NTNU – Trondheim
Norwegian University of
Science and Technology

Modelica Project Assignments

TTK4130 Modeling and Simulation

Wastewater treatment is a common form of pollution control consisting of collection sewers, pumping stations and treatment plants. The treatment plants are built to clean the wastewater to return the water into streams or for reuse. In the first stage of wastewater treatment solids are removed by sedimentation, while in the second stage biological processes are exploited to further purify the water.

This project aims to model the second stage in Dymola, which is a highly nonlinear and challenging process to operate. The simulation model used in this project is based on the report "Benchmark Simulation Model no. 1 (BSM1)" by J. Alex et al. (2008), which can be found under `http://apps.ensic.inpl-nancy.fr/benchmarkWWTP/Pdf/Description_BSM1_20080619.pdf`.

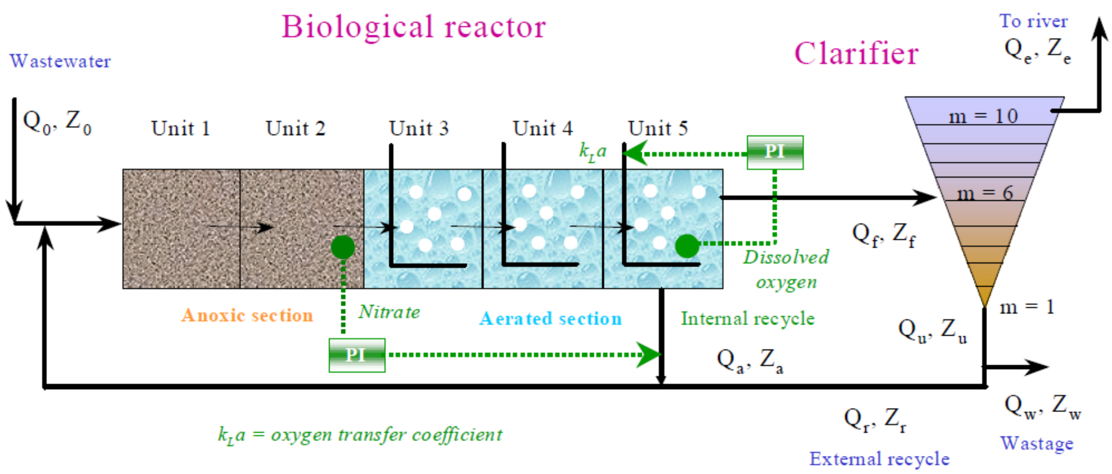

Figure 1: Wastewater treatment plant from "Benchmark Simulation Model no. 1 (BSM1)" by J. Alex et al. (2008)

Assignment 1 (Implementation of wastewater treatment plant simulation)

(a) Download the wastewater treatment package for Dymola by Gerald Reichl from:
`https://github.com/modelica-3rdparty/WasteWater/tree/master/WasteWater`

(b) Familiarise yourself with the wastewater treatment package by following the steps given in the readme file and run the benchmark example in ASM1.

- Once downloaded open "package.mo" in Dymola.
- Navigate to WasteWater/ASM1/examples and open the BenchPlant flowsheet.
- Go to the simulation tab, translate the model, run the script "*dymola_bench.mos*" found in the downloaded package to load the initial values of each unit and simulate it for 14 days.
- Once simulated it automatically saves a .mat file with the results. The results can be viewed in the "VariableBrowser" window for each unit in the flowsheet.

(c) Develop a simulation model for the BSM1 benchmark in Dymola with the help of the WasteWater treatment package using the *ASM1/examples/BenchPlant* for the open-loop case (no active controllers). See Figure 2 as an example of an open-loop Dymola Implementation. The following changes to the ASM1 benchmark need to be made for adjustment:

- Delete the feedback controllers
- Change the parameters in *ASM1/interfaces/ASM1base* and in *ASM1/interfaces/stoichiometry* to match those in Table 3 of the BSM1 report. Hint: Change the parameters in *ASM1/interfaces/ASM1base*

to parameters, since temperature is not a variable anymore. Hence, remove the exponential expressions with the corresponding pre-exponential factors.

