Benchmark Simulation Model no. 2 (BSM2)


Summary

This document presents in details the final state of Benchmark Simulation Model no. 2 (BSM2). The model equations to be implemented for the proposed layout, the procedure to test the implementation and the performance criteria to be used are described, as well as the sensors and control handles. Further explanations concerning the reasoning which ended up with the choices made can be found in the Corresponding Technical Reports. Furthermore supplementary informations are given in the documents accompanying the software.

Copyright © 2018 Jens Alex, Lorenzo Benedetti, John Copp, Krist V. Gernaey, Ulf Jeppsson, Ingmar Nopens, Marie-Noëlle Pons, Jean-Philippe Steyer and Peter Vanrolleghem
Table of Contents

1. INTRODUCTION 4

2. MODELING OF THE ACTIVATED SLUDGE SECTION 6
  2.1. General characteristics 6
  2.2. Bioprocess model 6
    2.2.1. List of variables 6
    2.2.2. List of processes 7
    2.2.3. Observed conversion rates 7
    2.2.4. Biological parameter values 8
  2.3. Detailed activated sludge section layout 9
    2.3.1. Bioreactor (general characteristics) 9
    2.3.2. Reactor mass balances (general formula) 9
    2.3.3. Secondary clarifier 10
    2.3.4. Plant effluent composition 13
  2.4. Oxygen transfer coefficient 13
  2.5. Oxygen concentration at saturation 13

3. MODELING OF THE ANAEROBIC DIGESTER 13
  3.1. Introduction 13
    3.1.1. Acid-base equations 13
    3.1.2. Temperature dependencies 14
  3.2. Model equations 14
    3.2.1. Process rates 14
    3.2.2. Process inhibition 15
    3.2.3. Liquid phase equations 16
    3.2.4. Gas phase equations 19
  3.3. ADM1 DAE implementation 20
    3.3.1. ODE and DAE systems 21
    3.3.2. Time constants in ADM1 21
    3.3.3. pH and $S_{H_2}$ solvers 21
  3.4. ADM1 benchmark model parameters 22

4. MODELING OF THE PRIMARY CLARIFIER 22

5. MODELING OF THE THICKENER AND DEWATERING UNIT 27
  5.1. Thickener 27
  5.2. Dewatering unit 27

6. MODELING OF THE REJECT WATER STORAGE TANK 28
  6.1. General definitions 28
  6.2. Storage tank behaviour 29
    6.2.1. Variation of the liquid volume 29
    6.2.2. Variation of a concentration 30
  6.3. Implementation 30

7. ASM/ADM AND ADM/ASM INTERFACES 30

8. INFLOW DATA 32

9. INITIALIZATION 33

10. EVALUATION 33

11. SET-UP OF A DEFAULT CONTROLLER 34
  11.1. Controller variables 34
11.2. Controller type 34
12. PERFORMANCE ASSESSMENT 35
13. SENSORS AND CONTROL HANDLES 38
13.1. Introduction 38
13.2. Sensors 38
13.3. Sensor model description 39
13.3.1. Continuously measuring sensors 40
13.3.2. Discontinuously measuring sensors 42
13.3.3. Conclusions 43
13.4. Control handles 43
13.5. Alternative description 44
13.5.1. Model for sensor class A and actuator model 44
13.5.2. Model for sensor class B0 and C0 45
13.5.3. Model for sensor class B1 and C1 45
13.5.4. Model for sensor D 46
14. CONCLUDING REMARKS 46
15. REFERENCES 46

APPENDICES
Appendix 1: Practical BSM1 plant layout 47
Appendix 2: Source codes of ASM/ADM/ASM interfaces for BSM2 48
A2.1. MATLAB code 48
A2.1. FORTRAN code 64
A2.2.1. ASM/ADM 64
A2.2.2. ADM/ASM 72
Appendix 3: Steady-state results 78
Appendix 4: Open-loop results 89
Appendix 5: Closed-loop results with ideal sensors and actuators 92
Appendix 6: Closed-loop results with realistic sensors and actuators 95
1. INTRODUCTION

Wastewater treatment plants (WWTPs) are large non-linear systems subject to large perturbations in influent flow rate and pollutant load, together with uncertainties concerning the composition of the incoming wastewater. Nevertheless these plants have to be operated continuously, meeting stricter and stricter regulations.

Many control strategies have been proposed in the literature but their evaluation and comparison, either practical or based on simulation, is difficult. This is due to a number of reasons, including: (1) the variability of the influent; (2) the complexity of the biological and biochemical phenomena; (3) the large range of time constants (varying from a few minutes to several days); (4) the lack of standard evaluation criteria (among other things, due to region specific effluent requirements and cost levels).

It is thus difficult to judge the particular influence of an applied control strategy on reported plant performance increase, as the reference situation is often not properly characterized. Due to the complexity of the systems it takes much effort to develop alternative controller approaches and, as a consequence, a fair comparison between different control strategies is only made seldomly. And even if this is done, it remains difficult to conclude to what extent the proposed solution is process or location specific.

To enhance the acceptance of innovating control strategies, the performance evaluation should be based on a rigorous methodology including a reference simulation model, a precise plant layout, well-defined controllers, performance criteria and test procedures.

From 1998 to 2004, the development of benchmark tools for simulation-based evaluation of control strategies for activated sludge plants has been undertaken in Europe by Working Groups of COST Action 682 and 624 (Alex et al., 1999). This development work is now continued under the umbrella of the IWA Task Group on Benchmarking of Control Strategies for WWTPs.

The benchmark is a simulation environment defining a plant layout, a simulation model, influent loads, test procedures and evaluation criteria. For each of these items, compromises were pursued to combine plainness with realism and accepted standards. Once the user has validated the simulation code, any control strategy can be applied and the performance can be evaluated according to a defined set of criteria. The benchmark is not linked to a particular simulation platform: direct coding (C/C++, Fortran, Matlab®) as well as commercial WWTP simulation software packages (such as Simba®, West®, GPS-X®) can be used. For this reason, the full set of equations and all the parameter values are available in the present document.

The first layout (BSM1) (Alex et al., 2009) was relatively simple (Figure 1). The BSM1 plant is composed of a five-compartment activated sludge reactor consisting of two anoxic tanks followed by three aerobic tanks. It thus combines nitrification with pre-denitrification in a configuration that is commonly used for achieving biological nitrogen removal in full-scale plants. The activated sludge reactors are followed by a secondary clarifier.

Figure 1: General overview of the BSM1 plant within BSM2
The BSM2 layout (Figure 2) includes BSM1 for the biological treatment of the wastewater and the sludge treatment. A primary clarifier, a thickener for the sludge wasted from the BSM1 clarifier, a digester for treatment of the solids wasted from the primary clarifier and the thickened secondary sludge as well as a dewatering unit have been also added. The liquids collected in the thickening and dewatering steps are recycled ahead of the primary settler. Different possible control handles such as pumps, valves, aeration, etc. are also shown in Figure 2.

Figure 2: General overview of the BSM2 plant

The purpose of the present document is to describe in details the BSM2 benchmark system. An important part of the development of BSM2 has been to implement the ADM1 model for the anaerobic digester. This has implied some slight changes with respect to the original version of ADM1 as well as the development of calculation procedures in order to have a reasonable calculation time for the whole BSM2 plant (Rosen et al., 2006; Rosen and Jeppsson, 2009). Furthermore, interfaces to transform the ASM1 variables into ADM1 variables (and vice-versa) had to be implemented (Nopens et al., 2009). More details on the model development for some units can be found in the other sections of the Technical Report. Finally, to facilitate the understanding of the modelling, Figure 3 summarizes the notations used for the various flow rates throughout the BSM2 plant.

Figure 3: BSM2 plant simplified layout with notation used for flow rates
2. MODELING OF THE ACTIVATED SLUDGE SECTION

2.1. General characteristics

The plant is designed for an average influent dry-weather flow rate of 20,648.36 m$^3$.d$^{-1}$ and an average biodegradable COD in the influent of 592.53 g.m$^{-3}$. Its hydraulic retention time (based on average dry weather flow rate and total tank volume – i.e. primary clarifier (900 m$^3$) + biological reactor (12,000 m$^3$) + secondary clarifier (6,000 m$^3$) – of 18,900 m$^3$) is 22 hours.

The influent dynamics are defined for 609 days by means of a single file, which takes into account rainfall effect and temperature.

2.2. Bioprocess model

The Activated Sludge Model no. 1 (ASM1; Henze et al., 1987) has been selected to describe the biological phenomena taking place in the biological reactor (Figure 4).

Figure 4: General overview of ASM1

2.2.1. List of variables

The list of state variables, with their definition and appropriate notation, is given in Table 1. To these variables should be added $T_{as}$ for temperature.

Table 1: List of ASM1 variables

<table>
<thead>
<tr>
<th>Definition</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soluble inert organic matter</td>
<td>$S_I$</td>
</tr>
<tr>
<td>Readily biodegradable substrate</td>
<td>$S_S$</td>
</tr>
<tr>
<td>Particulate inert organic matter</td>
<td>$X_I$</td>
</tr>
<tr>
<td>Slowly biodegradable substrate</td>
<td>$X_S$</td>
</tr>
<tr>
<td>Active heterotrophic biomass</td>
<td>$X_{BH}$</td>
</tr>
<tr>
<td>Active autotrophic biomass</td>
<td>$X_{BA}$</td>
</tr>
<tr>
<td>Particulate products arising from biomass decay</td>
<td>$X_P$</td>
</tr>
<tr>
<td>Oxygen</td>
<td>$S_O$</td>
</tr>
<tr>
<td>Nitrate and nitrite nitrogen</td>
<td>$S_{NO}$</td>
</tr>
<tr>
<td>$NH_4^+$ + $NH_3$ nitrogen</td>
<td>$S_{NH}$</td>
</tr>
<tr>
<td>Soluble biodegradable organic nitrogen</td>
<td>$S_{ND}$</td>
</tr>
<tr>
<td>Particulate biodegradable organic nitrogen</td>
<td>$X_{ND}$</td>
</tr>
<tr>
<td>Alkalinity</td>
<td>$S_{ALK}$</td>
</tr>
</tbody>
</table>
2.2.2. List of processes

Eight basic processes ($\rho_k$, $k = 1$ to 8) are used to describe the biological behavior of the system.

- $j = 1$: Aerobic growth of heterotrophs

$$
\rho_{1,\text{as}} = \mu_{\text{HT}} \left( \frac{S_S}{K_S + S_S} \right) \left( \frac{S_O}{K_{\text{O,H}} + S_O} \right) X_{B,H} \tag{1}
$$

with:

$$
\mu_{\text{HT}} = \mu_{\text{H}} \cdot \exp \left( \frac{\mu_{\text{H}}}{3} \cdot \left( T_{\text{as}} - 15 \right) \right) \tag{2}
$$

- $j = 2$: Anoxic growth of heterotrophs

$$
\rho_{2,\text{as}} = \mu_{\text{HT}} \left( \frac{S_S}{K_S + S_S} \right) \left( \frac{K_{\text{O,H}}}{K_{\text{O,H}} + S_O} \right) \left( \frac{S_{\text{NO}}}{K_{\text{NO}} + S_{\text{NO}}} \right) \eta_{\text{X,B,H}} \tag{3}
$$

- $j = 3$: Aerobic growth of autotrophs

$$
\rho_{3,\text{as}} = \mu_{\text{AT}} \left( \frac{S_{\text{NH}}}{K_{\text{NH}} + S_{\text{NH}}} \right) \left( \frac{S_O}{K_{\text{O,A}} + S_O} \right) X_{B,A} \tag{4}
$$

with:

$$
\mu_{\text{AT}} = \mu_{\text{A}} \cdot \exp \left( \frac{\mu_{\text{A}}}{0.3} \cdot \left( T_{\text{as}} - 15 \right) \right) \tag{5}
$$

- $j = 4$: Decay of heterotrophs

$$
\rho_{4,\text{as}} = b_{\text{HT}} X_{B,H} \tag{6}
$$

with:

$$
b_{\text{HT}} = b_{\text{H}} \cdot \exp \left( \frac{b_{\text{H}}}{0.2} \cdot \left( T_{\text{as}} - 15 \right) \right) \tag{7}
$$

- $j = 5$: Decay of autotrophs

$$
\rho_{5,\text{as}} = b_{\text{AT}} X_{B,A} \tag{8}
$$

with $b_{\text{AT}} = b_{\text{A}} \cdot \exp \left( \frac{b_{\text{A}}}{0.03} \cdot \left( T_{\text{as}} - 15 \right) \right) \tag{9}

- $j = 6$: Ammonification of soluble organic nitrogen

$$
k_{\text{as}} = k_{\text{a}} \cdot \exp \left( \frac{k_{\text{a}}}{0.04} \cdot \left( T_{\text{as}} - 15 \right) \right) \tag{10}
$$

- $j = 7$: Hydrolysis of entrapped organics

$$
k_{\text{as}} = k_{\text{h}} \cdot \exp \left( \frac{k_{\text{h}}}{2.5} \cdot \left( T_{\text{as}} - 15 \right) \right) \tag{12}
$$

- $j = 8$: Hydrolysis of entrapped organic nitrogen

$$
k_{\text{as}} = k_{\text{h}} \cdot \exp \left( \frac{k_{\text{h}}}{2.5} \cdot \left( T_{\text{as}} - 15 \right) \right) \tag{13}
$$

2.2.3. Observed conversion rates

The observed conversion rates ($r_k$) result from combinations of the basic processes:

- $S_1$ ($k = 1$)
2.2.4. Biological parameter values

The base biological parameter values used in the activated sludge section of BSM2 correspond to a temperature of 15°C. The stoichiometric parameters are listed in Table 2 and the kinetic parameters in Table 3.
Table 2: Stoichiometric parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma_A$</td>
<td>g cell COD formed.(g N oxidized)$^{-1}$</td>
<td>0.24</td>
</tr>
<tr>
<td>$\gamma_H$</td>
<td>g cell COD formed.(g COD oxidized)$^{-1}$</td>
<td>0.67</td>
</tr>
<tr>
<td>$f_0$</td>
<td>dimensionless</td>
<td>0.08</td>
</tr>
<tr>
<td>$i_{XB}$</td>
<td>g N.(g COD)$^{-1}$ in biomass</td>
<td>0.08</td>
</tr>
<tr>
<td>$i_{XP}$</td>
<td>g N.(g COD)$^{-1}$ in particulate products</td>
<td>0.06</td>
</tr>
</tbody>
</table>

Table 3: Kinetic parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_H$</td>
<td>d$^{-1}$</td>
<td>4.0</td>
</tr>
<tr>
<td>$K_S$</td>
<td>g COD.m$^{-3}$</td>
<td>10.0</td>
</tr>
<tr>
<td>$K_{O,H}$</td>
<td>g (–COD).m$^{-3}$</td>
<td>0.2</td>
</tr>
<tr>
<td>$K_{NO}$</td>
<td>g NO$\text{-N}.m^{-3}$</td>
<td>0.5</td>
</tr>
<tr>
<td>$b_H$</td>
<td>d$^{-1}$</td>
<td>0.3</td>
</tr>
<tr>
<td>$\eta_S$</td>
<td>dimensionless</td>
<td>0.8</td>
</tr>
<tr>
<td>$\eta_H$</td>
<td>dimensionless</td>
<td>0.8</td>
</tr>
<tr>
<td>$k_{H}$</td>
<td>g slowly biodegradable COD.(g cell COD . d)$^{-1}$</td>
<td>3.0</td>
</tr>
<tr>
<td>$K_X$</td>
<td>g slowly biodegradable COD.(g cell COD)$^{-1}$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\mu_A$</td>
<td>d$^{-1}$</td>
<td>0.5</td>
</tr>
<tr>
<td>$K_{NH}$</td>
<td>g NH$\text{-N}.m^{-3}$</td>
<td>1.0</td>
</tr>
<tr>
<td>$b_A$</td>
<td>d$^{-1}$</td>
<td>0.05</td>
</tr>
<tr>
<td>$K_{O,A}$</td>
<td>g (–COD).m$^{-3}$</td>
<td>0.4</td>
</tr>
<tr>
<td>$k_a$</td>
<td>m$^3$.m$^{-3}$.d$^{-1}$</td>
<td>0.05</td>
</tr>
</tbody>
</table>

2.3. Detailed activated sludge section layout

2.3.1. Bioreactor (general characteristics)

According to Figure 1, the general characteristics of the bioreactor for the default case are:
- Number of compartments: 5
- Non-aerated compartments: compartments 1-2
- Aerated compartments: compartments 3-5

For each compartment, the following variables have been defined ($k = 1$ to 5):
- Flow rate: $Q_k$
- Concentration: $Z_{as,k}$
- Volume:
  - Non-aerated compartments: $V_{as,1} = V_{as,2} = 1,500$ m$^3$
  - Aerated compartments: $V_{as,3} = V_{as,4} = V_{as,5} = 3,000$ m$^3$
- Reaction rate: $r_k$

2.3.2. Reactor mass balances (general formula)

The general equations for mass balancing are as follows:

For $k = 1$ (compartment 1)

$$\frac{dZ_{as,1}}{dt} = \frac{1}{V_{as,1}} (Q_{int} Z_{int} + Q_r Z_r + Q_{po} Z_{po} + r_{Z_1} V_{as,1} - Q_i Z_{as,1})$$  \hspace{1cm} (28)

$$Q_i = Q_{int} + Q_r + Q_{po}$$  \hspace{1cm} (29)

where $Q_{int}$ is the internal recycle from compartment 5, $Q_r$ the external recycle from the underflow of the secondary clarifier and $Q_{po}$ the overflow rate of the primary clarifier.

For $k = 2$ to 5

$$\frac{dZ_{as,k}}{dt} = \frac{1}{V_{as,k}} (Q_{k-1} Z_{as,k-1} + r_{Z,k} V_{as,k} - Q_k Z_{as,k})$$  \hspace{1cm} (30)

$$Q_k = Q_{k-1}$$  \hspace{1cm} (31)
Special case for oxygen ($S_{O,as,k}$)

\[
\frac{dS_{O,as,k}}{dt} = \frac{1}{V_{as,k}} \left( Q_{k-1} S_{O,as,k-1} + r_{Z,k} V_{as,k} + \left(K_L d_h\right) r_{as,k} \left(S_{O,as,k}^{\text{sat}} - S_{O,as,k}\right) - Q_k S_{O,as,k}\right)
\]  

(32)

where the saturation concentration for oxygen is $S_{O,as,k}^{\text{sat}}$ and is a function of temperature (cf §2.4).

$r_{Z,k}$ stands for the appropriate conversion rate, depending upon the state variable considered (cf §§ 2.2.3).

- Temperature ($T_{as}$)

An adiabatic temperature balance is assumed (no heat exchange with the environment)

\[
\frac{dT_{as,k}}{dt} = \frac{1}{V_{as,k}} \left( Q_{k-1} T_{as,k-1} - Q_k T_{as,k}\right)
\]  

(33)

- Miscellaneous

\[
Z_{\text{int}} = Z_{as,5}
\]  

(34)

\[
Z_f = Z_{as,5}
\]  

(35)

\[
Z_w = Z_r
\]  

(36)

\[
Q_f = Q_{sc,e} + Q_e + Q_w = Q_{sc,e} + Q_{sc,u}
\]  

(37)

Where $Q_{sc,e}$ and $Q_{sc,u}$ are the overflow and underflow rates from the secondary clarifier respectively and $Q_w$ is the wastage flow rate.

### 2.3.3. Secondary clarifier

The secondary clarifier is modeled as a 10 layers non-reactive unit (i.e. no biological reaction). The 6th layer (counting from bottom to top) is the feed layer. The secondary clarifier has an area ($A$) of 1,500 m$^2$. The height of each layer $m$ ($z_m$) is equal to 0.4 m, for a total height of 4 m. Therefore the secondary clarifier volume is equal to 6,000 m$^3$.

The solid flux due to gravity is $\nu_s(X_{sc}) X_{sc}$ where $X_{sc}$ is the total sludge concentration. A double-exponential settling velocity function (Takács et al., 1991) has been selected:

\[
\nu_s(X_{sc}) = \max\left[0, \min\left\{v_0, \nu_0 \left(e^{-r_h (X_{sc} - X_{min})} - e^{-r_p (X_{sc} - X_{min})}\right)\right]\right]
\]  

(38)

with $X_{min} = f_{ns} X_f$. $X_f$ is the total solid concentration from the biological reactor. The parameter values for the settling velocity function are given in Table 4.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum settling velocity</td>
<td>$v_0$</td>
<td>250.0</td>
</tr>
<tr>
<td>Maximum Vesilind settling velocity</td>
<td>$v_0$</td>
<td>474</td>
</tr>
<tr>
<td>Hindered zone settling parameter</td>
<td>$r_h$</td>
<td>0.000576</td>
</tr>
<tr>
<td>Flocculant zone settling parameter</td>
<td>$r_p$</td>
<td>0.00286</td>
</tr>
<tr>
<td>Non-settleable fraction</td>
<td>$f_{ns}$</td>
<td>0.00228</td>
</tr>
</tbody>
</table>

The upward ($v_{up}$) and downward ($v_{dn}$) velocities are calculated as:

\[
v_{dn} = \frac{Q_{sc,u}}{A} = \frac{Q_e + Q_w}{A}
\]  

(39)

\[
v_{up} = \frac{Q_{sc,e}}{A}
\]  

(40)

According to these notations, the mass balances for the sludge are written as:

For the feed layer ($m = 6$):
For the intermediate layers below the feed layer \((m = 2 \text{ to } m = 5)\):
\[
\frac{dX_{sc,m}}{dt} = v_{dn} \left(X_{sc,m+1} - X_{sc,m}\right) + \min(J_s,m,J_{s,m+1}) - \min(J_s,m,J_{s,m-1})
\]
\(z_m\) (42)

For the bottom layer \((m = 1)\):
\[
\frac{dX_{sc,1}}{dt} = v_{dn} \left(X_{sc,2} - X_{sc,1}\right) + \min(J_s,2,J_{s,1})
\]
\(z_1\) (43)

For the intermediate clarification layers above the feed layer \((m = 7 \text{ to } m = 9)\):
\[
\frac{dX_{sc,m}}{dt} = v_{up} \left(X_{sc,m-1} - X_{sc,m}\right) + J_{sc,m+1} - J_{sc,m}
\]
\(z_m\) (44)

\[
J_{sc,j} = \begin{cases} 
\min(v_{s,j}X_{sc,j}, v_{s,j-1}X_{sc,j-1}) & \text{if } X_{sc,j-1} > X_t \\
& \text{or} \\
v_{s,j}X_{sc,j} & \text{if } X_{sc,j-1} \leq X_t 
\end{cases}
\] (45)

For the top layer \((m = 10)\):
\[
\frac{dX_{sc,10}}{dt} = v_{up} \left(X_{sc,9} - X_{sc,10}\right) - J_{sc,10}
\]
\(z_{10}\) (46)

\[
J_{sc,10} = \begin{cases} 
\min(v_{s,10}X_{sc,10}, v_{s,9}X_{sc,9}) & \text{if } X_{sc,9} > X_t \\
& \text{or} \\
v_{s,10}X_{sc,10} & \text{if } X_{sc,9} \leq X_t 
\end{cases}
\] (47)

The threshold concentration \(X_t\) is equal to 3,000 g.m\(^{-3}\).

For the soluble components (including dissolved oxygen and temperature \((T_{sc})\)), each layer represents a completely mixed volume and the concentrations of soluble components are calculated accordingly:

For the feed layer \((m = 6)\):
\[
\frac{dZ_{sc,m}}{dt} = \frac{Q_tZ_f}{A} - \left(v_{dn} + v_{up}\right)Z_{sc,m}
\]
\(z_m\) (48)

For the layers \(m = 1 \text{ to } 5\):
\[
\frac{dZ_{sc,m}}{dt} = v_{dn} \left(Z_{sc,m+1} - Z_{sc,m}\right)
\]
\(z_m\) (49)

For the layers \(m = 7 \text{ to } 10\):
\[
\frac{dZ_{sc,m}}{dt} = v_{up} \left(Z_{sc,m-1} - Z_{sc,m}\right)
\]
\(z_m\) (50)

The concentrations in the recycle and wastage flow are equal to those of the first layer (bottom layer):
\[
Z_u = Z_{sc,1}
\] (51)
Calculation of the sludge concentration is straightforward from the concentrations in compartment 5 of the activated sludge reactor:

\[
X_f = \frac{1}{f_{\text{COD-SS}}} \left( X_{\text{S,as},5} + X_{\text{P,as},5} + X_{\text{L,as},5} + X_{\text{B,H,as},5} + X_{\text{B,A,as},5} \right) \\
= 0.75 \left( X_{\text{S,as},5} + X_{\text{P,as},5} + X_{\text{L,as},5} + X_{\text{B,H,as},5} + X_{\text{B,A,as},5} \right) \tag{52}
\]

given a COD to SS conversion factor, \( f_{\text{COD-SS}} \), equal to 4/3. The same principle is applied for \( X_u \) (in the clarifier underflow) and \( X_c \) (at the secondary clarifier exit).

To calculate the distribution of particulate concentrations in the recycle and the wastage flows, their ratios with respect to the total solid concentration are assumed to remain constant across the clarifier:

\[
\frac{X_{\text{S,as},5}}{X_f} = \frac{X_{\text{S,sc},1}}{X_u} \tag{53}
\]

Similar equations hold for \( X_{\text{P,sc},1}, X_{\text{I,sc},1}, X_{\text{B,H,sc},1}, \) and \( X_{\text{B,A,sc},1} \). Note that this assumption means that the dynamics of the fractions of particulate concentrations in the inlet of the clarifier will be directly propagated to the clarifier underflow and overflow, without taking into account the normal retention time in the clarifier.

The sludge age calculation is based on the total amount of biomass present in the system, i.e. the reactor and the settler:

\[
SRT = \frac{T X_{\text{as}} + TX_{\text{sc}}}{\phi_e + \phi_w} \tag{54}
\]

where \( TX_{\text{as}} \) is the total amount of biomass present in the reactor:

\[
TX_{\text{as}} = \sum_{k=1}^{n} \left( X_{\text{B,H,as},k} + X_{\text{B,A,as},k} \right) V_{\text{as},k} \quad \text{with } n = 5 \tag{55}
\]

\( TX_{\text{sc}} \) is the total amount of biomass present in the secondary clarifier:

\[
TX_{\text{sc}} = \sum_{j=1}^{m} \left( X_{\text{B,H,sc},j} + X_{\text{B,A,sc},j} \right) z_j \cdot A \quad \text{with } m = 10 \tag{56}
\]

\( \phi_e \) is the loss rate of biomass in the secondary clarifier overflow:

\[
\phi_e = \left( X_{\text{B,H,sc},m} + X_{\text{B,A,sc},m} \right) Q_{\text{sc},e} \tag{57}
\]

and \( \phi_w \) is the loss rate of biomass in the wastage flow:

\[
\phi_w = \left( X_{\text{B,H,sc},1} + X_{\text{B,A,sc},1} \right) Q_w \tag{58}
\]

In an actual plant the sludge age is measured based on the total amount of solids present in the system:

\[
SRT_{\text{meas}} = \frac{TSS_{\text{as}} + TSS_{\text{sc}}}{\psi_e + \psi_w} \tag{59}
\]

where \( TSS_{\text{as}} \) is the total amount of solids present in the reactor:

\[
TSS_{\text{as}} = \sum_{k=1}^{n} TSS_{\text{as},k} \cdot V_{\text{as},k} \quad \text{with } n = 5 \tag{60}
\]

\( TSS_{\text{sc}} \) is the total amount of solids present in the clarifier:

\[
TSS_{\text{sc}} = \sum_{j=1}^{m} TSS_{\text{sc},j} \cdot z_j \cdot A \quad \text{with } m = 10 \tag{62}
\]

\( \psi_e \) is the loss rate of solids in the secondary clarifier overflow:

\[
\psi_e = TSS_{\text{sc},m} \cdot Q_{\text{sc},e} \tag{64}
\]
with \( TSS_{sc,m} = \frac{1}{f_{COD-SS}} \left( X_{S,sc,m} + X_{P,sc,m} + X_{L,sc,m} + X_{B,H,sc,m} + X_{B,A,sc,m} \right) \) \hspace{1cm} (65)

for \( m = 10 \).

\( \psi_w \) is the loss rate of solids in the wastage flow:

\[ \psi_w = TSS_{sc,1} \cdot \dot{Q}_w \] \hspace{1cm} (66)

with \( TSS_{sc,1} = \frac{1}{f_{COD-SS}} \left( X_{S,sc,1} + X_{P,sc,1} + X_{L,sc,1} + X_{B,H,sc,1} + X_{B,A,sc,1} \right) \) \hspace{1cm} (67)

### 2.3.4. Plant effluent composition

In BSM2, the plant effluent composition is based on the secondary clarifier overflow and the raw wastewater that might be by-passed at the inlet of the plant. For any composition state variable:

\[ Z_i = \left( Q_{sc,e} \cdot Z_{sc,10} + Q_{bypass} \cdot Z_1 \right) \pm \left( Q_{sc,e} + Q_{bypass} \right) \] \hspace{1cm} (68)

where \( Z_i \) is the concentration in the raw wastewater.

### 2.4. Oxygen transfer coefficient

The oxygen transfer coefficient, \( K_{La} \), depends on temperature. ASCE (1993) presents the generally accepted dependency of the oxygen transfer coefficient \( K_{La} \) on temperature:

\[ K_{La} = \frac{1.024(T^{1.5})}{T} \cdot K_{La}(15^\circ C) \] \hspace{1cm} (69)

### 2.5. Oxygen concentration at saturation

Solubility of oxygen is dependent on temperature, decreasing with decreasing temperature.

\[ S_{O_2,sat}(T_{as}) = 0.9997743214 \cdot \frac{8}{10.5} \cdot 67915 \cdot K(T_{K}) \] \hspace{1cm} (70)

With \( K(T_{K}) = 56.12 e^{0.48 T_{*} + 0.23 \ln T_{*}} \) \hspace{1cm} (71)

The formula is valid in the range 273.15 \( K \) — 348.15 \( K \), where \( T_{*} = T_{as}/100 \) (K), \( T_{K} = T_{as}^2/273.15 \), \( A = -66.7354 \), \( B = 87.4755 \), \( C = 24.4526 \).

### 3. MODELING OF THE ANAEROBIC DIGESTER

#### 3.1. Introduction

The ADM1 implementation deviates somewhat from the model description in Batstone et al. (2002). There are mainly three reasons for this. Firstly, the ADM1 is implemented so that it is consistent with the other sections of BSM2. Secondly, the computational requirements must be regarded. Thirdly, no explicit values are given in Batstone et al. (2002) with regard to carbon and nitrogen contents of some state variables.

#### 3.1.1. Acid-base equations

The acid-base equilibrium equations play an important role in ADM1 (e.g. for pH calculations). For persons primarily familiar with AS models, these equations may create a problem as they do not normally appear in those. Moreover, (Batstone et al., 2002) focus more on how the implementation should be done by implicit algebraic equations and is not completely clear on the ODE implementation. The general model matrix describes the transformations of valerate \( (S_{va,total}) \), butyrate, propionate, acetate, inorganic carbon and inorganic nitrogen. However, all these substances are made up by acid-base pairs (e.g. \( S_{va,total} = S_{va}^- + S_{hva}^+ \)). It is suggested in Batstone et al. (2002) that when using ODEs, the equations are defined for each acid and base, respectively. Based on our experiences it is more advantageous to implement the ODEs based on the total and one of the acid-base components instead. The remaining part can always be calculated as the total minus the calculated part. This approach actually makes the model more understandable also in other respects and due to numerical issues (we are subtracting very small and similar sized numbers) the error of calculated outputs are much closer to the solution a differential-algebraic equation (DAE) setup would provide (when using a numerical solver with the same tolerance to integrate the ODEs). Using valerate as an example, the process rate \( (A4) \) in (Batstone et al., 2002) is:
Benchmark Simulation Model no. 2 (BSM2)

\[ K_{A,BVA} \left( S_{va} - S_{H^+} - K_{A,va} S_{BVA} \right) \]  

(72)

and herein we replace \( S_{BVA} \) by \( S_{va, total} - S_{va} \) and get

\[ K_{A,BVA} \left( S_{va} - K_{A,va} S_{va} - S_{BVA} \right) \]  

(73)

and, consequently, change the stoichiometry since \( S_{va} \) is not affected when the equilibrium of \( S_{H^+} \) is changing. If using the suggested implicit solver to calculate the pH (or \( S_{H^+} \)) at every integration step (see below) then the above problem will no longer be an issue.

The choice of a ODE or DAE system for modeling the pH should not affect the overall results of the model. The DAE can be said to be an approximation of the ODE since, naturally, the pH dynamics are not instantaneous. However, it is very common to model the dynamics as a DAE system in biochemical/chemical engineering. Thus, the rate coefficients \( k_{A,Bi} \) should be defined in such a way that the ODE produces the same results as the DAE. In Batstone et al. (2002), it is recommended that the coefficients should be chosen so that they are at least one order of magnitude faster (larger) than the fastest time constant of the remaining system and the value \( 1.1 \times 10^8 \text{ M}^{-1} \text{d}^{-1} \) is recommended. However, this is not sufficient. For the ODE to yield identical results, the rate coefficients need to be larger and a value of \( 1.1 \times 10^{10} \text{ M}^{-1} \text{d}^{-1} \) is more appropriate.

3.1.2. Temperature dependencies

In order to better allow for reasonable results for different temperatures within the digester, the benchmark ADM1 implementation now uses the complete information as stated in the ADM1 STR with regard to temperature dependency of several physiochemical parameters (see the table for physiochemical parameters). This means that a model user can work with different temperatures when investigating the system without having to recalculate these parameters. The parameters that are now considered to be functions of temperature are: \( K_W, K_{a,CO2}, K_{a,IN}, K_{H,CO2}, K_{H,CH4}, K_{H,H2} \) and \( p_{gas,h2o} \) (i.e. water vapor saturation pressure).

The \( K_a \) values for the organic acids are not assumed to vary within the selected temperature range (0 - 60 °C) and are assumed to be constants (see also Batstone et al. (2002), p. 39). For an even better temperature dependency of the AD model many of the biochemical parameter values would also need to be described as functions of temperature.

3.2. Model equations

3.2.1. Process rates

The biochemical process rates are defined as:

Disintegration:
\[ \rho_{1,ad} = k_{dis} \cdot X_c \]  

(74)

Hydrolysis of carbohydrates:
\[ \rho_{2,ad} = k_{hyd,ch} \cdot X_{ch} \]  

(75)

Hydrolysis of proteins:
\[ \rho_{3,ad} = k_{hyd,pr} \cdot X_{pr} \]  

(76)

Hydrolysis of lipids:
\[ \rho_{4,ad} = k_{hyd,li} \cdot X_{li} \]  

(77)

Uptake of sugars:
\[ \rho_{5,ad} = k_{m,su} \cdot \frac{S_{su}}{K_{S,su} + S_{su}} \cdot X_{su} \cdot I_5 \]  

(78)

Uptake of amino-acids:
\[ \rho_{6,ad} = k_{m,aa} \cdot \frac{S_{aa}}{K_{S,aa} + S_{aa}} \cdot X_{aa} \cdot I_6 \]  

(79)

Uptake of LCFA:
\[ \rho_{7,ad} = k_{m,fa} \cdot \frac{S_{fa}}{K_{S,fa} + S_{fa}} \cdot X_{fa} \cdot I_7 \]  

(80)

Uptake of valerate:
ρ\text{8,ad} = \frac{S_{va}}{K_{S,va} + S_{va}} \cdot X_{c4} \cdot \frac{S_{va}}{S_{bu} + S_{va}} \cdot I_8 \quad (81)

Uptake of butyrate:

ρ\text{9,ad} = \frac{S_{bu}}{K_{S,bu} + S_{bu}} \cdot X_{c4} \cdot \frac{S_{Bu}}{S_{bu} + S_{bu}} \cdot I_9 \quad (82)

Uptake of propionate:

ρ\text{10,ad} = \frac{S_{pro}}{K_{S,pro} + S_{pro}} \cdot X_{pro} \cdot I_{10} \quad (83)

Uptake of acetate:

ρ\text{11,ad} = \frac{S_{ac}}{K_{S,ac} + S_{ac}} \cdot X_{ac} \cdot I_{11} \quad (84)

Uptake of hydrogen:

ρ\text{12,ad} = \frac{S_{h2}}{K_{S,h2} + S_{h2}} \cdot X_{h2} \cdot I_{12} \quad (85)

Decay of $X_{su}$:

ρ\text{13,ad} = k_{dec,Xsu} \cdot X_{su} \quad (86)

Decay of $X_{aa}$:

ρ\text{14,ad} = k_{dec,Xaa} \cdot X_{aa} \quad (87)

Decay of $X_{fa}$:

ρ\text{15,ad} = k_{dec,Xfa} \cdot X_{fa} \quad (88)

Decay of $X_{c4}$:

ρ\text{16,ad} = k_{dec,Xc4} \cdot X_{c4} \quad (89)

Decay of $X_{pro}$:

ρ\text{17,ad} = k_{dec,Xpro} \cdot X_{pro} \quad (90)

Decay of $X_{ac}$:

ρ\text{18,ad} = k_{dec,Xac} \cdot X_{ac} \quad (91)

Decay of $X_{h2}$:

ρ\text{19,ad} = k_{dec,Xh2} \cdot X_{h2} \quad (92)

In the expressions for $\rho_{8,ad}$ and $\rho_{9,ad}$ (Eqs. 81 and 82), a small constant (1.10^{-6}) can be added at the denominator to the sum ($S_{va} + S_{bu}$) in order to avoid division by zero in the case of poor choice of initial conditions for $S_{va}$ and $S_{bu}$, respectively.

