parameters from a given input-output experiment with data acquired through simulation. A general structure for model identification is shown in Figure 2.5. The challenge facing model identification step is that some of the parameters may be identifiable while others are unidentifiable. A parameter is identifiable if there is a unique solution for the parameter.

30

On the other hand a parameter is unidentifiable if there are an infinite number of parameter values. This means that different sets of parameter values lead to the same model behaviour. In this case, it is therefore necessary to examine which combinations of parameters are the best.
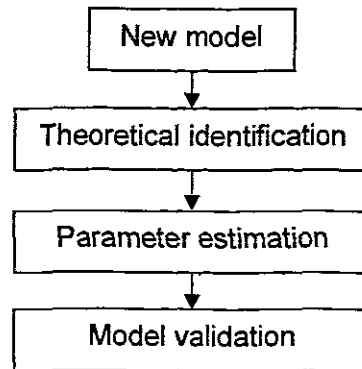
Figure 2.4: A simple model identification and validation methodology
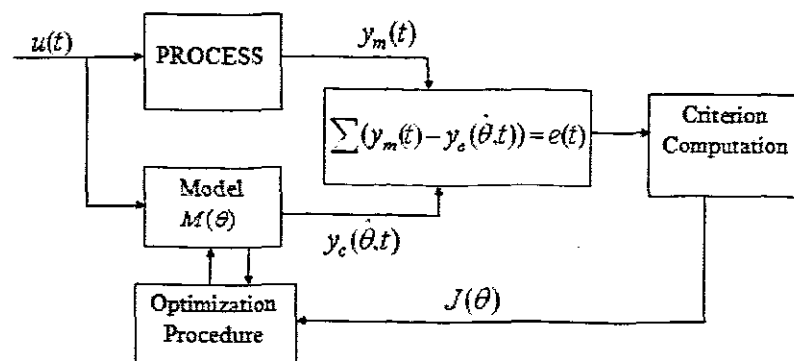
Figure 2.5: A general structure for model identification

(Adapted from Walter and Pronzato, 1995)

Parameter identification technique for non-linear systems is based on a Taylor series expansion of the measured variables with respect to time. The coefficients of the power series contain the model parameters. The law that governs whether these parameters can be determined from the Taylor series coefficients is represented by algebraic equations. Parameters are identifiable if the algebraic equations can be solved.

To obtain optimal values of the parameters, parameter estimation techniques are employed. When conducting parameter estimations, some of the model parameters are set at default values obtained from the literature and some need to be calculated. These parameters may be determined using the respirometric methods given in Vanrolleghem et al. (1999) and also in Peterson et al. (2000). In Lukasse (1999) identification schemes for simplified models are discussed.

Parameter estimation involves the determination of 'optimal' values of the parameters of a given model by making use of measured data. There are four important classical techniques used for parameter estimation; these are Bayesian estimation, Maximum likelihood estimation, Least squares estimation and Weighted least squares estimation. These techniques are used to fit a model to data and providing estimates for the model's parameters. A Bayes estimator is an estimator or decision rule that minimizes the posterior expected value of a loss function (Donald et al. 2006).

In maximum likelihood estimation, for a fixed set of data and underlying probability model, the values of the model parameters that make the data "more likely" are picked (Papadimitriou and Zhang, 2008). Maximum likelihood estimation gives a unique and easy means to determine solution in the case of the normal distribution and many other problems.

In least squares estimation, the best fit is that instance of the model for which the sum of squared residuals has its least value. A residual is the difference between an observed value and the value given by the model. Weighted least squares estimation differs from ordinary least-squares method in that it does not assume homogeneity of variance, but requires that the conditional variance be estimated prior to the estimation. The estimated variance needs not be constant across observations; they are treated as if it were the true variance when the error is computed. It is ordinary least squares if every observation is given equal weight.

Modern estimation (robust estimation method) incorporates the minimization of the median of the squares of the error instead of their sum. The disadvantage of robust estimation techniques is that they require more computation time. Methods for estimation are either conducted offline or online. When performing offline estimation, a complete set of data is available in time-series form.

Because of this form, it is then simple to apply the estimators given above to this data set. The idea is to perform simulation with constant parameter values over the time interval and finding the minimum of the performance index. From this point it is clear that parameter estimation is an optimization problem. There are different algorithms used for solving this optimization problem.

Peterson et al. (2001) present a systematic model calibration procedure for the Activated Sludge Model No. 1. It was underlined that it is very important to define the purpose of the model carefully since this will determine how to approach the model calibration. The aim of this study was to obtain a good description of the Nitrogen (N) removal capacity, since the model was to be applied for process optimisation in a later stage. It was thus important to have a good description of the process dynamics.

Koch et al. (1999) presented a calibration and validation of activated sludge model no 3 for a Swiss municipal wastewater. In this work, ASM3 was tested against experimental data as well as full scale experiments from various WWTPs. As a result of these tests a set of kinetic and stoichiometric parameters was obtained.

2.5.2.1 Comparative analysis of model estimation techniques

Maximum likelihood provides a consistent approach to parameter estimation problems and can be developed for a large variety of estimation problems. On the other hand the numerical estimation is usually non-trivial. Bayesian estimation requires a prior distribution of all unknown parameters, this is very difficult to achieve in practice. Weighted least squares estimation is based on the assumption that the weights are known exactly. This is however not possible in real applications. The main disadvantage of least-squares fitting is its sensitivity to outliers (extreme values that the response errors sometimes follow). According to researchers squaring the residuals magnifies the effects of extreme data points and as a result outliers have a large influence on the fit.

However there are ways in which to minimize the influence of the outliner. The advantage is that squaring the residuals places greater weight on parameters further from the original and makes all the values positive. Based on this analysis, the least squares method is identified as the convenient estimation method and is therefore used in this project.

2.6 Advanced control strategies for dissolved oxygen control

Several control strategies for the DO concentration have been suggested in the past. Although most of these strategies are implemented in simulation or on simple lab scaled plants, some real time implementation has been reported as well. One difficulty with controlling the DO is that its process dynamics are nonlinear. The gain of the DO process varies with respect to the airflow rate due to the nonlinear oxygen transfer function (K_La). K_La describes the rate at which oxygen is transferred to the system. From this it is clear that a linear controller may not be able to provide high control performance for all operating conditions.

Many high performance controllers for linear systems exist because of the extensive research done in controller design for linear systems. As a result of this there are many linear controllers reported for the activated sludge systems since the 70s. These include linear PI controllers with feed-forward from the respiration and the flow rates, dead-beat controllers based on a recursively estimated models with a linear K_La and controllers designs based on pole placement. Carlsson et al. (1994a) reported such controllers with auto-tuning based on online estimation of K_La. Stare et al. (2007) presented a proportional integral (PI) controller tuned from step response experiments. Given that the DO process is not a linear system its dynamics is often linearized before application.

Lindberg and Carlsson (1996) proposed a method which gives a high performance controller for all operating conditions. This strategy has two parts: parameter estimation part and controller design part. In parameter estimation the respiration rate and the K_La parameters were estimated. Respiration rate characterises the DO process and associated degradation of wastewater contaminants. A decrease in respiration indicates that toxics in the system. An estimation algorithm is used to find these two parameters based on measurements from DO and airflow rates.

And then a Kalman filter described by the nonlinearity of the airflow rate was used to track the time varying respiration rate and the K_La. The controller encompassed the nonlinear characteristics of the K_La. The controller design is based on the linearized DO process, obtained by using the inverse of the estimated K_La. A pole-placement design with a PI controller and a linear quadratic (LQ) controller which use feed-forward from the respiration rate were then used to control the linearized DO process.

According to Lindberg (1997) the advantage of a nonlinear controller compared to auto-tuning of PID-regulators is that it requires only one single tuning (estimation) event, while auto-tuning of a gain scheduling regulator may require several tunings for this type of nonlinear process.

Most of the control approaches are based on a constant DO set point (normally 2mg/l) control but in reality it is more practical to use a time varying DO set point. Both strategies have benefits and disadvantages. A constant DO set point may lead to unnecessary use of aeration energy during low loads. A varying DO set point only caters for what is necessary for the biological reactions but not for the mechanical activities such as mixing. A varying DO set point is chosen as the most convenient option in this project because of its ability to reduce aeration energy related cost.

With a time varying DO set point a fair trade off for the DO concentration in the aerobic tank is accomplished. In Lindberg (1997), Rehnstrom (2000) and Stare et al (2007) a DO set-point controller is suggested where the set-point is selected so that a pre-specified ammonium concentration in the last aerated zone is obtained see Figure 2.6. In some cases the set-point controller was implemented in the pilot plant and proved the possibility to not only save energy due to a lower DO set-point but to also reduce the effluent nitrate concentration.
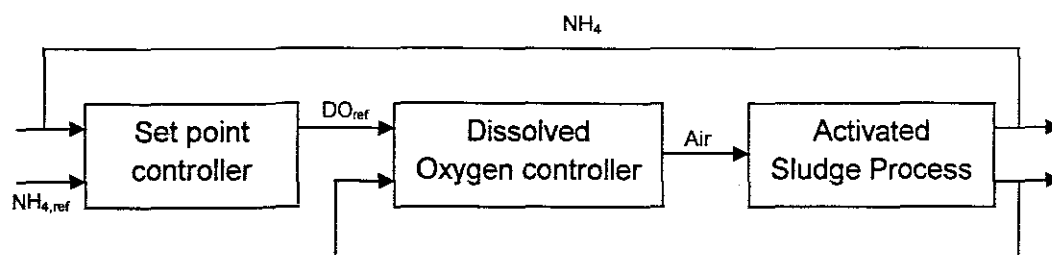
Figure 2.6: Cascade DO control structure
(Adapted from Lindberg, 1997)

Because of the demands on strict effluent quality and the need to reduce the impact of the many ASP disturbances, the energy consumption and the use of chemicals is bound to increase. It is therefore important to incorporate minimization of operational costs in the control strategies. To attain high controller performance at a low cost requires the use of more advanced model based control strategies in which effects of the disturbances are considered in the control law.

35

The controller set points could then be optimised in order to ensure that the process is running in an economically beneficial zone. Yuan et al. (1997), Lukasse (1999), Olsson and Newell (1999), Weijers (2000), Ingildsen (2002), Ingildsen et al (2002), Zarrad et al (2003) and Yuan and Keller (2003) propose several model based automatic control strategies. Lindberg (1997) applied a nonlinear linear quadratic (LQ) controller with feed forward from influent DO and influent respiration rate to the linearized DO process while Zarrad et al (2003) used a Linear Quadratic Gaussian (LQG) controllers.

Online minimisation of a multi objective criterion describing the operational cost and control performance is also being reported in some papers. In this approach the demand on the effluent water and the operational cost are incorporated in the control strategy through a cost function. The effluent demands are often incorporated as a constraint in the control design.

Economic optimization of the ASP is reported in Vanrolleghem and Gillot (2002) and in Cadet (2004). Samuelsson (2005) presented an optimal operating ranges and economic control of the denitrification process. Chotkowski et al. (2005) investigated a nonlinear model predictive controller (NMPC) and a direct reference adaptive controller (DMRAC) for varying DO set point. Cadet et al (2004) presented a multi-criteria control strategy that ensures that cost decreases while performance is kept good.

The possibility to obtain simplified process models allowed for more applications of model based control strategies. This allows for the use of hierarchical control structures. Brdys and Zhang (2001) presented a three level hierarchical structure where the state trajectory is calculated on the high level, the DO trajectory is optimised on the medium layer and the DO control is implemented on the bottom layer.

Holenda et al (2007) and Caraman et al. (2007) presented and evaluated the control performance of a model predictive controller (MPC) on a simple ASP. The aim of the controller is to maintain DO concentration at a certain point. The predictive controller uses an internal neural network model that is able to handle the complexities of the ASP. Stare et al. (2007) also presented an MPC to calculate control trajectories based on cost minimization. Nejjari et al. (1999a) present nonlinear adaptive feedback linearizing control based on the nonlinear model of the process and the observer estimator.

Observer estimators are software sensor for online estimation of process parameters and states. These sensors estimate the process parameters which are then used in the control law to calculate a suitable control signal. Results of simulation of the estimation and control algorithms showed robustness against disturbances and varying kinetic parameters.

Nejjari et al. (1999b) present an optimal controller with prediction and adaptive capabilities and the control law that is based on time predictions of the controlled variables and minimization of a cost function. The technique is based on a nonlinear process model derived from mass balance equations and an estimator for online tracking of unmeasured state variables and time varying parameters. The estimated state and set points are used to form a reference models. The objective is to minimise the error between the reference model and the predicted output. This is captured by a quadratic cost function.

2.6.1 Comparative analysis of the advanced method for dissolved oxygen control

Most of the control strategies found in the literature are based on simulation. Only a few of those strategies are tested practically on pilot plants. This is a reflection of how much grounds still needs to be covered for real time implementation of these strategies on full scale plants.

From the literature survey, it is clear that control of dissolved oxygen (DO) concentration is the most significant strategies for controlling the activated sludge process. This is because the strategy has potential to reduce cost while keeping the effluent concentration within standards. This is the reason why DO control is chosen as the ASP control strategy in this project. Some strategies control DO concentration towards a constant DO set point (normally 2mg/l) while others employ a varying DO set point.

There are various approaches used to control DO in the ASP. These methods are compared in Table 2.1 and Table 2.2. Most of the controllers implemented perform well but the disadvantage is their slow adaptation towards disturbances. In this project the set point trajectory is calculated based on prediction of the inflow disturbance. This set point is then compared to the measured DO values in order to calculate the appropriate control signal. This ensures robustness against disturbances and correction of the control signal based on online measurements from the process.

37

Paper	Purpose	Controller, control strategy used	Controlled variable	Manipulated variable	Model	Simulation results	Real time results	Comments (Advantage & disadvantages)
Lindberg, 1997	Control of N removal ASP (Benchmark plant)	A pole placement design with a nonlinear PI controller, based on linearization of DO process, K_La taken into account	DO	K_La		Good disturbance rejection at low and high loads	Good performance and disturbance rejection	A combination of the nonlinear PI controller and feed-forward controller should produce good results under various conditions
Lindberg, 1997	Control of N removal ASP (Benchmark plant)	A linear quadratic (LQ) controller using feed forward from respiration rate	DO	K_La		Reject disturbance slower at high loads	Good performance	
Zarrad et al 2003	Improve ASP (operation and monitoring (Benchmark plant)	LQ G, optimal Disturbance accommodating controller law	K_La	DO	State space + disturbance models (ASM1)	Controller stabilizes the closed loop system, not very good results	N/A	
Chotkowski et al., 2005	Control ASP for nutrient and phosphorus removal	NMPC, SQP is used to solve NMPC optimization problem	DO	K_La	ASM2d	Excellent results	N/A	Combining the two controllers should further enhance results
		Direct reference MAC				Fair results require fast sampling which may be close to the limit		
Stare et al. 2007	Control of N removal ASP (Benchmark plant) w.r.t operating costs	PI, controller tuned from step response experiments	DO	K_La		Better control w.r.t. energy used, under aeration under high load		

Paper	Purpose	Controller, control Method	Controlled variable	Manipulated variable	Model	Simulation results	Real time results	Comments
Stare et al. 2007	Control of N removal ASP (Benchmark plant) w.r.t operating costs	Cascade PI controllers, SNH5 PI + DO PI controllers	SNH5 DO	DO set point K_LA		Better removal of ammonia, save energy		
Stare et al. 2007	Control of N removal ASP (Benchmark plant) w.r.t operating costs	Cascade + Feed-forward controllers, based on SNH5 and influent SNH	Influent SNH SNH5 DO	DO set point 1 DO set point 2 DO sp 1 + DO sp 2 = DO K_La		Reduces effluent SNH peak, better SNH removal		
Stare et al. 2007	Control of N removal ASP (Benchmark plant) w.r.t operating costs	MPC, calculates control trajectory based on minimisation of a cost function	SNH5 , SNO5	DO set point, external carbon flow rate	ASM1	Further improvements w.r.t ammonia,	N/A	
Holenda et al., 2007	Control ASP to meet effluent requirements (Benchmark)	MPC, parameter tuning by trial and error method	DO	K_La	ASM1	Good results depending on the parameters	N/A	The trial and error method for selecting parameters of the controllers is not the best option to have as this takes time and becomes difficult on systems with high disturbances
Caraman et al., 2007	Control ASP to keep substrate below 20 mg/l (Benchmark)	MPC, parameters chosen randomly	DO concentration	Dilution rate	ASM1	Control objective is fulfilled	N/A	

2.7   Conclusion

In this chapter, the existing activated sludge process (ASP) control strategies are reviewed. Firstly, the different types of model based control strategies are discussed, followed by a review of the existing ASP models. The most widely known IWA models are briefly described. All the four IWA models are considered in this research. Secondly, various model reduction methods and model validation methods are discussed. Finally, a review of the different advanced control systems applied to the ASP is presented. The advantages and disadvantages of each strategy are highlighted. This provides a basis to the reasons for choosing the control strategy used in this research project. A detail description of the optimal control strategy is given in Chapter 3.

CHAPTER THREE
OPTIMAL CONTROL THEORY

3.1 Introduction

Due to uncertainties around global economy, a lot of research is focussed at finding ways that ensure that the best system performance is achieved at a lower cost. The best performance depends on a large number of possibilities. Optimal control theory provides a means in which the best possibility can be identified.

Optimal control theory involves the formulation of an optimal control problem. The optimal control problem requires formulation of a cost function, a model of the controlled process as well as a mathematical representation of the constraints subjecting the process inputs and outputs. The cost function enables for the quantification of the effects of the possibilities which are predicted by the model. The basic optimal control problem is therefore to find the possibility which will minimise or maximise the cost function (Hersch, 2006). The model is a vital tool because it is not feasible to experiment with a real process to find out the effect of different decisions.

The main focus of this project is to develop a method for optimal control of the activated sludge process. Covered in the chapter is mainly the theory necessary for the development of this project. In this chapter, basic definitions of a system and different control structures are firstly recalled. Then the optimization procedure is described starting with a description of how the optimal control problem is formulated. The final sections cover the methods used for finding solutions to the general optimal control problems.

Both the model building and the optimization process require large amounts of calculations. For this reason the optimal control problem can be decomposed into smaller sub-problems that are easier to handle. Thus in this chapter, the methods for decomposition of the problem together with numerical methods for finding solutions are also covered.

3.2 System definition

3.2.1 Input/Output representation

Systems are usually represented by block diagrams in which the main parts of the system are represented by blocks that are connected by lines to show their relationships.

A system can be regarded as a black box where the internal processes transform inputs into outputs as shown in Figure 3.1 (Levine, 2000:66).
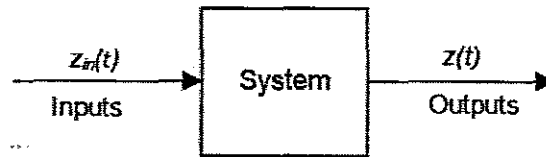
Figure 3.1: Black box diagram of a system

The process inputs are classified as either manipulated or disturbance variables and the outputs are classified as measured or unmeasured variables as shown in Figure 3.2 (Bequette, 2003:3).
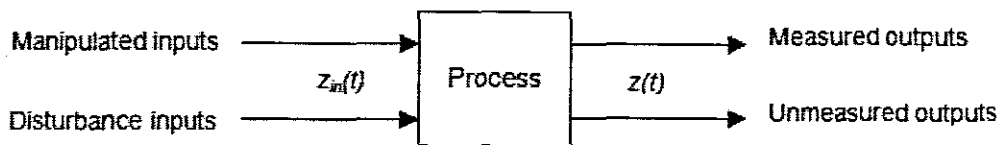
Figure 3.2: Extended block diagram of a system

Both types of inputs are capable of affecting the behaviour of the process and thus the process outputs. Process outputs are variables whose evolution is of great interest, they are often expected to follow pre-specified values called set-points. Manipulated inputs are inputs that can be adjusted to ensure that the process outputs reach set points. Often the inputs and outputs are not permitted to evolve arbitrarily but they are required to satisfy certain conditions called constraints. The outputs which satisfy the constraints are called admissible.

3.2.2 State space representation

In state space representation, a summary of the status of the system at a particular point in time is represented by what is known as state variables. The knowledge of the state at some initial time, and the knowledge of the system inputs after the initial time, allows for the determination of the state at a later time. The state variables are expressed as vectors; this simplifies the system in cases where there are a lot of state variables.

42

State space is the state at a given time belonging to an n-dimensional space of the values of the vector of the state variables as a function of time (Kanjilal, 1995:161-162). A state space representation is illustrated in Figure 3.3.
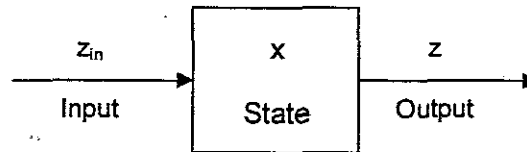
Figure 3.3: System state space representation

3.3   Control system definition

A control system is a mechanism used to alter the behaviour of another system by altering its manipulated inputs. A typical control system consists of a desired output, a controlled system, an actuator, a sensor and a controller. A sensor monitors the controlled system and provides its output to the controller. A controller then tries to get that output closer to the desired one by sending commands to the actuator for it to change the output of the system. The controller's internal control law or set of rules are used to determine these commands.

The outputs of the controller are referred to as the control signals or just controls. Control signals are variables that alter the process inputs until the process outputs reach set points. The evolution of the control variables over time are called policies or control trajectories. Control systems are classified into various categories. Open loop control system is a system that uses a controller to control the process but does not measure its outputs. In open loop control the process outputs have no effect upon the control signal (Liptak, 2006:137) as shown in Figure 3.4.
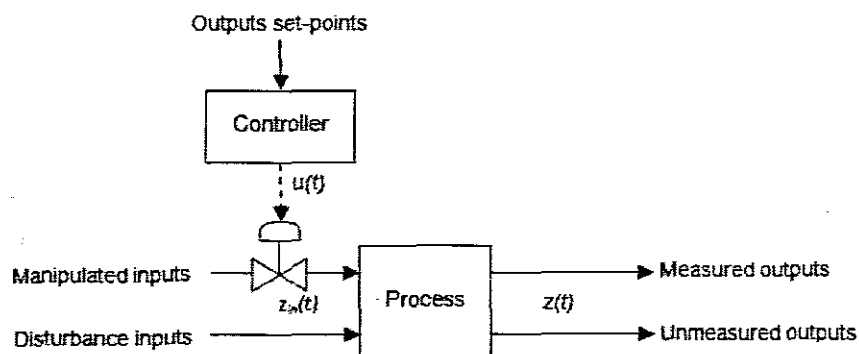
Figure 3.4: Block diagram of an open loop control system

43

Closed loop control system comprises a feedback link such that the control is a function of the output of the controlled system (Liptak, 2006:137), Figure 3.5. Model based control systems were originally developed for linear processes but have been extended for application on non linear systems (Kozak & Huba., 2004:171). When using model based control systems, the controller uses measured outputs together with measurements of the disturbance inputs to make decisions about the proper values of the manipulated inputs. Figure 3.6 (Bequette, 2003:3) illustrates this point.

In model based adaptive control, the control adapts itself as the model of the system changes. In hierarchical or multi level control systems, the control is distributed over many levels. Similar to the outputs, the controls are not permitted to evolve arbitrarily but they are required to satisfy given constraints. The controls which satisfy the constraints are also called admissible.
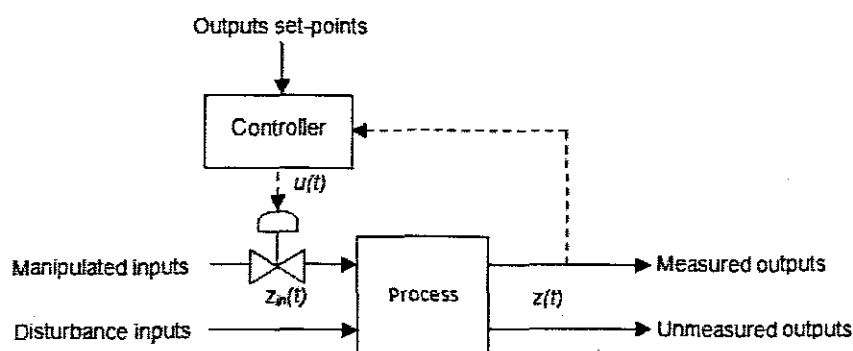
Figure 3.5: Block diagram of a closed loop control system
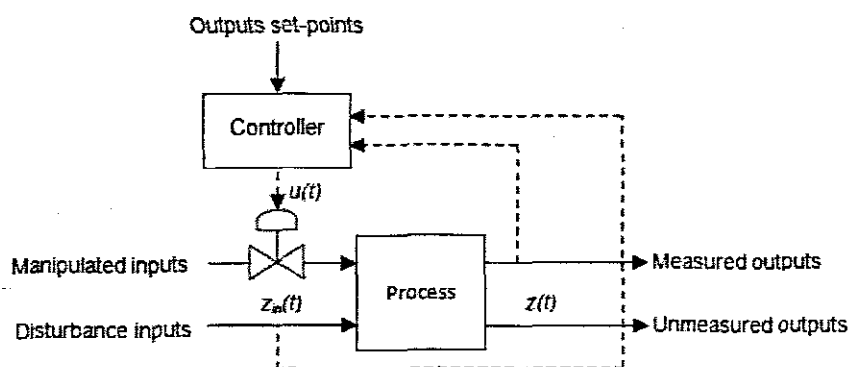
Figure 3.6: Extended block diagram of a closed loop control system

3.4   System optimization

The following information on system optimization was extracted from Titli and Singh (1978) and Naidu (2002).

It is possible to have a large number of control trajectories which enables the outputs to reach set points whilst satisfying the constraints. Optimization enables to choose the best amongst all these control trajectories. The best control trajectory is the one that meets the process objectives with less penalties e.g. effort, time, monetary unit, or whatever is defined by the user. This is referred to as the control objective. To enable the researcher to make a wise choice amongst the various possible admissible controls, it is necessary to formulate a cost function and a process model. A cost function is a quantitative expression of the control objectives. This is often referred to as the optimality criterion. An optimality criterion is a means to quantify the effects of the controls. A model is used for prediction of the effect of those decisions. The model is necessary because the number of possible decisions for one process is relatively large and it is often extremely expensive and sometimes dangerous to experiment with the system itself to find the effect of a particular decision.

Optimal control theory describes the use of controls to maximise or minimise the cost function. The optimization or optimal control problem is therefore to find the set of admissible controls which minimises or maximises the cost function by using the model. Solution to the optimization problem results in an optimal trajectory of the control variable or simply the optimal control trajectory. For dynamic systems with random disturbances, it is not possible to optimise the criterion with certainty; the best way is to work on assumptions of the disturbances.

3.4.1   Cost functions

The choice of the cost function is an important and difficult problem which is often based on intuition or experience. Cost functions fall into two categories; discrete objectives that are optimised at once, and continuous objectives that are maintained at optimal values during the course of time. For dynamic problems, the cost function depends upon the control trajectories and the evolution of the system in an integral fashion.

For discrete objectives the cost functions may be formulated when:

- Minimising the time required to transfer from one dynamic state to another

45

- Minimising the energy used in transferring from one state to another
- Minimising the error in arriving at the final state

For continuous objectives the cost functions may be formulated when:

- Regulating a system to remain near a fixed nominal condition in spite of disturbance
- Following a nominal path with minimum error, even though the system parameters have uncertain values
- Responding to arbitrary changes in the desired nominal state with well-behaved transients in state and control variables

Cost function often consists of weighting factors used as design parameters to help achieve desired result. This helps to guarantee the stability of the controlled system. This is good for systems with many dynamic states and controls or those which are inherently stable.

The state of the dynamic process evolves over time, therefore the criteria to be optimised is often expressed in terms of final values and the time integrals on the state and control variables. Two of the commonly used cost functions for dynamic processes are shown below.

$$J = \int_0^T (x - x^*)^T Q(x - x^*) dt \tag{3.1}$$

$$J = h\left(x(T) + \int_{t_0}^T g(x(t), u(t), t) dt \right) \tag{3.2}$$

where T is the time horizon; x is the state vector; x^* is a reference state vector or set point; Q is the weighting factor and u is the control vector. Equation (3.1) is a problem for minimising the mean square error and Equation (3.2) is the general form which is often used in dynamic optimisation. Problems with two different control objectives lead to different control trajectories. However there is only one cost function to be optimised. The cost function is therefore the weighted sum of the two objectives.

3.4.2 Process model

Optimization techniques generally consist of a mathematical procedure and for this procedure it is necessary to have a mathematical representation of the system under consideration. In other words the behaviour of the real process is represented by a set of mathematical relations which constitute a mathematical model of the process. The building of the model is called identification. The accuracy of identification could be measured by the difference between the output of the real process and that of the model.

Identification is the first step before optimisation can be performed since the results of the optimisation will be critically dependent upon the validity of the model. For this reason it is necessary to reach a compromise between the complexity (accurate representation of the process) of a model and simplicity of the identifiable model for the optimization phase. Identification is usually split into two phases; structure determination of the model and its parameter estimation. The structure of the model is usually determined from a priori physical knowledge of the process.

Parameter estimation enables us to put precise values on the parameters such that the model describes the real process and not a class of such processes. Parameter estimation requires experimental data of the inputs and outputs of the system. An adaptive modelling approach allows us to perform the structure determination and the parameter estimation simultaneously as shown in Figure 3.7. This technique is referred to as the least square estimation.

In least squares estimation, the unknown values of the parameters θ are estimated by finding numerical values for the parameters that minimise the sum of the squared deviations between the process states and the model states.
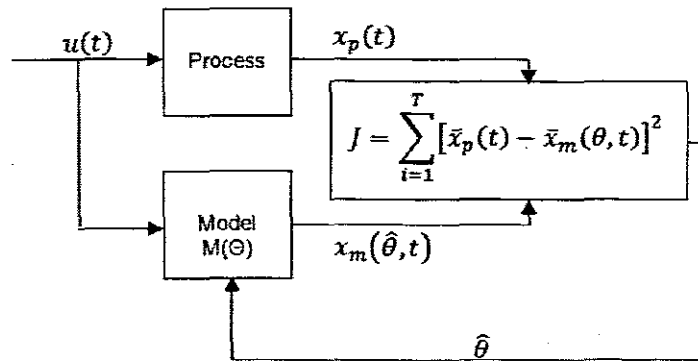
Figure 3.7: Least square estimation technique

Mathematical models consist of algebraic equations for static processes, integral-differential equations for dynamic systems, partial differential equations for distributed parameter systems and difference equations for discrete time systems. Linear dynamic time invariant systems are often described by Equations (3.3 & 3.4) (Naidu, 2002:2).

$$\dot{x}(t) = Ax(t) + Bu(t) \quad x(0) = x_0 \tag{3.3}$$

$$y(t) = Cx(t) + Du(t) \tag{3.4}$$

47

where x is an n dimensional state vector; A is an n by n constant matrix; u is an m dimensional control vector; B is an n by m constant matrix; y is an l dimensional output vector; C is an l by n constant matrix; D is an l by m constant matrix and x_0 is the initial state and it is assumed to be known. For time varying linear systems, A, B, C and D matrices are time varying as shown in Equations (3.5 & 3.6) (Naidu, 2002:101).

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad x(0) = x_0 \tag{3.5}$$

$$y(t) = C(t)x(t) + D(t)u(t) \tag{3.6}$$

A general nonlinear dynamic system is often described by the state space equations as shown in Equations (3.7 & 3.8) (Naidu, 2002:2):

$$\dot{x}(t) = f(x(t), u(t), t) \quad x(0) = x_0 \tag{3.7}$$

$$y(t) = f_1(x(t), u(t), t) \tag{3.8}$$

where f and f_1 are continuous differentiable vector function. Equations (3.7 & 3.8) are used to describe many physical systems since many physical systems are nonlinear in nature. A discrete time adaptation of the nonlinear dynamic system equations above is presented in Equations (3.9 & 3.10).

$$x(k+1) = f(x(k), u(k), k) \quad x(0) = x_0 \tag{3.9}$$

$$y(k) = f_1(x(k), u(k), k) \tag{3.10}$$

where k is the time instant and f_2 is the continuous differentiable. The state of the system at the instant $k+1$ is related to the state at the instant k by Equations (3.9 & 3.10).

3.4.3 Formulation of the optimal control problem

As mentioned previously, a mathematical formulation of the optimal control problem requires the following:

- A mathematical description of the process to be controlled
- A statement of the physical constraints
- Specification of the cost functional

The general nonlinear dynamic system Equations (3.7 & 3.8) and the cost function for general dynamic optimization in Equation (3.2) are used here to demonstrate the optimal control problem formulation.

48

The optimal control problem is therefore to find admissible control u^* which causes the system

$$\dot{x}(t) = f(x(t), u(t), t), \; x(0) = x_0$$
$$y(t) = f_1(x(t), u(t), t)$$

subject to the constraints $f_2(x(t), u(t), t) \geq 0$ (where f_2 is the constraints function) to follow admissible trajectory x^* that minimises the performance measure

$$J = h\left(x((T), T) + \int_{t_0}^{T} g(x(t), u(t), t) dt\right)$$

The cost functional is subjected to the model and the physical constraints of the inputs and outputs. Since there is more than one equation in the formulation of the optimal control problem, they all need to be combined to form a single equation having the meaning of the cost function and physically expressed by the same measures. The combination of the cost function and state equations result in an equation referred to as Lagrangian. The first requirement before combining the state equations and the cost function is to reformulate the state equations so they equal to zero.

Hence:

$$f(x(t), x(t), u(t), t) = 0 \tag{3.11}$$

The reformulated state equations and the cost functions are then adjoined by means of Lagrange multipliers as shown in Equation (3.12). This results in a problem that can then be solved to give the optimal solutions of the cost functional for the considered model equation.

$$L = h(x(t_f), t_f) + \int \left[g(x(t), u(t), t) + \lambda^T(t) f(x(t), u(t), t)\right] dt \tag{3.12}$$

where L is the Lagrangian and λ are the Lagrange multipliers. Lagrange multipliers are coordinating variables for the models and the cost functional. The optimal control problem is therefore to find u^* which minimises the Lagrangian according to state and control variables and maximise it according to the Lagrange's variables (multipliers).

3.4.4 Solution to the formulated optimal control problem

Once a mathematical formulation of the optimal control problem is obtained, indirect (not analytical) methods from classical theories are used to obtain solutions.

49

The most known of these indirect methods are mathematical programming, dynamic programming and variational methods (maximum principle and classical variational calculus). Since optimal control problems are complex and nonlinear, variational methods are employed.

According to Titli and Singh (1978), linear dynamic system problems are solved using variational methods, dynamic programming and gradient methods. The same techniques together with quasilinearisation may be used for nonlinear dynamic systems. Complex dynamic system problems are solved using decomposition-coordination methods in mathematical programming and in variational calculus, leading to hierarchical control and optimization.

Optimal control problems involve ordinary differential equations and therefore are difficult to solve analytically. These problems are better solved numerically using new computational algorithms. Numerical algorithms available include: Sequential gradient restoration algorithms and Control parameterization based algorithms. In the latter, the optimal control problem is discretized resulting in a standard mathematical programming. Sequential quadratic programming is then used to solve this problem.

3.4.4.1 Calculus of variation

The basic theorem used in finding an extreme value of a function is the necessary condition that the differential of the function vanishes at an extreme point (Naidu, 2002:19). The theorem is similar for a functional; the variation is zero on an extremal curve (Naidu, 2002:26). This point is illustrated in Equation (3.13); the differential of the Lagrangian in Equation (3.12) is equated to zero.

$$\partial L = 0 \tag{3.13}$$

Because the Lagrangian is a function of three variables (the states, the controls and the Lagrange multipliers), extreme points are obtained where partial variations with respect to each of the variables equals zero.

$$\partial L / \partial x = 0 \tag{3.14}$$

$$\partial L / \partial u = 0 \tag{3.15}$$

$$\partial L / \partial \lambda = 0 \tag{3.16}$$

Calculus of variations therefore provides first-order necessary conditions, Equations (3.14 – 3.16) which must be satisfied for optimality. Solutions to Equations (3.14 – 3.16) result in the optimal control trajectory.

50

3.4.5 Decomposition of the optimal control problem

The optimization phase requires a lot of calculations and these increases as the order or dimension of the problem increases. The order depends on the model being used and on the physical plant. The computation required for problems with high dimensions surpasses the capabilities of a single computer. .

To deal with such problems decomposition methods are used to breakdown large scale problems into a set of smaller and simpler problems. Decomposition-coordination methods and a multi level control structure may then be used for solution of the optimal control problem. A global control problem is usually represented as shown in Figure 3.8 (Titli and Singh, 1978:18). Such a controller is called a simple level simple objective controller. Its implementation is useful only in simple systems with less complexity.
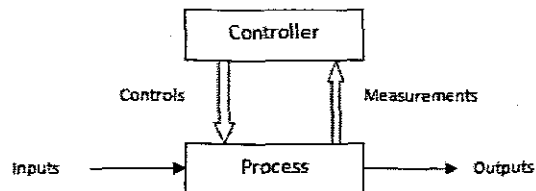
Figure 3.8: Simple level simple objective controller

For implementation on a complex system, the global problem may be considered to consist of independent subsystems and a simple level multi objective control structure could be defined as in Figure 3.9 (Titli and Singh, 1978:18).
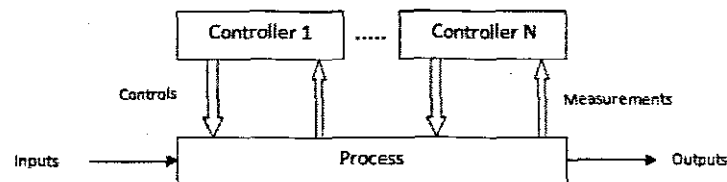
Figure 3.9: Single level multi objective controller

According to Titli and Singh (1978), since the independence assumption is clearly unrealistic, the solution obtained by the global controller is sub-optimal, and the degree of sub-optimality depends on the interactions between the sub systems.

51

When the interactions between the subsystems are strong and a criterion function is associated with each controller, conflicts could arise between the controllers if none of them have priority of action.

To resolve these conflicts, a second level of control which takes into accounts the interactions and modifies if necessary the objectives of any controller of the lower level is used. Thus the resulting control structure has multiple levels and multiple objectives. Basically the control is considered to consist of controllers which are distributed in a hierarchy having a pyramid structure such that on the first level there is a controller to control each one of the interconnected subsystems, Figure 3.10
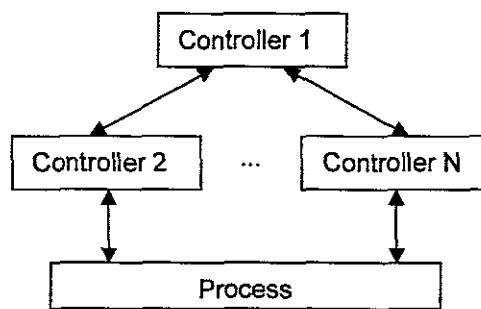
Figure 3.10: Two level control structure

In this structure the controls are distributed over different levels where some of the controllers only have an indirect access to the process to be controlled. Such controllers are inferior in the hierarchy and depend on information from controllers' superior to them. The system is called multi objective since the controllers have different objectives. Some controllers are used to act as coordinators so as to resolve conflicts between the different levels. The objectives of the controllers at the lower levels are independent of each other and are generally unknown to other units. The controllers have different functions from each other.

Controllers on the higher levels must obtain the same solution as controllers used on a global approach. The important objective of these controllers is to ensure that some necessary condition which is normally essentially satisfied by the global approach is also satisfied. The two important factors regarding multi level structures are task distribution and coordination. Tasks in the multi level structures are either distributed through horizontal or vertical divisions.

52

3.4.5.1 Horizontal division

A global problem is formulated as a set of separable lower order sub-problems. The original time horizon is coarsely partitioned into small time elements. On each subinterval of this partition, a sub-problem is defined. This method incorporates the use of a two levels hierarchical structure shown in Figure 3.11. At the lower level of the hierarchy, each sub-problem is solved by a local controller. The structure of the lower level problem is identical to the structure of the original problem. At the upper level the coordinating variables are introduced, in this level a global problem is defined depending only on these global variables. These variables are used for coordination of the actions of the lower level controllers.
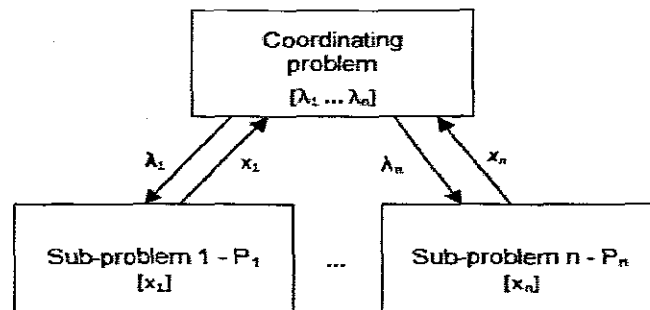
Figure 3.11: A hierarchical two level optimization structure

In principle the global problem P, whose objective is to extremise a global criterion is decomposed into a certain number of sub-problems P_i. The solutions of P_i are found locally such that solution of $[P_1 \ P_2 \ . \ . \ . \ P_n]$ implies the solution of the global problem P. Such a relation cannot in general be satisfied because of the existence of the interaction between the P_i's and the resulting conflicts. For this reason, it is necessary to introduce an intervention vector or coordination parameter λ and to replace P_i by $P_i(\lambda)$ such that the relation is satisfied. The solution $[P_1(\lambda) \ P_2(\lambda) \ . \ . \ . \ P_n(\lambda)]$ now implies the accurate solution of the global problem P. Each sub-problem $P_i(\lambda)$ is solved in the lower level and the coordination parameters λ are calculated in the upper level.

Hierarchical optimization involves choosing λ from some initial values λ_0 and iterating to the final value λ^* which solves the coordination problem in hierarchical control. Instead of solving the original problem directly, the upper level problem is solved.

For each set of values for the global or coordinating variables it is assumed that the local variables have been chosen optimally by the lower level optimization of each of the sub problems. The overall solution is then found by optimization at the upper level. Each iteration and update of the global variables involves an optimization of the sub-problems. Solution on the upper level shows feasibility of solution for the lower level.

3.4.5.2 Vertical division

In this division, the control task is divided vertically into basic control tasks. Figure 3.12 (Titli and Singh, 1978:21). The levels are regulation or direct control, optimization, adaptation and self organising. In the self organizing layer, the structure of the model and the control are chosen as the process environment changes. In the adaptation layer, the model and the control law directly adapt themselves as the process changes. The global optimal control problem is solved in the optimization level. The calculated trajectory of the optimal control variable is sampled and the resulting control variables are used as set points for the controller on the direct control layer.
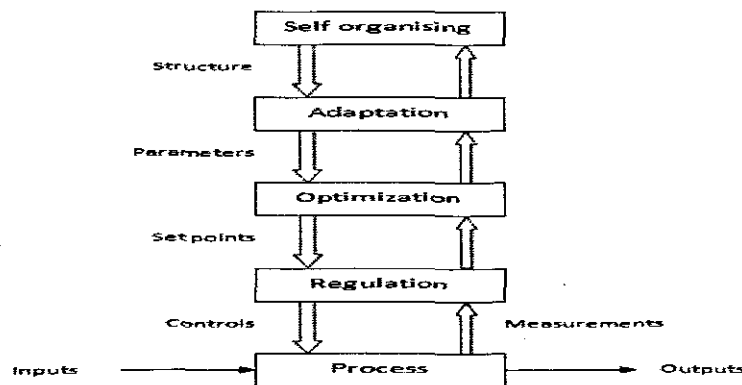
Figure 3.12: Hierarchical multilayer optimization structure

3.4.6 Numerical methods for solution of the optimal control problem

Because of the complexity of the system equations, the performance index and the constraints, it is useful to use numerical methods to solve the optimal control problem. This is generally a two point boundary value problem and its solution is very complex. In summary, the goal of the optimal control problem is to find the following:

- The state variables
- The coordinating or influence variables
- The control variables

54

That simultaneously satisfies the following conditions:

- The system equations
- The coordinating equations
- The optimality conditions
- The initial and final conditions

Numerical methods for the solution of this problem involve either dynamic programming or iterative procedures. In dynamic programming, many solutions satisfying the specified boundary conditions at one end are generated using unspecified boundary conditions as parameters (Bryson & Ho, 1975: 214). If the correct range of parameters is chosen then some of the solutions will pass through or near the desired boundary conditions at the other end.

In iterative procedures, successive linearization is used. A nominal solution is chosen that satisfies none, one, two or three of the four conditions above. This nominal solution is modified by successive linearization so that the remaining conditions are also satisfied. (Bryson & Ho, 1975: 213) The most used methods for iterative procedures are; Gradient methods, Neighbouring extremal methods and quasilinearization methods. The gradient method is used in this project.

3.4.6.1 Gradient methods

In optimal control theory, nonlinear programming is used to find the minimum of a function $f(x)$ or simply values of the variables that minimizes the function $f(x)$. There are many different ways to optimize nonlinear functions. The gradient search method also known as the steepest descent method is one of the direct search algorithms used. This is an iterative method often applied to systems that are too large to be handled by other direct methods. Such problems arise regularly when numerically solving partial differential equations. This method can also be used to solve unconstrained optimization problems such as energy minimization.

In this method, an arbitrary point $x(0)$ is chosen as the starting point in the search towards the minimum point x^* of the function $f(x)$. Series of steps $x(1)$, $x(2)$,....$x(i)$ are then taken until reaching the value close enough to the optimal solution x^*. When a step is taken, it is necessary to be in the direction in which the function $f(x)$ is decreasing and is doing so

most quickly. In basic calculus, for a given point $x(i)$ the gradient of the function, $f'(x(i))$, points in the direction of greatest increase of the function $f(x)$. The direction of greatest decrease of the function is the opposite of the direction of greatest increase, $-f'(x(i))$. The error or the difference between the iterated point and the current point,

$$e(i) = x(i) - x \tag{3.17}$$

indicates how far the point is from the minimum point.

The residual,

$$r(i) = -f'(x(i)) \tag{3.18}$$

is the direction of steepest descent for the function $f(x)$.

Therefore the next point in a search towards the minimum point can be found using the following equation:

$$x(i+1) = x(i) + \alpha r(i) \tag{3.19}$$

which is the same as

$$x(i+1) = x(i) - \alpha f'(x(i)) \tag{3.20}$$

When starting from an initial point $x(0)$, the first step towards the minimum point is therefore the point $x(1) = x(0) + \alpha r(0)$ which is similar to $x(1) = x(0) + \alpha(-f'(0))$. The minimum point is reached when the gradient is zero. It can be seen that α has a big influence on the speed of convergence. For this reason the choice of α is a vital one. As guidance the following equations are used when determining the value of α.

If $f(x(i)) - f(x(i-1)) < 0$ increase α \hfill (3.21)

If $f(x(i)) - f(x(i-1)) > 0$ decrease α and restart from $x(i-1)$ \hfill (3.22)

Figure 3.13 gives the flow chart of the gradient descent method.
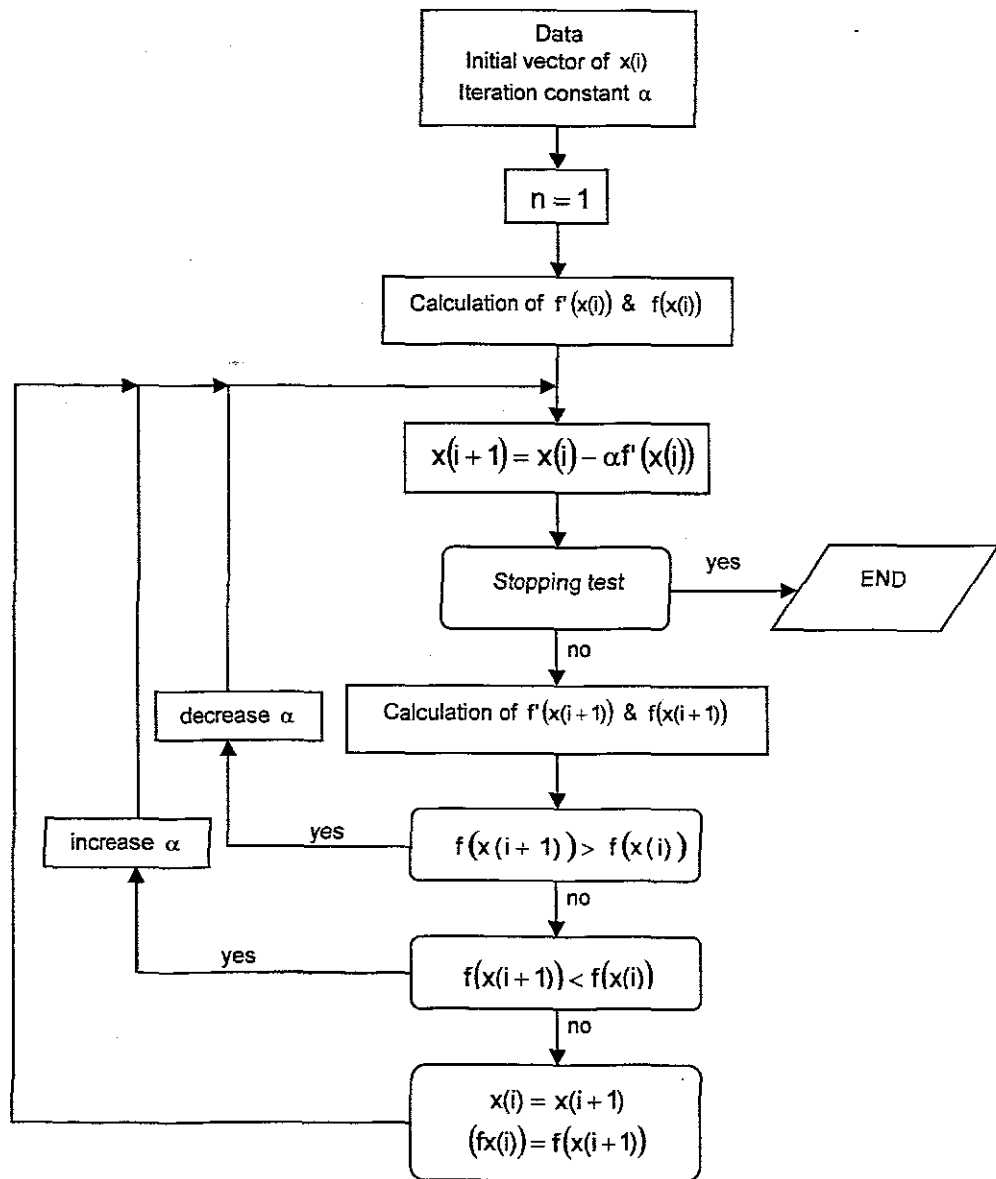
Figure 3.13: Flow chart of the gradient method

3.5   Conclusion

In this chapter, optimal control theory is studied. The theory provides a theoretical basis which is applied in this project to solve the optimization problem of a biological wastewater treatment process known as the activated sludge process. In the project the optimal control problem is to find the best dissolved oxygen trajectory that may be implemented on a controller to be tracked by the process in real time. In the next chapter the model of the Athlone and Benchmark activated sludge processes are derived. These models will then be used in Chapter 5 when formulating the optimal control problem.

57

CHAPTER FOUR
DERIVATIONS OF THE MODELS OF THE WASTEWATER TREATMENT PLANTS

4.1 Introduction

The goal of the research project is to calculate optimal dissolved oxygen trajectory for the activated sludge process (ASP) of a Wastewater Treatment Plant (WWTP). In order to explore this process and find the trajectory, it is necessary to determine a model of a WWTP and to use it for prediction of the plant behaviour. Chapter four presents derivations of the models that represent activities of the two WWTPs considered in the research project; these are the International Water Association (IWA) Benchmark Simulation plant and the Athlone WWTP in Cape Town, South Africa.

The plants models are derived based on the mass balance theory; which states that whatever mass is in the output and in the internal system combined must account for whatever mass entered the system (Prescot et al., 1997). To represent materials produced or consumed due to biological processes in the internal system of a WWTP, there are a number of Activated Sludge Models (ASMs) available in the literature. In this project the four ASMs developed by the IWA Task Group on mathematical modelling for design and operation of biological wastewater treatment are used (Henze et al., 2000).

This chapter is organized as follows; in section 4.2 the 4 IWA ASMs are described. The reduced versions of these models are described in section 4.3. In section 4.4, the mass balance models are derived for the Athlone and Benchmark WWTPs yielding sets of continuous differential equations that are used to describe the plants. These continuous mass balance equations are then transferred into their discrete counterparts so that they can be used to analyse and execute complex calculations. The attained discrete mass balance equations are then represented in state space form. State space form gives an explicit presentation of the model as a set of inputs, state variables and outputs related by differential equations. In state space representation the number of inputs, states variables and outputs are written in vector forms and the differential equations in matrix form.

4.2 Full activated sludge process models

In this section, the activated sludge models are presented; these are the Activated Sludge Model No. 1 (ASM1), Activated Sludge Model No. 2 (ASM2), Activated Sludge Model No 2d (ASM2d) and Activated Sludge Model No. 3 (ASM3).

Each model represents a Wastewater Treatment Plant (WWTP) designed to carry specific biological activities. The ASM1 describes the removal of carbon and nitrogen materials in wastewater. The ASM2 is similar to ASM1 but includes removal of phosphorus. The other two models; the ASM2d and the ASM3 are corrections to the defects realized in the ASM1 and ASM2 models over the years. A more detailed description of the models may be found in the IWA scientific and technical report no. 9 (Henze et al., 2000).

With these models the user may gain insight into the behaviour of the WWTPs without having to disturb the actual plants. The models capture different wastewater components and biological processes for treatment of these components. The wastewater components in the models are classified as either particulate (X) or soluble materials (S). Particulate materials flow through the system and finally settle in the clarifier where they are removed or fed back to the system via the bottom of the clarifier. Soluble materials change in concentrations due to the biological reactions in the system. Small part of these materials leaves the process as effluent wastewater through the top of the clarifier. In these models, biological processes are described by a set of non linear differential equations.

Activated sludge models are normally presented in a matrix format making it easy to relate a particular wastewater component to biological processes influencing it. Once a component is presented in terms of biological processes affecting it, the resulting set of equations may be used to predict the behaviour of that particular wastewater component. Wastewater components and biological processes are related by means of stoichiometric and kinetic parameters. Typical values of these parameters are given on the IWA scientific and technical report no. 9.

4.2.1 Activated sludge model no. 1

The overall ASM1 information summarized below was extracted from Henze et al. (1987 & 2000).

Basic concepts of the Activated Sludge Model No. 1 (ASM1) biological model are adapted from the prior model developed at the University of Cape Town (Dold et al, 1980). The ASM1 was completed and presented in 1987. ASM1 model represents the activated sludge process (ASP) designed for the purpose of reducing carbon and nitrogen in wastewater. This is achieved by means of three biological processes; carbon oxidation, nitrification and denitrification.

Carbon and nitrogen materials in wastewater are classified as Chemical Oxygen Demand (COD) and Total Kjeldahl (TKN) respectively. In the model, carbon and nitrogen are divided into the following forms. Carbon material (COD) is divided into; readily biodegradable substrate (S_S), slowly biodegradable substrate (X_S), soluble non-biodegradable inert material (S_I), particulate non-biodegradable inert material (X_I), heterotrophic biomass ($X_{B,H}$), autotrophic biomass ($X_{B,A}$) and inert particulate products arising from biomass decay (X_P). Nitrogenous material (TKN) is divided into; ammonia nitrogen (S_{NH}), soluble biodegradable organic nitrogen (S_{ND}), particulate biodegradable organic nitrogen (X_{ND}) and nitrate and nitrite nitrogen (S_{NO}).

As summarised in Jeppsson (1997: 3), the sum of all carbon materials amounts to the total COD and the sum of all nitrogen materials amounts to the total TKN.

$$COD_{total} = S_I + S_S + X_S + X_{B,H} + X_{B,A} + X_I + X_P \tag{4.1}$$

$$TKN_{total} = S_{NH} + S_{ND} + X_{ND} + S_{NO} \tag{4.2}$$

In addition to the carbon and nitrogen materials, the model captures the concentrations of dissolved oxygen (S_O) and alkalinity (S_{ALK}). Since carbon reduction depends on the concentration of substrates and oxygen, dissolved oxygen in the model reflects negative COD while alkalinity enables for the monitoring of nitrite nitrogen and pH in wastewater (Jeppsson, 1997).

As mentioned above, the reduction of carbon and nitrogen contents in wastewater is achieved through carbon oxidation, nitrification and denitrification processes. Carbon oxidation occurs when in the presence of oxygen, active heterotrophic biomass ($X_{B,H}$) uses readily biodegradable substrate (S_S) for energy, synthesis and growth. The result of this process is the production of more heterotrophic biomass and the decrease in COD concentrations hence the process is called aerobic growth of heterotrophs.

Denitrification occurs when in the absence of oxygen, nitrate and nitrite nitrogen (S_{NO}) gets used as electron acceptor by the active heterotrophic biomass ($X_{B,H}$). The result is the production of more heterotrophic biomass as well as nitrogen gas; this is called anoxic growth of heterotrophs. Nitrification occurs when in the presence of oxygen, some portion of ammonia nitrogen (S_{NH}) is used for the growth of active autotrophic biomass ($X_{B,A}$) and some is converted into nitrate and nitrite nitrogen (S_{NO}). The result of this process is the production of more autotrophic biomass and the regeneration of nitrite nitrogen. This process also affects the alkalinity of the wastewater.

The model also captures the processes describing the decay of active heterotrophic biomass ($X_{B,H}$) and active autotrophic biomass ($X_{B,A}$) due to lysis. The resulting non biodegradable material is classified as inert particulate products arising from biomass decay (X_P) and the resulting biodegradable material is added into slowly biodegradable substrates (X_S).

The contents of nitrate and nitrite nitrogen (S_{NO}) are reduced through ammonification. In this process nitrate and nitrite nitrogen is converted into ammonia nitrogen. Through hydrolysis, slowly biodegradable substrate (X_S) and particulate biodegradable organic nitrogen (X_{ND}) are broken down and allowed to become part of the readily biodegradable substrate (S_S) and soluble biodegradable organic nitrogen (S_{ND}) respectively.

The soluble non-biodegradable inert material (S_I) and particulate non-biodegradable inert material (X_I) are unaffected by the biological reactions. The soluble inert material leaves the system with the effluent wastewater while the particulate material leaves the system with the sludge. The ASM1 describes thirteen wastewater components described by the vector $x \in R^{13}$ and eight biological processes described by the vector $P \in R^8$.

The components and processes of the ASM1 model may be presented in vector forms as:

$$x = \begin{bmatrix} S_I & S_S & X_I & X_S & X_{BH} & X_{BA} & X_P & S_O & S_{NO} & S_{NH} & S_{ND} & X_{ND} & S_{ALK} \end{bmatrix}^T$$

$$P = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \end{bmatrix}^T$$

where:

$$P_1 = \hat{\mu}_H \left(\frac{S_S}{K_S + S_S} \right) \left(\frac{S_O}{K_{OH} + S_O} \right) X_{BH} \tag{4.3}$$

$$P_2 = \hat{\mu}_H \left(\frac{S_S}{K_S + S_S} \right) \left(\frac{K_{O,H}}{K_{O,H} + S_O} \right) \left(\frac{S_{NO}}{K_{NO} + S_{NO}} \right) \eta_g X_{B,H} \tag{4.4}$$

$$P_3 = \hat{\mu}_A \left(\frac{S_{NH}}{K_{NH} + S_{NH}} \right) \left(\frac{S_O}{K_{O,A} + S_O} \right) X_{B,A} \tag{4.5}$$

$$P_4 = b_H X_{B,H} \tag{4.6}$$

$$P_5 = b_A X_{B,A} \tag{4.7}$$

$$P_6 = k_a S_{ND} X_{B,H} \tag{4.8}$$

$$P_7 = k_h \left(\frac{X_S / X_{B,H}}{K_X + X_S / X_{B,H}} \right) \left[\left(\frac{S_O}{K_{O,H} + S_O} \right) + \eta_h \left(\frac{K_{O,H}}{K_{O,H} + S_O} \right) \left(\frac{S_{NO}}{K_{NO} + S_{NO}} \right) \right] X_{B,H} \tag{4.9}$$

$$P_8 = k_h \cdot \frac{X_{ND}}{X_S} \cdot \left(\frac{X_S / X_{B,H}}{K_X + X_S / X_{B,H}} \right) \left[\left(\frac{S_O}{K_{O,H} + S_O} \right) + \eta_h \left(\frac{K_{O,H}}{K_{O,H} + S_O} \right) \left(\frac{S_{NO}}{K_{NO} + S_{NO}} \right) \right] X_{B,H} \tag{4.10}$$

A Peterson matrix representation of the ASM1 is given in Figure 4.1 while descriptions and values and values of the ASM1 parameters are given in Table 4.1. The parameters values of the ASM1 model are adapted from Henze et al. (2000).

Figure 4.1: Activated Sludge Model No. 1 in a matrix format

Component → i	1 S_I	2 S_S	3 X_I	4 X_S	5 $X_{B,H}$	6 $X_{B,A}$	7 X_P	8 S_O	9 S_{NO}	10 S_{NH}	11 S_{ND}	12 X_{ND}	13 S_{ALK}	Process Rate, ρ_j [ML^{-3}T^{-1}]
↓ j Process														
1 Aerobic growth of heterotrophs		$-\dfrac{1}{Y_H}$			1			$-\dfrac{1-Y_H}{Y_H}$		$-i_{XB}$			$-\dfrac{i_{XB}}{14}$	$\mu_H\left(\dfrac{S_S}{K_S+S_S}\right)\left(\dfrac{S_O}{K_{O,H}+S_O}\right)X_{B,H}$
2 Anoxic growth of heterotrophs		$-\dfrac{1}{Y_H}$			1				$-\dfrac{1-Y_H}{2.86\,Y_H}$	$-i_{XB}$			$\dfrac{1-Y_H}{14\cdot 2.86\,Y_H}-\dfrac{i_{XB}}{14}$	$\mu_H\left(\dfrac{S_S}{K_S+S_S}\right)\left(\dfrac{K_{O,H}}{K_{O,H}+S_O}\right)\left(\dfrac{S_{NO}}{K_{NO}+S_{NO}}\right)\eta_g X_{B,H}$
3 Aerobic growth of autotrophs						1		$-\dfrac{4.57-Y_A}{Y_A}$	$\dfrac{1}{Y_A}$	$-i_{XB}-\dfrac{1}{Y_A}$			$-\dfrac{i_{XB}}{14}-\dfrac{1}{7\,Y_A}$	$\mu_A\left(\dfrac{S_{NH}}{K_{NH}+S_{NH}}\right)\left(\dfrac{S_O}{K_{O,A}+S_O}\right)X_{B,A}$
4 Decay of heterotrophs				$1-f_P$	-1		f_P					$i_{XB}-f_P i_{XP}$		$b_H X_{B,H}$
5 Decay of autotrophs				$1-f_P$		-1	f_P					$i_{XB}-f_P i_{XP}$		$b_A X_{B,A}$
6 Ammonification of soluble organic nitrogen										1	-1		$\dfrac{1}{14}$	$k_a S_{ND} X_{B,H}$
7 'Hydrolysis' of entrapped organics		1		-1										$k_h\dfrac{X_S/X_{B,H}}{K_X+(X_S/X_{B,H})}\left[\left(\dfrac{S_O}{K_{O,H}+S_O}\right)+\eta_h\left(\dfrac{K_{O,H}}{K_{O,H}+S_O}\right)\left(\dfrac{S_{NO}}{K_{NO}+S_{NO}}\right)\right]X_{B,H}$
8 'Hydrolysis' of entrapped organic nitrogen											1	-1		$\rho_7(X_{ND}/X_S)$
Observed Conversion Rates [ML^{-3}T^{-1}]							$r_i=\sum_j v_{ij}\rho_j$				$r_i=\sum_j v_{ij}\rho_j$			
Stoichiometric Parameters:	Soluble inert organic matter [M(COD)L^{-3}]	Readily biodegradable substrate [M(COD)L^{-3}]	Particulate inert organic matter [M(COD)L^{-3}]	Slowly biodegradable substrate [M(COD)L^{-3}]	Active heterotrophic biomass [M(COD)L^{-3}]	Active autotrophic biomass [M(COD)L^{-3}]	Particulate products arising from biomass decay [M(COD)L^{-3}]	Oxygen (negative COD) [M(-COD)L^{-3}]	Nitrate and nitrite nitrogen [M(N)L^{-3}]	NH$_4$+NH$_3$ nitrogen [M(N)L^{-3}]	Soluble biodegradable organic nitrogen [M(N)L^{-3}]	Particulate biodegradable organic nitrogen [M(N)L^{-3}]	Alkalinity – Molar units	Kinetic Parameters:

Stoichiometric Parameters:
Heterotrophic yield Y_H
Autotrophic yield Y_A
Fraction of biomass yielding particulate products, f_P
Mass N/Mass COD in biomass: i_{XB}
Mass N/Mass COD in products from biomass i_{XP}

Kinetic Parameters:
Heterotrophic growth and decay: $\mu_H, K_S, K_{O,H}, K_{NO}, b_H$
Autotrophic growth and decay: $\mu_A, K_{NH}, K_{O,A}, b_A$
Correction factor for anoxic growth of heterotrophs η_g
Ammonification k_a
Hydrolysis k_h, K_X
Correction factor for anoxic hydrolysis η_h

Figure 4.1: Activated Sludge Model No. 1 in a matrix format

(Adapted from Jeppsson, 1997: 15)

63

Table 4.1: ASM1 model parameter values at 15 °C

Symbol	Description	Value	Unit
Y_H	Heterotrophic yield	0.67	g COD
Y_A	Autotrophic yield	0.24	g COD
f_P	Fraction of biomass yielding particulate products	0.08	-
i_{XB}	COD mass in biomass	0.086	g N
i_{XP}	COD mass in products from biomass	0.06	g N
μ_H	Heterotrophic maximum specific growth rate	3	day^{-1}
b_H	Heterotrophic decay rate	0.2	day^{-1}
μ_A	Autotrophic maximum specific growth rate	0.3	day^{-1}
b_A	Correction factor for anoxic growth of heterotrophs	0.1	day^{-1}
η_g	Correction factor for anoxic growth of heterotrophs	0.8	-
k_a	Ammonification rate	0.04	m^{3}
k_h	Maximum specific hydrolysis rate	1	g COD
η_h	Correction factor for anoxic hydrolysis	0.4	-
K_S	Half saturation coefficient (hsc) for heterotrophs	20	$g\ COD\ m^{-3}$
$K_{O,H}$	Oxygen hsc for heterotrophs	0.2	$g\ O_2\ m^{-3}$
K_{NO}	Nitrate hsc for denitrifying heterotrophs	0.5	$g\ NO_3\text{-}N\ m^{-3}$
$K_{O,A}$	Oxygen hsc for autotrophs	0.4	$g\ O_2\ m^{-3}$
K_{NH}	Ammonia hsc for autotrophs	1	$g\ NH_3\text{-}N\ m^{-3}$
K_X	Hsc for hydrolysis of slowly biodegradable substrate	0.01	g COD

4.2.2 Activated sludge model no. 2

The overall ASM2 information below was extracted from Henze et al. (1995 & 2000).

Similar to the Activated Sludge Model No. 1 (ASM1), the Activated Sludge Model No. 2 (ASM2) describes the removal of carbon and nitrogen in wastewater. The most significant difference between ASM1 and ASM2 is that the latter incorporates phosphorus removal. ASM2 describes both the biological and chemical (chemical precipitation) methods for the removal of phosphorus. Biological phosphorus removal is achieved when the mixed liquor is exposed to an anaerobic/aerobic sequence.

Under anaerobic conditions microorganisms that accrue higher levels of intracellular phosphorus assimilate and store volatile fatty acids (VFAs) and other fermentation products, (Erdal et al, 2001). Such microorganisms are often referred to as phosphorus-accumulating-organisms (PAOs). During this period phosphorus is released in the anaerobic zone to produce the energy needed to take up the fermentation products.

In the ASM2, carbon material (COD) is divided into; fermentable, readily biodegradable organic substrate (S_F), fermentation products (S_A), soluble inert non-biodegradable organic material (S_I), particulate inert non-biodegradable organic material (X_I), slowly biodegradable substrate (X_S), heterotrophic organisms (X_H), nitrifying organisms (X_{AUT}) and total suspended solids (X_{TSS}). Nitrogenous material (TKN) is divided into; ammonium plus ammonia nitrogen (S_{NH4}), nitrate & nitrite nitrogen (S_{NO3}) and dinitrogen (S_{N2}).

Phosphorus material is divided into: phosphate (S_{PO4}), phosphate-accumulating organisms (X_{PAO}), poly phosphate (X_{PP}), a cell internal storage product of phosphorus-accumulating organisms (X_{PHA}), metal-hydroxides (X_{MeOH}) and metal phosphates (X_{MeP}). The model also captures the concentrations of dissolved oxygen (S_{O2}) and alkalinity (S_{ALK}). ASM2 describes nineteen wastewater compounds described by the vector $x \in R^{19}$ and nineteen biological processes described by the vector $P \in R^{19}$. The components and processes of the ASM2 model may be presented in vector forms as:

$$x = \begin{bmatrix} S_{O2} & S_F & S_A & S_{NH4} & S_{NO3} & S_{PO4} & S_I & S_{ALK} & S_{N2} & X_I & X_S & X_H & X_{PAO} & X_{PP} & X_{PHA} & X_{AUT} & X_{TSS} & X_{MeOH} & X_{MeP} \end{bmatrix}^T$$

$$P = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 & P_9 & P_{10} & P_{11} & P_{12} & P_{13} & P_{14} & P_{15} & P_{16} & P_{17} & P_{18} & P_{19} \end{bmatrix}^T$$

where:

$$P_1 = K_h \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{X_S/X_H}{K_X + X_S/X_H} \cdot X_H \tag{4.11}$$

$$P_2 = K_h \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{X_S/X_H}{K_X + X_S/X_H} \cdot X_H \tag{4.12}$$

$$P_3 = K_h \cdot \eta_{fe} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{X_S/X_H}{K_X + X_S/X_H} \cdot X_H \tag{4.13}$$

$$P_4 = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{K_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.14}$$

$$P_5 = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.15}$$

$$P_6 = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.16}$$

$$P_7 = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot X_H \tag{4.17}$$

$$P_8 = q_{fe} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_F}{K_{fe} + S_F} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.18}$$

$$P_9 = b_H \cdot X_H \tag{4.19}$$

$$P_{10} = q_{PHA} \frac{S_A}{K_A + S_A} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{PP}/X_{PAO}}{K_{PP} + X_{PP}/X_{PAO}} \cdot X_{PAO} \tag{4.20}$$

$$P_{11} = q_{PP} \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{PO4}}{K_{PS} + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA} + X_{PHA}/X_{PAO}} \cdot \frac{K_{MAX} - X_{PP}/X_{PAO}}{K_{IPP} + K_{MAX} - X_{PP}/X_{PAO}} \cdot X_{PAO} \tag{4.21}$$

$$P_{12} = \mu_{PAO} \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA} + X_{PHA}/X_{PAO}} \cdot X_{PAO} \tag{4.22}$$

$$P_{13} = b_{PAO} \cdot X_{PAO} \cdot S_{ALK}/(K_{ALK} + S_{ALK}) \tag{4.23}$$

$$P_{14} = b_{PP} \cdot X_{PP} \cdot S_{ALK}/(K_{ALK} + S_{ALK}) \tag{4.24}$$

$$P_{15} = b_{PHA} \cdot X_{PHA} \cdot S_{ALK}/(K_{ALK} + S_{ALK}) \tag{4.25}$$

$$P_{16} = \mu_{AUT} \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_{AUT} \tag{4.26}$$

$$P_{17} = b_{AUT} \cdot X_{AUT} \tag{4.27}$$

$$P_{18} = k_{PRE} \cdot S_{PO4} \cdot X_{MeOH} \tag{4.28}$$

$$P_{19} = k_{RED} \cdot X_{MeP} \cdot S_{ALK}/(K_{ALK} + S_{ALK}) \tag{4.29}$$

A Peterson matrix representation of the ASM2 is given in Figure 4.2. Descriptions and values of the ASM2 parameters in Figure 4.2 are given in Table 4.2; the ASM2 parameter values are adapted from Henze et al. (1995).

Table 4.2: ASM2 model parameter values at 10 $^{\circ}$C

Symbol	Name	Value	Unit
K_h	Hydrolysis rate constant	2	d^{-1}
K_{O2}	Saturation/inhibition coefficient for S_{O2}	0.20	$g\ O_2\ m^{-3}$
K_X	Saturation coefficient for particulate COD	0.1	$g\ X_S\ (g\ X_H)^{-1}$
n_{NO3}	Reduction factor for denitrification	0.60	-
K_{NO3}	Saturation coefficient for particulate nitrate	0.5	$g\ N\ m^{-3}$
n_{fe}	Anaerobic hydrolysis reduction factor	0.40	-
μ_H	Maximum growth rate on substrate	3	$g\ X_S\ (g\ X_H)^{-1}\ d^{-1}$
K_F	Saturation (Sat.) coefficient for growth on S_F	4	$g\ COD\ m^{-3}$
K_{NH4}	Saturation coefficient for ammonium	0.05	$g\ N\ m^{-3}$
K_P	Sat. coefficient for phosphorus in growth	0.01	$g\ P\ m^{-3}$
K_{ALK}	Saturation coefficient for alkalinity	0.1	$mole\ HCO_3\ m^{-3}$
K_A	Saturation coefficient for S_A	4	$g\ COD\ m^{-3}$
q_{fe}	Maximum rate for fermentation	1.5	$g\ S_F\ (g\ X_H)^{-1}\ d^{-1}$
K_{fe}	Saturation coefficient for fermentation of S_F	4	$g\ COD\ m^{-3}$
q_{PHA}	Rate constant for storage of X_{PHA}	2	$g\ X_{PHA}\ (g\ X_{PAO})^{-1}\ d^{-1}$
K_{PP}	Saturation coefficient for poly-phosphate	0.01	$g\ X_{PP}\ (g\ X_{PAO})^{-1}$
q_{PP}	Rate constant for storage of X_{PP}	1	$g\ X_{PP}\ (g\ X_{PAO})^{-1}\ d^{-1}$
K_{PS}	Sat. coefficient for phosphorus in storage	0.2	$g\ P\ m^{-3}$
K_{PHA}	Saturation coefficient for PHA	0.01	$g\ X_{PHA}\ (g\ X_{PAO})^{-1}$
K_{MAX}	Maximum ration or X_{PP}/X_{PAO}	0.34	$g\ X_{PP}\ (g\ X_{PAO})^{-1}$
K_{iPP}	Inhibition coefficient for PP storage	0.02	$g\ X_{PP}\ (g\ X_{PAO})^{-1}$
μ_{PAO}	Maximum growth rate for X_{PAO}	0.67	d^{-1}
b_{PAO}	Rate constant for lysis of X_{PAO}	0.1	d^{-1}
b_{PP}	Rate constant for lysis of X_{PP}	0.1	d^{-1}
b_{PHA}	Rate constant for lysis of X_{PHA}	0.1	d^{-1}
μ_{AUT}	Maximum growth rate of X_{AUT}	0.35	d^{-1}
k_{PRE}	Rate constant for precipitation	1	$m^3\ (g\ Fe\ (OH)_3)^{-1}\ d^{-1}$
k_{RED}	Rate constant for re-dissolution	0.6	d^{-1}
f_{SI}	Production of S_I in hydrolysis	0	$g\ COD\ (g\ COD)^{-1}$
Y_{PAO}	Yield coefficient for biomass	0.625	$g\ COD\ (g\ COD)^{-1}$
Y_{PO4}	PP requirement for biomass storage	0.40	$g\ P\ (g\ COD)^{-1}$
Y_{PHA}	Biomass requirement for PP storage	0.2	$g\ COD\ (g\ P)^{-1}$
Y_A	Yield coefficient of autotrophic biomass		$g\ COD\ (g\ N)^{-1}$
Y_H	Yield coefficient of autotrophic biomass		$g\ COD\ (g\ COD)^{-1}$
f_{XI}	Fraction of inert COD generated in biomass lysis		$g\ COD\ (g\ COD)^{-1}$
i_{PMB}	P content of biomass, XH, XPAO, XAUT	0.02	$g\ P\ (g\ COD)^{-1}$
i_{NBM}	N content of biomass, XH, XPAO, XAUT	0.07	$g\ N\ (g\ COD)^{-1}$
b_H	Rate constant for lysis and decay		d^{-1}
b_{AUT}	Decay rate of XAUT		d^{-1}

Processes \ States	SO2	SF	SA	SNH4	SNO3	SPO4	SI	SALK	SN2	XI	XS	XH	XPAO	XPP	XPHA	XAUT	XTSS	XMeOH	XMeP	Process Rates
Aerobic hydrolysis		1		0.01				0.001		−1							−0.75			P1
Anoxic hydrolysis		1		0.01				0.001		−1							−0.75			
Anaerobic hydrolysis		1		0.01				0.001		−1							−0.75			P3
Aerobic growth on SF	$1-\dfrac{1}{YH}$	$-\dfrac{1}{YH}$										1								P4
Aerobic growth on SA	$1-\dfrac{1}{YH}$		$-\dfrac{1}{YH}$									1								P5
Anoxic growth on SF		$-\dfrac{1}{YH}$			$-\dfrac{1-YH}{2.86YH}$				$\dfrac{1-YH}{2.86YH}$			1								P6
Anoxic growth on SA			$-\dfrac{1}{YH}$		$-\dfrac{1-YH}{2.86YH}$				$\dfrac{1-YH}{2.86YH}$			1								P7
Fermentation		−1	1																	
Lysis of XH										fXI	1 − fXI	−1								P9
Storage of XPHA			−1			−YPO4								−YP04	1					P10
Storage of XPP	−YPHA					−1								1	−YPHA					P11
Aerobic growth of XPHA	$1-\dfrac{1}{YH}$					−iPBM							1		$-\dfrac{1}{YH}$					P12
Lysis of XPAO			1			0.031				fXI	1 − fXI		−1							P13
Lysis of XPP						1								−1						P14
Lysis of XPHA															−1−1					P15
Growth of XAUT	$\dfrac{4.57-YA}{YA}$			$iNBM \; \dfrac{1}{YA}$	$\dfrac{1}{YA}$	−iPBM										1				P16
Lysis of XAUT				0.031		0.01				fXI	1 − fXI					−1				P17
Precipitation						−1		0.048									1.42	3.45	4.87	P18
Redissolution						1		−0.048									−1.42	3.45	−4.87	P19

Figure 4.2: Activated Sludge Model No. 2 in a matrix format

67

4.2.3 Activated sludge model no. 2d

The overall ASM2d information summarized below was extracted from Henze et al. (1999 & 2000).

Activated Sludge Model No. 2d (ASM2d) is an extension of Activated Sludge Model No.2 (ASM2) includes two denitrification processes relating to the biological removal of phosphorus. In the ASM2d, the denitrification activity of poly-phosphates (PAOs) is considered to inhibit nitrites in wastewater (Yoshida et al., 2009). ASM2d presents the same wastewater compounds and biological processes as the ASM2, with the addition of two biological processes which are anoxic growth of X_{PAO} and anoxic storage of X_{PP}.

ASM2d describes nineteen wastewater components described by the vector $x \in R^{19}$ and twenty-one biological processes described by the vector $P \in R^{21}$.

The components and processes of the ASM2d model may be presented in a vector form as:

$$x = \begin{bmatrix} S_{O2} & S_F & S_A & S_{NH4} & S_{NO3} & S_{PO4} & S_I & S_{ALK} & S_{N2} & X_I & X_S & X_H & X_{PAO} & X_{PP} & X_{PHA} & X_{AUT} & X_{TSS} & X_{MeOH} & X_{MeP} \end{bmatrix}^T$$

$$P = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 & P_9 & P_{10} & P_{11} & P_{12} & P_{13} & P_{14} & P_{15} & P_{16} & P_{17} & P_{18} & P_{19} & P_{20} & P_{21} \end{bmatrix}$$

where:

$$P_1 = K_h \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{X_S / X_H}{K_X + X_S / X_H} \cdot X_H \tag{4.30}$$

$$P_2 = K_h \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{X_S / X_H}{K_X + X_S / X_H} \cdot X_H \tag{4.31}$$

$$P_3 = K_h \cdot \eta_{fe} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{X_S / X_H}{K_X + X_S / X_H} \cdot X_H \tag{4.32}$$

$$P_4 = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{K_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.33}$$

$$P_5 = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.34}$$

$$P_6 = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.35}$$

$$P_7 = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot X_H \tag{4.36}$$

$$P_8 = q_{fe} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_F}{K_{fe} + S_F} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.37}$$

$$P_9 = b_H \cdot X_H \tag{4.38}$$

68

States / Processes	SO2	SF	SA	SNH4	SNO3	SPO4	SI	SALK	SN2	XI	XS	XH	XPAO	XPP	XPHA	XAUT	XTSS	XMeOH	XMeP	Process rates
Aerobic hydrolysis		1		0.01				0.001			-1						-0.75			P1
Anoxic hydrolysis		1		0.01				0.001			-1						-0.75			P2
Anaerobic hydrolysis		1		0.01				0.001			-1						-0.75			P3
Aerobic growth on SF	$1-\dfrac{1}{YH}$	$-\dfrac{1}{YH}$										1								P4
Aerobic growth on SA	$1-\dfrac{1}{YH}$		$-\dfrac{1}{YH}$									1								P5
Anoxic growth on SF		$-\dfrac{1}{YH}$			$\dfrac{1-YH}{2.86YH}$				$\dfrac{1-YH}{2.86YH}$			1								P6
Anoxic growth on SA			$-\dfrac{1}{YH}$		$\dfrac{1-YH}{2.86YH}$				$\dfrac{1-YH}{2.86YH}$			1								P7
Fermentation		-1	1																	P8
Lysis of XH										fXI	$1-fXI$	-1								P9
Storage of XPHA			-1			YPO4								$-YPO4$	1					P10
Storage of XPP	$-YPHA$					-1								1	$-YPHA$					P11
Anoxic storage of XPP					-0.07	-1			0.07					1	$-YPHA$					P12
Aerobic growth of XPHA	$1-\dfrac{1}{YH}$					$-iPBM$							1		$-\dfrac{1}{YH}$					P13
Anoxic growth on XPP					-0.21	$-iPBM$			0.21				1		$-\dfrac{1}{YH}$					P14
Lysis of XPAO						0.01				fXI	$1-fXI$		-1							P15
Lysis of XPP						1								-1						P16
Lysis of XPHA			1												-1					P17
Growth of XAUT	$\dfrac{4.57-YA}{YA}$			-4.24	$\dfrac{1}{YA}$	$-iPBM$										1				P18
Lysis of XAUT				0.031		0.01				fXI	$1-fXI$					-1				P19
Precipitation						-1		0.048									1.42	-3.45	4.87	P20
Re-dissolution						1		-0.048									-1.42	3.45	-4.87	P21

Figure 4.3: Activated Sludge Model No. 2d in a matrix format

70

4.2.4 Activated sludge model no. 3

The overall ASM1 information summarized below was extracted from Gujer et al. (1999 & 2000). The Activated Sludge Model No. 3 was developed to correct defects recognized by the COST group while experimenting with the Activated Sludge Model No. 1 (ASM1). Some of the shortcomings were due to the fact that in the past the models were required to be simple due to computational limitations. As a result of this, various biological processes in the ASM1 were replaced by a single process. In the ASM3, previously combined conversion processes in ASM1 are now separated to give a more realistic representation of the activated sludge process. Also captured in the ASM3, is the representation of internal storage compounds, which have an important role in the metabolism of the organisms (Henze et al, 2000). Similar to the ASM1, the ASM3 describes oxygen consumption, sludge production, nitrification and denitrification processes used for the removal of carbon and nitrogen in wastewater.

In the ASM3 carbon material (COD) is divided into; readily biodegradable substrate (S_S), soluble non-biodegradable material (S_I), particulate non-biodegradable material (X_I), slowly biodegradable substrate (X_S), heterotrophic biomass (X_H), a cell internal storage product heterotrophic organisms (X_{STO}), nitrifying organisms (X_A) and suspended solids (X_{SS}). Nitrogenous material (TKN) is divided into; ammonia nitrogen (S_{NH4}), dinitrogen (S_{N2}) and nitrate & nitrite nitrogen (S_{NOX}). ASM3 also comprises the concentrations of dissolved oxygen (S_{O2}) and alkalinity (S_{ALK}). ASM3 describes thirteen wastewater components and twelve biological processes described by the vectors $x \in R^{13}$ and $P \in R^{12}$.

The components and processes of the ASM3 model may be presented in a vector form as: $x = [S_{O2} \quad S_I \quad S_S \quad S_{NH4} \quad S_{N2} \quad S_{NOX} \quad S_{ALK} \quad X_I \quad X_S \quad X_H \quad X_{STO} \quad X_A \quad X_{SS}]^T$

$P = [P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5 \quad P_6 \quad P_7 \quad P_8 \quad P_9 \quad P_{10} \quad P_{11} \quad P_{12}]^T$

where:

$$P_1 = k_H \cdot \frac{X_S/X_H}{K_X + X_S/X_H} \cdot X_H \tag{4.51}$$

$$P_2 = k_{STO} \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_S}{K_S + S_S} \cdot X_H \tag{4.52}$$

$$P_3 = k_{STO} \cdot \eta_{NOX} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NOX}}{K_{NOX} + S_{NOX}} \cdot \frac{S_S}{K_S + S_S} \cdot X_H \tag{4.53}$$

$$P_4 = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{STO}/X_H}{K_{STO} + X_{STO}/X_H} \cdot X_H \tag{4.54}$$

$$P_5 = \mu_H \cdot \eta_{NOX} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NOX}}{K_{NOX} + S_{NOX}} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{STO}/X_H}{K_{STO} + X_{STO}/X_H} \cdot X_H \tag{4.55}$$

71

$$P_6 = b_{H,O2} \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot X_H \tag{4.56}$$

$$P_7 = b_{H,NOX} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NOX}}{K_{NOX} + S_{NOX}} \cdot X_H \tag{4.57}$$

$$P_8 = b_{STO,O2} \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot X_{STO} \tag{4.58}$$

$$P_9 = b_{STO,NOX} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NOX}}{K_{NOX} + S_{NOX}} \cdot X_{STO} \tag{4.59}$$

$$P_{10} = \mu_A \cdot \frac{S_{O2}}{K_{A,O2} + S_{O2}} \cdot \frac{S_{NH4}}{K_{A,NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{A,ALK} + S_{ALK}} \cdot X_A \tag{4.60}$$

$$P_{11} = b_{A,O2} \cdot \frac{S_{O2}}{K_{A,O2} + S_{O2}} \cdot X_A \tag{4.61}$$

$$P_{12} = b_{A,NOX} \cdot \frac{K_{A,O2}}{K_{A,O2} + S_{O2}} \cdot \frac{S_{NOX}}{K_{A,NOX} + S_{NOX}} \cdot X_A \tag{4.62}$$

A Peterson matrix representation of the ASM3 is given in Figure 4.4. Descriptions and values of the ASM3 parameters are given in Table 4.3; the ASM3 parameters values are also adapted from Gujer et al. (1999).

Table 4.3: ASM3 model parameter values at 10 ^{0}C

Symbol	Name	Value	Unit
k_H	Hydrolysis rate constant	2	g COD_{XS} (g COD_{XH})$^{-1}$ d^{-1}
k_{STO}	Storage rate constant	2.5	g COD_{SS} (g COD_{XH})$^{-1}$ d^{-1}
$b_{H,NOX}$	Anoxic endogenous respiration rate of X_H	0.05	d^{-1}
$b_{A,NOX}$	Anoxic endogenous respiration rate of X_A	0.02	d^{-1}
μ_H	Heterotrophic maximum growth rate	1	d^{-1}
μ_A	Autotrophic maximum growth rate	0.35	d^{-1}
η_{NOX}	Anoxic reduction factor	0.6	-
$b_{H,O2}$	Aerobic endogenous respiration rate of X_H	0.1	d^{-1}
$b_{STO,O2}$	Aerobic respiration rate of X_{STO}	0.1	d^{-1}
$b_{A,O2}$	Aerobic endogenous respiration rate of X_A	0.05	d^{-1}
$b_{STO,NOX}$	Anoxic respiration rate of X_{STO}	0.05	d^{-1}
$K_{A,O2}$	Oxygen saturation for nitrifiers	0.5	g O_2 m^{-3}
$K_{A,NH4}$	Ammonium substrate saturation for X_A	1	g N m^{-3}
K_{NOX}	Saturation constant for S_{NOX}	0.5	g NO_3-N m^{-3}
K_X	Hydrolysis saturation constant	1	g COD_{XS} (g COD_{XH})$^{-1}$ d^{-1}
K_{O2}	Saturation constant for S_{NO2}	0.2	g O_2 m^{-3}
K_{NH4}	Saturation constant for alkalinity for X_H	0.001	g N m^{-3}
K_{ALK}	Saturation constant for X_H	0.1	mole HCO_3 m^{-3}
f_{SI}	Production of S_I in hydrolysis	0	g COD_{SI} (g COD_{XS})$^{-1}$
$Y_{STO,O2}$	Aerobic yield of stored product per S_S	0.85	g COD_{XSTO} (g COD_{SS})$^{-1}$
$Y_{STO,NOX}$	Anoxic yield of stored product per S_S	0.80	g COD_{XSTO} (g COD_{SS})$^{-1}$
$Y_{H,O2}$	Aerobic yield of heterotrophic biomass	0.63	g COD_{XH} (g COD_{XSTO})$^{-1}$
$Y_{H,NOX}$	Anoxic yield of heterotrophic biomass	0.54	g COD_{XH} (g COD_{XSTO})$^{-1}$
Y_A	Yield of autotrophic biomass per NO3-N	0.24	g COD_{XA} (g N_{SNOX})$^{-1}$
f_{XI}	Production of X_I in endogenous respiration	0.20	g COD_{XI} (g COD_{XBM})$^{-1}$
K_S	Saturation constant for substrate S_S	2	g COD_{SS} m^{-3}
$K_{A,ALK}$	Bicarbonate saturation for nitrifiers	0.5	mole HCO_3 m^{-3}
$i_{N,BM}$	N content of biomass, X_H, X_A	0.07	g N (g COD_{XBM})$^{-1}$
$b_{H,NOX}$	Anoxic endogenous respiration rate of X_H	0.05	d^{-1}

States / Processes	SO2	SI	SS	SNH4	SN2	SNOX	SALK	XI	XS	XH	XSTO	XA	XSS	Process rates (P₁ - P₁₂)
Hydrolysis		fSI	1	0.01			0.001		-1				iXS	$k_H \cdot \dfrac{X_S/X_H}{K_X + X_S/X_H} \cdot X_H$
Aerobic storage of SS	-0.15		-1	0.03			0.002				YH02		0.51	$k_{STO} \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_S}{K_S+S_S} \cdot X_H$
Anoxic storage of SS			-1	0.03	0.07	-0.07	0.007				YHNOX		0.48	$k_{STO} \cdot \eta_{NOX} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{NOX}}{K_{NOX}+S_{NOX}} \cdot \dfrac{S_S}{K_S+S_S} \cdot X_H$
Anoxic growth on XH	-0.6			-0.07			-0.005			1	$-\dfrac{1}{YH02}$		-0.06	$\mu_H \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \dfrac{X_{STO}/X_H}{K_{STO}+X_{STO}/X_H} \cdot X_H$
Anoxic growth - denitrification				-0.07	0.3	-0.3	0.016			1	$-\dfrac{1}{YHNOX}$		-0.21	$\mu_H \cdot \eta_{NOX} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{NOX}}{K_{NOX}+S_{NOX}} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \dfrac{X_{STO}/X_H}{K_{STO}+X_{STO}/X_H} \cdot X_H$
Aerobic endog. respiration	-0.8			0.066			0.005	fI		-1			-0.75	$b_{H,O2} \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot X_H$
Anoxic endog. respiration				0.066	0.28	-0.28	0.025	fI		-1			-0.75	$b_{H,NOX} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{NOX}}{K_{NOX}+S_{NOX}} \cdot X_H$
Aerobic respiration of XSTO	-1										-1		-0.6	$b_{STO,O2} \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot X_{STO}$
Anoxic respiration of XSTO					0.35	-0.35	0.025				-1		-0.6	$b_{STO,NOX} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{NOX}}{K_{NOX}+S_{NOX}} \cdot X_{STO}$
Aerobic growth of XA	-18.04			-4.24		$\dfrac{1}{YA}$	-0.6					1	0.9	$\mu_A \cdot \dfrac{S_{O2}}{K_{A,O2}+S_{O2}} \cdot \dfrac{S_{NH4}}{K_{A,NH4}+S_{NH4}} \cdot \dfrac{S_{ALK}}{K_{A,ALK}+S_{ALK}} \cdot X_A$
Aerobic endog. respiration	-0.8			0.066			0.005	fI				-1	-0.75	$b_{A,O2} \cdot \dfrac{S_{O2}}{K_{A,O2}+S_{O2}} \cdot X_A$
Anoxic endog. respiration				0.066	0.28	-0.28	0.025	fI				-1	-0.75	$b_{A,NOX} \cdot \dfrac{K_{A,O2}}{K_{A,O2}+S_{O2}} \cdot \dfrac{S_{NOX}}{K_{NOX}+S_{NOX}} \cdot X_A$
	Dissolved oxygen	Inert soluble organic material	Readily biodegradable organic substrates	Ammonium plus ammonia nitrogen	Dinitrogen	Nitrate plus nitrite nitrogen	Alkalinity of wastewater	Inert particulate organic matter	Slowly biodegradable substrates	Heterotrophs	A cell internal storage product of PAO	Autotrophs	Suspended solids	

Figure 4.4: Activated Sludge Model No. 3 in a matrix format

73

The biological models described above are used to form the mass balance models for the Benchmark simulation plant and the Athlone plant. Based on the Peterson matrix tables above, the full set of ordinary differential equations describing the dynamic behaviour of a respective wastewater component may be formulated. Given a set of initial conditions, values of the parameters of the models and typical disturbances (i.e. influent wastewater flow and concentration), these equations may be used in simulations to predict effluent wastewater concentrations. Table 4.4 gives a summation of the dimensions of the vectors of the components and processes of the models.

Table 4.4: Dimensions of the vectors of the components and processes of the full models

DIMENSIONS OF THE VECTORS OF THE MODELS' COMPONENTS AND PROCESSES					
Description	Symbol	ASM1	ASM2	ASM2d	ASM3
Wastewater components	x	R^{13}	R^{19}	R^{19}	R^{13}
Process rates	P	R^{8}	R^{19}	R^{21}	R^{12}

4.3 Reduced activated sludge process models

The full biological models are used for prediction of the behaviour of the activated sludge processes and for prediction of variations in the effluent quality during dynamic loading conditions. These processes are time varying and extremely nonlinear. They are also subject to significant instability and high dimensions in terms of the number of state variables (wastewater components), processes and parameters. To accurately predict the behaviour of a true plant, activated sludge models require online modifications based on available data from sensors. Since the parameters of the models vary with time, model for online control purpose are related to quantities and variables that are possible to be measured online. All state variables of the reduced models developed in the thesis are supposed to be measurable.

Due to sensor unavailability and process complexities, influent wastewater concentrations, which are essential for prediction of the model behaviour are not easily characterised. It is therefore difficult to estimate and update varying model parameters online. These results in model identifiability problems; where models do not have unique set of parameters that explains certain behaviour. To counter the identification and sensor problems regarding full models, the reduced order models with smaller number of states, processes and parameters, capable to adequately describe major dynamic behaviour of carbonaceous, nitrogenous and phosphorus activities of the activated sludge process (ASP) are developed.

74

By reducing the numerical complexities and dimensions in the biological models, the overall mass balance models of the plants are also reduced. This will also reduces the long computational time required for process optimization. A fundamental requirement for the reduced models is that they contain a minimum number of state variables and parameters to allow for model identification based on online measurements, whilst maintaining the basic mechanistic structure of the reference model. As far as parameters are concerned, their number naturally decreases with decreasing of the number of state variables allowing for a possible identifiable model structure to be formed.

In this section, the IWA activated sludge models (ASM2, ASM2d and ASM3) are reduced based on the same reduction procedure used to reduce ASM1 in Lukasse (1999). The reduced models are later on used for optimal control of the ASP. Parameters of the reduced models are later on estimated to ensure that the reduced models keep the same prediction capabilities as the full models.

4.3.1   Reduction of the biological models

One of the objectives of this research project is to apply reduced models as an online tool for optimal control of the ASP. Because the dynamics of the ASP are very stiff and classified by different time scales; slow (weeks), medium (hours), and fast (minutes). The models are usually reduced on the basis of the different time scales separation. This means that the reduced models incorporate only variables with dynamics of interest (Lukasse, 1999:48).

In the activated sludge process, biomass is considered to be largely unknown and with slow dynamics, carbon, nitrogen and phosphorus substrates are characterised as variables with medium dynamics and the DO concentration is considered to have the fastest dynamics. Since carbon, nitrogen and phosphorus concentrations are the objects of interest in this project, biomass (variables with slow dynamics) are neglected and assumed to be in steady state while the variable with fast dynamics (DO) is ignored.

DO concentration is chosen as the control variable in this research project. This is due to its ability to affect the efficiency of the ASP. It is controlled separately on the layer of direct control shown in Figure 5.1. The DO control structure is designed in such a way that sufficient amount of oxygen is supplied in the aerobic reactor.

75

For this reason DO concentration (S_O for ASM1 and S_{O2} for the other three models) is not included in all the reduced biological models. Alkalinity (S_{ALK}) is also not included in the reduced models as it does not affect any other process and its dynamics are merely included in the full model to give an indication of pH in wastewater (Jeppsson, 2004).

4.3.1.1 Reduced activated sludge model no. 1

du Plessis (2009) identified the reduced activated sludge model no. 1 (reduced-ASM1) in Lukasse (1999). The results proved that the reduced-ASM1 model still captures the essential part of the carbon oxidation, nitrification and denitrification processes of the ASP. The model describes three wastewater components and four biological processes that mainly affect carbonification and nitrification/denitrification processes . A Peterson matrix representation of the reduced-ASM1 is given in Figure 4.5.

States / Processes	SS	SNO	SNH	Process rates ($P_1 - P_4$)
Aerobic growth of	$-\dfrac{1}{YH}$		$-iXB$	$\hat{\mu}_H \cdot \dfrac{S_s}{K_s + S_s} \cdot \dfrac{S_o}{K_{OH} + S_o} \cdot X_{BH}$
Anoxic growth of	$-\dfrac{1}{YH}$	$-\dfrac{1-YH}{2.86YH}$	$-iXB$	$\hat{\mu}_H \cdot \dfrac{S_s}{K_s + S_s} \cdot \dfrac{K_{OH}}{K_{OH} + S_o} \cdot \dfrac{S_{NO}}{K_{NO} + S_{NO}} \cdot \eta_g X_{BH}$
Aerobic growth of		$\dfrac{1}{YA}$	$-iXB - \dfrac{1}{YH}$	$\hat{\mu}_A \cdot \dfrac{S_{NH}}{K_{NH} + S_{NH}} \cdot \dfrac{S_o}{K_{OA} + S_o} X_{BA}$
Hydrolysis	1			$k_h \cdot \dfrac{X_s/X_{BH}}{K_x + X_s/X_{BH}} \left[\dfrac{S_o}{K_{OH} + S_o} + \eta_h \cdot \dfrac{K_{OH}}{K_{OH} + S_o} \cdot \dfrac{S_{NO}}{K_{OH} + S_{NO}} \right] X_{BH}$
	Readily biodegradable substrates	Nitrate and nitrite nitrogen	Ammonia and ammonium nitrogen	

Figure 4.5: Reduced Activated Sludge Model No.1 in a Peterson matrix format

Below is a description of how the full ASM1 was reduced.

As mentioned already, due to the time varying characteristic of the model parameters, models for online control purpose are related to quantities and variables that are measurable online. ASM1 variables possible to measure online are S_S, S_{NH} and S_{NO}.

The dynamics of heterotrophic organisms (X_{BH}), autotrophic organisms (X_{BA}) and slowly biodegradable substrate (X_S) are considered to be very slow hence these variables are treated as constants.

$$P_3 = \hat{\mu}_A \cdot \frac{S_{NH}}{K_{NH} + S_{NH}} \cdot \frac{S_O}{K_{O,A} + S_O} \cdot X_{B,A} \tag{4.65}$$

$$P_4 = k_h \cdot \frac{X_S/X_{B,H}}{K_X + X_S/X_{B,H}} \left[\frac{S_O}{K_{O,H} + S_O} + \eta_h \cdot \frac{K_{O,H}}{K_{O,H} + S_O} \cdot \frac{S_{NO}}{K_{NO} + S_{NO}} \right] X_{B,H} \tag{4.66}$$

The above described reduced-ASM1 model is implemented as it is in this project.

4.3.1.2 Reduced activated sludge model no. 2

The formation of the reduced Activated Sludge Model No. 2 (reduced-ASM2) is done using the same analysis of the full model as for the reduced-ASM1 model above. The reduced ASM2 has eight wastewater compounds and twelve processes in total.

In the reduced model dissolved oxygen (S_{O2}), alkalinity (S_{ALK}), heterotrophic organisms (X_H), nitrifying organisms (X_{AUT}), soluble inert non-biodegradable organic material (S_I) and inert non-biodegradable organic material (X_I), are omitted in the same way as in the reduced Activated Sludge Model no.1. Similar to reduced-ASM1, ammonia nitrogen (S_{NH4}) is multiplied by a factor f (1.2) to account for the organic nitrogen in the system.

Dinitrogen (S_{N2}) is not considered because it is a nitrogen gas that disperses into the atmosphere. The total suspended solids (X_{TSS}) are negligible and therefore ignored. Chemical processes (precipitation and re-dissolution) are removed from the model because phosphorus is removed biologically in activated sludge systems; as a result the chemical compounds (X_{MeP}, X_{MeOH}) are not included in the reduced model. Similar to the other organisms, dynamics of phosphate-accumulating organisms (X_{PAO}) are very slow and therefore they are assumed to be in steady state and are treated as constants. Similar to the full ASM2, the reduced-ASM2 describes carbon oxidation, nitrification, denitrification and phosphorus removal processes of the ASP. A Peterson matrix representation of the reduced-ASM2 is given in Figure 4.6.

78

Processes \ States	SF	SA	SNH4	SNO3	SPO4	XS	XPP	XPHA	Process rates (P$_1$ - P$_{12}$)
Aerobic hydrolysis	$1-f_{SI}$		0.01			-1			$K_h \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H$
Anoxic hydrolysis	$1-f_{SI}$		0.01			-1			$K_h \cdot \eta_{NO3} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \dfrac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H$
Anaerobic hydrolysis	$1-f_{SI}$		0.01			-1			$K_h \cdot \eta_{fe} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \dfrac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H$
Aerobic growth on SF	$\dfrac{1}{YH}$								$\mu_H \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_F}{K_F+S_F} \cdot \dfrac{S_F}{S_F+S_A} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{PO4}}{K_P+S_{PO4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H$
Aerobic growth on SA		$\dfrac{1}{YH}$							$\mu_H \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_A}{K_A+S_A} \cdot \dfrac{S_A}{S_F+S_A} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{PO4}}{K_P+S_{PO4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H$
Anoxic growth on SF	$\dfrac{1}{YH}$			$\dfrac{1}{2.86}\dfrac{YH}{YH}$					$\mu_H \cdot \eta_{NO3} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_F}{K_F+S_F} \cdot \dfrac{S_F}{S_F+S_A} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H$
Anoxic growth on SA		$\dfrac{1}{YH}$		$\dfrac{1}{2.86}\dfrac{YH}{YH}$					$\mu_H \cdot \eta_{NO3} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_A}{K_A+S_A} \cdot \dfrac{S_A}{S_F+S_A} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \dfrac{S_{PO4}}{K_P+S_{PO4}} \cdot X_H$
Fermentation	-1	1							$q_{fe} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \dfrac{S_F}{K_{fe}+S_F} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H$
Storage of XPHA		-1			YP04		-YP04	1	$q_{PHA} \cdot \dfrac{S_A}{K_A+S_A} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \dfrac{X_{PP}/X_{PAO}}{K_{PP}+X_{PP}/X_{PAO}} \cdot X_{PAO}$
Storage of XPP					-1		1	-YPHA	$q_{PP} \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{PO4}}{K_{PS}+S_{PO4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \dfrac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot \dfrac{K_{MAX}-X_{PP}/X_{PAO}}{K_{IPP}+K_{MAX}-X_{PP}/X_{PAO}} \cdot X_{PAO}$
Aerobic growth of XPAO					-iPBM			$\dfrac{1}{YA}$	$\mu_{PAO} \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \dfrac{S_{PO4}}{K_P+S_{PO4}} \cdot \dfrac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot X_{PAO}$
Aerobic growth of XAUT			$-iNBM-\dfrac{1}{YA}$	$\dfrac{1}{YA}$					$\mu_{AUT} \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{PO4}}{K_P+S_{PO4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT}$
	Fermentable readily biodegradable organic substrate	Fermentation products	Ammonium plus ammonia nitrogen	Nitrate plus nitrite nitrogen	Inorganic soluble phosphorus	Slowly biodegradable substrates	Poly-phosphate	A cell internal storage product of PAO	

Figure 4.6: Reduced Activated Sludge Model No.2 in a matrix format

The components of the ASM2 model described by the vector $x \in R^8$ may be presented in a vector form as:

$$x = [S_F \quad S_A \quad S_{NH4} \quad S_{NO3} \quad S_{PO4} \quad X_S \quad X_{PP} \quad X_{PHA}]^T$$

The vector of the process rates described by the vector $P \in R^{12}$ may be presented in a vector form as:

$$P = [P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5 \quad P_6 \quad P_7 \quad P_8 \quad P_9 \quad P_{10} \quad P_{11} \quad P_{12}]^T$$

where:

$$P_1 = K_h \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{X_S/X_H}{K_X + X_S/X_H} \cdot X_H \tag{4.67}$$

$$P_2 = K_h \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{X_S/X_H}{K_X + X_S/X_H} \cdot X_H \tag{4.68}$$

$$P_3 = K_h \cdot \eta_{fe} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{X_S/X_H}{K_X + X_S/X_H} \cdot X_H \tag{4.69}$$

$$P_4 = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{K_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.70}$$

$$P_5 = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{K_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.71}$$

$$P_6 = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.72}$$

$$P_7 = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot X_H \tag{4.73}$$

$$P_8 = q_{fe} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_F}{K_{fe} + S_F} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.74}$$

$$P_9 = q_{PHA} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{PP}/X_{PAO}}{K_{PP} + X_{PP}/X_{PAO}} \cdot X_{PAO} \tag{4.75}$$

$$P_{10} = q_{PP} \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{PO4}}{K_{PS} + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA} + X_{PHA}/X_{PAO}} \cdot \frac{K_{MAX} - X_{PP}/X_{PAO}}{K_{IPP} + K_{MAX} - X_{PP}/X_{PAO}} \cdot X_{PAO} \tag{4.76}$$

$$P_{11} = \mu_{PAO} \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA} + X_{PHA}/X_{PAO}} \cdot X_{PAO} \tag{4.77}$$

$$P_{12} = \mu_{AUT} \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_{AUT} \tag{4.78}$$

4.3.1.3 Reduced activated sludge model no. 2d

The formation of the reduced Activated Sludge Model No. 2d (reduced-ASM2d) is also done using the same analysis of the full model as above for reduced-ASM1. Similar to the full ASM2d, the reduced-ASM2d describes eight wastewater components and fourteen biological processes in total.

The reduced-ASM2d still describes carbon oxidation, nitrification, denitrification, biological phosphorus removal processes of the ASP. A Peterson matrix representation of the reduced-ASM2d is given in Figure 4.7.

The components of the reduced ASM2d model described by the vector $x \in R^8$ may be presented in a vector form as:

$$x = \begin{bmatrix} S_F & S_A & S_{NH4} & S_{NO3} & S_{PO4} & X_S & X_{PP} & X_{PHA} \end{bmatrix}^T$$

The vector of the process rates described by the vector $P \in R^{14}$ may be presented in a vector form as:

$$P = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 & P_9 & P_{10} & P_{11} & P_{12} & P_{13} & P_{14} \end{bmatrix}^T$$

where:

$$P_1 = K_h \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{X_S / X_H}{K_X + X_S / X_H} \cdot X_H \tag{4.79}$$

$$P_2 = K_h \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{X_S / X_H}{K_X + X_S / X_H} \cdot X_H \tag{4.80}$$

$$P_3 = K_h \cdot \eta_{fe} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{X_S / X_H}{K_X + X_S / X_H} \cdot X_H \tag{4.81}$$

$$P_4 = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{K_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.82}$$

$$P_5 = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.83}$$

$$P_6 = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.84}$$

$$P_7 = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot X_H \tag{4.85}$$

$$P_8 = q_{fe} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_F}{K_{fe} + S_F} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{4.86}$$

$$P_9 = q_{PHA} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{PP} / X_{PAO}}{K_{PP} + X_{PP} / X_{PAO}} \cdot X_{PAO} \tag{4.87}$$

$$P_{10} = q_{PP} \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{PO4}}{K_{PS} + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{PHA} / X_{PAO}}{K_{PHA} + X_{PHA} / X_{PAO}} \cdot \frac{K_{MAX} - X_{PP} / X_{PAO}}{K_{IPP} + K_{MAX} - X_{PP} / X_{PAO}} \cdot X_{PAO} \tag{4.88}$$

$$P_{11} = \mu_{PAO} \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{X_{PHA} / X_{PAO}}{K_{PHA} + X_{PHA} / X_{PAO}} \cdot X_{PAO} \tag{4.89}$$

$$P_{12} = P_{11} \cdot \eta_{NO3} \cdot \frac{K_{O2}}{S_{O2}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \tag{4.90}$$

$$P_{13} = \mu_{AUT} \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_{AUT} \tag{4.91}$$

$$P_{14} = P_{13} \cdot \eta_{NO3} \cdot \frac{K_{O2}}{S_{O2}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \tag{4.92}$$

State / Processes	SF	SA	SNH4	SNO3	SPO4	XS	XPP	XPHA	Process rates (P1 - P14)
Aerobic hydrolysis	$1-f_{SI}$		0.01			-1			$K_h \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H$
Anoxic hydrolysis	$1-f_{SI}$		0.01			-1			$K_h \cdot \eta_{NO3} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \dfrac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H$
Anaerobic hydrolysis	$1-f_{SI}$		0.01			-1			$K_h \cdot \eta_{fe} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \dfrac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H$
Aerobic growth on SF	$-\dfrac{1}{YH}$								$\mu_H \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_F}{K_F+S_F} \cdot \dfrac{S_F}{S_F+S_A} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{PO4}}{K_P+S_{PO4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H$
Aerobic growth on SA		$-\dfrac{1}{YH}$							$\mu_H \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_A}{K_A+S_A} \cdot \dfrac{S_A}{S_F+S_A} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{PO4}}{K_P+S_{PO4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H$
Anoxic growth on SF	$-\dfrac{1}{YH}$			$\dfrac{1-YH}{2.86\,YH}$					$\mu_H \cdot \eta_{NO3} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \dfrac{S_F}{K_F+S_F} \cdot \dfrac{S_F}{S_F+S_A} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{PO4}}{K_P+S_{PO4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H$
Anoxic growth on SA		$-\dfrac{1}{YH}$		$\dfrac{1-YH}{2.86\,YH}$					$\mu_H \cdot \eta_{NO3} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \dfrac{S_A}{K_A+S_A} \cdot \dfrac{S_A}{S_F+S_A} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{PO4}}{K_P+S_{PO4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H$
Fermentation	-1	1							$q_{fe} \cdot \dfrac{K_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \dfrac{S_F}{K_F+S_F} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H$
Storage of XPHA		-1			YPO4		-YPO4	1	$q_{PHA} \cdot \dfrac{S_A}{K_A+S_A} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \dfrac{X_{PP}/X_{PAO}}{K_{PP}+X_{PP}/X_{PAO}} \cdot X_{PAO}$
Aerobic storage of XPP					-1		1	YPHA	$q_{PP} \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{PO4}}{K_{PS}+S_{PO4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \dfrac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot \dfrac{K_{MAX}-X_{PP}/X_{PAO}}{K_{IPP}+K_{MAX}-X_{PP}/X_{PAO}} \cdot X_{PAO}$
Anoxic storage of XPP				-0.07	-1		1	YPHA	$P_{10} \cdot \eta_{NO3} \cdot \dfrac{K_{O2}}{S_{O2}} \cdot \dfrac{S_{NO3}}{K_{NO3}+S_{NO3}}$
Aerobic growth of XPAO					iPBM			$\dfrac{1}{YH}$	$\mu_{PAO} \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{PO4}}{K_P+S_{PO4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \dfrac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot X_{PAO}$
Anoxic growth of XPAO				-0.24	iPBM			$\dfrac{1}{YH}$	$P_{12} \cdot \eta_{NO3} \cdot \dfrac{K_{O2}}{S_{O2}} \cdot \dfrac{S_{NO3}}{K_{NO3}+S_{NO3}}$
Aerobic growth of XAUT			-4.24	$\dfrac{1}{YA}$	iPBM				$\mu_{AUT} \cdot \dfrac{S_{O2}}{K_{O2}+S_{O2}} \cdot \dfrac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \dfrac{S_{PO4}}{K_P+S_{PO4}} \cdot \dfrac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT}$
	Fermentable readily biodegradable organic substrate	Fermentation products	Ammonium plus ammonia nitrogen	Nitrate plus nitrite nitrogen	Inorganic soluble phosphorus	Slowly biodegradable substrates	Poly-phosphate	A cell internal storage product of PAO	

Figure 4.7: Reduced Activated Sludge Model No.2d in a matrix format

4.3.1.4 Reduced activated sludge model no. 3

The formation of the reduced Activated Sludge Model No. 3 (reduced-ASM3) is also done using the same analysis of the full model as above for reduced-ASM3. Ammonia nitrogen (S_{NH4}) in the reduced-ASM3 model is also multiplied by a factor **f** (1.2) to account for the organic nitrogen in the system. The reduced-ASM3 describes three wastewater components and four biological processes in total. Similar to the full ASM3, the reduced-ASM3 model still describes carbon oxidation, nitrification and denitrification processes of the ASP. A Peterson matrix representation of the reduced-ASM3 is given in Figure 4.8.

States \ Processes	SS	SNH4	SNOX	Process rates ($P_1 - P_4$)
Hydrolysis	1	0.01		$k_H \cdot \dfrac{X_S / X_H}{K_X + X_S / X_H} \cdot X_H$
Anoxic growth on XH		-0.07		$\mu_H \cdot \dfrac{S_{O2}}{K_{O2} + S_{O2}} \cdot \dfrac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \dfrac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot$ $\dfrac{X_{STO} / X_H}{K_{STO} + X_{STO} / X_H} \cdot X_H$
Anoxic growth - denitrification		-0.07	-0.3	$\mu_H \cdot \eta_{NOX} \cdot \dfrac{K_{O2}}{K_{O2} + S_{O2}} \cdot \dfrac{S_{NOX}}{K_{NOX} + S_{NOX}} \cdot \dfrac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot$ $\dfrac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \dfrac{X_{STO} / X_H}{K_{STO} + X_{STO} / X_H} \cdot X_H$
Aerobic growth of XA		-4.24	$\dfrac{1}{YA}$	$\mu_A \cdot \dfrac{S_{O2}}{K_{A,O2} + S_{O2}} \cdot \dfrac{S_{NH4}}{K_{A,NH4} + S_{NH4}} \cdot$ $\dfrac{S_{ALK}}{K_{A,ALK} + S_{ALK}} \cdot X_A$
	Readily biodegradable organic substrates	Ammonium plus ammonia nitrogen	Nitrate plus nitrite nitrogen	

Figure 4.8: Reduced Activated Sludge Model No.3 in a matrix format

The components of the reduced ASM3 model described by the vector $x \in R^3$ may be presented in a vector form as:

$$x = \begin{bmatrix} S_S & S_{NH4} & S_{NOX} \end{bmatrix}^T$$

The vector of the process rates described by the vector $P \in R^4$ may be presented in a vector form as:

$$P = [P_1 \quad P_2 \quad P_3 \quad P_4]^T$$

where:

$$P_1 = k_H \cdot \frac{X_S/X_H}{K_X + X_S/X_H} \cdot X_H \tag{4.93}$$

$$P_2 = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{STO}/X_H}{K_{STO} + X_{STO}/X_H} \cdot X_H \tag{4.94}$$

$$P_3 = \mu_H \cdot \eta_{NOX} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NOX}}{K_{NOX} + S_{NOX}} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{STO}/X_H}{K_{STO} + X_{STO}/X_H} \cdot X_H \tag{4.95}$$

$$P_4 = \mu_A \cdot \frac{S_{O2}}{K_{A,O2} + S_{O2}} \cdot \frac{S_{NH4}}{K_{A,NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{A,ALK} + S_{ALK}} \cdot X_A \tag{4.96}$$

The reduced biological models described above are used to form the reduced mass balance models for the Benchmark simulation plant and the Athlone plant. Given a set of initial conditions, values of the parameters of the models and typical disturbances (i.e. influent wastewater flow and concentration), these equations may be used in simulations to predict effluent wastewater concentrations. Table 4.5 gives a summation of the dimensions of the vectors of the components and processes of the reduced models.

Table 4.5: Dimensions of the vectors of the components and processes of the reduced models

DIMENSIONS OF THE VECTORS OF THE MODELS' COMPONENTS AND PROCESSES					
Description	Symbol	Reduced -ASM1	Reduced - ASM2	Reduced - ASM2d	Reduced - ASM3
Wastewater components	X	R^3	R^8	R^8	R^3
Process rates	P	R^4	R^{12}	R^{14}	R^4

4.4 Mass balance models

Using mass balance principle, the overall models for the considered plants are derived, with the IWA models representing the mass of the wastewater components produced or consumed due to the biological reactions within a bioreactor. In the preceding section, the biological activated sludge mathematical models are described in details. The mass balance models with the full biological models are referred to as the full mass balance models and the mass balance models with the reduced biological models are from this point referred to as the reduced mass balance models.

This section presents the derivations of the mass balance models of the Athlone and Benchmark wastewater treatment plants (WWTPs). Mass balance models are based on the physical layouts of the WWTPs, they account for materials entering the system, produced or consumed in the system and those leaving the system. Basically the biological models describe biological reactions occurring inside the tanks while mass balance equations describe transportation of wastewater compounds to and from the tanks. The tanks are characterised by volumes (V), flow rates (Q) as well as states of the wastewater components (x).

Fundamental to the mass balance is the conservation of mass principle, i.e. matter cannot disappear or be created [Yunus et al., 2004]. The generic equation for a mass balance around a certain system is:

Time variation of the mass of the component in the system (per unit volume) = Mass of the input component ± Mass of the component produced or consumed due to the reaction – Mass of the output component (Dochain & Vanrolleghem, 2009) (4.97)

4.4.1 Mass balance models in continuous form

The generic mass balance Equation (4.97) above in its rate form can be represented with mathematical notations as follows:

$$V \cdot \frac{d(x)}{dt} = Q_{in} x_{in} \pm V \cdot r - Q_{out} x_{out}$$ (4.98)

where Q_{in} is the input flow rate; x_{in} is the components concentration in the input; Q_{out} is the output flow rate; x_{out} is the components concentration in the output; r is the rate (per unit volume) at which components are produced or consumed. Equation (4.98) expresses that the rate of change of the mass balance of a given component into the tank is equal to the rate of change of the input component ± rate of change of the mass of the produced or consumed component due to reaction minus rate of change of the output component. Equation (4.98) is applied further for development of the mass balance model of the Athlone process and the Benchmark process plants.