- In the aeration tanks "nitri" change the aeration equation to match the one given in section 2.3.2 of the BSM1 report. Remove the now superfluous parameters defined in the nitri tank model.
- Change the volume and Kla values of the tanks as specified in section 2.3.1 of the BSM1 report. The Kla value for the open-loop case for tank 5 can be found in the text in section 4.
- Specify the pump flowrates as given in the plant description in section 2.1 in the BSM1 report. The pump flowrate of Q_a in the open-loop case can be found in the text in section 4 and the flow rate of Q_r is $18446 m^3 d^{-1}$ in the open-loop case.
- Delete the blowers since they are not required for the aeration equation anymore.
- Delete the temperature inputs since these are not required anymore either.

(d) Test your dynamic model of the plant on different process conditions using the same initial conditions as in the script file of ASM1 for the 14 day period.

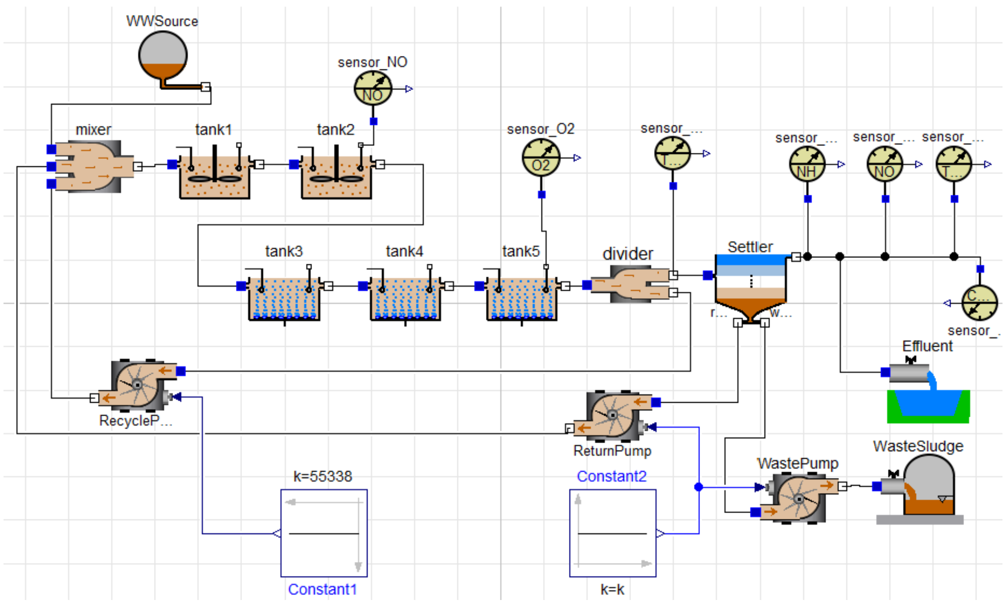

Figure 2: BSM1 Open-loop Dymola Implementation

Assignment 2 (Dynamic open-loop simulations)

Open-loop refers to the simulation of the plant without employing active controllers and disregarding process noise. This is generally a good first step to verify your simulation, since the simulation data is then always identical for the same conditions.

(a) Run your open-loop dynamic simulation for 100 days to reach steady-state based on the in-fluent data given in the introduction, Table 5 and the specifications in section 4 of the BSM1 report.

- Again use the script file of ASM1 to set the initial values of the different units. In theory the initial values can be set arbitrarily, since steady-state values are invariant of the initial conditions.
- Change the WWSource to constants, which are given in Table 5.
- Compare your simulation results with those in the file "Verificationdata.pdf" using the component names as shown in Figure 2.

- Rename the .mat file generated to ensure it is not overwritten. This file can be used in the future to initialize your simulation.

(b) Initialize your next simulation using the previous data and run it for 14 days using the influent data contained in the file "$Inf\_dry.txt$" for ASM1, as is the case in the Benchplant in ASM1.

- To initialize your simulation from previous data follow the following steps: Translate your model, at the top under simulation select *conintue/importinitial...* and select your previous .mat file. Select 100 to use the data after 100 days for initialization and start the next simulation at 0.
- Run the simulation. It is not required anymore to run the script file, since the simulation now uses the initial values of the .mat file.
- This exercise highlights a good way to obtain valid initialization values, which are often difficult to set otherwise.