The acid-base rates for the ODE implementation are as follows:

\begin{align*}
\text{3.2.2. Process inhibition} \\
\text{The process inhibition terms are expressed as:}
\end{align*}
Batstone et al. (2002) used switch functions to account for inhibition due to pH. These functions are, however, discontinuous and in a stiff system, such a switch can favour numerical instabilities. To reduce this risk, a number of alternative functions can be used to express the inhibition due to pH. Siegrist et al. (2002) used a Hill inhibition function based on the hydrogen ion concentration. This solution has been chosen for BSM2. For the ADM1, this gives the following expressions:

\[
I_{\text{ph},\text{aa}} = \frac{K_{\text{ph}}^{\text{a}}}{S_{\text{h}} + K_{\text{ph}}^{\text{a}}} \text{ with } K_{\text{ph}} = 10^{\frac{\text{pH}_{1,\text{aa}} + \text{pH}_{1,\text{aa}}}{2}} \quad \text{and} \quad n_{\text{aa}} = \frac{3.0}{\text{pH}_{1,\text{aa}} - \text{pH}_{1,\text{aa}}} \\
I_{\text{ph},\text{ac}} = \frac{K_{\text{ph}}^{\text{a}}}{S_{\text{h}} + K_{\text{ph}}^{\text{a}}} \text{ with } K_{\text{ph}} = 10^{\frac{\text{pH}_{1,\text{ac}} + \text{pH}_{1,\text{ac}}}{2}} \quad \text{and} \quad n_{\text{ac}} = \frac{3.0}{\text{pH}_{1,\text{ac}} - \text{pH}_{1,\text{ac}}} \\
I_{\text{ph},\text{h2}} = \frac{K_{\text{ph}}^{\text{a}}}{S_{\text{h}} + K_{\text{ph}}^{\text{a}}} \text{ with } K_{\text{ph}} = 10^{\frac{\text{pH}_{1,\text{h2}} + \text{pH}_{1,\text{h2}}}{2}} \quad \text{and} \quad n_{\text{h2}} = \frac{3.0}{\text{pH}_{1,\text{h2}} - \text{pH}_{1,\text{h2}}} 
\]

### 3.2.3. Liquid phase equations

The influent liquid flow rate to the anaerobic digester is calculated as:

\[
Q_{\text{ad}} = Q_{\text{tu}} + Q_{\text{pu}} 
\]

where \(Q_{\text{tu}}\) is the flow rate from the thickener underflow and \(Q_{\text{pu}}\) is the flow rate from the primary settler underflow.

The water-phase equations are written as:

\[
\frac{dS_{\text{u,si}}}{dt} = \frac{Q_{\text{ad}}}{V_{\text{ad,liq}}} (S_{\text{u,si}} - S_{\text{u}}) + \rho_{2,\text{ad}} + (1 - f_{\text{a,li}}) \rho_{3,\text{ad}} - \rho_{5,\text{ad}} \\
\frac{dS_{\text{ai}}}{dt} = \frac{Q_{\text{ad}}}{V_{\text{ad,liq}}} (S_{\text{ai}} - S_{\text{ai}}) + \rho_{3,\text{ad}} - \rho_{6,\text{ad}} 
\]
\[
\frac{dS_{fa}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (S_{fa,i} - S_{fa}) + f_{fa,liq} \rho_{4,ad} - \rho_{7,ad}
\]

(119)

\[
\frac{dS_{va}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (S_{va,i} - S_{va}) + (1 - Y_{aa}) f_{va,aa} \rho_{6,ad} - \rho_{8,ad}
\]

(120)

\[
\frac{dS_{bu}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (S_{bu,i} - S_{su}) + (1 - Y_{su}) f_{bu,su} \rho_{5} + (1 - Y_{aa}) f_{bu,aa} \rho_{6,ad} - \rho_{9,ad}
\]

(121)

\[
\frac{dS_{pro}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (S_{pro,i} - S_{pro}) + (1 - Y_{su}) f_{pro,si} \rho_{5,ad} + (1 - Y_{aa}) f_{pro,aa} \rho_{6,ad} + (1 - Y_{ca}) 0.54 \rho_{8,ad} - \rho_{10,ad}
\]

(122)

\[
\frac{dS_{ac}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (S_{ac,i} - S_{ac}) + (1 - Y_{su}) f_{ac,si} \rho_{5,ad} + (1 - Y_{aa}) f_{ac,aa} \rho_{6,ad} + (1 - Y_{ca}) 0.7 \rho_{7,ad}
\]

(123)

\[
+ (1 - Y_{ca}) 0.31 \rho_{8,ad} + (1 - Y_{ca}) 0.8 \rho_{9,ad} + (1 - Y_{ca}) 0.57 \rho_{10,ad} - \rho_{11,ad}
\]

\[
\frac{dS_{h2}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (S_{h2,i} - S_{h2}) + (1 - Y_{su}) f_{h2,si} \rho_{5,ad} + (1 - Y_{aa}) f_{h2,aa} \rho_{6,ad} + (1 - Y_{ca}) 0.3 \rho_{7,ad}
\]

(124)

\[
+ (1 - Y_{ca}) 0.15 \rho_{8,ad} + (1 - Y_{ca}) 0.2 \rho_{9,ad} + (1 - Y_{ca}) 0.43 \rho_{10,ad} - \rho_{12,ad} - \rho_{T,8}
\]

\[
\frac{dS_{ch4}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (S_{ch4,i} - S_{ch4}) + (1 - Y_{ac}) \rho_{11,ad} + (1 - Y_{h2}) \rho_{12,ad} - \rho_{T,9}
\]

(125)

\[
\frac{dS_{ic}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (S_{ic,i} - S_{ic}) - \sum_{j=1}^{19} \sum_{i=1, i \neq 24}^{19} C_j V_{ic,j} \rho_{j,ad} - \rho_{T,10}
\]

(126)*

(127)

(128)

*More specifically, the sum in equation 126 is calculated as:

(129)

where

(130)

(131)

(132)

(133)

(134)
Benchmark Simulation Model no. 2 (BSM2)

Differential equations 143 to 154 (particulate matter)

\[
\frac{dX_c}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (X_{c,i} - X_c) - \rho_{1,ad} + \rho_{3,ad} + \rho_{14,ad} + \rho_{15,ad} + \rho_{16,ad} + \rho_{17,ad} + \rho_{19,ad}
\]

(143)

\[
\frac{dX_{ch}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (X_{ch,i} - X_{ch}) + f_{ch,xc} \rho_{1,ad} - \rho_{2,ad}
\]

(144)

\[
\frac{dX_{pr}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (X_{pr,i} - X_{pr}) + f_{pr,xc} \rho_{1,ad} - \rho_{3,ad}
\]

(145)

\[
\frac{dX_{li}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (X_{li,i} - X_{li}) + f_{li,xc} \rho_{1,ad} - \rho_{4,ad}
\]

(146)

\[
\frac{dX_{su}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (X_{su,i} - X_{su}) + Y_{su} \rho_{5,ad} - \rho_{13,ad}
\]

(147)

\[
\frac{dX_{I}}{dt} = \frac{Q_{ad}}{V_{ad,liq}} (X_{I,i} - X_{I}) + f_{xi,xc} \rho_{1,ad}
\]

(154)

Differential equations 155 and 156 (cations and anions)
\[ \frac{dS_{\text{cat}}}{dt} = \frac{Q_{\text{ad}}}{V_{\text{ad,liq}}} (S_{\text{cat}}^{+} - S_{\text{cat}}^{-}) \] (155)

\[ \frac{dS_{\text{an}}}{dt} = \frac{Q_{\text{ad}}}{V_{\text{ad,liq}}} (S_{\text{an}}^{+} - S_{\text{an}}^{-}) \] (156)

Differential equations 157 to 162 (ion states for ODE implementation)

\[ \frac{dS_{\text{va}}}{dt} = -\rho_{A,4} \] (157)

\[ \frac{dS_{\text{ba}}}{dt} = -\rho_{A,5} \] (158)

\[ \frac{dS_{\text{peo}}}{dt} = -\rho_{A,6} \] (159)

\[ \frac{dS_{\text{ac}}}{dt} = -\rho_{A,7} \] (160)

\[ \frac{dS_{\text{hco3}}}{dt} = -\rho_{A,10} \] (161)

\[ \frac{dS_{\text{nh3}}}{dt} = -\rho_{A,11} \] (162)

The algebraic equations involved for pH calculation are given as:

\[ \frac{dS_{\text{gas,cb4}}}{dt} = -\frac{S_{\text{gas,cb4}}Q_{\text{gas}}}{V_{\text{ad,gas}}} + \rho_{T,9} \frac{V_{\text{ad,liq}}}{V_{\text{ad,gas}}} \] (168)

3.2.4. Gas phase equations

Differential equations 167 to 169 describe the fate of the gas phase components:
The necessary algebraic equations are:

\[ Q_{\text{gas}} = k_p \left( P_{\text{gas}} - P_{\text{atm}} \right) \frac{P_{\text{gas}}}{P_{\text{atm}}} \]  

A problem with this way of calculating the gas flow rate is that it may give rise to numerical problems in the solution of the equations. Multiple steady states as well as numerical instability have been reported among users. An alternative way of calculating the gas flow rate is also given in Batstone et al. (2002):

\[ Q_{\text{gas}} = k_p \left( P_{\text{gas}} - P_{\text{atm}} \right) \frac{P_{\text{gas}}}{P_{\text{atm}}} \]  

with

\[ \text{The alternative expression assumes an overpressure in the head space. Consequently, the flow rate is calculated at a higher pressure compared to the first expression. To compensate for this, the expression needs to rewritten into} \]

\[ Q_{\text{gas}} = \frac{k_p}{P_{\text{atm}}} \left( P_{\text{gas}} - P_{\text{atm}} \right) \]  

\[ \text{to obtain the flow rate at atmospheric pressure. Although this compensation factor is included, the two expressions will not yield to identical results. Depending on the operational overpressure, which is a function of the value of parameter } k_p \text{ (related to the friction in the gas outlet), the alternative expression results in a slightly smaller flow rate. The reason for this is that the liquid-gas transfer rates } (\rho_{T,8}, \rho_{T,9}, \rho_{T,10}) \text{ will be different. A comparison of the two expressions when the same overpressure is applied shows very similar results (the relative error in the range of } 1.10^{-5}) \text{. For BSM2, the alternative way (assuming an overpressure in the head space) of calculating the gas flow rate is used. Also note that if the physical or operational conditions of the digester model are changed (volume, load etc.), for example if applying the ADM1 as a stand-alone model outside the framework of BSM2, then the parameter } k_p \text{ will have to be adjusted to achieve a reasonable overpressure in the head space.} \]

### 3.3. ADM1 DAE implementation

It has been realized that the ODE implementation may be problematic for use in the BSM2 framework. The model must be able to handle dynamic inputs, time discrete and event-driven control actions as well as stochastic inputs or noise and still be sufficiently efficient and fast to allow for extensive simulations. The ADM1 is a very stiff system\(^1\) with time constants ranging from fractions of a second to months. This makes the simulation of such a system challenging and in order to avoid excessively long simulation times, one needs to be somewhat creative when implementing the model.

Some solvers are so called stiff solvers and, consequently, capable of solving stiff systems. However, a problem common to all stiff solvers is the difficulty to handle dynamic input - including noise. The more stochastic or random an input variable behaves, the more problematic is the simulation using a stiff solver. The reason for this is that in stiff solvers, predictions of future state values are carried out. However, predictions of future state

\(^1\) A system is called stiff, when the range of the time constants is large. This means that some of the system states react quickly whereas some others react sluggishly.
values affected by stochastic inputs will result in poor results, slowing down the solver by limiting its ability to use long integration steps. Simulation of the BSM2 is, thus, subject to the following dilemma. BSM2, which includes ASM1 and ADM1 models, is a very stiff system and, consequently, a stiff solver should be used. However, since BSM2 is a control simulation benchmark, noise must be included, calling for an explicit (i.e. non-stiff) solver.

In this section, the differential algebraic equation model implementation of ADM1 is presented. Two different DAE models are discussed: a model with algebraic pH ($S_{\text{H}^+}$) calculations and a model with algebraic pH and $S_{\text{H}_2}$ calculations (DAE$_{\text{pH}, S_{\text{H}_2}}$).

### 3.3.1 ODE and DAE systems

When the states of a system are described only by ordinary differential equations, the system is said to be an ODE system. If the system is stiff, it is sometimes possible to rewrite some of the system equations in order to omit the fastest states. The rationale for this is that from the slower state’s point of view, the fast states can be considered instantaneous and possible to describe by algebraic equations. Such systems are normally referred to as differential algebraic equation (DAE) systems. By rewriting an ODE system to a DAE system, the stiffness can be decreased, allowing for explicit solvers to be used and for stochastic elements to be incorporated. The drawback is that the DAE system is only an approximation of the original system and the effect of this approximation must be considered and investigated for each specific simulation model.

### 3.3.2 Time constants in ADM1

As mentioned before, the ADM1 includes time constants in a wide range; from milliseconds for pH to weeks or months for the states describing various fractions of active biomass. Since most control actions affecting the anaerobic digester are fairly slow, it makes sense to investigate which fast states can be approximated by algebraic equations. In Batstone et al. (2002), it is suggested that the pH ($S_{\text{H}^+}$) state is calculated by algebraic equations. However, this will only partially solve the stiffness problem. There are other fast states and a closer investigation suggests that the state describing hydrogen ($S_{\text{H}_2}$) also needs to be approximated by an algebraic equation.

### 3.3.3 pH and $S_{\text{H}_2}$ solvers

As mentioned above, stiffness of the ADM1 can be reduced by approximating the differential equations of the pH and $S_{\text{H}_2}$ states by algebraic equations. Different solutions can be proposed to solve them.

An implicit algebraic equation for the pH calculation is given in (Batstone et al., 2002). It has been suggested to calculate the $S_{\text{H}^+}$ and, consequently, the pH from the sum of all charges, which is supposed to be zero. The obtained implicit algebraic equations are non-linear and therefore can be solved only by an iterative numerical method. In the MATLAB-Simulink implementation of BSM2, the Newton-Raphson method used in Volcke (2006) for calculation of the pH and equilibrium concentrations was implemented. By using this method the new value of the unknown state is calculated at each iteration step $k$ as:

$$S_{\text{H}^+, k+1} = S_{\text{H}^+, k} - \frac{E(S_{\text{H}^+, k})}{dE(S_{\text{H}^+, k})/dS_{\text{H}^+}|_{S_{\text{H}^+, k}}}$$

where $S_{\text{H}^+, k}$ is the value of the state obtained from the previous iteration step and $E(S_{\text{H}^+, k})$ is the value of the algebraic equation that has to be zero for the equilibrium, i.e.:

$$E(S_{\text{H}^+, k}) = S_{\text{cat}, k} + S_{\text{shi}, k} + S_{\text{H}^+, k} - S_{\text{HCO}_3, k} - S_{\text{SC}, k}/64 - S_{\text{NH}, k}/112 - S_{\text{PN}, k}/160 - S_{\text{VA}, k}/208 - K_W S_{\text{H}^+, k} - S_{\text{m}, k}$$

The gradient of the algebraic equation, $dE(S_{\text{H}^+, k})/dS_{\text{H}^+}|_{S_{\text{H}^+, k}}$, is also needed for calculation of the new state value. Since this expression is rather complicated, it is not presented here. The iteration is repeated as long as $E(S_{\text{H}^+, k})$ remains larger than the predefined tolerance value, which in the present case is set to $10^{-12}$. Normally only two or three iterations are required to solve the equation at each time step.

In the FORTRAN implementation, a one-dimension optimization routine (Golden section) is used (Pons et al., 1983) to find the minimum of
\[ O(S_{\text{h}^+,k}) = \left(S_{\text{cat}^+,k} + S_{\text{nh}4^+,k} + S_{\text{H}^+,k}\right) \left( S_{\text{no}3^-_{-k}} + \frac{S_{\text{ac}^-_{-k}}}{64} + \frac{S_{\text{pr}^-_{-k}}}{112} + \frac{S_{\text{bu}^-_{-k}}}{160} + \frac{S_{\text{va}^-_{-k}}}{208} + \frac{K_W}{S_{\text{H}_2^+,k}} + S_{\text{an}^-_{-k}} \right) \]  

(179)

in the pH = 0 to 14 interval with a relative tolerance of $10^{-7}$.

The differential equation for the $S_{\text{h}2}$ state, explicitly given in the ODE implementation of this report, can be approximated by an algebraic equation in a similar way as was the case for the $S_{\text{H}2}$ state, simply by setting its differential to zero (assuming fast dynamics). In the MATLAB-SIMULINK implementation, the iteration is carried out in the same way as for the $S_{\text{H}2}$ calculation, this time using:

\[ E(S_{\text{h}2,k}) = \frac{Q_{\text{ad}}}{V_{\text{ad,liq}}} \left( S_{\text{h}2,1} - S_{\text{h}2,k} \right) + \left( \frac{1}{f_{\text{h}2,ma}} \right) \rho_5_{\text{ad}} + \left( f_{\text{h}2,ma} \right) \rho_6_{\text{ad}} + \left( \frac{1}{f_{\text{h}a}} \right) \rho_7_{\text{ad}} + \left( \frac{1}{f_{\text{h}a}} \right) \rho_8_{\text{ad}} + \frac{1}{f_{\text{h}a}} \rho_9_{\text{ad}} + \left( \frac{1}{f_{\text{h}a}} \right) \rho_{10,\text{ad}} - \rho_{12,\text{ad}} - \rho_{T,8} \]

(180)

and the gradient of $E(S_{\text{h}2,k+1})$.

In the FORTRAN implementation, the minimum of

\[ O(S_{\text{h}2,k}) = \frac{Q_{\text{ad}}}{V_{\text{ad,liq}}} \left( S_{\text{h}2,1} - S_{\text{h}2,k} \right) + \left( \frac{1}{f_{\text{h}2,ma}} \right) \rho_5_{\text{ad}} + \left( f_{\text{h}2,ma} \right) \rho_6_{\text{ad}} + \left( \frac{1}{f_{\text{h}a}} \right) \rho_7_{\text{ad}} + \left( \frac{1}{f_{\text{h}a}} \right) \rho_8_{\text{ad}} + \frac{1}{f_{\text{h}a}} \rho_9_{\text{ad}} + \left( \frac{1}{f_{\text{h}a}} \right) \rho_{10,\text{ad}} - \rho_{12,\text{ad}} - \rho_{T,8} \]

(181)

is searched for in the interval $[0, 10^{-5}]$ with a relative tolerance of $10^{-7}$.

The expression of the gradient is quite complex. To obtain the gradients for the $S_{\text{H}2}$ and $S_{\text{h}2}$ equations, it is recommended that a tool for handling mathematics symbolically is used (e.g. Maple or Mathematica).

### 3.4. ADM1 benchmark model parameters

Tables 5 to 8 summarize the ADM1 model parameters used in BSM2.

### 4. MODELING OF THE PRIMARY CLARIFIER

The flow rate at the inlet of the primary clarifier ($Q_{\text{pi}}$) is given by:

\[ Q_{\text{pi}} = Q_{\text{plant}} + Q_{\text{to}} + Q_{\text{st,bypass}} + Q_{\text{st,out}} \]

(182)

where $Q_{\text{plant}}$ is the flow rate of raw wastewater which will be treated in the plant, $Q_{\text{to}}$ is the overflow rate from the thickener, $Q_{\text{st,bypass}}$ is the flow rate bypassed from the sludge tank and $Q_{\text{st,out}}$ is the flow rate from the sludge tank.

If the flow rate into the BSM2 system $Q_{\text{i}}$ is larger than 60,000 m$^3$.d$^{-1}$

\[ Q_{\text{bypass}} = Q_{\text{i}} - 60,000 \]

(183)

\[ Q_{\text{plant}} = 60,000 \]

(184)

where $Q_{\text{bypass}}$ the flow rate of raw wastewater which is bypassed.

For any influent fraction as well as temperature, the following equation holds:

\[ Z_{\text{pi}} \cdot Q_{\text{pi}} = Z_{\text{plant}} \cdot Q_{\text{plant}} + Z_{\text{to}} \cdot Q_{\text{to}} + Z_{\text{st,bypass}} \cdot Q_{\text{st,bypass}} + Z_{\text{st,out}} \cdot Q_{\text{st,out}} \]

(185)

with

\[ Z_{\text{plant}} = Z_i \]

(186)

The model proposed by Otterpohl and Freund (1992) and Otterpohl et al. (1994) can be described by one completely mixed tank and a separation of the effluent of the tank into a primary clarifier effluent and a primary sludge (Figure 4). The model description originally does not consider primary sludge. The formulas regarding primary sludge are added by simple mass balance considerations. Table 9 summarizes the variables and their assumed values.
Figure 4: Primary clarifier model assumption
Table 5: Stoichiometric parameter values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
<th>Process(es)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{SI,xc}$</td>
<td>0.1</td>
<td>-</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$f_{XI,xc}$</td>
<td>0.2</td>
<td>-</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$f_{ch,xc}$</td>
<td>0.2</td>
<td>-</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$f_{pr,xc}$</td>
<td>0.2</td>
<td>-</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$f_{li,xc}$</td>
<td>0.3</td>
<td>-</td>
<td>1</td>
<td>To maintain N balance for disintegration</td>
</tr>
<tr>
<td>$N_{ec}$</td>
<td>0.0376/14</td>
<td>c</td>
<td>1, 13-19</td>
<td>Note: $1-f_{ch,xc}-f_{pr,xc}-f_{SI,xc}-f_{li,xc}-f_{XI,xc} = 0$</td>
</tr>
<tr>
<td>$N_f$</td>
<td>0.06/14</td>
<td>kmole N.(kg COD)$^{-1}$</td>
<td>1</td>
<td>6% on weight basis in the ASM section</td>
</tr>
<tr>
<td>$N_{ea}$</td>
<td>0.007</td>
<td>kmole N.(kg COD)$^{-1}$</td>
<td>1, 6</td>
<td></td>
</tr>
<tr>
<td>$C_{xc}$</td>
<td>0.02786</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>13-19</td>
<td></td>
</tr>
<tr>
<td>$C_{fl}$</td>
<td>0.03</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>1</td>
<td>$C_{12}$ in Eq. (126)</td>
</tr>
<tr>
<td>$C_{ch}$</td>
<td>0.0313</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>1, 2</td>
<td>$C_{14}$ in Eq. (126)</td>
</tr>
<tr>
<td>$C_{pr}$</td>
<td>0.03</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>1, 3</td>
<td>$C_{15}$ in Eq. (126)</td>
</tr>
<tr>
<td>$C_{li}$</td>
<td>0.022</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>1, 4</td>
<td>$C_{16}$ in Eq. (126)</td>
</tr>
<tr>
<td>$C_{xl}$</td>
<td>0.03</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>1</td>
<td>$C_{24}$ in Eq. (126)</td>
</tr>
<tr>
<td>$C_{su}$</td>
<td>0.0313</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>2, 5</td>
<td>$C_{1}$ in Eq. (126)</td>
</tr>
<tr>
<td>$C_{aa}$</td>
<td>0.03</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>3, 6</td>
<td>$C_{2}$ in Eq. (126)</td>
</tr>
<tr>
<td>$N_{bac}$</td>
<td>0.08/14</td>
<td>kmole N.(kg COD)$^{-1}$</td>
<td>5-19</td>
<td>8% on weight basis in the ASM section</td>
</tr>
<tr>
<td>$C_{su}$</td>
<td>0.025</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>5, 6, 9</td>
<td>$C_{3}$ in Eq. (126)</td>
</tr>
<tr>
<td>$C_{pro}$</td>
<td>0.0268</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>5, 6, 8, 10</td>
<td>$C_{6}$ in Eq. (126)</td>
</tr>
<tr>
<td>$C_{ac}$</td>
<td>0.0313</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>5-11</td>
<td>$C_{7}$ in Eq. (126)</td>
</tr>
<tr>
<td>$C_{bac}$</td>
<td>0.0313</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>5-19</td>
<td>$C_{17-23}$ in Eq. (126)</td>
</tr>
<tr>
<td>$Y_{su}$</td>
<td>0.06</td>
<td>kmole COD$_X$. (kg COD$_S$)$^{-1}$</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>$Y_{fa}$</td>
<td>0.06</td>
<td>kmole COD$_X$. (kg COD$_S$)$^{-1}$</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>$Y_{c4}$</td>
<td>0.06</td>
<td>kmole COD$_X$. (kg COD$_S$)$^{-1}$</td>
<td>8, 9</td>
<td></td>
</tr>
<tr>
<td>$Y_{pro}$</td>
<td>0.04</td>
<td>kmole COD$_X$. (kg COD$_S$)$^{-1}$</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>$C_{ch4}$</td>
<td>0.0156</td>
<td>kmole C.(kg COD)$^{-1}$</td>
<td>11, 12</td>
<td>$C_{0}$ in Eq. (10)</td>
</tr>
<tr>
<td>$Y_{ac}$</td>
<td>0.05</td>
<td>kmole COD$_X$. (kg COD$_S$)$^{-1}$</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>$Y_{a2}$</td>
<td>0.06</td>
<td>kmole COD$_X$. (kg COD$_S$)$^{-1}$</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

Note that $C_{ba}$ and $C_{IN}$, i.e. $C_8$ and $C_{11}$, are equal to zero in Eq. 126.
Table 6: Biochemical parameter values. The unit M is defined as kmole.m\(^{-3}\) according to Batstone et al. (2002)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
<th>Process(es)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k_{ds})</td>
<td>0.5</td>
<td>d(^{-1})</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(k_{hyd,chl})</td>
<td>10</td>
<td>d(^{-1})</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>(k_{hyd,pr})</td>
<td>10</td>
<td>d(^{-1})</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>(k_{hyd,li})</td>
<td>10</td>
<td>d(^{-1})</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>(K_{S,IN})</td>
<td>1.10(^{-4})</td>
<td>M</td>
<td>5-12</td>
<td></td>
</tr>
<tr>
<td>(k_{m, su})</td>
<td>30</td>
<td>d(^{-1})</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>(K_{S, su})</td>
<td>0.5 kg COD.m(^{-3})</td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>(pH_{UL, aa})</td>
<td>5.5</td>
<td>-</td>
<td>5-10</td>
<td>in (I_{S, 10})</td>
</tr>
<tr>
<td>(pH_{LL, aa})</td>
<td>4</td>
<td>-</td>
<td>5-10</td>
<td>in (I_{S, 10})</td>
</tr>
<tr>
<td>(k_{m, aa})</td>
<td>50</td>
<td>d(^{-1})</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>(K_{S, aa})</td>
<td>0.3 kg COD.m(^{-3})</td>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>(k_{m, fa})</td>
<td>6</td>
<td>d(^{-1})</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>(K_{S, fa})</td>
<td>0.4 kg COD.m(^{-3})</td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>(K_{a, fa})</td>
<td>5.10(^{-6}) kg COD.m(^{-3})</td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>(k_{m, c4})</td>
<td>20</td>
<td>d(^{-1})</td>
<td>8, 9</td>
<td></td>
</tr>
<tr>
<td>(K_{S, c4})</td>
<td>0.2 kg COD.m(^{-3})</td>
<td></td>
<td>8, 9</td>
<td></td>
</tr>
<tr>
<td>(K_{12,2, c4})</td>
<td>1.10(^{-5}) kg COD.m(^{-3})</td>
<td></td>
<td>8, 9</td>
<td></td>
</tr>
<tr>
<td>(k_{m, pro})</td>
<td>13</td>
<td>d(^{-1})</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>(K_{S, pro})</td>
<td>0.1 kg COD.m(^{-3})</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>(K_{12,2, pro})</td>
<td>3.5(^{-6}) kg COD.m(^{-3})</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>(k_{m, ac})</td>
<td>8</td>
<td>d(^{-1})</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>(K_{S, ac})</td>
<td>0.15 kg COD.m(^{-3})</td>
<td></td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>(K_{12, NI3})</td>
<td>0.0018 M</td>
<td></td>
<td>11</td>
<td>in (I_{11})</td>
</tr>
<tr>
<td>(pH_{UL, ac})</td>
<td>7</td>
<td>-</td>
<td>11</td>
<td>in (I_{11})</td>
</tr>
<tr>
<td>(pH_{LL, ac})</td>
<td>6</td>
<td>-</td>
<td>11</td>
<td>in (I_{11})</td>
</tr>
<tr>
<td>(k_{a, h2})</td>
<td>35</td>
<td>d(^{-1})</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>(K_{S, h2})</td>
<td>7.10(^{-6}) kg COD.m(^{-3})</td>
<td></td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>(pH_{UL, h2})</td>
<td>6</td>
<td>-</td>
<td>12</td>
<td>in (I_{12})</td>
</tr>
<tr>
<td>(pH_{LL, h2})</td>
<td>5</td>
<td>-</td>
<td>12</td>
<td>in (I_{12})</td>
</tr>
<tr>
<td>(k_{ac, Xsu})</td>
<td>0.02</td>
<td>d(^{-1})</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>(k_{ac, Xaa})</td>
<td>0.02</td>
<td>d(^{-1})</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>(k_{ac, Xfa})</td>
<td>0.02</td>
<td>d(^{-1})</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>(k_{ac, Xc4})</td>
<td>0.02</td>
<td>d(^{-1})</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>(k_{ac, Xpro})</td>
<td>0.02</td>
<td>d(^{-1})</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>(k_{ac, Xac})</td>
<td>0.02</td>
<td>d(^{-1})</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>(k_{ac, Xh2})</td>
<td>0.02</td>
<td>d(^{-1})</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>
Table 7: Physiochemical parameter values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>0.083145</td>
<td>Bar.M⁻¹.K⁻¹</td>
<td></td>
</tr>
<tr>
<td>$T_{base}$</td>
<td>298.15</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>$T_{ad}$</td>
<td>308.15</td>
<td>K</td>
<td>= 35°C</td>
</tr>
<tr>
<td>$K_W$</td>
<td>$10^{-14.4}$</td>
<td>M</td>
<td>≈ 2.08·10⁻¹⁴</td>
</tr>
<tr>
<td>$K_{a,va}$</td>
<td>$10^{4.86}$</td>
<td>M</td>
<td>≈ 1.38·10⁵</td>
</tr>
<tr>
<td>$K_{a,bu}$</td>
<td>$10^{4.82}$</td>
<td>M</td>
<td>≈ 1.5·10⁵</td>
</tr>
<tr>
<td>$K_{a,pro}$</td>
<td>$10^{4.88}$</td>
<td>M</td>
<td>≈ 1.32·10⁵</td>
</tr>
<tr>
<td>$K_{a,ac}$</td>
<td>$10^{4.76}$</td>
<td>M</td>
<td>≈ 1.74·10⁵</td>
</tr>
<tr>
<td>$K_{a,IN}$</td>
<td>$10^{6.13}$</td>
<td>M</td>
<td>≈ 1.11·10⁹</td>
</tr>
<tr>
<td>$K_{a,co2}$</td>
<td>$10^{9.25}$</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>$k_{A,Bva}$</td>
<td>$1.10^{10}$</td>
<td>M⁻¹.d⁻¹</td>
<td>Set to be at least three orders of magnitude higher than the fastest time constant of the system</td>
</tr>
<tr>
<td>$k_{A,Bbu}$</td>
<td>$1.10^{10}$</td>
<td>M⁻¹.d⁻¹</td>
<td></td>
</tr>
<tr>
<td>$k_{A,Bpro}$</td>
<td>$1.10^{10}$</td>
<td>M⁻¹.d⁻¹</td>
<td></td>
</tr>
<tr>
<td>$k_{A,Bac}$</td>
<td>$1.10^{10}$</td>
<td>M⁻¹.d⁻¹</td>
<td></td>
</tr>
<tr>
<td>$k_{A,Bco2}$</td>
<td>$1.10^{10}$</td>
<td>M⁻¹.d⁻¹</td>
<td></td>
</tr>
<tr>
<td>$k_{A,BIN}$</td>
<td>$1.10^{10}$</td>
<td>M⁻¹.d⁻¹</td>
<td></td>
</tr>
<tr>
<td>$P_{atm}$</td>
<td>1.013</td>
<td>bar</td>
<td></td>
</tr>
<tr>
<td>$p_{gas,h2o}$</td>
<td>0.0557</td>
<td>bar</td>
<td></td>
</tr>
<tr>
<td>$K_a$</td>
<td>5.10⁴</td>
<td>m³.d⁻¹.bar⁻¹</td>
<td></td>
</tr>
<tr>
<td>$K_{H,co2}$</td>
<td>200</td>
<td>d⁻¹</td>
<td></td>
</tr>
<tr>
<td>$K_{H,ch4}$</td>
<td></td>
<td>M⁻¹.bar⁻¹</td>
<td>≈ 0.0271</td>
</tr>
<tr>
<td>$K_{H,h2}$</td>
<td></td>
<td>M⁻¹.bar⁻¹</td>
<td>≈ 0.00116</td>
</tr>
<tr>
<td>$K_{H,h2}$</td>
<td></td>
<td>M⁻¹.bar⁻¹</td>
<td>≈ 7.38·10⁻⁴</td>
</tr>
</tbody>
</table>

Table 8: Physical parameter values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{ad,liq}$</td>
<td>3400</td>
<td>m³</td>
</tr>
<tr>
<td>$V_{ad,gas}$</td>
<td>100</td>
<td>m³</td>
</tr>
</tbody>
</table>

For the evaluation of the concentrations within the tank simple CSTR formulas holds:

(187)

Given the "mean" fraction of particulate COD from overall COD:

(188)

the COD removal efficiency is calculated as:
with the hydraulic retention time \( t_h \) (d) and a correction factor \( f_{corr} \). The parameter \( f_X \) is used as a constant parameter describing the mean value of the COD particulate to COD total ratio. For the calculation of the "mean" hydraulic retention time a first order low pass is used to calculate the mean influent flow rate:

\[
(189) \text{with the "smoothing" time constant } t_m = 3/24 \text{ d. The hydraulic retention time is then calculated as }
\]

\[
(190) \text{The removal efficiency with respect to the particulate COD is written as: }
\]

\[
(192) \text{as the soluble COD is not affected. With this factor, the effluent concentrations calculates as: }
\]

\[
(193) \text{with the factor }
\]

\[
(194) \text{where the factor } f_{sx,k} \text{ is 0 for all soluble fractions, is 1 for all particulate fractions except } X_S \text{ and is } f_{XS} \text{ for } X_S.
\]

The primary sludge concentration follows from the mass balance:

\[
(195)
\]

For the primary sludge flow rate, a proportional flow rate to the influent flow rate is used:

\[
Q_{pu}(t) = f_{PS} q(t) \text{ with } f_{PS} = 0.007 \tag{196}
\]

Table 9: Main primary settler parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{corr} )</td>
<td>correction factor removal efficiency (tuning parameter)</td>
<td>0.65</td>
</tr>
<tr>
<td>( f_k )</td>
<td>effluent concentration factor</td>
<td>-</td>
</tr>
<tr>
<td>( f_{PS} )</td>
<td>ratio of primary sludge flow rate to the influent flow rate</td>
<td>0.007</td>
</tr>
<tr>
<td>( f_{SX,k} )</td>
<td>structural factor defining the fraction of particulate matter of each fraction</td>
<td>-</td>
</tr>
<tr>
<td>( f_X )</td>
<td>ratio of particulate COD from total COD (mean value)</td>
<td>0.85</td>
</tr>
<tr>
<td>( f_{XS} )</td>
<td>particulate fraction of XS</td>
<td>0.5</td>
</tr>
<tr>
<td>( Z_{pc,k}(t) )</td>
<td>concentrations in the mixing tank (CSTR)</td>
<td></td>
</tr>
<tr>
<td>( Z_{pl,k}(t) )</td>
<td>influent concentrations</td>
<td></td>
</tr>
<tr>
<td>( Z_{ps,k}(t) )</td>
<td>effluent concentrations</td>
<td></td>
</tr>
<tr>
<td>( Z_{pu,k}(t) )</td>
<td>primary sludge concentrations</td>
<td></td>
</tr>
<tr>
<td>( \eta_{COD}(t) )</td>
<td>COD removal efficiency (total)</td>
<td>%</td>
</tr>
<tr>
<td>( \eta_{CODp}(t) )</td>
<td>COD removal efficiency (particulate)</td>
<td>%</td>
</tr>
<tr>
<td>( Q_{in}(t) )</td>
<td>influent flow rate</td>
<td>m³.d⁻¹</td>
</tr>
<tr>
<td>( Q_{in}(t) )</td>
<td>mean influent flow rate</td>
<td>m³.d⁻¹</td>
</tr>
<tr>
<td>( Q_{ps}(t) )</td>
<td>primary sludge flow rate</td>
<td>m³.d⁻¹</td>
</tr>
<tr>
<td>( Q_{pu}(t) )</td>
<td>primary settler overflow, to the activated sludge section</td>
<td>m³.d⁻¹</td>
</tr>
<tr>
<td>( t_h(t) )</td>
<td>hydraulic retention time</td>
<td>d</td>
</tr>
<tr>
<td>( t_m )</td>
<td>smoothing time constant for the ( q_m ) calculation</td>
<td>d</td>
</tr>
<tr>
<td>( V_{pc} )</td>
<td>volume of primary clarifier (900 m³)</td>
<td>m³</td>
</tr>
</tbody>
</table>
5. MODELING OF THE THICKENER AND DEWATERING UNIT

For simplicity, these two units are supposed to have an ideal behavior and have no volume.