4.4.1.1 Athlone mass balance model

The Athlone plant is situated in the middle part of Cape Town, South Africa. The plant has been in operation since 1921 and it is also faced with problems related to plant deterioration. Typical plant problems include faulty sludge handling apparatus and

constant failing of the air blowers (South African Government, 2009). The plant consists of three biological reactors; Anaerobic, Anoxic and Aerobic. This organization of biological reactors is referred to as the pre-nitrification activated sludge process. The Athlone plant is capable to treat approximately 80 million litre of water daily (80 000 m^3/day).

The layout of the Athlone plant showing different flows through the system is presented in Figure 4.9 where Q_p is the recycled flow rate for Tank 2; Q_a is the recycle flow rate for Tank 3; Q_w is the waste sludge flow rate; Q_0 is the influent flow rate; Q_1 is the output flow rate for Tank 1; Q_2 is the output flow rate for Tank 2; Q_3 is the output flow rate for Tank 3; Q_e is the effluent flow rate, Q_r is the settler recycle flow rate, x_0 is the vector of components concentrations in the influent wastewater, x_1 is the vector of components concentrations in Tank 1, x_2 is the vector of components concentrations in Tank 2, x_3 is the vector of components concentration in Tank 3, x_4 is the vector of difference between components concentrations from Tank 3 and from the wasted sludge, x_e is the vector of components concentrations in the effluent wastewater and x_r is the vector of recycle concentrations of the components of the wastewater sludge.

The values of the flow rates and the tank volumes for the Athlone wastewater treatment plant are given in Table 4.6.

Table 4.6: Athlone plant flow rates and tank volumes

Symbol	Name	Value	Unit
Q_a	Tank 3 to Tank 2 recycled flow rate	39917	m^3/day
Q_p	Tank 2 to Tank 1 recycled flow rate	39917	m^3/day
Q_w	Flow rate for the wasted sludge	3840	m^3/day
Q_0	Influent flow rate	40003	m^3/day
Q_r	Recycled flow rate from the settler	$Q_0 + Q_r - Q_w - Q_e$	m^3/day
Q_e	Effluent flow rate	$Q_0 - Q_w$	m^3/day
V_1	Tank 1 volume	1148	m^3
V_2	Tank 2 volume	1148	m^3
V_3	Tank 3 volume	5273	m^3
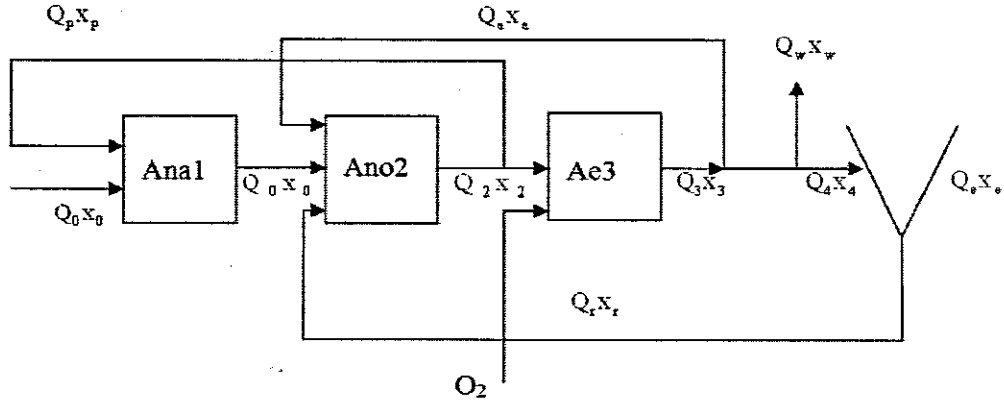
Figure 4.9: Layout of the Athlone plant

The mass balance model of the Athlone plant is derived as follows:

a) Firstly, the input and output flow rates of each tank and the settler are derived

$$Q_1 = Q_0 + Q_p \tag{4.99}$$

$$Q_2 = Q_1 + Q_a + Q_r = Q_0 + Q_p + Q_a + Q_r \tag{4.100}$$

$$Q_3 = Q_2 - Q_p = Q_0 + Q_a + Q_r \tag{4.101}$$

$$Q_4 = Q_3 - Q_a - Q_w = Q_0 + Q_r - Q_w \tag{4.102}$$

$$Q_r = Q_4 - Q_e = Q_0 + Q_r - Q_w - Q_e \tag{4.103}$$

$$Q_e = Q_0 - Q_w \tag{4.104}$$

b) Secondly, the input and output mass balance equations for the settler components are derived. The settler is assumed to be ideal with no biological reactions; meaning that there is no production or consumption of components in the settler. The input concentration of the settler components is:

$$x_4 = x_a = x_w = x_3 \tag{4.105}$$

Mass balance equation of components at the input of the settler is:

$$Q_4 x_4 = Q_4 x_3 = Q_3 x_3 - Q_a x_a - Q_w x_w = (Q_0 + Q_r - Q_w) x_3 \tag{4.106}$$

Mass balance equation of components at the output of the settler is:

$$Q_r x_r = Q_4 x_4 - Q_e x_e = (Q_0 + Q_r - Q_w) x_3 - (Q_0 - Q_w) x_e \tag{4.107}$$

c) Finally, equation for calculating concentrations of the recycled components is derived. Based on Equation (4.107), equation for concentration of the recycled components is:

$$x_r = \frac{(Q_0 + Q_r - Q_w) x_3 - (Q_0 - Q_w) x_e}{Q_r} \tag{4.108}$$

The recycle concentration values depend on whether the wastewater component is particulate or soluble. For the soluble materials, the effluent concentration is:

$$x_e = x_3 \tag{4.109}$$

Therefore

$$x_r = \frac{(Q_0 + Q_r - Q_w)x_3 - (Q_0 - Q_w)x_3}{Q_r} = x_3 \tag{4.110}$$

For the particulate materials, the effluent concentration is:

$$x_e = 0 \tag{4.111}$$

Therefore

$$x_r = \frac{(Q_0 + Q_r - Q_w)x_3}{Q_r} \tag{4.112}$$

For the purpose of simplification a coefficient λ for representation of the fraction of the flow rates is introduced as:

$$\lambda = \frac{(Q_0 + Q_r - Q_w)}{Q_r} \tag{4.113}$$

Therefore

$$x_r = \lambda x_3 \tag{4.114}$$

With all elements described above, and using the generic mass balance Equation (4.98) as a guide, the mass balance equations for the components in each of the three Athlone tanks can be derived.

d) Mass balance equation for Tank 1:

Table 4.7 gives the inputs and outputs rates of change of mass flows of the components going in and out of Tank 1 respectively.

Table 4.7: Components for Tank 1

Inputs	Outputs	Reaction rates
$Q_0 x_0$ $Q_p x_p$	$Q_1 x_1$	r_1

Mass balance equation for components in Tank 1 is:

$$\frac{dx_1}{dt} = \frac{1}{V1}\left[Q_0 x_0 + Q_p x_2 - Q_1 x_1\right] + r_1 \qquad x_1(0) = x_{10} \tag{4.115}$$

where r_1 is the rate (per unit volume) at which components in Tank 1 are produced or consumed, and the vector x_{10} contains the initial conditions for every component.

e) Mass balance equation for Tank 2:

Table 4.8 gives the inputs and outputs rates of change of mass of the components going in and out of Tank 2 respectively.

Table 4.8: Components for Tank 2

Inputs	Outputs	Reaction rates
$Q_1 x_1$ $Q_a x_3$ $Q_r x_3$ for soluble material $Q_r \lambda x_3$ for particulate material	$Q_2 x_2$	r_2

For soluble material, mass balance equation for components in Tank 2 is:

$$\frac{dx_2}{dt} = \frac{1}{V2}[Q_1 x_1 + Q_a x_3 + Q_r x_3 - Q_2 x_2] + r_2 \qquad x_2(0) = x_{20} \qquad (4.116)$$

For particulate material, mass balance equation for components in Tank 2 is:

$$\frac{dx_2}{dt} = \frac{1}{V2}[Q_1 x_1 + Q_a x_3 + Q_r \lambda x_3 - Q_2 x_2] + r_2 \qquad (4.117)$$

where r_2 is the rate at which components in Tank 2 are produced or consumed and x_{20} is the vector of the initial conditions for every component

f) Mass balance equation for Tank 3:

Table 4.9 gives the inputs and outputs mass of the components going in and out of Tank 3 respectively.

Table 4.9: Components for Tank 3

Inputs	Outputs	Reaction rates
$(Q_2 - Q_p)x_2$	$Q_3 x_3$	r_3

Mass balance equation for components in Tank 3 is:

$$\frac{dx_3}{dt} = \frac{1}{V3}[(Q_2 - Q_p)x_2 - Q_3 x_3] + r_3 \qquad x_3(0) = x_{30} \qquad (4.118)$$

where r_3 is the rate (per unit volume) at which components in Tank 3 are produced or consumed and x_{30} is the vector of the initial conditions for every component.

In this tank (the aerobic tank), the equation for the dissolved oxygen concentration has additional member determined by the flow rate of the supplied air.

89

This flow rate is the control signal to the activated sludge process. The equation for the dissolved oxygen is:

$$\frac{dS_{o,3}}{dt} = \frac{1}{V3}\left[(Q_2 - Q_p)S_{o,2} - Q_3 S_{o,3}\right] + r_3 + K_{La}(S_{o,sat} - S_{o,3}) \qquad\qquad S_{o,3}(0) = S_{o,3_0} \qquad\qquad (4.119)$$

where K_{La} is the oxygen transfer function and $S_{o,sat}$ is the oxygen concentration at saturation and $S_{o,3_0}$ is the initial condition for dissolved oxygen concentration. The value of K_{La} depends on the air flow rate to the aerobic tank and can be expressed by different linear or nonlinear functions of the air flow rate. $K_{La} = f(u)$, where u is the air flow rate.

4.4.1.2 Benchmark mass balance model

The Benchmark plant is a simulation plant model developed by the COST 682 working group No. 2 (Copp, 2002); its setup provides for nitrification with pre-denitrification processes used for nitrogen removal. The Benchmark plant is made up of two anoxic and three aerobic reactors for biological processes and one secondary settler for sedimentation purposes. The Benchmark simulation plant is part of a simulation environment that includes influent loads, control strategies and a detailed description of the derived mathematical model based on the ASM1 biological model.

The results of the different control strategies used on the Benchmark plant are well documented, hence the Benchmark simulation environment is a good platform for testing and comparing different control approaches.

The layout of the Benchmark plant showing different flows through the system is presented in Figure 4.10 where Q_a is the recycled flow from Tank 5; Q_w is the wasted sludge flow; Q_0 is the influent flow; Q_1 is the flow through Tank 1; Q_2 is the flow through Tank 2; Q_3 is the flow through Tank 3; Q_4 is the flow through Tank 4; Q_5 is the flow through Tank 5; Q_e is the effluent flow, Q_r is the settler recycled flow, x_1 is the vector of concentrations of Tank 1 process variables. x_2 is the vector of concentrations of Tank 2 process variables, x_3 is the vector of concentrations of Tank 3 process variables, x_4 is the vector of concentrations of Tank 4 process variables, x_5 is the vector of concentrations of Tank 5 process variables, x_6 is the difference between concentrations from Tank 5 and from the internal recycle flow, x_e is the effluent concentrations and x_r is the vector of recycled concentrations of the process variables.

90

The values of the flow rates and the tank volumes for the Benchmark wastewater treatment plant are given in Table 4.10.

Table 4.10: Benchmark plant flow rates and tank volumes

Symbol	Name	Value	Unit
Q_a	Tank 5 to Tank 1 recycled flow rate	55338	m³/day
Q_w	Flow rate for the wasted sludge	385	m³/day
Q_0	Influent flow rate	18446	m³/day
Q_r	Settler recycled flow rate	$Q_w + Q_a + Q_r - Q_e$	m³/day
Q_e	Effluent flow rate	$Q_0 - Q_w$	m³/day
V_1	Tank 1 volume	1000	m³
V_2	Tank 2 volume	1000	m³
V_3	Tank 3 volume	1000	m³
V_4	Tank 4 volume	1333	m³
V_5	Tank 5 volume	1333	m³
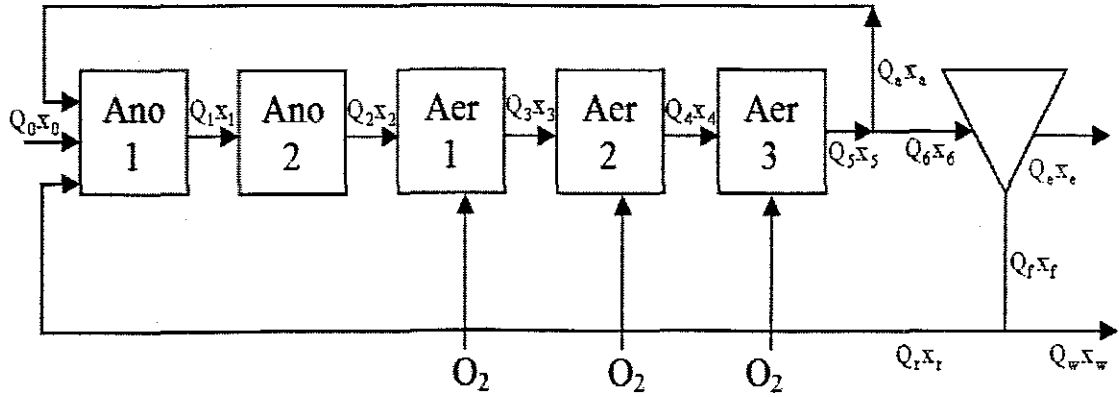
Figure 4.10: Layout of the Benchmark plant

The mass balance model of the Benchmark plant below is derived according to the Benchmark guidelines in Copp (2002).

The mass balance model of the Benchmark plant is derived as follows:

a) Similar to what is derivation for Athlone plant, input and output flow rates of the tanks are derived

$$Q_1 = Q_2 = Q_3 = Q_4 = Q_5 = Q_0 + Q_a + Q_r \qquad (4.120)$$

$$Q_6 = Q_5 - Q_a = Q_0 + Q_r \qquad (4.121)$$

$$Q_f = Q_r + Q_w \qquad (4.122)$$

$$Q_e = Q_6 - Q_f = Q_0 + Q_r - Q_r - Q_w \qquad (4.123)$$

$$Q_e = Q_0 - Q_w \qquad (4.124)$$

b) Then the input and output mass balance equations for the settler components are derived.

91

This settler is also considered to be ideal where no biological reactions take place and separation of the compounds is done at once. The settler used in the thesis is different from the settler used in the COST benchmark model.

c) The input concentration of the settler is:

$$x_6 = x_a = x_5$$ (4.125)

Mass balance equation of components at the input of the settler is

$$Q_6 x_6 = Q_6 x_5 = (Q_5 - Q_a) x_5 = (Q_0 + Q_r) x_5$$ (4.126)

Mass balance equations of components at the output of the settler is

$$Q_r x_f = Q_6 x_6 - Q_e x_e = (Q_0 + Q_r) x_5 - (Q_0 - Q_w) x_e$$ (4.127)

d) Finally the equation for the recycle flow concentration is derived.

$$Q_r x_f = Q_r x_r + Q_w x_w$$ (4.128)

The concentrations for the recycled and the waste flow are the same therefore:

$$Q_r x_f = Q_r x_r + Q_w x_r = (Q_r + Q_w) x_r$$ (4.129)

The recycle flow concentration can then be determined using the following equation:

$$(Q_0 + Q_r) x_5 - (Q_0 - Q_w) x_e = (Q_r + Q_w) x_r$$ (4.130)

Therefore:

$$x_r = \frac{(Q_0 + Q_r) x_5 - (Q_0 - Q_w) x_e}{(Q_r + Q_w)}$$ (4.131)

For the soluble materials the effluent concentration is:

$$x_e = x_5$$ (4.132)

Therefore

$$x_r = \frac{(Q_0 + Q_r) x_5 - (Q_0 - Q_w) x_5}{(Q_r + Q_w)}$$ (4.133)

Therefore

$$x_r = x_5$$ (4.134)

For the particulate materials the effluent concentration is:

$$x_e = 0$$ (4.135)

Therefore

$$x_r = \frac{(Q_0 + Q_r) x_5}{(Q_r + Q_w)}$$ (4.136)

A coefficient λ for representation of the fraction of flow rates is therefore:

$$\lambda = \frac{(Q_0 + Q_r)}{(Q_r + Q_w)}$$ (4.137)

Therefore

$$x_r = \lambda x_5 \tag{4.138}$$

The mass balance equations for the components in each of the five Benchmark tanks can then be derived as follows:

e) Mass balance equation for Tank 1:

Table 4.11 gives the inputs and outputs rate of change of mass of the components going in and out of Tank 1 respectively.

Table 4.11: Components for Tank 1

Inputs	Outputs	Reaction
$Q_0 x_0$ $Q_a x_a$ $Q_r x_5$ for soluble material $Q_r \lambda x_5$ for particulate material	$Q_1 x_1$	r_1

For soluble material, mass balance equation for components in Tank 1 is:

$$\frac{dx_1}{dt} = \frac{1}{V_1}[Q_0 x_0 + Q_a x_5 + Q_r x_5 - Q_1 x_1] + r_1 \qquad x_1(0) = x_{10} \tag{4.139}$$

For particulate material, mass balance equation for components in Tank 1 is:

$$\frac{dx_1}{dt} = \frac{1}{V_1}[Q_0 x_0 + Q_a x_5 + Q_r \lambda x_5 - Q_1 x_1] + r_1 \qquad x_1(0) = x_{10} \tag{4.140}$$

where r_1 is the rate (per unit volume) at which components in Tank 1 are produced or consumed, and the vector x_0 is the initial conditions vector.

f) Mass balance for Tank n (where n = tanks 2 to 5)

Table 4.12 gives the inputs and outputs rates of change of mass of the components going in and out of Tank 2 respectively.

Table 4.12: Components for Tank n

Inputs	Outputs	Reaction
$Q_{n-1} x_{n-1}$	$Q_n x_n$	r_n

Mass balance equation for components in Tank n is:

$$\frac{dx_n}{dt} = \frac{1}{V_n}[Q_{n-1} x_{n-1} - Q_n x_n] + r_n \qquad x_n(0) = x_{n0} \tag{4.141}$$

where r_n is the rate at which components in tanks 2-5 are produced or consumed.

For the aerobic tanks (Tank p where p = tanks 3 - 5), the equation for the dissolved oxygen is:

$$\frac{dS_{O,p}}{dt} = \frac{1}{V_p}\left[Q_{p-1}S_{O,p-1} - Q_p S_{O,p}\right] + r_p + K_{La}\left(S_{Osat} - S_{O,p}\right) \qquad S_{O,p}(0) = S_{O,p0} \qquad (4.142)$$

where K_{La_p} is the oxygen transfer function and $S_{o,sat}$ is the oxygen concentration at saturation.

Equations (4.115 - 4.119) and Equations (4.139 - 4.142) make up the mass balance model of the Athlone and the Benchmark Plants respectively. The above Athlone and Benchmark mass balance equations are used in the next section for the determination of the discrete mass balance models.

4.4.2 Mass balance models in discrete form

It is simpler to analyse the system with discrete equations on the computer programs. For this reason the continuous mass balance equations on section 4.4.1 are transferred into their discrete counterparts. The discrete mass balance equations are merely approximations of the continuous mass balance equations.

To produce a model that is in a discrete form, the differential terms of the mass balance equations are replaced with first difference equations as shown below.

$$\frac{dx}{dt} = \frac{x(k+1) - x(k)}{\Delta t} \qquad (4.143)$$

where Δt is the sampling time

4.4.2.1 Discrete form of the Athlone mass balance model

The discrete mass balance model of the Athlone plant is derived as follows:

a) Discrete mass balance equation for Tank 1 derivation:

$$\frac{dx_1}{dt} = \frac{1}{V1}\left[Q_0 x_0 + Q_p x_2 - Q_1 x_1\right] + r_1 \qquad x_1(0) = x_{10} \qquad (4.144)$$

$$\frac{x_1(k+1) - x_1(k)}{\Delta t} = \frac{1}{V1}\left[Q_0 x_0 + Q_p x_2 - Q_1 x_1\right] + r_1 \qquad (4.145)$$

$$x_1(k+1) = \frac{\Delta t}{V1}Q_0 x_0 + \left(1 - \frac{\Delta t}{V1}Q_1\right)x_1(k) + \frac{\Delta t}{V1}Q_p x_2(k) + r_1\Delta t \qquad x_1(0) = x_{10} \qquad (4.146)$$

b) Discrete mass balance equation for Tank 2 derivation:

$$\frac{dx_2}{dt} = \frac{1}{V2}\left[Q_1 x_1 + Q_a x_3 + Q_r x_3 - Q_2 x_2\right] + r_2 \qquad x_2(0) = x_{20} \qquad (4.147)$$

$$\frac{x_2(k+1) - x_2(k)}{\Delta t} = \frac{1}{V2}\left[Q_1 x_1(k) + Q_a x_3(k) + Q_r x_3(k) - Q_2 x_2(k)\right] + r_2 \qquad (4.148)$$

94

For the soluble materials:

$$x_2(k+1) = \frac{\Delta t}{V2} Q_1 x_1(k) + \left(1 - \frac{\Delta t}{V2} Q_2\right) x_2(k) + \frac{\Delta t}{V2}(Q_a + Q_r) x_3(k) + r_2 \Delta t \; , \; x_2(0) = x_{20} \qquad (4.149)$$

For the particulate materials:

$$x_2(k+1) = \frac{\Delta t}{V2} Q_1 x_1(k) + \left(1 - \frac{\Delta t}{V2} Q_2\right) x_2(k) + \frac{\Delta t}{V2}(Q_a + \lambda Q_r) x_3(k) + r_2 \Delta t \; , \; x_2(0) = x_{20} \qquad (4.150)$$

c) Discrete mass balance equation for Tank 3 derivation:

$$\frac{dx_3}{dt} = \frac{1}{V3}\left[(Q_2 - Q_p) x_2 - Q_3 x_3\right] + r_3 \qquad\qquad x_3(0) = x_{30} \qquad (4.151)$$

$$\frac{x_3(k+1) - x_3(k)}{\Delta t} = \frac{1}{V3}\left[(Q_2 - Q_p) x_2(k) - Q_3 x_3(k)\right] + r_3 \qquad (4.152)$$

$$x_3(k+1) = \frac{\Delta t}{V3}(Q_2 - Q_p) x_2(k) + \left(1 - \frac{\Delta t}{V3} Q_3\right) x_3(k) + r_3 \Delta t \qquad\qquad x_3(0) = x_{30} \qquad (4.153)$$

For the dissolved oxygen concentration (only for the aerobic tank) the discrete form of the equation is:

$$S_{o,3}(k+1) = \frac{\Delta t}{V_3}(Q_2 - Q_p) S_{o,2}(k) + \left(1 - \frac{\Delta t}{V_3} Q_3\right) S_{o,3}(k) + \Delta t r_{So,3} + \Delta t K_{LA}\left[S_{o,SAT} - S_{o,3}(k)\right] \qquad S_{o,3}(0) = S_{o,3_0} \quad (4.154)$$

Equations (4.144 – 4.154) make up the discrete mass balance model of the Athlone Plant.

4.4.2.2 Discrete form of the Benchmark mass balance model

a) Discrete mass balance equation for Tank 1 derivation:

$$\frac{dx_1}{dt} = \frac{1}{V_1}\left[Q_0 x_0 + Q_a x_5 + Q_r x_5 - Q_1 x_1\right] + r_1 \qquad\qquad x_1(0) = x_{10} \qquad (4.155)$$

$$\frac{x_1(k+1) - x_1(k)}{\Delta t} = \frac{1}{V_1}\left[Q_0 x_0 + Q_a x_5(k) + Q_r x_r(k) - Q_1 x_1(k)\right] + r_1 \qquad (4.156)$$

For the soluble materials:

$$x_1(k+1) = \frac{\Delta t}{V_1} Q_0 x_0 + \left(1 - \frac{\Delta t}{V_1} Q_1\right) x_1(k) + \frac{\Delta t}{V_1}(Q_a + Q_r) x_5(k) + r_1 \Delta t \; \; x_1(0) = x_{10} \qquad (4.157)$$

For the particulate materials:

$$x_1(k+1) = \frac{\Delta t}{V_1} Q_0 x_0 + \left(1 - \frac{\Delta t}{V_1} Q_1\right) x_1(k) + \frac{\Delta t}{V_1}(Q_a + \lambda Q_r) x_5(k) + r_1 \Delta t \; \; x_1(0) = x_{10} \qquad (4.158)$$

b) Discrete mass balance equation for Tank n (n = 2 - 5) derivation:

95

$$\frac{dx_n}{dt} = \frac{1}{V_n}[Q_{n-1}x_{n-1} - Q_n x_n] + r_n \qquad\qquad x_n(0) = x_{n0} \qquad\qquad (4.159)$$

$$\frac{x_n(k+1) - x_n(k)}{\Delta t} = \frac{1}{V_n}[Q_{n-1}x_{n-1}(k) - Q_n x_n(k)] + r_n \qquad\qquad (4.160)$$

$$x_n(k+1) = \frac{\Delta t}{V_n}Q_{n-1}x_{n-1}(k) + \left(1 - \frac{\Delta t}{V_n}Q_n\right)x_n(k) + r_n\Delta t \qquad\qquad x_n(0) = x_{n0} \qquad (4.161)$$

For the dissolved oxygen concentration (only for aerobic tanks) the discrete form of the equation is:

$$S_{o,n}(k+1) = \frac{\Delta t}{V_n}Q_{n-1}S_{o,n-1}(k) + \left(1 - \frac{\Delta t Q_n}{V_n}\right)S_{o,n}(k) + \Delta t r_{S_{o,n}} + \Delta t K_{LA}[S_{o,SAT} - S_{o,n}(k)] \qquad S_{o,n}(0) = S_{o,n0} \quad (4.162)$$

Equations (4.155 – 4.162) make up the discrete mass balance model of the Athlone Plant. The discrete Athlone and Benchmark mass balance models are used in section 4.4.3 to derive the state space mass balance models.

4.4.3 State space mass balance models

For simulation purpose, a better and simpler representation of the model is necessary. State space model provides a convenient and compact way to represent and analyse a system. This is especially useful when dealing with multi-input, multi-output time varying systems. In this section, the discrete mass balance models derived in the sections above are presented in state space form.

4.4.3.1 State space representation of the full mass balance models

The full mass balance model may be graphically presented as in Figure 4.11, where u_1 is the air flow rate supplied to the process, x_i is a vector of the inflow concentrations and x is a vector of the components concentrations.
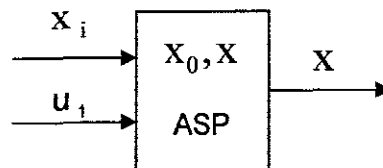
Figure 4.11: Graphical representation of the full mass balance model

The full mass balance equations in section 4.4.2 may then be presented by the following state space model equation:

$$x(k+1) = Ax(k) + r(x,k) + Bx_i(k) + K_{La}(u_1)^T[S_{o,sat} - S_o(k)] \quad x(0) = x_0, \quad k = \overline{0,K-1} \qquad (4.163)$$

$$\frac{dx_n}{dt} = \frac{1}{V_n}\left[Q_{n-1}x_{n-1} - Q_n x_n\right] + r_n \qquad\qquad x_n(0) = x_{n0} \qquad\qquad (4.159)$$

$$\frac{x_n(k+1) - x_n(k)}{\Delta t} = \frac{1}{V_n}\left[Q_{n-1}x_{n-1}(k) - Q_n x_n(k)\right] + r_n \qquad\qquad (4.160)$$

$$x_n(k+1) = \frac{\Delta t}{V_n}Q_{n-1}x_{n-1}(k) + \left(1 - \frac{\Delta t}{V_n}Q_n\right)x_n(k) + r_n\Delta t \qquad x_n(0) = x_{n0} \qquad (4.161)$$

For the dissolved oxygen concentration (only for aerobic tanks) the discrete form of the equation is:

$$S_{O,n}(k+1) = \frac{\Delta t}{V_n}Q_{n-1}S_{O,n-1}(k) + \left(1 - \frac{\Delta t Q_n}{V_n}\right)S_{O,n}(k) + \Delta t r_{SO,n} + \Delta t K_{LA}\left[S_{O,SAT} - S_{O,n}(k)\right] \qquad S_{O,n}(0) = S_{O,n0} \quad (4.162)$$

Equations (4.155 – 4.162) make up the discrete mass balance model of the Athlone Plant. The discrete Athlone and Benchmark mass balance models are used in section 4.4.3 to derive the state space mass balance models.

4.4.3 State space mass balance models

For simulation purpose, a better and simpler representation of the model is necessary. State space model provides a convenient and compact way to represent and analyse a system. This is especially useful when dealing with multi-input, multi-output time varying systems. In this section, the discrete mass balance models derived in the sections above are presented in state space form.

4.4.3.1 State space representation of the full mass balance models

The full mass balance model may be graphically presented as in Figure 4.11, where u_i is the air flow rate supplied to the process, x_i is a vector of the inflow concentrations and x is a vector of the components concentrations.

Figure 4.11: Graphical representation of the full mass balance model

The full mass balance equations in section 4.4.2 may then be presented by the following state space model equation:

$$x(k+1) = Ax(k) + r(x,k) + Bx_i(k) + K_{La}(u_i)^T\left[S_{O,sat} - S_O(k)\right] \quad x(0) = x_0, \quad k = \overline{0, K-1} \qquad (4.163)$$

where A is the matrix with coefficients of the components concentrations; x is a vector of the components concentrations; r is the rate (per unit volume) at which components are produced or consumed; B is the matrix with coefficients of the input wastewater concentrations; x_i is a vector of the input wastewater concentrations; K_{La} is a vector of oxygen transfer functions; u_i is the vector of the air flow rate, s_o is the vector of the dissolved oxygen concentration; $s_{o,sat}$ is a vector of the dissolved oxygen saturation; k is the time instant and x_0 is the vector of initial conditions for the wastewater components.

From the Peterson matrix tables of the respective models,

$$r(x,k) = C^T P(x,k) \tag{4.164}$$

where C is a matrix of the yield coefficients and P is a vector of the process rates.

The state space model Equation (4.161) can then be written in the following way:

$$x(k+1) = Ax(k) + C^T P(x,k) + Bx_i(k) + K_{La}(u_i)^T \left[S_{o,sat} - S_o(k) \right] \ x(0) = x_0, \ k = \overline{0,K-1} \tag{4.165}$$

4.4.3.2 State space representation of the reduced mass balance models

The reduced mass balance model may be graphically presented as in Figure 4.12, where airflow rate is ignored and the dissolved oxygen concentration is now the control input. u is the dissolved oxygen control input in the model used for the optimal control problem formulation.
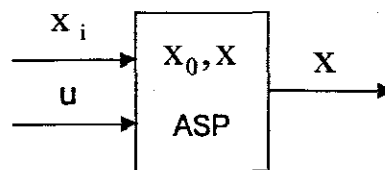
Figure 4.12: Graphical representation of the reduced mass balance model

For the reduced mass balance model, the state space model Equation (4.165) may be written in the following way:

$$x(k+1) = ARx(k) + CR^T PR(x,u,k) + BRx_i(k), \qquad x(0) = x_0, \qquad k = \overline{0,K-1} \tag{4.166}$$

where u is the vector of the dissolved oxygen concentration, AR is the matrix with coefficients of the components concentrations, BR is the matrix with coefficients of the

97

input wastewater concentrations, CR is a matrix of the yield coefficients and PR is a vector of the process rates.

4.4.3.3 Components of the state space mass balance model of the Athlone plant

From the discrete mass balance equations of the Athlone plant, the following vectors and matrices are deduced.

The vector for the process variables for the whole system is:

$$x = [x_1 \ x_2 \ x_3]^T$$

where x_n is a vector of the process variables (for each tank) as given in the Peterson matrix tables.

The vector for the process control variables for the whole system is:

$$u = [u_1 \ u_2 \ u_3]^T$$

where u_n is a control variable for each tank.

The vector for the inflow concentrations is:

$$x_i = [x_1]$$

where x_1 is a vector of the inflow concentration in the first tank.

The vector for the process rates for the whole system is:

$$P = [P_1 \ P_2 \ P_3]^T$$

where P_n is a vector of all process rates (for each tank) as given in the Peterson matrix.

The matrix c for the whole system is:

$$C = \Delta t \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix}$$

where C_n is a matrix of the stoichiometric parameters (for each tank) as given in the Peterson matrix and Δt is the sampling time.

The matrix A for the whole system (depends on the mass balance models) is:

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix}$$

where:

98

$$A_{11} = \left[\operatorname{diag}(1 - \frac{\Delta t}{V_1} Q_1) \right] \quad A_{12} = \left[\operatorname{diag}(\frac{\Delta t}{V_1} Q_P) \right] \quad A_{22} = \left[\operatorname{diag}(1 - \frac{\Delta t}{V_2} Q_2) \right] \quad A_{21} = \left[\operatorname{diag}(\frac{\Delta t}{V2} Q_1) \right] \quad A_{33} = \left[\operatorname{diag}(1 - \frac{\Delta t}{V_3} Q_3) \right]$$

$$A_{32} = \left[\operatorname{diag}(\frac{\Delta t}{V_3} Q_2) \right]$$

$$A_{23} = \left[\operatorname{diag}(\frac{\Delta t}{V2} (Q_a + Q_r)) \right] \text{ for the soluble materials}$$

$$A_{23} = \left[\operatorname{diag}(\frac{\Delta t}{V2} (Q_a + \lambda Q_r)) \right] \text{ for the particulate materials}$$

The matrix B for the whole system (depends on the number of influent wastewater components in the model) is:

$$B = \begin{bmatrix} B_1 & 0 & 0 \end{bmatrix}^T$$

$$B_1 = \left[\operatorname{diag}(\frac{Q_0}{V_1} \Delta t) \right]$$

Where Q_1 is the flow into Tank 1, Q_2 is the flow into Tank 2, Q_3 is the flow into Tank 3, V_1 is the volume of Tank 1, V_2 is the volume of Tank 2 and V_3 is the volume of Tank 3.

4.4.3.4 Components of the state space mass balance model of the Benchmark plant

From the discrete mass balance equations of the Benchmark plant, the following vectors and matrices are deduced.

The vector for the process variables for the whole system is:

$$x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix}^T$$

The vector for the process control variables for the whole system is:

$$u = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 \end{bmatrix}^T$$

The vector for the inflow concentrations for the whole system is:

$$x_i = \begin{bmatrix} x_1 \end{bmatrix}$$

The vector for the process rates for the whole system for the whole system is:

$$P = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 \end{bmatrix}^T$$

The matrix c for the whole system for the whole system is:

$$C = \Delta t \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 & 0 \\ 0 & 0 & C_3 & 0 & 0 \\ 0 & 0 & 0 & C_4 & 0 \\ 0 & 0 & 0 & 0 & C_5 \end{bmatrix}$$

The matrix A for the whole system (depends on the mass balance models) is:

$$A = \begin{bmatrix} A_{11} & 0 & 0 & 0 & A_{15} \\ A_{21} & A_{22} & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & 0 \\ 0 & 0 & 0 & A_{54} & A_{55} \end{bmatrix}$$

where:

$$A_{11} = \left[\text{diag}(1 - \frac{\Delta t}{V_1} Q_1) \right] \quad A_{22} = \left[\text{diag}(1 - \frac{\Delta t}{V_2} Q_2) \right] \quad A_{33} = \left[\text{diag}(1 - \frac{\Delta t}{V_3} Q_3) \right] \quad A_{44} = \left[\text{diag}(1 - \frac{\Delta t}{V_4} Q_4) \right]$$

$$A_{55} = \left[\text{diag}(1 - \frac{\Delta t}{V_5} Q_5) \right] \quad A_{21} = \left[\text{diag}(\frac{\Delta t}{V_2} Q_1) \right] \quad A_{32} = \left[\text{diag}(\frac{\Delta t}{V_3} Q_2) \right] \quad A_{43} = \left[\text{diag}(\frac{\Delta t}{V_4} Q_3) \right] \quad A_{54} = \left[\text{diag}(\frac{\Delta t}{V_5} Q_4) \right]$$

For the soluble materials

$$A_{15} = \left[\text{diag}\left\{ \frac{\Delta t}{V_1}(Q_a + Q_r) \right\} \right]$$

For the particulate materials

$$A_{15} = \left[\text{diag}\left\{ \frac{\Delta t}{V_1}(Q_a + \lambda Q_r) \right\} \right]$$

The matrix B for the whole system (depends on the number of influent wastewater components in the model) is:

$$B = \begin{bmatrix} B_1 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$B_1 = \left[\text{diag}(\frac{\Delta t}{V_1} Q_0) \right]$$

where Δt is the sampling time, Q_1 is the flow into Tank 1; Q_2 is the flow into Tank 2; Q_3 is the flow into Tank 3; Q_4 is the flow into Tank 4; Q_5 is the flow into Tank 5; V_1 is the volume of Tank 1; V_2 is the volume of Tank 2; V_3 is the volume of Tank 1; V_4 is the volume of Tank 4 and V_5 is the volume of Tank 5.

100

For the reduced models of both the Benchmark and Athlone plants, the term corresponding to s_{NO} in B_i is multiplied by a factor f. The matrices A, B, C and P are respectively denoted A_R, B_R, C_R and P_R in order to differentiate them from the full model matrices.

4.4.3.5 Summary of the vector and matrices of the mass balance models

The dimensions of the vector and the matrices of the mass balance models depend on the type of the biological models considered. Table 4.13 and Table 4.14 give summaries of the dimensions of all elements of each of the full and reduced mass balance models including the models' matrices.

Table 4.13: Dimensions of the various elements and matrices of the full models

DIMENSIONS OF THE VARIOUS ELEMENTS OF THE PLANTS MODELS					
Plants	Vectors / matrices	ASM1	ASM2	ASM2d	ASM3
Vector and matrices of the each tank					
BIOLOGICAL MODELS	x_n, x_i	R^{13}	R^{19}	R^{19}	R^{13}
	P_n	R^{8}	R^{19}	R^{21}	R^{12}
	A_n, B_i	$R^{13\times13}$	$R^{19\times19}$	$R^{19\times19}$	$R^{13\times13}$
	C_n	$R^{8\times13}$	$R^{19\times19}$	$R^{21\times19}$	$R^{12\times13}$
ATHLONE PLANT	x	R^{39}	R^{57}	R^{57}	R^{39}
	x_i	R^{13}	R^{19}	R^{19}	R^{13}
	P	R^{24}	R^{57}	R^{63}	R^{36}
	A	$R^{39\times39}$	$R^{57\times57}$	$R^{57\times57}$	$R^{39\times39}$
	C	$R^{24\times39}$	$R^{57\times57}$	$R^{63\times57}$	$R^{39\times36}$
	B	$R^{39\times13}$	$R^{57\times19}$	$R^{57\times19}$	$R^{39\times13}$
BENCHMARK PLANT	x	R^{65}	R^{95}	R^{95}	R^{65}
	x_i	R^{13}	R^{19}	R^{19}	R^{13}
	P	R^{40}	R^{95}	R^{105}	R^{60}
	A	$R^{65\times65}$	$R^{95\times95}$	$R^{95\times95}$	$R^{65\times65}$
	C	$R^{40\times65}$	$R^{95\times95}$	$R^{105\times95}$	$R^{60\times65}$
	B	$R^{65\times13}$	$R^{95\times19}$	$R^{95\times19}$	$R^{65\times13}$

Table 4.14: Dimensions of the various elements and matrices of the reduced models

DIMENSIONS OF THE VARIOUS ELEMENTS OF THE PLANTS MODELS					
Plants	Vectors / matrices	r-ASM1	r-ASM2	r-ASM2d	r-ASM3
Vector and matrices of the each tank					
BIOLOGICAL MODELS	x_n, x_i	R^3	R^6	R^6	R^3
	PR_n	R^4	R^{12}	R^{14}	R^4
	AR_n, BR	$R^{3\times3}$	$R^{8\times8}$	$R^{8\times8}$	$R^{3\times3}$
	CR_n	$R^{4\times3}$	$R^{12\times8}$	$R^{14\times8}$	$R^{4\times3}$
ATHLONE PLANT	x, x_i	R^9	R^{24}	R^{24}	R^9
	PR	R^{12}	R^{36}	R^{42}	R^{12}
	AR	$R^{9\times9}$	$R^{24\times24}$	$R^{24\times24}$	$R^{9\times9}$
	CR	$R^{12\times9}$	$R^{36\times24}$	$R^{42\times24}$	$R^{12\times9}$
	BR	$R^{9\times3}$	$R^{24\times8}$	$R^{24\times8}$	$R^{9\times3}$
BENCHMARK PLANT	x, x_i	R^{15}	R^{40}	R^{40}	R^{15}
	PR	R^{20}	R^{60}	R^{70}	R^{20}
	AR	$R^{15\times15}$	$R^{40\times40}$	$R^{40\times40}$	$R^{15\times15}$
	CR	$R^{20\times15}$	$R^{60\times40}$	$R^{70\times40}$	$R^{20\times15}$
	BR	$R^{15\times3}$	$R^{40\times8}$	$R^{40\times8}$	$R^{15\times3}$

4.5 Conclusion

In this section, the mathematical mass balance models of the Benchmark and Athlone wastewater treatment plants are derived. The biological activities within these plants are represented by the activated sludge models. The full mass balance models are reduced in order to acquire identifiable models with less number of state variables that are measurable. Finally both the full and reduced mass balance models are presented in discrete state space form which makes them easier to handle and apply in simulation. In the next chapter the mass balance models are implemented in the MATLAB simulation environment.

CHAPTER FIVE
PARAMETER ESTIMATION AND SIMULATIONS OF THE MODELS

5.1 Introduction

In this thesis, a method for optimization of the dissolved oxygen (DO) trajectory for the activated sludge processes (ASP) is developed. The aim is to calculate DO values sufficient for the ASP to produce effluent wastewater that meets the South African municipal standards at a lower cost. This project is part of the main research project at the Electrical Engineering Department of the Cape Peninsula University of Technology in South Africa. The main research project is aimed at developing a cost effective method for real time control of the Athlone and Benchmark wastewater treatment plants by manipulating the concentration of DO in the plants. This is achieved by regulating the airflow rate into the plants.

Tasks of the main project are distributed over a three layer hierarchical structure shown in Figure 5.1.
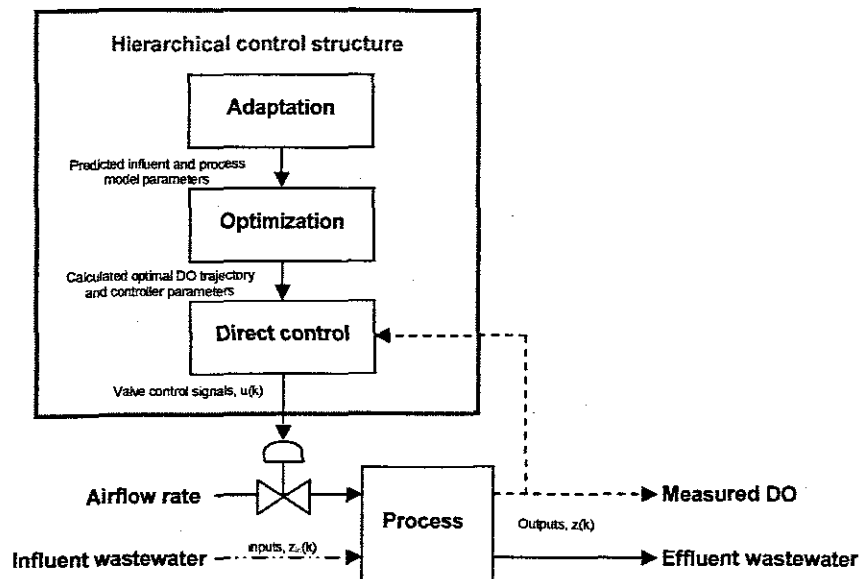
Figure 5.1: Control structure for the activated sludge process

In the adaptation layer, concentration of the influent wastewater is predicted based on historical influent concentration and data from the weather files. In the optimization layer, optimal DO trajectories are calculated based on predicted influent wastewater

concentrations as well as on the municipal limits for the effluent concentrations. The optimal DO trajectories are passed onto the layer of direct control where they are used as set points for the controller to track. These trajectories are used for calculation of the controller parameters and the control signals for the air regulating valve.

As stated preciously, the thesis focuses only on the DO trajectory optimization part of the main project. This part requires only the adaptation and optimization layers. A full description of the layer of direct control and the controller developed is found in Nketoane (2009). Sequence of tasks in the adaptation and optimization layers are broken down in Figure 5.2. Simulation and calculation tasks of the adaptation and optimization layers are carried on MATLAB programming environment.
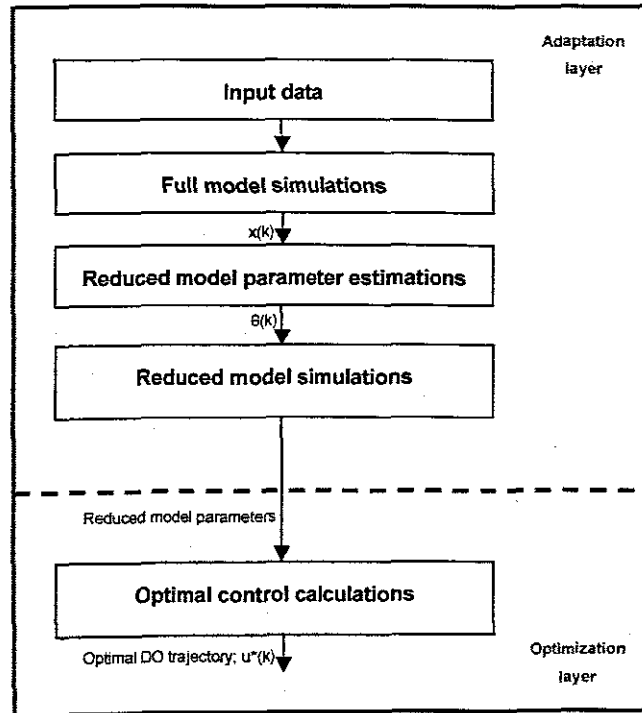
Figure 5.2: Sequence of tasks in the adaptation and optimization layers

The full and reduced mass balance models are developed in Chapter 4. In this chapter, simulation procedure for the mass balance models is provided. Simulation results are used to formulate and solve the optimal control problem for Benchmark and Athlone plants. The simulation procedure is the same for all the mass balance models. To avoid repetition, only the results of the Benchmark plant with ASM1 are shown.

The programs for simulations of the other mass balance models are provided on the Appendices. The other reason for this is because there has been a lot of different work done on the Benchmark plant with the ASM1 model, therefore a platform for comparison of the results is provided.

Calculations / simulations are performed as follows:

- The Benchmark and Athlone plants' layout are each implemented with the following biological models:
  - Activated Sludge Model No. 1 (ASM1)
  - Activated Sludge Model No. 2 (ASM2)
  - Activated Sludge Model No. 2d (ASM2d)
  - Activated Sludge Model No. 3 (ASM3)

5.2 Simulation procedure

The simulation procedure defined in the simulation benchmark (Copp, 2002), is used here. This is a two way step procedure involving simulations to steady state followed by dynamic simulations using one of the influent data files described in the simulation benchmark. Dynamic simulations follow a steady state simulation to ensure that the starting point is consistent and does not influence the generated dynamic output. Table 5.1 contains data common in all the simulation files for the control input of both plants.

Table 5.1: Data for simulations

Plants Oxygen components	Benchmark	Athlone
KLa - Tank 1	n/a	n/a
KLa - Tank 2	n/a	n/a
KLa - Tank 3	10 hr-1	3.5 hr-1
KLa - Tank 4	10 hr-1	n/a
KLa - Tank 5	3.5 hr-1	n/a
S O,sat - Tank 1	n/a	n/a
S O,sat - Tank 2	n/a	n/a
S O,sat - Tank 3	8 gO2m-3	8 gO2m-3
S O,sat - Tank 4	8 gO2m-3	n/a
S O,sat - Tank 5	8 gO2m-3	n/a
Delta Time (Sampling time)	0.01042	0.01042

5.2.1 Steady state simulations

In this step, steady states of all state variables are calculated for 100 days using the constant influent wastewater composition (average values in the dry influent file) shown in Table 5.2. Simulation is done for the following mass balance models:

- Benchmark plant with Activated Sludge Model No. 1 (ASM1)
- Benchmark plant with Activated Sludge Model No. 2 (ASM2)
- Benchmark plant with Activated Sludge Model No. 2d (ASM2d)
- Benchmark plant with Activated Sludge Model No. 3 (ASM3)
- Athlone plant with Activated Sludge Model No. 1 (ASM1)
- Athlone plant with Activated Sludge Model No. 2 (ASM2)
- Athlone plant with Activated Sludge Model No. 2d (ASM2d)
- Athlone plant with Activated Sludge Model No. 3 (ASM3)

Initial conditions for steady state simulations are provided in Table 5.3. The flow-weighted *dry weather* data (Table 5.2) is used for this purpose and steady state is defined by simulating 100 days using a constant influent.

Table 5.2: Constant influent composition for steady state simulations

Components	Values
SO	0.00
SNH	31.56
SNO	0.00
SI	30.00
SALK	7.00
SND	6.95
XI	51.20
XS	202.32
XBH	28.17
XBA	0.00
SS	69.50
XP	0.00
XND	10.59

The constant influent concentrations in Table 5.3 are average values of the dry weather influent file specified by the COST Working Group (Copp, 2000).

106

Table 5.3: Initial conditions for steady state simulation of the mass balance models

Components \ Tank	1	2	3	4	5
SI	15	15	15	15	15
SS	1.4	0.7	0.6	0.5	0.4
XI	574.6	574.6	574.6	574.6	574.6
XS	41.1	38.2	32.4	27.8	24.7
XBH	1275.9	1276.7	1278.6	1279.6	1279.7
XBA	74.2	74.2	74.5	74.8	74.9
XP	224.4	224.8	225.2	225.7	226.1
SO	4e-4.		0.9	1.2	0.2
SNO	2.7	1.8	3.3	4.6	5.2
SNH	4.0	4.2	2.8	1.5	0.9
SND	0.6	0.4	0.4	0.4	0.3
XND	2.6	2.5	2.2	1.9.	1.8
SALK	2.5	2.5	2.3	2.1	2.1

5.2.2 Dynamic simulations

In this section, the obtained steady state results are used as initial conditions for dynamic simulation of the plants using the dry weather file (Figure 5.3), the concentrations not included in the file are assumed to be zero. Calculations are done for 14 days for the same mass balance models in point 2 above.
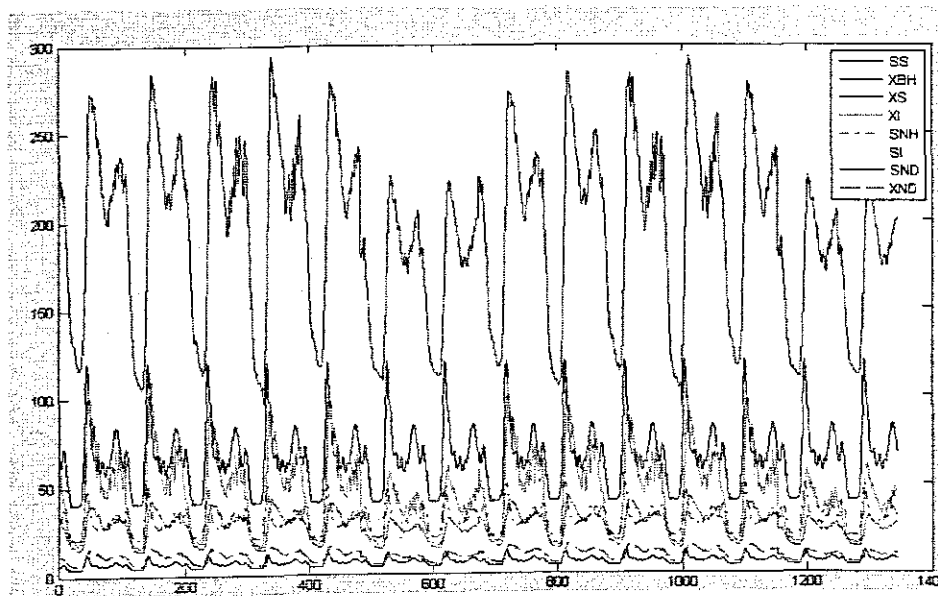
Figure 5.3: Trajectories of the influent wastewater concentrations from dry weather file

107

The plants are implemented with the reduced ASM1, ASM2, ASM2d and ASM3 biological models. The results from dynamic simulations are used to formulate the structure of the reduced models.

The output data generated from these simulations are used to build and calibrate the reduced models.

5.3 Simulation of the full mass balance models

The models derived in Chapter 4 are used in this section for simulation of the behaviour of the real Athlone and Benchmark plants. Simulation is based on the full state space model Equation (4.163);

$$x(k+1) = Ax(k) + C^T P(x,k) + Bx_i(k) + K_{La}^T [S_{O,sat} \quad S_O(k)] \qquad x(0) = x_0$$

Figure 5.4 shows the flow chart with the procedure on how the full mass balance models are implemented in MATLAB.

Firstly, values of the influent wastewater composition, typical model parameter values and plant data are introduced. Then the matrices of the models (derived in section 4.3) are calculated. At each instance k from $k=0$ till $k=K-1$, the process rates (k) are calculated followed by calculations of the mass balance model outputs. K is the final time of the prediction horizon. Finally simulation results are plotted. These results may be used to examine the dynamic performance of the process.
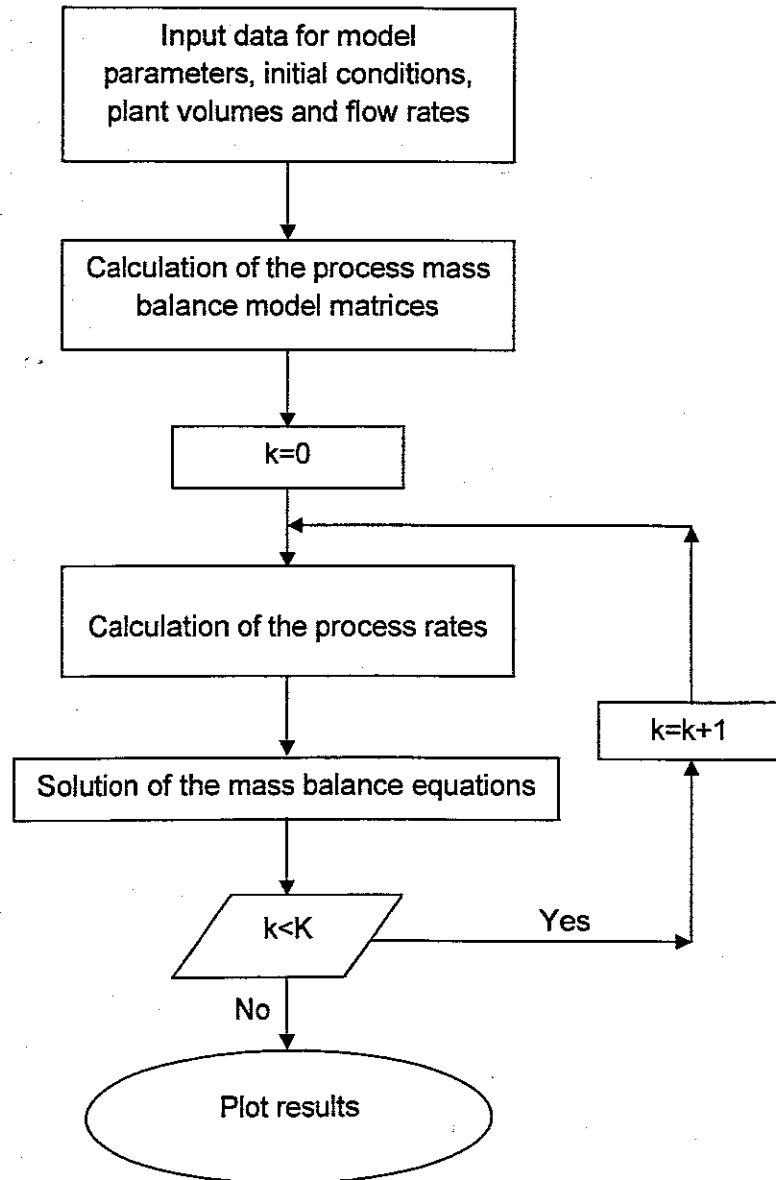
Figure 5.4: Flow chart for simulation of the full mass balance models

Values of the influent wastewater compounds and the model parameters are adapted from the IWA technical report (Henze et al., 2000). The influent wastewater concentrations are obtained from a 14 days dry weather file. The data from the dry weather file is supplied at fifteen minutes intervals.

A fifteen minute interval corresponds to a sampling period of 0.01042. Steady state simulations are ran over a 100 days period and dynamic simulations are ran over 14 days.

Therefore the prediction horizon (K) is 9597 for steady state and 1344 for dynamic simulations.

Calculations in MATLAB gives better results when a smaller sampling period is used. For this reason a sampling period of 0.001042 is chosen. With this sampling period the prediction horizon (K) becomes 95969 for steady state simulations and 95969 for dynamic simulations. Since only 1344 samples are provided on a 14 day dry weather file a MATLAB function called spline is used for interpolation between the existing points of the data. New 10 sampling points then are taken between every 2 points of the initial data.

The Athlone plant data used in simulations are presented in Table 4.6 while the Benchmark plant data are in Table 4.10. Parameter values for the full ASM1, ASM2, ASM2d and ASM3 models are presented in Chapter 4; see Table 4.1 till Table 4.3.

Table 5.4 presents the results from steady state simulations of the full mass balance models. These results are then used as initial conditions for dynamic simulations.

Table 5.4: Initial conditions for dynamic simulation of the mass balance models

Tank / Components	1	2	3	4	5
SO	0.4442	0.6052	2.4577	4.1315	2.7336
SNH	4.9639	3.5994	1.4408	0.4273	0.1634
SNO	26.0224	27.3786	29.7912	31.0889	31.6029
SI	30	30	30	30	30
SALK	3.2415	3.0472	2.7207	2.5556	2.50
SND	1.2434	0.9498	0.7914	0.7031	0.6414
XI	1252	1252	1252	1252	1252
XS	74.9448	66.6212	57.2967	50.0176	44.5204
XBH	2680	2683	2685	2685	2685
XA	166.2464	166.5816	167.0738	167.2829	167.3101
SS	2.2766	1.1926	0.9806	0.8626	0.7743
XP	514.0746	514.7800	515.7209	516.6621	517.6030
XND	4.8343	4.4095	3.9169	3.5256	3.5256

5.3.1 Simulation data for the Athlone plant with ASM1 model

The MATLAB file "AASM1_Simulation.m" given in Appendix E.1 is used for simulation of the Athlone plant with ASM1.

From the full mass balance equations of the Athlone plant with ASM1, the following vectors and matrices are deduced and used in simulations. The vector for the system states of the **n-th** tanks is:

$$x_n = [S_{In} \quad S_{Sn} \quad X_{In} \quad X_{Sn} \quad X_{BHn} \quad X_{BAn} \quad X_{Pn} \quad S_{On} \quad S_{NOn} \quad S_{NHn} \quad S_{NDn} \quad X_{NDn} \quad S_{ALKn}]^T \in R^{13} \qquad \overline{n=1,3}$$

$$x = [x_1 \quad x_2 \quad x_3]^T \in R^{39}$$

The vector for the input concentration is:

$$x_i = [S_{Ii} \quad S_{Si} \quad X_{Ii} \quad X_{Si} \quad X_{BHi} \quad X_{BAi} \quad X_{Pi} \quad S_{Oi} \quad S_{NOi} \quad S_{NHi} \quad S_{NDi} \quad X_{NDi} \quad S_{ALKi}]^T \in R^{13}$$

The vector for the system dissolved oxygen is:

$$S_O = [0 \quad . . \quad 0 \quad S_{O1} \quad 0 \quad . . \quad 0 \quad 0 \quad . . \quad 0 \quad S_{O2} \quad 0 \quad . . \quad 0 \quad 0 \quad . . \quad 0 \quad S_{O3} \quad 0 \quad . . \quad 0]^T \in R^{39}$$

The vector for the system oxygen transfer function is:

$$K_La = [0 \quad . . \quad 0 \quad K_La \quad 0 \quad . . \quad 0 \quad 0 \quad . . \quad 0 \quad K_La \quad 0 \quad . . \quad 0 \quad 0 \quad . . \quad 0 \quad K_La \quad 0 \quad . . \quad 0]^T \in R^{39}$$

The vector for the system process rates for the **n-th** tanks is:

$$P_n = [p_{1n} \quad p_{2n} \quad p_{3n} \quad p_{4n} \quad p_{5n} \quad p_{6n} \quad p_{7n} \quad p_{8n}]^T \in R^8$$

$$P = [P_1 \quad P_2 \quad P_3]^T \in R^{24}$$

The matrix C for the whole system is:

$$C = \Delta t \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix} \in R^{24 \times 39}$$

where:

$$C_n \in R^{8 \times 13} \qquad \overline{n=1,3}$$

The matrix A for the whole system is:

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix} \in R^{39 \times 39}$$

The matrix B for the whole system is:

$$B = [B_1 \quad 0 \quad 0]^T \in R^{39 \times 13}$$

where:

$$B_1 \in R^{13 \times 13}$$

111

5.3.2 Simulation data for the Athlone plant with ASM2 model

The MATLAB file "*AASM2_Simulation.m*" given in Appendix F.1 is used for simulation of the Athlone plant with ASM2.

From the full mass balance equations of the Athlone plant with ASM2, the following vectors and matrices are deduced.

The vector for the system states for the **n-th** tank is:

$$x_n = [S_{O2n} \ S_{Fn} \ S_{An} \ S_{NH4n} \ S_{NO3n} \ S_{PO4n} \ S_{In} \ S_{ALKn} \ S_{N2n} \ X_{In} \ X_{Sn} \ X_{Hn} \ X_{PAOn} \ X_{PPn} \ X_{PHAn} \ X_{AUTn} \ X_{TSSn} \ X_{MeOHn} \ X_{MePn}]^T$$

$$x_n \in R^{19} \quad \overline{n = 1,3}$$

$$x = [x_1 \ x_2 \ x_3]^T \in R^{57}$$

The vector for the input concentration is:

$$x_i = [S_{O2i} \ S_{Fi} \ S_{Ai} \ S_{NH4i} \ S_{NO3i} \ S_{PO4i} \ S_{Ii} \ S_{ALKi} \ S_{N2i} \ X_{Ii} \ X_{Si} \ X_{Hi} \ X_{PAOi} \ X_{PPi} \ X_{PHAi} \ X_{AUTi} \ X_{TSSi} \ X_{MeOHi} \ X_{MePi}]^T \in R^{19}$$

The vector for the system dissolved oxygen is:

$$S_O = [0 \ . \ . \ 0 \ S_{O1} \ 0 \ . \ . \ 0 \ 0 \ . \ . \ 0 \ S_{O2} \ 0 \ . \ . \ 0 \ 0 \ . \ . \ 0 \ S_{O3} \ 0 \ . \ . \ 0]^T \in R^{57}$$

The vector for the system oxygen transfer function is:

$$K_La(u) = [0 \ . \ . \ 0 \ K_La(u_1) \ 0 \ . \ . \ 0 \ 0 \ . \ . \ 0 \ K_La(u_2) \ 0 \ . \ . \ 0 \ 0 \ . \ . \ 0 \ K_La(u_3) \ 0 \ . \ . \ 0]^T \in R^{57}$$

The vector for the system process rates for the **n-th** tank is:

$$P_n = [\rho_{1n} \ \rho_{2n} \ \rho_{3n} \ \rho_{4n} \ \rho_{5n} \ \rho_{6n} \ \rho_{7n} \ \rho_{8n} \ \rho_{9n} \ \rho_{10n} \ \rho_{11n} \ \rho_{12n} \ \rho_{13n} \ \rho_{14n} \ \rho_{15n} \ \rho_{16n} \ \rho_{17n} \ \rho_{18n} \ \rho_{19n}]^T \in R^{19}$$

$$P = [P_1 \ P_2 \ P_3]^T \in R^{57}$$

The matrix c for the whole system is:

$$C = \Delta t \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix} \in R^{57 \times 57}$$

where:

$$C_n \in R^{19 \times 19} \quad \overline{n = 1,3}$$

The matrix A for the whole system is:

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix} \in R^{57 \times 57}$$

The matrix B for the whole system is:

$$B = [B_1 \ 0 \ 0]^T \in R^{57 \times 19}$$

where: $B_1 \in R^{19 \times 19}$

112

5.3.3 Simulation data for the Athlone plant ASM2d model

The MATLAB file "*AASM2d_Simulation.m*" given in Appendix G.1 is used for simulation of the Athlone plant with ASM2d.

From the full mass balance equations of the Athlone plant with ASM2d, the following vectors and matrices are deduced.

The vector for the system states for the n-th tank is:

$$x_n = \begin{bmatrix} S_{O2n} & S_{Fn} & S_{An} & S_{NH4n} & S_{NO3n} & S_{PO4n} & S_{In} & S_{ALKn} & S_{N2n} & X_{In} & X_{Sn} & X_{Hn} & X_{PAOn} & X_{PPn} & X_{PHAn} & X_{AUTn} & X_{TSSn} & X_{MeOHn} & X_{MePn} \end{bmatrix}^T$$

$$x_n \in R^{19} \qquad \overline{n = 1,3}$$

$$x = [x_1 \quad x_2 \quad x_3]^T \in R^{57}$$

The vector for the input concentration is:

$$x_i = \begin{bmatrix} S_{O2i} & S_{Fi} & S_{Ai} & S_{NH4i} & S_{NO3i} & S_{PO4i} & S_{Ii} & S_{ALKi} & S_{N2i} & X_{Ii} & X_{Si} & X_{Hi} & X_{PAOi} & X_{PPi} & X_{PHAi} & X_{AUTi} & X_{TSSi} & X_{MeOHi} & X_{MePi} \end{bmatrix}^T \in R^{19}$$

The vector for the system dissolved oxygen is:

$$S_O = \begin{bmatrix} 0 & .. & 0 & S_{o1} & 0 & .. & 0 & 0 & .. & 0 & S_{o2} & 0 & .. & 0 & 0 & .. & 0 & S_{o3} & 0 & .. & 0 \end{bmatrix}^T \in R^{57}$$

The vector for the system oxygen transfer function is:

$$K_La(u) = \begin{bmatrix} 0 & .. & 0 & K_La(u_1) & 0 & .. & 0 & 0 & .. & 0 & K_La(u_2) & 0 & .. & 0 & 0 & .. & 0 & K_La(u_3) & 0 & .. & 0 \end{bmatrix}^T \in R^{57}$$

The vector for the system process rates for the n-th tank is:

$$P_n = \begin{bmatrix} P_{1n} & P_{2n} & P_{3n} & P_{4n} & P_{5n} & P_{6n} & P_{7n} & P_{8n} & P_{9n} & P_{10n} & P_{11n} & P_{12n} & P_{13n} & P_{14n} & P_{15n} & P_{16n} & P_{17n} & P_{18n} & P_{19n} \end{bmatrix}^T \in R^{21}$$

$$P = \begin{bmatrix} P_1 & P_2 & P_3 \end{bmatrix}^T \in R^{63}$$

The matrix c for the whole system is:

$$C = \Delta t \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix} \in R^{63 \times 57}$$

where:

$$C_n \in R^{21 \times 19} \quad \overline{n = 1,3}$$

The matrix A for the whole system is:

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix} \in R^{57 \times 57}$$

The matrix B for the whole system is:

$$B = \begin{bmatrix} B_1 & 0 & 0 \end{bmatrix}^T$$

where:

$$B_1 \in R^{19 \times 19}$$

5.3.4 Simulation data for the Athlone plant with ASM3 model

The MATLAB file "*AASM3_Simulation.m*" given in Appendix H.1 is used for simulation of the Athlone plant with ASM3.

From the full mass balance equations of the Athlone plant with ASM3, the following vectors and matrices are deduced.

The vector for the system states for the n-th tank is:

$$x_n = [S_{O2n} \quad S_{In} \quad S_{Sn} \quad S_{NH4n} \quad S_{N2n} \quad S_{NOXn} \quad S_{ALKn} \quad X_{In} \quad X_{Sn} \quad X_{Hn} \quad X_{STOn} \quad X_{An} \quad X_{SSn}]^T \in R^{13} \qquad \overline{n = 1,3}$$

$$x = [x_1 \quad x_2 \quad x_3]^T \in R^{39}$$

The vector for the input concentration is:

$$x_i = [S_{O2i} \quad S_{Ii} \quad S_{Si} \quad S_{NH4i} \quad S_{N2i} \quad S_{NOXi} \quad S_{ALKi} \quad X_{Ii} \quad X_{Si} \quad X_{Hi} \quad X_{STOi} \quad X_{Ai} \quad X_{SSi}]^T \in R^{13}$$

The vector for the system dissolved oxygen is:

$$S_O = [0 \, . \, . \, 0 \; S_{O1} \; 0 \, . \, . \, 0 \; 0 \, . \, . \, 0 \; S_{O2} \; 0 \, . \, . \, 0 \; 0 \, . \, . \, 0 \; S_{O3} \; 0 \, . \, . \, 0]^T \in R^{39}$$

The vector for the system oxygen transfer function is:

$$K_L a(u) = [0 \, . \, . \, 0 \; K_L a(u_1) \; 0 \, . \, . \, 0 \; 0 \, . \, . \, 0 \; K_L a(u_2) \; 0 \, . \, . \, 0 \; 0 \, . \, . \, 0 \; K_L a(u_3) \; 0 \, . \, . \, 0]^T \in R^{39}$$

The vector for the system process rates for the n-th tank is:

$$P_n = [p_{1n} \quad p_{2n} \quad p_{3n} \quad p_{4n} \quad p_{5n} \quad p_{6n} \quad p_{7n} \quad p_{8n} \quad p_{9n} \quad p_{10n} \quad p_{11n} \quad p_{12n}]^T \in R^{12}$$

$$P = [P_1 \quad P_2 \quad P_3]^T \in R^{36}$$

The matrix c for the whole system is:

$$C = \Delta t \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix} \in R^{36 \times 39}$$

where:

$$C_n \in R^{12 \times 13} \quad \overline{n = 1,3}$$

The matrix A for the whole system is:

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix} \in R^{39 \times 39}$$

The matrix B for the whole system is:

$$B = [B_1 \quad 0 \quad 0]^T \in R^{39 \times 13}$$

where:

$$B_1 \in R^{13 \times 13}$$

114

5.3.5 Simulation data and results of the Benchmark plant with ASM1 model

The MATLAB file "BASM1_Simulation.m" given in Appendix A.1 is used for simulation of the Benchmark plant with ASM1. From the full mass balance equations of the Benchmark plant with ASM1, the following vectors and matrices are deduced.

The vector for the system states for the n-th tank is:

$$x_n = [S_{In} \quad S_{Sn} \quad X_{In} \quad X_{Sn} \quad X_{BHn} \quad X_{BAn} \quad X_{Pn} \quad S_{On} \quad S_{NOn} \quad S_{NHn} \quad S_{NDn} \quad X_{NDn} \quad S_{ALKn}]^T \in R^{13} \quad \overline{n = 1,5}$$

$$x = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5]^T \in R^{65}$$

The vector for the input concentration is:

$$x_i = [S_{Ii} \quad S_{Si} \quad X_{Ii} \quad X_{Si} \quad X_{BHi} \quad X_{BAi} \quad X_{Pi} \quad S_{Oi} \quad S_{NOi} \quad S_{NHi} \quad S_{NDi} \quad X_{NDi} \quad S_{ALKi}]^T \in R^{13}$$

The vector for the system dissolved oxygen is:

$$S_O = [0 \ .. \ 0 \ S_{O1} \ 0 \ .. \ 0 \ 0 \ .. \ 0 \ S_{O2} \ 0 \ .. \ 0 \ 0 \ .. \ 0 \ S_{O3} \ 0 \ .. \ 0]^T \in R^{65}$$

The vector for the system oxygen transfer function is:

$$K_La(u) = [0 \ .. \ 0 \ K_La(u_1) \ 0 \ .. \ 0 \ 0 \ .. \ 0 \ K_La(u_2) \ 0 \ .. \ 0 \ 0 \ .. \ 0 \ K_La(u_5) \ 0 \ .. \ 0]^T \in R^{65}$$

The vector for the system process rates for the n-th tank is:

$$P_n = [p_{1n} \quad p_{2n} \quad p_{3n} \quad p_{4n} \quad p_{5n} \quad p_{6n} \quad p_{7n} \quad p_{8n}]^T \in R^8$$

$$P = [P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5]^T \in R^{40}$$

The matrix c for the whole system is:

$$C = \Delta t \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 & 0 \\ 0 & 0 & C_3 & 0 & 0 \\ 0 & 0 & 0 & C_4 & 0 \\ 0 & 0 & 0 & 0 & C_5 \end{bmatrix} \in R^{40 \times 65}$$

where:

$$C_n = \begin{bmatrix} 0 & -1.4925 & 0 & 0 & 1 & 0 & 0 & -0.4925 & 0 & -0.08 & 0 & 0 & -0.0057 \\ 0 & -1.4925 & 0 & 0 & 1 & 0 & 0 & 0 & -0.1722 & -0.08 & 0 & 0 & 0.0066 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -18.0417 & 4.1667 & -4.2467 & 0 & 0 & -0.6010 \\ 0 & 0 & 0 & 0.92 & -1 & 0 & 0.08 & 0 & 0 & 0 & 0 & 0.0752 & 0 \\ 0 & 0 & 0 & 0.92 & 0 & -1 & 0.08 & 0 & 0 & 0 & 0 & 0.0752 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0.0714 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \end{bmatrix} \in R^{8 \times 13} \quad \overline{n = 1,5}$$

The matrix A for the whole system is:

$$A = \begin{bmatrix} A_{11} & 0 & 0 & 0 & A_{15} \\ A_{21} & A_{22} & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & 0 \\ 0 & 0 & 0 & A_{54} & A_{55} \end{bmatrix} \in R^{65 \times 65}$$

where:

115

$$A_{11} = \begin{bmatrix}
0.9039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.9039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.9039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.9039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.9039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.9039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.9039 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9039 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9039 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9039 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9039 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9039 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9039
\end{bmatrix} \in R^{13 \times 13}$$

$$A_{15} = \begin{bmatrix}
0.0769 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.0769 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.0953 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.0953 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.0953 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.0953 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0953 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0769 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0769 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0769 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0769 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0953 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0769
\end{bmatrix} \in R^{13 \times 13}$$

$$A_{21} = \begin{bmatrix}
0.0961 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.0961 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.0961 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.0961 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.0961 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.0961 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0961 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0961 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0961 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0961 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0961 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0961 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0961
\end{bmatrix} \in R^{13 \times 13}$$

$$A_{22} = A_{11} \in R^{13 \times 13}$$

$$A_{32} = \begin{bmatrix}
0.0721 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.0721 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.0721 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.0721 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0721 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0721
\end{bmatrix} \in R^{13 \times 13}$$

$$A_{33} = \begin{bmatrix} 0.9279 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9279 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9279 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9279 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9279 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9279 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.9279 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9279 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9279 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9279 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9279 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9279 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9279 \end{bmatrix} \in R^{13 \times 13}$$

$$A_{43} = A_{54} = A_{32} \in R^{13 \times 13}$$

$$A_{55} = A_{44} = A_{33} \in R^{13 \times 13}$$

The matrix B for the whole system is:

$$B = [B_1 \quad 0 \quad 0 \quad 0 \quad 0]^T \in R^{65 \times 13}$$

where:

$$B_1 = \begin{bmatrix} 0.0192 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0192 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0192 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0192 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0192 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0192 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0192 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0192 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0192 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0192 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0192 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0192 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0192 \end{bmatrix} \in R^{13 \times 13}$$

The steady state and dynamic simulation results of the wastewater components concentrations for the Benchmark plants' five tanks are shown in Figures (5.5 – 5.14).
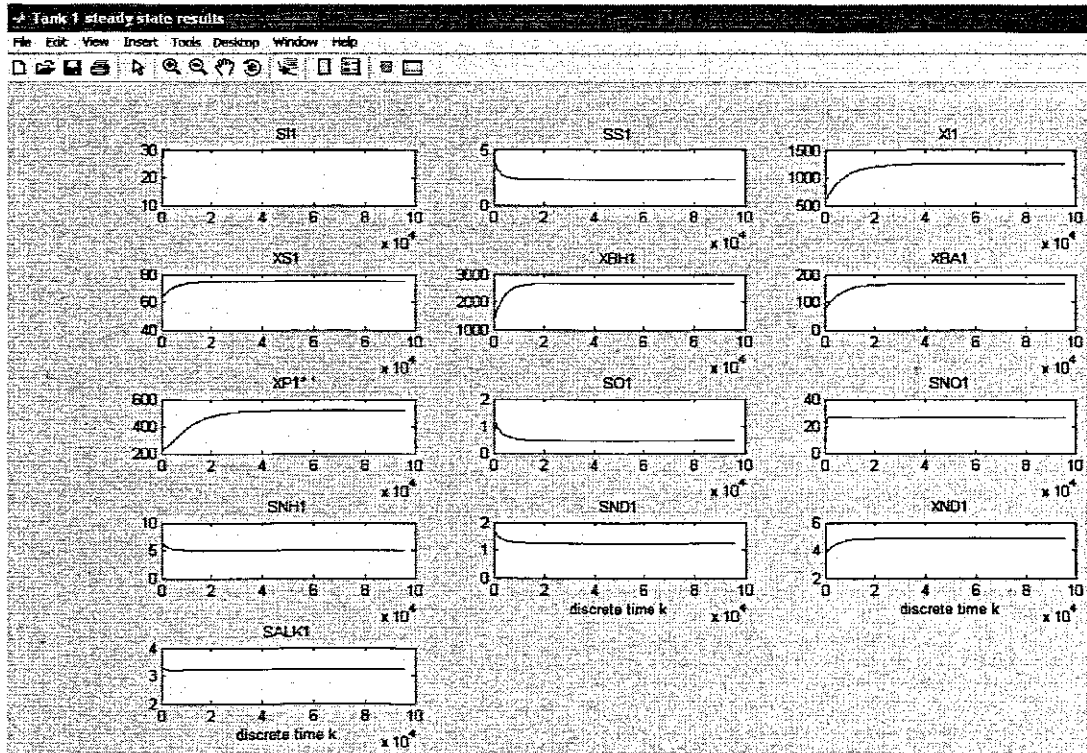
Figure 5.5: Steady state results for the process variables in Tank 1
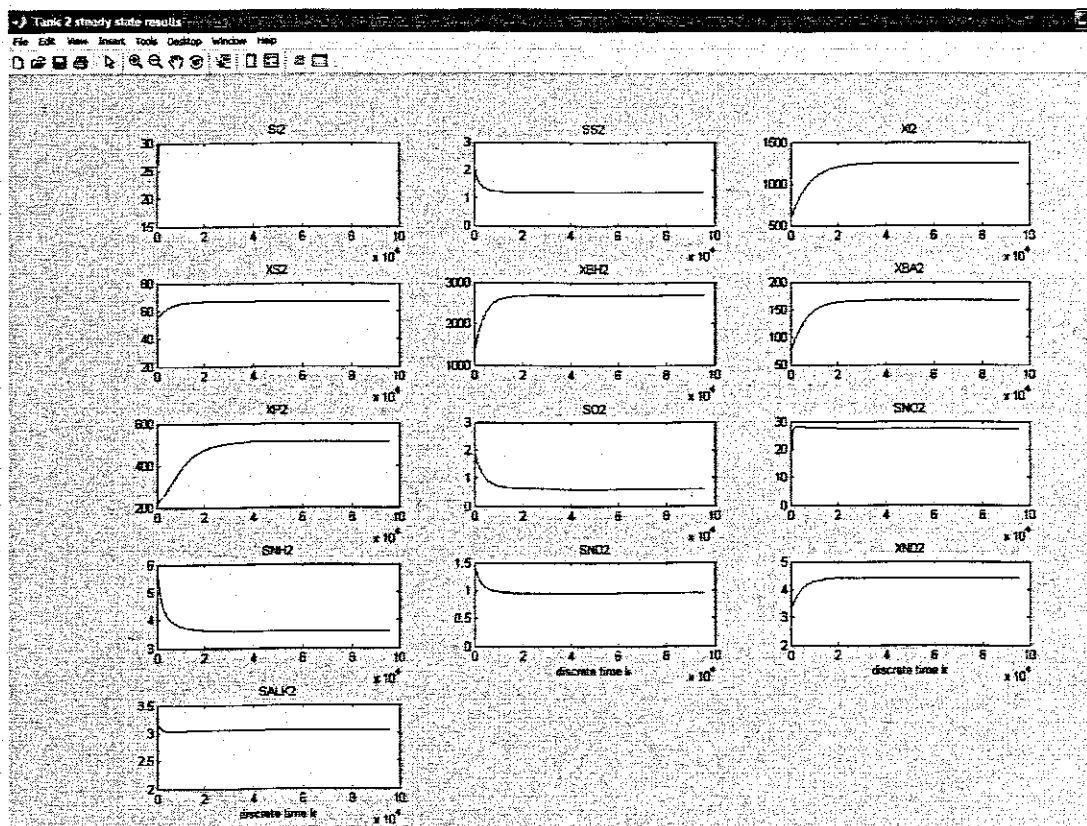
Figure 5.6: Steady state results for the process variables in Tank 2

118
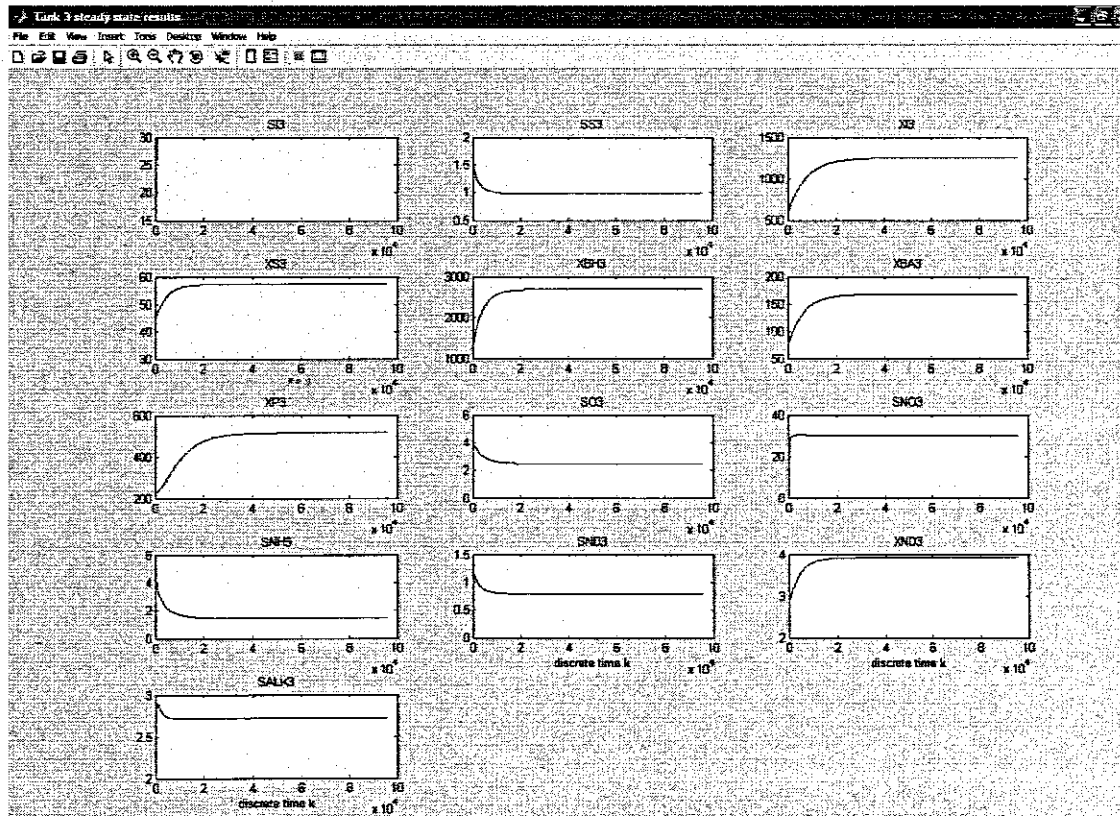
Figure 5.7: Steady state results for the process variables in Tank 3
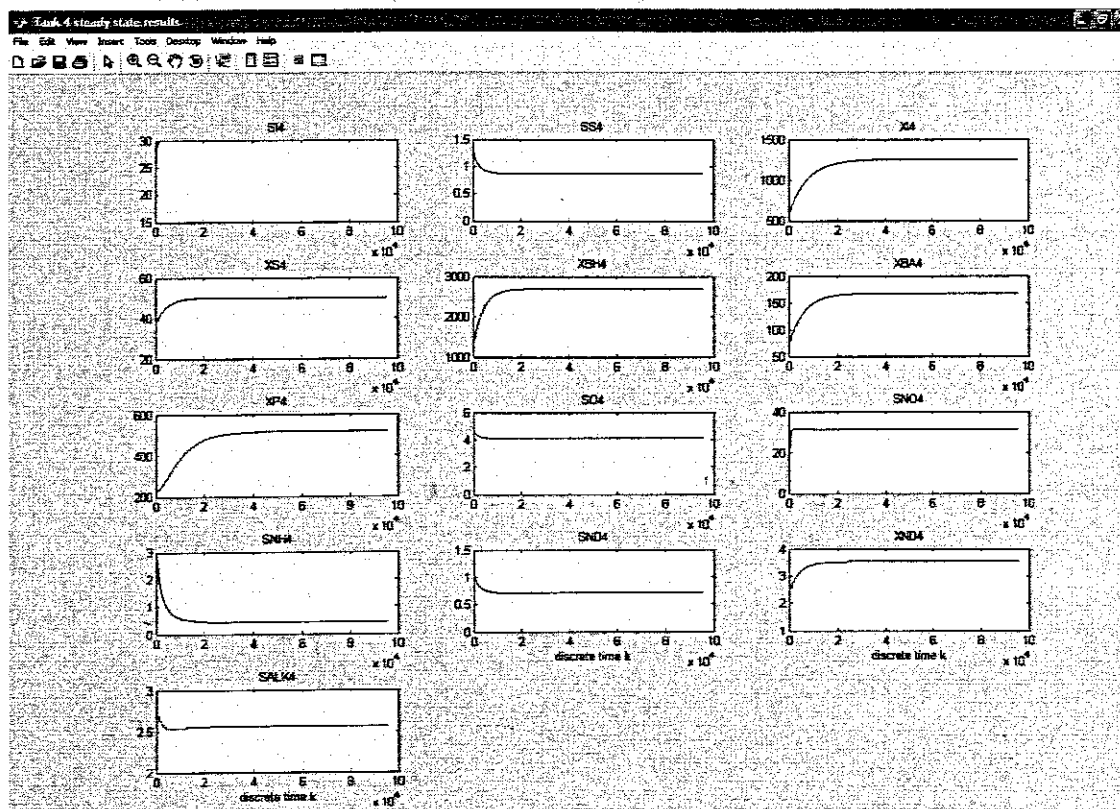
Figure 5.8: Steady state results for the process variables in Tank 4

119
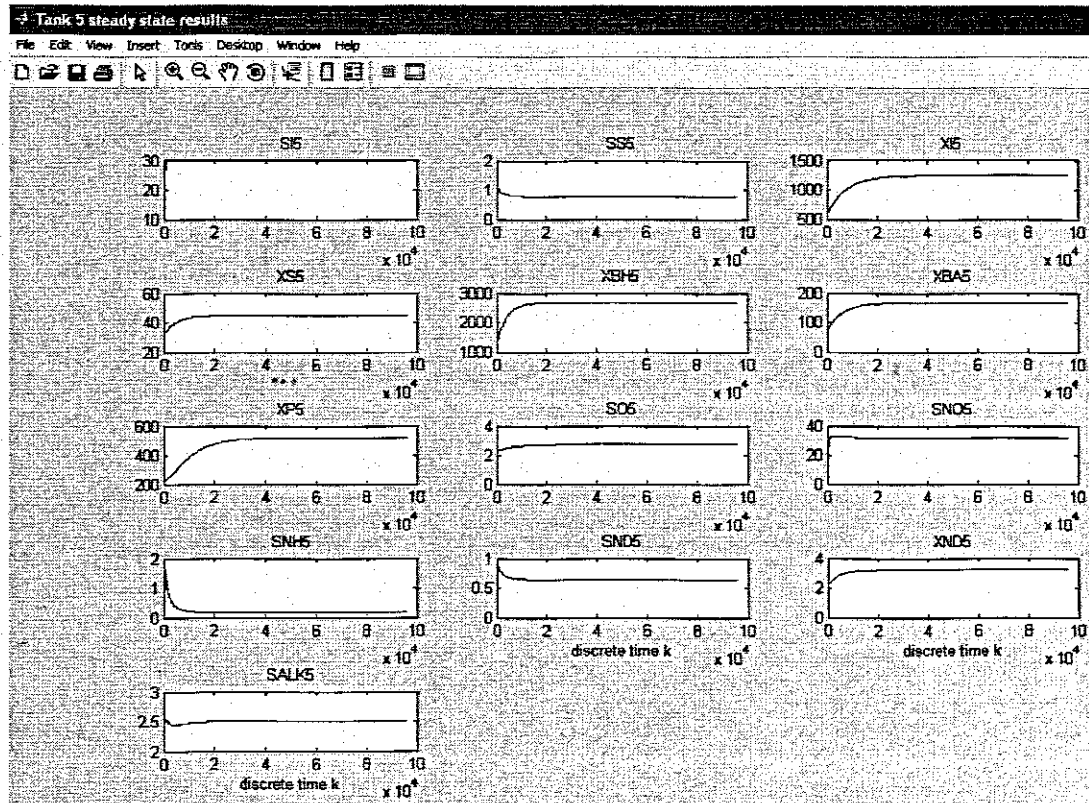
Figure 5.9: Steady state results for the process variables in Tank 5
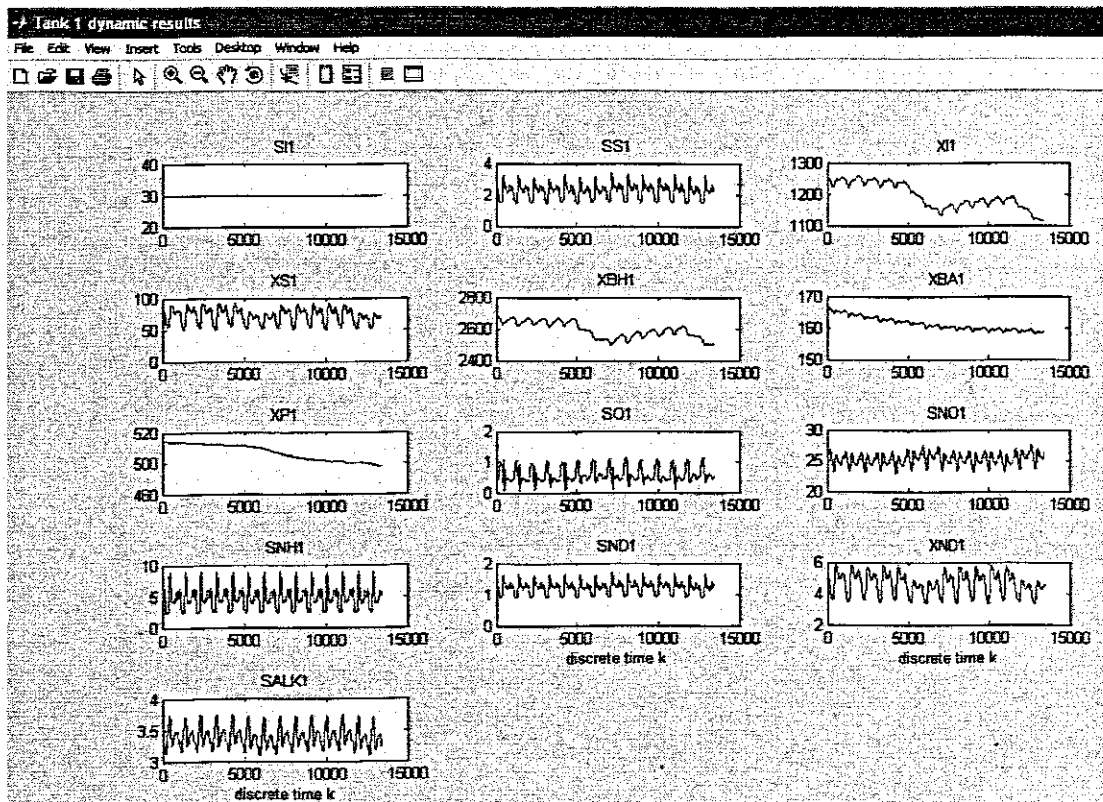
Figure 5.10: Dynamic results for the process variables in Tank 1

120
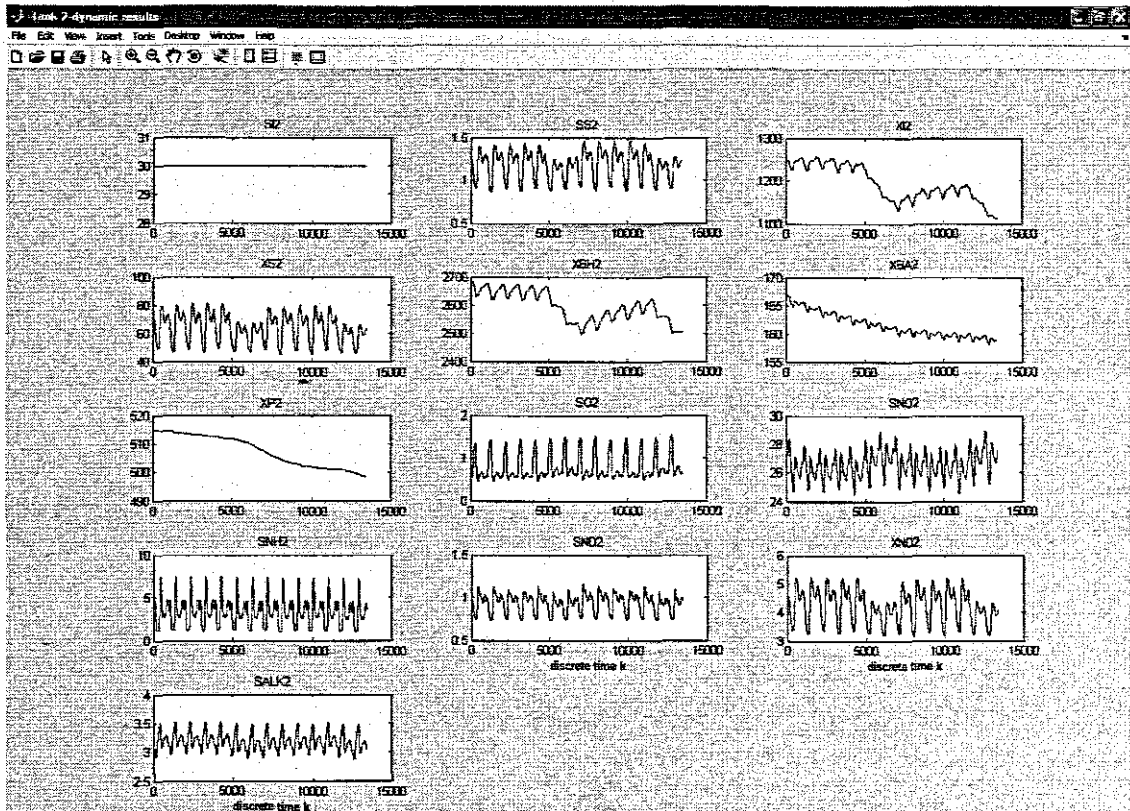
Figure 5.11: Dynamic results for the process variables in Tank 2
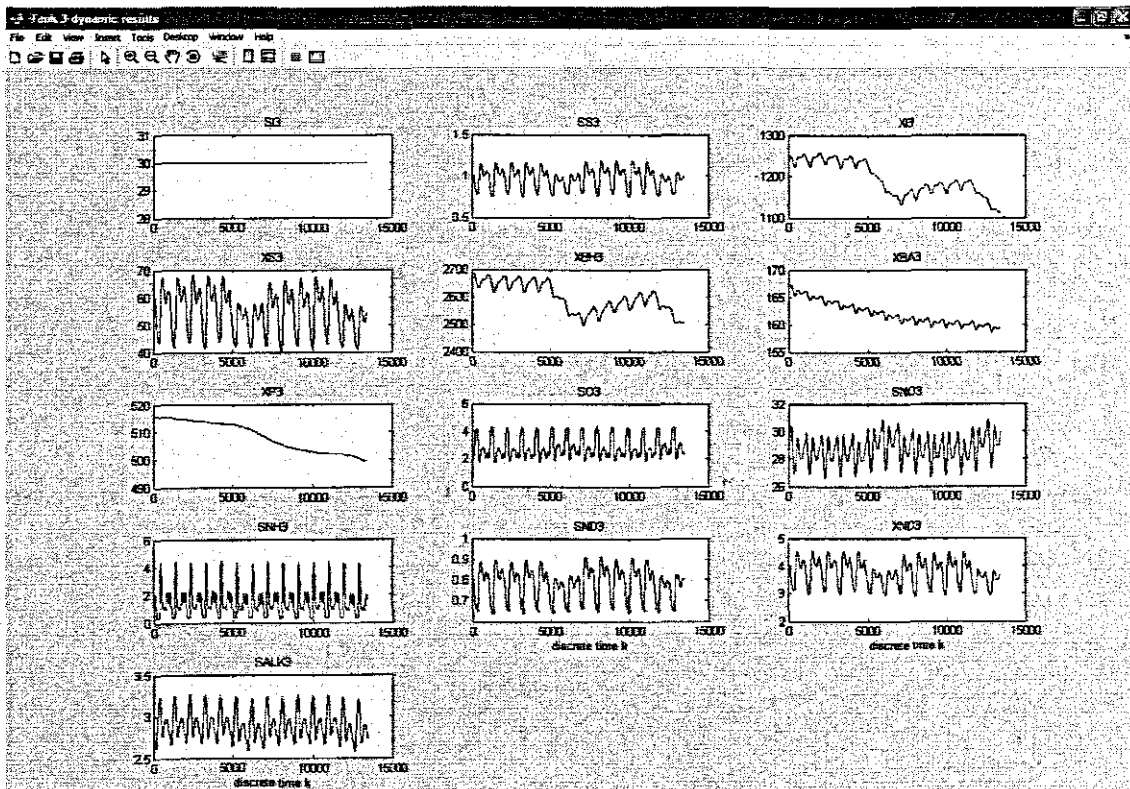
Figure 5.12: Dynamic results for the process variables in Tank 3

121
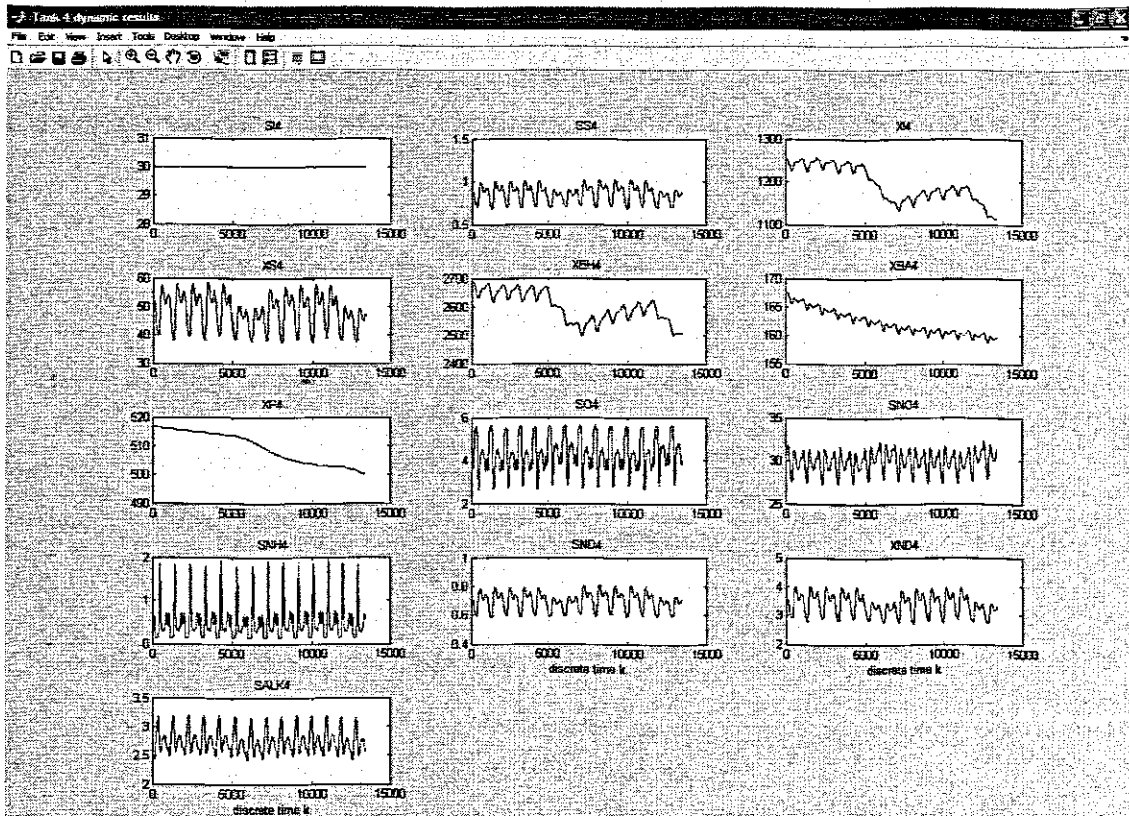
Figure 5.13: Dynamic results for the process variables in Tank 4
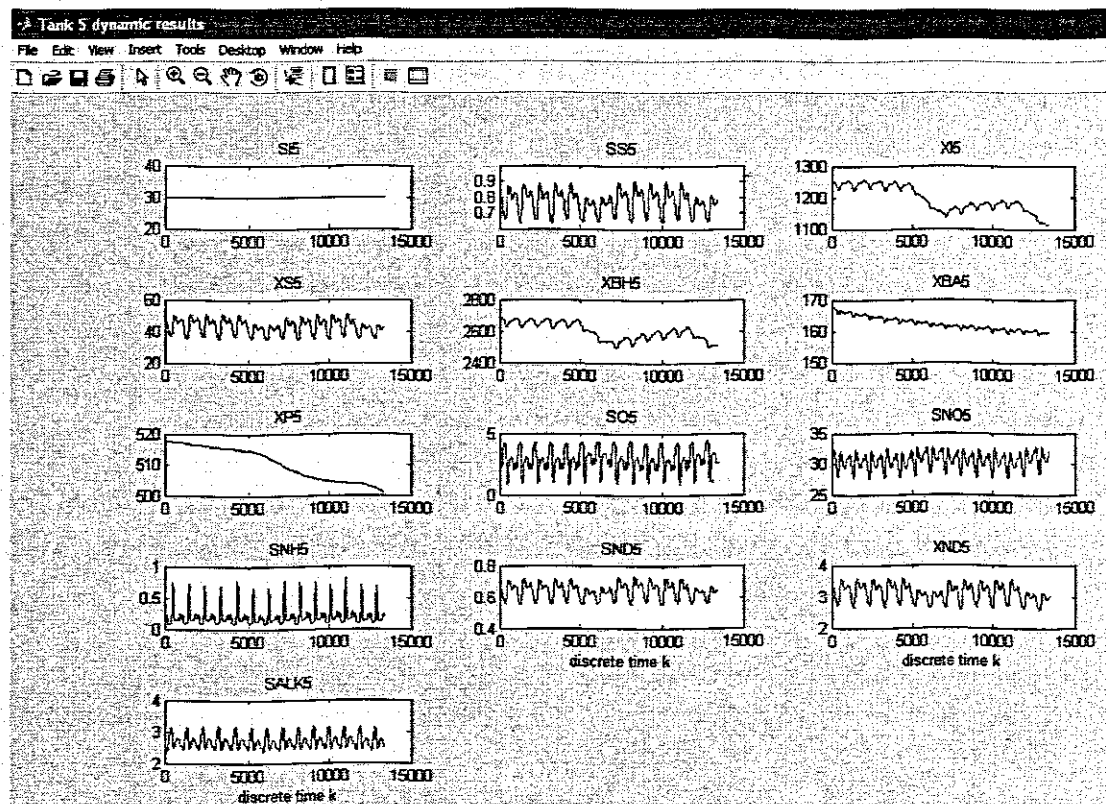
Figure 5.14: Dynamic results for the process variables in Tank 5

122

5.3.6 Simulation data for the Benchmark plant with ASM2 model

The MATLAB file "*BASM2_Simulation.m*" given in Appendix B.1 is used for simulation of the Benchmark plant with ASM2.

From the full mass balance equations of the Benchmark plant with ASM2, the following vectors and matrices are deduced.

The vector for the system states for the n-th tank is:

$$x_n = [S_{O2n} \ S_{Fn} \ S_{An} \ S_{NH4n} \ S_{NO3n} \ S_{PO4n} \ S_{In} \ S_{ALKn} \ S_{N2n} \ X_{In} \ X_{Sn} \ X_{Hn} \ X_{PAOn} \ X_{PPn} \ X_{PHAn} \ X_{AUTn} \ X_{TSSn} \ X_{MeOHn} \ X_{MePn}]^T$$

$$x_n \in R^{19} \qquad \overline{n = 1,5}$$

$$x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T \in R^{95}$$

The vector for the input concentration is:

$$x_i = [S_{O2i} \ S_{Fi} \ S_{Ai} \ S_{NH4i} \ S_{NO3i} \ S_{PO4i} \ S_{Ii} \ S_{ALKi} \ S_{N2i} \ X_{Ii} \ X_{Si} \ X_{Hi} \ X_{PAOi} \ X_{PPi} \ X_{PHAi} \ X_{AUTi} \ X_{TSSi} \ X_{MeOHi} \ X_{MePi}]^T \in R^{19}$$

The vector for the system dissolved oxygen is:

$$S_O = [0 \ . \ . \ 0 \ S_{O1} \ 0 \ . \ . \ 0 \ 0 \ . \ . \ 0 \ S_{O2} \ 0 \ . \ . \ 0 \ 0 \ . \ . \ 0 \ S_{O5} \ 0 \ . \ . \ 0]^T \in R^{95}$$

The vector for the system oxygen transfer function is:

$$K_La(u) = [0 \ . \ . \ 0 \ K_La(u_1) \ 0 \ . \ . \ 0 \ 0 \ . \ . \ 0 \ K_La(u_2) \ 0 \ . \ . \ 0 \ 0 \ . \ . \ 0 \ K_La(u_5) \ 0 \ . \ . \ 0]^T \in R^{95}$$

The vector for the system process rates for the n-th tank is:

$$P_n = [P_{1n} \ P_{2n} \ P_{3n} \ P_{4n} \ P_{5n} \ P_{6n} \ P_{7n} \ P_{8n} \ P_{9n} \ P_{10n} \ P_{11n} \ P_{12n} \ P_{13n} \ P_{14n} \ P_{15n} \ P_{16n} \ P_{17n} \ P_{18n} \ P_{19n}]^T \in R^{19}$$

$$P = [P_1 \ P_2 \ P_3 \ P_4 \ P_5]^T \in R^{95}$$

The matrix c for the whole system is:

$$C = \Delta t \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 & 0 \\ 0 & 0 & C_3 & 0 & 0 \\ 0 & 0 & 0 & C_4 & 0 \\ 0 & 0 & 0 & 0 & C_5 \end{bmatrix} \in R^{95 \times 95}$$

where:

$$C_n \in R^{19 \times 19} \quad \overline{n = 1,5}$$

The matrix A for the whole system is:

$$A = \begin{bmatrix} A_{11} & 0 & 0 & 0 & A_{15} \\ A_{21} & A_{22} & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & 0 \\ 0 & 0 & 0 & A_{54} & A_{55} \end{bmatrix} \in R^{95 \times 95}$$

The matrix B for the whole system is:

$$B = [B_1 \ 0 \ 0 \ 0 \ 0]^T \in R^{95 \times 19}$$

where: $B_1 \in R^{19 \times 19}$

5.3.7 Simulation data for the Benchmark plant with ASM2d model

The MATLAB file *"BASM2d_Simulation.m"* given in Appendix C.1 is used for simulation of the Benchmark plant with ASM2d.

From the full mass balance equations of the Benchmark plant with ASM2d, the following vectors and matrices are deduced.

The vector for the system states for the n-th tank is:

$$x_n = [S_{O2n} \quad S_{Fn} \quad S_{An} \quad S_{NH4n} \quad S_{NO3n} \quad S_{PO4n} \quad S_{In} \quad S_{ALKn} \quad S_{N2n} \quad X_{In} \quad X_{Sn} \quad X_{Hn} \quad X_{PAOn} \quad X_{PPn} \quad X_{PHAn} \quad X_{AUTn} \quad X_{TSSn} \quad X_{MeOHn} \quad X_{MePn}]^T$$

$$x_n \in R^{19} \qquad \overline{n = 1,5}$$

$$x = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5]^T \in R^{95}$$

The vector for the input concentration is:

$$x_i = [S_{O2i} \quad S_{Fi} \quad S_{Ai} \quad S_{NH4i} \quad S_{NO3i} \quad S_{PO4i} \quad S_{Ii} \quad S_{ALKi} \quad S_{N2i} \quad X_{Ii} \quad X_{Si} \quad X_{Hi} \quad X_{PAOi} \quad X_{PPi} \quad X_{PHAi} \quad X_{AUTi} \quad X_{TSSi} \quad X_{MeOHi} \quad X_{MePi}]^T \in R^{19}$$

The vector for the system dissolved oxygen is:

$$S_O = [0 \dots 0 \ S_{O1} \ 0 \dots 0 \ 0 \dots 0 \ S_{O2} \ 0 \dots 0 \ 0 \dots 0 \ S_{O5} \ 0 \dots 0]^T \in R^{95}$$

The vector for the system oxygen transfer function is:

$$K_L a(u) = [0 \dots 0 \ K_L a(u_1) \ 0 \dots 0 \ 0 \dots 0 \ K_L a(u_2) \ 0 \dots 0 \ 0 \dots 0 \ K_L a(u_5) \ 0 \dots 0]^T \in R^{95}$$

The vector for the system process rates for the n-th tank is:

$$P_n = [P_{1n} \quad P_{2n} \quad P_{3n} \quad P_{4n} \quad P_{5n} \quad P_{6n} \quad P_{7n} \quad P_{8n} \quad P_{9n} \quad P_{10n} \quad P_{11n} \quad P_{12n} \quad P_{13n} \quad P_{14n} \quad P_{15n} \quad P_{16n} \quad P_{17n} \quad P_{18n} \quad P_{19n}]^T \in R^{21}$$

$$P = [P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5]^T \in R^{105}$$

The matrix c for the whole system is:

$$C = \Delta t \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 & 0 \\ 0 & 0 & C_3 & 0 & 0 \\ 0 & 0 & 0 & C_4 & 0 \\ 0 & 0 & 0 & 0 & C_5 \end{bmatrix} \in R^{105 \times 95}$$

where:

$$C_n \in R^{21 \times 19} \quad \overline{n = 1,5}$$

The matrix A for the whole system is:

$$A = \begin{bmatrix} A_{11} & 0 & 0 & 0 & A_{15} \\ A_{21} & A_{22} & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & 0 \\ 0 & 0 & 0 & A_{54} & A_{55} \end{bmatrix} \in R^{95 \times 95}$$

The matrix B for the whole system is:

$$B = [B_1 \quad 0 \quad 0 \quad 0 \quad 0]^T \in R^{95 \times 19}$$

where:

$$B_1 \in R^{19 \times 19}$$

5.3.8 Simulation data for the Benchmark plant with ASM3 model

The MATLAB file "*BASM3_Simulation.m*" given in Appendix D.1 is used for simulation of the Benchmark plant with ASM3.

From the full mass balance equations of the Benchmark plant with ASM3, the following vectors and matrices are deduced.

The vector for the system states for the n-th tank is:

$$x_n = [S_{O2n} \quad S_{ln} \quad S_{Sn} \quad S_{NH4n} \quad S_{N2n} \quad S_{NOXn} \quad S_{ALKn} \quad X_{ln} \quad X_{Sn} \quad X_{Hn} \quad X_{STOn} \quad X_{An} \quad X_{SSn}]^T \in R^{13} \quad \overline{n = 1,5}$$

$$x = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5]^T \in R^{65}$$

The vector for the input concentration is:

$$x_i = [S_{O2i} \quad S_{li} \quad S_{Si} \quad S_{NH4i} \quad S_{N2i} \quad S_{NOXi} \quad S_{ALKi} \quad X_{li} \quad X_{Si} \quad X_{Hi} \quad X_{STOi} \quad X_{Ai} \quad X_{SSi}]^T \in R^{13}$$

The vector for the system dissolved oxygen is:

$$S_O = [0 \ .. \ 0 \ S_{O1} \ 0 \ .. \ 0 \ 0 \ .. \ 0 \ S_{O2} \ 0 \ .. \ 0 \ 0 \ .. \ 0 \ S_{O3} \ 0 \ .. \ 0]^T \in R^{65}$$

The vector for the system oxygen transfer function is:

$$K_La(u) = [0 \ .. \ 0 \ K_La(u_1) \ 0 \ .. \ 0 \ 0 \ .. \ 0 \ K_La(u_2) \ 0 \ .. \ 0 \ 0 \ .. \ 0 \ K_La(u_5) \ 0 \ .. \ 0]^T \in R^{65}$$

The vector for the system process rates for the n-th tank is:

$$P_n = [P_{1n} \quad P_{2n} \quad P_{3n} \quad P_{4n} \quad P_{5n} \quad P_{6n} \quad P_{7n} \quad P_{8n} \quad P_{9n} \quad P_{10n} \quad P_{11n} \quad P_{12n}]^T \in R^{12}$$

$$P = [P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5]^T \in R^{60}$$

The matrix c for the whole system is:

$$C = \Delta t \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 & 0 \\ 0 & 0 & C_3 & 0 & 0 \\ 0 & 0 & 0 & C_4 & 0 \\ 0 & 0 & 0 & 0 & C_5 \end{bmatrix} \in R^{65 \times 65}$$

where:

$$C_n \in R^{12 \times 13} \quad \overline{n = 1,5}$$

The matrix A for the whole system is:

125

$$A = \begin{bmatrix} A_{11} & 0 & 0 & 0 & A_{15} \\ A_{21} & A_{22} & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & 0 \\ 0 & 0 & 0 & A_{54} & A_{55} \end{bmatrix} \in R^{65 \times 65}$$

The matrix B for the whole system is:

$$B = \begin{bmatrix} B_1 & 0 & 0 & 0 & 0 \end{bmatrix}^T \in R^{65 \times 13}$$

where:

$$B_1 \in R^{13 \times 13}$$

5.3.9 Discussion of the results of the steady state and dynamic simulations of the full models

Steady state results for all the variables of the full mass balance model of the Benchmark plant with ASM1 biological model are shown in Table 5.4. Steady state is achieved for all the process variables of all the plants mass balance models. Table 5.5 shows a comparison between steady state results achieved in this project (highlighted in grey) and those from the COST Benchmark Group (highlighted in black) for the Benchmark plant with ASM1. The results are very close to each other and this shows that the procedure for steady state simulation was applied correctly. This procedure is applied to all the mass balance models.

Table 5.5: Comparison of the steady state results achieved in the thesis and those achieved by the COST Benchmark Group

Components	Tank 1	Tank 2	Tank 3	Tank 4	Tank 5
SNH- thesis	4.9639	3.5994	1.4408	0.4273	0.1634
SNH- Benchmark	7.92	8.34	5.55	2.97	1.73
SI - thesis	30	30	30	30	30
SI - Benchmark	30	30	30	30	30
SALK - thesis	3.2415	3.0472	2.7207	2.5556	2.50
SALK - Benchmark	4.93	5.08	4.67	4.29	4.13
SND- thesis	1.2434	0.9498	0.7914	0.7031	0.6414
SND- Benchmark	1.22	0.882	0.829	0.767	0.688
XS - thesis	74.9448	66.6212	57.2967	50.0176	44.5204
XS - Benchmark	82.1	76.4	64.9	55.7	49.3
XND- thesis	4.8343	4.4095	3.9169	3.5256	3.5256
XND- Benchmark	5.28	5.03	4.39	3.88	3.53

5.4   Simulations of the reduced models

Computer simulations of the reduced models must verify that the main characteristics of the process dynamics are similar to those from the full models. To ensure that the reduced mass balance models provide similar prediction capabilities as the full mass balance models, parameters of the reduced biological models are calculated.

Some parameters are considered to be constants with values as suggested by the task group (Henze et al, 1987) and some parameters are considered to be time varying and unknown. These parameters need to be identified. Using an offline optimization method, the parameters of the reduced models are identified by least squares parameter estimation method. In section 5.5.1 below, all the unknown parameters are presented and calculated before they are used in section 5.5.2 for simulation.

5.4.1   Estimation of the parameters of the reduced models

The following information of parameter estimation methods is taken from Dochain and Vanrolleghem (2001). Reduced models need to be calibrated to be sure that they give the same prediction capabilities as the full models. This requires estimation of the model parameters. Parameter estimation refers to the determination of the values of the parameters of the model with the aid of experimental data.

The first important step in parameter estimation is to provide a set of data for estimation of the parameters. This data is used to calculate the estimates of the parameters using an appropriate estimation algorithm. In this case, the data used is acquired from simulation of the full models. The second step is to select which parameters are to be estimated.

Selection of parameters to be estimated is important because these parameters have to be identifiable and to be capable of influencing the process behavior in a maximum way. Therefore it is important to carry parameter identifiability analysis before model calibration or parameter estimation is conducted. This idea addresses the question of whether a given output uniquely determines all of the model parameters. (Bellu et al., 2009).

Typical methods used for selection of the parameters are: structural and practical identifiability analysis, sensitivity analysis and numerical properties of the estimation algorithm.

127

Structural identifiability analysis allows one to find out the possibly identifiable parameters or combinations thereof. In this analysis, one of the parameters can be estimated while the others are set to constant. When this parameter is found, it means its value is conditioned by the choice of the values of the other parameters. This method works well in situation where the data is sufficiently rich in information.

In practical identifiability analysis, a sub-selection of the parameter set after an analysis of the parameter estimation error is done. By eliminating the parameter that causes identifiability problems and giving it an assumed value, estimation of the other parameters may be highly facilitated. This means that estimates of the parameters are conditioned by the choice of the value of the other parameter. This method works well in situation where the data is not sufficiently rich in information.

Sensitivity analysis is used to pre select parameter subsets that ensure reliable estimation. In numerical properties of the estimation problem, the parameter estimation problem is structured in a stepwise procedure; a large parameter estimation problem is split into a series of smaller problems.

The sensitivity function represents a change in the variables of the process for a change of a parameter. The large the values of the sensitivity functions, the more accurately a single parameter can be identified. In the case of several parameters, the sensitivity functions of the parameters as functions of the independent variable of the measurement (time) have to be linearly independent. Otherwise the parameters are not individually identifiable because a change in one parameter can be compensated for by changes in the other parameters (du Plessis, 2009). Since nonlinearity complicates the estimation problem considerably, it is advantageous to separate the linear from the nonlinear parameters and create subsets of these two types of parameters.

To reduce the estimations complications due to parameter nonlinearity, the model is written in a form linear in the parameters. This formation is termed re-parameterisation. The disadvantage with this method is that the physical meaning of parameters may be lost. Minimising the level of nonlinearity of the parameters maximises the quality of parameter estimation results.

128

Once a model is written in a form linear in the parameters, parameters can be calculated analytically from the objective function, this becomes an optimization problem. In parameter estimation, the objective functions are conventionally arranged such that small values of those functions represent close agreement between model and data.