(c) In a separate file reformulate the equation system in tank5 to automatically change Kla to give an output concentration of dissolved oxygen "SO" of exactly $2g(COD).m^{-3}$. Explain the observations on Kla.

- Duplicate your simulation model
- Change Kla from parameter to variable in tank5. The model now has 1 extra degree of freedom, hence we have to add an extra equation to compensate.
- Add an equation constraining "SO" to be equal to $2g(COD).m^{-3}$ in tank5.
- Simulate the plant. You should now be able to plot for tank5 the variation of Kla against time.

(d) To gauge the performance of the wastewater treatment plant several performance indices have been proposed in literature. Implement the following performance indices given in section 6 to be calculated by Dymola: EQ, PE, AE, IQ and SP to be calculated over the full time period from the first day to the 14th day (hint: integral definitions can be reformulated as differential equations with zero initial conditions). Here is an example on how to create the quality indicator for AE:

- Change the measurement ports of tank3, tank4 and tank5 to output the Kla value using "Modelica.Blocks.Interfaces.RealOutput". See the sensors of ASM1 as an example on how to do this.
- Duplicate a sensor block and change the equations to the equations as shown in Figure 3 for AE.
- Connect the tanks output of Kla to this new block
- Simulate your model. In the results you should be able to now find the "$Aeration\_energy$" unit under which the AE is calculated over time.
- For the other quality indicators follow very similar steps. Examples on how to extract other variable information can be found in the sensor blocks in ASM1.
- For SP ignore "TSS", since these are only relevant if you start calculation after 7 days.

```
model Aeration_energy

  extends WasteWater.Icons.sensor_O2;
  Modelica.Blocks.Interfaces.RealInput Kla3
      a;
  Modelica.Blocks.Interfaces.RealInput Kla4
      a;
  Modelica.Blocks.Interfaces.RealInput Kla5
      a;
  Modelica.Blocks.Interfaces.RealOutput AE(start=0) a;

  Real T(start=1e-3);

equation
  der(T)  =  1.0;
  der(AE*T) =  2/1.8/1000*1333*(Kla3 + Kla4 + Kla5);

end Aeration energy;
```

Figure 3: Example quality indicator for aeration energy

Assignment 3 (Open-loop sensitivity analysis)

In Assignment 3 we employ the simulation set-up developed in Assignments 1 and 2 to investigate the effects of different key process parameters on the efficiency of the wastewater treatment plant. To carry-out this analysis please use a constant Kla value for tank5 for the nominal case as given in section 4 of the BSM1 report. For the influent data utilise the file ″$Inf\_dry.txt$″ for ASM1 initialised at steady-state, which may be different for each variation of the process parameters.

Gauge the effect of the following parameters on the plant using the performance indices developed previously, explain your observations and state the advantages and disadvantages:

(a) Implement the remaining ME quality indicator over the full 14 days and combine all quality indicators for the overall cost indicator (OCI).

- For ME use the ″if″$else$ expression in Modelica

(b) Volumetric flow-rates of the pumps

(c) Kla values of the aeration tanks

(d) Y_A, Y_H and SO_{sat} (for these cases ignore advantages and disadvantages)

Assignment 4 (Multiple-input, Multiple-output control system design pairing)

Systems with more than one manipulated variable and controlled variable are referred to as ″Multiple-Input, Multiple-Output″ systems, often abbreviated as ″MIMO″. For nearly all important processes there are two controlled variables, e.g. product throughput and quality. An added complexity of MIMO systems is the fact that process interactions are present, i.e. each manipulated variable may affect both controlled variables. In the presence of significant interactions the choice of the most suitable manipulated variables is not obvious.

(a) Determine the steady-state gain matrix by incurring small perturbations on the manipulated variables and calculating the changes in the controlled variable. The steady-state gain matrix can be used to evaluate the interactions of the different manipulated variables on the controlled variables.