5.1. Thickener

The thickener thickens the sludge wasted from the bottom of the clarifier prior to its mixing with the primary sludge from the primary clarifier and its digestion (Figure 5).

\[
\text{inlet} \quad Q_{\text{w},u} \quad Z_u \quad \text{overflow} \quad Q_{\text{to},z_{\text{to}}} \quad \text{underflow} \quad Q_{\text{tu},z_{\text{tu}}}
\]

Figure 5: Thickener model assumption

The suspended solid concentration at the inlet of the thickener is \( TSS_{\text{sc},1} \) (Eq. 67). The percentage of suspended solids in the underflow of the thickener is \( \rho_{\text{thick}} \) (= 7%). The percentage of suspended solids removed is \( TSS_{\text{rem}} \) (= 98%). The thickening factor \( f_{\text{thick}} \) is calculated as:

\[
(197)
\]

Let

\[
(198)
\]

Then, the thinning factor is equal to:

\[
(199)
\]

If the thickening factor is larger than 1, the variables in the underflow are calculated as:

For any particulate fraction:

For any soluble fraction and temperature:

Underflow rate:

\[
(200)
\]

The variables in the overflow are calculated as:

For any particulate fraction:

For any soluble fraction and temperature:

Overflow rate:

\[
(201)
\]

If the thickening factor is lower than 1, there is an error.

5.2. Dewatering unit

The dewatering unit thickens the digested sludge from the digester (flow rate \( Q_{\text{ad}} \)). The reject water is recycled to the inlet of the primary settler (Figure 6). The model of the dewatering unit is similar to the model of the thickener.

\[
\text{inlet} \quad Q_{\text{ad},z_{\text{ad}}} \quad \text{overflow} \quad Q_{\text{do},z_{\text{do}}} \quad \text{underflow} \quad Q_{\text{du},z_{\text{du}}}
\]

Figure 6: Dewatering unit model assumption
The suspended solid concentration at the inlet of the dewatering unit is:

\[ X_{S,ad}, X_{P,ad}, X_{I,H,ad}, X_{B,A,ad} \]

where \( X_{S,ad}, X_{P,ad}, X_{I,H,ad}, X_{B,A,ad} \) are outputs from the ADM-ASM interface (see §§7).

The percentage of suspended solids in the underflow of the thickener is \( p_{\text{dewat}} = 28\% \). The percentage of suspended solids removed is \( TSS_{\text{rem}} = 98\% \). The dewatering factor \( (f_{\text{dewat}}) \) is calculated as:

\[ (203) \]

Let \[ (204) \]

Then the dewatering factor is equal to:

\[ (205) \]

If the dewatering factor is larger than 1, the variables in the underflow are calculated as:

For any particulate fraction:

\[ Z_{\text{du}} = Z_{ad} \cdot f_{\text{dewat}} \]

For any soluble fraction and temperature:

\[ (206) \]

Underflow rate:

\[ (207) \]

The variables in the overflow are calculated as:

For any particulate fraction:

\[ (208) \]

For any soluble fraction and temperature:

\[ (209) \]

Overflow rate:

\[ (210) \]

If the dewatering factor is lower than 1, there is an error.

6. MODELING OF THE REJECT WATER STORAGE TANK

A storage tank has been set on the recycle line from the dewatering unit to the inlet of the primary clarifier, after the ADM to ASM interface. Its behaviour depends upon the flow rate from the dewatering unit, the available storage volume and the fate of the stored reject water. The reject water cannot be stored in the tank when the tank is full. In such a case, the reject water has to be recycled directly to the inlet of the primary clarifier until the storage tank has been emptied and can receive water again. Furthermore, a limit has been set for the reject water outflow rate from the tank \( (Q_{st,set}) \).

6.1 General definitions

The various notations used for the model are given in Table 10, in agreement with Figure 7. The height tolerance is used to define the minimal and maximal volumes that can be handled in the tank.

Table 10: Reject water storage tank variables

<table>
<thead>
<tr>
<th>Definition</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total tank volume</td>
<td>( V_{st,\text{total}} )</td>
</tr>
<tr>
<td>Liquid volume in tank</td>
<td>( V_{st} )</td>
</tr>
<tr>
<td>Maximal liquid volume in tank</td>
<td>( V_{st,\text{max}} )</td>
</tr>
<tr>
<td>Minimal liquid volume in tank</td>
<td>( V_{st,\text{min}} )</td>
</tr>
<tr>
<td>Height</td>
<td>( H_{st} )</td>
</tr>
<tr>
<td>Area</td>
<td>( A_{st} )</td>
</tr>
<tr>
<td>Liquid flow rate from the dewatering unit overflow</td>
<td>( Q_{do} )</td>
</tr>
<tr>
<td>Liquid flow rate at the inlet of the storage tank</td>
<td>( Q_{st,in} )</td>
</tr>
<tr>
<td>Liquid flow rate at the outlet of the storage tank</td>
<td>( Q_{st,out} )</td>
</tr>
<tr>
<td>Liquid flow rate in the bypass of the storage tank</td>
<td>( Q_{st,bypass} )</td>
</tr>
<tr>
<td>Liquid flow rate setpoint at the outlet of the storage tank</td>
<td>( Q_{st,set} )</td>
</tr>
</tbody>
</table>
6.2. Storage tank behaviour

6.2.1. Variation of liquid volume

The following relations define the various flow rates needed for the mass balances.

When the liquid volume in the storage tank is between the minimal and the maximal values, the total reject water flow rate from the dewatering unit is admitted to the storage tank and the outflow rate is equal to the set point. No reject water is bypassed:

\[
\text{If } (V_{st} \leq V_{st,max} \text{ and } V_{st} \geq V_{st,min}) \text{ then} \\
Q_{st,in} = Q_{do} \\
Q_{st,out} = Q_{st,set} \\
Q_{st,bypass} = 0 \\
\]

(208)

When the liquid volume in the storage tank is larger than the maximal value and the reject water flow rate from the dewatering unit is larger than the outflow rate set point, the reject water from the dewatering unit is bypassed to the primary clarifier and the outflow rate from the storage tank is set to 0:

\[
\text{If } (V_{st} \geq V_{st,max} \text{ and } Q_{do} > Q_{st,set}) \text{ then} \\
Q_{st,in} = 0 \\
Q_{st,out} = 0 \\
Q_{st,bypass} = Q_{do} \\
\]

(209)

When the liquid volume in the storage tank is larger than the maximal value and the reject water flow rate from the dewatering unit is smaller than the outflow rate set point, the reject water from the dewatering unit is fed to the storage tank. The outflow rate is equal to its set point.

\[
\text{If } (V_{st} \geq V_{st,max} \text{ and } Q_{do} \leq Q_{st,set}) \text{ then} \\
Q_{st,in} = Q_{do} \\
Q_{st,out} = Q_{st,set} \\
Q_{st,bypass} = 0 \\
\]

(210)

When the liquid volume in the storage tank is smaller than the minimal value, the reject water from the dewatering unit is fed into the storage tank and the outflow rate is set to 0:

\[
\text{If } (V_{st} \leq V_{st,min}) \text{ then} \\
Q_{st,in} = Q_{do} \\
Q_{st,out} = 0 \\
Q_{st,bypass} = 0 \\
\]

(211)

The variations of the storage tank volume are given as:

\[
\frac{dV_{st}}{dt} = Q_{st,in} - Q_{st,out} \\
\]

(212)
6.2.2. Variation of a concentration

The tank is assumed to be non-reactive. Given a volume $V_{st}$, an influent flow rate $Q_{st,i}$, an influent concentration $C_{do}$ and an effluent flow rate $Q_{st,out}$, the corresponding concentration $C_{st}$ is calculated by integration of Equation (213).

\[
\frac{dC_{st}}{dt} = \frac{1}{V_{st}} (Q_{st,i} C_{do} - Q_{st,out} C_{st}) = \frac{Q_{st,i}}{V_{st}} (C_{do} - C_{st})
\]  

(213)

6.3. Implementation

The implementation is performed with the following parameters:

- $V_{st,\text{total}} = 160$ m$^3$
- $V_{stank,\text{max}} = 0.9 \cdot V_{st,\text{total}}$
- $V_{stank,\text{min}} = 0.1 \cdot V_{st,\text{total}}$
- $Q_{st,\text{set}} \leq 1500$ m$^3$.d$^{-1}$

For initialization, the tank is half full ($V_{st} = 0.5 \cdot V_{st,\text{total}}$) when a sludge tank is considered in the system. It is full when a sludge tank is not considered. In such a case the flow from the dewatering unit overflow is bypassed directly to the primary clarifier.

7. ASM/ADM and ADM/ASM interfaces

The purpose of the ASM/ADM interface is to transform the state variables from the activated sludge section corresponding to the ASM1 formulation into state variables usable in the anaerobic digester corresponding to the ADM1 formulation. The opposite function is assigned to the ADM/ASM interface.

The ASM/ADM interface is applied after mixing the primary sludge from the underflow of the primary clarifier with the thickened secondary sludge wasted from the secondary clarifier. This means that for any ASM1 state variable ($Z_{as}$) to be transformed into an ADM1 state variable ($Z_{ad}$):

\[
Z_{as} = \frac{Z_{pu} \cdot Q_{pu} + Z_{tu} \cdot Q_{tu}}{Q_{pu} + Q_{tu}} = \frac{Z_{pu} \cdot Q_{pu} + Z_{tu} \cdot Q_{tu}}{Q_{ad}}
\]  

(214)

The ADM/ASM interface is applied to transform any $Z_{ad}$ state variable at the outlet of the digester (except pH and temperature) into a $Z_{as}$ state variable, used in the dewatering unit. pH of ASM/ADM and ADM/ASM interfaces are identical to current (at every time step) pH in the digester. The temperature within the interfaces is equal to the digester temperature, i.e. 35°C. The temperature at the output of interface ADM/ASM at time $t$ is equal to the temperature at the inlet of interface ASM/ADM at time $t$.

Both interfaces are built from sets of rules (Nopens et al., 2009). To help the developer, the MATLAB and FORTRAN codes are given in Appendix A2. The following equations guarantee the charge balance (with $T_{base} = 298.15$ K (25°C) and $T_{ad} = T_{base} + 10$ K (35°C)):

\[
\alpha_{ac}^{\text{ch}} = \alpha_{bu}^{\text{ch}} = \alpha_{va}^{\text{ch}} = \frac{-1}{C_i} \frac{1 + 10^{pK_a - pH_{ad}}}{1 + 10^{pK_a - pH_{ad}}}
\]  

(215)

with $pK_a = 4.76$, 4.88, 4.82, 4.86 resp. $(T=25^\circ C)$ with $C_i$ respectively equal to 64, 112, 160, 208

\[
\alpha_{IN}^{\text{ch}} = \frac{10^{pK_a - pH_{ad}}}{1 + 10^{pK_a - pH_{ad}}}
\]  

(216)

with $pK_a = 9.25 - \log_{10} \left\{ \exp \left[ \frac{51965}{100 \cdot R} \left( T_{base} - T_{ad} \right) \right] \right\}$

(217)

\[
\alpha_{IC}^{\text{ch}} = \frac{-1}{1 + 10^{pK_a - pH_{ad}}}
\]  

(218)

with $pK_a = 6.35 - \log_{10} \left\{ \exp \left[ \frac{7646}{100 \cdot R} \left( T_{base} - T_{ad} \right) \right] \right\}$
(conversion from AS unit g N.m\(^{-3}\) into AD unit kmole N.m\(^{-3}\)) \(220\)

(conversion from AS unit g N.m\(^{-3}\) into AD unit kmole N.m\(^{-3}\)) \(221\)

(conversion from AS unit mole HCO\(_3\).m\(^{-3}\) into interface unit kmole HCO\(_3\).m\(^{-3}\)) \(222\)

Note that \(R = 0.083145\) bar.m\(^3\). K\(^{-1}\).kmol\(^{-1}\)

For the ASM/ADM interface:

It can be noted that based on the definition of the BSM TG ASM/ADM interface, the variables \(S_{\text{ac}}, S_{\text{pro}}, S_{\text{bu}}\) and \(S_{\text{va}}\) will always be zero, but they are added here to provide a general description. Special care must be taken to the \(S_{\text{NO}}\) concentration of the ASM input. The ASM/ADM interface immediately converts any influent nitrate to nitrogen gas (with an associated COD loss), i.e. a direct denitrification. However, in the charge balancing equation (the \(S_{\text{IC}}\) equation above) the true influent nitrate concentration from the ASM (prior to interface internal denitrification) should be used.

For the ADM/ASM interface:

It can be noted that based on the definition of the BSM TG ADM/ASM interface the variable \(S_{\text{NO}}\) will always be zero, but it is added here to provide a general description.

As the ADM1 fulfils absolutely the charge balance via \(S_{\text{an}}\) and \(S_{\text{cat}}\), it is necessary to calculate \(S_{\text{an}}\) and \(S_{\text{cat}}\) to have a closed charge balance. The ADM1 does not calculate all ions and, thus, the charge balance is not absolutely closed, but of course all processes of the ADM1 respect the charge balance relatively.

To calculate \(S_{\text{an}}\) and \(S_{\text{cat}}\) of the influent to ADM1, the full charge balance should be used, i.e. it should include also the \(\text{OH}^-\) and \(\text{H}^+\) ions:

\[
(226)
\]

with \[
\text{and}
\]

leading to:

\[
(227)
\]

If the result is greater than zero, it is assumed:

\[
(228)
\]

and

\[
(229)
\]

If the result is smaller than zero,

\[
(230)
\]

and

\[
(231)
\]

Note that the above \(S_{\text{an}}\) and \(S_{\text{cat}}\) equations are only relevant in the ASM/ADM interface.
8. INFLUENT DATA

In BSM2 the evaluation of the plant performance is done on a full year (364 days). A dynamic stabilization period is required before evaluation. The file starts on 63 days before Jan 1st. The first 245 days (i.e. $63 + 182$) serve for the stabilization under dynamic conditions.

The 609 days ($63 + 182 + 364 = 609$ days) dynamic file for BSM2 can be loaded from the CD. The structure of the file is as follows: time, $S_I$, $S_{SS}$, $X_I$, $X_{SS}$, $X_{HB}$, $X_{B, H}$, $X_{B, A}$, $X_P$, $S_O$, $S_{NO}$, $S_{NH}$, $S_{NDD}$, $X_{NDD}$, $S_{ALK}$, $TSS$, $Q_i$, $T$ and five dummy states for further extension. Some of these variables are plotted in Figure 8. $TSS$ gives the total suspended solids in the influent according to:

$$TSS = 0.75 \left( X_S + X_I + X_{B, H} + X_{B, A} + X_P \right)$$  \quad (232)

![Figure 8: Some of the influent characteristics with respect to time: $S_I$, $S_{SS}$, $X_I$, $X_{SS}$, $S_{NH}$, $T$ and $Q_i$](image)
9. INITIALIZATION

Any initial values can be selected by the user. However, a 1000-day period of stabilization in closed-loop using constant inputs (Table 11) with no noise on the measurements has to be completed before using the influent file (609 days). Noise on measurements should be used with the dynamic files.

Table 11: Influent values for the stabilization period

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{I,stab}$</td>
<td>27.22619062</td>
<td>g COD.m$^{-3}$</td>
</tr>
<tr>
<td>$S_{S,stab}$</td>
<td>58.17618568</td>
<td>g COD.m$^{-3}$</td>
</tr>
<tr>
<td>$X_{I,stab}$</td>
<td>92.49900106</td>
<td>g COD.m$^{-3}$</td>
</tr>
<tr>
<td>$X_{S,stab}$</td>
<td>363.943473</td>
<td>g COD.m$^{-3}$</td>
</tr>
<tr>
<td>$X_{B,H,stab}$</td>
<td>50.68328815</td>
<td>g COD.m$^{-3}$</td>
</tr>
<tr>
<td>$X_{B,A,stab}$</td>
<td>0</td>
<td>g COD.m$^{-3}$</td>
</tr>
<tr>
<td>$X_{P,stab}$</td>
<td>0</td>
<td>g COD.m$^{-3}$</td>
</tr>
<tr>
<td>$S_{O,stab}$</td>
<td>0</td>
<td>g (-COD)/m$^3$</td>
</tr>
<tr>
<td>$S_{NO,stab}$</td>
<td>0</td>
<td>g N.m$^{-3}$</td>
</tr>
<tr>
<td>$S_{NH,stab}$</td>
<td>23.85946563</td>
<td>g N.m$^{-3}$</td>
</tr>
<tr>
<td>$S_{ND,stab}$</td>
<td>5.651606031</td>
<td>g N.m$^{-3}$</td>
</tr>
<tr>
<td>$X_{ND,stab}$</td>
<td>16.12981606</td>
<td>g N.m$^{-3}$</td>
</tr>
<tr>
<td>$S_{ALK,stab}$</td>
<td>7</td>
<td>mole.m$^{-3}$</td>
</tr>
<tr>
<td>$TSS_{stab}$</td>
<td>380.3443217</td>
<td>g.m$^{-3}$</td>
</tr>
<tr>
<td>$Q_{i,stab}$</td>
<td>20648.36121</td>
<td>m$^3$.d$^{-1}$</td>
</tr>
<tr>
<td>$T_{stab}$</td>
<td>14.85808006</td>
<td>°C</td>
</tr>
</tbody>
</table>

The following operation conditions are applied during the stabilization period:
- Internal recycle flow rate $Q_{int} = 61,944$ m$^3$.d$^{-1}$
- External recycle flowrate $Q_{r} = 20,648$ m$^3$.d$^{-1}$
- Wastage flowrate $Q_{w} = 300$ m$^3$.d$^{-1}$
- External carbon flowrate in 1st anoxic reactor $Q_{EC1} = 2$ m$^3$.d$^{-1}$
- External carbon concentration: 400,000 g COD.m$^{-3}$
- Oxygen transfer coefficients: $K_{L}a_3 = 120$ d$^{-1}$ and $K_{L}a_5 = 60$ d$^{-1}$
- Flow rate from the reject water storage tank $Q_{st, set} = 0$.  

Appendix A3 summarizes the steady-state results obtained under these conditions.

10. EVALUATION

For evaluation of the simulation results over a fixed period of time ($t_{obs} = t_f - t_0$), average values are to be calculated as follows (The user should be aware that all the integrals for performance assessment are calculated by rectangular integration with a time step of 15 min):

- Flow rate (m$^3$.d$^{-1}$):

  \[ Q_{int} = 61,944 \text{ m}^3\text{.d}^{-1} \tag{233} \]

- Concentration for compound $Z_k$ (mass.m$^{-3}$) in flow $Q$ must be flow proportional:

  \[ C_{Z_k} = \frac{Q}{Q_{int}} C_{Z_k} \tag{234} \]
11. SET-UP OF A DEFAULT CONTROLLER

A default controller is proposed so the closed-loop simulation and the implementation of the evaluation criteria can be tested before the user implements his/her own control strategy. The primary control objective for the default strategies is to maintain the dissolved oxygen concentration in the fifth compartment at a predetermined set point value (2 g (-COD).m\(^{-3}\)) by manipulation of the oxygen transfer coefficient in the fourth reactor in such a way that: \(K_{L}a_3 = K_{L}a_4; K_{L}a_5 = K_{L}a_4/2\). Actuators models are used for the three oxygen transfer coefficients. The modeling principles of the sensors are given in Section 13 of this document. Furthermore, external carbon addition rate is 2.0 m\(^3\).d\(^{-1}\). Finally, two different wastage flow rates are imposed dependent on time of the year (Table 12), assuming day 0 is at the start of the 609 days period. For this reason a first-order filter (time constant = 0.0001 day) is added to simulate the response of the wastage pump.

### Table 12: Wastage flowrate in function of time

<table>
<thead>
<tr>
<th>Time (d)</th>
<th>(Q_w \text{ (m}^3\text{.d}^{-1}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ≤ (t) &lt; 182</td>
<td>300</td>
</tr>
<tr>
<td>182 ≤ (t) &lt; 364</td>
<td>450</td>
</tr>
<tr>
<td>364 ≤ (t) &lt; 546</td>
<td>300</td>
</tr>
<tr>
<td>546 ≤ (t) &lt; 608</td>
<td>450</td>
</tr>
</tbody>
</table>

Appendices A4 to A6 summarize the results obtained in dynamic conditions in open loop (A4), in closed-loop with ideal sensors and actuators (A5) and in closed-loop with realistic sensors and actuators (A6).

11.1. Controller variables

For the dissolved oxygen control in second aerated compartment, the probe is assumed to be of class A with a measurement range of 0 to 10 g (-COD).m\(^{-3}\) and a measurement noise of 0.25 g (-COD).m\(^{-3}\). The manipulated variable is the oxygen transfer coefficient, \(K_{L}a_4\).

Constraints are applied on recirculation flows. In the test case, \(Q_{int}\) is maintained constant and is set at \(Q_{i,stab}\). The external recycle flow rate \(Q_r\) is maintained constant and is set to \(Q_r = Q_{r,stab}\). There are also constraints on oxygen transfer in compartment 4: \(K_{L}a_4 = 0 \text{ to } 10 \text{ d}^{-1}\).

11.2. Controller type

The suggested controller is of the PI type. Its performance is assessed by:
- the Integral of Absolute Error (IAE)

\[
IAE = \int_{t_0}^{t_f} |e| \, dt
\]  

(235)

where \(e\) is the error:

\[
e = Z_{\text{setpoint}} - Z_{\text{meas}}
\]

(236)

- the Integral of Squared Error (ISE)

\[
ISE = \int_{t_0}^{t_f} e^2 \, dt
\]

(237)

- the maximal deviation from set point:

\[
Dev^\text{max} = \max \left\{ |e| \right\}
\]

(238)

- the error variance:

\[
Var(e) = \frac{\sigma^2}{\eta}
\]

(239)

with
Benchmark Simulation Model no. 2 (BSM2)

12. PERFORMANCE ASSESSMENT

The flow-weighted average values of the effluent concentrations over the evaluation should obey the limits given in Table 13. Total nitrogen \( (N_{\text{tot}}) \) is calculated as the sum of \( S_{\text{NO}_e} \) and \( S_{\text{NKj,e}} \), where \( S_{\text{NKj}} \) is the Kjeldahl nitrogen concentration.

Table 13: Effluent quality limits

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{\text{tot}} )</td>
<td>&lt;18 g N.m(^{-3})</td>
</tr>
<tr>
<td>( \text{COD}_{\text{tot}} )</td>
<td>&lt;100 g COD.m(^{-3})</td>
</tr>
<tr>
<td>( S_{\text{NH}} )</td>
<td>&lt;4 g N.m(^{-3})</td>
</tr>
<tr>
<td>( \text{TSS} )</td>
<td>&lt;30 g SS.m(^{-3})</td>
</tr>
<tr>
<td>( \text{BOD}_5 )</td>
<td>&lt;10 g BOD.m(^{-3})</td>
</tr>
</tbody>
</table>

The percentage of time the effluent limits are not met must be reported, as well as the number of violations. The number of violations is defined as the number of crossings of the limit (from below to above the limit).

The performance assessment is made at two levels.
- The **first level** concerns the local control loops, assessed by IAE (Integral of the Absolute Error) and ISE (Integral of the Squared Error) criteria, by maximal deviation from set points, and by error variance. Basically, this serves as a proof that the proposed control strategy has been applied properly.
- The **second level** provides measures for the effect of the control strategy as such on plant performance and it can be divided into four sub-levels:
  - the **effluent quality**: levies or fines are to be paid due to the discharge of pollution in the receiving water bodies. The Effluent Quality Index \( (EQI) \) (kg pollution unit.d\(^{-1}\)) is averaged over the period of observation \( t_{\text{obs}} \) (d) (i.e. 364 days = 1 year) based on a weighting of the effluent loads of compounds that have a major influence on the quality of the receiving water and that are usually included in regional legislation. It is defined as:

\[
\text{EQI} = \frac{1}{t_{\text{obs}} \cdot 1000} \int_{t=245 \text{ days}}^{t=609 \text{ days}} \left( B_{\text{TSS}} \cdot \text{TSS}_e(t) + B_{\text{COD}} \cdot \text{COD}_e(t) + B_{\text{NKj}} \cdot S_{\text{NKj,e}}(t) + B_{\text{NO}} \cdot S_{\text{NO}_e}(t) + B_{\text{BOD5}} \cdot \text{BOD}_e(t) \right) \cdot Q_e(t) \, dt \tag{246}
\]

where
\[ S_{NK,j,e} = S_{NH,e} + S_{ND,e} + X_{ND,e} + i_{XB}(X_{B,H,e} + X_{X,A,e}) + i_{XP}(X_{P,e} + X_{I,e}) \]  
\[ TSS_{e} = 0.75 \left( X_{S,e} + X_{I,e} + X_{B,H,e} + X_{B,A,e} + X_{P,e} \right) \]  
\[ BOD_{S,e} = 0.25 \left( S_{S,e} + X_{S,e} + (1 - f_p) \cdot (X_{B,H,e} + X_{B,A,e}) \right) \]  
\[ COD_{S,e} = S_{S,e} + X_{S,e} + X_{I,e} + X_{B,H,e} + X_{B,A,e} + X_{P,e} \]  
\[ Q_{e} = Q_{ae,e} + Q_{bypass} \]

and the \( B_i \) are weighting factors for the different types of pollution to convert them into pollution units (Table 14). The concentrations are to be expressed in g.m\(^{-3}\). The values for \( B_i \) have been deduced from Vanrolleghem et al. (1996).

Table 14: \( B_i \) values

<table>
<thead>
<tr>
<th>Factor</th>
<th>( B_{TSS} )</th>
<th>( B_{COD} )</th>
<th>( B_{NKj} )</th>
<th>( B_{NO} )</th>
<th>( B_{BOD} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value (g pollution unit.g(^{-1}))</td>
<td>2</td>
<td>1</td>
<td>30</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

The 95% percentiles of the effluent ammonia (\( S_{NH,e,95} \)), effluent total nitrogen (\( N_{tot,e,95} \)) and total suspended solids (\( TSS_{e,95} \)) have to be shown as well. These percentiles represent the \( S_{NH}, N_{tot} \) and \( TSS \) effluent concentrations that are exceeded 5% of the time.

- the cost factors for operation
- the sludge production to be disposed (\( SP \)) (kg.d\(^{-1}\))

The sludge production, \( SP \), is calculated from the total solid flow from wastage and the solids accumulated in the system over the period of time considered (the last 364 days of the weather file).

\[ SP = \frac{1}{t_{obs}} \left( TSS_{609 \text{ days}} - TSS_{245 \text{ days}} \right) + 0.75 \cdot \int_{t=245 \text{ days}}^{t=609 \text{ days}} \left( X_{S,w} + X_{I,w} + X_{B,H,w} + X_{B,A,w} \right) Q_{w} \left( t \right) \, dt \]  
\[ \text{where } TSS \left( t \right) \text{ is the amount of solids in the system at time } t, \text{ i.e.} \]  
\[ TSS \left( t \right) = TSS_{as} \left( t \right) + TSS_{sc} \left( t \right) \]  
\( TSS_{as} \) and \( TSS_{sc} \) are given respectively by equations 60 and 62.

- the total sludge production (\( SP_{\text{total}} \)) (kg.d\(^{-1}\)) takes into account the sludge to be disposed and the sludge lost at the weir:

\[ SP_{\text{total}} = SP + \frac{0.75}{t_{obs}} \int_{t=245 \text{ days}}^{t=609 \text{ days}} \left( X_{S,w} + X_{I,w} + X_{B,H,w} + X_{B,A,w} \right) Q_{w} \left( t \right) \, dt \]  

- the aeration energy (\( AE \)) (kWh.d\(^{-1}\)) and the pumping energy (\( PE \)) (kWh.d\(^{-1}\)) (internal and external flow recycle pumps).

The pumping energy depends on how the various tanks can be arranged on the available space. Considering the state-of-the-art design rules an arrangement with two parallel lines, similar to the one shown in Appendix 1, can be proposed for the activated ludge section of BSM2. In BSM2 the pumping energy is calculated as:

\[ PE = \frac{1}{t_{obs}} \int_{t=245 \text{ days}}^{t=609 \text{ days}} \left( 0.004 \cdot Q_{\text{im}} \left( t \right) + 0.008 \cdot Q_{r} \left( t \right) + 0.05 \cdot Q_{w} \left( t \right) + 0.075 \cdot Q_{\text{ps}} + 0.06 \cdot Q_{\text{du}} + 0.004 \cdot Q_{du} \right) \, dt \]  

with the flow rates expressed in m\(^3\).d\(^{-1}\).

The aeration energy \( AE \) should take into account the plant peculiarities (type of diffuser, bubble size, depth of submersion, etc.) and is calculated from the \( K_{L,a} \) according to the following relation, valid for Degrémont DP230 porous disks at an immersion depth of 4 m:

\[ AE = \frac{S_{\text{out}}}{t_{obs}} \cdot 1.8 \cdot 1000 \int_{t=245 \text{ days}}^{t=609 \text{ days}} \sum_{k=1}^{s} Y_{a,k} \cdot K_{L,a} \left( t \right) \, dt \]  

with \( K_{L,a} \) given in d\(^{-1}\) and \( k \) referring to the compartment number.
- the consumption of external carbon source \((EC)\) (kg COD.d\(^{-1}\)) that could be added to improve denitrification (see Section 7 on control and handles)

\[
EC = \frac{COD_{EC}}{t_{obs}} \cdot 1000 \int_{t_{245days}}^{t_{609days}} \left( \sum_{k=1}^{n} Q_{EC,k} \right) \cdot dt
\]  \hspace{1cm} (257)

where \(Q_{EC,k}\) is the flow rate of external carbon added to compartment \(k\) and \(COD_{EC} = 400,000\) g COD.m\(^{-3}\) is the concentration of readily biodegradable substrate in the external carbon source.

- the mixing energy \((ME)\) (kWh.d\(^{-1}\))

The compartments in anoxic state should be mixed to avoid settling. Mixing energy is a function of the compartment volume.

\[
ME = \frac{24}{t_{obs}} \int_{t_{245days}}^{t_{609days}} \sum_{k=1}^{n} \left[ 0.005 \cdot V_{ak,k} \cdot \left( \begin{array}{c} 1 \\
\text{if } K_{L,a_k} \left( t \right) < 20 \text{ d}^{-1} \\
0 \text{ otherwise}
\end{array} \right) \right] \cdot dt
\]  \hspace{1cm} (258)

- the methane production \((MET_{prod})\) (kg.d\(^{-1}\))

\[
MET_{prod} = \frac{16 \cdot P_{\text{am}}}{RT_{ad} \cdot t_{obs}} \int_{t_{245days}}^{t_{609days}} \frac{Q_{\text{gas}} \left( t \right) \cdot p_{\text{gas,cha}} \left( t \right)}{p_{\text{gas}} \left( t \right)} \cdot dt
\]  \hspace{1cm} (259)

- the heating energy \((HE)\) (kWh.d\(^{-1}\))

It is necessary to heat the digester influent to the digester operating temperature \((T_{ad})\):

\[
HE = \frac{1000 \cdot 4.186}{86400 \cdot t_{obs}} \int_{t_{245days}}^{t_{609days}} \left( T_{ad} - T_{ad,i} \right) \cdot Q_{\text{ad}} \left( t \right) \cdot dt
\]  \hspace{1cm} (260)

\[
\text{with } T_{ad,i} = \frac{T_{pu} \cdot Q_{pu} + T_{in} \cdot Q_{in}}{Q_{ad}} + 273.15
\]  \hspace{1cm} (261)

assuming \(T_{pu}\) and \(T_{in}\) are given in °C and \(T_{ad}\) in K.

The methane produced in the digester is used to generate the necessary heat energy. The net heating energy is calculated as:

\[
HE_{\text{net}} = \max \left( 0, HE - 7 \cdot MET_{\text{prod}} \right)
\]  \hspace{1cm} (262)

- controller output variations

The maximum values and the variance of the manipulated variables variations should be given. This will provide an indication on peak loads and the wear of the pumps and aeration devices.

Furthermore, in case the user applies other influent data files than the one defined with BSM2 an Influent Quality Index \((IQI)\) index is proposed to compare the influent qualities:

\[
IQI = \frac{1}{t_{obs}} \int_{t_{245days}}^{t_{609days}} \left( B_{\text{TSS}} \cdot T_{\text{Ss}} \left( t \right) + B_{\text{COD}} \cdot COD_{\text{B}} \left( t \right) + B_{\text{DNK}} \cdot S_{\text{DNK}} \left( t \right) + B_{\text{BODs}} \cdot BOD_{\text{S}} \left( t \right) \right) \cdot Q_{\text{t}} \left( t \right) \cdot dt
\]  \hspace{1cm} (263)

with:

\[
S_{\text{DNK}} = S_{\text{NHI}} + S_{\text{ND}} + X_{\text{ND}} + \left( i_{\text{X}} \cdot \left( X_{\text{B,HI}} + X_{\text{X,AA}} \right) \right) + i_{\text{XP}} \cdot \left( X_{\text{P}} + X_{\text{I}} \right)
\]  \hspace{1cm} (264)

\[
T_{\text{Ss}} = 0.75 \cdot \left( X_{\text{S,s}} + X_{\text{I}} + X_{\text{B,HI}} + X_{\text{B,AA}} + X_{\text{P}} \right)
\]  \hspace{1cm} (265)

\[
BOD_{\text{S}} = 0.65 \cdot \left( X_{\text{S,s}} + X_{\text{S,AA}} + \left( 1 - f_{\text{P}} \right) \cdot \left( X_{\text{B,HI}} + X_{\text{B,AA}} \right) \right)
\]  \hspace{1cm} (266)

\[
COD = S_{\text{S,s}} + S_{\text{I}} + X_{\text{S,s}} + X_{\text{I}} + X_{\text{B,HI}} + X_{\text{B,AA}} + X_{\text{P,s}}
\]  \hspace{1cm} (267)

- Finally an Overall Cost Index \((OCI)\) is calculated:

\[
OCI = AE + PE + 3 \cdot SP + 3 \cdot EC + ME - 6 \cdot MET_{\text{prod}} + HE_{\text{net}}
\]  \hspace{1cm} (268)
13. SENSORS AND CONTROL HANDLES

13.1. Introduction

To test your own control strategy on the BSM2 plant, appropriate sensors and actuators must be selected. To avoid unrealistic control behaviour, the dynamic behaviour of sensors and actuators (control handles) as well as additional measurement noise must be considered. To allow for a wide range of different strategies to be tested (within the confinement of the physical plant layout), a significant number of sensors and control handles are available. Their mathematical descriptions focus on simplicity rather than completely accurate reproductions of their true behaviour.

The principle for any good control strategy implies that the number of sensors and control actions should be minimised within the framework of the selected control strategy, due to the investment and maintenance costs, etc (Rieger et al., 2003).

For initialisation purposes, first test of control concepts, or evaluation of the theoretical potential of control options, it is of course a valid option to use ideal sensors (no noise, no delay). For internal flows (e.g. return sludge, internal recycle), which are basically control handles, it can be assumed that the flow rates are known or can be measured without errors and delays. For such an ideal sensor, no specific sensor model is required. But the usage of ideal sensors should be reported when discussing a specific control strategy.

13.2. Sensors

The aim of the sensor classification is to describe different sensor types but also to limit the number of sensor classes in order to ease the comparison of the simulation results. The benchmark deals with control strategies, therefore only a few related criterions are used and only one minimal measuring interval of 5 minutes is taken into account. It is not intended to define a user configurable class, since this would make it difficult to compare different benchmark studies. Should it nevertheless be impossible to choose a class, the benchmark model user is requested to describe the specific sensor in detail.

The main parameter to describe the sensor dynamics of the sensor classes is the “Response time”. This parameter is defined in an ISO norm (ISO 2003) and characterises the sensor dynamics based on a step response as presented in Figure 9.

![Figure 9: Definition of response time](image)

In the norm the response time is the sum of the delay and the rise (or fall) time. The delay is defined as the time to reach 10% of the final value of a step response ($t_d$). Thus, the delay time in this context is not exactly the same as a transport delay time or dead-time defined in control engineering. The overall time to reach (and not to leave) a band from 90% - 110% of the final value of the step response is introduced as response time (here $t_r$). To describe the dynamics of a sensor it is assumed that the two values delay time and response time (as defined by Figure 6) are given.
For the definition of the benchmark sensor classes a response time \( t_r \) is proposed. The six sensor classes are shown in Table 14 and a list of typical sensors is provided in Table 15.