The model parameters are then adjusted to achieve a minimum in these functions, yielding best fit parameters. This adjustment of parameters is therefore a minimisation problem. The functions to be minimised are called loss, merit, cost or objective functions. The choice of these functions is one of the problems to be solved when model parameters are to be estimated. The best know objective function for parameter estimation is the sum of squared errors functions referred to as the Ordinary Least Squares Estimation, shown in Equation (5.1). This is the objective function used in this project.

$$J(\theta) = \sum_{k=1}^{K} (y(k) - \bar{y}(k, \theta))^2 \tag{5.1}$$

where $y(k)$ are the observations which are obtained by simulation of the full model, K is the total number of observations available, $\bar{y}(k, \theta)$ are the reduced model predictions for a given parameter set θ, θ is the given parameter set.

The optimization problem is to find the values of θ which minimize the least square function. The least-squares technique is defined as a mathematical procedure by which the unknown parameters of a mathematical model are chosen such that the sum of the squares of some chosen error (Ikonem and Najim, 2002) is minimized. In general a model linear in the parameters can be written in the following form

$$y(k) = \phi^T(k)\theta(k) \tag{5.2}$$

where θ is the parameter vector, ϕ is the termed regressor, $y(k)$ contains the terms that are independent of the parameters, and k refers to the discrete time

As can be seen in the Peterson matrix representation of each biological model, the process rates vector (P) may be expressed in terms of the kinetic parameters.
For reduced models the vector of the process rates can then be expressed in terms of unknown parameters (θ). The model parameters and reaction rates can then be presented in a vector form as follows:

$$\theta = [\theta_1 \ \ldots \ \theta_v]^T \tag{5.3}$$

$$P_n = [p_{n,1} \ \ldots \ p_{n,n2}]^T \tag{5.4}$$

where v is the total number of the unknown kinetic parameters to be estimated and n_2 is the total number of process rates in the Peterson matrix for the **n-th** tank.

Then $p_n = p\theta$, where $p_n \in R^{n_2 v}$, $P_n \in R^{n_2}$

The process rates vector (P) in the state space model equation of the whole plant (Equation 4.165) may then be rewritten in terms of process rates and unknown parameters ($p\theta$):

$$x(k+1) = Ax(k) + C^T p(x,u,k)\theta + Bx_i(k) \tag{5.5}$$

Where:

$p = [p_1 \ p_2 \ \cdots \ p_n \ \cdots \ p_N]^T \in R^{Nn_2 v}$ where N is the number of tanks in the considered mass balance model and is the vector containing part of the process rate multiplying the parameter vector.

Data for $x(k)$ and $u(k)$ for the period $k = \overline{1K}$ may then be substituted in Equation (5.5). Rewriting the model to have a form of Equation (5.2) yields,

$$y(k) = x(k+1) - Ax(k) - Bx_i(k) = C^T p(x,u,k)\theta \tag{5.6}$$

where $y(k)$ and ϕ corresponds to:

$$y(k) = x(k+1) - Ax(k) - Bx_i(k) \tag{5.7}$$

$$\phi(k) = C^T p(x,u,k) \tag{5.8}$$

In the simple case of a linear equation with unknown parameters, linear regression can be applied; in this case a least squares objective function in Equations (5.1) is used. The objective function in a matrix format becomes

$$J(\theta) = \sum_{k=0}^{K} (y(k) - \hat{y}(k,\theta))^T (y(k) - \hat{y}(k,\theta)) \tag{5.9}$$

Equation (5.9) can then be expressed in terms of Equation (5.2) as

$$J(\theta) = \sum_{k=0}^{K} (y(k) - \phi(k)^T \hat{\theta}(k))^T (y(k) - \phi(k)^T \hat{\theta}(k)) \tag{5.10}$$

Therefore

$$J(\theta) = \sum_{k=0}^{K} y(k)^T y(k) - \left(\hat{\theta}(k)^T \phi(k)^T\right) y(k) - y(k)^T \hat{\theta}(k)\phi(k) - \hat{\theta}(k)^T \phi(k)^T \hat{\theta}(k)\phi(k) \tag{5.11}$$

The least squares estimator that minimises this criterion can be deduced from the first derivative of the function in Equation (5.11) towards the parameter θ to be:

$$\hat{\theta} = \left[\sum_{k=0}^{K} \phi(k)^T \phi(k) \right]^{-1} \left[\sum_{k=0}^{K} \phi(k)^T y(k) \right] \tag{5.12}$$

Equation (5.12) can be written in a vector matrix form as

$$\hat{\theta}(k) = \left(\phi(k)^T \phi(k) \right)^{-1} \phi(k)^T y(k) \tag{5.13}$$

where:

$$\hat{\theta} = \begin{bmatrix} \theta(0) \\ \cdot \\ \cdot \\ \cdot \\ \theta(K) \end{bmatrix}, \quad \phi = \begin{bmatrix} \phi(0) & & \\ & \cdot & \\ & & \cdot \\ & & \phi(K) \end{bmatrix}, \quad y = \begin{bmatrix} y(0) \\ \cdot \\ \cdot \\ \cdot \\ y(K) \end{bmatrix}$$

Equation (5.13) is used in the subsequent section to calculate model parameter values for the different models. In this section, stoichiometric parameters in the matrices c as well as parameters of the switching functions are accepted to be constant and known. Values of these parameters are the same as in the full models.

The growth rates parameters are considered to be time varying and unknown, these parameters are determined in the next section through least square estimation techniques described above. The flow chart for calculation of the parameters is presented in Figure 5.15. The parameters to be estimated are selected because they are very important to the bacterial growth and therefore important for the whole process behaviour.
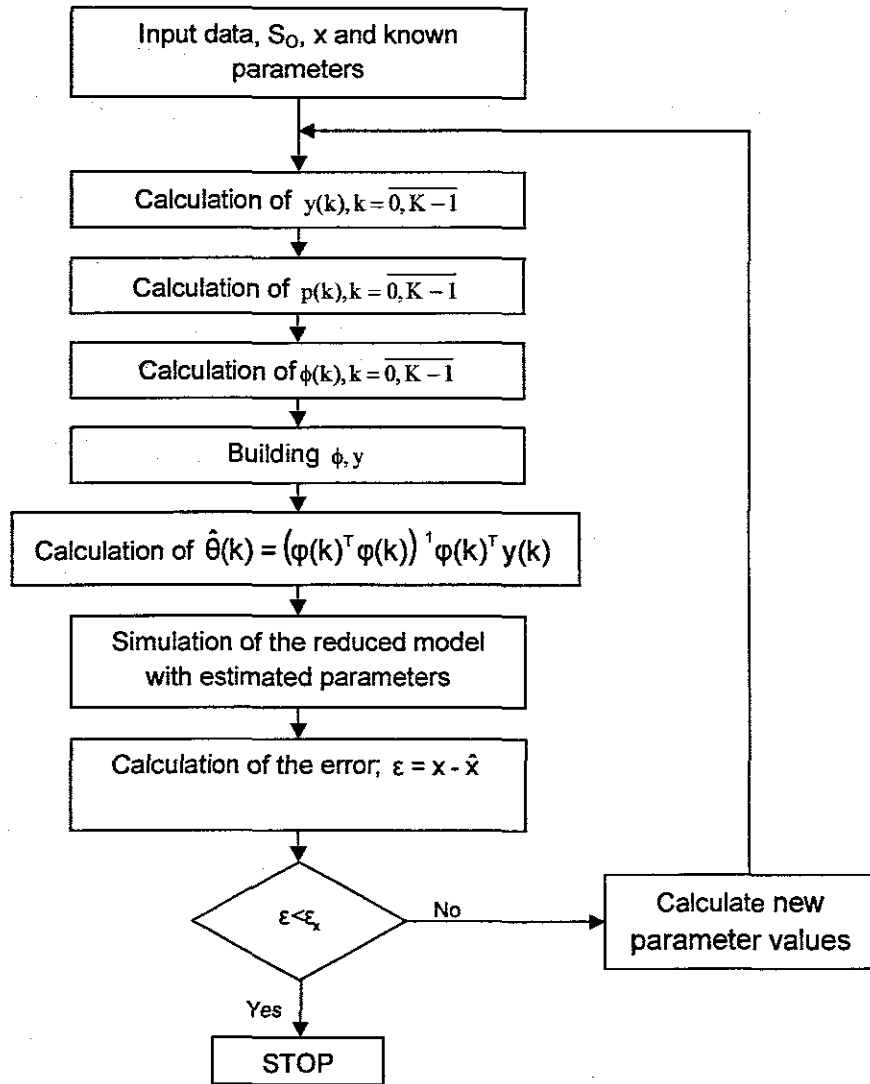
Figure 5.15: Flow chart with the algorithm for calculation of the estimated parameters

Estimations of the parameters of the reduced biological models are calculated, estimation is done for the following:

- Reduced ASM1
- Reduced ASM2
- Reduced ASM2d
- Reduced ASM3

5.4.1.1 Calculation of the reduced-ASM1 parameters

The following derivations are based upon the equations for the reduced-ASM1 model process rates. Equations (4.63 – 4.66). The vector for the reduced-ASM1 model parameters to be estimated is selected to be:

$$\theta = \begin{bmatrix} \mu_H & \mu_A & k_h \end{bmatrix}^T$$

The unknown parameters vector for the reduced-ASM1 model therefore becomes:

$$\theta = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T$$

where $\theta_1 = \mu_H$ $\theta_2 = \mu_A$ $\theta_3 = k_h$

The vector for the reduced-ASM1 model process rates can then be expressed in terms of unknown parameters for the **n-th** tank as follows:

$$P_n = \begin{bmatrix} p_{n1}\theta_1 & p_{n2}\theta_1 & p_{n3}\theta_2 & p_{n4}\theta_3 \end{bmatrix}^T \in R^{4x3} \tag{5.14}$$

where $v = 3$ $n2 = 4$ $N = 3$

$$p_{n1}\theta_1 = \theta_1 \cdot \left(\frac{S_s}{K_s + S_s} \right)\left(\frac{S_o}{K_{O,H} + S_o} \right) X_{B,H} \tag{5.15}$$

$$p_{n2}\theta_1 = \theta_1 \cdot \left(\frac{S_s}{K_s + S_s} \right)\left(\frac{K_{O,H}}{K_{O,H} + S_o} \right)\left(\frac{S_{NO}}{K_{NO} + S_{NO}} \right)\eta_g X_{B,H} \tag{5.16}$$

$$p_{n3}\theta_2 = \theta_2 \cdot \left(\frac{S_{NH}}{K_{NH} + S_{NH}} \right)\left(\frac{S_o}{K_{O,A} + S_o} \right) \tag{5.17}$$

$$p_{n4}\theta_3 = \theta_3 \cdot \frac{X_s/X_{B,H}}{K_x + X_s/X_{B,H}}\left[\left(\frac{S_o}{K_{O,H} + S_o} \right) + \eta_h\left(\frac{K_{O,H}}{K_{O,H} + S_o} \right)\left(\frac{S_{NO}}{K_{NO} + S_{NO}} \right) \right] X_{B,H} \tag{5.18}$$

The process rates equations represented in a matrix form for the **n-th** tank are as follows:

$$P_n = \begin{bmatrix} p_{n1}\theta_1 \\ p_{n2}\theta_1 \\ p_{n3}\theta_2 \\ p_{n4}\theta_3 \end{bmatrix} = \begin{bmatrix} p_{n1} & 0 & 0 \\ p_{n2} & 0 & 0 \\ 0 & p_{n3} & 0 \\ 0 & 0 & p_{n4} \end{bmatrix}\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \tag{5.19}$$

For the whole system the matrix of the process rate becomes:

$$P = \begin{bmatrix} p1 \\ \cdot \\ \cdot \\ p_N \end{bmatrix} \in R^{12x3}$$

From Equation (5.19) the matrix with process rates can then be used in Equation (5.8) to calculate ϕ which is then used together with $y(k)$ to calculate unknown reduced-ASM1 model parameters in Equation (5.13).

The results from the calculation of the reduced-ASM1 model parameters are shown in Figure 5.16.

133
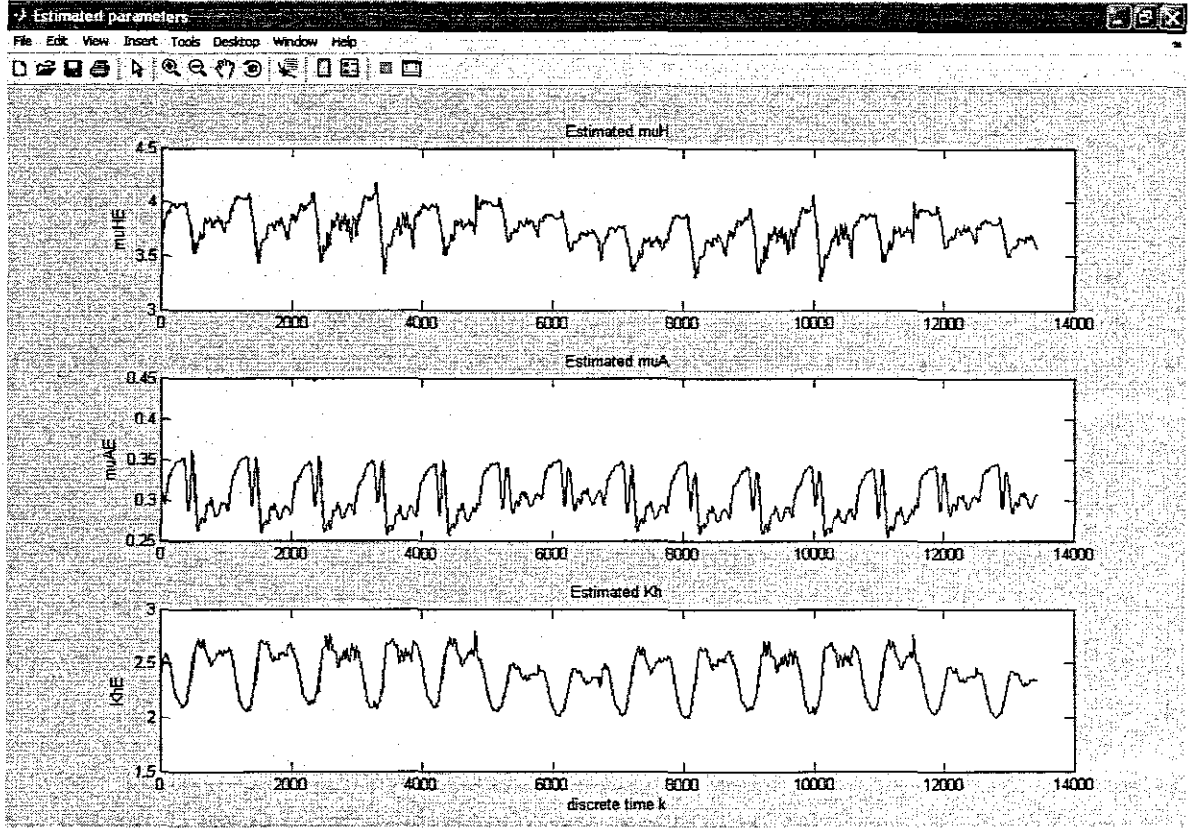
Figure 5.16: Calculated parameters for the Benchmark plant with the ASM1 biological model

5.4.1.2 Calculation of the reduced-ASM2 parameters

The following derivations are based upon the equations for the reduced-ASM2 model process rates Equations (4.67 – 4.78). The vector for the reduced-ASM2 model parameters to be estimated is selected to be:

$$\theta = \begin{bmatrix} K_h & \mu_H & q_{fe} & q_{PHA} & q_{PP} & \mu_{PAO} & \mu_{AUT} \end{bmatrix}^T$$

The unknown parameters vector for the reduced-ASM2 model therefore becomes:

$$\theta = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 & \theta_7 \end{bmatrix}^T \tag{5.20}$$

where $\theta_1 = K_h$ $\theta_2 = \mu_H$ $\theta_3 = q_{fe}$ $\theta_4 = q_{PHA}$ $\theta_5 = q_{PP}$ $\theta_6 = \mu_{PAO}$ $\theta_7 = \mu_{AUT}$

The vector for the reduced-ASM2 model process rates can then be expressed in terms of unknown parameters for the n-th tank as follows:

$$P_n = \begin{bmatrix} p_{n1}\theta_1 & p_{n2}\theta_1 & p_{n3}\theta_1 & p_{n4}\theta_2 & p_{n5}\theta_2 & p_{n6}\theta_2 & p_{n7}\theta_2 & p_{n8}\theta_3 & p_{n9}\theta_4 & p_{n10}\theta_5 & p_{n11}\theta_6 & p_{n12}\theta_7 \end{bmatrix}^T \in R^{12 \times 7} \tag{5.21}$$

where $v = 7$ $n2 = 12$ $N = 3$

$$p_{n1}\theta_1 = \theta_1 \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{X_S/X_H}{K_X + X_S/X_H} \cdot X_H \tag{5.22}$$

134

$$p_{n2}\theta_1 = \theta_1 \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H \qquad (5.23)$$

$$p_{n3}\theta_1 = \theta_1 \cdot \eta_{fe} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H \qquad (5.24)$$

$$p_{n4}\theta_2 = \theta_2 \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_F}{K_F+S_F} \cdot \frac{S_F}{K_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (5.25)$$

$$p_{n5}\theta_2 = \theta_2 \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_A}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (5.26)$$

$$p_{n6}\theta_2 = \theta_2 \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_F}{K_F+S_F} \cdot \frac{S_F}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (5.27)$$

$$p_{n7}\theta_2 = \theta_2 \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_A}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot X_H \qquad (5.28)$$

$$p_{n8}\theta_3 = \theta_3 \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_F}{K_{fe}+S_F} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (5.29)$$

$$p_{n9}\theta_4 = \theta_4 \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{PP}/X_{PAO}}{K_{PP}+X_{PP}/X_{PAO}} \cdot X_{PAO} \qquad (5.30)$$

$$p_{n10}\theta_5 = \theta_5 \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{PO4}}{K_{PS}+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot \frac{K_{MAX}-X_{PP}/X_{PAO}}{K_{IPP}+K_{MAX}-X_{PP}/X_{PAO}} \cdot X_{PAO} \qquad (5.31)$$

$$p_{n11}\theta_6 = \theta_6 \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot X_{PAO} \qquad (5.32)$$

$$p_{n12}\theta_7 = \theta_7 \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT} \qquad (5.33)$$

The process rates equations represented in a matrix form for the **n-th** tank are as follows:

$$p_n = \begin{bmatrix} p_{n1}\theta_1 \\ p_{n2}\theta_1 \\ p_{n3}\theta_1 \\ p_{n4}\theta_2 \\ p_{n5}\theta_2 \\ p_{n6}\theta_2 \\ p_{n7}\theta_2 \\ p_{n8}\theta_3 \\ p_{n9}\theta_4 \\ p_{n10}\theta_5 \\ p_{n11}\theta_6 \\ p_{n12}\theta_7 \end{bmatrix} = \begin{bmatrix} p_{n1} & 0 & 0 & 0 & 0 & 0 & 0 \\ p_{n2} & 0 & 0 & 0 & 0 & 0 & 0 \\ p_{n3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & p_{n4} & 0 & 0 & 0 & 0 & 0 \\ 0 & p_{n5} & 0 & 0 & 0 & 0 & 0 \\ 0 & p_{n6} & 0 & 0 & 0 & 0 & 0 \\ 0 & p_{n7} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{n8} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{n9} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{n10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & p_{n11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{n12} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \end{bmatrix} \qquad (5.34)$$

For the whole system the matrix of the process rate becomes:

$$P = \begin{bmatrix} p1 \\ \cdot \\ \cdot \\ p_N \end{bmatrix} \in R^{36 \times 7}$$

From Equation (5.34) the matrix with process rates can then be used in Equation (5.8) to calculate ϕ which is then used together with $y(k)$ to calculate unknown reduced-ASM2 parameters in Equation (5.13).

5.4.1.3 Calculation of the reduced-ASM2d parameters

The following derivations are based upon the Equations (4.79 – 4.92) for the reduced-ASM2d model process rates. The vector for the reduced-ASM2d model parameters to be estimated is selected to be:

$$\theta = \begin{bmatrix} K_h & \mu_H & q_{fe} & q_{PHA} & q_{PP} & \mu_{PAO} & \mu_{AUT} \end{bmatrix}^T$$

The unknown parameters vector for the reduced-ASM2d model therefore becomes:

$$\theta = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 & \theta_7 \end{bmatrix}^T \tag{5.35}$$

where $\theta_1 = K_h$ $\theta_2 = \mu_H$ $\theta_3 = q_{fe}$ $\theta_4 = q_{PHA}$ $\theta_5 = q_{PP}$ $\theta_6 = \mu_{PAO}$ $\theta_7 = \mu_{AUT}$

The vector for the reduced-ASM2d model process rates can then be expressed in terms of unknown parameters for the **n-th** tank as follows:

$$P_n = \begin{bmatrix} p_{n1}\theta_1 & p_{n2}\theta_1 & p_{n3}\theta_1 & p_{n4}\theta_2 & p_{n5}\theta_2 & p_{n6}\theta_2 & p_{n7}\theta_2 & p_{n8}\theta_3 \\ p_{n9}\theta_4 & p_{n10}\theta_5 & p_{n11}\theta_6 & p_{n12}\theta_6 & p_{n13}\theta_7 & p_{n14}\theta_7 \end{bmatrix}^T \tag{5.36}$$

where $v = 7$ $n2 = 14$ $N = 3$ $p_n \in R^{14 \times 7}$

$$p_{n1}\theta_1 = \theta_1 \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H \tag{5.37}$$

$$p_{n2}\theta_1 = \theta_1 \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H \tag{5.38}$$

$$p_{n3}\theta_1 = \theta_1 \cdot \eta_{fe} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H \tag{5.39}$$

$$p_{n4}\theta_2 = \theta_2 \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_F}{K_F+S_F} \cdot \frac{S_F}{K_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \tag{5.40}$$

$$p_{n5}\theta_2 = \theta_2 \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_A}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \tag{5.41}$$

$$p_{n6}\theta_2 = \theta_2 \cdot \eta_{NO_3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_F}{K_F+S_F} \cdot \frac{S_F}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \tag{5.42}$$

$$p_{n7}\theta_2 = \theta_2 \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_A}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot X_H \tag{5.43}$$

$$p_{n8}\theta_3 = \theta_3 \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_F}{K_{fe}+S_F} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \tag{5.44}$$

$$p_{n9}\theta_4 = \theta_4 \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{PP}/X_{PAO}}{K_{PP}+X_{PP}/X_{PAO}} \cdot X_{PAO} \tag{5.45}$$

$$p_{n10}\theta_5 = \theta_5 \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{PO4}}{K_{PS}+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot \frac{K_{MAX}-X_{PP}/X_{PAO}}{K_{IPP}+K_{MAX}-X_{PP}/X_{PAO}} \cdot X_{PAO} \tag{5.46}$$

136

$$p_{m11}\theta_6 = \theta_6 \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot X_{PAO} \qquad (5.47)$$

$$p_{m12}\theta_6 = p_{11}\theta_6 \cdot p_{11} \cdot \eta_{NO3} \cdot \frac{K_{O2}}{S_{O2}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \qquad (5.48)$$

$$p_{m13}\theta_7 = \theta_7 \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT} \qquad (5.49)$$

$$p_{m14}\theta_7 = p_{13}\theta_7 \cdot \eta_{NO3} \cdot \frac{K_{O2}}{S_{O2}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \qquad (5.50)$$

The process rates equations represented in a matrix form for the **n-th** tank are as follows:

$$
p_n =
\begin{bmatrix}
p_{n1}\theta_1 \\
p_{n2}\theta_1 \\
p_{n3}\theta_1 \\
p_{n4}\theta_2 \\
p_{n5}\theta_2 \\
p_{n6}\theta_2 \\
p_{n7}\theta_2 \\
p_{n8}\theta_3 \\
p_{n9}\theta_4 \\
p_{n10}\theta_5 \\
p_{n11}\theta_6 \\
p_{n12}\theta_6 \\
p_{n13}\theta_7 \\
p_{n14}\theta_7
\end{bmatrix}
=
\begin{bmatrix}
p_{n1} & 0 & 0 & 0 & 0 & 0 & 0 \\
p_{n2} & 0 & 0 & 0 & 0 & 0 & 0 \\
p_{n3} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & p_{n4} & 0 & 0 & 0 & 0 & 0 \\
0 & p_{n5} & 0 & 0 & 0 & 0 & 0 \\
0 & p_{n6} & 0 & 0 & 0 & 0 & 0 \\
0 & p_{n7} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & p_{n8} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & p_{n9} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & p_{n10} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & p_{n11} & 0 \\
0 & 0 & 0 & 0 & 0 & p_{n12} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & p_{n13} \\
0 & 0 & 0 & 0 & 0 & 0 & p_{n14}
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\
\theta_2 \\
\theta_3 \\
\theta_4 \\
\theta_5 \\
\theta_6 \\
\theta_7
\end{bmatrix}
\qquad (5.51)
$$

For the whole system the matrix of the process rate becomes:

$$P = \begin{bmatrix} p1 \\ \cdot \\ \cdot \\ \cdot \\ p_N \end{bmatrix} \in R^{42 \times 7}$$

From Equation (5.51) the matrix with process rates can then be used in Equation (5.8) to calculate ϕ which is then used together with $y(k)$ to calculate unknown reduced-ASM2d model parameters in Equation (5.13).

5.4.1.4 Calculation of the reduced-ASM3 parameters

The following derivations are based upon the equations for the reduced-ASM3 model process rates Equations (4.93 – 4.96). The vector for the reduced-ASM3 model parameters to be estimated is selected to be:

$$\theta = \begin{bmatrix} k_H & \mu_H & \mu_A \end{bmatrix}^T \qquad (5.52)$$

The unknown parameters vector for the reduced-ASM3 model therefore becomes:

$$\theta = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T \qquad (5.53)$$

where $\theta_1 = k_H$ $\theta_2 = \mu_H$ $\theta_3 = \mu_A$

137

The vector for the reduced-ASM3 model process rates can then be expressed in terms of unknown parameters for the n-th tank as follows:

$$P_n = [p_{n1}\theta_1 \quad p_{n2}\theta_1 \quad p_{n3}\theta_2 \quad p_{n4}\theta_3]^T \tag{5.54}$$

where $v = 3 \quad n2 = 4 \quad N = 3 \quad p_n \in R^{4x3}$

$$p_{n1}\theta_1 = \theta_1 \cdot \frac{X_S/X_H}{K_X + X_S/X_H} \cdot X_H \tag{5.55}$$

$$p_{n2}\theta_2 = \theta_2 \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{STO}/X_H}{K_{STO} + X_{STO}/X_H} \cdot X_H \tag{5.56}$$

$$p_{n3}\theta_2 = \theta_2 \cdot \eta_{NOX} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{NOX}}{K_{NOX} + S_{NOX}} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{STO}/X_H}{K_{STO} + X_{STO}/X_H} \cdot X_H \tag{5.57}$$

$$p_{n4}\theta_3 = \theta_3 \cdot \frac{S_{O2}}{K_{A,O2} + S_{O2}} \cdot \frac{S_{NH4}}{K_{A,NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{A,ALK} + S_{ALK}} \cdot X_A \tag{5.58}$$

The process rates equations represented in a matrix form for the n-th tank are as follows:

$$p_n = \begin{bmatrix} p_{n1}\theta_1 \\ p_{n2}\theta_2 \\ p_{n3}\theta_2 \\ p_{n4}\theta_3 \end{bmatrix} = \begin{bmatrix} p_{n1} & 0 & 0 \\ 0 & p_{n2} & 0 \\ 0 & p_{n3} & 0 \\ 0 & 0 & p_{n4} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \tag{5.59}$$

For the whole system the matrix of the process rate becomes:

$$P = \begin{bmatrix} p1 \\ \cdot \\ \cdot \\ p_N \end{bmatrix} \in R^{12x3}$$

From Equation (5.59) the matrix with process rates can then be used in Equation (5.8) to calculate ϕ which is then used together with $y(k)$ to calculate unknown reduced-ASM3 parameters in Equation (5.13).

Using the estimated parameters, the reduced models can then be simulated. The programs used for estimation of the parameters of the models are incorporated in the programs used for simulation of the reduced models.

5.4.1.5 Discussion of the results

The calculated values of the parameters for the reduced mass balance models are very close to constant parameter values used during simulations of the full mass balance models (muA = 0.5, muH = 4, kH = 3). This proofs the effectiveness of the Least squares method.

138

5.4.2 Simulation of the mass balance models with reduced biological models

In this section, dynamic simulation of the plants mass balance models incorporating reduced biological models is described. The flow chart for simulation of the reduced plants models is given in Figure 5.17. In this section, simulation results of both full and reduced models are shown so that a clear comparison can be made between the prediction capabilities of these models. The full model results are presented by a dotted solid line while the reduced models results are presented by a solid line. When running simulation of reduced models, the control trajectory is the S_o trajectory obtained from the simulation of the full models.
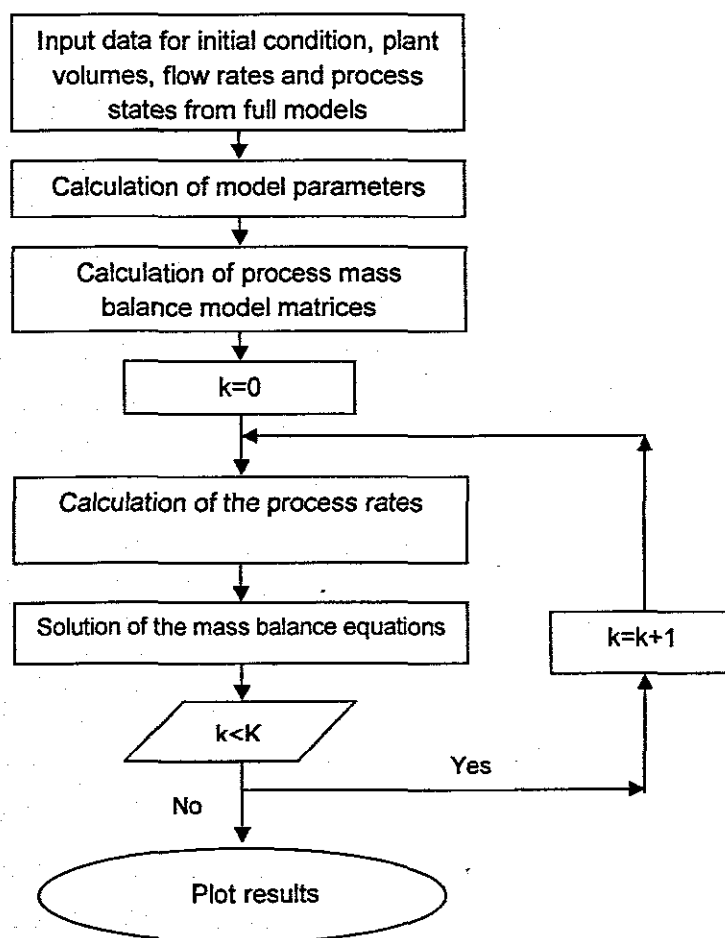
Figure 5.17: Flow chart for simulation of the reduced models

Firstly, values of the influent wastewater composition and plant data are introduced followed by calculations of the model parameter values. Then the matrices of the models (derived in section 4.3) are calculated.

Then at each instance (k) from k = 0 till k = K-1, the process rates (P) are calculated followed by calculations of the mass balance model outputs. K is the final time of the prediction horizon. Finally simulation results are plotted. These results may be used to examine the dynamic performance of the process. Calculations are done in a similar manner as for dynamic simulations of the full models.

The plants with reduced models are simulated for 14 days (in the same way as with the full dynamic models). Simulation is done for the following:

- Benchmark plant with reduced ASM1
- Benchmark plant with reduced ASM2
- Benchmark plant with reduced ASM2d
- Benchmark plant with reduced ASM3
- Athlone plant with reduced ASM1
- Athlone plant with reduced ASM2
- Athlone plant with reduced ASM2d
- Athlone plant with reduced ASM3

5.4.2.1 Simulation data for the Athlone plant with reduced-ASM1 model

The MATLAB file "*AASM1_Simulation*.m" given in Appendix E.1 is used for simulation of the Athlone plant with ASM1.

From the reduced mass balance equations of the Athlone plant with r-ASM1, the following vectors and matrices are deduced. The vector for the system states for the **n-th** tank is:

$$x_n = [S_{Sn} \quad S_{NOn} \quad S_{NHn}]^T \in R^3 \qquad \overline{n = 13}$$

$$x = [x_1 \quad x_2 \quad x_3]^T \in R^9$$

The vector for the input concentration is:

$$x_i = [S_{Si} \quad S_{NOi} \quad S_{NHi}]^T \in R^3$$

The vector for the process control variables is:

$$u = [S_{O1} \quad S_{O2} \quad S_{O3}]^T \in R^3$$

The vector for the system process rates for the **n-th** tank is:

$$PR_n = [p_{1n} \quad p_{2n} \quad p_{3n} \quad p_{4n}]^T \in R^4$$

$$PR = [P_1 \quad P_2 \quad P_3]^T \in R^{12}$$

5.4.2.2 Simulation data for the Athlone plant with reduced-ASM2 model

The MATLAB file "*AASM2_Simulation.m*" given in Appendix F.1 is used for simulation of the Athlone plant with ASM2.

From the reduced mass balance equations of the Athlone plant with r-ASM2, the following vectors and matrices are deduced.

The vector for the system states for the **n-th** tank is:

$$x_n = [S_{Fn} \quad S_{An} \quad S_{NHn} \quad S_{NO3n} \quad S_{PO4n} \quad X_{Sn} \quad X_{PPn} \quad X_{PHAn}]^T \in R^8 \qquad \overline{n = 1,3}$$

$$x = [x_1 \quad x_2 \quad x_3]^T \in R^{24}$$

The vector for the input concentration is:

$$x_i = [S_{Fi} \quad S_{Ai} \quad S_{NHi} \quad S_{NO3i} \quad S_{PO4i} \quad X_{Si} \quad X_{PPi} \quad X_{PHAi}]^T \in R^8$$

The vector for the process control variables is:

$$u = [S_{O1} \quad S_{O2} \quad S_{O3}]^T \in R^3$$

The vector for the system process rates for the **n-th** tank is:

$$PR_n = [p_{1n} \quad p_{2n} \quad p_{3n} \quad p_{4n} \quad p_{5n} \quad p_{6n} \quad p_{7n} \quad p_{8n} \quad p_{9n} \quad p_{10n} \quad p_{11n} \quad p_{12n}]^T \in R^{12}$$

$$PR = [P_1 \quad P_2 \quad P_3]^T \in R^{36}$$

5.4.2.3 Simulation data for the Athlone plant with reduced-ASM2d model

The MATLAB file "*AASM2d_Simulation.m*" given in Appendix G.1 is used for simulation of the Athlone plant with ASM2d.

From the reduced mass balance equations of the Athlone plant with r-ASM2d, the following vectors and matrices are deduced.

The vector for the system states for the **n-th** tank is:

$$x_n = [S_{Fn} \quad S_{An} \quad S_{NHn} \quad S_{NO3n} \quad S_{PO4n} \quad X_{Sn} \quad X_{PPn} \quad X_{PHAn}]^T \in R^8 \quad \overline{n = 1,3}$$

$$x = [x_1 \quad x_2 \quad x_3]^T \in R^{24}$$

The vector for the input concentration is:

$$x_i = [S_{Fi} \quad S_{Ai} \quad S_{NHi} \quad S_{NO3i} \quad S_{PO4i} \quad X_{Si} \quad X_{PPi} \quad X_{PHAi}]^T \in R^8$$

The vector for the process control variables is:

$$u = [S_{O1} \quad S_{O2} \quad S_{O3}]^T \in R^3$$

The vector for the system process rates for the **n-th** tank is:

$$P_n = [p_{1n} \quad p_{2n} \quad p_{3n} \quad p_{4n} \quad p_{5n} \quad p_{6n} \quad p_{7n} \quad p_{8n} \quad p_{9n} \quad p_{10n} \quad p_{11n} \quad p_{12n} \quad p_{13n} \quad p_{14n}]^T \in R^{14}$$

141

$$PR = \begin{bmatrix} P_1 & P_2 & P_3 \end{bmatrix}^T \in R^{42}$$

5.4.2.4 Simulation data for the Athlone plant with reduced-ASM3 model

The MATLAB file "*AASM3_Simulation.m*" given in Appendix H.1 is used for simulation of the Benchmark plant with ASM3. From the reduced mass balance equations of the Athlone plant with r-ASM3, the following vectors and matrices are deduced. The vector for the system states for the **n-th** tank is:

$$x_n = \begin{bmatrix} S_{Sn} & S_{NH4n} & S_{NOXn} \end{bmatrix}^T \in R^{3}, \qquad \overline{n = 1,3}$$

$$x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T \in R^{9}$$

The vector for the input concentration is:

$$x_i = \begin{bmatrix} S_{Si} & S_{NH4i} & S_{NOXi} \end{bmatrix}^T \in R^{3}$$

The vector for the process control variables is:

$$u = \begin{bmatrix} S_{O1} & S_{O2} & S_{O3} \end{bmatrix}^T \in R^{3}$$

The vector for the system process rates for the **n-th** tank is:

$$PR_n = \begin{bmatrix} p_{1n} & p_{2n} & p_{3n} & p_{4n} \end{bmatrix}^T \in R^{4}$$

$$PR = \begin{bmatrix} P_1 & P_2 & P_3 \end{bmatrix}^T \in R^{12}$$

5.4.2.5 Simulation data and results of the Benchmark plant with reduced-ASM1 model

The MATLAB file "*BASM1_Simulation.m*" given in Appendix A.1 is used for simulation of the Benchmark plant with ASM1. From the reduced mass balance equations of the Benchmark plant with r-ASM1, the following vectors and matrices are deduced. The vector for the system states for the **n-th** tank is:

$$x_n = \begin{bmatrix} S_{Sn} & S_{NOn} & S_{NHn} \end{bmatrix}^T \in R^{3} \qquad \overline{n = 1,5}$$

$$x \in R^{15}$$

The vector for the input concentration is:

$$x_i = \begin{bmatrix} S_{Si} & S_{NOi} & S_{NHi} \end{bmatrix}^T \in R^{3}$$

The vector for the process control variables is:

$$u = \begin{bmatrix} S_{O1} & S_{O2} & S_{O3} & S_{O4} & S_{O5} \end{bmatrix}^T \in R^{5}$$

The vector for the system process rates for the **n-th** tank is:

$$PR_n = \begin{bmatrix} p_{1n} & p_{2n} & p_{3n} & p_{4n} \end{bmatrix}^T \in R^{4}$$

$$PR = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 \end{bmatrix}^T \in R^{20}$$

The matrix CR for the whole system is:

$$CR = \Delta t \begin{bmatrix} CR_1 & 0 & 0 & 0 & 0 \\ 0 & CR_2 & 0 & 0 & 0 \\ 0 & 0 & CR_3 & 0 & 0 \\ 0 & 0 & 0 & CR_4 & 0 \\ 0 & 0 & 0 & 0 & CR_5 \end{bmatrix} \in R^{20 \times 15}$$

where:

$$CR_n = \begin{bmatrix} -1.4925 & 0 & -0.08 \\ -1.4925 & -0.1722 & -0.08 \\ 0 & 4.1667 & -4.2467 \\ 1 & 0 & 0 \end{bmatrix} \in R^{4 \times 3} \qquad \overline{n = 1,5}$$

The matrix AR for the whole system is:

$$AR = \begin{bmatrix}
0.9039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0769 & 0 & 0 \\
0 & 0.9039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0769 & 0 \\
0 & 0 & 0.9039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0769 \\
0.0961 & 0 & 0 & 0.9039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.0961 & 0 & 0 & 0.9039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.0961 & 0 & 0 & 0.9039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.0721 & 0 & 0 & 0.9279 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0.9279 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0.9279 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0.9279 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0.9279 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0.9279 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0.9279 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0.9279 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0721 & 0 & 0 & 0.9279
\end{bmatrix} \in R^{15 \times 15}$$

The matrix BR for the whole system is:

$$BR = \begin{bmatrix}
0.0192 & 0 & 0 \\
0 & 0.0231 & 0 \\
0 & 0 & 0.0192 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix} \in R^{15 \times 3}$$

The simulation results of the wastewater components concentrations for the plants' five tanks are shown in Figures (5.18 - 5.22).
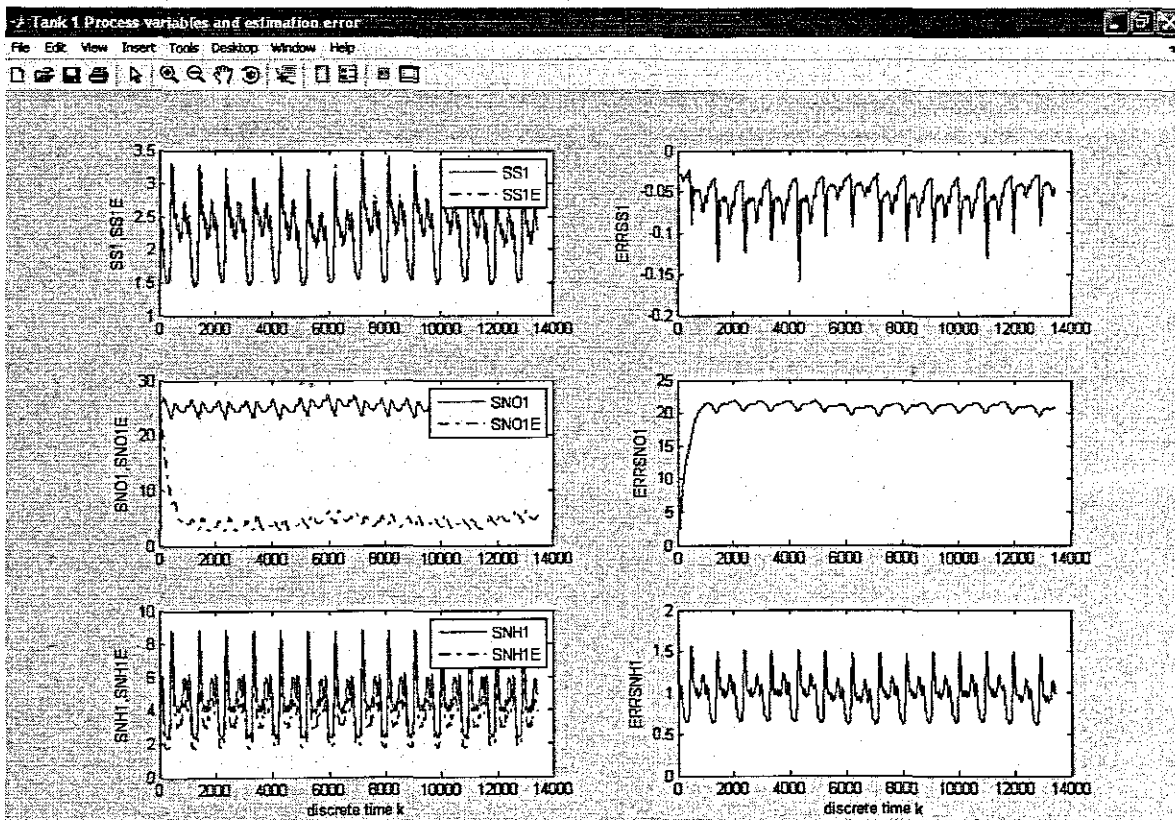
Figure 5.18: Benchmark_ASM1 process variables and estimation errors for Tank1
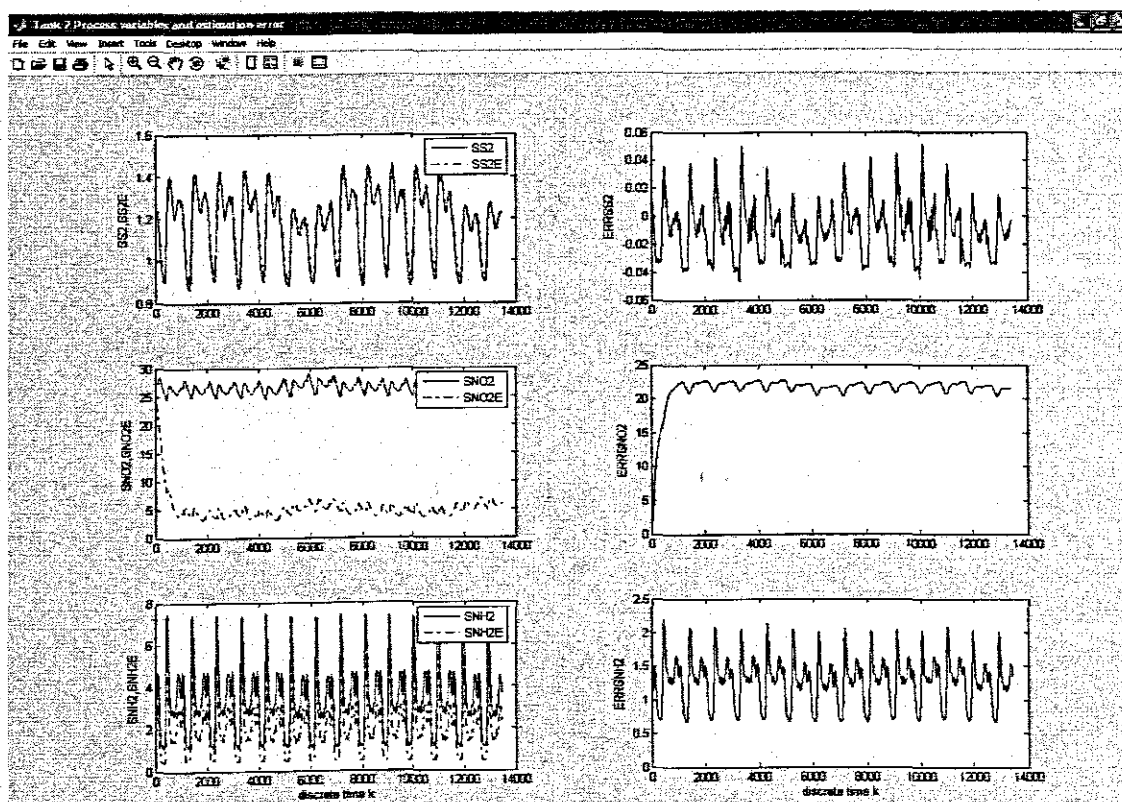
Figure 5.19: Benchmark_ASM1 process variables and estimation errors for Tank2

144
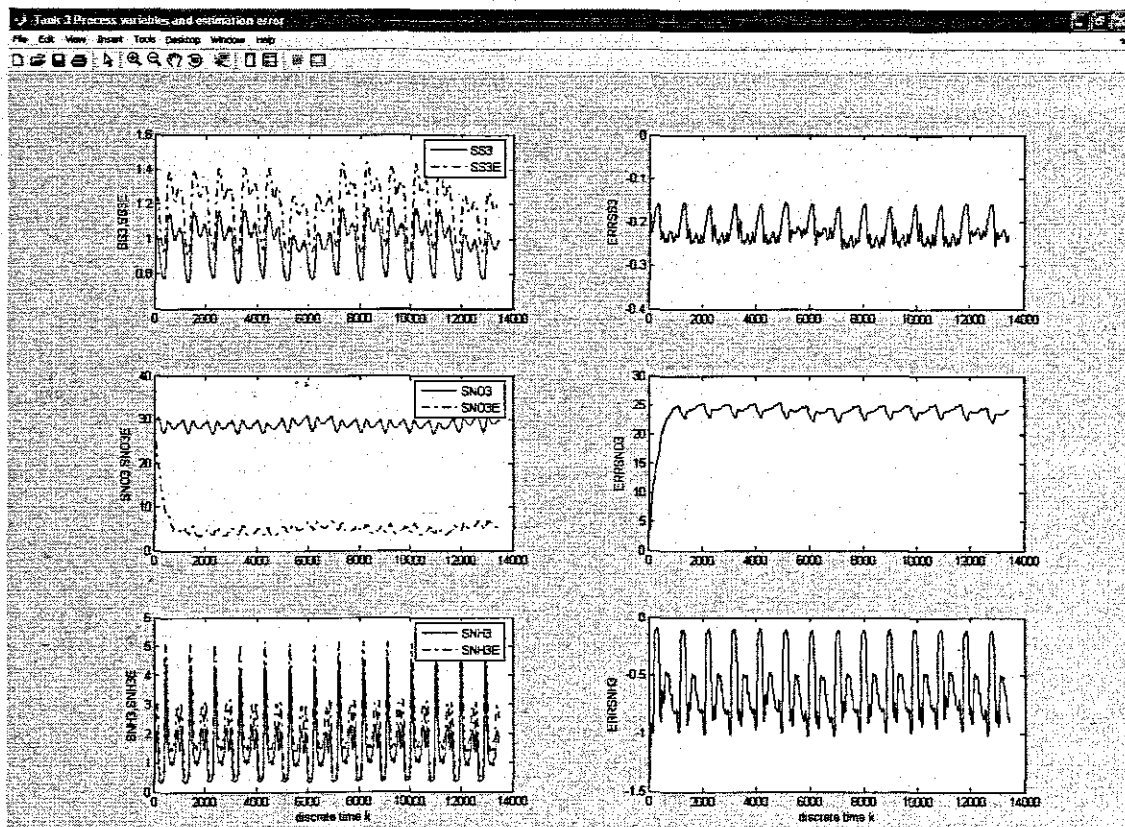
Figure 5.20: Benchmark_ASM1 process variables and estimation errors for Tank3
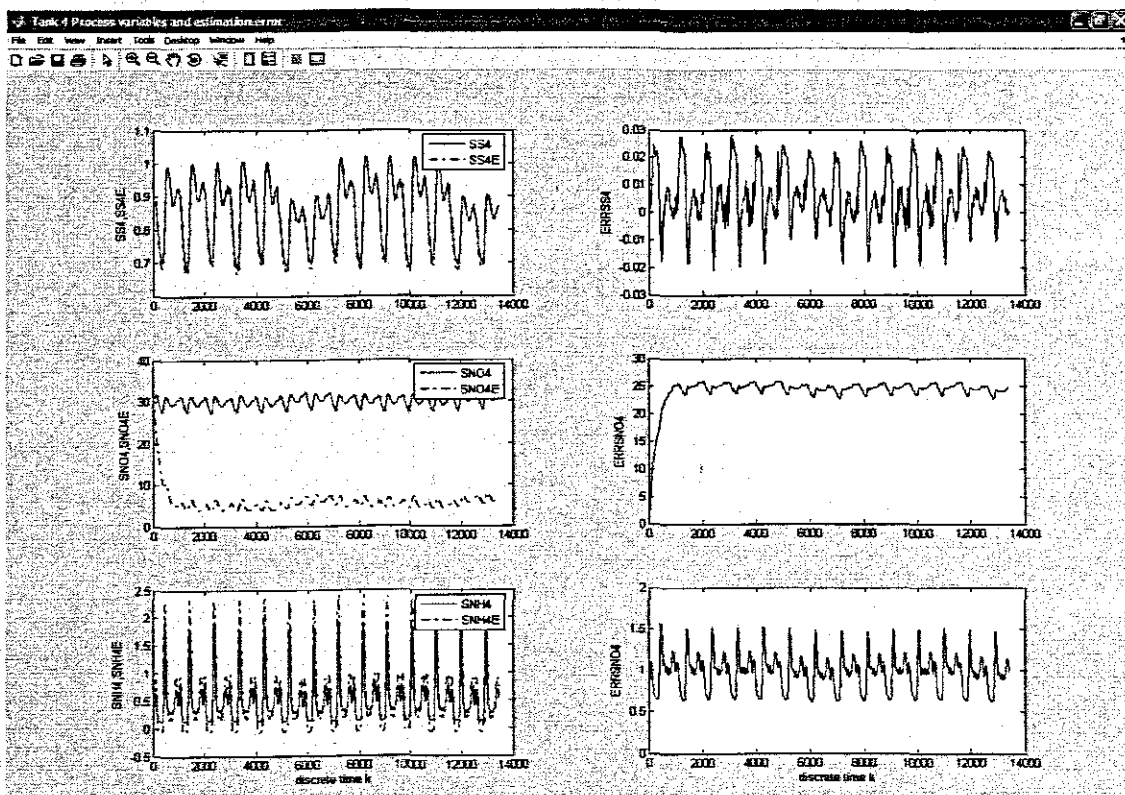
Figure 5.21: Benchmark_ASM1 process variables and estimation errors for Tank4

145
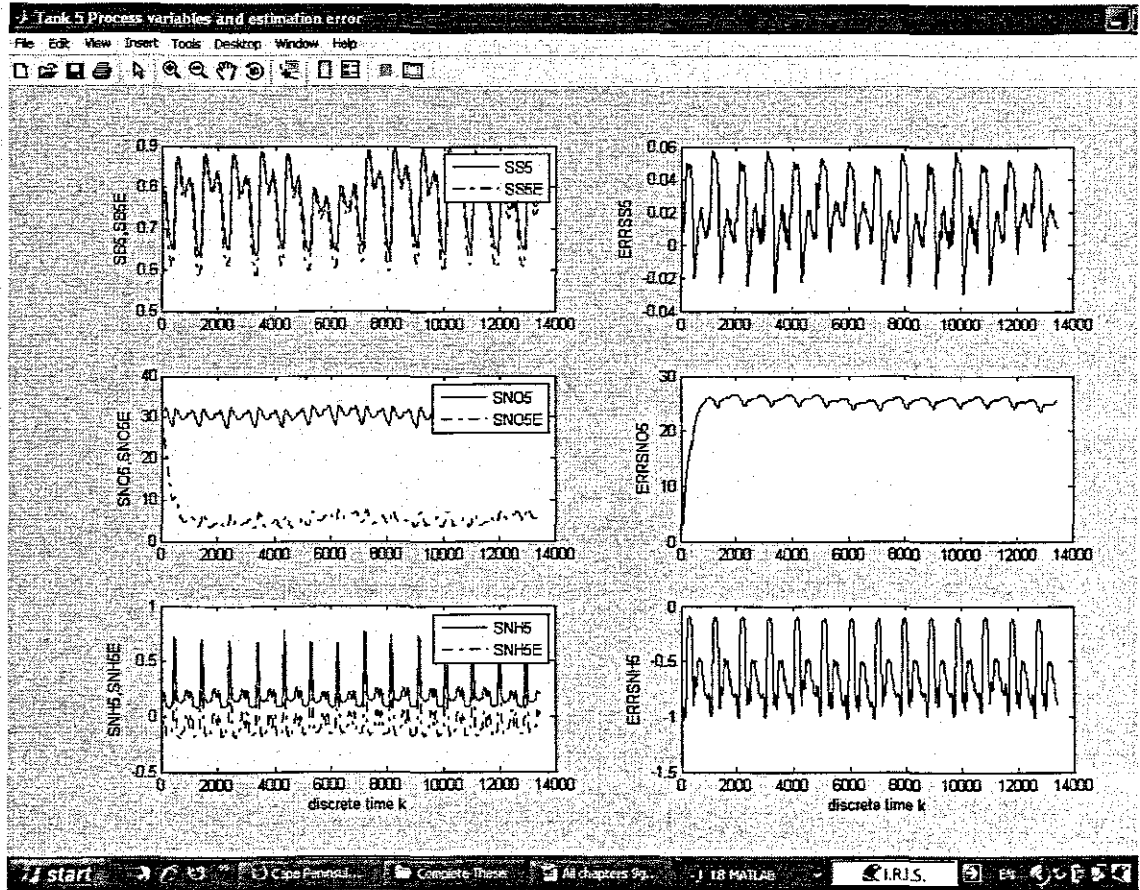
Figure 5.22: Benchmark_ASM1 Tank 5 process variables and estimation errors

5.4.2.6 Simulation data for the Benchmark plant with reduced-ASM2 model

The MATLAB file "*BASM2_Simulation*.m" given in Appendix B.1 is used for simulation of the Benchmark plant with ASM2.

From the reduced mass balance equations of the Benchmark plant with r-ASM2, the following vectors and matrices are deduced.

The vector for the system states for the **n-th** tank is:

$$x_n = [S_{Fn} \quad S_{An} \quad S_{NH4n} \quad S_{NO3n} \quad S_{PO4n} \quad X_{Sn} \quad X_{PPn} \quad X_{PHAn}]^T \in R^8 \qquad \overline{n = 1,5}$$

$$x = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5]^T \in R^{40}$$

The vector for the input concentration is:

$$x_i = [S_{Fi} \quad S_{Ai} \quad S_{NH4i} \quad S_{NO3i} \quad S_{PO4i} \quad X_{Si} \quad X_{PPi} \quad X_{PHAi}]^T \in R^8$$

The vector for the process control variables is:

$$u = [S_{O1} \quad S_{O2} \quad S_{O3} \quad S_{O4} \quad S_{O5}]^T \in R^5$$

The vector for the system process rates for the **n-th** tank is:

146

$$PR_n = \begin{bmatrix} p_{1n} & p_{2n} & p_{3n} & p_{4n} & p_{5n} & p_{6n} & p_{7n} & p_{8n} & p_{9n} & p_{10n} & p_{11n} & p_{12n} \end{bmatrix}^T \in R^{12}$$

$$PR = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 \end{bmatrix}^T \in R^{60}$$

5.4.2.7 Simulation data for the Benchmark plant with reduced-ASM2d model

The MATLAB file *"BASM2d_Simulation.m"* given in Appendix C.1 is used for simulation of the Benchmark plant with ASM2d.

From the reduced mass balance equations of the Benchmark plant with r-ASM2d, the following vectors and matrices are deduced.

The vector for the system states for the **n-th** tank is:

$$x_n = \begin{bmatrix} S_{Fn} & S_{An} & S_{NH4n} & S_{NO3n} & S_{PO4n} & X_{Sn} & X_{PPn} & X_{PHAn} \end{bmatrix}^T \in R^8 \quad \overline{n=1,5}$$

$$x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix}^T \in R^{40}$$

The vector for the input concentration is:

$$x_i = \begin{bmatrix} S_{Fi} & S_{Ai} & S_{NH4i} & S_{NO3i} & S_{PO4i} & X_{Si} & X_{PPi} & X_{PHAi} \end{bmatrix}^T \in R^8$$

The vector for the process control variables is:

$$u = \begin{bmatrix} S_{O1} & S_{O2} & S_{O3} & S_{O4} & S_{O5} \end{bmatrix}^T \in R^5$$

The vector for the system process rates for the **n-th** tank is:

$$PR_n = \begin{bmatrix} p_{1n} & p_{2n} & p_{3n} & p_{4n} & p_{5n} & p_{6n} & p_{7n} & p_{8n} & p_{9n} & p_{10n} & p_{11n} & p_{12n} & p_{13n} & p_{14n} \end{bmatrix}^T \in R^{14}$$

$$PR = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 \end{bmatrix}^T \in R^{70}$$

5.4.2.8 Simulation data for the Benchmark plant with reduced-ASM3 model

The MATLAB file *"BASM3_Simulation.m"* given in Appendix D.1 is used for simulation of the Benchmark plant with ASM3.

From the reduced mass balance equations of the Benchmark plant with r-ASM3, the following vectors and matrices are deduced.

The vector for the system states for the **n-th** tank is:

$$x_n = \begin{bmatrix} S_{Sn} & S_{NH4n} & S_{NOXn} \end{bmatrix}^T \in R^3 \quad \overline{n=1,5}$$

$$x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix}^T \in R^{15}$$

The vector for the input concentration is:

$$x_i = \begin{bmatrix} S_{Si} & S_{NH4i} & S_{NOXi} \end{bmatrix}^T \in R^3$$

The vector for the process control variables is:

147

$$u = [S_{01} \quad S_{02} \quad S_{03} \quad S_{04} \quad S_{05}]^T \in R^5$$

The vector for the system process rates for the **n-th** tank is:

$$PR_n = [p_{1n} \quad p_{2n} \quad p_{3n} \quad p_{4n}]^T \in R^4$$

$$PR = [P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5]^T \in R^{20}$$

5.4.3 Discussion of the results

The results for the variables for the reduced mass balance models are plotted on the same graph with corresponding process variables from the full mass balance model. As can be seen in Figures (18 - 22), the trajectories of the full models and reduced model display common behaviour which are very close to each other.

5.5 Conclusion

In this section, the state space models of the Athlone and Benchmark plants with reduced biological models are implemented on the MATLAB simulation environment. Before the reduced models are used in simulation, their parameters are firstly calculated based on data from the full model simulations. The estimated trajectories of the reduced models parameters are used in Chapter 6 for optimization of the control trajectories.

CHAPTER SIX
DEVELOPMENT OF METHOD FOR OPTIMAL CONTROL CALCULATION FOR THE ACTIVATED SLUDGE PROCESSES

6.1   Introduction

The purpose of this project is to develop a method for calculation of the optimal dissolved oxygen trajectory as a part of the control strategy to be implemented on the activated sludge process (ASP) of the Benchmark and Athlone wastewater treatment plants. The control strategy is based on the manipulation of DO in the aeration tanks. The objective of the control strategy is to calculate optimal DO trajectory that causes a process to satisfy the physical constraints and minimize the performance criterion (Katebi, 1998). The obtained trajectory is then used as a sequence of set points for the controller to track. In order to formulate the optimal control problem in the ASP, it is necessary to represent the controlled processes mathematically and to describe the aim of optimization mathematically by some criterion (Lindberg, 1997).

The optimal control strategy is described in Chapter 3 while mathematical models of the Benchmark and Athlone ASPs are developed in Chapter 4. In this project, all 4 IWA models (ASM1, ASM2, ASM2d and ASM3) are used to describe biological activities inside the reactors. The models are all used because each model describes different ASP setup. To decrease complexities of the problem, biological models are reduced in the previous Chapter 5 before their use. For optimal control calculation an optimal control problem is formulated for Benchmark with reduced ASM1, Benchmark with reduced ASM2, Benchmark with reduced ASM2d, Benchmark with reduced ASM3, Athlone with reduced ASM1, Athlone with reduced ASM2, Athlone with reduced ASM2d and Athlone with reduced ASM3.

The control strategy and the optimal control problem formulation are common for all these different process setups. In this chapter, the common optimal control problem is formulated and solved based on the principles of decomposition and coordination.

6.1.1   Formulation of the optimal control problem for Benchmark and Athlone plants

The optimal control problem is to find the trajectory of the dissolved oxygen concentration as a control variable $u(k), k = \overline{0, K-1}$ in such a way that the deviation of the process variables $x(k), k = \overline{1, K}$ from the effluent set points $x^{sp}(k), k = \overline{1, K}$ is minimal. The effluent set points are

149

determined by the requirements of the municipality and standards for quality of the effluent leaving the wastewater treatment plants.

As developed in Chapter 4, the plants are described by the following discrete state space model:

$$x(k+1) = Ax(k) + C^T P(x,u,k) + Bx_i(k) \qquad x(0) = x_0, \ k = \overline{0,K-1}$$

The following discrete quadratic cost functional is selected as the criterion for ASP optimization:

$$J(x,u,k) = \frac{1}{2} \| x^{SP}(K) - x(K) \|^2_Q + \frac{1}{2} \sum_{k=0}^{K-1} \| x^{SP}(k) - x(k) \|^2_Q + \| u(k) \|^2_R \tag{6.1}$$

The constraints of the state and control variables are described mathematically through the following equations:

$$x_{min}(k) \le x(k) \le x_{max}(k), \ k = \overline{0,K} \tag{6.2}$$

$$u_{min}(k) \le u(k) \le u_{max}, \ k = \overline{0,K-1} \tag{6.3}$$

where k is the number of steps in the optimization interval, Q and R are positively definite square matrices. A minimum value of the objective functional reflects a minimum deviation of the process variables from the effluent set points. The optimal control problem is therefore to find the trajectory of the control variable $u(k), k = \overline{0,K-1}$ that minimises the objective functional:

$$J(x,u,k) = \frac{1}{2} \| x^{SP}(K) - x(K) \|^2_Q + \frac{1}{2} \sum_{k=0}^{K-1} \| x^{SP}(k) - x(k) \|^2_Q + \| u(k) \|^2_R \rightarrow min$$

subject to the state equations, the physical and the process limitations over the state and control variables:

$$x(k+1) = Ax(k) + C^T P(x,u,k) + Bx_i(k) \qquad x(0) = x_0, \ k = \overline{0,K-1}$$

$$x_{min}(k) \le x(k) \le x_{max}(k), \ k = \overline{0,K}$$

$$u_{min}(k) \le u(k) \le u_{max}, \ k = \overline{0,K-1}$$

6.1.2 Formulation of the Lagrangian for Benchmark and Athlone plants

The state equations and the cost functional are put together to form the Lagrangian functional.

- The first step is to reformulate the state equations so they are equal to zero

$$-x(k+1) + Ax(k) + C^T P(x,u,k) + Bx_i(k) = 0 \tag{6.4}$$

- The second step is to combine the state equations and the cost functional by means of the coordinating variables known as the Lagrange multipliers, λ

$$L = \frac{1}{2} // x^{SP}(K) - x(K) //^2_Q +$$
$$+ \sum_{k=0}^{K-1} \left\{ \frac{1}{2} \left[// x^{SP}(k) - x(k) //^2_Q + // u(k) //^2_R \right] + \lambda(k)^T \left[-x(k+1) + Ax(k) + C^T P(x,u,k) + Bx_i(k) \right] \right\} \tag{6.5}$$

- Finally the Lagrangian is simplified and written as follows

$$L = \frac{1}{2} // x^{SP}(k) - x(k) //^2_Q +$$
$$+ \sum_{k=0}^{K-1} \left\{ \frac{1}{2} \left[// x^{SP}(k) - x(k) //^2_Q + // u(k) //^2_R \right] - \lambda(k)^T x(k+1) + \lambda(k)^T Ax(k) + \lambda(k)^T C^T P(x,u,k) + \lambda(k)^T Bx_i(k) \right\} \tag{6.6}$$

6.1.3 Application of calculus of variation technique

Applying calculus of variation method, firstly the Lagrangian is partially differentiated with respect to the state, control and conjugate variables. Then the resulting partial differential equations are set to zero as shown below. Solution to these equations results in the optimal trajectories of the state, control and conjugate variables.

$$\frac{\partial L(k)}{\partial x(k)} = 0 = e_x(k) \quad \text{where } k = \overline{1,K} \tag{6.7}$$

$$\frac{\partial L(k)}{\partial u(k)} = 0 = e_u(k) \quad \text{where } k = \overline{0,K-1} \tag{6.8}$$

$$\frac{\partial L(k)}{\partial \lambda(k)} = 0 = e_\lambda(k) \quad \text{where } k = \overline{0,K-1} \tag{6.9}$$

Using chain rule, partial differentiation of Equation (6.6) with respect to the state, control and conjugate variables results in the following equations:

$$\frac{\partial L}{\partial x(k)} = -Q\left[x^{SP}(k) - x(k) \right] - \lambda(k-1) + A^T \lambda(k) + \left(C^T \frac{\partial P(x,u,k)}{\partial x(k)} \right)^T \lambda(k) = e_x(k), \ k = \overline{1,K-1} \tag{6.10}$$

$$\frac{\partial L}{\partial u(k)} = R\left[u(k) \right] + \left(C^T \frac{\partial P(x,u,k)}{\partial u(k)} \right)^T \lambda(k) = e_u(k), \ k = \overline{0,K-1} \tag{6.11}$$

$$\frac{\partial L}{\partial \lambda(k)} = -x(k+1) + Ax(k) + C^T P(x,u,k) + Bx_i(k) = e_\lambda(k), \ k = \overline{0,K-1} \tag{6.12}$$

$$\frac{\partial L}{\partial x(K)} = -Q\left[x^{SP}(K) - x(K) \right] - \lambda(K-1), \ k = K \tag{6.13}$$

These equations need to be solved simultaneously in order to obtain the optimal solution. $P(x,u,k)$ depends on the used biological models (i.e. reduced ASM1, ASM2, ASM2d and ASM3) given in Chapter 4. The derivatives $\frac{\partial P(x,u,k)}{\partial x(k)}$ and $\frac{\partial P(x,u,k)}{\partial u(k)}$ can be calculated using the chain rule method.

6.1.3.1 Derivatives of the process rates according to the state vectors

Reduced-ASM1

These derivatives are based on process rates equations (4.63 – 4.66)

$$\frac{\partial P_1}{\partial S_S} = \hat{\mu}_H \cdot \frac{K_S}{(K_S + S_S)^2} \cdot \frac{S_O}{K_{O,H} + S_O} \cdot X_{B,H} \tag{6.14}$$

$$\frac{\partial P_2}{\partial S_S} = \hat{\mu}_H \cdot \frac{K_S}{(K_S + S_S)^2} \cdot \frac{K_{O,H}}{K_{O,H} + S_O} \cdot \frac{S_{NO}}{K_{NO} + S_{NO}} \cdot \eta_g \cdot X_{B,H} \tag{6.15}$$

$$\frac{\partial P_2}{\partial S_{NO}} = \hat{\mu}_H \cdot \frac{S_S}{K_S + S_S} \cdot \frac{K_{O,H}}{K_{O,H} + S_O} \cdot \frac{K_{NO}}{(K_{NO} + S_{NO})^2} \cdot \eta_g \cdot X_{B,H} \tag{6.16}$$

$$\frac{\partial P_3}{\partial S_{NH}} = \hat{\mu}_A \cdot \frac{K_{NH}}{(K_{NH} + S_{NH})^2} \cdot \frac{S_O}{K_{O,A} + S_O} \cdot X_{B,A} \tag{6.17}$$

$$\frac{\partial P_4}{\partial S_{NO}} = k_h \cdot \eta_h \cdot \frac{X_S / X_{B,H}}{K_X + X_S / X_{B,H}} \cdot \frac{K_{O,H}}{K_{O,H} + S_O} \cdot \frac{K_{NO}}{(K_{NO} + S_{NO})^2} \cdot X_{B,H} \tag{6.18}$$

Reduced-ASM2

These derivatives are based on process rates equations (4.67 – 4.78)

$$\frac{\partial P_1}{\partial X_S} = K_h \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_X X_H}{(K_X X_H + X_S)^2} \cdot X_H \tag{6.19}$$

$$\frac{\partial P_2}{\partial S_{NO3}} = K_h \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{-K_{NO3}}{(K_{NO3} + S_{NO3})^2} \cdot \frac{X_S / X_H}{K_X + X_S / X_H} \cdot X_H \tag{6.20}$$

$$\frac{\partial P_2}{\partial X_S} = K_h \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{K_X X_H}{(K_X X_H + X_S)^2} \cdot X_H \tag{6.21}$$

$$\frac{\partial P_3}{\partial S_{NO3}} = K_h \cdot \eta_{fe} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{-K_{NO3}}{(K_{NO3} + S_{NO3})^2} \cdot \frac{X_S / X_H}{K_X + X_S / X_H} \cdot X_H \tag{6.22}$$

$$\frac{\partial P_3}{\partial X_S} = K_h \cdot \eta_{fe} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{K_X X_H}{(K_X X_H + X_S)^2} \cdot X_H \tag{6.23}$$

$$\frac{\partial P_4}{\partial S_F} = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{2 S_F K_F + S_F^2}{(K_F + S_F)^2 (K_F + S_A)} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{6.24}$$

$$\frac{\partial P_4}{\partial S_A} = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{-S_F}{(K_F + S_A)^2} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{6.25}$$

$$\frac{\partial P_4}{\partial S_{NH4}} = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{K_F + S_A} \cdot \frac{K_{NH4}}{(K_{NH4} + S_{NH4})^2} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{6.26}$$

152

$$\frac{\partial P_4}{\partial S_{PO4}} = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{K_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{K_P}{(K_P + S_{PO4})^2} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \qquad (6.27)$$

$$\frac{\partial P_5}{\partial S_F} = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{-S_A}{(S_F + S_A)^2} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \qquad (6.28)$$

$$\frac{\partial P_5}{\partial S_A} = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{2 S_A K_A S_F + S_A^2 K_A + S_A^2 S_F}{[(K_A + S_A)(S_F + S_A)]^2} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \qquad (6.29)$$

$$\frac{\partial P_5}{\partial S_{NH4}} = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{K_{NH4}}{(K_{NH4} + S_{NH4})^2} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \qquad (6.30)$$

$$\frac{\partial P_5}{\partial S_{PO4}} = \mu_H \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{K_P}{(K_P + S_{PO4})^2} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \qquad (6.31)$$

$$\frac{\partial P_6}{\partial S_F} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{2 S_F K_A S_A + S_F^2 K_A + S_F^2 S_A}{[(K_A + S_F)(S_F + S_A)]^2} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \qquad (6.32)$$

$$\frac{\partial P_6}{\partial S_A} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{-S_F}{(S_F + S_A)^2} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \qquad (6.33)$$

$$\frac{\partial P_6}{\partial S_{NH4}} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{S_F + S_A} \cdot \frac{K_{NH4}}{(K_{NH4} + S_{NH4})^2} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \qquad (6.34)$$

$$\frac{\partial P_6}{\partial S_{NO3}} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{K_{NO3}}{(K_{NO3} + S_{NO3})^2} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \qquad (6.35)$$

$$\frac{\partial P_7}{\partial S_F} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{-S_A}{(S_F + S_A)^2} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot X_H \qquad (6.36)$$

$$\frac{\partial P_7}{\partial S_A} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{2 S_A K_A S_F + S_A^2 K_A + S_A^2 S_F}{[(K_A + S_A)(S_F + S_A)]^2} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} X_H \qquad (6.37)$$

$$\frac{\partial P_7}{\partial S_{NH4}} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{K_{NH4}}{(K_{NH4} + S_{NH4})^2} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot X_H \qquad (6.38)$$

$$\frac{\partial P_7}{\partial S_{NO3}} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{K_{NO3}}{(K_{NO3} + S_{NO3})^2} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot X_H \qquad (6.39)$$

$$\frac{\partial P_7}{\partial S_{PO4}} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{K_P}{(K_P + S_{PO4})^2} \cdot X_H \qquad (6.40)$$

$$\frac{\partial P_8}{\partial S_F} = q_{fe} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{K_{fe}}{(K_{fe} + S_F)^2} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \qquad (6.41)$$

$$\frac{\partial P_8}{\partial S_{NO3}} = q_{fe} \cdot \frac{K_{O2}}{K_{O2} + S_{O2}} \cdot \frac{-K_{NO3}}{(K_{NO3} + S_{NO3})^2} \cdot \frac{S_F}{K_{fe} + S_F} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \qquad (6.42)$$

$$\frac{\partial P_9}{\partial S_A} = q_{PHA} \cdot \frac{K_A}{(K_A + S_A)^2} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{PP}/X_{PAO}}{K_{PP} + X_{PP}/X_{PAO}} \cdot X_{PAO} \qquad (6.43)$$

$$\frac{\partial P_9}{\partial X_{PP}} = q_{PHA} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{K_{PP} X_{PAO}}{(K_{PP} X_{PAO} + X_{PP})^2} \cdot X_{PAO} \qquad (6.44)$$

$$\frac{\partial P_{10}}{\partial S_{PO4}} = q_{PP} \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{K_{PS}}{(K_{PS} + S_{PO4})^2} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA} + X_{PHA}/X_{PAO}} \cdot \frac{K_{MAX} - X_{PP}/X_{PAO}}{K_{IPP} + K_{MAX} - X_{PP}/X_{PAO}} \cdot X_{PAO} \qquad (6.45)$$

$$\frac{\partial P_{10}}{\partial X_{PP}} = q_{PP} \cdot \frac{S_{O2}}{K_{O2} + S_{O2}} \cdot \frac{S_{PO4}}{K_{PS} + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA} + X_{PHA}/X_{PAO}} \cdot \frac{-K_{IPP} X_{PAO}}{(K_{IPP} X_{PAO} + K_{MAX} X_{PAO} - X_{PP})^2} \cdot X_{PAO} \qquad (6.46)$$

153

$$\frac{\partial P_{10}}{\partial X_{PHA}} = q_{PP} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{PO4}}{K_{PS}+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{K_{PHA}X_{PAO}}{\left(K_{PHA}X_{PAO}+X_{PHA}\right)^2} \cdot \frac{K_{MAX}-X_{PP}/X_{PAO}}{K_{IPP}+K_{MAX}-X_{PP}/X_{PAO}} \cdot X_{PAO} \qquad (6.47)$$

$$\frac{\partial P_{11}}{\partial S_{NH4}} = \mu_{PAO} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NH4}}{\left(K_{NH4}+S_{NH4}\right)^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot X_{PAO} \qquad (6.48)$$

$$\frac{\partial P_{11}}{\partial S_{PO4}} = \mu_{PAO} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{K_P}{\left(K_P+S_{PO4}\right)^2} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot X_{PAO} \qquad (6.49)$$

$$\frac{\partial P_{11}}{\partial X_{PHA}} = \mu_{PAO} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{K_{PHA}X_{PAO}}{\left(K_{PHA}X_{PAO}+X_{PHA}\right)^2} \cdot X_{PAO} \qquad (6.50)$$

$$\frac{\partial P_{12}}{\partial S_{NH4}} = \mu_{AUT} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NH4}}{\left(K_{NH4}+S_{NH4}\right)^2} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT} \qquad (6.51)$$

$$\frac{\partial P_{12}}{\partial S_{PO4}} = \mu_{AUT} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{K_P}{\left(K_P+S_{PO4}\right)^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT} \qquad (6.52)$$

Reduced-ASM2d

These derivatives are based on process rates equations (4.69 – 4.92)

$$\frac{\partial P_1}{\partial X_S} = K_h \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_X X_H}{\left(K_X X_H+X_S\right)^2} \cdot X_H \qquad (6.53)$$

$$\frac{\partial P_2}{\partial S_{NO3}} = K_h \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{-K_{NO3}}{\left(K_{NO3}+S_{NO3}\right)^2} \cdot \frac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H \qquad (6.54)$$

$$\frac{\partial P_2}{\partial X_S} = K_h \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{K_X X_H}{\left(K_X X_H+X_S\right)^2} \cdot X_H \qquad (6.55)$$

$$\frac{\partial P_3}{\partial S_{NO3}} = K_h \cdot \eta_{fe} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{-K_{NO3}}{\left(K_{NO3}+S_{NO3}\right)^2} \cdot \frac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H \qquad (6.56)$$

$$\frac{\partial P_3}{\partial X_S} = K_h \cdot \eta_{fe} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{K_X X_H}{\left(K_X X_H+X_S\right)^2} \cdot X_H \qquad (6.57)$$

$$\frac{\partial P_4}{\partial S_F} = \mu_H \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{2S_F K_F+S_F{}^2}{\left(K_F+S_F\right)^2\left(K_F+S_A\right)} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.58)$$

$$\frac{\partial P_4}{\partial S_A} = \mu_H \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_F}{K_F+S_F} \cdot \frac{-S_F}{\left(K_F+S_A\right)^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.59)$$

$$\frac{\partial P_4}{\partial S_{NH4}} = \mu_H \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_F}{K_F+S_F} \cdot \frac{S_F}{K_F+S_A} \cdot \frac{K_{NH4}}{\left(K_{NH4}+S_{NH4}\right)^2} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.60)$$

$$\frac{\partial P_4}{\partial S_{PO4}} = \mu_H \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_F}{K_F+S_F} \cdot \frac{S_F}{K_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{K_P}{\left(K_P+S_{PO4}\right)^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.61)$$

$$\frac{\partial P_5}{\partial S_F} = \mu_H \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{-S_A}{\left(S_F+S_A\right)^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.62)$$

$$\frac{\partial P_5}{\partial S_A} = \mu_H \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{2S_A K_A S_F+S_A{}^2 K_A+S_A{}^2 S_F}{\left[\left(K_A+S_A\right)\left(S_F+S_A\right)\right]^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.63)$$

$$\frac{\partial P_5}{\partial S_{NH4}} = \mu_H \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_A}{S_F+S_A} \cdot \frac{K_{NH4}}{\left(K_{NH4}+S_{NH4}\right)^2} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.64)$$

$$\frac{\partial P_5}{\partial S_{PO4}} = \mu_H \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_A}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{K_P}{(K_P+S_{PO4})^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.65)$$

$$\frac{\partial P_6}{\partial S_F} = \mu_H \cdot \eta_{NO_3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{2S_F K_A S_A + S_F^2 K_A + S_F^2 S_A}{[(K_A+S_F)(S_F+S_A)]^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.66)$$

$$\frac{\partial P_6}{\partial S_A} = \mu_H \cdot \eta_{NO_3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_F}{K_F+S_F} \cdot \frac{-S_F}{(S_F+S_A)^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.67)$$

$$\frac{\partial P_6}{\partial S_{NH4}} = \mu_H \cdot \eta_{NO_3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_F}{K_F+S_F} \cdot \frac{S_F}{S_F+S_A} \cdot \frac{K_{NH4}}{(K_{NH4}+S_{NH4})^2} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.68)$$

$$\frac{\partial P_6}{\partial S_{NO3}} = \mu_H \cdot \eta_{NO_3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_F}{K_F+S_F} \cdot \frac{S_F}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{K_{NO3}}{(K_{NO3}+S_{NO3})^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.69)$$

$$\frac{\partial P_7}{\partial S_F} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{-S_A}{(S_F+S_A)^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot X_H \qquad (6.70)$$

$$\frac{\partial P_7}{\partial S_A} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{2S_A K_A S_F + S_A^2 K_A + S_A^2 S_F}{[(K_A+S_A)(S_F+S_A)]^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} X_H \qquad (6.71)$$

$$\frac{\partial P_7}{\partial S_{NH4}} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_A}{S_F+S_A} \cdot \frac{K_{NH4}}{(K_{NH4}+S_{NH4})^2} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot X_H \qquad (6.72)$$

$$\frac{\partial P_7}{\partial S_{NO3}} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_A}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{K_{NO3}}{(K_{NO3}+S_{NO3})^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot X_H \qquad (6.73)$$

$$\frac{\partial P_7}{\partial S_{PO4}} = \mu_H \cdot \eta_{NO3} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_A}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{K_P}{(K_P+S_{PO4})^2} \cdot X_H \qquad (6.74)$$

$$\frac{\partial P_8}{\partial S_F} = q_{fe} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{K_{fe}}{(K_{fe}+S_F)^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.75)$$

$$\frac{\partial P_8}{\partial S_{NO3}} = q_{fe} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{-K_{NO3}}{(K_{NO3}+S_{NO3})^2} \cdot \frac{S_F}{K_{fe}+S_F} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \qquad (6.76)$$

$$\frac{\partial P_9}{\partial S_A} = q_{PHA} \cdot \frac{K_A}{(K_A+S_A)^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{PP}/X_{PAO}}{K_{PP}+X_{PP}/X_{PAO}} \cdot X_{PAO} \qquad (6.77)$$

$$\frac{\partial P_9}{\partial X_{PP}} = q_{PHA} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{K_{PP}X_{PAO}}{(K_{PP}X_{PAO}+X_{PP})^2} \cdot X_{PAO} \qquad (6.78)$$

$$\frac{\partial P_{10}}{\partial S_{PO4}} = q_{PP} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{PS}}{(K_{PS}+S_{PO4})^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot \frac{K_{MAX}-X_{PP}/X_{PAO}}{K_{IPP}+K_{MAX}-X_{PP}/X_{PAO}} \cdot X_{PAO} \qquad (6.79)$$

$$\frac{\partial P_{10}}{\partial X_{PP}} = q_{PP} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{PO4}}{K_{PS}+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot \frac{-K_{IPP}X_{PAO}}{(K_{IPP}X_{PAO}+K_{MAX}X_{PAO}-X_{PP})^2} \cdot X_{PAO} \qquad (6.80)$$

$$\frac{\partial P_{10}}{\partial X_{PHA}} = q_{PP} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{PO4}}{K_{PS}+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{K_{PHA}X_{PAO}}{(K_{PHA}X_{PAO}+X_{PHA})^2} \cdot \frac{K_{MAX}-X_{PP}/X_{PAO}}{K_{IPP}+K_{MAX}-X_{PP}/X_{PAO}} \cdot X_{PAO} \qquad (6.81)$$

$$\frac{\partial P_{11}}{\partial S_{NH4}} = \mu_{PAO} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NH4}}{(K_{NH4}+S_{NH4})^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot X_{PAO} \qquad (6.82)$$

$$\frac{\partial P_{11}}{\partial S_{PO4}} = \mu_{PAO} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{K_P}{(K_P+S_{PO4})^2} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot X_{PAO} \qquad (6.83)$$

$$\frac{\partial P_{11}}{\partial X_{PHA}} = \mu_{PAO} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{K_{PHA}X_{PAO}}{(K_{PHA}X_{PAO}+X_{PHA})^2} \cdot X_{PAO} \qquad (6.84)$$

155

$$\frac{\partial P_{12}}{\partial S_{NH4}} = \mu_{PAO} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NH4}}{(K_{NH4}+S_{NH4})^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot X_{PAO} \cdot \eta_{NO3} \cdot$$
$$\frac{K_{O2}}{S_{O2}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \tag{6.85}$$

$$\frac{\partial P_{12}}{\partial S_{NO3}} = \mu_{PAO} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot X_{PAO} \cdot \eta_{NO3} \cdot$$
$$\frac{K_{O2}}{S_{O2}} \cdot \frac{K_{NO3}}{(K_{NO3}+S_{NO3})^2} \tag{6.86}$$

$$\frac{\partial P_{12}}{\partial S_{PO4}} = \mu_{PAO} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{K_P}{(K_P+S_{PO4})^2} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot X_{PAO} \cdot \eta_{NO3} \cdot$$
$$\frac{K_{O2}}{S_{O2}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \tag{6.87}$$

$$\frac{\partial P_{12}}{\partial X_{PHA}} = \mu_{PAO} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{K_{PHA}X_{PAO}}{(K_{PHA}X_{PAO}+X_{PHA})^2} \cdot X_{PAO} \cdot \eta_{NO3} \cdot$$
$$\frac{K_{O2}}{S_{O2}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \tag{6.88}$$

$$\frac{\partial P_{13}}{\partial S_{NH4}} = \mu_{AUT} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NH4}}{(K_{NH4}+S_{NH4})^2} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT} \tag{6.89}$$

$$\frac{\partial P_{13}}{\partial S_{PO4}} = \mu_{AUT} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{K_P}{(K_P+S_{PO4})^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT} \tag{6.90}$$

$$\frac{\partial P_{14}}{\partial S_{NH4}} = \mu_{AUT} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NH4}}{(K_{NH4}+S_{NH4})^2} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT} \cdot \eta_{NO3} \cdot \frac{K_{O2}}{S_{O2}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \tag{6.91}$$

$$\frac{\partial P_{14}}{\partial S_{NO3}} = \mu_{AUT} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT} \cdot \eta_{NO3} \cdot \frac{K_{O2}}{S_{O2}} \cdot \frac{K_{NO3}}{(K_{NO3}+S_{NO3})^2} \tag{6.92}$$

$$\frac{\partial P_{14}}{\partial S_{PO4}} = \mu_{AUT} \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{K_P}{(K_P+S_{PO4})^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT} \cdot \eta_{NO3} \cdot \frac{K_{O2}}{S_{O2}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \tag{6.93}$$

Reduced-ASM3

These derivatives are based on process rates equations (4.93 – 4.96)

$$\frac{\partial P_1}{\partial x} = 0 \tag{6.94}$$

$$\frac{\partial P_2}{\partial S_{NH4}} = \mu_H \cdot \frac{S_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NH4}}{(K_{NH4}+S_{NH4})^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{STO}/X_H}{K_{STO}+X_{STO}/X_H} \cdot X_H \tag{6.95}$$

$$\frac{\partial P_3}{\partial S_{NH4}} = \mu_H \cdot \eta_{NOX} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{S_{NOX}}{K_{NOX}+S_{NOX}} \cdot \frac{K_{NH4}}{(K_{NH4}+S_{NH4})^2} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{STO}/X_H}{K_{STO}+X_{STO}/X_H} \cdot X_H \tag{6.96}$$

$$\frac{\partial P_3}{\partial S_{NOX}} = \mu_H \cdot \eta_{NOX} \cdot \frac{K_{O2}}{K_{O2}+S_{O2}} \cdot \frac{K_{NOX}}{(K_{NOX}+S_{NOX})^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{STO}/X_H}{K_{STO}+X_{STO}/X_H} \cdot X_H \tag{6.97}$$

$$\frac{\partial P_4}{\partial S_{NH4}} = \mu_A \cdot \frac{S_{O2}}{K_{A,O2}+S_{O2}} \cdot \frac{K_{A,NH4}}{(K_{A,NH4}+S_{NH4})^2} \cdot \frac{S_{ALK}}{K_{A,ALK}+S_{ALK}} \cdot X_A \tag{6.98}$$

6.1.3.2 Derivatives of the process rates according to the control vectors

Reduced-ASM1

These derivatives are based on process rates equations (4.63 – 4.66)

$$\frac{\partial P_1}{\partial S_O} = \hat{\mu}_H \cdot \frac{S_S}{K_S + S_S} \cdot \frac{K_{O,H}}{(K_{O,H} + S_O)^2} \cdot X_{B,H} \tag{6.99}$$

$$\frac{\partial P_2}{\partial S_O} = \hat{\mu}_H \cdot \frac{S_S}{K_S + S_S} \cdot \frac{-K_{O,H}}{(K_{O,H} + S_O)^2} \cdot \frac{S_{NO}}{K_{NO} + S_{NO}} \cdot \eta_g \cdot X_{B,H} \tag{6.100}$$

$$\frac{\partial P_3}{\partial S_O} = \hat{\mu}_A \cdot \frac{S_{NH}}{K_{NH} + S_{NH}} \cdot \frac{K_{O,A}}{(K_{O,A} + S_O)^2} \cdot X_{B,A} \tag{6.101}$$

$$\frac{\partial P_4}{\partial S_O} = k_h \cdot \frac{X_S / X_{B,H}}{K_X + X_S / X_{B,H}} \left[\frac{K_{O,H}}{(K_{O,H} + S_O)^2} + \eta_h \cdot \frac{-K_{O,H}}{(K_{O,H} + S_O)^2} \cdot \frac{S_{NO}}{K_{NO} + S_{NO}} \right] X_{B,H} \tag{6.102}$$

Reduced-ASM2

These derivatives are based on process rates equations (4.67 – 4.78)

$$\frac{\partial P_1}{\partial S_{O2}} = K_h \cdot \frac{K_{O2}}{(K_{O2} + S_{O2})^2} \cdot \frac{X_S / X_H}{K_X + X_S / X_H} \cdot X_H \tag{6.103}$$

$$\frac{\partial P_2}{\partial S_{O2}} = K_h \cdot \eta_{NO3} \cdot \frac{-K_{O2}}{(K_{O2} + S_{O2})^2} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{X_S / X_H}{K_X + X_S / X_H} \cdot X_H \tag{6.104}$$

$$\frac{\partial P_3}{\partial S_{O2}} = K_h \cdot \eta_{fe} \cdot \frac{-K_{O2}}{(K_{O2} + S_{O2})^2} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{X_S / X_H}{K_X + X_S / X_H} \cdot X_H \tag{6.105}$$

$$\frac{\partial P_4}{\partial S_{O2}} = \mu_H \cdot \frac{K_{O2}}{(K_{O2} + S_{O2})^2} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{K_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{6.106}$$

$$\frac{\partial P_5}{\partial S_{O2}} = \mu_H \cdot \frac{K_{O2}}{(K_{O2} + S_{O2})^2} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{6.107}$$

$$\frac{\partial P_6}{\partial S_{O2}} = \mu_H \cdot \eta_{NO3} \cdot \frac{-K_{O2}}{(K_{O2} + S_{O2})^2} \cdot \frac{S_F}{K_F + S_F} \cdot \frac{S_F}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{6.108}$$

$$\frac{\partial P_7}{\partial S_{O2}} = \mu_H \cdot \eta_{NO3} \cdot \frac{-K_{O2}}{(K_{O2} + S_{O2})^2} \cdot \frac{S_A}{K_A + S_A} \cdot \frac{S_A}{S_F + S_A} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot X_H \tag{6.109}$$

$$\frac{\partial P_8}{\partial S_{O2}} = q_{fe} \cdot \frac{-K_{O2}}{(K_{O2} + S_{O2})^2} \cdot \frac{K_{NO3}}{K_{NO3} + S_{NO3}} \cdot \frac{S_F}{K_{fe} + S_F} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_H \tag{6.110}$$

$$\frac{\partial P_9}{\partial S_{O2}} = 0 \tag{6.111}$$

$$\frac{\partial P_{10}}{\partial S_{O2}} = q_{PP} \cdot \frac{K_{O2}}{(K_{O2} + S_{O2})^2} \cdot \frac{S_{PO4}}{K_{PS} + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{X_{PHA} / X_{PAO}}{K_{PHA} + X_{PHA} / X_{PAO}} \cdot \frac{K_{MAX} - X_{PP} / X_{PAO}}{K_{IPP} + K_{MAX} - X_{PP} / X_{PAO}} \cdot X_{PAO} \tag{6.112}$$

$$\frac{\partial P_{11}}{\partial S_{O2}} = \mu_{PAO} \cdot \frac{K_{O2}}{(K_{O2} + S_{O2})^2} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{X_{PHA} / X_{PAO}}{K_{PHA} + X_{PHA} / X_{PAO}} \cdot X_{PAO} \tag{6.113}$$

$$\frac{\partial P_{12}}{\partial S_{O2}} = \mu_{AUT} \cdot \frac{K_{O2}}{(K_{O2} + S_{O2})^2} \cdot \frac{S_{NH4}}{K_{NH4} + S_{NH4}} \cdot \frac{S_{PO4}}{K_P + S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK} + S_{ALK}} \cdot X_{AUT} \tag{6.114}$$

Reduced-ASM2d

These derivatives are based on process rates equations (4.79 – 4.92)

$$\frac{\partial P_1}{\partial S_{O2}} = K_h \cdot \frac{K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H \tag{6.115}$$

$$\frac{\partial P_2}{\partial S_{O2}} = K_h \cdot \eta_{NO3} \cdot \frac{-K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H \tag{6.116}$$

$$\frac{\partial P_3}{\partial S_{O2}} = K_h \cdot \eta_{fe} \cdot \frac{-K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{X_S/X_H}{K_X+X_S/X_H} \cdot X_H \tag{6.117}$$

$$\frac{\partial P_4}{\partial S_{O2}} = \mu_H \cdot \frac{K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{S_F}{K_F+S_F} \cdot \frac{S_F}{K_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \tag{6.118}$$

$$\frac{\partial P_5}{\partial S_{O2}} = \mu_H \cdot \frac{K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_A}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \tag{6.119}$$

$$\frac{\partial P_6}{\partial S_{O2}} = \mu_H \cdot \eta_{NO_3} \cdot \frac{-K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{S_F}{K_F+S_F} \cdot \frac{S_F}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \tag{6.120}$$

$$\frac{\partial P_7}{\partial S_{O2}} = \mu_H \cdot \eta_{NO3} \cdot \frac{-K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{S_A}{K_A+S_A} \cdot \frac{S_A}{S_F+S_A} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot X_H \tag{6.121}$$

$$\frac{\partial P_8}{\partial S_{O2}} = q_{fe} \cdot \frac{-K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{K_{NO3}}{K_{NO3}+S_{NO3}} \cdot \frac{S_F}{K_{fe}+S_F} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_H \tag{6.122}$$

$$\frac{\partial P_9}{\partial S_{O2}} = 0 \tag{6.123}$$

$$\frac{\partial P_{10}}{\partial S_{O2}} = q_{PP} \cdot \frac{K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{S_{PO4}}{K_{PS}+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot \frac{K_{MAX}-X_{PP}/X_{PAO}}{K_{IPP}+K_{MAX}-X_{PP}/X_{PAO}} \cdot X_{PAO} \tag{6.124}$$

$$\frac{\partial P_{11}}{\partial S_{O2}} = \mu_{PAO} \cdot \frac{K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot X_{PAO} \tag{6.125}$$

$$\frac{\partial P_{12}}{\partial S_{O2}} = \mu_{PAO} \cdot \frac{-K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{X_{PHA}/X_{PAO}}{K_{PHA}+X_{PHA}/X_{PAO}} \cdot \eta_{NO3} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \tag{6.126}$$

$$\frac{\partial P_{13}}{\partial S_{O2}} = \mu_{AUT} \cdot \frac{K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT} \tag{6.127}$$

$$\frac{\partial P_{14}}{\partial S_{O2}} = \mu_{AUT} \cdot \frac{-K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{PO4}}{K_P+S_{PO4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot X_{AUT} \cdot \eta_{NO3} \cdot \frac{S_{NO3}}{K_{NO3}+S_{NO3}} \tag{6.128}$$

Reduced-ASM3

These derivatives are based on process rates equations (4.93 – 4.96)

$$\frac{\partial P_1}{\partial S_{O2}} = 0 \tag{6.129}$$

$$\frac{\partial P_2}{\partial S_{O2}} = \mu_H \cdot \frac{K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{STO}/X_H}{K_{STO}+X_{STO}/X_H} \cdot X_H \tag{6.130}$$

$$\frac{\partial P_3}{\partial S_{O2}} = \mu_H \cdot \eta_{NOX} \cdot \frac{-K_{O2}}{(K_{O2}+S_{O2})^2} \cdot \frac{S_{NOX}}{K_{NOX}+S_{NOX}} \cdot \frac{S_{NH4}}{K_{NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{ALK}+S_{ALK}} \cdot \frac{X_{STO}/X_H}{K_{STO}+X_{STO}/X_H} \cdot X_H \tag{6.131}$$

$$\frac{\partial P_4}{\partial S_{O2}} = \mu_A \cdot \frac{K_{A,O2}}{(K_{A,O2}+S_{O2})^2} \cdot \frac{S_{NH4}}{K_{A,NH4}+S_{NH4}} \cdot \frac{S_{ALK}}{K_{A,ALK}+S_{ALK}} \cdot X_A \tag{6.132}$$

158

6.1.4 Decomposition of the optimal control problem

The solutions to Equations (6.10 – 6.13) where the derivatives of the process rates are given by equations in the previous point determine solution to the original optimal control problem. These equations are with non linear dynamics and with many variables and therefore numerical and decomposition techniques are used to find the solution. To solve these equations the problem is first decomposed in time. This type of decomposition is based on the prediction of the conjugate variables. These variables appear in the necessary conditions for optimality of all other variables. The optimal solution for all other variables depends on the optimal solution for the conjugate variables.

The problem solution is proposed to be done in a two level hierarchy where the first level task is to calculate the state and control variables at separate time moments and coordinating variables are selected and improved in the second level. The calculating structure is shown in Figure 6.1. The state, control and conjugate variables are determined through equations (6.133 - 6.135).
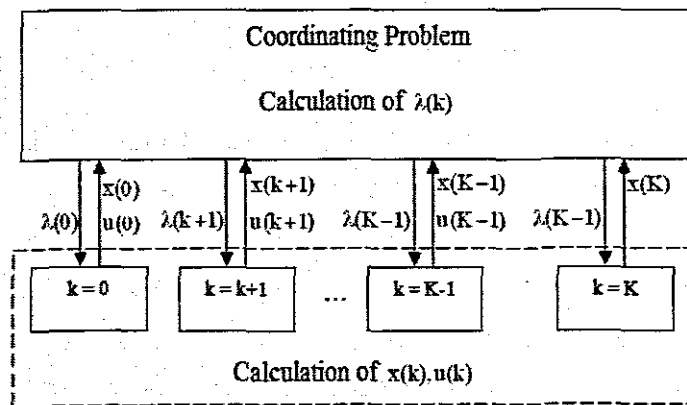
Figure 6.1: Two level hierarchical structure

6.1.5 Algorithm for calculation based on the gradient method

- First the coordinator determines values of the coordinating conjugate variables, $\lambda^i(k) = \lambda(k), k = \overline{0,K-1}$ where $i=1$

- Then the lower level determines the values of the state and control variables based on the calculated values of the coordinating variables. The problem in the lower level is decomposed into multiple sub problems. These sub problems are solved by gradient procedures of the following kind:

$$x^{i,q+1}(k) = x^{i,q}(k) - \alpha_x e_x^{i,q}(k) \quad \text{where } k = \overline{1,K-1} \tag{6.133}$$

159

$$u^{\ell,q+1}(k) = u^{\ell,q}(k) - \alpha_u e_x^{\ell,q}(k) \quad \text{where } k = \overline{1,K-1} \tag{6.134}$$

$$x^{\ell+1}(K) = x^{sp}(K) - Q^{-1}\lambda^{\ell}(K-1) \quad \text{where } k = K \tag{6.135}$$

where $e_x(k) = \dfrac{\partial L}{\partial x(k)}$, $e_u(k) = \dfrac{\partial L}{\partial u(k)}$, q and ℓ are the iteration indexes for the gradient procedure, α_x and α_u are the steps of the gradient procedures. Steps of the gradient procedures are selected small in order to achieve convergence of the calculations. Calculation of the gradient procedures starts from q = 1 and ℓ = 1 with initial values for the control trajectory.

The values of the gradients are used to determine how close the solution is to the optimal one under given values of the coordinating variables. In order to do this evaluation a norm of the gradients is calculated and compared with some small positive number close to zero.

$$\left\| e_x^{\ell,q}(k) \right\| \le \varepsilon_x, \varepsilon_x > 0, \varepsilon_x < 0 \qquad \text{where } \varepsilon_x = 0.01 \tag{6.136}$$

$$\left\| e_u^{\ell,q}(k) \right\| \le \varepsilon_u, \varepsilon_u > 0, \varepsilon_u < 0 \qquad \text{where } \varepsilon_u = 0.01 \tag{6.137}$$

If these inequality conditions are fulfilled, the gradient procedures are reaching the point close to the optimal solution and the calculations on the first level can stop.

The obtained control and state trajectories are sent to a procedure to check if they belong to the area of constraints determined by the inequalities. This is implemented by projection of the obtained values over the area of constraints as follows:

$$x^{\ell}(k) = \begin{bmatrix} x_{min}(k), & \text{if} & x^{\ell,q} \le x_{min}(k) \\ x^{\ell,q}, & \text{if} & x_{min}(k) \le x^{\ell,q} \le x_{max}(k) \\ x_{max}(k), & \text{if} & x^{\ell,q} \ge x_{max}(k) \end{bmatrix} \qquad k = \overline{1,K} \tag{6.138}$$

$$u^{\ell}(k) = \begin{bmatrix} u_{min}(k), \text{if} & u^{\ell,q} \le u_{min}(k) \\ u^{\ell,q}, \text{if} & u_{min}(k) \le u^{\ell,q} \le u_{max}(k) \\ u_{max}(k), \text{if} & u^{\ell,q} \ge u_{max}(k) \end{bmatrix} \qquad k = \overline{1,K-1} \tag{6.139}$$

The obtained trajectories are sent to the coordinating level to be used in solving the coordinating problem.

160

The coordinating problem is also built on the bases of gradient method for calculation of the optimal trajectory of the coordinating variables.

The procedure is based on the solution from the first level sub problems used to calculate the gradient e_λ according to the necessary condition for optimality.

The value of the gradient e_λ is used to check how close to optimal solution the obtained one is. The norm of the gradient is calculated and compared according to:

$$\| e_\lambda{}'(k) \| \leq \varepsilon_0, \varepsilon_0 > 0, \varepsilon_0 = 0.01 \tag{6.140}$$

If equation (6.140) is fulfilled, the obtained solution is optimal and the calculations stop otherwise the new value of the coordinating variable is calculated.

Using a gradient procedure:

$$\lambda^{l+1}(k) = \lambda^l(k) + \alpha_\lambda e_\lambda^l(k), k = \overline{0, K-1}$$ where α_n is the step of the gradient procedure.

The new values are sent to the sub problems of the first level and the calculations are repeated until equation (6.140) is fulfilled. For the next iteration of the coordinating problem the control and state variables obtained from the first iteration are treated as initial conditions. If the equation cannot be reached a maximum number of calculation is determined $\rightarrow M$ and the iterations stop when $l = M$.

6.1.6 Implementation of the above algorithm in MATLAB

All the developed MATLAB programs are presented in the Appendices. A flow chart showing the calculation procedure is presented in Figure 6.2. All the underlined words indicate the names of the MATLAB different functions used for calculations.

161
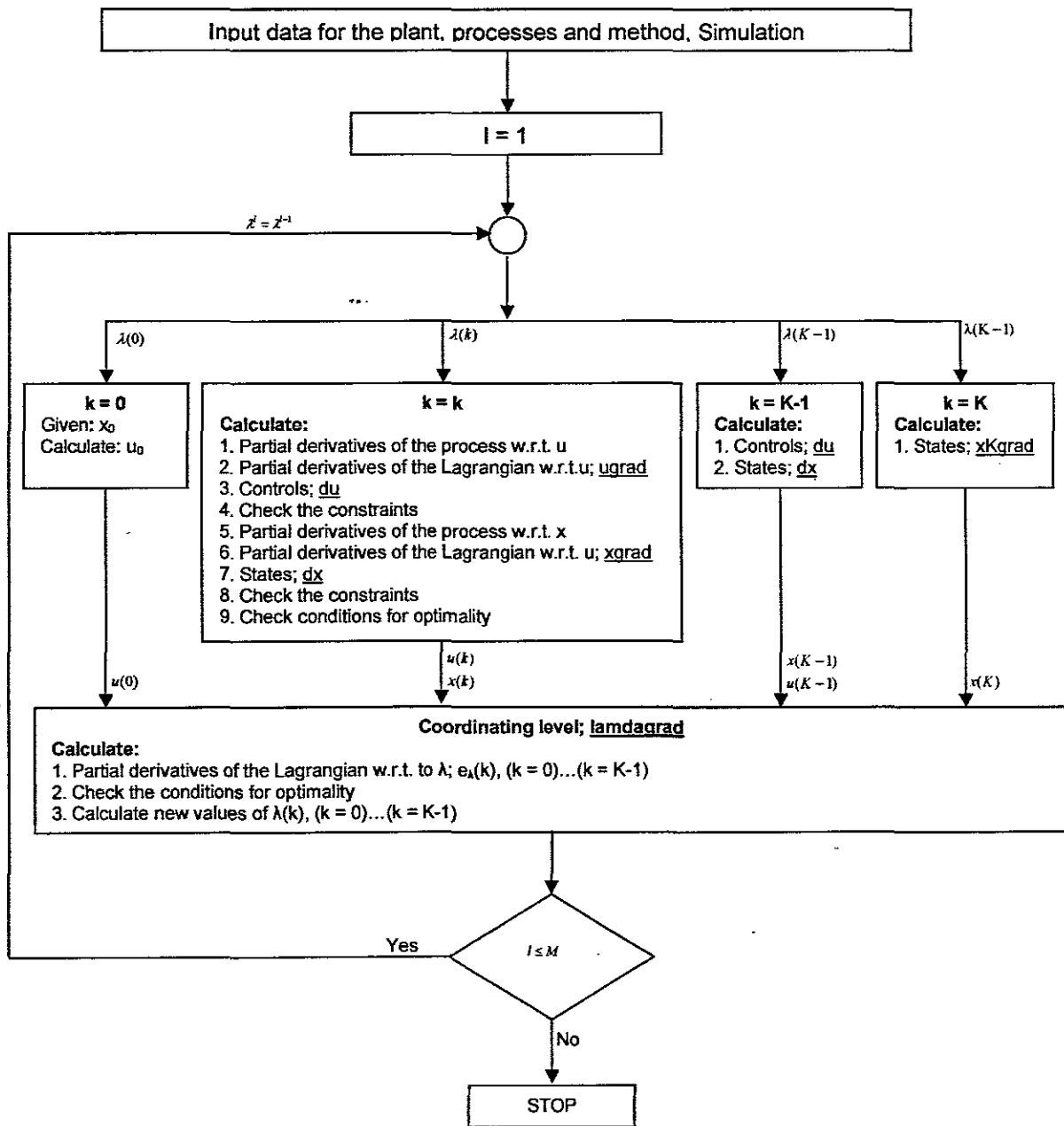
Figure 6.2: Flow chart showing the calculation procedure used in MATLAB

Calculation for the optimal control is done in the following order:

1. At first, a script file with all input data is created:

   - Coefficients of IWA model

   - Initial parameters for the optimal control problem

   - Initial trajectories for the conjugate variables

   - Initial data for the control and state variables

162

2. Script file for calculation of the first level

- Calculate the gradients with respect to the state and the control variables
- Check the conditions for the end of gradient procedure
- Calculate the improved values of state and control variables
- Calculate the state and control gradients using the above improved values
- Continue with the procedure until the conditions are satisfied or till the maximum number of iterations is reached
- When the iteration stops, the obtained trajectories are sent to the second level

3. Script file for calculation on the second level

- Calculate the gradient with respect to the coordinating variables
- Check the conditions, if fulfilled the optimal solution is obtained
- If the condition is not satisfied, calculate the improved values of the coordinate variables and sent to the sub-problem on the first level
- Solve the problems on the first level and send values of state and control solutions to the second level
- Repeat calculation until the conditions are satisfied or till the maximum number of iteration on the second level is reached
- The optimal solution is saved and printed

6.1.7 Solutions to the optimal control problems

Optimal control problems were formulated for Benchmark with reduced ASM1, Benchmark with reduced ASM2, Benchmark with reduced ASM2d, Benchmark with reduced ASM3, Athlone with reduced ASM1, Athlone with reduced ASM2, Athlone with reduced ASM2d and Athlone with reduced ASM3. The MATLAB code for optimal control calculations is generic for all the mass balance models. The difference is in the number of states for every biological model and the number of tanks for each plant. The results for optimal control of the Benchmark plant with ASM1 biological models are shown on the subsequent section.

6.1.7.1 Results for Benchmark problem with reduced ASM1

Reduced ASM1 has 3 state variables; S_S, S_{NO} and S_{NH}. The MATLAB file "*BASM1_Main.m*" given in Appendix A.5 is used for calculation of the optimal control problem for the Benchmark plant with ASM1. The obtained optimal states and control trajectories are presented in Figure 6.3 and Figure 6.4 respectively.
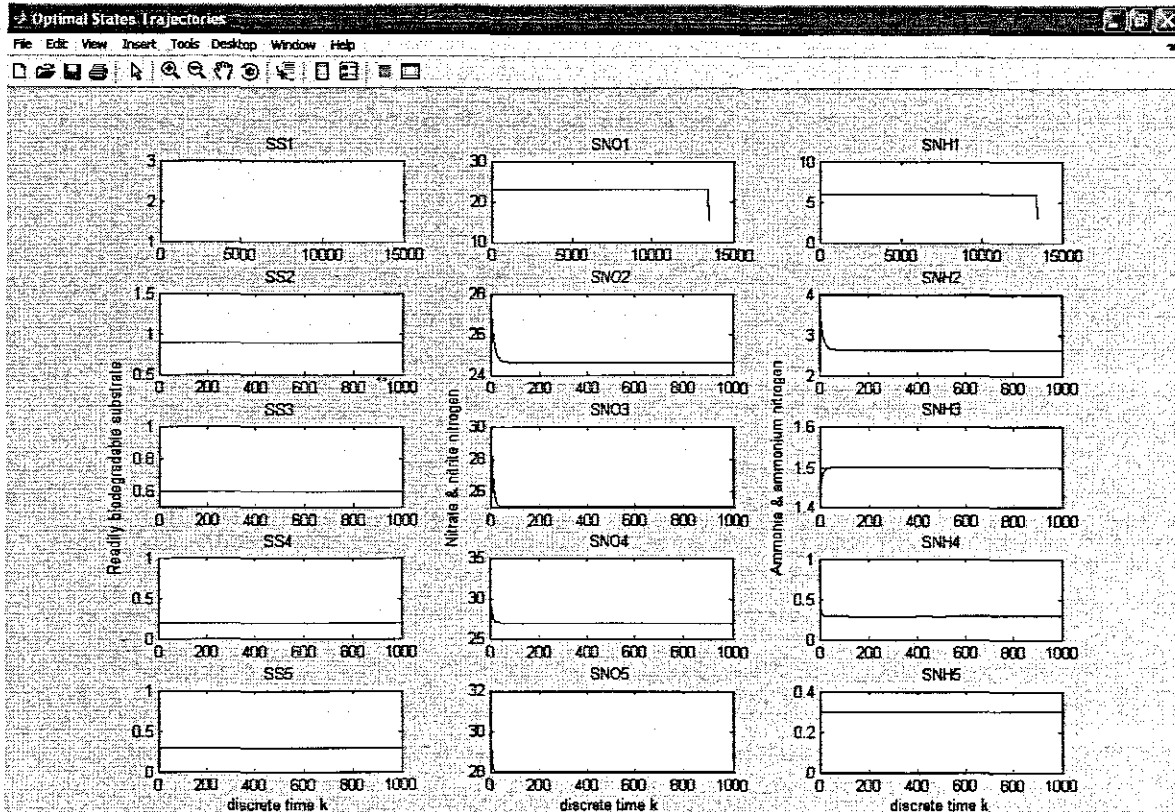
Figure 6.3: Optimal state trajectories for Benchmark plant with reduced ASM1 model
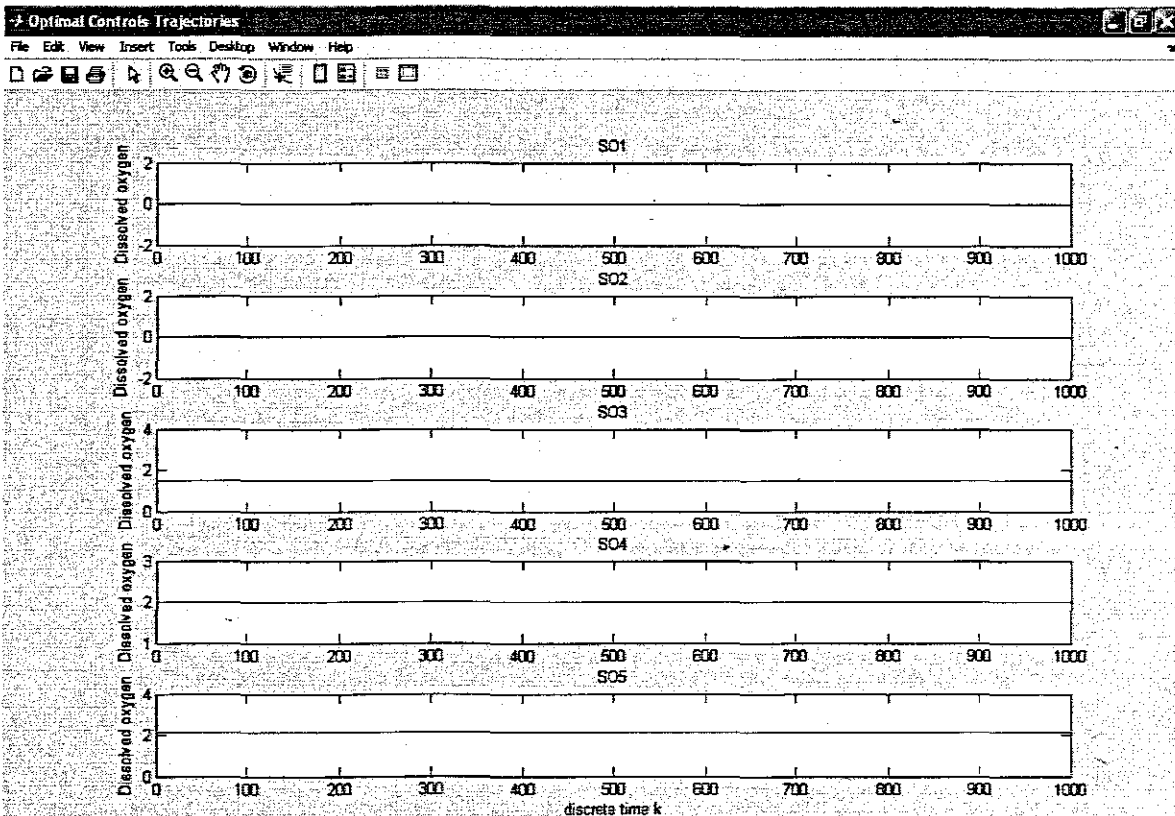
Figure 6.4: Optimal control trajectories for Benchmark plant with reduced ASM1 model

6.1.7.2 Discussion of the results
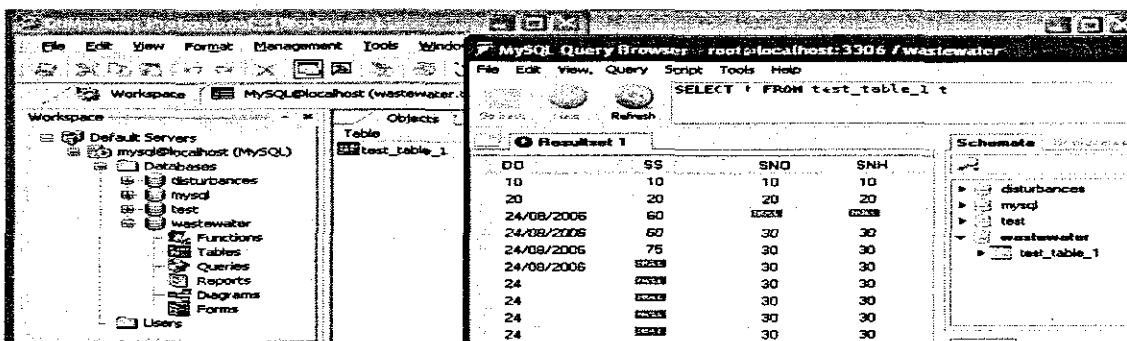
In this section, the results for calculations of the optimal control trajectories for the reduced models are discussed. Calculations are performed for the Benchmark plant with reduced ASM1 biological models.

In this mass balance model, the obtained control trajectories ensure that the system reaches the set points for the state variables. Both the state and the control trajectories are within the constraints specified in the programs.

6.2   Conclusion

Compared to a typical constant dissolved oxygen trajectory of 2mg/l, the obtained optimal DO trajectory ensures that a much less aeration energy will be required. The strategy ensures that the optimal state trajectories be underneath the limitations from the municipality.

The obtained optimal control trajectories are sent to MySQL database where they can be accessed and implemented on a controller. MySQL database is used as a storage facility to create a history of all the data calculated in Matlab and measured from the process. The wastewater database created so far consists of all the reduced mass balance models with the fields for dissolved oxygen values calculated for each aeration tanks. Figure 6.5 shows a database created for storing the sampled optimal DO trajectories for all the reduced mass balance models.

Figure 6.5: MySQL database for historical data storage

CHAPTER SEVEN
CONCLUSION AND FUTURE RECOMMENDATIONS

7.1 Introduction

The purpose of this project is to develop a method for calculation of the optimal dissolved oxygen trajectory as part of the control strategy to be implemented on the activated sludge processes of the wastewater treatment plants.

There is growing interest in research activities relating to finding better ways of controlling the wastewater treatment plants. This is due to an increase in strict requirements on the quality of effluent concentrations imposed on these plants. Dissolved oxygen is chosen as the control trajectory because it has the potential to influence the behaviour of the activated sludge processes as well as ensure that the operational cost can be minimised.

Therefore the demand is to find a way to meet effluent requirements at lower costs. The optimal control strategy makes it possible to incorporate the cost and effluent constraints on a single problem. This project is based on formulation and solutions of such problems.

7.2 Aims and objectives

The aim of the project was to develop methods for calculation of the dissolved oxygen optimal trajectory that may later be used for real time optimal control of activated sludge processes.

The project aim is divided into the following objectives:

1. Literature review and analysis of the optimal control approaches and their application to wastewater treatment processes

2. Development of the full mass balance models for the Benchmark and Athlone plants based on the Activated Sludge model No. 1 (ASM1), Activated Sludge model No. 2 (ASM2), Activated Sludge model No. 2d (ASM2d) and Activated Sludge model No. 3 (ASM3) biological models

3. Reduction of the mass balance models given in objective 2

4. Development of the estimation procedure for calculation of parameters of the reduced mass balance models

5. Development of methods for optimal control calculations for the models considered in objective 2 (plants and reduced models)

166

6. Development of software in MATLAB environment for solution of the problems for parameter estimation, simulation and optimal control calculation

7.3 Thesis deliverables

7.3.1 Literature review on the activated sludge process control

The initial step involved a review of the activated sludge process as an object for control. The different control strategies with different control variables were investigated and compared to each other..

