ADERSA

MELISSA

Memorandum of Understanding
ECT/FG/MMM/97.012

ESTEC/Contract N° 12924/98/NL/MV
Contract change notice No 03 of August 2001

TECHNICAL NOTE : 64.4

Control of nitrite of the nitrifying compartment

PRELIMINARY STUDY

Version : 1
Issue : 0

J.-J. LECLERCQ

July 2002

10, rue de la Croix Martre
91873 PALAISEAU Cedex
Téléphone : (33) 1 60 13 53 53
Télécopie : (33) 1 69 20 05 63
Courriel : adersa@adersa.com

adersa

Document Change Log

Version	Issue	Date	Observation
0	0	July 2002	Draft
1	0	July 2002	Original version

TABLE OF CONTENTS

Abbreviations or notations:
NH_3 : ammonia (gaseous or solvated)
NO_2 : nitrite ion
NO_3 : nitrate ion
N_s : Nitrosomonas
N_b : Nitrobacter

Note :
In a figure, a graph ij is located by its row i and its column j.

1. INTRODUCTION

The function of the nitrifying compartment is the transformation of ammonia into nitrate. During one step of this transformation, nitrite which is a poison for man and a lot of bacteria is produced. As the on line measurement of nitrite is impossible (a sensor is not yet available), an estimator has been built and is on the way to be validated (a first validation has been done in TN 48.3). The aim of the control of nitrite, which is based on the estimator result, is to minimize the production of this component at the column output. Control of nitrite is obtained by acting on the input flow.
The present note describes :

- the principle of the control;
- how it has been built and checked at each step;
- the tools that have been elaborated to check the programme of the control all along its future evolution.

2. PRINCIPLE OF THE CONTROL

The model based control lies on a reduced model of the nitrifying column built by LGCB. At each sampling period, the control computes the predicted behaviour of the nitrite concentration on a horizon H in the future (scenario method). The first point of the scenario is the measurements at the column output, except for nitrite and biomass for which the estimations replace the measurements. The input data of the internal model are the measurements at column input in the gas and liquid streams. They are supposed constant all along the scenario duration H.
An iterative algorithm computes the liquid flow rate that fits the maximum NO_2 of horizon H on the NO_2 constraint. This control computed flow rate is compared to the requested flow rate (flow rate fixed by the operator) and is applied to the process only if it is the lower of the two.

The bias that occurs inevitably because of process/model mismatch is cancelled by a bias estimator.
In detail, the 17 measurements at each sampling period follow :

- gas and liquid flow rates;
- O_2, CO_2 and NH_3 (if any) concentrations of the gas input stream;
- O_2, CO_2 , NH_3 , NO_3 , SO_4 and PO_4 concentrations of the liquid input and output streams.

3. INTERNAL MODEL OF THE CONTROL

3.1. Notation

The notation comes from TN 44.2 and is recalled hereafter, for each substrate of the problem and for each tank of the column :

a : molar concentration in the gas phase

b : molar concentration in the liquid phase of the molecular form

c : molar concentration at the thermodynamic equilibrium

d_G : molar concentration in the incoming gas flow

d_L : molar concentration in the incoming liquid flow of the molecular form

q_G : gas flow rate (1/h)

q_L : liquid flow rate (1/h)

r : mean volumetric production or consumption rate (mol/1/h)

K : volumetric transfer coefficient in liquid phase (notation K_{La} in TN 27.1)

k : dissociation constant of acid/base equilibrium

In the liquid phase, the concentration of a molecular form, x, and the one of its ionic form x' are linked by the relation which implies the dissociation constant k :

$$x' = k \cdot x$$

with $k = \frac{K_b . [H^+]}{K_e}$ for NH_3 solvated

$k = 0$ for the other compounds

3.2. Recall of general equations for any tank of the column

The following equations have been established in the Technical Note 44.2 related to the nitrifying compartment.

