ASModel 1.0
A Scilab/Scicos toolbox for activated sludge modelling and wastewater treatment plant simulation

Fabrice Béline

August 2004

Contents

Chapter 1

Introduction

ASModel is a toolbox for modelling of activated sludge. It could be used to simulate wastewater treatment plant. At the moment two models are implemented:

1. The well known Activated Sludge Model number 1 (ASM1)
2. The Piggery Wastewater Treatment number 1 (PiWaT1) developed by Cemagref

These models are implemented with intermittent aeration process. The block diagrams included in this toolbox allow to build easily a treatment plant, to define all characteristics and finally simulate the plant. Others block diagrams included in Scicos could be used in order to save results or to draw dynamic graphics.

Only few block diagrams are developed (7) but sources are available and could be used for more development. All contributions could be sent at *asmodel@cemagref.fr* and would be integrated in the next version with the name of the contributors.

If you have some comments or questions, you can contact the authors at *asmodel@cemagref.fr*.

Remarks

The ASModel palette (ASModel.cos) and examples (Demo1.cos, Demo2.cos, Demo3.cos, Demo4.cos) were built using scilab 3.0 and can't be used with previous version. Anyway, this toolbox could work with previous version but ASModel palette and examples need to be rebuilt.

Chapter 2

Description of the models

2.1 ASM1: Activated Sludge Model number 1

The ASM1 model was initially developed by the IWA task group on mathematical modelling for design and operation of biological wastewater treatment. The model incorporates phenomena such as carbon oxidation, nitrification and denitrification. The model distinguihes 12 components:

1. Soluble inert COD
2. Readily biodegradable COD
3. Particulate inert COD
4. Slowly Biodegradable COD
5. Active heterotrophic biomass
6. Active autotrophic biomass
7. Particulate inert COD from biomass
8. Dissolved oxygen
9. Nitrate+nitrite
10. Ammonium+ammonia
11. Soluble organic nitrogen
12. Particulate organic nitrogen

and 8 biological processes:

1. Aerobic growth of heterotrophs
2. Anoxic growth of heterotrophs
3. Aerobic growth of autotrophs
4. Decay of heterotrophs
5. Decay of autotrophs
6. Ammonification of soluble organic nitrogen
7. Hydrolysis of entrapped organics
8. Hydrolysis of entrapped organic nitrogen

Kinetic and stoichiometry of all processes are presented on table 2.1 and 2.2. Default values for this model are proposed on table 2.3.

Table 2.1: Process kinetics for organic matter oxidation, nitrification and denitrification for model ASM1

j	Process rate	Definition
ρ_1	$\mu_{\mathrm{H}}\left(\frac{\mathrm{S_S}}{\mathrm{S_S+K_S}}\right)\left(\frac{\mathrm{S_O}}{\mathrm{S_O+K_{OH}}}\right)\mathrm{X_{BH}}$	Aerobic growth of heterotrophs
ρ_2	$\mu_{\mathrm{H}}\eta_{\mathrm{G}}\left(\frac{\mathrm{S_S}}{\mathrm{S_S+K_S}}\right)\left(\frac{\mathrm{K_{OH}}}{\mathrm{S_O+K_{OH}}}\right)\left(\frac{\mathrm{S_{NO}}}{\mathrm{S_{NO}+K_{NO}}}\right)\mathrm{X_{BH}}$	Anoxic growth of heterotrophs
ρ_3	$\mu_{\mathrm{A}}\left(\frac{\mathrm{S_{NH}}}{\mathrm{S_{NH}+K_{NH}}}\right)\left(\frac{\mathrm{S_O}}{\mathrm{S_O+K_{OA}}}\right)\mathrm{X_{BA}}$	Aerobic growth of autotrophs
ρ_4	$\mathrm{b_H}\cdot\mathrm{X_{BH}}$	Decay of heterotrophs
ρ_5	$\mathrm{b_A}\cdot\mathrm{X_{BA}}$	Decay of autotrophs
ρ_6	$\mathrm{k_A}\cdot\mathrm{S_{ND}}\cdot\mathrm{X_{BH}}$	Ammonification of soluble organic nitrogen
ρ_7	$\mathrm{k_H}(\frac{\mathrm{X_{BH}/X_S}}{\mathrm{K_X+X_{BH}/X_S}})[(\frac{\mathrm{S_O}}{\mathrm{S_O+K_{OH}}})+\eta_{\mathrm{H}}(\frac{\mathrm{K_{OH}}}{\mathrm{S_O+K_{OH}}})(\frac{\mathrm{S_{NO}}}{\mathrm{S_{NO}+K_{NO}}})]\mathrm{X_{BH}}$	Hydrolysis of entrapped organics
ρ_8	$\rho_7\left(\frac{\mathrm{X_{ND}}}{\mathrm{X_S}}\right)$	Hydrolysis of entrapped organic nitrogen

Table 2.2: Process stoichiometry for organic matter oxidation, nitrification and denitrification for model ASM1

	1 $\mathrm{S_I}$	2 $\mathrm{S_S}$	3 $\mathrm{X_I}$	4 $\mathrm{X_S}$	5 $\mathrm{X_{BH}}$	6 $\mathrm{X_{BA}}$	7 $\mathrm{X_P}$	8 $\mathrm{S_O}$	9 $\mathrm{S_{NO}}$	10 $\mathrm{S_{NH}}$	11 $\mathrm{S_{ND}}$	12 $\mathrm{X_{ND}}$
ρ_1		$\frac{-1}{\mathrm{Y_H}}$			1			$\frac{-(1-\mathrm{Y_H})}{\mathrm{Y_H}}$		$-\mathrm{i_{XB}}$		
ρ_2		$\frac{-1}{\mathrm{Y_H}}$			1				$-\frac{1-\mathrm{Y_H}}{2.86\mathrm{Y_H}}$	$-\mathrm{i_{XB}}$		
ρ_3						1		$-\frac{4.57-\mathrm{Y_A}}{\mathrm{Y_A}}$	$\frac{1}{\mathrm{Y_A}}$	$-\mathrm{i_{XB}}-\frac{1}{\mathrm{Y_A}}$		
ρ_4				$1-\mathrm{f_P}$	-1		$\mathrm{f_P}$					$\mathrm{i_{XB}}-\mathrm{f_P i_{XP}}$
ρ_5				$1-\mathrm{f_P}$		-1	$\mathrm{f_P}$					$\mathrm{i_{XB}}-\mathrm{f_P i_{XP}}$
ρ_6										1	-1	
ρ_7		1		-1								
ρ_8											1	-1

Table 2.3: Default values at 20°C for the model ASM1

Symbol	Definition	Default Value
$\mathrm{Y_H}$	Aerobic yield for heterotrophic biomass ($\mathrm{gCOD.gCOD^{-1}}$)	0.67
μ_H	Maximum specific growth rate for heterotrophic biomass ($\mathrm{day^{-1}}$)	6.0
$\mathrm{b_H}$	Decay coefficient for heterotrophic biomass ($\mathrm{day^{-1}}$)	0.62
$\mathrm{K_S}$	$\mathrm{S_S}$ half-saturation coefficient for heterotrophic biomass ($\mathrm{gCOD.m^{-3}}$)	20
$\mathrm{K_{OH}}$	$\mathrm{S_O}$ half-saturation coefficient for heterotrophic biomass ($\mathrm{gCOD.m^{-3}}$)	0.2
$\mathrm{K_{NO}}$	$\mathrm{S_{NO}}$ half-saturation coefficient for heterotrophic biomass under anoxic conditions ($\mathrm{gN.m^{-3}}$)	0.5
η_G	Correction factor for μ_H under anoxic conditions (Dimensionless)	0.8
η_H	Correction factor for hydrolysis under anoxic conditions (Dimensionless)	0.4
$\mathrm{k_H}$	Maximum specific hydrolysis rate ($\mathrm{gCOD(gCOD.day)^{-1}}$)	3.0
$\mathrm{K_X}$	Half saturation coefficient for hydrolysis of slowly biodegradable substrate ($\mathrm{gCODgCOD^{-1}}$)	0.03
$\mathrm{k_A}$	Ammonification rate ($\mathrm{m^{-3}.COD(g.day)^{-1}}$)	0.08
$\mathrm{f_P}$	Fraction of biomass leading to inert particulate products (Dimensionless)	0.08
$\mathrm{i_{XB}}$	Mass of nitrogen per mass of COD in biomass ($\mathrm{gN.gCOD^{-1}}$)	0.086
$\mathrm{i_{XP}}$	Mass of nitrogen per mass of COD in products from biomass ($\mathrm{gN.gCOD^{-1}}$)	0.06
$\mathrm{Y_A}$	Yield for autotrophic biomass ($\mathrm{gCOD.gN^{-1}}$)	0.24
μ_A	Maximum specific growth rate for $\mathrm{NH_4^+}$ oxidisers autotrophic biomass ($\mathrm{day^{-1}}$)	0.8
$\mathrm{b_A}$	Decay coefficient for autotrophic biomass ($\mathrm{day^{-1}}$)	0.04
$\mathrm{K_{OA}}$	$\mathrm{S_O}$ half-saturation coefficient for autotrophic biomass ($\mathrm{gCOD.m^{-3}}$)	0.4
$\mathrm{K_{NH}}$	$\mathrm{S_{NH}}$ half-saturation coefficient for autotrophic biomass ($\mathrm{gN.m^{-3}}$)	1.0

2.2 PiWaT1: Piggery Wastewater Treatment model number 1

The PiWaT1 model was initially developed for the intermittent aeration process treating piggery wastewater. Various phenomena such as carbon oxidation, nitrification and denitrification were implemented. The model distinguihes 15 components:

1. Soluble inert COD
2. Readily biodegradable COD
3. Particulate inert COD
4. Slowly Biodegradable COD
5. Active heterotrophic biomass
6. Active ammonium oxidizing biomass
7. Active nitrite oxidizing biomass
8. Particulate inert COD from biomass
9. Dissolved oxygen
10. Nitrite
11. Nitrate
12. Ammonium+ammonia
13. Soluble organic nitrogen
14. Particulate organic nitrogen
15. Particulate inert organic nitrogen

and 11 biological processes:

1. Aerobic growth of heterotrophs
2. Anoxic growth of heterotrophs with $\mathrm{S_{NOI}}$
3. Anoxic growth of heterotrophs with $\mathrm{S_{NOA}}$
4. Aerobic growth of $\mathrm{NH_4^+}$ oxidisers autotrophs
5. Aerobic growth of $\mathrm{NO_2^-}$ oxidisers autotrophs
6. Decay of heterotrophs
7. Decay of $\mathrm{NH_4^+}$ oxidisers autotrophs
8. Decay of $\mathrm{NO_2^-}$ oxidisers autotrophs
9. Ammonification of soluble organic nitrogen
10. Hydrolysis of entrapped organics
11. Hydrolysis of entrapped organic nitrogen

Kinetic and stoichiometry parameters are presented on table 2.4 and 2.5. Default values for this model are presented on table 2.6.