It became apparent that the dissolved oxygen control strategy is one of the most favoured methods due to its ability to save the aeration related energy which makes up more than 40% of the total running costs.

Most of the controllers in the literature implemented with a constant DO set point of 2mg/l. Some researchers investigated the option of using a varying DO set point to further reduce the aeration effort.

Based on this idea, model based controllers which calculate DO set points based on predicted behaviour of the process are implemented. Optimal control theory is one of the model based strategies capable to calculate the optimal DO trajectories which may be used as a varying DO set points for controllers.

The different control strategies available in the literature are compared in Chapter two. These include model predictive controllers, model adaptive controllers and PI controllers.

7.3.2 Development of an approach for calculation of the optimal control strategy

In Chapter three, a procedure for calculation of the optimal control variables is given. The procedure involves formulation of the optimal control problem based on a model of the process and its solution introducing decomposition of the optimal control problem in a set of interconnected sub-problems and coordination of their solutions in order to obtain the initial problem solution.

7.3.3 Development of the full mass balance models for different cases of plant structures and biological models

Based on mass balance principle, the models of the Benchmark and Athlone plants are derived. The plant models incorporates the following biological models; ASM1, ASM2, ASM2d and ASM3. The following mass balance models are developed: Benchmark plant with ASM1, Benchmark plant with ASM2, Benchmark plant with ASM2d, Benchmark plant with ASM3, Benchmark plant with ASM1, Benchmark plant with ASM2, Benchmark plant with ASM2d and Benchmark plant with ASM3.

7.3.4 Development of the reduced mass balance models for different cases of plant structures and biological models

Models for online control are required to be identifiable with state variables that are measurable online. For this reason the biological models are reduced and the reduced mass balance models of the Benchmark and Athlone plants were obtained. The following reduced mass balance models are developed: Benchmark plant with reduced ASM1, Benchmark plant with reduced ASM2, Benchmark plant with reduced ASM2d, Benchmark plant with reduced ASM3, Benchmark plant with reduced ASM1, Benchmark plant with reduced ASM2, Benchmark plant with reduced ASM2d and Benchmark plant with reduced ASM3.

7.3.5 Development of method for estimation of the kinetic parameters of the reduced biological models

A method for parameter estimation of the reduced models is developed based on the Least Square technique. For every of the considered above combinations of plant structures and biological models special vector of parameters to be estimated is selected. Mathematical derivation of the equation for calculation of the selected parameters is provided.

7.3.6 Development of a decomposition-coordinating method for solution of the optimal control problem

The optimal control problems for the reduced mass balance models are formulated based on the optimal control theory covered in Chapter three. The method for solution is obtained based on the Lagrange's method and decomposition coordination approach for time domain decomposition of the problem. Special coordinating vector is introduced in order fully to decompose the Lagrange functional in time domain.

168

7.3.7 MATLAB codes for simulations

The methods developed for simulation of the mass balance models as well as for parameter estimation and the optimal control calculation are implemented in MATLAB simulation environment. The software programs presented in Table 7.1 are developed for the deliverables described above.

Table 7.1: M-files for implementation of the optimal control strategy

Description of the files	m-files for the Benchmark Plant
Main program with calculation of the optimal control	BASM1_Main
Simulation of the full and reduced models	BASM1_ Simulation
Calculation of the process rates for the full model	P_BASM1F
Calculation of the process rates for the reduced model	P_BASM1R
Calculation of the process rates for the reduced model with estimated parameters	P_BASM1RE
Partial derivatives of the Lagrangian w.r.t. the controls	BASM1_ugrad
Partial derivatives of the Lagrangian w.r.t. the states	BASM1_zgrad
Partial derivatives of the Lagrangian w.r.t. the states at the final moment	BASM1_zKgrad
Partial derivatives of the Lagrangian w.r.t. the Lagrange multipliers	BASM1_lamdagrad
Partial derivatives of the process rates w.r.t. the controls	BASM1_du
Partial derivatives of the process rates w.r.t. the states	BASM1_dz
Calculation of the value of the criterion	BASM1_criterion

The m-files shown in Table 7.1 are generic and may be implemented to solve the optimal control problems of all the mass balance models derived on the thesis. The Benchmark plant with the ASM1 biological models is used in this case, as it has lot of data (on the literature) to be used for comparison.

7.3.8 Benefits of the deliverables

The advantage of using a time changing DO set-point is that only the energy needed is used. A constant DO creates loss because the process is supplied with what is not required.

Using the optimal control strategy the appropriate optimal DO trajectory is calculated and as a result the optimal DO set points can be achieved.

The error between the simulation outputs of the full and the reduced mass balance models shows the following:

- The mass balance principle for acquiring the plant models and the Least squares principle for calculating parameters of the reduced models were applied properly
- These principles produces good results even for a nonlinear system like the wastewater treatment plant

The obtained values of the parameters of the reduced models are within close proximities of the original values. These values are not fictitious and out of range.

..

Finally the state trajectories based on the obtained optimal DO trajectory was calculated. These values were within the constraints from the municipality.

The steady state results for Benchmark plant with ASM1 model are very close to the results by the COST Benchmark Group.

7.4 Application of the results

The methods used in this project; mass balance principle, Least Squares principle, optimal control theory and method for decomposition of the optimal control problem are generic and can be applied to any system. These methods are good with nonlinear systems with multiple inputs and outputs. The decomposition method makes it easier to solve the problem on simulation programming environments faster than the sequential global approach of solution.

The obtained DO trajectory may be sampled and tested in a SCADA system for simulation and for a pilot or real plant implementation. The obtained results can be used for educational and experimental purposes, at schools, universities, design institutions and wastewater plants. The also can be used as a basis for future research in the field of wastewater treatment control.

7.5 Future research

The next initiative is to apply the developed methods to a pilot plant. This will involve online estimation of the parameters. It will be interesting to see how quickly the system keeps up with the delays and disturbances on a real process.

7.6   Publications

Kujane, K. & Tzoneva, R. 2008. Investigation and Development of Methods for Optimal Control of the Activated Sludge Process. Paper presented at the Control Systems Day Conference, Wits University, Johannesburg, 25 June 2008.

Kujane, K. & Tzoneva, R. 2009. Method for Simulation of the Benchmark Plant with ASM1 and ASM3 Biological Models. International Journal of Modelling and Simulation (submitted to journal).

REFERENCES

Bastin, G. & Dochain, D. 1990. *On-line Estimation and Adaptive Control of Bioreactor.* New York: Elsevier

Bequette, B.W. 2003. *Process Control: Modeling, Design and Simulation.* Upper Saddle River, NJ: Prentice Hall.

Brdys, M.A. & Zhang, Y. 2001. Robust Hierarchical Optimizing Control of Municipal Wastewater Treatment Plants. 9th *IFAC Symposium on Large Scale Systems: Theory & Applications:* 531-538, 18-20 July.

Bryson, A. E. & Ho, Y. 1975. *Applied Optimal Control: OPTIMIZATION, ESTIMATION AND CONTROL.* California: Hemisphere.

Cadet, C., Béteau J.F. & Hernandez , S.C. 2004. Multicriteria control strategy for cost/quality compromise in wastewater treatment plants. *Control Engineering Practice,* 12:335-347.

Camacho, E. F. & Bordons, C. 2004. *Model Predictive Control.* 2nd ed. London: Springer.

Carlsson, B. & Rehnstrom, A. 2002. Control of an activated sludge process with nitrogen removal—a benchmark study. *Water Science and Technology,* 45(4):135–142.

Carlsson, B. 1997. The use of automatic control in wastewater treatment plants. *Nordisk konferens, kvaverening och biologisk fosfor rening.* Stockholm, 20-30 Jan.

Carlsson, B., Lindberg, C.F., Hasselblad, S. & Xu, S. 1994. On-line estimation of the respiration rate and the oxygen transfer rate at Kungsangen wastewater plant in Uppsala. *Water Science and Technology,* 30(4):255–263.

Chachuat, B., Roche, N. & Lati, M.A. 2005. Optimal aeration control of industrial alternating activated sludge plants. *Biochemical Engineering Journal,* 23(3):277,289.

Copp, J.B. 2002. *The COST simulation benchmark: description and simulator manual (COST Action 624 & 682).* Luxembourg: European Commission.

COST WWTP. n.d. *Benchmark.* http://www.ensic.inpl-nancy.fr/COSTWWTP/ [16 February 2009].

Dochain, D. & Vanrolleghem, P.A. 2001. *Dynamical Modelling and Estimation in Wastewater Treatment Processes.* London: IWA.

Dold, P.L., Ekama, G.A. & Marais, G.v.R. 1980. A general model for the activated sludge process. Progress Water Technology, 12:47-77.

Du Plessis, C.S. 2009. Investigation of process parameters and development of a mathematical model for the purposes of control design and implementation for a wastewater treatment process. Unpublished PhD thesis, Cape Peninsula University of Technology, Cape Town.

Gujer, W., Henze, M., Mino, T. & van Loosdrecht, M.C.M. 1999. Activated Sludge Model No. 3. *Water Science and Technology,* 39(1):183,193.

Henze, M., Grady Jr., C.P.L., Gujer, W., Marais, G.v.R. & Matsuo, T. 1987. *Activated sludge model No. 1: IAWPRC Scientific and Technical Report No. 1.* London: IAWPRC.

Henze, M., Gujer, W., Mino, T. & Loodrecht, M. 2000. *Activated Sludge Models: ASM1, ASM2, ASM2D and ASM3: Report No. 9.* London: IWA.

Henze, M., Gujer, W., Mino, T., Matsuo, T., Wentzel, M.C. & Marais, G.v.R. 1995. *Activated sludge model No. 2: IAWQ Scientific and Technical Report No. 3.* London: IAWQ.

Henze, M., Gujer, W., Mino, T., Matsuo, T., Wentzel, M.C., Marais, G.v.R. & Loosdrecht, M.C.M. 1998. *Outline Activated Sludge Model No. 2d.* London: IAWQ.

Holenda, B. 2007. Development of modelling, control and optimization tools for the activated sludge process. PhD thesis. Pannon University, Pannon.

Holenda, B., Domokos, E., Reedey, A. & Fazakas, J. 2005a. Application of Model Predictive Control for the dissolved oxygen control of the COST simulation benchmark. Presented at Chemeca 2005, Brisbane, 10-12 Nov 2005.

Holenda, B., Domokos, E., Redey, A. & Fazakas, J. 2005b. Dissolved oxygen control of small-size wastewater treatment plants using model predictive control. *Presented at 10th EuCheMS-DCE International Conference*, Rimini, 4-7 Sept 2005.

Holenda, B., Domokos, E., Redey, A. & Fazakas, J. 2005c. Effluent water quality optimization of the alternating activated sludge process using genetic algorithm. *Presented at 10th EuCheMS-DCE International Conference*, Rimini, 4-7 Sept 2005.

Holenda, B., Domokos, E., Redey, A. & Fazakas, J. 2007a. Aeration optimization of a wastewater treatment plant using genetic algorithm. *Optimal Control Applications and Methods*, 8(3):191,208.

Holenda, B., Domokos, E., Redey, A. & Fazakas, J. 2007b. Dissolved oxygen control of the activated sludge wastewater treatment process using model predictive control. *Computers and chemical Engineering*, 32(2008):1270-1278. June 4.

Ikonen, E. & Najim, K. 2002. *Advanced Process Identification and Control.* New York, Marcel Dekker.

Ingilsen, P. 2002. Realising Full-Scale Control in Wastewater Treatment Systems Using In situ Nutrient Sensors. PhD. thesis, Lund University, Lund.

Ingildsen, P. & G. Olsson. 2001. *Get more out of your wastewater treatment plant complexity made simple.* Denmark: Danfoss.

Ingildsen, P., Jeppson, U. & Olsson, G. 2002. Dissolved oxygen controller based on on-line measurements of ammonia combining feedforward and feedback. *Water Science Technology*, 45(4-5):453-460.

IWA. n.d. *IWA Task Group on Benchmarking of Control Strategies for WWTPs.* http://www.benchmarkwwtp.org/ [16 February 2009].

Jeppsson, U. & Olsson, G. 1993. Reduced order models for on-line parameter identifications of the activated sludge process. *Water Science Technology*, 28(11):173-183.

Jeppsson, U. 1996. Modelling aspects of wastewater treatment processes. PhD Thesis, Lund Institute of Technology, Lund.

Jeppsson, U. & Pons, M.N. 2004. The COST benchmark simulation model–current state and future perspective. *Control Eng. Pract*, 12 (3), 299–304.

Keesman, K.J. 2002. State and parameter estimation in biotechnical batch reactors. *Control Engineering Practice*, 10:219-225.

Kozak, S. & Huba, M. 2004. *Control Systems Design 2003*. Oxford: IFAC.

Levine, W.S. 2000. *Control System Fundamentals*. Boca Raton, FL: CRC.

Lindberg, C. & Carlsson, B. 1993. Evaluation of some methods for identifying the oxygen transfer rate and the respiration rate in an activated sludge process. *Technical Report UPTEC 93032R*, 41.

Lindberg, C. 1995. Control of wastewater treatment plants. Licentiate thesis. Uppsala University, Uppsala.

Lindberg, C.F. & Carlsson, B. 1996a. Adaptive control of external carbon flow rate in an activated sludge process. *Wat. Sci. Tech*, 34(3):173,180.

Lindberg, C.F. & Carlsson, B. 1996b. Efficient control of the DO in an activated sludge process. Vatten, 52(3):209-212.

Lindberg, C. & Carlson, B. 1996c. Estimation of the respiration rate and oxygen transfer function utilizing a slow do sensor. *Water Science Technology*, 33(1):325-333.

Lindberg, C. & Carlsson, B. 1996d. Nonlinear and set-point control of the dissolved oxygen dynamic in an activated sludge process. *Wat. Sci. Tech*. 34(3):135.142.

Lindberg, C. 1997. Control and estimation strategies applied to the Activated Sludge Process. PhD. Thesis, Uppsala University, Uppsala.

Liptak, B. G. 2006. *Process control and optimization*. 4th ed. Boca Raton, FL: CRC.

Lukasse, L.J.S., Keesman, K.J. & van Straten, G. 1996. Grey-box identification of dissolved oxygen dynamics in activated sludge process, *Proceedings of the 13th World Congress of IFAC*, San Francisco, 485-490.

Lukasse, L.J.S., Keesman, K.J. & van Straten, G. 1997. Estimation of BOD respiration rate and kinetics of activated sludge. *Water Research*, 31(9):2278-2286.

Lukasse, L.J.S., Keesman, K.J. & van Straten, G. 1999. A recursively identified model for short-term predictions of Ammonium concentrations in alternating activated sludge processes. Journal *of Process Control*, 9:87-100.

Maciejowski, J.M. 2002. *Predictive Control with constraints*. Harlow: Prentice-Hall.

Motor Management Success. 2001. **http://www.cee1.org/ind/mot-sys/ww/Kennewick_case.pdf** [11 May 2008].

Naidu, D. *Optimal Control Systems.* IDAHO: CRC.

Nejjari, F., Benhammou, A., Dahhou, B. & Roux, G. 1997. Nonlinear multivariable control of a biological wastewater treatment process. Proceedings of EEC 97, Brussels, n.d.

Nejjari, F., Benhammou, A., Dahhou, B. & Roux, G. 1999. Nonlinear multivariable adaptive control of an activated sludge wastewater treatment process. *International Journal of Adaptive Control and Signal Processing,* 13(5):347-365.

Nejjari, F., BenYoussef, C., Benhammou, A. & Dahhou, B. 1996. Procedures for state and parameter estimation of a biological wastewater treatment. *CESA'96 IMACS,* Lille, 9-12 July. 1: 238-243.

Nejjari, F., Roux, G., Dahhou, B. & Benhammou, A. 1999. Estimation and optimal control design of biological wastewater treatment process. *Mathematics Control Simulation,* 48:269-280.

Nketoane, P. 2009. Design and implementation of a nonlinear controller in PLC as a part of an Adroit SCADA system for optimal adaptive control of the activated sludge process. Unpublished MTech thesis, Cape Peninsula University of Technology. Cape Town.

Olsson, G. & Newell, B. 1999. *Wastewater Treatment Systems: Modelling, Diagnosis and Control.* London: IWA.

Olsson, G. 1992. Control of wastewater treatment systems. ISA Transactions, 31(1):87-96.

Petersen, B., Gernaey, K. & Vanrolleghem, P.A. 2000. Improved theoretical identifiability of model parameters by combined respirometric-titrimetric measurements: A generalisation of results. In:*Proceedings IMACS 3rd Symposium on Mathematical Modelling.* Vienna. 2-4 February 2000, Vienna University of Technology, 2:639–642.

Petersen. B., Gernaey, K., Henze, M. and Vanrolleghem, P.A. 2002. Evaluation of an ASM1 model calibration procedure on a municipal-industrial wastewater treatment plant. *Hydroinformatics.* 4:15-38.

Rauch, W. & Harremoes, P. 1999. Genetic algorithms in real time control applied to minimize transient pollution from urban wastewater systems. *Water Res,* 33(5):1265,1277.

Samuelsson, P. & Carlsson. B. 2001. Feedforward control of the external carbon flow rate in an activated sludge process. *Water Science and Technology,* 43(1):115-122.

Samuelsson, P., Halvarsson, B., & Carlsson, B. 2005a. Cost-efficient operation of a Denitrifying activated sludge Process—an initial study. *Technical Report 2005-010.* Uppsala, Uppsala University Press.

Samuelsson, P., Halvarsson, B. & Carlsson, B. 2007. Cost efficient operation of a denitrifying activated sludge process. *Water Research,* 41(11):2325-2332.

Schütze, M., Campisano, A., Colas H., Schilling, W. & Vanrolleghem, P.A. 2004. Real time control of urban wastewater systems – where do we stand today? *Journal of Hydrology.* 299(34): 335-348.

Schütze, M.R., Butler, D. & Beck, M.B. 2002. *Modelling, Simulation and Control of Urban Wastewater Systems*. London: Springer.

SIMBA 4, Userguide 2001. Simulation of biological wastewater treatment. *IFAK system GmbH*, Magdeburg, n.d.

Singh, M.G. & Titli, A. 1978. *Systems: Decomposition. Optimization and Control*. Oxford: Pergamon.

Smets, I.Y., Haegebaert, V.J., Carrette, R. & van Impe, J.F. 2003. Linearization of the activated sludge model ASM1for fast and reliable prediction. *Water Research*, 37(8): 1831-1851.

Smits, I.Y.M., Versyck, K.J.E. & van Impe, J.F.M. 2002. Optimal control theory: a generic tool for identification and control of (bio-)chemical reactors. *Annual Reviews in Control*. 26: 57-73.

South Africa. National Government. 1996. http://www.info.gov.za/documents/constitution/1996/96cons7.htm#155 [15 May 2008].

Stare, A., Vrecko. D., Hvala, N. & Strmcnik, S. 2007. Comparison of control strategies for nitrogen removal in an activated sludge process in terms of operating costs: A simulation study. *Water Research*, 41(9):2004-2014.

Steffens. M. A.,& Lant, P. A. 1999. Multivariable control of nutrient-removing activated sludge systems. *Water Research*, 33(12): 2864–2878.

Steffens, M.A., Lant, P.A. & Newell, R.B. 1997. A systematic approach for reducing complex biological wastewater treatment models. *Water Research*, 31(3): 590-606.

Vanrolleghem, P. & Gillot, S. 2002. Robustness and economic measures as control benchmark perform criteria. *Water Science and Technology*, 45(4–5):117–126.

Vanrolleghem, P. A. 1994. *On-line modelling of activated sludge processes: Development of an adaptive sensor*. PhD thesis. University of Gent, Gent.

Walter, E. & Pronzato, L. 1995. On the identifiability and distinguishability of nonlinear parametric models. In: Proc. Symp. *Applications of modelling and control in agriculture and bioindustries*, IMACS, Brussels, n.d.

WRC. 2002. http://www.wrc.org.za/downloads/knowledgereview/2002/Municipal.pdf [15 May 2008].

Youssef, C., Dahhou, B., Zeng, F. & Rols, T. 1996. Estimation and filtering of nonlinear systems. Application to a wastewater treatment process. *In International Journal System Science*, 27: 497-505.

Yuan, Z. & Keller, J. 2003. Integrated control of nitrate recirculation and external carbon addition in a predenitrification system. *Water Sci. Technol.*. 48 (11): 345–354.

Yuan, Z., Bogaert, H., Vanrolleghem, P., Thoeye. C., Vansteenkiste, G. & Verstraete, W., 1997. Control of external carbon addition to predenitrifying systems. *J. Environ. Eng.* 123 (11):1080–1086.

APPENDICES

APPENDIX A: DEVELOPED MATLAB CODE FOR OPTIMAL CONTROL OF THE BENCHMARK PLANT WITH THE ACTIVATED SLUDGE MODEL NO.1 BIO-MODEL

A.1: MATLAB script file – BASM1_Simulation.m

```
% M-File: BASM1_Simulation.m
% ========================= M-FILE DESCRIPTION =====================
% This m-file is used for:
% Simulation of the Benchmark plant using full ASM1 model
% Calculation of the parameters for the reduced ASM1 model
% Simulation of the Benchmark plant using the reduced ASM1 model

clear all          % Clear the workspace
clc                % Clear the command window


% ======================= PARAMETER DEFINITION =====================
% Declaration of the global variables
global muH bH muA bA etag ka kh etah Ks KOH KNO KNH KOA KX

% Declaration of the global variables
K1    = 95969;      % Prediction horizon or interval (100 days sampled every 0.001042)
DT    = 0.001042;   % Sampling time

% Initialization of the model parameters, typical values at 15 0C
% Stochiometric parameters
YH    = 0.67;       % Heterotrophic yield
YA    = 0.24;       % Autotrophic yield
fP    = 0.08;       % Fraction of biomass yielding particulate products
iXB   = 0.08;       % COD mass in biomass
iXP   = 0.06;       % COD mass in products from biomass
% Kinetic parameters
muH   = 4;          % Heterotrophic maximum specific growth rate
bH    = 0.3;        % Heterotrophic decay rate
muA   = 0.5;        % Autotrophic maximum specific growth rate
bA    = 0.05;       % Correction factor for anoxic growth of heterotrophs
etag  = 0.8;        % Correction factor for anoxic growth of heterotrophs
ka    = 0.05;       % Ammonification rate
kh    = 3;          % Maximum specific hydrolysis rate
etah  = 0.8;        % Correction factor for anoxic hydrolysis
% Monod switching functions
Ks    = 10.0;       % Half saturation coefficient (hsc) for heterotrophs
KOH   = 0.2;        % Oxygen hsc for heterotrophs
KNO   = 0.5;        % Nitrate hsc for denitrifying heterotrophs
KNH   = 1.0;        % Ammonia hsc for autotrophs
KOA   = 0.4;        % Oxygen hsc for autotrophs
KX    = 0.1;        % Hsc for hydrolysis of slowly biodegradable substrate

% Initialization of the Benchmark process data
V1    = 1000;       % Tank 1 volume
V2    = 1000;       % Tank 2 volume
V3    = 1333;       % Tank 3 volume
V4    = 1333;       % Tank 4 volume
V5    = 1333;       % Tank 5 volume

Q0    = 18446;      % Influent flow rate
Qa    = 55338;      % Internal recirculation rate
Qr    = 18446;      % Sludge recycled rate
Qw    = 385;        % Wasted sludge rate
Q     = Q0 + Qa + Qr;% Flow through the tanks
% Coeficient for fraction of flow rates for the particulate materials
LAMBDA = (Q0+Qr)/(Qr+Qw);

% Oxygen parameters
```

```
SOsat = 8;              % Oxygen saturation point
KLa1  = 155;            % Oxygen transfer function in Tank 1
KLa2  = 155;            % Oxygen transfer function in Tank 2
KLa3  = 240;            % Oxygen transfer function in Tank 3
KLa4  = 240;            % Oxygen transfer function in Tank 4
KLa5  = 84;             % Oxygen transfer function in Tank 5


KL = [KLa1 KLa2 KLa3 KLa4 KLa5]; KLa = diag(KL);


% ============================ PROCESS MODEL ========================
% Calculation of the MATRIX A
% Matrix A represent weighting factors of the accumulated state variables
% from different flows towards state variables in each tank as derived
% through mass balance principles, see chapter 3 of the Thesis
IV1 = DT/ V1; IV11 = IV1 * Q; A11 = 1 - IV11;
IV2 = DT/ V2; IV22 = IV2 * Q; A22 = 1 - IV22;
IV3 = DT/ V3; IV33 = IV3 * Q; A33 = 1 - IV33;
IV4 = DT/ V4; IV44 = IV4 * Q; A44 = 1 - IV44;
IV5 = DT/ V5; IV55 = IV5 * Q; A55 = 1 - IV55;


A15a = IV1 * (Qa + Qr);         % Soluble materials recycled from Tank 5
A15b = IV1 * (Qa + LAMBDA*Qr);% Particulate materials recycled from Tank 5
A21 = IV22; A32 = IV33; A43 = IV44; A54 = IV55;


A11 = A11 * ones(1,13); A11 = diag(A11);
A22 = A22 * ones(1,13); A22 = diag(A22);
A33 = A33 * ones(1,13); A33 = diag(A33);
A44 = A44 * ones(1,13); A44 = diag(A44);
A55 = A55 * ones(1,13); A55 = diag(A55);


A15 = [A15a,A15a,A15b,A15b,A15b,A15b,A15b,A15a,A15a,A15a,A15a,A15b,A15a];
A15 = diag(A15);
A21 = A21 * ones(1,13); A21 = diag(A21);
A32 = A32 * ones(1,13); A32 = diag(A32);
A43 = A43 * ones(1,13); A43 = diag(A43);
A54 = A54 * ones(1,13); A54 = diag(A54);


A10 = zeros(13,13); A20 = zeros(13,26); A30 = zeros(13,39);


A = [A11 A30 A15;A21 A22 A30; A10 A32 A33 A20;A20 A43 A44 A10;A30 A54 A55];


% Calculation of the MATRIX B
% Matrix B represent weighting factors of the state variables towards the
% influent wastewater composition as derived through mass balance
% principles, see chapter 3 of the Thesis
IV0 = DT / V1*Q0;
B11 = IV0 * ones(1,13); B11 = diag(B11);


B = [B11; zeros(52,13)];


% Formulation of the MATRIX C
% Matrix C represent weighting factors of the state variables towards the
% processes on the ASM1 matrix table
c1 = -(1/YH);
c2 = (-(1-YH))/(YH);
c3 = -(iXB/14);
c4 = (-(1-YH))/(2.86*YH);
c5 = ((1-YH)/(14*2.86*YH))-(iXB/14);
c6 = (-4.57/YA)+1;
c7 = (1/YA);
c8 = -iXB-(1/YA);
c9 = -(iXB/14)-(1/(7*YA));
c10 = 1 - fP;
c11 = iXB-(fP*iXP);


CC = [[0 c1 0 0   1  0   0  c2 0  -iXB 0   0   c3  ];
```

179

```
            [0  c1  0  0    1   0   0   0   c4 -iXB  0   0   c5  ];
            [0  0   0  0    0   1   0   c6  c7  c8   0   0   c9  ];
            [0  0   0  c10 -1   0   fP  0   0   0    0  c11  0   ];
            [0  0   0  c10  0  -1   fP  0   0   0    0  c11  0   ];
            [0  0   0  0    0   0   0   0   0   1   -1   0   1/14];
            [0  1   0 -1    0   0   0   0   0   0    0   0   0   ];
            [0  0   0  0    0   0   0   0   0   0    1  -1   0   ]];

   C = DT*[CC zeros(8,52); zeros(8,13) CC zeros(8,39);
          zeros(8,26) CC zeros(8,26); zeros(8,39) CC zeros(8,13);
          zeros(8,52) CC];


   % ======================== SIMULATION DATA ========================
   % Average values of the inflow concentrations based on the dry weather file
   SIi = 30.0; SSi = 69.5; XIi = 51.2; XSi = 202.32; XBHi = 28.17; XBAi = 0.0;
   XPi = 0.0; SOi = 0; SNOi = 0.0; SNHi = 31.56; SNDi = 6.95; XNDi = 10.59;
   SALKi = 7.0;
   % SI SS XI XS XBH XBA XP SO SNO SNH SND XND SALK;
   xi = [SIi*ones(1,K1);  SSi*ones(1,K1);   XIi*ones(1,K1);   XSi*ones(1,K1);
        XBHi*ones(1,K1); XBAi*ones(1,K1);  XPi*ones(1,K1);   SOi*ones(1,K1);
        SNOi*ones(1,K1); SNHi*ones(1,K1);  SNDi*ones(1,K1);  XNDi*ones(1,K1);
        SALKi*ones(1,K1)];


   % Values of the initial conditions
   % For tank 1
   SI_1   = 30; SS_1  = 2.8082; XI_1   = 1149.1252;
   XS_1   = 82.1349; XBH_1 = 2551.7658; XBA_1 = 148.3894;
   XP_1   = 448.8519; SO_1  = 0.0042984; SNO_1 = 5.3699;
   SNH_1  = 7.9179; SND_1 = 1.2166; XND_1 = 5.2849;
   SALK_1 = 4.9277;
   x10 = [SI_1; SS_1; XI_1; XS_1; XBH_1; XBA_1; XP_1; SO_1; SNO_1; SNH_1;
         SND_1; XND_1; SALK_1 ]*0.5;


   % For tank 2
   SI_2   = 30; SS_2  = 1.4588; XI_2   = 1149.1252;
   XS_2   = 76.3862; XBH_2 = 2553.3851; XBA_2 = 148.3091;
   XP_2   = 449.5227; SO_2  = 6.3132e-05; SNO_2 = 3.662;
   SNH_2  = 8.3444; SND_2 = 0.88206; XND_2 = 5.0291;
   SALK_2 = 5.0802;
   x20 = [SI_2; SS_2; XI_2; XS_2; XBH_2; XBA_2; XP_2; SO_2; SNO_2; SNH_2;
         SND_2; XND_2; SALK_2 ]*0.5;


   % For tank 3
   SI_3   = 30; SS_3  = 1.1495; XI_3   = 1149.1252;
   XS_3   = 64.8549; XBH_3 = 2557.1314; XBA_3 = 148.9413;
   XP_3   = 450.4183; SO_3  = 1.7184; SNO_3 = 6.5409;
   SNH_3  = 5.5479; SND_3 = 0.82889; XND_3 = 4.3924;
   SALK_3 = 4.6748;
   x30 = [SI_3; SS_3; XI_3; XS_3; XBH_3; XBA_3; XP_3; SO_3; SNO_3; SNH_3;
         SND_3; XND_3; SALK_3 ]*0.5;


   % For tank 4
   SI_4   = 30; SS_4  = 0.99532; XI_4   = 1149.1252;
   XS_4   = 55.694; XBH_4 = 2559.1826; XBA_4 = 149.5271;
   XP_4   = 451.3147; SO_4   = 2.4289; SNO_4 = 9.299;
   SNH_4  = 2.9674; SND_4 = 0.76679; XND_4 = 3.879;
   SALK_4 = 4.2935;
   x40 = [SI_4; SS_4; XI_4; XS_4; XBH_4; XBA_4; XP_4; SO_4; SNO_4; SNH_4;
         SND_4; XND_4; SALK_4 ]*0.5;


   % For tank 5
   SI_5   = 30; SS_5  = 0.88949; XI_5   = 1149.1252;
   XS_5   = 49.3056; XBH_5 = 2559.3436; XBA_5 = 149.7971;
   XP_5   = 452.2111; SO_5  = 0.49094; SNO_5 = 10.4152;
   SNH_5  = 1.7333; SND_5 = 0.68828; XND_5 = 3.5272;
   SALK_5 = 4.1256;
```

```
x50 = [SI_5; SS_5; XI_5; XS_5; XBH_5; XBA_5; XP_5; SO_5; SNO_5; SNH_5;
       SND_5; XND_5; SALK_5 ]*0.5;


x0 = [x10;x20;x30;x40;x50];
x  = [x0 zeros(65,K1)];

% Steady state simulation of the model
for k = 1:K1
    xk  = x(:,k);
    xik = xi(:,k);
    Pk  = P_BASM1F(xk);     % Calculation of the process rates

    SOk = [xk(8,1);xk(21,1);xk(34,1);xk(47,1);xk(60,1)];
    xxk = DT*KLa*(SOsat-SOk);
    xx1k =
[zeros(7,1);xxk(1);zeros(12,1);xxk(2);zeros(12,1);xxk(3);zeros(12,1);xxk(4);zeros(12,1)
;xxk(5);zeros(5,1)];
    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xx1k;
end

% Full model response to the steady state influent concentrations
% Tank 1
figure(1)
subplot(5,3,1);plot(x(1,:));title('SI1')
subplot(5,3,2);plot(x(2,:));title('SS1')
subplot(5,3,3);plot(x(3,:));title('XI1')
subplot(5,3,4);plot(x(4,:));title('XS1')
subplot(5,3,5);plot(x(5,:));title('XBH1')
subplot(5,3,6);plot(x(6,:));title('XBA1')
subplot(5,3,7);plot(x(7,:));title('XP1')
subplot(5,3,8);plot(x(8,:));title('SO1')
subplot(5,3,9);plot(x(9,:));title('SNO1')
subplot(5,3,10);plot(x(10,:));title('SNH1')
subplot(5,3,11);plot(x(11,:));title('SND1')
xlabel('discrete time k')
subplot(5,3,12);plot(x(12,:));title('XND1')
xlabel('discrete time k')
subplot(5,3,13);plot(x(13,:));title('SALK1')
xlabel('discrete time k')
set(figure(1),'name','Tank 1 steady state results','numbertitle','off')

% Tank 2
figure(2)
subplot(5,3,1);plot(x(14,:));title('SI2')
subplot(5,3,2);plot(x(15,:));title('SS2')
subplot(5,3,3);plot(x(16,:));title('XI2')
subplot(5,3,4);plot(x(17,:));title('XS2')
subplot(5,3,5);plot(x(18,:));title('XBH2')
subplot(5,3,6);plot(x(19,:));title('XBA2')
subplot(5,3,7);plot(x(20,:));title('XP2')
subplot(5,3,8);plot(x(21,:));title('SO2')
subplot(5,3,9);plot(x(22,:));title('SNO2')
subplot(5,3,10);plot(x(23,:));title('SNH2')
subplot(5,3,11);plot(x(24,:));title('SND2')
xlabel('discrete time k')
subplot(5,3,12);plot(x(25,:));title('XND2')
xlabel('discrete time k')
subplot(5,3,13);plot(x(26,:));title('SALK2')
xlabel('discrete time k')
set(figure(2),'name','Tank 2 steady state results','numbertitle','off')

% Tank 3
figure(3)
subplot(5,3,1);plot(x(27,:));title('SI3')
subplot(5,3,2);plot(x(28,:));title('SS3')
subplot(5,3,3);plot(x(29,:));title('XI3')
```

181

```
subplot(5,3,4);plot(x(30,:));title('XS3')
subplot(5,3,5);plot(x(31,:));title('XBH3')
subplot(5,3,6);plot(x(32,:));title('XBA3')
subplot(5,3,7);plot(x(33,:));title('XP3')
subplot(5,3,8);plot(x(34,:));title('SO3')
subplot(5,3,9);plot(x(35,:));title('SNO3')
subplot(5,3,10);plot(x(36,:));title('SNH3')
subplot(5,3,11);plot(x(37,:));title('SND3')
xlabel('discrete time k')
subplot(5,3,12);plot(x(38,:));title('XND3')
xlabel('discrete time k')
subplot(5,3,13);plot(x(39,:));title('SALK3')
xlabel('discrete time k')
set(figure(3),'name','Tank 3 steady state results','numbertitle','off')


% Tank 4
figure(4)
subplot(5,3,1);plot(x(40,:));title('SI4')
subplot(5,3,2);plot(x(41,:));title('SS4')
subplot(5,3,3);plot(x(42,:));title('XI4')
subplot(5,3,4);plot(x(43,:));title('XS4')
subplot(5,3,5);plot(x(44,:));title('XBH4')
subplot(5,3,6);plot(x(45,:));title('XBA4')
subplot(5,3,7);plot(x(46,:));title('XP4')
subplot(5,3,8);plot(x(47,:));title('SO4')
subplot(5,3,9);plot(x(48,:));title('SNO4')
subplot(5,3,10);plot(x(49,:));title('SNH4')
subplot(5,3,11);plot(x(50,:));title('SND4')
xlabel('discrete time k')
subplot(5,3,12);plot(x(51,:));title('XND4')
xlabel('discrete time k')
subplot(5,3,13);plot(x(52,:));title('SALK4');
xlabel('discrete time k')
set(figure(4),'name','Tank 4 steady state results','numbertitle','off')


% Tank 5
figure(5)
subplot(5,3,1);plot(x(53,:));title('SI5')
subplot(5,3,2);plot(x(54,:));title('SS5')
subplot(5,3,3);plot(x(55,:));title('XI5')
subplot(5,3,4);plot(x(56,:));title('XS5')
subplot(5,3,5);plot(x(57,:));title('XBH5')
subplot(5,3,6);plot(x(58,:));title('XBA5')
subplot(5,3,7);plot(x(59,:));title('XP5')
subplot(5,3,8);plot(x(60,:));title('SO5')
subplot(5,3,9);plot(x(61,:));title('SNO5')
subplot(5,3,10);plot(x(62,:));title('SNH5')
subplot(5,3,11);plot(x(63,:));title('SND5')
xlabel('discrete time k')
subplot(5,3,12);plot(x(64,:));title('XND5')
xlabel('discrete time k')
subplot(5,3,13);plot(x(65,:));title('SALK5');
xlabel('discrete time k')
set(figure(5),'name','Tank 5 steady state results','numbertitle','off')


% Data for dynamic simulations
K = 13436;   % Prediction horizon for dynamic simulations (14 days sampled 0.001042)
DT  = 0.001042;

% Dynamics of the inflow concentrations for dry weather file
load dry.txt

t    = 1344;
SIi = dry(1:t,7)'; SSi = dry(1:t,2)'; XIi = dry(1:t,5)'; XSi = dry(1:t,4)';
XBHi = dry(1:t,3)'; XBAi = XBAi*zeros(1,t); XPi = XPi*zeros(1,t);
SOi = SOi*zeros(1,t); SNOi = SNOi*zeros(1,t); SNHi = dry(1:t,6)';
```
182

```
SNDi = dry(1:t,8)'; XNDi = dry(1:t,9)'; SALKi =  SALKi* ones(1,t);

t1 = [dry(:,1)']; t2 = [0:0.001042:14];

SIi = spline(t1,SIi,t2);   SSi = spline(t1,SSi,t2);   XIi = spline(t1,XIi,t2);
XSi = spline(t1,XSi,t2);   XBHi = spline(t1,XBHi,t2); XBAi = spline(t1,XBAi,t2);
XPi = spline(t1,XPi,t2);   SOi = spline(t1,SOi,t2);   SNOi = spline(t1,SNOi,t2);
SNHi = spline(t1,SNHi,t2); SNDi = spline(t1,SNDi,t2); XNDi = spline(t1,XNDi,t2);
SALKi = spline(t1,SALKi,t2);

xi = [SIi;SSi;XIi;XSi;XBHi;XBAi;XPi;SOi;SNOi;SNHi;SNDi;XNDi;SALKi];
x = [x(:,(K1+1)) zeros(65,K)];

% Dynamic simulation of the full model
for k = 1:K
    xk  = x(:,k);
    xik = xi(:,k);
    Pk  = P_BASM1F(xk);

    SOk  = [xk(8,1);xk(21,1);xk(34,1);xk(47,1);xk(60,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xx1k =
[zeros(7,1);xxk(1);zeros(12,1);xxk(2);zeros(12,1);xxk(3);zeros(12,1);xxk(4);zeros(12,1)
;xxk(5);zeros(5,1)];
    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xx1k;
end

% Full model response to the dynamic influent concentrations
% Tank 1
figure(6)
subplot(5,3,1);plot(x(1,:));title('SI1')
subplot(5,3,2);plot(x(2,:));title('SS1')
subplot(5,3,3);plot(x(3,:));title('XI1')
subplot(5,3,4);plot(x(4,:));title('XS1')
subplot(5,3,5);plot(x(5,:));title('XBH1')
subplot(5,3,6);plot(x(6,:));title('XBA1')
subplot(5,3,7);plot(x(7,:));title('XP1')
subplot(5,3,8);plot(x(8,:));title('SO1')
subplot(5,3,9);plot(x(9,:));title('SNO1')
subplot(5,3,10);plot(x(10,:));title('SNH1')
subplot(5,3,11);plot(x(11,:));title('SND1')
xlabel('discrete time k')
subplot(5,3,12);plot(x(12,:));title('XND1')
xlabel('discrete time k')
subplot(5,3,13);plot(x(13,:));title('SALK1')
xlabel('discrete time k')
set(figure(6),'name','Tank 1 dynamic results','numbertitle','off')

% Tank 2
figure(7)
subplot(5,3,1);plot(x(14,:));title('SI2')
subplot(5,3,2);plot(x(15,:));title('SS2')
subplot(5,3,3);plot(x(16,:));title('XI2')
subplot(5,3,4);plot(x(17,:));title('XS2')
subplot(5,3,5);plot(x(18,:));title('XBH2')
subplot(5,3,6);plot(x(19,:));title('XBA2')
subplot(5,3,7);plot(x(20,:));title('XP2')
subplot(5,3,8);plot(x(21,:));title('SO2')
subplot(5,3,9);plot(x(22,:));title('SNO2')
subplot(5,3,10);plot(x(23,:));title('SNH2')
subplot(5,3,11);plot(x(24,:));title('SND2')
xlabel('discrete time k')
subplot(5,3,12);plot(x(25,:));title('XND2')
xlabel('discrete time k')
subplot(5,3,13);plot(x(26,:));title('SALK2')
xlabel('discrete time k')
```

```
set(figure(7),'name','Tank 2 dynamic results','numbertitle','off')

% Tank 3
figure(8)
subplot(5,3,1);plot(x(27,:));title('SI3')
subplot(5,3,2);plot(x(28,:));title('SS3')
subplot(5,3,3);plot(x(29,:));title('XI3')
subplot(5,3,4);plot(x(30,:));title('XS3')
subplot(5,3,5);plot(x(31,:));title('XBH3')
subplot(5,3,6);plot(x(32,:));title('XBA3')
subplot(5,3,7);plot(x(33,:));title('XP3')
subplot(5,3,8);plot(x(34,:));title('SO3')
subplot(5,3,9);plot(x(35,:));title('SNO3')
subplot(5,3,10);plot(x(36,:));title('SNH3')
subplot(5,3,11);plot(x(37,:));title('SND3')
xlabel('discrete time k')
subplot(5,3,12);plot(x(38,:));title('XND3')
xlabel('discrete time k')
subplot(5,3,13);plot(x(39,:));title('SALK3')
xlabel('discrete time k')
set(figure(8),'name','Tank 3 dynamic results','numbertitle','off')

% Tank 4
figure(9)
subplot(5,3,1);plot(x(40,:));title('SI4')
subplot(5,3,2);plot(x(41,:));title('SS4')
subplot(5,3,3);plot(x(42,:));title('XI4')
subplot(5,3,4);plot(x(43,:));title('XS4')
subplot(5,3,5);plot(x(44,:));title('XBH4')
subplot(5,3,6);plot(x(45,:));title('XBA4')
subplot(5,3,7);plot(x(46,:));title('XP4')
subplot(5,3,8);plot(x(47,:));title('SO4')
subplot(5,3,9);plot(x(48,:));title('SNO4')
subplot(5,3,10);plot(x(49,:));title('SNH4')
subplot(5,3,11);plot(x(50,:));title('SND4')
xlabel('discrete time k')
subplot(5,3,12);plot(x(51,:));title('XND4')
xlabel('discrete time k')
subplot(5,3,13);plot(x(52,:));title('SALK4');
xlabel('discrete time k')
set(figure(9),'name','Tank 4 dynamic results','numbertitle','off')

% Tank 5
figure(10)
subplot(5,3,1);plot(x(53,:));title('SI5')
subplot(5,3,2);plot(x(54,:));title('SS5')
subplot(5,3,3);plot(x(55,:));title('XI5')
subplot(5,3,4);plot(x(56,:));title('XS5')
subplot(5,3,5);plot(x(57,:));title('XBH5')
subplot(5,3,6);plot(x(58,:));title('XBA5')
subplot(5,3,7);plot(x(59,:));title('XP5')
subplot(5,3,8);plot(x(60,:));title('SO5')
subplot(5,3,9);plot(x(61,:));title('SNO5')
subplot(5,3,10);plot(x(62,:));title('SNH5')
subplot(5,3,11);plot(x(63,:));title('SND5')
xlabel('discrete time k')
subplot(5,3,12);plot(x(64,:));title('XND5')
xlabel('discrete time k')
subplot(5,3,13);plot(x(65,:));title('SALK5');
xlabel('discrete time k')
set(figure(10),'name','Tank 5 dynamic results','numbertitle','off')

% ===================== REDUCED MODEL FORMULATION =========================
% Form the vector of the states (z) for the reduced model, based on the
% ASM1 biological model
%      SS,      SNO,     SNH
```

184

```
z1 = [x(2,:); x(9,:);  x(10,:)];       %1     2      3
z2 = [x(15,:); x(22,:); x(23,:)];      %4     5      6
z3 = [x(28,:); x(35,:); x(36,:)];      %7     8      9
z4 = [x(41,:); x(48,:); x(49,:)];      %10    11     12
z5 = [x(54,:); x(61,:); x(62,:)];      %13    14     15
z  = [z1;z2;z3;z4;z5];

% Formation of the vector of the control variables (DO Concentration)
u1 = x(8,:);
u2 = x(21,:);
u3 = x(34,:);
u4 = x(47,:);
u5 = x(60,:);
u  = [u1; u2; u3; u4; u5];

% Dynamic values of the inflow concentrations for dry weather
zi = [xi(2,:); xi(9,:); xi(10,:)];

% Steady state values for the biomass XBH, XBA, XS
XBH = [2551.76*ones(1,K+1); 2553.38*ones(1,K+1); 2557.13*ones(1,K+1);
       2559.18*ones(1,K+1); 2559.34*ones(1,K+1)];
XBA = [148.389*ones(1,K+1); 148.309*ones(1,K+1); 148.941*ones(1,K+1);
       149.527*ones(1,K+1); 149.797*ones(1,K+1)];
XS  = [82.135*ones(1,K+1); 76.386*ones(1,K+1); 64.855*ones(1,K+1);
       55.694*ones(1,K+1); 49.306*ones(1,K+1)];

% MATRIX A for the reduced model (AR)
IV1 = DT/ V1; IV11 = IV1 * Q; A11r = 1 - IV11;
IV2 = DT/ V2; IV22 = IV2 * Q; A22r = 1 - IV22;
IV3 = DT/ V3; IV33 = IV3 * Q; A33r = 1 - IV33;
IV4 = DT/ V4; IV44 = IV4 * Q; A44r = 1 - IV44;
IV5 = DT/ V5; IV55 = IV5 * Q; A55r = 1 - IV55;

A15r = IV1 * (Qa + Qr);
A21r = IV22; A32r = IV33; A43r = IV44; A54r = IV55;

A11r = A11r * ones(1,3); A11r = diag(A11r);
A22r = A22r * ones(1,3); A22r = diag(A22r);
A33r = A33r * ones(1,3); A33r = diag(A33r);
A44r = A44r * ones(1,3); A44r = diag(A44r);
A55r = A55r * ones(1,3); A55r = diag(A55r);

A15r = [A15r, A15r, A15r];
A15r = diag(A15r);

A21r = A21r * ones(1,3); A21r = diag(A21r);
A32r = A32r * ones(1,3); A32r = diag(A32r);
A43r = A43r * ones(1,3); A43r = diag(A43r);
A54r = A54r * ones(1,3); A54r = diag(A54r);

A10r = zeros(3,3); A20r = zeros(3,6); A30r = zeros(3,9);
AR = [A11r A30r A15r;A21r A22r A30r;A10r A32r A33r A20r;A20r A43r A44r A10r;A30r A54r
A55r];

% MATRIX B for the reduced model(BR)
IV0 = DT / V1*Q0;
f = 1.2;
B11r = [IV0 IV0*f^IV0];
B11r = diag(B11r);
BR = [B11r; zeros(12,3)];

% MATRIX C for the reduced model (CR)
c1 = -(1/YH);
c2 = -(1-YH)/(2.86*YH);
c3 = (1/YA);
c4 = -iXB-(1/YA);
```

185

```
    CCR = [[c1 0   -iXB ];
          [c1 c2 -iXB ];
          [0  c3 c4   ];
          [1  0  0    ]];

    CR = DT*[[CCR zeros(4,12)];[zeros(4,3) CCR zeros(4,9)];(DT/V3)*[zeros(4,6) CCR
    zeros(4,6)];
             [zeros(4,9) CCR zeros(4,3)]; [zeros(4,12) CCR]];

    % Parameter estimation
    for k = 1:K
        zk = z(:,k);
        uk = u(:,k);
        XBHk = XBH(:,k);
        XBAk = XBA(:,k);
        XSk  = XS(:,k);
        zik  = zi(:,k);
        Pk = P_BASM1R(zk,uk,XBHk,XBAk,XSk);
        FIk = CR'*Pk;
        Y(:,k) = z(:,k+1)-AR*z(:,k)-BR*zi(:,k);
        theta(:,k)=(FIk'*FIk)^(-1)*FIk'*Y(:,k);
    end

    muHE(1,:) = theta(1,:);
    muAE(1,:) = theta(2,:);
    KhE(1,:)  = theta(3,:);

    % Dynamic simulation of the reduced model with the estimated parareters
    zE = [z(1,1);z(2,1);z(3,1);z(4,1);z(5,1);z(6,1);z(7,1);z(8,1);z(9,1);z(10,1); z(11,1);
    z(12,1);z(13,1); z(14,1); z(15,1)];
    ERR = z(:,1)-zE;

    % Dynamic simulation of the reduced model with the estimated parameters
    for k = 1:K
        zEk = zE(:,k);
        uk = u(:,k);
        zik = zi(:,k);

        XBHk = XBH(:,k);
        XBAk = XBA(:,k);
        XSk = XS(:,k);

        muHEk = muHE(:,k);
        muAEk = muAE(:,k);
        KhEk  = KhE(:,k);

        Pk = P_BASM1RE(zEk,uk,XBHk,XBAk,XSk,muHEk,muAEk,KhEk);
        zE(:,k+1) = AR*zE(:,k) + CR'*Pk + BR*zi(:,k);
        ERR(:,k+1)=z(:,k+1)-zE(:,k+1);
    end

    % Vector of process full, estimated and error variables SNH, SNO,SS, Tank 1
    SS1=z(1,:);SNO1=z(2,:);SNH1=z(3,:);
    SS1E=zE(1,:);SNO1E=zE(2,:);SNH1E=zE(3,:);
    ERRSS1=ERR(1,:);ERRSNO1=ERR(2,:);ERRSNH1=ERR(3,:);
    % Calculation of the weighted errors for tank1
    eSS1=(ERRSS1*ERRSS1')/(SS1*SS1');
    eSNO1=(ERRSNO1*ERRSNO1')/(SNO1*SNO1');
    eSNH1=(ERRSNH1*ERRSNH1')/(SNH1*SNH1');

    % Vector of process full, estimated and error variables SNH, SNO,SS, Tank 2
    SS2=z(4,:);SNO2=z(5,:);SNH2=z(6,:);
    SS2E=zE(4,:);SNO2E=zE(5,:);SNH2E=zE(6,:);
    ERRSS2=ERR(4,:);ERRSNO2=ERR(5,:);ERRSNH2=ERR(6,:);
    % Calculation of the weighted errors for tank 2
    eSS2=(ERRSS2*ERRSS2')/(SS2*SS2');
```

186

```
eSNO2=(ERRSNO2*ERRSNO2')/(SNO2*SNO2');
eSNH2=(ERRSNH2*ERRSNH2')/(SNH2*SNH2');

% Vector of process full, estimated and error variables SNH, SNO,SS, Tank 3
SS3=z(7,:);SNO3=z(8,:);SNH3=z(9,:);
SS3E=zE(7,:);SNO3E=zE(8,:);SNH3E=zE(9,:);
ERRSS3=ERR(7,:);ERRSNO3=ERR(8,:);ERRSNH3=ERR(9,:);
% Calculation of the weighted errors for tank 3
eSS3=(ERRSS3*ERRSS3')/(SS3*SS3');
eSNO3=(ERRSNO3*ERRSNO3')/(SNO3*SNO3');
eSNH3=(ERRSNH3*ERRSNH3')/(SNH3*SNH3');

% Vector of process full, estimated and error variables SS, SNO,SNH, Tank 4
SS4=z(10,:);SNO4=z(11,:);SNH4=z(12,:);
SS4E=zE(10,:);SNO4E=zE(11,:);SNH4E=zE(12,:);
ERRSS4=ERR(10,:);ERRSNO4=ERR(11,:);ERRSNH4=ERR(12,:);
% Calculation of the weighted errors for tank 4
eSS4=(ERRSS4*ERRSS4')/(SS4*SS4');
eSNO4=(ERRSNO4*ERRSNO4')/(SNO4*SNO4');
eSNH4=(ERRSNH4*ERRSNH4')/(SNH4*SNH4');

% Vector of process full, estimated and error variables SS, SNO,SNH, Tank 5
SS5=z(13,:);SNO5=z(14,:);SNH5=z(15,:);
SS5E=zE(13,:);SNO5E=zE(14,:);SNH5E=zE(15,:);
ERRSS5=ERR(13,:);ERRSNO5=ERR(14,:);ERRSNH5=ERR(15,:);
% Calculation of the weighted errors for tank 5
eSS5=(ERRSS5*ERRSS5')/(SS5*SS5');
eSNO5=(ERRSNO5*ERRSNO5')/(SNO5*SNO5');
eSNH5=(ERRSNH5*ERRSNH5')/(SNH5*SNH5');

k = 1:K;
% Graphs of the estimated parameters
figure(11)
subplot(3,1,1);plot(k,muHE,'k');title('Estimated muH');ylabel('muHE');
subplot(3,1,2);plot(k,muAE,'k');title('Estimated muA');ylabel('muAE');
subplot(3,1,3);plot(k,KhE,'k');title('Estimated Kh');ylabel('KhE');
set(figure(11),'name','Estimated parameters','numbertitle','off')
xlabel('discrete time k')

k = 1:K+1;
% Graphs of the reduced model response to the dynamic influent
% concentrations
% Tank 1
figure(12)
subplot(3,2,1);plot(k,SS1); hold on;plot(k,SS1E,'k-.');ylabel('SS1,SS1E');
legend('SS1','SS1E');subplot(3,2,2); plot(k,ERRSS1,'k');ylabel('ERRSS1');

subplot(3,2,3);plot(k,SNO1); hold on;plot(k,SNO1E,'k-.');
legend('SNO1','SNO1E');
ylabel('SNO1,SNO1E');subplot(3,2,4); plot(k,ERRSNO1,'k');ylabel('ERRSNO1');

subplot(3,2,5);plot(k,SNH1); hold on;plot(k,SNH1E,'k-.');
legend('SNH1','SNH1E');
ylabel('SNH1,SNH1E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNH1,'k');ylabel('ERRSNH1');
xlabel('discrete time k')
set(figure(12),'name','Tank 1 Process variables and estimation
error','numbertitle','off')

% Tank 2
figure(13)
subplot(3,2,1);plot(k,SS2); hold on;plot(k,SS2E,'k-.');ylabel('SS2,SS2E');
legend('SS2','SS2E');subplot(3,2,2); plot(k,ERRSS2,'k');ylabel('ERRSS2');

subplot(3,2,3);plot(k,SNO2); hold on;plot(k,SNO2E,'k-.');
legend('SNO2','SNO2E');
```

187

```
ylabel('SNO2,SNO2E');
subplot(3,2,4); plot(k,ERRSNO2,'k');ylabel('ERRSNO2');

subplot(3,2,5);plot(k,SNH2); hold on;plot(k,SNH2E,'k-.');
legend('SNH2','SNH2E');
ylabel('SNH2,SNH2E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNH2,'k');ylabel('ERRSNH2');
xlabel('discrete time k')
set(figure(13),'name','Tank 2 Process variables and estimation
error','numbertitle','off')

% Tank 3
figure(14)
subplot(3,2,1);plot(k,SS3); hold on;plot(k,SS3E,'k-.');ylabel('SS3,SS3E');
legend('SS3','SS3E');subplot(3,2,2); plot(k,ERRSS3,'k');ylabel('ERRSS3');

subplot(3,2,3);plot(k,SNO3);hold on;plot(k,SNO3E,'k-.');
legend('SNO3','SNO3E');
ylabel('SNO3,SNO3E');subplot(3,2,4); plot(k,ERRSNO3,'k');ylabel('ERRSNO3');

subplot(3,2,5);plot(k,SNH3); hold on;plot(k,SNH3E,'k-.');
legend('SNH3','SNH3E');
ylabel('SNH3,SNH3E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNH3,'k');
ylabel('ERRSNH3');xlabel('discrete time k')
set(figure(14),'name','Tank 3 Process variables and estimation
error','numbertitle','off')

% Tank 4
figure(15)
subplot(3,2,1);plot(k,SS4); hold on;plot(k,SS4E,'k-.');ylabel('SS4,SS4E');
legend('SS4','SS4E');subplot(3,2,2); plot(k,ERRSS4,'k');ylabel('ERRSS4');

subplot(3,2,3);plot(k,SNO4); hold on;plot(k,SNO4E,'k-.');
ylabel('SNO4,SNO4E');legend('SNO4','SNO4E');
subplot(3,2,4); plot(k,ERRSNO4,'k');ylabel('ERRSNO4');

subplot(3,2,5);plot(k,SNH4); hold on;plot(k,SNH4E,'k-.');
legend('SNH4','SNH4E');
ylabel('SNH4,SNH4E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNH1,'k');ylabel('ERRSNH4');
ylabel('ERRSNO4');xlabel('discrete time k')
set(figure(15),'name','Tank 4 Process variables and estimation
error','numbertitle','off')

% Tank 5
figure(16)
subplot(3,2,1);plot(k,SS5); hold on;plot(k,SS5E,'k-.');ylabel('SS5,SS5E');
legend('SS5','SS5E');subplot(3,2,2); plot(k,ERRSS5,'k');ylabel('ERRSS5');

subplot(3,2,3);plot(k,SNO5); hold on;plot(k,SNO5E,'k-.');
ylabel('SNO5,SNO5E');legend('SNO5','SNO5E');
subplot(3,2,4); plot(k,ERRSNO5,'k');ylabel('ERRSNO5');

subplot(3,2,5);plot(k,SNH5); hold on;plot(k,SNH5E,'k-.');
legend('SNH5','SNH5E');
ylabel('SNH5,SNH5E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNH3,'k');
ylabel('ERRSNH5');xlabel('discrete time k')
set(figure(16),'name','Tank 5 Process variables and estimation
error','numbertitle','off')

% ======================= OPTIMAL CONTROL DATA =========================
% Declaration of the data used in the formulation of the optimal control
% problem, see chapter 4 of the Thesis
```

188

```
% Setpoints of the state and control variables for optimal control
% problem formulation
SSsp1 = 2 * ones(1,K+1); SSsp2 = 1.5 * ones(1,K+1);
SSsp3 = 0.8 * ones(1,K+1); SSsp4 = 0.7 * ones(1,K+1);
SSsp5 = 0.6 * ones(1,K+1);
SNOsp1 = 23 * ones(1,K+1); SNOsp2 = 25 * ones(1,K+1);
SNOsp3 = 25 * ones(1,K+1); SNOsp4 = 27 * ones(1,K+1);
SNOsp5 = 28 * ones(1,K+1);
SNHsp1 = 6 * ones(1,K+1); SNHsp2 = 3 * ones(1,K+1);
SNHsp3 = 1.5 * ones(1,K+1); SNHsp4 = 0.3 * ones(1,K+1);
SNHsp5 = 0.1 * ones(1,K+1);
zsp = [SSsp1; SNOsp1; SNHsp1; SSsp2; SNOsp2; SNHsp2; SSsp3; SNOsp3; SNHsp3;
       SSsp4; SNOsp4; SNHsp4; SSsp5; SNOsp5; SNHsp5];

% Constraints of the state and control variables for optimal control
% problem formulation
SSmn1 = 1 * ones(1,K+1); SSmn2 = 0.9 * ones(1,K+1);
SSmn3 = 0.4 * ones(1,K+1); SSmn4 = 0.2 * ones(1,K+1);
SSmn5 = 0.3 * ones(1,K+1);
SNOmn = 15 * ones(1,K+1);
SNHmn1 = 3 * ones(1,K+1); SNHmn2 = 1.5 * ones(1,K+1);
SNHmn3 = 0.5 * ones(1,K+1); SNHmn4 = 0.01 * ones(1,K+1);
SNHmn5 = 0.3 * ones(1,K+1);
zmin =[SSmn1; SNOmn; SNHmn1; SSmn2; SNOmn; SNHmn2; SSmn3; SNOmn; SNHmn3;
       SSmn4; SNOmn; SNHmn4; SSmn5; SNOmn; SNHmn5];

SSmx1 = 4 * ones(1, K+1); SSmx2 = 3 * ones(1,K+1);
SSmx3 = 0.6 * ones(1,K+1); SSmx4 = 0.4 * ones(1,K+1);
SSmx5 = 1.2 * ones(1,K+1);
SNOmx = 30 * ones(1,K+1);
SNHmx1 = 12 * ones(1,K+1); SNHmx2 = 6 * ones(1,K+1);
SNHmx3 = 3 * ones(1,K+1); SNHmx4 = 0.6 * ones(1,K+1);
SNHmx5 = 0.2 * ones(1,K+1);
zmax =[SSmx1; SNOmx; SNHmx1; SSmx2; SNOmx; SNHmx2; SSmx3; SNOmx; SNHmx3;
       SSmx4; SNOmx; SNHmx4; SSmx5; SNOmx; SNHmx5];

umin1 = 0.007 * ones(1, K);
umin2 = 0.008 * ones(1, K);
umin3 = 1.5 * ones(1, K);
umin4 = 2.0 * ones(1, K);
umin5 = 2.1 * ones(1, K);
umin = [umin1; umin2; umin3; umin4; umin5];

umax1 = 1.7 * ones(1, K);
umax2 = 1.8 * ones(1, K);
umax3 = 4.5 * ones(1, K);
umax4 = 4.0 * ones(1, K);
umax5 = 4.1 * ones(1, K);
umax = [umax1; umax2; umax3; umax4; umax5];

% Initial values of the coordinating and the control variables for optimal
% control problem formulation
lamda = 0.1 * ones(15,K) - 15.96;

% Number of iterations of the coordinating level
L = 50;
% Number of iterations of the first level
q1 = 20;

% Values of the error for termination of the calculations at the
% first level
epsu = 0.01;
epsz = 0.01;

% Value of the error for termination of the calculations at the
% coordinating level
```

```
epslamda = 0.01;

% Values of the steps in the gradient procedures
alu = 0.05;
alz = 0.05;
alp = 0.08;

% Values of the coefficients for changing of the step values
alpha = 2;
beta = 3;
gama = 3;

% Weighting matrices Qw and Rw in the criterion
Qw = diag([1.0 1.0 1.0 1.0 1.0 1.0 1.5 1.5 1.5 2 2 2 3 3 3]);
Rw = diag ([0.1 0.1 0.5 0.5 0.5]);
```

A.2: MATLAB script file – P_BASM1F.m

```
% M-File: P_BASM1F.m
% ============================ M-FILE DESCRIPTION ============================
% This m-file is used for:
% Calculation of the process rates for the full model

function Pk = P_BASM1F(xk)

global muH bH muA bA etag ka kh etah Ks KOH KNO KNH KOA KX;
% TANK 1
P(1) = muH*(xk(2) /(Ks+xk(2))) *(xk(8) /(KOH+xk(8))) * xk(5);
P(2) = muH*(xk(2) /(Ks+xk(2))) *(KOH/(KOH+xk(8))) * (xk(9) /(KNO+xk(9)))  * etag *
xk(5);
P(3) = muA*(xk(10)/(KNH+xk(10)))*(xk(8) /(KOA+xk(8))) * xk(6);
P(4) = bH*xk(5);
P(5) = bA*xk(6);
P(6) = ka*xk(11)*xk(5);
P(7) = kh*((xk(4) /(xk(5))) /(KX + (xk(4) /xk(5)))) *((xk(8) /(KOH+xk(8)))  + (etah *
(KOH/(KOH+xk(8)))) *(xk(9) /(KNO+xk(9)))) *xk(5);
P(8) = P(7) *(xk(12)/xk(4));

% TANK 2
P(9) = muH*(xk(15)/(Ks+xk(15)))*(xk(21)/(KOH+xk(21)))* xk(18);
P(10)= muH*(xk(15)/(Ks+xk(15)))*(KOH/(KOH+xk(21)))* (xk(22)/(KNO+xk(22))) * etag *
xk(18);
P(11)= muA*(xk(23)/(KNH+xk(23)))*(xk(21)/(KOA+xk(21)))* xk(19);
P(12)= bH*xk(18);
P(13)= bA*xk(19);
P(14)= ka*xk(24)*xk(18);
P(15)= kh*((xk(17)/(xk(18)))/(KX + (xk(17)/xk(18))))*((xk(21)/(KOH+xk(21))) + (etah *
(KOH/(KOH+xk(21))))*(xk(22)/(KNO+xk(22))))*xk(18);
P(16)= P(15) *(xk(25)/xk(17));

% TANK 3
P(17)= muH*(xk(28)/(Ks+xk(28)))*(xk(34)/(KOH+xk(34)))* xk(31);
P(18)= muH*(xk(28)/(Ks+xk(28)))*(KOH/(KOH+xk(34)))* (xk(35)/(KNO+xk(35))) * etag *
xk(31);
P(19)= muA*(xk(36)/(KNH+xk(36)))*(xk(34)/(KOA+xk(34)))* xk(32);
P(20)= bH*xk(31);
P(21)= bA*xk(32);
P(22)= ka*xk(37)*xk(31);
P(23)= kh*((xk(30)/(xk(31)))/(KX + (xk(30)/xk(31))))*((xk(34)/(KOH+xk(34))) + (etah *
(KOH/(KOH+xk(34))))*(xk(35)/(KNO+xk(35))))*xk(31);
P(24)= P(23) *(xk(38)/xk(30));

% TANK 4
P(25)= muH*(xk(41)/(Ks+xk(41)))*(xk(47)/(KOH+xk(47)))* xk(44);
```

190

```
P(26)= muH*(xk(41)/(Ks+xk(41)))*(KOH/(KOH+xk(47)))* (xk(48)/(KNO+xk(48))) * etag *
xk(44);
P(27)= muA*(xk(49)/(KNH+xk(49)))*(xk(47)/(KOA+xk(47)))* xk(45);
P(28)= bH*xk(44);
P(29)= bA*xk(45);
P(30)= ka*xk(50)*xk(44);P(33)= muH*(xk(54)/(Ks+xk(54)))*(xk(60)/(KOH+xk(60)))* xk(57);
P(31)= kh*((xk(43)/(xk(44)))/(KX + (xk(43)/xk(44))))*((xk(47)/(KOH+xk(47))) + (etah *
(KOH/(KOH+xk(47))))*(xk(48)/(KNO+xk(48))))*xk(44);
P(32)= P(31)*(xk(51)/xk(43));

% TANK 5
P(33)= muH*(xk(54)/(Ks+xk(54)))*(xk(60)/(KOH+xk(60)))* xk(57);
P(34)= muH*(xk(54)/(Ks+xk(54)))*(KOH/(KOH+xk(60)))* (xk(61)/(KNO+xk(61))) * etag *
xk(57);
P(35)= muA*(xk(62)/(KNH+xk(62)))*(xk(60)/(KOA+xk(60)))* xk(58);
P(36)= bH*xk(57);
P(37)= bA*xk(58);
P(38)= ka*xk(63)*xk(57);
P(39)= kh*((xk(56)/(xk(57)))/(KX + (xk(56)/xk(57))))*((xk(60)/(KOH+xk(60))) + (etah *
(KOH/(KOH+xk(60))))*(xk(61)/(KNO+xk(61))))*xk(57);
P(40)= P(39)*(xk(64)/xk(56));

Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20) P(21) P(22) P(23) P(24) P(25) P(26) P(27) P(28) P(29)
P(30) P(31) P(32) P(33) P(34) P(35) P(36) P(37) P(38) P(39) P(40)]';
```

A.3: MATLAB script file – P_BASM1R.m

```
% M-File: P_BASM1R.m
% ========================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model

function Pk = P_BASM1R(zk,uk,XBHk,XBAk,XSk)

global etag etah Ks KOH KNO KNH KOA KX;
% TANK 1
P(1)= (zk(1)/(Ks+zk(1)))*(uk(1)/(KOH+uk(1)))* XBHk(1);
P(2)= (zk(1)/(Ks+zk(1)))*(KOH/(KOH+uk(1)))* (zk(2)/(KNO+zk(2))) * etag * XBHk(1);
P(3)= (zk(3)/(KNH+zk(3)))*(uk(1)/(KOA+uk(1)))* XBAk(1);
P(4)= ((XSk(1)/XBHk(1))/(KX + (XSk(1)/XBHk(1))))*((uk(1)/(KOH+uk(1))) + etah *
(KOH/(KOH+uk(1)))*(zk(2)/(KNO+zk(2))))* XBHk(1);
P1k = [P(1) 0 0; P(2) 0 0; 0 P(3) 0; 0 0 P(4)];

% TANK 2
P(5)= (zk(4)/(Ks+zk(4)))*(uk(2)/(KOH+uk(2)))* XBHk(2);
P(6)= (zk(4)/(Ks+zk(4)))*(KOH/(KOH+uk(2)))* (zk(5)/(KNO+zk(5))) * etag * XBHk(2);
P(7)= (zk(6)/(KNH+zk(6)))*(uk(2)/(KOA+uk(2)))* XBAk(2);
P(8)= ((XSk(2)/XBHk(2))/(KX + (XSk(2)/XBHk(2))))*((uk(2)/(KOH+uk(2))) + etah *
(KOH/(KOH+uk(2)))*(zk(5)/(KNO+zk(5))))*XBHk(2);
P2k = [P(5) 0 0; P(6) 0 0; 0 P(7) 0; 0 0 P(8)];

% TANK 3
P(9) = (zk(7)/(Ks+zk(7)))*(uk(3)/(KOH+uk(3)))* XBHk(3);
P(10)= (zk(7)/(Ks+zk(7)))*(KOH/(KOH+uk(3)))* (zk(8)/(KNO+zk(8))) * etag * XBHk(3);
P(11)= (zk(9)/(KNH+zk(9)))*(uk(3)/(KOA+uk(3)))* XBAk(3);
P(12)= ((XSk(3)/XBHk(3))/(KX + (XSk(3)/XBHk(3))))*((uk(3)/(KOH+uk(3))) + etah *
(KOH/(KOH+uk(3)))*(zk(8)/(KNO+zk(8))))*XBHk(3);
P3k = [P(9) 0 0; P(10) 0 0; 0 P(11) 0; 0 0 P(12)];

% TANK 4
P(13)= (zk(10)/(Ks+zk(10)))*(uk(4)/(KOH+uk(4)))* XBHk(4);
P(14)= (zk(10)/(Ks+zk(10)))*(KOH/(KOH+uk(4)))* (zk(11)/(KNO+zk(11))) * etag * XBHk(4);
```

191

```
P(15)= (zk(12)/(KNH+zk(12)))*(uk(4)/(KOA+uk(4)))* XBAk(4);
P(16)= ((XSk(4)/XBHk(4))/(KX + (XSk(4)/XBHk(4))))*((uk(4)/(KOH+uk(4))) + etah *
(KOH/(KOH+uk(4)))*(zk(11)/(KNO+zk(11))))*XBHk(4);
P4k = [P(13) 0 0; P(14)  0 0; 0 P(15) 0; 0 0 P(16)];

% TANK 5
P(17)= (zk(13)/(Ks+zk(13)))*(uk(5)/(KOH+uk(5)))* XBHk(5);
P(18)= (zk(13)/(Ks+zk(13)))*(KOH/(KOH+uk(5)))* (zk(14)/(KNO+zk(14))) * etag * XBHk(5);
P(19)= (zk(15)/(KNH+zk(15)))*(uk(5)/(KOA+uk(5)))* XBAk(5);
P(20)= ((XSk(5)/XBHk(5))/(KX + (XSk(5)/XBHk(5))))*((uk(5)/(KOH+uk(5))) + etah *
(KOH/(KOH+uk(5)))*(zk(14)/(KNO+zk(14))))*XBHk(5);
P5k = [P(17) 0 0; P(18) 0 0; 0 P(19) 0; 0 0 P(20)];

%Matrix Pk for the whole process
Pk = [P1k; P2k; P3k; P4k; P5k];
```

A.4: MATLAB script file – P_BASM1RE.m

```
% M-File: P_BASM1RE.m
% =================== M-FILE DESCRIPTION ===========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model

% Description of the function function
function Pk = P_BASM1RE(zEk,uk,XBHk,XBAk,XSk,muHEk,muAEk,KhEk)

global etag etah Ks KOH KNO KNH KOA KX;
% TANK 1
P(1)= muHEk  * (zEk(1)/(Ks+zEk(1)))*(uk(1)/(KOH+uk(1)))* XBHk(1);
P(2)= muHEk  * (zEk(1)/(Ks+zEk(1)))*(KOH/(KOH+uk(1)))* (zEk(2)/(KNO+zEk(2))) * etag *
XBHk(1);
P(3)= muAEk  * (zEk(3)/(KNH+zEk(3)))*(uk(1)/(KOA+uk(1)))* XBAk(1);
P(4)= KhEk   * ((XSk(1)/XBHk(1))/(KX + (XSk(1)/XBHk(1))))*((uk(1)/(KOH+uk(1))) + etah *
(KOH/(KOH+uk(1)))*(zEk(2)/(KNO+zEk(2))))* XBHk(1);

% TANK 2
P(5)= muHEk  * (zEk(4)/(Ks+zEk(4)))*(uk(2)/(KOH+uk(2)))* XBHk(2);
P(6)= muHEk  * (zEk(4)/(Ks+zEk(4)))*(KOH/(KOH+uk(2)))* (zEk(5)/(KNO+zEk(5))) * etag *
XBHk(2);
P(7)= muAEk  * (zEk(6)/(KNH+zEk(6)))*(uk(2)/(KOA+uk(2)))* XBAk(2);
P(8)= KhEk   * ((XSk(2)/XBHk(2))/(KX + (XSk(2)/XBHk(2))))*((uk(2)/(KOH+uk(2))) + etah *
(KOH/(KOH+uk(2)))*(zEk(5)/(KNO+zEk(5))))*XBHk(2);

% TANK 3
P(9) = muHEk  * (zEk(7)/(Ks+zEk(7)))*(uk(3)/(KOH+uk(3)))* XBHk(3);
P(10)= muHEk  * (zEk(7)/(Ks+zEk(7)))*(KOH/(KOH+uk(3)))* (zEk(8)/(KNO+zEk(8))) * etag *
XBHk(3);
P(11)= muAEk  * (zEk(9)/(KNH+zEk(9)))*(uk(3)/(KOA+uk(3)))* XBAk(3);
P(12)= KhEk   * ((XSk(3)/XBHk(3))/(KX + (XSk(3)/XBHk(3))))*((uk(3)/(KOH+uk(3))) + etah *
(KOH/(KOH+uk(3)))*(zEk(8)/(KNO+zEk(8))))*XBHk(3);

% TANK 4
P(13)= muHEk  * (zEk(10)/(Ks+zEk(10)))*(uk(4)/(KOH+uk(4)))* XBHk(4);
P(14)= muHEk  * (zEk(10)/(Ks+zEk(10)))*(KOH/(KOH+uk(4)))* (zEk(11)/(KNO+zEk(11))) * etag
* XBHk(4);
P(15)= muAEk  * (zEk(12)/(KNH+zEk(12)))*(uk(4)/(KOA+uk(4)))* XBAk(4);
P(16)= KhEk   * ((XSk(4)/XBHk(4))/(KX + (XSk(4)/XBHk(4))))*((uk(4)/(KOH+uk(4))) + etah *
(KOH/(KOH+uk(4)))*(zEk(11)/(KNO+zEk(11))))*XBHk(4);

% TANK 5
P(17)= muHEk  * (zEk(13)/(Ks+zEk(13)))*(uk(5)/(KOH+uk(5)))* XBHk(5);
P(18)= muHEk  * (zEk(13)/(Ks+zEk(13)))*(KOH/(KOH+uk(5)))* (zEk(14)/(KNO+zEk(14))) * etag
* XBHk(5);
```

```
P(19)= muAEk * (zEk(15)/(KNH+zEk(15)))*(uk(5)/(KOA+uk(5)))* XBAk(5);
P(20)= KhEk  * ((XSk(5)/XBHk(5))/(KX + (XSk(5)/XBHk(5))))*((uk(5)/(KOH+uk(5))) + etah *
(KOH/(KOH+uk(5)))*(zEk(14)/(KNO+zEk(14))))*XBHk(5);


%Matrix Pk for the whole process
Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20)]';
```

A.5: MATLAB script file – BASM1_Main.m

```
% M-file: BASM1_Main.m
% ======================= M-FILE DESCRIPTION  =======================
% This m-file contains the main program for calculation of the optimal DO
% trajectory for Benchmark plant with ASM1 biological model

% Declaration of the global variables
global  bH bA etag ka etah Ks KOH KNO KNH KOA KX

% Calls the function with simulation results
BASM1_Simulation

l = 1;                      % Initialization of the initial iteration step
while l<=L                  % The program is repeated untill the optimization
                            % horizon is reached

    for k = 1:K-1

        % Recalls the model predicted state and control trajectories
        zk = z(:,k);
        uk = u(:,k);
        % Recalls the steady state trajectory for the biomass XBH, XBA, XS
        XBHk = XBH(:,k);
        XBAk = XBA(:,k);
        XSk  = XS(:,k);
        % Recalls the minimum and maximum limits of the controls
        umink = umin(:,k);
        umaxk = umax(:,k);
        % Recalls the initial trajectory of the coordinating variables
        lamdak = lamda(:,k);
        % Calculates the process rates based on the trajectories above
        gk = BASM1_du(zk, uk, XBHk, XBAk, XSk, muHEk, muAEk, KhEk);

        k1 = k+1; %
        muHEk1 = muHE(:,k1);
        muAEk1 = muAE(:,k1);
        KhEk1  = KhE(:,k1);
        % Recalls the model predicted state and control trajectories
        zk1 = z(:,k1);
        uk1 = u(:,k1);
        % Recalls the steady state trajectory for the biomass XBH, XBA, XS
        XBHk1 = XBH(:,k1);
        XBAk1 = XBA(:,k1);
        XSk1 = XS(:,k1);
        % Recalls the minimum and maximum limits of the states
        zmink1 = zmin(:,k1);
        zmaxk1 = zmax(:,k1);
        % Recalls the initial trajectory of the coordinating variables
        lamdak1= lamda(:,k1);
        % Recalls the states set points
        zspk1 = zsp(:,k1);

        % Calculates the process rates based on the trajectories above
        dk = BASM1_dz(zk1, uk1, XBHk1, XBAk1, XSk1, muHEk1, muAEk1, KhEk1);

        % Calculates the gradients of the control variables
```

193

```matlab
        uk = BASM1_ugrad(uk,lamdak,Rw,CR,gk,alu,epsu,umink,umaxk,gama,q1);
        % Assigns the control variables with new values
        u(:,k) = uk;

        % Calculates the gradients of the state variables
        zk1 = BASM1_zgrad(zk,lamdak1,lamdak, Qw, CR, dk, alz, epsz, zmink1, zmaxk1,AR,
beta, zspk1, q1);
        % Assigns the state variables with new values
        z(:,k1) = zk1;
    end

% Calculation of the last value of the state vector
% Recalls initial conditions of the state variables
%K = K+1;
lamdak1 = lamda(:,K); % Recalls cordinating variables for the final moment
zmink1 = zmin(:,K+1); % Recalls minimum state value for the final moment
zmaxk1 = zmax(:,K+1); % Recalls maximum state value for the final moment
zspk1 = zsp(:,K+1);    % Recalls state set point for the final moment

zk1 = BASM1_zKgrad(lamdak, Qw, zmink1, zmaxk1, zspk1);;
% Assigns the state variables at the final moment with new values
z(:,K+1) = zk1;

normp1 = 0;
for k=1:K
        % Recalls initial conditions of the state variables
        %zik = zi(:,k);
        zk = z(:,k);
        uk = u(:,k);
        XBHk = XBH(:,k);
        XBAk = XBA(:,k);
        XSk  = XS(:,k); %?
        % Calculates the process rates based on the reduced model
        Pk = P_BASM1RE(zk,uk,XBHk,XBAk,XSk,muHEk,muAEk,KhEk);
        elamda(:,k) = -z(:,k+1)+AR*z(:,k)+CR'*Pk+BR*zi(:,k);   % calculation of the
gradient
        P(:,k) = Pk;
        normp1 = normp1+sum(elamda(:,k).*elamda(:,k));
    end

    normp = sqrt(normp1);
%check the value of the vector of the conjugate variables
if normp <= epslamda
    break
else l = l + 1;
    lamda = lamda + alp*elamda;
    alp = alp/alpha;
end
end

% Calculation of the criterion based on the optima trajectories
J = BASM1_criterion (z,u,zsp,Qw,Rw,K);

% printing the results
J;
normp;
alp;

% printing the conjugate variables and the ep(k)
figure(17);
subplot(2,1,1);plot(lamda');
xlabel('Time');ylabel('Coordinating variables Y(k)');
title('Lagrange multipliers trajectory');
subplot(2,1,2);plot(elamda')
xlabel('Time');ylabel('ep(k)');
title('Lagrange multipliers gradient');
```

194

```
set(figure(17),'name','Trajectory and gradient of the Lagrange
multipliers','numbertitle','off')

% printing the state and control variables for every tank
% Tank 1
figure(18)
subplot(5,3,1);plot(z(1,:));title('SS1');
subplot(5,3,2);plot(z(2,:));title('SNO1')
subplot(5,3,3);plot(z(3,:));title('SNH1')

% Tank 2
subplot(5,3,4);plot(z(4,:));title('SS2')
subplot(5,3,5);plot(z(5,:));title('SNO2')
subplot(5,3,6);plot(z(6,:));title('SNH2');

% Tank 3
subplot(5,3,7);plot(z(7,:));title('SS3')
ylabel('Readily biodegradable substrate')
subplot(5,3,8);plot(z(8,:));title('SNO3')
ylabel('Nitrate & nitrite nitrogen')
subplot(5,3,9);plot(z(9,:));title('SNH3')
ylabel('Ammonia & ammonium nitrogen')

% Tank 4
subplot(5,3,10);plot(z(10,:));title('SS4')
subplot(5,3,11);plot(z(11,:));title('SNO4')
subplot(5,3,12);plot(z(12,:));title('SNH4');

% Tank 5
subplot(5,3,13);plot(z(13,:));xlabel('discrete time k');title('SS5')
subplot(5,3,14);plot(z(14,:));xlabel('discrete time k');title('SNO5')
subplot(5,3,15);plot(z(15,:));xlabel('discrete time k');title('SNH5')
set(figure(18),'name','Optimal States Trajectories','numbertitle','off')

figure(19);
subplot(5,1,1);plot(u(1,:));title('SO1');ylabel('Dissolved oxygen');
subplot(5,1,2);plot(u(2,:));title('SO2');ylabel('Dissolved oxygen');
subplot(5,1,3);plot(u(3,:));title('SO3');ylabel('Dissolved oxygen');
subplot(5,1,4);plot(u(4,:));title('SO4');ylabel('Dissolved oxygen');
subplot(5,1,5);plot(u(5,:));title('SO5');ylabel('Dissolved oxygen');
xlabel('discrete time k')
set(figure(19),'name','Optimal Controls Trajectories','numbertitle','off')
```

A.6: MATLAB script file – BASM1_du.m

```
% M-File: BASM1_du.m
% ========================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Partial derivative of the Lagrangian with respect to u; dp(z,u)/du

function gk = BASM1_du(zk, uk, XBHk, XBAk, XSk, muHEk, muAEk, KhEk)
global etag etah Ks KOH KNO KNH KOA KX
% For tank 1
g11 = muHEk*XBHk(1)*(zk(1)/(Ks+zk(1)))*(KOH/(KOH+uk(1))^2);
g12 = muHEk*(zk(1)/(Ks+zk(1)))*(zk(2)/(KNO+zk(2)))*etag*XBHk(1)*(-KOH/(KOH+uk(1))^2);
g13 = muAEk*(zk(3)/(KNH+zk(3)))*(KOA/(KOA+uk(1))^2)*XBAk(1);
g14 = (KhEk*XSk(1)/(KX+(XSk(1)/(XBHk(1)))))*((KOH/(KOH+uk(1))^2)+etah*(-
KOH/(KOH+uk(1))^2)*(zk(2)/KNO+zk(2)));
% For tank 2
g21 = muHEk*XBHk(2)*(zk(4)/(Ks+zk(4)))*(KOH/(KOH+uk(2))^2);
g22 = muHEk*(zk(4)/(Ks+zk(4)))*(zk(5)/(KNO+zk(5)))*etag*XBHk(2)*(-KOH/(KOH+uk(2))^2);
g23 = muAEk*(zk(6)/(KNH+zk(6)))*(KOA/(KOA+uk(2))^2)*XBAk(2);
g24 = (KhEk*XSk(2)/(KX+(XSk(2)/(XBHk(2)))))*((KOH/(KOH+uk(2))^2)+etah*(-
KOH/(KOH+uk(2))^2)*(zk(5)/KNO+zk(5)));
```

195

```
% For tank 3
g31 = muHEk*XBHk(3)*(zk(7)/(Ks+zk(7)))*(KOH/(KOH+uk(3))^2);
g32 = muHEk*(zk(7)/(Ks+zk(7)))*(zk(8)/(KNO+zk(8)))*etag*XBHk(3)*(-KOH/(KOH+uk(3))^2);
g33 = muAEk*(zk(9)/(KNH+zk(9)))*(KOA/(KOA+uk(3))^2)*XBAk(3);
g34 = (KhEk*XSk(3)/(KX+(XSk(3)/(XBHk(3)))))*((KOH/(KOH+uk(3))^2)+etah*(-
KOH/(KOH+uk(3))^2)*(zk(8)/KNO+zk(8)));
% For tank 4
g41 = muHEk*XBHk(4)*(zk(10)/(Ks+zk(10)))*(KOH/(KOH+uk(4))^2);
g42 = muHEk*(zk(10)/(Ks+zk(10)))*(zk(11)/(KNO+zk(11)))*etag*XBHk(4)*(-
KOH/(KOH+uk(4))^2);
g43 = muAEk*zk(12)/(KNH+zk(12))*(KOA/(KOA+uk(4))^2)*XBAk(4);
g44 = (KhEk*XSk(4)/(KX+(XSk(4)/(XBHk(4)))))*((KOH/(KOH+uk(4))^2)+etah*(-
KOH/(KOH+uk(4))^2)*(zk(11)/KNO+zk(11)));
% For tank 5
g51 = muHEk*XBHk(5)*(zk(13)/(Ks+zk(13)))*(KOH/(KOH+uk(5))^2);
g52 = muHEk*(zk(13)/(Ks+zk(13)))*(zk(14)/(KNO+zk(14)))*etag*XBHk(5)*(-
KOH/(KOH+uk(5))^2);
g53 = muAEk*zk(15)/(KNH+zk(15))*(KOA/(KOA+uk(5))^2)*XBAk(5);
g54 = (KhEk*XSk(5)/(KX+(XSk(5)/(XBHk(5)))))*((KOH/(KOH+uk(5))^2)+etah*(-
KOH/(KOH+uk(5))^2)*(zk(14)/KNO+zk(14)));
%Building the matrix gk
gk = [g11 0 0 0 0; g12 0 0 0 0; g13 0 0 0 0; g14 0 0 0 0;
       0 g21 0 0 0; 0 g22 0 0 0; 0 g23 0 0 0; 0 g24 0 0 0;
       0 0 g31 0 0; 0 0 g32 0 0; 0 0 g33 0 0; 0 0 g34 0 0;
       0 0 0 g41 0; 0 0 0 g42 0; 0 0 0 g43 0; 0 0 0 g44 0;
       0 0 0 0 g51; 0 0 0 0 g52; 0 0 0 0 g53; 0 0 0 0 g54];

% The matrix gk is used for calculation of the gragient eu(k) at every moment k=0,k-1
```

A.7: MATLAB script file – BASM1_dz.m

```
% M-File: BASM1_dz.m
% ========================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Partial derivative of the Lagrangian with respect to z; dp(z,u)/dz

function dk = BASM1_dz(zk, uk, XBHk, XBAk, XSk, muHEk, muAEk, KhEk)
global  etag etah Ks KOH KNO KNH KOA KX
% For tank 1
d11 = muHEk*XBHk(1)*(uk(1)/(KOH+uk(1))* (Ks/(Ks+zk(1))^2));
d12 = muHEk*(etag*XBHk(1)*(KOH/KOH+uk(1))*(zk(2)/(KNO+zk(2)))*(Ks/(Ks+zk(1))^2));
d13 = muHEk*etag*XBHk(1)*(zk(1)/(Ks+zk(1)))*(KOH/KOH+uk(1))*(KNO/(KNO+zk(2))^2);
d14 = muAEk*XBAk(1)*(uk(1)/(KOA+uk(1)))*(KNH/(KNH+zk(3))^2);
d15 = KhEk*(XSk(1)/(KX+(XSk(1)/(XBHk(1)))))*etah*(KOH/(KOH+uk(1)))*(KNO/(KNO+zk(2))^2);

% For tank 2
d21 = muHEk*XBHk(2)*(uk(2)/(KOH+uk(2))* (Ks/(Ks+zk(4))^2));
d22 = muHEk*etag*XBHk(2)*(KOH/KOH+uk(2))*(zk(5)/(KNO+zk(5)))*(Ks/(Ks+zk(4))^2);
d23 = muHEk*etag*XBHk(2)*(KOH/KOH+uk(2))*(KNO/(KNO+zk(5))^2)*(zk(4)/(Ks+zk(4)));
d24 = muAEk*XBAk(2)*(uk(2)/(KOA+uk(2)))*(KNH/(KNH+zk(6))^2);
d25 = KhEk*(XSk(2)/(KX+(XSk(2)/(XBHk(2)))))*etah*(KOH/(KOH+uk(2)))*(KNO/(KNO+zk(5))^2);
% For tank 3
d31 = muHEk*XBHk(3)*(uk(3)/(KOH+uk(3))* (Ks/(Ks+zk(7))^2));
d32 = muHEk*etag*XBHk(3)*(KOH/KOH+uk(3))*(zk(8)/(KNO+zk(8)))*(Ks/(Ks+zk(7))^2);
d33 = muHEk*etag*XBHk(3)*(KOH/KOH+uk(3))*(KNO/(KNO+zk(8))^2)*(zk(7)/(Ks+zk(7)));
d34 = muAEk*XBAk(3)*(uk(3)/(KOA+uk(3)))*(KNH/(KNH+zk(9))^2);
d35 = KhEk*(XSk(3)/(KX+(XSk(3)/(XBHk(3)))))*etah*(KOH/(KOH+uk(3)))*(KNO/(KNO+zk(8))^2);
% For tank 4
d41 = muHEk*XBHk(4)*(uk(4)/(KOH+uk(4))* (Ks/(Ks+zk(10))^2));
d42 = muHEk*etag*XBHk(4)*(KOH/KOH+uk(4))*(zk(11)/(KNO+zk(11)))*(Ks/(Ks+zk(10))^2);
d43 = muHEk*etag*XBHk(4)*(KOH/KOH+uk(4))*(KNO/(KNO+zk(11))^2)*(zk(10)/(Ks+zk(10)));
d44 = muAEk*XBAk(4)*(uk(4)/(KOA+uk(4)))*(KNH/(KNH+zk(12))^2);
d45 =
KhEk*(XSk(4)/(KX+(XSk(4)/(XBHk(4)))))*etah*(KOH/(KOH+uk(4)))*(KNO/(KNO+zk(11))^2);
```

```
% For tank 5
d51 = muHEk*XBHk(5)*(uk(5)/(KOH+uk(5))* (Ks/(Ks+zk(13))^2));
d52 = muHEk*etag*XBHk(5)*(KOH/KOH+uk(5))*(zk(14)/(KNO+zk(14)))*(Ks/(Ks+zk(13))^2);
d53 = muHEk*etag*XBHk(5)*(KOH/KOH+uk(5))*(KNO/(KNO+zk(14))^2)*(zk(13)/(Ks+zk(13)));
d54 = muAEk*XBAk(5)*(uk(5)/(KOA+uk(5)))*(KNH/(KNH+zk(15))^2);
d55 =
KhEk*(XSk(5)/(KX+(XSk(5)/(XBHk(5)))))*etah*(KOH/(KOH+uk(5)))*(KNO/(KNO+zk(14))^2);
% Building the matrix dk
% The matrix dk is used for calculation of the gradient ex(k) at every moment k=0,k-1
dk = [d11 0 0 0 0 0 0 0 0 0 0 0 0 0 0; d12 d13 0  0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 d14 0 0 0 0 0 0 0 0 0 0 0 0; 0 d15 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 d21 0 0  0 0 0 0 0 0 0 0; 0 0 0 d22 d23 0  0 0 0 0 0 0 0 0 0;
      0 0 0  0 0 d24 0 0 0 0 0 0 0 0; 0 0 0 0 d25 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 d31 0 0  0 0.0 0 0 0; 0 0 0 0 0 0 d32 d33 0  0 0 0 0 0;
      0 0 0 0 0 0  0 0 d34 0 0 0 0 0; 0 0 0 0 0 0 0 d35 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 d41 0 0  0 0 0; 0 0 0 0 0 0 0 0 0 d42 d43 0  0 0 0;
      0 0 0 0 0 0 0 0 0  0 0 d44 0 0 0; 0 0 0 0 0 0 0 0 0 0 0 d45 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 d51 0 0 ; 0 0 0 0 0 0 0 0 0 0 0 0 0 d52 d53 0 ;
      0 0 0 0 0 0 0 0 0 0 0 0  0 0 d54; 0 0 0 0 0 0 0 0 0 0 0 0 0 0 d55 0];
```

A.8: MATLAB script file – BASM1_ugrad.m

```
% M-File: BASM1_ugrad.m
% ============================== M-FILE DESCRIPTION ==========================
% This m-file is used for:
% Calculation of the partial derivative of the Lagrangian with respect to u
% This function also checks if the conditions for optimality are met

function uk = BASM1_ugrad(uk,lamdak,Rw,CR,gk,alu,epsu,umink,umaxk,gama,q1)
q = 1;% q is the index of the gradient procedure
while q <= q1
    euk = Rw * uk + (CR'*gk)'*lamdak;
    normeu = norm(euk);          % calculation of the norm of the gradient
    if normeu <= epsu
        uk = uk;
        break
    else q = q+1; % iterate q to improve uk
        uk = uk - alu*euk ;
        alu = alu/gama;
    end      % end condition for norm
end            % end condition for number of step of the gradient procedure

% checking for the control constraints
for i = 1:5
if  uk(1) < umink(1)
    uk(1) = umink(1);
else if uk(1) > umaxk(1)
        uk(1) = umaxk(1);
else uk(1) = uk(1);
end
end
end
```

A.9: MATLAB script file – BASM1_zgrad.m

```
% M-File: BASM1_zgrad.m
% ============================== M-FILE DESCRIPTION ==========================
% This m-file is used for:
% Partial derivative of the Lagrangian with respect to z
% This function also checks if the conditions for optimality are met
```

```
function zk = BASM1_zgrad(zk,lamdak1,lamdak, Qw, CR, dk, alz, epsz, zmink1, zmaxk1,AR,
beta, zspk1, q1)

q = 1;
while q <= q1
    ezk = -Qw * (zspk1-zk)-lamdak + AR'*lamdak + (CR'*dk)'*lamdak1;
    normez = norm(ezk);
    if normez<= epsz
        zk = zk ;
        break
    else q = q+1;
        zk = zk - alz*ezk;
        alz = alz/beta;
    end
end

% checking for the state constraints
for i = 1:15

if zk(1) < zmink1(1)
    zk(1) = zmink1(1);
else if zk(1) > zmaxk1(1)
        zk(1) = zmaxk1(1);
else zk(1) = zk(1);
end
end
end
```

A.10: MATLAB script file – BASM1_zKgrad.m

```
% M-File: BASM1_zKgrad.m
% ======================= M-FILE DESCRIPTION =======================
% This m-file is used for:
% Calculation for the value of vector z at the last moment

function zK1 = BASM1_zKgrad(lamdak, Qw, zmink1, zmaxk1, zspk1,n)
zK1 = -zspk1+inv(Qw)*lamdak;

% checking for the state constraints in Tank 1
for i = 1:15
    if zK1(i) <= zmink1(i)
        zK1(i) = zmink1(i);
    else if zK1(i) >= zmaxk1(i)
            zK1(i) = zmaxk1(i);
        else zK1(i) = zK1(i);
        end
    end
end
```

A.11: MATLAB script file – BASM1_lamdagrad.m

```
% M-File: BASM1_lamdagrad.m
% ======================= M-FILE DESCRIPTION =======================
% This m-file is used for:
% Partial derivative of the Lagrangian with respect to lamda
% This function also checks if the conditions for optimality are met

function [lamdak,normp, elamdak] = BASM1_lamdagrad(zk, zk1, zik, uk, XBHk, XBAk,lamdak,
Pk, AR, CR, BR, zi, epslamda, q1, alp, alpha)

q = 1;
normp1 = 0;
```

```
while q <= q1
    elamdak = -zk1+AR*zk+CR'*Pk+BR*zik;  % calculation of the gradient
    normp1 = normp1+sum(elamdak.*elamdak);
    normp = sqrt(normp1);

%check the value of the vector of the conjugate variables
if normp <= epslamda
    lamdak = lamdak;
    l = l + 1;
    break
else q = q+1;
    lamdak = lamdak + alp*elamdak;
    alp = alp/alpha;
end
end
```

A.12: MATLAB script file – BASM1_criterion.m

```
% M-File: BASM1_criterion.m
% ========================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Calculation for the value of the criterion

function J = BASM1_criterion(z, u, zsp, Qw, Rw, K)
J = 0.0;
for k = 1:K
J=J + sum((zsp(:,k)-z(:,k))'*Qw*(zsp(:,k)-z(:,k))+sum (u(:,k)'*Rw*u(:,k)));
end
J = J + sum((zsp(:,K+1)-z(:,K+1)'*Qw*(zsp(:,K+1)-z(:,K+1))));
```

APPENDIX B: DEVELOPED MATLAB CODE FOR OPTIMAL CONTROL OF THE BENCHMARK PLANT WITH THE ACTIVATED SLUDGE MODEL NO.2 BIO-MODEL

B.1: MATLAB script file – BASM2_Simulation.m

```
% M-File: BASM2_Simulation.m
% ======================== M-FILE DESCRIPTION ========================
% This m-file is used for:
% Simulation of the Benchmark plant using full ASM2 model
% Calculation of the parameters for the reduced ASM2 model
% Simulation of the Benchmark plant using the reduced ASM2 model

clear all          % Clear the workspace
clc                % Clear the command window

% ======================== PARAMETER DEFINITION ========================
% Declaration of the global variables
global fSI YH fXI YPO4 YPHA iPBM YA iNBM Kh etaNO3 etafe muH qfe bH qPHA
global qPP muPAO bPAO bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4
global KALK KA Kfe KPP KPS KP KPHA KIPP KMAX

K1  = 10;          % Prediction horizon or interval
DT = 0.001042;       % Sampling time

% Initialization of the model parameters, typical values at 20 0C
% Stochiometric parameters
fSI     = 0.0;
YH      = 0.625;
fXI     = 0.10;
%YPAO = 0.625;
YPO4    = 0.40;
YPHA    = 0.20;
iPBM    = 0.02;
YA      = 0.24;
iNBM    = 0.07;
% Kinetic parameters
Kh      = 3;
etaNO3  = 0.8;
etafe   = 0.40;
muH     = 0.6;
qfe     = 3;
bH      = 0.4;
qPHA    = 3;
qPP     = 1.5;
muPAO   = 1;
bPAO    = 0.2;
bPP     = 0.2;
bPHA    = 0.2;
muAUT   = 1;
bAUT    = 0.15;
kPRE    = 1;
kRED    = 0.6;
% Monod switching functions
KO2     = 0.2;
KX      = 0.1;
KNO3    = 0.5;
KF      = 4;
KNH4    = 0.05;
KALK    = 0.1;
KA      = 4;
Kfe     = 20;
KPP     = 0.01;
KPS     = 0.20;
```

```
KP       = 0.01;
KPHA     = 0.01;
KIPP     = 0.02;
KMAX     = 0.34;

% Initialization of the Benchmark process data
V1   = 1000;        % Tank 1 volume
V2   = 1000;        % Tank 2 volume
V3   = 1333;        % Tank 3 volume
V4   = 1333;        % Tank 4 volume
V5   = 1333;        % Tank 5 volume

Q0   = 18446;       % Influent flow rate
Qa   = 55338;       % Internal recirculation rate
Qr   = 18446;       % Sludge recycled rate
Qw   = 385;         % Wasted sludge rate
Q    = Q0 + Qa + Qr;% Flow through the tanks

% Coeficient for fraction of flow rates for the particulate materials
LAMBDA = (Q0+Qr)/(Qr+Qw);

% Oxygen parameters
SOsat = 8;          % Oxygen saturation point
KLa3  = 240;        % Oxygen transfer function in Tank 3
KLa4  = 240;        % Oxygen transfer function in Tank 4
KLa5  = 84;         % Oxygen transfer function in Tank 5

KL = [KLa3 KLa4 KLa5];
KLa = diag(KL);

% ========================= PROCESS MODEL =========================
% Calculation of the MATRIX A
% Matrix A represent weighting factors of the accumulated state variables
% from different flows towards state variables in each tank as derived
% through mass balance principles, see chapter 3 of the Thesis
IV1 = DT/ V1; IV11 = IV1 * Q; A11 = 1 - IV11;
IV2 = DT/ V2; IV22 = IV2 * Q; A22 = 1 - IV22;
IV3 = DT/ V3; IV33 = IV3 * Q; A33 = 1 - IV33;
IV4 = DT/ V4; IV44 = IV4 * Q; A44 = 1 - IV44;
IV5 = DT/ V5; IV55 = IV5 * Q; A55 = 1 - IV55;

A15a = IV1 * (Qa + Qr);
A15b = IV1 * (Qa + LAMBDA*Qr);
A21 = IV22;
A32 = IV33;
A43 = IV44;
A54 = IV55;

A11 = A11 * ones(1,19); A11 = diag(A11);
A22 = A22 * ones(1,19); A22 = diag(A22);
A33 = A33 * ones(1,19); A33 = diag(A33);
A44 = A44 * ones(1,19); A44 = diag(A44);
A55 = A55 * ones(1,19); A55 = diag(A55);

A15 =
[A15a,A15a,A15a,A15a,A15a,A15a,A15a,A15a,A15a,A15b,A15b,A15b,A15b,A15b,A15b,A15b,A15b,A15b,A15b];
A15 = diag(A15);
A21 = A21 * ones(1,19); A21 = diag(A21);
A32 = A32 * ones(1,19); A32 = diag(A32);
A43 = A43 * ones(1,19); A43 = diag(A43);
A54 = A54 * ones(1,19); A54 = diag(A54);

A10 = zeros(19,19); A20 = zeros(19,38); A30 = zeros(19,57);

A = [A11 A30 A15;A21 A22 A30;A10 A32 A33 A20;A20 A43 A44 A10;A30 A54 A55];
```

```
% Calculation of the MATRIX B
% Matrix B represent weighting factors of the state variables towards the
% influent wastewater composition as derived through mass balance
% principles, see chapter 3 of the Thesis
IV0 = DT/ V1*Q0;
B11 = IV0 * ones(1,19); B11 = diag(B11);
B   = [B11; zeros(76,19)];


% Formulation of the MATRIX C
% Matrix C represent weighting factors of the state variables towards the
% processes on the ASM1 matrix table
%      SO2    SF    SA      SNH4    SNO3   SPO4     SI      SALK    SN2  XI
%      XS      XH     XPAO    XPP    XPHA   XAUT    XTSS    XMeOH  XMeP
CC = [[0   1  0 0.01    0 0   0 0.001   0 0 -1   0 0 0 0 0      -0.75  0     0];
      [0   1  0 0.01    0 0   0 0.001   0 0 -1   0 0 0 0 0      -0.75  0     0];
      [0   1  0 0.01    0 0   0 0.001   0 0 -1   0 0 0 0 0      -0.75  0     0];
      [-0.59 -1.59  0 -0.022   0 -0.004  0 -0.001 0 0 0 1 0    0     0     0      0.9     0
0];
      [-0.59  0  -1.59    -0.0070   0 -0.02   0 0.021 0 0 0 1 0 0   0     0      0.9     0
0];
      [0   -1.59  0 -0.022    -0.21 -0.004   0 0.014 0.21  0 0 1 0 0    0     0      0.9     0
0];
      [0   0  -1.59    -0.07    -0.21 -0.02   0 0.036 0.21  0 0 1 0 0 0    0     0.9     0
0];
      [0   -1 1 0.03    0 0.01   0 0.014 0 0 0    0      0      0      0     0      0      0];
      [0   0  0 0.031    0 0   0.01   0 0 0.002 0 0.10 0.9  -1   0 0    0     0      -0.15    0
0];
      [0   0  -1  0    0 0.4   0 -0.004 0 0 0 0 0    -0.4 1   0     -0.69    0     0];
      [-0.2  0  0 0    0 -1 0 0.048 0 0 0 0 0    1   -0.20    0      3.11    0     0];
      [-0.6  0  0 -0.07    0 -0.02 0 -0.0004 0 0 0 0 1   0    -1.6 0    -0.06    0
0];
      [0   0  0 0.031    0 0.01   0 0.002 0 0.10 0.9  0 -1   0 0    0     -0.15    0     0];
      [0   0  0 0    0 1   0 -0.048 0 0 0    0      0      -1   0     0      -3.23    0     0];
      [0   0  1 0    0 0   0 -0.0016 0 0 0    0      0      0      -1   0    -0.6    0     0];
      [-18   0  0 -4.24    4.17  -0.02 0 -0.6 0 0 0 0 0 0    1    0.9    0     0];
      [0   0  0 0.031    0 0.01   0 0.002 0 0.10 0.9  0 0 0    0     -1   -0.15    0     0];
      [0   0  0 0    0 -1 0 0 0.048    0 0 0 0 0    0      0      1.42 -3.45 4.87];
      [0   0  0 0    0 1   0 0 -0.048    0 0 0 0 0    0      -1.42    3.45 -4.87]];

C = DT*[CC zeros(19,76); zeros(19,19) CC zeros(19,57);
        zeros(19,38) CC zeros(19,38); zeros(19,57) CC zeros(19,19);
        zeros(19,76) CC];


% ===================== SIMULATION DATA =======================
% Average values of the inflow concentrations based on tne dry weather file
SO2i = 0.0;     SFi   = 30;    SAi    = 20;   SNH4i = 31.56; SNO3i = 0.0;
SPO4i = 3.6;    SIi   = 30;    SALKi = 7;    SN2i = 6.95;   XIi   = 51.2;
XSi   = 202.32; XHi   = 28.17; XPAOi  = 0.0; XPPi = 0.0;    XPHAi = 0.0;
XAUTi = 0.0;    XTSSi = 180;   XMeOHi = 0.0; XMePi = 0.0;

xi = [SO2i*ones(1,K1);  SFi*ones(1,K1);    SAi*ones(1,K1);    SNH4i*ones(1,K1);
     SNO3i*ones(1,K1); SPO4i*ones(1,K1); SIi*ones(1,K1);  SALKi*ones(1,K1);
     SN2i*ones(1,K1);  XIi*ones(1,K1);    XSi*ones(1,K1);    XHi*ones(1,K1);
     XPAOi*ones(1,K1); XPPi*ones(1,K1);   XPHAi*ones(1,K1); XAUTi*ones(1,K1);
     XTSSi*ones(1,K1); XMeOHi*ones(1,K1); XMePi*ones(1,K1)                ]*0.001;


% Values of the initial conditions
% For tank 1
SO2_1  = 4.2984433e-03; SF_1   = 2.8082131e+00; SA_1   = 2.8082131e+00;
SNH4_1 = 7.9178845e+00; SNO3_1 = 5.3699400e+00; SPO4_1 = 3.6;
SI_1   = 3.0000000e+01; SALK_1 = 4.9277103e+00; SN2_1  = 1.2166405e+00;
XI_1   = 1.1491252e+03; XS_1   = 8.2134908e+01; XH_1   = 2.5517658e+03;
XPAO_1 = 0.1; XPP_1  = 0.1; XPHA_1 = 0.1; XAUT_1  = 1.4838943e+02;
XTSS_1 = 180;XMeOH_1 = 0.1;XMeP_1  = 0.1;
x10 = [SO2_1; SF_1; SA_1; SNH4_1; SNO3_1; SPO4_1; SI_1;   SALK_1; SN2_1;
      XI_1;   XS_1;  XH_1; XPAO_1; XPP_1;  XPHA_1; XAUT_1; XTSS_1; XMeOH_1;
```

```
         XMeP_1];

% For tank 2
SO2_2   = 6.3131911e-05; SF_2    = 1.4587940e+00; SA_2    = 1.4587940e+00;
SNH4_2  = 8.3444148e+00; SNO3_2  = 3.6619672e+00; SPO4_2  = 3.6;
SI_2    = 3.0000000e+01; SALK_2  = 5.0801748e+00; SN2_2   = 8.8206477e-01;
XI_2    = 1.1491252e+03; XS_2    = 7.6386187e+01; XH_2    = 2.5533851e+03;
XPAO_2  = 0.1;XPP_2     = 0.1; XPHA_2  = 0.1; XAUT_2  = 1.4830914e+02;
XTSS_2  = 180;XMeOH_2  = 0.1; XMeP_2  = 0.1;
x20 = [SO2_2; SF_2; SA_2; SNH4_2; SNO3_2; SPO4_2; SI_2;    SALK_2; SN2_2;
       XI_2;  XS_2; XH_2; XPAO_2; XPP_2;  XPHA_2; XAUT_2; XTSS_2; XMeOH_2;
       XMeP_2];

% For tank 3
SO2_3   = 1.7183778e+00; SF_3    = 1.1495418e+00; SA_3    = 1.1495418e+00;
SNH4_3  = 5.5479452e+00; SNO3_3  = 6.5408820e+00; SPO4_3  = 3.6;
SI_3    = 3.0000000e+01; SALK_3  = 4.6747902e+00; SN2_3   = 8.2888682e-01;
XI_3    = 1.1491252e+03; XS_3    = 6.4854922e+01; XH_3    = 2.5571314e+03;
XPAO_3  = 0.1; XPP_3    = 0.1; XPHA_3  = 0.1; XAUT_3   = 1.4894126e+02;
XTSS_3  = 180; XMeOH_3  = 0.1; XMeP_3  = 0.1;
x30 = [SO2_3; SF_3; SA_3; SNH4_3; SNO3_3; SPO4_3; SI_3;    SALK_3; SN2_3;
       XI_3;  XS_3; XH_3; XPAO_3; XPP_3;  XPHA_3; XAUT_3; XTSS_3; XMeOH_3;
       XMeP_3];

% For tank 4
SO2_4   = 2.4288838e+00; SF_4    = 9.9532389e-01; SA_4    = 9.9532389e-01;
SNH4_4  = 2.9673854e+00; SNO3_4  = 9.2989988e+00; SPO4_4  = 3.6;
SI_4    = 3.0000000e+01; SALK_4  = 4.2934562e+00; SN2_4   = 7.6678656e-01;
XI_4    = 1.1491252e+03; XS_4    = 5.5693982e+01; XH_4    = 2.5591826e+03;
XPAO_4  = 0.1; XPP_4 = 0.1; XPHA_4  = 0.1; XAUT_4   = 1.4952712e+02;
XTSS_4  = 180; XMeOH_4 = 0.1; XMeP_4  = 0.1;
x40 = [SO2_4; SF_4; SA_4; SNH4_4; SNO3_4; SPO4_4; SI_4;    SALK_4; SN2_4;
       XI_4;  XS_4; XH_4; XPAO_4; XPP_4;  XPHA_4; XAUT_4; XTSS_4; XMeOH_4;
       XMeP_4];

% For tank 5
SO2_5   = 4.9094351e-01; SF_5    = 8.8949280e-01; SA_5    = 8.8949280e-01;
SNH4_5  = 1.7333316e+00; SNO3_5  = 1.0415220e+01; SPO4_5  = 3.6;
SI_5    = 3.0000000e+01; SALK_5  = 4.1255794e+00; SN2_5   = 6.8828001e-01;
XI_5    = 1.1491252e+03; XS_5    = 4.9305586e+01; XH_5    = 2.5593436e+03;
XPAO_5  = 0.1; XPP_5     = 0.1; XPHA_5  = 0.1; XAUT_5  = 1.4979714e+02;
XTSS_5  = 180; XMeOH_5  = 0.1; XMeP_5  = 0.1;
x50 = [SO2_5; SF_5; SA_5; SNH4_5; SNO3_5; SPO4_5; SI_5;    SALK_5; SN2_5;
       XI_5;  XS_5; XH_5; XPAO_5; XPP_5;  XPHA_5; XAUT_5; XTSS_5; XMeOH_5;
       XMeP_5];

x0 = [x10;x20;x30;x40;x50];
x = [x0 zeros(95,K1)]*0.01;

% Steady state simulation of the model
for k = 1:K1
    xk  = x(:,k);
    xik = xi(:,k);
    Pk  = P_BASM2F(xk);

    SOk  = [xk(39,1);xk(58,1);xk(77,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xxlk = [zeros(38,1);xxk(1);zeros(18,1);xxk(2);zeros(18,1);xxk(3);zeros(18,1)];

    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xxlk;
end

% Full model response to the steady state influent concentrations
% Tank 1
figure(1)
subplot(5,4,1);plot(x(1,:));title('SO2_1')
```
                                 203

```
subplot(5,4,2);plot(x(2,:));title('SF1')
subplot(5,4,3);plot(x(3,:));title('SA1')
subplot(5,4,4);plot(x(4,:));title('SNH4_1')
subplot(5,4,5);plot(x(5,:));title('SNO3_1')
subplot(5,4,6);plot(x(6,:));title('SPO4_1')
subplot(5,4,7);plot(x(7,:));title('SI1')
subplot(5,4,8);plot(x(8,:));title('SALK1')
subplot(5,4,9);plot(x(9,:));title('SN2_1')
subplot(5,4,10);plot(x(10,:));title('XI1');
subplot(5,4,11);plot(x(11,:));title('XS1')
subplot(5,4,12);plot(x(12,:));title('XH1')
subplot(5,4,13);plot(x(13,:));title('XPAO1')
subplot(5,4,14);plot(x(14,:));title('XPP1')
subplot(5,4,15);plot(x(15,:));title('XPHA1')
subplot(5,4,16);plot(x(16,:));title('XAUT1')
xlabel('discrete time k')
subplot(5,4,17);plot(x(17,:));title('XTSS1')
xlabel('discrete time k')
subplot(5,4,18);plot(x(18,:));title('XMeOH1')
xlabel('discrete time k')
subplot(5,4,19);plot(x(19,:));title('XMeP1')
xlabel('discrete time k')
set(figure(1),'name','Tank 1 steady state results','numbertitle','off')


% Tank 2
figure(2)
subplot(5,4,1);plot(x(20,:));title('SO2_2')
subplot(5,4,2);plot(x(21,:));title('SF2')
subplot(5,4,3);plot(x(22,:));title('SA2')
subplot(5,4,4);plot(x(23,:));title('SNH4_2')
subplot(5,4,5);plot(x(24,:));title('SNO3_2')
subplot(5,4,6);plot(x(25,:));title('SPO4_2')
subplot(5,4,7);plot(x(26,:));title('SI2')
subplot(5,4,8);plot(x(27,:));title('SALK2')
subplot(5,4,9);plot(x(28,:));title('SN2_2')
subplot(5,4,10);plot(x(29,:));title('XI2');
subplot(5,4,11);plot(x(30,:));title('XS2')
subplot(5,4,12);plot(x(31,:));title('XH2')
subplot(5,4,13);plot(x(32,:));title('XPAO2')
subplot(5,4,14);plot(x(33,:));title('XPP2')
subplot(5,4,15);plot(x(34,:));title('XPHA2')
subplot(5,4,16);plot(x(35,:));title('XAUT2')
xlabel('discrete time k')
subplot(5,4,17);plot(x(36,:));title('XTSS2')
xlabel('discrete time k')
subplot(5,4,18);plot(x(37,:));title('XMeOH2')
xlabel('discrete time k')
subplot(5,4,19);plot(x(38,:));title('XMeP2')
xlabel('discrete time k')
set(figure(2),'name','Tank 2 steady state results','numbertitle','off')


% Tank 3
figure(3)
subplot(5,4,1);plot(x(39,:));title('SO2_3')
subplot(5,4,2);plot(x(40,:));title('SF3')
subplot(5,4,3);plot(x(41,:));title('SA3')
subplot(5,4,4);plot(x(42,:));title('SNH4_3')
subplot(5,4,5);plot(x(43,:));title('SNO3_3')
subplot(5,4,6);plot(x(44,:));title('SPO4_3')
subplot(5,4,7);plot(x(45,:));title('SI3')
subplot(5,4,8);plot(x(46,:));title('SALK3')
subplot(5,4,9);plot(x(47,:));title('SN2_3')
subplot(5,4,10);plot(x(48,:));title('XI3');
subplot(5,4,11);plot(x(49,:));title('XS3')
subplot(5,4,12);plot(x(50,:));title('XH3')
subplot(5,4,13);plot(x(51,:));title('XPAO3')
```

```
subplot(5,4,14);plot(x(52,:));title('XPP3')
subplot(5,4,15);plot(x(53,:));title('XPHA3')
subplot(5,4,16);plot(x(54,:));title('XAUT3')
xlabel('discrete time k')
subplot(5,4,17);plot(x(55,:));title('XTSS3')
xlabel('discrete time k')
subplot(5,4,18);plot(x(56,:));title('XMeOH3')
xlabel('discrete time k')
subplot(5,4,19);plot(x(57,:));title('XMeP3')
xlabel('discrete time k')
set(figure(3),'name','Tank 3 steady state results','numbertitle','off')

% Tank 4
figure(4)
subplot(5,4,1);plot(x(58,:));title('SO2_4')
subplot(5,4,2);plot(x(59,:));title('SF4')
subplot(5,4,3);plot(x(60,:));title('SA4')
subplot(5,4,4);plot(x(61,:));title('SNH4_4')
subplot(5,4,5);plot(x(62,:));title('SNO3_4')
subplot(5,4,6);plot(x(63,:));title('SPO4_4')
subplot(5,4,7);plot(x(64,:));title('SI4')
subplot(5,4,8);plot(x(65,:));title('SALK4')
subplot(5,4,9);plot(x(66,:));title('SN2_4')
subplot(5,4,10);plot(x(67,:));title('XI4');
subplot(5,4,11);plot(x(68,:));title('XS4')
subplot(5,4,12);plot(x(69,:));title('XH4')
subplot(5,4,13);plot(x(70,:));title('XPAO4')
subplot(5,4,14);plot(x(71,:));title('XPP4')
subplot(5,4,15);plot(x(72,:));title('XPHA4')
subplot(5,4,16);plot(x(73,:));title('XAUT4')
xlabel('discrete time k')
subplot(5,4,17);plot(x(74,:));title('XTSS4')
xlabel('discrete time k')
subplot(5,4,18);plot(x(75,:));title('XMeOH4')
xlabel('discrete time k')
subplot(5,4,19);plot(x(76,:));title('XMeP4')
xlabel('discrete time k')
set(figure(4),'name','Tank 4 steady state results','numbertitle','off')

% Tank 5
figure(5)
subplot(5,4,1);plot(x(77,:));title('SO2_5')
subplot(5,4,2);plot(x(78,:));title('SF5')
subplot(5,4,3);plot(x(79,:));title('SA5')
subplot(5,4,4);plot(x(80,:));title('SNH4_5')
subplot(5,4,5);plot(x(81,:));title('SNO3_5')
subplot(5,4,6);plot(x(82,:));title('SPO4_5')
subplot(5,4,7);plot(x(83,:));title('SI5')
subplot(5,4,8);plot(x(84,:));title('SALK5')
subplot(5,4,9);plot(x(85,:));title('SN2_5')
subplot(5,4,10);plot(x(86,:));title('XI5');
subplot(5,4,11);plot(x(87,:));title('XS5')
subplot(5,4,12);plot(x(88,:));title('XH5')
subplot(5,4,13);plot(x(89,:));title('XPAO5')
subplot(5,4,14);plot(x(90,:));title('XPP5')
subplot(5,4,15);plot(x(91,:));title('XPHA5')
subplot(5,4,16);plot(x(92,:));title('XAUT5')
xlabel('discrete time k')
subplot(5,4,17);plot(x(93,:));title('XTSS5')
xlabel('discrete time k')
subplot(5,4,18);plot(x(94,:));title('XMeOH5')
xlabel('discrete time k')
subplot(5,4,19);plot(x(95,:));title('XMeP5')
xlabel('discrete time k')
set(figure(5),'name','Tank 5 steady state results','numbertitle','off')
```

```
% Data for dynamic simulations
K     = 10;      % Prediction horizon for dynamic simulations /
% Dynamics of the inflow concentrations for dry weather file
load dry.txt
% The order is  t, SS XBH XS XI SNH SI SND XND QO
%                 1  2   3  4  5   6  7   8   9  10


xi = [SO2i*ones(1,K);  SFi*ones(1,K);    SAi*ones(1,K);   dry(1:K,6)';
      SNO3i*ones(1,K); SPO4i*ones(1,K);  dry(1:K,7)';     SALKi*ones(1,K);
      SN2i*ones(1,K);  dry(1:K,5)';      dry(1:K,4)';     dry(1:K,3)';
      XPAOi*ones(1,K); XPPi*ones(1,K);   XPHAi*ones(1,K); XAUTi*ones(1,K);
      XTSSi*ones(1,K); XMeOHi*ones(1,K); XMePi*ones(1,K)];

% The order is                ..
%                 1  2  3  4  5  6  7  8  9  10  11  12  13   14 15 16 17
%                 18 19
x = [x(:,(K1+1)) zeros(95,K)];

% Dynamic simulation of the model
for k = 1:K
    xk  = x(:,k);
    xik = xi(:,k);
    Pk  = P_BASM2F(xk);

    SOk  = [xk(39,1);xk(58,1);xk(77,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xx1k = [zeros(38,1);xxk(1);zeros(18,1);xxk(2);zeros(18,1);xxk(3);zeros(18,1)];

    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xx1k;
end

% Full model response to the dynamic influent concentrations
% Tank 1
figure(6)
subplot(5,4,1);plot(x(1,:));title('SO2_1')
subplot(5,4,2);plot(x(2,:));title('SF1')
subplot(5,4,3);plot(x(3,:));title('SA1')
subplot(5,4,4);plot(x(4,:));title('SNH4_1')
subplot(5,4,5);plot(x(5,:));title('SNO3_1')
subplot(5,4,6);plot(x(6,:));title('SPO4_1')
subplot(5,4,7);plot(x(7,:));title('SI1')
subplot(5,4,8);plot(x(8,:));title('SALK1')
subplot(5,4,9);plot(x(9,:));title('SN2_1')
subplot(5,4,10);plot(x(10,:));title('XI1');
subplot(5,4,11);plot(x(11,:));title('XS1')
subplot(5,4,12);plot(x(12,:));title('XH1')
subplot(5,4,13);plot(x(13,:));title('XPAO1')
subplot(5,4,14);plot(x(14,:));title('XPP1')
subplot(5,4,15);plot(x(15,:));title('XPHA1')
subplot(5,4,16);plot(x(16,:));title('XAUT1')
xlabel('discrete time k')
subplot(5,4,17);plot(x(17,:));title('XTSS1')
xlabel('discrete time k')
subplot(5,4,18);plot(x(18,:));title('XMeOH1')
xlabel('discrete time k')
subplot(5,4,19);plot(x(19,:));title('XMeP1')
xlabel('discrete time k')
set(figure(6),'name','Tank 1 dynamic results','numbertitle','off')


% Tank 2
figure(7)
subplot(5,4,1);plot(x(20,:));title('SO2_2')
subplot(5,4,2);plot(x(21,:));title('SF2')
subplot(5,4,3);plot(x(22,:));title('SA2')
subplot(5,4,4);plot(x(23,:));title('SNH4_2')
```

```
subplot(5,4,5);plot(x(24,:));title('SNO3_2')
subplot(5,4,6);plot(x(25,:));title('SPO4_2')
subplot(5,4,7);plot(x(26,:));title('SI2')
subplot(5,4,8);plot(x(27,:));title('SALK2')
subplot(5,4,9);plot(x(28,:));title('SN2_2')
subplot(5,4,10);plot(x(29,:));title('XI2');
subplot(5,4,11);plot(x(30,:));title('XS2')
subplot(5,4,12);plot(x(31,:));title('XH2')
subplot(5,4,13);plot(x(32,:));title('XPAO2')
subplot(5,4,14);plot(x(33,:));title('XPP2')
subplot(5,4,15);plot(x(34,:));title('XPHA2')
subplot(5,4,16);plot(x(35,:));title('XAUT2')
xlabel('discrete time k')
subplot(5,4,17);plot(x(36,:));title('XTSS2')
xlabel('discrete time k')
subplot(5,4,18);plot(x(37,:));title('XMeOH2')
xlabel('discrete time k')
subplot(5,4,19);plot(x(38,:));title('XMeP2')
xlabel('discrete time k')
set(figure(7),'name','Tank 2 dynamic results','numbertitle','off')


% Tank 3
figure(8)
subplot(5,4,1);plot(x(39,:));title('SO2_3')
subplot(5,4,2);plot(x(40,:));title('SF3')
subplot(5,4,3);plot(x(41,:));title('SA3')
subplot(5,4,4);plot(x(42,:));title('SNH4_3')
subplot(5,4,5);plot(x(43,:));title('SNO3_3')
subplot(5,4,6);plot(x(44,:));title('SPO4_3')
subplot(5,4,7);plot(x(45,:));title('SI3')
subplot(5,4,8);plot(x(46,:));title('SALK3')
subplot(5,4,9);plot(x(47,:));title('SN2_3')
subplot(5,4,10);plot(x(48,:));title('XI3');
subplot(5,4,11);plot(x(49,:));title('XS3')
subplot(5,4,12);plot(x(50,:));title('XH3')
subplot(5,4,13);plot(x(51,:));title('XPAO3')
subplot(5,4,14);plot(x(52,:));title('XPP3')
subplot(5,4,15);plot(x(53,:));title('XPHA3')
subplot(5,4,16);plot(x(54,:));title('XAUT3')
xlabel('discrete time k')
subplot(5,4,17);plot(x(55,:));title('XTSS3')
xlabel('discrete time k')
subplot(5,4,18);plot(x(56,:));title('XMeOH3')
xlabel('discrete time k')
subplot(5,4,19);plot(x(57,:));title('XMeP3')
xlabel('discrete time k')
set(figure(8),'name','Tank 3 dynamic results','numbertitle','off')


% Tank 4
figure(9)
subplot(5,4,1);plot(x(58,:));title('SO2_4')
subplot(5,4,2);plot(x(59,:));title('SF4')
subplot(5,4,3);plot(x(60,:));title('SA4')
subplot(5,4,4);plot(x(61,:));title('SNH4_4')
subplot(5,4,5);plot(x(62,:));title('SNO3_4')
subplot(5,4,6);plot(x(63,:));title('SPO4_4')
subplot(5,4,7);plot(x(64,:));title('SI4')
subplot(5,4,8);plot(x(65,:));title('SALK4')
subplot(5,4,9);plot(x(66,:));title('SN2_4')
subplot(5,4,10);plot(x(67,:));title('XI4');
subplot(5,4,11);plot(x(68,:));title('XS4')
subplot(5,4,12);plot(x(69,:));title('XH4')
subplot(5,4,13);plot(x(70,:));title('XPAO4')
subplot(5,4,14);plot(x(71,:));title('XPP4')
subplot(5,4,15);plot(x(72,:));title('XPHA4')
subplot(5,4,16);plot(x(73,:));title('XAUT4')
```

```
xlabel('discrete time k')
subplot(5,4,17);plot(x(74,:));title('XTSS4')
xlabel('discrete time k')
subplot(5,4,18);plot(x(75,:));title('XMeOH4')
xlabel('discrete time k')
subplot(5,4,19);plot(x(76,:));title('XMeP4')
xlabel('discrete time k')
set(figure(9),'name','Tank 4 dynamic results','numbertitle','off')


% Tank 5
figure(10)
subplot(5,4,1);plot(x(77,:));title('SC2_5')
subplot(5,4,2);plot(x(78,:));title('SF5')
subplot(5,4,3);plot(x(79,:));title('SA5')
subplot(5,4,4);plot(x(80,:));title('SNH4_5')
subplot(5,4,5);plot(x(81,:));title('SNO3_5')
subplot(5,4,6);plot(x(82,:));title('SPO4_5')
subplot(5,4,7);plot(x(83,:));title('SI5')
subplot(5,4,8);plot(x(84,:));title('SALK5')
subplot(5,4,9);plot(x(85,:));title('SN2_5')
subplot(5,4,10);plot(x(86,:));title('XI5');
subplot(5,4,11);plot(x(87,:));title('XS5')
subplot(5,4,12);plot(x(88,:));title('XH5')
subplot(5,4,13);plot(x(89,:));title('XPAO5')
subplot(5,4,14);plot(x(90,:));title('XPP5')
subplot(5,4,15);plot(x(91,:));title('XPHA5')
subplot(5,4,16);plot(x(92,:));title('XAUT5')
xlabel('discrete time k')
subplot(5,4,17);plot(x(93,:));title('XTSS5')
xlabel('discrete time k')
subplot(5,4,18);plot(x(94,:));title('XMeCH5')
xlabel('discrete time k')
subplot(5,4,19);plot(x(95,:));title('XMeP5')
xlabel('discrete time k')
set(figure(10),'name','Tank 5 dynamic results','numbertitle','off')


% ======================= REDUCED MODEL FORMULATION =======================
% Form the vector of the states (z) for the reduced model, based on the
% ASM1 prediction
%       SF,      SA,      SNH4,    SNO3,      SPO4,    XS,      XPP,      XPHA
z1 = [x(2,:);  x(3,:);  x(4,:); x(5,:); x(6,:); x(11,:); x(14,:); x(15,:)];
z2 = [x(21,:); x(22,:); x(23,:);x(24,:);x(25,:);x(30,:); x(33,:); x(34,:)];
z3 = [x(40,:); x(41,:); x(42,:);x(43,:);x(44,:);x(49,:); x(52,:); x(53,:)];
z4 = [x(59,:); x(60,:); x(61,:);x(62,:);x(63,:);x(68,:); x(71,:); x(72,:)];
z5 = [x(78,:); x(79,:); x(80,:);x(81,:);x(82,:);x(87,:); x(90,:); x(91,:)];
z  = [z1;z2;z3;z4;z5];


% Formation of the vector of the control variables
u1 = x(1,:);
u2 = x(20,:);
u3 = x(39,:);
u4 = x(58,:);
u5 = x(77,:);
u  = [u1; u2; u3; u4; u5];


% Dynamic values of the inflow concentrations for dry weather
zi = [SFi*ones(1,K);   SAi*ones(1,K); SNH4i*ones(1,K); SNO3i*ones(1,K);
      SPO4i*ones(1,K); XSi*ones(1,K); XPPi*ones(1,K);  XPHAi*ones(1,K)];


% Steady state values for the biomass XH, XAUT, SALK, XPAO
XH   = [2551.76*ones(1,K); 2553.38*ones(1,K); 2557.13*ones(1,K);
        2559.18*ones(1,K); 2559.34*ones(1,K)];
XAUT = [148.389*ones(1,K); 148.309*ones(1,K); 148.941*ones(1,K);
        149.527*ones(1,K); 149.797*ones(1,K)];
SALK = [4.9*ones(1,K);     148.309*ones(1,K); 148.941*ones(1,K);
        149.527*ones(1,K); 149.797*ones(1,K)];
```