- Potential manipulated variables: Kla of tanks 3, 4 and 5 (tank3.Kla, tank4.Kla, tank5.Kla), flowrates of the outer recycle loop (ReturnPump.Q) and inner recycle loop (RecyclePump.Q), see Figure 2.
- Controlled variables: amount of nitrate after tank 2 (tank2.SNo) and amount of dissolved oxygen after tank 5 (tank5.So)
- Use the constant influent-data to set-up your nominal simulation and set the parameter values to the open-loop case as in Assignment 2 a), e.g. tank5.Kla = 84.0
- Record the nominal steady-state values of the controlled variables (tank2.SNo,tank5.So) after running the simulation for 100 days
- Carry-out small perturbations on the manipulated variables (not too small though, since otherwise you run into issues with the accuracy of the solver). Record the steady-state values of the controlled variables.
- Here an example on how you might progress on tank3.Kla:
  - Increase tank3.Kla from 240 to 242 (try to make a small change, but not too small) and record the change made
  - Set the tolerance of the solver to 1e-12 to ensure that the recorded difference is not in the range of the numerical error
  - Run the simulation for 100 days to reach the new steady-state
  - Record the controlled variables (tank2.SNo,tank5.So) after 100 days
  - Return the value of tank3.Kla to 240
- Repeat this procedure for all potential manipulated variables. You should then have 5x2 values of the controlled variables.
- Establish the gain matrix using the previously determined values: $K = \begin{bmatrix} \frac{\Delta y_1}{\Delta u_1} & \frac{\Delta y_2}{\Delta u_1} \\ \vdots & \vdots \\ \frac{\Delta y_1}{\Delta u_5} & \frac{\Delta y_2}{\Delta u_5} \end{bmatrix}$
  where Δy_i is the change of the ith controlled variable from its nominal steady-state value and Δu_i the incurred change of the ith manipulated variable from its nominal value. For example if we increase tank3.Kla from 240 to 242 then $\Delta u_1 = 242 - 240 = 2$. If we now recorded a change of $tank2.SNo$ from 0.7 to 0.6, then $\Delta y_1 = 0.6 - 0.7 = -0.1$, and the first element in the matrix is given by $\frac{-0.1}{2} = -0.05$.

(b) Use Bristols relative gain array method (RGA) to systematically choose the manipulated variables to control the nitrate content after tank2 and the oxygen content after tank5. RGA is one of many methods to make this decision.

- Generally for two controlled variables we want to choose two manipulated variables to obtain a well-behaved square control system. Therefore, divide the gain matrix K into 10 possible pairings (5 choose 2) of 2x2 control systems, i.e. eliminate all but two rows of matrix K and create in this fashion 10 unique sub-matrices corresponding to two inputs and two outputs.
- We will refer to these square matrices as:
  $K_{i,j} = \begin{bmatrix} \frac{\Delta y_1}{\Delta u_i} & \frac{\Delta y_2}{\Delta u_i} \\ \frac{\Delta y_1}{\Delta u_j} & \frac{\Delta y_2}{\Delta u_j} \end{bmatrix}$
  where $K_{i,j}$ is a square matrix for the ith and jth control input.
- The RGA can then be calculated using the following formula for each pairing:

$$RGA_{i,j} = K_{i,j} * K_{i,j}^{-T} = \begin{bmatrix} \lambda & 1-\lambda \\ 1-\lambda & \lambda \end{bmatrix} \tag{1}$$

where $RGA_{i,j}$ is the RGA for the ith and jth control input pairing and $*$ refers to **elementwise** multiplication. If calculated properly the matrix should have a form as shown on the right-hand side.

- The RGA can be used to gauge the goodness of the control pairings as follows:
  - Interactions are small if relative-gains are close to 1, therefore choose pairings corresponding to RGA elements close to 1.
  - Avoid pairings with negative RGA elements.
  - Large RGA elements correspond to processes that are very sensitive to small changes and hence should be avoided.
  - Example:

    $RGA_{1,2} = \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix}$

    Here the first input (row 1) should control the second output (column 2) since 0.8 is considerably closer to 1 than 0.2, while the second input (row 2) should control the first output (column 1). This still leads to a control system that has significant interactions. Nonetheless the RGA elements are relatively small and hence the control system is likely to be robust. We also see that we can use the closeness of λ to 1 as selection criteria.
- Based on the different RGAs choose the best input-output pairing and explain your choice. Also comment on which input-output pairing would not work without modifications.