Table 2.4: Process kinetics for organic matter oxidation, nitrification and denitrification for model PiWaT1

j	Process rate	Definition
ρ_1	$\mu_\mathrm{H}\left(\frac{\mathrm{S_S}}{\mathrm{S_S+K_S}}\right)\left(\frac{\mathrm{S_O}}{\mathrm{S_O+K_{OH}}}\right)\left(\frac{\mathrm{S_{NH}}}{\mathrm{S_{NH}+K_{NH}}}\right)\mathrm{X_{BH}}$	Aerobic growth of heterotrophs
ρ_2	$\mu_\mathrm{H}\eta_\mathrm{G}\left(\frac{\mathrm{S_S}}{\mathrm{S_S+K_S}}\right)\left(\frac{\mathrm{K_{AD}}}{\mathrm{S_O+K_{AD}}}\right)\left(\frac{\mathrm{S_{NH}}}{\mathrm{S_{NH}+K_{NH}}}\right)\left(\frac{\mathrm{S_{NOI}}}{\mathrm{S_{NOI}+K_{NOI}}}\right)\left(\frac{\mathrm{S_{NOI}}}{\mathrm{S_{NOI}+S_{NOA}}}\right)\mathrm{X_{BH}}$	Anoxic growth of heterotrophs with $\mathrm{S_{NOI}}$
ρ_3	$\mu_\mathrm{H}\eta_\mathrm{G}\left(\frac{\mathrm{S_S}}{\mathrm{S_S+K_S}}\right)\left(\frac{\mathrm{K_{AD}}}{\mathrm{S_O+K_{AD}}}\right)\left(\frac{\mathrm{S_{NH}}}{\mathrm{S_{NH}+K_{NH}}}\right)\left(\frac{\mathrm{S_{NOA}}}{\mathrm{S_{NOA}+K_{NOA}}}\right)\left(\frac{\mathrm{S_{NOA}}}{\mathrm{S_{NOI}+S_{NOA}}}\right)\mathrm{X_{BH}}$	Anoxic growth of heterotrophs with $\mathrm{S_{NOA}}$
ρ_4	$\mu_\mathrm{AI}\left(\frac{\mathrm{S_{NH}}}{\mathrm{S_{NH}+K_{NAI}}}\right)\left(\frac{\mathrm{S_O}}{\mathrm{S_O+K_{OAI}}}\right)\left(\frac{\mathrm{S_{NH}}}{\mathrm{S_{NH}+K_{NH}}}\right)\mathrm{X_{BAI}}$	Aerobic growth of $\mathrm{NH_4^+}$ oxidisers autotrophs
ρ_5	$\mu_\mathrm{AA}\left(\frac{\mathrm{S_{NOI}}}{\mathrm{S_{NOI}+K_{NAA}}}\right)\left(\frac{\mathrm{S_O}}{\mathrm{S_O+K_{OAA}}}\right)\left(\frac{\mathrm{S_{NH}}}{\mathrm{S_{NH}+K_{NH}}}\right)\mathrm{X_{BAA}}$	Aerobic growth of $\mathrm{NO_2^-}$ oxidisers autotrophs
ρ_6	$\mathrm{b_H}\cdot\mathrm{X_{BH}}$	Decay of heterotrophs
ρ_7	$\mathrm{b_{AI}}\cdot\mathrm{X_{BAI}}$	Decay of $\mathrm{NH_4^+}$ oxidisers autotrophs
ρ_8	$\mathrm{b_{AA}}\cdot\mathrm{X_{BAA}}$	Decay of $\mathrm{NO_2^-}$ oxidisers autotrophs
ρ_9	$\mathrm{k_A}\cdot\mathrm{S_{ND}}\cdot\mathrm{X_{BH}}$	Ammonification of soluble organic nitrogen
ρ_{10}	$\mathrm{K_H}[(\frac{\mathrm{S_O}}{\mathrm{S_O+K_{OH}}}+\eta_\mathrm{H}(\frac{\mathrm{K_{AD}}}{\mathrm{S_O+K_{AD}}})(\frac{\mathrm{S_{NOI}}}{\mathrm{S_{NOI}+K_{NOI}}})(\frac{\mathrm{S_{NOI}}}{\mathrm{S_{NOI}+S_{NOA}}})+\eta_\mathrm{H}(\frac{\mathrm{K_{AD}}}{\mathrm{S_O+K_{AD}}})(\frac{\mathrm{S_{NOA}}}{\mathrm{S_{NOA}+K_{NOA}}})(\frac{\mathrm{S_{NOA}}}{\mathrm{S_{NOI}+S_{NOA}}})]\mathrm{X_S}$	Hydrolysis of entrapped organics
ρ_{11}	$\rho_{10}\left(\frac{\mathrm{X_{ND}}}{\mathrm{X_S}}\right)$	Hydrolysis of entrapped organic nitrogen

Table 2.5: Process stoichiometry for organic matter oxidation, nitrification and denitrification for model PiWaT1

	1 $\mathrm{S_I}$	2 $\mathrm{S_S}$	3 $\mathrm{X_I}$	4 $\mathrm{X_S}$	5 $\mathrm{X_{BH}}$	6 $\mathrm{X_{BAI}}$	7 $\mathrm{X_{BAA}}$	8 $\mathrm{X_P}$	9 $\mathrm{S_O}$	10 $\mathrm{S_{NOI}}$	11 $\mathrm{S_{NOA}}$	12 $\mathrm{S_{NH}}$	13 $\mathrm{S_{ND}}$	14 $\mathrm{X_{ND}}$	15 $\mathrm{X_{NI}}$
ρ_1		$\frac{-1}{\mathrm{Y_H}}$			1				$\frac{-(1-\mathrm{Y_H})}{\mathrm{Y_H}}$			$-\mathrm{i_{BN}}$			
ρ_2		$\frac{-1}{\mathrm{Y_{HD}}}$			1					$-\frac{1-\mathrm{Y_{HD}}}{1.71\mathrm{Y_{HD}}}$		$-\mathrm{i_{BN}}$			
ρ_3		$\frac{-1}{\mathrm{Y_{HD}}}$			1					$\frac{1-\mathrm{Y_{HD}}}{1.14\mathrm{Y_{HD}}}$	$-\frac{1-\mathrm{Y_{HD}}}{1.14\mathrm{Y_{HD}}}$	$-\mathrm{i_{BN}}$			
ρ_4						1			$-\frac{3.43-\mathrm{Y_{AI}}}{\mathrm{Y_{AI}}}$	$\frac{1}{\mathrm{Y_{AI}}}$		$-\mathrm{i_{BN}}-\frac{1}{\mathrm{Y_{AI}}}$			
ρ_5							1		$-\frac{1.14-\mathrm{Y_{AA}}}{\mathrm{Y_{AA}}}$	$-\frac{1}{\mathrm{Y_{AA}}}$	$\frac{1}{\mathrm{Y_{AA}}}$	$-\mathrm{i_{BN}}$			
ρ_6				$1-\mathrm{f_P}$	-1			$\mathrm{f_P}$						$\mathrm{i_{BN}}-\mathrm{f_P}\cdot\mathrm{i_{BN}}$	$\mathrm{f_P}\cdot\mathrm{i_{BN}}$
ρ_7				$1-\mathrm{f_P}$		-1		$\mathrm{f_P}$						$\mathrm{i_{BN}}-\mathrm{f_P}\cdot\mathrm{i_{BN}}$	$\mathrm{f_P}\cdot\mathrm{i_{BN}}$
ρ_8				$1-\mathrm{f_P}$			-1	$\mathrm{f_P}$						$\mathrm{i_{BN}}-\mathrm{f_P}\cdot\mathrm{i_{BN}}$	$\mathrm{f_P}\cdot\mathrm{i_{BN}}$
ρ_9												1	-1		
ρ_{10}		1		-1											
ρ_{11}													1	-1	

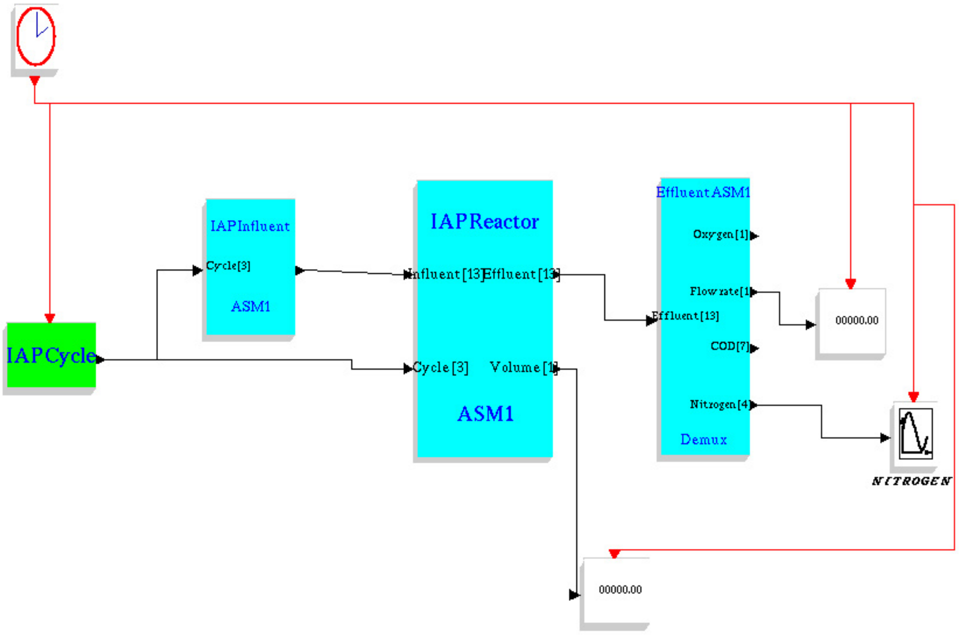
Table 2.6: Default values at 20°C for the model PiWaT1