208

```
XPAO = [76.386*ones(1,K);    76.386*ones(1,K); 64.855*ones(1,K);
        55.694*ones(1,K); 49.306*ones(1,K)];

% MATRIX A for the reduced model (AR)
A11r = 1 - IV11; A22r = 1 - IV22; A33r = 1 - IV33;
A44r = 1 - IV44; A55r = 1 - IV55;
A15ar = IV1 * (Qa + Qr);
A15br = IV1 * (Qa + LAMBDA*Qr);
A21r = IV22; A32r = IV33; A43r = IV44; A54r = IV55;

A11r = A11r * ones(1,8); A11r = diag(A11r);
A22r = A22r * ones(1,8); A22r = diag(A22r);
A33r = A33r * ones(1,8); A33r = diag(A33r);
A44r = A44r * ones(1,8); A44r = diag(A44r);
A55r = A55r * ones(1,8); A55r = diag(A55r);

A15r = [A15ar, A15ar, A15ar, A15ar, A15ar, A15br, A15br, A15br];
A15r = diag(A15r);

A21r = A21r * ones(1,8); A21r = diag(A21r);
A32r = A32r * ones(1,8); A32r = diag(A32r);
A43r = A43r * ones(1,8); A43r = diag(A43r);
A54r = A54r * ones(1,8); A54r = diag(A54r);

A10r = zeros(8,8); A20r = zeros(8,16); A30r = zeros(8,24);

AR = [A11r A30r A15r;A21r A22r A30r;A10r A32r A33r A20r;A20r A43r A44r A10r;A30r A54r
A55r];

% MATRIX B for the reduced model (BR)
IV0 = DT / V1*Q0;
f = 1.2;
B11r = [IV0*ones(1,3) IV0*f IV0*ones(1,4)];
B11r = diag(B11r);
BR = [B11r; zeros(32,8)];

% MATRIX C for the reduced model (CR)
%      SF     SA      SNH4    SNO3    SPO4      XS        XPP     XPHA
CCR = [[1   0 0.01    0 0    -1     0 0];
       [1   0 0.01    0 0    -1     0 0];
       [1   0 0.01    0 0    -1     0 0];
       [-1.59  0 -0.022   0 -0.004   0 0     0];
       [0   -1.59    -0.0070  0 -0.02    0 0     0];
       [-1.59  0 -0.022   -0.21 -0.004   0   0   0];
       [0   -1.59    -0.07    -0.21 -0.02    0   0 0];
       [-1 1 0.03    0 0.01    0        0     0];
       [0   -1    0   0 0.4     0       -0.4 1];
       [0   0 0    0 -1    0        1       -0.20];
       [0   0 -0.07    0 -0.02  0       0      -1.6];
       [0   0 -4.24    4.17   -0.02  0   0 0]];

CR = DT*[CCR zeros(12,32); zeros(12,8) CCR zeros(12,24);
         zeros(12,16) CCR zeros(12,16); zeros(12,24) CCR zeros(12,8);
         zeros(12,32) CCR];

% Parameter estimation
for k = 1:K
    zk = z(:,k);
    uk = u(:,k);

    XHk   = XH(:,k);
    XAUTk = XAUT(:,k);
    XPAOk = XPAO(:,k);
    SALKk = SALK(:,k);
```

209

```
    zik    = zi(:,k);
    Pk     = P_BASM2R(zk,uk,XHk,XAUTk,XPAOk,SALKk);
    FIk    = CR'*Pk;
    Y(:,k)      = z(:,k+1)-AR*z(:,k)-BR*zi(:,k);
    theta(:,k) = (FIk'*FIk)^(-1)*FIk'*Y(:,k);
end


kHe(1,:)      = theta(1,:);
muHe(1,:)     = theta(2,:);
qfee(1,:)     = theta(3,:);
qPHAe(1,:)    = theta(4,:);
qPPe(1,:)     = theta(5,:);
muPAOe(1,:)   = theta(6,:);
muAUTe(1,:)   = theta(7,:);           ...

% Dynamic simulation of the reduced model with the estimated parameters
zE = [z(1,1);z(2,1);z(3,1);z(4,1);z(5,1);z(6,1);z(7,1);z(8,1);z(9,1);z(10,1); z(11,1);
z(12,1);z(13,1); z(14,1);
z(15,1);z(16,1);z(17,1);z(18,1);z(19,1);z(20,1);z(21,1);z(22,1);z(23,1);z(24,1);z(25,1)
; z(26,1); z(27,1);z(28,1); z(29,1);
z(30,1);z(31,1);z(32,1);z(33,1);z(34,1);z(35,1);z(36,1);z(37,1);z(38,1);z(39,1);z(40,1)
];
ERR = z(:,1)-zE;


% Dynamic simulation of the reduced model with the estimated parameters
for k = 1:K
    zEk    = zE(:,k);
    uk     = u(:,k);
    zik    = zi(:,k);

    XHk    = XH(:,k);
    XAUTk  = XAUT(:,k);
    XPAOk  = XPAO(:,k);
    SALKk  = SALK(:,k);

    kHek       = theta(1,k);
    muHek      = theta(2,k);
    qfeek      = theta(3,k);
    qPHAek     = theta(4,k);
    qPPek      = theta(5,k);
    muPAOek    = theta(6,k);
    muAUTek    = theta(7,k);

    Pk =
P_BASM2RE(zEk,uk,XHk,XAUTk,XPAOk,SALKk,kHek,muHek,qfeek,qPHAek,qPPek,muPAOek,muAUTek);
    zE(:,k+1) = AR*zE(:,k) + CR'*Pk + BR*zi(:,k);
    ERR(:,k+1)=z(:,k+1)-zE(:,k+1);
end

% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 1
SF1=z(1,:);SA1=z(2,:);SNH4_1=z(3,:);SNO3_1=z(4,:);
SF1E=zE(1,:);SA1E=zE(2,:);SNH4_1E=zE(3,:);SNO3_1E=zE(4,:);
ERRSF1=ERR(1,:);ERRSA1=ERR(2,:);ERRSNH4_1=ERR(3,:);ERRSNO3_1=ERR(4,:);

SPO4_1=z(5,:);XS1=z(6,:);XPP1=z(7,:);XPHA1=z(8,:);
SPO4_1E=zE(5,:);XS1E=zE(6,:);XPP1E=zE(7,:);XPHA1E=zE(8,:);
ERRSPO4_1=ERR(5,:);ERRXS1=ERR(6,:);ERRXPP1=ERR(7,:);ERRXPHA1=ERR(8,:);

% Calculation of the weighted errors for tank 1
eSF1=(ERRSF1*ERRSF1')/(SF1*SF1');
eSA1=(ERRSA1*ERRSA1')/(SA1*SA1');
eSNH4_1=(ERRSNH4_1*ERRSNH4_1')/(SNH4_1*SNH4_1');
eSNO3_1=(ERRSNO3_1*ERRSNO3_1')/(SNO3_1*SNO3_1');
eSPO4_1=(ERRSPO4_1*ERRSPO4_1')/(SPO4_1*SPO4_1');
eXS1=(ERRXS1*ERRXS1')/(XS1*XS1');
```

```
eXPP1=(ERRXPP1*ERRXPP1')/(XPP1*XPP1');
eXPHA1=(ERRXPHA1*ERRXPHA1')/(XPHA1*XPHA1');


% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 2
SF2=z(9,:);SA2=z(10,:);SNH4_2=z(11,:);SNO3_2=z(12,:);
SF2E=zE(9,:);SA2E=zE(10,:);SNH4_2E=zE(11,:);SNO3_2E=zE(12,:);
ERRSF2=ERR(9,:);ERRSA2=ERR(10,:);ERRSNH4_2=ERR(11,:);ERRSNO3_2=ERR(12,:);


SPO4_2=z(13,:);XS2=z(14,:);XPP2=z(15,:);XPHA2=z(16,:);
SPO4_2E=zE(13,:);XS2E=zE(14,:);XPP2E=zE(15,:);XPHA2E=zE(16,:);
ERRSPO4_2=ERR(13,:);ERRXS2=ERR(14,:);ERRXPP2=ERR(15,:);ERRXPHA2=ERR(16,:);


% Calculation of the weighted errors for tank 2
eSF2=(ERRSF2*ERRSF2')/(SF2*SF2');
eSA2=(ERRSA2*ERRSA2')/(SA2*SA2');
eSNH4_2=(ERRSNH4_2*ERRSNH4_2')/(SNH4_2*SNH4_2');
eSNO3_2=(ERRSNO3_2*ERRSNO3_2')/(SNO3_2*SNO3_2');
eSPO4_2=(ERRSPO4_2*ERRSPO4_2')/(SPO4_2*SPO4_2');
eXS2=(ERRXS2*ERRXS2')/(XS2*XS2');
eXPP2=(ERRXPP2*ERRXPP2')/(XPP2*XPP2');
eXPHA2=(ERRXPHA2*ERRXPHA2')/(XPHA2*XPHA2');


% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 3
SF3=z(17,:);SA3=z(18,:);SNH4_3=z(19,:);SNO3_3=z(20,:);
SF3E=zE(17,:);SA3E=zE(18,:);SNH4_3E=zE(19,:);SNO3_3E=zE(20,:);
ERRSF3=ERR(17,:);ERRSA3=ERR(18,:);ERRSNH4_3=ERR(19,:);ERRSNO3_3=ERR(20,:);


SPO4_3=z(21,:);XS3=z(22,:);XPP3=z(23,:);XPHA3=z(24,:);
SPO4_3E=zE(21,:);XS3E=zE(22,:);XPP3E=zE(23,:);XPHA3E=zE(24,:);
ERRSPO4_3=ERR(21,:);ERRXS3=ERR(22,:);ERRXPP3=ERR(23,:);ERRXPHA3=ERR(24,:);


% Calculation of the weighted errors for tank 3
eSF3=(ERRSF3*ERRSF3')/(SF3*SF3');
eSA3=(ERRSA3*ERRSA3')/(SA3*SA3');
eSNH4_3=(ERRSNH4_3*ERRSNH4_3')/(SNH4_3*SNH4_3');
eSNO3_3=(ERRSNO3_3*ERRSNO3_3')/(SNO3_3*SNO3_3');
eSPO4_3=(ERRSPO4_3*ERRSPO4_3')/(SPO4_3*SPO4_3');
eXS3=(ERRXS3*ERRXS3')/(XS3*XS3');
eXPP3=(ERRXPP3*ERRXPP3')/(XPP3*XPP3');
eXPHA3=(ERRXPHA3*ERRXPHA3')/(XPHA3*XPHA3');


% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 4
SF4=z(9,:);SA4=z(10,:);SNH4_4=z(11,:);SNO3_4=z(12,:);
SF4E=zE(9,:);SA4E=zE(10,:);SNH4_4E=zE(11,:);SNO3_4E=zE(12,:);
ERRSF4=ERR(9,:);ERRSA4=ERR(10,:);ERRSNH4_4=ERR(11,:);ERRSNO3_4=ERR(12,:);


SPO4_4=z(13,:);XS4=z(14,:);XPP4=z(15,:);XPHA4=z(16,:);
SPO4_4E=zE(13,:);XS4E=zE(14,:);XPP4E=zE(15,:);XPHA4E=zE(16,:);
ERRSPO4_4=ERR(13,:);ERRXS4=ERR(14,:);ERRXPP4=ERR(15,:);ERRXPHA4=ERR(16,:);


% Calculation of the weighted errors for tank 4
eSF4=(ERRSF4*ERRSF4')/(SF4*SF4');
eSA4=(ERRSA4*ERRSA4')/(SA4*SA4');
eSNH4_4=(ERRSNH4_4*ERRSNH4_4')/(SNH4_4*SNH4_4');
eSNO3_4=(ERRSNO3_4*ERRSNO3_4')/(SNO3_4*SNO3_4');
eSPO4_4=(ERRSPO4_4*ERRSPO4_4')/(SPO4_4*SPO4_4');
eXS4=(ERRXS4*ERRXS4')/(XS4*XS4');
eXPP4=(ERRXPP4*ERRXPP4')/(XPP4*XPP4');
eXPHA4=(ERRXPHA4*ERRXPHA4')/(XPHA4*XPHA4');


% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 3
SF5=z(17,:);SA5=z(18,:);SNH4_5=z(19,:);SNO3_5=z(20,:);
```

```
SF5E=zE(17,:);SA5E=zE(18,:);SNH4_5E=zE(19,:);SNO3_5E=zE(20,:);
ERRSF5=ERR(17,:);ERRSA5=ERR(18,:);ERRSNH4_5=ERR(19,:);ERRSNO3_5=ERR(20,:);

SPO4_5=z(21,:);XS5=z(22,:);XPP5=z(23,:);XPHA5=z(24,:);
SPO4_5E=zE(21,:);XS5E=zE(22,:);XPP5E=zE(23,:);XPHA5E=zE(24,:);
ERRSPO4_5=ERR(21,:);ERRXS5=ERR(22,:);ERRXPP5=ERR(23,:);ERRXPHA5=ERR(24,:);

% Calculation of the weighted errors for tank 5
eSF5=(ERRSF5*ERRSF5')/(SF5*SF5');
eSA5=(ERRSA5*ERRSA5')/(SA5*SA5');
eSNH4_5=(ERRSNH4_5*ERRSNH4_5')/(SNH4_5*SNH4_5');
eSNO3_5=(ERRSNO3_5*ERRSNO3_5')/(SNO3_5*SNO3_5');
eSPO4_5=(ERRSPO4_5*ERRSPO4_5')/(SPO4_5*SPO4_5');
eXS5=(ERRXS5*ERRXS5')/(XS5*XS5');
eXPP5=(ERRXPP5*ERRXPP5')/(XPP5*XPP5');
eXPHA5=(ERRXPHA5*ERRXPHA5')/(XPHA5*XPHA5');

k = 1:K;
% Graphs of the estimated parameters
figure(11)
subplot(4,2,1);plot(k,kHe,'k');title('Estimated kH');ylabel('kHe');
subplot(4,2,2);plot(k,muHe,'k');title('Estimated muH');ylabel('muHe');
subplot(4,2,3);plot(k,qfee,'k');title('Estimated qfe');ylabel('qfee');
subplot(4,2,4);plot(k,qPHAe,'k');title('Estimated qPHA');ylabel('qPHAe');
subplot(4,2,5);plot(k,qPPe,'k');title('Estimated qPP');ylabel('qPPe');
subplot(4,2,6);plot(k,muPAOe,'k');title('Estimated muPAO');
ylabel('muPAOe');xlabel('discrete time k')
subplot(4,2,7);plot(k,muAUTe,'k');title('Estimated muAUT');
ylabel('muAUTe');xlabel('discrete time k')
set(figure(11),'name','Estimated parameters','numbertitle','off')

k = 1:K+1;
% Graphs of the reduced model response to the dynamic influent
% concentrations

figure(12)
subplot(4,4,1);plot(k,SF1);hold on; plot(k,SF1E,'k-.');ylabel('SF1,SF1E');
legend('SF1','SF1E');subplot(4,4,2); plot(k,ERRSF1,'k');ylabel('ERRSF1');

subplot(4,4,3);plot(k,SA1);hold on; plot(k,SA1E,'k-.');ylabel('SA1,SA1E');
legend('SA1','SA1E');subplot(4,4,4); plot(k,ERRSA1,'k');ylabel('ERRSA1');

subplot(4,4,5);plot(k,SNH4_1);hold on; plot(k,SNH4_1E,'k-.');
legend('SNH4_1','SNH4_1E');ylabel('SNH4_1,SNH4_1E');subplot(4,4,6);
plot(k,ERRSNH4_1,'k');ylabel('ERRSNH4_1');

subplot(4,4,7);plot(k,SNO3_1);hold on; plot(k,SNO3_1E,'k-.')
legend('SNO3_1','SNO3_1E');ylabel('SNO3_1,SNO3_1E');subplot(4,4,8);
plot(k,ERRSNO3_1,'k');ylabel('ERRSNO3_1');

subplot(4,4,9);plot(k,SPO4_1);hold on; plot(k,SPO4_1E,'k-.');
legend('SPO4_1','SPO4_1E');ylabel('SPO4_1,SPO4_1E');subplot(4,4,10);
plot(k,ERRSPO4_1,'k');ylabel('ERRSPO4_1');

subplot(4,4,11);plot(k,XS1);hold on; plot(k,XS1E,'k-.');ylabel('XS1,XS1E');
legend('XS1','XS1E');subplot(4,4,12); plot(k,ERRXS1,'k');ylabel('ERRXS1');

subplot(4,4,13);plot(k,XPP1);hold on; plot(k,XPP1E,'k-.')
legend('XPP1','XPP1E');ylabel('XPP1,XPP1E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP1,'k');
ylabel('ERRXPP1');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA1);hold on; plot(k,XPHA1E,'k-.')
legend('XPHA1','XPHA1E');ylabel('XPHA1,XPHA1E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA1,'k');
ylabel('ERRXPHA1');xlabel('discrete time k')
```

212

```
set(figure(12),'name','Tank 1 Process variables and estimation
error','numbertitle','off')

figure(13)
subplot(4,4,1);plot(k,SF2);hold on; plot(k,SF2E,'k-.');ylabel('SF2,SF2E');
legend('SF2','SF2E');subplot(4,4,2); plot(k,ERRSF2,'k');ylabel('ERRSF2');

subplot(4,4,3);plot(k,SA2);hold on; plot(k,SA2E,'k-.');ylabel('SA2,SA2E');
legend('SA2','SA2E');subplot(4,4,4); plot(k,ERRSA2,'k');ylabel('ERRSA2');

subplot(4,4,5);plot(k,SNH4_2);hold on; plot(k,SNH4_2E,'k-.');
legend('SNH4_2','SNH4_2E');ylabel('SNH4_2,SNH4_2E');subplot(4,4,6);
plot(k,ERRSNH4_2,'k');ylabel('ERRSNH4_2');

subplot(4,4,7);plot(k,SNO3_2);hold on; plot(k,SNO3_2E,'k-.')
legend('SNO3_2','SNO3_2E');ylabel('SNO3_2,SNO3_2E');subplot(4,4,8);
plot(k,ERRSNO3_2,'k');ylabel('ERRSNO3_2');

subplot(4,4,9);plot(k,SPO4_2);hold on; plot(k,SPO4_2E,'k-.')
legend('SPO4_2','SPO4_2E');ylabel('SPO4_2,SPO4_2E');subplot(4,4,10);
plot(k,ERRSPO4_2,'k');ylabel('ERRSPO4_2');

subplot(4,4,11);plot(k,XS2);hold on; plot(k,XS2E,'k-.');ylabel('XS2,XS2E');
legend('XS2';'XS2E');subplot(4,4,12); plot(k,ERRXS2,'k');ylabel('ERRXS2');

subplot(4,4,13);plot(k,XPP2);hold on; plot(k,XPP2E,'k-.');
legend('XPP2','XPP2E');ylabel('XPP2,XPP2E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP2,'k');
ylabel('ERRXPP2');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA2);hold on; plot(k,XPHA2E,'k-.')
legend('XPHA2','XPHA2E');ylabel('XPHA2,XPHA2E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA2,'k');
ylabel('ERRXPHA2');xlabel('discrete time k')
set(figure(13),'name','Tank 2 Process variables and estimation
error','numbertitle','off')

figure(14)
subplot(4,4,1);plot(k,SF3);hold on; plot(k,SF3E,'k-.');ylabel('SF3,SF3E');
legend('SF3','SF3E');subplot(4,4,2); plot(k,ERRSF3,'k');ylabel('ERRSF3');

subplot(4,4,3);plot(k,SA3);hold on; plot(k,SA3E,'k-.');ylabel('SA3,SA3E');
legend('SA3','SA3E');subplot(4,4,4); plot(k,ERRSA3,'k');

subplot(4,4,5);plot(k,SNH4_3);hold on; plot(k,SNH4_3E,'k-.')
legend('SNH4_3','SNH4_3E');ylabel('SNH4_3,SNH4_3E');subplot(4,4,6);
plot(k,ERRSNH4_3,'k');ylabel('ERRSNH4_3');

subplot(4,4,7);plot(k,SNO3_3);hold on; plot(k,SNO3_3E,'k-.')
legend('SNO3_3','SNO3_3E');ylabel('SNO3_3,SNO3_3E');subplot(4,4,8);
plot(k,ERRSNO3_3,'k');ylabel('ERRSNO3_3');

subplot(4,4,9);plot(k,SPO4_3);hold on; plot(k,SPO4_3E,'k-.')
legend('SPO4_3','SPO4_3E');ylabel('SPO4_3,SPO4_3E');subplot(4,4,10);
plot(k,ERRSPO4_3,'k');ylabel('ERRSPO4_3');

subplot(4,4,11);plot(k,XS3);hold on; plot(k,XS3E,'k-.');ylabel('XS3,XS3E');
legend('XS3','XS3E');subplot(4,4,12); plot(k,ERRXS3,'k');ylabel('ERRXS3');

subplot(4,4,13);plot(k,XPP3);hold on; plot(k,XPP3E,'k-.')
legend('XPP3','XPP3E');ylabel('XPP3,XPP3E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP3,'k');
ylabel('ERRXPP3');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA3);hold on; plot(k,XPHA3E,'k-.')
legend('XPHA3','XPHA3E');ylabel('XPHA3,XPHA3E');xlabel('discrete time k')
```

213

```
subplot(4,4,16); plot(k,ERRXPHA3,'k');
ylabel('ERRXPHA3');xlabel('discrete time k')
set(figure(14),'name','Tank 3 Process variables and estimation
error','numbertitle','off')

figure(15)
subplot(4,4,1);plot(k,SF4);hold on; plot(k,SF4E,'k-.');ylabel('SF4,SF4E');
legend('SF4','SF4E');subplot(4,4,2); plot(k,ERRSF4,'k');ylabel('ERRSF4');

subplot(4,4,3);plot(k,SA4);hold on; plot(k,SA4E,'k-.');ylabel('SA4,SA4E');
legend('SA4','SA4E');subplot(4,4,4); plot(k,ERRSA4,'k');ylabel('ERRSA4');

subplot(4,4,5);plot(k,SNH4_4);hold on; plot(k,SNH4_4E,'k-.')
legend('SNH4_4','SNH4_4E');ylabel('SNH4_4,SNH4_4E');subplot(4,4,6);
plot(k,ERRSNH4_4,'k');ylabel('ERRSNH4_4');

subplot(4,4,7);plot(k,SNO3_4);hold on; plot(k,SNO3_4E,'k-.')
legend('SNO3_4','SNO3_4E');ylabel('SNO3_4,SNO3_4E');subplot(4,4,8);
plot(k,ERRSNO3_4,'k');ylabel('ERRSNO3_4');

subplot(4,4,9);plot(k,SPO4_4);hold on; plot(k,SPO4_4E,'k-.')
legend('SPO4_4','SPO4_4E');ylabel('SPO4_4,SPO4_4E');subplot(4,4,10);
plot(k,ERRSPO4_4,'k');ylabel('ERRSPO4_4');

subplot(4,4,11);plot(k,XS4);hold on; plot(k,XS4E,'k-.');ylabel('XS4,XS4E');
legend('XS4','XS4E');subplot(4,4,12); plot(k,ERRXS4,'k');ylabel('ERRXS4');

subplot(4,4,13);plot(k,XPP4);hold on; plot(k,XPP4E,'k-.')
legend('XPP4','XPP4E');ylabel('XPP4,XPP4E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP4,'k');
ylabel('ERRXPP4');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA4);hold on; plot(k,XPHA4E,'k-.')
legend('XPHA4','XPHA4E');ylabel('XPHA4,XPHA4E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA4,'k');
ylabel('ERRXPHA4');xlabel('discrete time k')
set(figure(15),'name','Tank 4 Process variables and estimation
error','numbertitle','off')

figure(16)
subplot(4,4,1);plot(k,SF5);hold on; plot(k,SF5E,'k-.');ylabel('SF5,SF5E');
legend('SF5','SF5E');subplot(4,4,2); plot(k,ERRSF5,'k');ylabel('ERRSF5');

subplot(4,4,3);plot(k,SA5);hold on; plot(k,SA5E,'k-.');ylabel('SA5,SA5E');
legend('SA5','SA5E');subplot(4,4,4); plot(k,ERRSA5,'k');ylabel('ERRSA5');

subplot(4,4,5);plot(k,SNH4_5);hold on; plot(k,SNH4_5E,'k-.')
legend('SNH4_5','SNH4_5E');ylabel('SNH4_5,SNH4_5E');subplot(4,4,6);
plot(k,ERRSNH4_5,'k');ylabel('ERRSNH4_5');

subplot(4,4,7);plot(k,SNO3_5);hold on; plot(k,SNO3_5E,'k-.')
legend('SNO3_5','SNO3_5E');ylabel('SNO3_5,SNO3_5E');subplot(4,4,8);
plot(k,ERRSNO3_5,'k');ylabel('ERRSNO3_5');

subplot(4,4,9);plot(k,SPO4_5);hold on; plot(k,SPO4_5E,'k-.')
legend('SPO4_5','SPO4_5E');ylabel('SPO4_5,SPO4_5E');subplot(4,4,10);
plot(k,ERRSPO4_5,'k');ylabel('ERRSPO4_5');

subplot(4,4,11);plot(k,XS5);hold on; plot(k,XS5E,'k-.');ylabel('XS5,XS5E');
legend('XS5','XS5E');subplot(4,4,12); plot(k,ERRXS5,'k');ylabel('ERRXS5');

subplot(4,4,13);plot(k,XPP5);hold on; plot(k,XPP5E,'k-.')
legend('XPP5','XPP5E');ylabel('XPP5,XPP5E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP5,'k');
ylabel('ERRXPP5');xlabel('discrete time k')
```

214

```
subplot(4,4,15);plot(k,XPHA5);hold on; plot(k,XPHA5E,'k-.')
legend('XPHA5','XPHA5E');ylabel('XPHA5,XPHA5E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA5,'k');
ylabel('ERRXPHA5');xlabel('discrete time k')
set(figure(16),'name','Tank 5 Process variables and estimation
error','numbertitle','off')
```

B.2: MATLAB script file – P_BASM2F.m

```
% M-File: P_BASM2F.m
% ===================== M-FILE DESCRIPTION =======================
% This m-file is used for:
% Calculation of the process rates for the full model

function[P] = P_BASM2F(xk)

global fSI YH fXI YPO4 YPHA iPBM YA iNBM Kh etaNO3 etafe muH qfe bH qPHA qPP muPAO bPAO
bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4 KALK KA Kfe KPP KPS KP KPHA KIPP KMAX
% TANK 1
P(1)= Kh * (xk(1)/(KO2+xk(1))) * ((xk(11)/xk(12))/(KX+(xk(11)/xk(12)))) * xk(12);
P(2)= Kh * etaNO3 * (KO2/(KO2+xk(1))) * (xk(5)/(KNO3+xk(5))) *
((xk(11)/xk(12))/(KX+(xk(11)/xk(12)))) * xk(12);
P(3)= Kh * etafe * (KO2/(KO2+xk(1))) * (KNO3/(KNO3+xk(5))) *
((xk(11)/xk(12))/(KX+(xk(11)/xk(12)))) * xk(12);
P(4)= muH * (xk(1)/(KO2+xk(1))) * (xk(2)/(KF+xk(2))) * (xk(2)/(xk(2)+xk(3))) *
(xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) * (xk(8)/(KALK+xk(8))) * xk(12);
P(5)= muH * (xk(1)/(KO2+xk(1))) * (xk(3)/(KA+xk(3))) * (xk(3)/(xk(2)+xk(3))) *
(xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) * (xk(8)/(KALK+xk(8))) * xk(12);
P(6)= muH * etaNO3 * (KO2/(KO2+xk(1))) * (xk(2)/(KF+xk(2))) * (xk(2)/(xk(2)+xk(3))) *
(xk(4)/(KNH4+xk(4))) * (xk(5)/(KNO3+xk(5))) * (xk(8)/(KALK+xk(8))) * xk(12);
P(7)= muH * etaNO3 * (KO2/(KO2+xk(1))) * (xk(3)/(KA+xk(3))) * (xk(3)/(xk(2)+xk(3))) *
(xk(4)/(KNH4+xk(4))) * (xk(5)/(KNO3+xk(5))) * (xk(8)/(KALK+xk(8)))* (xk(6)/(KP+xk(6)))
* xk(12);
P(8)= qfe * (KO2/(KO2+xk(1))) * (KNO3/(KNO3+xk(5))) * (xk(2)/(Kfe+xk(2))) *
(xk(8)/(KALK+xk(8)))* xk(12);
P(9)= bH * xk(12);
P(10)= qPHA * (xk(3)/(KA+xk(3))) * (xk(8)/(KALK+xk(8))) *
((xk(14)/xk(13))/(KPP+(xk(14)/xk(13)))) * xk(13);
P(11)= qPP * (xk(1)/(KO2+xk(1))) * (xk(6)/(KPS+xk(6))) * (xk(8)/(KALK+xk(8))) *
((xk(15)/xk(13))/(KPHA+(xk(15)/xk(13)))) * ((KMAX-(xk(14)/xk(13)))/(KPP+KMAX-
(xk(14)/xk(13)))) * xk(13);
P(12)= muPAO * (xk(1)/(KO2+xk(1))) * (xk(4)/(KNH4+xk(4))) * (xk(8)/(KALK+xk(8)))*
(xk(6)/(KP+xk(6))) * ((xk(15)/xk(13))/(KPHA+(xk(15)/xk(13)))) * xk(13);
P(13)= bPAO * xk(13) * (xk(8)/(KALK+xk(8)));
P(14)= bPP * xk(14) * (xk(8)/(KALK+xk(8)));
P(15)= bPHA * xk(15) * (xk(8)/(KALK+xk(8)));
P(16)= muAUT * (xk(1)/(KO2+xk(1))) * (xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) *
(xk(8)/(KALK+xk(8))) * xk(16);
P(17)= bAUT * xk(16);
P(18)= kPRE * xk(6) * xk(18);
P(19)= kRED * xk(19) * (xk(8)/(KALK+xk(8)));

% TANK 2
P(20)= Kh * (xk(20)/(KO2+xk(20))) * ((xk(30)/xk(31))/(KX+(xk(30)/xk(31)))) * xk(31);
P(21)= Kh * etaNO3 * (KO2/(KO2+xk(20))) * (xk(24)/(KNO3+xk(24))) *
((xk(30)/xk(31))/(KX+(xk(30)/xk(31)))) * xk(31);
P(22)= Kh * etafe * (KO2/(KO2+xk(20))) * (KNO3/(KNO3+xk(24))) *
((xk(30)/xk(31))/(KX+(xk(30)/xk(31)))) * xk(31);
P(23)= muH * (xk(20)/(KO2+xk(20))) * (xk(21)/(KF+xk(21))) * (xk(21)/(xk(21)+xk(22))) *
(xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) * (xk(27)/(KALK+xk(27))) * xk(31);
P(24)= muH * (xk(20)/(KO2+xk(20))) * (xk(22)/(KA+xk(22))) * (xk(22)/(xk(21)+xk(22))) *
(xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) * (xk(27)/(KALK+xk(27))) * xk(31);
```

215

```
P(25)= muH * etaNO3 * (KO2/(KO2+xk(20))) * (xk(21)/(KF+xk(21))) *
(xk(21)/(xk(21)+xk(22))) * (xk(23)/(KNH4+xk(23))) * (xk(24)/(KNO3+xk(24))) *
(xk(27)/(KALK+xk(27))) * xk(31);
P(26)= muH * etaNO3 * (KO2/(KO2+xk(20))) * (xk(22)/(KA+xk(22))) *
(xk(22)/(xk(21)+xk(22))) * (xk(23)/(KNH4+xk(23))) * (xk(24)/(KNO3+xk(24))) *
(xk(27)/(KALK+xk(27)))* (xk(25)/(KP+xk(25))) * xk(31);
P(27)= qfe * (KO2/(KO2+xk(20))) * (KNO3/(KNO3+xk(24))) * (xk(21)/(Kfe+xk(21))) *
(xk(27)/(KALK+xk(27)))* xk(31);
P(28)= bH * xk(31);
P(29)= qPHA * (xk(22)/(KA+xk(22))) * (xk(27)/(KALK+xk(27))) *
((xk(33)/xk(32))/(KPP+(xk(33)/xk(32)))) * xk(32);
P(30)= qPP * (xk(20)/(KO2+xk(20))) * (xk(25)/(KPS+xk(25))) * (xk(27)/(KALK+xk(27))) *
((xk(34)/xk(32))/(KPHA+(xk(34)/xk(32)))) * ((KMAX-(xk(33)/xk(32)))/(KPP+KMAX-
(xk(33)/xk(32)))) * xk(32);
P(31)= muPAO * (xk(20)/(KO2+xk(20))) * (xk(23)/(KNH4+xk(23))) * (xk(27)/(KALK+xk(27)))*
(xk(25)/(KP+xk(25))) * ((xk(34)/xk(32))/(KPHA+(xk(34)/xk(32)))) * xk(32);
P(32)= bPAO * xk(32) * (xk(27)/(KALK+xk(27)));
P(33)= bPP * xk(33) * (xk(27)/(KALK+xk(27)));
P(34)= bPHA * xk(34) * (xk(27)/(KALK+xk(27)));
P(35)= muAUT * (xk(20)/(KO2+xk(20))) * (xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) *
(xk(27)/(KALK+xk(27))) * xk(35);
P(36)= bAUT * xk(35);
P(37)= kPRE * xk(25) * xk(37);
P(38)= kRED * xk(38) * (xk(27)/(KALK+xk(27)));

% TANK 3
P(39)= Kh * (xk(39)/(KO2+xk(39))) * ((xk(49)/xk(50))/(KX+(xk(49)/xk(50)))) * xk(50);
P(40)= Kh * etaNO3 * (KO2/(KO2+xk(39))) * (xk(43)/(KNO3+xk(43))) *
((xk(49)/xk(50))/(KX+(xk(49)/xk(50)))) * xk(50);
P(41)= Kh * etafe * (KO2/(KO2+xk(39))) * (KNO3/(KNO3+xk(43))) *
((xk(49)/xk(50))/(KX+(xk(49)/xk(50)))) * xk(50);
P(42)= muH * (xk(39)/(KO2+xk(39))) * (xk(40)/(KF+xk(40))) * (xk(40)/(xk(40)+xk(41))) *
(xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) * (xk(46)/(KALK+xk(46))) * xk(50);
P(43)= muH * (xk(39)/(KO2+xk(39))) * (xk(41)/(KA+xk(41))) * (xk(41)/(xk(40)+xk(41))) *
(xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) * (xk(46)/(KALK+xk(46))) * xk(50);
P(44)= muH * etaNO3 * (KO2/(KO2+xk(39))) * (xk(40)/(KF+xk(40))) *
(xk(40)/(xk(40)+xk(41))) * (xk(42)/(KNH4+xk(42))) * (xk(43)/(KNO3+xk(43))) *
(xk(46)/(KALK+xk(46))) * xk(50);
P(45)= muH * etaNO3 * (KO2/(KO2+xk(39))) * (xk(41)/(KA+xk(41))) *
(xk(41)/(xk(40)+xk(41))) * (xk(42)/(KNH4+xk(42))) * (xk(43)/(KNO3+xk(43))) *
(xk(46)/(KALK+xk(46)))* (xk(44)/(KP+xk(44))) * xk(50);
P(46)= qfe * (KO2/(KO2+xk(39))) * (KNO3/(KNO3+xk(43))) * (xk(40)/(Kfe+xk(40))) *
(xk(46)/(KALK+xk(46)))* xk(50);
P(47)= bH * xk(50);
P(48)= qPHA * (xk(41)/(KA+xk(41))) * (xk(46)/(KALK+xk(46))) *
((xk(52)/xk(51))/(KPP+(xk(52)/xk(51)))) * xk(51);
P(49)= qPP * (xk(39)/(KO2+xk(39))) * (xk(44)/(KPS+xk(44))) * (xk(46)/(KALK+xk(46))) *
((xk(53)/xk(51))/(KPHA+(xk(53)/xk(51)))) * ((KMAX-(xk(52)/xk(51)))/(KPP+KMAX-
(xk(52)/xk(51)))) * xk(51);
P(50)= muPAO * (xk(39)/(KO2+xk(39))) * (xk(42)/(KNH4+xk(42))) * (xk(46)/(KALK+xk(46)))*
(xk(44)/(KP+xk(44))) * ((xk(53)/xk(51))/(KPHA+(xk(53)/xk(51)))) * xk(51);
P(51)= bPAO * xk(51) * (xk(46)/(KALK+xk(46)));
P(52)= bPP * xk(52) * (xk(46)/(KALK+xk(46)));
P(53)= bPHA * xk(53) * (xk(46)/(KALK+xk(46)));
P(54)= muAUT * (xk(39)/(KO2+xk(39))) * (xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) *
(xk(46)/(KALK+xk(46))) * xk(54);
P(55)= bAUT * xk(54);
P(56)= kPRE * xk(44) * xk(56);
P(57)= kRED * xk(57) * (xk(46)/(KALK+xk(46)));

% TANK 4
P(58)= Kh * (xk(58)/(KO2+xk(58))) * ((xk(68)/xk(69))/(KX+(xk(68)/xk(69)))) * xk(69);
P(59)= Kh * etaNO3 * (KO2/(KO2+xk(58))) * (xk(62)/(KNO3+xk(62))) *
((xk(68)/xk(69))/(KX+(xk(68)/xk(69)))) * xk(69);
P(60)= Kh * etafe * (KO2/(KO2+xk(58))) * (KNO3/(KNO3+xk(62))) *
((xk(68)/xk(69))/(KX+(xk(68)/xk(69)))) * xk(69);
```

216

```
P(61)= muH * (xk(58)/(KO2+xk(58))) * (xk(59)/(KF+xk(59))) * (xk(59)/(xk(59)+xk(60))) *
(xk(61)/(KNH4+xk(61))) * (xk(63)/(KP+xk(63))) * (xk(65)/(KALK+xk(65))) * xk(69);
P(62)= muH * (xk(58)/(KO2+xk(58))) * (xk(60)/(KA+xk(60))) * (xk(60)/(xk(59)+xk(60))) *
(xk(61)/(KNH4+xk(61))) * (xk(63)/(KP+xk(63))) * (xk(65)/(KALK+xk(65))) * xk(69);
P(63)= muH * etaNO3 * (KO2/(KO2+xk(58))) * (xk(59)/(KF+xk(59))) *
(xk(59)/(xk(59)+xk(60))) * (xk(61)/(KNH4+xk(61))) * (xk(62)/(KNO3+xk(62))) *
(xk(65)/(KALK+xk(65))) * xk(69);
P(64)= muH * etaNO3 * (KO2/(KO2+xk(58))) * (xk(60)/(KA+xk(60))) *
(xk(60)/(xk(59)+xk(60))) * (xk(61)/(KNH4+xk(61))) * (xk(62)/(KNO3+xk(62))) *
(xk(65)/(KALK+xk(65)))* (xk(63)/(KP+xk(63))) * xk(69);
P(65)= qfe * (KO2/(KO2+xk(58))) * (KNO3/(KNO3+xk(62))) * (xk(59)/(Kfe+xk(59))) *
(xk(65)/(KALK+xk(65)))* xk(69);
P(66)= bH * xk(69);
P(67)= qPHA * (xk(60)/(KA+xk(60))) * (xk(65)/(KALK+xk(65))) *
((xk(71)/xk(70))/(KPP+(xk(71)/xk(70)))) * xk(70);
P(68)= qPP * (xk(58)/(KO2+xk(58))) * (xk(63)/(KPS+xk(63))) * (xk(65)/(KALK+xk(65))) *
((xk(72)/xk(70))/(KPHA+(xk(72)/xk(70)))) * ((KMAX-(xk(71)/xk(70)))/(KPP+KMAX-
(xk(71)/xk(70)))) * xk(70);
P(69)= muPAO * (xk(58)/(KO2+xk(58))) * (xk(61)/(KNH4+xk(61))) * (xk(65)/(KALK+xk(65)))*
(xk(63)/(KP+xk(63))) * ((xk(72)/xk(70))/(KPHA+(xk(72)/xk(70)))) * xk(70);
P(70)= bPAO * xk(70) * (xk(65)/(KALK+xk(65)));
P(71)= bPP * xk(71) * (xk(65)/(KALK+xk(65)));
P(72)= bPHA * xk(72) * (xk(65)/(KALK+xk(65)));
P(73)= muAUT * (xk(58)/(KO2+xk(58))) * (xk(61)/(KNH4+xk(61))) * (xk(63)/(KP+xk(63))) *
(xk(65)/(KALK+xk(65))) * xk(73);
P(74)= bAUT * xk(73);
P(75)= kPRE * xk(63) * xk(75);
P(76)= kRED * xk(76) * (xk(65)/(KALK+xk(65)));

% TANK 5
P(77)= Kh * (xk(77)/(KO2+xk(77))) * ((xk(87)/xk(88))/(KX+(xk(87)/xk(88)))) * xk(88);
P(78)= Kh * etaNO3 * (KO2/(KO2+xk(77))) * (xk(81)/(KNO3+xk(81))) *
((xk(87)/xk(88))/(KX+(xk(87)/xk(88)))) * xk(88);
P(79)= Kh * etafe * (KO2/(KO2+xk(77))) * (KNO3/(KNO3+xk(81))) *
((xk(87)/xk(88))/(KX+(xk(87)/xk(88)))) * xk(88);
P(80)= muH * (xk(77)/(KO2+xk(77))) * (xk(78)/(KF+xk(78))) * (xk(78)/(xk(78)+xk(79))) *
(xk(80)/(KNH4+xk(80))) * (xk(82)/(KP+xk(82))) * (xk(84)/(KALK+xk(84))) * xk(88);
P(81)= muH * (xk(77)/(KO2+xk(77))) * (xk(79)/(KA+xk(79))) * (xk(79)/(xk(78)+xk(79))) *
(xk(80)/(KNH4+xk(80))) * (xk(82)/(KP+xk(82))) * (xk(84)/(KALK+xk(84))) * xk(88);
P(82)= muH * etaNO3 * (KO2/(KO2+xk(77))) * (xk(78)/(KF+xk(78))) *
(xk(78)/(xk(78)+xk(79))) * (xk(80)/(KNH4+xk(80))) * (xk(81)/(KNO3+xk(81))) *
(xk(84)/(KALK+xk(84))) * xk(88);
P(83)= muH * etaNO3 * (KO2/(KO2+xk(77))) * (xk(79)/(KA+xk(79))) *
(xk(79)/(xk(78)+xk(79))) * (xk(80)/(KNH4+xk(80))) * (xk(81)/(KNO3+xk(81))) *
(xk(84)/(KALK+xk(84)))* (xk(82)/(KP+xk(82))) * xk(88);
P(84)= qfe * (KO2/(KO2+xk(77))) * (KNO3/(KNO3+xk(81))) * (xk(78)/(Kfe+xk(78))) *
(xk(84)/(KALK+xk(84)))* xk(88);
P(85)= bH * xk(88);
P(86)= qPHA * (xk(79)/(KA+xk(79))) * (xk(84)/(KALK+xk(84))) *
((xk(90)/xk(89))/(KPP+(xk(90)/xk(89)))) * xk(89);
P(87)= qPP * (xk(77)/(KO2+xk(77))) * (xk(82)/(KPS+xk(82))) * (xk(84)/(KALK+xk(84))) *
((xk(91)/xk(89))/(KPHA+(xk(91)/xk(89)))) * ((KMAX-(xk(90)/xk(89)))/(KPP+KMAX-
(xk(90)/xk(89)))) * xk(89);
P(88)= muPAO * (xk(77)/(KO2+xk(77))) * (xk(80)/(KNH4+xk(80))) * (xk(84)/(KALK+xk(84)))*
(xk(82)/(KP+xk(82))) * ((xk(91)/xk(89))/(KPHA+(xk(91)/xk(89)))) * xk(89);
P(89)= bPAO * xk(89) * (xk(84)/(KALK+xk(84)));
P(90)= bPP * xk(90) * (xk(84)/(KALK+xk(84)));
P(91)= bPHA * xk(91) * (xk(84)/(KALK+xk(84)));
P(92)= muAUT * (xk(77)/(KO2+xk(77))) * (xk(80)/(KNH4+xk(80))) * (xk(82)/(KP+xk(82))) *
(xk(84)/(KALK+xk(84))) * xk(92);
P(93)= bAUT * xk(92);
P(94)= kPRE * xk(82) * xk(94);
P(95)= kRED * xk(95) * (xk(84)/(KALK+xk(84)));


P = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20) P(21) P(22) P(23) P(24) P(25) P(26) P(27) P(28) P(29)
```

```
P(19)= etaNO3 * (KO2/(KO2+uk(2))) * (zk(10)/(KA+zk(10))) * (zk(10)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(12)/(KNO3+zk(12))) * (SALKk(2)/(KALK+SALKk(2)))*
(zk(13)/(KP+zk(13))) * XHk(2);
P(20)= (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(9)/(Kfe+zk(9))) *
(SALKk(2)/(KALK+SALKk(2)))* XHk(2);
P(21)= (zk(10)/(KA+zk(10))) * (SALKk(2)/(KALK+SALKk(2))) *
((XPAOk(2)/XPAOk(2))/(KPP+(XPAOk(2)/XPAOk(2)))) * XPAOk(2);
P(22)= (uk(2)/(KO2+uk(2))) * (zk(13)/(KPS+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) *
((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * ((KMAX-(XPAOk(2)/XPAOk(2)))/(KPP+KMAX-
(XPAOk(2)/XPAOk(2)))) * XPAOk(2);
P(23)= (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (SALKk(2)/(KALK+SALKk(2)))*
(zk(13)/(KP+zk(13))) * ((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * XPAOk(2);
P(24)= (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XAUTk(2);

P2k = [[P(13) P(14) P(15) 0      0      0      0      0      0      0      0      0     ];
%Kh(2)
      [0      0      0      P(16)  P(17)  P(18)  P(19)  0      0      0      0      0     ];
%muH(2)
      [0      0      0      0      0      0      0      P(20)  0      0      0      0     ];
%qfe(2)
      [0      0      0      0      0      0      0      0      P(21)  0      0      0     ];
%qPP(2)
      [0    -  0      0      0      0      0      0      0      0      P(22)  0      0     ];
%qPHA(2)
      [0      0      0      0      0      0      0      0      0      0      P(23)  0     ];
%muPAO(2)
      [0      0      0      0      0      0      0      0      0      0      0
P(24)]];%muAUT(2)


% TANK 3
P(25)= (uk(3)/(KO2+uk(3))) * ((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(26)= etaNO3 * (KO2/(KO2+uk(3))) * (zk(20)/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(27)= etafe * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(28)= (uk(3)/(KO2+uk(3))) * (zk(17)/(KF+zk(17))) * (zk(17)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(29)= (uk(3)/(KO2+uk(3))) * (zk(18)/(KA+zk(18))) * (zk(18)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(30)= etaNO3 * (KO2/(KO2+uk(3))) * (zk(17)/(KF+zk(17))) * (zk(17)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(20)/(KNO3+zk(20))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(31)= etaNO3 * (KO2/(KO2+uk(3))) * (zk(18)/(KA+zk(18))) * (zk(18)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(20)/(KNO3+zk(20))) * (SALKk(3)/(KALK+SALKk(3)))*
(zk(21)/(KP+zk(21))) * XHk(3);
P(32)= (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(17)/(Kfe+zk(17))) *
(SALKk(3)/(KALK+SALKk(3)))* XHk(3);
P(33)= (zk(18)/(KA+zk(18))) * (SALKk(3)/(KALK+SALKk(3))) *
((XPAOk(3)/XPAOk(3))/(KPP+(XPAOk(3)/XPAOk(3)))) * XPAOk(3);
P(34)= (uk(3)/(KO2+uk(3))) * (zk(21)/(KPS+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) *
((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * ((KMAX-(XPAOk(3)/XPAOk(3)))/(KPP+KMAX-
(XPAOk(3)/XPAOk(3)))) * XPAOk(3);
P(35)= (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (SALKk(3)/(KALK+SALKk(3)))*
(zk(21)/(KP+zk(21))) * ((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * XPAOk(3);
P(36)= (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XAUTk(3);

P3k = [[P(25) P(26) P(27) 0      0      0      0      0      0      0      0      0     ];
%Kh(3)
      [0      0      0      P(28)  P(29)  P(30)  P(31)  0      0      0      0      0     ];
%muH(3)
      [0      0      0      0      0      0      0      P(32)  0      0      0      0     ];
%qfe(3)
      [0      0      0      0      0      0      0      0      P(33)  0      0      0     ];
%qPP(3)
```

```
          [0      0     0      0      0      0      0      0      0      P(34) 0     0    ];
%qPHA(3)
          [0      0     0      0      0      0      0      0      0      P(35) 0    ];
%muPAO(3)
          [0      0     0      0      0      0      0      0      0      0
P(36)]];%muAUT(3)


% TANK 4
P(37)= (uk(4)/(KO2+uk(4))) * ((zk(30)/XHk(4))/(KX+(zk(30)/XHk(4)))) * XHk(4);
P(38)= etaNO3 * (KO2/(KO2+uk(4))) * (zk(27)/(KNO3+zk(27))) *
((zk(30)/XHk(4))/(KX+(zk(30)/XHk(4)))) * XHk(4);
P(39)= etafe * (KO2/(KO2+uk(4))) * (KNO3/(KNO3+zk(27))) *
((zk(30)/XHk(4))/(KX+(zk(30)/XHk(4)))) * XHk(4);
P(40)= (uk(4)/(KO2+uk(4))) * (zk(25)/(KF+zk(25))) * (zk(25)/(zk(25)+zk(26))) *
(zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) * (SALKk(4)/(KALK+SALKk(4))) * XHk(4);
P(41)= (uk(4)/(KO2+uk(4))) * (zk(26)/(KA+zk(26))) * (zk(26)/(zk(25)+zk(26))) *
(zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) * (SALKk(4)/(KALK+SALKk(4))) * XHk(4);
P(42)= etaNO3 * (KO2/(KO2+uk(4))) * (zk(25)/(KF+zk(25))) * (zk(25)/(zk(25)+zk(26))) *
(zk(27)/(KNH4+zk(27))) * (zk(27)/(KNO3+zk(27))) * (SALKk(4)/(KALK+SALKk(4))) * XHk(4);
P(43)= etaNO3 * (KO2/(KO2+uk(4))) * (zk(26)/(KA+zk(26))) * (zk(26)/(zk(25)+zk(26))) *
(zk(27)/(KNH4+zk(27))) * (zk(27)/(KNO3+zk(27))) * (SALKk(4)/(KALK+SALKk(4)))*
(zk(29)/(KP+zk(29))) * XHk(4);
P(44)= (KO2/(KO2+uk(4))) * (KNO3/(KNO3+zk(27))) * (zk(25)/(Kfe+zk(25))) *
(SALKk(4)/(KALK+SALKk(4)))* XHk(4);
P(45)= (zk(26)/(KA+zk(26))) * (SALKk(4)/(KALK+SALKk(4))) *
((XPAOk(4)/XPAOk(4))/(KPP+(XPAOk(4)/XPAOk(4)))) * XPAOk(4);
P(46)= (uk(4)/(KO2+uk(4))) * (zk(29)/(KPS+zk(29))) * (SALKk(4)/(KALK+SALKk(4))) *
((zk(32)/XPAOk(4))/(KPHA+(zk(32)/XPAOk(4)))) * ((KMAX-(XPAOk(4)/XPAOk(4)))/(KPP+KMAX-
(XPAOk(4)/XPAOk(4)))) * XPAOk(4);
P(47)= (uk(4)/(KO2+uk(4))) * (zk(27)/(KNH4+zk(27))) * (SALKk(4)/(KALK+SALKk(4)))*
(zk(29)/(KP+zk(29))) * ((zk(32)/XPAOk(4))/(KPHA+(zk(32)/XPAOk(4)))) * XPAOk(4);
P(48)= (uk(4)/(KO2+uk(4))) * (zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) *
(SALKk(4)/(KALK+SALKk(4))) * XAUTk(4);


P4k = [[P(37) P(38) P(39) 0      0      0      0      0      0      0      0      0    ];
%Kh(4)
          [0      0     0      P(40) P(41) P(42) P(43) 0      0      0      0      0    ];
%muH(4)
          [0      0     0      0      0      0      0      P(44) 0      0      0      0    ];
%qfe(4)
          [0      0     0      0      0      0      0      0      P(45) 0      0      0    ];
%qPP(4)
          [0      0     0      0      0      0      0      0      P(46) 0      0    ];
%qPHA(4)
          [0      0     0      0      0      0      0      0      0      P(47) 0    ];
%muPAO(4)
          [0      0     0      0      0      0      0      0      0      0
P(48)]];%muAUT(4)


% TANK 5
P(49)= (uk(5)/(KO2+uk(5))) * ((zk(38)/XHk(5))/(KX+(zk(38)/XHk(5)))) * XHk(5);
P(50)= etaNO3 * (KO2/(KO2+uk(5))) * (zk(36)/(KNO3+zk(36))) *
((zk(38)/XHk(5))/(KX+(zk(38)/XHk(5)))) * XHk(5);
P(51)= etafe * (KO2/(KO2+uk(5))) * (KNO3/(KNO3+zk(36))) *
((zk(38)/XHk(5))/(KX+(zk(38)/XHk(5)))) * XHk(5);
P(52)= (uk(5)/(KO2+uk(5))) * (zk(33)/(KF+zk(33))) * (zk(33)/(zk(33)+zk(34))) *
(zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) * (SALKk(5)/(KALK+SALKk(5))) * XHk(5);
P(53)= (uk(5)/(KO2+uk(5))) * (zk(34)/(KA+zk(34))) * (zk(34)/(zk(33)+zk(34))) *
(zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) * (SALKk(5)/(KALK+SALKk(5))) * XHk(5);
P(54)= etaNO3 * (KO2/(KO2+uk(5))) * (zk(33)/(KF+zk(33))) * (zk(33)/(zk(33)+zk(34))) *
(zk(35)/(KNH4+zk(35))) * (zk(36)/(KNO3+zk(36))) * (SALKk(5)/(KALK+SALKk(5))) * XHk(5);
P(55)= etaNO3 * (KO2/(KO2+uk(5))) * (zk(34)/(KA+zk(34))) * (zk(34)/(zk(33)+zk(34))) *
(zk(35)/(KNH4+zk(35))) * (zk(36)/(KNO3+zk(36))) * (SALKk(5)/(KALK+SALKk(5)))*
(zk(37)/(KP+zk(37))) * XHk(5);
P(56)= (KO2/(KO2+uk(5))) * (KNO3/(KNO3+zk(36))) * (zk(33)/(Kfe+zk(33))) *
(SALKk(5)/(KALK+SALKk(5)))* XHk(5);
```

```
P(57)= (zk(34)/(KA+zk(34))) * (SALKk(5)/(KALK+SALKk(5))) *
((XPAOk(5)/XPAOk(5))/(KPP+(XPAOk(5)/XPAOk(5)))) * XPAOk(5);
P(58)= (uk(5)/(KO2+uk(5))) * (zk(37)/(KPS+zk(37))) * (SALKk(5)/(KALK+SALKk(5))) *
((zk(40)/XPAOk(5))/(KPHA+(zk(40)/XPAOk(5)))) * ((KMAX-(XPAOk(5)/XPAOk(5)))/(KPP+KMAX-
(XPAOk(5)/XPAOk(5)))) * XPAOk(5);
P(59)= (uk(5)/(KO2+uk(5))) * (zk(35)/(KNH4+zk(35))) * (SALKk(5)/(KALK+SALKk(5)))*
(zk(37)/(KP+zk(37))) * ((zk(40)/XPAOk(5))/(KPHA+(zk(40)/XPAOk(5)))) * XPAOk(5);
P(60)= (uk(5)/(KO2+uk(5))) * (zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) *
(SALKk(5)/(KALK+SALKk(5))) * XAUTk(5);


P5k = [[P(49) P(50) P(51) 0      0      0      0      0      0      0      0      0     ];
%Kh(5)
       [0      0      0      P(52) P(53) P(54) P(55) 0      0      0      0      0     ];
%muH(5)
       [0      0      0      0      0      0      0      P(56) 0      0      0      0     ];
%qfe(5)
       [0      0      0      0      0      0      0      0      P(57) 0      0      0     ];
%qPP(5)
       [0      0      0      0      0      0      0      0      0      P(58) 0      0     ];
%qPHA(5)
       [0      0      0      0      0      0      0      0      0      0      P(59) 0     ];
%muPAO(5)
       [0      0      0      0      0      0      0      0      0      0      0
P(60)]];%muAUT(5)

%Matrix Pk for the whole process
Pk = [P1k'; P2k'; P3k'; P4k'; P5k'];
```

B.4: MATLAB script file – P_BASM2RE.m

```
% M-File: P_BASM2RE.m
% ========================== M-FILE DESCRIPTION ===========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model

% Description of the function function
function Pk =
P_BASM2RE(zk,uk,XHk,XAUTk,XPAOk,SALKk,kHek,muHek,qfeek,qPHAek,qPPek,muPAOek,muAUTek)


global fSI YH fXI YPO4 YPHA iPBM YA iNBM Kh etaNO3 etafe muH qfe bH qPHA qPP muPAO bPAO
bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4 KALK KA Kfe KPP KPS KP KPHA KIPP KMAX
% TANK 1
P(1)= kHek * (uk(1)/(KO2+uk(1))) * ((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(2)= kHek * etaNO3 * (KO2/(KO2+uk(1))) * (zk(4)/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(3)= kHek * etafe * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(4)= muHek * (uk(1)/(KO2+uk(1))) * (zk(1)/(KF+zk(1))) * (zk(1)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(5)= muHek * (uk(1)/(KO2+uk(1))) * (zk(2)/(KA+zk(2))) * (zk(2)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(6)= muHek * etaNO3 * (KO2/(KO2+uk(1))) * (zk(1)/(KF+zk(1))) * (zk(1)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(4)/(KNO3+zk(4))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(7)= muHek * etaNO3 * (KO2/(KO2+uk(1))) * (zk(2)/(KA+zk(2))) * (zk(2)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(4)/(KNO3+zk(4))) * (SALKk(1)/(KALK+SALKk(1)))*
(zk(5)/(KP+zk(5))) * XHk(1);
P(8)= qfeek * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(1)/(Kfe+zk(1))) *
(SALKk(1)/(KALK+SALKk(1)))* XHk(1);
P(9)= qPHAek * (zk(2)/(KA+zk(2))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(7)/XPAOk(1))/(KPP+(zk(7)/XPAOk(1)))) * XPAOk(1);
P(10)= qPPek * (uk(1)/(KO2+uk(1))) * (zk(5)/(KPS+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * ((KMAX-(zk(7)/XPAOk(1)))/(KPP+KMAX-
(zk(7)/XPAOk(1)))) * XPAOk(1);
```

221

```
P(11)= muPAOek * (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) *
(SALKk(1)/(KALK+SALKk(1)))* (zk(5)/(KP+zk(5))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * XPAOk(1);
P(12)= muAUTek * (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XAUTk(1);

% TANK 2
P(13)= kHek * (uk(2)/(KO2+uk(2))) * ((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(14)= kHek * etaNO3 * (KO2/(KO2+uk(2))) * (zk(12)/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(15)= kHek * etafe * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(16)= muHek * (uk(2)/(KO2+uk(2))) * (zk(9)/(KF+zk(9))) * (zk(9)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(17)= muHek * (uk(2)/(KO2+uk(2))) * (zk(10)/(KA+zk(10))) * (zk(10)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(18)= muHek * etaNO3 * (KO2/(KO2+uk(2))) * (zk(9)/(KF+zk(9))) * (zk(9)/(zk(9)+zk(10)))
* (zk(11)/(KNH4+zk(11))) * (zk(12)/(KNO3+zk(12))) * (SALKk(2)/(KALK+SALKk(2))) *
XHk(2);
P(19)= muHek * etaNO3 * (KO2/(KO2+uk(2))) * (zk(10)/(KA+zk(10))) *
(zk(10)/(zk(9)+zk(10))) * (zk(11)/(KNH4+zk(11))) * (zk(12)/(KNO3+zk(12))) *
(SALKk(2)/(KALK+SALKk(2)))* (zk(13)/(KP+zk(13))) * XHk(2);
P(20)= qfeek * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(9)/(Kfe+zk(9))) *
(SALKk(2)/(KALK+SALKk(2)))* XHk(2);
P(21)= qPHAek * (zk(10)/(KA+zk(10))) * (SALKk(2)/(KALK+SALKk(2))) *
((XPAOk(2)/XPAOk(2))/(KPP+(XPAOk(2)/XPAOk(2)))) * XPAOk(2);
P(22)= qPPek * (uk(2)/(KO2+uk(2))) * (zk(13)/(KPS+zk(13))) * (SALKk(2)/(KALK+SALKk(2)))
* ((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * ((KMAX-(XPAOk(2)/XPAOk(2)))/(KPP+KMAX-
(XPAOk(2)/XPAOk(2)))) * XPAOk(2);
P(23)= muPAOek * (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) *
(SALKk(2)/(KALK+SALKk(2)))* (zk(13)/(KP+zk(13))) *
((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * XPAOk(2);
P(24)= muAUTek * (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XAUTk(2);

% TANK 3
P(25)= kHek * (uk(3)/(KO2+uk(3))) * ((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(26)= kHek * etaNO3 * (KO2/(KO2+uk(3))) * (zk(20)/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(27)= kHek * etafe * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(28)= muHek * (uk(3)/(KO2+uk(3))) * (zk(17)/(KF+zk(17))) * (zk(17)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(29)= muHek * (uk(3)/(KO2+uk(3))) * (zk(18)/(KA+zk(18))) * (zk(18)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(30)= muHek * etaNO3 * (KO2/(KO2+uk(3))) * (zk(17)/(KF+zk(17))) *
(zk(17)/(zk(17)+zk(18))) * (zk(19)/(KNH4+zk(19))) * (zk(20)/(KNO3+zk(20))) *
(SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(31)= muHek * etaNO3 * (KO2/(KO2+uk(3))) * (zk(18)/(KA+zk(18))) *
(zk(18)/(zk(17)+zk(18))) * (zk(19)/(KNH4+zk(19))) * (zk(20)/(KNO3+zk(20))) *
(SALKk(3)/(KALK+SALKk(3))) * (zk(21)/(KP+zk(21))) * XHk(3);
P(32)= qfeek * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(17)/(Kfe+zk(17))) *
(SALKk(3)/(KALK+SALKk(3)))* XHk(3);
P(33)= qPHAek * (zk(18)/(KA+zk(18))) * (SALKk(3)/(KALK+SALKk(3))) *
((XPAOk(3)/XPAOk(3))/(KPP+(XPAOk(3)/XPAOk(3)))) * XPAOk(3);
P(34)= qPPek * (uk(3)/(KO2+uk(3))) * (zk(21)/(KPS+zk(21))) * (SALKk(3)/(KALK+SALKk(3)))
* ((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * ((KMAX-(XPAOk(3)/XPAOk(3)))/(KPP+KMAX-
(XPAOk(3)/XPAOk(3)))) * XPAOk(3);
P(35)= muPAOek * (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) *
(SALKk(3)/(KALK+SALKk(3)))* (zk(21)/(KP+zk(21))) *
((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * XPAOk(3);
P(36)= muAUTek * (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XAUTk(3);

% TANK 4
P(37)= kHek * (uk(4)/(KO2+uk(4))) * ((zk(30)/XHk(4))/(KX+(zk(30)/XHk(4)))) * XHk(4);
```

```
P(38)= kHek * etaNO3 * (KO2/(KO2+uk(4))) * (zk(27)/(KNO3+zk(27))) *
((zk(30)/XHk(4))/(KX+(zk(30)/XHk(4)))) * XHk(4);
P(39)= kHek * etafe * (KO2/(KO2+uk(4))) * (KNO3/(KNO3+zk(27))) *
((zk(30)/XHk(4))/(KX+(zk(30)/XHk(4)))) * XHk(4);
P(40)= muHek * (uk(4)/(KO2+uk(4))) * (zk(25)/(KF+zk(25))) * (zk(25)/(zk(25)+zk(26))) *
(zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) * (SALKk(4)/(KALK+SALKk(4))) * XHk(4);
P(41)= muHek * (uk(4)/(KO2+uk(4))) * (zk(26)/(KA+zk(26))) * (zk(26)/(zk(25)+zk(26))) *
(zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) * (SALKk(4)/(KALK+SALKk(4))) * XHk(4);
P(42)= muHek * etaNO3 * (KO2/(KO2+uk(4))) * (zk(25)/(KF+zk(25))) *
(zk(25)/(zk(25)+zk(26))) * (zk(27)/(KNH4+zk(27))) * (zk(27)/(KNO3+zk(27))) *
(SALKk(4)/(KALK+SALKk(4))) * XHk(4);
P(43)= muHek * etaNO3 * (KO2/(KO2+uk(4))) * (zk(26)/(KA+zk(26))) *
(zk(26)/(zk(25)+zk(26))) * (zk(27)/(KNH4+zk(27))) * (zk(27)/(KNO3+zk(27))) *
(SALKk(4)/(KALK+SALKk(4)))* (zk(29)/(KP+zk(29))) * XHk(4);
P(44)= qfeek * (KO2/(KO2+uk(4))) * (KNO3/(KNO3+zk(27))) * (zk(25)/(Kfe+zk(25))) *
(SALKk(4)/(KALK+SALKk(4)))* XHk(4);
P(45)= qPHAek * (zk(26)/(KA+zk(26))) * (SALKk(4)/(KALK+SALKk(4))) *
((XPAOk(4)/XPAOk(4))/(KPP+(XPAOk(4)/XPAOk(4)))) * XPAOk(4);
P(46)= qPPek * (uk(4)/(KO2+uk(4))) * (zk(29)/(KPS+zk(29))) * (SALKk(4)/(KALK+SALKk(4)))
* ((zk(32)/XPAOk(4))/(KPHA+(zk(32)/XPAOk(4)))) * ((KMAX-(XPAOk(4)/XPAOk(4)))/(KPP+KMAX-
(XPAOk(4)/XPAOk(4)))) * XPAOk(4);
P(47)= muPAOek * (uk(4)/(KO2+uk(4))) * (zk(27)/(KNH4+zk(27))) *
(SALKk(4)/(KALK+SALKk(4)))* (zk(29)/(KP+zk(29))) *
((zk(32)/XPAOk(4))/(KPHA+(zk(32)/XPAOk(4)))) * XPAOk(4);
P(48)= muAUTek * (uk(4)/(KO2+uk(4))) * (zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) *
(SALKk(4)/(KALK+SALKk(4))) * XAUTk(4);


% TANK 5
P(49)= kHek * (uk(5)/(KO2+uk(5))) * ((zk(38)/XHk(5))/(KX+(zk(38)/XHk(5)))) * XHk(5);
P(50)= kHek * etaNO3 * (KO2/(KO2+uk(5))) * (zk(36)/(KNO3+zk(36))) *
((zk(38)/XHk(5))/(KX+(zk(38)/XHk(5)))) * XHk(5);
P(51)= kHek * etafe * (KO2/(KO2+uk(5))) * (KNO3/(KNO3+zk(36))) *
((zk(38)/XHk(5))/(KX+(zk(38)/XHk(5)))) * XHk(5);
P(52)= muHek * (uk(5)/(KO2+uk(5))) * (zk(33)/(KF+zk(33))) * (zk(33)/(zk(33)+zk(34))) *
(zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) * (SALKk(5)/(KALK+SALKk(5))) * XHk(5);
P(53)= muHek * (uk(5)/(KO2+uk(5))) * (zk(34)/(KA+zk(34))) * (zk(34)/(zk(33)+zk(34))) *
(zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) * (SALKk(5)/(KALK+SALKk(5))) * XHk(5);
P(54)= muHek * etaNO3 * (KO2/(KO2+uk(5))) * (zk(33)/(KF+zk(33))) *
(zk(33)/(zk(33)+zk(34))) * (zk(35)/(KNH4+zk(35))) * (zk(36)/(KNO3+zk(36))) *
(SALKk(5)/(KALK+SALKk(5))) * XHk(5);
P(55)= muHek * etaNO3 * (KO2/(KO2+uk(5))) * (zk(34)/(KA+zk(34))) *
(zk(34)/(zk(33)+zk(34))) * (zk(35)/(KNH4+zk(35))) * (zk(36)/(KNO3+zk(36))) *
(SALKk(5)/(KALK+SALKk(5)))* (zk(37)/(KP+zk(37))) * XHk(5);
P(56)= qfeek * (KO2/(KO2+uk(5))) * (KNO3/(KNO3+zk(36))) * (zk(33)/(Kfe+zk(33))) *
(SALKk(5)/(KALK+SALKk(5)))* XHk(5);
P(57)= qPHAek * (zk(34)/(KA+zk(34))) * (SALKk(5)/(KALK+SALKk(5))) *
((XPAOk(5)/XPAOk(5))/(KPP+(XPAOk(5)/XPAOk(5)))) * XPAOk(5);
P(58)= qPPek * (uk(5)/(KO2+uk(5))) * (zk(37)/(KPS+zk(37))) * (SALKk(5)/(KALK+SALKk(5)))
* ((zk(40)/XPAOk(5))/(KPHA+(zk(40)/XPAOk(5)))) * ((KMAX-(XPAOk(5)/XPAOk(5)))/(KPP+KMAX-
(XPAOk(5)/XPAOk(5)))) * XPAOk(5);
P(59)= muPAOek * (uk(5)/(KO2+uk(5))) * (zk(35)/(KNH4+zk(35))) *
(SALKk(5)/(KALK+SALKk(5)))* (zk(37)/(KP+zk(37))) *
((zk(40)/XPAOk(5))/(KPHA+(zk(40)/XPAOk(5)))) * XPAOk(5);
P(60)= muAUTek * (uk(5)/(KO2+uk(5))) * (zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) *
(SALKk(5)/(KALK+SALKk(5))) * XAUTk(5);


Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20) P(21) P(22) P(23) P(24) P(25) P(26) P(27) P(28) P(29)
P(30) P(31) P(32) P(33) P(34) P(35) P(36) P(37) P(38) P(39) P(40) P(41) P(42) P(43)
P(44) P(45) P(46) P(47) P(48) P(49) P(50) P(51) P(52) P(53) P(54) P(55) P(56) P(57)
P(58) P(59) P(60)]';
```

APPENDIX C: DEVELOPED MATLAB CODE FOR OPTIMAL CONTROL OF THE BENCHMARK PLANT WITH THE ACTIVATED SLUDGE MODEL NO.2d BIO-MODEL

C.1: MATLAB script file – BASM2d_Simulation.m

```
% M-File: BASM2d_Simulation.m
% ========================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Simulation of the Benchmark plant using full ASM2d model
% Calculation of the parameters for the reduced ASM2d model
% Simulation of the Benchmark plant using the reduced ASM2d model

clear all          % Clear the workspace
clc                % Clear the command window


% ========================= PARAMETER DEFINITION =========================
% Declaration of the global variables
global fSI YH fXI YPO4 YPHA iPBM YA iNBM Kh etaNO3 etafe muH qfe bH qPHA
global qPP muPAO bPAO bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4
global KALK KA Kfe KPP KPS KP KPHA KIPP KMAX

K1       = 5;           % Prediction horizon or interval
DT       = 0.01042;     % Sampling time

% Initialization of the model parameters, typical values at 20 0C
% Stochiometric parameters
fSI      = 0.0;
YH       = 0.625;
fXI      = 0.10;
YPO4     = 0.40;
YPHA     = 0.20;
iPBM     = 0.02;
YA       = 0.24;
iNBM     = 0.07;
% Kinetic parameters
Kh       = 3;
etaNO3   = 0.8;
etafe    = 0.4;
muH      = 0.6;
qfe      = 3;
bH       = 0.4;
qPHA     = 0.3;
qPP      = 1.5;
muPAO    = 1;
bPAO     = 0.2;
bPP      = 0.2;
bPHA     = 0.2;
muAUT    = 1;
bAUT     = 0.15;
kPRE     = 1;
kRED     = 0.6;
% Monod switching functions
KO2      = 0.2;
KX       = 0.1;
KNO3     = 0.5;
KF       = 4;
KNH4     = 0.05;
KALK     = 0.1;
KA       = 4;
Kfe      = 20;
KPP      = 0.1;
KPS      = 0.20;
KP       = 0.1;
```

```
KPHA      = 0.1;
KIPP      = 0.02;
KMAX      = 0.34;

% Initialization of the Benchmark process data
V1    = 1000;         % Tank 1 volume
V2    = 1000;         % Tank 2 volume
V3    = 1333;         % Tank 3 volume
V4    = 1333;         % Tank 4 volume
V5    = 1333;         % Tank 5 volume

Q0    = 18446;        % Influent flow rate
Qa    = 55338;        % Internal recirculation rate
Qr    = 18446;        % Sludge·recycled rate
Qw    = 385;          % Wasted sludge rate
Q     = Q0 + Qa + Qr;% Flow through the tanks

% Coeficient for fraction of flow rates for the particulate materials
LAMBDA = (Q0+Qr)/(Qr+Qw);

% Oxygen parameters
SOsat = 8;            % Oxygen saturation point
KLa3  = 240;          % Oxygen transfer function in Tank 3
KLa4  = 240;          % Oxygen transfer function in Tank 4
KLa5  = 84;           % Oxygen transfer function in Tank 5

KL  = [KLa3 KLa4 KLa5];
KLa = diag(KL);

% =========================== PROCESS MODEL ===========================
% Calculation of the MATRIX A
% Matrix A represent weighting factors of the accumulated state variables
% from different flows towards state variables in each tank as derived
% through mass balance principles, see chapter 3 of the Thesis
IV1 = DT/ V1; IV11 = IV1 * Q; A11 = 1 - IV11;
IV2 = DT/ V2; IV22 = IV2 * Q; A22 = 1 - IV22;
IV3 = DT/ V3; IV33 = IV3 * Q; A33 = 1 - IV33;
IV4 = DT/ V4; IV44 = IV4 * Q; A44 = 1 - IV44;
IV5 = DT/ V5; IV55 = IV5 * Q; A55 = 1 - IV55;

A15a = IV1 * (Qa + Qr);        % Soluble materials recycled from Tank 5
A15b = IV1 * (Qa + LAMBDA*Qr); % Particulate materials recycled from Tank 5
A21 = IV22;
A32 = IV33;
A43 = IV44;
A54 = IV55;

A11 = A11 * ones(1,19); A11 = diag(A11);
A22 = A22 * ones(1,19); A22 = diag(A22);
A33 = A33 * ones(1,19); A33 = diag(A33);
A44 = A44 * ones(1,19); A44 = diag(A44);
A55 = A55 * ones(1,19); A55 = diag(A55);

A15 =
[A15a,A15a,A15a,A15a,A15a,A15a,A15a,A15a,A15a,A15b,A15b,A15b,A15b,A15b,A15b,A15b,A15b,A
15b,A15b];
A15 = diag(A15);
A21 = A21 * ones(1,19); A21 = diag(A21);
A32 = A32 * ones(1,19); A32 = diag(A32);
A43 = A43 * ones(1,19); A43 = diag(A43);
A54 = A54 * ones(1,19); A54 = diag(A54);

A10 = zeros(19,19); A20 = zeros(19,38); A30 = zeros(19,57);

A = [A11 A30 A15; A21 A22 A30; A10 A32 A33 A20; A20 A43 A44 A10; A30 A54 A55];
```

225

```matlab
% Calculation of the MATRIX B
% Matrix B represent weighting factors of the state variables towards the
% influent wastewater composition as derived through mass balance
% principles, see chapter 3 of the Thesis
IV0 = DT/ V1*Q0;
B11 = IV0 * ones(1,19); B11 = diag(B11);
B = [B11; zeros(76,19)];


% Formulation of the MATRIX C
% Matrix C represent weighting factors of the state variables towards the
% processes on the ASM1 matrix table
%       SO2    SF     SA     SNH4    SNO3     SPO4    SI     SALK    SN2 XI
%       XS     XH     XPAO    XPP     XPHA    XAUT    XTSS    XMeOH   XMeP
CC =[[0 1 0 0 0.01 0 0 0 0.001 0 -1 0 0 0 0 0 -0.75 0 0];
  [0 1 0 0 0.01 0 0 0 0.001 0 -1 0 0 0 0 0 -0.75 0 0];
  [0 1 0 0 0.01 0 0 0 0.001 0 -1 0 0 0 0 0 -0.75 0 0];
  [-0.6 -1.6 0 0 -0.022 0 0 -0.004 -0.001 0 0 1 0 0 0 0 0.9 0 0];
  [-0.6 0 -1.6 0 -0.07 0 0 -0.02 0.021 0 0 1 0 0 0 0 0.9 0 0];
  [0 -1.6 0 0 -0.022 0.21 -0.21 -0.004 0.014 0 0 1 0 0 0 0 0.9 0 0];
  [0 0 -1.6 0 -0.07 0.21 -0.21 -0.02 0.036 0 0 1 0 0 0 0 0.9 0 0];
  [0 -1 1 0 0.03 0 0 0.01 -0.014 0 0 0 0 0 0 0 0 0 0];
  [0 0 0 0 0.031 0 0 0.01 0.002 0.1 0.9 -1 0 0 0 0 -0.15 0 0];
  [0 0 -1 0 0 0 0 0.4 0.009 0 0 0 0 -0.4 1 0 -0.69 0 0];
  [-0.2 0 0 0 0 0 0 -1 0.016 0 0 0 0 1 -0.2 0 3.11 0 0];
  [0 0 0 0 0 0.07 -0.07 -1 0.021 0 0 0 0 1 -0.2 0 3.11 0 0];
  [-0.6 0 0 0 -0.07 0 0 -0.02 -0.004 0 0 0 1 0 -1.6 0 -0.06 0 0];
  [0 0 0 0 -0.07 0.21 -0.21 -0.02 0.011 0 0 0 1 0 -1.6 0 -0.06 0 0];
  [0 0 0 0 0.031 0 0 0.01 0.002 0.1 0.9 0 -1 0 0 0 -0.15 0 0];
  [0 0 0 0 0 0 0 1 -0.016 0 0 0 0 -1 0 0 -3.23 0 0]
  [0 0 1 0 0 0 0 -0.016 0 0 0 0 0 -1 0 -0.6 0 0];
  [-18 0 0 0 -4.24 0 4.17 -0.02 -0.6 0 0 0 0 0 0 1 0.9 0 0];
  [0 0 0 0 0.031 0 0 0.01 0.002 0.1 0.9 0 0 0 0 0 -1 -0.15 0 0];
  [0 0 0 0 0 0 0 -1 0.048 0 0 0 0 0 0 0 1.42 -3.45 4.87];
  [0 0 0 0 0 0 0 1 -0.048 0 0 0 0 0 0 0 -1.42 3.45 -4.87];]

C = DT*[CC zeros(21,76); zeros(21,19) CC zeros(21,57);
        zeros(21,38) CC zeros(21,38); zeros(21,57) CC zeros(21,19);
        zeros(21,76) CC;];


% ======================== SIMULATION DATA ========================
% Average values of the inflow concentrations based on the dry weather file
SO2i  = 0.0;     SFi   = 30;    SAi   = 20; SNH4i = 31.56; SNO3i = 0.0;
SPO4i = 3.6;     SIi   = 30;    SALKi = 7;  SN2i  = 6.95;  XIi   = 51.2;
XSi   = 202.32; XHi    = 28.17; XPAOi = 0.0; XPPi = 0.0;   XPHAi = 0.0;
XAUTi = 0.0;     XTSSi = 180;   XMeOHi = 0.0; XMePi = 0.0;

xi = [SO2i*ones(1,K1);  SFi*ones(1,K1);     SAi*ones(1,K1);    SNH4i*ones(1,K1);
      SNO3i*ones(1,K1); SPO4i*ones(1,K1);   SIi*ones(1,K1);    SALKi*ones(1,K1);
      SN2i*ones(1,K1);  XIi*ones(1,K1);     XSi*ones(1,K1);    XHi*ones(1,K1);
      XPAOi*ones(1,K1); XPPi*ones(1,K1);    XPHAi*ones(1,K1);  XAUTi*ones(1,K1);
      XTSSi*ones(1,K1); XMeOHi*ones(1,K1);  XMePi*ones(1,K1)]*0.01;


% Values of the initial conditions
% For tank 1
SO2_1  = 4.2984433e-03; SF_1   = 2.8082131e+00; SA_1   = 2.8082131e+00;
SNH4_1 = 7.9178845e+00; SNO3_1 = 5.3699400e+00; SPO4_1 = 3.6;
SI_1   = 3.0000000e+01; SALK_1 = 4.9277103e+00; SN2_1  = 1.2166405e+00;
XI_1   = 1.1491252e+03; XS_1   = 8.2134908e+01; XH_1   = 2.5517658e+03;
XPAO_1 = 0.1; XPP_1  = 0.1; XPHA_1 = 0.1; XAUT_1 = 1.4838943e+02;
XTSS_1 = 180; XMeOH_1 = 0.1; XMeP_1 = 0.1;
x10 = [SO2_1; SF_1; SA_1; SNH4_1; SNO3_1; SPO4_1; SI_1;   SALK_1; SN2_1;
       XI_1;  XS_1; XH_1; XPAO_1; XPP_1;  XPHA_1; XAUT_1; XTSS_1; XMeOH_1;
       XMeP_1];


% For tank 2
SO2_2  = 6.3131911e-05; SF_2   = 1.4587940e+00; SA_2   = 1.4587940e+00;
```

```
SNH4_2 = 8.3444148e+00; SNO3_2 = 3.6619672e+00; SPO4_2 = 3.6;
SI_2   = 3.0000000e+01; SALK_2 = 5.0801748e+00; SN2_2  = 8.8206477e-01;
XI_2   = 1.1491252e+03; XS_2   = 7.6386187e+01; XH_2   = 2.5533851e+03;
XPAO_2 = 0.1;XPP_2   = 0.1; XPHA_2 = 0.1; XAUT_2 = 1.4830914e+02;
XTSS_2 = 180;XMeOH_2 = 0.1; XMeP_2 = 0.1;
x20 = [SO2_2; SF_2; SA_2; SNH4_2; SNO3_2; SPO4_2; SI_2;   SALK_2; SN2_2;
       XI_2;  XS_2; XH_2; XPAO_2; XPP_2;  XPHA_2; XAUT_2; XTSS_2; XMeOH_2;
       XMeP_2];

% For tank 3
SO2_3  = 1.7183778e+00; SF_3   = 1.1495418e+00; SA_3   = 1.1495418e+00;
SNH4_3 = 5.5479452e+00; SNO3_3 = 6.5408820e+00; SPO4_3 = 3.6;
SI_3   = 3.0000000e+01; SALK_3 = 4.6747902e+00; SN2_3  = 8.2888682e-01;
XI_3   = 1.1491252e+03; XS_3 .. = 6.4854922e+01; XH_3   = 2.5571314e+03;
XPAO_3 = 0.1; XPP_3   = 0.1; XPHA_3  = 0.1; XAUT_3  = 1.4894126e+02;
XTSS_3 = 180; XMeOH_3 = 0.1; XMeP_3  = 0.1;
x30 = [SO2_3; SF_3; SA_3; SNH4_3; SNO3_3; SPO4_3; SI_3;   SALK_3; SN2_3;
       XI_3;  XS_3; XH_3; XPAO_3; XPP_3;  XPHA_3; XAUT_3; XTSS_3; XMeOH_3;
       XMeP_3];

% For tank 4
SO2_4   = 2.4288838e+00;SF_4    = 9.9532389e-01;SA_4    = 9.9532389e-01;
SNH4_4  = 2.9673854e+00;SNO3_4  = 9.2989988e+00;SPO4_4  = 3.6;
SI_4    = 3.0000000e+01;SALK_4  = 4.2934562e+00;SN2_4   = 7.6678656e-01;
XI_4    = 1.1491252e+03;XS_4    = 5.5693982e+01;XH_4    = 2.5591826e+03;
XPAO_4  = 0.1;XPP_4 = 0.1;XPHA_4     = 0.1;XAUT_4  = 1.4952712e+02;
XTSS_4  = 180;XMeOH_4 = 0.1;XMeP_4  = 0.1;
x40 = [SO2_4; SF_4; SA_4; SNH4_4; SNO3_4; SPO4_4; SI_4;   SALK_4; SN2_4;
       XI_4;  XS_4; XH_4; XPAO_4; XPP_4;  XPHA_4; XAUT_4; XTSS_4; XMeOH_4;
       XMeP_4];

% For tank 5
SO2_5   = 4.9094351e-01;SF_5    = 8.8949280e-01;SA_5    = 8.8949280e-01;
SNH4_5  = 1.7333316e+00;SNO3_5  = 1.0415220e+01;SPO4_5  = 3.6;
SI_5    = 3.0000000e+01;SALK_5  = 4.1255794e+00;SN2_5   = 6.8828001e-01;
XI_5    = 1.1491252e+03;XS_5    = 4.9305586e+01;XH_5    = 2.5593436e+03;
XPAO_5  = 0.1;XPP_5   = 0.1;XPHA_5 = 0.1;XAUT_5  = 1.4979714e+02;
XTSS_5  = 180;XMeOH_5 = 0.1;XMeP_5 = 0.1;
x50 = [SO2_5; SF_5; SA_5; SNH4_5; SNO3_5; SPO4_5; SI_5;   SALK_5; SN2_5;
       XI_5;  XS_5; XH_5; XPAO_5; XPP_5;  XPHA_5; XAUT_5; XTSS_5; XMeOH_5;
       XMeP_5];

x0 = [x10;x20;x30;x40;x50]*0.01;
x  = [x0 zeros(95,K1)];

% Steady state simulation of the model
for k = 1:K1
    xk  = x(:,k);
    xik = xi(:,k);
    Pk  = P_BASM2dF(xk);

    SOk  = [xk(39,1);xk(58,1);xk(77,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xx1k = [zeros(38,1);xxk(1);zeros(18,1);xxk(2);zeros(18,1);xxk(3);zeros(18,1)];

    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xx1k;
end

% Full model response to the steady state influent concentrations
% Tank 1
figure(1)
subplot(5,4,1);plot(x(1,:));title('SO2_1')
subplot(5,4,2);plot(x(2,:));title('SF1')
subplot(5,4,3);plot(x(3,:));title('SA1')
subplot(5,4,4);plot(x(4,:));title('SNH4_1')
subplot(5,4,5);plot(x(5,:));title('SNO3_1')
```

```
subplot(5,4,6);plot(x(6,:));title('SPO4_1')
subplot(5,4,7);plot(x(7,:));title('SI1')
subplot(5,4,8);plot(x(8,:));title('SALK1')
subplot(5,4,9);plot(x(9,:));title('SN2_1')
subplot(5,4,10);plot(x(10,:));title('XI1');
subplot(5,4,11);plot(x(11,:));title('XS1')
subplot(5,4,12);plot(x(12,:));title('XH1')
subplot(5,4,13);plot(x(13,:));title('XPAO1')
subplot(5,4,14);plot(x(14,:));title('XPP1')
subplot(5,4,15);plot(x(15,:));title('XPHA1')
subplot(5,4,16);plot(x(16,:));title('XAUT1')
xlabel('discrete time k')
subplot(5,4,17);plot(x(17,:));title('XTSS1')
xlabel('discrete time k')
subplot(5,4,18);plot(x(18,:));title('XMeOH1')
xlabel('discrete time k')
subplot(5,4,19);plot(x(19,:));title('XMeP1')
xlabel('discrete time k')
set(figure(1),'name','Tank 1 steady state results','numbertitle','off')

% Tank 2
figure(2)
subplot(5,4,1);plot(x(20,:));title('SC2_2')
subplot(5,4,2);plot(x(21,:));title('SF2')
subplot(5,4,3);plot(x(22,:));title('SA2')
subplot(5,4,4);plot(x(23,:));title('SNH4_2')
subplot(5,4,5);plot(x(24,:));title('SNO3_2')
subplot(5,4,6);plot(x(25,:));title('SPO4_2')
subplot(5,4,7);plot(x(26,:));title('SI2')
subplot(5,4,8);plot(x(27,:));title('SALK2')
subplot(5,4,9);plot(x(28,:));title('SN2_2')
subplot(5,4,10);plot(x(29,:));title('XI2');
subplot(5,4,11);plot(x(30,:));title('XS2')
subplot(5,4,12);plot(x(31,:));title('XH2')
subplot(5,4,13);plot(x(32,:));title('XPAO2')
subplot(5,4,14);plot(x(33,:));title('XPP2')
subplot(5,4,15);plot(x(34,:));title('XPHA2')
subplot(5,4,16);plot(x(35,:));title('XAUT2')
xlabel('discrete time k')
subplot(5,4,17);plot(x(36,:));title('XTSS2')
xlabel('discrete time k')
subplot(5,4,18);plot(x(37,:));title('XMeOH2')
xlabel('discrete time k')
subplot(5,4,19);plot(x(38,:));title('XMeP2')
xlabel('discrete time k')
set(figure(2),'name','Tank 2 steady state results','numbertitle','off')

% Tank 3
figure(3)
subplot(5,4,1);plot(x(39,:));title('SO2_3')
subplot(5,4,2);plot(x(40,:));title('SF3')
subplot(5,4,3);plot(x(41,:));title('SA3')
subplot(5,4,4);plot(x(42,:));title('SNH4_3')
subplot(5,4,5);plot(x(43,:));title('SNO3_3')
subplot(5,4,6);plot(x(44,:));title('SPO4_3')
subplot(5,4,7);plot(x(45,:));title('SI3')
subplot(5,4,8);plot(x(46,:));title('SALK3')
subplot(5,4,9);plot(x(47,:));title('SN2_3')
subplot(5,4,10);plot(x(48,:));title('XI3');
subplot(5,4,11);plot(x(49,:));title('XS3')
subplot(5,4,12);plot(x(50,:));title('XH3')
subplot(5,4,13);plot(x(51,:));title('XPAO3')
subplot(5,4,14);plot(x(52,:));title('XPP3')
subplot(5,4,15);plot(x(53,:));title('XPHA3')
subplot(5,4,16);plot(x(54,:));title('XAUT3')
xlabel('discrete time k')
```

```
subplot(5,4,17);plot(x(55,:));title('XTSS3')
xlabel('discrete time k')
subplot(5,4,18);plot(x(56,:));title('XMeOH3')
xlabel('discrete time k')
subplot(5,4,19);plot(x(57,:));title('XMeP3')
xlabel('discrete time k')
set(figure(3),'name','Tank 3 steady state results','numbertitle','off')


% Tank 4
figure(4)
subplot(5,4,1);plot(x(58,:));title('SO2_4')
subplot(5,4,2);plot(x(59,:));title('SF4')
subplot(5,4,3);plot(x(60,:));title('SA4')
subplot(5,4,4);plot(x(61,:));title('SNH4_4')
subplot(5,4,5);plot(x(62,:));title('SNO3_4')
subplot(5,4,6);plot(x(63,:));title('SPO4_4')
subplot(5,4,7);plot(x(64,:));title('SI4')
subplot(5,4,8);plot(x(65,:));title('SALK4')
subplot(5,4,9);plot(x(66,:));title('SN2_4')
subplot(5,4,10);plot(x(67,:));title('XI4');
subplot(5,4,11);plot(x(68,:));title('XS4')
subplot(5,4,12);plot(x(69,:));title('XH4')
subplot(5,4,13);plot(x(70,:));title('XPAO4')
subplot(5,4,14);plot(x(71,:));title('XPP4')
subplot(5,4,15);plot(x(72,:));title('XPHA4')
subplot(5,4,16);plot(x(73,:));title('XAUT4')
xlabel('discrete time k')
subplot(5,4,17);plot(x(74,:));title('XTSS4')
xlabel('discrete time k')
subplot(5,4,18);plot(x(75,:));title('XMeOH4')
xlabel('discrete time k')
subplot(5,4,19);plot(x(76,:));title('XMeP4')
xlabel('discrete time k')
set(figure(4),'name','Tank 4 steady state results','numbertitle','off')


% Tank 5
figure(5)
subplot(5,4,1);plot(x(77,:));title('SO2_5')
subplot(5,4,2);plot(x(78,:));title('SF5')
subplot(5,4,3);plot(x(79,:));title('SA5')
subplot(5,4,4);plot(x(80,:));title('SNH4_5')
subplot(5,4,5);plot(x(81,:));title('SNO3_5')
subplot(5,4,6);plot(x(82,:));title('SPO4_5')
subplot(5,4,7);plot(x(83,:));title('SI5')
subplot(5,4,8);plot(x(84,:));title('SALK5')
subplot(5,4,9);plot(x(85,:));title('SN2_5')
subplot(5,4,10);plot(x(86,:));title('XI5');
subplot(5,4,11);plot(x(87,:));title('XS5')
subplot(5,4,12);plot(x(88,:));title('XH5')
subplot(5,4,13);plot(x(89,:));title('XPAO5')
subplot(5,4,14);plot(x(90,:));title('XPP5')
subplot(5,4,15);plot(x(91,:));title('XPHA5')
subplot(5,4,16);plot(x(92,:));title('XAUT5')
xlabel('discrete time k')
subplot(5,4,17);plot(x(93,:));title('XTSS5')
xlabel('discrete time k')
subplot(5,4,18);plot(x(94,:));title('XMeOH5')
xlabel('discrete time k')
subplot(5,4,19);plot(x(95,:));title('XMeP5')
xlabel('discrete time k')
set(figure(5),'name','Tank 5 steady state results','numbertitle','off')


% Data for dynamic simulations
K    = 5;    % Prediction horizon for dynamic simulations
% Dynamics of the inflow concentrations for dry weather file
load dry.txt
```

229

```
% The order is t, SS XBH XS XI SNH SI SND XND QO
%                   1  2   3  4  5   6  7   8   9   10

xi = [SO2i*ones(1,K);   SFi*ones(1,K);    SAi*ones(1,K);    dry(1:K,6)';
      SNO3i*ones(1,K);  SPO4i*ones(1,K);  dry(1:K,7)';      SALKi*ones(1,K);
      SN2i*ones(1,K);   dry(1:K,5)';      dry(1:K,4)';      dry(1:K,3)';
      XPAOi*ones(1,K);  XPPi*ones(1,K);   XPHAi*ones(1,K);  XAUTi*ones(1,K);
      XTSSi*ones(1,K);  XMeOHi*ones(1,K); XMePi*ones(1,K)                  ];

% The order is
%                   1  2   3  4  5   6  7   8   9   10   11  12  13   14 15 16 17
%                   18 19
x = [x(:,(K1+1)) zeros(95,K)];

% Dynamic simulation of the model
for k = 1:K
    xk  = x(:,k);
    xik = xi(:,k);
    Pk  = P_BASM2dF(xk);

    SOk  = [xk(39,1);xk(58,1);xk(77,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xx1k = [zeros(38,1);xxk(1);zeros(18,1);xxk(2);zeros(18,1);xxk(3);zeros(18,1)];

    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xx1k;
end

% Full model response to the dynamic influent concentrations
% Tank 1
figure(6)
subplot(5,4,1);plot(x(1,:));title('SO2_1')
subplot(5,4,2);plot(x(2,:));title('SF1')
subplot(5,4,3);plot(x(3,:));title('SA1')
subplot(5,4,4);plot(x(4,:));title('SNH4_1')
subplot(5,4,5);plot(x(5,:));title('SNO3_1')
subplot(5,4,6);plot(x(6,:));title('SPO4_1')
subplot(5,4,7);plot(x(7,:));title('SI1')
subplot(5,4,8);plot(x(8,:));title('SALK1')
subplot(5,4,9);plot(x(9,:));title('SN2_1')
subplot(5,4,10);plot(x(10,:));title('XI1');
subplot(5,4,11);plot(x(11,:));title('XS1')
subplot(5,4,12);plot(x(12,:));title('XH1')
subplot(5,4,13);plot(x(13,:));title('XPAO1')
subplot(5,4,14);plot(x(14,:));title('XPP1')
subplot(5,4,15);plot(x(15,:));title('XPHA1')
subplot(5,4,16);plot(x(16,:));title('XAUT1')
xlabel('discrete time k')
subplot(5,4,17);plot(x(17,:));title('XTSS1')
xlabel('discrete time k')
subplot(5,4,18);plot(x(18,:));title('XMeOH1')
xlabel('discrete time k')
subplot(5,4,19);plot(x(19,:));title('XMeP1')
xlabel('discrete time k')
set(figure(6),'name','Tank 1 dynamic results','numbertitle','off')


% Tank 2
figure(7)
subplot(5,4,1);plot(x(20,:));title('SO2_2')
subplot(5,4,2);plot(x(21,:));title('SF2')
subplot(5,4,3);plot(x(22,:));title('SA2')
subplot(5,4,4);plot(x(23,:));title('SNH4_2')
subplot(5,4,5);plot(x(24,:));title('SNO3_2')
subplot(5,4,6);plot(x(25,:));title('SPO4_2')
subplot(5,4,7);plot(x(26,:));title('SI2')
subplot(5,4,8);plot(x(27,:));title('SALK2')
```

```
subplot(5,4,9);plot(x(28,:));title('SN2_2')
subplot(5,4,10);plot(x(29,:));title('XI2');
subplot(5,4,11);plot(x(30,:));title('XS2')
subplot(5,4,12);plot(x(31,:));title('XH2')
subplot(5,4,13);plot(x(32,:));title('XPAO2')
subplot(5,4,14);plot(x(33,:));title('XPP2')
subplot(5,4,15);plot(x(34,:));title('XPHA2')
subplot(5,4,16);plot(x(35,:));title('XAUT2')
xlabel('discrete time k')
subplot(5,4,17);plot(x(36,:));title('XTSS2')
xlabel('discrete time k')
subplot(5,4,18);plot(x(37,:));title('XMeOH2')
xlabel('discrete time k')
subplot(5,4,19);plot(x(38,:));title('XMeP2')
xlabel('discrete time k')
set(figure(7),'name','Tank 2 dynamic results','numbertitle','off')

% Tank 3
figure(8)
subplot(5,4,1);plot(x(39,:));title('SO2_3')
subplot(5,4,2);plot(x(40,:));title('SF3')
subplot(5,4,3);plot(x(41,:));title('SA3')
subplot(5,4,4);plot(x(42,:));title('SNH4_3')
subplot(5,4,5);plot(x(43,:));title('SNO3_3')
subplot(5,4,6);plot(x(44,:));title('SPO4_3')
subplot(5,4,7);plot(x(45,:));title('SI3')
subplot(5,4,8);plot(x(46,:));title('SALK3')
subplot(5,4,9);plot(x(47,:));title('SN2_3')
subplot(5,4,10);plot(x(48,:));title('XI3');
subplot(5,4,11);plot(x(49,:));title('XS3')
subplot(5,4,12);plot(x(50,:));title('XH3')
subplot(5,4,13);plot(x(51,:));title('XPAO3')
subplot(5,4,14);plot(x(52,:));title('XPP3')
subplot(5,4,15);plot(x(53,:));title('XPHA3')
subplot(5,4,16);plot(x(54,:));title('XAUT3')
xlabel('discrete time k')
subplot(5,4,17);plot(x(55,:));title('XTSS3')
xlabel('discrete time k')
subplot(5,4,18);plot(x(56,:));title('XMeOH3')
xlabel('discrete time k')
subplot(5,4,19);plot(x(57,:));title('XMeP3')
xlabel('discrete time k')
set(figure(8),'name','Tank 3 dynamic results','numbertitle','off')

% Tank 4
figure(9)
subplot(5,4,1);plot(x(58,:));title('SO2_4')
subplot(5,4,2);plot(x(59,:));title('SF4')
subplot(5,4,3);plot(x(60,:));title('SA4')
subplot(5,4,4);plot(x(61,:));title('SNH4_4')
subplot(5,4,5);plot(x(62,:));title('SNO3_4')
subplot(5,4,6);plot(x(63,:));title('SPO4_4')
subplot(5,4,7);plot(x(64,:));title('SI4')
subplot(5,4,8);plot(x(65,:));title('SALK4')
subplot(5,4,9);plot(x(66,:));title('SN2_4')
subplot(5,4,10);plot(x(67,:));title('XI4');
subplot(5,4,11);plot(x(68,:));title('XS4')
subplot(5,4,12);plot(x(69,:));title('XH4')
subplot(5,4,13);plot(x(70,:));title('XPAO4')
subplot(5,4,14);plot(x(71,:));title('XPP4')
subplot(5,4,15);plot(x(72,:));title('XPHA4')
subplot(5,4,16);plot(x(73,:));title('XAUT4')
xlabel('discrete time k')
subplot(5,4,17);plot(x(74,:));title('XTSS4')
xlabel('discrete time k')
subplot(5,4,18);plot(x(75,:));title('XMeOH4')
```