### Table 14: Sensor classes. A measuring interval equal to 0 means continuous measurement

<table>
<thead>
<tr>
<th>Sensor classes</th>
<th>Response time ( t_r ) [min]</th>
<th>Measuring interval ( t_i ) [min]</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>1</td>
<td>0</td>
<td>Ion sensitive, optical without filtration</td>
</tr>
<tr>
<td>Class B₀</td>
<td>10</td>
<td>0</td>
<td>Gas sensitive + fast filtration</td>
</tr>
<tr>
<td>Class B₁</td>
<td>10</td>
<td>5</td>
<td>Photometric + fast filtration</td>
</tr>
<tr>
<td>Class C₀</td>
<td>20</td>
<td>0</td>
<td>Gas-sensitive + slow filtration</td>
</tr>
<tr>
<td>Class C₁</td>
<td>20</td>
<td>5</td>
<td>Photometric + slow filtration or sedimentation</td>
</tr>
<tr>
<td>Class D</td>
<td>30</td>
<td>30</td>
<td>Photometric or titrimetric for total components</td>
</tr>
</tbody>
</table>

The response time includes the whole system with filtration unit and measuring system. Class A is a more or less ideal sensor; the response time of 1 minute is chosen in order to prevent unrealistic control applications. Class B contains mainly classical on-line analyzers with a fast filtration and short sample loops. In Class C, analyzers with a slow filtration or sedimentation unit are described. Class D includes all batch measurements like respirometer and sensors for total components. To take into account continuously and discontinuously measuring sensors, the classes B and C are divided into two subclasses. Five minutes is selected as the measuring interval, which is a typical minimum value for photometric analyzers. Longer intervals are not useful for control actions and are therefore neglected.

Additional to choosing the sensor class, the user has to define the measuring range for each sensor. Depending on the chosen measurement range, the standard deviation is assumed to be 2.5% of the maximum measurement value (see sensor model description).

Real measurement signals always include measurement noise, which can lead to unwanted control actions or slow down the reaction. Therefore, noise is included in the sensor model. The idea is not to model noise exactly, but to take into account some of its effects. In order to get comparable benchmark simulation results, the noise signal is defined. Choice of a random signal would have required running each benchmark simulation a large number of times in order to eliminate the influence of the random signal. The noise signal is chosen with a standard deviation of 1, which is multiplied with the defined noise level (2.5% of the maximum measurement value). The noise is white zero-mean normally distributed noise. Other types of noise would be too specific and the sensors within one class would not be comparable.

As an illustration, oxygen and nitrate sensors, which can be used in the activated sludge section, can very easily be described as:
- oxygen sensor: Class A, measurement range: 0-10 g (-COD).m\(^{-3}\), measurement noise \( \delta = 0.25 \) g (-COD).m\(^{-3}\).
- nitrate sensor: Class B₀ with a measurement range 0-20 g N.m\(^{-3}\), measurement noise \( \delta = 0.5 \) g N.m\(^{-3}\).

### 13.3. Sensor model description

To ensure identical implementation and behaviour of the sensor models, it is necessary to describe the model in detail. The following description is the result of a Simulink implementation and takes into account a number of performance issues which are similar for most of the simulation systems.

The proposed sensor classes contain a set of continuous (A, B₀, C₀) and time-discrete sensor models (B₁, C₁, D). Continuous models are preferred to time-discrete ones for implementing the continuous sensors for performance reasons. The discontinuous sensors B₁ and C₁ are modelled in a similar way but include an output sample and hold function. Sensor class D is modelled only in discrete form.
13.3.1. Continuously measuring sensors

For the sensor classes A, B₀ and C₀ the approach is shown in Figure 10:

![Simulink model of sensor class A, B₀ and C₀](image)

**Figure 10:** Simulink model of sensor class A, B₀ and C₀

The original sensor signal \( u \) is transformed by a linear transfer function (block Transfer Fcn). This transfer function is used to implement the expected time response of the sensor. Real time behaviour of sensors is typically a combination of transport+delay time behaviour (or dead time) caused by sample transport and preparation and a first or higher order dynamics (time constants) caused by different reasons, e.g. a mixing tank.

To have a sensor model with the same response time, a series of equal first order delay transfer functions is assumed. The number of first order transfer functions in series \( n \) determines the ratio of delay time and response time (as defined in Figure 9). Table 16 shows the parameters for the response-time modelling (see specific sensor model) of the continuously operating sensors.

For the sensor class A a response time \( t_r \) of 1 min and a system order of \( n = 2 \) is suggested. The assumed transfer function is:

\[
(269)
\]

The problem is to find \( \tau \) such as \( t_r = 1 \) min, using either Simulink or the time-domain function:

\[
(270)
\]

With \( \tau = \frac{0.257}{3.89} \), the ratio of the delay time to the rise time \( (R_{d/y}) \) is equal to 0.133. Thus the transfer function is only a small fraction of the response time as typical for this sensor class.

For the sensor classes B and C, a system order of \( n = 8 \) is assumed to mimic the behaviour of the sensors. For class B a response time of 10 min and for class C of 20 min is selected. The transfer function is

\[
(271)
\]

with \( \tau = \frac{t_r}{11.7724} \).

This will lead to a ratio of the delay time to the response time equal to 0.392. In this case, the delay time is approximately 40% of the response time. This is assumed to consider the significant effect of the transport of the sample for the sensor classes B and C. The step responses for the classes A, B₀ and C₀ are presented in Figure 11.
Table 15: Typical sensor characteristics within the proposed classification scheme

<table>
<thead>
<tr>
<th>Measured variable</th>
<th>Sensor types</th>
<th>$t_d$ (min)</th>
<th>$t_i$ (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLSS (g.m$^{-3}$)</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Turbidity (FNU or g TSS.m$^{-3}$)</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_{NH_4}$ (ion sensitive)</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_{NO_x}$ (ion sensitive)</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_{NO_x}$ (UV)</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$C_{COD}$, $S_{COD}$ (UV/Vis)</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Flow rate (m$^3$.d$^{-1}$)</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Water level (m)</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Temperature (°C)</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pH</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_O$ (g (-COD).m$^{-3}$)</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sludge blanket height (m)</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_{NH_4}$ (gas sensitive + normal filtration)</td>
<td>B$_0$</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>$S_{NO_x}$ (UV + normal filtration)</td>
<td>B$_0$</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>$S_{NH_4}$ (photometric + normal filtration)</td>
<td>B$_1$</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>$S_{NO_3}$ (photometric + normal filtration)</td>
<td>B$_1$</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>$S_{NO_2}$ (photometric + normal filtration)</td>
<td>B$_1$</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>$S_{PO_4}$ (photometric + normal filtration)</td>
<td>B$_1$</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>$S_{NH_4}$ (gas sensitive + slow filtration or sedimentation)</td>
<td>C$_0$</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>$S_{NO_x}$ (UV + slow filtration or sedimentation)</td>
<td>C$_0$</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>$S_{NH_4}$ (photometric + slow filtration or sedimentation)</td>
<td>C$_1$</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>$S_{NO_3}$ (photometric + slow filtration or sedimentation)</td>
<td>C$_1$</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>$S_{NO_2}$ (photometric + slow filtration or sedimentation)</td>
<td>C$_1$</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>$S_{PO_4}$ (photometric + slow filtration or sedimentation)</td>
<td>C$_1$</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>$C_{COD}$ (thermal chemical oxidation + photometric)</td>
<td>D</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>TOC (thermal oxidation + IR detector)</td>
<td>D</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>$C_N$ (thermal oxidation + IR detector or chemoluminescence detector)</td>
<td>D</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>$C_T$ (thermal chemical oxidation + photometric)</td>
<td>D</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Respirometer</td>
<td>D</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Titration biosensor (alkalinity)</td>
<td>D</td>
<td>30</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 16: Parameters for response time modelling

<table>
<thead>
<tr>
<th>Sensor class</th>
<th>$t_r$ (min)</th>
<th>$n$</th>
<th>$\tau$ (min)</th>
<th>$R_{td/tr}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>2</td>
<td>0.257</td>
<td>0.133</td>
</tr>
<tr>
<td>B$_0$</td>
<td>10</td>
<td>8</td>
<td>0.849</td>
<td>0.392</td>
</tr>
<tr>
<td>C$_0$</td>
<td>20</td>
<td>8</td>
<td>1.699</td>
<td>0.392</td>
</tr>
</tbody>
</table>
The noise is modelled with a constant noise level $n_l$. In the Simulink model presented in Figure 10, the noise signal (white noise with a standard deviation $\delta=1$) is multiplied by the noise level $n_l$ and the maximum value of the measurement interval $y_{\text{max}}$. A normal distributed (standard deviation 1), frequency limited noise signal should be used and created by the user. The signal could be created using a sample time of 1 min and be interpolated using linear interpolation to provide a continuous noise signal. Using the sample time of 1 min together with the linear interpolation will limit the frequency spectrum of the noise (cut-off of high frequencies - pink noise. The noise is added to the delayed measurement signal and limited to the measurement interval $(0, y_{\text{max}})$.

13.3.2. Discontinuously measuring sensors

Sensor classes $B_1$, $C_1$ and $D$ are operated discontinuously using a sampling interval $t_i$. An example of an implementation using a Simulink model is presented in Figures 12 and 13. The implementation is similar to that used in the model for the continuously measuring sensors but includes an additional output sample and hold function.

![Simulink diagram](image)
Sensor class D represents batch-type reactors, for which any of the continuous delay times are negligible, compared to the batch operation of the measurement. An appropriate Simulink implementation is demonstrated in Figure 13. This model adds noise to the original signal, limits the sum to the measuring range \((0, y_{\text{max}})\) and uses a sample and hold function followed by a unit delay \(y(k) = u_3(k-1)\). Figure 14 shows examples of the output signal for all sensor classes.

13.3.3. Conclusions
Table 17 summarizes the recommended sensor parameter values for BSM2. Except for the plant influent flow rate, all the other flows are not explicitly measured but can be considered as known for simplicity.

13.4. Control handles
For reasons of simplicity, all available control handles are considered to be ideal with regard to their behaviour. In the closed-loop test case, only one control handle is used: the oxygen transfer rate in reactor number 4 \((K_L a_4)\). The following control handles are considered to exist for the implementation of new control strategies on the benchmark plant:

- internal flow recirculation rate \((Q_i)\);
- return sludge flow rate \((Q_r)\);
- wastage flow rate \((Q_w)\);
- anoxic/aerobic volume – all five biological reactors are equipped with both aerators and mechanical mixing devices, i.e. in a discrete fashion the volumes for anoxic and aerobic behaviour can be modified;
- aeration intensity individually for each reactor \((K_L a_1, K_L a_2, K_L a_3, K_L a_4, K_L a_5)\), taking into account the dynamics of the aeration system;
- external carbon source flow rate \((Q_{EC1}, Q_{EC2}, Q_{EC3}, Q_{EC4}, Q_{EC5})\) where the carbon source is considered to consist of readily biodegradable substrate, i.e. COD$_{EC}$;
- influent distribution by use of step feed (fractions of the influent flow to each of the five biological reactors: \(f_{Q_{i1}}, f_{Q_{i2}}, f_{Q_{i3}}, f_{Q_{i4}}, f_{Q_{i5}}\));
- distribution of internal flow recirculation (fractions of the internal recirculation flow to each of the five biological reactors: \(f_{Q_{int1}}, f_{Q_{int2}}, f_{Q_{int3}}, f_{Q_{int4}}, f_{Q_{int5}}\));
- distribution of return sludge flow (fractions of the return sludge flow to each of the five biological reactors: \(f_{Q_{r1}}, f_{Q_{r2}}, f_{Q_{r3}}, f_{Q_{r4}}, f_{Q_{r5}}\));
- reject water flow rate \((Q_{st, set})\)

The above selection gives about 30 individual control handles to manipulate the defined benchmark plant and dramatically increases its flexibility. Such a number of available control handles may not be realistic for a real plant but is defined for the benchmark plant in order to allow for basically any type of general control strategy. The defined limitations for the different control handles are given in Table 18.
The non-ideal aeration system ($K_L a_1 - K_L a_5$) is defined with significant dynamics. A response time of $t_r = 4$ min is considered (see Rieger et al., 2005). A second order time delay function gives a reasonable model of this process. The time constant of each of the two identical first order delays is $\tau = t_r/3.89 = 1.03$ min.

**Table 17: Recommended BSM2 sensor parameters**

<table>
<thead>
<tr>
<th>Measured variable</th>
<th>Class</th>
<th>Measurement range</th>
<th>Measurement noise ($\delta$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow rate (m$^3$.d$^{-1}$) high range</td>
<td>A</td>
<td>0-100 000</td>
<td>2500</td>
</tr>
<tr>
<td>Water level (m)</td>
<td>A</td>
<td>0-5</td>
<td>0.125</td>
</tr>
<tr>
<td>Temperature (°C)</td>
<td>A</td>
<td>5-25</td>
<td>0.5</td>
</tr>
<tr>
<td>pH</td>
<td>A</td>
<td>5-9</td>
<td>0.1</td>
</tr>
<tr>
<td>$S_O$ (g (-COD).m$^{-3}$)</td>
<td>A</td>
<td>0-10</td>
<td>0.25</td>
</tr>
<tr>
<td>Sludge blanket level (m)</td>
<td>A</td>
<td>0-5</td>
<td>0.125</td>
</tr>
<tr>
<td>$S_{NO}$ (g N.m$^{-3}$)</td>
<td>$B_0$</td>
<td>0-20</td>
<td>0.5</td>
</tr>
<tr>
<td>$S_{NH}$ (g N.m$^{-3}$) low range</td>
<td>$B_0$</td>
<td>0-20</td>
<td>0.5</td>
</tr>
<tr>
<td>$S_{NH}$ (g N.m$^{-3}$) high range</td>
<td>$B_0$</td>
<td>0-50</td>
<td>1.25</td>
</tr>
<tr>
<td>$S_{ALK}$ (mole HCO$_3$-m$^{-3}$)</td>
<td>$B_0$</td>
<td>0-20</td>
<td>0.5</td>
</tr>
<tr>
<td>Mixed-liquor suspended solids (g.m$^{-3}$)</td>
<td>A</td>
<td>0-10 000</td>
<td>250</td>
</tr>
<tr>
<td>Effluent total suspended solids (g.m$^{-3}$)</td>
<td>A</td>
<td>0-200</td>
<td>5</td>
</tr>
<tr>
<td>COD$_{tot}$ (g COD.m$^{-3}$)</td>
<td>D</td>
<td>0-1 000</td>
<td>25</td>
</tr>
<tr>
<td>OUR (g (-COD).m$^{-3}$.d$^{-1}$)</td>
<td>D</td>
<td>0-2 000</td>
<td>50</td>
</tr>
</tbody>
</table>

![Figure 14: Pulse response of sensor classes.](image)

**13.5. Alternative description**

To clarify the sensor and actuator models, a presentation in form of differential and difference equations is also presented in this section. The notations are summarized in Table 19.

**713.5.1 Model for sensor class A and actuator model**

\begin{align}
\text{(272)}
\end{align}

\begin{align}
\text{(273)}
\end{align}

\begin{align}
\text{(274)}
\end{align}

\begin{align}
\text{(275)}
\end{align}
13.5.2. Model for sensor class B₀ and C₀

Table 18: Available control handles and their limitations

<table>
<thead>
<tr>
<th>Control handle</th>
<th>Minimum value</th>
<th>Maximum value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{\text{ut}}$ (m³.d⁻¹)</td>
<td>0</td>
<td>309,720.</td>
<td>Max = 500% of $Q_{\text{i,stab}}$</td>
</tr>
<tr>
<td>$Q_i$ (m³.d⁻¹)</td>
<td>0</td>
<td>41,295.</td>
<td>Max = 200% of $Q_{\text{i,stab}}$</td>
</tr>
<tr>
<td>$Q_w$ (m³.d⁻¹)</td>
<td>0</td>
<td>1844.6</td>
<td>Max = 10% of $Q_{\text{i,stab}}$</td>
</tr>
<tr>
<td>$K_{L_a} A_4$ (d⁻¹)</td>
<td>0</td>
<td>240</td>
<td>Reactor 4</td>
</tr>
<tr>
<td>$Q_{\text{EC1}}$ (m³.d⁻¹)</td>
<td>0</td>
<td>5</td>
<td>Reactor 1</td>
</tr>
</tbody>
</table>

Carbon source conc. 400,000 g COD.m⁻³ available as COD₅ (e.g. 25% ethanol solution)

Table 19: Variables used in the sensor models

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u(t)$</td>
<td>ideal measurement signal from process</td>
</tr>
<tr>
<td>$x_1(t) \ldots x_7(t)$</td>
<td>internal states for dynamic part of sensor model</td>
</tr>
<tr>
<td>$u_2(t)$</td>
<td>delayed measurement signal (intermediate variable)</td>
</tr>
<tr>
<td>$y_2(t), y_3(k), y_4(k)$</td>
<td>intermediate signals</td>
</tr>
<tr>
<td>$y(t)$</td>
<td>real measurement signal from sensor (delayed, noisy, limited)</td>
</tr>
<tr>
<td>$\tau$</td>
<td>time constant for one first order time delay</td>
</tr>
<tr>
<td>$t_i$</td>
<td>sampling interval for discontinuous sensor models</td>
</tr>
</tbody>
</table>

13.5.3. Model for sensor class B₁ and C₁

$y_3(k) = y_2(t; t = k\cdot t_i)$
\[ y(t) = y_3(k, k = \text{floor}(t/t_i)) \]  

13.5.4. Model for sensor D

\[ y_3(k) = y_2(t, t = k \cdot t_i) \]  

\[ y_4(k) = y_3(k-1) \]  

\[ y(t) = y_4(k, k = \text{floor}(t/t_i)) \]  

14. CONCLUDING REMARKS

The aim of this Technical Report is to give the details of the models used in BSM2. Further explanations concerning the reasoning which ended up with the choices made can be found in the Corresponding Technical Reports. Furthermore supplementary informations are given in the documents accompanying the software.

15. REFERENCES


Appendix 1: Practical BSM1 plant layout

[Diagram of plant layout with labeled components such as activated sludge tanks, secondary clarifiers, and flow lines.]

Q: Pipe length: 50 m; Static head: 0.25 m; 30 degree elbows

Activated Sludge Tanks:
- V_a = 2750 + 3(1333/2) = 5668.5 m³
- A_w = 750 m² (10 m x 75 m)

Secondary Clarifiers:
- A_c = 2(1541 ft)² = 2(1541 m)²
- Q_c = 0.68 m³/s

Mixing: 20% activated sludge from activated sludge tank 1, 80% from activated sludge tank 2

Gravity flow box

Reference to WTP
Appendix 2: Source codes of ASM/ADM/ASM interfaces for BSM2

A2.1. MATLAB code

/*
* New version (no 3) of the ASM1 to ADM1 interface based on discussions
* within the IWA TG BSM community during 2002-2006. Now also including charge
* balancing and temperature dependency for applicable parameters.
* Model parameters are defined in adm1init_bsm2.m
* u is the input in ASM1 terminology + extra dummy states, 21 variables
* plus one extra input = dynamic pH from the ADM1 system (needed for
* accurate charge balancing - also used the ADM1 to ASM1 interface).
* If temperature control of AD is used then the operational temperature
* of the ADM1 should also be an input rather than a defined parameter.
* Temperature in the ADM1 and the ASM1 to ADM1 and the ADM1 to ASM1
* interfaces should be identical at every time instant.
* Input vector:
* u[0] : Si = soluble inert organic material (g COD/m3)
* u[1] : Ss = readily biodegradable substrate (g COD/m3)
* u[2] : Xi = particulate inert organic material (g COD/m3)
* u[3] : Xs = slowly biodegradable substrate (g COD/m3)
* u[4] : Xbh = active heterotrophic biomass (g COD/m3)
* u[5] : Xba = active autotrophic biomass (g COD/m3)
* u[6] : Xp = particulate product arising from biomass decay (g COD/m3)
* u[7] : So = oxygen (g -COD/m3)
* u[8] : Sno = nitrate and nitrite nitrogen (g N/m3)
* u[9] : Snh = ammonia and ammonium nitrogen (g N/m3)
* u[10] : Snd = soluble biogradable organic nitrogen (g N/m3)
* u[11] : Xnd = particulate biogradable organic nitrogen (g N/m3)
* u[12] : Salk = alkalinity (mole HCO3- /m3)
* u[13] : TSS = total suspended solids (internal use) (mg SS/l)
* u[14] : flow rate (m3/d)
* u[15] : temperature (deg C)
* u[16:20] : dummy states for future use
* u[21] : pH in the anaerobic digester
* y is the output in ADM1 terminology + extra dummy states, 33 variables
* y[0] : Ssu = monosacharides (kg COD/m3)
* y[1] : Saa = amino acids (kg COD/m3)
* y[2] : Sfa = long chain fatty acids (LCFA) (kg COD/m3)
* y[3] : Sva = total valerate (kg COD/m3)
* y[4] : Sbu = total butyrate (kg COD/m3)
* y[5] : Spro = total propionate (kg COD/m3)
* y[6] : Sac = total acetate (kg COD/m3)
* y[7] : Sh2 = hydrogen gas (kg COD/m3)
* y[8] : Sch4 = methane gas (kg COD/m3)
* y[9] : Sic = inorganic carbon (kmole C/m3)
* y[10] : Sin = inorganic nitrogen (kmole N/m3)
* y[11] : Si = soluble inerts (kg COD/m3)
* y[12] : Xc = composites (kg COD/m3)
* y[13] : Xch = carbohydrates (kg COD/m3)
* y[14] : Xpr = proteins (kg COD/m3)
* y[15] : Xli = lipids (kg COD/m3)
* y[16] : Xsu = sugar degraders (kg COD/m3)
* y[17] : Xaa = amino acid degraders (kg COD/m3)
* y[18] : Xfa = LCFA degraders (kg COD/m3)
* y[19] : Xc4 = valerate and butyrate degraders (kg COD/m3)
* y[20] : Xpro = propionate degraders (kg COD/m3)
* $y_{[21]}$: Xac = acetate degraders (kg COD/m$^3$)
* $y_{[22]}$: Xh$_2$ = hydrogen degraders (kg COD/m$^3$)
* $y_{[23]}$: Xi = particulate inerts (kg COD/m$^3$)
* $y_{[24]}$: scat+ = cations (metallic ions, strong base) (kmole/m$^3$)
* $y_{[25]}$: san− = anions (metallic ions, strong acid) (kmole/m$^3$)
* $y_{[26]}$: flow rate (m$^3$/d)
* $y_{[27]}$: temperature (deg C)
* $y_{[28:32]}$: dummy states for future use

* ASM1 → ADM1 conversion, version 3 for BSM2
* Copyright: John Copp, Primodal Inc., Canada; Ulf Jeppsson, Lund
* University, Sweden; Damien Batstone, Univ of Queensland, Australia, Ingmar Nopens, Univ of Ghent, Belgium,
* Marie-Noelle Pons, Nancy, France, Peter Vanrolleghem,
* Univ. Laval, Canada, Jens Alex, IFAK, Germany and
* Eveline Volcke, Univ of Ghent, Belgium.

```
#define S_FUNCTION_NAME asm2adm_v3_bsm2

#include "simstruc.h"
#include <math.h>

#define PAR ssGetArg(S,0)

/*
 * mdlInitializeSizes - initialize the sizes array
 */
static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumContStates(  S, 0);   /* number of continuous states     */
    ssSetNumDiscStates(  S, 0);   /* number of discrete states       */
    ssSetNumInputs(      S, 22);  /* number of inputs                */
    ssSetNumOutputs(     S, 33);  /* number of outputs               */
    ssSetDirectFeedThrough(S, 1);  /* direct feedthrough flag         */
    ssSetNumSampleTimes( S, 1);   /* number of sample times          */
    ssSetNumSFcnParams(  S, 1);   /* number of input arguments       */
    ssSetNumRWork(       S, 0);   /* number of real work vector elements */
    ssSetNumIWork(       S, 0);   /* number of integer work vector elements */
    ssSetNumPWork(       S, 0);   /* number of pointer work vector elements */
}

/*
 * mdlInitializeSampleTimes - initialize the sample times array
 */
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);
}

/*
 * mdlInitializeConditions - initialize the states
 */
static void mdlInitializeConditions(double *x0, SimStruct *S)
{
}
```
* mdlOutputs - compute the outputs
*/

static void mdlOutputs(double *y, double *x, double *u, SimStruct *S, int tid)
{
    double CODequiv, fnaa, fnxc, fnbac, fni, fnsi, fnsi_adm, frlixs, frlibac, frxs_adm,
    fdegrade_adm, frxs_as, fdegrade_as;
    double  R, T_base, T_op, pK_w_base, pK_a_va_base, pK_a_bu_base, pK_a_pro_base,
    pK_a_ac_base, pK_a_co2_base, pK_a_IN_base;
    double  pH_adm, pK_w, pK_a_co2, pK_a_IN, alfa_va, alfa_bu, alfa_pro, alfa_ac, alfa_co2, alfa_IN,
    alfa_NO, alfa_alk, factor;
    double CODdemand, remaina, remainb, remainc, remaind, ScatminusSan;
    double sorgn, xorgn, xprtemp, xprtemp2, xlitemp, xlitemp2, xlitemp3, xchtemp, xchtemp2,
    xchtemp3;
    double biomass, biomass_nobio, biomass_bioN, remainCOD, inertX, xc, noninertX, inertS,
    utemp[22], utemp2[22];

    /* parameters defined in adm1init_bsm2.m, INTERFACEPAR */
    CODequiv = mxGetPr(PAR)[0];
    fnaa = mxGetPr(PAR)[1];
    fnxc = mxGetPr(PAR)[2];
    fnbac = mxGetPr(PAR)[3];
    fni = mxGetPr(PAR)[4];
    fnsi = mxGetPr(PAR)[5];
    fnsi_adm = mxGetPr(PAR)[6];
    frlixs = mxGetPr(PAR)[7];
    frlibac = mxGetPr(PAR)[8];
    frxs_adm = mxGetPr(PAR)[9];
    fdegrade_adm = mxGetPr(PAR)[10];
    frxs_as = mxGetPr(PAR)[11]; /* not used in ASM2ADM */
    fdegrade_as = mxGetPr(PAR)[12]; /* not used in ASM2ADM */
    R = mxGetPr(PAR)[13];
    T_base = mxGetPr(PAR)[14];
    T_op = mxGetPr(PAR)[15]; /* should be an input variable if dynamic temperature control is used */
    pK_w_base = mxGetPr(PAR)[16];
    pK_a_va_base = mxGetPr(PAR)[17];
    pK_a_bu_base = mxGetPr(PAR)[18];
    pK_a_pro_base = mxGetPr(PAR)[19];
    pK_a_ac_base = mxGetPr(PAR)[20];
    pK_a_co2_base = mxGetPr(PAR)[21];
    pK_a_IN_base = mxGetPr(PAR)[22];

    pH_adm = u[21];

    factor = (1.0/T_base - 1.0/T_op)/(100.0*R);
    pK_w = pK_w_base - log10(exp(55900.0*factor));
    pK_a_co2 = pK_a_co2_base - log10(exp(7646.0*factor));
    pK_a_IN = pK_a_IN_base - log10(exp(51965.0*factor));
    alfa_va = 1.0/208.0*(-1.0/(1.0 + pow(10, pK_a_va_base - pH_adm)));
    alfa_bu = 1.0/160.0*( -1.0/(1.0 + pow(10, pK_a_bu_base - pH_adm)));
    alfa_pro = 1.0/112.0*(-1.0/(1.0 + pow(10, pK_a_pro_base - pH_adm)));
    alfa_ac = 1.0/64.0*(-1.0/(1.0 + pow(10, pK_a_ac_base - pH_adm)));
    alfa_co2 = -1.0/(1.0 + pow(10, pK_a_co2 - pH_adm));
    alfa_IN = (pow(10, pK_a_IN - pH_adm))/(1.0 + pow(10, pK_a_IN - pH_adm));
    alfa_NO = 1.0/14000.0; /* convert mgN/l into kmoleN/m3 */
    alfa_alk = -0.001; /* convert moleHCO3/m3 into kmoleHCO3/m3 */
    alfa_NO = -1.0/14000.0; /* convert mgN/l into kmoleN/m3 */
for (i = 0; i < 22; i++) {
    utemp[i] = u[i];
    utemp2[i] = u[i];
}

for (i = 0; i < 32; i++)
    y[i] = 0.0;

/*============================================================================
====================*/
/* Let CODdemand be the COD demand of available electron
* acceptors prior to the anaerobic digester, i.e. oxygen and nitrate */
CODdemand = u[7] + CODequiv*u[8];
utemp[7] = 0;
utemp[8] = 0;

/*============================================================================
====================*/
/* if extreme detail was used then some extra NH4 would be transformed
* into N bound in biomass and some biomass would be formed when
* removing the CODdemand (based on the yield). But on a total COD balance
* approach the below is correct (neglecting the N need for biomass growth)
* The COD is reduced in a hierarchical approach in the order:
* 1) SS; 2) XS; 3) XBH; 4) XBA. It is no real improvement to remove SS and add
* biomass. The net result is the same. */
if (CODdemand > u[1]) { /* check if COD demand can be fulfilled by SS*/
    remaina = CODdemand - u[1];
    utemp[1] = 0.0;
    if (remaina > u[3]) { /* check if COD demand can be fulfilled by XS*/
        remainb = remaina - u[3];
        utemp[3] = 0.0;
        if (remainb > u[4]) { /* check if COD demand can be fulfilled by XBH */
            remainc = remainb - u[4];
            utemp[4] = 0.0;
            if (remainc > u[5]) { /* check if COD demand can be fulfilled by XBA */
                remaind = remainc - u[5];
                utemp[5] = 0.0;
                utemp[7] = remaind;
                /* if here we are in trouble, carbon shortage: an error printout
                   should be given */
                /* and execution stopped */
            } else { /* reduced all COD demand by use of SS, XS, XBH and XBA */
            }
        } else { /* reduced all COD demand by use of SS, XS and XBH */
        }
    } else { /* reduced all COD demand by use of SS and XS */
    }
} else { /* reduced all COD demand by use of SS */
}
Benchmark Simulation Model no. 2 (BSM2)

/*============================================================================
====================*/
/* SS becomes part of amino acids when transformed into ADM
* and any remaining SS is mapped to monosacharides (no N contents)
* Enough SND must be available for mapping to amino acids */

sorgn = u[10]/fnaa; /* Saa COD equivalent to SND */
if (sorgn >= utemp[1]) { /* not all SND-N in terms of COD fits into amino acids */
y[1] = utemp[1]; /* map all SS COD into Saa */
    utemp[1] = 0.0; /* all SS used */
}
else { /* all SND-N fits into amino acids */
y[1] = sorgn; /* map all SND related COD into Saa */
    utemp[1] = utemp[1] - sorgn; /* excess SS, which will become sugar in ADM1 i.e. no nitrogen association */
    utemp[10] = 0.0; /* all SND used */
}
/*============================================================================
====================*/
/* XS becomes part of Xpr (proteins) when transformed into ADM
* and any remaining XS is mapped to Xch and Xli (no N contents)
* Enough XND must be available for mapping to Xpr */

xorgn = u[11]/fnaa; /* Xpr COD equivalent to XND */
if (xorgn >= utemp[3]) { /* not all XND-N in terms of COD fits into Xpr */
    xprtemp = utemp[3]; /* map all XS COD into Spr */
    utemp[3] = 0.0; /* all XS used */
    xlitemp = 0.0;
    xchtemp = 0.0;
}
else { /* all XND-N fits into Xpr */
    xprtemp = xorgn; /* map all XND related COD into Xpr */
    xchtemp = (1.0 - frlixs)*(utemp[3] - xorgn); /* part of XS COD not associated with N */
    xlitemp = frlixs*(utemp[3] - xorgn); /* part of XS COD not associated with N */
    utemp[3] = 0.0; /* all XS used */
    utemp[11] = 0.0; /* all XND used */
}
/*============================================================================
====================*/
/* Biomass becomes part of Xpr and XI when transformed into ADM
* and any remaining XBH and XBA is mapped to Xch and Xli (no N contents)
* Remaining XND-N can be used as nitrogen source to form Xpr */

biomass = utemp[4] + utemp[5];
biomass_nobio = biomass*(1.0 - frxs_adm); /* part which is mapped to XI */
biomass_bioN = (biomass*fnbac - biomass_nobio*fxni);
if (biomass_bioN < 0.0) {
    /* Problems: if here we should print 'ERROR: not enough biomass N to map the requested inert part' */
}
if ((biomass_bioN/fnaa) <= (biomass - biomass_nobio)) { /* all biomass N used */
    xprtemp2 = biomass_bioN/fnaa;  /* all biomass N used */
    remainCOD = biomass - biomass_nobio - xprtemp2;
}
else if ((utemp[11]/fnaa) > remainCOD) { /* use part of remaining XND-N to form proteins */
    xprtemp2 = xprtemp2 + remainCOD;
    remainCOD = 0.0;
    utemp[4] = 0.0;
    utemp[5] = 0.0;
}
else { /* use all remaining XND-N to form proteins */
    xprtemp2 = xprtemp2 + utemp[11]/fnaa;
    remainCOD = remainCOD - utemp[11]/fnaa;
    utemp[11] = 0.0;
}

xlitemp2 = frlibac*remainCOD;  /* part of the COD not associated with N */
xchtemp2 = (1.0 - frlibac)*remainCOD; /* part of the COD not associated with N */

else { /* all biomass COD used */
    xprtemp2 = biomass - biomass_nobio; /* all biomass COD used */
    N in XND */
}

utemp[4] = 0.0;
utemp[5] = 0.0;

/*============================================================================
====================*/
/* direct mapping of XI and XP to ADM1 XI (if fdegrade_ad = 0)
 * assumption: same N content in both ASM1 and ADM1 particulate inerts */

inertX = (1-fdegrade_adm)*(utemp[2] + utemp[6]);

/* special case: IF part of XI and XP in the ASM can be degraded in the AD
 * we have no knowledge about the contents so we put it in as composts (Xc)
 * we need to keep track of the associated nitrogen
 * N content which may be different, take first from XI&XP-N, then XND-N, then SND-N,
 * then SNH. A similar principle could be used for other states. */

xc = 0.0;
xlitemp3 = 0.0;
xchtemp3 = 0.0;
if (fdegrade_adm > 0) { /* N in XI&XP(ASM) not enough */
    x = noninertX*fxni/fnxc;
    noninertX = noninertX - noninertX*fxni/fnxc;
    if (utemp[11] < (noninertX*fnxc)) { /* N in XND not enough */
        x = x + utemp[11]/fnxc;
        noninertX = noninertX - utemp[11]/fnxc;
        utemp[11] = 0.0;
        if (utemp[10] < (noninertX*fnxc)) { /* N in SND not enough */
            x = x + utemp[10]/fnxc;
            noninertX = noninertX - utemp[10]/fnxc;
            utemp[10] = 0.0;
            if (utemp[9] < (noninertX*fnxc)) { /* N in SNH not enough */
                x = x + utemp[9]/fnxc;
            }
noninertX = noninertX - utemp[9]/fnxc;
utemp[9] = 0.0;
/* Should be a WARNING printout: Nitrogen shortage when converting biodegradable
XI&XP
* Putting remaining XI&XP as lipids (50%) and carbohydrates (50%) */
xlitemp3 = 0.5*noninertX;
xchtemp3 = 0.5*noninertX;
noninertX = 0.0;
}
else { /* N in SNH enough for mapping */
xc = xc + noninertX;
noninertX = 0.0;
}
else { /* N in SND enough for mapping */
xc = xc + noninertX;
noninertX = 0.0;
}
else { /* N in XND enough for mapping */
xc = xc + noninertX;
noninertX = 0.0;
}
else { /* N in XI&XP(ASM) enough for mapping */
xc = xc + noninertX;
noninertX = 0;
}