Symbol	Definition	Default Value
$\mathrm{Y_H}$	Aerobic yield for heterotrophic biomass $(\mathrm{gCOD.gCOD^{-1}})$	0.6
$\mathrm{Y_{HD}}$	Anoxic yield for heterotrophic biomass $(\mathrm{gCOD.gCOD^{-1}})$	0.53
μ_H	Maximum specific growth rate for heterotrophic biomass $(\mathrm{day^{-1}})$	6
$\mathrm{b_H}$	Decay coefficient for heterotrophic biomass $(\mathrm{day^{-1}})$	0.62
$\mathrm{K_S}$	$\mathrm{S_S}$ half-saturation coefficient for heterotrophic biomass $(\mathrm{gCOD.m^{-3}})$	20
$\mathrm{K_{OH}}$	$\mathrm{S_O}$ half-saturation coefficient for heterotrophic biomass under aerobic conditions $(\mathrm{gCOD.m^{-3}})$	0.05
$\mathrm{K_{AD}}$	$\mathrm{S_O}$ inhibition coefficient for heterotrophic biomass under anoxic conditions $(\mathrm{gCOD.m^{-3}})$	0.05
$\mathrm{K_{NH}}$	$\mathrm{S_{NH}}$ half-saturation coefficient for biomass $(\mathrm{gN.m^{-3}})$	0.05
$\mathrm{K_{NOI}}$	$\mathrm{S_{NOI}}$ half-saturation coefficient for heterotrophic biomass under anoxic conditions $(\mathrm{gN.m^{-3}})$	0.5
$\mathrm{K_{NOA}}$	$\mathrm{S_{NOA}}$ half-saturation coefficient for heterotrophic biomass under anoxic conditions $(\mathrm{gN.m^{-3}})$	0.5
η_G	Correction factor for μ_H under anoxic conditions (Dimensionless)	0.8
η_H	Correction factor for hydrolysis under anoxic conditions (Dimensionless)	0.4
$\mathrm{K_H}$	Maximum specific hydrolysis rate $(\mathrm{gCOD(gCOD.day)^{-1}})$	10
$\mathrm{k_A}$	Ammonification rate $(\mathrm{m^{-3}.COD(g.day)^{-1}})$	0.08
$\mathrm{f_P}$	Fraction of biomass leading to inert particulate products (Dimensionless)	0.08
$\mathrm{i_{BN}}$	Mass of nitrogen per mass of COD $(\mathrm{gN.gCOD^{-1}})$	0.086
$\mathrm{Y_{AI}}$	Yield for $\mathrm{NH_4^+}$ oxidisers autotrophic biomass $(\mathrm{gCOD.gN^{-1}})$	0.15
μ_AI	Maximum specific growth rate for $\mathrm{NH_4^+}$ oxidisers autotrophic biomass $(\mathrm{day^{-1}})$	0.8
$\mathrm{b_{AI}}$	Decay coefficient for $\mathrm{X_{BAI}}$ autotrophic biomass $(\mathrm{day^{-1}})$	0.05
$\mathrm{K_{OAI}}$	$\mathrm{S_O}$ half-saturation coefficient for $\mathrm{X_{BAI}}$ biomass $(\mathrm{gCOD.m^{-3}})$	0.3
$\mathrm{K_{NAI}}$	$\mathrm{S_{NH}}$ half-saturation coefficient for $\mathrm{X_{BAI}}$ biomass $(\mathrm{gN.m^{-3}})$	5
$\mathrm{Y_{AA}}$	Yield for $\mathrm{NO_2^-}$ oxidisers autotrophic biomass $(\mathrm{gCOD.gN^{-1}})$	0.041
μ_AA	Maximum specific growth rate for $\mathrm{NO_2^-}$ oxidisers autotrophic biomass $(\mathrm{day^{-1}})$	0.79
$\mathrm{b_{AA}}$	Decay coefficient for $\mathrm{X_{BAA}}$ autotrophic biomass $(\mathrm{day^{-1}})$	0.033
$\mathrm{K_{OAA}}$	$\mathrm{S_O}$ half-saturation coefficient for $\mathrm{X_{BAA}}$ biomass $(\mathrm{gCOD.m^{-3}})$	1.1
$\mathrm{K_{NAA}}$	$\mathrm{S_{NOI}}$ half-saturation coefficient for $\mathrm{X_{BAA}}$ biomass $(\mathrm{gN.m^{-3}})$	2.5

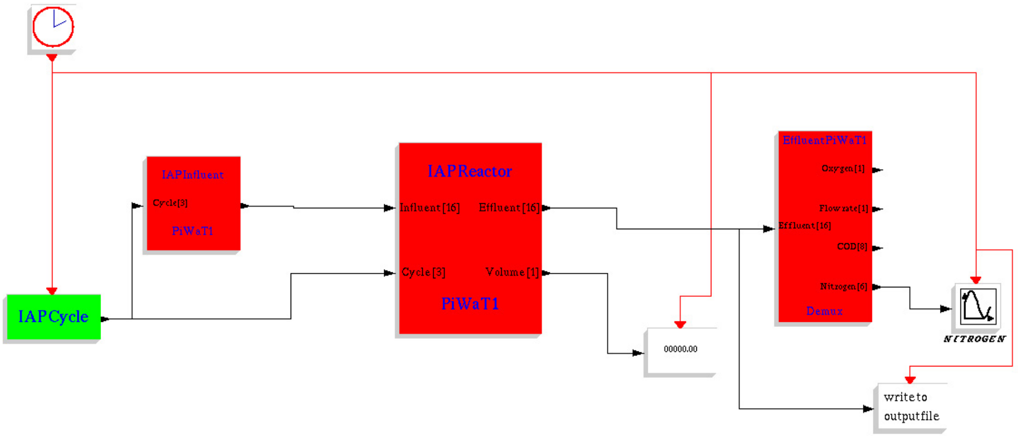
Chapter 3

Examples

3.1 Example 1 using model ASM1



3.2 Example 2 using model PiWaT1


IAPInfluent
Cycle[3]
PiWaT1
IAPReactor
Influent[16]
Effluent[16]
Cycle[3]
Volume[1]
PiWaT1
IAPCycle
EffluentPiWaT1
Oxygen[1]
Flowrate[1]
Effluent[16]
COD[8]
Nitrogen[6]
Demux
00000.00
NITROGEN
write to
outputfile

Chapter 4

Blocks

4.1 IAP Cycle

Library

ASModel

Description

This block allows user to define the cycle characteristics: determination of the duration and the action of each phase (4 maximum).

Dialog Box

Cycle characteristics	
Phase 1 [Time(hrs);Feeding(0/1);Aeration(0/1);Withdrawal(0/1)]	[1;1;0;0]
Phase 2 [Time(hrs);Feeding(0/1);Aeration(0/1);Withdrawal(0/1)]	[1;1;0;0]
Phase 3 [Time(hrs);Feeding(0/1);Aeration(0/1);Withdrawal(0/1)]	[1;1;0;0]
Phase 4 [Time(hrs);Feeding(0/1);Aeration(0/1);Withdrawal(0/1)]	[1;1;0;0]

- Phase 1 [Time(hrs);Feeding(0/1);Aeration(0/1);Withdrawal(0/1)]: a vector of positive integer [a1,b1,c1,d1] corresponding for phase 1 to:
  - a1: duration in hours of the phase
  - b1: aeration on (0=no, 1=yes)
  - c1: feeding pump on (0=no, 1=yes)
  - d1: withdrawal pump on (0=no, 1=yes)
- Phase 2 [Time(hrs);Feeding(0/1);Aeration(0/1);Withdrawal(0/1)]: a vector of positive integer [a2,b2,c2,d2] corresponding for phase 2 to:
  - a2: duration in hours of the phase
  - b2: aeration on (0=no, 1=yes)
  - c2: feeding pump on (0=no, 1=yes)

  - d2: withdrawal pump on (0=no, 1=yes)

- Phase 3 [Time(hrs);Feeding(0/1);Aeration(0/1);Withdrawal(0/1)]: a vector of positive integer [a3,b3,c3,d3] corresponding for phase 3 to:
  - a3: duration in hours of the phase
  - b3: aeration on (0=no, 1=yes)
  - c3: feeding pump on (0=no, 1=yes)
  - d3: withdrawal pump on (0=no, 1=yes)
- Phase 4 [Time(hrs);Feeding(0/1);Aeration(0/1);Withdrawal(0/1)]: a vector of positive integer [a4,b4,c4,d4] corresponding for phase 4 to:
  - a4: duration in hours of the phase
  - b4: aeration on (0=no, 1=yes)
  - c4: feeding pump on (0=no, 1=yes)
  - d4: withdrawal pump on (0=no, 1=yes)

Default properties

- always active: yes
- direct-feedthrough: no
- zero-crossing: no
- mode: no
- number/sizes of inputs: 0/
- number/sizes of outputs: 1/3
- number/sizes of activation inputs: 1/1
- number/sizes of activation outputs: 0/
- continuous-time state:
- discrete-time state:
- name of computational function: *IAP_Cycle*

Interfacing function

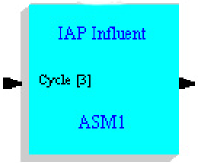
`./macros/IAP_Cycle.sci`

Computational function (type 1)

```
#include "stdio.h"

#define P1_T rpar[0]/24
#define P1_F rpar[1]
#define P1_A rpar[2]
#define P1_W rpar[3]
#define P2_T rpar[4]/24
#define P2_F rpar[5]
#define P2_A rpar[6]
#define P2_W rpar[7]
#define P3_T rpar[8]/24
#define P3_F rpar[9]
```

```
#define P3_A rpar[10]
#define P3_W rpar[11]
#define P4_T rpar[12]/24
#define P4_F rpar[13]
#define P4_A rpar[14]
#define P4_W rpar[15]

void IAP_Cycle(int * flag, int *nevprt, double* t, double* xd,
               double* x, int* nx, double* z, int *nz,
               double* tvec, int* ntvec, double* rpar, int* nrpar,
               int* ipar, int* nipar, double* y1,int* ny1)

{
  int nb_cycle;
  float cycle;
  float Tps_Cycle;

  if (*flag==1||*flag==6)
    {
      cycle= P1_T + P2_T + P3_T + P4_T;
      nb_cycle=*t/cycle;
      Tps_Cycle=*t-nb_cycle*cycle;
      if (Tps_Cycle<=P1_T)
        {
          y1[0]=P1_F;
          y1[1]=P1_A;
          y1[2]=P1_W;
        }
      else
        {
          if (Tps_Cycle<=(P1_T+P2_T))
            {
              y1[0]=P2_F;
              y1[1]=P2_A;
              y1[2]=P2_W;
            }
          else
            {
              if (Tps_Cycle<=(P1_T+P2_T+P3_T))
                {
                  y1[0]=P3_F;
                  y1[1]=P3_A;
                  y1[2]=P3_W;
                }
              else
                {
                  y1[0]=P4_F;
                  y1[1]=P4_A;
                  y1[2]=P4_W;
                }
            }
        }
    }
  return;
}
```

4.2 IAP Influent ASM1



Library

ASModel

Description

This block allows user to define the characteristics of the influent according to ASM1 model and the flow rate of the feeding pump.