231

```
xlabel('discrete time k')
subplot(5,4,19);plot(x(76,:));title('XMeP4')
xlabel('discrete time k')
set(figure(9),'name','Tank 4 dynamic results','numbertitle','off')


% Tank 5
figure(10)
subplot(5,4,1);plot(x(77,:));title('SO2_5')
subplot(5,4,2);plot(x(78,:));title('SF5')
subplot(5,4,3);plot(x(79,:));title('SA5')
subplot(5,4,4);plot(x(80,:));title('SNH4_5')
subplot(5,4,5);plot(x(81,:));title('SNO3_5')
subplot(5,4,6);plot(x(82,:));title('SPO4_5')
subplot(5,4,7);plot(x(83,:));title('SI5')
subplot(5,4,8);plot(x(84,:));title('SALK5')
subplot(5,4,9);plot(x(85,:));title('SN2_5')
subplot(5,4,10);plot(x(86,:));title('XI5');
subplot(5,4,11);plot(x(87,:));title('XS5')
subplot(5,4,12);plot(x(88,:));title('XH5')
subplot(5,4,13);plot(x(89,:));title('XPAO5')
subplot(5,4,14);plot(x(90,:));title('XPP5')
subplot(5,4,15);plot(x(91,:));title('XPHA5')
subplot(5,4,16);plot(x(92,:));title('XAUT5')
xlabel('discrete time k')
subplot(5,4,17);plot(x(93,:));title('XTSS5')
xlabel('discrete time k')
subplot(5,4,18);plot(x(94,:));title('XMeOH5')
xlabel('discrete time k')
subplot(5,4,19);plot(x(95,:));title('XMeP5')
xlabel('discrete time k')
set(figure(10),'name','Tank 5 dynamic results','numbertitle','off')


% ====================== REDUCED MODEL FORMULATION ======================
% Form the vector of the states (z) for the reduced model, based on the
% ASM1 prediction
%      SF,       SA,      SNH4,     SNO3,     SPO4,      XS,       XPP,      XPHA
z1 = [x(2,:); x(3,:); x(4,:); x(5,:);  x(6,:);  x(11,:); x(14,:); x(15,:)];
z2 = [x(21,:); x(22,:);x(23,:);x(24,:);x(25,:); x(30,:); x(33,:); x(34,:)];
z3 = [x(40,:); x(41,:);x(42,:);x(43,:);x(44,:); x(49,:); x(52,:); x(53,:)];
z4 = [x(59,:); x(60,:);x(61,:);x(62,:);x(63,:); x(68,:); x(71,:); x(72,:)];
z5 = [x(78,:); x(79,:);x(80,:);x(81,:);x(82,:); x(87,:); x(90,:); x(91,:)];
z  = [z1;z2;z3;z4;z5];

% Formation of the vector of the control variables
u1 = x(1,:);
u2 = x(20,:);
u3 = x(39,:);
u4 = x(58,:);
u5 = x(77,:);
u  = [u1; u2; u3; u4; u5];

% Dynamic values of the inflow concentrations for dry weather
zi = [SFi*ones(1,K);   SAi*ones(1,K); SNH4i*ones(1,K); SNO3i*ones(1,K);
      SPO4i*ones(1,K); XSi*ones(1,K); XPPi*ones(1,K);  XPHAi*ones(1,K)]*0.001;

% Steady state values for the biomass XH, XA, SALK, XPP, XPAO, XPHA
XH   = [2551.76*ones(1,K); 2553.38*ones(1,K); 2557.13*ones(1,K);
        2559.18*ones(1,K); 2559.34*ones(1,K)];
XA   = [148.389*ones(1,K); 148.309*ones(1,K); 148.941*ones(1,K);
        149.527*ones(1,K); 149.797*ones(1,K)];
SALK = [4.9*ones(1,K); 148.309*ones(1,K); 148.941*ones(1,K);
        149.527*ones(1,K); 149.797*ones(1,K)];
XPAO = [0.1*ones(1,K); 76.386*ones(1,K); 64.855*ones(1,K);
        55.694*ones(1,K); 49.306*ones(1,K)];
```

```matlab
% MATRIX A for the reduced model (AR)
A11r = 1 - IV11; A22r = 1 - IV22; A33r = 1 - IV33;
A44r = 1 - IV44; A55r = 1 - IV55;
A15ar = IV1 * (Qa + Qr);
A15br = IV1 * (Qa + LAMBDA*Qr);
A21r = IV22; A32r = IV33; A43r = IV44; A54r = IV55;


A11r = A11r * ones(1,8); A11r = diag(A11r);
A22r = A22r * ones(1,8); A22r = diag(A22r);
A33r = A33r * ones(1,8); A33r = diag(A33r);
A44r = A44r * ones(1,8); A44r = diag(A44r);
A55r = A55r * ones(1,8); A55r = diag(A55r);


A15r = [A15ar, A15ar, A15ar, A15ar, A15ar, A15br, A15br, A15br];
A15r = diag(A15r);


A21r = A21r * ones(1,8); A21r = diag(A21r);
A32r = A32r * ones(1,8); A32r = diag(A32r);
A43r = A43r * ones(1,8); A43r = diag(A43r);
A54r = A54r * ones(1,8); A54r = diag(A54r);


A10r = zeros(8,8); A20r = zeros(8,16); A30r = zeros(8,24);

AR = [A11r A30r A15r;A21r A22r A30r;A10r A32r A33r A20r;A20r A43r A44r A10r;A30r A54r
A55r];


% MATRIX B for the reduced model (BR)
IV0 = DT / V1*Q0;
f = 1.2;
B11r = [IV0*ones(1,3) IV0*f IV0*ones(1,4)];
B11r = diag(B11r);
BR = [B11r; zeros(32,8)];

% MATRIX C for the reduced model (CR)
%      SF     SA     SNH4    SNO3    SPC4       XS       XPP    XPHA
% MATRIX C for the reduced model (CR)
%      SF     SA     SNH4    SNO3    SPC4       XS       XPP    XPHA
 CCR=[[1     0     0.01    0        0         -1   0     0];
      [1     0     0.01    0        0         -1   0     0];
      [1     0     0.01    0        0         -1   0     0];
      [-1.6  0     -0.022  0       -0.004      0   0     0];
      [0     -1.6  -0.07   0       -0.02       0   0     0];
      [-1.6  0     -0.022  -0.21   -0.004      0   0     0];
      [0     -1.6  -0.07   -0.21   -0.02       0   0     0];
      [-1    1     0.03    0        0.01       0   0     0];
      [0     -1    0       0        0.4        0   -0.4  1];
      [0     0     0       0       -1          0   1     -0.2];
      [0     0     0       -0.07   -1          0   1     -0.2];
      [0     0     -0.07   0       -0.02       0   0     -1.6];
      [0     0     -0.07   -0.21   -0.02       0   0     -1.6];
      [0     0     -4.24   4.17    -0.02       0   0     0]];

CR = DT*[CCR zeros(14,32); zeros(14,8) CCR zeros(14,24);
         zeros(14,16) CCR zeros(14,16); zeros(14,24) CCR zeros(14,8);
         zeros(14,32) CCR];

% Parameter estimation
for k = 1:K
    zk = z(:,k);
    uk = u(:,k);

    XHk = XH(:,k);
    XAk = XA(:,k);
    SALKk = SALK(:,k);
    XPAOk = XPAO(:,k);
```

```
    zik  = zi(:,k);
    Pk = P_BASM2dR(zk,uk,XHk,XAk,SALKk,XPAOk);
    FIk = CR'*Pk;
    Y(:,k) = z(:,k+1)-AR*z(:,k)-BR*zi(:,k);
    theta(:,k)=(FIk'*FIk)^(-1)*FIk'*Y(:,k);
end

kHe(1,:)     = theta(1,:);
muHe(1,:)    = theta(2,:);
qfee(1,:)    = theta(3,:);
qPHAe(1,:)   = theta(4,:);
qPPe(1,:)    = theta(5,:);
muPAOe(1,:)  = theta(6,:);
muAUTe(1,:)  = theta(7,:);

% Dynamic simulation of the reduced model with the estimated parameters
zE = [x(2,1);x(3,1);x(4,1);x(5,1);x(6,1);x(11,1);x(14,1);x(15,1);x(21,1);x(22,1);
x(23,1); x(24,1);x(25,1); x(30,1);
x(33,1);x(34,1);x(40,1);x(41,1);x(42,1);x(43,1);x(44,1);x(45,1);x(52,1);x(53,1);x(59,1)
; x(60,1); x(61,1);x(62,1); x(63,1);
x(68,1);x(71,1);x(72,1);x(78,1);x(79,1);x(80,1);x(81,1);x(82,1);x(87,1);x(90,1);x(91,1)
];
ERR = z(:,1)-zE;

% Dynamic simulation of the reduced model with the estimated parameters
for k = 1:K
    zk = z(:,k);
    uk = u(:,k);
    zik = zi(:,k);

    XHk = XH(:,k);
    XAk = XA(:,k);
    SALKk = SALK(:,k);
    XPAOk = XPAO(:,k);

    kHek     = kHe(:,k);
    muHek    = muHe(:,k);
    qfeek    = qfee(:,k);
    qPHAek   = qPHAe(:,k);
    qPPek    = qPPe(:,k);
    muPAOek  = muPAOe(:,k);
    muAUTek  = muAUTe(:,k);

    Pk =
P_BASM2dRE(zk,uk,XHk,XAk,SALKk,XPAOk,kHek,muHek,qfeek,qPHAek,qPPek,muPAOek,muAUTek);
    zE(:,k+1) = AR*zE(:,k) + CR'*Pk + BR*zi(:,k);
    ERR(:,k+1)=z(:,k+1)-zE(:,k+1);
end

% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 1
SF1=z(1,:);SA1=z(2,:);SNH4_1=z(3,:);SNO3_1=z(4,:);
SF1E=zE(1,:);SA1E=zE(2,:);SNH4_1E=zE(3,:);SNO3_1E=zE(4,:);
ERRSF1=ERR(1,:);ERRSA1=ERR(2,:);ERRSNH4_1=ERR(3,:);ERRSNO3_1=ERR(4,:);

SPO4_1=z(5,:);XS1=z(6,:);XPP1=z(7,:);XPHA1=z(8,:);
SPO4_1E=zE(5,:);XS1E=zE(6,:);XPP1E=zE(7,:);XPHA1E=zE(8,:);
ERRSPO4_1=ERR(5,:);ERRXS1=ERR(6,:);ERRXPP1=ERR(7,:);ERRXPHA1=ERR(8,:);

% Calculation of the weighted errors for tank 1
eSF1=(ERRSF1*ERRSF1')/(SF1*SF1');
eSA1=(ERRSA1*ERRSA1')/(SA1*SA1');
eSNH4_1=(ERRSNH4_1*ERRSNH4_1')/(SNH4_1*SNH4_1');
eSNO3_1=(ERRSNO3_1*ERRSNO3_1')/(SNO3_1*SNO3_1');
eSPO4_1=(ERRSPO4_1*ERRSPO4_1')/(SPO4_1*SPO4_1');
eXS1=(ERRXS1*ERRXS1')/(XS1*XS1');
```

```
eXPP1=(ERRXPP1*ERRXPP1')/(XPP1*XPP1');
eXPHA1=(ERRXPHA1*ERRXPHA1')/(XPHA1*XPHA1');

% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 2
SF2=z(9,:);SA2=z(10,:);SNH4_2=z(11,:);SNO3_2=z(12,:);
SF2E=zE(9,:);SA2E=zE(10,:);SNH4_2E=zE(11,:);SNO3_2E=zE(12,:);
ERRSF2=ERR(9,:);ERRSA2=ERR(10,:);ERRSNH4_2=ERR(11,:);ERRSNO3_2=ERR(12,:);


SPO4_2=z(13,:);XS2=z(14,:);XPP2=z(15,:);XPHA2=z(16,:);
SPO4_2E=zE(13,:);XS2E=zE(14,:);XPP2E=zE(15,:);XPHA2E=zE(16,:);
ERRSPO4_2=ERR(13,:);ERRXS2=ERR(14,:);ERRXPP2=ERR(15,:);ERRXPHA2=ERR(16,:);


% Calculation of the weighted errors for tank 2
eSF2=(ERRSF2*ERRSF2')/(SF2*SF2');
eSA2=(ERRSA2*ERRSA2')/(SA2*SA2');
eSNH4_2=(ERRSNH4_2*ERRSNH4_2')/(SNH4_2*SNH4_2');
eSNO3_2=(ERRSNO3_2*ERRSNO3_2')/(SNO3_2*SNO3_2');
eSPO4_2=(ERRSPO4_2*ERRSPO4_2')/(SPO4_2*SPO4_2');
eXS2=(ERRXS2*ERRXS2')/(XS2*XS2');
eXPP2=(ERRXPP2*ERRXPP2')/(XPP2*XPP2');
eXPHA2=(ERRXPHA2*ERRXPHA2')/(XPHA2*XPHA2');

% Vector of process full, estimated and error variables
% SF,SNH4,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 3
SF3=z(17,:);SA3=z(18,:);SNH4_3=z(19,:);SNO3_3=z(20,:);
SF3E=zE(17,:);SA3E=zE(18,:);SNH4_3E=zE(19,:);SNO3_3E=zE(20,:);
ERRSF3=ERR(17,:);ERRSA3=ERR(18,:);ERRSNH4_3=ERR(19,:);ERRSNO3_3=ERR(20,:);


SPO4_3=z(21,:);XS3=z(22,:);XPP3=z(23,:);XPHA3=z(24,:);
SPO4_3E=zE(21,:);XS3E=zE(22,:);XPP3E=zE(23,:);XPHA3E=zE(24,:);
ERRSPO4_3=ERR(21,:);ERRXS3=ERR(22,:);ERRXPP3=ERR(23,:);ERRXPHA3=ERR(24,:);


% Calculation of the weighted errors for tank 3
eSF3=(ERRSF3*ERRSF3')/(SF3*SF3');
eSA3=(ERRSA3*ERRSA3')/(SA3*SA3');
eSNH4_3=(ERRSNH4_3*ERRSNH4_3')/(SNH4_3*SNH4_3');
eSNO3_3=(ERRSNO3_3*ERRSNO3_3')/(SNO3_3*SNO3_3');
eSPO4_3=(ERRSPO4_3*ERRSPO4_3')/(SPO4_3*SPO4_3');
eXS3=(ERRXS3*ERRXS3')/(XS3*XS3');
eXPP3=(ERRXPP3*ERRXPP3')/(XPP3*XPP3');
eXPHA3=(ERRXPHA3*ERRXPHA3')/(XPHA3*XPHA3');

% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 4
SF4=z(9,:);SA4=z(10,:);SNH4_4=z(11,:);SNO3_4=z(12,:);
SF4E=zE(9,:);SA4E=zE(10,:);SNH4_4E=zE(11,:);SNO3_4E=zE(12,:);
ERRSF4=ERR(9,:);ERRSA4=ERR(10,:);ERRSNH4_4=ERR(11,:);ERRSNO3_4=ERR(12,:);


SPO4_4=z(13,:);XS4=z(14,:);XPP4=z(15,:);XPHA4=z(16,:);
SPO4_4E=zE(13,:);XS4E=zE(14,:);XPP4E=zE(15,:);XPHA4E=zE(16,:);
ERRSPO4_4=ERR(13,:);ERRXS4=ERR(14,:);ERRXPP4=ERR(15,:);ERRXPHA4=ERR(16,:);


% Calculation of the weighted errors for tank 4
eSF4=(ERRSF4*ERRSF4')/(SF4*SF4');
eSA4=(ERRSA4*ERRSA4')/(SA4*SA4');
eSNH4_4=(ERRSNH4_4*ERRSNH4_4')/(SNH4_4*SNH4_4');
eSNO3_4=(ERRSNO3_4*ERRSNO3_4')/(SNO3_4*SNO3_4');
eSPO4_4=(ERRSPO4_4*ERRSPO4_4')/(SPO4_4*SPO4_4');
eXS4=(ERRXS4*ERRXS4')/(XS4*XS4');
eXPP4=(ERRXPP4*ERRXPP4')/(XPP4*XPP4');
eXPHA4=(ERRXPHA4*ERRXPHA4')/(XPHA4*XPHA4');

% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 3
SF5=z(17,:);SA5=z(18,:);SNH4_5=z(19,:);SNO3_5=z(20,:);
```

```
SF5E=zE(17,:);SA5E=zE(18,:);SNH4_5E=zE(19,:);SNO3_5E=zE(20,:);
ERRSF5=ERR(17,:);ERRSA5=ERR(18,:);ERRSNH4_5=ERR(19,:);ERRSNO3_5=ERR(20,:);

SPO4_5=z(21,:);XS5=z(22,:);XPP5=z(23,:);XPHA5=z(24,:);
SPO4_5E=zE(21,:);XS5E=zE(22,:);XPP5E=zE(23,:);XPHA5E=zE(24,:);
ERRSPO4_5=ERR(21,:);ERRXS5=ERR(22,:);ERRXPP5=ERR(23,:);ERRXPHA5=ERR(24,:);

% Calculation of the weighted errors for tank 5
eSF5=(ERRSF5*ERRSF5')/(SF5*SF5');
eSA5=(ERRSA5*ERRSA5')/(SA5*SA5');
eSNH4_5=(ERRSNH4_5*ERRSNH4_5')/(SNH4_5*SNH4_5');
eSNO3_5=(ERRSNO3_5*ERRSNO3_5')/(SNO3_5*SNO3_5');
eSPO4_5=(ERRSPO4_5*ERRSPO4_5')/(SPO4_5*SPO4_5');
eXS5=(ERRXS5*ERRXS5')/(XS5*XS5');
eXPP5=(ERRXPP5*ERRXPP5')/(XPP5*XPP5');
eXPHA5=(ERRXPHA5*ERRXPHA5')/(XPHA5*XPHA5');

k = 1:K;
% Graphs of the estimated parameters
figure(11)
subplot(4,2,1);plot(k,kHe,'k');title('Estimated kH');ylabel('kHe');
subplot(4,2,2);plot(k,muHe,'k');title('Estimated muH');ylabel('muHe');
subplot(4,2,3);plot(k,qfee,'k');title('Estimated qfe');ylabel('qfee');
subplot(4,2,4);plot(k,qPHAe,'k');title('Estimated qPHA');ylabel('qPHAe');
subplot(4,2,5);plot(k,qPPe,'k');title('Estimated qPP');ylabel('qPPe');
subplot(4,2,6);plot(k,muPAOe,'k');title('Estimated muPAO');
ylabel('muPAOe');xlabel('discrete time k')
subplot(4,2,7);plot(k,muAUTe,'k');title('Estimated muAUT');
ylabel('muAUTe');xlabel('discrete time k')
set(figure(11),'name','Estimated parameters','numbertitle','off')

k = 1:K+1;
% Graphs of the reduced model response to the dynamic influent
% concentrations

figure(12)
subplot(4,4,1);plot(k,SF1);hold on; plot(k,SF1E,'k-.');ylabel('SF1,SF1E');
legend('SF1','SF1E');subplot(4,4,2); plot(k,ERRSF1,'k');ylabel('ERRSF1');

subplot(4,4,3);plot(k,SA1);hold on; plot(k,SA1E,'k-.');ylabel('SA1,SA1E');
legend('SA1','SA1E');subplot(4,4,4); plot(k,ERRSA1,'k');ylabel('ERRSA1');

subplot(4,4,5);plot(k,SNH4_1);hold on; plot(k,SNH4_1E,'k-.');
legend('SNH4_1','SNH4_1E');ylabel('SNH4_1,SNH4_1E');subplot(4,4,6);
plot(k,ERRSNH4_1,'k');ylabel('ERRSNH4_1');

subplot(4,4,7);plot(k,SNO3_1);hold on; plot(k,SNO3_1E,'k-.')
legend('SNO3_1','SNO3_1E');ylabel('SNO3_1,SNO3_1E');subplot(4,4,8);
plot(k,ERRSNO3_1,'k');ylabel('ERRSNO3_1');

subplot(4,4,9);plot(k,SPO4_1);hold on; plot(k,SPO4_1E,'k-.');
legend('SPO4_1','SPO4_1E');ylabel('SPO4_1,SPO4_1E');subplot(4,4,10);
plot(k,ERRSPO4_1,'k');ylabel('ERRSPO4_1');

subplot(4,4,11);plot(k,XS1);hold on; plot(k,XS1E,'k-.');ylabel('XS1,XS1E');
legend('XS1','XS1E');subplot(4,4,12); plot(k,ERRXS1,'k');ylabel('ERRXS1');

subplot(4,4,13);plot(k,XPP1);hold on; plot(k,XPP1E,'k-.')
legend('XPP1','XPP1E');ylabel('XPP1,XPP1E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP1,'k');
ylabel('ERRXPP1');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA1);hold on; plot(k,XPHA1E,'k-.')
legend('XPHA1','XPHA1E');ylabel('XPHA1,XPHA1E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA1,'k');
ylabel('ERRXPHA1');xlabel('discrete time k')
```

```
set(figure(12),'name','Tank 1 Process variables and estimation
error','numbertitle','off')

figure(13)
subplot(4,4,1);plot(k,SF2);hold on; plot(k,SF2E,'k-.');ylabel('SF2,SF2E');
legend('SF2','SF2E');subplot(4,4,2); plot(k,ERRSF2,'k');ylabel('ERRSF2');

subplot(4,4,3);plot(k,SA2);hold on; plot(k,SA2E,'k-.');ylabel('SA2,SA2E');
legend('SA2','SA2E');subplot(4,4,4); plot(k,ERRSA2,'k');ylabel('ERRSA2');

subplot(4,4,5);plot(k,SNH4_2);hold on; plot(k,SNH4_2E,'k-.');
legend('SNH4_2','SNH4_2E');ylabel('SNH4_2,SNH4_2E');subplot(4,4,6);
plot(k,ERRSNH4_2,'k');ylabel('ERRSNH4_2');

subplot(4,4,7);plot(k,SNO3_2);hold on; plot(k,SNO3_2E,'k-.')
legend('SNO3_2','SNO3_2E');ylabel('SNO3_2,SNO3_2E');subplot(4,4,8);
plot(k,ERRSNO3_2,'k');ylabel('ERRSNO3_2');

subplot(4,4,9);plot(k,SPO4_2);hold on; plot(k,SPO4_2E,'k-.')
legend('SPO4_2','SPO4_2E');ylabel('SPO4_2,SPO4_2E');subplot(4,4,10);
plot(k,ERRSPO4_2,'k');ylabel('ERRSPO4_2');

subplot(4,4,11);plot(k,XS2);hold on; plot(k,XS2E,'k-.');ylabel('XS2,XS2E');
legend('XS2','XS2E');subplot(4,4,12); plot(k,ERRXS2,'k');ylabel('ERRXS2');

subplot(4,4,13);plot(k,XPP2);hold on; plot(k,XPP2E,'k-.');
legend('XPP2','XPP2E');ylabel('XPP2,XPP2E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP2,'k');
ylabel('ERRXPP2');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA2);hold on; plot(k,XPHA2E,'k-.')
legend('XPHA2','XPHA2E');ylabel('XPHA2,XPHA2E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA2,'k');
ylabel('ERRXPHA2');xlabel('discrete time k')
set(figure(13),'name','Tank 2 Process variables and estimation
error','numbertitle','off')

figure(14)
subplot(4,4,1);plot(k,SF3);hold on; plot(k,SF3E,'k-.');ylabel('SF3,SF3E');
legend('SF3','SF3E');subplot(4,4,2); plot(k,ERRSF3,'k');ylabel('ERRSF3');

subplot(4,4,3);plot(k,SA3);hold on; plot(k,SA3E,'k-.');ylabel('SA3,SA3E');
legend('SA3','SA3E');subplot(4,4,4); plot(k,ERRSA3,'k');

subplot(4,4,5);plot(k,SNH4_3);hold on; plot(k,SNH4_3E,'k-.')
legend('SNH4_3','SNH4_3E');ylabel('SNH4_3,SNH4_3E');subplot(4,4,6);
plot(k,ERRSNH4_3,'k');ylabel('ERRSNH4_3');

subplot(4,4,7);plot(k,SNO3_3);hold on; plot(k,SNO3_3E,'k-.')
legend('SNO3_3','SNO3_3E');ylabel('SNO3_3,SNO3_3E');subplot(4,4,8);
plot(k,ERRSNO3_3,'k');ylabel('ERRSNO3_3');

subplot(4,4,9);plot(k,SPO4_3);hold on; plot(k,SPO4_3E,'k-.')
legend('SPO4_3','SPO4_3E');ylabel('SPO4_3,SPO4_3E');subplot(4,4,10);
plot(k,ERRSPO4_3,'k');ylabel('ERRSPO4_3');

subplot(4,4,11);plot(k,XS3);hold on; plot(k,XS3E,'k-.');ylabel('XS3,XS3E');
legend('XS3','XS3E');subplot(4,4,12); plot(k,ERRXS3,'k');ylabel('ERRXS3');

subplot(4,4,13);plot(k,XPP3);hold on; plot(k,XPP3E,'k-.')
legend('XPP3','XPP3E');ylabel('XPP3,XPP3E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP3,'k');
ylabel('ERRXPP3');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA3);hold on; plot(k,XPHA3E,'k-.')
legend('XPHA3','XPHA3E');ylabel('XPHA3,XPHA3E');xlabel('discrete time k')
```

```
subplot(4,4,16); plot(k,ERRXPHA3,'k');
ylabel('ERRXPHA3');xlabel('discrete time k')
set(figure(14),'name','Tank 3 Process variables and estimation
error','numbertitle','off')

figure(15)
subplot(4,4,1);plot(k,SF4);hold on; plot(k,SF4E,'k-.');ylabel('SF4,SF4E');
legend('SF4','SF4E');subplot(4,4,2); plot(k,ERRSF4,'k');ylabel('ERRSF4');

subplot(4,4,3);plot(k,SA4);hold on; plot(k,SA4E,'k-.');ylabel('SA4,SA4E');
legend('SA4','SA4E');subplot(4,4,4); plot(k,ERRSA4,'k');ylabel('ERRSA4');

subplot(4,4,5);plot(k,SNH4_4);hold on; plot(k,SNH4_4E,'k-.')
legend('SNH4_4','SNH4_4E');ylabel('SNH4_4,SNH4_4E');subplot(4,4,6);
plot(k,ERRSNH4_4,'k');ylabel('ERRSNH4_4');

subplot(4,4,7);plot(k,SNO3_4);hold on; plot(k,SNO3_4E,'k-.')
legend('SNO3_4','SNO3_4E');ylabel('SNO3_4,SNO3_4E');subplot(4,4,8);
plot(k,ERRSNO3_4,'k');ylabel('ERRSNO3_4');

subplot(4,4,9);plot(k,SPO4_4);hold on; plot(k,SPO4_4E,'k-.')
legend('SPO4_4','SPO4_4E');ylabel('SPO4_4,SPO4_4E');subplot(4,4,10);
plot(k,ERRSPO4_4,'k');ylabel('ERRSPO4_4');

subplot(4,4,11);plot(k,XS4);hold on; plot(k,XS4E,'k-.');ylabel('XS4,XS4E');
legend('XS4','XS4E');subplot(4,4,12); plot(k,ERRXS4,'k');ylabel('ERRXS4');

subplot(4,4,13);plot(k,XPP4);hold on; plot(k,XPP4E,'k-.')
legend('XPP4','XPP4E');ylabel('XPP4,XPP4E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP4,'k');
ylabel('ERRXPP4');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA4);hold on; plot(k,XPHA4E,'k-.')
legend('XPHA4','XPHA4E');ylabel('XPHA4,XPHA4E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA4,'k');
ylabel('ERRXPHA4');xlabel('discrete time k')
set(figure(15),'name','Tank 4 Process variables and estimation
error','numbertitle','off')

figure(16)
subplot(4,4,1);plot(k,SF5);hold on; plot(k,SF5E,'k-.');ylabel('SF5,SF5E');
legend('SF5','SF5E');subplot(4,4,2); plot(k,ERRSF5,'k');ylabel('ERRSF5');

subplot(4,4,3);plot(k,SA5);hold on; plot(k,SA5E,'k-.');ylabel('SA5,SA5E');
legend('SA5','SA5E');subplot(4,4,4); plot(k,ERRSA5,'k');ylabel('ERRSA5');

subplot(4,4,5);plot(k,SNH4_5);hold on; plot(k,SNH4_5E,'k-.')
legend('SNH4_5','SNH4_5E');ylabel('SNH4_5,SNH4_5E');subplot(4,4,6);
plot(k,ERRSNH4_5,'k');ylabel('ERRSNH4_5');

subplot(4,4,7);plot(k,SNO3_5);hold on; plot(k,SNO3_5E,'k-.')
legend('SNO3_5','SNO3_5E');ylabel('SNO3_5,SNO3_5E');subplot(4,4,8);
plot(k,ERRSNO3_5,'k');ylabel('ERRSNO3_5');

subplot(4,4,9);plot(k,SPO4_5);hold on; plot(k,SPO4_5E,'k-.')
legend('SPO4_5','SPO4_5E');ylabel('SPO4_5,SPO4_5E');subplot(4,4,10);
plot(k,ERRSPO4_5,'k');ylabel('ERRSPO4_5');

subplot(4,4,11);plot(k,XS5);hold on; plot(k,XS5E,'k-.');ylabel('XS5,XS5E');
legend('XS5','XS5E');subplot(4,4,12); plot(k,ERRXS5,'k');ylabel('ERRXS5');

subplot(4,4,13);plot(k,XPP5);hold on; plot(k,XPP5E,'k-.')
legend('XPP5','XPP5E');ylabel('XPP5,XPP5E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP5,'k');
ylabel('ERRXPP5');xlabel('discrete time k')
```

238

```
subplot(4,4,15);plot(k,XPHA5);hold on; plot(k,XPHA5E,'k-.')
legend('XPHA5','XPHA5E');ylabel('XPHA5,XPHA5E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA5,'k');
ylabel('ERRXPHA5');xlabel('discrete time k')
set(figure(16),'name','Tank 5 Process variables and estimation
error','numbertitle','off')
```

C.2: MATLAB script file – P_BASM2dF.m

```
% M-File: P_BASM2dF.m
% ======================= M-FILE DESCRIPTION =======================
% This m-file is used for:
% Calculation of the process rates for the full model

function Pk = P_BASM2dF(xk)

global fSI YH fXI YPO4 YPHA iPBM YAUT iNBM Kh etaNO3 etafe muH qfe bH qPHA qPP
muPAO bPAO bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4 KALK KA Kfe KPP KPS KP
KPHA KIPP KMAX
    % TANK 1
    P(1)= Kh * (xk(1)/(KO2+xk(1))) * ((xk(11)/xk(12))/(KX+(xk(11)/xk(12)))) * xk(12);
    P(2)= Kh * etaNO3 * (KO2/(KO2+xk(1))) * (xk(5)/(KNO3+xk(5))) *
((xk(11)/xk(12))/(KX+(xk(11)/xk(12)))) * xk(12);
    P(3)= Kh * etafe * (KO2/(KO2+xk(1))) * (KNO3/(KNO3+xk(5))) *
((xk(11)/xk(12))/(KX+(xk(11)/xk(12)))) * xk(12);
    P(4)= muH * (xk(1)/(KO2+xk(1))) * (xk(2)/(KF+xk(2))) * (xk(2)/(xk(2)+xk(3))) *
(xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) * (xk(8)/(KALK+xk(8))) * xk(12);
    P(5)= muH * (xk(1)/(KO2+xk(1))) * (xk(3)/(KA+xk(3))) * (xk(3)/(xk(2)+xk(3))) *
(xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) * (xk(8)/(KALK+xk(8))) * xk(12);
    P(6)= muH * etaNO3 * (KO2/(KO2+xk(1))) * (KNO3/(KNO3+xk(5))) * (xk(2)/(KF+xk(2))) *
(xk(2)/(xk(2)+xk(3))) * (xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) *
(xk(8)/(KALK+xk(8))) * xk(12);
    P(7)= muH * etaNO3 * (KO2/(KO2+xk(1))) * (KNO3/(KNO3+xk(5))) * (xk(3)/(KA+xk(3))) *
(xk(3)/(xk(2)+xk(3))) * (xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) *
(xk(8)/(KALK+xk(8)))* (xk(6)/(KP+xk(6))) * xk(12);
    P(8)= qfe * (KO2/(KO2+xk(1))) * (KNO3/(KNO3+xk(5))) * (xk(2)/(KF+xk(2))) *
(xk(8)/(KALK+xk(8)))* xk(12);
    P(9)= bH * xk(12);
    P(10)= qPHA * (xk(3)/(KA+xk(3))) * (xk(8)/(KALK+xk(8))) *
((xk(14)/xk(13))/(KPP+(xk(14)/xk(13)))) * xk(13);
    P(11)= qPP * (xk(1)/(KO2+xk(1))) * (xk(6)/(KPS+xk(6))) * (xk(8)/(KALK+xk(8))) *
((xk(15)/xk(13))/(KPHA+(xk(15)/xk(13)))) * ((KMAX-(xk(14)/xk(13)))/(KPP+KMAX-
(xk(14)/xk(13)))) * xk(13);
    P(12)= qPP * (xk(1)/(KO2+xk(1))) * (xk(6)/(KPS+xk(6))) * (xk(8)/(KALK+xk(8))) *
((xk(15)/xk(13))/(KPHA+(xk(15)/xk(13)))) * ((KMAX-(xk(14)/xk(13)))/(KPP+KMAX-
(xk(14)/xk(13)))) * xk(13) * etaNO3 * KO2/xk(1) * (xk(5)/(KNO3+xk(5)));
    P(13)= muPAO * (xk(1)/(KO2+xk(1))) * (xk(4)/(KNH4+xk(4))) * (xk(8)/(KALK+xk(8)))*
(xk(6)/(KP+xk(6))) * ((xk(15)/xk(13))/(KPHA+(xk(15)/xk(13)))) * xk(13);
    P(14)= muPAO * (xk(1)/(KO2+xk(1))) * (xk(4)/(KNH4+xk(4))) * (xk(8)/(KALK+xk(8)))*
(xk(6)/(KP+xk(6))) * ((xk(15)/xk(13))/(KPHA+(xk(15)/xk(13)))) * xk(13) * etaNO3 *
KO2/xk(1) * (xk(5)/(KNO3+xk(5)));
    P(15)= bPAO * xk(13) * (xk(8)/(KALK+xk(8)));
    P(16)= bPP * xk(14) * (xk(8)/(KALK+xk(8)));
    P(17)= bPHA * xk(15) * (xk(8)/(KALK+xk(8)));
    P(18)= muAUT * (xk(1)/(KO2+xk(1))) * (xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) *
(xk(8)/(KALK+xk(8))) * xk(16);
    P(19)= bAUT * xk(16);
    P(20)= kPRE * xk(6) * xk(18);
    P(21)= kRED * xk(19) * (xk(8)/(KALK+xk(8)));

    % TANK 2
    P(22)= Kh * (xk(20)/(KO2+xk(20))) * ((xk(30)/xk(31))/(KX+(xk(30)/xk(31)))) *
xk(31);
```

239

```
    P(23)= Kh * etaNO3 * (KO2/(KO2+xk(20))) * (xk(24)/(KNO3+xk(24))) *
((xk(30)/xk(31))/(KX+(xk(30)/xk(31)))) * xk(31);
    P(24)= Kh * etafe * (KO2/(KO2+xk(20))) * (KNO3/(KNO3+xk(24))) *
((xk(30)/xk(31))/(KX+(xk(30)/xk(31)))) * xk(31);
    P(25)= muH * (xk(20)/(KO2+xk(20))) * (xk(21)/(KF+xk(21))) *
(xk(21)/(xk(21)+xk(22))) * (xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) *
(xk(27)/(KALK+xk(27))) * xk(31);
    P(26)= muH * (xk(20)/(KO2+xk(20))) * (xk(22)/(KA+xk(22))) *
(xk(22)/(xk(21)+xk(22))) * (xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) *
(xk(27)/(KALK+xk(27))) * xk(31);
    P(27)= muH * etaNO3 * (KO2/(KO2+xk(20))) * (KNO3/(KNO3+xk(24))) *
(xk(21)/(KF+xk(21))) * (xk(21)/(xk(21)+xk(22))) * (xk(23)/(KNH4+xk(23))) *
(xk(25)/(KP+xk(25))) * (xk(27)/(KALK+xk(27))) * xk(31);
    P(28)= muH * etaNO3 * (KO2/(KO2+xk(20))) * (KNO3/(KNO3+xk(24))) *
(xk(22)/(KA+xk(22))) * (xk(22)/(xk(21)+xk(22))) * (xk(23)/(KNH4+xk(23))) *
(xk(25)/(KP+xk(25))) * (xk(27)/(KALK+xk(27)))* (xk(25)/(KP+xk(25))) * xk(31);
    P(29)= qfe * (KO2/(KO2+xk(20))) * (KNO3/(KNO3+xk(24))) * (xk(21)/(KF+xk(21))) *
(xk(27)/(KALK+xk(27)))* xk(31);
    P(30)= bH * xk(31);
    P(31)= qPHA * (xk(22)/(KA+xk(22))) * (xk(27)/(KALK+xk(27))) *
((xk(33)/xk(32))/(KPP+(xk(33)/xk(32)))) * xk(32);
    P(32)= qPP * (xk(20)/(KO2+xk(20))) * (xk(25)/(KPS+xk(25))) * (xk(27)/(KALK+xk(27)))
* ((xk(34)/xk(32))/(KPHA+(xk(34)/xk(32)))) * ((KMAX-(xk(33)/xk(32)))/(KPP+KMAX-
(xk(33)/xk(32)))) * xk(32);
    P(33)= qPP * (xk(20)/(KO2+xk(20))) * (xk(25)/(KPS+xk(25))) * (xk(27)/(KALK+xk(27)))
* ((xk(34)/xk(32))/(KPHA+(xk(34)/xk(32)))) * ((KMAX-(xk(33)/xk(32)))/(KPP+KMAX-
(xk(33)/xk(32)))) * xk(32) * etaNO3 * KO2/xk(20) * (xk(24)/(KNO3+xk(24)));
    P(34)= muPAO * (xk(20)/(KO2+xk(20))) * (xk(23)/(KNH4+xk(23))) *
(xk(27)/(KALK+xk(27)))* (xk(25)/(KP+xk(25))) * ((xk(34)/xk(32))/(KPHA+(xk(34)/xk(32))))
* xk(32);
    P(35)= muPAO * (xk(20)/(KO2+xk(20))) * (xk(23)/(KNH4+xk(23))) *
(xk(27)/(KALK+xk(27)))* (xk(25)/(KP+xk(25))) * ((xk(34)/xk(32))/(KPHA+(xk(34)/xk(32))))
* xk(32) * etaNO3 * KO2/xk(20) * (xk(24)/(KNO3+xk(24)));
    P(36)= bPAO * xk(32) * (xk(27)/(KALK+xk(27)));
    P(37)= bPP * xk(33) * (xk(27)/(KALK+xk(27)));
    P(38)= bPHA * xk(34) * (xk(27)/(KALK+xk(27)));
    P(39)= muAUT * (xk(20)/(KO2+xk(20))) * (xk(23)/(KNH4+xk(23))) *
(xk(25)/(KP+xk(25))) * (xk(27)/(KALK+xk(27))) * xk(35);
    P(40)= bAUT * xk(35);
    P(41)= kPRE * xk(25) * xk(37);
    P(42)= kRED * xk(38) * (xk(27)/(KALK+xk(27)));


    % TANK 3
    P(43)= Kh * (xk(39)/(KO2+xk(39))) * ((xk(49)/xk(50))/(KX+(xk(49)/xk(50)))) *
xk(50);
    P(44)= Kh * etaNO3 * (KO2/(KO2+xk(39))) * (xk(43)/(KNO3+xk(43))) *
((xk(49)/xk(50))/(KX+(xk(49)/xk(50)))) * xk(50);
    P(45)= Kh * etafe * (KO2/(KO2+xk(39))) * (KNO3/(KNO3+xk(43))) *
((xk(49)/xk(50))/(KX+(xk(49)/xk(50)))) * xk(50);
    P(46)= muH * (xk(39)/(KO2+xk(39))) * (xk(40)/(KF+xk(40))) *
(xk(40)/(xk(40)+xk(41))) * (xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) *
(xk(46)/(KALK+xk(46))) * xk(50);
    P(47)= muH * (xk(39)/(KO2+xk(39))) * (xk(41)/(KA+xk(41))) *
(xk(41)/(xk(40)+xk(41))) * (xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) *
(xk(46)/(KALK+xk(46))) * xk(50);
    P(48)= muH * etaNO3 * (KO2/(KO2+xk(39))) * (KNO3/(KNO3+xk(43))) *
(xk(40)/(KF+xk(40))) * (xk(40)/(xk(40)+xk(41))) * (xk(42)/(KNH4+xk(42))) *
(xk(44)/(KP+xk(44))) * (xk(46)/(KALK+xk(46))) * xk(50);
    P(49)= muH * etaNO3 * (KO2/(KO2+xk(39))) * (KNO3/(KNO3+xk(43))) *
(xk(41)/(KA+xk(41))) * (xk(41)/(xk(40)+xk(41))) * (xk(42)/(KNH4+xk(42))) *
(xk(44)/(KP+xk(44))) * (xk(46)/(KALK+xk(46)))* (xk(44)/(KP+xk(44))) * xk(50);
    P(50)= qfe * (KO2/(KO2+xk(39))) * (KNO3/(KNO3+xk(43))) * (xk(40)/(KF+xk(40))) *
(xk(46)/(KALK+xk(46)))* xk(50);
    P(51)= bH * xk(50);
    P(52)= qPHA * (xk(41)/(KA+xk(41))) * (xk(46)/(KALK+xk(46))) *
((xk(52)/xk(51))/(KPP+(xk(52)/xk(51)))) * xk(51);
```

240

```
    P(53)= qPP * (xk(39)/(KO2+xk(39))) * (xk(44)/(KPS+xk(44))) * (xk(46)/(KALK+xk(46)))
* ((xk(53)/xk(51))/(KPHA+(xk(53)/xk(51)))) * ((KMAX-(xk(52)/xk(51)))/(KPP+KMAX-
(xk(52)/xk(51)))) * xk(51);
    P(54)= qPP * (xk(39)/(KO2+xk(39))) * (xk(44)/(KPS+xk(44))) * (xk(46)/(KALK+xk(46)))
* ((xk(53)/xk(51))/(KPHA+(xk(53)/xk(51)))) * ((KMAX-(xk(52)/xk(51)))/(KPP+KMAX-
(xk(52)/xk(51)))) * xk(51) * etaNO3 * KO2/xk(39) * (xk(43)/(KNO3+xk(43)));
    P(55)= muPAO * (xk(39)/(KO2+xk(39))) * (xk(42)/(KNH4+xk(42))) *
(xk(46)/(KALK+xk(46)))* (xk(44)/(KP+xk(44))) * ((xk(53)/xk(51))/(KPHA+(xk(53)/xk(51))))
* xk(51);
    P(56)= muPAO * (xk(39)/(KO2+xk(39))) * (xk(42)/(KNH4+xk(42))) *
(xk(46)/(KALK+xk(46)))* (xk(44)/(KP+xk(44))) * ((xk(53)/xk(51))/(KPHA+(xk(53)/xk(51))))
* xk(51) * etaNO3 * KO2/xk(39) * (xk(43)/(KNO3+xk(43)));
    P(57)= bPAO * xk(51) * (xk(46)/(KALK+xk(46)));
    P(58)= bPP * xk(52) * (xk(46)/(KALK+xk(46)));
    P(59)= bPHA * xk(53) * (xk(46)/(KALK+xk(46)));
    P(60)= muAUT * (xk(39)/(KO2+xk(39))) * (xk(42)/(KNH4+xk(42))) *
(xk(44)/(KP+xk(44))) * (xk(46)/(KALK+xk(46))) * xk(54);
    P(61)= bAUT * xk(54);
    P(62)= kPRE * xk(44) * xk(56);
    P(63)= kRED * xk(57) * (xk(46)/(KALK+xk(46)));


    % TANK 4
    P(64)= Kh * (xk(58)/(KO2+xk(58))) * ((xk(68)/xk(69))/(KX+(xk(68)/xk(69)))) *
xk(69);
    P(65)= Kh * etaNO3 * (KO2/(KO2+xk(58))) * (xk(62)/(KNO3+xk(62))) *
((xk(68)/xk(69))/(KX+(xk(68)/xk(69)))) * xk(69);
    P(66)= Kh * etafe * (KO2/(KO2+xk(58))) * (KNO3/(KNO3+xk(62))) *
((xk(68)/xk(69))/(KX+(xk(68)/xk(69)))) * xk(69);
    P(67)= muH * (xk(58)/(KO2+xk(58))) * (xk(59)/(KF+xk(59))) *
(xk(59)/(xk(59)+xk(60))) * (xk(61)/(KNH4+xk(61))) * (xk(63)/(KP+xk(63))) *
(xk(65)/(KALK+xk(65))) * xk(69);
    P(68)= muH * (xk(58)/(KO2+xk(58))) * (xk(60)/(KA+xk(60))) *
(xk(60)/(xk(59)+xk(60))) * (xk(61)/(KNH4+xk(61))) * (xk(63)/(KP+xk(63))) *
(xk(65)/(KALK+xk(65))) * xk(69);
    P(69)= muH * etaNO3 * (KO2/(KO2+xk(58))) * (KNO3/(KNO3+xk(62))) *
(xk(59)/(KF+xk(59))) * (xk(59)/(xk(59)+xk(60))) * (xk(61)/(KNH4+xk(61))) *
(xk(63)/(KP+xk(63))) * (xk(65)/(KALK+xk(65))) * xk(69);
    P(70)= muH * etaNO3 * (KO2/(KO2+xk(58))) * (KNO3/(KNO3+xk(62))) *
(xk(60)/(KA+xk(60))) * (xk(60)/(xk(59)+xk(60))) * (xk(61)/(KNH4+xk(61))) *
(xk(63)/(KP+xk(63))) * (xk(65)/(KALK+xk(65)))* (xk(63)/(KP+xk(63))) * xk(69);
    P(71)= qfe * (KO2/(KO2+xk(58))) * (KNO3/(KNO3+xk(62))) * (xk(59)/(KF+xk(59))) *
(xk(65)/(KALK+xk(65)))* xk(69);
    P(72)= bH * xk(69);
    P(73)= qPHA * (xk(60)/(KA+xk(60))) * (xk(65)/(KALK+xk(65))) *
((xk(71)/xk(70))/(KPP+(xk(71)/xk(70)))) * xk(70);
    P(74)= qPP * (xk(58)/(KO2+xk(58))) * (xk(63)/(KPS+xk(63))) * (xk(65)/(KALK+xk(65)))
* ((xk(72)/xk(70))/(KPHA+(xk(72)/xk(70)))) * ((KMAX-(xk(71)/xk(70)))/(KPP+KMAX-
(xk(71)/xk(70)))) * xk(70);
    P(75)= qPP * (xk(58)/(KO2+xk(58))) * (xk(63)/(KPS+xk(63))) * (xk(65)/(KALK+xk(65)))
* ((xk(72)/xk(70))/(KPHA+(xk(72)/xk(70)))) * ((KMAX-(xk(71)/xk(70)))/(KPP+KMAX-
(xk(71)/xk(70)))) * xk(70) * etaNO3 * KO2/xk(58) * (xk(62)/(KNO3+xk(62)));
    P(76)= muPAO * (xk(58)/(KO2+xk(58))) * (xk(61)/(KNH4+xk(61))) *
(xk(65)/(KALK+xk(65)))* (xk(63)/(KP+xk(63))) * ((xk(72)/xk(70))/(KPHA+(xk(72)/xk(70))))
* xk(70);
    P(77)= muPAO * (xk(58)/(KO2+xk(58))) * (xk(61)/(KNH4+xk(61))) *
(xk(65)/(KALK+xk(65)))* (xk(63)/(KP+xk(63))) * ((xk(72)/xk(70))/(KPHA+(xk(72)/xk(70))))
* xk(70) * etaNO3 * KO2/xk(58) * (xk(62)/(KNO3+xk(62)));
    P(78)= bPAO * xk(70) * (xk(65)/(KALK+xk(65)));
    P(79)= bPP * xk(71) * (xk(65)/(KALK+xk(65)));
    P(80)= bPHA * xk(72) * (xk(65)/(KALK+xk(65)));
    P(81)= muAUT * (xk(58)/(KO2+xk(58))) * (xk(61)/(KNH4+xk(61))) *
(xk(63)/(KP+xk(63))) * (xk(65)/(KALK+xk(65))) * xk(73);
    P(82)= bAUT * xk(73);
    P(83)= kPRE * xk(63) * xk(75);
    P(84)= kRED * xk(76) * (xk(65)/(KALK+xk(65)));
```

241

```
% TANK 5
   P(85)= Kh * (xk(77)/(KO2+xk(77))) * ((xk(87)/xk(88))/(KX+(xk(87)/xk(88)))) *
xk(88);
   P(86)= Kh * etaNO3 * (KO2/(KO2+xk(77))) * (xk(81)/(KNO3+xk(81))) *
((xk(87)/xk(88))/(KX+(xk(87)/xk(88)))) * xk(88);
   P(87)= Kh * etafe * (KO2/(KO2+xk(77))) * (KNO3/(KNO3+xk(81))) *
((xk(87)/xk(88))/(KX+(xk(87)/xk(88)))) * xk(88);
   P(88)= muH * (xk(77)/(KO2+xk(77))) * (xk(78)/(KF+xk(78))) *
(xk(78)/(xk(78)+xk(79))) * (xk(80)/(KNH4+xk(80))) * (xk(82)/(KP+xk(82))) *
(xk(84)/(KALK+xk(84))) * xk(88);
   P(89)= muH * (xk(77)/(KO2+xk(77))) * (xk(79)/(KA+xk(79))) *
(xk(79)/(xk(78)+xk(79))) * (xk(80)/(KNH4+xk(80))) * (xk(82)/(KP+xk(82))) *
(xk(84)/(KALK+xk(84))) * xk(88);
   P(90)= muH * etaNO3 * (KO2/(KO2+xk(77))) * (KNO3/(KNO3+xk(81))) *
(xk(78)/(KF+xk(78))) * (xk(78)/(xk(78)+xk(79))) * (xk(80)/(KNH4+xk(80))) *
(xk(82)/(KP+xk(82))) * (xk(84)/(KALK+xk(84))) * xk(88);
   P(91)= muH * etaNO3 * (KO2/(KO2+xk(77))) * (KNO3/(KNO3+xk(81))) *
(xk(79)/(KA+xk(79))) * (xk(79)/(xk(78)+xk(79))) * (xk(80)/(KNH4+xk(80))) *
(xk(82)/(KP+xk(82))) * (xk(84)/(KALK+xk(84)))* (xk(82)/(KP+xk(82))) * xk(88);
   P(92)= qfe * (KO2/(KO2+xk(77))) * (KNO3/(KNO3+xk(81))) * (xk(78)/(KF+xk(78))) *
(xk(84)/(KALK+xk(84)))* xk(88);
   P(93)= bH * xk(88);
   P(94)= qPHA * (xk(79)/(KA+xk(79))) * (xk(84)/(KALK+xk(84))) *
((xk(90)/xk(89))/(KPP+(xk(90)/xk(89)))) * xk(89);
   P(95)= qPP * (xk(77)/(KO2+xk(77))) * (xk(82)/(KPS+xk(82))) * (xk(84)/(KALK+xk(84)))
* ((xk(91)/xk(89))/(KPHA+(xk(91)/xk(89)))) * ((KMAX-(xk(90)/xk(89)))/(KPP+KMAX-
(xk(90)/xk(89)))) * xk(89);
   P(96)= qPP * (xk(77)/(KO2+xk(77))) * (xk(82)/(KPS+xk(82))) * (xk(84)/(KALK+xk(84)))
* ((xk(91)/xk(89))/(KPHA+(xk(91)/xk(89)))) * ((KMAX-(xk(90)/xk(89)))/(KPP+KMAX-
(xk(90)/xk(89)))) * xk(89) * etaNO3 * KO2/xk(77) * (xk(81)/(KNO3+xk(81)));
   P(97)= muPAO * (xk(77)/(KO2+xk(77))) * (xk(80)/(KNH4+xk(80))) *
(xk(84)/(KALK+xk(84)))* (xk(82)/(KP+xk(82))) * ((xk(91)/xk(89))/(KPHA+(xk(91)/xk(89))))
* xk(89);
   P(98)= muPAO * (xk(77)/(KO2+xk(77))) * (xk(80)/(KNH4+xk(80))) *
(xk(84)/(KALK+xk(84)))* (xk(82)/(KP+xk(82))) * ((xk(91)/xk(89))/(KPHA+(xk(91)/xk(89))))
* xk(89) * etaNO3 * KO2/xk(77) * (xk(81)/(KNO3+xk(81)));
   P(99)= bPAO * xk(89) * (xk(84)/(KALK+xk(84)));
   P(100)= bPP * xk(90) * (xk(84)/(KALK+xk(84)));
   P(101)= bPHA * xk(91) * (xk(84)/(KALK+xk(84)));
   P(102)= muAUT * (xk(77)/(KO2+xk(77))) * (xk(80)/(KNH4+xk(80))) *
(xk(82)/(KP+xk(82))) * (xk(84)/(KALK+xk(84))) * xk(92);
   P(103)= bAUT * xk(92);
   P(104)= kPRE * xk(82) * xk(94);
   P(105)= kRED * xk(95) * (xk(84)/(KALK+xk(84)));

   Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14)
P(15) P(16) P(17) P(18) P(19) P(20) P(21) P(22) P(23) P(24) P(25) P(26) P(27) P(28)
P(29) P(30) P(31) P(32) P(33) P(34) P(35) P(36) P(37) P(38) P(39) P(40) P(41) P(42)
P(43) P(44) P(45) P(46) P(47) P(48) P(49) P(50) P(51) P(52) P(53) P(54) P(55) P(56)
P(57) P(58) P(59) P(60) P(61) P(62) P(63) P(64) P(65) P(66) P(67) P(68) P(69) P(70)
P(71) P(72) P(73) P(74) P(75) P(76) P(77) P(78) P(79) P(80) P(81) P(82) P(83) P(84)
P(85) P(86) P(87) P(88) P(89) P(90) P(91) P(92) P(93) P(94) P(95) P(96) P(97) P(98)
P(99) P(100) P(101) P(102) P(103) P(104) P(105)]';
```

C.3: MATLAB script file – P_BASM2dR.m

```
% M-File: P_BASM2dR.m
% ========================= M-FILE DESCRIPTION ==========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model

function    Pk = P_BASM2dR(zk,uk,XHk,XAk,SALKk,XPAOk)
```

242

```
global     fSI YH fXI YPO4 YPHA iPBM YAUT iNBM Kh etaNO3 etafe muH qfe bH qPHA qPP
muPAO bPAO bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4 KALK KA Kfe KPP KPS KP
KPHA KIPP KMAX
% TANK 1
%     SF      SA      SNH4      SNO3      SPO4      XS      XPP      XPHA
P(1)= (uk(1)/(KO2+uk(1))) * ((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(2)= etaNO3 * (KO2/(KO2+uk(1))) * (zk(4)/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(3)= etafe * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(4)= (uk(1)/(KO2+uk(1))) * (zk(1)/(KF+zk(1))) * (zk(1)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(5)= (uk(1)/(KO2+uk(1))) * (zk(2)/(KA+zk(2))) * (zk(2)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/.(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(6)= etaNO3 * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(1)/(KF+zk(1))) *
(zk(1)/(zk(1)+zk(2))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(7)= etaNO3 * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(2)/(KA+zk(2))) *
(zk(2)/(zk(1)+zk(2))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1)))* (zk(5)/(KP+zk(5))) * XHk(1);
P(8)= (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(1)/(KF+zk(1))) *
(SALKk(1)/(KALK+SALKk(1)))* XHk(1);
P(9)= (zk(2)/(KA+zk(2))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(7)/XPAOk(1))/(KPP+(zk(7)/XPAOk(1)))) * XPAOk(1);
P(10)= (uk(1)/(KO2+uk(1))) * (zk(5)/(KPS+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * ((KMAX-(zk(7)/XPAOk(1)))/(KPP+KMAX-
(zk(7)/XPAOk(1)))) * XPAOk(1);
P(11)= (uk(1)/(KO2+uk(1))) * (zk(5)/(KPS+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * ((KMAX-(zk(7)/XPAOk(1)))/(KPP+KMAX-
(zk(7)/XPAOk(1)))) * XPAOk(1) * etaNO3 * KO2/uk(1) * (zk(4)/(KNO3+zk(4)));
P(12)= (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) * (SALKk(1)/(KALK+SALKk(1)))*
(zk(5)/(KP+zk(5))) * ((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * XPAOk(1);
P(13)= (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) * (SALKk(1)/(KALK+SALKk(1)))*
(zk(5)/(KP+zk(5))) * ((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * XPAOk(1) * etaNO3 *
KO2/uk(1) * (zk(4)/(KNO3+zk(4)));
P(14)= (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XAk(1);


P1k  = [[P(1)     P(2)     P(3)      0       0       0       0       0       0       0
0      0        0        0     ]; %Kh(1)
       [0        0        0        P(4)    P(5)    P(6)    P(7)      0       0       0
0      0        0        0     ]; %Muh(1)
       [0        0        0        0       0       0       0       P(8)     0       0
0      0        0        0     ]; %qfe(1)
       [0        0        0        0       0       0       0        0      P(9)     0
0      0        0        0     ]; %qpha(1)
       [0        0        0        0       0       0       0        0       0      P(10)
P(11)   0        0        0     ]; %qpp(1)
       [0        0        0        0       0       0       0        0       0       0
0      P(12)    P(13)    0     ]; %mupao(1)
       [0        0        0       P(14)] ]; %muAut(1)

% TANK 2
P(15)= (uk(2)/(KO2+uk(2))) * ((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(16)= etaNO3 * (KO2/(KO2+uk(2))) * (zk(12)/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(17)= etafe * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(18)= (uk(2)/(KO2+uk(2))) * (zk(9)/(KF+zk(9))) * (zk(9)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(19)= (uk(2)/(KO2+uk(2))) * (zk(10)/(KA+zk(10))) * (zk(10)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(20)= etaNO3 * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(9)/(KF+zk(9))) *
(zk(9)/(zk(9)+zk(10))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XHk(2);
```

243

```
P(21)= etaNO3 * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(10)/(KA+zk(10))) *
(zk(10)/(zk(9)+zk(10))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2)))* (zk(13)/(KP+zk(13))) * XHk(2);
P(22)= (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(9)/(KF+zk(9))) *
(SALKk(2)/(KALK+SALKk(2)))* XHk(2);
P(23)= (zk(10)/(KA+zk(10))) * (SALKk(2)/(KALK+SALKk(2))) *
((zk(15)/XPAOk(2))/(KPP+(zk(15)/XPAOk(2)))) * XPAOk(2);
P(24)= (uk(2)/(KO2+uk(2))) * (zk(13)/(KPS+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) *
((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * ((KMAX-(zk(15)/XPAOk(2)))/(KPP+KMAX-
(zk(15)/XPAOk(2)))) * XPAOk(2);
P(25)= (uk(2)/(KO2+uk(2))) * (zk(13)/(KPS+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) *
((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * ((KMAX-(zk(15)/XPAOk(2)))/(KPP+KMAX-
(zk(15)/XPAOk(2)))) * XPAOk(2) * etaNO3 * KO2/uk(2) * (zk(12)/(KNO3+zk(12)));
P(26)= (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (SALKk(2)/(KALK+SALKk(2)))*
(zk(13)/(KP+zk(13))) * ((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * XPAOk(2);
P(27)= (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (SALKk(2)/(KALK+SALKk(2)))*
(zk(13)/(KP+zk(13))) * ((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * XPAOk(2) * etaNO3
* KO2/uk(2) * (zk(12)/(KNO3+zk(12)));
P(28)= (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XAk(2);


P2k = [[P(15)    P(16)   P(17)    0        0        0        0        0        0        0
0      0       0        0      ]; %Kh(2)
       [0     -  0        0        P(18)    P(19)    P(20)    P(21)    0        0        0
0      0       0        0      ]; %Muh(2)
       [0        0        0        0        0        0        0        P(22)    0        0
0      0       0        0      ]; %qfe(2)
       [0        0        0        0        0        0        0        0        P(23)    0
0      0       0        0      ]; %qpha(2)
       [0        0        0        0        0        0        0        0        0        P(24)
P(25)  0       0        0      ]; %qpp(2)
       [0        0        0        0        0        0        0        0        0        0
0      P(26)   P(27)    0      ]; %mupao(2)
       [0        0        0        0        0        0        0        0        0        0
0      0       0        P(28)] ]; %muAut(2)


% TANK 3
P(29)= (uk(3)/(KO2+uk(3))) * ((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(30)= etaNO3 * (KO2/(KO2+uk(3))) * (zk(20)/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(31)= etafe * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(32)= (uk(3)/(KO2+uk(3))) * (zk(17)/(KF+zk(17))) * (zk(17)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(33)= (uk(3)/(KO2+uk(3))) * (zk(18)/(KA+zk(18))) * (zk(18)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(34)= etaNO3 * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(17)/(KF+zk(17))) *
(zk(17)/(zk(17)+zk(18))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(35)= etaNO3 * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(18)/(KA+zk(18))) *
(zk(18)/(zk(17)+zk(18))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3)))* (zk(21)/(KP+zk(21))) * XHk(3);
P(36)= (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(17)/(KF+zk(17))) *
(SALKk(3)/(KALK+SALKk(3)))* XHk(3);
P(37)= (zk(18)/(KA+zk(18))) * (SALKk(3)/(KALK+SALKk(3))) *
((zk(23)/XPAOk(3))/(KPP+(zk(23)/XPAOk(3)))) * XPAOk(3);
P(38)= (uk(3)/(KO2+uk(3))) * (zk(21)/(KPS+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) *
((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * ((KMAX-(zk(23)/XPAOk(3)))/(KPP+KMAX-
(zk(23)/XPAOk(3)))) * XPAOk(3);
P(39)= (uk(3)/(KO2+uk(3))) * (zk(21)/(KPS+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) *
((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * ((KMAX-(zk(23)/XPAOk(3)))/(KPP+KMAX-
(zk(23)/XPAOk(3)))) * XPAOk(3) * etaNO3 * KO2/uk(3) * (zk(20)/(KNO3+zk(20)));
P(40)= (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (SALKk(3)/(KALK+SALKk(3)))*
(zk(21)/(KP+zk(21))) * ((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * XPAOk(3);
```

244

```
P(41)= (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (SALKk(3)/(KALK+SALKk(3)))*
(zk(21)/(KP+zk(21))) * ((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * XPAOk(3) * etaNO3
* KO2/uk(3) * (zk(20)/(KNO3+zk(20)));
P(42)= (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XAk(3);


P3k = [[P(29)    P(30)    P(31)    0        0       0       0       0       0       0
0      0        0        0     ]; %Kh(2)
       [0        0        0        P(32)    P(33)   P(34)   P(35)   0       0       0
0      0        0        0     ]; %Muh(2)
       [0        0        0        0        0       0       0       P(36)   0       0
0      0        0        0     ]; %qfe(2)
       [0        0        0        0        0       0       0       0       P(37)   0
0      0        0        0     ]; %qpha(2)
       [0        0        0        0        0       0       0       0       0       P(38)
P(29)  0        0        0     ]; %qpp(2)
       [0        0        0        0        0       0       0       0       0       0
0      P(40)    P(41)    0     ]; %mupao(2)
       [0        0        0        0        0       0       0       0       0       0
0      0        0        P(42)] ]; %muAut(2)


% TANK 4
P(43)= (uk(4)/(KO2+uk(4))) * ((zk(30)/XHk(4))/(KX+(zk(30)/XHk(4)))) * XHk(4);
P(44)= etaNO3 * (KO2/(KO2+uk(4))) * (zk(28)/(KNO3+zk(28))) *
((zk(30)/XHk(4))/(KX+(zk(30)/XHk(4)))) * XHk(4);
P(45)= etafe * (KO2/(KO2+uk(4))) * (KNO3/(KNO3+zk(28))) *
((zk(30)/XHk(4))/(KX+(zk(30)/XHk(4)))) * XHk(4);
P(46)= (uk(4)/(KO2+uk(4))) * (zk(25)/(KF+zk(25))) * (zk(25)/(zk(25)+zk(26))) *
(zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) * (SALKk(4)/(KALK+SALKk(4))) * XHk(4);
P(47)= (uk(4)/(KO2+uk(4))) * (zk(26)/(KA+zk(26))) * (zk(26)/(zk(25)+zk(26))) *
(zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) * (SALKk(4)/(KALK+SALKk(4))) * XHk(4);
P(48)= etaNO3 * (KO2/(KO2+uk(4))) * (KNO3/(KNO3+zk(28))) * (zk(25)/(KF+zk(25))) *
(zk(25)/(zk(25)+zk(26))) * (zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) *
(SALKk(4)/(KALK+SALKk(4))) * XHk(4);
P(49)= etaNO3 * (KO2/(KO2+uk(4))) * (KNO3/(KNO3+zk(28))) * (zk(26)/(KA+zk(26))) *
(zk(26)/(zk(25)+zk(26))) * (zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) *
(SALKk(4)/(KALK+SALKk(4)))* (zk(29)/(KP+zk(29))) * XHk(4);
P(50)= (KO2/(KO2+uk(4))) * (KNO3/(KNO3+zk(28))) * (zk(25)/(KF+zk(25))) *
(SALKk(4)/(KALK+SALKk(4)))* XHk(4);
P(51)= (zk(26)/(KA+zk(26))) * (SALKk(4)/(KALK+SALKk(4))) *
((zk(31)/XPAOk(4))/(KPP+(zk(31)/XPAOk(4)))) * XPAOk(4);
P(52)= (uk(4)/(KO2+uk(4))) * (zk(29)/(KPS+zk(29))) * (SALKk(4)/(KALK+SALKk(4))) *
((zk(32)/XPAOk(4))/(KPHA+(zk(32)/XPAOk(4)))) * ((KMAX-(zk(31)/XPAOk(4)))/(KPP+KMAX-
(zk(31)/XPAOk(4)))) * XPAOk(4);
P(53)= (uk(4)/(KO2+uk(4))) * (zk(29)/(KPS+zk(29))) * (SALKk(4)/(KALK+SALKk(4))) *
((zk(32)/XPAOk(4))/(KPHA+(zk(32)/XPAOk(4)))) * ((KMAX-(zk(31)/XPAOk(4)))/(KPP+KMAX-
(zk(31)/XPAOk(4)))) * XPAOk(4) * etaNO3 * KO2/uk(4) * (zk(28)/(KNO3+zk(28)));
P(54)= (uk(4)/(KO2+uk(4))) * (zk(27)/(KNH4+zk(27))) * (SALKk(4)/(KALK+SALKk(4)))*
(zk(29)/(KP+zk(29))) * ((zk(32)/XPAOk(4))/(KPHA+(zk(32)/XPAOk(4)))) * XPAOk(4);
P(55)= (uk(4)/(KO2+uk(4))) * (zk(27)/(KNH4+zk(27))) * (SALKk(4)/(KALK+SALKk(4)))*
(zk(29)/(KP+zk(29))) * ((zk(32)/XPAOk(4))/(KPHA+(zk(32)/XPAOk(4)))) * XPAOk(4) * etaNO3
* KO2/uk(4) * (zk(28)/(KNO3+zk(28)));
P(56)= (uk(4)/(KO2+uk(4))) * (zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) *
(SALKk(4)/(KALK+SALKk(4))) * XAk(4);


P4k = [[P(43)    P(44)    P(45)    0        0       0       0       0       0       0
0      0        0        0     ]; %Kh(2)
       [0        0        0        P(46)    P(47)   P(48)   P(49)   0       0       0
0      0        0        0     ]; %Muh(2)
       [0        0        0        0        0       0       0       P(50)   0       0
0      0        0        0     ]; %qfe(2)
       [0        0        0        0        0       0       0       0       P(51)   0
0      0        0        0     ]; %qpha(2)
       [0        0        0        0        0       0       0       0       0       P(52)
P(53)  0        0        0     ]; %qpp(2)
```

245

```
            [0          0          0          0          0          0          0          0          0          0
0           P(54)      P(55)      0        ]; %mupao(2)
            [0          0          0          0          0          0          0          0          0          0
0           0          0          P(56)]   ]; %muAut(2)


% TANK 5
P(57)= (uk(5)/(KO2+uk(5))) * ((zk(38)/XHk(5))/(KX+(zk(38)/XHk(5)))) * XHk(5);
P(58)= etaNO3 * (KO2/(KO2+uk(5))) * (zk(36)/(KNO3+zk(36))) *
((zk(38)/XHk(5))/(KX+(zk(38)/XHk(5)))) * XHk(5);
P(59)= etafe * (KO2/(KO2+uk(5))) * (KNO3/(KNO3+zk(36))) *
((zk(38)/XHk(5))/(KX+(zk(38)/XHk(5)))) * XHk(5);
P(60)= (uk(5)/(KO2+uk(5))) * (zk(33)/(KF+zk(33))) * (zk(33)/(zk(33)+zk(34))) *
(zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) * (SALKk(5)/(KALK+SALKk(5))) * XHk(5);
P(61)= (uk(5)/(KO2+uk(5))) *·(zk(34)/(KA+zk(34))) * (zk(34)/(zk(33)+zk(34))) *
(zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) * (SALKk(5)/(KALK+SALKk(5))) * XHk(5);
P(62)= etaNO3 * (KO2/(KO2+uk(5))) * (KNO3/(KNO3+zk(36))) * (zk(33)/(KF+zk(33))) *
(zk(33)/(zk(33)+zk(34))) * (zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) *
(SALKk(5)/(KALK+SALKk(5))) * XHk(5);
P(63)= etaNO3 * (KO2/(KO2+uk(5))) * (KNO3/(KNO3+zk(36))) * (zk(34)/(KA+zk(34))) *
(zk(34)/(zk(33)+zk(34))) * (zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) *
(SALKk(5)/(KALK+SALKk(5)))* (zk(37)/(KP+zk(37))) * XHk(5);
P(64)= (KO2/(KO2+uk(5))) * (KNO3/(KNO3+zk(36))) * (zk(33)/(KF+zk(33))) *
(SALKk(5)/(KALK+SALKk(5)))* XHk(5);
P(65)= (zk(34)/(KA+zk(34))) * (SALKk(5)/(KALK+SALKk(5))) *
((zk(39)/XPAOk(5))/(KPP+(zk(39)/XPAOk(5)))) * XPAOk(5);
P(66)= (uk(5)/(KO2+uk(5))) * (zk(37)/(KPS+zk(37))) * (SALKk(5)/(KALK+SALKk(5))) *
((zk(40)/XPAOk(5))/(KPHA+(zk(40)/XPAOk(5)))) * ((KMAX-(zk(39)/XPAOk(5)))/(KPP+KMAX-
(zk(39)/XPAOk(5)))) * XPAOk(5);
P(67)= (uk(5)/(KO2+uk(5))) * (zk(37)/(KPS+zk(37))) * (SALKk(5)/(KALK+SALKk(5))) *
((zk(40)/XPAOk(5))/(KPHA+(zk(40)/XPAOk(5)))) * ((KMAX-(zk(39)/XPAOk(5)))/(KPP+KMAX-
(zk(39)/XPAOk(5)))) * XPAOk(5) * etaNO3 * KO2/uk(5) * (zk(36)/(KNO3+zk(36)));
P(68)= (uk(5)/(KO2+uk(5))) * (zk(35)/(KNH4+zk(35))) * (SALKk(5)/(KALK+SALKk(5)))*
(zk(37)/(KP+zk(37))) * ((zk(40)/XPAOk(5))/(KPHA+(zk(40)/XPAOk(5)))) * XPAOk(5);
P(69)= (uk(5)/(KO2+uk(5))) * (zk(35)/(KNH4+zk(35))) * (SALKk(5)/(KALK+SALKk(5)))*
(zk(37)/(KP+zk(37))) * ((zk(40)/XPAOk(5))/(KPHA+(zk(40)/XPAOk(5)))) * XPAOk(5) * etaNO3
* KO2/uk(5) * (zk(36)/(KNO3+zk(36)));
P(70)= (uk(5)/(KO2+uk(5))) * (zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) *
(SALKk(5)/(KALK+SALKk(5))) * XAk(5);


P5k = [[P(57)     P(58)      P(59)      0          0          0          0          0          0          0
0           0          0          0        ]; %Kh(2)
            [0          0          0          P(60)      P(61)      P(62)      P(63)      0          0          0
0           0          0          0        ]; %Muh(2)
            [0          0          0          0          0          0          0          P(64)      0          0
0           0          0          0        ]; %qfe(2)
            [0          0          0          0          0          0          0          0          P(65)      0
0           0          0          0        ]; %qpha(2)
            [0          0          0          0          0          0          0          0          0          P(66)
P(67)       0          0          0        ]; %qpp(2)
            [0          0          0          0          0          0          0          0          0          0
0           P(68)      P(69)      0        ]; %mupao(2)
            [0          0          0          0          0          0          0          0          0          0
0           0          0          P(70)]   ]; %muAut(2)

%Matrix Pk for the whole process
Pk = [P1k'; P2k'; P3k'; P4k'; P5k'];
```

C.4: MATLAB script file – P_BASM2dRE.m

```
% M-File: P_BASM2dRE.m
% ======================== M-FILE DESCRIPTION ========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
```

246

```matlab
% model

% Description of the function function
function    Pk =
P_BASM2dRE(zk,uk,XHk,XAk,SALKk,XPAOk,kHek,muHek,qfeek,qPHAek,qPPek,muPAOek,muAUTek)

global     fSI YH fXI YPO4 YPHA iPBM YAUT iNBM Kh etaNO3 etafe muH qfe bH qPHA qPP
muPAO bPAO bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4 KALK KA Kfe KPP KPS KP
KPHA KIPP KMAX

% TANK 1
%        SF      SA      SNH4    SNO3    SPO4      XS        XPP      XPHA
P(1)= kHek * (uk(1)/(KO2+uk(1))) * ((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(2)= kHek * etaNO3 * (KO2/(KO2+uk(1))) * (zk(4)/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(3)= kHek * etafe * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(4)= muHek * (uk(1)/(KO2+uk(1))) * (zk(1)/(KF+zk(1))) * (zk(1)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(5)= muHek * (uk(1)/(KO2+uk(1))) * (zk(2)/(KA+zk(2))) * (zk(2)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(6)= muHek * etaNO3 * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(1)/(KF+zk(1))) *
(zk(1)/(zk(1)+zk(2))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(7)= muHek * etaNO3 * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(2)/(KA+zk(2))) *
(zk(2)/(zk(1)+zk(2))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1)))* (zk(5)/(KP+zk(5))) * XHk(1);
P(8)= qfeek * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(1)/(KF+zk(1))) *
(SALKk(1)/(KALK+SALKk(1)))* XHk(1);
P(9)= qPHAek * (zk(2)/(KA+zk(2))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(7)/XPAOk(1))/(KPP+(zk(7)/XPAOk(1)))) * XPAOk(1);
P(10)= qPPek * (uk(1)/(KO2+uk(1))) * (zk(5)/(KPS+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * ((KMAX-(zk(7)/XPAOk(1)))/(KPP+KMAX-
(zk(7)/XPAOk(1)))) * XPAOk(1);
P(11)= qPPek * (uk(1)/(KO2+uk(1))) * (zk(5)/(KPS+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * ((KMAX-(zk(7)/XPAOk(1)))/(KPP+KMAX-
(zk(7)/XPAOk(1)))) * XPAOk(1) * etaNO3 * KO2/uk(1) * (zk(4)/(KNO3+zk(4)));
P(12)= muPAOek * (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) *
(SALKk(1)/(KALK+SALKk(1)))* (zk(5)/(KP+zk(5))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * XPAOk(1);
P(13)= muPAOek * (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) *
(SALKk(1)/(KALK+SALKk(1)))* (zk(5)/(KP+zk(5))) *.
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * XPAOk(1) * etaNO3 * KO2/uk(1) *
(zk(4)/(KNO3+zk(4)));
P(14)= muAUTek * (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XAk(1);


% TANK 2
P(15)= kHek * (uk(2)/(KO2+uk(2))) * ((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(16)= kHek * etaNO3 * (KO2/(KO2+uk(2))) * (zk(12)/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(17)= kHek * etafe * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(18)= muHek * (uk(2)/(KO2+uk(2))) * (zk(9)/(KF+zk(9))) * (zk(9)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(19)= muHek * (uk(2)/(KO2+uk(2))) * (zk(10)/(KA+zk(10))) * (zk(10)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(20)= muHek * etaNO3 * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(9)/(KF+zk(9))) *
(zk(9)/(zk(9)+zk(10))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(21)= muHek * etaNO3 * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(10)/(KA+zk(10)))
* (zk(10)/(zk(9)+zk(10))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2)))* (zk(13)/(KP+zk(13))) * XHk(2);
P(22)= qfeek * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(9)/(KF+zk(9))) *
(SALKk(2)/(KALK+SALKk(2)))* XHk(2);
```

```
P(23)= qPHAek * (zk(10)/(KA+zk(10))) * (SALKk(2)/(KALK+SALKk(2))) *
((zk(15)/XPAOk(2))/(KPP+(zk(15)/XPAOk(2)))) * XPAOk(2);
P(24)= qPPek * (uk(2)/(KO2+uk(2))) * (zk(13)/(KPS+zk(13))) * (SALKk(2)/(KALK+SALKk(2)))
* ((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * ((KMAX-(zk(15)/XPAOk(2)))/(KPP+KMAX-
(zk(15)/XPAOk(2)))) * XPAOk(2);
P(25)= qPPek * (uk(2)/(KO2+uk(2))) * (zk(13)/(KPS+zk(13))) * (SALKk(2)/(KALK+SALKk(2)))
* ((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * ((KMAX-(zk(15)/XPAOk(2)))/(KPP+KMAX-
(zk(15)/XPAOk(2)))) * XPAOk(2) * etaNO3 * KO2/uk(2) * (zk(12)/(KNO3+zk(12)));
P(26)= muPAOek * (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) *
(SALKk(2)/(KALK+SALKk(2)))* (zk(13)/(KP+zk(13))) *
((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * XPAOk(2);
P(27)= muPAOek * (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) *
(SALKk(2)/(KALK+SALKk(2)))* (zk(13)/(KP+zk(13))) *
((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * XPAOk(2) * etaNO3 * KO2/uk(2) *
(zk(12)/(KNO3+zk(12)));
P(28)= muAUTek * (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XAk(2);

% TANK 3
P(29)= kHek * (uk(3)/(KO2+uk(3))) * ((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(30)= kHek * etaNO3 * (KO2/(KO2+uk(3))) * (zk(20)/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(31)= kHek * etafe * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(32)= muHek * (uk(3)/(KO2+uk(3))) * (zk(17)/(KF+zk(17))) * (zk(17)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(33)= muHek * (uk(3)/(KO2+uk(3))) * (zk(18)/(KA+zk(18))) * (zk(18)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(34)= muHek * etaNO3 * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(17)/(KF+zk(17)))
* (zk(17)/(zk(17)+zk(18))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(35)= muHek * etaNO3 * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(18)/(KA+zk(18)))
* (zk(18)/(zk(17)+zk(18))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3)))* (zk(21)/(KP+zk(21))) * XHk(3);
P(36)= qfeek * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(17)/(KF+zk(17))) *
(SALKk(3)/(KALK+SALKk(3)))* XHk(3);
P(37)= qPHAek * (zk(18)/(KA+zk(18))) * (SALKk(3)/(KALK+SALKk(3))) *
((zk(23)/XPAOk(3))/(KPP+(zk(23)/XPAOk(3)))) * XPAOk(3);
P(38)= qPPek * (uk(3)/(KO2+uk(3))) * (zk(21)/(KPS+zk(21))) * (SALKk(3)/(KALK+SALKk(3)))
* ((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * ((KMAX-(zk(23)/XPAOk(3)))/(KPP+KMAX-
(zk(23)/XPAOk(3)))) * XPAOk(3);
P(39)= qPPek * (uk(3)/(KO2+uk(3))) * (zk(21)/(KPS+zk(21))) * (SALKk(3)/(KALK+SALKk(3)))
* ((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * ((KMAX-(zk(23)/XPAOk(3)))/(KPP+KMAX-
(zk(23)/XPAOk(3)))) * XPAOk(3) * etaNO3 * KO2/uk(3) * (zk(20)/(KNO3+zk(20)));
P(40)= muPAOek * (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) *
(SALKk(3)/(KALK+SALKk(3)))* (zk(21)/(KP+zk(21))) *
((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * XPAOk(3);
P(41)= muPAOek * (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) *
(SALKk(3)/(KALK+SALKk(3)))* (zk(21)/(KP+zk(21))) *
((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * XPAOk(3) * etaNO3 * KO2/uk(3) *
(zk(20)/(KNO3+zk(20)));
P(42)= muAUTek * (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XAk(3);

% TANK 4
P(43)= kHek * (uk(4)/(KO2+uk(4))) * ((zk(30)/XHk(4))/(KX+(zk(30)/XHk(4)))) * XHk(4);
P(44)= kHek * etaNO3 * (KO2/(KO2+uk(4))) * (zk(28)/(KNO3+zk(28))) *
((zk(30)/XHk(4))/(KX+(zk(30)/XHk(4)))) * XHk(4);
P(45)= kHek * etafe * (KO2/(KO2+uk(4))) * (KNO3/(KNO3+zk(28))) *
((zk(30)/XHk(4))/(KX+(zk(30)/XHk(4)))) * XHk(4);
P(46)= muHek * (uk(4)/(KO2+uk(4))) * (zk(25)/(KF+zk(25))) * (zk(25)/(zk(25)+zk(26))) *
(zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) * (SALKk(4)/(KALK+SALKk(4))) * XHk(4);
P(47)= muHek * (uk(4)/(KO2+uk(4))) * (zk(26)/(KA+zk(26))) * (zk(26)/(zk(25)+zk(26))) *
(zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) * (SALKk(4)/(KALK+SALKk(4))) * XHk(4);
```

```
P(48)= muHek * etaNO3 * (KO2/(KO2+uk(4))) * (KNO3/(KNO3+zk(28))) * (zk(25)/(KF+zk(25)))
* (zk(25)/(zk(25)+zk(26))) * (zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) *
(SALKk(4)/(KALK+SALKk(4))) * XHk(4);
P(49)= muHek * etaNO3 * (KO2/(KO2+uk(4))) * (KNO3/(KNO3+zk(28))) * (zk(26)/(KA+zk(26)))
* (zk(26)/(zk(25)+zk(26))) * (zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) *
(SALKk(4)/(KALK+SALKk(4)))* (zk(29)/(KP+zk(29))) * XHk(4);
P(50)= qfeek * (KO2/(KO2+uk(4))) * (KNO3/(KNO3+zk(28))) * (zk(25)/(KF+zk(25))) *
(SALKk(4)/(KALK+SALKk(4)))* XHk(4);
P(51)= qPHAek * (zk(26)/(KA+zk(26))) * (SALKk(4)/(KALK+SALKk(4))) *
((zk(31)/XPAOk(4))/(KPP+(zk(31)/XPAOk(4)))) * XPAOk(4);
P(52)= qPPek * (uk(4)/(KO2+uk(4))) * (zk(29)/(KPS+zk(29))) * (SALKk(4)/(KALK+SALKk(4)))
* ((zk(32)/XPAOk(4))/(KPHA+(zk(32)/XPAOk(4)))) * ((KMAX-(zk(31)/XPAOk(4)))/(KPP+KMAX-
(zk(31)/XPAOk(4)))) * XPAOk(4);
P(53)= qPPek * (uk(4)/(KO2+uk(4))) * (zk(29)/(KPS+zk(29))) * (SALKk(4)/(KALK+SALKk(4)))
* ((zk(32)/XPAOk(4))/(KPHA+(zk(32)/XPAOk(4)))) * ((KMAX-(zk(31)/XPAOk(4)))/(KPP+KMAX-
(zk(31)/XPAOk(4)))) * XPAOk(4) * etaNO3 * KO2/uk(4) * (zk(28)/(KNO3+zk(28)));
P(54)= muPAOek * (uk(4)/(KO2+uk(4))) * (zk(27)/(KNH4+zk(27))) *
(SALKk(4)/(KALK+SALKk(4)))* (zk(29)/(KP+zk(29))) *
((zk(32)/XPAOk(4))/(KPHA+(zk(32)/XPAOk(4)))) * XPAOk(4);
P(55)= muPAOek * (uk(4)/(KO2+uk(4))) * (zk(27)/(KNH4+zk(27))) *
(SALKk(4)/(KALK+SALKk(4)))* (zk(29)/(KP+zk(29))) *
((zk(32)/XPAOk(4))/(KPHA+(zk(32)/XPAOk(4)))) * XPAOk(4) * etaNO3 * KO2/uk(4) *
(zk(28)/(KNO3+zk(28)));
P(56)= muAUTek * (uk(4)/(KO2+uk(4))) * (zk(27)/(KNH4+zk(27))) * (zk(29)/(KP+zk(29))) *
(SALKk(4)/(KALK+SALKk(4))) * XAk(4);
% TANK 5
P(57)= kHek * (uk(5)/(KO2+uk(5))) * ((zk(38)/XHk(5))/(KX+(zk(38)/XHk(5)))) * XHk(5);
P(58)= kHek * etaNO3 * (KO2/(KO2+uk(5))) * (zk(36)/(KNO3+zk(36))) *
((zk(38)/XHk(5))/(KX+(zk(38)/XHk(5)))) * XHk(5);
P(59)= kHek * etafe * (KO2/(KO2+uk(5))) * (KNO3/(KNO3+zk(36))) *
((zk(38)/XHk(5))/(KX+(zk(38)/XHk(5)))) * XHk(5);
P(60)= muHek * (uk(5)/(KO2+uk(5))) * (zk(33)/(KF+zk(33))) * (zk(33)/(zk(33)+zk(34))) *
(zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) * (SALKk(5)/(KALK+SALKk(5))) * XHk(5);
P(61)= muHek * (uk(5)/(KO2+uk(5))) * (zk(34)/(KA+zk(34))) * (zk(34)/(zk(33)+zk(34))) *
(zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) * (SALKk(5)/(KALK+SALKk(5))) * XHk(5);
P(62)= muHek * etaNO3 * (KO2/(KO2+uk(5))) * (KNO3/(KNO3+zk(36))) * (zk(33)/(KF+zk(33)))
* (zk(33)/(zk(33)+zk(34))) * (zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) *
(SALKk(5)/(KALK+SALKk(5))) * XHk(5);
P(63)= muHek * etaNO3 * (KO2/(KO2+uk(5))) * (KNO3/(KNO3+zk(36))) * (zk(34)/(KA+zk(34)))
* (zk(34)/(zk(33)+zk(34))) * (zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) *
(SALKk(5)/(KALK+SALKk(5)))* (zk(37)/(KP+zk(37))) * XHk(5);
P(64)= qfeek * (KO2/(KO2+uk(5))) * (KNO3/(KNO3+zk(36))) * (zk(33)/(KF+zk(33))) *
(SALKk(5)/(KALK+SALKk(5)))* XHk(5);
P(65)= qPHAek * (zk(34)/(KA+zk(34))) * (SALKk(5)/(KALK+SALKk(5))) *
((zk(39)/XPAOk(5))/(KPP+(zk(39)/XPAOk(5)))) * XPAOk(5);
P(66)= qPPek * (uk(5)/(KO2+uk(5))) * (zk(37)/(KPS+zk(37))) * (SALKk(5)/(KALK+SALKk(5)))
* ((zk(40)/XPAOk(5))/(KPHA+(zk(40)/XPAOk(5)))) * ((KMAX-(zk(39)/XPAOk(5)))/(KPP+KMAX-
(zk(39)/XPAOk(5)))) * XPAOk(5);
P(67)= qPPek * (uk(5)/(KO2+uk(5))) * (zk(37)/(KPS+zk(37))) * (SALKk(5)/(KALK+SALKk(5)))
* ((zk(40)/XPAOk(5))/(KPHA+(zk(40)/XPAOk(5)))) * ((KMAX-(zk(39)/XPAOk(5)))/(KPP+KMAX-
(zk(39)/XPAOk(5)))) * XPAOk(5) * etaNO3 * KO2/uk(5) * (zk(36)/(KNO3+zk(36)));
P(68)= muPAOek * (uk(5)/(KO2+uk(5))) * (zk(35)/(KNH4+zk(35))) *
(SALKk(5)/(KALK+SALKk(5)))* (zk(37)/(KP+zk(37))) *
((zk(40)/XPAOk(5))/(KPHA+(zk(40)/XPAOk(5)))) * XPAOk(5);
P(69)= muPAOek * (uk(5)/(KO2+uk(5))) * (zk(35)/(KNH4+zk(35))) *
(SALKk(5)/(KALK+SALKk(5)))* (zk(37)/(KP+zk(37))) *
((zk(40)/XPAOk(5))/(KPHA+(zk(40)/XPAOk(5)))) * XPAOk(5) * etaNO3 * KO2/uk(5) *
(zk(36)/(KNO3+zk(36)));
P(70)= muAUTek * (uk(5)/(KO2+uk(5))) * (zk(35)/(KNH4+zk(35))) * (zk(37)/(KP+zk(37))) *
(SALKk(5)/(KALK+SALKk(5))) * XAk(5);


Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20) P(21) P(22) P(23) P(24) P(25) P(26) P(27) P(28) P(29)
P(30) P(31) P(32) P(33) P(34) P(35) P(36) P(37) P(38) P(39) P(40) P(41) P(42) P(43)
P(44) P(45) P(46) P(47) P(48) P(49) P(50) P(51) P(52) P(53) P(54) P(55) P(56) P(57)
P(58) P(59) P(60) P(61) P(62) P(63) P(64) P(65) P(66) P(67) P(68) P(69) P(70)]';
```

APPENDIX D: DEVELOPED MATLAB CODE FOR OPTIMAL CONTROL OF THE BENCHMARK PLANT WITH THE ACTIVATED SLUDGE MODEL NO.3 BIO-MODEL

D.1: MATLAB script file – BASM3_Simulation.m

```
% M-File: BASM3_Simulation.m
% ======================== M-FILE DESCRIPTION ========================
% This m-file is used for:
% Simulation of the Benchmark plant using full ASM3 model
% Calculation of the parameters for the reduced ASM3 model
% Simulation of the Benchmark plant using the reduced ASM3 model


clear all         % Clear the workspace
clc               % Clear the command window


% ======================== PARAMETER DEFINITION ========================
% Declaration of the global variables
global kH kSTO etaNOX muH bH_O2 bH_NOX bSTO_O2 bSTO_NOX muA bA_O2 bA_NOX
global KX KO2 KS KNOX KNH4 KALK KSTO KA_O2 KA_NH4 KA_ALK KA_NOX

K1 = 95969;             % Prediction horizon or interval
DT = 0.001042;          % Sampling time

% Initialization of the model parameters, typical values at 20 OC
% Stochiometric parameters
fSI       = 0.0;
YSTO_O2   = 0.85;
YSTO_NOX  = 0.80;
YH_O2     = 0.63;
YH_NOX    = 0.54;
YA        = 0.24;
fI        = 0.2;
% Kinetic parameters
kH        = 3;
kSTO      = 5;
etaNOX    = 0.6;
muH       = 2;
bH_O2     = 0.2;
bH_NOX    = 0.1;
bSTO_O2   = 0.2;
bSTO_NOX  = 0.1;
muA       = 1;
bA_O2     = 0.15;
bA_NOX    = 0.05;
% Monod switching functions
KX        = 1;
KO2       = 0.2;
KS        = 2;
KNOX      = 0.5;
KNH4      = 0.01;
KALK      = 0.1;
KSTO      = 1;
KA_O2     = 0.5;
KA_NH4    = 1;
KA_ALK    = 0.5;
KA_NOX    = 0.5;


% Initialization of the Benchmark process data
V1    = 1000;           % Tank 1 volume
V2    = 1000;           % Tank 2 volume
V3    = 1333;           % Tank 3 volume
V4    = 1333;           % Tank 4 volume
V5    = 1333;           % Tank 5 volume

Q0    = 18446;          % Influent flow rate
Qa    = 55338;          % Internal recirculation rate
```

```
Qr    = 18446;         % Sludge recycled rate
Qw    = 385;           % Wasted sludge rate
Q     = Q0 + Qa + Qr;% Flow through the tanks

% Coeficient for fraction of flow rates for the particulate materials
LAMBDA = (Q0+Qr)/(Qr+Qw);


% Oxygen parameters
SOsat = 8;             % Oxygen saturation point
KLa1  = 0;             % Oxygen transfer function in Tank 1
KLa2  = 0;             % Oxygen transfer function in Tank 2
KLa3  = 240;           % Oxygen transfer function in Tank 3
KLa4  = 240;           % Oxygen transfer function in Tank 4
KLa5  = 84;            % Oxygen transfer function in Tank 5


KL = [KLa1 KLa2 KLa3 KLa4 KLa5]; KLa = diag(KL);


% =========================== PROCESS MODEL ===========================
% Calculation of the MATRIX A
% Matrix A represent weighting factors of the accumulated state variables
% from different flows towards state variables in each tank as derived
% through mass balance principles, see chapter 3 of the Thesis
IV1 = DT/ V1; IV11 = IV1 * Q; A11 = 1 - IV11;
IV2 = DT/ V2; IV22 = IV2 * Q; A22 = 1 - IV22;
IV3 = DT/ V3; IV33 = IV3 * Q; A33 = 1 - IV33;
IV4 = DT/ V4; IV44 = IV4 * Q; A44 = 1 - IV44;
IV5 = DT/ V5; IV55 = IV5 * Q; A55 = 1 - IV55;


A15a = IV1 * (Qa + Qr);        % Soluble materials recycled from Tank 5
A15b = IV1 * (Qa + LAMBDA*Qr);% Particulate materials recycled from Tank 5
A21 = IV22;
A32 = IV33;
A43 = IV44;
A54 = IV55;


A11 = A11 * ones(1,13); A11 = diag(A11);
A22 = A22 * ones(1,13); A22 = diag(A22);
A33 = A33 * ones(1,13); A33 = diag(A33);
A44 = A44 * ones(1,13); A44 = diag(A44);
A55 = A55 * ones(1,13); A55 = diag(A55);


A15 = [A15a,A15a,A15a,A15a,A15a,A15a,A15a,A15b,A15b,A15b,A15b,A15b,A15b];
A15 = diag(A15);
A21 = A21 * ones(1,13); A21 = diag(A21);
A32 = A32 * ones(1,13); A32 = diag(A32);
A43 = A43 * ones(1,13); A43 = diag(A43);
A54 = A54 * ones(1,13); A54 = diag(A54);


A10 = zeros(13,13); A20 = zeros(13,26); A30 = zeros(13,39);

A = [A11 A30 A15; A21 A22 A30; A10 A32 A33 A20; A20 A43 A44 A10; A30 A54 A55];

% Calculation of the MATRIX B
% Matrix B represent weighting factors of the state variables towards the
% influent wastewater composition as derived through mass balance
% principles, see chapter 3 of the Thesis
IV0 = DT/ V1*Q0;
B11 = IV0 * ones(1,13); B11 = diag(B11);
B = [B11; zeros(52,13)];


% Formulation of the MATRIX C
% Matrix C represent weighting factors of the state variables towards the
% processes on the ASM1 matrix table
CC = [[0   fSI   1    0.01    0    0    0.001  0  -1 0   0         0  -0.75];
      [-0.15  0  -1   0.03    0    0    0.002  0   0 0   YSTO_O2   0   0.51 ];
      [0     0  -1   0.03  0.07  -0.07  0.007  0   0 0   YSTO_NOX  0   0.48 ];
```

251

```
      [-0.60  0 0 -0.07 0      0      -0.005 0  0  1   -1/YH_O2 0  -0.06];
      [0     0 0 -0.07  0.30  -0.30  0.016  0  0  1   -1/YH_NOX 0 -0.21];
      [-0.80 0 0 0.066 0      0       0.005  fI 0 -1  0        0  -0.75];
      [0     0 0 0.066  0.28  -0.28   0.025  fI 0 -1  0        0  -0.75];
      [-1    0 0 0      0      0       0      0  0  0  -1       0  -0.60];
      [0     0 0 0      0.35  -0.35    0.025  0  0  0  -1       0  -0.60];
      [-18.04 0 0 -4.24 0      1/YA   -0.60  0  0  0  0        1   0.90 ];
      [-0.80 0 0 0.066  0      0       0.005  fI 0  0  0        -1 -0.75];
      [0     0 0 0.066  0.28  -0.28   0.025  fI 0  0  0        -1 -0.75]];

   C = DT*[CC zeros(12,52); zeros(12,13) CC zeros(12,39);
           zeros(12,26) CC zeros(12,26); zeros(12,39) CC zeros(12,13);
           zeros(12,52) CC];

% ================================= SIMULATION DATA =============================
% Constant inflow concentrations for steady state simulations
SO2i =  0.0;SIi = 30.0;SSi = 69.5;SNH4i = 31.56;SN2i = 6.95;SNOXi = 0.0;
SALKi= 7.0;XIi = 51.2;XSi= 202.32;XHi = 28.17;XSTOi=0;XAi = 0.0;XSSi = 125;


% Order of the states
%SO2   SI    SS    SNH4  SN2   SNOX   SALK  XI XS    XH   XSTO XA XSS
xi = [SO2i*ones(1,K1); SIi*ones(1,K1)/0.1;  SSi*ones(1,K1);  SNH4i*ones(1,K1);
      SN2i*ones(1,K1); SNOXi*ones(1,K1);SALKi*ones(1,K1);XIi*ones(1,K1);
      XSi*ones(1,K1);  XHi*ones(1,K1);  XSTOi*ones(1,K1);XAi*ones(1,K1);
      XSSi*ones(1,K1)]*0.1;


% Values of the initial conditions
% For tank 1
SO2_1  = 4.2984433e-03; SI_1   = 3.0000000e+01; SS_1   = 2.8082131e+00;
SNH4_1 = 7.9178845e+00; SN2_1  = 1.2166405e+00; SNOX_1 = 5.3699400e+00;
SALK_1 = 4.9277103e+00; XI_1   = 1.1491252e+03; XS_1   = 8.2134908e+01;
XH_1   = 2.5517658e+03; XSTO_1 = 4.4885186e+00; XA_1   = 1.4838943e+02;
XSS_1  = 5.2848894e+00;
x10 = [SO2_1; SI_1; SS_1; SNH4_1; SN2_1; SNOX_1; SALK_1; XI_1; XS_1;
       XH_1; XSTO_1; XA_1;XSS_1 ]*0.1;


% For tank 2
SO2_2  = 6.3131911e-05; SI_2   = 3.0000000e+01; SS_2   = 1.4587940e+00;
SNH4_2 = 8.3444148e+00; SN2_2  = 8.8206477e-01; SNOX_2 = 3.6619672e+00;
SALK_2 = 5.0801748e+00; XI_2   = 1.1491252e+03; XS_2   = 7.6386187e+01;
XH_2   = 2.5533851e+03; XSTO_2 = 4.4952273e+00; XA_2   = 1.4830914e+02;
XSS_2  = 5.0290873e+00;
x20 = [SO2_2; SI_2; SS_2; SNH4_2; SN2_2; SNOX_2; SALK_2; XI_2; XS_2;
       XH_2; XSTO_2; XA_2;XSS_2 ]*0.1;


% For tank 3
SO2_3  = 1.7183778e+00; SI_3   = 3.0000000e+01; SS_3   = 1.1495418e+00;
SNH4_3 = 5.5479452e+00; SN2_3  = 8.2888682e-01; SNOX_3 = 6.5408820e+00;
SALK_3 = 4.6747902e+00; XI_3   = 1.1491252e+03; XS_3   = 6.4854922e+01;
XH_3   = 2.5571314e+03; XSTO_3 = 4.3924277e+00; XA_3   = 1.4894126e+02;
XSS_3  = 4.5041834e+00;
x30 = [SO2_3; SI_3; SS_3; SNH4_3; SN2_3; SNOX_3; SALK_3; XI_3; XS_3;
       XH_3; XSTO_3; XA_3;XSS_3 ]*0.1;


% For tank 4
SO2_4  = 2.4288838e+00; SI_4   = 3.0000000e+01; SS_4   = 9.9532389e-01;
SNH4_4 = 2.9673854e+00; SN2_4  = 7.6678656e-01; SNOX_4 = 9.2989988e+00;
SALK_4 = 4.2934562e+00; XI_4   = 1.1491252e+03; XS_4   = 5.5693982e+01;
XH_4   = 2.5591826e+03; XSTO_4 = 3.8790101e+00; XA_4   = 1.4952712e+02;
XSS_4  = 4.5131469e+00;
x40 = [SO2_4; SI_4; SS_4; SNH4_4; SN2_4; SNOX_4; SALK_4; XI_4; XS_4;
       XH_4; XSTO_4; XA_4;XSS_4 ]*0.1;


% For tank 5
SO2_5  = 4.9094351e-01; SI_5   = 3.0000000e+01; SS_5   = 8.8949280e-01;
SNH4_5 = 1.7333316e+00; SN2_5  = 6.8828001e-01; SNOX_5 = 1.0415220e+01;
```

```
SALK_5 = 4.1255794e+00; XI_5    = 1.1491252e+03; XS_5    = 4.9305586e+01;
XH_5    = 2.5593436e+03; XSTO_5 = 3.5271755e+00; XA_5    = 1.4979714e+02;
XSS_5 = 4.5221112e+00;
x50 = [SO2_5; SI_5; SS_5; SNH4_5; SN2_5; SNOX_5; SALK_5; XI_5; XS_5;
       XH_5; XSTO_5; XA_5;XSS_5 ]*0.1;


x0 = [x10;x20;x30;x40;x50];
x = [x0 zeros(65,K1)];


% Steady state simulation of the model
for k = 1:K1
    xk  = x(:,k);
    xik = xi(:,k);
    Pk  = P_BASM3F(xk);    ..

    SOk  = [xk(1,1);xk(14,1);xk(27,1);xk(40,1);xk(53,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xx1k =
[xxk(1);zeros(12,1);xxk(2);zeros(12,1);xxk(3);zeros(12,1);xxk(4);zeros(12,1);xxk(5);zer
os(12,1)];
    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xx1k;
end


% Full model response to constant influent concentrations
% Tank 1
figure(1)
subplot(5,3,1);plot(x(1,:));title('SO2_1')
subplot(5,3,2);plot(x(2,:));title('SI1')
subplot(5,3,3);plot(x(3,:));title('SS1')
subplot(5,3,4);plot(x(4,:));title('SNH4_1')
subplot(5,3,5);plot(x(5,:));title('SN2_1')
subplot(5,3,6);plot(x(6,:));title('SNOX1')
subplot(5,3,7);plot(x(7,:));title('SALK1')
subplot(5,3,8);plot(x(8,:));title('XI1')
subplot(5,3,9);plot(x(9,:));title('XS1')
subplot(5,3,10);plot(x(10,:));title('XH1')
subplot(5,3,11);plot(x(11,:));title('XSTO1')
xlabel('discrete time k')
subplot(5,3,12);plot(x(12,:));title('XA1')
xlabel('discrete time k')
subplot(5,3,13);plot(x(13,:));title('XSS1')
xlabel('discrete time k')
set(figure(1),'name','Tank 1 steady state results','numbertitle','off')


% Tank 2
figure(2)
subplot(5,3,1);plot(x(14,:));title('SO2_2')
subplot(5,3,2);plot(x(15,:));title('SI2')
subplot(5,3,3);plot(x(16,:));title('SS2')
subplot(5,3,4);plot(x(17,:));title('SNH4_2')
subplot(5,3,5);plot(x(18,:));title('SN2_2')
subplot(5,3,6);plot(x(19,:));title('SNOX2')
subplot(5,3,7);plot(x(20,:));title('SALK2')
subplot(5,3,8);plot(x(21,:));title('XI2')
subplot(5,3,9);plot(x(22,:));title('XS2')
subplot(5,3,10);plot(x(23,:));title('XH2')
subplot(5,3,11);plot(x(24,:));title('XSTO2')
xlabel('discrete time k')
subplot(5,3,12);plot(x(25,:));title('XA2')
xlabel('discrete time k')
subplot(5,3,13);plot(x(26,:));title('XSS2')
xlabel('discrete time k')
set(figure(2),'name','Tank 2 steady state results','numbertitle','off')


% Tank 3
figure(3)
```

253

```matlab
subplot(5,3,1);plot(x(27,:));title('SO2_3')
subplot(5,3,2);plot(x(28,:));title('SI3')
subplot(5,3,3);plot(x(29,:));title('SS3')
subplot(5,3,4);plot(x(30,:));title('SNH4_3')
subplot(5,3,5);plot(x(31,:));title('SN2_3')
subplot(5,3,6);plot(x(32,:));title('SNOX3')
subplot(5,3,7);plot(x(33,:));title('SALK3')
subplot(5,3,8);plot(x(34,:));title('XI3')
subplot(5,3,9);plot(x(35,:));title('XS3')
subplot(5,3,10);plot(x(36,:));title('XH3')
subplot(5,3,11);plot(x(37,:));title('XSTO3')
xlabel('discrete time k')
subplot(5,3,12);plot(x(38,:));title('XA3')
xlabel('discrete time k')
subplot(5,3,13);plot(x(39,:));title('XSS3')
xlabel('discrete time k')
set(figure(3),'name','Tank 3 steady state results','numbertitle','off')

% Tank 4
figure(4)
subplot(5,3,1);plot(x(40,:));title('SO2_4')
subplot(5,3,2);plot(x(41,:));title('SI4')
subplot(5,3,3);plot(x(42,:));title('SS4')
subplot(5,3,4);plot(x(43,:));title('SNH4_4')
subplot(5,3,5);plot(x(44,:));title('SN2_4')
subplot(5,3,6);plot(x(45,:));title('SNOX4')
subplot(5,3,7);plot(x(46,:));title('SALK4')
subplot(5,3,8);plot(x(47,:));title('XI4')
subplot(5,3,9);plot(x(48,:));title('XS4')
subplot(5,3,10);plot(x(49,:));title('XH4')
subplot(5,3,11);plot(x(50,:));title('XSTO4')
xlabel('discrete time k')
subplot(5,3,12);plot(x(51,:));title('XA4')
xlabel('discrete time k')
subplot(5,3,13);plot(x(52,:));title('XSS4')
xlabel('discrete time k')
set(figure(4),'name','Tank 4 steady state results','numbertitle','off')

% Tank 5
figure(5)
subplot(5,3,1);plot(x(53,:));title('SO2_5')
subplot(5,3,2);plot(x(54,:));title('SI5')
subplot(5,3,3);plot(x(55,:));title('SS5')
subplot(5,3,4);plot(x(56,:));title('SNH4_5')
subplot(5,3,5);plot(x(57,:));title('SN2_5')
subplot(5,3,6);plot(x(58,:));title('SNOX5')
subplot(5,3,7);plot(x(59,:));title('SALK5')
subplot(5,3,8);plot(x(60,:));title('XI5')
subplot(5,3,9);plot(x(61,:));title('XS5')
subplot(5,3,10);plot(x(62,:));title('XH5')
subplot(5,3,11);plot(x(63,:));title('XSTO5')
xlabel('discrete time k')
subplot(5,3,12);plot(x(64,:));title('XA5')
xlabel('discrete time k')
subplot(5,3,13);plot(x(65,:));title('XSS5')
xlabel('discrete time k')
set(figure(5),'name','Tank 5 steady state results','numbertitle','off')

% Data for dynamic simulations
K = 13436;   % Prediction horizon for dynamic simulations (14 days sampled 0.001042)
DT  = 0.001042;

% Dynamics of the inflow concentrations for dry weather file
load dry.txt    % The order is t, SS XBH XS XI SNH SI SND YND Q0
%                                 1  2   3  4  5   6  7   8  9  10
% ======================== SIMULATION DATA =============================
```

254

```
% Constant inflow concentrations for steady state simulations
% SO2 SI SS SNH4 SN2 SNOX SALK XI XS XH XSTO XA XSS

t    = 1344;
SO2i = SO2i*zeros(1,t); SIi = dry(1:t,7)'; SSi = dry(1:t,2)'; SNH4i = dry(1:t,6)';
SN2i = dry(1:t,8)';SNOXi = SNOXi*zeros(1,t); SALKi = SALKi*ones(1,t);
XIi = dry(1:t,5)';XSi = dry(1:t,4)';XHi = dry(1:t,3)';
XSTOi = XSTOi*zeros(1,t);  XAi = XAi*zeros(1,t); XSSi =XSSi* ones(1,t);

t1 = [dry(:,1)']; t2 = [0:0.001042:14];

SO2i = spline(t1,SO2i,t2);   SIi = spline(t1,SIi,t2);   SSi = spline(t1,SSi,t2);
SNH4i = spline(t1,SNH4i,t2);  SN2i = spline(t1,SN2i,t2); SNOXi = spline(t1,SNOXi,t2);
SALKi = spline(t1,SALKi,t2);   XIi = spline(t1,XIi,t2);   XSi = spline(t1,XSi,t2);
XHi = spline(t1,XHi,t2); XSTOi = spline(t1,XSTOi,t2); XAi = spline(t1,XAi,t2);
XSSi = spline(t1,XSSi,t2);

xi = [SO2i;SIi;SSi;SNH4i;SN2i;SNOXi;SALKi;XIi;XSi;XHi;XSTOi;XAi;XSSi]*0.1;
x = [x(:,(K1+1)) zeros(65,K)];

% Dynamic simulation of the model
for k = 1:K
    xk   = x(:,k);
    xik  = xi(:,k);
    Pk   = P_BASM3F(xk);  % Calculation of the process rates

    SOk  = [xk(1,1);xk(14,1);xk(27,1);xk(40,1);xk(53,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xxlk =
[xxk(1);zeros(12,1);xxk(2);zeros(12,1);xxk(3);zeros(12,1);xxk(4);zeros(12,1);xxk(5);zer
os(12,1)];
    x(:,k+1)  = A*x(:,k) + C'*Pk + B*xi(:,k) + xxlk;
end

% Full model response to the dynamic influent concentrations
% Tank 1
figure(6)
subplot(5,3,1);plot(x(1,:));title('SO2_1')
subplot(5,3,2);plot(x(2,:));title('SI1')
subplot(5,3,3);plot(x(3,:));title('SS1')
subplot(5,3,4);plot(x(4,:));title('SNH4_1')
subplot(5,3,5);plot(x(5,:));title('SN2_1')
subplot(5,3,6);plot(x(6,:));title('SNOX1')
subplot(5,3,7);plot(x(7,:));title('SALK1')
subplot(5,3,8);plot(x(8,:));title('XI1')
subplot(5,3,9);plot(x(9,:));title('XS1')
subplot(5,3,10);plot(x(10,:));title('XH1')
subplot(5,3,11);plot(x(11,:));title('XSTO1')
xlabel('discrete time k')
subplot(5,3,12);plot(x(12,:));title('XA1')
xlabel('discrete time k')
subplot(5,3,13);plot(x(13,:));title('XSS1')
xlabel('discrete time k')
set(figure(6),'name','Tank 1 dynamic results','numbertitle','off')

% Tank 2
figure(7)
subplot(5,3,1);plot(x(14,:));title('SO2_2')
subplot(5,3,2);plot(x(15,:));title('SI2')
subplot(5,3,3);plot(x(16,:));title('SS2')
subplot(5,3,4);plot(x(17,:));title('SNH4_2')
subplot(5,3,5);plot(x(18,:));title('SN2_2')
subplot(5,3,6);plot(x(19,:));title('SNOX2')
subplot(5,3,7);plot(x(20,:));title('SALK2')
subplot(5,3,8);plot(x(21,:));title('XI2')
subplot(5,3,9);plot(x(22,:));title('XS2')
```

```
subplot(5,3,10);plot(x(23,:));title('XH2')
subplot(5,3,11);plot(x(24,:));title('XSTO2')
xlabel('discrete time k')
subplot(5,3,12);plot(x(25,:));title('XA2')
xlabel('discrete time k')
subplot(5,3,13);plot(x(26,:));title('XSS2')
xlabel('discrete time k')
set(figure(7),'name','Tank 2 dynamic results','numbertitle','off')


% Tank 3
figure(8)
subplot(5,3,1);plot(x(27,:));title('SO2_3')
subplot(5,3,2);plot(x(28,:));title('SI3')
subplot(5,3,3);plot(x(29,:));title('SS3')
subplot(5,3,4);plot(x(30,:));title('SNH4_3')
subplot(5,3,5);plot(x(31,:));title('SN2_3')
subplot(5,3,6);plot(x(32,:));title('SNOX3')
subplot(5,3,7);plot(x(33,:));title('SALK3')
subplot(5,3,8);plot(x(34,:));title('XI3')
subplot(5,3,9);plot(x(35,:));title('XS3')
subplot(5,3,10);plot(x(36,:));title('XH3')
subplot(5,3,11);plot(x(37,:));title('XSTO3')
xlabel('discrete time k')
subplot(5;3,12);plot(x(38,:));title('XA3')
xlabel('discrete time k')
subplot(5,3,13);plot(x(39,:));title('XSS3')
xlabel('discrete time k')
set(figure(8),'name','Tank 3 dynamic results','numbertitle','off')


% Tank 4
figure(9)
subplot(5,3,1);plot(x(40,:));title('SO2_4')
subplot(5,3,2);plot(x(41,:));title('SI4')
subplot(5,3,3);plot(x(42,:));title('SS4')
subplot(5,3,4);plot(x(43,:));title('SNH4_4')
subplot(5,3,5);plot(x(44,:));title('SN2_4')
subplot(5,3,6);plot(x(45,:));title('SNOX4')
subplot(5,3,7);plot(x(46,:));title('SALK4')
subplot(5,3,8);plot(x(47,:));title('XI4')
subplot(5,3,9);plot(x(48,:));title('XS4')
subplot(5,3,10);plot(x(49,:));title('XH4')
subplot(5,3,11);plot(x(50,:));title('XSTO4')
xlabel('discrete time k')
subplot(5,3,12);plot(x(51,:));title('XA4')
xlabel('discrete time k')
subplot(5,3,13);plot(x(52,:));title('XSS4')
xlabel('discrete time k')
set(figure(9),'name','Tank 4 dynamic results','numbertitle','off')


% Tank 5
figure(10)
subplot(5,3,1);plot(x(53,:));title('SO2_5')
subplot(5,3,2);plot(x(54,:));title('SI5')
subplot(5,3,3);plot(x(55,:));title('SS5')
subplot(5,3,4);plot(x(56,:));title('SNH4_5')
subplot(5,3,5);plot(x(57,:));title('SN2_5')
subplot(5,3,6);plot(x(58,:));title('SNOX5')
subplot(5,3,7);plot(x(59,:));title('SALK5')
subplot(5,3,8);plot(x(60,:));title('XI5')
subplot(5,3,9);plot(x(61,:));title('XS5')
subplot(5,3,10);plot(x(62,:));title('XH5')
subplot(5,3,11);plot(x(63,:));title('XSTO5')
xlabel('discrete time k')
subplot(5,3,12);plot(x(64,:));title('XA5')
xlabel('discrete time k')
subplot(5,3,13);plot(x(65,:));title('XSS5')
```

256

```
xlabel('discrete time k')
set(figure(10),'name','Tank 5 dynamic results','numbertitle','off')

% ========================= REDUCED MODEL FORMULATION =========================
% Form the vector of the states (z) for the reduced model, based on the
% ASM1 prediction
%        SS,          SNH4,        SNOX,
z1 = [x(3,:);  x(4,:);   x(6,:) ];    %1      2      3
z2 = [x(16,:); x(17,:);  x(19,:)];    %4      5      6
z3 = [x(29,:); x(30,:);  x(32,:)];    %7      8      9
z4 = [x(42,:); x(43,:);  x(45,:)];    %10     11     12
z5 = [x(55,:); x(56,:);  x(58,:)];    %13     14     15
z  = [z1;z2;z3;z4;z5];

% Formation of the vector of the control variables
u1 = x(1,:);
u2 = x(14,:);
u3 = x(27,:);
u4 = x(40,:);
u5 = x(53,:);
u  = [u1; u2; u3; u4; u5];

% Dynamic values of the inflow concentrations for dry weather
zi  =[xi(3,:); xi(4,:); xi(6,:)]*0.1;
% Steady state values for the biomass XH, XA, XS, SALK, XSTO
XH   = [160*ones(1,K); 160*ones(1,K); 161*ones(1,K);
        162*ones(1,K); 160*ones(1,K)];
XA   = [6.5*ones(1,K); 6.6*ones(1,K); 6.7*ones(1,K);
        6*ones(1,K); 6.3*ones(1,K)];
XS   = [20*ones(1,K); 20*ones(1,K); 21*ones(1,K);
        20*ones(1,K); 21*ones(1,K)];
XSTO = [26*ones(1,K); 26*ones(1,K); 26*ones(1,K);
        25*ones(1,K); 25*ones(1,K)];
SALK = [0.5*ones(1,K); 0.4*ones(1,K); 0.5*ones(1,K);
        0.4*ones(1,K); 0.5*ones(1,K)];

% MATRIX A for the reduced model (AR)
A11r = 1 - IV11; A22r = 1 - IV22; A33r = 1 - IV33;
A44r = 1 - IV44; A55r = 1 - IV55;
A15r = IV1 * (Qa + Qr);
A21r = IV22; A32r = IV33; A43r = IV44; A54r = IV55;

A11r = A11r * ones(1,3); A11r = diag(A11r);
A22r = A22r * ones(1,3); A22r = diag(A22r);
A33r = A33r * ones(1,3); A33r = diag(A33r);
A44r = A44r * ones(1,3); A44r = diag(A44r);
A55r = A55r * ones(1,3); A55r = diag(A55r);

A15r = [A15r, A15r, A15r];
A15r = diag(A15r);

A21r = A21r * ones(1,3); A21r = diag(A21r);
A32r = A32r * ones(1,3); A32r = diag(A32r);
A43r = A43r * ones(1,3); A43r = diag(A43r);
A54r = A54r * ones(1,3); A54r = diag(A54r);

A10r = zeros(3,3); A20r = zeros(3,6); A30r = zeros(3,9);

AR = [A11r A30r A15r;A21r A22r A30r;A10r A32r A33r A20r;A20r A43r A44r A10r;A30r A54r
A55r];

% MATRIX B for the reduced model (BR)
IV0 = DT / V1*Q0;
f = 1.2;
B11r = [IV0 IV0 IV0*f];
B11r = diag(B11r);
```

257

```
    BR = [B11r; zeros(12,3)];

    % MATRIX C for the reduced model (CR)
    CCR=[[1  0.01     0     ];
         [0  -0.07    0     ];
         [0  -0.07   -0.30];
         [0  -4.24   1/YA]];

    CR = DT*[CCR zeros(4,12);zeros(4,3) CCR zeros(4,9);zeros(4,6) CCR zeros(4,6);
            zeros(4,9) CCR zeros(4,3);zeros(4,12) CCR];

    % Parameter estimation
    for k = 1:K
        zk = z(:,k);
        uk = u(:,k);

        XHk  = XH(:,k);
        XAk  = XA(:,k);
        XSk  = XS(:,k);
        XSTOk = XSTO(:,k);
        SALKk = SALK(:,k);

        zik  = zi(:,k);
        Pk = P_BASM3R(zk,uk,XHk,XAk,XSk,XSTOk,SALKk);
        FIk = CR'*Pk;
        Y(:,k) = z(:,k+1)-AR*z(:,k)-BR*zi(:,k);
        theta(:,k)=(FIk'*FIk)^(-1)*FIk'*Y(:,k);
    end

    muHE(1,:) = theta(1,:);
    muAE(1,:) = theta(2,:);
    KhE(1,:)  = theta(3,:);

    % Dynamic simulation of the reduced model with the estimated parameters
    zE = [z(1,1);z(2,1);z(3,1);z(4,1);z(5,1);z(6,1);z(7,1);z(8,1);z(9,1);z(10,1); z(11,1);
    z(12,1);z(13,1); z(14,1); z(15,1)];
    ERR = z(:,1)-zE;

    % Dynamic simulation of the reduced model with the estimated parameters
    for k = 1:K
        zEk = zE(:,k);
        uk = u(:,k);
        zik = zi(:,k);

        XHk  = XH(:,k);
        XAk  = XA(:,k);
        XSk  = XS(:,k);
        XSTOk = XSTO(:,k);
        SALKk = SALK(:,k);

        muHEk = muHE(:,k);
        muAEk = muAE(:,k);
        KhEk  = KhE(:,k);

        Pk = P_BASM3RE(zEk,uk,XHk,XAk,XSk,XSTOk,SALKk,muHEk,muAEk,KhEk);
        zE(:,k+1) = AR*zE(:,k) + CR'*Pk + BR*zi(:,k);
        ERR(:,k+1)=z(:,k+1)-zE(:,k+1);
    end

    % Vector of process full, estimated and error variables SS, SNH4, SNOX, Tank 1
    SS1=z(1,:);SNH4_1=z(2,:);SNOX1=z(3,:);
    SS1E=zE(1,:);SNH4_1E=zE(2,:);SNOX1E=zE(3,:);
    ERRSS1=ERR(1,:);ERRSNH4_1=ERR(2,:);ERRSNOX1=ERR(3,:);
    % Calculation of the weighted errors for tank1
    eSS1=(ERRSS1*ERRSS1')/(SS1*SS1');
    eSNH4_1=(ERRSNH4_1*ERRSNH4_1')/(SNH4_1*SNH4_1');
```

```
eSNOX1=(ERRSNOX1*ERRSNOX1')/(SNOX1*SNOX1');

% Vector of process full, estimated and error variables SS, SNH4,SNOX, Tank 2
SS2=z(4,:);SNH4_2=z(5,:);SNOX2=z(6,:);
SS2E=zE(4,:);SNH4_2E=zE(5,:);SNOX2E=zE(6,:);
ERRSS2=ERR(4,:);ERRSNH4_2=ERR(5,:);ERRSNOX2=ERR(6,:);
% Calculation of the weighted errors for tank 2
eSS2=(ERRSS2*ERRSS2')/(SS2*SS2');
eSNH4_2=(ERRSNH4_2*ERRSNH4_2')/(SNH4_2*SNH4_2');
eSNOX2=(ERRSNOX2*ERRSNOX2')/(SNOX2*SNOX2');

% Vector of process full, estimated and error variables SS, SNH4,SNOX, Tank 3
SS3=z(7,:);SNH4_3=z(8,:);SNOX3=z(9,:);
SS3E=zE(7,:);SNH4_3E=zE(8,:);SNOX3E=zE(9,:);
ERRSS3=ERR(7,:);ERRSNH4_3=ERR(8,:);ERRSNOX3=ERR(9,:);
% Calculation of the weighted errors for tank 3
eSS3=(ERRSS3*ERRSS3')/(SS3*SS3');
eSNH4_3=(ERRSNH4_3*ERRSNH4_3')/(SNH4_3*SNH4_3');
eSNOX3=(ERRSNOX3*ERRSNOX3')/(SNOX3*SNOX3');

% Vector of process full, estimated and error variables SS, SNH4,SNOX, Tank 4
SS4=z(10,:);SNH4_4=z(11,:);SNOX4=z(12,:);
SS4E=zE(10,:);SNH4_4E=zE(11,:);SNOX4E=zE(12,:);
ERRSS4=ERR(10,:);ERRSNH4_4=ERR(11,:);ERRSNOX4=ERR(12,:);
% Calculation of the weighted errors for tank 4
eSS4=(ERRSS4*ERRSS4')/(SS4*SS4');
eSNH4_4=(ERRSNH4_4*ERRSNH4_4')/(SNH4_4*SNH4_4');
eSNOX4=(ERRSNOX4*ERRSNOX4')/(SNOX4*SNOX4');

% Vector of process full, estimated and error variables SS, SNH4,SNOX, Tank 5
SS5=z(13,:);SNH4_5=z(14,:);SNOX5=z(15,:);
SS5E=zE(13,:);SNH4_5E=zE(14,:);SNOX5E=zE(15,:);
ERRSS5=ERR(13,:);ERRSNH4_5=ERR(14,:);ERRSNOX5=ERR(15,:);
% Calculation of the weighted errors for tank 5
eSS5=(ERRSS5*ERRSS5')/(SS5*SS5');
eSNH4_5=(ERRSNH4_5*ERRSNH4_5')/(SNH4_5*SNH4_5');
eSNOX5=(ERRSNOX5*ERRSNOX5')/(SNOX5*SNOX5');

k = 1:K;
% Graphs of the estimated parameters
figure(11)
subplot(3,1,1);plot(k,muHE,'k');title('Estimated muH');ylabel('muHE');
subplot(3,1,2);plot(k,muAE,'k');title('Estimated muA');ylabel('muAE');
subplot(3,1,3);plot(k,KhE,'k');title('Estimated Kh');ylabel('KhE');
set(figure(11),'name','Estimated parameters','numbertitle','off')
xlabel('discrete time k')

k = 1:K+1;
% Graphs of the reduced model response to the dynamic influent
% concentrations
% Tank 1

figure(12)
subplot(3,2,1);plot(k,SS1); hold on;plot(k,SS1E,'k-.');ylabel('SS1,SS1E');
legend('SS1','SS1E');subplot(3,2,2); plot(k,ERRSS1,'k');ylabel('ERRSS1');

subplot(3,2,3);plot(k,SNH4_1); hold on;plot(k,SNH4_1E,'k-.');
legend('SNH4_1','SNH4_1E');ylabel('SNH4_1,SNH4_1E');
subplot(3,2,4); plot(k,ERRSNH4_1,'k');ylabel('ERRSNH4_1');

subplot(3,2,5);plot(k,SNOX1); hold on;plot(k,SNOX1E,'k-.');
legend('SNOX1','SNOX1E');ylabel('SNOX1,SNOX1E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNOX1,'k');ylabel('ERRSNOX1');xlabel('discrete time k')
set(figure(12),'name','BASM3 Tank 1 Process variables and estimation
error','numbertitle','off')
```

```
% Tank 2
figure(13)
subplot(3,2,1);plot(k,SS2); hold on;plot(k,SS2E,'k-.');ylabel('SS2,SS2E');
legend('SS2','SS2E');subplot(3,2,2); plot(k,ERRSS2,'k');ylabel('ERRSS2');

subplot(3,2,3);plot(k,SNH4_2); hold on;plot(k,SNH4_2E,'k-.');
legend('SNH4_2','SNH4_2E');ylabel('SNH4_1,SNH4_1E');
subplot(3,2,4); plot(k,ERRSNH4_2,'k');ylabel('ERRSNH4_2');

subplot(3,2,5);plot(k,SNOX2); hold on;plot(k,SNOX2E,'k-.');
legend('SNOX2','SNOX2E');ylabel('SNOX2,SNOX2E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNOX2,'k');ylabel('ERRSNOX2');xlabel('discrete time k')
set(figure(12),'name','BASM3 Tank 2 Process variables and estimation
error','numbertitle','off').

% Tank 3
figure(14)
subplot(3,2,1);plot(k,SS3); hold on;plot(k,SS3E,'k-.');ylabel('SS3,SS3E');
legend('SS3','SS3E');subplot(3,2,2); plot(k,ERRSS3,'k');ylabel('ERRSS3');

subplot(3,2,3);plot(k,SNH4_3); hold on;plot(k,SNH4_3E,'k-.');
legend('SNH4_3','SNH4_3E');ylabel('SNH4_3,SNH4_3E');
subplot(3,2,4); plot(k,ERRSNH4_3,'k');ylabel('ERRSNH4_3');

subplot(3,2,5);plot(k,SNOX3); hold on;plot(k,SNOX3E,'k-.');
legend('SNOX3','SNOX3E');ylabel('SNOX1,SNOX1E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNOX3,'k');ylabel('ERRSNOX3');
ylabel('ERRSNOX3');xlabel('discrete time k')
set(figure(14),'name','BASM3 Tank 3 Process variables and estimation
error','numbertitle','off')

% Tank 4
figure(15)
subplot(3,2,1);plot(k,SS4); hold on;plot(k,SS4E,'k-.');ylabel('SS4,SS4E');
legend('SS4','SS4E');subplot(3,2,2); plot(k,ERRSS4,'k');ylabel('ERRSS4');

subplot(3,2,3);plot(k,SNH4_4); hold on;plot(k,SNH4_4E,'k-.');
legend('SNH4_4','SNH4_4E');ylabel('SNH4_4,SNH4_4E');
subplot(3,2,4); plot(k,ERRSNH4_4,'k');ylabel('ERRSNH4_4');

subplot(3,2,5);plot(k,SNOX4); hold on;plot(k,SNOX4E,'k-.');
legend('SNOX4','SNOX4E');ylabel('SNOX4,SNOX4E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNOX1,'k');ylabel('ERRSNOX4');
ylabel('ERRSNO4');xlabel('discrete time k')
set(figure(15),'name','BASM3 Tank 4 Process variables and estimation
error','numbertitle','off')

% Tank 5
figure(16)
subplot(3,2,1);plot(k,SS5); hold on;plot(k,SS5E,'k-.');ylabel('SS5,SS5E');
legend('SS5','SS5E');subplot(3,2,2); plot(k,ERRSS5,'k');ylabel('ERRSS5');

subplot(3,2,3);plot(k,SNH4_5); hold on;plot(k,SNH4_5E,'k-.');
legend('SNH4_5','SNH4_5E');ylabel('SNH4_5,SNH4_5E');
subplot(3,2,4); plot(k,ERRSNH4_5,'k');ylabel('ERRSNH4_5');

subplot(3,2,5);plot(k,SNOX5); hold on;plot(k,SNOX5E,'k-.');
legend('SNOX5','SNOX5E');ylabel('SNOX5,SNOX5E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNOX3,'k');
ylabel('ERRSNOX5');xlabel('discrete time k')
set(figure(16),'name','BASM3 Tank 5 Process variables and estimation
error','numbertitle','off')
```