/*============================================================================
====================*/
/* Mapping of ASM SI to ADM1 SI
* N content may be different, take first from SI-N, then SND-N, then XND-N,
* then SNH. Similar principle could be used for other states. */

inertS = 0.0;
if (fsni < fsni_adm) { /* N in SI(ASM) not enough */
inertS = utemp[0]*fsni/fsni_adm;
utemp[0] = utemp[0] - utemp[0]*fsni/fsni_adm;
if (utemp[10] < (utemp[0]*fsni_adm)) { /* N in SND not enough */
inertS = inertS + utemp[10]/fsni_adm;
utemp[0] = utemp[0] - utemp[10]/fsni_adm;
utemp[10] = 0.0;
if (utemp[11] < (utemp[0]*fsni_adm)) { /* N in XND not enough */
inertS = inertS + utemp[11]/fsni_adm;
utemp[0] = utemp[0] - utemp[11]/fsni_adm;
utemp[11] = 0.0;
if (utemp[9] < (utemp[0]*fsni_adm)) { /* N in SNH not enough */
inertS = inertS + utemp[9]/fsni_adm;
utemp[0] = utemp[0] - utemp[9]/fsni_adm;
utemp[9] = 0.0;
/* Here there should be a warning printout: Nitrogen shortage when converting SI
* Putting remaining SI as monosacharides */
utemp[1] = utemp[1] + utemp[0];
utemp[0] = 0.0;
}
else { /* N in SNH enough for mapping */
inertS = inertS + utemp[0];
    utemp[0] = 0.0;
}
}
else { /* N in XND enough for mapping */
inertS = inertS + utemp[0];
    utemp[0] = 0.0;
}
else { /* N in SND enough for mapping */
inertS = inertS + utemp[0];
    utemp[0] = 0.0;
}
else { /* N in SI(ASM) enough for mapping */
inertS = inertS + utemp[0];
    utemp[10] = utemp[10] + utemp[0]*(fsni-fsni_adm); /* put remaining N as SND */
    utemp[0] = 0.0;
}

/*============================================================================
====================*/
/* Define the outputs including charge balance */

y[0] = utemp[1]/1000.0;
y[1] = y[1]/1000.0;
y[11] = inertS/1000.0;
y[12] = xc/1000.0;
y[13] = (xchtemp + xchtemp2 + xchtemp3)/1000.0;
y[14] = (xprtemp + xprtemp2)/1000.0;
y[15] = (xlttemp + xlttemp2 + xlttemp3)/1000.0;
y[23] = (biomass_nobio + inertX)/1000.0;
y[26] = u[14]; /* flow rate */
y[27] = T_op - 273.15; /* temperature, degC */
y[28] = u[16]; /* dummy state */
y[29] = u[17]; /* dummy state */
y[30] = u[18]; /* dummy state */
y[31] = u[19]; /* dummy state */
y[32] = u[20]; /* dummy state */

/* charge balance, output S_IC */
y[9] = (utemp2[8]*alfa_NO + utemp2[9]*alfa_NH + utemp2[12]*alfa_alk) - (y[3]*alfa_va + y[4]*alfa_ac + y[5]*alfa_pro + y[6]*alfa_ac + y[10]*alfa_IN)/alfa_co2;

/* calculate anions and cations based on full charge balance including H+ and OH- */
ScatminusSan = y[3]*alfa_va + y[4]*alfa_ac + y[5]*alfa_pro + y[6]*alfa_co2 + pow(10, (-pK_w + pH_adm)) - pow(10, -pH_adm);

if (ScatminusSan > 0) {
    y[24] = ScatminusSan;
    y[25] = 0.0;
}
else {
    y[24] = 0.0;
    y[25] = -1.0*ScatminusSan;
}

/* Finally there should be a input-output mass balance check here of COD and N */
}

/* mdlUpdate - perform action at major integration time step */
static void mdlUpdate(double *x, double *u, SimStruct *S, int tid)
{

/* mdlDerivatives - compute the derivatives */
static void mdlDerivatives(double *dx, double *x, double *u, SimStruct *S, int tid)
{

/* mdlTerminate - called when the simulation is terminated. */
static void mdlTerminate(SimStruct *S)
{

#ifdef MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif

/* New version (no 3) of the ADM1 to ASM1 interface based on discussions * within the IWA TG BSM community during 2002-2006. Now also including charge * balancing and temperature dependency for applicable parameters. * Model parameters are defined in adm1init_bsm2.m * u is the input in ADM1 terminology + extra dummy states, 33 variables * plus two extra inputs: 1) dynamic pH from the ADM1 system (needed for * accurate charge balancing - also used the ASM1 to ADM1 interface) and * 2) wastewater temperature into the ASM2ADM interface, which is used as * the output temperature from the ADM2ASM interface (assume heat exchangers etc). * If temperature control of AD is used then the operational temperature * of the ADM1 should also be an input rather than a defined parameter. * Temperature in the ADM1 and the ASM1 to ADM1 and the ADM1 to ASM1 * interfaces should be identical at every time instant. * The interface assumes identical N-content of particulate inerts in both * AD and AS. The same holds for biomass. The N-content of soluble inerts may vary. * * u is the input in ADM1 terminology + extra dummy states, 33 variables * u[0] : Ssu = monosacharides (kg COD/m3) * u[1] : Saa = amino acids (kg COD/m3) * u[2] : Sfa = long chain fatty acids (LCFA) (kg COD/m3)
* u[3] : Sva = total valerate (kg COD/m3)
* u[4] : Sbu = total butyrate (kg COD/m3)
* u[5] : Spro = total propionate (kg COD/m3)
* u[6] : Sac = total acetate (kg COD/m3)
* u[7] : Sh2 = hydrogen gas (kg COD/m3)
* u[8] : Sch4 = methane gas (kg COD/m3)
* u[9] : Sic = inorganic carbon (kmole C/m3)
* u[10] : Sin = inorganic nitrogen (kmole N/m3)
* u[11] : Si = soluble inerts (kg COD/m3)
* u[12] : Xc = composites (kg COD/m3)
* u[13] : Xch = carbohydrates (kg COD/m3)
* u[14] : Xpr = proteins (kg COD/m3)
* u[15] : Xli = lipids (kg COD/m3)
* u[16] : Xsu = sugar degraders (kg COD/m3)
* u[17] : Xaa = amino acid degraders (kg COD/m3)
* u[18] : Xfa = LCFA degraders (kg COD/m3)
* u[19] : Xc4 = valerate and butyrate degraders (kg COD/m3)
* u[20] : Xpro = propionate degraders (kg COD/m3)
* u[21] : Xac = acetate degraders (kg COD/m3)
* u[22] : Xh2 = hydrogen degraders (kg COD/m3)
* u[23] : Xi = particulate inerts (kg COD/m3)
* u[24] : scat+ = cations (metallic ions, strong base) (kmole/m3)
* u[25] : san- = anions (metallic ions, strong acid) (kmole/m3)
* u[26] : flow rate (m^3/d)
* u[27] : temperature (deg C)
* u[28-32] : dummy states for future use
* u[33] : dynamic pH from the ADM1
* u[34] : wastewater temperature into the ASM2ADM interface, deg C

* Output vector:
  * y[0] : Si = soluble inert organic material (g COD/m3)
  * y[1] : Ss = readily biodegradable substrate (g COD/m3)
  * y[2] : Xi = particulate inert organic material (g COD/m3)
  * y[3] : Xs = slowly biodegradable substrate (g COD/m3)
  * y[4] : Xbh = active heterotrophic biomass (g COD/m3)
  * y[5] : Xba = active autotrophic biomass (g COD/m3)
  * y[6] : Xp = particulate product arising from biomass decay (g COD/m3)
  * y[7] : So = oxygen (g -COD/m3)
  * y[8] : Sno = nitrate and nitrite nitrogen (g N/m3)
  * y[9] : Snh = ammonia and ammonium nitrogen (g N/m3)
  * y[10] : Snd = soluble biogradable organic nitrogen (g N/m3)
  * y[11] : Xnd = particulate biogradable organic nitrogen (g N/m3)
  * y[12] : Salk = alkalinity (mole HCO3-/m3)
  * y[13] : TSS = total suspended solids (internal use) (mg SS/l)
  * y[14] : flow rate (m^3/d)
  * y[15] : temperature (deg C)
  * y[16-20] : dummy states for future use

* ADM1 --> ASM1 conversion, version 3 for BSM2
* Copyright: John Copp, Primodal Inc., Canada; Ulf Jeppsson, Lund
  * University, Sweden; Damien Batstone, Univ of Queensland,
  * Australia, Ingmar Nopens, Univ of Ghent, Belgium,
  * Marie-Noelle Pons, Nancy, France, Peter Vanrolleghem,
  * Univ. Laval, Canada, Jens Alex, IFAK, Germany and
  * Eveline Volcke, Univ of Ghent, Belgium.
 */

#define S_FUNCTION_NAME adm2asm_v3_bsm2
#include "simstruc.h"
#include <math.h>

#define PAR    ssGetArg(S,0)

/*
 * mdlInitializeSizes - initialize the sizes array
 */
static void mdlInitializeSizes(SimStruct *S)
{
  ssSetNumContStates( S, 0); /* number of continuous states */
  ssSetNumDiscStates( S, 0); /* number of discrete states */
  ssSetNumInputs( S, 35); /* number of inputs */
  ssSetNumOutputs( S, 21); /* number of outputs */
  ssSetDirectFeedThrough(S, 1); /* direct feedthrough flag */
  ssSetNumSampleTimes( S, 1); /* number of sample times */
  ssSetNumSFcnParams( S, 1); /* number of input arguments */
  ssSetNumRWork( S, 0); /* number of real work vector elements */
  ssSetNumIWork( S, 0); /* number of integer work vector elements */
  ssSetNumPWork( S, 0); /* number of pointer work vector elements */
}

/*
 * mdlInitializeSampleTimes - initialize the sample times array
 */
static void mdlInitializeSampleTimes(SimStruct *S)
{
  ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
  ssSetOffsetTime(S, 0, 0.0);
}

/*
 * mdlInitializeConditions - initialize the states
 */
static void mdlInitializeConditions(double *x0, SimStruct *S)
{
}

/*
 * mdlOutputs - compute the outputs
 */
static void mdlOutputs(double *y, double *x, double *u, SimStruct *S, int tid)
{
  double CODequiv, fnaa, fnxc, fnbac, fni, fni_adm, frlixs, frlibac, frxs_adm, fdegrade_adm, frxs_as, fdegrade_as;
  double  R, T_base, T_op, pK_w_base, pK_a_va_base, pK_a_pro_base, pK_a_ac_base, pK_a_co2_base, pK_a_IN_base;
  double  pH_adm, pK_a_co2, pK_a_IN, alfa_va, alfaBu, alfa_pro, alfa_ac, alfa_co2, alfa_IN, alfa_NH, alfa_alk, alfa_NO, factor;
  double XPtemp, XStemp, XStemp2;
  double biomass, biomass_nobio, biomass_bioN, remainCOD, inertX, noninertX, inertS, utemp[35];
  int i;

  /* parameters defined in adm1init_bsm2.m, INTERFACEPAR */
  CODequiv = mxGetPr(PAR)[0]; /* not used in ADM2ASM */
  fnaa = mxGetPr(PAR)[1];
  fnxc = mxGetPr(PAR)[2];
  fnbac = mxGetPr(PAR)[3];
  fni = mxGetPr(PAR)[4];

  /*...*/
fsni = mxGetPr(PAR)[5];
fsni_adm = mxGetPr(PAR)[6];
frlixs = mxGetPr(PAR)[7];    /* not used in ADM2ASM */
frlibac = mxGetPr(PAR)[8];   /* not used in ADM2ASM */
frxs_adm = mxGetPr(PAR)[9];  /* not used in ADM2ASM */
fdgrade_adm = mxGetPr(PAR)[10]; /* not used in ADM2ASM */
frxs_as = mxGetPr(PAR)[11];
fgrade_as = mxGetPr(PAR)[12];
R = mxGetPr(PAR)[13];
T_base = mxGetPr(PAR)[14];
T_op = mxGetPr(PAR)[15];    /* should be an input variable if dynamic temperature control is used */
pK_w_base = mxGetPr(PAR)[16];
pK_a va_base = mxGetPr(PAR)[17];
pK_bu_base = mxGetPr(PAR)[18];
pK_a pro_base = mxGetPr(PAR)[19];
pK_a co2_base = mxGetPr(PAR)[20];
pK_a IN_base = mxGetPr(PAR)[21];
pK_a IN_base = mxGetPr(PAR)[22];

factor = (1.0/T_base - 1.0/T_op)/(100.0*R);
pK_w = pK_w_base - log10(exp(55900.0*factor));
pK_a co2 = pK_a co2_base - log10(exp(7646.0*factor));
pK_a IN = pK_a IN_base - log10(exp(51965.0*factor));
alfa_va = 1.0/208.0*(-1.0/(1.0 + pow(10, pK_a va_base - pH_adm)));
alfa_bu = 1.0/160.0*(-1.0/(1.0 + pow(10, pK_bu_base - pH_adm)));
alfa pro = 1.0/120.0*(-1.0/(1.0 + pow(10, pK_a pro_base - pH_adm)));
alfa ac = 1.0/64.0*(-1.0/(1.0 + pow(10, pK_a ac_base - pH_adm)));
alfa co2 = -1.0/(1.0 + pow(10, pK_a co2_base - pH_adm));
alfa IN = (pow(10, pK_a IN_base - pH_adm))/(1.0 + pow(10, pK_a IN_base - pH_adm));
alfa NH = 1.0/14000.0;    /* convert mgN/l into kmoleN/m3 */
alfa alk = -0.001;        /* convert moleHCO3/m3 into kmoleHCO3/m3 */
alfa NO = -1.0/14000.0;   /* convert mgN/l into kmoleN/m3 */

for (i = 0; i < 35; i++)
    utemp[i] = u[i];
for (i = 0; i < 21; i++)
y[i] = 0.0;

/*============================================================================
====================*/
/* Biomass becomes part of XS and XP when transformed into ASM */
/* Assume Npart of formed XS to be fnxc and Npart of XP to be fxni */
/* Remaining N goes into the ammonia pool (also used as source if necessary) */

biomass_nobio = biomass*(1.0 - frxs_as);    /* part which is mapped to XP */
biomass_bioN = (biomass*fnbac - biomass_nobio*fxni);
remainCOD = 0.0;
if (biomass_bioN < 0.0) {
    /* Problems: if here we should print 'WARNING: not enough biomass N to map the requested inert part of biomass' */
    /* We map as much as we can, and the remains go to XS! */
    XPtemp = biomass*fnbac/fxni;
    biomass_nobio = XPtemp;
biomass_bioN = 0.0;

} else {
    XPtemp = biomass_nobio;
}

if ((biomass_bioN/fnxc) <= (biomass - biomass_nobio)) {
    XStemp = biomass_bioN/fnxc; /* all biomass N used */
    remainCOD = biomass - biomass_nobio - XStemp;
    if ((utemp[10]*14000.0/fnaa) >= remainCOD) { /* use part of remaining S_IN to form XS */
        XStemp = XStemp + remainCOD;
    }
    else {
        /* Problems: if here we should print 'ERROR: not enough nitrogen to map the requested XS part of biomass' */
        /* System failure! */
    }
}
else {
    XStemp = biomass - biomass_nobio; /* all biomass COD used */
}

utemp[10] = utemp[10] + biomass*fnbac/14000.0 - XPtemp*fxni/14000.0 - XStemp*fnxc/14000.0; /* any remaining N in S_IN */
y[6] = XPtemp; /* inert part of biomass */

/*============================================================================
====================*/
/* mapping of inert XI in AD into XI and possibly XS in AS */
/* assumption: same N content in both ASM1 and ADM1 particulate inerts */
/* special case: if part of XI in AD can be degraded in AS */
/* we have no knowledge about the contents so we put it in as part substrate (XS) */
/* we need to keep track of the associated nitrogen */
/* N content may be different, take first from XI-N then S_IN, */
/* * Similar principle could be used for other states. */
inertX = (1.0-fdegrade_as)*utemp[23]*1000.0;
XStemp2 = 0.0;
noninertX = 0.0;
if (fdegrade_as > 0.0) {
    noninertX = fdegrade_as*utemp[23]*1000.0;
    if (fxni < fnxc) { /* N in XI(AD) not enough */
        XStemp2 = noninertX*fxni/fnxc;
        noninertX = noninertX - noninertX*fxni/fnxc;
        if ((utemp[10]*14000.0) < (noninertX*fnxc)) { /* N in SNH not enough */
            XStemp2 = XStemp2 + (utemp[10]*14000.0)/fnxc;
            noninertX = noninertX - (utemp[10]*14000.0)/fnxc;
            utemp[10] = 0.0;
        }
    } /* Problems: if here we should print 'WARNING: Nitrogen shortage when converting biodegradable XI' */
    /* Mapping what we can to XS and putting remaining XI back into XI of ASM */
    inertX = inertX + noninertX;
} else { /* N in S_IN enough for mapping */
    XStemp2 = XStemp2 + noninertX;
    noninertX = 0.0;
}
else { /* N in XI(AD) enough for mapping */
    XStemp2 = XStemp2 + noninertX;
    utemp[10] = utemp[10] + noninertX*(fxni - fnxc)/14000.0; /* put remaining N as S_IN */
    noninertX = 0;
}
}

y[2] = inertX; /* Xi = Xi*fdegrate_AS + possibly nonmappable XS */

/*============================================================================
====================*/
/* Mapping of ADM SI to ASM1 SI */
* It is assumed that this mapping will be 100% on COD basis
  * N content may be different, take first from SI-N then from S_IN.
  * Similar principle could be used for other states. */

inertS = 0.0;
if (fsni_adm < fsni) { /* N in SI(AD) not enough */
    inertS = utemp[11]*fsni_adm/fsni;
    if ((utemp[10]*14.0) < (utemp[11]*fsni)) { /* N in S_IN not enough */
        inertS = inertS + utemp[10]*14.0/fsni;
        utemp[10] = 0.0;
        /* Problems: if here we should print 'ERROR: Nitrogen shortage when converting SI' */
        /* System failure: nowhere to put SI */
    } else { /* N in S_IN enough for mapping */
        inertS = inertS + utemp[11];
        utemp[11] = 0.0;
    }
}
else { /* N in SI(AD) enough for mapping */
    inertS = inertS + utemp[11];
    utemp[11] = 0.0;
}

y[0] = inertS*1000.0; /* Si = Si */

/*============================================================================
====================*/
/* Define the outputs including charge balance */
/* nitrogen in biomass, composites, proteins */
* Xnd is the nitrogen part of Xs in ASM1. Therefore Xnd should be based on the
  * same variables as constitutes Xs, ie AD biomass (part not mapped to XP), xc and xpr if we assume
  * there is no nitrogen in carbohydrates and lipids. The N content of Xi is
  * not included in Xnd in ASM1 and should in my view not be included. */

y[11] = fnxc*(XStemp + XStemp2) + fnxc*1000.0*utemp[12] + fnaa*1000.0*utemp[14];
/* Snd is the nitrogen part of Ss in ASM1. Therefore Snd should be based on the
   * same variables as constitutes Ss, and we assume
* there is only nitrogen in the amino acids. The N content of Si is
* not included in Snd in ASM1 and should in my view not be included. */

```c
y[10] = fnaa*1000.0*utemp[1];
/* sh2 and sch4 assumed to be stripped upon reentry to ASM side */
= sum all S except Sh2, Sch4, Si, Sic, Sin */
y[9] = utemp[10]*14000.0;   /* Snh = S_IN including adjustments above */
y[14] = utemp[26];   /* flow rate */
y[15] = u[34];      /* temperature, degC, should be equal to AS temperature into the AD/AS interface */
y[16] = utemp[28];    /* dummy state */
y[17] = utemp[29];    /* dummy state */
y[18] = utemp[30];    /* dummy state */
y[19] = utemp[31];    /* dummy state */
y[20] = utemp[32];    /* dummy state */
/* charge balance, output S_alk (molHCO3/m3) */
y[12] = (u[3]*alfa_va + u[4]*alfa_bu + u[5]*alfa_pro + u[6]*alfa_ac + u[9]*alfa_co2 + u[10]*alfa_IN -
y[8]*alfa_NO - y[9]*alfa_NH)/alfa_alk;
/* Finally there should be a input-output mass balance check here of COD and N */
```

```c
/* mdlUpdate - perform action at major integration time step */
static void mdlUpdate(double *x, double *u, SimStruct *S, int tid)
{
}
/*
 * mdlDerivatives - compute the derivatives */
static void mdlDerivatives(double *dx, double *x, double *u, SimStruct *S, int tid)
{
}
/*
 * mdlTerminate - called when the simulation is terminated. */
static void mdlTerminate(SimStruct *S)
{
}
#endif  /* MATLAB_MEX_FILE */ /* Is this file being compiled as a MEX-file? */
#include "simulink.c"    /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"     /* Code generation registration function */
#endif
```
A2.1. FORTRAN code

A2.2.1. ASM/ADM

SUBROUTINE asm2admv2(asmm, adm, totCODin, totCODout, totNin, totNout, & ph, ancat, warn)

IMPLICIT NONE
INTEGER i, warn
DOUBLE PRECISION fnaa, fnxc, fnxb, fni, fchi, flix
DOUBLE PRECISION fnbac, fni, fni, fli
DOUBLE PRECISION xx, nxb, nxc, naa, ni, nbac, xni, sni, sni_adm
DOUBLE PRECISION fli, fnx, frli, fdegrade
DOUBLE PRECISION CODEquiv, CODdemand
DOUBLE PRECISION remaina, remainb, remainc, remaind, remainCOD
DOUBLE PRECISION demand
DOUBLE PRECISION sorgn, remorgn, sinertn, remCOD, xinertn, xorgn
DOUBLE PRECISION xprtemp, xlneg, xchtemp, biomass_nobio, biomass
DOUBLE PRECISION biomass_bioN, xprtemp2, xlneg2, xchtemp2
DOUBLE PRECISION inertX, xc, xlneg3, xchtemp3, noninertX, inertS
DOUBLE PRECISION totTKNin, totCODin, totCODout, totTKNout
DOUBLE PRECISION totNin, totNout, ph, ancat, pkk, tfac, bigr

c New version of ASM1 ADM1 interface based on discussions
c within the BSM community during 2002-2006.

c asmm is the input in ASM1 terminology, 13 state variables
asmm[1] : Si = soluble inert organic material (g COD/m3)
asmm[2] : Ss = readily biodegradable substrate (g COD/m3)
asmm[3] : Xi = particulate inert organic material (g COD/m3)
asmm[4] : Xs = slowly biodegradable substrate (g COD/m3)
asmm[5] : Xbh = active heterotrophic biomass (g COD/m3)
asmm[6] : Xba = active autotrophic biomass (g COD/m3)
asmm[7] : Xp = particulate product arising from biomass decay (g COD/m3)
asmm[8] : So = oxygen (g -COD/m3)
asmm[9] : Sno = nitrate and nitrite nitrogen (g N/m3)
asmm[10] : Snh = ammonia and ammonium nitrogen (g N/m3)
asmm[11] : Snd = soluble biogradable organic nitrogen (g N/m3)
asmm[12] : Xnd = particulate biogradable organic nitrogen (g N/m3)
asmm[13] : Salk = alkalinity (mole HCO3- /m3)
DOUBLE PRECISION asmm(13), xtemp(13)

c adm is the output in ADM1 terminology, 24 state variables
adm[1] : Ssu = monosacharides (kg COD/m3)
ad[2] : Saa = amino acids (kg COD/m3)
ad[3] : Sfa = long chain fatty acids (LCFA) (kg COD/m3)
ad[4] : Sva = total valerate (kg COD/m3)
ad[5] : Sbu = total butyrate (kg COD/m3)
ad[6] : Spro = total propionate (kg COD/m3)
ad[7] : Sac = total acetate (kg COD/m3)
ad[8] : Sh2 = hydrogen gas (kg COD/m3)
ad[9] : Sch4 = methane gas (kg COD/m3)
ad[10] : Sic = inorganic carbon (kmole C/m3)
ad[11] : Sin = inorganic nitrogen (kmole N/m3)
ad[12] : Si = soluble inerts (kg COD/m3)
ad[13] : Xc = composites (kg COD/m3)
ad[14] : Xch = carbohydrates (kg COD/m3)
ad[15] : Xpr = proteins (kg COD/m3)
ad[16] : Xli = lipids (kg COD/m3)
ad[17] : Xsu = sugar degraders (kg COD/m3)
c adm[18] : Xaa = amino acid degraders (kg COD/m3)
c adm[19] : Xfa = LCFA degraders (kg COD/m3)
c adm[20] : Xc4 = valerate and butyrate degraders (kg COD/m3)
c adm[21] : Xpro = propionate degraders (kg COD/m3)
c adm[22] : Xac = acetate degraders (kg COD/m3)
c adm[23] : Xh2 = hydrogen degraders (kg COD/m3)
c adm[24] : Xi = particulate inerts (kg COD/m3)

DOUBLE PRECISION adm(24)
DOUBLE PRECISION alfachac, alfachpro, alfachbu, alfachva
DOUBLE PRECISION alfachin, alfachic, alfachnh
DOUBLE PRECISION alfachalk, alfachno
DOUBLE PRECISION chargeasm1, chargeadm1

tfac=1.d00/298.15d00-1.d00/(273.15d00+35.d00)
bigr=0.08314d-00
c

c ASM1 --> ADM1 conversion

Copyright: John Copp, Primodal Inc., Canada Ulf Jeppsson, Lund
University, Sweden Damien Batstone, Univ of Queensland,
Australia and Ingmar Nopens, Univ of Ghent, Belgium

do i=1,13
   xtemp(i)=asmm(i)
endo
do i=1,24
   adm(i) = 0.d0
endo

warn=0

c Parameters

CODequiv is the conversion factor for COD demand of nitrate
c exact value of ASM1 2.86
   CODequiv = 40.d0/14.d0
c fraction of N in amino acids and Xpr as in ADM1 report
   fnaa = 0.098d0
   N content of biomass based on BSM1, same in AS and AD
   fnbac = 0.08d0
   N content of composite material based on BSM2
   fnxc = 0.0376d0
   N content of inerts XI and XP, same in AS and AD
   fxni = 0.06d0
   N content of SI, zero in ASM1 and BSM1
   fsni = 0.d0
   N content of SI in the AD system
   fsni_adm = 0.06d0

c fnbac, fxni and fsni are adjusted to fit the benchmark values of iXB=0.08
   and iXP=0.06 and iSI=0.
c i.e 8% N content mgCOD/l <-> mgN/l = iXB, in ADM1 8.75%
   nbac = fnbac/14.d0*14000.d0
   nxc = fnxc/14.d0*14000.d0
   naa = fnaa/14.d0*14000.d0
   xni = fxni/14.d0*14000.d0
   xni = fxni/14.d0*14000.d0
   sni = fsni/14.d0*14000.d0
   sni_adm = fsni_adm/14.d0*14000.d0
c lipid fraction of non-nitrogenous XS in BSM2
frlixs = 0.7d0

c anaerobically degradable fraction of biomass in BSM2
frxs = 0.68d0

c lipid fraction of non-nitrogenous biomass in BSM2
frlixb = 0.4d0

c amount of XI and XP degradable in AD, zero in BSM2
fdegrade = 0.d0

c Let CODdemand be the COD demand of available electron
c acceptors prior to the anaerobic digester, i.e. oxygen and nitrate
CODdemand = asmm(8) + CODequiv*asmm(9)
xtemp(8) = 0.d0
xtemp(9) = 0.d0

c if extreme detail was used then some extra NH4 would be transformed
c into N bound in biomass and some biomass would be formed when
c removing the CODdemand (based on the yield). But on a total COD balance
c approach the below is correct (neglecting the N need for biomass growth)

if (CODdemand .GT. asmm(2)) then
  remaina = CODdemand - asmm(2)
  xtemp(2) = 0.d0
  if (remaina .GT. asmm(4)) then
    remainb = remaina - asmm(4)
    xtemp(4) = 0.d0
    if (remainb .GT. asmm(5)) then
      remainc = remainb - asmm(5)
      xtemp(10) = xtemp(10) + asmm(5)*fnbac
      xtemp(5) = 0.d0
      if (remainc .GT. asmm(6)) then
        remaind = remainc - asmm(6)
        xtemp(10) = xtemp(10) + asmm(6)*fnbac
        xtemp(6) = 0.d0
        c if here we run into problems, carbon shortage
        xtemp(8) = remaind
      else
        xtemp(6) = asmm(6) - remainc
        xtemp(10) = xtemp(10) + remainc*fnbac
        endif
    else
      xtemp(5) = asmm(5) - remainb
      xtemp(10) = xtemp(10) + remainb*fnbac
    endif
  else
    xtemp(4) = asmm(4) - remaina
  endif
else
  xtemp(2) = asmm(2) - CODdemand
endif

c SS becomes part of amino acids when transformed into ADM

c and any remaining SS is mapped to monosacharides (no N contents)
c Enough SND must be available for mapping to amino acids
c Saa COD equivalent to SND
sorgn = asmm(11)/fnaa

if (sorgn .GE. xtemp(2)) then
c not all SND-N fits into amino acids
c map all SS COD into Saa
   adm(2) = xtemp(2)
c the remaining part of SND
   xtemp(11) = xtemp(11) - xtemp(2) * fnaa
c all SS used
   xtemp(2) = 0.d0
else
c all SND-N fits into amino acids
c map all SND related COD into Saa
   adm(2) = sorgn
c the rest of the SS COD, later mapped into sugar
   xtemp(2) = xtemp(2) - sorgn
c all SND used
   xtemp(11) = 0.d0
endif

C XS becomes part of Xpr (proteins) when transformed into ADM
C and any remaining XS is mapped to Xch and Xli (no N contents)
c Enough XND must be available for mapping to Xpr

c Xpr COD equivalent to XND
   xorgn = asmm(12)/fnaa
   if (xorgn .GE. xtemp(4)) then
      c not all XND-N fits into Xpr
      c map all XS COD into Xpr
      xprtemp = xtemp(4)
c the remaining part of XND
      xtemp(12) = xtemp(12) - xtemp(4) * fnaa
      c all XS used
      xtemp(4) = 0.d0
      xlitemp = 0.d0
      xchtemp = 0.d0
   else
      c all XND-N fits into Xpr
      c map all XND related COD into Xpr
      xprtemp = xorgn
      c part of XS COD not associated with N
      xlitemp = frlixs*(xtemp(4) - xorgn)
c part of XS COD not associated with N
      xchtemp = (1.d0-frlixs)*(xtemp(4) - xorgn)
c all XS used
      xtemp(4) = 0.d0
      c all XND used
      xtemp(12) = 0.d0
   endif

C Biomass becomes part of Xpr and XI when transformed into ADM
C and any remaining XBH and XBA is mapped to Xch and Xli (no N contents)
c Remaining XND-N can be used as nitrogen source to form Xpr
   biomass = xtemp(5) + xtemp(6)
c part which is mapped to XI
   biomass_nobio = biomass*(1.d0-frxs)
bioN = (biomass*fnbac - biomass_nobio*fxni)
   if (bioN .LT. 0.d0) then
      disp('ERROR: not enough biomass N to map the requested inert part')
      warn=1
   endif
   if ((bioN/fnna) .LE. (biomass - biomass_nobio)) then
      c all biomass N used

xprtemp2 = biomass_bioN/fnaa
remainCOD = biomass - biomass_nobio - xprtemp2

c use part of remaining XND-N to form proteins
  if ((xtemp(12)/fnaa) .GT. remainCOD) then
    xprtemp2 = xprtemp2 + remainCOD
    xtemp(12) = xtemp(12) - remainCOD*fnaa
    remainCOD = 0.d0
    xtemp(5) = 0.d0
    xtemp(6) = 0
  endif

c use all remaining XND-N to form proteins
  else
    xprtemp2 = xprtemp2 + xtemp(12)/fnaa
    remainCOD = remainCOD - xtemp(12)/fnaa
    xtemp(12) = 0.d0
  endif

c part of the COD not associated with N
  xlitemp2 = frlixb*remainCOD

c part of the COD not associated with N
  xchtemp2 = (1.d0-frlixb)*remainCOD
  else
    c all biomass COD used
    xprtemp2 = biomass - biomass_nobio

  c any remaining N in XND
    xtemp(12) = xtemp(12) + biomass*fnbac - biomass_nobio*fxni
    &- xprtemp2*fnaa
  endif
  xtemp(5) = 0.d0
  xtemp(6) = 0.d0

c direct mapping of XI and XP to ADM1 XI

c assumption: same N content in both ASM1 and ADM1 particulate inerts
  inertX = (1-fdegrade)*(xtemp(3) + xtemp(7))

c special case: if part of XI and XP in ASM can be degraded in AD

c we have no knowledge about the contents so we put it in as composites (xc)
c we need to keep track of the associated nitrogen

c N content may be different, take first from XI&XP-N, then XND-N, then
SND-N,
c then SNH. Similar principle could be used for other states.
  xc = 0.d0
  xlitemp3 = 0.d0
  xchtemp3 = 0.d0
  if (fdegrade .GT. 0.d0) then
    noninertX = fdegrade*(xtemp(3) + xtemp(7))
    c N in XI&XP(ASM) not enough
      if ((noninertX*fxni) .LT. (noninertX*fnxc)) then
        xc = noninertX*fxni/fnxc
        noninertX = noninertX - noninertX*fxni/fnxc
      endif
  c N in XND not enough
    if (xtemp(12) .LT. (noninertX*fnxc)) then
      xc = xc + xtemp(12)/fnxc
      noninertX = noninertX - xtemp(12)/fnxc
      xtemp(12) = 0.d0
  c N in SND not enough
    if (xtemp(11) .LT. (noninertX*fnxc)) then
      xc = xc + xtemp(11)/fnxc
      noninertX = noninertX - xtemp(11)/fnxc
      xtemp(11) = 0.d0
  c N in SNH not enough
    if (xtemp(10) .LT. (noninertX*fnxc)) then
      xc = xc + xtemp(10)/fnxc
noninertX = noninertX - xtemp(10)/fnxc
xtemp(10) = 0.d0
c disp('ERROR: Nitrogen shortage when converting biodegradable XI&XP')
c disp('Putting remaining XI&XP as lipids (50%) and carbohydrates (50%)')
xlitemp3 = 0.5d0*noninertX
xchtemp3 = 0.5d0*noninertX
noninertX = 0.d0
c N in SNH enough for mapping
else
  xc = xc + noninertX
  xtemp(10) = xtemp(10) - noninertX*fnxc
  noninertX = 0.d0
endif
c N in SND enough for mapping
else
  xc = xc + noninertX
  xtemp(11) = xtemp(11) - noninertX*fnxc
  noninertX = 0.d0
endif
c N in XND enough for mapping
else
  xc = xc + noninertX
  xtemp(12) = xtemp(12) - noninertX*fnxc
  noninertX = 0.d0
endif
c N in XI&XP(ASM) enough for mapping
else
  xc = xc + noninertX
  xtemp(12) = xtemp(12) + noninertX*(fxni-fnxc)
  noninertX = 0.d0
endif
c N in SI(ASM) not enough
if ((xtemp(1)*fsni) .LT. (xtemp(1)*fsni_adm)) then
  inertS = xtemp(1)*fsni/fsni_adm
  xtemp(1) = xtemp(1) - xtemp(1)*fsni/fsni_adm
endif
c N in SND not enough
if (xtemp(11) .LT. (xtemp(1)*fsni_adm)) then
  inertS = inertS + xtemp(11)/fsni_adm
  xtemp(11) = xtemp(11) - xtemp(11)/fsni_adm
  xtemp(11) = 0.d0
endif
c N in XND not enough
if (xtemp(12) .LT. (xtemp(1)*fsni_adm)) then
  inertS = inertS + xtemp(12)/fsni_adm
  xtemp(1) = xtemp(1) - xtemp(12)/fsni_adm
  xtemp(12) = 0.d0
endif
c N in SNH not enough
if (xtemp(10) .LT. (xtemp(1)*fsni_adm)) then
  inertS = inertS + xtemp(10)/fsni_adm
  xtemp(1) = xtemp(1) - xtemp(10)/fsni_adm
  xtemp(10) = 0.d0
endif
c disp('ERROR: Nitrogen shortage when converting SI')
c disp('Putting remaining SI as monosacharides')
  xtemp(2) = xtemp(2) + xtemp(1)
xtemp(1) = 0.d0
c N in SNH enough for mapping
else
  inertS = inertS + xtemp(1)
  xtemp(10) = xtemp(10) - xtemp(1)*fsni_adm
  xtemp(1) = 0.d0
end if

c N in XND enough for mapping
else
  inertS = inertS + xtemp(1)
  xtemp(12) = xtemp(12) - xtemp(1)*fsni_adm
  xtemp(1) = 0.d0
end if

c N in SND enough for mapping
else
  inertS = inertS + xtemp(1)
  xtemp(11) = xtemp(11) - xtemp(1)*fsni_adm
  xtemp(1) = 0.d0
end if

c N in SI(ASM) enough for mapping
else
  inertS = inertS + xtemp(1)
c put remaining N as SND
  xtemp(11) = xtemp(11) + xtemp(1)*(fsni-fsni_adm)
  xtemp(1) = 0.d0
endif

adm(1) = xtemp(2)/1000.d0
adm(2) = adm(2)/1000.d0
c
adm(10) = xtemp(13)/1000.d0
adm(11) = (xtemp(10) + xtemp(11) + xtemp(12))/14000
adm(12) = inertS/1000.d0
adm(13) = xc/1000.d0
adm(14) = (xchtemp + xchtemp2 + xchtemp3)/1000.d0
adm(15) = (xprtemp + xprtemp2)/1000.d0
adm(16) = (xlitemp + xlitemp2 + xlitemp3)/1000.d0
adm(24) = (biomass_nobio + inertX)/1000.d0

C Calculation of adm(10) (Sic)
alfachac=(-(1.d00/64.d00)/(1.d00+10.d00**(4.76d00 -ph))
alfachpro=-(1.d00/112.d00)/(1.d00+10.d00**(4.88d00 -ph))
alfachbu=-(1.d00/160.d00)/(1.d00+10.d00**(4.82d00 -ph))
alfachva=-(1.d00/208.d00)/(1.d00+10.d00**(4.86d00 -ph))
pkk=-dlog10((10.d00**(-9.25d0))
& *dexp(51965.d00/bigr/100.d00*tfac))
alfachin=(10.d00**(pkk-ph))/(1.d00+10.d00**(pkk-ph))
pkk=-dlog10((10.d00**(-6.35d0))
& *dexp(7646.d00/bigr/100.d00*tfac))
alfachic=-1.d00/(1.d00+10.d00**(pkk-ph))
c modifie / PV
alfachnh=1.d00/14.d00
alfachno=-1.d00/14.d00
alfachalk=-1.d00
chargeasm1=(asmm(13)*alfachalk+asmm(10)*alfachnh+
& asmm(9)*alfachno)/1000.
chargeadm1=adm(7)*alfachac+adm(6)*alfachpro+
& adm(5)*alfachbu+adm(4)*alfachva+adm(11)*alfachin
adm(10)=(chargeasm1-chargeadm1)/alfachic
pkk=-dlog10((10.d00**(-14)))*dexp(55900.d00/bigr/100.d00*tfac))
ancat=chargeadm1+adm(10)*alfachic-10.d0**(-ph)+10.d0**(-pkk+ph)
c check mass balances
totCODin = 0.d0
do i=1,7
    totCODin=totCODin+asmm(i)
enddo

totNin = asmm(9) + asmm(10) + asmm(11) + asmm(12) +
&nbac/1000.d0*(asmm(5) + asmm(6)) + sni/1000.d0*asmm(1) +
&xni/1000.d0*(asmm(3) + asmm(7))

totCODout=0.d0
   do i=1,9
    totCODout=totCODout+adm(i)*1000.d0
   enddo
   do i=12,24
    totCODout=totCODout+adm(i)*1000.d0
   enddo
   totNout = nbac*(adm(17)+adm(18)+adm(19)+adm(20)+adm(21)
&+adm(22)+adm(23)) + naa*(adm(2) + adm(15)) + adm(11)*14000.d0 +
&sni_adm*adm(12) + nxc*adm(13) + xni*adm(24)

RETURN
END
A2.2.2. ADM/ASM

SUBROUTINE adm2asmv2(adm,asmm,totCODin,totCODout,totTKNin, &totTKNout,ph,warn)

IMPLICIT NONE
INTEGER i,warn

C ADM2ASM Transformation model for conversion of ADM1 variables
C into ASM1 variables.
C ADM2ASM(x) returns the 13 state variables of the ASM1. The
C input vector x represents the first 24 state variables of ADM1. A number
C of parameters are required and are currently defined within this file. As
C many of them are also used in ASM1 and ADM1 they should probably only
C be defined in the initialisation files for those models to avoid
C different values if they are changed in one file and not in another.