Dialog Box

Feed flow rate	
Feed flow rate (m3/h)	20
Influent characteristics	
Soluble inert COD - Si (gDCO/m3)	3000
Readily biodegradable COD - Ss (gDCO/m3)	6500
Particulate inert COD - Xi (gDCO/m3)	10000
Slowly Biodegradable COD - Xs (gDCO/m3)	15000
Active heterotrophic biomass - Xbh (gDCO/m3)	0
Active autotrophic biomass - Xba (gDCO/m3)	0
Particulate inert COD from biomass - Xp (gDCO/m3)	0
Dissolved oxygen - So (gO2/m3)	0
Nitrate+nitrite - Sno (gN/m3)	0
Ammonium+ammonia - Snh (gN/m3)	3500
Soluble organic nitrogen - Snd (gN/m3)	100
Particulate organic nitrogen - Xnd (gN/m3)	300

- Feed flow rate (m3/h): a real representing the flow rate of the feeding pump in m3/h
- Soluble inert COD - Si (gDCO/m3): a real representing the soluble inert COD concentration of the influent in gCOD/m3
- Readily biodegradable COD - Ss (gDCO/m3): a real representing the readily biodegradable COD concentration of the influent in gCOD/m3
- Particulate inert COD - Xi (gDCO/m3): a real representing the particulate inert COD concentration of the influent in gCOD/m3
- Slowly Biodegradable COD - Xs (gDCO/m3): a real representing the slowly Biodegradable COD concentration of the influent in gCOD/m3
- Active heterotrophic biomass - Xbh (gDCO/m3): a real representing the active heterotrophic biomass concentration of the influent in gCOD/m3
- Active autotrophic biomass - Xba (gDCO/m3): a real representing the active autotrophic biomass concentration of the influent in gCOD/m3

- Particulate inert COD from biomass - Xp (gDCO/m3): a real representing the particulate inert COD from biomass concentration of the influent in gCOD/m3
- Dissolved oxygen - So (gO2/m3): a real representing the dissolved oxygen concentration of the influent in gO2/m3
- Nitrate+nitrite - Sno (gN/m3): a real representing the nitrate+nitrite concentration of the influent in gN/m3
- Ammonium+ammonia - Snh (gN/m3): a real representing the ammonium+ammonia concentration of the influent in gN/m3
- Soluble organic nitrogen - Snd (gN/m3): a real representing the soluble organic nitrogen concentration of the influent in gN/m3
- Particulate organic nitrogen - Xnd (gN/m3): a real representing the particulate organic nitrogen concentration of the influent in gN/m3

Default properties

- always active: no
- direct-feedthrough: yes
- zero-crossing: no
- mode: no
- number/sizes of inputs: 1/3
- number/sizes of outputs: 1/13
- number/sizes of activation inputs: 0/
- number/sizes of activation outputs: 0/
- continuous-time state:
- discrete-time state:
- name of computational function: *Influent_ASM1*

Interfacing function

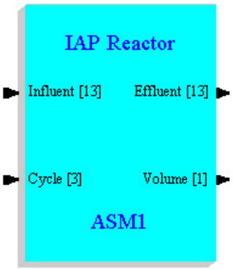
`./macros/Influent_ASM1.sci`

Computational function (type 1)

```
#include "stdio.h"

#define  Si rpar[1]
#define  Ss rpar[2]
#define  Xi rpar[3]
#define  Xs rpar[4]
#define  Xbh rpar[5]
#define  Xba rpar[6]
#define  Xp rpar[7]
#define  So rpar[8]
#define  Sno rpar[9]
#define  Snh rpar[10]
#define  Snd rpar[11]
#define  Xnd rpar[12]

#define Feeding u1[0]
```

```
#define Aeration u1[1]
#define Withdraw u1[2]

void Influent_ASM1(int * flag, int *nevprt, double* t, double* xd,
                  double* x, int* nx, double* z, int *nz,
                  double* tvec, int* ntvec, double* rpar, int* nrpar,
                  int* ipar, int* nipar, double* u1,int* nu1,
                  double* y1, int* ny1)

{
  float Qa;

  if (Feeding==1)
    {
    Qa=rpar[0]*24;
    }
  else
    {
    Qa=0;
    }
  if (*flag==1||*flag==6)
    {
      y1[0]=Qa;
      y1[1]=Si;
      y1[2]=Ss;
      y1[3]=Xi;
      y1[4]=Xs;
      y1[5]=Xbh;
      y1[6]=Xba;
      y1[7]=Xp;
      y1[8]=So;
      y1[9]=Sno;
      y1[10]=Snh;
      y1[11]=Snd;
      y1[12]=Xnd;
    }
  else
    {
      if (*flag==0)
        {
        }
    }
  return;
}
```

4.3 IAP Reactor ASM1



Library

ASModel

Description

This block simulate the Intermittent Aeration Process (IAP) reactor using ASM1 model and allows user to define stoichiometric and kinetic constants of the ASM1 model, the aeration capacity, the initial states of the reactor, the temperature and the withdrawal flow rate.

Dialog Box

Stoichiometric constants - Model ASM1	
Heterotrophic yield - Yh (gDCO/gDCO)	0.67
Autotrophic yield - Ya (gDCO/gN)	0.24
Inert fraction of biomass - fp (-)	0.08
Nitrogen content of active biomass - ixb (gN/gDCO)	0.086
Nitrogen content of inert biomass - ixp (gN/gDCO)	0.06

Kinetic constants - Model ASM1	
Heterotrophic max. specific growth rate - muh (1/d)	6.0
Heterotrophic decay rate - bh (1/d)	0.62
Oxygen half-saturation coefficient for heterotrophs - Koh (gO2/m3)	0.2
Ss half-saturation coefficient for heterotrophs - Ks (gDCO/m3)	20
Sno half-saturation coefficient for heterotrophs - Kno (gN/m3)	0.5
Anoxic growth factor reduction - ng (-)	0.8
Autotrophic max. specific growth rate - mua (1/d)	0.8
Autotrophic decay rate - ba (1/d)	0.04
Oxygen half-saturation coefficient for Autotrophs - Koa (gO2/m3)	1.0
Snh half-saturation coefficient for Autotrophs - Knh (gN/m3)	1.0
Xs max. specific hydrolysis rate - kh (gDCO/(gDCO.d))	3
Xs half-saturation coefficient - Kx (gDCO/gDCO)	0.03
Anoxic hydrolysis factor reduction - nh (-)	0.4
Ammonification rate - ka (m3.DCO/(g.d))	0.08
Aeration capacity	
Mass transfer coefficient - Kla (1/d)	3000
Alpha factor(-)	0.8
Beta factor (-)	0.95
Initial States	
Soluble inert COD - Si (gDCO/m3)	30
Readily biodegradable COD - Ss (gDCO/m3)	10
Particulate inert COD - Xi (gDCO/m3)	10
Slowly Biodegradable COD - Xs (gDCO/m3)	15
Active heterotrophic biomass - Xbh (gDCO/m3)	500
Active autotrophic biomass - Xba (gDCO/m3)	50
Particulate inert COD from biomass - Xp (gDCO/m3)	10
Dissolved oxygen - So (gO2/m3)	1
Nitrate+nitrite - Sno (gN/m3)	10
Ammonium+ammonia - Snh (gN/m3)	10
Soluble organic nitrogen - Snd (gN/m3)	10
Particulate organic nitrogen - Xnd (gN/m3)	10
Volume (m3)	200
Save final states (0/1)	0
Load initial states (0/1)	0
Temperature	
Temperature (Deg. Celcius)	20
Flow rate	
Withdrawal flow rate (m3/h)	20

- Heterotrophic yield - Yh (gDCO/gDCO): a real representing the heterotrophic yield in gDCO/gDCO
- Autotrophic yield - Ya (gDCO/gN): a real representing the autotrophic yield in gDCO/gN
- Inert fraction of biomass - fp (-): a real representing the inert fraction of biomass
- Nitrogen content of active biomass - ixb (gN/gDCO): a real representing the nitrogen content of active biomass in gN/gDCO

