Out: April 10, 2018
Recommended completion: April 27, 2018

**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`.

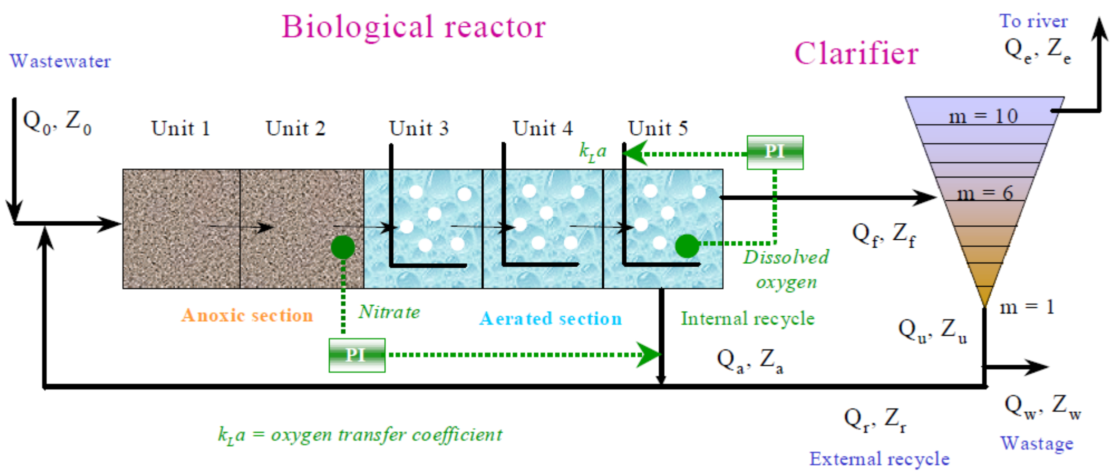


Figure 1: Wastewater treatment plant from ”Benchmark Simulation Model no. 1 (BSM1)” by J. Alex et al. (2008)

**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.

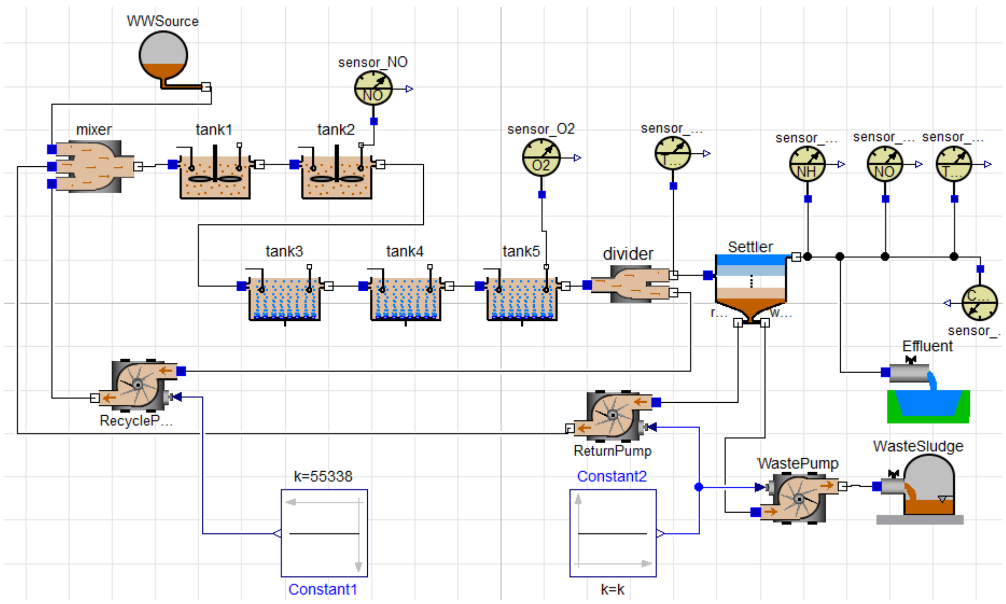


Figure 2: BSM1 Open-loop Dymola Implementation

**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.

**Assignment 3 (Performance indicators I)**

To gauge the performance of the wastewater treatment plant several performance indices have been proposed in literature.