C Parameters:
C fnaa          fraction of N in amino acids as in ADM1 report (default
C                0.098)
C fnxc          N content of composites adjusted from ADM1 report
C                (default 0.0376)
C fnbac          N content of biomass based on BSM1 (default 0.08)
C fsni_adm       N content of soluble inerts
C fnbac and fsni_adm are adjusted to fit the benchmark values of iXB=0.08
C and
C iXP=0.06.
C fxni N content of inerts XI and XP, same in AS and AD
C fni = N content of SI, zero in ASM and BSM2
C nbac= fnbac/14*14000 conversion into kmol N/m3 from mg N/l for biomass
C nxb = fnxb/14*14000 conversion into kmol N/m3 from mg N/l for biomass
C nxc = fnxc/14*14000 conversion into kmol N/m3 from mg N/l for
C composites
C nnaa = fnaa/14*14000 conversion into kmol N/m3 from mg N/l for amino
C acids
C ni = fni/14*14000 conversion into kmol N/m3 from mg N/l for
C inerts

DOUBLE PRECISION fnaa,fnxc,fnbac,fsni_adm,nxc,naa,ni
DOUBLE PRECISION iXI,fxni,fsni,nbac,sni_adm,xni,snl
DOUBLE PRECISION frxs_AS,fdegrade_AS

C adm is the input in ADM1 terminology, 24 variables
C adm[1] : Ssu = monosacharides (kg COD/m3)
C adm[2] : Saa = amino acids (kg COD/m3)
C adm[3] : Sfa = long chain fatty acids (LCFA) (kg COD/m3)
C adm[4] : Sva = total valerate (kg COD/m3)
C adm[5] : Sbu = total butyrate (kg COD/m3)
C adm[6] : Spro = total propionate (kg COD/m3)
C adm[7] : Sac = total acetate (kg COD/m3)
C adm[8] : Sh2 = hydrogen gas (kg COD/m3)
C adm[9] : Sch4 = methane gas (kg COD/m3)
C adm[10]: Sic = inorganic carbon (kmole C/m3)
C adm[11]: Sin = inorganic nitrogen (kmole N/m3)
C adm[12] : Sinert = soluble inerts (kg COD/m3)
C adm[13] : Xc = composites (kg COD/m3)
C adm[14] : Xch = carbohydrates (kg COD/m3)
C adm[15] : Xpr = proteins (kg COD/m3)
C adm[16] : Xli = lipids (kg COD/m3)
C adm[17] : Xsu = sugar degraders (kg COD/m3)
C adm[18] : Xaa = amino acid degraders (kg COD/m3)
Benchmark Simulation Model no. 2 (BSM2)

c adm[19] : Xfa = LCFA degraders (kg COD/m3)
c adm[20] : Xc4 = valerate and butyrate degraders (kg COD/m3)
c adm[21] : Xpro = propionate degraders (kg COD/m3)
c adm[22] : Xac = acetate degraders (kg COD/m3)
c adm[23] : Xh2 = hydrogen degraders (kg COD/m3)
c adm[24] : xinert = particulate inerts (kg COD/m3)

DOUBLE PRECISION adm(24)
c

c asmm is the output in ASM1 terminology, 13 variables

c asmm[1] : Si = soluble inert organic material (g COD/m3)
c asmm[2] : Ss = readily biodegradable substrate (g COD/m3)
c asmm[3] : Xi = particulate inert organic material (g COD/m3)
c asmm[4] : Xs = slowly biodegradable substrate (g COD/m3)
c asmm[5] : Xbh = active heterotrophic biomass (g COD/m3)
c asmm[6] : Xba = active autotrophic biomass (g COD/m3)
c asmm[7] : Xp = particulate product arising from biomass decay (g COD/m3)
c asmm[8] : So = oxygen (g -COD/m3)
c asmm[9] : Sno = nitrate and nitrite nitrogen (g N/m3)
c asmm[10] : Snh = ammonia and ammonium nitrogen (g N/m3)
c asmm[11] : Snd = soluble biogradable organic nitrogen (g N/m3)
c asmm[12] : Xnd = particulate biogradable organic nitrogen (g N/m3)
c asmm[13] : Salk = alkalinity (mole HCO3-/m3)

DOUBLE PRECISION asmm(13)

DOUBLE PRECISION xtemp(24)

DOUBLE PRECISION biomass,biomass_nobio,biomass_bioN
DOUBLE PRECISION remainCOD,xtemp,xstemp
DOUBLE PRECISION totCODout,totTKNout,totCODin,totTKNin,ph
DOUBLE PRECISION inertX,xtemp2,noninertX
DOUBLE PRECISION inertS

DOUBLE PRECISION alfachac, alfachpro, alfachbu, alfachva
DOUBLE PRECISION alfachin, alfachic, alfachnh
DOUBLE PRECISION alfachalk, alfachno
DOUBLE PRECISION chargeasm1, chargeadm1, pkk, tfac, bigr

tfac=1.d00/298.15d00-1.d00/(273.15d00+35.d00)
bigr=0.08314d-00

c ADM1 --> ASM1

c Copyright: John Copp, Hydromantis Inc. and Ulf Jeppsson, Lund University

do i=1,13
    asmm(i)=0.d0
endo
c
Set parameter values

fnaa = 0.098d0
fnxc = 0.0376d0
fnbac = 0.08d0
fsni_adm = 0.06d0
fxni=0.06d00
fsni=0.d00
nbac=fnbac/14.d0*14000.d0
nxc = fnxc/14.d0*14000
naa = fnaa/14.d0*14000.d0
sni_adm = fsni_adm/14.d0*14000.d0
xni=fxni/14.d0*14000.d0
sni=fsni/14.d0*14000.d0
frxs_AS=0.79d0
fdegrade_AS=0.d00
warn=0

do i=1,24
    xtemp(i)=adm(i)
endo
c Biomass becomes part of XS and XP when transformed into ASM
c Assume N part of formed XS to be fnxc and N part of XP to be fxni
c Remaining N goes into ammonia pool
    biomass=0.d0
    do i=17,23
        biomass = biomass + xtemp(i)*1000.d0
    enddo
c Part of biomass mapped into XP
    biomass_nobio=biomass*(1.d0-frxs_AS)
    biomass_bioN=biomass*fnbac-biomass_nobio*fxni
    remainCOD=0.d0
    if(biomass_bioN.LT.0.d0) then
        warn=1
        xptemp=biomass*fnbac/fxni
        biomass_bioN=0.d0
    else
        xptemp=biomass_nobio
    end if

    if((biomass_bioN/fnxc).LE.(biomass-biomass_nobio)) then
        c all biomass N used
        xtemp=biomass_bioN/fnxc
        remainCOD=biomass-biomass_nobio-xtemp
    c use part of remaining S_IN to form XS
    if((xtemp(11)*14000.d0/fnxc).GT.remainCOD) then
        xtemp=xtemp+remainCOD
    else
        write(*,*) 'System failure'
        warn=2
    end if
    else
        c all biomass COD used
        xtemp=biomass-biomass_nobio
    end if

c Any remaining N in S_IN
    xtemp(11)=xtemp(11)+biomass*fnbac/14000.d0 -xptemp*fxni/14000.d0
    &-xstemp*fnxc/14000.d0

c Xs = all X from ADM except Xi + biomass
    asmm(4)=xtemp
    do i=13,16
        asmm(4)=asmm(4)+xtemp(i)*1000.
    end do

c inert part of biomass
    asmm(7)=xtemp

c mapping of inert XI in AD into XI and possibly XS in AS
c Assumption: same N content in both ASM1 and ADM1 particulate inerts
c Special case: if part of XI in AD can be degraded in AS
c We have no knowledge about the contents so we put it in as part substrate
(XS)
c We need to keep track of the associated nitrogen

c N content might be different, take first from XI-N then S_IN
inertX=(1.d0-fdegrade_AS)*xtemp(24)*1000.d0
xstemp2=0.d0
noninertX=0.d0

if(fdegrade_AS.GT.0.d0) then
    noninertX=fdegrade_AS*xtemp(24)*1000.d0
end if

if(fxni.LT.fnxc) then
    xstemp2=noninertX*fxni/fnxc
    noninertX=noninertX-noninertX*fxni/fnxc
end if

if((xtemp(11)*14000.d0).LT.(noninertX*fnxc)) then
    xstemp2=xstemp2+xtemp(11)*14000.d0/fnxc
    noninertX=noninertX-xtemp(11)*14000.d0/fnxc
    xtemp(11)=0.d0
    warn=3
    inertX=inertX+noninertX
else
    xstemp2=xstemp2+noninertX
    c N in SNH not enough
end if

if((xtemp(11)*14.d0).LT.(xtemp(12)*fsni)) then
    inertS=xtemp(12)*fsni/fsni
    xtemp(12)=xtemp(12)-xtemp(11)*14.d0/fsni
    xtemp(11)=0.d0
    warn=5
else
    xtemp(11)=xtemp(11)-xtemp(12)*fsni/14.d0
xtemp(12)=0.d0
    c N in SI(AD) enough for mapping
    end if
end if

if(fsni_adm.LT.fsni) then
    inertS=xtemp(12)*fsni_adm/fsni
    xtemp(12)=xtemp(12)-xtemp(11)*14.d0/fsni
    xtemp(11)=0.d0
    warn=5
else
    xtemp(11)=xtemp(11)-xtemp(12)*fsni/14.d0
xtemp(12)=0.d0
    c N in SI(AD) enough for mapping
    end if
end if

if((xtemp(11)*14.d0).LT.(xtemp(12)*fsni)) then
    inertS=xtemp(12)*fsni/fsni
    xtemp(12)=xtemp(12)-xtemp(11)*14.d0/fsni
    xtemp(11)=0.d0
    warn=5
else
    xtemp(11)=xtemp(11)-xtemp(12)*fsni/14.d0
xtemp(12)=0.d0
    c N in SI(AD) enough for mapping
    end if
end if

if((xtemp(11)*14.d0).LT.(xtemp(12)*fsni)) then
    inertS=xtemp(12)*fsni/fsni
    xtemp(12)=xtemp(12)-xtemp(11)*14.d0/fsni
    xtemp(11)=0.d0
    warn=5
else
    xtemp(11)=xtemp(11)-xtemp(12)*fsni/14.d0
xtemp(12)=0.d0
    c N in SI(AD) enough for mapping
    end if
end if

inertS=0.d0

if(fxni.LT.fnxc) then
    xstemp2=noninertX*fxni/fnxc
    noninertX=noninertX-xinertX
end if

if((xtemp(11)*14000.d0).LT.(noninertX*fnxc)) then
    xstemp2=xstemp2+xtemp(11)*14000.d0/fnxc
    noninertX=noninertX-xtemp(11)*14000.d0/fnxc
else
    xstemp2=xstemp2+noninertX
    c N in SI(AD) enough for mapping
end if

if((xtemp(11)*14.d0).LT.(xtemp(12)*fsni)) then
    inertS=xtemp(12)*fsni/fsni
    xtemp(12)=xtemp(12)-xtemp(11)*14.d0/fsni
else
    xtemp(11)=xtemp(11)-xtemp(12)*fsni/14.d0
xtemp(12)=0.d0
    c N in SI(AD) enough for mapping
end if

if(fsni_adm.LT.fsni) then
    inertS=xtemp(12)*fsni_adm/fsni
    xtemp(12)=xtemp(12)-xtemp(11)*14.d0/fsni
else
    xtemp(11)=xtemp(11)-xtemp(12)*fsni/14.d0
xtemp(12)=0.d0
    c N in SI(AD) enough for mapping
end if

if((xtemp(11)*14.d0).LT.(xtemp(12)*fsni)) then
    inertS=xtemp(12)*fsni/fsni
    xtemp(12)=xtemp(12)-xtemp(11)*14.d0/fsni
else
    xtemp(11)=xtemp(11)-xtemp(12)*fsni/14.d0
xtemp(12)=0.d0
    c N in SI(AD) enough for mapping
end if

if(fxni.LT.fnxc) then
    xstemp2=noninertX*fxni/fnxc
    noninertX=noninertX-xinertX
end if

xstemp2=xstemp2+noninertX

xtemp(11)=xtemp(11)-xtemp(12)*fsni/14.d0
xtemp(12)=0.d0

end if
C Si as SI
   asmm(1)=inertS*1000.

C nitrogen in biomass, composites, proteins
C Xnd is the nitrogen part of XS in ASM1. Should be based on
C the same variables as constitutes XS, xc and xpo (no nitrogen in lipids
and carbohydrates)
   asmm(12)=fnxc*(xstemp+xstemp2)+nxc*xtemp(13)+nna*xtemp(15)

C sh2=x(8) and sch4=x(9) assumed to be stripped upon reentry to ASM side
C Ss = sum of all solubles except Sh2, Sch4, Si, Sic, Sin
   do i=1,7
      asmm(2) = asmm(2)+xtemp(i)*1000.d0
   enddo

C Snd is the nitrogen part of Ss in ASM1. Therefore Snd should be based on
C same variables as constitutes Ss, and we assume
C there is only nitrogen in the amino acids. The N content of Si is
C not included in ASM1
   asmm(11) = nna*xtemp(2)

C Snh = Sin including adjustments above
   asmm(10) = xtemp(11)*14000.d0

C Calculation of Salk
   alfachac=(-1.d0/64.d0)/(1.d0+10.d0**((4.76d0-ph)))
   alfachpro=(-1.d0/112.d0)/(1.d0+10.d0**((4.88d0-ph)))
   alfachbu=(-1.d0/160.d0)/(1.d0+10.d0**((4.82d0-ph)))
   alfachva=(-1.d0/208.d0)/(1.d0+10.d0**((4.86d0-ph)))
   pkk=-dlog10((10.d0**(-9.25d0)*dexp(51965.d0/bigr/100.d0*tfac))
   & dexp(51965.d0/bigr/100.d0*tfac))
   alfachin=(10.d0**((pkk-ph))/(1.d0+10.d0**((pkk-ph)))
   pkk=-dlog10((10.d0**(-6.35d0))
   & dexp(7646.d0/bigr/100.d0*tfac))
   alfachic=-1.d0/(1.d0+10.d0**((pkk-ph)))

C modifie / PV
   alfachnh=1.d0/14.d0
   alfachno=-1.d0/14.d0
   alfachalk=-1.d0

   chargeasm1=asmm(10)*alfachnh+asmm(9)*alfachno
   chargeadm1=(adm(7)*alfachac+adm(6)*alfachpro+adm(11)*alfachin
   & +adm(5)*alfachbu+adm(4)*alfachva+adm(10)*alfachic)*1000.
   asmm(13)=chargeasm1-chargeadm1

C check mass balances
   totCODin=0.d0
   do i=1,7
      totCODin=totCODin + adm(i)
   enddo
   do i=12,24
      totCODin = totCODin + adm(i)*1000.d0
   enddo
   totTKNin=0.d0
   do i=17,23
      totTKNin=totTKNin+nbac*adm(i)
   enddo
   totTKNin = totTKNin + nxc*adm(13) + nna*adm(15) + nna*adm(2) +
   & adm(11)*14000.d0 + sni_adm*adm(12) + xni*adm(24)
totCODout=0.d0
do i=1,7
    totCODout = totCODout + asmm(i)
enddo

cc!!Note SI N not included here
totTKNout = asmm(10) + asmm(11) + asmm(12) + fSni*asmm(1) + &fNbac*(asmm(5) + asmm(6)) + fXni*(asmm(3) + asmm(7))
return
end
APPENDIX 3: Steady-state results

These results were obtained with FORTRAN (integration using a Runge-Kutta 4 algorithm with a constant integration step = 0.005 h) and MATLAB-Simulink (Solver: ode45, absolute tolerance = 10^{-8}, relative tolerance = 10^{-5}). For details on operation conditions see text.

### CONCENTRATION values

<table>
<thead>
<tr>
<th>Input variables raw wastewater</th>
<th>Values</th>
<th>Output variables</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_I$ (g COD.m^{-3})</td>
<td>27.22619062</td>
<td>$S_I$ (g COD.m^{-3})</td>
<td>28.064300</td>
<td>28.0643</td>
</tr>
<tr>
<td>$S_{II}$ (g COD.m^{-3})</td>
<td>58.17618568</td>
<td>$S_{II}$ (g COD.m^{-3})</td>
<td>0.671945</td>
<td>0.67336</td>
</tr>
<tr>
<td>$X_I$ (g COD.m^{-3})</td>
<td>92.49900106</td>
<td>$X_I$ (g COD.m^{-3})</td>
<td>5.918820</td>
<td>5.9191</td>
</tr>
<tr>
<td>$X_{II}$ (g COD.m^{-3})</td>
<td>363.943473</td>
<td>$X_{II}$ (g COD.m^{-3})</td>
<td>0.123290</td>
<td>0.12329</td>
</tr>
<tr>
<td>$X_{TH}$ (g COD.m^{-3})</td>
<td>50.68328815</td>
<td>$X_{TH}$ (g COD.m^{-3})</td>
<td>8.662210</td>
<td>8.6614</td>
</tr>
<tr>
<td>$X_{BA}$ (g COD.m^{-3})</td>
<td>0</td>
<td>$X_{BA}$ (g COD.m^{-3})</td>
<td>0.649823</td>
<td>0.6484</td>
</tr>
<tr>
<td>$X_{V}$ (g COD.m^{-3})</td>
<td>0</td>
<td>$X_{V}$ (g COD.m^{-3})</td>
<td>3.748460</td>
<td>3.7485</td>
</tr>
<tr>
<td>$S_{S}$ (g COD.m^{-3})</td>
<td>58.17618568</td>
<td>$S_{S}$ (g COD.m^{-3})</td>
<td>1.373160</td>
<td>1.3748</td>
</tr>
<tr>
<td>$X_{SI}$ (g COD.m^{-3})</td>
<td>92.49900106</td>
<td>$X_{SI}$ (g COD.m^{-3})</td>
<td>1.373160</td>
<td>1.3748</td>
</tr>
<tr>
<td>$S_{SH}$ (g COD.m^{-3})</td>
<td>23.85946563</td>
<td>$S_{SH}$ (g COD.m^{-3})</td>
<td>0.158835</td>
<td>0.15845</td>
</tr>
<tr>
<td>$S_{ND}$ (g COD.m^{-3})</td>
<td>5.651606031</td>
<td>$S_{ND}$ (g COD.m^{-3})</td>
<td>0.560279</td>
<td>0.55943</td>
</tr>
<tr>
<td>$X_{SH}$ (g COD.m^{-3})</td>
<td>16.12981606</td>
<td>$X_{SH}$ (g COD.m^{-3})</td>
<td>0.009243</td>
<td>0.0092428</td>
</tr>
<tr>
<td>$S_{ALK}$ (mole HCO3.m^{-3})</td>
<td>7</td>
<td>$S_{ALK}$ (mole HCO3.m^{-3})</td>
<td>4.564060</td>
<td>4.5646</td>
</tr>
<tr>
<td>$TSS$ (g COD.m^{-3})</td>
<td>380.3443217</td>
<td>$TSS$ (g COD.m^{-3})</td>
<td>14.326277</td>
<td>14.3255</td>
</tr>
<tr>
<td>$Q$ (m^3.d^{-1})</td>
<td>20648.36121</td>
<td>$Q$ (m^3.d^{-1})</td>
<td>20640.768000</td>
<td>20640.7792</td>
</tr>
<tr>
<td>$T$ (°C)</td>
<td>14.85808006</td>
<td>$T$ (°C)</td>
<td>14.8581</td>
<td>14.8581</td>
</tr>
</tbody>
</table>

### LOAD values

<table>
<thead>
<tr>
<th>Input variables raw wastewater</th>
<th>Values</th>
<th>Output variables</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_I$ (kg COD.d^{-1})</td>
<td>562.1762184</td>
<td>$S_I$ (kg COD.d^{-1})</td>
<td>579.2687054</td>
<td>579.2690197</td>
</tr>
<tr>
<td>$S_{II}$ (kg COD.d^{-1})</td>
<td>1201.242896</td>
<td>$S_{II}$ (kg COD.d^{-1})</td>
<td>13.8964085</td>
<td>13.89867508</td>
</tr>
<tr>
<td>$X_I$ (kg COD.d^{-1})</td>
<td>1909.952785</td>
<td>$X_I$ (kg COD.d^{-1})</td>
<td>122.1699095</td>
<td>122.1748362</td>
</tr>
<tr>
<td>$X_{II}$ (kg COD.d^4)</td>
<td>7514.836291</td>
<td>$X_{II}$ (kg COD.d^4)</td>
<td>2.544800287</td>
<td>2.544801668</td>
</tr>
<tr>
<td>$X_{TH}$ (kg COD.d^4)</td>
<td>1046.526841</td>
<td>$X_{TH}$ (kg COD.d^4)</td>
<td>178.794667</td>
<td>178.778045</td>
</tr>
<tr>
<td>$X_{BA}$ (kg COD.d^4)</td>
<td>0</td>
<td>$X_{BA}$ (kg COD.d^4)</td>
<td>13.39426909</td>
<td>13.38348123</td>
</tr>
<tr>
<td>$X_{V}$ (kg COD.d^4)</td>
<td>0</td>
<td>$X_{V}$ (kg COD.d^4)</td>
<td>77.37196083</td>
<td>77.37196083</td>
</tr>
<tr>
<td>$S_{S}$ (kg COD.d^4)</td>
<td>0</td>
<td>$S_{S}$ (kg COD.d^4)</td>
<td>28.34307699</td>
<td>28.37694324</td>
</tr>
<tr>
<td>$S_{SH}$ (kg COD.d^4)</td>
<td>0</td>
<td>$S_{SH}$ (kg COD.d^4)</td>
<td>189.7794773</td>
<td>189.7878366</td>
</tr>
<tr>
<td>$S_{ND}$ (kg COD.d^4)</td>
<td>492.6588647</td>
<td>$S_{ND}$ (kg COD.d^4)</td>
<td>3.278476385</td>
<td>3.270531464</td>
</tr>
<tr>
<td>$X_{SH}$ (kg COD.d^4)</td>
<td>116.6964028</td>
<td>$X_{SH}$ (kg COD.d^4)</td>
<td>11.56458885</td>
<td>11.54707111</td>
</tr>
<tr>
<td>$X_{ND}$ (kg COD.d^4)</td>
<td>333.0542683</td>
<td>$X_{ND}$ (kg COD.d^4)</td>
<td>0.190784063</td>
<td>0.190778594</td>
</tr>
<tr>
<td>$S_{ALK}$ (mole HCO3.d^{-1})</td>
<td>144.5385285</td>
<td>$S_{ALK}$ (mole HCO3.d^{-1})</td>
<td>94.2057036</td>
<td>94.21690074</td>
</tr>
<tr>
<td>$TSS$ (kg COD.d^{-1})</td>
<td>7853.486938</td>
<td>$TSS$ (kg COD.d^{-1})</td>
<td>295.705365</td>
<td>295.6894824</td>
</tr>
</tbody>
</table>
### CONCENTRATION values

<table>
<thead>
<tr>
<th>Input variables raw wastewater</th>
<th>Values</th>
<th>Output variables</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_I$ (g COD.m$^{-3}$)</td>
<td>27.22619062</td>
<td>$S_I$ (g COD.m$^{-3}$)</td>
<td>130.863000</td>
<td>130.8682</td>
</tr>
<tr>
<td>$S_S$ (g COD.m$^{-3}$)</td>
<td>58.17618568</td>
<td>$S_S$ (g COD.m$^{-3}$)</td>
<td>262.103000</td>
<td>258.5788</td>
</tr>
<tr>
<td>$X_I$ (g COD.m$^{-3}$)</td>
<td>92.49900106</td>
<td>$X_I$ (g COD.m$^{-3}$)</td>
<td>314241.0000</td>
<td>314238.8343</td>
</tr>
<tr>
<td>$X_S$ (g COD.m$^{-3}$)</td>
<td>363.943473</td>
<td>$X_S$ (g COD.m$^{-3}$)</td>
<td>47664.900000</td>
<td>47667.026</td>
</tr>
<tr>
<td>$X_{BA}$ (g COD.m$^{-3}$)</td>
<td>50.68328815</td>
<td>$X_{BA}$ (g COD.m$^{-3}$)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$X_P$ (g COD.m$^{-3}$)</td>
<td>0</td>
<td>$X_P$ (g COD.m$^{-3}$)</td>
<td>11426.900000</td>
<td>11427.473</td>
</tr>
<tr>
<td>$S_O$ (g -COD.m$^{-3}$)</td>
<td>0</td>
<td>$S_O$ (g -COD.m$^{-3}$)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_NO$ (g N.m$^{-3}$)</td>
<td>23.85946563</td>
<td>$S_NO$ (g N.m$^{-3}$)</td>
<td>1443.440000</td>
<td>1442.7931</td>
</tr>
<tr>
<td>$S_{NH}$ (g N.m$^{-3}$)</td>
<td>5.651606031</td>
<td>$S_{NH}$ (g N.m$^{-3}$)</td>
<td>0.543236</td>
<td>0.54232</td>
</tr>
<tr>
<td>$X_{ND}$ (g N.m$^{-3}$)</td>
<td>16.12981606</td>
<td>$X_{ND}$ (g N.m$^{-3}$)</td>
<td>1841.030000</td>
<td>1841.1071</td>
</tr>
<tr>
<td>$S_{ALK}$ (mole HCO$_3$.m$^{-3}$)</td>
<td>7</td>
<td>$S_{ALK}$ (mole HCO$_3$.m$^{-3}$)</td>
<td>97.823400</td>
<td>97.8462</td>
</tr>
<tr>
<td>$TSS$ (g.m$^{-3}$)</td>
<td>380.3443217</td>
<td>$TSS$ (g.m$^{-3}$)</td>
<td>279999.600000</td>
<td>280000</td>
</tr>
<tr>
<td>$Q$ (m$^3$.d$^{-1}$)</td>
<td>14.85808006</td>
<td>$Q$ (m$^3$.d$^{-1}$)</td>
<td>9.582480</td>
<td>9.582</td>
</tr>
<tr>
<td>$T$ (°C)</td>
<td></td>
<td>$T$ (°C)</td>
<td>14.858100</td>
<td>14.8581</td>
</tr>
</tbody>
</table>

### LOAD values

<table>
<thead>
<tr>
<th>Input variables raw wastewater</th>
<th>Values</th>
<th>Output variables</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_I$ (kg COD.d$^{-1}$)</td>
<td>562.1762184</td>
<td>$S_I$ (kg COD.d$^{-1}$)</td>
<td>1.25399208</td>
<td>1.253979092</td>
</tr>
<tr>
<td>$S_S$ (kg COD.d$^{-1}$)</td>
<td>1201.242896</td>
<td>$S_S$ (kg COD.d$^{-1}$)</td>
<td>2.511596755</td>
<td>2.477702062</td>
</tr>
<tr>
<td>$X_I$ (kg COD.d$^{-1}$)</td>
<td>1909.952785</td>
<td>$X_I$ (kg COD.d$^{-1}$)</td>
<td>3011.208098</td>
<td>3011.03651</td>
</tr>
<tr>
<td>$X_S$ (kg COD.d$^{-1}$)</td>
<td>7514.836291</td>
<td>$X_S$ (kg COD.d$^{-1}$)</td>
<td>456.747951</td>
<td>456.745431</td>
</tr>
<tr>
<td>$X_{BA}$ (kg COD.d$^{-1}$)</td>
<td>1046.526841</td>
<td>$X_{BA}$ (kg COD.d$^{-1}$)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$X_P$ (kg COD.d$^{-1}$)</td>
<td>0</td>
<td>$X_P$ (kg COD.d$^{-1}$)</td>
<td>109.4980407</td>
<td>109.4980463</td>
</tr>
<tr>
<td>$S_O$ (kg -COD.d$^{-1}$)</td>
<td>0</td>
<td>$S_O$ (kg -COD.d$^{-1}$)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_NO$ (kg N.d$^{-1}$)</td>
<td>492.6588647</td>
<td>$S_NO$ (kg N.d$^{-1}$)</td>
<td>13.83173493</td>
<td>13.82483438</td>
</tr>
<tr>
<td>$S_{NH}$ (kg N.d$^{-1}$)</td>
<td>116.96964028</td>
<td>$S_{NH}$ (kg N.d$^{-1}$)</td>
<td>0.005205548</td>
<td>0.00519651</td>
</tr>
<tr>
<td>$S_{ND}$ (kg N.d$^{-1}$)</td>
<td>333.0542683</td>
<td>$S_{ND}$ (kg N.d$^{-1}$)</td>
<td>17.64163315</td>
<td>17.64148823</td>
</tr>
<tr>
<td>$S_{ALK}$ (kmole HCO$_3$.d$^{-1}$)</td>
<td>144.5385285</td>
<td>$S_{ALK}$ (kmole HCO$_3$.d$^{-1}$)</td>
<td>0.937390774</td>
<td>0.937562288</td>
</tr>
<tr>
<td>$TSS$ (kg.d$^{-1}$)</td>
<td>7853.486938</td>
<td>$TSS$ (kg.d$^{-1}$)</td>
<td>2683.090567</td>
<td>2682.96</td>
</tr>
<tr>
<td>CONCENTRATION values</td>
<td>PRIMARY EFFLUENT</td>
<td>FORTRAN</td>
<td>MATLAB</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------</td>
<td>---------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>Output variables</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_1$ (g COD.m$^{-3}$)</td>
<td>28.067000</td>
<td>28.067</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_6$ (g COD.m$^{-3}$)</td>
<td>59.075600</td>
<td>59.0473</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_I$ (g COD.m$^{-3}$)</td>
<td>49.336300</td>
<td>49.3362</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_S$ (g COD.m$^{-3}$)</td>
<td>186.585000</td>
<td>186.5845</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{BH}$ (g COD.m$^{-3}$)</td>
<td>26.611600</td>
<td>26.6115</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{BA}$ (g COD.m$^{-3}$)</td>
<td>0.049527</td>
<td>0.0495</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_P$ (g COD.m$^{-3}$)</td>
<td>0.341502</td>
<td>0.3415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_O$ (g -COD.m$^{-3}$)</td>
<td>0.017526</td>
<td>0.0175</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_NO$ (g N.m$^{-3}$)</td>
<td>0.117352</td>
<td>0.1174</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_NH$ (g N.m$^{-3}$)</td>
<td>34.926900</td>
<td>34.9215</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{ND}$ (g N.m$^{-3}$)</td>
<td>5.545710</td>
<td>5.5457</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{ND}$ (g N.m$^{-3}$)</td>
<td>8.268320</td>
<td>8.2683</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{ALK}$ (mole HCO$_3^-$m$^{-3}$)</td>
<td>7.696350</td>
<td>7.6965</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$TSS$ (g.m$^{-3}$)</td>
<td>197.192947</td>
<td>197.1925</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Q$ (m$^3$.d$^{-1}$)</td>
<td>20938.776000</td>
<td>20939</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T$ (°C)</td>
<td>14.858100</td>
<td>14.8581</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CONCENTRATION values</th>
<th>PRIMARY UNDERFLOW</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output variables</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_1$ (g COD.m$^{-3}$)</td>
<td>28.067000</td>
<td>28.0672</td>
<td></td>
</tr>
<tr>
<td>$S_6$ (g COD.m$^{-3}$)</td>
<td>59.075600</td>
<td>59.0473</td>
<td></td>
</tr>
<tr>
<td>$X_I$ (g COD.m$^{-3}$)</td>
<td>6480.700000</td>
<td>6480.7</td>
<td></td>
</tr>
<tr>
<td>$X_S$ (g COD.m$^{-3}$)</td>
<td>24509.300000</td>
<td>24509</td>
<td></td>
</tr>
<tr>
<td>$X_{BH}$ (g COD.m$^{-3}$)</td>
<td>3495.640000</td>
<td>3495.6</td>
<td></td>
</tr>
<tr>
<td>$X_{BA}$ (g COD.m$^{-3}$)</td>
<td>6.505740</td>
<td>6.5001</td>
<td></td>
</tr>
<tr>
<td>$X_P$ (g COD.m$^{-3}$)</td>
<td>44.858800</td>
<td>44.8571</td>
<td></td>
</tr>
<tr>
<td>$S_O$ (g -COD.m$^{-3}$)</td>
<td>0.017526</td>
<td>0.0175</td>
<td></td>
</tr>
<tr>
<td>$S_NO$ (g N.m$^{-3}$)</td>
<td>0.117352</td>
<td>0.1174</td>
<td></td>
</tr>
<tr>
<td>$S_NH$ (g N.m$^{-3}$)</td>
<td>34.926900</td>
<td>34.9215</td>
<td></td>
</tr>
<tr>
<td>$S_{ND}$ (g N.m$^{-3}$)</td>
<td>5.545710</td>
<td>5.5457</td>
<td></td>
</tr>
<tr>
<td>$X_{ND}$ (g N.m$^{-3}$)</td>
<td>1086.110000</td>
<td>1086.1</td>
<td></td>
</tr>
<tr>
<td>$S_{ALK}$ (mole HCO$_3^-$m$^{-3}$)</td>
<td>7.696350</td>
<td>7.6965</td>
<td></td>
</tr>
<tr>
<td>$TSS$ (g.m$^{-3}$)</td>
<td>25902.753405</td>
<td>25903</td>
<td></td>
</tr>
<tr>
<td>$Q$ (m$^3$.d$^{-1}$)</td>
<td>147.604800</td>
<td>147.6047</td>
<td></td>
</tr>
<tr>
<td>$T$ (°C)</td>
<td>14.858100</td>
<td>14.8581</td>
<td></td>
</tr>
</tbody>
</table>
## Concentration Values