- Nitrogen content of inert biomass - ixp (gN/gDCO): a real representing the nitrogen content of inert biomass in gN/gDCO
- Heterotrophic max. specific growth rate - muh (1/d): a real representing the heterotrophic maximum specific growth rate in 1/d
- Heterotrophic decay rate - bh (1/d): a real representing the heterotrophic decay rate in 1/d
- Oxygen half-saturation coefficient for heterotrophs - Koh (gO2/m3): a real representing the oxygen half-saturation coefficient for heterotrophs in gO2/m3
- Ss half-saturation coefficient for heterotrophs - Ks (gDCO/m3): a real representing the Ss half-saturation coefficient for heterotrophs in gDCO/m3
- Sno half-saturation coefficient for heterotrophs - Kno (gN/m3): a real representing the Sno half-saturation coefficient for heterotrophs in gN/m3
- Anoxic growth factor reduction - ng (-): a real representing the anoxic growth factor reduction
- Autotrophic max. specific growth rate - mua (1/d): a real representing the autotrophic maximum specific growth rate in 1/d
- Autotrophic decay rate - ba (1/d): a real representing the autotrophic decay rate in 1/d
- Oxygen half-saturation coefficient for Autotrophs - Koa (gO2/m3): a real representing the oxygen half-saturation coefficient for Autotrophs in gO2/m3
- Snh half-saturation coefficient for Autotrophs - Knh (gN/m3): a real representing the Snh half-saturation coefficient for Autotrophs in gN/m3
- Xs max. specific hydrolysis rate - kh (gDCO/(gDCO.d)): a real representing the Xs maximum specific hydrolysis rate in gDCO/(gDCO.d
- Xs half-saturation coefficient - Kx (gDCO/gDCO): a real representing the Xs half-saturation coefficient in gDCO/gDCO
- Anoxic hydrolysis factor reduction - nh (-): a real representing the anoxic hydrolysis factor reduction
- Ammonification rate - ka (m3.DCO/(g.d)): a real representing the ammonification rate in m3.DCO/(g.d)
- Mass transfer coefficient - Kla (1/d): a real representing the mass transfer coefficient in 1/d
- Alpha factor(-): a real representing the alpha factor
- Beta factor (-): a real representing the beta factor
- Soluble inert COD - Si (gDCO/m3): a real representing the initial soluble inert COD concentration of the reactor in gDCO/m3
- Readily biodegradable COD - Ss (gDCO/m3): a real representing the initial readily biodegradable COD concentration of the reactor in gDCO/m3
- Particulate inert COD - Xi (gDCO/m3): a real representing the initial particulate inert COD concentration of the reactor in gDCO/m3
- Slowly Biodegradable COD - Xs (gDCO/m3): a real representing the initial slowly Biodegradable COD concentration of the reactor in gDCO/m3
- Active heterotrophic biomass - Xbh (gDCO/m3): a real representing the initial active heterotrophic biomass concentration of the reactor in gDCO/m3
- Active autotrophic biomass - Xba (gDCO/m3): a real representing the initial active autotrophic biomass concentration of the reactor in gDCO/m3
- Particulate inert COD from biomass - Xp (gDCO/m3): a real representing the initial particulate inert COD from biomass concentration of the reactor in gDCO/m3

- Dissolved oxygen - So (gO2/m3): a real representing the initial dissolved oxygen concentration of the reactor in gO2/m3
- Nitrate+nitrite - Sno (gN/m3): a real representing the initial nitrate+nitrite concentration of the reactor in gN/m3
- Ammonium+ammonia - Snh (gN/m3): a real representing the initial ammonium+ammonia concentration of the reactor in gN/m3
- Soluble organic nitrogen - Snd (gN/m3): a real representing the initial soluble organic nitrogen concentration of the reactor in gN/m3
- Particulate organic nitrogen - Xnd (gN/m3): a real representing the initial particulate organic nitrogen concentration of the reactor in gN/m3
- Volume (m3): a real representing the initial volume of the reactor in m”
- Save final states (0/1): Save the final concentrations and volume of the reactor (0:no, 1: yes)
- Load initial states (0/1): Load and use the initial concentrations and volume of the reactor from a file “state.asm1” (0:no, 1: yes)
- Temperature (Deg. Celcius): a real representing the temperature in Deg. celcius
- Withdrawal flow rate (m3/h): a real representing the flow rate of the withdrawal pump in m3/h

Default properties

- always active: yes
- direct-feedthrough: no
- zero-crossing: no
- mode: no
- number/sizes of inputs: 2/13/3
- number/sizes of outputs: 2/13/1
- number/sizes of activation inputs: 0/
- number/sizes of activation outputs: 0/
- continuous-time state:
- discrete-time state:
- name of computational function: *IAPreacteur_ASM1*

Interfacing function

`./macros/IAPreacteur_ASM1.sci`

Computational function (type 1)

```
#include "stdio.h"
#include "stdlib.h"
#include "math.h"

#define  Si x[0]
#define  Ss x[1]
#define  Xi x[2]
#define  Xs x[3]
#define  Xbh x[4]
```

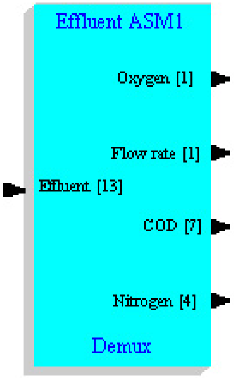
```
#define  Xba x[5]
#define  Xp x[6]
#define  So x[7]
#define  Sno x[8]
#define  Snh x[9]
#define  Snd x[10]
#define  Xnd x[11]
#define  Vr x[12]

#define Yh rpar[0]
#define Ya rpar[1]
#define fp rpar[2]
#define ixb rpar[3]
#define ixp rpar[4]

#define muh rpar[5]
#define bh rpar[6]
#define Koh rpar[7]
#define Ks rpar[8]
#define Kno rpar[9]
#define ng rpar[10]
#define mua rpar[11]
#define ba rpar[12]
#define Koa rpar[13]
#define Knh rpar[14]
#define kh rpar[15]
#define Kx rpar[16]
#define nh rpar[17]
#define ka rpar[18]

#define Alpha rpar[20]
#define Beta rpar[21]

#define Temperature rpar[22]

#define FILE_OUT rpar[24]
#define FILE_IN rpar[25]

#define Qa u1[0]
#define Sia u1[1]
#define Ssa u1[2]
#define Xia u1[3]
#define Xsa u1[4]
#define Xbha u1[5]
#define Xbaa u1[6]
#define Xpa u1[7]
#define Soa u1[8]
#define Snoa u1[9]
#define Snha u1[10]
#define Snda u1[11]
#define Xnda u1[12]

#define Feeding u2[0]
#define Aeration u2[1]
#define Withdraw u2[2]
```

```
#define Cs Beta*(-0.0000661858*pow(Temperature,3)+0.0075161751
                 *pow(Temperature,2)-0.3999077443*Temperature+14.58958)

#define process1 muh*(Ss/(Ks+Ss))*(So/(Koh+So))*Xbh
#define process2 ng*muh*(Ss/(Ks+Ss))*(Koh/(Koh+So))*(Sno/(Kno+Sno))*Xbh
#define process3 mua*(Snh/(Knh+Snh))*(So/(Koa+So))*Xba
#define process4 bh*Xbh
#define process5 ba*Xba
#define process6 ka*Snd*Xbh
#define process7 kh*((Xs/Xbh)/(Kx+(Xs/Xbh)))*((So/(Koh+So))+nh*(Koh/(Koh+So))
                 *(Sno/(Kno+Sno)))*Xbh
#define process8 process7*(Xnd/Xs)

void IAPreacteur_ASM1(int * flag, int *nevprt, double* t, double* xd,
                    double* x, int* nx, double* z, int *nz, double* tvec,
                    int* ntvec, double* rpar, int* nrpar, int* ipar,
                    int* nipar, double* u1, int* nu1,double* u2, int* nu2,
                    double* y1, int* ny1, double* y2, int* ny2)

{
  float Qs;
  float Kla;
  float temp1,temp2,temp3,temp4,temp5,temp6,temp7,temp8,temp9,temp10,temp11,
        temp12,temp13;
  FILE *stream;
  char file[]="state.asm1";

  if (Withdraw==1)
    {
      Qs=rpar[23]*24;
    }
  else
    {
      Qs=0;
    }


  if (Aeration==1)
    {
      Kla=Alpha*rpar[19];
    }
  else
    {
      Kla=0;
    }


 if (*flag==4 && FILE_IN==1)
    {
      /*  Initialisation */
      if ((stream=fopen(file, "r"))==NULL)
        {
          /*printf("Ouverture du fichier impossible");*/
          exit(-1);
        }
      else
```

```
       {
         fscanf(stream, "%f%f%f%f%f%f%f%f%f%f%f%f%f",&temp1,&temp2,
                         &temp3,&temp4,&temp5,&temp6,&temp7,&temp8,
                         &temp9,&temp10,&temp11,&temp12,&temp13);
         fclose(stream);
         Si=temp1;
         Ss=temp2;
         Xi=temp3;
         Xs=temp4;
         Xbh=temp5;
         Xba=temp6;
         Xp=temp7;
         So=temp8;
         Sno=temp9;
         Snh=temp10;
         Snd=temp11;
         Xnd=temp12;
         Vr=temp13;
       }
   }
else
  {
    if (*flag==5 && FILE_OUT==1)
      {
        /*  Ending */
        if ((stream=fopen(file, "w"))==NULL)
          {
            exit(-1);
          }
        else
          {
            fprintf(stream,"%.1f\t%.1f\t%.1f\t%.1f\t%.1f\t%.1f\t%.1f
                            \t%.1f\t%.1f\t%.1f\t%.1f\t%.1f\t%.1f\n",
                            y1[1],y1[2],y1[3],y1[4],y1[5],y1[6],y1[7],
                            y1[8],y1[9],y1[10],y1[11],y1[12],y2[0]);
            fclose(stream);
          }

      }
    else
      {
        if (*flag==1||*flag==6)
          {
            y1[0]=Qs;
            y1[1]=Si;
            y1[2]=Ss;
            y1[3]=Xi;
            y1[4]=Xs;
            y1[5]=Xbh;
            y1[6]=Xba;
            y1[7]=Xp;
            y1[8]=So;
            y1[9]=Sno;
            y1[10]=Snh;
            y1[11]=Snd;
            y1[12]=Xnd;
```

```
             y2[0]=Vr;
           }
         else
           {
             if (*flag==0)
               {
                 xd[0]=Qa*(Sia-Si)/Vr;
                 xd[1]=Qa*(Ssa-Ss)/Vr-1/Yh*process1-1/Yh*process2
                       +process7;
                 xd[2]=Qa*(Xia-Xi)/Vr;
                 xd[3]=Qa*(Xsa-Xs)/Vr+(1-fp)*process4 +(1-fp)*process5
                       -process7;
                 xd[4]=Qa*(Xbha-Xbh)/Vr+process1 +process2 -process4;
                 xd[5]=Qa*(Xbaa-Xba)/Vr+process3 -process5;
                 xd[6]=Qa*(Xpa-Xp)/Vr+fp*process4 +fp*process5;
                 xd[7]=Qa*(Soa-So)/Vr-((1-Yh)/Yh)*process1
                       -((4.57-Ya)/Ya)*process3 + Kla*(Cs-So);
                 xd[8]=Qa*(Snoa-Sno)/Vr-((1-Yh)/(2.86*Yh))*process2
                       +1/Ya*process3;
                 xd[9]=Qa*(Snha-Snh)/Vr-ixb*process1 -ixb*process2
                       -(ixb+1/Ya)*process3 +process6;
                 xd[10]=Qa*(Snda-Snd)/Vr-process6 +process8;
                 xd[11]=Qa*(Xnda-Xnd)/Vr+(ixb-fp*ixp)*process4
                        +(ixb-fp*ixp)*process5 -process8;
                 xd[12]=Qa-Qs;
               }
           }
       }
   }

 return;
}
```

4.4 Effluent ASM1 Demux



Library

ASModel

Description

This block allow is a demultiplexer for ASM1 effluent. This block allow user to divide the effluent characteristics into 4 output including dissolved oxygen (DO), Flow rate, Chemical Oxygen Demand (COD) and Nitrogen.