D.2: MATLAB script file – P_BASM3F.m

```matlab
% M-File: P_BASM3F.m
% ============================= M-FILE DESCRIPTION ==========================
% This m-file is used for:
% Calculation of the process rates for the full model

function Pk  = P_BASM3F(xk)

global kH kSTO etaNOX muH bH_O2 bH_NOX bSTO_O2 bSTO_NOX muA bA_O2 bA_NOX KX KO2 KS KNOX
KNH4 KALK KSTO KA_O2 KA_NH4 KA_ALK KA_NOX;
% TANK 1
P(1)= kH*((xk(9)/xk(10))/(KX+(xk(9)/xk(10))))* xk(10);
P(2)= kSTO * (xk(1)/(KO2+xk(1))) * (xk(3)/(KS+xk(3)))* xk(10);
P(3)= kSTO * etaNOX * (KO2/(KO2+xk(1))) * (xk(6)/(KNOX+xk(6))) * (xk(3)/(KS+xk(3))) *
xk(10);
P(4)= muH * (xk(1)/(KO2+xk(1))) * (xk(4)/(KNH4+xk(4))) * (xk(7)/(KALK+xk(7))) *
((xk(11)/xk(10))/(KSTO+(xk(11)/xk(10)))) * xk(10);
P(5)= muH * etaNOX * (KO2/(KO2+xk(1))) * (xk(6)/(KNOX+xk(6))) * (xk(4)/(KNH4+xk(4))) *
(xk(7)/(KALK+xk(7))) * ((xk(11)/xk(10))/(KSTO+(xk(11)/xk(10)))) * xk(10);
P(6)= bH_O2 * (xk(1)/(KO2+xk(1))) * xk(10);
P(7)= bH_NOX * (KO2/(KO2+xk(1))) * (xk(6)/(KNOX+xk(6))) * xk(10);
P(8)= bSTO_O2 * (xk(1)/(KO2+xk(1))) * xk(11);
P(9)= bSTO_NOX * (KO2/(KO2+xk(1))) * (xk(6)/(KNOX+xk(6))) * xk(11);
P(10)= muA * (xk(1)/(KA_O2+xk(1))) * (xk(4)/(KA_NH4+xk(4))) * (xk(7)/(KA_ALK+xk(7))) *
xk(12);
P(11)= bA_O2 * (xk(1)/(KA_O2+xk(1))) * xk(12);
P(12)= bA_NOX * (KA_O2/(KA_O2+xk(1))) * (xk(6)/(KA_NOX+xk(6))) * xk(12);


% TANK 2
P(13)= kH*((xk(22)/xk(23))/(KX+(xk(22)/xk(23))))* xk(23);
P(14)= kSTO * (xk(14)/(KO2+xk(14))) * (xk(16)/(KS+xk(16)))* xk(23);
P(15)= kSTO * etaNOX * (KO2/(KO2+xk(14))) * (xk(19)/(KNOX+xk(19))) *
(xk(16)/(KS+xk(16))) * xk(23);
P(16)= muH * (xk(14)/(KO2+xk(14))) * (xk(17)/(KNH4+xk(17))) * (xk(20)/(KALK+xk(20))) *
((xk(24)/xk(23))/(KSTO+(xk(24)/xk(23)))) * xk(23);
P(17)= muH * etaNOX * (KO2/(KO2+xk(14))) * (xk(19)/(KNOX+xk(19))) *
(xk(17)/(KNH4+xk(17))) * (xk(20)/(KALK+xk(20))) *
((xk(24)/xk(23))/(KSTO+(xk(24)/xk(23)))) * xk(23);
P(18)= bH_O2 * (xk(14)/(KO2+xk(14))) * xk(23);
P(19)= bH_NOX * (KO2/(KO2+xk(14))) * (xk(19)/(KNOX+xk(19))) * xk(23);
P(20)= bSTO_O2 * (xk(14)/(KO2+xk(14))) * xk(24);
P(21)= bSTO_NOX * (KO2/(KO2+xk(14))) * (xk(19)/(KNOX+xk(19))) * xk(24);
P(22)= muA * (xk(14)/(KA_O2+xk(14))) * (xk(17)/(KA_NH4+xk(17))) *
(xk(20)/(KA_ALK+xk(20))) * xk(25);
P(23)= bA_O2 * (xk(14)/(KA_O2+xk(14))) * xk(25);
P(24)= bA_NOX * (KA_O2/(KA_O2+xk(14))) * (xk(19)/(KA_NOX+xk(19))) * xk(25);


% TANK 3
P(25)= kH*((xk(35)/xk(36))/(KX+(xk(35)/xk(36))))* xk(36);
P(26)= kSTO * (xk(27)/(KO2+xk(27))) * (xk(29)/(KS+xk(29)))* xk(36);
P(27)= kSTO * etaNOX * (KO2/(KO2+xk(27))) * (xk(32)/(KNOX+xk(32))) *
(xk(29)/(KS+xk(29))) * xk(36);
P(28)= muH * (xk(27)/(KO2+xk(27))) * (xk(30)/(KNH4+xk(30))) * (xk(33)/(KALK+xk(33))) *
((xk(37)/xk(36))/(KSTO+(xk(37)/xk(36)))) * xk(36);
P(29)= muH * etaNOX * (KO2/(KO2+xk(27))) * (xk(32)/(KNOX+xk(32))) *
(xk(30)/(KNH4+xk(30))) * (xk(33)/(KALK+xk(33))) *
((xk(37)/xk(36))/(KSTO+(xk(37)/xk(36)))) * xk(36);
P(30)= bH_O2 * (xk(27)/(KO2+xk(27))) * xk(36);
P(31)= bH_NOX * (KO2/(KO2+xk(27))) * (xk(32)/(KNOX+xk(32))) * xk(36);
P(32)= bSTO_O2 * (xk(27)/(KO2+xk(27))) * xk(37);
P(33)= bSTO_NOX * (KO2/(KO2+xk(27))) * (xk(32)/(KNOX+xk(32))) * xk(37);
P(34)= muA * (xk(27)/(KA_O2+xk(27))) * (xk(30)/(KA_NH4+xk(30))) *
(xk(33)/(KA_ALK+xk(33))) * xk(38);
P(35)= bA_O2 * (xk(27)/(KA_O2+xk(27))) * xk(38);
P(36)= bA_NOX * (KA_O2/(KA_O2+xk(27))) * (xk(32)/(KA_NOX+xk(32))) * xk(38);


% TANK 4
```

261

```
P(37)= kH*((xk(48)/xk(49))/(KX+(xk(48)/xk(49))))* xk(49);
P(38)= kSTO * (xk(40)/(KO2+xk(40))) * (xk(42)/(KS+xk(42)))* xk(49);
P(39)= kSTO * etaNOX * (KO2/(KO2+xk(40))) * (xk(45)/(KNOX+xk(45))) *
(xk(42)/(KS+xk(42))) * xk(49);
P(40)= muH * (xk(40)/(KO2+xk(40))) * (xk(43)/(KNH4+xk(43))) * (xk(46)/(KALK+xk(46))) *
((xk(50)/xk(49))/(KSTO+(xk(50)/xk(49)))) * xk(49);
P(41)= muH * etaNOX * (KO2/(KO2+xk(40))) * (xk(45)/(KNOX+xk(45))) *
(xk(43)/(KNH4+xk(43))) * (xk(46)/(KALK+xk(46))) *
((xk(50)/xk(49))/(KSTO+(xk(50)/xk(49)))) * xk(49);
P(42)= bH_O2 * (xk(40)/(KO2+xk(40))) * xk(49);
P(43)= bH_NOX * (KO2/(KO2+xk(40))) * (xk(45)/(KNOX+xk(45))) * xk(49);
P(44)= bSTO_O2 * (xk(40)/(KO2+xk(40))) * xk(50);
P(45)= bSTO_NOX * (KO2/(KO2+xk(40))) * (xk(45)/(KNOX+xk(45))) * xk(50);
P(46)= muA * (xk(40)/(KA_O2+xk(40))) * (xk(43)/(KA_NH4+xk(43))) *
(xk(46)/(KA_ALK+xk(46))) * xk(51);
P(47)= bA_O2 * (xk(40)/(KA_O2+xk(40))) * xk(51);
P(48)= bA_NOX * (KA_O2/(KA_O2+xk(40))) * (xk(45)/(KA_NOX+xk(45))) * xk(51);

% TANK 5
P(49)= kH*((xk(61)/xk(62))/(KX+(xk(61)/xk(62))))* xk(62);
P(50)= kSTO * (xk(53)/(KO2+xk(53))) * (xk(55)/(KS+xk(55)))* xk(62);
P(51)= kSTO * etaNOX * (KO2/(KO2+xk(53))) * (xk(58)/(KNOX+xk(58))) *
(xk(55)/(KS+xk(55))) * xk(62);
P(52)= muH * (xk(53)/(KO2+xk(53))) * (xk(56)/(KNH4+xk(56))) * (xk(59)/(KALK+xk(59))) *
((xk(63)/xk(62))/(KSTO+(xk(63)/xk(62)))) * xk(62);
P(53)= muH * etaNOX * (KO2/(KO2+xk(53))) * (xk(58)/(KNOX+xk(58))) *
(xk(56)/(KNH4+xk(56))) * (xk(59)/(KALK+xk(59))) *
((xk(63)/xk(62))/(KSTO+(xk(63)/xk(62)))) * xk(62);
P(54)= bH_O2 * (xk(53)/(KO2+xk(53))) * xk(62);
P(55)= bH_NOX * (KO2/(KO2+xk(53))) * (xk(58)/(KNOX+xk(58))) * xk(62);
P(56)= bSTO_O2 * (xk(53)/(KO2+xk(53))) * xk(63);
P(57)= bSTO_NOX * (KO2/(KO2+xk(53))) * (xk(58)/(KNOX+xk(58))) * xk(63);
P(58)= muA * (xk(53)/(KA_O2+xk(53))) * (xk(56)/(KA_NH4+xk(56))) *
(xk(59)/(KA_ALK+xk(59))) * xk(64);
P(59)= bA_O2 * (xk(53)/(KA_O2+xk(53))) * xk(64);
P(60)= bA_NOX * (KA_O2/(KA_O2+xk(53))) * (xk(58)/(KA_NOX+xk(58))) * xk(64);

Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20) P(21) P(22) P(23) P(24) P(25) P(26) P(27) P(28) P(29)
P(30) P(31) P(32) P(33) P(34) P(35) P(36) P(37) P(38) P(39) P(40) P(41) P(42) P(43)
P(44) P(45) P(46) P(47) P(48) P(49) P(50) P(51) P(52) P(53) P(54) P(55) P(56) P(57)
P(58) P(59) P(60)]';
```

D.3: MATLAB script file – P_BASM3R.m

```
% M-File: P_BASM3R.m
% ========================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model

function Pk = P_BASM1R(zk,uk,XHk,XAk,XSk,XSTOk,SALKk);

global kH kSTO etaNOX muH bH_O2 bH_NOX bSTO_O2 bSTO_NOX muA bA_O2 bA_NOX KX KO2 KS KNOX
KNH4 KALK KSTO KA_O2 KA_NH4 KA_ALK KA_NOX;
% TANK 1
P(1)= ((XSk(1)/XHk(1))/(KX+(XSk(1)/XHk(1))))* XHk(1);
P(2)= (uk(1)/(KO2+uk(1))) * (zk(2)/(KNH4+zk(2))) * (SALKk(1)/(KALK+SALKk(1))) *
((XSTOk(1)/XHk(1))/(KSTO+(XSTOk(1)/XHk(1)))) * XHk(1);
P(3)= etaNOX * (KO2/(KO2+uk(1))) * (zk(3)/(KNOX+zk(3))) * (zk(2)/(KNH4+zk(2))) *
(SALKk(1)/(KALK+SALKk(1))) * ((XSTOk(1)/XHk(1))/(KSTO+(XSTOk(1)/XHk(1)))) * XHk(1);
P(4)= (uk(1)/(KA_O2+uk(1))) * (zk(2)/(KA_NH4+zk(2))) * (SALKk(1)/(KA_ALK+SALKk(1))) *
XAk(1);
```

```matlab
P1k = [P(1) 0 0; 0 P(2) 0; 0 P(3) 0; 0 0 P(4)];

% TANK 2
P(5)= ((XSk(2)/XHk(2))/(KX+(XSk(2)/XHk(2))))* XHk(2);
P(6)= (uk(2)/(KO2+uk(2))) * (zk(5)/(KNH4+zk(5))) * (SALKk(2)/(KALK+SALKk(2))) *
((XSTOk(2)/XHk(2))/(KSTO+(XSTOk(2)/XHk(2)))) * XHk(2);
P(7)= etaNOX * (KO2/(KO2+uk(1))) * (zk(6)/(KNOX+zk(6))) * (zk(5)/(KNH4+zk(5))) *
(SALKk(2)/(KALK+SALKk(2))) * ((XSTOk(2)/XHk(2))/(KSTO+(XSTOk(2)/XHk(2)))) * XHk(2);
P(8)= (uk(2)/(KA_O2+uk(2))) * (zk(5)/(KA_NH4+zk(5))) * (SALKk(2)/(KA_ALK+SALKk(2))) *
XAk(2);
P2k = [P(5) 0 0; 0 P(6) 0; 0 P(7) 0; 0 0 P(8)];

% TANK 3
P(9) = ((XSk(3)/XHk(3))/(KX+(XSk(3)/XHk(3))))* XHk(3);
P(10)= (uk(3)/(KO2+uk(3))) * (zk(8)/(KNH4+zk(8))) * (SALKk(3)/(KALK+SALKk(3))) *
((XSTOk(3)/XHk(3))/(KSTO+(XSTOk(3)/XHk(3)))) * XHk(3);
P(11)= etaNOX * (KO2/(KO2+uk(2))) * (zk(9)/(KNOX+zk(9))) * (zk(8)/(KNH4+zk(8))) *
(SALKk(3)/(KALK+SALKk(3))) * ((XSTOk(3)/XHk(3))/(KSTO+(XSTOk(3)/XHk(3)))) * XHk(3);
P(12)= (uk(3)/(KA_O2+uk(3))) * (zk(8)/(KA_NH4+zk(8))) * (SALKk(3)/(KA_ALK+SALKk(3))) *
XAk(3);
P3k = [P(9) 0 0; 0 P(10) 0; 0 P(11) 0; 0 0 P(12)];

% TANK 4
P(13)= ((XSk(4)/XHk(4))/(KX+(XSk(4)/XHk(4))))* XHk(4);
P(14)= (uk(4)/(KO2+uk(4))) * (zk(11)/(KNH4+zk(11))) * (SALKk(4)/(KALK+SALKk(4))) *
((XSTOk(4)/XHk(4))/(KSTO+(XSTOk(4)/XHk(4)))) * XHk(4);
P(15)= etaNOX * (KO2/(KO2+uk(4))) * (zk(12)/(KNOX+zk(12))) * (zk(11)/(KNH4+zk(11))) *
(SALKk(4)/(KALK+SALKk(4))) * ((XSTOk(4)/XHk(4))/(KSTO+(XSTOk(4)/XHk(4)))) * XHk(4);
P(16)= (uk(4)/(KA_O2+uk(4))) * (zk(11)/(KA_NH4+zk(11))) * (SALKk(4)/(KA_ALK+SALKk(4)))
* XAk(4);
P4k = [P(13) 0 0; 0 P(14) 0; 0 P(15) 0; 0 0 P(16)];

% TANK 5
P(17)= ((XSk(5)/XHk(5))/(KX+(XSk(5)/XHk(5))))* XHk(5);
P(18)= (uk(5)/(KO2+uk(5))) * (zk(14)/(KNH4+zk(14))) * (SALKk(5)/(KALK+SALKk(5))) *
((XSTOk(5)/XHk(5))/(KSTO+(XSTOk(5)/XHk(5)))) * XHk(5);
P(19)= etaNOX * (KO2/(KO2+uk(5))) * (zk(15)/(KNOX+zk(15))) * (zk(14)/(KNH4+zk(14))) *
(SALKk(5)/(KALK+SALKk(5))) * ((XSTOk(5)/XHk(5))/(KSTO+(XSTOk(5)/XHk(5)))) * XHk(5);
P(20)= (uk(5)/(KA_O2+uk(5))) * (zk(14)/(KA_NH4+zk(14))) * (SALKk(5)/(KA_ALK+SALKk(5)))
* XAk(5);
P5k = [P(17) 0 0; 0 P(18) 0; 0 P(19) 0; 0 0 P(20)];

%Matrix Pk for the whole process
Pk = [P1k; P2k; P3k; P4k; P5k];
```

D.4: MATLAB script file – P_BASM3RE.m

```matlab
% M-File: P_BASM3RE.m
% ================================ M-FILE DESCRIPTION ============================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model

% Description of the function function
function Pk = P_BASM1RE(zk,uk,XHk,XAk,XSk,XSTOk,SALKk,muHEk,muAEk,KhEk);

global kH kSTO etaNOX muH bH_O2 bH_NOX bSTO_O2 bSTO_NOX muA bA_O2 bA_NOX KX KO2 KS KNOX
KNH4 KALK KSTO KA_O2 KA_NH4 KA_ALK KA_NOX;
% TANK 1
P(1)= KhEk * ((XSk(1)/XHk(1))/(KX+(XSk(1)/XHk(1))))* XHk(1);
P(2)= muHEk * (uk(1)/(KO2+uk(1))) * (zk(2)/(KNH4+zk(2))) * (SALKk(1)/(KALK+SALKk(1))) *
((XSTOk(1)/XHk(1))/(KSTO+(XSTOk(1)/XHk(1)))) * XHk(1);
```

```
P(3)= muHEk * etaNOX * (KO2/(KO2+uk(1))) * (zk(3)/(KNOX+zk(3))) * (zk(2)/(KNH4+zk(2)))
* (SALKk(1)/(KALK+SALKk(1))) * ((XSTOk(1)/XHk(1))/(KSTO+(XSTOk(1)/XHk(1)))) * XHk(1);
P(4)= muAEk * (uk(1)/(KA_O2+uk(1))) * (zk(2)/(KA_NH4+zk(2))) *
(SALKk(1)/(KA_ALK+SALKk(1))) * XAk(1);

% TANK 2
P(5)= KhEk * ((XSk(2)/XHk(2))/(KX+(XSk(2)/XHk(2))))* XHk(2);
P(6)= muHEk * (uk(2)/(KO2+uk(2))) * (zk(5)/(KNH4+zk(5))) * (SALKk(2)/(KALK+SALKk(2))) *
((XSTOk(2)/XHk(2))/(KSTO+(XSTOk(2)/XHk(2)))) * XHk(2);
P(7)= muHEk * etaNOX * (KO2/(KO2+uk(1))) * (zk(6)/(KNOX+zk(6))) * (zk(5)/(KNH4+zk(5)))
* (SALKk(2)/(KALK+SALKk(2))) * ((XSTOk(2)/XHk(2))/(KSTO+(XSTOk(2)/XHk(2)))) * XHk(2);
P(8)= muAEk * (uk(2)/(KA_O2+uk(2))) * (zk(5)/(KA_NH4+zk(5))) *
(SALKk(2)/(KA_ALK+SALKk(2))) * XAk(2);

% TANK 3
P(9) = KhEk * ((XSk(3)/XHk(3))/(KX+(XSk(3)/XHk(3))))* XHk(3);
P(10)= muHEk * (uk(3)/(KO2+uk(3))) * (zk(8)/(KNH4+zk(8))) * (SALKk(3)/(KALK+SALKk(3)))
* ((XSTOk(3)/XHk(3))/(KSTO+(XSTOk(3)/XHk(3)))) * XHk(3);
P(11)= muHEk * etaNOX * (KO2/(KO2+uk(2))) * (zk(9)/(KNOX+zk(9))) * (zk(8)/(KNH4+zk(8)))
* (SALKk(3)/(KALK+SALKk(3))) * ((XSTOk(3)/XHk(3))/(KSTO+(XSTOk(3)/XHk(3)))) * XHk(3);
P(12)= muAEk * (uk(3)/(KA_O2+uk(3))) * (zk(8)/(KA_NH4+zk(8))) *
(SALKk(3)/(KA_ALK+SALKk(3))) * XAk(3);

% TANK 4
P(13)= KhEk * ((XSk(4)/XHk(4))/(KX+(XSk(4)/XHk(4))))* XHk(4);
P(14)= muHEk * (uk(4)/(KO2+uk(4))) * (zk(11)/(KNH4+zk(11))) *
(SALKk(4)/(KALK+SALKk(4))) * ((XSTOk(4)/XHk(4))/(KSTO+(XSTOk(4)/XHk(4)))) * XHk(4);
P(15)= muHEk * etaNOX * (KO2/(KO2+uk(4))) * (zk(12)/(KNOX+zk(12))) *
(zk(11)/(KNH4+zk(11))) * (SALKk(4)/(KALK+SALKk(4))) *
((XSTOk(4)/XHk(4))/(KSTO+(XSTOk(4)/XHk(4)))) * XHk(4);
P(16)= muAEk * (uk(4)/(KA_O2+uk(4))) * (zk(11)/(KA_NH4+zk(11))) *
(SALKk(4)/(KA_ALK+SALKk(4))) * XAk(4);

% TANK 5
P(17)= KhEk * ((XSk(5)/XHk(5))/(KX+(XSk(5)/XHk(5))))* XHk(5);
P(18)= muHEk * (uk(5)/(KO2+uk(5))) * (zk(14)/(KNH4+zk(14))) *
(SALKk(5)/(KALK+SALKk(5))) * ((XSTOk(5)/XHk(5))/(KSTO+(XSTOk(5)/XHk(5)))) * XHk(5);
P(19)= muHEk * etaNOX * (KO2/(KO2+uk(5))) * (zk(15)/(KNOX+zk(15))) *
(zk(14)/(KNH4+zk(14))) * (SALKk(5)/(KALK+SALKk(5))) *
((XSTOk(5)/XHk(5))/(KSTO+(XSTOk(5)/XHk(5)))) * XHk(5);
P(20)= muAEk * (uk(5)/(KA_O2+uk(5))) * (zk(14)/(KA_NH4+zk(14))) *
(SALKk(5)/(KA_ALK+SALKk(5))) * XAk(5);

Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20)]';
```

APPENDIX E: DEVELOPED MATLAB CODE FOR OPTIMAL CONTROL OF THE ATHLONE PLANT WITH THE ACTIVATED SLUDGE MODEL NO.1 BIO-MODEL

E.1: MATLAB script file – AASM1_Simulation.m

```
% M-File: AASM1_Simulation.m
% ========================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Simulation of the Athlone plant using full ASM1 model
% Calculation of the parameters for the reduced ASM1 model
% Simulation of the Athlone plant using the reduced ASM1 model

clear all          % Clear the workspace
clc                % Clear the command window


% ===================== PARAMETER DEFINITION =====================
% Declaration of the global variables
global muH bH muA bA etag ka kh etah Ks KOH KNO KNH KOA KX


% Declaration of the global variables
K1   = 1344;       % Prediction horizon or interval
DT   = 0.01042;    % Sampling time

% Initialization of the model parameters, typical values at 15 0C
% Stochiometric parameters
YH   = 0.67;       % Heterotrophic yield
YA   = 0.24;       % Autotrophic yield
fP   = 0.08;       % Fraction of biomass yielding particulate products
iXB  = 0.08;       % COD mass in biomass
iXP  = 0.06;       % COD mass in products from biomass
% Kinetic parameters
muH  = 0.4;        % Heterotrophic maximum specific growth rate
bH   = 3;          % Heterotrophic decay rate
muA  = 0.5;        % Autotrophic maximum specific growth rate
bA   = 0.05;       % Correction factor for anoxic growth of heterotrophs
etag = 0.8;        % Correction factor for anoxic growth of heterotrophs
ka   = 0.05;       % Ammonification rate
kh   = 3;          % Maximum specific hydrolysis rate
etah = 0.8;        % Correction factor for anoxic hydrolysis
% Monod switching functions
Ks   = 10.0;       % Half saturation coefficient (hsc) for heterotrophs
KOH  = 0.2;        % Oxygen hsc for heterotrophs
KNO  = 0.5;        % Nitrate hsc for denitrifying heterotrophs
KNH  = 1.0;        % Ammonia hsc for autotrophs
KOA  = 0.4;        % Oxygen hsc for autotrophs
KX   = 0.1;        % Hsc for hydrolysis of slowly biodegradable substrate

% Initialization of the Athlone process data
V1   = 1148;       % Tank 1 volume
V2   = 1148;       % Tank 2 volume
V3   = 5273;       % Tank 3 volume

Q0   = 40003;      % Influent flow rate
Qa   = 39916.8;    % Recycled flow rate from tank 3
Qp   = 39916.8;    % Recycled flow rate from tank 2
Qr   = 40003;      % Recycled flow rate from the settler
Qw   = 38400.0;    % Waste flow rate (3840)
Q1   = Qp + Q0;    % flow through the tank 1
Q2   = Qp + Q0 + Qa + Qr;   % flow through the tank 2
Q3   = Q2;         % flow through the tank 3

%Q0   = 18446;     % Influent flow rate
%Qa   = 55338;     % Internal recirculation rate
```

```matlab
%Qr    = 18446;        % Sludge recycled rate
%Qw    = 395;          % Wasted sludge rate

% Coeficient for fraction of flow rates for the particulate materials
LAMBDA = (Qp+Q0+Qr-Qw)/(Qr);

% Oxygen parameters
SOsat = 8;             % Oxygen saturation point
KLa1  = 0;             % Oxygen transfer function in Tank 1
KLa2  = 0;             % Oxygen transfer function in Tank 2
KLa3  = 84;            % Oxygen transfer function in Tank 3

KL = [KLa1 KLa2 KLa3]; KLa = diag(KL);


% ======================== PROCESS MODEL =========================
% Calculation of the MATRIX A
% Matrix A represent weighting factors of the accumulated state variables
% from different flows towards state variables in each tank as derived
% through mass balance principles, see chapter 3 of the Thesis
IV1 = DT/ V1; IV11 = IV1 * Q1; A11 = 1 - IV11;
IV2 = DT/ V2; IV22 = IV2 * Q2; A22 = 1 - IV22;
IV3 = DT/ V3; IV33 = IV3 * Q3; A33 = 1 - IV33;


A12 = IV1 * Qp;
A23a = IV2 * (Qa + Qr);
A23b = IV2 * (Qa + (LAMBDA*Qr));
A21 = IV2*Q1;
A32 = IV3*Q2;


A11 = A11 * ones(1,13); A11 = diag(A11);
A12 = A12 * ones(1,13); A12 = diag(A12);
A13 = zeros(13,13);


A21 = A21 * ones(1,13); A21 = diag(A21);
A22 = A22 * ones(1,13); A22 = diag(A22);
A23 = [A23a,A23a,A23b,A23b,A23b,A23b,A23b,A23a,A23a,A23a,A23a,A23b,A23a];
A23 = diag(A23);


A31 = zeros(13,13);
A32 = A32 * ones(1,13); A32 = diag(A32);
A33 = A33 * ones(1,13); A33 = diag(A33);


A = [A11 A12 A13; A21 A22 A23; A31 A32 A33];

% Calculation of the MATRIX B
% Matrix B represent weighting factors of the state variables towards the
% influent wastewater composition as derived through mass balance
% principles, see chapter 3 of the Thesis
IV0 = DT/V1*Q0;
B11 = IV0 * ones(1,13); B11 = diag(B11);


B = [B11; zeros(26,13)];

% Formulation of the MATRIX C
% Matrix C represent weighting factors of the state variables towards the
% processes on the ASM1 matrix table
c1 = -(1/YH);
c2 = (-(1-YH))/(YH);
c3 = -(iXB/14);
c4 = (-(1-YH))/(2.86*YH);
c5 = ((1-YH)/(14*2.86*YH))-(iXB/14);
c6 = (-4.57/YA)+1;
c7 = (1/YA);
c8 = -iXB-(1/YA);
c9 = -(iXB/14)-(1/(7*YA));
```

266

```
c10 = 1 - fP;
c11 = iXB-(fP*iXP);

CC = [[0  c1 0  0    1  0   0  c2 0  -iXB 0  0   c3  ];
      [0  c1 0  0    1  0   0  0  c4 -iXB 0  0   c5  ];
      [0  0  0  0    0  1   0  c6 c7 c8   0  0   c9  ];
      [0  0  0  c10 -1  0   fP 0  0  0    0  c11 0   ];
      [0  0  0  c10  0  -1  fP 0  0  0    0  c11 0   ];
      [0  0  0  0    0  0   0  0  0  1    -1 0   1/14];
      [0  1  0 -1    0  0   0  0  0  0    0  0   0   ];
      [0  0  0  0    0  0   0  0  0  0    1  -1  0   ]];

C = DT*[CC zeros(8,26); zeros(8,13) CC zeros(8,13); zeros(8,26) CC];

% =============================== SIMULATION DATA ============================
% Average values of the inflow concentrations based on the dry weather file
SIi = 30.0; SSi = 69.5; XIi = 51.2; XSi = 202.32; XBHi = 28.17; XBAi = 0.0;
XPi = 0.0; SOi = 0.0; SNOi = 0.0; SNHi = 31.56; SNDi = 6.95; XNDi = 10.59;
SALKi = 7.0;

% SI SS XI XS XBH XBA XP SO SNO SNH SND XND SALK;
xi = [SIi*ones(1,K1);  SSi*ones(1,K1);   XIi*ones(1,K1);  XSi*ones(1,K1);
      XBHi*ones(1,K1); XBAi*ones(1,K1); XPi*ones(1,K1);  SOi*ones(1,K1);
      SNOi*ones(1,K1); SNHi*ones(1,K1); SNDi*ones(1,K1); XNDi*ones(1,K1);
      SALKi*ones(1,K1)];

% Values of the initial conditions
% For tank 1
SI_1  = 3.0000000e+01; SS_1  = 2.8082131e+00; XI_1  = 1.1491252e+03;
XS_1  = 8.2134908e+01; XBH_1 = 2.5517658e+03; XBA_1 = 1.4838943e+02;
XP_1  = 4.4885186e+02; SO_1  = 4.2984433e-03; SNO_1 = 5.3699400e+00;
SNH_1 = 7.9178845e+00; SND_1 = 1.2166405e+00; XND_1 = 5.2848894e+00/4;
SALK_1 = 4.9277103e+00;
x10 = [SI_1; SS_1; XI_1; XS_1; XBH_1; XBA_1; XP_1; SO_1; SNO_1; SNH_1;
       SND_1; XND_1; SALK_1 ];
x10 = (x10.*rand(13,1)/2);

% For tank 2
SI_2  = 3.0000000e+01; SS_2  = 1.4587940e+00; XI_2  = 1.1491252e+03;
XS_2  = 7.6386187e+01; XBH_2 = 2.5533851e+03; XBA_2 = 1.4830914e+02;
XP_2  = 4.4952273e+02; SO_2  = 6.3131911e-05; SNO_2 = 3.6619672e+00;
SNH_2 = 8.3444148e+00; SND_2 = 8.8206477e-01; XND_2 = 5.0290873e+00/4;
SALK_2 = 5.0801748e+00;
x20 = [SI_2; SS_2; XI_2; XS_2; XBH_2; XBA_2; XP_2; SO_2; SNO_2; SNH_2;
       SND_2; XND_2; SALK_2 ];
x20 = (x20.*rand(13,1)/2);

% For tank 3
SI_3  = 3.0000000e+01; SS_3  = 1.1495418e+00; XI_3  = 1.1491252e+03;
XS_3  = 6.4854922e+01; XBH_3 = 2.5571314e+03; XBA_3 = 1.4894126e+02;
XP_3  = 4.5041834e+02; SO_3  = 1.7183778e+00; SNO_3 = 6.5408820e+00;
SNH_3 = 5.5479452e+00; SND_3 = 8.2888682e-01; XND_3 = 4.3924277e+00/4;
SALK_3 = 4.6747902e+00;
x30 = [SI_3; SS_3; XI_3; XS_3; XBH_3; XBA_3; XP_3; SO_3; SNO_3; SNH_3;
       SND_3; XND_3; SALK_3 ];
x30 = (x30.*rand(13,1)/2);

x0 = [x10;x20;x30]*0.0001;
x  = [x0 zeros(39,K1)];

% Steady state simulation of the model
for k = 1:K1
    xk  = x(:,k);
    xik = xi(:,k);
    Pk  = P_AASM1F(xk); % Calculation of the process rates
```

267

```
%SOk   = [xk(34,1)];
%xxk   = DT*KLa*(SOsat-SOk);
%xxlk  = [zeros(33,1);xxk(1);zeros(5,1)];
%x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xxlk;

SOk    = [xk(8,1);xk(21,1);xk(34,1)];
xxk    = DT*KLa*(SOsat-SOk);
xxlk   = [zeros(7,1);xxk(1);zeros(12,1);xxk(2);zeros(12,1);xxk(3);zeros(5,1)];
x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xxlk;

end

% Full model response to the steady state influent concentrations
% Tank 1
figure(1)
subplot(5,3,1);plot(x(1,:));title('SI1')
subplot(5,3,2);plot(x(2,:));title('SS1')
subplot(5,3,3);plot(x(3,:));title('XI1')
subplot(5,3,4);plot(x(4,:));title('XS1')
subplot(5,3,5);plot(x(5,:));title('XBH1')
subplot(5,3,6);plot(x(6,:));title('XBA1')
subplot(5,3,7);plot(x(7,:));title('XP1')
subplot(5,3,8);plot(x(8,:));title('SO1')
subplot(5,3,9);plot(x(9,:));title('SNO1')
subplot(5,3,10);plot(x(10,:));title('SNH1')
subplot(5,3,11);plot(x(11,:));title('SND1')
xlabel('discrete time k')
subplot(5,3,12);plot(x(12,:));title('XND1')
xlabel('discrete time k')
subplot(5,3,13);plot(x(13,:));title('SALK1')
xlabel('discrete time k')
set(figure(1),'name','Tank 1 steady state results','numbertitle','off')

% Tank 2
figure(2)
subplot(5,3,1);plot(x(14,:));title('SI2')
subplot(5,3,2);plot(x(15,:));title('SS2')
subplot(5,3,3);plot(x(16,:));title('XI2')
subplot(5,3,4);plot(x(17,:));title('XS2')
subplot(5,3,5);plot(x(18,:));title('XBH2')
subplot(5,3,6);plot(x(19,:));title('XBA2')
subplot(5,3,7);plot(x(20,:));title('XP2')
subplot(5,3,8);plot(x(21,:));title('SO2')
subplot(5,3,9);plot(x(22,:));title('SNO2')
subplot(5,3,10);plot(x(23,:));title('SNH2')
subplot(5,3,11);plot(x(24,:));title('SND2')
xlabel('discrete time k')
subplot(5,3,12);plot(x(25,:));title('XND2')
xlabel('discrete time k')
subplot(5,3,13);plot(x(26,:));title('SALK2')
xlabel('discrete time k')
set(figure(2),'name','Tank 2 steady state results','numbertitle','off')

% Tank 3
figure(3)
subplot(5,3,1);plot(x(27,:));title('SI3')
subplot(5,3,2);plot(x(28,:));title('SS3')
subplot(5,3,3);plot(x(29,:));title('XI3')
subplot(5,3,4);plot(x(30,:));title('XS3')
subplot(5,3,5);plot(x(31,:));title('XBH3')
subplot(5,3,6);plot(x(32,:));title('XBA3')
subplot(5,3,7);plot(x(33,:));title('XP3')
subplot(5,3,8);plot(x(34,:));title('SO3')
subplot(5,3,9);plot(x(35,:));title('SNO3')
subplot(5,3,10);plot(x(36,:));title('SNH3')
subplot(5,3,11);plot(x(37,:));title('SND3')
```

268

```
xlabel('discrete time k')
subplot(5,3,12);plot(x(38,:));title('XND3')
xlabel('discrete time k')
subplot(5,3,13);plot(x(39,:));title('SALK3')
xlabel('discrete time k')
set(figure(3),'name','Tank 3 steady state results','numbertitle','off')


% Declaration of the global variables
K     = 1344;        % Prediction horizon or interval
% Dynamics of the inflow concentrations for dry weather file
load dry.txt

% The order is t, SS XBH XS XI SNH SI SND XND Q0
%              1  2   3  4  5   6  7   8   9  10


xi = [dry(1:K,7)'; dry(1:K,2)'; dry(1:K,5)'; dry(1:K,4)'; dry(1:K,3)';
      XBAi*zeros(1,K); XPi*zeros(1,K); S0i*zeros(1,K); SNOi*zeros(1,K);
      dry(1:K,6)'; dry(1:K,8)'; dry(1:K,9)'; SALKi* ones(1,K)];

% The order is SI SS XI XS XBH XBA XP SO SNO SNH  SND XND SALK
%               1  2  3  4   5   6   7  8   9  10   11  12  13


x = [x(:,K1+1) zeros(39,K)]*0.0000001;


% Dynamic simulation of the model
for k    = 1:K
    xk   = x(:,k);
    xik  = xi(:,k);
    Pk   = P_AASM1F(xk);

    SOk  = [xk(34,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xx1k = [zeros(33,1);xxk(1);zeros(5,1)];
    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xx1k;
end

% Full model response to the dynamic influent concentrations
% Tank 1
figure(4)
subplot(5,3,1);plot(x(1,:));title('SI1')
subplot(5,3,2);plot(x(2,:));title('SS1')
subplot(5,3,3);plot(x(3,:));title('XI1')
subplot(5,3,4);plot(x(4,:));title('XS1')
subplot(5,3,5);plot(x(5,:));title('XBH1')
subplot(5,3,6);plot(x(6,:));title('XBA1')
subplot(5,3,7);plot(x(7,:));title('XP1')
subplot(5,3,8);plot(x(8,:));title('SO1')
subplot(5,3,9);plot(x(9,:));title('SNO1')
subplot(5,3,10);plot(x(10,:));title('SNH1')
subplot(5,3,11);plot(x(11,:));title('SND1')
xlabel('discrete time k')
subplot(5,3,12);plot(x(12,:));title('XND1')
xlabel('discrete time k')
subplot(5,3,13);plot(x(13,:));title('SALK1')
xlabel('discrete time k')
set(figure(4),'name','Tank 1 dynamic results','numbertitle','off')

% Tank 2
figure(5)
subplot(5,3,1);plot(x(14,:));title('SI2')
subplot(5,3,2);plot(x(15,:));title('SS2')
subplot(5,3,3);plot(x(16,:));title('XI2')
subplot(5,3,4);plot(x(17,:));title('XS2')
subplot(5,3,5);plot(x(18,:));title('XBH2')
subplot(5,3,6);plot(x(19,:));title('XBA2')
subplot(5,3,7);plot(x(20,:));title('XP2')
```

269

```
subplot(5,3,8);plot(x(21,:));title('SO2')
subplot(5,3,9);plot(x(22,:));title('SNO2')
subplot(5,3,10);plot(x(23,:));title('SNH2')
subplot(5,3,11);plot(x(24,:));title('SND2')
xlabel('discrete time k')
subplot(5,3,12);plot(x(25,:));title('XND2')
xlabel('discrete time k')
subplot(5,3,13);plot(x(26,:));title('SALK2')
xlabel('discrete time k')
set(figure(5),'name','Tank 2 dynamic results','numbertitle','off')

% Tank 3
figure(6)
subplot(5,3,1);plot(x(27,:));title('SI3')
subplot(5,3,2);plot(x(28,:));title('SS3')
subplot(5,3,3);plot(x(29,:));title('XI3')
subplot(5,3,4);plot(x(30,:));title('XS3')
subplot(5,3,5);plot(x(31,:));title('XBH3')
subplot(5,3,6);plot(x(32,:));title('XBA3')
subplot(5,3,7);plot(x(33,:));title('XP3')
subplot(5,3,8);plot(x(34,:));title('SO3')
subplot(5,3,9);plot(x(35,:));title('SNO3')
subplot(5,3,10);plot(x(36,:));title('SNH3')
subplot(5;3,11);plot(x(37,:));title('SND3')
xlabel('discrete time k')
subplot(5,3,12);plot(x(38,:));title('XND3')
xlabel('discrete time k')
subplot(5,3,13);plot(x(39,:));title('SALK3')
xlabel('discrete time k')
set(figure(6),'name','Tank 3 dynamic results','numbertitle','off')

% ====================== REDUCED MODEL FORMULATION ======================
% Form the vector of the states (z) for the reduced model, based on the
% ASM1 prediction
%       SS,      SNO,      SNH
z1 = [x(2,:);   x(9,:);   x(10,:)];
z2 = [x(15,:);  x(22,:);  x(23,:)];
z3 = [x(28,:);  x(35,:);  x(36,:)];
z  = [z1;z2;z3];

% Formation of the vector of the control variables
u1 = x(8,:);
u2 = x(21,:);
u3 = x(34,:);
u  = [u1; u2; u3];

% Dynamic values of the inflow concentrations for dry weather
%     SS            SNO             SNH
zi = [dry(1:K,2)'; SNOi*zeros(1,K); dry(1:K,6)'];

% Steady state values for the biomass XBH, XBA, XS
XBH = [XBH_1*ones(1,K); XBH_2*ones(1,K); XBH_3*ones(1,K)];
XBA = [XBA_1*ones(1,K); XBA_2*ones(1,K); XBA_3*ones(1,K)];
XS  = [XS_1*ones(1,K); XS_2*ones(1,K); XS_3*ones(1,K)];

% MATRIX A for the reduced model (AR)
A11r = 1 - IV11; A22r = 1 - IV22; A33r = 1 - IV33;

A12r = IV1 * Qp; A23r = IV2 * (Qa + Qr);
A21r = IV2*Q1; A32r = IV3*Q2;

A11r = A11r * ones(1,3); A11r = diag(A11r);
A12r = A12r * ones(1,3); A12r = diag(A12r);
A13r = zeros(3,3);

A21r = A21r * ones(1,3); A21r = diag(A21r);
```

270

```
A22r = A22r * ones(1,3); A22r = diag(A22r);
A23r = [A23r,A23r,A23r];
A23r = diag(A23r);

A31r = zeros(3,3);
A32r = A32r * ones(1,3); A32r = diag(A32r);
A33r = A33r * ones(1,3); A33r = diag(A33r);

AR = [A11r A12r A13r; A21r A22r A23r; A31r A32r A33r];

% MATRIX B for the reduced model(BR)
IV0 = DT / V1*Q0;
f = 1.2;
B11r = [IV0 IV0*f IV0];       ..
B11r = diag(B11r);
BR = [B11r; zeros(6,3)];

% MATRIX C for the reduced model (CR)
c1 = -(1/YH);
c2 = -(1-YH)/(2.86*YH);
c3 = (1/YA);
c4 = -iXB-(1/YA);

CCR = [[c1 0   -iXB ];
       [c1 c2 -iXB ];
       [0  c3 c4   ];
       [1  0  0    ]];
CR = DT*[CCR zeros(4,6); zeros(4,3) CCR zeros(4,3); zeros(4,6) CCR];

% Parameter estimation
% Calculation of the estimated coefficients
for k = 1:K
    zk   = z(:,k);
    uk   = u(:,k);
    XBHk = XBH(:,k);
    XBAk = XBA(:,k);
    XSk  = XS(:,k);

    zik  = zi(:,k);
    Pk   = P_AASM1R(zk,uk,XBHk,XBAk,XSk);
    FIk  = CR'*Pk;
    Y(:,k) = z(:,k+1)-AR*z(:,k)-BR*zi(:,k);
    theta(:,k)=(FIk'*FIk)^(-1)*FIk'*Y(:,k);
end

muHE(1,:) = theta(1,:);
muAE(1,:) = theta(2,:);
KhE(1,:)  = theta(3,:);

% Dynamic simulation of the reduced model with the estimated parameters
zE=[x(2,1);x(9,1);x(10,1);x(15,1);x(22,1);x(23,1);x(28,1);x(35,1);x(36,1)];
ERR = z(:,1)-zE;

for k = 1:K
    zEk  = zE(:,k);
    uk   = u(:,k);
   XBHk = XBH(:,k);
    XBAk = XBA(:,k);
    XSk  = XS(:,k);

    zik = zi(:,k);

    muHEk = muHE(:,k);
    muAEk = muAE(:,k);
    KhEk  = KhE(:,k);
```

271

```
        Pk = P_AASM1RE(zEk,uk,XBHk,XBAk,XSk,muHEk,muAEk,KhEk);
        zE(:,k+1) = AR*zE(:,k) + CR'*Pk + BR*zi(:,k);
        ERR(:,k+1)=z(:,k+1)-zE(:,k+1);
end

% Vector of process full, estimated and error variables SNH, SNO,SS, Tank 1
SS1=z(1,:);SNO1=z(2,:);SNH1=z(3,:);
SS1E=zE(1,:);SNO1E=zE(2,:);SNH1E=zE(3,:);
ERRSS1=ERR(1,:);ERRSNO1=ERR(2,:);ERRSNH1=ERR(3,:);
% Calculation of the weighted errors for tank1
eSS1=(ERRSS1*ERRSS1')/(SS1*SS1');
eSNO1=(ERRSNO1*ERRSNO1')/(SNO1*SNO1');
eSNH1=(ERRSNH1*ERRSNH1')/(SNH1*SNH1');

% Vector of process full, estimated and error variables SNH, SNO,SS, Tank 2
SS2=z(4,:);SNO2=z(5,:);SNH2=z(6,:);
SS2E=zE(4,:);SNO2E=zE(5,:);SNH2E=zE(6,:);
ERRSS2=ERR(4,:);ERRSNO2=ERR(5,:);ERRSNH2=ERR(6,:);
% Calculation of the weighted errors for tank 2
eSS2=(ERRSS2*ERRSS2')/(SS2*SS2');
eSNO2=(ERRSNO2*ERRSNO2')/(SNO2*SNO2');
eSNH2=(ERRSNH2*ERRSNH2')/(SNH2*SNH2');

% Vector of process full, estimated and error variables SNH, SNO,SS, Tank 3
SS3=z(7,:);SNO3=z(8,:);SNH3=z(9,:);
SS3E=zE(7,:);SNO3E=zE(8,:);SNH3E=zE(9,:);
ERRSS3=ERR(7,:);ERRSNO3=ERR(8,:);ERRSNH3=ERR(9,:);
% Calculation of the weighted errors for tank 3
eSS3=(ERRSS3*ERRSS3')/(SS3*SS3');
eSNO3=(ERRSNO3*ERRSNO3')/(SNO3*SNO3');
eSNH3=(ERRSNH3*ERRSNH3')/(SNH3*SNH3');

k = 1:K;
% Graphs of the estimated parameters
figure(7)
subplot(3,1,1);plot(k,muHE,'k');title('Estimated muH');ylabel('muHE');
subplot(3,1,2);plot(k,muAE,'k');title('Estimated muA');ylabel('muAE');
subplot(3,1,3);plot(k,KhE,'k');title('Estimated Kh');ylabel('KhE');
set(figure(7),'name','Estimated parameters','numbertitle','off')
xlabel('discrete time k')

k = 1:K+1;
% Graphs of the reduced model response to the dynamic influent
% concentrations
% Tank 1
figure(8)
subplot(3,2,1);plot(k,SS1); hold on;plot(k,SS1E,'k-.');ylabel('SS1,SS1E');
legend('SS1','SS1E');subplot(3,2,2); plot(k,ERRSS1,'k');ylabel('ERRSS1');

subplot(3,2,3);plot(k,SNO1); hold on;plot(k,SNO1E,'k-.');
legend('SNO1','SNO1E');
ylabel('SNO1,SNO1E');subplot(3,2,4); plot(k,ERRSNO1,'k');ylabel('ERRSNO1');

subplot(3,2,5);plot(k,SNH1); hold on;plot(k,SNH1E,'k-.');
legend('SNH1','SNH1E');
ylabel('SNH1,SNH1E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNH1,'k');ylabel('ERRSNH1');
xlabel('discrete time k')
set(figure(8),'name','Tank 1 Process variables and estimation
error','numbertitle','off')
% Tank 2
figure(9)
subplot(3,2,1);plot(k,SS2); hold on;plot(k,SS2E,'k-.');ylabel('SS2,SS2E');
legend('SS2','SS2E');subplot(3,2,2); plot(k,ERRSS2,'k');ylabel('ERRSS2');

subplot(3,2,3);plot(k,SNO2); hold on;plot(k,SNO2E,'k-.');
```

```
legend('SNO2','SNO2E');
ylabel('SNO2,SNO2E');
subplot(3,2,4); plot(k,ERRSNO2,'k');ylabel('ERRSNO2');

subplot(3,2,5);plot(k,SNH2); hold on;plot(k,SNH2E,'k-.');
legend('SNH2','SNH2E');
ylabel('SNH2,SNH2E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNH2,'k');ylabel('ERRSNH2');
xlabel('discrete time k')
set(figure(9),'name','Tank 2 Process variables and estimation
error','numbertitle','off')
% Tank 3
figure(10)
subplot(3,2,1);plot(k,SS3); hold on;plot(k,SS3E,'k-.');ylabel('SS3,SS3E');
legend('SS3','SS3E');subplot(3,2,2); plot(k,ERRSS3,'k');ylabel('ERRSS3');

subplot(3,2,3);plot(k,SNO3);hold on;plot(k,SNO3E,'k-.');
legend('SNO3','SNO3E');
ylabel('SNO3,SNO3E');subplot(3,2,4); plot(k,ERRSNO3,'k');ylabel('ERRSNO3');

subplot(3,2,5);plot(k,SNH3); hold on;plot(k,SNH3E,'k-.');
legend('SNH3','SNH3E');
ylabel('SNH3,SNH3E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNH3,'k');
ylabel('ERRSNH3');xlabel('discrete time k')
set(figure(10),'name','Tank 3 Process variables and estimation
error','numbertitle','off')
```

E.2: MATLAB script file – P_AASM1F.m

```matlab
% M-File: P_AASM1F.m
% ========================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Calculation of the process rates for the full model

% Description of the function
function Pk = P_AASM1F(xk)

% Definition of the global variables
global muH bH muA bA etag ka kh etah Ks KOH KNO KNH KOA KX;
% TANK 1
P(1) = muH*(xk(2) /(Ks+xk(2))) *(xk(8) /(KOH+xk(8))) * xk(5);
P(2) = muH*(xk(2) /(Ks+xk(2))) *(KOH/(KOH+xk(8))) * (xk(9) /(KNO+xk(9)))* etag * xk(5);
P(3) = muA*(xk(10)/(KNH+xk(10)))*(xk(8) /(KOA+xk(8))) * xk(6);
P(4) = bH*xk(5);
P(5) = bA*xk(6);
P(6) = ka*xk(11)*xk(5);
P(7) = kh*((xk(4) /(xk(5))) /(KX + (xk(4) /xk(5)))) *((xk(8) /(KOH+xk(8)))+ (etah *
(KOH/(KOH+xk(8)))) *(xk(9) /(KNO+xk(9)))) *xk(5);
P(8) = P(7) *(xk(12)/xk(4));

% TANK 2
P(9) = muH*(xk(15)/(Ks+xk(15)))*(xk(21)/(KOH+xk(21)))* xk(18);
P(10)= muH*(xk(15)/(Ks+xk(15)))*(KOH/(KOH+xk(21)))* (xk(22)/(KNO+xk(22)))* etag *
xk(18);
P(11)= muA*(xk(23)/(KNH+xk(23)))*(xk(21)/(KOA+xk(21)))* xk(19);
P(12)= bH*xk(18);
P(13)= bA*xk(19);
P(14)= ka*xk(24)*xk(18);
P(15)= kh*((xk(17)/(xk(18)))/(KX + (xk(17)/xk(18))))*((xk(21)/(KOH+xk(21)))+ (etah *
(KOH/(KOH+xk(21)))) *(xk(22)/(KNO+xk(22)))) *xk(18);
P(16)= P(15)*(xk(25)/xk(17));

% TANK 3
P(17)= muH*(xk(28)/(Ks+xk(28)))*(xk(34)/(KOH+xk(34)))* xk(31);
P(18)= muH*(xk(28)/(Ks+xk(28)))*(KOH/(KOH+xk(34)))* (xk(35)/(KNO+xk(35)))* etag *
xk(31);
P(19)= muA*(xk(36)/(KNH+xk(36)))*(xk(34)/(KOA+xk(34)))* xk(32);
P(20)= bH*xk(31);
P(21)= bA*xk(32);
P(22)= ka*xk(37)*xk(31);
P(23)= kh*((xk(30)/(xk(31)))/(KX + (xk(30)/xk(31))))*((xk(34)/(KOH+xk(34)))+ (etah *
(KOH/(KOH+xk(34)))) *(xk(35)/(KNO+xk(35)))) *xk(31);
P(24)= P(23)*(xk(38)/xk(30));

Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20) P(21) P(22) P(23) P(24)]';
```

E.3: MATLAB script file – P_AASM1R.m

```matlab
% M-File: P_AASM1R.m
% ========================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model

% Description of the function
function Pk = P_AASM1R(zk,uk,XBHk,XBAk,XSk)

% Definition of the global variables
global etag etah Ks KOH KNO KNH KOA KX;
% TANK 1
```

```matlab
P(1)= (zk(1)/(Ks+zk(1)))*(uk(1)/(KOH+uk(1)))* XBHk(1);
P(2)= (zk(1)/(Ks+zk(1)))*(KOH/(KOH+uk(1)))* (zk(2)/(KNO+zk(2)))* etag* XBHk(1);
P(3)= (zk(3)/(KNH+zk(3)))*(uk(1)/(KOA+uk(1)))* XBAk(1);
P(4)= ((XSk(1)/XBHk(1))/(KX + (XSk(1)/XBHk(1))))*((uk(1)/(KOH+uk(1)))+etah *
(KOH/(KOH+uk(1)))*(zk(2)/(KNO+zk(2))))* XBHk(1);
P1k = [P(1) 0 0; P(2) 0 0; 0 P(3) 0; 0 0 P(4)];

% TANK 2
P(5)= (zk(4)/(Ks+zk(4)))*(uk(2)/(KOH+uk(2)))* XBHk(2);
P(6)= (zk(4)/(Ks+zk(4)))*(KOH/(KOH+uk(2)))* (zk(5)/(KNO+zk(5)))* etag* XBHk(2);
P(7)= (zk(6)/(KNH+zk(6)))*(uk(2)/(KOA+uk(2)))* XBAk(2);
P(8)= ((XSk(2)/XBHk(2))/(KX + (XSk(2)/XBHk(2))))*((uk(2)/(KOH+uk(2)))+etah *
(KOH/(KOH+uk(2)))*(zk(5)/(KNO+zk(5))))*XBHk(2);
P2k = [P(5) 0 0; P(6) 0 0; 0 P(7) 0; 0 0 P(8)];

% TANK 3
P(9) = (zk(7)/(Ks+zk(7)))*(uk(3)/(KOH+uk(3)))* XBHk(3);
P(10)= (zk(7)/(Ks+zk(7)))*(KOH/(KOH+uk(3)))* (zk(8)/(KNO+zk(8))) * etag * XBHk(3);
P(11)= (zk(9)/(KNH+zk(9)))*(uk(3)/(KOA+uk(3)))* XBAk(3);
P(12)= ((XSk(3)/XBHk(3))/(KX + (XSk(3)/XBHk(3))))*((uk(3)/(KOH+uk(3)))+ etah *
(KOH/(KOH+uk(3)))*(zk(8)/(KNO+zk(8))))*XBHk(3);
P3k = [P(9) 0 0; P(10) 0 0; 0 P(11) 0; 0 0 P(12)];

% Matrix Pk for the whole process
Pk = [P1k; P2k; P3k];
```

E.4: MATLAB script file – P_AASM1RE.m

```matlab
% M-File: P_AASM1RE.m
% =========================== M-FILE DESCRIPTION ===========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model

% Description of the function function
function Pk = P_AASM1RE(zk,uk,XBHk,XBAk,XSk,muHEk,muAEk,KhEk)

global etag etah Ks KOH KNO KNH KOA KX
% TANK 1
P(1)= muHEk * (zk(1)/(Ks+zk(1)))*(uk(1)/(KOH+uk(1)))* XBHk(1);
P(2)= muHEk * (zk(1)/(Ks+zk(1)))*(KOH/(KOH+uk(1)))* (zk(2)/(KNO+zk(2)))* etag *
XBHk(1);
P(3)= muAEk * (zk(3)/(KNH+zk(3)))*(uk(1)/(KOA+uk(1)))* XBAk(1);
P(4)= KhEk *((XSk(1)/XBHk(1))/(KX + (XSk(1)/XBHk(1))))*((uk(1)/(KOH+uk(1)))+ etah *
(KOH/(KOH+uk(1)))*(zk(2)/(KNO+zk(2))))* XBHk(1);

% TANK 2
P(5)= muHEk * (zk(4)/(Ks+zk(4)))*(uk(2)/(KOH+uk(2)))* XBHk(2);
P(6)= muHEk * (zk(4)/(Ks+zk(4)))*(KOH/(KOH+uk(2)))* (zk(5)/(KNO+zk(5)))* etag *
XBHk(2);
P(7)= muAEk * (zk(6)/(KNH+zk(6)))*(uk(2)/(KOA+uk(2)))* XBAk(2);
P(8)= KhEk*((XSk(2)/XBHk(2))/(KX + (XSk(2)/XBHk(2))))*((uk(2)/(KOH+uk(2)))+ etah *
(KOH/(KOH+uk(2)))*(zk(5)/(KNO+zk(5))))*XBHk(2);

% TANK 3
P(9) = muHEk * (zk(7)/(Ks+zk(7)))*(uk(3)/(KOH+uk(3)))* XBHk(3);
P(10)= muHEk * (zk(7)/(Ks+zk(7)))*(KOH/(KOH+uk(3)))* (zk(8)/(KNO+zk(8)))* etag *
XBHk(3);
P(11)= muAEk * (zk(9)/(KNH+zk(9)))*(uk(3)/(KOA+uk(3)))* XBAk(3);
P(12)= KhEk*((XSk(3)/XBHk(3))/(KX + (XSk(3)/XBHk(3))))*((uk(3)/(KOH+uk(3)))+ etah *
(KOH/(KOH+uk(3)))*(zk(8)/(KNO+zk(8))))*XBHk(3);

%Matrix Pk for the whole process
Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12)]';
```

APPENDIX F: DEVELOPED MATLAB CODE FOR OPTIMAL CONTROL OF THE ATHLONE PLANT WITH THE ACTIVATED SLUDGE MODEL NO.2 BIO-MODEL

F.1: MATLAB script file – AASM2_Simulation.m

```
% M-File: AASM2_Simulation.m
% ========================= M-FILE DESCRIPTION ==========================
% This m-file is used for:
% Simulation of the Athlone plant using full ASM2 model
% Calculation of the parameters for the reduced ASM2 model
% Simulation of the Athlone plant using the reduced ASM2 model

clear all          % Clear the workspace
clc                % Clear the command window


% ========================= PARAMETER DEFINITION ========================
% Declaration of the global variables
global fSI YH fXI YPO4 YPHA iPBM YA iNBM Kh etaNO3 etafe muH qfe bH qPHA
global qPP muPAO bPAO bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4
global KALK KA Kfe KPP KPS KP KPHA KIPP KMAX

K1  = 11;            % Prediction horizon or interval
DT = 0.01042;        % Sampling time

% Initialization of the model parameters, typical values at 20 0C
% Stochiometric parameters
fSI      = 0.0;
YH       = 0.625;
fXI      = 0.10;
%YPAO = 0.625;
YPO4     = 0.40;
YPHA     = 0.20;
iPBM     = 0.02;
YA       = 0.24;
iNBM     = 0.07;
% Kinetic parameters
Kh       = 3;
etaNO3   = 0.8;
etafe    = 0.40;
muH      = 0.6;
qfe      = 3;
bH       = 0.4;
qPHA     = 3;
qPP      = 1.5;
muPAO    = 1;
bPAO     = 0.2;
bPP      = 0.2;
bPHA     = 0.2;
muAUT    = 1;
bAUT     = 0.15;
kPRE     = 1;
kRED     = 0.6;
% Monod switching functions
KO2      = 0.2;
KX       = 0.1;
KNO3     = 0.5;
KF       = 4;
KNH4     = 0.05;
KALK     = 0.1;
KA       = 4;
Kfe      = 20;
KPP      = 0.01;
KPS      = 0.20;
KP       = 0.01;
```

```
KPHA     = 0.01;
KIPP     = 0.02;
KMAX     = 0.34;


% Initialization of the Athlone process data
V1  = 1148;          % Tank 1 volume
V2  = 1148;          % Tank 2 volume
V3  = 5273;          % Tank 3 volume


Q0  = 40003;         % Influent flow rate
Qa  = 39916.8;       % Recycled flow rate from tank 3
Qp  = 39916.8;       % Recycled flow rate from tank 2
Qr  = 40003;         % Recycled flow rate from the settler
Qw  = 38400.0;       % Waste flow rate (3840)
Q1  = Qp + Q0;       % flow through the tank 1
Q2  = Qp + Q0 + Qa + Qr;    % flow through the tank 2
Q3  = Q2;                   % flow through the tank 3


% Coeficient for fraction of flow rates for the particulate materials
LAMBDA = (Qp+Q0+Qr-Qw)/(Qr);


% Oxygen parameters
SOsat  = 8;          % Oxygen saturation point
KLa    = 4.8;        % Oxygen transfer function in Tank 1


% ============================ PROCESS MODEL ============================
% Calculation of the MATRIX A
% Matrix A represent weighting factors of the accumulated state variables
% from different flows towards state variables in each tank as derived
% through mass balance principles, see chapter 3 of the Thesis
IV1 = DT/ V1; IVi1 = IV1 * Q1; A11 = 1 - IVi1;
IV2 = DT/ V2; IV22 = IV2 * Q2; A22 = 1 - IV22;
IV3 = DT/ V3; IV33 = IV3 * Q3; A33 = 1 - IV33;


A12 = IV1 * Qp;
A23a = IV2 * (Qa + Qr);
A23b = IV2 * (Qa + LAMBDA*Qr);
A21 = IV2*Q1;
A32 = IV3*Q2;


A11 = A11 * ones(1,19); A11 = diag(A11);
A12 = A12 * ones(1,19); A12 = diag(A12);
A13 = zeros(19,19);


A21 = A21 * ones(1,19); A21 = diag(A21);
A22 = A22 * ones(1,19); A22 = diag(A22);
A23 =
[A23a,A23a,A23a,A23a,A23a,A23a,A23a,A23a,A23a,A23b,A23b,A23b,A23b,A23b,A23b,A23b,A
23b,A23b,A23b];
A23 = diag(A23);


A31 = zeros(19,19);
A32 = A32 * ones(1,19); A32 = diag(A32);
A33 = A33 * ones(1,19); A33 = diag(A33);


A = [A11 A12 A13; A21 A22 A23; A31 A32 A33];

% Calculation of the MATRIX B
% Matrix B represent weighting factors of the state variables towards the
% influent wastewater composition as derived through mass balance
% principles, see chapter 3 of the Thesis
IV0 = DT/ V1*Q0;
B11 = IV0 * ones(1,19); B11 = diag(B11);


B = [B11; zeros(38,19)];
```

277

```
% Formulation of the MATRIX C
% Matrix C represent weighting factors of the state variables towards the
% processes on the ASM1 matrix table
%     SO2   SF    SA      SNH4   SN03    SPO4    SI      SALK    SN2   XI
%     XS         XH      XPAO   XPP     XPHA    XAUT    XTSS    XMeOH   XMeP
CC = [[0   1    0 0.01    0 0    0 0.001    0 0  -1    0 0 0 0 0     -0.75    0     0];
      [0   1    0 0.01    0 0    0 0.001    0 0  -1    0 0 0 0 0     -0.75    0     0];
      [0   1    0 0.01    0 0    0 0.001    0 0  -1    0 0 0 0 0     -0.75    0     0];
      [-0.59 -1.59  0 -0.022    0 -0.004  0 -0.001 0 0 0 1 0      0      0      0     0.9      0
0];
      [-0.59  0  -1.59   -0.0070   0 -0.02  0 0.021 0 0 0 1 0 0      0      0     0.9      0
0];
      [0  -1.59  0 -0.022    -0.21 -0.004  0 0.014 0.21  0 0 1 0 0      0      0     0.9      0
0];
      [0   0  -1.59   -0.07    -0.21 -0.02  0 0.036 0.21  0 0 1 0 0 0      0     0.9      0
0];
      [0  -1 1 0.03     0 0.01  0 0.014 0 0 0      0      0      0      0      0      0      0];
      [0   0    0 0.031    0 0.01  0 0.002 0 0.10 0.9  -1    0 0      0      0     -0.15     0
0];
      [0   0  -1    0      0 0.4   0 -0.004 0 0 0 0 0      -0.4 1      0     -0.69     0     0];
      [-0.2 0  0 0      0 -1  0 0.048 0 0 0 0      1      -0.20   0      3.11     0     0];
      [-0.6 0  0 -0.07    0 -0.02 0 -0.0004 0 0 0 0 1      0     -1.6 0     -0.06     0
0];
      [0   0    0 0.031    0 0.01  0 0.002 0 0.10 0.9  0 -1    0 0      0     -0.15     0     0];
      [0   0    0 0      0 1    0 -0.048 0 0 0      0      0     -1    0      0     -3.23     0     0];
      [0   0    1 0      0 0    0 -0.0016 0 0 0      0      0      0     -1    0     -0.6 0     0];
      [-18  0    0 -4.24   4.17 -0.02 0 -0.6 0 0 0 0 0 0      1      0.9     0     0];
      [0   0    0 0.031    0 0.01  0 0.002 0 0.10 0.9  0 0 0      -1    -0.15     0     0];
      [0   0    0 0      0 -1  0 0 0.048    0 0 0 0 0      0      0      1.42 -3.45 4.87];
      [0   0    0 0      0 1    0 -0.048 0 0 0 0 0 0      0     -1.42   3.45 -4.87]];

  C = DT*[CC zeros(19,38); zeros(19,19) CC zeros(19,19); zeros(19,38) CC];

% ============================== SIMULATION DATA ==============================
% Average values of the inflow concentrations based on the dry weather file
SO2i  = 0.0;     SFi   = 30;    SAi   = 20; SNH4i = 31.56; SNO3i = 0.0;
SPO4i = 3.6;     SIi   = 30;    SALKi = 7;    SN2i  = 6.95;   XIi   = 51.2;
XSi   = 202.32;  XHi   = 28.17; XPAOi = 0.0; XPPi  = 0.0;     XPHAi = 0.0;
XAUTi = 0.0;     XTSSi = 180;   XMeOHi = 0.0; XMePi = 0.0;

xi = [SO2i*ones(1,K1); SFi*ones(1,K1); SAi*ones(1,K1); SNH4i*ones(1,K1);
      SNO3i*ones(1,K1); SPO4i*ones(1,K1); SIi*ones(1,K1); SALKi*ones(1,K1);
      SN2i*ones(1,K1); XIi*ones(1,K1); XSi*ones(1,K1); XHi*ones(1,K1);
      XPAOi*ones(1,K1); XPPi*ones(1,K1); XPHAi*ones(1,K1); XAUTi*ones(1,K1);
      XTSSi*ones(1,K1); XMeOHi*ones(1,K1); XMePi*ones(1,K1)];

% Values of the initial conditions
% For tank 1
SO2_1  = 4.2984433e-03; SF_1   = 2.8082131e+00; SA_1   = 2.8082131e+00;
SNH4_1 = 7.9178845e+00; SNO3_1 = 5.3699400e+00; SPO4_1 = 3.6;
SI_1   = 3.0000000e+01; SALK_1 = 4.9277103e+00; SN2_1  = 1.2166405e+00;
XI_1   = 1.1491252e+03; XS_1   = 8.2134908e+01; XH_1   = 2.5517658e+03;
XPAO_1 = 0.1; XPP_1   = 0.1; XPHA_1  = 0.1; XAUT_1  = 1.4838943e+02;
XTSS_1 = 180;XMeOH_1 = 0.1;XMeP_1  = 0.1;
x10 = [SO2_1; SF_1; SA_1; SNH4_1; SNO3_1; SPO4_1; SI_1;   SALK_1; SN2_1;
       XI_1;  XS_1; XH_1; XPAO_1; XPP_1;  XPHA_1; XAUT_1; XTSS_1; XMeOH_1;
       XMeP_1];

% For tank 2
SO2_2  = 6.3131911e-05; SF_2   = 1.4587940e+00; SA_2   = 1.4587940e+00;
SNH4_2 = 8.3444148e+00; SNO3_2 = 3.6619672e+00; SPO4_2 = 3.6;
SI_2   = 3.0000000e+01; SALK_2 = 5.0801748e+00; SN2_2  = 8.8206477e-01;
XI_2   = 1.1491252e+03; XS_2   = 7.6386187e+01; XH_2   = 2.5533851e+03;
XPAO_2 = 0.1;XPP_2   = 0.1; XPHA_2 = 0.1; XAUT_2 = 1.4830914e+02;
XTSS_2 = 180;XMeOH_2 = 0.1; XMeP_2 = 0.1;
x20 = [SO2_2; SF_2; SA_2; SNH4_2; SNO3_2; SPO4_2; SI_2;   SALK_2; SN2_2;
```

```
                XI_2;   XS_2; XH_2; XPAO_2; XPP_2;  XPHA_2; XAUT_2; XTSS_2; XMeOH_2;
                XMeP_2];

% For tank 3
SO2_3   = 1.7183778e+00; SF_3    = 1.1495418e+00; SA_3    = 1.1495418e+00;
SNH4_3  = 5.5479452e+00; SNO3_3  = 6.5408820e+00; SPO4_3  = 3.6;
SI_3    = 3.0000000e+01; SALK_3  = 4.6747902e+00; SN2_3   = 8.2888682e-01;
XI_3    = 1.1491252e+03; XS_3    = 6.4854922e+01; XH_3    = 2.5571314e+03;
XPAO_3  = 0.1; XPP_3    = 0.1; XPHA_3   = 0.1; XAUT_3    = 1.4894126e+02;
XTSS_3  = 180; XMeOH_3  = 0.1; XMeP_3   = 0.1;
x30 = [SO2_3; SF_3; SA_3; SNH4_3; SNO3_3; SPO4_3; SI_3;    SALK_3; SN2_3;
        XI_3;   XS_3; XH_3; XPAO_3; XPP_3;  XPHA_3; XAUT_3; XTSS_3; XMeOH_3;
        XMeP_3];


x0 = [x10;x20;x30];
x = [x0 zeros(57,K1)];

% Steady state simulation of the model
for k = 1:K1
    xk  = x(:,k);
    xik = xi(:,k);
    Pk   = P_AASM2F(xk);

    SOk  = [xk(39,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xx1k = [zeros(38,1);xxk(1);zeros(18,1)];
    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xx1k;
end


% Full model response to the steady state influent concentrations
% Tank 1
figure(1)
subplot(5,4,1);plot(x(1,:));title('SO2_1')
subplot(5,4,2);plot(x(2,:));title('SF1')
subplot(5,4,3);plot(x(3,:));title('SA1')
subplot(5,4,4);plot(x(4,:));title('SNH4_1')
subplot(5,4,5);plot(x(5,:));title('SNO3_1')
subplot(5,4,6);plot(x(6,:));title('SPO4_1')
subplot(5,4,7);plot(x(7,:));title('SI1')
subplot(5,4,8);plot(x(8,:));title('SALK1')
subplot(5,4,9);plot(x(9,:));title('SN2_1')
subplot(5,4,10);plot(x(10,:));title('XI1');
subplot(5,4,11);plot(x(11,:));title('XS1')
subplot(5,4,12);plot(x(12,:));title('XH1')
subplot(5,4,13);plot(x(13,:));title('XPAO1')
subplot(5,4,14);plot(x(14,:));title('XPP1')
subplot(5,4,15);plot(x(15,:));title('XPHA1')
subplot(5,4,16);plot(x(16,:));title('XAUT1')
xlabel('discrete time k')
subplot(5,4,17);plot(x(17,:));title('XTSS1')
xlabel('discrete time k')
subplot(5,4,18);plot(x(18,:));title('XMeOH1')
xlabel('discrete time k')
subplot(5,4,19);plot(x(19,:));title('XMeP1')
xlabel('discrete time k')
set(figure(1),'name','Tank 1 steady state results','numbertitle','off')

% Tank 2
figure(2)
subplot(5,4,1);plot(x(20,:));title('SO2_2')
subplot(5,4,2);plot(x(21,:));title('SF2')
subplot(5,4,3);plot(x(22,:));title('SA2')
subplot(5,4,4);plot(x(23,:));title('SNH4_2')
subplot(5,4,5);plot(x(24,:));title('SNO3_2')
subplot(5,4,6);plot(x(25,:));title('SPO4_2')
subplot(5,4,7);plot(x(26,:));title('SI2')
```

```
subplot(5,4,8);plot(x(27,:));title('SALK2')
subplot(5,4,9);plot(x(28,:));title('SN2_2')
subplot(5,4,10);plot(x(29,:));title('XI2');
subplot(5,4,11);plot(x(30,:));title('XS2')
subplot(5,4,12);plot(x(31,:));title('XH2')
subplot(5,4,13);plot(x(32,:));title('XPAO2')
subplot(5,4,14);plot(x(33,:));title('XPP2')
subplot(5,4,15);plot(x(34,:));title('XPHA2')
subplot(5,4,16);plot(x(35,:));title('XAUT2')
xlabel('discrete time k')
subplot(5,4,17);plot(x(36,:));title('XTSS2')
xlabel('discrete time k')
subplot(5,4,18);plot(x(37,:));title('XMeOH2')
xlabel('discrete time k')
subplot(5,4,19);plot(x(38,:));title('XMeP2')
xlabel('discrete time k')
set(figure(2),'name','Tank 2 steady state results','numbertitle','off')

% Tank 3
figure(3)
subplot(5,4,1);plot(x(39,:));title('SO2_3')
subplot(5,4,2);plot(x(40,:));title('SF3')
subplot(5,4,3);plot(x(41,:));title('SA3')
subplot(5,4,4);plot(x(42,:));title('SNH4_3')
subplot(5,4,5);plot(x(43,:));title('SNO3_3')
subplot(5,4,6);plot(x(44,:));title('SPO4_3')
subplot(5,4,7);plot(x(45,:));title('SI3')
subplot(5,4,8);plot(x(46,:));title('SALK3')
subplot(5,4,9);plot(x(47,:));title('SN2_3')
subplot(5,4,10);plot(x(48,:));title('XI3');
subplot(5,4,11);plot(x(49,:));title('XS3')
subplot(5,4,12);plot(x(50,:));title('XH3')
subplot(5,4,13);plot(x(51,:));title('XPAO3')
subplot(5,4,14);plot(x(52,:));title('XPP3')
subplot(5,4,15);plot(x(53,:));title('XPHA3')
subplot(5,4,16);plot(x(54,:));title('XAUT3')
xlabel('discrete time k')
subplot(5,4,17);plot(x(55,:));title('XTSS3')
xlabel('discrete time k')
subplot(5,4,18);plot(x(56,:));title('XMeOH3')
xlabel('discrete time k')
subplot(5,4,19);plot(x(57,:));title('XMeP3')
xlabel('discrete time k')
set(figure(3),'name','Tank 3 steady state results','numbertitle','off')

% Data for dynamic simulations
K    = 11;      % Prediction horizon for dynamic simulations

% Dynamic of the inflow concentrations for dry weather file
load dry.txt

% The order is t, SS XBH XS XI SNH SI SND XND Q0
%               1, 2,  3, 4, 5,  6, 7,  8,  9, 10

xi = [SO2i*ones(1,K); SFi*ones(1,K); SAi*ones(1,K); dry(1:K,6)';
      SNO3i*ones(1,K); SPO4i*ones(1,K); dry(1:K,7)'; SALKi*ones(1,K);
      SN2i*ones(1,K); dry(1:K,5)'; dry(1:K,4)'; dry(1:K,3)';
      XPAOi*ones(1,K); XPPi*ones(1,K); XPHAi*ones(1,K); XAUTi*ones(1,K);
      XTSSi*ones(1,K); XMeOHi*ones(1,K); XMePi*ones(1,K)];

x = [x(:,K1+1) zeros(57,K)];

% Dynamic simulation of the model
for k = 1:K
    xk  = x(:,k);
    xik = xi(:,k);
```

```
    Pk   = P_AASM2F(xk);

    SOk  = [xk(39,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xx1k = [zeros(38,1);xxk(1);zeros(18,1)];
    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xx1k;
end

% Full model response to the dynamic influent concentrations
% Tank 1
figure(4)
subplot(5,4,1);plot(x(1,:));title('SO2_1')
subplot(5,4,2);plot(x(2,:));title('SF1')
subplot(5,4,3);plot(x(3,:));title('SA1')
subplot(5,4,4);plot(x(4,:));title('SNH4_1')
subplot(5,4,5);plot(x(5,:));title('SNO3_1')
subplot(5,4,6);plot(x(6,:));title('SPO4_1')
subplot(5,4,7);plot(x(7,:));title('SI1')
subplot(5,4,8);plot(x(8,:));title('SALK1')
subplot(5,4,9);plot(x(9,:));title('SN2_1')
subplot(5,4,10);plot(x(10,:));title('XI1');
subplot(5,4,11);plot(x(11,:));title('XS1')
subplot(5,4,12);plot(x(12,:));title('XH1')
subplot(5,4,13);plot(x(13,:));title('XPAO1')
subplot(5,4,14);plot(x(14,:));title('XPP1')
subplot(5,4,15);plot(x(15,:));title('XPHA1')
subplot(5,4,16);plot(x(16,:));title('XAUT1')
xlabel('discrete time k')
subplot(5,4,17);plot(x(17,:));title('XTSS1')
xlabel('discrete time k')
subplot(5,4,18);plot(x(18,:));title('XMeOH1')
xlabel('discrete time k')
subplot(5,4,19);plot(x(19,:));title('XMeP1')
xlabel('discrete time k')
set(figure(4),'name','Tank 1 dynamic results','numbertitle','off')


% Tank 2
figure(5)
subplot(5,4,1);plot(x(20,:));title('SO2_2')
subplot(5,4,2);plot(x(21,:));title('SF2')
subplot(5,4,3);plot(x(22,:));title('SA2')
subplot(5,4,4);plot(x(23,:));title('SNH4_2')
subplot(5,4,5);plot(x(24,:));title('SNO3_2')
subplot(5,4,6);plot(x(25,:));title('SPO4_2')
subplot(5,4,7);plot(x(26,:));title('SI2')
subplot(5,4,8);plot(x(27,:));title('SALK2')
subplot(5,4,9);plot(x(28,:));title('SN2_2')
subplot(5,4,10);plot(x(29,:));title('XI2');
subplot(5,4,11);plot(x(30,:));title('XS2')
subplot(5,4,12);plot(x(31,:));title('XH2')
subplot(5,4,13);plot(x(32,:));title('XPAO2')
subplot(5,4,14);plot(x(33,:));title('XPP2')
subplot(5,4,15);plot(x(34,:));title('XPHA2')
subplot(5,4,16);plot(x(35,:));title('XAUT2')
xlabel('discrete time k')
subplot(5,4,17);plot(x(36,:));title('XTSS2')
xlabel('discrete time k')
subplot(5,4,18);plot(x(37,:));title('XMeOH2')
xlabel('discrete time k')
subplot(5,4,19);plot(x(38,:));title('XMeP2')
xlabel('discrete time k')
set(figure(5),'name','Tank 2 dynamic results','numbertitle','off')

% Tank 3
figure(6)
subplot(5,4,1);plot(x(39,:));title('SO2_3')
```

281

```
subplot(5,4,2);plot(x(40,:));title('SF3')
subplot(5,4,3);plot(x(41,:));title('SA3')
subplot(5,4,4);plot(x(42,:));title('SNH4_3')
subplot(5,4,5);plot(x(43,:));title('SNO3_3')
subplot(5,4,6);plot(x(44,:));title('SPO4_3')
subplot(5,4,7);plot(x(45,:));title('SI3')
subplot(5,4,8);plot(x(46,:));title('SALK3')
subplot(5,4,9);plot(x(47,:));title('SN2_3')
subplot(5,4,10);plot(x(48,:));title('XI3');
subplot(5,4,11);plot(x(49,:));title('XS3')
subplot(5,4,12);plot(x(50,:));title('XH3')
subplot(5,4,13);plot(x(51,:));title('XPAO3')
subplot(5,4,14);plot(x(52,:));title('XPP3')
subplot(5,4,15);plot(x(53,:));title('XPHA3')
subplot(5,4,16);plot(x(54,:));title('XAUT3')
xlabel('discrete time k')
subplot(5,4,17);plot(x(55,:));title('XTSS3')
xlabel('discrete time k')
subplot(5,4,18);plot(x(56,:));title('XMeOH3')
xlabel('discrete time k')
subplot(5,4,19);plot(x(57,:));title('XMeP3')
xlabel('discrete time k')
set(figure(6),'name','Tank 3 dynamic results','numbertitle','off')

% ===================== REDUCED MODEL FORMULATION =====================
% Form the vector of the states (z) for the reduced model, based on the
% ASM1 prediction
%         SF,       SA,      SNH4,     SNO3,       SPO4,    XS,      XPP,     XPHA
z1 = [x(2,:); x(3,:); x(4,:); x(5,:);  x(6,:);  x(11,:); x(14,:); x(15,:)];
z2 = [x(21,:);x(22,:);x(23,:);x(24,:); x(25,:); x(30,:); x(33,:); x(34,:)];
z3 = [x(40,:); x(41,:);x(42,:);x(43,:); x(44,:); x(49,:);x(52,:); x(53,:)];
z  = [z1;z2;z3];

% Formation of the vector of the control variables
u1 = x(1,:);
u2 = x(20,:);
u3 = x(39,:);
u  = [u1; u2; u3];

% Dynamic values of the inflow concentrations for dry weather
zi = [SFi*ones(1,K); SAi*ones(1,K); SNH4i*ones(1,K); SNO3i*ones(1,K);
      SPO4i*ones(1,K); XSi*ones(1,K); XPPi*ones(1,K); XPHAi*ones(1,K)];

% Steady state values for the biomass XH, XAUT, SALK, XPAO
XH   = [2551.76*ones(1,K); 2553.38*ones(1,K); 2557.13*ones(1,K);
         2559.18*ones(1,K); 2559.34*ones(1,K)];
XAUT = [148.389*ones(1,K); 148.309*ones(1,K); 148.941*ones(1,K);
         149.527*ones(1,K); 149.797*ones(1,K)];
SALK = [4.9*ones(1,K); 148.309*ones(1,K); 148.941*ones(1,K);
         149.527*ones(1,K); 149.797*ones(1,K)];
XPAO = [0.1*ones(1,K); 76.386*ones(1,K); 64.855*ones(1,K);
         55.694*ones(1,K); 49.306*ones(1,K)];

% MATRIX A for the reduced model (AR)
A11r = 1 - IV11; A22r = 1 - IV22; A33r = 1 - IV33;

A12r = IV1 * Qp;
A23ar = IV2 * (Qa + Qr); A23br = IV2 * (Qa + LAMBDA*Qr);

A21r = IV2*Q1; A32r = IV3*Q2;

A11r = A11r * ones(1,8); A11r = diag(A11r);
A12r = A12r * ones(1,8); A12r = diag(A12r);
A13r = zeros(8,8);

A21r = A21r * ones(1,8); A21r = diag(A21r);
```

282

```matlab
A22r = A22r * ones(1,8); A22r = diag(A22r);
A23r = [A23ar,A23ar,A23ar,A23ar,A23ar,A23br,A23br,A23br];
A23r = diag(A23r);

A31r = zeros(8,8);
A32r = A32r * ones(1,8); A32r = diag(A32r);
A33r = A33r * ones(1,8); A33r = diag(A33r);

AR = [A11r A12r A13r; A21r A22r A23r; A31r A32r A33r];

% MATRIX B for the reduced model (BR)
IV0 = DT / V1*Q0;
f = 1.2;
B11r = [IV0*ones(1,3) IV0*f IV0*ones(1,4)];
B11r = diag(B11r);
BR = [B11r; zeros(16,8)];

% MATRIX C for the reduced model (CR)
%      SF     SA      SNH4    SNO3    SPO4      XS       XPP     XPHA
CCR = [[1   0 0.01    0 0   -1     0 0];
       [1   0 0.01    0 0   -1     0 0];
       [1   0 0.01    0 0   -1     0 0];
       [-1.59  0 -0.022   0 -0.004   0 0    0];
       [0  -1.59    -0.0070  0 -0.02   0 0    0];
       [-1.59  0 -0.022    -0.21 -0.004   0 0   0];
       [0  -1.59    -0.07   -0.21 -0.02   0 0 0];
       [-1 1 0.03    0 0.01   0      0   0];
       [0  -1    0    0 0.4    0      -0.4 1];
       [0   0 0    0 -1    0      1   -0.20];
       [0   0 -0.07   0 -0.02  0      0    -1.6];
       [0   0 -4.24   4.17  -0.02  0  0 0]];
CR = DT*[CCR zeros(12,16); zeros(12,8) CCR zeros(12,8); zeros(12,16) CCR];

% Parameter estimation
for k = 1:K
    zk = z(:,k);
    uk = u(:,k);

    XHk  = XH(:,k);
    XAUTk = XAUT(:,k);
    XPAOk = XPAO(:,k);
    SALKk = SALK(:,k);

    zik  = zi(:,k);
    Pk = P_AASM2R(zk,uk,XHk,XAUTk,XPAOk,SALKk);
    FIk = CR'*Pk;
    Y(:,k) = z(:,k+1)-AR*z(:,k)-BR*zi(:,k);
    theta(:,k)=(FIk'*FIk)^(-1)*FIk'*Y(:,k);
end

kHe(1,:)     = theta(1,:);
muHe(1,:)    = theta(2,:);
qfee(1,:)    = theta(3,:);
qPHAe(1,:)   = theta(4,:);
qPPe(1,:)    = theta(5,:);
muPAOe(1,:)  = theta(6,:);
muAUTe(1,:)  = theta(7,:);

% Dynamic simulation of the reduced model with the estimated parameters
zE = [x(2,1);x(3,1);x(4,1);x(5,1);x(6,1);x(11,1);x(14,1);x(15,1);x(21,1);x(22,1);
x(23,1); x(24,1);x(25,1); x(30,1);
x(33,1);x(34,1);x(40,1);x(41,1);x(42,1);x(43,1);x(44,1);x(45,1);x(52,1);x(53,1)];
ERR = z(:,1)-zE;

% Dynamic simulation of the reduced model with the estimated parameters
for k = 1:K
```

283

```
        zk = z(:,k);
        uk = u(:,k);
        zik = zi(:,k);

        XHk = XH(:,k);
        XAUTk = XAUT(:,k);
        XPAOk = XPAO(:,k);
        SALKk = SALK(:,k);

        kHek     = kHe(:,k);
        muHek    = muHe(:,k);
        qfeek    = qfee(:,k);
        qPHAek   = qPHAe(:,k);
        qPPek    = qPPe(:,k);
        muPAOek  = muPAOe(:,k);
        muAUTek  = muAUTe(:,k);

        Pk =
P_AASM2RE(zk,uk,XHk,XAUTk,XPAOk,SALKk,kHek,muHek,qfeek,qPHAek,qPPek,muPAOek,muAUTek);
        zE(:,k+1) = AR*zE(:,k) + CR'*Pk + BR*zi(:,k);
        ERR(:,k+1)=z(:,k+1)-zE(:,k+1);
end

% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 1
SF1=z(1,:);SA1=z(2,:);SNH4_1=z(3,:);SNO3_1=z(4,:);
SF1E=zE(1,:);SA1E=zE(2,:);SNH4_1E=zE(3,:);SNO3_1E=zE(4,:);
ERRSF1=ERR(1,:);ERRSA1=ERR(2,:);ERRSNH4_1=ERR(3,:);ERRSNO3_1=ERR(4,:);

SPO4_1=z(5,:);XS1=z(6,:);XPP1=z(7,:);XPHA1=z(8,:);
SPO4_1E=zE(5,:);XS1E=zE(6,:);XPP1E=zE(7,:);XPHA1E=zE(8,:);
ERRSPO4_1=ERR(5,:);ERRXS1=ERR(6,:);ERRXPP1=ERR(7,:);ERRXPHA1=ERR(8,:);

% Calculation of the weighted errors for tank 1
eSF1=(ERRSF1*ERRSF1')/(SF1*SF1');
eSA1=(ERRSA1*ERRSA1')/(SA1*SA1');
eSNH4_1=(ERRSNH4_1*ERRSNH4_1')/(SNH4_1*SNH4_1');
eSNO3_1=(ERRSNO3_1*ERRSNO3_1')/(SNO3_1*SNO3_1');
eSPO4_1=(ERRSPO4_1*ERRSPO4_1')/(SPO4_1*SPO4_1');
eXS1=(ERRXS1*ERRXS1')/(XS1*XS1');
eXPP1=(ERRXPP1*ERRXPP1')/(XPP1*XPP1');
eXPHA1=(ERRXPHA1*ERRXPHA1')/(XPHA1*XPHA1');

% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 2
SF2=z(9,:);SA2=z(10,:);SNH4_2=z(11,:);SNO3_2=z(12,:);
SF2E=zE(9,:);SA2E=zE(10,:);SNH4_2E=zE(11,:);SNO3_2E=zE(12,:);
ERRSF2=ERR(9,:);ERRSA2=ERR(10,:);ERRSNH4_2=ERR(11,:);ERRSNO3_2=ERR(12,:);

SPO4_2=z(13,:);XS2=z(14,:);XPP2=z(15,:);XPHA2=z(16,:);
SPO4_2E=zE(13,:);XS2E=zE(14,:);XPP2E=zE(15,:);XPHA2E=zE(16,:);
ERRSPO4_2=ERR(13,:);ERRXS2=ERR(14,:);ERRXPP2=ERR(15,:);ERRXPHA2=ERR(16,:);

% Calculation of the weighted errors for tank 2
eSF2=(ERRSF2*ERRSF2')/(SF2*SF2');
eSA2=(ERRSA2*ERRSA2')/(SA2*SA2');
eSNH4_2=(ERRSNH4_2*ERRSNH4_2')/(SNH4_2*SNH4_2');
eSNO3_2=(ERRSNO3_2*ERRSNO3_2')/(SNO3_2*SNO3_2');
eSPO4_2=(ERRSPO4_2*ERRSPO4_2')/(SPO4_2*SPO4_2');
eXS2=(ERRXS2*ERRXS2')/(XS2*XS2');
eXPP2=(ERRXPP2*ERRXPP2')/(XPP2*XPP2');
eXPHA2=(ERRXPHA2*ERRXPHA2')/(XPHA2*XPHA2');

% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 3
SF3=z(17,:);SA3=z(18,:);SNH4_3=z(19,:);SNO3_3=z(20,:);
```

```
set(figure(8),'name','Tank 1 Process variables and estimation
error','numbertitle','off')

figure(9)
subplot(4,4,1);plot(k,SF2);hold on; plot(k,SF2E,'k-.');ylabel('SF2,SF2E');
legend('SF2','SF2E');subplot(4,4,2); plot(k,ERRSF2,'k');ylabel('ERRSF2');

subplot(4,4,3);plot(k,SA2);hold on; plot(k,SA2E,'k-.');ylabel('SA2,SA2E');
legend('SA2','SA2E');subplot(4,4,4); plot(k,ERRSA2,'k');ylabel('ERRSA2');

subplot(4,4,5);plot(k,SNH4_2);hold on; plot(k,SNH4_2E,'k-.')
legend('SNH4_2','SNH4_2E');ylabel('SNH4_2,SNH4_2E');
subplot(4,4,6); plot(k,ERRSNH4_2,'k');ylabel('ERRSNH4_2');

subplot(4,4,7);plot(k,SNO3_2);hold on; plot(k,SNO3_2E,'k-.')
legend('SNO3_2','SNO3_2E');ylabel('SNO3_2,SNO3_2E');
subplot(4,4,8); plot(k,ERRSNO3_2,'k');ylabel('ERRSNO3_2');

subplot(4,4,9);plot(k,SPO4_2);hold on; plot(k,SPO4_2E,'k-.')
legend('SPO4_2','SPO4_2E');ylabel('SPO4_2,SPO4_2E');
subplot(4,4,10); plot(k,ERRSPO4_2,'k');ylabel('ERRSPO4_2');

subplot(4,4,11);plot(k,XS2);hold on; plot(k,XS2E,'k-.');ylabel('XS2,XS2E');
legend('XS2','XS2E');subplot(4,4,12); plot(k,ERRXS2,'k');ylabel('ERRXS2');

subplot(4,4,13);plot(k,XPP2);hold on; plot(k,XPP2E,'k-.');
legend('XPP2','XPP2E');ylabel('XPP2,XPP2E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP2,'k');
ylabel('ERRXPP2');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA2);hold on; plot(k,XPHA2E,'k-.')
legend('XPHA2','XPHA2E');ylabel('XPHA2,XPHA2E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA2,'k');
ylabel('ERRXPHA2');xlabel('discrete time k')
set(figure(9),'name','Tank 2 Process variables and estimation
error','numbertitle','off')

figure(10)
subplot(4,4,1);plot(k,SF3);hold on; plot(k,SF3E,'k-.');ylabel('SF3,SF3E');
legend('SF3','SF3E');subplot(4,4,2); plot(k,ERRSF3,'k');ylabel('ERRSF3');

subplot(4,4,3);plot(k,SA3);hold on; plot(k,SA3E,'k-.');ylabel('SA3,SA3E');
legend('SA3','SA3E');subplot(4,4,4); plot(k,ERRSA3,'k');ylabel('ERRSA3');

subplot(4,4,5);plot(k,SNH4_3);hold on; plot(k,SNH4_3E,'k-.')
legend('SNH4_3','SNH4_3E');ylabel('SNH4_3,SNH4_3E');
subplot(4,4,6); plot(k,ERRSNH4_3,'k');ylabel('ERRSNH4_3');

subplot(4,4,7);plot(k,SNO3_3);hold on; plot(k,SNO3_3E,'k-.')
legend('SNO3_3','SNO3_3E');ylabel('SNO3_3,SNO3_3E');
subplot(4,4,8); plot(k,ERRSNO3_3,'k');ylabel('ERRSNO3_3');

subplot(4,4,9);plot(k,SPO4_3);hold on; plot(k,SPO4_3E,'k-.')
legend('SPO4_3','SPO4_3E');ylabel('SPO4_3,SPO4_3E');
subplot(4,4,10); plot(k,ERRSPO4_3,'k');ylabel('ERRSPO4_3');

subplot(4,4,11);plot(k,XS3);hold on; plot(k,XS3E,'k-.')
legend('XS3','XS3E');ylabel('XS3,XS3E');
subplot(4,4,12); plot(k,ERRXS3,'k');ylabel('ERRXS3');

subplot(4,4,13);plot(k,XPP3);hold on; plot(k,XPP3E,'k-.')
legend('XPP3','XPP3E');ylabel('XPP3,XPP3E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP3,'k');
ylabel('ERRXPP3');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA3);hold on; plot(k,XPHA3E,'k-.')
```

```
legend('XPHA3','XPHA3E');ylabel('XPHA3,XPHA3E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA3,'k');
ylabel('ERRXPHA3');xlabel('discrete time k')
set(figure(10),'name','Tank 3 Process variables and estimation
error','numbertitle','off')
```

F.2: MATLAB script file – P_AASM2F.m

```
% M-File: P_AASM2F.m
% ========================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Calculation of the process rates for the full model

function P = P_AASM2F(xk)

global fSI YH fXI YPO4 YPHA iPBM YA iNBM Kh etaNO3 etafe muH qfe bH qPHA qPP muPAO bPAO
bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4 KALK KA Kfe KPP KPS KP KPHA KIPP KMAX

% TANK 1
P(1)= Kh * (xk(1)/(KO2+xk(1))) * ((xk(11)/xk(12))/(KX+(xk(11)/xk(12)))) * xk(12);
P(2)= Kh * etaNO3 * (KO2/(KO2+xk(1))) * (xk(5)/(KNO3+xk(5))) *
((xk(11)/xk(12))/(KX+(xk(11)/xk(12)))) * xk(12);
P(3)= Kh * etafe * (KO2/(KO2+xk(1))) * (KNO3/(KNO3+xk(5))) *
((xk(11)/xk(12))/(KX+(xk(11)/xk(12)))) * xk(12);
P(4)= muH * (xk(1)/(KO2+xk(1))) * (xk(2)/(KF+xk(2))) * (xk(2)/(xk(2)+xk(3))) *
(xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) * (xk(8)/(KALK+xk(8))) * xk(12);
P(5)= muH * (xk(1)/(KO2+xk(1))) * (xk(3)/(KA+xk(3))) * (xk(3)/(xk(2)+xk(3))) *
(xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) * (xk(8)/(KALK+xk(8))) * xk(12);
P(6)= muH * etaNO3 * (KO2/(KO2+xk(1))) * (xk(2)/(KF+xk(2))) * (xk(2)/(xk(2)+xk(3))) *
(xk(4)/(KNH4+xk(4))) * (xk(5)/(KNO3+xk(5))) * (xk(8)/(KALK+xk(8))) * xk(12);
P(7)= muH * etaNO3 * (KO2/(KO2+xk(1))) * (xk(3)/(KA+xk(3))) * (xk(3)/(xk(2)+xk(3))) *
(xk(4)/(KNH4+xk(4))) * (xk(5)/(KNO3+xk(5))) * (xk(8)/(KALK+xk(8)))* (xk(6)/(KP+xk(6)))
* xk(12);
P(8)= qfe * (KO2/(KO2+xk(1))) * (KNO3/(KNO3+xk(5))) * (xk(2)/(Kfe+xk(2))) *
(xk(8)/(KALK+xk(8)))* xk(12);
P(9)= bH * xk(12);
P(10)= qPHA * (xk(3)/(KA+xk(3))) * (xk(8)/(KALK+xk(8))) *
((xk(14)/xk(13))/(KPP+(xk(14)/xk(13)))) * xk(13);
P(11)= qPP * (xk(1)/(KO2+xk(1))) * (xk(6)/(KPS+xk(6))) * (xk(8)/(KALK+xk(8))) *
((xk(15)/xk(13))/(KPHA+(xk(15)/xk(13)))) * ((KMAX-(xk(14)/xk(13)))/(KPP+KMAX-
(xk(14)/xk(13)))) * xk(13);
P(12)= muPAO * (xk(1)/(KO2+xk(1))) * (xk(4)/(KNH4+xk(4))) * (xk(8)/(KALK+xk(8)))*
(xk(6)/(KP+xk(6))) * ((xk(15)/xk(13))/(KPHA+(xk(15)/xk(13)))) * xk(13);
P(13)= bPAO * xk(13) * (xk(8)/(KALK+xk(8)));
P(14)= bPP * xk(14) * (xk(8)/(KALK+xk(8)));
P(15)= bPHA * xk(15) * (xk(8)/(KALK+xk(8)));
P(16)= muAUT * (xk(1)/(KO2+xk(1))) * (xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) *
(xk(8)/(KALK+xk(8))) * xk(16);
P(17)= bAUT * xk(16);
P(18)= kPRE * xk(6) * xk(18);
P(19)= kRED * xk(19) * (xk(8)/(KALK+xk(8)));

% TANK 2
P(20)= Kh * (xk(20)/(KO2+xk(20))) * ((xk(30)/xk(31))/(KX+(xk(30)/xk(31)))) * xk(31);
P(21)= Kh * etaNO3 * (KO2/(KO2+xk(20))) * (xk(24)/(KNO3+xk(24))) *
((xk(30)/xk(31))/(KX+(xk(30)/xk(31)))) * xk(31);
P(22)= Kh * etafe * (KO2/(KO2+xk(20))) * (KNO3/(KNO3+xk(24))) *
((xk(30)/xk(31))/(KX+(xk(30)/xk(31)))) * xk(31);
P(23)= muH * (xk(20)/(KO2+xk(20))) * (xk(21)/(KF+xk(21))) * (xk(21)/(xk(21)+xk(22))) *
(xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) * (xk(27)/(KALK+xk(27))) * xk(31);
P(24)= muH * (xk(20)/(KO2+xk(20))) * (xk(22)/(KA+xk(22))) * (xk(22)/(xk(21)+xk(22))) *
(xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) * (xk(27)/(KALK+xk(27))) * xk(31);
P(25)= muH * etaNO3 * (KO2/(KO2+xk(20))) * (xk(21)/(KF+xk(21))) *
(xk(21)/(xk(21)+xk(22))) * (xk(23)/(KNH4+xk(23))) * (xk(24)/(KNO3+xk(24))) *
(xk(27)/(KALK+xk(27))) * xk(31);
```

287

```
P(26)= muH * etaNO3 * (KO2/(KO2+xk(20))) * (xk(22)/(KA+xk(22))) *
(xk(22)/(xk(21)+xk(22))) * (xk(23)/(KNH4+xk(23))) * (xk(24)/(KNO3+xk(24))) *
(xk(27)/(KALK+xk(27)))* (xk(25)/(KP+xk(25))) * xk(31);
P(27)= qfe * (KO2/(KO2+xk(20))) * (KNO3/(KNO3+xk(24))) * (xk(21)/(Kfe+xk(21))) *
(xk(27)/(KALK+xk(27)))* xk(31);
P(28)= bH * xk(31);
P(29)= qPHA * (xk(22)/(KA+xk(22))) * (xk(27)/(KALK+xk(27))) *
((xk(33)/xk(32))/(KPP+(xk(33)/xk(32)))) * xk(32);
P(30)= qPP * (xk(20)/(KO2+xk(20))) * (xk(25)/(KPS+xk(25))) * (xk(27)/(KALK+xk(27))) *
((xk(34)/xk(32))/(KPHA+(xk(34)/xk(32)))) * ((KMAX-(xk(33)/xk(32)))/(KPP+KMAX-
(xk(33)/xk(32)))) * xk(32);
P(31)= muPAO * (xk(20)/(KO2+xk(20))) * (xk(23)/(KNH4+xk(23))) * (xk(27)/(KALK+xk(27)))*
(xk(25)/(KP+xk(25))) * ((xk(34)/xk(32))/(KPHA+(xk(34)/xk(32)))) * xk(32);
P(32)= bPAO * xk(32) * (xk(27)/(KALK+xk(27)));
P(33)= bPP * xk(33) * (xk(27)/(KALK+xk(27)));
P(34)= bPHA * xk(34) * (xk(27)/(KALK+xk(27)));
P(35)= muAUT * (xk(20)/(KO2+xk(20))) * (xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) *
(xk(27)/(KALK+xk(27))) * xk(35);
P(36)= bAUT * xk(35);
P(37)= kPRE * xk(25) * xk(37);
P(38)= kRED * xk(38) * (xk(27)/(KALK+xk(27)));


% TANK 3
P(39)= Kh * (xk(39)/(KO2+xk(39))) * ((xk(49)/xk(50))/(KX+(xk(49)/xk(50)))) * xk(50);
P(40)= Kh * etaNO3 * (KO2/(KO2+xk(39))) * (xk(43)/(KNO3+xk(43))) *
((xk(49)/xk(50))/(KX+(xk(49)/xk(50)))) * xk(50);
P(41)= Kh * etafe * (KO2/(KO2+xk(39))) * (KNO3/(KNO3+xk(43))) *
((xk(49)/xk(50))/(KX+(xk(49)/xk(50)))) * xk(50);
P(42)= muH * (xk(39)/(KO2+xk(39))) * (xk(40)/(KF+xk(40))) * (xk(40)/(xk(40)+xk(41))) *
(xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) * (xk(46)/(KALK+xk(46))) * xk(50);
P(43)= muH * (xk(39)/(KO2+xk(39))) * (xk(41)/(KA+xk(41))) * (xk(41)/(xk(40)+xk(41))) *
(xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) * (xk(46)/(KALK+xk(46))) * xk(50);
P(44)= muH * etaNO3 * (KO2/(KO2+xk(39))) * (xk(40)/(KF+xk(40))) *
(xk(40)/(xk(40)+xk(41))) * (xk(42)/(KNH4+xk(42))) * (xk(43)/(KNO3+xk(43))) *
(xk(46)/(KALK+xk(46))) * xk(50);
P(45)= muH * etaNO3 * (KO2/(KO2+xk(39))) * (xk(41)/(KA+xk(41))) *
(xk(41)/(xk(40)+xk(41))) * (xk(42)/(KNH4+xk(42))) * (xk(43)/(KNO3+xk(43))) *
(xk(46)/(KALK+xk(46)))* (xk(44)/(KP+xk(44))) * xk(50);
P(46)= qfe * (KO2/(KO2+xk(39))) * (KNO3/(KNO3+xk(43))) * (xk(40)/(Kfe+xk(40))) *
(xk(46)/(KALK+xk(46)))* xk(50);
P(47)= bH * xk(50);
P(48)= qPHA * (xk(41)/(KA+xk(41))) * (xk(46)/(KALK+xk(46))) *
((xk(52)/xk(51))/(KPP+(xk(52)/xk(51)))) * xk(51);
P(49)= qPP * (xk(39)/(KO2+xk(39))) * (xk(44)/(KPS+xk(44))) * (xk(46)/(KALK+xk(46))) *
((xk(53)/xk(51))/(KPHA+(xk(53)/xk(51)))) * ((KMAX-(xk(52)/xk(51)))/(KPP+KMAX-
(xk(52)/xk(51)))) * xk(51);
P(50)= muPAO * (xk(39)/(KO2+xk(39))) * (xk(42)/(KNH4+xk(42))) * (xk(46)/(KALK+xk(46)))*
(xk(44)/(KP+xk(44))) * ((xk(53)/xk(51))/(KPHA+(xk(53)/xk(51)))) * xk(51);
P(51)= bPAO * xk(51) * (xk(46)/(KALK+xk(46)));
P(52)= bPP * xk(52) * (xk(46)/(KALK+xk(46)));
P(53)= bPHA * xk(53) * (xk(46)/(KALK+xk(46)));
P(54)= muAUT * (xk(39)/(KO2+xk(39))) * (xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) *
(xk(46)/(KALK+xk(46))) * xk(54);
P(55)= bAUT * xk(54);
P(56)= kPRE * xk(44) * xk(56);
P(57)= kRED * xk(57) * (xk(46)/(KALK+xk(46)));


P = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20) P(21) P(22) P(23) P(24) P(25) P(26) P(27) P(28) P(29)
P(30) P(31) P(32) P(33) P(34) P(35) P(36) P(37) P(38) P(39) P(40) P(41) P(42) P(43)
P(44) P(45) P(46) P(47) P(48) P(49) P(50) P(51) P(52) P(53) P(54) P(55) P(56) P(57)]';
```

F.3: MATLAB script file – P_AASM2R.m

```
% M-File: P_AASM2R.m
% ========================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model

function    Pk = P_AASM2R(zk,uk,XHk,XAUTk,XPAOk,SALKk)

global fSI YH fXI YPO4 YPHA iPBM YA iNBM Kh etaNO3 etafe muH qfe bH qPHA qPP muPAO bPAO
bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4 KALK KA Kfe KPP KPS KP KPHA KIPP KMAX
% TANK 1
P(1)= (uk(1)/(KO2+uk(1))) * ((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(2)= etaNO3 * (KO2/(KO2+uk(1))) * (zk(4)/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(3)= etafe * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(4)= (uk(1)/(KO2+uk(1))) * (zk(1)/(KF+zk(1))) * (zk(1)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(5)= (uk(1)/(KO2+uk(1))) * (zk(2)/(KA+zk(2))) * (zk(2)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(6)= etaNO3 * (KO2/(KO2+uk(1))) * (zk(1)/(KF+zk(1))) * (zk(1)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(4)/(KNO3+zk(4))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(7)= etaNO3 * (KO2/(KO2+uk(1))) * (zk(2)/(KA+zk(2))) * (zk(2)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(4)/(KNO3+zk(4))) * (SALKk(1)/(KALK+SALKk(1)))*
(zk(5)/(KP+zk(5))) * XHk(1);
P(8)= (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(1)/(Kfe+zk(1))) *
(SALKk(1)/(KALK+SALKk(1)))* XHk(1);
P(9)= (zk(2)/(KA+zk(2))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(7)/XPAOk(1))/(KPP+(zk(7)/XPAOk(1)))) * XPAOk(1);
P(10)= (uk(1)/(KO2+uk(1))) * (zk(5)/(KPS+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * ((KMAX-(zk(7)/XPAOk(1)))/(KPP+KMAX-
(zk(7)/XPAOk(1)))) * XPAOk(1);
P(11)= (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) * (SALKk(1)/(KALK+SALKk(1)))*
(zk(5)/(KP+zk(5))) * ((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * XPAOk(1);
P(12)= (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XAUTk(1);


P1k = [[P(1) P(2) P(3) 0    0    0    0    0    0    0     0     0    ];
       [0    0    0    P(4) P(5) P(6) P(7) 0    0    0     0     0    ];
       [0    0    0    0    0    0    0    P(8) 0    0     0     0    ];
       [0    0    0    0    0    0    0    0    P(9) 0     0     0    ];
       [0    0    0    0    0    0    0    0    0    P(10) 0     0    ];
       [0    0    0    0    0    0    0    0    0    0     P(11) 0    ];
       [0    0    0    0    0    0    0    0    0    0     0     P(12)]];

% TANK 2
P(13)= (uk(2)/(KO2+uk(2))) * ((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(14)= etaNO3 * (KO2/(KO2+uk(2))) * (zk(12)/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(15)= etafe * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(16)= (uk(2)/(KO2+uk(2))) * (zk(9)/(KF+zk(9))) * (zk(9)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(17)= (uk(2)/(KO2+uk(2))) * (zk(10)/(KA+zk(10))) * (zk(10)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(18)= etaNO3 * (KO2/(KO2+uk(2))) * (zk(9)/(KF+zk(9))) * (zk(9)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(12)/(KNO3+zk(12))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(19)= etaNO3 * (KO2/(KO2+uk(2))) * (zk(10)/(KA+zk(10))) * (zk(10)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(12)/(KNO3+zk(12))) * (SALKk(2)/(KALK+SALKk(2)))*
(zk(13)/(KP+zk(13))) * XHk(2);
P(20)= (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(9)/(Kfe+zk(9))) *
(SALKk(2)/(KALK+SALKk(2)))* XHk(2);
```

```
P(21)= (zk(10)/(KA+zk(10))) * (SALKk(2)/(KALK+SALKk(2))) *
((XPAOk(2)/XPAOk(2))/(KPP+(XPAOk(2)/XPAOk(2)))) * XPAOk(2);
P(22)= (uk(2)/(KO2+uk(2))) * (zk(13)/(KPS+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) *
((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * ((KMAX-(XPAOk(2)/XPAOk(2)))/(KPP+KMAX-
(XPAOk(2)/XPAOk(2)))) * XPAOk(2);
P(23)= (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (SALKk(2)/(KALK+SALKk(2)))*
(zk(13)/(KP+zk(13))) * ((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * XPAOk(2);
P(24)= (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XAUTk(2);


P2k = [[P(13) P(14) P(15)   0      0      0      0      0      0      0      0   0];
       [0      0      0    P(16)  P(17)  P(18)  P(19)   0      0      0      0   0];
       [0      0      0      0      0      0      0    P(20)   0      0      0   0];
       [0      0      0      0      0      0      0      0    P(21)   0      0   0];
       [0      0      0      0      0      0      0      0      0    P(22)  0   0];
       [0      0      0      0      0      0      0      0      0   0     P(23) 0];
       [0      0      0      0      0      0      0      0      0   0     0  P(24)]];

% TANK 3
P(25)= (uk(3)/(KO2+uk(3))) * ((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(26)= etaNO3 * (KO2/(KO2+uk(3))) * (zk(20)/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(27)= etafe * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(28)= (uk(3)/(KO2+uk(3))) * (zk(17)/(KF+zk(17))) * (zk(17)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(29)= (uk(3)/(KO2+uk(3))) * (zk(18)/(KA+zk(18))) * (zk(18)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(30)= etaNO3 * (KO2/(KO2+uk(3))) * (zk(17)/(KF+zk(17))) * (zk(17)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(20)/(KNO3+zk(20))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(31)= etaNO3 * (KO2/(KO2+uk(3))) * (zk(18)/(KA+zk(18))) * (zk(18)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(20)/(KNO3+zk(20))) * (SALKk(3)/(KALK+SALKk(3)))*
(zk(21)/(KP+zk(21))) * XHk(3);
P(32)= (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(17)/(Kfe+zk(17))) *
(SALKk(3)/(KALK+SALKk(3)))* XHk(3);
P(33)= (zk(18)/(KA+zk(18))) * (SALKk(3)/(KALK+SALKk(3))) *
((XPAOk(3)/XPAOk(3))/(KPP+(XPAOk(3)/XPAOk(3)))) * XPAOk(3);
P(34)= (uk(3)/(KO2+uk(3))) * (zk(21)/(KPS+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) *
((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * ((KMAX-(XPAOk(3)/XPAOk(3)))/(KPP+KMAX-
(XPAOk(3)/XPAOk(3)))) * XPAOk(3);
P(35)= (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (SALKk(3)/(KALK+SALKk(3)))*
(zk(21)/(KP+zk(21))) * ((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * XPAOk(3);
P(36)= (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XAUTk(3);


P3k = [[P(25) P(26) P(27)   0      0      0      0      0      0      0      0   0];
       [0      0      0    P(28)  P(29)  P(30)  P(31)   0      0      0      0   0];
       [0      0      0      0      0      0      0    P(32)   0      0      0   0];
       [0      0      0      0      0      0      0      0    P(33)   0      0   0];
       [0      0      0      0      0      0      0      0      0    P(34)  0   0];
       [0      0      0      0      0      0      0      0      0   0     P(35) 0];
       [0      0      0      0      0      0      0      0      0   0     0  P(36)]];

%Matrix Pk for the whole process
Pk = [P1k'; P2k'; P3k'];
```

F.4: MATLAB script file – P_AASM2RE.m

```
% M-File: P_AASM2RE.m
% ============================= M-FILE DESCRIPTION ==========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model
```

290

```
% Description of the function function
function Pk =
P_AASM2RE(zk,uk,XHk,XAUTk,XPAOk,SALKk,kHek,muHek,qfeek,qPHAek,qPPek,muPAOek,muAUTek)

global fSI YH fXI YPO4 YPHA iPBM YA iNBM Kh etaNO3 etafe muH qfe bH qPHA qPP muPAO bPAO
bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4 KALK KA Kfe KPP KPS KP KPHA KIPP KMAX
% TANK 1
P(1)= kHek * (uk(1)/(KO2+uk(1))) * ((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(2)= kHek * etaNO3 * (KO2/(KO2+uk(1))) * (zk(4)/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(3)= kHek * etafe * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(4)= muHek * (uk(1)/(KO2+uk(1))) * (zk(1)/(KF+zk(1))) * (zk(1)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(5)= muHek * (uk(1)/(KO2+uk(1))) * (zk(2)/(KA+zk(2))) * (zk(2)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(6)= muHek * etaNO3 * (KO2/(KO2+uk(1))) * (zk(1)/(KF+zk(1))) * (zk(1)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(4)/(KNO3+zk(4))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(7)= muHek * etaNO3 * (KO2/(KO2+uk(1))) * (zk(2)/(KA+zk(2))) * (zk(2)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(4)/(KNO3+zk(4))) * (SALKk(1)/(KALK+SALKk(1)))*
(zk(5)/(KP+zk(5))) * XHk(1);
P(8)= qfeek * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(1)/(Kfe+zk(1))) *
(SALKk(1)/(KALK+SALKk(1)))* XHk(1);
P(9)= qPHAek * (zk(2)/(KA+zk(2))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(7)/XPAOk(1))/(KPP+(zk(7)/XPAOk(1)))) * XPAOk(1);
P(10)= qPPek * (uk(1)/(KO2+uk(1))) * (zk(5)/(KPS+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * ((KMAX-(zk(7)/XPAOk(1)))/(KPP+KMAX-
(zk(7)/XPAOk(1)))) * XPAOk(1);
P(11)= muPAOek * (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) *
(SALKk(1)/(KALK+SALKk(1)))* (zk(5)/(KP+zk(5))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * XPAOk(1);
P(12)= muAUTek * (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XAUTk(1);


% TANK 2
P(13)= kHek * (uk(2)/(KO2+uk(2))) * ((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(14)= kHek * etaNO3 * (KO2/(KO2+uk(2))) * (zk(12)/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(15)= kHek * etafe * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(16)= muHek * (uk(2)/(KO2+uk(2))) * (zk(9)/(KF+zk(9))) * (zk(9)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(17)= muHek * (uk(2)/(KO2+uk(2))) * (zk(10)/(KA+zk(10))) * (zk(10)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(18)= muHek * etaNO3 * (KO2/(KO2+uk(2))) * (zk(9)/(KF+zk(9))) * (zk(9)/(zk(9)+zk(10)))
* (zk(11)/(KNH4+zk(11))) * (zk(12)/(KNO3+zk(12))) * (SALKk(2)/(KALK+SALKk(2))) *
XHk(2);
P(19)= muHek * etaNO3 * (KO2/(KO2+uk(2))) * (zk(10)/(KA+zk(10))) *
(zk(10)/(zk(9)+zk(10))) * (zk(11)/(KNH4+zk(11))) * (zk(12)/(KNO3+zk(12))) *
(SALKk(2)/(KALK+SALKk(2)))* (zk(13)/(KP+zk(13))) * XHk(2);
P(20)= qfeek * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(9)/(Kfe+zk(9))) *
(SALKk(2)/(KALK+SALKk(2)))* XHk(2);
P(21)= qPHAek * (zk(10)/(KA+zk(10))) * (SALKk(2)/(KALK+SALKk(2))) *
((XPAOk(2)/XPAOk(2))/(KPP+(XPAOk(2)/XPAOk(2)))) * XPAOk(2);
P(22)= qPPek * (uk(2)/(KO2+uk(2))) * (zk(13)/(KPS+zk(13))) * (SALKk(2)/(KALK+SALKk(2)))
* ((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * ((KMAX-(XPAOk(2)/XPAOk(2)))/(KPP+KMAX-
(XPAOk(2)/XPAOk(2)))) * XPAOk(2);
P(23)= muPAOek * (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) *
(SALKk(2)/(KALK+SALKk(2)))* (zk(13)/(KP+zk(13))) *
((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * XPAOk(2);
P(24)= muAUTek * (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XAUTk(2);


% TANK 3
P(25)= kHek * (uk(3)/(KO2+uk(3))) * ((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
```