### Anoxygenic Reactor NO 2

<table>
<thead>
<tr>
<th>Output variables</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_I$ (g COD.m$^{-3}$)</td>
<td>28.064300</td>
<td>28.0643</td>
</tr>
<tr>
<td>$S_S$ (g COD.m$^{-3}$)</td>
<td>1.337900</td>
<td>1.3412</td>
</tr>
<tr>
<td>$X_I$ (g COD.m$^{-3}$)</td>
<td>1532.270000</td>
<td>1532.3</td>
</tr>
<tr>
<td>$X_S$ (g COD.m$^{-3}$)</td>
<td>58.863100</td>
<td>58.8579</td>
</tr>
<tr>
<td>$X_{BH}$ (g COD.m$^{-3}$)</td>
<td>2245.750000</td>
<td>2245.4</td>
</tr>
<tr>
<td>$X_{BA}$ (g COD.m$^{-3}$)</td>
<td>166.697000</td>
<td>166.5512</td>
</tr>
<tr>
<td>$X_P$ (g COD.m$^{-3}$)</td>
<td>965.720000</td>
<td>965.6773</td>
</tr>
<tr>
<td>$S_O$ (g -COD.m$^{-3}$)</td>
<td>0.000109</td>
<td>0.00010907</td>
</tr>
<tr>
<td>$S_{NO}$ (g N.m$^{-3}$)</td>
<td>2.219260</td>
<td>2.2207</td>
</tr>
<tr>
<td>$S_{NH}$ (g N.m$^{-3}$)</td>
<td>7.203620</td>
<td>7.2028</td>
</tr>
<tr>
<td>$S_{ND}$ (g N.m$^{-3}$)</td>
<td>0.687265</td>
<td>0.6862</td>
</tr>
<tr>
<td>$X_{NO}$ (g N.m$^{-3}$)</td>
<td>3.742710</td>
<td>3.7424</td>
</tr>
<tr>
<td>$X_{ND}$ (g N.m$^{-3}$)</td>
<td>5.565480</td>
<td>5.5659</td>
</tr>
<tr>
<td>S$_{ALK}$ (mole HCO$_3$.m$^{-3}$)</td>
<td>5.5659</td>
<td>5.5659</td>
</tr>
<tr>
<td>TSS (g.m$^{-3}$)</td>
<td>3726.975075</td>
<td>3726.5</td>
</tr>
<tr>
<td>$Q$ (m$^3$.d$^{-1}$)</td>
<td>103532.78</td>
<td>103532.78</td>
</tr>
<tr>
<td>$T$ (°C)</td>
<td>14.856600</td>
<td>14.8581</td>
</tr>
</tbody>
</table>

### Aerobic Reactor NO 2

<table>
<thead>
<tr>
<th>Output variables</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_I$ (g COD.m$^{-3}$)</td>
<td>28.064300</td>
<td>28.0643</td>
</tr>
<tr>
<td>$S_S$ (g COD.m$^{-3}$)</td>
<td>0.778875</td>
<td>0.7806</td>
</tr>
<tr>
<td>$X_I$ (g COD.m$^{-3}$)</td>
<td>1532.270000</td>
<td>1532.3</td>
</tr>
<tr>
<td>$X_S$ (g COD.m$^{-3}$)</td>
<td>37.391100</td>
<td>37.388</td>
</tr>
<tr>
<td>$X_{BH}$ (g COD.m$^{-3}$)</td>
<td>2245.990000</td>
<td>2245.6</td>
</tr>
<tr>
<td>$X_{BA}$ (g COD.m$^{-3}$)</td>
<td>167.980000</td>
<td>167.8339</td>
</tr>
<tr>
<td>$X_P$ (g COD.m$^{-3}$)</td>
<td>968.847000</td>
<td>968.804</td>
</tr>
<tr>
<td>$S_O$ (g -COD.m$^{-3}$)</td>
<td>1.427790</td>
<td>1.4284</td>
</tr>
<tr>
<td>$S_{NO}$ (g N.m$^{-3}$)</td>
<td>8.405400</td>
<td>8.4066</td>
</tr>
<tr>
<td>$S_{NH}$ (g N.m$^{-3}$)</td>
<td>0.693511</td>
<td>0.6922</td>
</tr>
<tr>
<td>$S_{ND}$ (g N.m$^{-3}$)</td>
<td>0.610313</td>
<td>0.6094</td>
</tr>
<tr>
<td>$X_{NO}$ (g N.m$^{-3}$)</td>
<td>2.681730</td>
<td>2.6815</td>
</tr>
<tr>
<td>$X_{ND}$ (g N.m$^{-3}$)</td>
<td>4.658610</td>
<td>4.659</td>
</tr>
<tr>
<td>S$_{ALK}$ (mole HCO$_3$.m$^{-3}$)</td>
<td>4.658610</td>
<td>4.659</td>
</tr>
<tr>
<td>TSS (g.m$^{-3}$)</td>
<td>3714.358575</td>
<td>3713.9</td>
</tr>
<tr>
<td>$Q$ (m$^3$.d$^{-1}$)</td>
<td>103532.78</td>
<td>103532.78</td>
</tr>
<tr>
<td>$T$ (°C)</td>
<td>14.856600</td>
<td>14.8581</td>
</tr>
</tbody>
</table>
### Benchmark Simulation Model no. 2 (BSM2)

#### Concentration Values

<table>
<thead>
<tr>
<th>Output variables</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_t$ (g COD.m$^{-3}$)</td>
<td>28.064300</td>
<td>28.0643</td>
</tr>
<tr>
<td>$X_t$ (g COD.m$^{-3}$)</td>
<td>3036.240000</td>
<td>3036.2</td>
</tr>
<tr>
<td>$X_S$ (g COD.m$^{-3}$)</td>
<td>63.245400</td>
<td>63.2392</td>
</tr>
<tr>
<td>$X_{BH}$ (g COD.m$^{-3}$)</td>
<td>4443.550000</td>
<td>4442.8</td>
</tr>
<tr>
<td>$X_{BA}$ (g COD.m$^{-3}$)</td>
<td>332.885000</td>
<td>332.5957</td>
</tr>
<tr>
<td>$X_P$ (g COD.m$^{-3}$)</td>
<td>1922.890000</td>
<td>1922.8</td>
</tr>
<tr>
<td>$S_O$ (g -COD.m$^{-3}$)</td>
<td>1.373160</td>
<td>1.3748</td>
</tr>
<tr>
<td>$S_NO$ (g N.m$^{-3}$)</td>
<td>9.194400</td>
<td>9.1948</td>
</tr>
<tr>
<td>$S_NH$ (g N.m$^{-3}$)</td>
<td>0.158835</td>
<td>0.1585</td>
</tr>
<tr>
<td>$S_{ND}$ (g N.m$^{-3}$)</td>
<td>4.741270</td>
<td>4.7411</td>
</tr>
<tr>
<td>$X_{ND}$ (g N.m$^{-3}$)</td>
<td>4.741520</td>
<td>4.7411</td>
</tr>
<tr>
<td>$S_{ALK}$ (mole HCO$_3^-$·m$^{-3}$)</td>
<td>4.564060</td>
<td>4.5646</td>
</tr>
<tr>
<td>$TSS$ (g.m$^{-3}$)</td>
<td>7349.107800</td>
<td>7348.3</td>
</tr>
<tr>
<td>$Q$ (m$^3$.d$^{-1}$)</td>
<td>300.000000</td>
<td>300</td>
</tr>
<tr>
<td>$T$ (°C)</td>
<td>14.856600</td>
<td>14.8581</td>
</tr>
</tbody>
</table>

#### Concentration Values

<table>
<thead>
<tr>
<th>Output variables</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_t$ (g COD.m$^{-3}$)</td>
<td>28.064300</td>
<td>28.0643</td>
</tr>
<tr>
<td>$X_t$ (g COD.m$^{-3}$)</td>
<td>67.689300</td>
<td>67.6876</td>
</tr>
<tr>
<td>$X_S$ (g COD.m$^{-3}$)</td>
<td>1.409980</td>
<td>1.4098</td>
</tr>
<tr>
<td>$X_{BH}$ (g COD.m$^{-3}$)</td>
<td>99.063400</td>
<td>99.0462</td>
</tr>
<tr>
<td>$X_{BA}$ (g COD.m$^{-3}$)</td>
<td>7.421270</td>
<td>7.4147</td>
</tr>
<tr>
<td>$X_P$ (g COD.m$^{-3}$)</td>
<td>42.868400</td>
<td>42.8659</td>
</tr>
<tr>
<td>$S_O$ (g -COD.m$^{-3}$)</td>
<td>1.373160</td>
<td>1.3748</td>
</tr>
<tr>
<td>$S_NO$ (g N.m$^{-3}$)</td>
<td>9.194400</td>
<td>9.1948</td>
</tr>
<tr>
<td>$S_NH$ (g N.m$^{-3}$)</td>
<td>0.158835</td>
<td>0.1585</td>
</tr>
<tr>
<td>$S_{ND}$ (g N.m$^{-3}$)</td>
<td>0.105706</td>
<td>0.1057</td>
</tr>
<tr>
<td>$X_{ND}$ (g N.m$^{-3}$)</td>
<td>0.1058706</td>
<td>0.1057</td>
</tr>
<tr>
<td>$S_{ALK}$ (mole HCO$_3^-$·m$^{-3}$)</td>
<td>4.564060</td>
<td>4.5646</td>
</tr>
<tr>
<td>$TSS$ (g.m$^{-3}$)</td>
<td>163.839263</td>
<td>163.8182</td>
</tr>
<tr>
<td>$Q$ (m$^3$.d$^{-1}$)</td>
<td>269.133600</td>
<td>269.1373</td>
</tr>
<tr>
<td>$T$ (°C)</td>
<td>14.856600</td>
<td>14.8581</td>
</tr>
<tr>
<td>Output variables</td>
<td>FORTRAN</td>
<td>MATLAB</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>$S_l$ (g COD.m$^{-3}$)</td>
<td>28.064300</td>
<td>28.0643</td>
</tr>
<tr>
<td>$S_h$ (g COD.m$^{-3}$)</td>
<td>0.671945</td>
<td>0.6734</td>
</tr>
<tr>
<td>$X_l$ (g COD.m$^{-3}$)</td>
<td>28920.100000</td>
<td>28923</td>
</tr>
<tr>
<td>$X_h$ (g COD.m$^{-3}$)</td>
<td>602.410000</td>
<td>602.4207</td>
</tr>
<tr>
<td>$X_{BH}$ (g COD.m$^{-3}$)</td>
<td>42324.600000</td>
<td>42323</td>
</tr>
<tr>
<td>$X_{BA}$ (g COD.m$^{-3}$)</td>
<td>3170.720000</td>
<td>3168.3</td>
</tr>
<tr>
<td>$X_P$ (g COD.m$^{-3}$)</td>
<td>18315.400000</td>
<td>18317</td>
</tr>
<tr>
<td>$S_0$ (g -COD.m$^{-3}$)</td>
<td>1.373160</td>
<td>1.3748</td>
</tr>
<tr>
<td>$S_{NO}$ (g N.m$^{-3}$)</td>
<td>9.194400</td>
<td>9.1948</td>
</tr>
<tr>
<td>$S_{NH}$ (g N.m$^{-3}$)</td>
<td>0.158835</td>
<td>0.1585</td>
</tr>
<tr>
<td>$S_{ND}$ (g N/m$^3$)</td>
<td>0.560279</td>
<td>0.5594</td>
</tr>
<tr>
<td>$X_{ND}$ (g N/m$^3$)</td>
<td>45.162800</td>
<td>45.1637</td>
</tr>
<tr>
<td>$S_{ALK}$ (mole HCO$_3^-$m$^{-3}$)</td>
<td>4.564060</td>
<td>4.5646</td>
</tr>
<tr>
<td>$TSS$ (g.m$^{-3}$)</td>
<td>69999.922500</td>
<td>70000</td>
</tr>
<tr>
<td>$Q$ (m$^3$.d$^{-1}$)</td>
<td>30.866160</td>
<td>30.8627</td>
</tr>
<tr>
<td>$T$ (°C)</td>
<td>14.856600</td>
<td>14.8581</td>
</tr>
<tr>
<td>CONCENTRATION values</td>
<td>ASM2ADM OUTPUT</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td><strong>Output variables</strong></td>
<td><strong>FORTRAN</strong></td>
<td></td>
</tr>
<tr>
<td>S\textsubscript{su} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>S\textsubscript{aa} (kg COD.m\textsuperscript{-3})</td>
<td>0.043902</td>
<td></td>
</tr>
<tr>
<td>S\textsubscript{fa} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>S\textsubscript{va} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>S\textsubscript{bu} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>S\textsubscript{pro} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>S\textsubscript{ac} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>S\textsubscript{h2} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>S\textsubscript{ch4} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>S\textsubscript{IC} (kmole C.m\textsuperscript{-3})</td>
<td>0.007918</td>
<td></td>
</tr>
<tr>
<td>S\textsubscript{IN} (kmole N.m\textsuperscript{-3})</td>
<td>0.001972</td>
<td></td>
</tr>
<tr>
<td>S\textsubscript{I} (kg COD.m\textsuperscript{-3})</td>
<td>0.028067</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{c} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{ch} (kg COD.m\textsuperscript{-3})</td>
<td>3.723580</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{pr} (kg COD.m\textsuperscript{-3})</td>
<td>15.924300</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{li} (kg COD.m\textsuperscript{-3})</td>
<td>8.046860</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{su} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{aa} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{fa} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{c4} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{pro} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{ac} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{h2} (kg COD.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{I} (kg COD.m\textsuperscript{-3})</td>
<td>17.011000</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{cat} (kmole.m\textsuperscript{-3})</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>X\textsubscript{an} (kmole.m\textsuperscript{-3})</td>
<td>0.005210</td>
<td></td>
</tr>
<tr>
<td>TSS (kg.m\textsuperscript{-3})</td>
<td>Not used in ADM</td>
<td></td>
</tr>
<tr>
<td>Q (m\textsuperscript{3}.d\textsuperscript{-1})</td>
<td>178.471000</td>
<td></td>
</tr>
<tr>
<td>T (°C)</td>
<td>35.000000</td>
<td></td>
</tr>
</tbody>
</table>
## Concentration Values

<table>
<thead>
<tr>
<th>Output Variables</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ssu (kg COD.m⁻³)</td>
<td>0.012395</td>
<td>0.0124</td>
</tr>
<tr>
<td>Ssu (kg COD.m⁻³)</td>
<td>0.005543</td>
<td>0.0055</td>
</tr>
<tr>
<td>Sfa (kg COD.m⁻³)</td>
<td>0.107409</td>
<td>0.1074</td>
</tr>
<tr>
<td>Sva (kg COD.m⁻³)</td>
<td>0.012332</td>
<td>0.0123</td>
</tr>
<tr>
<td>Sbu (kg COD.m⁻³)</td>
<td>0.014003</td>
<td>0.014</td>
</tr>
<tr>
<td>Spro (kg COD.m⁻³)</td>
<td>0.017584</td>
<td>0.0176</td>
</tr>
<tr>
<td>Sac (kg COD.m⁻³)</td>
<td>0.092837</td>
<td>0.0893</td>
</tr>
<tr>
<td>Sh2 (kg COD.m⁻³)</td>
<td>0.000000</td>
<td>2.51E-07</td>
</tr>
<tr>
<td>Sgas,h2 (kg COD.m⁻³)</td>
<td>0.000011</td>
<td>1.10E-05</td>
</tr>
<tr>
<td>pH</td>
<td>7.270000</td>
<td>7.2631</td>
</tr>
<tr>
<td>Pgas,h2 (bar)</td>
<td>1.013</td>
<td>1.10E-05</td>
</tr>
<tr>
<td>Pgas,h2 (bar)</td>
<td>0.000011</td>
<td>1.10E-05</td>
</tr>
<tr>
<td>Pgas,h2 (bar)</td>
<td>1.662680</td>
<td>1.6535</td>
</tr>
<tr>
<td>Pgas,h2 (bar)</td>
<td>0.013276</td>
<td>0.0135</td>
</tr>
<tr>
<td>Pgas,h2 (bar)</td>
<td>0.665582</td>
<td>0.6619</td>
</tr>
<tr>
<td>Pgas,h2 (bar)</td>
<td>0.340125</td>
<td>0.3469</td>
</tr>
<tr>
<td>Pgas,h2 (bar)</td>
<td>1.061890</td>
<td>1.0645</td>
</tr>
<tr>
<td>Pgas,h2 (bar)</td>
<td>2686.256739</td>
<td>2708.3</td>
</tr>
</tbody>
</table>

## Digestion Output

- **Q** (m³.d⁻¹): 178.471000
- **T** (°C): 35
- **pH**: 7.270000
- **TSS** (kg.m⁻³): not used in ADM
- **Qgas** (m³.d⁻¹): 2686.256739

### Footnotes

- T-dependent
- Normalized to $P_{atm}$
<table>
<thead>
<tr>
<th>CONCENTRATION values</th>
<th>ADM2ASM OUTPUT</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output variables</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_I$ (g COD.m$^{-3}$)</td>
<td>130.8674</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_S$ (g COD.m$^{-3}$)</td>
<td>258.5789</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_I$ (g COD.m$^{-3}$)</td>
<td>17216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_S$ (g COD.m$^{-3}$)</td>
<td>2611.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{BH}$ (g COD.m$^{-3}$)</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{BA}$ (g COD.m$^{-3}$)</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_P$ (g COD.m$^{-3}$)</td>
<td>626.0654</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_O$ (g -COD.m$^{-3}$)</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{NO}$ (g N.m$^{-3}$)</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{NH}$ (g N.m$^{-3}$)</td>
<td>1442.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{ND}$ (g N.m$^{-3}$)</td>
<td>0.5432</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{ND}$ (g N.m$^{-3}$)</td>
<td>100.8669</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{ALK}$ (mole HCO$_3^-$ m$^{-3}$)</td>
<td>97.846</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$TSS$ (g.m$^{-3}$)</td>
<td>15340</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Q$ (m$^3$.d$^{-1}$)</td>
<td>178.4674</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T$ ($^\circ$C)</td>
<td>14.8581</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CONCENTRATION values</th>
<th>DEWATERING EFFLUENT</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output variables</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_I$ (g COD.m$^{-3}$)</td>
<td>130.863000</td>
<td>130.8674</td>
<td></td>
</tr>
<tr>
<td>$S_S$ (g COD.m$^{-3}$)</td>
<td>262.103000</td>
<td>258.5789</td>
<td></td>
</tr>
<tr>
<td>$X_I$ (g COD.m$^{-3}$)</td>
<td>363.869000</td>
<td>363.8587</td>
<td></td>
</tr>
<tr>
<td>$X_S$ (g COD.m$^{-3}$)</td>
<td>55.192600</td>
<td>55.1931</td>
<td></td>
</tr>
<tr>
<td>$X_{BH}$ (g COD.m$^{-3}$)</td>
<td>0.000000</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$X_{BA}$ (g COD.m$^{-3}$)</td>
<td>0.000000</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$X_P$ (g COD.m$^{-3}$)</td>
<td>13.231600</td>
<td>13.2317</td>
<td></td>
</tr>
<tr>
<td>$S_O$ (g -COD.m$^{-3}$)</td>
<td>0.000000</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$S_{NO}$ (g N.m$^{-3}$)</td>
<td>0.000000</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$S_{NH}$ (g N.m$^{-3}$)</td>
<td>1443.440000</td>
<td>1442.8</td>
<td></td>
</tr>
<tr>
<td>$S_{ND}$ (g N.m$^{-3}$)</td>
<td>0.543236</td>
<td>0.5432</td>
<td></td>
</tr>
<tr>
<td>$X_{ND}$ (g N.m$^{-3}$)</td>
<td>2.131780</td>
<td>2.1318</td>
<td></td>
</tr>
<tr>
<td>$S_{ALK}$ (mole HCO$_3^-$ m$^{-3}$)</td>
<td>97.823400</td>
<td>97.846</td>
<td></td>
</tr>
<tr>
<td>$TSS$ (g.m$^{-3}$)</td>
<td>324.219900</td>
<td>324.2126</td>
<td></td>
</tr>
<tr>
<td>$Q$ (m$^3$.d$^{-1}$)</td>
<td>168.888480</td>
<td>168.8853</td>
<td></td>
</tr>
<tr>
<td>$T$ ($^\circ$C)</td>
<td>14.858100</td>
<td>14.8581</td>
<td></td>
</tr>
</tbody>
</table>
### Effluent average concentrations based on load

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effluent average flow rate</td>
<td>m³.d⁻¹</td>
<td>20640.000000</td>
<td>20640.7792</td>
</tr>
<tr>
<td>Effluent average $S_I$ concentration</td>
<td>g COD.m⁻³</td>
<td>28.060000</td>
<td>28.0643</td>
</tr>
<tr>
<td>Effluent average $S_O$ concentration</td>
<td>g COD.m⁻³</td>
<td>0.671900</td>
<td>0.67336</td>
</tr>
<tr>
<td>Effluent average $X_I$ concentration</td>
<td>g COD.m⁻³</td>
<td>5.919000</td>
<td>5.9191</td>
</tr>
<tr>
<td>Effluent average $X_S$ concentration</td>
<td>g COD.m⁻³</td>
<td>0.123300</td>
<td>0.12329</td>
</tr>
<tr>
<td>Effluent average $X_{BH}$ concentration</td>
<td>g COD.m⁻³</td>
<td>8.662000</td>
<td>8.6614</td>
</tr>
<tr>
<td>Effluent average $X_{BA}$ concentration</td>
<td>g COD.m⁻³</td>
<td>0.648900</td>
<td>0.6484</td>
</tr>
<tr>
<td>Effluent average $S_{SO}$ concentration</td>
<td>g COD.m⁻³</td>
<td>1.373000</td>
<td>1.3748</td>
</tr>
<tr>
<td>Effluent average $S_{NO}$ concentration</td>
<td>g N.m⁻³</td>
<td>9.190000</td>
<td>9.1948</td>
</tr>
<tr>
<td>Effluent average $S_{NH}$ concentration (limit = 4 g N.m⁻³)</td>
<td>g N.m⁻³</td>
<td>0.158800</td>
<td>0.15845</td>
</tr>
<tr>
<td>Effluent average $S_{ND}$ concentration</td>
<td>g N.m⁻³</td>
<td>0.560300</td>
<td>0.55943</td>
</tr>
<tr>
<td>Effluent average $X_{ND}$ concentration</td>
<td>g N.m⁻³</td>
<td>0.009243</td>
<td>0.0092428</td>
</tr>
<tr>
<td>Effluent average $S_{ALK}$ concentration</td>
<td>mole HCO₃⁻.m⁻³</td>
<td>4.564000</td>
<td>4.5646</td>
</tr>
<tr>
<td>Effluent average TSS concentration (limit = 30 g SS.m⁻³)</td>
<td>g.m⁻³</td>
<td>14.330000</td>
<td>14.3255</td>
</tr>
<tr>
<td>Effluent average Temperature</td>
<td>°C</td>
<td>14.856600</td>
<td>14.8581</td>
</tr>
<tr>
<td>Effluent average Kjeldahl N concentration</td>
<td>g N.m⁻³</td>
<td>2.053000</td>
<td>2.052</td>
</tr>
<tr>
<td>Effluent average total N concentration (limit = 18 g N.m⁻³)</td>
<td>g N.m⁻³</td>
<td>11.250000</td>
<td>11.2468</td>
</tr>
<tr>
<td>Effluent average total COD concentration (limit = 100 g COD.m⁻³)</td>
<td>g COD.m⁻³</td>
<td>47.840000</td>
<td>47.8383</td>
</tr>
<tr>
<td>Effluent average BOD₅ concentration (limit = 10 g.m⁻³)</td>
<td>g.m⁻³</td>
<td>2.340000</td>
<td>2.3404</td>
</tr>
</tbody>
</table>

### Effluent average load

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effluent average $S_I$ load</td>
<td>kg COD.d⁻¹</td>
<td>579.158400</td>
<td>579.269</td>
</tr>
<tr>
<td>Effluent average $S_O$ load</td>
<td>kg COD.d⁻¹</td>
<td>13.868016</td>
<td>13.8988</td>
</tr>
<tr>
<td>Effluent average $X_I$ load</td>
<td>kg COD.d⁻¹</td>
<td>122.168160</td>
<td>122.1748</td>
</tr>
<tr>
<td>Effluent average $X_S$ load</td>
<td>kg COD.d⁻¹</td>
<td>2.544912</td>
<td>2.5447</td>
</tr>
<tr>
<td>Effluent average $X_{BH}$ load</td>
<td>kg COD.d⁻¹</td>
<td>178.783680</td>
<td>178.7776</td>
</tr>
<tr>
<td>Effluent average $X_{BA}$ load</td>
<td>kg COD.d⁻¹</td>
<td>13.393296</td>
<td>13.3835</td>
</tr>
<tr>
<td>Effluent average $X_P$ load</td>
<td>kg COD.d⁻¹</td>
<td>77.358720</td>
<td>77.3722</td>
</tr>
<tr>
<td>Effluent average $S_{SO}$ load</td>
<td>kg COD.d⁻¹</td>
<td>28.338720</td>
<td>28.376</td>
</tr>
<tr>
<td>Effluent average $S_{NO}$ load</td>
<td>kg N.d⁻¹</td>
<td>189.764160</td>
<td>189.7888</td>
</tr>
<tr>
<td>Effluent average $S_{NH}$ load</td>
<td>kg N.d⁻¹</td>
<td>3.277632</td>
<td>3.2706</td>
</tr>
<tr>
<td>Effluent average $S_{ND}$ load</td>
<td>kg N.d⁻¹</td>
<td>11.564592</td>
<td>11.547</td>
</tr>
<tr>
<td>Effluent average $X_{ND}$ load</td>
<td>kg N.d⁻¹</td>
<td>0.190776</td>
<td>0.19078</td>
</tr>
<tr>
<td>Effluent average $S_{ALK}$ load</td>
<td>kmole HCO₃⁻.d⁻¹</td>
<td>94.200960</td>
<td>94.216</td>
</tr>
<tr>
<td>Effluent average TSS load</td>
<td>kg.d⁻¹</td>
<td>295.771200</td>
<td>295.6896</td>
</tr>
<tr>
<td>Effluent average Kjeldahl N load</td>
<td>kg N.d⁻¹</td>
<td>42.373920</td>
<td>42.3541</td>
</tr>
<tr>
<td>Effluent average total N load</td>
<td>kg N.d⁻¹</td>
<td>232.200000</td>
<td>232.1429</td>
</tr>
<tr>
<td>Effluent average total COD load</td>
<td>kg COD.d⁻¹</td>
<td>987.417600</td>
<td>987.4206</td>
</tr>
<tr>
<td>Effluent average BOD₅ load</td>
<td>kg.d⁻¹</td>
<td>48.297600</td>
<td>48.3079</td>
</tr>
</tbody>
</table>
### Other output quality variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Influent quality ((IQI)) index</td>
<td>kg poll.units.d(^{-1})</td>
<td>74700.000000</td>
<td>74746.1235</td>
</tr>
<tr>
<td>Effluent quality ((EQI)) index</td>
<td>kg poll.units.d(^{-1})</td>
<td>4840.000000</td>
<td>4843.9256</td>
</tr>
<tr>
<td>Average sludge production for disposal per day</td>
<td>kg SS.d(^{-1})</td>
<td>2683.000000</td>
<td>2682.9648</td>
</tr>
<tr>
<td>Average sludge production released into effluent per day</td>
<td>kg SS.d(^{-1})</td>
<td>296.000000</td>
<td>295.6896</td>
</tr>
<tr>
<td>Total average sludge production per day</td>
<td>kg SS.d(^{-1})</td>
<td>2979.000000</td>
<td>2978.6544</td>
</tr>
</tbody>
</table>

### ‘Energy’ related variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average aeration energy</td>
<td>kWh.d(^{-1})</td>
<td>4000.000000</td>
<td>4000</td>
</tr>
<tr>
<td>Average pumping energy</td>
<td>kWh.d(^{-1})</td>
<td>440.900000</td>
<td>441.5576</td>
</tr>
<tr>
<td>Average carbon source dosage</td>
<td>kg COD.d(^{-1})</td>
<td>800.000000</td>
<td>800</td>
</tr>
<tr>
<td>Average mixing energy</td>
<td>kWh.d(^{-1})</td>
<td>768.000000</td>
<td>768</td>
</tr>
<tr>
<td>Average heating energy</td>
<td>kWh.d(^{-1})</td>
<td>4180.000000</td>
<td>4179.8063</td>
</tr>
<tr>
<td>Average methane gas production (1 kg = 13.8928 kWh)</td>
<td>kg CH(_4).d(^{-1})</td>
<td>1065.000000</td>
<td>1065.3522</td>
</tr>
<tr>
<td>Average hydrogen gas production</td>
<td>kg H(_2).d(^{-1})</td>
<td>0.0035541</td>
<td></td>
</tr>
<tr>
<td>Average carbon dioxide gas production</td>
<td>kg CO(_2).d(^{-1})</td>
<td>1535.4117</td>
<td></td>
</tr>
<tr>
<td>Average total gas flow rate from AD (normalized to (P_{\text{atm}}))</td>
<td>'normal' m(^3).d(^{-1})</td>
<td>2562.580000</td>
<td>2708.3428</td>
</tr>
</tbody>
</table>

### Operational cost index

<table>
<thead>
<tr>
<th>Variable (including weight factor)</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sludge production cost index</td>
<td>–</td>
<td>8049.000000</td>
<td>8048.8944</td>
</tr>
<tr>
<td>Aeration energy cost index</td>
<td>–</td>
<td>4000.000000</td>
<td>4000</td>
</tr>
<tr>
<td>Pumping energy cost index</td>
<td>–</td>
<td>440.900000</td>
<td>441.5576</td>
</tr>
<tr>
<td>Carbon source dosage cost index</td>
<td>–</td>
<td>2400.000000</td>
<td>2400</td>
</tr>
<tr>
<td>Mixing energy cost index</td>
<td>–</td>
<td>768.000000</td>
<td>768</td>
</tr>
<tr>
<td>Heating energy cost index</td>
<td>–</td>
<td>0.000000</td>
<td>0</td>
</tr>
<tr>
<td>Net energy production from methane index (subtracted from rest)</td>
<td>–</td>
<td>6392.1131</td>
<td></td>
</tr>
<tr>
<td>Total Operational Cost Index ((OCI))</td>
<td>–</td>
<td>9267.000000</td>
<td>9266.3389</td>
</tr>
</tbody>
</table>
APPENDIX 4: Open-loop results

These results were obtained with FORTRAN (integration using a Runge-Kutta 4 algorithm with a constant integration step = 0.005 h) and MATLAB-Simulink (Solver: ode45, absolute tolerance = $10^{-8}$, relative tolerance = $10^{-5}$). For details on operation conditions see text.

### Effluent average concentrations based on load, including bypass

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effluent average flow rate</td>
<td>m$^3$.d$^{-1}$</td>
<td>20661.600000</td>
<td>20661.2186</td>
</tr>
<tr>
<td>Effluent average $S_I$ concentration</td>
<td>g COD.m$^{-3}$</td>
<td>28.050000</td>
<td>28.0493</td>
</tr>
<tr>
<td>Effluent average $S_S$ concentration</td>
<td>g COD.m$^{-3}$</td>
<td>0.809400</td>
<td>0.8123</td>
</tr>
<tr>
<td>Effluent average $X_I$ concentration</td>
<td>g COD.m$^{-3}$</td>
<td>6.489000</td>
<td>6.4896</td>
</tr>
<tr>
<td>Effluent average $X_S$ concentration</td>
<td>g COD.m$^{-3}$</td>
<td>0.348600</td>
<td>0.34855</td>
</tr>
<tr>
<td>Effluent average $X_{BH}$ concentration</td>
<td>g COD.m$^{-3}$</td>
<td>9.692000</td>
<td>9.6917</td>
</tr>
<tr>
<td>Effluent average $X_{BA}$ concentration</td>
<td>g COD.m$^{-3}$</td>
<td>0.675100</td>
<td>0.67445</td>
</tr>
<tr>
<td>Effluent average $X_P$ concentration</td>
<td>g COD.m$^{-3}$</td>
<td>3.991000</td>
<td>3.9911</td>
</tr>
<tr>
<td>Effluent average $S_O$ concentration</td>
<td>g -COD.m$^{-3}$</td>
<td>1.031000</td>
<td>1.0319</td>
</tr>
<tr>
<td>Effluent average $S_NO$ concentration</td>
<td>g N.m$^{-3}$</td>
<td>7.471000</td>
<td>7.4746</td>
</tr>
<tr>
<td>Effluent average $S_{NO}$ concentration (limit = 4 g N.m$^{-3}$)</td>
<td>g N.m$^{-3}$</td>
<td>1.651000</td>
<td>1.6491</td>
</tr>
<tr>
<td>Effluent average $S_{ND}$ concentration</td>
<td>g N.m$^{-3}$</td>
<td>0.320800</td>
<td>0.32018</td>
</tr>
<tr>
<td>Effluent average $X_{ND}$ concentration</td>
<td>g N.m$^{-3}$</td>
<td>0.020180</td>
<td>0.02018</td>
</tr>
<tr>
<td>Effluent average $S_{ALK}$ concentration</td>
<td>mole HCO$_3$^{-}.m$^{-3}$</td>
<td>4.796000</td>
<td>4.796</td>
</tr>
<tr>
<td>Effluent average $TSS$ concentration (limit = 30 g SS.m$^{-3}$)</td>
<td>g.m$^{-3}$</td>
<td>15.900000</td>
<td>15.8965</td>
</tr>
<tr>
<td>Effluent average Temperature</td>
<td>°C</td>
<td>14.8604</td>
<td>14.8604</td>
</tr>
<tr>
<td>Effluent average Kjeldahl N concentration</td>
<td>g N.m$^{-3}$</td>
<td>3.732000</td>
<td>3.7294</td>
</tr>
<tr>
<td>Effluent average total N concentration (limit = 18 g N.m$^{-3}$)</td>
<td>g N.m$^{-3}$</td>
<td>11.200000</td>
<td>11.204</td>
</tr>
<tr>
<td>Effluent average total COD concentration (limit = 100 g COD.m$^{-3}$)</td>
<td>g COD.m$^{-3}$</td>
<td>50.050000</td>
<td>50.0559</td>
</tr>
<tr>
<td>Effluent average BOD$_5$ concentration (limit = 10 g.m$^{-3}$)</td>
<td>g.m$^{-3}$</td>
<td>2.674000</td>
<td>2.6742</td>
</tr>
</tbody>
</table>

### Effluent average load, including bypass

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effluent average $S_I$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>579.557880</td>
<td>579.5322</td>
</tr>
<tr>
<td>Effluent average $S_S$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>16.723499</td>
<td>16.761</td>
</tr>
<tr>
<td>Effluent average $X_I$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>134.073122</td>
<td>134.0829</td>
</tr>
<tr>
<td>Effluent average $X_S$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>7.202634</td>
<td>7.2014</td>
</tr>
<tr>
<td>Effluent average $X_{BH}$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>200.252277</td>
<td>200.2421</td>
</tr>
<tr>
<td>Effluent average $X_{BA}$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>13.948646</td>
<td>13.9349</td>
</tr>
<tr>
<td>Effluent average $X_P$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>82.460446</td>
<td>82.4607</td>
</tr>
<tr>
<td>Effluent average $S_O$ load</td>
<td>kg -COD.d$^{-1}$</td>
<td>21.302110</td>
<td>21.3202</td>
</tr>
<tr>
<td>Effluent average $S_NO$ load</td>
<td>kg N.d$^{-1}$</td>
<td>154.362814</td>
<td>154.435</td>
</tr>
<tr>
<td>Effluent average $S_{ND}$ load</td>
<td>kg N.d$^{-1}$</td>
<td>34.112302</td>
<td>34.0731</td>
</tr>
<tr>
<td>Effluent average $X_{ND}$ load</td>
<td>kg N.d$^{-1}$</td>
<td>12.454812</td>
<td>12.4366</td>
</tr>
<tr>
<td>Effluent average $S_{ALK}$ load</td>
<td>kg N.d$^{-1}$</td>
<td>0.416951</td>
<td>0.41694</td>
</tr>
<tr>
<td>Effluent average $TSS$ load</td>
<td>kmole HCO$_3$^{-}.d$^{-1}$</td>
<td>99.093034</td>
<td>99.0915</td>
</tr>
<tr>
<td>Effluent average $Kjeldahl N$ load</td>
<td>kg.d$^{-1}$</td>
<td>328.519440</td>
<td>328.4415</td>
</tr>
<tr>
<td>Effluent average total N load</td>
<td>kg N.d$^{-1}$</td>
<td>77.109091</td>
<td>77.0534</td>
</tr>
<tr>
<td>Effluent average total COD load</td>
<td>kg COD.d$^{-1}$</td>
<td>231.409920</td>
<td>231.4884</td>
</tr>
<tr>
<td>Effluent average BOD$_5$ load</td>
<td>kg.d$^{-1}$</td>
<td>1034.113080</td>
<td>1034.2151</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effluent average $S_I$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>579.557880</td>
<td>579.5322</td>
</tr>
<tr>
<td>Effluent average $S_S$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>16.723499</td>
<td>16.761</td>
</tr>
<tr>
<td>Effluent average $X_I$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>134.073122</td>
<td>134.0829</td>
</tr>
<tr>
<td>Effluent average $X_S$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>7.202634</td>
<td>7.2014</td>
</tr>
<tr>
<td>Effluent average $X_{BH}$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>200.252277</td>
<td>200.2421</td>
</tr>
<tr>
<td>Effluent average $X_{BA}$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>13.948646</td>
<td>13.9349</td>
</tr>
<tr>
<td>Effluent average $X_P$ load</td>
<td>kg COD.d$^{-1}$</td>
<td>82.460446</td>
<td>82.4607</td>
</tr>
<tr>
<td>Effluent average $S_O$ load</td>
<td>kg -COD.d$^{-1}$</td>
<td>21.302110</td>
<td>21.3202</td>
</tr>
<tr>
<td>Effluent average $S_NO$ load</td>
<td>kg N.d$^{-1}$</td>
<td>154.362814</td>
<td>154.435</td>
</tr>
<tr>
<td>Effluent average $S_{ND}$ load</td>
<td>kg N.d$^{-1}$</td>
<td>34.112302</td>
<td>34.0731</td>
</tr>
<tr>
<td>Effluent average $X_{ND}$ load</td>
<td>kg N.d$^{-1}$</td>
<td>12.454812</td>
<td>12.4366</td>
</tr>
<tr>
<td>Effluent average $S_{ALK}$ load</td>
<td>kg N.d$^{-1}$</td>
<td>0.416951</td>
<td>0.41694</td>
</tr>
<tr>
<td>Effluent average $TSS$ load</td>
<td>kmole HCO$_3$^{-}.d$^{-1}$</td>
<td>99.093034</td>
<td>99.0915</td>
</tr>
<tr>
<td>Effluent average $Kjeldahl N$ load</td>
<td>kg.d$^{-1}$</td>
<td>328.519440</td>
<td>328.4415</td>
</tr>
<tr>
<td>Effluent average total N load</td>
<td>kg N.d$^{-1}$</td>
<td>77.109091</td>
<td>77.0534</td>
</tr>
<tr>
<td>Effluent average total COD load</td>
<td>kg COD.d$^{-1}$</td>
<td>231.409920</td>
<td>231.4884</td>
</tr>
<tr>
<td>Effluent average BOD$_5$ load</td>
<td>kg.d$^{-1}$</td>
<td>1034.113080</td>
<td>1034.2151</td>
</tr>
</tbody>
</table>
## Other output quality variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Influent quality ($IQI$) index</td>
<td>kg poll.units.d$^{-1}$</td>
<td>74700.000000</td>
<td>74783.3138</td>
</tr>
<tr>
<td>Effluent quality ($EQI$) index</td>
<td>kg poll.units.d$^{-1}$</td>
<td>5660.000000</td>
<td>5657.5529</td>
</tr>
<tr>
<td>Average sludge production for disposal per day</td>
<td>kg SS.d$^{-1}$</td>
<td>2646.000000</td>
<td>2651.8714</td>
</tr>
<tr>
<td>Average sludge production released into effluent per day</td>
<td>kg SS.d$^{-1}$</td>
<td>325.000000</td>
<td>328.4415</td>
</tr>
<tr>
<td>Total average sludge production per day</td>
<td>kg SS.d$^{-1}$</td>
<td>2971.000000</td>
<td>2980.3128</td>
</tr>
</tbody>
</table>