Implement the following performance indicators given in section 6 to be calculated by Dymola: EQ, PE, AE and IQ 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.

```
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 4 (Performance indicators II)**

Implement the performance indicators ME and SP. Together with the previous assignment it is now possible to calculate the overall cost indicator ($OCI$).

- Implement ME and SP following the steps as in Assignment 3 for AE.
- For SP ignore "TSS", since these are only relevant if you start calculation after 7 days.
- For ME use the "$if$"$else$ expression in Modelica.
- Ensure that your performance indicators give reasonable values by comparing to the values given in the Table on page 35 of the BSM1 report, e.g. for SP the units of the biomass concentrations ($X$s) need to be adjusted by dividing by a factor of 1000.
- Combine all quality indicators for the overall cost indicator ($OCI$).

**Assignment 5 (Open-loop sensitivity analysis I)**

In Assignment 5 we employ the simulation set-up developed in Assignments 1, 2, 3 and 4 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.

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) Volumetric flow-rates of the pumps

(b) $Y_A$ and $Y_H$

**Assignment 6 (Open-loop sensitivity analysis II)**

In Assignment 6 we employ the simulation set-up developed in Assignments 1, 2, 3 and 4 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.

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) Kla values of the aeration tanks

(b) $SO_{sat}$

**Assignment 7 (Multiple-input, Multiple-output control system design pairing I)**

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.

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 $\Delta y_i$ is the change of the ith controlled variable from its nominal steady-state value and $\Delta u_i$ 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 *tank*2.*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 $\frac{-0.1}{2} = -0.05$.

**Assignment 8 (Multiple-input, Multiple-output control system design pairing II)**
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. For this you need to steady-state gain matrix ($K$) of assignment 7.

- 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. An example matlab script is shown in Figure 4 to obtain the 10 RGAs, where $K$ is the previously established steady-state gain matrix from assignment 7.
- 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 $\lambda$ to 1 or 0 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.