291

```
P(26)= kHek * etaNO3 * (KO2/(KO2+uk(3))) * (zk(20)/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(27)= kHek * etafe * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(28)= muHek * (uk(3)/(KO2+uk(3))) * (zk(17)/(KF+zk(17))) * (zk(17)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(29)= muHek * (uk(3)/(KO2+uk(3))) * (zk(18)/(KA+zk(18))) * (zk(18)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(30)= muHek * etaNO3 * (KO2/(KO2+uk(3))) * (zk(17)/(KF+zk(17))) *
(zk(17)/(zk(17)+zk(18))) * (zk(19)/(KNH4+zk(19))) * (zk(20)/(KNO3+zk(20))) *
(SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(31)= muHek * etaNO3 * (KO2/(KO2+uk(3))) * (zk(18)/(KA+zk(18))) *
(zk(18)/(zk(17)+zk(18))) * (zk(19)/(KNH4+zk(19))) * (zk(20)/(KNO3+zk(20))) *
(SALKk(3)/(KALK+SALKk(3)))* (zk(21)/(KP+zk(21))) * XHk(3);
P(32)= qfeek * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(17)/(Kfe+zk(17))) *
(SALKk(3)/(KALK+SALKk(3)))* XHk(3);
P(33)= qPHAek * (zk(18)/(KA+zk(18))) * (SALKk(3)/(KALK+SALKk(3))) *
((XPAOk(3)/XPAOk(3))/(KPP+(XPAOk(3)/XPAOk(3)))) * XPAOk(3);
P(34)= qPPek * (uk(3)/(KO2+uk(3))) * (zk(21)/(KPS+zk(21))) * (SALKk(3)/(KALK+SALKk(3)))
* ((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * ((KMAX-(XPAOk(3)/XPAOk(3)))/(KPP+KMAX-
(XPAOk(3)/XPAOk(3)))) * XPAOk(3);
P(35)= muPAOek * (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) *
(SALKk(3)/(KALK+SALKk(3)))* (zk(21)/(KP+zk(21))) *
((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * XPAOk(3);
P(36)= muAUTek * (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XAUTk(3);

Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20) P(21) P(22) P(23) P(24) P(25) P(26) P(27) P(28) P(29)
P(30) P(31) P(32) P(33) P(34) P(35) P(36)]';
```

APPENDIX G: DEVELOPED MATLAB CODE FOR OPTIMAL CONTROL OF THE ATHLONE PLANT WITH THE ACTIVATED SLUDGE MODEL NO.2d BIO-MODEL

G.1: MATLAB script file – AASM2d_Simulation.m

```matlab
% M-File: AASM2d_Simulation.m
% ============================ M-FILE DESCRIPTION ========================
% This m-file is used for:
% Simulation of the Athlone plant using full ASM2d model
% Calculation of the parameters for the reduced ASM2d model
% Simulation of the Athlone plant using the reduced ASM2d model

clear all          % Clear the workspace
clc                % Clear the command window


% ========================== PARAMETER DEFINITION =======================
% Declaration of the global variables
global fSI YH fXI YPO4 YPHA iPBM YA iNBM Kh etaNO3 etafe muH qfe bH qPHA
global qPP muPAO bPAO bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4
global KALK KA Kfe KPP KPS KP KPHA KIPP KMAX


K1  = 11;          % Prediction horizon or interval
DT = 0.01042;      % Sampling time

% Initialization of the model parameters, typical values at 20 0C
% Stochiometric parameters
fSI     = 0.0;
YH      = 0.625;
fXI     = 0.10;
YPO4    = 0.40;
YPHA    = 0.20;
iPBM    = 0.02;
YA      = 0.24;
iNBM    = 0.07;
% Kinetic parameters
Kh      = 3 ;
etaNO3  = 0.8;
etafe   = 0.40;
muH     = 0.6;
qfe     = 3;
bH      = 0.4;
qPHA    = 3;
qPP     = 1.5;
muPAO   = 1;
bPAO    = 0.2;
bPP     = 0.2;
bPHA    = 0.2;
muAUT   = 1;
bAUT    = 0.15;
kPRE    = 1;
kRED    = 0.6;
% Monod switching functions
KO2     = 0.2;
KX      = 0.1;
KNO3    = 0.5;
KF      = 4;
KNH4    = 0.05;
KALK    = 0.1;
KA      = 4;
Kfe     = 20;
KPP     = 0.01;
KPS     = 0.20;
KP      = 0.01;
KPHA    = 0.01;
```

```
KIPP    = 0.02;
KMAX    = 0.34;


% Initialization of the Athlone process data
V1   = 1148;          % Tank 1 volume
V2   = 1148;          % Tank 2 volume
V3   = 5273;          % Tank 3 volume


Q0   = 40003;         % Influent flow rate
Qa   = 39916.8;       % Recycled flow rate from tank 3
Qp   = 39916.8;       % Recycled flow rate from tank 2
Qr   = 40003;         % Recycled flow rate from the settler
Qw   = 38400.0;       % Waste flow rate (3840)
Q1   = Qp + Q0;       % flow through the tank 1
Q2   = Qp + Q0 + Qa + Qr;    % flow through the tank 2
Q3   = Q2;                    % flow through the tank 3


% Coeficient for fraction of flow rates for the particulate materials
LAMBDA = (Qp+Q0+Qr-Qw)/(Qr);


% Oxygen parameters
SOsat  = 8;           % Oxygen saturation point
KLa    = 4.8;         % Oxygen transfer function in Tank 1


% ========================= PROCESS MODEL ===========================
% Calculation of the MATRIX A
% Matrix A represent weighting factors of the accumulated state variables
% from different flows towards state variables in each tank as derived
% through mass balance principles, see chapter 3 of the Thesis
IV1 = DT/ V1; IV11 = IV1 * Q1; A11 = 1 - IV11;
IV2 = DT/ V2; IV22 = IV2 * Q2; A22 = 1 - IV22;
IV3 = DT/ V3; IV33 = IV3 * Q3; A33 = 1 - IV33;

A12 = IV1 * Qp;
A23a = IV2 * (Qa + Qr);
A23b = IV2 * (Qa + LAMBDA*Qr);
A21 = IV2*Q1;
A32 = IV3*Q2;

A11 = A11 * ones(1,19); A11 = diag(A11);
A12 = A12 * ones(1,19); A12 = diag(A12);
A13 = zeros(19,19);

A21 = A21 * ones(1,19); A21 = diag(A21);
A22 = A22 * ones(1,19); A22 = diag(A22);
A23 =
[A23a,A23a,A23a,A23a,A23a,A23a,A23a,A23a,A23a,A23b,A23b,A23b,A23b,A23b,A23b,A23b,A
23b,A23b];
A23 = diag(A23);


A31 = zeros(19,19);
A32 = A32 * ones(1,19); A32 = diag(A32);
A33 = A33 * ones(1,19); A33 = diag(A33);


A = [A11 A12 A13; A21 A22 A23; A31 A32 A33];

% Calculation of the MATRIX B
% Matrix B represent weighting factors of the state variables towards the
% influent wastewater composition as derived through mass balance
% principles, see chapter 3 of the Thesis
IV0 = DT/ V1*Q0;
B11 = IV0 * ones(1,19); B11 = diag(B11);

B = [B11; zeros(38,19)];

% Formulation of the MATRIX C
```

294

```
% Matrix C represent weighting factors of the state variables towards the
% processes on the ASM1 matrix table
CC=[[0 1 0 0 0.01 0 0 0 0.001 0 -1 0 0 0 0 0 -0.75 0 0];
 [0 1 0 0 0.01 0 0 0 0.001 0 -1 0 0 0 0 0 -0.75 0 0];
 [0 1 0 0 0.01 0 0 0 0.001 0 -1 0 0 0 0 0 -0.75 0 0];
 [-0.6 -1.6 0 0 -0.022 0 0 -0.004 -0.001 0 0 1 0 0 0 0 0.9 0 0];
 [-0.6 0 -1.6 0 -0.07 0 0 -0.02 0.021 0 0 1 0 0 0 0 0.9 0 0];
 [0 -1.6 0 0 -0.022 0.21 -0.21 -0.004 0.014 0 0 1 0 0 0 0 0.9 0 0];
 [0 0 -1.6 0 -0.07 0.21 -0.21 -0.02 0.036 0 0 1 0 0 0 0 0.9 0 0];
 [0 -1 1 0 0.03 0 0 0.01 -0.014 0 0 0 0 0 0 0 0 0 0];
 [0 0 0 0 0.031 0 0 0.01 0.002 0.1 0.9 -1 0 0 0 0 -0.15 0 0];
 [0 0 -1 0 0 0 0 0.4 0.009 0 0 0 0 -0.4 1 0 -0.69 0 0];
 [-0.2 0 0 0 0 0 0 -1 0.016 0 0 0 0 1 -0.2 0 3.11 0 0];
 [0 0 0 0 0 0.07 -0.07 -1 0.021 0 0 0 0 1 -0.2 0 3.11 0 0];
 [-0.6 0 0 0 -0.07 0 0 -0.02 -0.004 0 0 0 1 0 -1.6 0 -0.06 0 0];
 [0 0 0 0 -0.07 0.21 -0.21 -0.02 0.011 0 0 0 1 0 -1.6 0 -0.06 0 0];
 [0 0 0 0 0.031 0 0 0.01 0.002 0.1 0.9 0 -1 0 0 0 -0.15 0 0];
 [0 0 0 0 0 0 0 1 -0.016 0 0 0 0 -1 0 0 -3.23 0 0]
 [0 0 1 0 0 0 0 0 -0.016 0 0 0 0 0 -1 0 -0.6 0 0];
 [-18 0 0 0 -4.24 0 4.17 -0.02 -0.6 0 0 0 0 0 0 1 0.9 0 0];
 [0 0 0 0 0.031 0 0 0.01 0.002 0.1 0.9 0 0 0 0 -1 -0.15 0 0];
 [0 0 0 0 0 0 0 -1 0.048 0 0 0 0 0 0 0 1.42 -3.45 4.87];
 [0 0 0 0 0 0 0 1 -0.048 0 0 0 0 0 0 0 -1.42 3.45 -4.87];]

C = DT*[CC zeros(21,38); zeros(21,19) CC zeros(21,19); zeros(21,38) CC];

% ================================= SIMULATION DATA =============================
% Average values of the inflow concentrations based on the dry weather file
SO2i  = 0.0;      SFi   = 30;     SAi   = 20;   SNH4i = 31.56; SNO3i = 0.0;
SPO4i = 3.6;      SIi   = 30;     SALKi = 7;    SN2i  = 6.95;  XIi   = 51.2;
XSi   = 202.32;   XHi   = 28.17; XPAOi = 0.0;  XPPi  = 0.0;    XPHAi = 0.0;
XAUTi = 0.0;      XTSSi = 180;    XMeOHi = 0.0; XMePi = 0.0;

xi = [SO2i*ones(1,K1); SFi*ones(1,K1); SAi*ones(1,K1); SNH4i*ones(1,K1);
      SNO3i*ones(1,K1); SPO4i*ones(1,K1); SIi*ones(1,K1); SALKi*ones(1,K1);
      SN2i*ones(1,K1); XIi*ones(1,K1); XSi*ones(1,K1); XHi*ones(1,K1);
      XPAOi*ones(1,K1); XPPi*ones(1,K1); XPHAi*ones(1,K1); XAUTi*ones(1,K1);
      XTSSi*ones(1,K1); XMeOHi*ones(1,K1); XMePi*ones(1,K1)];

% Values of the initial conditions
% For tank 1
SO2_1  = 4.2984433e-03; SF_1   = 2.8082131e+00; SA_1   = 2.8082131e+00;
SNH4_1 = 7.9178845e+00; SNO3_1 = 5.3699400e+00; SPO4_1 = 3.6;
SI_1   = 3.0000000e+01; SALK_1 = 4.9277103e+00; SN2_1  = 1.2166405e+00;
XI_1   = 1.1491252e+03; XS_1   = 8.2134908e+01; XH_1   = 2.5517658e+03;
XPAO_1 = 0.1; XPP_1   = 0.1; XPHA_1 = 0.1; XAUT_1   = 1.4838943e+02;
XTSS_1 = 180;XMeOH_1 = 0.1;XMeP_1 = 0.1;
x10 = [SO2_1; SF_1; SA_1; SNH4_1; SNO3_1; SPO4_1; SI_1;   SALK_1; SN2_1;
       XI_1;  XS_1; XH_1; XPAO_1; XPP_1;  XPHA_1; XAUT_1; XTSS_1; XMeOH_1;
       XMeP_1];

% For tank 2
SO2_2  = 6.3131911e-05; SF_2   = 1.4587940e+00; SA_2   = 1.4587940e+00;
SNH4_2 = 8.3444148e+00; SNO3_2 = 3.6619672e+00; SPO4_2 = 3.6;
SI_2   = 3.0000000e+01; SALK_2 = 5.0801748e+00; SN2_2  = 8.8206477e-01;
XI_2   = 1.1491252e+03; XS_2   = 7.6386187e+01; XH_2   = 2.5533851e+03;
XPAO_2 = 0.1;XPP_2    = 0.1; XPHA_2 = 0.1; XAUT_2 = 1.4830914e+02;
XTSS_2 = 180;XMeOH_2 = 0.1; XMeP_2 = 0.1;
x20 = [SO2_2; SF_2; SA_2; SNH4_2; SNO3_2; SPO4_2; SI_2;   SALK_2; SN2_2;
       XI_2;  XS_2; XH_2; XPAO_2; XPP_2;  XPHA_2; XAUT_2; XTSS_2; XMeOH_2;
       XMeP_2];

% For tank 3
SO2_3  = 1.7183778e+00; SF_3   = 1.1495418e+00; SA_3   = 1.1495418e+00;
SNH4_3 = 5.5479452e+00; SNO3_3 = 6.5408820e+00; SPO4_3 = 3.6;
SI_3   = 3.0000000e+01; SALK_3 = 4.6747902e+00; SN2_3  = 8.2888682e-01;
```

```
XI_3    = 1.1491252e+03; XS_3   = 6.4854922e+01; XH_3    = 2.5571314e+03;
XPAO_3 = 0.1; XPP_3   = 0.1; XPHA_3 = 0.1; XAUT_3   = 1.4894126e+02;
XTSS_3 = 180; XMeOH_3 = 0.1; XMeP_3  = 0.1;
x30 = [SO2_3; SF_3; SA_3; SNH4_3; SNO3_3; SPO4_3; SI_3;   SALK_3; SN2_3;
       XI_3;  XS_3; XH_3; XPAO_3; XPP_3;  XPHA_3; XAUT_3; XTSS_3; XMeOH_3;
       XMeP_3];

x0 = [x10;x20;x30];
x = [x0 zeros(57,K1)];

% Steady state simulation of the model
for k = 1:K1
    xk = x(:,k);
    xik = xi(:,k);
    Pk = P_AASM2dF(xk);

    SOk  = [xk(39,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xx1k = [zeros(38,1);xxk(1);zeros(18,1)];
    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xx1k;
end

% Full model response to the steady state influent concentrations
% Tank 1
figure(1)
subplot(5,4,1);plot(x(1,:));title('SO2_1')
subplot(5,4,2);plot(x(2,:));title('SF1')
subplot(5,4,3);plot(x(3,:));title('SA1')
subplot(5,4,4);plot(x(4,:));title('SNH4_1')
subplot(5,4,5);plot(x(5,:));title('SNO3_1')
subplot(5,4,6);plot(x(6,:));title('SPO4_1')
subplot(5,4,7);plot(x(7,:));title('SI1')
subplot(5,4,8);plot(x(8,:));title('SALK1')
subplot(5,4,9);plot(x(9,:));title('SN2_1')
subplot(5,4,10);plot(x(10,:));title('XI1');
subplot(5,4,11);plot(x(11,:));title('XS1')
subplot(5,4,12);plot(x(12,:));title('XH1')
subplot(5,4,13);plot(x(13,:));title('XPAO1')
subplot(5,4,14);plot(x(14,:));title('XPP1')
subplot(5,4,15);plot(x(15,:));title('XPHA1')
subplot(5,4,16);plot(x(16,:));title('XAUT1')
xlabel('discrete time k')
subplot(5,4,17);plot(x(17,:));title('XTSS1')
xlabel('discrete time k')
subplot(5,4,18);plot(x(18,:));title('XMeOH1')
xlabel('discrete time k')
subplot(5,4,19);plot(x(19,:));title('XMeP1')
xlabel('discrete time k')
set(figure(1),'name','Tank 1 steady state results','numbertitle','off')

% Tank 2
figure(2)
subplot(5,4,1);plot(x(20,:));title('SO2_2')
subplot(5,4,2);plot(x(21,:));title('SF2')
subplot(5,4,3);plot(x(22,:));title('SA2')
subplot(5,4,4);plot(x(23,:));title('SNh4_2')
subplot(5,4,5);plot(x(24,:));title('SNO3_2')
subplot(5,4,6);plot(x(25,:));title('SPO4_2')
subplot(5,4,7);plot(x(26,:));title('SI2')
subplot(5,4,8);plot(x(27,:));title('SALK2')
subplot(5,4,9);plot(x(28,:));title('SN2_2')
subplot(5,4,10);plot(x(29,:));title('XI2');
subplot(5,4,11);plot(x(30,:));title('XS2')
subplot(5,4,12);plot(x(31,:));title('XH2')
subplot(5,4,13);plot(x(32,:));title('XPAO2')
subplot(5,4,14);plot(x(33,:));title('XPP2')
```

296

```
subplot(5,4,15);plot(x(34,:));title('XPHA2')
subplot(5,4,16);plot(x(35,:));title('XAUT2')
xlabel('discrete time k')
subplot(5,4,17);plot(x(36,:));title('XTSS2')
xlabel('discrete time k')
subplot(5,4,18);plot(x(37,:));title('XMeOH2')
xlabel('discrete time k')
subplot(5,4,19);plot(x(38,:));title('XMeP2')
xlabel('discrete time k')
set(figure(2),'name','Tank 2 steady state results','numbertitle','off')

% Tank 3
figure(3)
subplot(5,4,1);plot(x(39,:));title('SO2_3')
subplot(5,4,2);plot(x(40,:));title('SF3')
subplot(5,4,3);plot(x(41,:));title('SA3')
subplot(5,4,4);plot(x(42,:));title('SNH4_3')
subplot(5,4,5);plot(x(43,:));title('SNO3_3')
subplot(5,4,6);plot(x(44,:));title('SPO4_3')
subplot(5,4,7);plot(x(45,:));title('SI3')
subplot(5,4,8);plot(x(46,:));title('SALK3')
subplot(5,4,9);plot(x(47,:));title('SN2_3')
subplot(5,4,10);plot(x(48,:));title('XI3');
subplot(5,4,11);plot(x(49,:));title('XS3')
subplot(5,4,12);plot(x(50,:));title('XH3')
subplot(5,4,13);plot(x(51,:));title('XPAO3')
subplot(5,4,14);plot(x(52,:));title('XPP3')
subplot(5,4,15);plot(x(53,:));title('XPHA3')
subplot(5,4,16);plot(x(54,:));title('XAUT3')
xlabel('discrete time k')
subplot(5,4,17);plot(x(55,:));title('XTSS3')
xlabel('discrete time k')
subplot(5,4,18);plot(x(56,:));title('XMeOH3')
xlabel('discrete time k')
subplot(5,4,19);plot(x(57,:));title('XMeP3')
xlabel('discrete time k')
set(figure(3),'name','Tank 3 steady state results','numbertitle','off')

% Data for dynamic simulations
K     = 11;      % Prediction horizon for dynamic simulations

% Dynamic of the inflow concentrations for dry weather file
load dry.txt

% The order is t, SS XBH XS XI SNH SI SND XND Q0
%               1, 2, 3,  4, 5, 6,  7, 8,  9,  10

xi = [SO2i*ones(1,K); SFi*ones(1,K); SAi*ones(1,K); dry(1:K,6)';
      SNO3i*ones(1,K); SPO4i*ones(1,K); dry(1:K,7)'; SALKi*ones(1,K);
      SN2i*ones(1,K); dry(1:K,5)'; dry(1:K,4)'; dry(1:K,3)';
      XPAOi*ones(1,K); XPPi*ones(1,K); XPHAi*ones(1,K); XAUTi*ones(1,K);
      XTSSi*ones(1,K); XMeOHi*ones(1,K); XMePi*ones(1,K)];


x = [x(:,K1+1) zeros(57,K)];

% Dynamic simulation of the model
for k = 1:K
    xk = x(:,k);
    xik = xi(:,k);
    Pk = P_AASM2dF(xk);

    SOk  = [xk(39,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xxlk = [zeros(38,1);xxk(1);zeros(18,1)];
    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xxlk;
end
```

297

```
% Full model response to the dynamic influent concentrations
% Tank 1
figure(4)
subplot(5,4,1);plot(x(1,:));title('SO2_1')
subplot(5,4,2);plot(x(2,:));title('SF1')
subplot(5,4,3);plot(x(3,:));title('SA1')
subplot(5,4,4);plot(x(4,:));title('SNH4_1')
subplot(5,4,5);plot(x(5,:));title('SNO3_1')
subplot(5,4,6);plot(x(6,:));title('SPO4_1')
subplot(5,4,7);plot(x(7,:));title('SI1')
subplot(5,4,8);plot(x(8,:));title('SALK1')
subplot(5,4,9);plot(x(9,:));title('SN2_1')
subplot(5,4,10);plot(x(10,:));title('XI1');
subplot(5,4,11);plot(x(11,:));title('XS1')
subplot(5,4,12);plot(x(12,:));title('XH1')
subplot(5,4,13);plot(x(13,:));title('XPAO1')
subplot(5,4,14);plot(x(14,:));title('XPP1')
subplot(5,4,15);plot(x(15,:));title('XPHA1')
subplot(5,4,16);plot(x(16,:));title('XAUT1')
xlabel('discrete time k')
subplot(5,4,17);plot(x(17,:));title('XTSS1')
xlabel('discrete time k')
subplot(5,4,18);plot(x(18,:));title('XMeOH1')
xlabel('discrete time k')
subplot(5,4,19);plot(x(19,:));title('XMeP1')
xlabel('discrete time k')
set(figure(4),'name','Tank 1 dynamic results','numbertitle','off')

% Tank 2
figure(5)
subplot(5,4,1);plot(x(20,:));title('SO2_2')
subplot(5,4,2);plot(x(21,:));title('SF2')
subplot(5,4,3);plot(x(22,:));title('SA2')
subplot(5,4,4);plot(x(23,:));title('SNH4_2')
subplot(5,4,5);plot(x(24,:));title('SNO3_2')
subplot(5,4,6);plot(x(25,:));title('SPO4_2')
subplot(5,4,7);plot(x(26,:));title('SI2')
subplot(5,4,8);plot(x(27,:));title('SALK2')
subplot(5,4,9);plot(x(28,:));title('SN2_2')
subplot(5,4,10);plot(x(29,:));title('XI2');
subplot(5,4,11);plot(x(30,:));title('XS2')
subplot(5,4,12);plot(x(31,:));title('XH2')
subplot(5,4,13);plot(x(32,:));title('XPAO2')
subplot(5,4,14);plot(x(33,:));title('XPP2')
subplot(5,4,15);plot(x(34,:));title('XPHA2')
subplot(5,4,16);plot(x(35,:));title('XAUT2')
xlabel('discrete time k')
subplot(5,4,17);plot(x(36,:));title('XTSS2')
xlabel('discrete time k')
subplot(5,4,18);plot(x(37,:));title('XMeOH2')
xlabel('discrete time k')
subplot(5,4,19);plot(x(38,:));title('XMeP2')
xlabel('discrete time k')
set(figure(5),'name','Tank 2 dynamic results','numbertitle','off')

% Tank 3
figure(6)
subplot(5,4,1);plot(x(39,:));title('SO2_3')
subplot(5,4,2);plot(x(40,:));title('SF3')
subplot(5,4,3);plot(x(41,:));title('SA3')
subplot(5,4,4);plot(x(42,:));title('SNH4_3')
subplot(5,4,5);plot(x(43,:));title('SNO3_3')
subplot(5,4,6);plot(x(44,:));title('SPO4_3')
subplot(5,4,7);plot(x(45,:));title('SI3')
subplot(5,4,8);plot(x(46,:));title('SALK3')
subplot(5,4,9);plot(x(47,:));title('SN2_3')
```

```
subplot(5,4,10);plot(x(48,:));title('XI3');
subplot(5,4,11);plot(x(49,:));title('XS3')
subplot(5,4,12);plot(x(50,:));title('XH3')
subplot(5,4,13);plot(x(51,:));title('XPAO3')
subplot(5,4,14);plot(x(52,:));title('XPP3')
subplot(5,4,15);plot(x(53,:));title('XPHA3')
subplot(5,4,16);plot(x(54,:));title('XAUT3')
xlabel('discrete time k')
subplot(5,4,17);plot(x(55,:));title('XTSS3')
xlabel('discrete time k')
subplot(5,4,18);plot(x(56,:));title('XMeOH3')
xlabel('discrete time k')
subplot(5,4,19);plot(x(57,:));title('XMeP3')
xlabel('discrete time k')
set(figure(6),'name','Tank 3 dynamic results','numbertitle','off')

% ======================== REDUCED MODEL FORMULATION ========================
% Form the vector of the states (z) for the reduced model, based on the
% ASM1 prediction
%       SF,       SA,      SNH4,     SNO3,      SPO4,     XS,      XPP,      XPHA
z1 = [x(2,:); x(3,:); x(4,:); x(5,:); x(6,:); x(11,:); x(14,:);  x(15,:)];
z2 = [x(21,:);x(22,:);x(23,:);x(24,:);x(25,:);x(30,:); x(33,:);  x(34,:)];
z3 = [x(40,:); x(41,:); x(42,:);x(43,:);x(44,:);x(49,:);x(52,:); x(53,:)];
z  = [z1;z2;z3];

% Formation of the vector of the control variables
u1 = x(1,:);
u2 = x(20,:);
u3 = x(39,:);
u  = [u1; u2; u3];

% Dynamic values of the inflow concentrations for dry weather
zi  = [SFi*ones(1,K); SAi*ones(1,K); SNH4i*ones(1,K); SNO3i*ones(1,K);
       SPO4i*ones(1,K); XSi*ones(1,K); XPPi*ones(1,K); XPHAi*ones(1,K)];

% Steady state values for the biomass XH, XA, SALK, XPP, XPAO, XPHA
XH   = [2551.76*ones(1,K); 2553.38*ones(1,K); 2557.13*ones(1,K);
        2559.18*ones(1,K); 2559.34*ones(1,K)];
XA   = [148.389*ones(1,K); 148.309*ones(1,K); 148.941*ones(1,K);
        149.527*ones(1,K); 149.797*ones(1,K)];
SALK = [4.9*ones(1,K); 148.309*ones(1,K); 148.941*ones(1,K);
        149.527*ones(1,K); 149.797*ones(1,K)];
XPP  = [0.1*ones(1,K); 76.386*ones(1,K); 64.855*ones(1,K);
        55.694*ones(1,K); 49.306*ones(1,K)];
XPAO = [0.1*ones(1,K); 76.386*ones(1,K); 64.855*ones(1,K);
        55.694*ones(1,K); 49.306*ones(1,K)];
XPHA = [0.1*ones(1,K); 76.386*ones(1,K); 64.855*ones(1,K);
        55.694*ones(1,K); 49.306*ones(1,K)];

% MATRIX A for the reduced model (AR)
A11r = 1 - IV11; A22r = 1 - IV22; A33r = 1 - IV33;

A12r = IV1 * Qp;
A23ar = IV2 * (Qa + Qr); A23br = IV2 * (Qa + LAMBDA*Qr);

A21r = IV2*Q1; A32r = IV3*Q2;

A11r = A11r * ones(1,8); A11r = diag(A11r);
A12r = A12r * ones(1,8); A12r = diag(A12r);
A13r = zeros(8,8);

A21r= A21r * ones(1,8); A21r = diag(A21r);
A22r = A22r * ones(1,8); A22r = diag(A22r);
A23r = [A23ar,A23ar,A23ar,A23ar,A23ar,A23br,A23br,A23br];
A23r = diag(A23r);
```

299

```
A31r = zeros(8,8);
A32r = A32r * ones(1,8); A32r = diag(A32r);
A33r = A33r * ones(1,8); A33r = diag(A33r);


AR = [A11r A12r A13r; A21r A22r A23r; A31r A32r A33r];

% MATRIX B for the reduced model (BR)
IV0 = DT / V1*Q0;
f = 1.2;
B11r = [IV0*ones(1,3) IV0*f IV0*ones(1,4)];
B11r = diag(B11r);
BR = [B11r; zeros(16,8)];

% MATRIX C for the reduced model (CR)
%      SF    SA     SNH4   SNO3    SPO4       XS         XPP     XPHA
 CCR=[[1     0    0.01    0       0        -1   0      0];
      [1     0    0.01    0       0        -1   0      0];
      [1     0    0.01    0       0        -1   0      0];
      [-1.6  0   -0.022   0      -0.004     0   0      0];
      [0    -1.6 -0.07    0      -0.02      0   0      0];
      [-1.6  0   -0.022  -0.21   -0.004     0   0      0];
      [0    -1.6 -0.07   -0.21   -0.02      0   0      0];
      [-1    1    0.03     0       0.01     0   0      0];
      [0    -1    0        0       0.4      0  -0.4    1];
      [0     0    0        0      -1        0   1     -0.2];
      [0     0    0       -0.07   -1        0   1     -0.2];
      [0     0   -0.07     0      -0.02     0   0     -1.6];
      [0     0   -0.07    -0.21   -0.02     0   0     -1.6];
      [0     0   -4.24    4.17    -0.02     0   0      0]];

CR = DT*[CCR zeros(14,16); zeros(14,8) CCR zeros(14,8); zeros(14,16) CCR];

% Parameter estimation
for k = 1:K
    zk = z(:,k);
    uk = u(:,k);

    XHk  = XH(:,k);
    XAk  = XA(:,k);
    SALKk = SALK(:,k);
    XPAOk = XPAO(:,k);

    zik  = zi(:,k);
    Pk = P_AASM2dR(zk,uk,XHk,XAk,SALKk,XPAOk);
    FIk = CR'*Pk;
    Y(:,k) = z(:,k+1)-AR*z(:,k)-BR*zi(:,k);
    theta(:,k)=(FIk'*FIk)^(-1)*FIk'*Y(:,k);
end

kHe(1,:)    = theta(1,:);
muHe(1,:)   = theta(2,:);
qfee(1,:)   = theta(3,:);
qPHAe(1,:)  = theta(4,:);
qPPe(1,:)   = theta(5,:);
muPAOe(1,:) = theta(6,:);
muAUTe(1,:) = theta(7,:);

% Dynamic simulation of the reduced model with the estimated parameters
zE = [x(2,1);x(3,1);x(4,1);x(5,1);x(6,1);x(11,1);x(14,1);x(15,1);x(21,1);x(22,1);
x(23,1); x(24,1);x(25,1); x(30,1);
x(33,1);x(34,1);x(40,1);x(41,1);x(42,1);x(43,1);x(44,1);x(45,1);x(52,1);x(53,1)];
ERR = z(:,1)-zE;

% Dynamic simulation of the reduced model with the estimated parameters
for k = 1:K
    zk = z(:,k);
```

```
    uk = u(:,k);
    zik = zi(:,k);

    XHk = XH(:,k);
    XAk = XA(:,k);
    SALKk = SALK(:,k);
    XPAOk = XPAO(:,k);

    kHek    = kHe(:,k);
    muHek   = muHe(:,k);
    qfeek   = qfee(:,k);
    qPHAek  = qPHAe(:,k);
    qPPek   = qPPe(:,k);
    muPAOek = muPAOe(:,k);
    muAUTek = muAUTe(:,k);

    Pk =
P_AASM2dRE(zk,uk,XHk,XAk,SALKk,XPAOk,kHek,muHek,qfeek,qPHAek,qPPek,muPAOek,muAUTek);
    zE(:,k+1) = AR*zE(:,k) + CR'*Pk + BR*zi(:,k);
    ERR(:,k+1)=z(:,k+1)-zE(:,k+1);
end

% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 1
SF1=z(1,:);SA1=z(2,:);SNH4_1=z(3,:);SNO3_1=z(4,:);
SF1E=zE(1,:);SA1E=zE(2,:);SNH4_1E=zE(3,:);SNO3_1E=zE(4,:);
ERRSF1=ERR(1,:);ERRSA1=ERR(2,:);ERRSNH4_1=ERR(3,:);ERRSNO3_1=ERR(4,:);


SPO4_1=z(5,:);XS1=z(6,:);XPP1=z(7,:);XPHA1=z(8,:);
SPO4_1E=zE(5,:);XS1E=zE(6,:);XPP1E=zE(7,:);XPHA1E=zE(8,:);
ERRSPO4_1=ERR(5,:);ERRXS1=ERR(6,:);ERRXPP1=ERR(7,:);ERRXPHA1=ERR(8,:);

% Calculation of the weighted errors for tank 1
eSF1=(ERRSF1*ERRSF1')/(SF1*SF1');
eSA1=(ERRSA1*ERRSA1')/(SA1*SA1');
eSNH4_1=(ERRSNH4_1*ERRSNH4_1')/(SNH4_1*SNH4_1');
eSNO3_1=(ERRSNO3_1*ERRSNO3_1')/(SNO3_1*SNO3_1');
eSPO4_1=(ERRSPO4_1*ERRSPO4_1')/(SPO4_1*SPO4_1');
eXS1=(ERRXS1*ERRXS1')/(XS1*XS1');
eXPP1=(ERRXPP1*ERRXPP1')/(XPP1*XPP1');
eXPHA1=(ERRXPHA1*ERRXPHA1')/(XPHA1*XPHA1');

% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 2
SF2=z(9,:);SA2=z(10,:);SNH4_2=z(11,:);SNO3_2=z(12,:);
SF2E=zE(9,:);SA2E=zE(10,:);SNH4_2E=zE(11,:);SNO3_2E=zE(12,:);
ERRSF2=ERR(9,:);ERRSA2=ERR(10,:);ERRSNH4_2=ERR(11,:);ERRSNO3_2=ERR(12,:);


SPO4_2=z(13,:);XS2=z(14,:);XPP2=z(15,:);XPHA2=z(16,:);
SPO4_2E=zE(13,:);XS2E=zE(14,:);XPP2E=zE(15,:);XPHA2E=zE(16,:);
ERRSPO4_2=ERR(13,:);ERRXS2=ERR(14,:);ERRXPP2=ERR(15,:);ERRXPHA2=ERR(16,:);

% Calculation of the weighted errors for tank 2
eSF2=(ERRSF2*ERRSF2')/(SF2*SF2');
eSA2=(ERRSA2*ERRSA2')/(SA2*SA2');
eSNH4_2=(ERRSNH4_2*ERRSNH4_2')/(SNH4_2*SNH4_2');
eSNO3_2=(ERRSNO3_2*ERRSNO3_2')/(SNO3_2*SNO3_2');
eSPO4_2=(ERRSPO4_2*ERRSPO4_2')/(SPO4_2*SPO4_2');
eXS2=(ERRXS2*ERRXS2')/(XS2*XS2');
eXPP2=(ERRXPP2*ERRXPP2')/(XPP2*XPP2');
eXPHA2=(ERRXPHA2*ERRXPHA2')/(XPHA2*XPHA2');

% Vector of process full, estimated and error variables
% SF,SA,SNH4,SNO3,SPO4,XS,XPP,XPHA, Tank 3
SF3=z(17,:);SA3=z(18,:);SNH4_3=z(19,:);SNO3_3=z(20,:);
SF3E=zE(17,:);SA3E=zE(18,:);SNH4_3E=zE(19,:);SNO3_3E=zE(20,:);
```

301

```
ERRSF3=ERR(17,:);ERRSA3=ERR(18,:);ERRSNH4_3=ERR(19,:);ERRSNO3_3=ERR(20,:);

SPO4_3=z(21,:);XS3=z(22,:);XPP3=z(23,:);XPHA3=z(24,:);
SPO4_3E=zE(21,:);XS3E=zE(22,:);XPP3E=zE(23,:);XPHA3E=zE(24,:);
ERRSPO4_3=ERR(21,:);ERRXS3=ERR(22,:);ERRXPP3=ERR(23,:);ERRXPHA3=ERR(24,:);

% Calculation of the weighted errors for tank 3
eSF3=(ERRSF3*ERRSF3')/(SF3*SF3');
eSA3=(ERRSA3*ERRSA3')/(SA3*SA3');
eSNH4_3=(ERRSNH4_3*ERRSNH4_3')/(SNH4_3*SNH4_3');
eSNO3_3=(ERRSNO3_3*ERRSNO3_3')/(SNO3_3*SNO3_3');
eSPO4_3=(ERRSPO4_3*ERRSPO4_3')/(SPO4_3*SPO4_3');
eXS3=(ERRXS3*ERRXS3')/(XS3*XS3');
eXPP3=(ERRXPP3*ERRXPP3')/(XPP3*XPP3');
eXPHA3=(ERRXPHA3*ERRXPHA3')/(XPHA3*XPHA3');

k = 1:K;
% Graphs of the estimated parameters
figure(7)
subplot(4,2,1);plot(k,kHe,'k');title('Estimated kH');ylabel('kHe');
subplot(4,2,2);plot(k,muHe,'k');title('Estimated muH');ylabel('muHe');
subplot(4,2,3);plot(k,qfee,'k');title('Estimated qfe');ylabel('qfee');
subplot(4,2,4);plot(k,qPHAe,'k');title('Estimated qPHA');ylabel('qPHAe');
subplot(4,2,5);plot(k,qPPe,'k');title('Estimated qPP');ylabel('qPPe');
subplot(4,2,6);plot(k,muPAOe,'k');title('Estimated muPAO');
ylabel('muPAOe');xlabel('discrete time k')
subplot(4,2,7);plot(k,muAUTe,'k');title('Estimated muAUT');
ylabel('muAUTe');xlabel('discrete time k')
set(figure(7),'name','Estimated parameters','numbertitle','off')

k = 1:K+1;
% Graphs of the reduced model response to the dynamic influent
% concentrations

figure(8)
subplot(4,4,1);plot(k,SF1);hold on; plot(k,SF1E,'k-.');ylabel('SF1,SF1E');
legend('SF1','SF1E');subplot(4,4,2); plot(k,ERRSF1,'k');ylabel('ERRSF1');

subplot(4,4,3);plot(k,SA1);hold on; plot(k,SA1E,'k-.');ylabel('SA1,SA1E');
legend('SA1','SA1E');subplot(4,4,4); plot(k,ERRSA1,'k');ylabel('ERRSA1');

subplot(4,4,5);plot(k,SNH4_1);hold on; plot(k,SNH4_1E,'k-.')
legend('SNH4_1','SNH4_1E');ylabel('SNH4_1,SNH4_1E');
subplot(4,4,6); plot(k,ERRSNH4_1,'k');ylabel('ERRSNH4_1');

subplot(4,4,7);plot(k,SNO3_1);hold on; plot(k,SNO3_1E,'k-.')
legend('SNO3_1','SNO3_1E');ylabel('SNO3_1,SNO3_1E');
subplot(4,4,8); plot(k,ERRSNO3_1,'k');ylabel('ERRSNO3_1');

subplot(4,4,9);plot(k,SPO4_1);hold on; plot(k,SPO4_1E,'k-.')
legend('SPO4_1','SPO4_1E');ylabel('SPO4_1,SPO4_1E');
subplot(4,4,10); plot(k,ERRSPO4_1,'k');ylabel('ERRSPO4_1');

subplot(4,4,11);plot(k,XS1);hold on; plot(k,XS1E,'k-.');ylabel('XS1,XS1E');
legend('XS1','XS1E');subplot(4,4,12); plot(k,ERRXS1,'k');ylabel('ERRXS1');

subplot(4,4,13);plot(k,XPP1);hold on; plot(k,XPP1E,'k-.')
legend('XPP1','XPP1E');ylabel('XPP1,XPP1E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP1,'k');
ylabel('ERRXPP1');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA1);hold on; plot(k,XPHA1E,'k-.')
legend('XPHA1','XPHA1E');ylabel('XPHA1,XPHA1E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA1,'k');
ylabel('ERRXPHA1');xlabel('discrete time k')
```

302

```
set(figure(8),'name','Tank 1 Process variables and estimation
error','numbertitle','off')

figure(9)
subplot(4,4,1);plot(k,SF2);hold on; plot(k,SF2E,'k-.');ylabel('SF2,SF2E');
legend('SF2','SF2E');subplot(4,4,2); plot(k,ERRSF2,'k');ylabel('ERRSF2');

subplot(4,4,3);plot(k,SA2);hold on; plot(k,SA2E,'k-.');ylabel('SA2,SA2E');
legend('SA2','SA2E');subplot(4,4,4); plot(k,ERRSA2,'k');ylabel('ERRSA2');

subplot(4,4,5);plot(k,SNH4_2);hold on; plot(k,SNH4_2E,'k-.')
legend('SNH4_2','SNH4_2E');ylabel('SNH4_2,SNH4_2E');
subplot(4,4,6); plot(k,ERRSNH4_2,'k');ylabel('ERRSNH4_2');

subplot(4,4,7);plot(k,SNO3_2);hold on; plot(k,SNO3_2E,'k-.')
legend('SNO3_2','SNO3_2E');ylabel('SNO3_2,SNO3_2E');
subplot(4,4,8); plot(k,ERRSNO3_2,'k');ylabel('ERRSNO3_2');

subplot(4,4,9);plot(k,SPO4_2);hold on; plot(k,SPO4_2E,'k-.')
legend('SPO4_2','SPO4_2E');ylabel('SPO4_2,SPO4_2E');
subplot(4,4,10); plot(k,ERRSPO4_2,'k');ylabel('ERRSPO4_2');

subplot(4,4,11);plot(k,XS2);hold on; plot(k,XS2E,'k-.');ylabel('XS2,XS2E');
legend('XS2','XS2E');subplot(4,4,12); plot(k,ERRXS2,'k');ylabel('ERRXS2');

subplot(4,4,13);plot(k,XPP2);hold on; plot(k,XPP2E,'k-.');
legend('XPP2','XPP2E');ylabel('XPP2,XPP2E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP2,'k');
ylabel('ERRXPP2');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA2);hold on; plot(k,XPHA2E,'k-.')
legend('XPHA2','XPHA2E');ylabel('XPHA2,XPHA2E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA2,'k');
ylabel('ERRXPHA2');xlabel('discrete time k')
set(figure(9),'name','Tank 2 Process variables and estimation
error','numbertitle','off')

figure(10)
subplot(4,4,1);plot(k,SF3);hold on; plot(k,SF3E,'k-.');ylabel('SF3,SF3E');
legend('SF3','SF3E');subplot(4,4,2); plot(k,ERRSF3,'k');ylabel('ERRSF3');

subplot(4,4,3);plot(k,SA3);hold on; plot(k,SA3E,'k-.');ylabel('SA3,SA3E');
legend('SA3','SA3E');subplot(4,4,4); plot(k,ERRSA3,'k');ylabel('ERRSA3');

subplot(4,4,5);plot(k,SNH4_3);hold on; plot(k,SNH4_3E,'k-.')
legend('SNH4_3','SNH4_3E');ylabel('SNH4_3,SNH4_3E');
subplot(4,4,6); plot(k,ERRSNH4_3,'k');ylabel('ERRSNH4_3');

subplot(4,4,7);plot(k,SNO3_3);hold on; plot(k,SNO3_3E,'k-.')
legend('SNO3_3','SNO3_3E');ylabel('SNO3_3,SNO3_3E');
subplot(4,4,8); plot(k,ERRSNO3_3,'k');ylabel('ERRSNO3_3');

subplot(4,4,9);plot(k,SPO4_3);hold on; plot(k,SPO4_3E,'k-.')
legend('SPO4_3','SPO4_3E');ylabel('SPO4_3,SPO4_3E');
subplot(4,4,10); plot(k,ERRSPO4_3,'k');ylabel('ERRSPO4_3');

subplot(4,4,11);plot(k,XS3);hold on; plot(k,XS3E,'k-.')
legend('XS3','XS3E');ylabel('XS3,XS3E');
subplot(4,4,12); plot(k,ERRXS3,'k');ylabel('ERRXS3');

subplot(4,4,13);plot(k,XPP3);hold on; plot(k,XPP3E,'k-.')
legend('XPP3','XPP3E');ylabel('XPP3,XPP3E');xlabel('discrete time k')
subplot(4,4,14); plot(k,ERRXPP3,'k');
ylabel('ERRXPP3');xlabel('discrete time k')

subplot(4,4,15);plot(k,XPHA3);hold on; plot(k,XPHA3E,'k-.')
```

```
legend('XPHA3','XPHA3E');ylabel('XPHA3,XPHA3E');xlabel('discrete time k')
subplot(4,4,16); plot(k,ERRXPHA3,'k');
ylabel('ERRXPHA3');xlabel('discrete time k')
set(figure(10),'name','Tank 3 Process variables and estimation
error','numbertitle','off')
```

G.2: MATLAB script file – P_AASM2dF.m

```
% M-File: P_AASM2dF.m
% ============================ M-FILE DESCRIPTION ============================
% This m-file is used for:
% Calculation of the process rates for the full model

function  P = P_AASM2dF(xk)

global fSI YH fXI YPO4 YPHA iPBM YAUT iNBM Kh etaNO3 etafe muH qfe bH qPHA qPP muPAO
bPAO bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4 KALK KA Kfe KPP KPS KP KPHA KIPP
KMAX
% TANK 1
P(1)= Kh * (xk(1)/(KO2+xk(1))) * ((xk(11)/xk(12))/(KX+(xk(11)/xk(12)))) * xk(12);
P(2)= Kh * etaNO3 * (KO2/(KO2+xk(1))) * (xk(5)/(KNO3+xk(5))) *
((xk(11)/xk(12))/(KX+(xk(11)/xk(12)))) * xk(12);
P(3)= Kh * etafe * (KO2/(KO2+xk(1))) * (KNO3/(KNO3+xk(5))) *
((xk(11)/xk(12))/(KX+(xk(11)/xk(12)))) * xk(12);
P(4)= muH * (xk(1)/(KO2+xk(1))) * (xk(2)/(KF+xk(2))) * (xk(2)/(xk(2)+xk(3))) *
(xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) * (xk(8)/(KALK+xk(8))) * xk(12);
P(5)= muH * (xk(1)/(KO2+xk(1))) * (xk(3)/(KA+xk(3))) * (xk(3)/(xk(2)+xk(3))) *
(xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) * (xk(8)/(KALK+xk(8))) * xk(12);
P(6)= muH * etaNO3 * (KO2/(KO2+xk(1))) * (KNO3/(KNO3+xk(5))) * (xk(2)/(KF+xk(2))) *
(xk(2)/(xk(2)+xk(3))) * (xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) *
(xk(8)/(KALK+xk(8))) * xk(12);
P(7)= muH * etaNO3 * (KO2/(KO2+xk(1))) * (KNO3/(KNO3+xk(5))) * (xk(3)/(KA+xk(3))) *
(xk(3)/(xk(2)+xk(3))) * (xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) *
(xk(8)/(KALK+xk(8)))* xk(12);
P(8)= qfe * (KO2/(KO2+xk(1))) * (KNO3/(KNO3+xk(5))) * (xk(2)/(KF+xk(2))) *
(xk(8)/(KALK+xk(8)))* xk(12);
P(9)= bH * xk(12);
P(10)= qPHA * (xk(3)/(KA+xk(3))) * (xk(8)/(KALK+xk(8))) *
((xk(14)/xk(13))/(KPP+(xk(14)/xk(13)))) * xk(13);
P(11)= qPP * (xk(1)/(KO2+xk(1))) * (xk(6)/(KPS+xk(6))) * (xk(8)/(KALK+xk(8))) *
((xk(15)/xk(13))/(KPHA+(xk(15)/xk(13)))) * ((KMAX-(xk(14)/xk(13)))/(KPP+KMAX-
(xk(14)/xk(13)))) * xk(13);
P(12)= qPP * (xk(1)/(KO2+xk(1))) * (xk(6)/(KPS+xk(6))) * (xk(8)/(KALK+xk(8))) *
((xk(15)/xk(13))/(KPHA+(xk(15)/xk(13)))) * ((KMAX-(xk(14)/xk(13)))/(KPP+KMAX-
(xk(14)/xk(13)))) * xk(13) * etaNO3 * KO2/xk(1) * (xk(5)/(KNO3+xk(5)));
P(13)= muPAO * (xk(1)/(KO2+xk(1))) * (xk(4)/(KNH4+xk(4))) * (xk(8)/(KALK+xk(8)))*
(xk(6)/(KP+xk(6))) * ((xk(15)/xk(13))/(KPHA+(xk(15)/xk(13)))) * xk(13);
P(14)= muPAO * (xk(1)/(KO2+xk(1))) * (xk(4)/(KNH4+xk(4))) * (xk(8)/(KALK+xk(8)))*
(xk(6)/(KP+xk(6))) * ((xk(15)/xk(13))/(KPHA+(xk(15)/xk(13)))) * xk(13) * etaNO3 *
KO2/xk(1) * (xk(5)/(KNO3+xk(5)));
P(15)= bPAO * xk(13) * (xk(8)/(KALK+xk(8)));
P(16)= bPP * xk(14) * (xk(8)/(KALK+xk(8)));
P(17)= bPHA * xk(15) * (xk(8)/(KALK+xk(8)));
P(18)= muAUT * (xk(1)/(KO2+xk(1))) * (xk(4)/(KNH4+xk(4))) * (xk(6)/(KP+xk(6))) *
(xk(8)/(KALK+xk(8))) * xk(16);
P(19)= bAUT * xk(16);
P(20)= kPRE * xk(6) * xk(18);
P(21)= kRED * xk(19) * (xk(8)/(KALK+xk(8)));

% TANK 2
P(22)= Kh * (xk(20)/(KO2+xk(20))) * ((xk(30)/xk(31))/(KX+(xk(30)/xk(31)))) * xk(31);
P(23)= Kh * etaNO3 * (KO2/(KO2+xk(20))) * (xk(24)/(KNO3+xk(24))) *
((xk(30)/xk(31))/(KX+(xk(30)/xk(31)))) * xk(31);
```

```
P(24)= Kh * etafe * (KO2/(KO2+xk(20))) * (KNO3/(KNO3+xk(24))) *
((xk(30)/xk(31))/(KX+(xk(30)/xk(31)))) * xk(31);
P(25)= muH * (xk(20)/(KO2+xk(20))) * (xk(21)/(KF+xk(21))) * (xk(21)/(xk(21)+xk(22))) *
(xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) * (xk(27)/(KALK+xk(27))) * xk(31);
P(26)= muH * (xk(20)/(KO2+xk(20))) * (xk(22)/(KA+xk(22))) * (xk(22)/(xk(21)+xk(22))) *
(xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) * (xk(27)/(KALK+xk(27))) * xk(31);
P(27)= muH * etaNO3 * (KO2/(KO2+xk(20))) * (KNO3/(KNO3+xk(24))) * (xk(21)/(KF+xk(21)))
* (xk(21)/(xk(21)+xk(22))) * (xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) *
(xk(27)/(KALK+xk(27))) * xk(31);
P(28)= muH * etaNO3 * (KO2/(KO2+xk(20))) * (KNO3/(KNO3+xk(24))) * (xk(22)/(KA+xk(22)))
* (xk(22)/(xk(21)+xk(22))) * (xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) *
(xk(27)/(KALK+xk(27))) * xk(31);
P(29)= qfe * (KO2/(KO2+xk(20))) * (KNO3/(KNO3+xk(24))) * (xk(21)/(KF+xk(21))) *
(xk(27)/(KALK+xk(27)))* xk(31);
P(30)= bH * xk(31);
P(31)= qPHA * (xk(22)/(KA+xk(22))) * (xk(27)/(KALK+xk(27))) *
((xk(33)/xk(32))/(KPP+(xk(33)/xk(32)))) * xk(32);
P(32)= qPP * (xk(20)/(KO2+xk(20))) * (xk(25)/(KPS+xk(25))) * (xk(27)/(KALK+xk(27))) *
((xk(34)/xk(32))/(KPHA+(xk(34)/xk(32)))) * ((KMAX-(xk(33)/xk(32)))/(KPP+KMAX-
(xk(33)/xk(32)))) * xk(32);
P(33)= qPP * (xk(20)/(KO2+xk(20))) * (xk(25)/(KPS+xk(25))) * (xk(27)/(KALK+xk(27))) *
((xk(34)/xk(32))/(KPHA+(xk(34)/xk(32)))) * ((KMAX-(xk(33)/xk(32)))/(KPP+KMAX-
(xk(33)/xk(32)))) * xk(32) * etaNO3 * KO2/xk(20) * (xk(24)/(KNO3+xk(24)));
P(34)= muPAO * (xk(20)/(KO2+xk(20))) * (xk(23)/(KNH4+xk(23))) * (xk(27)/(KALK+xk(27)))*
(xk(25)/(KP+xk(25))) * ((xk(34)/xk(32))/(KPHA+(xk(34)/xk(32)))) * xk(32);
P(35)= muPAO * (xk(20)/(KO2+xk(20))) * (xk(23)/(KNH4+xk(23))) * (xk(27)/(KALK+xk(27)))*
(xk(25)/(KP+xk(25))) * ((xk(34)/xk(32))/(KPHA+(xk(34)/xk(32)))) * xk(32) * etaNO3 *
KO2/xk(20) * (xk(24)/(KNO3+xk(24)));
P(36)= bPAO * xk(32) * (xk(27)/(KALK+xk(27)));
P(37)= bPP * xk(33) * (xk(27)/(KALK+xk(27)));
P(38)= bPHA * xk(34) * (xk(27)/(KALK+xk(27)));
P(39)= muAUT * (xk(20)/(KO2+xk(20))) * (xk(23)/(KNH4+xk(23))) * (xk(25)/(KP+xk(25))) *
(xk(27)/(KALK+xk(27))) * xk(35);
P(40)= bAUT * xk(35);
P(41)= kPRE * xk(25) * xk(37);
P(42)= kRED * xk(38) * (xk(27)/(KALK+xk(27)));

% TANK 3
P(43)= Kh * (xk(39)/(KO2+xk(39))) * ((xk(49)/xk(50))/(KX+(xk(49)/xk(50)))) * xk(50);
P(44)= Kh * etaNO3 * (KO2/(KO2+xk(39))) * (xk(43)/(KNO3+xk(43))) *
((xk(49)/xk(50))/(KX+(xk(49)/xk(50)))) * xk(50);
P(45)= Kh * etafe * (KO2/(KO2+xk(39))) * (KNO3/(KNO3+xk(43))) *
((xk(49)/xk(50))/(KX+(xk(49)/xk(50)))) * xk(50);
P(46)= muH * (xk(39)/(KO2+xk(39))) * (xk(40)/(KF+xk(40))) * (xk(40)/(xk(40)+xk(41))) *
(xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) * (xk(46)/(KALK+xk(46))) * xk(50);
P(47)= muH * (xk(39)/(KO2+xk(39))) * (xk(41)/(KA+xk(41))) * (xk(41)/(xk(40)+xk(41))) *
(xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) * (xk(46)/(KALK+xk(46))) * xk(50);
P(48)= muH * etaNO3 * (KO2/(KO2+xk(39))) * (KNO3/(KNO3+xk(43))) * (xk(40)/(KF+xk(40)))
* (xk(40)/(xk(40)+xk(41))) * (xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) *
(xk(46)/(KALK+xk(46))) * xk(50);
P(49)= muH * etaNO3 * (KO2/(KO2+xk(39))) * (KNO3/(KNO3+xk(43))) * (xk(41)/(KA+xk(41)))
* (xk(41)/(xk(40)+xk(41))) * (xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) *
(xk(46)/(KALK+xk(46))) * xk(50);
P(50)= qfe * (KO2/(KO2+xk(39))) * (KNO3/(KNO3+xk(43))) * (xk(40)/(KF+xk(40))) *
(xk(46)/(KALK+xk(46)))* xk(50);
P(51)= bH * xk(50);
P(52)= qPHA * (xk(41)/(KA+xk(41))) * (xk(46)/(KALK+xk(46))) *
((xk(52)/xk(51))/(KPP+(xk(52)/xk(51)))) * xk(51);
P(53)= qPP * (xk(39)/(KO2+xk(39))) * (xk(44)/(KPS+xk(44))) * (xk(46)/(KALK+xk(46))) *
((xk(53)/xk(51))/(KPHA+(xk(53)/xk(51)))) * ((KMAX-(xk(52)/xk(51)))/(KPP+KMAX-
(xk(52)/xk(51)))) * xk(51);
P(54)= qPP * (xk(39)/(KO2+xk(39))) * (xk(44)/(KPS+xk(44))) * (xk(46)/(KALK+xk(46))) *
((xk(53)/xk(51))/(KPHA+(xk(53)/xk(51)))) * ((KMAX-(xk(52)/xk(51)))/(KPP+KMAX-
(xk(52)/xk(51)))) * xk(51) * etaNO3 * KO2/xk(39) * (xk(43)/(KNO3+xk(43)));
P(55)= muPAO * (xk(39)/(KO2+xk(39))) * (xk(42)/(KNH4+xk(42))) * (xk(46)/(KALK+xk(46)))*
(xk(44)/(KP+xk(44))) * ((xk(53)/xk(51))/(KPHA+(xk(53)/xk(51)))) * xk(51);
```

```
P(56)= muPAO * (xk(39)/(KO2+xk(39))) * (xk(42)/(KNH4+xk(42))) * (xk(46)/(KALK+xk(46)))*
(xk(44)/(KP+xk(44))) * ((xk(53)/xk(51))/(KPHA+(xk(53)/xk(51)))) * xk(51) * etaNO3 *
KO2/xk(39) * (xk(43)/(KNO3+xk(43)));
P(57)= bPAO * xk(51) * (xk(46)/(KALK+xk(46)));
P(58)= bPP * xk(52) * (xk(46)/(KALK+xk(46)));
P(59)= bPHA * xk(53) * (xk(46)/(KALK+xk(46)));
P(60)= muAUT * (xk(39)/(KO2+xk(39))) * (xk(42)/(KNH4+xk(42))) * (xk(44)/(KP+xk(44))) *
(xk(46)/(KALK+xk(46))) * xk(54);
P(61)= bAUT * xk(54);
P(62)= kPRE * xk(44) * xk(56);
P(63)= kRED * xk(57) * (xk(46)/(KALK+xk(46)));


P = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20) P(21) P(22) P(23) P(24) P(25) P(26) P(27) P(28) P(29)
P(30) P(31) P(32) P(33) P(34) P(35) P(36) P(37) P(38) P(39) P(40) P(41) P(42) P(43)
P(44) P(45) P(46) P(47) P(48) P(49) P(50) P(51) P(52) P(53) P(54) P(55) P(56) P(57)
P(58) P(59) P(60) P(61) P(62) P(63)]';
```

G.3: MATLAB script file – P_AASM2dR.m

```
% M-File: P_AASM2dR.m
% ======================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model

function    Pk = P_AASM2dR(zk,uk,XHk,XAk,SALKk,XPAOk)

global      fSI YH fXI YPO4 YPHA iPBM YAUT iNBM Kh etaNO3 etafe muH qfe bH qPHA qPP
muPAO bPAO bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4 KALK KA Kfe KPP KPS KP
KPHA KIPP KMAX
% TANK 1
%     SF     SA     SNH4     SNO3     SPO4     XS     XPP     XPHA
P(1)= (uk(1)/(KO2+uk(1))) * ((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(2)= etaNO3 * (KO2/(KO2+uk(1))) * (zk(4)/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(3)= etafe * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(4)= (uk(1)/(KO2+uk(1))) * (zk(1)/(KF+zk(1))) * (zk(1)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(5)= (uk(1)/(KO2+uk(1))) * (zk(2)/(KA+zk(2))) * (zk(2)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(6)= etaNO3 * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(1)/(KF+zk(1))) *
(zk(1)/(zk(1)+zk(2))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(7)= etaNO3 * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(2)/(KA+zk(2))) *
(zk(2)/(zk(1)+zk(2))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(8)= (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(1)/(KF+zk(1))) *
(SALKk(1)/(KALK+SALKk(1)))* XHk(1);
P(9)= (zk(2)/(KA+zk(2))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(7)/XPAOk(1))/(KPP+(zk(7)/XPAOk(1)))) * XPAOk(1);
P(10)= (uk(1)/(KO2+uk(1))) * (zk(5)/(KPS+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * ((KMAX-(zk(7)/XPAOk(1)))/(KPP+KMAX-
(zk(7)/XPAOk(1)))) * XPAOk(1);
P(11)= (uk(1)/(KO2+uk(1))) * (zk(5)/(KPS+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * ((KMAX-(zk(7)/XPAOk(1)))/(KPP+KMAX-
(zk(7)/XPAOk(1)))) * XPAOk(1) * etaNO3 * KO2/uk(1) * (zk(4)/(KNO3+zk(4)));
P(12)= (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) * (SALKk(1)/(KALK+SALKk(1)))*
(zk(5)/(KP+zk(5))) * ((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * XPAOk(1);
P(13)= (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) * (SALKk(1)/(KALK+SALKk(1)))*
(zk(5)/(KP+zk(5))) * ((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * XPAOk(1) * etaNO3 *
KO2/uk(1) * (zk(4)/(KNO3+zk(4)));
```

306

```
P(14)= (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XAk(1);


P1k = [[P(1)        P(2)      P(3)         zeros(1,11)                    ];
       [zeros(1,3)  P(4)      P(5)         P(6)          P(7)   zeros(1,7)];
       [zeros(1,7)  P(8)      zeros(1,6)                                  ];
       [zeros(1,8)  P(9)      zeros(1,5)                                  ];
       [zeros(1,9)  P(10)     P(11)        zeros(1,3)                     ];
       [zeros(1,11) P(12)     P(13)        0                              ];
       [zeros(1,13) P(14)                                              ]]';
% TANK 2
P(15)= (uk(2)/(KO2+uk(2))) * ((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(16)= etaNO3 * (KO2/(KO2+uk(2))) * (zk(12)/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(17)= etafe * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(18)= (uk(2)/(KO2+uk(2))) * (zk(9)/(KF+zk(9))) * (zk(9)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(19)= (uk(2)/(KO2+uk(2))) * (zk(10)/(KA+zk(10))) * (zk(10)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(20)= etaNO3 * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(9)/(KF+zk(9))) *
(zk(9)/(zk(9)+zk(10))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(21)= etaNO3 * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(10)/(KA+zk(10))) *
(zk(10)/(zk(9)+zk(10))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2)))* XHk(2);
P(22)= (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(9)/(KF+zk(9))) *
(SALKk(2)/(KALK+SALKk(2)))* XHk(2);
P(23)= (zk(10)/(KA+zk(10))) * (SALKk(2)/(KALK+SALKk(2))) *
((zk(15)/XPAOk(2))/(KPP+(zk(15)/XPAOk(2)))) * XPAOk(2);
P(24)= (uk(2)/(KO2+uk(2))) * (zk(13)/(KPS+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) *
((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * ((KMAX-(zk(15)/XPAOk(2)))/(KPP+KMAX-
(zk(15)/XPAOk(2)))) * XPAOk(2);
P(25)= (uk(2)/(KO2+uk(2))) * (zk(13)/(KPS+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) *
((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * ((KMAX-(zk(15)/XPAOk(2)))/(KPP+KMAX-
(zk(15)/XPAOk(2)))) * XPAOk(2) * etaNO3 * KO2/uk(2) * (zk(12)/(KNO3+zk(12)));
P(26)= (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (SALKk(2)/(KALK+SALKk(2)))*
(zk(13)/(KP+zk(13))) * ((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * XPAOk(2);
P(27)= (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (SALKk(2)/(KALK+SALKk(2)))*
(zk(13)/(KP+zk(13))) * ((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * XPAOk(2) * etaNO3
* KO2/uk(2) * (zk(12)/(KNO3+zk(12)));
P(28)= (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XAk(2);


P2k = [[P(15)       P(16)     P(17)        zeros(1,11)                    ];
       [zeros(1,3)  P(18)     P(19)        P(20)         P(21)  zeros(1,7)];
       [zeros(1,7)  P(22)     zeros(1,6)                                  ];
       [zeros(1,8)  P(23)     zeros(1,5)                                  ];
       [zeros(1,9)  P(24)     P(25)        zeros(1,3)                     ];
       [zeros(1,11) P(26)     P(27)        0                              ];
       [zeros(1,13) P(28)                                              ]]';

% TANK 3
P(29)= (uk(3)/(KO2+uk(3))) * ((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(30)= etaNO3 * (KO2/(KO2+uk(3))) * (zk(20)/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(31)= etafe * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(32)= (uk(3)/(KO2+uk(3))) * (zk(17)/(KF+zk(17))) * (zk(17)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(33)= (uk(3)/(KO2+uk(3))) * (zk(18)/(KA+zk(18))) * (zk(18)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(34)= etaNO3 * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(17)/(KF+zk(17))) *
(zk(17)/(zk(17)+zk(18))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XHk(3);
```

307

```
P(35)= etaNO3 * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(18)/(KA+zk(18))) *
(zk(18)/(zk(17)+zk(18))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3)))* XHk(3);
P(36)= (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(17)/(KF+zk(17))) *
(SALKk(3)/(KALK+SALKk(3)))* XHk(3);
P(37)= (zk(18)/(KA+zk(18))) * (SALKk(3)/(KALK+SALKk(3))) *
((zk(23)/XPAOk(3))/(KPP+(zk(23)/XPAOk(3)))) * XPAOk(3);
P(38)= (uk(3)/(KO2+uk(3))) * (zk(21)/(KPS+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) *
((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * ((KMAX-(zk(23)/XPAOk(3)))/(KPP+KMAX-
(zk(23)/XPAOk(3)))) * XPAOk(3);
P(39)= (uk(3)/(KO2+uk(3))) * (zk(21)/(KPS+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) *
((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * ((KMAX-(zk(23)/XPAOk(3)))/(KPP+KMAX-
(zk(23)/XPAOk(3)))) * XPAOk(3) * etaNO3 * KO2/uk(3) * (zk(20)/(KNO3+zk(20)));
P(40)= (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (SALKk(3)/(KALK+SALKk(3)))*
(zk(21)/(KP+zk(21))) * ((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * XPAOk(3);
P(41)= (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (SALKk(3)/(KALK+SALKk(3)))*
(zk(21)/(KP+zk(21))) * ((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * XPAOk(3) * etaNO3
* KO2/uk(3) * (zk(20)/(KNO3+zk(20)));
P(42)= (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XAk(3);


P3k = [[P(29)          P(30)     P(31)        zeros(1,11)                    ];
       [zeros(1,3)     P(32)     P(33)        P(34)         P(35)   zeros(1,7)];
       [zeros(1,7)     P(36)     zeros(1,6)                                  ];
       [zeros(1,8)     P(37)     zeros(1,5)                                  ];
       [zeros(1,9)     P(38)     P(39)        zeros(1,3)                     ];
       [zeros(1,11)    P(40)     P(41)        0                             ];
       [zeros(1,13)    P(42)                                               ]]';

%Matrix Pk for the whole process
Pk = [P1k; P2k; P3k];
```

G.4: MATLAB script file – P_AASM2dRE.m

```
% M-File: P_AASM2dRE.m
% ===================== M-FILE DESCRIPTION =====================
% This m-file is used for:
% Description of the function function
function    Pk =
P_AASM2dRE(zk,uk,XHk,XAk,SALKk,XPAOk,kHek,muHek,qfeek,qPHAek,qPPek,muPAOek,muAUTek)

global fSI YH fXI YPO4 YPHA iPBM YAUT iNBM Kh etaNO3 etafe muH qfe bH qPHA qPP muPAO
bPAO bPP bPHA muAUT bAUT kPRE kRED KO2 KX KNO3 KF KNH4 KALK KA Kfe KPP KPS KP KPHA KIPP
KMAX

% TANK 1
% SF     SA     SNH4   SNO3   SPO4   XS      XPP     XPHA
% zk(1) zk(2) zk(3) zk(4) zk(5) zk(6) zk(7) zk(8)
P(1)= kHek * (uk(1)/(KO2+uk(1))) * ((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(2)= kHek * etaNO3 * (KO2/(KO2+uk(1))) * (zk(4)/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(3)= kHek * etafe * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) *
((zk(6)/XHk(1))/(KX+(zk(6)/XHk(1)))) * XHk(1);
P(4)= muHek * (uk(1)/(KO2+uk(1))) * (zk(1)/(KF+zk(1))) * (zk(1)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(5)= muHek * (uk(1)/(KO2+uk(1))) * (zk(2)/(KA+zk(2))) * (zk(2)/(zk(1)+zk(2))) *
(zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(6)= muHek * etaNO3 * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(1)/(KF+zk(1))) *
(zk(1)/(zk(1)+zk(2))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XHk(1);
P(7)= muHek * etaNO3 * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(2)/(KA+zk(2))) *
(zk(2)/(zk(1)+zk(2))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XHk(1);
```

```
P(8)= qfeek * (KO2/(KO2+uk(1))) * (KNO3/(KNO3+zk(4))) * (zk(1)/(KF+zk(1))) *
(SALKk(1)/(KALK+SALKk(1)))* XHk(1);
P(9)= qPHAek * (zk(2)/(KA+zk(2))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(7)/XPAOk(1))/(KPP+(zk(7)/XPAOk(1)))) * XPAOk(1);
P(10)= qPPek * (uk(1)/(KO2+uk(1))) * (zk(5)/(KPS+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * ((KMAX-(zk(7)/XPAOk(1)))/(KPP+KMAX-
(zk(7)/XPAOk(1)))) * XPAOk(1);
P(11)= qPPek * (uk(1)/(KO2+uk(1))) * (zk(5)/(KPS+zk(5))) * (SALKk(1)/(KALK+SALKk(1))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * ((KMAX-(zk(7)/XPAOk(1)))/(KPP+KMAX-
(zk(7)/XPAOk(1)))) * XPAOk(1) * etaNO3 * KO2/uk(1) * (zk(4)/(KNO3+zk(4)));
P(12)= muPAOek * (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) *
(SALKk(1)/(KALK+SALKk(1)))* (zk(5)/(KP+zk(5))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * XPAOk(1);
P(13)= muPAOek * (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) *
(SALKk(1)/(KALK+SALKk(1)))* (zk(5)/(KP+zk(5))) *
((zk(8)/XPAOk(1))/(KPHA+(zk(8)/XPAOk(1)))) * XPAOk(1) * etaNO3 * KO2/uk(1) *
(zk(4)/(KNO3+zk(4)));
P(14)= muAUTek * (uk(1)/(KO2+uk(1))) * (zk(3)/(KNH4+zk(3))) * (zk(5)/(KP+zk(5))) *
(SALKk(1)/(KALK+SALKk(1))) * XAk(1);


% TANK 2
% SF    SA    SNH4  SNO3  SPO4  XS    XPP   XPHA
% zk(9) zk(10) zk(11) zk(12) zk(13) zk(14) zk(15) zk(16)
P(15)= kHek * (uk(2)/(KO2+uk(2))) * ((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(16)= kHek * etaNO3 * (KO2/(KO2+uk(2))) * (zk(12)/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(17)= kHek * etafe * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) *
((zk(14)/XHk(2))/(KX+(zk(14)/XHk(2)))) * XHk(2);
P(18)= muHek * (uk(2)/(KO2+uk(2))) * (zk(9)/(KF+zk(9))) * (zk(9)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(19)= muHek * (uk(2)/(KO2+uk(2))) * (zk(10)/(KA+zk(10))) * (zk(10)/(zk(9)+zk(10))) *
(zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) * (SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(20)= muHek * etaNO3 * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(9)/(KF+zk(9))) *
(zk(9)/(zk(9)+zk(10))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(21)= muHek * etaNO3 * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(10)/(KA+zk(10)))
* (zk(10)/(zk(9)+zk(10))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XHk(2);
P(22)= qfeek * (KO2/(KO2+uk(2))) * (KNO3/(KNO3+zk(12))) * (zk(9)/(KF+zk(9))) *
(SALKk(2)/(KALK+SALKk(2)))* XHk(2);
P(23)= qPHAek * (zk(10)/(KA+zk(10))) * (SALKk(2)/(KALK+SALKk(2))) *
((zk(15)/XPAOk(2))/(KPP+(zk(15)/XPAOk(2)))) * XPAOk(2);
P(24)= qPPek * (uk(2)/(KO2+uk(2))) * (zk(13)/(KPS+zk(13))) * (SALKk(2)/(KALK+SALKk(2)))
* ((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * ((KMAX-(zk(15)/XPAOk(2)))/(KPP+KMAX-
(zk(15)/XPAOk(2)))) * XPAOk(2);
P(25)= qPPek * (uk(2)/(KO2+uk(2))) * (zk(13)/(KPS+zk(13))) * (SALKk(2)/(KALK+SALKk(2)))
* ((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * ((KMAX-(zk(15)/XPAOk(2)))/(KPP+KMAX-
(zk(15)/XPAOk(2)))) * XPAOk(2) * etaNO3 * KO2/uk(2) * (zk(12)/(KNO3+zk(12)));
P(26)= muPAOek * (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) *
(SALKk(2)/(KALK+SALKk(2)))* (zk(13)/(KP+zk(13))) *
((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * XPAOk(2);
P(27)= muPAOek * (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) *
(SALKk(2)/(KALK+SALKk(2)))* (zk(13)/(KP+zk(13))) *
((zk(16)/XPAOk(2))/(KPHA+(zk(16)/XPAOk(2)))) * XPAOk(2) * etaNO3 * KO2/uk(2) *
(zk(12)/(KNO3+zk(12)));
P(28)= muAUTek * (uk(2)/(KO2+uk(2))) * (zk(11)/(KNH4+zk(11))) * (zk(13)/(KP+zk(13))) *
(SALKk(2)/(KALK+SALKk(2))) * XAk(2);


% TANK 3
% SF    SA    SNH4  SNO3  SPO4  XS    XPP   XPHA
% zk(17) zk(18) zk(19) zk(20) zk(21) zk(22) zk(23) zk(24)
P(29)= kHek * (uk(3)/(KO2+uk(3))) * ((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(30)= kHek * etaNO3 * (KO2/(KO2+uk(3))) * (zk(20)/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
P(31)= kHek * etafe * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) *
((zk(22)/XHk(3))/(KX+(zk(22)/XHk(3)))) * XHk(3);
```

```
P(32)= muHek * (uk(3)/(KO2+uk(3))) * (zk(17)/(KF+zk(17))) * (zk(17)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(33)= muHek * (uk(3)/(KO2+uk(3))) * (zk(18)/(KA+zk(18))) * (zk(18)/(zk(17)+zk(18))) *
(zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) * (SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(34)= muHek * etaNO3 * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(17)/(KF+zk(17)))
* (zk(17)/(zk(17)+zk(18))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(35)= muHek * etaNO3 * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(18)/(KA+zk(18)))
* (zk(18)/(zk(17)+zk(18))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XHk(3);
P(36)= qfeek * (KO2/(KO2+uk(3))) * (KNO3/(KNO3+zk(20))) * (zk(17)/(KF+zk(17))) *
(SALKk(3)/(KALK+SALKk(3)))* XHk(3);
P(37)= qPHAek * (zk(18)/(KA+zk(18))) * (SALKk(3)/(KALK+SALKk(3))) *
((zk(23)/XPAOk(3))/(KPP+(zk(23)/XPAOk(3)))) * XPAOk(3);
P(38)= qPPek * (uk(3)/(KO2+uk(3))) * (zk(21)/(KPS+zk(21))) * (SALKk(3)/(KALK+SALKk(3)))
* ((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * ((KMAX-(zk(23)/XPAOk(3)))/(KPP+KMAX-
(zk(23)/XPAOk(3)))) * XPAOk(3);
P(39)= qPPek * (uk(3)/(KO2+uk(3))) * (zk(21)/(KPS+zk(21))) * (SALKk(3)/(KALK+SALKk(3)))
* ((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * ((KMAX-(zk(23)/XPAOk(3)))/(KPP+KMAX-
(zk(23)/XPAOk(3)))) * XPAOk(3) * etaNO3 * KO2/uk(3) * (zk(20)/(KNO3+zk(20)));
P(40)= muPAOek * (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) *
(SALKk(3)/(KALK+SALKk(3)))* (zk(21)/(KP+zk(21))) *
((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * XPAOk(3);
P(41)= muPAOek * (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) *
(SALKk(3)/(KALK+SALKk(3)))* (zk(21)/(KP+zk(21))) *
((zk(24)/XPAOk(3))/(KPHA+(zk(24)/XPAOk(3)))) * XPAOk(3) * etaNO3 * KO2/uk(3) *
(zk(20)/(KNO3+zk(20)));
P(42)= muAUTek * (uk(3)/(KO2+uk(3))) * (zk(19)/(KNH4+zk(19))) * (zk(21)/(KP+zk(21))) *
(SALKk(3)/(KALK+SALKk(3))) * XAk(3);

Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20) P(21) P(22) P(23) P(24) P(25) P(26) P(27) P(28) P(29)
P(30) P(31) P(32) P(33) P(34) P(35) P(36) P(37) P(38) P(39) P(40) P(41) P(42) ]';
```

APPENDIX H: DEVELOPED MATLAB CODE FOR OPTIMAL CONTROL OF THE ATHLONE PLANT WITH THE ACTIVATED SLUDGE MODEL NO.3 BIO-MODEL

H.1: MATLAB script file – AASM3_Simulation.m

```
% M-File: AASM3_Simulation.m
% ========================= M-FILE DESCRIPTION =========================
% This m-file is used for:
% Simulation of the Athlone plant using full ASM3 model
% Calculation of the parameters for the reduced ASM3 model
% Simulation of the Athlone plant using the reduced ASM3 model

clear all            % Clear the workspace
clc                  % Clear the command window


% ========================= PARAMETER DEFINITION =========================
% Declaration of the global variables
global kH kSTO etaNOX muH bH_O2 bH_NOX bSTO_O2 bSTO_NOX muA bA_O2 bA_NOX
global KX KO2 KS KNOX KNH4 KALK KSTO KA_O2 KA_NH4 KA_ALK KA_NOX

K1  = 1344;          % Prediction horizon or interval
DT = 0.001042;       % Sampling time

% Initialization of the model parameters, typical values at 20 0C
% Stochiometric parameters
fSI          = 0.0;
YSTO_O2      = 0.85;
YSTO_NOX     = 0.80;
YH_O2        = 0.63;
YH_NOX       = 0.54;
YA           = 0.24;
fI           =  0.2;
% Kinetic parameters
kH           = 3;
kSTO         = 5;
etaNOX       = 0.6;
muH          = 2;
bH_O2        = 0.2;
bH_NOX       = 0.1;
bSTO_O2      = 0.2;
bSTO_NOX     = 0.1;
muA          = 1;
bA_O2        = 0.15;
bA_NOX       = 0.05;
% Monod switching functions
KX           = 1;
KO2          = 0.2;
KS           = 2;
KNOX         = 0.5;
KNH4         = 0.01;
KALK         = 0.1;
KSTO         = 1;
KA_O2        = 0.5;
KA_NH4       = 1;
KA_ALK       = 0.5;
KA_NOX       = 0.5;


% Initialization of the Athlone process data
V1  = 1148;          % Tank 1 volume
V2  = 1148;          % Tank 2 volume
V3  = 5273;          % Tank 3 volume

Q0  = 40003;         % Influent flow rate
```

311

```
Qa    = 39916.8;        % Recycled flow rate from tank 3
Qp    = 39916.8;        % Recycled flow rate from tank 2
Qr    = 40003;          % Recycled flow rate from the settler
Qw    = 38400.0;        % Waste flow rate (3840)
Q1    = Qp + Q0;        % flow through the tank 1
Q2    = Qp + Q0 + Qa + Qr;   % flow through the tank 2
Q3    = Q2;                  % flow through the tank 3

% Coeficient for fraction of flow rates for the particulate materials
LAMBDA = (Qp+Q0+Qr-Qw)/(Qr);

% Oxygen parameters
SOsat  = 8;             % Oxygen saturation point
KLa    = 4.8;           % Oxygen transfer function in Tank 1

% ============================ PROCESS MODEL ============================
% Calculation of the MATRIX A
% Matrix A represent weighting factors of the accumulated state variables
% from different flows towards state variables in each tank as derived
% through mass balance principles, see chapter 3 of the Thesis
IV1 = DT/ V1; IV11 = IV1 * Q1; A11 = 1 - IV11;
IV2 = DT/ V2; IV22 = IV2 * Q2; A22 = 1 - IV22;
IV3 = DT/ V3; IV33 = IV3 * Q3; A33 = 1 - IV33;

A12 = IV1 * Qp;
A23a = IV2 * (Qa + Qr);
A23b = IV2 * (Qa + (LAMBDA*Qr));
A21 = IV2*Q1;
A32 = IV3*Q2;

A11 = A11 * ones(1,13); A11 = diag(A11);
A12 = A12 * ones(1,13); A12 = diag(A12);
A13 = zeros(13,13);

A21 = A21 * ones(1,13); A21 = diag(A21);
A22 = A22 * ones(1,13); A22 = diag(A22);
A23 = [A23a,A23a,A23b,A23b,A23b,A23b,A23b,A23a,A23a,A23a,A23a,A23b,A23a];
A23 = diag(A23);

A31 = zeros(13,13);
A32 = A32 * ones(1,13); A32 = diag(A32);
A33 = A33 * ones(1,13); A33 = diag(A33);

A = [A11 A12 A13; A21 A22 A23; A31 A32 A33];

% Calculation of the MATRIX B
% Matrix B represent weighting factors of the state variables towards the
% influent wastewater composition as derived through mass balance
% principles, see chapter 3 of the Thesis
IV0 = DT/V1*Q0;
B11 = IV0 * ones(1,13); B11 = diag(B11);

B = [B11; zeros(26,13)];

% Formulation of the MATRIX C
% Matrix C represent weighting factors of the state variables towards the
% processes on the ASM1 matrix table
CC=[[0      fSI 1   0.01    0    0        0.001  0  -1  0   0        0  -0.75];
    [-0.15  0   -1  0.03    0    0        0.002  0  0   0   YSTO_O2  0  0.51];
    [0      0   -1  0.03    0.07 -0.07    0.007  0  0   0   YSTO_NOX 0  0.48];
    [-0.60  0   0   -0.07   0    0        -0.005 0  0   1   -1/YH_O2 0  -0.06];
    [0      0   0   -0.07   0.30 -0.30    0.016  0  0   1   -1/YH_NOX 0 -0.21];
    [-0.80  0   0   0.066   0    0        0.005  fI 0   -1  0        0  -0.75 ];
    [0      0   0   0.066   0.28 -0.28    0.025  fI 0   -1  0        0  -0.75 ];
    [-1     0   0   0       0    0        0      0  0   0   -1       0  -0.60 ];
    [0      0   0   0       0.35 -0.35    0.025  0  0   0   -1       0  -0.60 ];
```

312

```
           [-18.04  0    0    -4.24   0     1/YA    -0.60  0  0  0  0    1  0.90  ];
           [-0.80   0    0     0.066  0     0        0.005 fI 0  0  0   -1 -0.75 ];
           [0       0    0     0.066  0.28 -0.28     0.025 fI 0  0  0   -1 -0.75 ]];
C = DT*[CC zeros(12,26) ; zeros(12,13) CC zeros(12,13) ; zeros(12,26) CC];


% ========================= SIMULATION DATA =========================
% Constant inflow concentrations for steady state simulations
SOi = 0.0; SIi = 30.0; SSi = 69.5; SNHi = 31.56; SNDi = 6.95;
SALKi = 7.0; SO2i = SOi; SNH4i = SNHi; SN2i = SNDi; SNOXi = 0.0;
XIi = 51.2; XSi = 202.32; XBHi = 28.17; XHi = XBHi; XSTOi = 0;
XAi = 0.0; XSSi = 125;

%SO2  SI   SS   SNH4 SN2   SNOX  SALK XI   XS   XH     XSTO XA
%XSS
xi = [SO2i*ones(1,K1); SIi*ones(1,K1); SSi*ones(1,K1); SNH4i*ones(1,K1);
      SN2i*ones(1,K1); SNOXi*ones(1,K1); SALKi*ones(1,K1); XIi*ones(1,K1);
      XSi*ones(1,K1); XHi*ones(1,K1); XSTOi*ones(1,K1); XAi*ones(1,K1);
      XSSi*ones(1,K1)];


% Values of the initial conditions
% For tank 1
SO2_1 = 4.2984433e-03; SI_1  = 3.0000000e+01; SS_1   = 2.8082131e+00;
SNH4_1 = 7.9178845e+00; SN2_1 = 1.2166405e+00; SNOX_1 = 5.3699400e+00;
SALK_1 = 4.9277103e+00; XI_1  = 1.1491252e+03; XS_1   = 8.2134908e+01;
XH_1   = 2.5517658e+03; XSTO_1 = 4.4885186e+00; XA_1   = 1.4838943e+02;
XSS_1 = 5.2848894e+00;
x10 = [SO2_1; SI_1; SS_1; SNH4_1; SN2_1; SNOX_1; SALK_1; XI_1; XS_1;
       XH_1; XSTO_1; XA_1;XSS_1 ];


% For tank 2
SO2_2 = 6.3131911e-05; SI_2  = 3.0000000e+01; SS_2   = 1.4587940e+00;
SNH4_2 = 8.3444148e+00; SN2_2 = 8.8206477e-01; SNOX_2 = 3.6619672e+00;
SALK_2 = 5.0801748e+00; XI_2  = 1.1491252e+03; XS_2   = 7.6386187e+01;
XH_2   = 2.5533851e+03; XSTO_2 = 4.4952273e+00; XA_2   = 1.4830914e+02;
XSS_2  = 5.0290873e+00;
x20 = [SO2_2; SI_2; SS_2; SNH4_2; SN2_2; SNOX_2; SALK_2; XI_2; XS_2;
       XH_2; XSTO_2; XA_2;XSS_2 ];


% For tank 3
SO2_3  = 1.7183778e+00; SI_3   = 3.0000000e+01; SS_3   = 1.1495418e+00;
SNH4_3 = 5.5479452e+00; SN2_3  = 8.2888682e-01; SNOX_3 = 6.5408820e+00;
SALK_3 = 4.6747902e+00; XI_3   = 1.1491252e+03; XS_3   = 6.4854922e+01;
XH_3   = 2.5571314e+03; XSTO_3 = 4.3924277e+00; XA_3   = 1.4894126e+02;
XSS_3  = 4.5041834e+00;
x30 = [SO2_3; SI_3; SS_3; SNH4_3; SN2_3; SNOX_3; SALK_3; XI_3; XS_3;
       XH_3; XSTO_3; XA_3;XSS_3 ];


x0 = [x10;x20;x30];
x  = [x0 zeros(39,K1)];

% Steady state simulation of the model
for k = 1:K1
    xk  = x(:,k);
    xik = xi(:,k);
    Pk  = P_AASM3F(xk);

    SOk  = [xk(27,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xxlk = [zeros(26,1);xxk(1);zeros(12,1)];
    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xxlk;
end

% Full model response to the steady state influent concentrations
% Tank 1
figure(1)
subplot(5,3,1);plot(x(1,:));title('SO2_1')
```

```
subplot(5,3,2);plot(x(2,:));title('SI1')
subplot(5,3,3);plot(x(3,:));title('SS1')
subplot(5,3,4);plot(x(4,:));title('SNH4_1')
subplot(5,3,5);plot(x(5,:));title('SN2_1')
subplot(5,3,6);plot(x(6,:));title('SNOX1')
subplot(5,3,7);plot(x(7,:));title('SALK1')
subplot(5,3,8);plot(x(8,:));title('XI1')
subplot(5,3,9);plot(x(9,:));title('XS1')
subplot(5,3,10);plot(x(10,:));title('XH1')
subplot(5,3,11);plot(x(11,:));title('XSTO1')
xlabel('discrete time k')
subplot(5,3,12);plot(x(12,:));title('XA1')
xlabel('discrete time k')
subplot(5,3,13);plot(x(13,:));title('XSS1')
xlabel('discrete time k')
set(figure(1),'name','Tank 1 steady state results','numbertitle','off')

% Tank 2
figure(2)
subplot(5,3,1);plot(x(14,:));title('SO2_2')
subplot(5,3,2);plot(x(15,:));title('SI2')
subplot(5,3,3);plot(x(16,:));title('SS2')
subplot(5,3,4);plot(x(17,:));title('SNH4_2')
subplot(5,3,5);plot(x(18,:));title('SN2_2')
subplot(5,3,6);plot(x(19,:));title('SNOX2')
subplot(5,3,7);plot(x(20,:));title('SALK2')
subplot(5,3,8);plot(x(21,:));title('XI2')
subplot(5,3,9);plot(x(22,:));title('XS2')
subplot(5,3,10);plot(x(23,:));title('XH2')
subplot(5,3,11);plot(x(24,:));title('XSTO2')
xlabel('discrete time k')
subplot(5,3,12);plot(x(25,:));title('XA2')
xlabel('discrete time k')
subplot(5,3,13);plot(x(26,:));title('XSS2')
xlabel('discrete time k')
set(figure(2),'name','Tank 2 steady state results','numbertitle','off')

% Tank 3
figure(3)
subplot(5,3,1);plot(x(27,:));title('SO2_3')
subplot(5,3,2);plot(x(28,:));title('SI3')
subplot(5,3,3);plot(x(29,:));title('SS3')
subplot(5,3,4);plot(x(30,:));title('SNH4_3')
subplot(5,3,5);plot(x(31,:));title('SN2_3')
subplot(5,3,6);plot(x(32,:));title('SNOX3')
subplot(5,3,7);plot(x(33,:));title('SALK3')
subplot(5,3,8);plot(x(34,:));title('XI3')
subplot(5,3,9);plot(x(35,:));title('XS3')
subplot(5,3,10);plot(x(36,:));title('XH3')
subplot(5,3,11);plot(x(37,:));title('XSTO3')
xlabel('discrete time k')
subplot(5,3,12);plot(x(38,:));title('XA3')
xlabel('discrete time k')
subplot(5,3,13);plot(x(39,:));title('XSS3')
xlabel('discrete time k')
set(figure(3),'name','Tank 3 steady state results','numbertitle','off')

% Data for dynamic simulations
K    = 1344;    % Prediction horizon for dynamic simulations

% Dynamic of the inflow concentrations for dry weather file
load dry.txt

% The order is t, SS XBH XS XI SNH SI SND XND Q0
%               1, 2, 3,  4, 5, 6,  7, 8,  9,  10
```

314

```
xi = [SO2i*zeros(1,K); dry(1:K,7)';      dry(1:K,2)';      dry(1:K,6)';
      SN2i*ones(1,K); SNOXi*zeros(1,K); SALKi* ones(1,K); dry(1:K,5)';
      dry(1:K,4)';     dry(1:K,3)';     XSTOi*ones(1,K);  XAi*ones(1,K);
      XSSi*ones(1,K)];

x = [x(:,K1+1) zeros(39,K)];

% Dynamic simulation of the model
for k    = 1:K
    xk   = x(:,k);
    xik  = xi(:,k);
    Pk   = P_AASM3F(xk); % Calculation of the process rates

    SOk  = [xk(27,1)];
    xxk  = DT*KLa*(SOsat-SOk);
    xx1k = [zeros(26,1);xxk(1);zeros(12,1)];
    x(:,k+1) = A*x(:,k) + C'*Pk + B*xi(:,k) + xx1k;
end

% Full model response to the dynamic influent concentrations
% Tank 1
figure(4)
subplot(5,3,1);plot(x(1,:));title('SO2_1')
subplot(5,3,2);plot(x(2,:));title('SI1')
subplot(5,3,3);plot(x(3,:));title('SS1')
subplot(5,3,4);plot(x(4,:));title('SNH4_1')
subplot(5,3,5);plot(x(5,:));title('SN2_1')
subplot(5,3,6);plot(x(6,:));title('SNOX1')
subplot(5,3,7);plot(x(7,:));title('SALK1')
subplot(5,3,8);plot(x(8,:));title('XI1')
subplot(5,3,9);plot(x(9,:));title('XS1')
subplot(5,3,10);plot(x(10,:));title('XH1')
subplot(5,3,11);plot(x(11,:));title('XSTO1')
xlabel('discrete time k')
subplot(5,3,12);plot(x(12,:));title('XA1')
xlabel('discrete time k')
subplot(5,3,13);plot(x(13,:));title('XSS1')
xlabel('discrete time k')
set(figure(4),'name','Tank 1 dynamic results','numbertitle','off')

% Tank 2
figure(5)
subplot(5,3,1);plot(x(14,:));title('SO2_2')
subplot(5,3,2);plot(x(15,:));title('SI2')
subplot(5,3,3);plot(x(16,:));title('SS2')
subplot(5,3,4);plot(x(17,:));title('SNH4_2')
subplot(5,3,5);plot(x(18,:));title('SN2_2')
subplot(5,3,6);plot(x(19,:));title('SNOX2')
subplot(5,3,7);plot(x(20,:));title('SALK2')
subplot(5,3,8);plot(x(21,:));title('XI2')
subplot(5,3,9);plot(x(22,:));title('XS2')
subplot(5,3,10);plot(x(23,:));title('XH2')
subplot(5,3,11);plot(x(24,:));title('XSTO2')
xlabel('discrete time k')
subplot(5,3,12);plot(x(25,:));title('XA2')
xlabel('discrete time k')
subplot(5,3,13);plot(x(26,:));title('XSS2')
xlabel('discrete time k')
set(figure(5),'name','Tank 2 dynamic results','numbertitle','off')

% Tank 3
figure(6)
subplot(5,3,1);plot(x(27,:));title('SO2_3')
subplot(5,3,2);plot(x(28,:));title('SI3')
subplot(5,3,3);plot(x(29,:));title('SS3')
subplot(5,3,4);plot(x(30,:));title('SNH4_3')
```

315

```
subplot(5,3,5);plot(x(31,:));title('SN2_3')
subplot(5,3,6);plot(x(32,:));title('SNOX3')
subplot(5,3,7);plot(x(33,:));title('SALK3')
subplot(5,3,8);plot(x(34,:));title('XI3')
subplot(5,3,9);plot(x(35,:));title('XS3')
subplot(5,3,10);plot(x(36,:));title('XH3')
subplot(5,3,11);plot(x(37,:));title('XSTO3')
xlabel('discrete time k')
subplot(5,3,12);plot(x(38,:));title('XA3')
xlabel('discrete time k')
subplot(5,3,13);plot(x(39,:));title('XSS3')
xlabel('discrete time k')
set(figure(6),'name','Tank 3 dynamic results','numbertitle','off')

% ======================= REDUCED MODEL FORMULATION =======================
% Form the vector of the states (z) for the reduced model, based on the
% ASM1 prediction
%      SS,       SNH4,       SNOX
z1 = [x(3,:);   x(4,:);   x(6,:) ];    %1      2      3
z2 = [x(16,:);  x(17,:);  x(19,:)];    %4      5      6
z3 = [x(29,:);  x(30,:);  x(32,:)];    %7      8      9
z  = [z1;z2;z3];

% Formation of the vector of the control variables
u1 = x(1,:);
u2 = x(14,:);
u3 = x(27,:);
u  = [u1; u2; u3];

% Dynamic values of the inflow concentrations for dry weather
%        SS              SNH4                SNOX
zi  = [SSi*ones(1,K);  SNH4i*ones(1,K);   SNOXi*ones(1,K)];

% Steady state values for the biomass XH, XA, XS, SALK, XSTO
XH     = [2551.76*ones(1,K);2553.38*ones(1,K);2557.13*ones(1,K);
          2559.18*ones(1,K);2559.34*ones(1,K)];
XA     = [148.389*ones(1,K);148.309*ones(1,K);148.941*ones(1,K);
          149.527*ones(1,K);149.797*ones(1,K)];
XS     = [82.135*ones(1,K);76.386*ones(1,K);64.855*ones(1,K);
          55.694*ones(1,K);49.306*ones(1,K)];
XSTO   = [448.9*ones(1,K);76.386*ones(1,K);64.855*ones(1,K);
          55.694*ones(1,K);49.306*ones(1,K)];
SALK   = [4.9*ones(1,K);148.309*ones(1,K);148.941*ones(1,K);
          149.527*ones(1,K);149.797*ones(1,K)];

% MATRIX A for the reduced model (AR)
A11r = 1 - IV11; A22r = 1 - IV22; A33r = 1 - IV33;

A12r = IV1 * Qp; A23r = IV2 * (Qa + Qr);
A21r = IV2*Q1; A32r = IV3*Q2;

A11r = A11r * ones(1,3); A11r = diag(A11r);
A12r = A12r * ones(1,3); A12r = diag(A12r);
A13r = zeros(3,3);

A21r = A21r * ones(1,3); A21r = diag(A21r);
A22r = A22r * ones(1,3); A22r = diag(A22r);
A23r = [A23r,A23r,A23r];
A23r = diag(A23r);

A31r = zeros(3,3);
A32r = A32r * ones(1,3); A32r = diag(A32r);
A33r = A33r * ones(1,3); A33r = diag(A33r);

AR = [A11r A12r A13r; A21r A22r A23r; A31r A32r A33r];
```

316

```
% MATRIX B for the reduced model(BR)
IV0 = DT / V1*Q0;
f = 1.2;
B11r = [IV0 IV0 IV0*f];
B11r = diag(B11r);
BR = [B11r; zeros(6,3)];

% MATRIX C for the reduced model (CR)
CCR=[[1  0.01    0    ];
     [0  -0.07   0    ];
     [0  -0.07  -0.30];
     [0  -4.24  1/YA]];

CR = DT*[CCR zeros(4,6); zeros(4,3) CCR zeros(4,3); zeros(4,6) CCR];

% Parameter estimation
for k = 1:K
    zk = z(:,k);
    uk = u(:,k);

    XHk   = XH(:,k);
    XAk   = XA(:,k);
    XSk   = XS(:,k);
    XSTOk = XSTO(:,k);
    SALKk = SALK(:,k);

    zik   = zi(:,k);
    Pk    = P_AASM3R(zk,uk,XHk,XAk,XSk,XSTOk,SALKk);
    FIk   = CR'*Pk;
    Y(:,k) = z(:,k+1)-AR*z(:,k)-BR*zi(:,k);
    theta(:,k)=(FIk'*FIk)^(-1)*FIk'*Y(:,k);
end

muHE(1,:) = theta(1,:);
muAE(1,:) = theta(2,:);
KhE(1,:)  = theta(3,:);

% Dynamic simulation of the reduced model with the estimated parameters
zE = [x(3,1);x(4,1);x(6,1);x(16,1);x(17,1);x(19,1);x(29,1);x(30,1);x(32,1)];
ERR = z(:,1)-zE;

% Dynamic simulation of the reduced model with the estimated parameters
for k = 1:K
    zk  = z(:,k);
    uk  = u(:,k);
    zik = zi(:,k);

    XHk = XH(:,k);
    XAk = XA(:,k);
    XSk = XS(:,k);
    XSTOk = XSTO(:,k);
    SALKk = SALK(:,k);

    muHEk = muHE(:,k);
    muAEk = muAE(:,k);
    KhEk  = KhE(:,k);

    Pk = P_AASM3RE(zk,uk,XHk,XAk,XSk,XSTOk,SALKk,muHEk,muAEk,KhEk);
    zE(:,k+1) = AR*zE(:,k) + CR'*Pk + BR*zi(:,k);
    ERR(:,k+1)=z(:,k+1)-zE(:,k+1);
end

% Vector of process full, estimated and error variables SS, SNH4, SNOX, Tank 1
SS1=z(1,:);SNH4_1=z(2,:);SNOX1=z(3,:);
SS1E=zE(1,:);SNH4_1E=zE(2,:);SNOX1E=zE(3,:);
ERRSS1=ERR(1,:);ERRSNH4_1=ERR(2,:);ERRSNOX1=ERR(3,:);
```

```
% Calculation of the weighted errors for tank1
eSS1=(ERRSS1*ERRSS1')/(SS1*SS1');
eSNH4_1=(ERRSNH4_1*ERRSNH4_1')/(SNH4_1*SNH4_1');
eSNOX1=(ERRSNOX1*ERRSNOX1')/(SNOX1*SNOX1');


% Vector of process full, estimated and error variables SS, SNH4,SNOX, Tank 2
SS2=z(4,:);SNH4_2=z(5,:);SNOX2=z(6,:);
SS2E=zE(4,:);SNH4_2E=zE(5,:);SNOX2E=zE(6,:);
ERRSS2=ERR(4,:);ERRSNH4_2=ERR(5,:);ERRSNOX2=ERR(6,:);
% Calculation of the weighted errors for tank 2
eSS2=(ERRSS2*ERRSS2')/(SS2*SS2');
eSNH4_2=(ERRSNH4_2*ERRSNH4_2')/(SNH4_2*SNH4_2');
eSNOX2=(ERRSNOX2*ERRSNOX2')/(SNOX2*SNOX2');


% Vector of process full, estimated and error variables SS, SNH4,SNOX, Tank 3
SS3=z(7,:);SNH4_3=z(8,:);SNOX3=z(9,:);
SS3E=zE(7,:);SNH4_3E=zE(8,:);SNOX3E=zE(9,:);
ERRSS3=ERR(7,:);ERRSNH4_3=ERR(8,:);ERRSNOX3=ERR(9,:);
% Calculation of the weighted errors for tank 3
eSS3=(ERRSS3*ERRSS3')/(SS3*SS3');
eSNH4_3=(ERRSNH4_3*ERRSNH4_3')/(SNH4_3*SNH4_3');
eSNOX3=(ERRSNOX3*ERRSNOX3')/(SNOX3*SNOX3');


k = 1:K;
% Graphs of the estimated parameters
figure(7)
subplot(3,1,1);plot(k,muHE,'k');title('Estimated muH');ylabel('muHE');
subplot(3,1,2);plot(k,muAE,'k');title('Estimated muA');ylabel('muAE');
subplot(3,1,3);plot(k,KhE,'k');title('Estimated Kh');ylabel('KhE');
set(figure(7),'name','Estimated parameters','numbertitle','off')
xlabel('discrete time k')


k = 1:K+1;
% Graphs of the reduced model response to the dynamic influent
% concentrations

figure(8)
subplot(3,2,1);plot(k,SS1);hold on; plot(k,SS1E,'k-.');ylabel('SS1,SS1E');
legend('SS1','SS1E');subplot(3,2,2); plot(k,ERRSS1,'k');ylabel('ERRSS1');

subplot(3,2,3);plot(k,SNH4_1);hold on; plot(k,SNH4_1E,'k-.');
legend('SNH4_1','SNH4_1E');ylabel('SNH4_1,SNH4_1E');
subplot(3,2,4); plot(k,ERRSNH4_1,'k');ylabel('ERRSNH4_1');

subplot(3,2,5);plot(k,SNOX1);hold on;plot(k,SNOX1E,'k-.');
legend('SNOX1','SNOX1E');ylabel('SNOX1,SNOX1E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNOX1,'k');ylabel('ERRSNOX1');
xlabel('discrete time k')
set(figure(8),'name','Tank 1 Process variables and estimation
error','numbertitle','off')


figure(9)
subplot(3,2,1);plot(k,SS2);hold on; plot(k,SS2E,'k-.');ylabel('SS2,SS2E');
legend('SS2','SS2E');subplot(3,2,2); plot(k,ERRSS2,'k');ylabel('ERRSS2');

subplot(3,2,3);plot(k,SNH4_2);hold on;plot(k,SNH4_2E,'k-.');
legend('SNH4_2','SNH4_2E');ylabel('SNH4_2,SNH4_2E');
subplot(3,2,4); plot(k,ERRSNH4_2,'k');ylabel('ERRSNH4_2');

subplot(3,2,5);plot(k,SNOX2);hold on; plot(k,SNOX2E,'k-.')
legend('SNOX2','SNOX2E');ylabel('SNOX2,SNOX2E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNOX2,'k');ylabel('ERRSNOX2');
xlabel('discrete time k')
set(figure(9),'name','Tank 2 Process variables and estimation
error','numbertitle','off')
```

```
figure(10)
subplot(3,2,1);plot(k,SS3);hold on; plot(k,SS3E,'k-.');ylabel('SS3,SS3E');
legend('SS3','SS3E');subplot(3,2,2); plot(k,ERRSS3,'k');ylabel('ERRSS3');

subplot(3,2,3);plot(k,SNH4_3);hold on;plot(k,SNH4_3E,'k-.');
legend('SNH4_3','SNH4_3E');ylabel('SNH4_3,SNH4_3E');subplot(3,2,4);
plot(k,ERRSNH4_3,'k');
ylabel('ERRSNH4_3');

subplot(3,2,5);plot(k,SNOX3);hold on; plot(k,SNOX3E,'k-.')
legend('SNOX3','SNOX3E');ylabel('SNOX3,SNOX3E');xlabel('discrete time k')
subplot(3,2,6); plot(k,ERRSNOX3,'k');
ylabel('ERRSNOX3');xlabel('discrete time k')
set(figure(10),'name','Tank 3 Process variables and estimation
error','numbertitle','off')
```

H.2: MATLAB script file – P_AASM3F.m

```
% M-File: P_AASM3F.m
% ========================= M-FILE DESCRIPTION ===========================
% This m-file is used for:
% Calculation of the process rates for the full model

function Pk  = P_AASM3F(xk)

global kH kSTO etaNOX muH bH_O2 bH_NOX bSTO_O2 bSTO_NOX muA bA_O2 bA_NOX KX KO2 KS KNOX
KNH4 KALK KSTO KA_O2 KA_NH4 KA_ALK KA_NOX;
% TANK 1
P(1)= kH*((xk(9)/xk(10))/(KX+(xk(9)/xk(10))))* xk(10);
P(2)= kSTO * (xk(1)/(KO2+xk(1))) * (xk(3)/(KS+xk(3)))* xk(10);
P(3)= kSTO * etaNOX * (KO2/(KO2+xk(1))) * (xk(6)/(KNOX+xk(6))) * (xk(3)/(KS+xk(3))) *
xk(10);
P(4)= muH * (xk(1)/(KO2+xk(1))) * (xk(4)/(KNH4+xk(4))) * (xk(7)/(KALK+xk(7))) *
((xk(11)/xk(10))/(KSTO+(xk(11)/xk(10)))) * xk(10);
P(5)= muH * etaNOX * (KO2/(KO2+xk(1))) * (xk(6)/(KNOX+xk(6))) * (xk(4)/(KNH4+xk(4))) *
(xk(7)/(KALK+xk(7))) * ((xk(11)/xk(10))/(KSTO+(xk(11)/xk(10)))) * xk(10);
P(6)= bH_O2 * (xk(1)/(KO2+xk(1))) * xk(10);
P(7)= bH_NOX * (KO2/(KO2+xk(1))) * (xk(6)/(KNOX+xk(6))) * xk(10);
P(8)= bSTO_O2 * (xk(1)/(KO2+xk(1))) * xk(11);
P(9)= bSTO_NOX * (KO2/(KO2+xk(1))) * (xk(6)/(KNOX+xk(6))) * xk(11);
P(10)= muA * (xk(1)/(KA_O2+xk(1))) * (xk(4)/(KA_NH4+xk(4))) * (xk(7)/(KA_ALK+xk(7))) *
xk(12);
P(11)= bA_O2 * (xk(1)/(KA_O2+xk(1))) * xk(12);
P(12)= bA_NOX * (KA_O2/(KA_O2+xk(1))) * (xk(6)/(KA_NOX+xk(6))) * xk(12);


% TANK 2
P(13)= kH*((xk(22)/xk(23))/(KX+(xk(22)/xk(23))))* xk(23);
P(14)= kSTO * (xk(14)/(KO2+xk(14))) * (xk(16)/(KS+xk(16)))* xk(23);
P(15)= kSTO * etaNOX * (KO2/(KO2+xk(14))) * (xk(19)/(KNOX+xk(19))) *
(xk(16)/(KS+xk(16))) * xk(23);
P(16)= muH * (xk(14)/(KO2+xk(14))) * (xk(17)/(KNH4+xk(17))) * (xk(20)/(KALK+xk(20))) *
((xk(24)/xk(23))/(KSTO+(xk(24)/xk(23)))) * xk(23);
P(17)= muH * etaNOX * (KO2/(KO2+xk(14))) * (xk(19)/(KNOX+xk(19))) *
(xk(17)/(KNH4+xk(17))) * (xk(20)/(KALK+xk(20))) *
((xk(24)/xk(23))/(KSTO+(xk(24)/xk(23)))) * xk(23);
P(18)= bH_O2 * (xk(14)/(KO2+xk(14))) * xk(23);
P(19)= bH_NOX * (KO2/(KO2+xk(14))) * (xk(19)/(KNOX+xk(19))) * xk(23);
P(20)= bSTO_O2 * (xk(14)/(KO2+xk(14))) * xk(24);
P(21)= bSTO_NOX * (KO2/(KO2+xk(14))) * (xk(19)/(KNOX+xk(19))) * xk(24);
P(22)= muA * (xk(14)/(KA_O2+xk(14))) * (xk(17)/(KA_NH4+xk(17))) *
(xk(20)/(KA_ALK+xk(20))) * xk(25);
P(23)= bA_O2 * (xk(14)/(KA_O2+xk(14))) * xk(25);
P(24)= bA_NOX * (KA_O2/(KA_O2+xk(14))) * (xk(19)/(KA_NOX+xk(19))) * xk(25);
```

```
% TANK 3
P(25)= kH*((xk(35)/xk(36))/(KX+(xk(35)/xk(36))))* xk(36);
P(26)= kSTO * (xk(27)/(KO2+xk(27))) * (xk(29)/(KS+xk(29)))* xk(36);
P(27)= kSTO * etaNOX * (KO2/(KO2+xk(27))) * (xk(32)/(KNOX+xk(32))) *
(xk(29)/(KS+xk(29))) * xk(36);
P(28)= muH * (xk(27)/(KO2+xk(27))) * (xk(30)/(KNH4+xk(30))) * (xk(33)/(KALK+xk(33))) *
((xk(37)/xk(36))/(KSTO+(xk(37)/xk(36)))) * xk(36);
P(29)= muH * etaNOX * (KO2/(KO2+xk(27))) * (xk(32)/(KNOX+xk(32))) *
(xk(30)/(KNH4+xk(30))) * (xk(33)/(KALK+xk(33))) *
((xk(37)/xk(36))/(KSTO+(xk(37)/xk(36)))) * xk(36);
P(30)= bH_O2 * (xk(27)/(KO2+xk(27))) * xk(36);
P(31)= bH_NOX * (KO2/(KO2+xk(27))) * (xk(32)/(KNOX+xk(32))) * xk(36);
P(32)= bSTO_O2 * (xk(27)/(KO2+xk(27))) * xk(37);
P(33)= bSTO_NOX * (KO2/(KO2+xk(27))) * (xk(32)/(KNOX+xk(32))) * xk(37);
P(34)= muA * (xk(27)/(KA_O2+xk(27))) * (xk(30)/(KA_NH4+xk(30))) *
(xk(33)/(KA_ALK+xk(33))) * xk(38);
P(35)= bA_O2 * (xk(27)/(KA_O2+xk(27))) * xk(38);
P(36)= bA_NOX * (KA_O2/(KA_O2+xk(27))) * (xk(32)/(KA_NOX+xk(32))) * xk(38);

Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12) P(13) P(14) P(15)
P(16) P(17) P(18) P(19) P(20) P(21) P(22) P(23) P(24) P(25) P(26) P(27) P(28) P(29)
P(30) P(31) P(32) P(33) P(34) P(35) P(36)]';
```

H.3: MATLAB script file – P_AASM3R.m

```
% M-File: P_AASM3R.m
% ========================== M-FILE DESCRIPTION ==========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model

function Pk = P_AASM1R(zk,uk,XHk,XAk,XSk,XSTOk,SALKk);

global KX KO2 KNH4 KALK KSTO etaNOX KNOX KA_O2 KA_NH4 KA_ALK ;

% TANK 1
P(1)= ((XSk(1)/XHk(1))/(KX+(XSk(1)/XHk(1))))* XHk(1);
P(2)= (uk(1)/(KO2+uk(1))) * (zk(2)/(KNH4+zk(2))) * (SALKk(1)/(KALK+SALKk(1))) *
((XSTOk(1)/XHk(1))/(KSTO+(XSTOk(1)/XHk(1)))) * XHk(1);
P(3)= etaNOX * (KO2/(KO2+uk(1))) * (zk(3)/(KNOX+zk(3))) * (zk(2)/(KNH4+zk(2))) *
(SALKk(1)/(KALK+SALKk(1))) * ((XSTOk(1)/XHk(1))/(KSTO+(XSTOk(1)/XHk(1)))) * XHk(1);
P(4)= (uk(1)/(KA_O2+uk(1))) * (zk(2)/(KA_NH4+zk(2))) * (SALKk(1)/(KA_ALK+SALKk(1))) *
XAk(1);
P1k = [P(1) 0 0; 0 P(2) 0; 0 P(3) 0; 0 0 P(4)];
% TANK 2
P(5)= ((XSk(2)/XHk(2))/(KX+(XSk(2)/XHk(2))))* XHk(2);
P(6)= (uk(2)/(KO2+uk(2))) * (zk(5)/(KNH4+zk(5))) * (SALKk(2)/(KALK+SALKk(2))) *
((XSTOk(2)/XHk(2))/(KSTO+(XSTOk(2)/XHk(2)))) * XHk(2);
P(7)= etaNOX * (KO2/(KO2+uk(2))) * (zk(6)/(KNOX+zk(6))) * (zk(5)/(KNH4+zk(5))) *
(SALKk(2)/(KALK+SALKk(2))) * ((XSTOk(2)/XHk(2))/(KSTO+(XSTOk(2)/XHk(2)))) * XHk(2);
P(8)= (uk(2)/(KA_O2+uk(2))) * (zk(5)/(KA_NH4+zk(5))) * (SALKk(2)/(KA_ALK+SALKk(2))) *
XAk(2);
P2k = [P(5) 0 0; 0 P(6) 0; 0 P(7) 0; 0 0 P(8)];
% TANK 3
P(9) = ((XSk(3)/XHk(3))/(KX+(XSk(3)/XHk(3))))* XHk(3);
P(10)= (uk(3)/(KO2+uk(3))) * (zk(8)/(KNH4+zk(8))) * (SALKk(3)/(KALK+SALKk(3))) *
((XSTOk(3)/XHk(3))/(KSTO+(XSTOk(3)/XHk(3)))) * XHk(3);
P(11)= etaNOX * (KO2/(KO2+uk(3))) * (zk(9)/(KNOX+zk(9))) * (zk(8)/(KNH4+zk(8))) *
(SALKk(3)/(KALK+SALKk(3))) * ((XSTOk(3)/XHk(3))/(KSTO+(XSTOk(3)/XHk(3)))) * XHk(3);
P(12)= (uk(3)/(KA_O2+uk(3))) * (zk(8)/(KA_NH4+zk(8))) * (SALKk(3)/(KA_ALK+SALKk(3))) *
XAk(3);
P3k = [P(9) 0 0; 0 P(10) 0; 0 P(11) 0; 0 0 P(12)];

%Matrix Pk for the whole process
```

```
Pk = [Plk; P2k; P3k];
```

H.4: MATLAB script file – P_AASM3RE.m

```
% M-File: P_AASM3RE.m
% ============================ M-FILE DESCRIPTION ========================
% This m-file is used for:
% Calculation of the process rates for parameter estimation of the reduced
% model

% Description of the function function
function Pk = P_AASM3RE(zk,uk,XHk,XAk,XSk,XSTOk,SALKk,muHEk,muAEk,KhEk);

global KX KO2 KNH4 KALK KSTO etaNOX KNOX KA_O2 KA_NH4 KA_ALK ;
% TANK 1
P(1)= KhEk * ((XSk(1)/XHk(1))/(KX+(XSk(1)/XHk(1))))* XHk(1);
P(2)= muHEk * (uk(1)/(KO2+uk(1))) * (zk(2)/(KNH4+zk(2))) * (SALKk(1)/(KALK+SALKk(1))) *
((XSTOk(1)/XHk(1))/(KSTO+(XSTOk(1)/XHk(1)))) * XHk(1);
P(3)= muHEk * etaNOX * (KO2/(KO2+uk(1))) * (zk(3)/(KNOX+zk(3))) * (zk(2)/(KNH4+zk(2)))
* (SALKk(1)/(KALK+SALKk(1))) * ((XSTOk(1)/XHk(1))/(KSTO+(XSTOk(1)/XHk(1)))) * XHk(1);
P(4)= muAEk * (uk(1)/(KA_O2+uk(1))) * (zk(2)/(KA_NH4+zk(2))) *
(SALKk(1)/(KA_ALK+SALKk(1))) * XAk(1);

% TANK 2
P(5)= KhEk * ((XSk(2)/XHk(2))/(KX+(XSk(2)/XHk(2))))* XHk(2);
P(6)= muHEk * (uk(2)/(KO2+uk(2))) * (zk(5)/(KNH4+zk(5))) * (SALKk(2)/(KALK+SALKk(2))) *
((XSTOk(2)/XHk(2))/(KSTO+(XSTOk(2)/XHk(2)))) * XHk(2);
P(7)= muHEk * etaNOX * (KO2/(KO2+uk(1))) * (zk(6)/(KNOX+zk(6))) * (zk(5)/(KNH4+zk(5)))
* (SALKk(2)/(KALK+SALKk(2))) * ((XSTOk(2)/XHk(2))/(KSTO+(XSTOk(2)/XHk(2)))) * XHk(2);
P(8)= muAEk * (uk(2)/(KA_O2+uk(2))) * (zk(5)/(KA_NH4+zk(5))) *
(SALKk(2)/(KA_ALK+SALKk(2))) * XAk(2);

% TANK 3
P(9) = KhEk * ((XSk(3)/XHk(3))/(KX+(XSk(3)/XHk(3))))* XHk(3);
P(10)= muHEk * (uk(3)/(KO2+uk(3))) * (zk(8)/(KNH4+zk(8))) * (SALKk(3)/(KALK+SALKk(3)))
* ((XSTOk(3)/XHk(3))/(KSTO+(XSTOk(3)/XHk(3)))) * XHk(3);
P(11)= muHEk * etaNOX * (KO2/(KO2+uk(3))) * (zk(9)/(KNOX+zk(9))) * (zk(8)/(KNH4+zk(8)))
* (SALKk(3)/(KALK+SALKk(3))) * ((XSTOk(3)/XHk(3))/(KSTO+(XSTOk(3)/XHk(3)))) * XHk(3);
P(12)= muAEk * (uk(3)/(KA_O2+uk(3))) * (zk(8)/(KA_NH4+zk(8))) *
(SALKk(3)/(KA_ALK+SALKk(3))) * XAk(3);

Pk = [P(1) P(2) P(3) P(4) P(5) P(6) P(7) P(8) P(9) P(10) P(11) P(12)]';
```