## 'Energy' related variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average aeration energy</td>
<td>kWh.d$^{-1}$</td>
<td>4000.000000</td>
<td>4000</td>
</tr>
<tr>
<td>Average pumping energy</td>
<td>kWh.d$^{-1}$</td>
<td>440.900000</td>
<td>441.5362</td>
</tr>
<tr>
<td>Average carbon source dosage</td>
<td>kg COD.d$^{-1}$</td>
<td>800.000000</td>
<td>800</td>
</tr>
<tr>
<td>Average mixing energy</td>
<td>kWh.d$^{-1}$</td>
<td>768.000000</td>
<td>768</td>
</tr>
<tr>
<td>Average heating energy</td>
<td>kWh.d$^{-1}$</td>
<td>4177.000000</td>
<td>4177.3089</td>
</tr>
<tr>
<td>Average methane gas production (1 kg = 13.8928 kWh)</td>
<td>kg CH$_4$.d$^{-1}$</td>
<td>1057.000000</td>
<td>1059.4972</td>
</tr>
<tr>
<td>Average hydrogen gas production</td>
<td>kg H$_2$.d$^{-1}$</td>
<td>0.003576</td>
<td>0.0036065</td>
</tr>
<tr>
<td>Average carbon dioxide gas production</td>
<td>kg CO$_2$.d$^{-1}$</td>
<td>1485.000000</td>
<td>1527.5286</td>
</tr>
<tr>
<td>Average total gas flow rate from AD (normalized to $P_{sat}$)</td>
<td>'normal' m$^3$.d$^{-1}$</td>
<td>2547.000000</td>
<td>2693.6501</td>
</tr>
</tbody>
</table>

## Operational cost index

<table>
<thead>
<tr>
<th>Variable (including weight factor)</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sludge production cost index</td>
<td>–</td>
<td>7938.000000</td>
<td>7955.6141</td>
</tr>
<tr>
<td>Aeration energy cost index</td>
<td>–</td>
<td>4000.000000</td>
<td>4000</td>
</tr>
<tr>
<td>Pumping energy cost index</td>
<td>–</td>
<td>440.900000</td>
<td>441.5362</td>
</tr>
<tr>
<td>Carbon source dosage cost index</td>
<td>–</td>
<td>2400.000000</td>
<td>2400</td>
</tr>
<tr>
<td>Mixing energy cost index</td>
<td>–</td>
<td>768.000000</td>
<td>768</td>
</tr>
<tr>
<td>Heating energy cost index</td>
<td>–</td>
<td>0.000000</td>
<td>0</td>
</tr>
<tr>
<td>Net energy production from methane index (subtracted from rest)</td>
<td>–</td>
<td>6356.9833</td>
<td></td>
</tr>
<tr>
<td>Total Operational Cost Index ($OCI$)</td>
<td>–</td>
<td>9208.000000</td>
<td>9208.1669</td>
</tr>
</tbody>
</table>

## Effluent violations

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>95% percentile of effluent SNH (Ammonia95)</td>
<td>g N.m$^{-3}$</td>
<td>4.650000</td>
<td>4.6523</td>
</tr>
<tr>
<td>95% percentile of effluent total N (TN95)</td>
<td>g N.m$^{-3}$</td>
<td>15.100000</td>
<td>15.1312</td>
</tr>
<tr>
<td>95% percentile of effluent TSS (TSS95)</td>
<td>g COD.m$^{-3}$</td>
<td>20.100000</td>
<td>20.1336</td>
</tr>
<tr>
<td>Maximum effluent total N limit (18 g N.m$^{-3}$) was violated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>during:</td>
<td>days</td>
<td>0.374920</td>
<td>0.39583</td>
</tr>
<tr>
<td>% of total evaluation time:</td>
<td>%</td>
<td>0.103000</td>
<td>0.10875</td>
</tr>
<tr>
<td>number of violations:</td>
<td>–</td>
<td>4.000000</td>
<td>5</td>
</tr>
<tr>
<td>Maximum effluent total COD limit (100 g COD.m$^{-3}$) was violated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>during:</td>
<td>days</td>
<td>0.218764</td>
<td>0.21875</td>
</tr>
<tr>
<td>% of total evaluation time:</td>
<td>%</td>
<td>0.060100</td>
<td>0.060096</td>
</tr>
<tr>
<td>number of violations:</td>
<td>–</td>
<td>3.000000</td>
<td>3</td>
</tr>
<tr>
<td>Maximum effluent ammonia limit (4 g N.m$^{-3}$) was violated</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
During: | days | 30.175600 | 30.1042 |
---|---|---|---|
% of total evaluation time: | % | 8.290000 | 8.2704 |
Number of violations: | – | 172.000000 | 171 |
**Maximum effluent TSS limit (30 g SS.m⁻³) was violated**

During: | days | 1.426880 | 1.4375 |
---|---|---|---|
% of total evaluation time: | % | 0.392000 | 0.39492 |
Number of violations: | – | 12.000000 | 12 |

**Maximum effluent BOD₅ limit (10 g.m⁻³) was violated**

During: | days | 0.436800 | 0.4375 |
---|---|---|---|
% of total evaluation time: | % | 0.120000 | 0.12019 |
Number of violations: | – | 5.000000 | 5 |
APPENDIX 5: Closed-loop results with ideal sensors and actuators

These results were obtained with FORTRAN (integration using a Runge-Kutta 4 algorithm with a constant integration step = 0.005 h) and MATLAB-Simulink (Solver: ode45, absolute tolerance = 10^{-8}, relative tolerance = 10^{-5}). For details on operation conditions see text. Details on PI controller settings are given in table headings.

### Effluent average concentrations based on load, including bypass

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effluent average flow rate</td>
<td>m³.d⁻¹</td>
<td>20661.600000</td>
<td>20661.0229</td>
</tr>
<tr>
<td>Effluent average S_i concentration</td>
<td>g COD.m⁻³</td>
<td>28.060000</td>
<td>28.0618</td>
</tr>
<tr>
<td>Effluent average S_x concentration</td>
<td>g COD.m⁻³</td>
<td>0.724600</td>
<td>0.72625</td>
</tr>
<tr>
<td>Effluent average X_i concentration</td>
<td>g COD.m⁻³</td>
<td>5.913000</td>
<td>5.9139</td>
</tr>
<tr>
<td>Effluent average X_S concentration</td>
<td>g COD.m⁻³</td>
<td>0.331300</td>
<td>0.33129</td>
</tr>
<tr>
<td>Effluent average X_BH concentration</td>
<td>g COD.m⁻³</td>
<td>9.892000</td>
<td>9.8922</td>
</tr>
<tr>
<td>Effluent average X_BA concentration</td>
<td>g COD.m⁻³</td>
<td>0.691500</td>
<td>0.69097</td>
</tr>
<tr>
<td>Effluent average X_P concentration</td>
<td>g COD.m⁻³</td>
<td>3.400000</td>
<td>3.4003</td>
</tr>
<tr>
<td>Effluent average X_SO concentration</td>
<td>g -COD.m⁻³</td>
<td>1.577000</td>
<td>1.5791</td>
</tr>
<tr>
<td>Effluent average X_NO concentration</td>
<td>g N.m⁻³</td>
<td>11.060000</td>
<td>11.0536</td>
</tr>
<tr>
<td>Effluent average X_NH concentration</td>
<td>g N.m⁻³</td>
<td>0.019080</td>
<td>0.019077</td>
</tr>
<tr>
<td>Effluent average X_ND concentration</td>
<td>g N.m⁻³</td>
<td>4.455000</td>
<td>4.4553</td>
</tr>
<tr>
<td>Effluent average S_SO concentration</td>
<td>g N.m⁻³</td>
<td>0.473600</td>
<td>0.47377</td>
</tr>
<tr>
<td>Effluent average S_NO concentration</td>
<td>g N.m⁻³</td>
<td>0.585900</td>
<td>0.58498</td>
</tr>
<tr>
<td>Effluent average S_NH concentration</td>
<td>g N.m⁻³</td>
<td>0.019080</td>
<td>0.019077</td>
</tr>
<tr>
<td>Effluent average S_ND concentration</td>
<td>g N.m⁻³</td>
<td>15.170000</td>
<td>15.1713</td>
</tr>
<tr>
<td>Effluent average S_ALK concentration</td>
<td>mole HCO₃⁻.m⁻³</td>
<td>4.455000</td>
<td>4.4553</td>
</tr>
<tr>
<td>Effluent average TSS concentration (limit = 30 g SS.m⁻³)</td>
<td>g.m⁻³</td>
<td>15.170000</td>
<td>15.1713</td>
</tr>
<tr>
<td>Effluent average Temperature</td>
<td>°C</td>
<td>14.860000</td>
<td>14.8603</td>
</tr>
<tr>
<td>Effluent average Kjeldahl N concentration</td>
<td>g N.m⁻³</td>
<td>2.484000</td>
<td>2.4833</td>
</tr>
<tr>
<td>Effluent average total N concentration (limit = 18 g N.m⁻³)</td>
<td>g N.m⁻³</td>
<td>13.540000</td>
<td>13.5369</td>
</tr>
<tr>
<td>Effluent average total COD concentration (limit = 100 g COD.m⁻³)</td>
<td>g COD.m⁻³</td>
<td>49.020000</td>
<td>49.0169</td>
</tr>
<tr>
<td>Effluent average BOD₅ concentration (limit = 10 g.m⁻³)</td>
<td>g.m⁻³</td>
<td>2.698000</td>
<td>2.6985</td>
</tr>
</tbody>
</table>

### Effluent average load, including bypass

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effluent average S_i load</td>
<td>kg COD.d⁻¹</td>
<td>579.764969</td>
<td>579.7858</td>
</tr>
<tr>
<td>Effluent average S_x load</td>
<td>kg COD.d⁻¹</td>
<td>14.971395</td>
<td>15.005</td>
</tr>
<tr>
<td>Effluent average X_i load</td>
<td>kg COD.d⁻¹</td>
<td>122.172041</td>
<td>122.1863</td>
</tr>
<tr>
<td>Effluent average X_S load</td>
<td>kg COD.d⁻¹</td>
<td>6.845188</td>
<td>6.8449</td>
</tr>
<tr>
<td>Effluent average X_BH load</td>
<td>kg COD.d⁻¹</td>
<td>204.384547</td>
<td>204.3795</td>
</tr>
<tr>
<td>Effluent average X_BA load</td>
<td>kg COD.d⁻¹</td>
<td>14.287496</td>
<td>14.2761</td>
</tr>
<tr>
<td>Effluent average X_P load</td>
<td>kg COD.d⁻¹</td>
<td>70.249440</td>
<td>70.2529</td>
</tr>
<tr>
<td>Effluent average X_SO load</td>
<td>kg -COD.d⁻¹</td>
<td>32.583343</td>
<td>32.6228</td>
</tr>
<tr>
<td>Effluent average X_NO load</td>
<td>kg N.d⁻¹</td>
<td>228.517296</td>
<td>228.3783</td>
</tr>
<tr>
<td>Effluent average X_NH load</td>
<td>kg N.d⁻¹</td>
<td>9.785334</td>
<td>9.7886</td>
</tr>
<tr>
<td>Effluent average X_ND load</td>
<td>kg N.d⁻¹</td>
<td>12.105631</td>
<td>12.0864</td>
</tr>
<tr>
<td>Effluent average X_SH load</td>
<td>kg N.d⁻¹</td>
<td>0.394223</td>
<td>0.39415</td>
</tr>
<tr>
<td>Effluent average S_ALK load</td>
<td>kmol HCO₃⁻.d⁻¹</td>
<td>92.047428</td>
<td>92.0513</td>
</tr>
<tr>
<td>Effluent average TSS load</td>
<td>kg.d⁻¹</td>
<td>313.436472</td>
<td>313.4547</td>
</tr>
<tr>
<td>Effluent average Kjeldahl N load</td>
<td>kg N.d⁻¹</td>
<td>51.323414</td>
<td>51.3079</td>
</tr>
<tr>
<td>Effluent average total N load</td>
<td>kg N.d⁻¹</td>
<td>279.758064</td>
<td>279.6863</td>
</tr>
<tr>
<td>Effluent average total COD load</td>
<td>kg COD.d⁻¹</td>
<td>1012.831632</td>
<td>1012.7305</td>
</tr>
<tr>
<td>Effluent average BOD₅ load</td>
<td>kg.d⁻¹</td>
<td>55.744997</td>
<td>55.7533</td>
</tr>
</tbody>
</table>
### Other output quality variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Influent quality (IQI) index</td>
<td>kg poll.units.d⁻¹</td>
<td>74700.000000</td>
<td>74783.3138</td>
</tr>
<tr>
<td>Effluent quality (EQI) index</td>
<td>kg poll.units.d⁻¹</td>
<td>5580.000000</td>
<td>5574.1681</td>
</tr>
<tr>
<td>Average sludge production for disposal per day</td>
<td>kg SS.d⁻¹</td>
<td>2702.000000</td>
<td>2707.7709</td>
</tr>
<tr>
<td>Average sludge production released into effluent per day</td>
<td>kg SS.d⁻¹</td>
<td>309.000000</td>
<td>313.4547</td>
</tr>
<tr>
<td>Total average sludge production per day</td>
<td>kg SS.d⁻¹</td>
<td>3011.000000</td>
<td>3021.2257</td>
</tr>
</tbody>
</table>

### 'Energy' related variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average aeration energy</td>
<td>kWh.d⁻¹</td>
<td>4223.000000</td>
<td>4222.6368</td>
</tr>
<tr>
<td>Average pumping energy</td>
<td>kWh.d⁻¹</td>
<td>444.800000</td>
<td>445.4525</td>
</tr>
<tr>
<td>Average carbon source dosage</td>
<td>kg COD.d⁻¹</td>
<td>800</td>
<td>800</td>
</tr>
<tr>
<td>Average mixing energy</td>
<td>kWh.d⁻¹</td>
<td>768.000000</td>
<td>768</td>
</tr>
<tr>
<td>Average heating energy</td>
<td>kWh.d⁻¹</td>
<td>4226.000000</td>
<td>4225.3462</td>
</tr>
<tr>
<td>Average methane gas production (1 kg = 13.8928 kWh)</td>
<td>kg CH₄.d⁻¹</td>
<td>1083.000000</td>
<td>1085.3603</td>
</tr>
<tr>
<td>Average hydrogen gas production</td>
<td>kg H₂.d⁻¹</td>
<td>0.003681</td>
<td>0.0037088</td>
</tr>
<tr>
<td>Average carbon dioxide gas production</td>
<td>kg CO₂.d⁻¹</td>
<td>1521.000000</td>
<td>1562.6873</td>
</tr>
<tr>
<td>Average total gas flow rate from AD (normalized to P_{atm})</td>
<td>'normal' m³.d⁻¹</td>
<td>2605.000000</td>
<td>2757.956</td>
</tr>
</tbody>
</table>

### Operational cost index

<table>
<thead>
<tr>
<th>Variable (including weight factor)</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sludge production cost index</td>
<td>–</td>
<td>8106.000000</td>
<td>8123.3128</td>
</tr>
<tr>
<td>Aeration energy cost index</td>
<td>–</td>
<td>4223.000000</td>
<td>4222.6368</td>
</tr>
<tr>
<td>Pumping energy cost index</td>
<td>–</td>
<td>444.800000</td>
<td>445.4525</td>
</tr>
<tr>
<td>Carbon source dosage cost index</td>
<td>–</td>
<td>2400.000000</td>
<td>2400</td>
</tr>
<tr>
<td>Mixing energy cost index</td>
<td>–</td>
<td>768.000000</td>
<td>768</td>
</tr>
<tr>
<td>Heating energy cost index</td>
<td>–</td>
<td>0.000000</td>
<td>0</td>
</tr>
<tr>
<td>Net energy production from methane index (subtracted from rest)</td>
<td>–</td>
<td>6512.1616</td>
<td></td>
</tr>
<tr>
<td>Total Operational Cost Index (OCI)</td>
<td>–</td>
<td>9443.000000</td>
<td>9447.2406</td>
</tr>
</tbody>
</table>
### Effluent violations

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>95% percentile of effluent $S_{NH}$ (Ammonia95)</td>
<td>g N.m⁻³</td>
<td>1.530000</td>
<td>1.5451</td>
</tr>
<tr>
<td>95% percentile of effluent total N (TN95)</td>
<td>g N.m⁻³</td>
<td>16.800000</td>
<td>16.756</td>
</tr>
<tr>
<td>95% percentile of effluent TSS (TSS95)</td>
<td>g COD.m⁻³</td>
<td>19.700000</td>
<td>19.7367</td>
</tr>
</tbody>
</table>

Maximum effluent total N limit (18 g N.m⁻³) was violated during:
- days: 4.258800
- % of total evaluation time: 1.170000
- number of violations: 32

Maximum effluent COD limit (100 g COD.m⁻³) was violated during:
- days: 0.208208
- % of total evaluation time: 0.057200
- number of violations: 3

Maximum effluent ammonia limit (4 g N.m⁻³) was violated during:
- days: 1.488760
- % of total evaluation time: 0.409000
- number of violations: 11

Maximum effluent TSS limit (30 g SS.m⁻³) was violated during:
- days: 1.259440
- % of total evaluation time: 0.346000
- number of violations: 11

Maximum effluent BOD₅ limit (10 g.m⁻³) was violated during:
- days: 0.458640
- % of total evaluation time: 0.126000
- number of violations: 6

### Controller performance

<table>
<thead>
<tr>
<th>Controller type</th>
<th>Unit</th>
<th>FORTRAN</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DO controller</strong></td>
<td></td>
<td>discrete PI, bias=120 1/d, $K=21.6$ m².g (–COD)⁻¹.d⁻¹, $T_i=0.00208$, $T_e=8.3e-4$ d</td>
<td>continuous PI with antiwindup, $K=25$ m².g (–COD)⁻¹.d⁻¹, $T_i=0.002$, $T_e=0.001$ d</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Setpoint SO₄</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Average of $e_{SO₄}$</td>
<td>mean($e$)</td>
<td>g (-COD).m⁻³</td>
<td>2.000000</td>
</tr>
<tr>
<td>Average of $</td>
<td>e_{SO₄}</td>
<td>$</td>
<td>mean (abs($e$))</td>
</tr>
<tr>
<td>IAE $e_{SO₄}$</td>
<td>integral of absolute error</td>
<td>g (-COD).m⁻³.d</td>
<td>10.953500</td>
</tr>
<tr>
<td>ISE $e_{SO₄}$</td>
<td>integral of square error max deviation from setpoint</td>
<td>(g (-COD)/m⁻³)².d</td>
<td>0.578263</td>
</tr>
<tr>
<td>Max $e_{SO₄}$</td>
<td>max deviation of setpoint</td>
<td>g (-COD).m⁻³</td>
<td>0.175859</td>
</tr>
<tr>
<td>Standard deviation of $e_{SO₄}$</td>
<td>std($e$)</td>
<td>g (-COD).m⁻³</td>
<td>0.039860</td>
</tr>
<tr>
<td>Variance of $e_{SO₄}$</td>
<td>var($e$)</td>
<td>(g (-COD).m⁻³)²</td>
<td>0.001589</td>
</tr>
<tr>
<td>Max deviation of $K_a$</td>
<td>max($K_a$)-min($K_a$)</td>
<td>d⁻¹</td>
<td>240.000000</td>
</tr>
<tr>
<td>Max deviation of $K_a$ in 1 sample</td>
<td>max(delta $K_a$)</td>
<td>d⁻¹</td>
<td>1.526198</td>
</tr>
<tr>
<td>Average value of $K_a$</td>
<td>mean($K_a$)</td>
<td>d⁻¹</td>
<td>63.614082</td>
</tr>
<tr>
<td>Standard deviation of $K_a$</td>
<td>std(delta $K_a$)</td>
<td>d⁻¹</td>
<td>0.226210</td>
</tr>
<tr>
<td>Variance of $K_a$</td>
<td>var(delta $K_a$)</td>
<td>(d⁻¹)²</td>
<td>0.051171</td>
</tr>
</tbody>
</table>
APPENDIX 5: Closed-loop results with realistic sensors and actuators

These results were obtained with MATLAB-Simulink (Solver: ode45, absolute tolerance =10⁻⁸, relative tolerance =10⁻⁵). For details on operation conditions see text. Details on PI controller settings are given in table headings.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effluent average flow rate</td>
<td>m³.d⁻¹</td>
<td>20661.0241</td>
</tr>
<tr>
<td>Effluent average flow rate</td>
<td>g COD.m⁻³</td>
<td>28.0618</td>
</tr>
<tr>
<td>Effluent average $S_I$ concentration</td>
<td>g COD.m⁻³</td>
<td>0.72634</td>
</tr>
<tr>
<td>Effluent average $S_S$ concentration</td>
<td>g COD.m⁻³</td>
<td>5.9139</td>
</tr>
<tr>
<td>Effluent average $X_I$ concentration</td>
<td>g COD.m⁻³</td>
<td>0.33132</td>
</tr>
<tr>
<td>Effluent average $X_S$ concentration</td>
<td>g COD.m⁻³</td>
<td>9.892</td>
</tr>
<tr>
<td>Effluent average $X_{BH}$ concentration</td>
<td>g COD.m⁻³</td>
<td>0.69096</td>
</tr>
<tr>
<td>Effluent average $X_{BA}$ concentration</td>
<td>g COD.m⁻³</td>
<td>3.4003</td>
</tr>
<tr>
<td>Effluent average $X_{P}$ concentration</td>
<td>g -COD.m⁻³</td>
<td>1.5798</td>
</tr>
<tr>
<td>Effluent average $S_{SO}$ concentration</td>
<td>g N.m⁻³</td>
<td>11.0455</td>
</tr>
<tr>
<td>Effluent average $S_{SN}$ concentration</td>
<td>g N.m⁻³</td>
<td>0.47429</td>
</tr>
<tr>
<td>Effluent average $S_{SO}$ concentration (limit = 4 g N.m⁻³)</td>
<td>g N.m⁻³</td>
<td>0.58502</td>
</tr>
<tr>
<td>Effluent average $S_{SN}$ concentration</td>
<td>g N.m⁻³</td>
<td>0.019079</td>
</tr>
<tr>
<td>Effluent average $X_{ALK}$ concentration</td>
<td>mole HCO₃⁻.m⁻³</td>
<td>4.4559</td>
</tr>
<tr>
<td>Effluent average Temperature</td>
<td>°C</td>
<td>15.1713</td>
</tr>
<tr>
<td>Effluent average Kjeldahl N concentration</td>
<td>g N.m⁻³</td>
<td>2.4839</td>
</tr>
<tr>
<td>Effluent average total N concentration (limit = 18 g N.m⁻³)</td>
<td>g N.m⁻³</td>
<td>13.5294</td>
</tr>
<tr>
<td>Effluent average total COD concentration (limit = 100 g COD.m⁻³)</td>
<td>g COD.m⁻³</td>
<td>49.0166</td>
</tr>
<tr>
<td>Effluent average BOD₅ concentration (limit = 10 g.m⁻³)</td>
<td>g.m⁻³</td>
<td>2.6985</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effluent average $S_I$ load</td>
<td>kg COD.d⁻¹</td>
<td>579.7859</td>
</tr>
<tr>
<td>Effluent average $S_S$ load</td>
<td>kg COD.d⁻¹</td>
<td>15.0069</td>
</tr>
<tr>
<td>Effluent average $X_I$ load</td>
<td>kg COD.d⁻¹</td>
<td>122.1863</td>
</tr>
<tr>
<td>Effluent average $X_S$ load</td>
<td>kg COD.d⁻¹</td>
<td>6.8453</td>
</tr>
<tr>
<td>Effluent average $X_{BH}$ load</td>
<td>kg COD.d⁻¹</td>
<td>204.3794</td>
</tr>
<tr>
<td>Effluent average $X_{BA}$ load</td>
<td>kg COD.d⁻¹</td>
<td>14.2759</td>
</tr>
<tr>
<td>Effluent average $X_{P}$ load</td>
<td>kg COD.d⁻¹</td>
<td>70.2528</td>
</tr>
<tr>
<td>Effluent average $S_{SO}$ load</td>
<td>kg N.d⁻¹</td>
<td>228.2123</td>
</tr>
<tr>
<td>Effluent average $S_{SN}$ load</td>
<td>kg N.d⁻¹</td>
<td>9.7993</td>
</tr>
<tr>
<td>Effluent average $X_{ALK}$ load</td>
<td>kg N.d⁻¹</td>
<td>12.0872</td>
</tr>
<tr>
<td>Effluent average $X_{ND}$ load</td>
<td>kg N.d⁻¹</td>
<td>0.39418</td>
</tr>
<tr>
<td>Effluent average $S_{ALK}$ load</td>
<td>kmol HCO₃⁻.d⁻¹</td>
<td>92.0639</td>
</tr>
<tr>
<td>Effluent average $TSS$ load</td>
<td>kg.d⁻¹</td>
<td>313.4547</td>
</tr>
<tr>
<td>Effluent average Kjeldahl N load</td>
<td>kg N.d⁻¹</td>
<td>51.3195</td>
</tr>
<tr>
<td>Effluent average total N load</td>
<td>kg N.d⁻¹</td>
<td>279.5318</td>
</tr>
<tr>
<td>Effluent average total COD load</td>
<td>kg COD.d⁻¹</td>
<td>1012.7325</td>
</tr>
<tr>
<td>Effluent average BOD₅ load</td>
<td>kg.d⁻¹</td>
<td>55.7538</td>
</tr>
</tbody>
</table>
**Other output quality variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Influent quality ((IQI)) index</td>
<td>kg poll.units.d(^{-1})</td>
<td>74783.3138</td>
</tr>
<tr>
<td>Effluent quality ((EQI)) index</td>
<td>kg poll.units.d(^{-1})</td>
<td>5572.8572</td>
</tr>
<tr>
<td>Average sludge production for disposal per day</td>
<td>kg SS.d(^{-1})</td>
<td>2707.769</td>
</tr>
<tr>
<td>Average sludge production released into effluent per day</td>
<td>kg SS.d(^{-1})</td>
<td>313.4547</td>
</tr>
<tr>
<td>Total average sludge production per day</td>
<td>kg SS.d(^{-1})</td>
<td>3021.2237</td>
</tr>
</tbody>
</table>

**'Energy' related variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average aeration energy</td>
<td>kWh.d(^{-1})</td>
<td>4225.4326</td>
</tr>
<tr>
<td>Average pumping energy</td>
<td>kWh.d(^{-1})</td>
<td>445.4525</td>
</tr>
<tr>
<td>Average carbon source dosage</td>
<td>kg COD.d(^{-1})</td>
<td>800</td>
</tr>
<tr>
<td>Average mixing energy</td>
<td>kWh.d(^{-1})</td>
<td>768</td>
</tr>
<tr>
<td>Average heating energy</td>
<td>kWh.d(^{-1})</td>
<td>4225.3434</td>
</tr>
<tr>
<td>Average methane gas production (1 kg = 13.8928 kWh)</td>
<td>kg CH(_4).d(^{-1})</td>
<td>1085.3599</td>
</tr>
<tr>
<td>Average hydrogen gas production</td>
<td>kg H(_2).d(^{-1})</td>
<td>0.0037088</td>
</tr>
<tr>
<td>Average carbon dioxide gas production</td>
<td>kg CO(_2).d(^{-1})</td>
<td>1562.6859</td>
</tr>
<tr>
<td>Average total gas flow rate from AD (normalized to (P_{\text{atm}}))</td>
<td>'normal' m(^3).d(^{-1})</td>
<td>2757.9546</td>
</tr>
</tbody>
</table>

**Operational cost index**

<table>
<thead>
<tr>
<th>Variable (including weight factor)</th>
<th>Unit</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sludge production cost index</td>
<td>–</td>
<td>8123.3069</td>
</tr>
<tr>
<td>Aeration energy cost index</td>
<td>–</td>
<td>4225.4326</td>
</tr>
<tr>
<td>Pumping energy cost index</td>
<td>–</td>
<td>445.4525</td>
</tr>
<tr>
<td>Carbon source dosage cost index</td>
<td>–</td>
<td>2400</td>
</tr>
<tr>
<td>Mixing energy cost index</td>
<td>–</td>
<td>768</td>
</tr>
<tr>
<td>Heating energy cost index</td>
<td>–</td>
<td>0</td>
</tr>
<tr>
<td>Net energy production from methane index (subtracted from rest)</td>
<td>–</td>
<td>6512.1596</td>
</tr>
<tr>
<td>Total Operational Cost Index ((OCI))</td>
<td>–</td>
<td>9450.0324</td>
</tr>
</tbody>
</table>
### Effluent violations

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>95% percentile of effluent ( S_{\text{NH}} ) (Ammonia95)</td>
<td>g N.m(^{-3})</td>
<td>1.5427</td>
</tr>
<tr>
<td>95% percentile of effluent total N (TN95)</td>
<td>g N.m(^{-3})</td>
<td>16.7525</td>
</tr>
<tr>
<td>95% percentile of effluent TSS (TSS95)</td>
<td>g COD.m(^{-3})</td>
<td>19.7367</td>
</tr>
</tbody>
</table>

**Maximum effluent total N limit (18 g N.m\(^{-3}\)) was violated**

- during: days | 4.2812
- % of total evaluation time: % | 1.1762
- number of violations: – | 32

**Maximum effluent total COD limit (100 g COD.m\(^{-3}\)) was violated**

- during: days | 0.20833
- % of total evaluation time: % | 0.057234
- number of violations: – | 3

**Maximum effluent ammonia limit (4 g N.m\(^{-3}\)) was violated**

- during: days | 1.4896
- % of total evaluation time: % | 0.40923
- number of violations: – | 11

**Maximum effluent TSS limit (30 g SS.m\(^{-3}\)) was violated**

- during: days | 1.25
- % of total evaluation time: % | 0.34341
- number of violations: – | 11

**Maximum effluent BOD\(_5\) limit (10 g.m\(^{-3}\)) was violated**

- during: days | 0.45833
- % of total evaluation time: % | 0.12592
- number of violations: – | 6

### Controller performance

<table>
<thead>
<tr>
<th>Qw controller</th>
<th>Unit</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller type</td>
<td>timer based ( Q_w ) actuator as first order filter used</td>
<td></td>
</tr>
<tr>
<td>Setpoint Qw</td>
<td>(m(^3).d(^{-1}))</td>
<td></td>
</tr>
<tr>
<td>Average of ( c_{Qw} )</td>
<td>mean(e)</td>
<td>m(^3).d(^{-1})</td>
</tr>
<tr>
<td>Average of (</td>
<td>c_{Qw}</td>
<td>)</td>
</tr>
<tr>
<td>IAE ( c_{Qw} )</td>
<td>integral of absolute error</td>
<td>m(^3)</td>
</tr>
<tr>
<td>ISE ( c_{Qw} )</td>
<td>integral of square error ( (m^3)^2.d )</td>
<td>383.78</td>
</tr>
<tr>
<td>Max ( c_{Qw} )</td>
<td>max deviation from setpoint</td>
<td>m(^3).d(^{-1})</td>
</tr>
<tr>
<td>Standard deviation of ( c_{Qw} )</td>
<td>std(e)</td>
<td>m(^3).d(^{-1})</td>
</tr>
<tr>
<td>Variance of ( c_{Qw} )</td>
<td>var(e)</td>
<td>(m(^3).d(^{-1}))(^2)</td>
</tr>
<tr>
<td>Max deviation of ( Q_w )</td>
<td>max((Q_{w}))-min((Q_{w}))</td>
<td>m(^3).d(^{-1})</td>
</tr>
<tr>
<td>Max deviation of ( Q_w ) in 1 sample</td>
<td>max(delta(Q_{w}))</td>
<td>m(^3).d(^{-1})</td>
</tr>
<tr>
<td>Average value of ( Q_w )</td>
<td>mean((Q_{w}))</td>
<td>m(^3).d(^{-1})</td>
</tr>
<tr>
<td>Standard deviation of delta(Q_{w})</td>
<td>std(delta(Q_{w}))</td>
<td>m(^3).d(^{-1})</td>
</tr>
<tr>
<td>Variance of delta(Q_{w})</td>
<td>var(delta(Q_{w}))</td>
<td>(m(^3).d(^{-1}))(^2)</td>
</tr>
</tbody>
</table>
### DO controller

**Controller type**

<table>
<thead>
<tr>
<th>Setpoint SO4</th>
<th>Unit</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average of $e_{SO4}$</td>
<td>$\text{mean}(e)$</td>
<td>g (-COD).m$^{-3}$</td>
</tr>
<tr>
<td>Average of</td>
<td>$\text{mean}(\text{abs}(e))$</td>
<td>g (-COD).m$^{-3}$</td>
</tr>
<tr>
<td>IAE $e_{SO4}$</td>
<td>integral of absolute error</td>
<td>g (-COD).m$^{-3}$.d</td>
</tr>
<tr>
<td>ISE $e_{SO4}$</td>
<td>integral of square error</td>
<td>$(g \cdot \text{(-COD).m}^{-3})^2$.d</td>
</tr>
<tr>
<td>Max $e_{SO4}$</td>
<td>setpoint</td>
<td>g (-COD).m$^{-3}$</td>
</tr>
<tr>
<td>Standard deviation of $e_{SO4}$</td>
<td>$\text{std}(e)$</td>
<td>g (-COD).m$^{-3}$</td>
</tr>
<tr>
<td>Variance of $e_{SO4}$</td>
<td>$\text{var}(e)$</td>
<td>$(g \cdot \text{(-COD).m}^{-3})^2$</td>
</tr>
<tr>
<td>Max deviation of $K_1a_4$</td>
<td>$\max(K_1a_4) - \min(K_1a_4)$</td>
<td>d$^{-1}$</td>
</tr>
<tr>
<td>Max deviation of $K_1a_4$ in 1 sample</td>
<td>$\max(\Delta K_1a_4)$</td>
<td>d$^{-1}$</td>
</tr>
<tr>
<td>Average value of $K_1a_4$</td>
<td>$\text{mean } K_1a_4$</td>
<td>d$^{-1}$</td>
</tr>
<tr>
<td>Standard deviation of $K_1a_4$</td>
<td>$\text{std}(\Delta K_1a_4)$</td>
<td>d$^{-1}$</td>
</tr>
<tr>
<td>Variance of $K_1a_4$</td>
<td>$\text{var}(\Delta K_1a_4)$</td>
<td>(d$^{-1}$)$^2$</td>
</tr>
</tbody>
</table>