```matlab
load('K.mat')
n   = size(K,1); % number of manipulated variables
k   = 0;
% tank3.Kla - tank4.Kla, tank5.Kla, Recyclepump.Q, Returnpump.Q
% tank4.Kla - tank5.Kla, Recyclepump.Q, Returnpump.Q
% tank5.Kla - Recyclepump.Q, Returnpump.Q
% Recyclepump.Q - Returnpump.Q
for i = 1:n
    for j = i+1:n
        k = k+1;
        RGA{k}     = [K(i,:);K(j,:)];
        RGA{k}     = RGA{k}.*transpose(inv(RGA{k}));
        RGAa       = RGA{k};
        lambda(k)  = abs(RGAa(1,1));
    end
end
```

Figure 4: Matlab script to calculate RGAs from steady-state gain matrix ($K$)

**Assignment 9 (Perfect settler modelling and implementation)**

Sedimentation (settling) in wastewater treatment describes the separation of suspended solid particles, which are heavier than water. Sedimentation depends on the differences in density between the solid particles and water. In our plant the settler is used to remove sludge produced in the bioreactors for recycling and disposal.

The settler is assumed to be designed such that all particles are removed, i.e. the surface area is sufficiently large that all particles can settle, which will be determined from Modelica in the next assignment. This means that the effluent stream contains no particles/biomass (variables $X$ in the flowsheet). Let $\mathcal{S} = \{S_I, S_S, S_O, S_{NO}, S_{NH}, S_{ND}, S_{ALK}\}$ be a set representing the different substrates and $\mathcal{X} = \{X_i, X_s, X_{bh}, X_{ba}, X_p, X_{nd}\}$ be the various biomass components and lastly let $\mathcal{F} = \{\text{Feed, Effluent, Recycle, Waste}\}$ refer to the different streams entering and leaving the settler.

*Settler equations based on assumption of perfect separation*

The equations are based on simple mass balances on a settler as given in Figure 5 with 1 feed stream and 3 output streams.

1. Firstly we know that all components entering the settler have to also leave the settler, since no chemical reaction or accumulation is taking place. Amount of component leaving or entering is given by the volumetric flowrate times the concentration of the respective component. This can be expressed as follows:
$$\sum_{j\in\mathcal{F}} Q_j C_i^j = 0 \quad \forall i \in \mathcal{S} \cup \mathcal{X} \tag{2}$$
where $Q_j$ are the flowrates of the various streams which by convention is negative for output flowrates and $C_i^j$ is the concentration of component $i$ in stream $j$.

2. The effluent stream contains no biomass, since we assume perfect separation. This leads to the following:
$$C_i^{\text{Effluent}} = 0 \quad \forall i \in \mathcal{X} \tag{3}$$

3. The volumetric flowrates entering the system have to equal the volumetric flowrates leaving the system, since we assume no accumulation to take place:

$$\sum_{j \in \mathcal{F}} Q_j = 0 \tag{4}$$

4. Waste stream and recycle stream have exactly the same compositions (since they are split from the bottom of the settler):

$$C_i^{\text{Recycle}} = C_i^{\text{Waste}} \quad \forall i \in \mathcal{S} \cup \mathcal{X} \tag{5}$$

5. The substrate concentration of the effluent stream is the same as that of the waste/recycle stream, since the substrate is not separated:

$$C_i^{\text{Effluent}} = C_i^{\text{Waste}} \quad \forall i \in \mathcal{S} \tag{6}$$

These are all the required equations needed to model the settler. Note that the concentrations and flowrate of the feed are known from the process. Further, the recycle and waste flowrates need to also be specified by the pump flowrates of the flowsheet, see Figure 2 for clarification. We can in addition carry-out a degrees of freedom analysis to see if the system is fully specified:

| Equation no. | New unknowns | No. of new unknowns | Number of equations |
|---|---|---|---|
| (2) | $C_i^j \quad \forall j \in \mathcal{F}/$ Feed, $Q_{\text{Effluent}}$ | 40 | 13 |
| (3) | - | 0 | 6 |
| (4) | - | 0 | 1 |
| (5) | - | 0 | 13 |
| (6) | - | 0 | 7 |
| | | 40 | 40 |

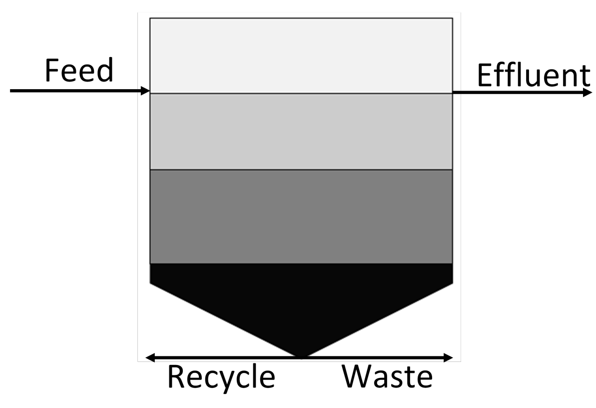


Figure 5: Settler input and output streams

*Settler implementation in Modelica*

- Right-click on your package directory and select New/Model...
- Name the model and write a small description.
- Go to the Modelica text view to start implementing the model.

- First we need to define the inputs and outputs. For a settler we have a single input and three outputs. Make sure to give each output a unique name. One output is utilised as recycle, the other is wastage and the last one as effluent. The following lines define an input and output with names "Input" and "Output1" respectively:
  - WasteWater.ASM1.Interfaces.WWFlowAsm1in Input annotation (Placement(transformation(extent={{-100,40},{-80,60}}),iconTransformation(extent={{-100,40},{-80,60}})));
  - WasteWater.ASM1.Interfaces.WWFlowAsm1out Output1 annotation (Placement(transformation(extent={{-100,40},{-80,60}}),iconTransformation(extent={{-100,40},{-80,60}})));

  It should be pointed out that the placement of the input/output can be easily changed in the icon menu rather than using the coordinates directly.
- Next create your own icon for the model by selecting the icon menu at the top (blue square). On the top left you are given the possibility to create simple icons from lines/rectangles/ellipses and polygons. The line and fill color is also editable. The input/output box representation can be moved as required. See Figure 6 for an example.

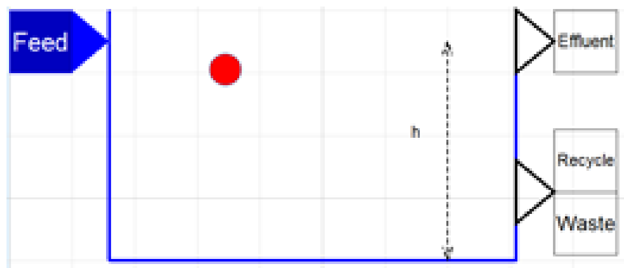


Figure 6: Icon example for settler

- Return to the Modelica text view. Next we would need to define the variables not covered by the inputs and outputs, but in this case however all the variables required are already defined.
- Type (if not already there) equation after the input/output definitions. Model the settler by implementing equations (2)-(6) in Modelica. See Figure 7 as an example on how to implement equation (2) in Modelica. See "Settler template.txt" to help you with the implementation, in which the equations (2), (5) and (6) are given, while (3) and (4) are missing.
- Add the settler to your flowsheet by replacing "Settler" in the original flowsheet for the case of the "infdry" input. Connect the streams as required, which should be easy if you named them consistently.
- Verify the new settler on the "infdry" input and run it for the full 14 days. Does it behave as you would expect? Use the script "settler_initial.mos" for initialization, which can be found on Blackboard in the assignments folder.
- Run the settler for the constant input and note down the new steady-state values of the variables in the verification data after 100 days.