Default properties

- always active: no
- direct-feedthrough: yes
- zero-crossing: no
- mode: no
- number/sizes of inputs: 1/13
- number/sizes of outputs: 4/1/1/7/4
- number/sizes of activation inputs: 0/
- number/sizes of activation outputs: 0/
- continuous-time state:
- discrete-time state:
- name of computational function: *ASM1_Demux*

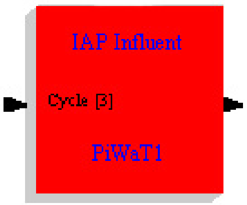
Interfacing function

`./macros/ASM1_Demux.sci`

Computational function (type 1)

```
#include "stdio.h"


void ASM1_Demux(int * flag, int *nevprt, double* t, double* xd,
                double* x, int* nx, double* z, int *nz,
                double* tvec, int* ntvec, double* rpar,
                int* nrpar, int* ipar, int* nipar, double* u1,
                int* nu1, double* y1, int* ny1, double* y2,
                int* ny2,double* y3, int* ny3, double* y4, int* ny4)

{
  if (*flag==1||*flag==6)
    {
      y1[0]=u1[8];
      y2[0]=u1[0];
      y3[0]=u1[1];
      y3[1]=u1[2];
      y3[2]=u1[3];
      y3[3]=u1[4];
      y3[4]=u1[5];
      y3[5]=u1[6];
      y3[6]=u1[7];
      y4[0]=u1[9];
      y4[1]=u1[10];
      y4[2]=u1[11];
      y4[3]=u1[12];
    }
  else
    {
      if (*flag==0)
        {
        }
    }
  return;
}
```

4.5 IAP Influent PiWaT1



Library

ASModel

Description

This block allows user to define the characteristics of the influent according to PiWaT1 model and the flow rate of the feeding pump.

Dialog Box

Feed flow rate	
Feed flow rate (m3/h)	20
Influent characteristics	
Soluble inert COD - Si (gDCO/m3)	3000
Readily biodegradable COD - Ss (gDCO/m3)	6500
Particulate inert COD - Xi (gDCO/m3)	10000
Slowly Biodegradable COD - Xs (gDCO/m3)	15000
Active heterotrophic biomass - Xbh (gDCO/m3)	0
Active ammonium oxidizing biomass - Xbai (gDCO/m3)	0
Active nitrite oxidizing biomass - Xbaa (gDCO/m3)	0
Particulate inert COD from biomass - Xp (gDCO/m3)	0
Dissolved oxygen - So (gO2/m3)	0
Nitrite - Snoi (gN/m3)	0
Nitrate - Snoa (gN/m3)	0
Ammonium+ammonia - Snh (gN/m3)	3500
Soluble organic nitrogen - Snd (gN/m3)	100
Particulate organic nitrogen - Xnd (gN/m3)	300
Particulate inert organic nitrogen - Xni (gN/m3)	500

- Feed flow rate (m3/h): a real representing the flow rate of the feeding pump in m3/h

- Soluble inert COD - Si (gDCO/m3): a real representing the soluble inert COD concentration of the influent in gCOD/m3
- Readily biodegradable COD - Ss (gDCO/m3): a real representing the readily biodegradable COD concentration of the influent in gCOD/m3
- Particulate inert COD - Xi (gDCO/m3): a real representing the particulate inert COD concentration of the influent in gCOD/m3
- Slowly Biodegradable COD - Xs (gDCO/m3): a real representing the slowly Biodegradable COD concentration of the influent in gCOD/m3
- Active heterotrophic biomass - Xbh (gDCO/m3): a real representing the active heterotrophic biomass concentration of the influent in gCOD/m3
- Active ammonium oxidizing biomass - Xbai (gDCO/m3): a real representing the active ammonium oxidizing biomass concentration of the influent in gCOD/m3
- Active nitrite oxidizing biomass - Xbai (gDCO/m3): a real representing the active nitrite oxidizing biomass concentration of the influent in gCOD/m3
- Particulate inert COD from biomass - Xp (gDCO/m3): a real representing the particulate inert COD from biomass concentration of the influent in gCOD/m3
- Dissolved oxygen - So (gO2/m3): a real representing the dissolved oxygen concentration of the influent in gO2/m3
- Nitrite - Snoi (gN/m3): a real representing the nitrite concentration of the influent in gN/m3
- Nitrate - Snoa (gN/m3): a real representing the nitrate concentration of the influent in gN/m3
- Ammonium+ammonia - Snh (gN/m3): a real representing the ammonium+ammonia concentration of the influent in gN/m3
- Soluble organic nitrogen - Snd (gN/m3): a real representing the soluble organic nitrogen concentration of the influent in gN/m3
- Particulate organic nitrogen - Xnd (gN/m3): a real representing the particulate organic nitrogen concentration of the influent in gN/m3
- Particulate inert organic nitrogen - Xnd (gN/m3): a real representing the particulate inert organic nitrogen concentration of the influent in gN/m3

Default properties

- always active: no
- direct-feedthrough: yes
- zero-crossing: no
- mode: no
- number/sizes of inputs: 1/3
- number/sizes of outputs: 1/16
- number/sizes of activation inputs: 0/
- number/sizes of activation outputs: 0/
- continuous-time state:
- discrete-time state:
- name of computational function: *Influent_PiWaT1*

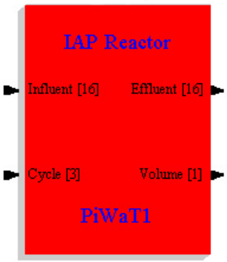
Interfacing function

`./macros/Influent_PiWaT1.sci`

Computational function (type 1)

```
#include "stdio.h"

#define  Si rpar[1]
#define  Ss rpar[2]
#define  Xi rpar[3]
#define  Xs rpar[4]
#define  Xbh rpar[5]
#define  Xbai rpar[6]
#define  Xbaa rpar[7]
#define  Xp rpar[8]
#define  So rpar[9]
#define  Snoi rpar[10]
#define  Snoa rpar[11]
#define  Snh rpar[12]
#define  Snd rpar[13]
#define  Xnd rpar[14]
#define  Xni rpar[15]

#define Feeding u1[0]
#define Aeration u1[1]
#define Withdraw u1[2]

void Influent_PiWaT1(int * flag, int *nevprt, double* t, double* xd,
             double* x, int* nx, double* z, int *nz, double* tvec,
             int* ntvec, double* rpar, int* nrpar, int* ipar,
             int* nipar, double* u1,
             int* nu1, double* y1, int* ny1)

{
  float Qa;

  if (Feeding==1)
    {
    Qa=rpar[0]*24;
    }
  else
    {
    Qa=0;
    }

  if (*flag==1||*flag==6)
    {
      y1[0]=Qa;
      y1[1]=Si;
      y1[2]=Ss;
      y1[3]=Xi;
      y1[4]=Xs;
      y1[5]=Xbh;
      y1[6]=Xbai;
      y1[7]=Xbaa;
      y1[8]=Xp;
      y1[9]=So;
      y1[10]=Snoi;
      y1[11]=Snoa;
      y1[12]=Snh;
      y1[13]=Snd;
      y1[14]=Xnd;
      y1[15]=Xni;
    }
  else
    {
      if (*flag==0)
        {
        }
    }
  return;
}
```

4.6 IAP Reactor PiWaT1



Library

ASModel

Description

This block simulate the Intermittent Aeration Process (IAP) reactor using PiWaT1 model and allows user to define stoichiometric and kinetic constants of the PiWaT1 model, the aeration capacity, the initial states of the reactor, the temperature and the withdrawal flow rate.

