**Matlab®/Simulink® implementation of ASM1, 2d and 3 module within BSM1**

Xavier Flores-Alsina1, Krist V. Gernaey1, Ulf Jeppsson2

1 Department of Chemical and Biochemical engineering, Technical University of Denmark., Kgs. Lyngby (Denmark)

2 Division of Industrial Electrical Engineering and Automation, Lund University, Lund (Sweden)

Note that this short document does not by any means provide a complete description of how to use the BSM1 system, however it may give some useful hints on the topic and avoid some unnecessary frustration. The document assumes that you are familiar with Matlab®/Simulink® and wastewater treatment process modelling and simulation. The provided models have been implemented in Matlab R2019 release and will also work with later versions.

## UNPACKING THE FILES

The files have been archived using zip. Just unzip the files with the software you normally use for this purpose.

## FILE DESCRIPTIONS

When the files are unpacked, you will find in the main directory 4 folders. The folder Documents con-tain three files: (1) Flores Alsina et al 2012…., (2) AGREEMENT\_ASM1, (3) README BSM1.pdf. There are also xls files with the simulation results. The other 3 folders named contains most of the files that are associated with ASM1, ASM2d and ASM3 implemented within BSM1.

## Simulink models

1. benchmark.slx (or .mdl) - simulate the plant without active control, i.e. in open loop, using dynamic input data

2. benchmark\_LT.slx (or .mdl) - simulate the plant without active control, i.e. in open loop, using full long term dynamic data

3. benchmarkss.slx (or .mdl) - simulate the plant without active control, i.e. in open loop, using constant input data

## C-files

1. asm1.c, asm2d.c and asm3.c - C-files containing Activated Sludge Model No.1, 2d and 3.

2. combiner\_XXX.c - adds two separate streams into one based on loads.

3. hyddelay\_XXX.c - a special delay function (a fast first-order exponential filter) to avoid alge-braic loops.

4. settler1d\_XXX.c - C-file for a 10-layer one-dimensional settler model.

These 4 files must be compiled on your local machine using the Matlab mex command before you can use the models (or use command mexall\_XXX).

## Initialisation files

The initialisation m-files have names associated with the model they influence:

1. asmXXXinit.m

2. settler1dinit\_XXX.m

All of these initialization files can be run using the benchmarkinit\_XXX.m script.

In that section you can select to simulate the process with electron dependency decay rates and reactive settler

## Input data

The various influent data files are provided in .mat format.

1. CONSTINFLUENT\_XXX.mat - the constant value influent file, which actually represents the average vales for one full year of dynamic data.

2. DRYINFLEUNT\_XXX.mat - the dynamic influent file which represents dry weather data for 14 days (15 min samples).

3. DYNINFLUENT\_XXX.mat - full dynamic influent data file for 609 days (15 min samples).

4. RAININFLUENT\_XXX.mat - the dynamic influent file which represents rain weather data for 14 days (15 min samples).

5. STORMINFLUENT\_XXX.mat - the dynamic influent file which represents storm weather data for 14 days (15 min samples).

## RUNNING THE BSM1

When the archive has been unzipped, you are ready to run the BSM1. A few simple instructions are given below to help you through the first time and to test the system on your computer.

1. Start Matlab and move to the directory containing the model you want to use

2. Command mexall\_XXX (if you have problems with the C-compiler you must solve this). If you change the C-files (which should normally not be done), you need to re-mex all the C-files that you have modified in order for Simulink to use the modified file.

3. Run the initialization file benchmarkinit (initiates all variables and parameters, loads the data files, etc.).

4. Command benchmarkss (the Simulink model will appear in a new window). The system will simulate 200 days forward using the constant influent data, the open loop configuration and solver ode15s.

5. Command stateset. The final values of the previous simulation have now been used to initialise the BSM1 model so that you can start your next simulation at the exact same position as where the last one ended. If you want to save all the generated data from a simulation you should simply use the Matlab save command.

If you want to run 14-days of dynamic simulation, follow step no. 6a; otherwise if you want to run the full 609-days of dynamic simulation, go to step no. 6b.

6. a. Command benchmark (the Simulink model will appear in a new window). The system will simulate 14 days forward using the dynamic influent data, the open loop configuration and solver ode45. You can choose between 3 different types of dynamic data: (i) dry weather influent data, (ii) rain weather influent data or (iii) storm weather influent data. You can select the influent data you want to simulate before running the model. Run this model twice, making sure to use the stateset command in between the simulations. Go to step no. 7.

b. Command benchmark\_LT (the Simulink model will appear in a new window). The system will simulate 609 days forward using the dynamic long term influent data, the open loop configuration and solver ode45.

7. After simulation all data are stored in the Matlab workspace and not to files. Use the who command to see what variables you have available.

8. When the simulation is finished: command perf\_plant\_dyn. The script will calculate and print to the screen the complete set of evaluation criteria for the overall plant performance for the simulation from day 7 to 14 (or command perf\_plant\_LT if you run the model using 609-days of dynamic data).

When you have reached this point you can be sure that the BSM1 with pH module on your computer works properly. Note that the long term dynamic simulation requires some time to finish.

## FINAL COMMENTS

Read the documentation about benchmarking from the web site carefully plus other available reports on BSM. Try to understand the structure of the different m-files and c-files to grasp how they relate to each other.

When you feel confident that you understand this implementation you may start to create your own in-fluent files for subsequent simulation studies. Remember that this implementation is simply a starting point and a fully verified simulation platform tor you to start working on your own and testing different scenarios that you’re interested in.

Enjoy!