```
model Settler "Ideal basin settler with variable height"

  WasteWater.ASM1.Interfaces.WWFlowAsm1in  Feed
  a;

  WasteWater.ASM1.Interfaces.WWFlowAsm1out Recycle
  a;

  WasteWater.ASM1.Interfaces.WWFlowAsm1out Waste
  a;

  WasteWater.ASM1.Interfaces.WWFlowAsm1out Effluent
  a;

  // Amount of solutes entering = amount of solutes leaving
  Feed.Si*Feed.Q + Waste.Si*Waste.Q  + Effluent.Si*Effluent.Q  + Recycle.Si*Recycle.Q = 0;
  Feed.Ss*Feed.Q + Waste.Ss*Waste.Q  + Effluent.Ss*Effluent.Q  + Recycle.Ss*Recycle.Q = 0;
  Feed.Xi*Feed.Q + Waste.Xi*Waste.Q  + Effluent.Xi*Effluent.Q  + Recycle.Xi*Recycle.Q = 0;
  Feed.Xs*Feed.Q + Waste.Xs*Waste.Q  + Effluent.Xs*Effluent.Q  + Recycle.Xs*Recycle.Q = 0;
  Feed.Xbh*Feed.Q + Waste.Xbh*Waste.Q  + Effluent.Xbh*Effluent.Q  + Recycle.Xbh*Recycle.Q = 0;
  Feed.Xba*Feed.Q + Waste.Xba*Waste.Q  + Effluent.Xba*Effluent.Q  + Recycle.Xba*Recycle.Q = 0;
  Feed.Xp*Feed.Q  + Waste.Xp*Waste.Q  + Effluent.Xp*Effluent.Q  + Recycle.Xp*Recycle.Q = 0;
  Feed.So*Feed.Q  + Waste.So*Waste.Q  + Effluent.So*Effluent.Q  + Recycle.So*Recycle.Q = 0;
  Feed.Sno*Feed.Q + Waste.Sno*Waste.Q  + Effluent.Sno*Effluent.Q  + Recycle.Sno*Recycle.Q = 0;
  Feed.Snh*Feed.Q + Waste.Snh*Waste.Q  + Effluent.Snh*Effluent.Q  + Recycle.Snh*Recycle.Q = 0;
  Feed.Snd*Feed.Q + Waste.Snd*Waste.Q  + Effluent.Snd*Effluent.Q  + Recycle.Snd*Recycle.Q = 0;
  Feed.Xnd*Feed.Q + Waste.Xnd*Waste.Q  + Effluent.Xnd*Effluent.Q  + Recycle.Xnd*Recycle.Q = 0;
  Feed.Salk*Feed.Q + Waste.Salk*Waste.Q  + Effluent.Salk*Effluent.Q  + Recycle.Salk*Recycle.Q = 0;