Dialog Box

Stoichiometric constants - Model PiWaT1	
Aerobic heterotrophic yield - Yh (gDCO/gDCO)	0.60
Anoxic heterotrophic yield - Yhd (gDCO/gDCO)	0.53
Ammonium oxidisers yield - Yai (gDCO/gN)	0.15
Nitrite oxidisers yield - Yaa (gDCO/gN)	0.041
Inert fraction of biomass - fp (-)	0.08
Nitrogen content of active biomass - ibn (gN/gDCO)	0.086
Kinetic constants (Hetrotrophic)- Model PiWaT1	
Heterotrophic max. specific growth rate - muh (1/d)	6.0
Heterotrophic decay rate - bh (1/d)	0.62
Oxygen half-saturation coefficient for heterotrophs - Koh (gO2/m3)	0.05
Oxygen inhibition coefficient for heterotrophs - Kad (gO2/m3)	0.05
Ss half-saturation coefficient for heterotrophs - Ks (gDCO/m3)	20
Nitrite half-saturation coefficient for heterotrophs - Knoi (gN/m3)	0.5
Nitrate half-saturation coefficient for heterotrophs - Knoa (gN/m3)	0.5
Anoxic growth factor reduction - ng (-)	0.8
Snh half-saturation coefficient - Knh (gN/m3)	0.05
Xs max. specific hydrolysis rate - Kh (gDCO/(gDCO.d))	10
Anoxic hydrolysis factor reduction - nh (-)	0.4
Ammonification rate - ka (m3.DCO/(g.d))	0.08
Kinetic constants (Autotrophic) - Model PiWaT1	
Ammonium oxidisers max. specific growth rate - muai (1/d)	0.8
Ammonium oxidisers decay rate - bai (1/d)	0.05
Ammonium half-saturation coefficient for Ammonium oxidisers - Knai (gN/m3)	5
Oxygen half-saturation coefficient for Ammonium oxidisers - Koai (gO2/m3)	0.3
Nitrite oxidisers max. specific growth rate - muaa (1/d)	0.79
Nitrite oxidisers decay rate - baa (1/d)	0.033
Nitrite half-saturation coefficient for Nitrite oxidisers - Knaa (gN/m3)	2.5
Oxygen half-saturation coefficient for Nitrite oxidisers - Koaa (gO2/m3)	1.1
Aeration capacity	
Mass transfer coefficient - Kla (1/d)	3000
Alpha factor(-)	0.8
Beta factor (-)	0.95
Temperature	
Temperature (Deg. Celcius)	20
Flow rate	
Withdrawal flow rate (m3/h)	20
Initial states	
Soluble inert COD Si (gDCO/m3)	30
Readily biodegradable COD - Ss (gDCO/m3)	10
Particulate inert COD - Xi (gDCO/m3)	10
Slowly Biodegradable COD - Xs (gDCO/m3)	15
Active heterotrophic biomass - Xbh (gDCO/m3)	500
Active ammonium oxidizing biomass - Xbai (gDCO/m3)	50
Active nitrite oxidizing biomass - Xbaa (gDCO/m3)	50
Particulate inert COD from biomass - Xp (gDCO/m3)	10
Dissolved oxygen - So (gO2/m3)	1
Nitrite - Snoi (gN/m3)	10
Nitrate - Snoa (gN/m3)	10
Ammonium+ammonia - Snh (gN/m3)	10
Soluble organic nitrogen - Snd (gN/m3)	10
Particulate organic nitrogen - Xnd (gN/m3)	10
Particulate inert organic nitrogen - Xni (gN/m3)	10
Volume (m3)	200
Use files	
Save final states (0/1)	0
Load initial states (0/1)	0

- Aerobic heterotrophic yield - Yh (gDCO/gDCO): a real representing the aerobic heterotrophic yield in gDCO/gDCO
- Anoxic heterotrophic yield - Yhd (gDCO/gDCO): a real representing the anoxic heterotrophic yield in gDCO/gDCO
- Ammonium oxidisers yield - Yai (gDCO/gN): a real representing the ammonium oxidisers yield in gDCO/gN
- Nitrite oxidisers yield - Yaa (gDCO/gN): a real representing the nitrite oxidisers yield in gDCO/gN
- Inert fraction of biomass - fp (-): a real representing the inert fraction of biomass
- Nitrogen content of active biomass - ibn (gN/gDCO): a real representing the nitrogen content of active biomass in gN/gDCO
- Heterotrophic max. specific growth rate - muh (1/d): a real representing the heterotrophic maximum specific growth rate in 1/d
- Heterotrophic decay rate - bh (1/d): a real representing the heterotrophic decay rate in 1/d
- Oxygen half-saturation coefficient for heterotrophs - Koh (gO2/m3): a real representing the oxygen half-saturation coefficient for heterotrophs in gO2/m3
- Oxygen inhibition coefficient for heterotrophs - Kad (gO2/m3): a real representing the oxygen inhibition coefficient for heterotrophs in gO2/m3
- Ss half-saturation coefficient for heterotrophs - Ks (gDCO/m3): a real representing the Ss half-saturation coefficient for heterotrophs in gDCO/m3
- Nitrite half-saturation coefficient for heterotrophs - Kno (gN/m3): a real representing the nitrite half-saturation coefficient for heterotrophs in gN/m3
- Nitrate half-saturation coefficient for heterotrophs - Kno (gN/m3): a real representing the nitrate half-saturation coefficient for heterotrophs in gN/m3
- Anoxic growth factor reduction - ng (-): a real representing the anoxic growth factor reduction
- Snh half-saturation coefficient - Knh (gN/m3): a real representing the Snh half-saturation coefficient in gN/m3
- Xs max. specific hydrolysis rate - Kh (gDCO/(gDCO.d)): a real representing the Xs maximum specific hydrolysis rate in gDCO/(gDCO.d)
- Anoxic hydrolysis factor reduction - nh (-): a real representing the anoxic hydrolysis factor reduction
- Ammonification rate - ka (m3.DCO/(g.d)): a real representing the ammonification rate in m3.DCO/(g.d)
- Ammonium oxidisers max. specific growth rate - muai (1/d): a real representing the ammonium oxidisers maximum specific growth rate in 1/d
- Ammonium oxidisers decay rate - bai (1/d): a real representing the ammonium oxidisers decay rate in 1/d
- Ammonium half-saturation coefficient for Ammonium oxidisers - Knai (gN/m3): a real representing the ammonium half-saturation coefficient for Ammonium oxidisers in g2/m3
- Oxygen half-saturation coefficient for Ammonium oxidisers - Koai (gO2/m3): a real representing the oxygen half-saturation coefficient for Ammonium oxidisers in gO2/m3
- Nitrite oxidisers max. specific growth rate - muaa (1/d): a real representing the nitrite oxidisers maximum specific growth rate in 1/d
- Nitrite oxidisers decay rate - baa (1/d): a real representing the nitrite oxidisers decay rate in 1/d
- Nitrite half-saturation coefficient for Nitrite oxidisers - Knaa (gN/m3): a real representing the nitrite half-saturation coefficient for Nitrite oxidisers in gN/m3
- Oxygen half-saturation coefficient for Nitrite oxidisers - Koaa (gO2/m3): a real representing the oxygen half-saturation coefficient for Nitrite oxidisers in gO2/m3

- Mass transfer coefficient - Kla (1/d): a real representing the mass transfer coefficient in 1/d
- Alpha factor(-): a real representing the alpha factor
- Beta factor (-): a real representing the beta factor
- Temperature (Deg. Celcius): a real representing the temperature in Deg. celcius
- Withdrawal flow rate (m3/h): a real representing the flow rate of the withdrawal pump in m3/h
- Soluble inert COD - Si (gDCO/m3): a real representing the initial soluble inert COD concentration of the reactor in gDCO/m3
- Readily biodegradable COD - Ss (gDCO/m3): a real representing the initial readily biodegradable COD concentration of the reactor in gDCO/m3
- Particulate inert COD - Xi (gDCO/m3): a real representing the initial particulate inert COD concentration of the reactor in gDCO/m3
- Slowly Biodegradable COD - Xs (gDCO/m3): a real representing the initial slowly Biodegradable COD concentration of the reactor in gDCO/m3
- Active heterotrophic biomass - Xbh (gDCO/m3): a real representing the initial active heterotrophic biomass concentration of the reactor in gDCO/m3
- Active ammonium oxidizing biomass - Xbai (gDCO/m3): a real representing the initial active ammonium oxidizing biomass concentration of the reactor in gDCO/m3
- Active nitrite oxidizing biomass - Xbaa (gDCO/m3): a real representing the initial active nitrite oxidizing biomass concentration of the reactor in gDCO/m3
- Particulate inert COD from biomass - Xp (gDCO/m3): a real representing the initial particulate inert COD from biomass concentration of the reactor in gDCO/m3
- Dissolved oxygen - So (gO2/m3): a real representing the initial dissolved oxygen concentration of the reactor in gO2/m3
- Nitrite - Snoi (gN/m3): a real representing the initial nitrite concentration of the reactor in gN/m3
- Nitrate - Snoa (gN/m3): a real representing the initial nitrate concentration of the reactor in gN/m3
- Ammonium+ammonia - Snh (gN/m3): a real representing the initial ammonium+ammonia concentration of the reactor in gN/m3
- Soluble organic nitrogen - Snd (gN/m3): a real representing the initial soluble organic nitrogen concentration of the reactor in gN/m3
- Particulate organic nitrogen - Xnd (gN/m3): a real representing the initial particulate organic nitrogen concentration of the reactor in gN/m3
- Particulate inert organic nitrogen - Xni (gN/m3): a real representing the initial particulate inert organic nitrogen concentration of the reactor in gN/m3
- Volume (m3): a real representing the initial volume of the reactor in m”
- Save final states (0/1): Save the final concentrations and volume of the reactor (0:no, 1: yes)
- Load initial states (0/1): Load and use the initial concentrations and volume of the reactor from a file “state.piwat1” (0:no, 1: yes)

Default properties

- always active: yes
- direct-feedthrough: no
- zero-crossing: no
- mode: no
- number/sizes of inputs: 2/16/3
- number/sizes of outputs: 2/16/1
- number/sizes of activation inputs: 0/
- number/sizes of activation outputs: 0/
- continuous-time state:
- discrete-time state:
- name of computational function: *IAPreacteur_PiWaT1*

Interfacing function

```
./macros/IAPreacteur_PiWaT1.sci
```

Computational function (type 1)

```
#include "stdio.h"
#include "stdlib.h"
#include "math.h"

#define  Si x[0]
#define  Ss x[1]
#define  Xi x[2]
#define  Xs x[3]
#define  Xbh x[4]
#define  Xbai x[5]
#define  Xbaa x[6]
#define  Xp x[7]
#define  So x[8]
#define  Snoi x[9]
#define  Snoa x[10]
#define  Snh x[11]
#define  Snd x[12]
#define  Xnd x[13]
#define  Xni x[14]
#define  Vr x[15]

#define Yh rpar[0]
#define Yhd rpar[1]
#define Yai rpar[2]
#define Yaa rpar[3]
#define fp rpar[4]
#define ibn rpar[5]


#define muh rpar[6]
```