  a
end Settler;
```

Figure 7: Settler example equations (2)

**Assignment 10 (Settler design)**

Based on the previous section we now want to determine a settler that can approach perfect separation by using Modelica. In this assignment we use the simplest model for the settler based on discrete particle settling, which is stated below. For more information please go to `https://www.it.uu.se/research/project/jass/material/sett98.pdf`.

*Force balance*

In the discrete particle settling model we assume the concentration of solids to be low enough, such that each particle can be treated individually. The model is then derived from a straight-forward force balance, see Figure 8.

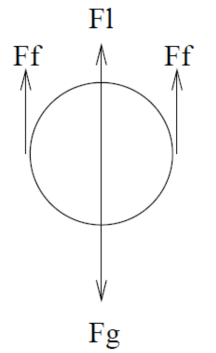


Figure 8: Forces acting on a single particle: $F_g$ gravitational force, $F_l$ lift force, $F_f$ frictional force

Using Newton's second law the force balance can be expressed as follows:

$$m\frac{dv}{dt} = F_g - F_l - F_f \tag{7}$$

where $v$ is the velocity of the particle.

The various forces can be expressed as:

$$F_g = mg = \rho_p V_p g \tag{8}$$
$$F_l = \rho_f V_p g \tag{9}$$
$$F_f = \frac{C_D A_p \rho_f v^2}{2} \tag{10}$$

where $\rho_p$ is the density of the particle, $V_p$ is the volume of the particle, $g$ the gravitational constant, $\rho_l$ is the fluid density, $C_D$ is the drag coefficient, $A_p$ the projected area perpendicular to the velocity.

Assuming steady state ($\frac{dv}{dt} = 0$) and substituting the expressions from (8)-(10) into (7), we arrive at the following expression for the particle velocity:

$$v = \sqrt{\frac{2g(\rho_p - \rho_f)V_p}{C_D A_p \rho_f}} \tag{11}$$

Assuming the particle to be spherical $V_p$ and $A_p$ are given by:

$$V_p = \frac{\pi d^3}{6}, \quad A_p = \pi d^2 \tag{12}$$

where $d$ is the diameter of the particle.

For laminar flow it can be shown that the drag coefficient can be determined as follows:

$$C_D = \frac{24}{Re}, \quad Re = \frac{v d \rho_f}{\mu} \tag{13}$$

where $\mu$ is the viscosity of the fluid.

Finally we arrive at Stokes law for the velocity of the particle substituting (12) and (13) into (11):

$$v = \frac{g(\rho_p - \rho_f)d^2}{18\mu} \tag{14}$$


Figure 9: Ideal settling basin with a height of $h$ and inflow rate $Q$

*Settler design*
The velocity of the particle in (14) can be used to design the settler. If we assume our settler to be approximately given as shown in Figure 9, then the length of the settler is set to allow for enough time for the particle to travel $h$ distance. The time a unit element resides in a basin is given by:

$$T = \frac{V}{Q} = \frac{Ah}{Q} \tag{15}$$

where $V$, $A$ and $h$ are the basin volume, surface area and height respectively.

The surface area $A$ and height $h$ are design parameters, whereas the inflow rate $Q$ is given by the process. To allow for enough time for the particle to settle the time $T$ has to exceed the required minimum time for the particle to reach the bottom of the basin according to (15):

$$T \geq h/v = h/\left(\frac{g(\rho_p - \rho_f)d^2}{18\mu}\right) \tag{16}$$

The aim of this assignment is to determine a sensible value of the surface area to allow for perfect separation using equations (15) and (16).

*Surface area from Modelica*
We assume that the surface area $A$ is variable to determine a sensible value for it from Modelica by finding the minimum required. The minimum is found by setting (16) as equality. Equations (15) and (16) need to be added to the previous settler model, which can be carried-out as follows:

- Right-click on your package directory and select New/Model...
- Name the model and write a small description.
- Go to the Modelica text view to start implementing the model.
- Type at the top ”extends Settler;”, assuming you named your settler ”Settler”. This immediately gives you all the equations and variable definitions of the previous model, so that you can ”extend” it.
- Next we need to define the variables that have not yet been defined below the extends command, which are as follows:
  - The variable surface area of the settler: Real A(start=10.0,fixed=false);
  - The time spent in the settler: Real T(start=1000.0,fixed=false);
  - The velocity of the particle: Real v(start=1000.0,fixed=false);
  - The density of the particle: Real rhop = 1200.0;
  - The density of the liquid: Real rhof = 1000.0;
  - The diameter of the particle: Real d = 0.0005;

  - The viscosity of the liquid: Real mu = 0.001;
  - Height of settler: Real h = 4.0;
  - Gravitational constant: Real g = 9.81;

  Note that the real variables have a start value even though they are not differential variables. This is to provide the nonlinear solver an initial guess to avoid division by zero errors.

- Next start the equation section of your model and define equations (15) and (16) (where (16) is used as equality, since we are interested in the minimum required)
- Go to the icon view and copy-over the icon you created previously.
- Next in your flowsheet you now want to use your new model. For this right-click on your current settler, select change class and select your new model. This replaces the current implemented model with the one you have now created.
- Run your model on the ”infdry” input and observe the surface area required. Does the surface area vary as expected?
- Note down approximately the maximum surface area and add 60% margin for safety. Assuming a width of 4m, what is the required length of your settler for perfect separation?