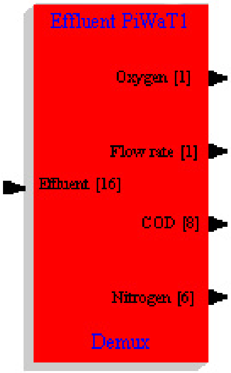
```
#define bh rpar[7]
#define Koh rpar[8]
#define Kad rpar[9]
#define Ks rpar[10]
#define Knoi rpar[11]
#define Knoa rpar[12]
#define ng rpar[13]
#define Knh rpar[14]
#define Kh rpar[15]
#define nh rpar[16]
#define ka rpar[17]

#define muai rpar[18]
#define bai rpar[19]
#define Knai rpar[20]
#define Koai rpar[21]
#define muaa rpar[22]
#define baa rpar[23]
#define Knaa rpar[24]
#define Koaa rpar[25]


#define Alpha rpar[27]
#define Beta rpar[28]

#define Temperature rpar[29]

#define FILE_OUT rpar[31]
#define FILE_IN rpar[32]

#define Qa u1[0]
#define Si_a u1[1]
#define Ss_a u1[2]
#define Xi_a u1[3]
#define Xs_a u1[4]
#define Xbh_a u1[5]
#define Xbai_a u1[6]
#define Xbaa_a u1[7]
#define Xp_a u1[8]
#define So_a u1[9]
#define Snoi_a u1[10]
#define Snoa_a u1[11]
#define Snh_a u1[12]
#define Snd_a u1[13]
#define Xnd_a u1[14]
#define Xni_a u1[15]

#define Feeding u2[0]
#define Aeration u2[1]
#define Withdraw u2[2]

#define Cs Beta*(-0.0000661858*pow(Temperature,3)
+0.0075161751*pow(Temperature,2)-0.3999077443*Temperature+14.58958)

#define process1 muh*(Ss/(Ks+Ss))*(So/(Koh+So))*(Snh/(Knh+Snh))*Xbh
#define process2 ng*muh*(Ss/(Ks+Ss))*(Kad/(Kad+So))*(Snh/(Knh+Snh))
```

```
                                                                *(Snoi/(Knoi+Snoi))*(Snoi/
#define process3 ng*muh*(Ss/(Ks+Ss))*(Kad/(Kad+So))*(Snh/(Knh+Snh))
                                                                *(Snoa/(Knoa+Snoa))*(Snoa/
#define process4 muai*(Snh/(Knai+Snh))*(So/(Koai+So))
                                                                *(Snh/(Knh+Snh))*Xbai
#define process5 muaa*(Snoi/(Knaa+Snoi))*(So/(Koaa+So))
                                                                *(Snh/(Knh+Snh))*Xbaa
#define process6 bh*Xbh
#define process7 bai*Xbai
#define process8 baa*Xbaa
#define process9 ka*Snd*Xbh
#define process10 Kh*Xs*((So/(Koh+So))+nh*(Kad/(Kad+So))*(Snoi/(Knoi+Snoi)
                                                                *(Snoi/(Snoa+Snoi)))+nh*(K
                                                                *(Snoa/(Snoa+Snoi))))
#define process11 process10*(Xnd/Xs)


void IAPreacteur_PiWaT1(int * flag, int *nevprt, double* t, double* xd,
                                double* x, int* nx, double* z, int *nz, double* tvec, int*
             double* rpar, int* nrpar, int* ipar, int* nipar, double* u1,
             int* nu1,double* u2, int* nu2, double* y1, int* ny1, double* y2,
             int* ny2)

{
  float Qs;
  float Kla;
  float temp1,temp2,temp3,temp4,temp5,temp6,temp7,temp8,temp9,temp10,
                        temp11,temp12,temp13,temp14,temp15,temp16;
  FILE *stream;
  char file[]="state.piwat1";



  if (Withdraw==1)
    {
      Qs=rpar[30]*24;
    }
  else
    {
      Qs=0;
    }
  if (Aeration==1)
    {
      Kla=Alpha*rpar[26];
    }
  else
    {
      Kla=0;
    }
 if (*flag==4 && FILE_IN==1)
    {
      /*  Initialisation */
      if ((stream=fopen(file, "r"))==NULL)
        {
          /*printf("Ouverture du fichier impossible");*/
          exit(-1);
```

```
       }
     else
       {
         fscanf(stream, "%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f",&temp1,&temp2,&temp3,&temp4,
                               &temp5,&temp6,&temp7,&temp8,&temp9,&temp10,&temp11,&temp12
                               &temp14,&temp15,&temp16);
         fclose(stream);
         Si=temp1;
         Ss=temp2;
         Xi=temp3;
         Xs=temp4;
         Xbh=temp5;
         Xbai=temp6;
         Xbaa=temp7;
         Xp=temp8;
         So=temp9;
         Snoi=temp10;
         Snoa=temp11;
         Snh=temp12;
         Snd=temp13;
         Xnd=temp14;
         Xni=temp15;
         Vr=temp16;
       }
   }
else
  {
    if (*flag==5 && FILE_OUT==1)
      {
        /*  Ending */
        if ((stream=fopen(file, "w"))==NULL)
          {
            exit(-1);
          }
        else
          {
            fprintf(stream,"%.1f\t%.1f\t%.1f\t%.1f\t%.1f\t%.1f\t%.1f\t%.1f\t
                                       %.1f\t%.1f\t%.1f\t%.1f\t%.1f\t%.1f\t%.1f\t%.1f\n",
                                       y1[4],y1[5],y1[6],y1[7],y1[8],y1[9],y1[10],y1[11],
                                       y1[14],y1[15],y2[0]);
            fclose(stream);
          }

      }
    else
      {
        if (*flag==1||*flag==6)
          {
            y1[0]=Qs;
            y1[1]=Si;
            y1[2]=Ss;
            y1[3]=Xi;
            y1[4]=Xs;
            y1[5]=Xbh;
            y1[6]=Xbai;
            y1[7]=Xbaa;
```

```
             y1[8]=Xp;
             y1[9]=So;
             y1[10]=Snoi;
             y1[11]=Snoa;
             y1[12]=Snh;
             y1[13]=Snd;
             y1[14]=Xnd;
             y1[15]=Xni;
             y2[0]=Vr;
           }
         else
           {
             if (*flag==0)
               {
                 xd[0]=Qa*(Si_a-Si)/Vr;
                 xd[1]=Qa*(Ss_a-Ss)/Vr-1/Yh*process1-1/Yhd*process2-1/Yhd
                                                *process3+process10;
                 xd[2]=Qa*(Xi_a-Xi)/Vr;
                 xd[3]=Qa*(Xs_a-Xs)/Vr+(1-fp)*process6 +(1-fp)*process7
                                                +(1-fp)*process8-process10;
                 xd[4]=Qa*(Xbh_a-Xbh)/Vr+process1 +process2+process3-process6;
                 xd[5]=Qa*(Xbai_a-Xbai)/Vr+process4 -process7;
                 xd[6]=Qa*(Xbaa_a-Xbaa)/Vr+process5 -process8;
                 xd[7]=Qa*(Xp_a-Xp)/Vr+fp*process6 +fp*process7+fp*process8;
                 xd[8]=Qa*(So_a-So)/Vr-((1-Yh)/Yh)*process1
                                                -((3.43-Yai)/Yai)*process4-((1.14-Yaa)/Yaa
                                                + Kla*(Cs-So);
                 xd[9]=Qa*(Snoi_a-Snoi)/Vr-((1-Yhd)/(1.71*Yhd))*process2
                                        +((1-Yhd)/(1.14*Yhd))*process3+1/Yai*process4
                                        -1/Yaa*process5;
                 xd[10]=Qa*(Snoa_a-Snoa)/Vr-((1-Yhd)/(1.14*Yhd))*process3
                                                +1/Yaa*process5;
                 xd[11]=Qa*(Snh_a-Snh)/Vr-ibn*process1 -ibn*process2
                                                -ibn*process3 -(ibn+1/Yai)*process4 -ibn*p
                                                +process9;
                 xd[12]=Qa*(Snd_a-Snd)/Vr-process9 +process11;
                 xd[13]=Qa*(Xnd_a-Xnd)/Vr+(ibn-fp*ibn)*process6
                                                +(ibn-fp*ibn)*process7+(ibn-fp*ibn)*proces
                                                -process11;
                 xd[14]=Qa*(Xni_a-Xni)/Vr+fp*ibn*process6+fp*ibn*process7
                                                +fp*ibn*process8;
                 xd[15]=Qa-Qs;
               }
           }
       }
   }
 return;
}
```

4.7 Effluent PiWaT1 Demux



Library

ASModel

Description

This block allow is a demultiplexer for PiWaT1 effluent. This block allow user to divide the effluent characteristics into 4 output including dissolved oxygen (DO), Flow rate, Chemical Oxygen Demand (COD) and Nitrogen.

Default properties

- always active: no
- direct-feedthrough: yes
- zero-crossing: no
- mode: no
- number/sizes of inputs: 1/16
- number/sizes of outputs: 4/1/1/8/6
- number/sizes of activation inputs: 0/
- number/sizes of activation outputs: 0/
- continuous-time state:
- discrete-time state:
- name of computational function: *PiWaT1_Demux*

Interfacing function

`./macros/PiWaT1_Demux.sci`

Computational function (type 1)

```
#include "stdio.h"


void PiWaT1_Demux(int * flag, int *nevprt, double* t, double* xd,
                        double* x, int* nx, double* z, int *nz, double* tvec,
                        int* ntvec, double* rpar, int* nrpar, int* ipar, int* nipar,
                        double* u1, int* nu1, double* y1, int* ny1, double* y2,
                        int* ny2,double* y3, int* ny3, double* y4, int* ny4)
{
  if (*flag==1||*flag==6)
    {
/* Oxygene */
      y1[0]=u1[9];
/* Flow rate */
      y2[0]=u1[0];
/* DCO */
      y3[0]=u1[1];
      y3[1]=u1[2];
      y3[2]=u1[3];
      y3[3]=u1[4];
      y3[4]=u1[5];
      y3[5]=u1[6];
      y3[6]=u1[7];
      y3[7]=u1[8];
/* AZOTE */
      y4[0]=u1[10];
      y4[1]=u1[11];
      y4[2]=u1[12];
      y4[3]=u1[13];
      y4[4]=u1[14];
      y4[5]=u1[15];
    }
  else
    {
      if (*flag==0)
        {
        }
    }
  return;
}
```