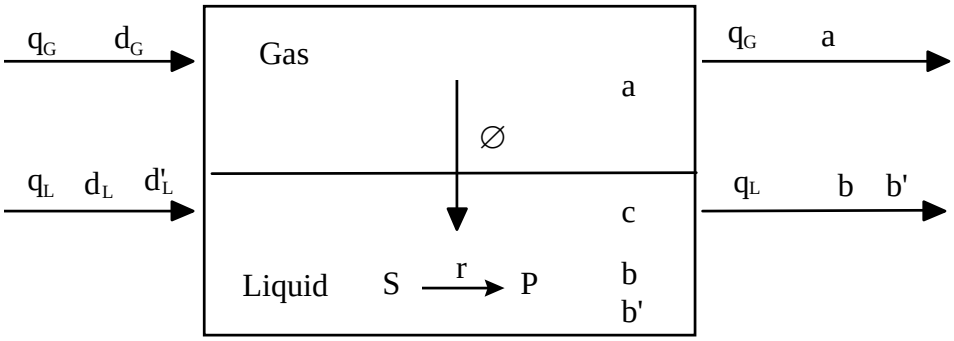

Figure 1 : General flow sheet of a tank
(S = substrate , P = product)

For a mono-phase substrate, the mass conservation law, in the liquid phase of an ideally stirred tank, leads to :

$$(1 + \tau_L \cdot p)\, b = \tau_L \cdot r + d_L \qquad (1)$$

with $\tau_L = \dfrac{V_L}{q_L}$

p : Laplace variable

For a bi-phase substrate, the mass conservation law, in the liquid and gas phases of an ideally stirred tank, leads to (relations A2.7 and A2.8 of TN 44.2) :

$$(1 + \tau_1 . p)\, b = G_1 . d_L + G_2 . d_G + G_3 . r \qquad (2)$$

$$a = \alpha_4 . b + \alpha_5 . d_G \qquad (3)$$

3.3. Internal model of the control

For the control, the internal model is based on the state given by the estimator : the nitrite and biomass concentrations of an equivalent column composed of only one tank.

So the state system established in TN44.2 (equation 15) has been simplified and adapted to an equivalent column where the fixed bed B is represented by only one ideally stirred tank and where the parts A and C of the column have a null volume :

$$\begin{cases} \dot{X} = A . X + B . U_0 + E . J . C_x \\ Y = C . X + D . U_0 \end{cases} \qquad (4)$$

The matrices A, B, C, D and E are computed by the routine ‘statesysim’ (see in annex 2) and the matrix of the limiting terms J is computed by the routine ‘irate’ (same annex).
U_0 is the vector of the measured inputs of the process (concentrations of the substrates in the gas and liquid streams) and C_x is the vector of biomass N_s ans N_b estimated at each sampling period.

3.4. Comparison: internal model of control / model of process

Four tests have been realized to compare the behaviour of the internal model versus the behaviour of the process : 2 steps of ammonia at low and high concentration and 2 steps of flow rate (Table 1). For each test, the step is done at time t=5 h.

Test	Total NH3 (mol/l)	Flow rate (l/h)
4	$2.1\ 10^{-3}$ (low load)	step from .77 to 1.16 (+50%)
3	step from $2.1\ 10^{-3}$ to $3.15\ 10^{-3}$	.77
2	$2.1\ 10^{-2}$ (high load)	step from .77 to 1.16 (+50%)
1	step from $2.1\ 10^{-2}$ to $3.15\ 10^{-2}$	.77

Table 1 : Specific conditions of the tests

The figure 3 shows the behaviour of NO_2 at column output on a duration of 100 h.
The comparison with the internal model is done on the following procedure (figure 2):

- The NO_2 and the biomass are estimated by means of the estimator all along the simulation.
- The simulation horizon is sampled each every 5 h from 0 to 100 h (which gives 21 samples).
- At each sample, the internal model is launched on the horizon H of the scenario. Its input data are the process measurements and the estimations of this sample. The behaviour of the NO_2 of this horizon H is compared to the NO_2 process that follow the sample during the duration H.

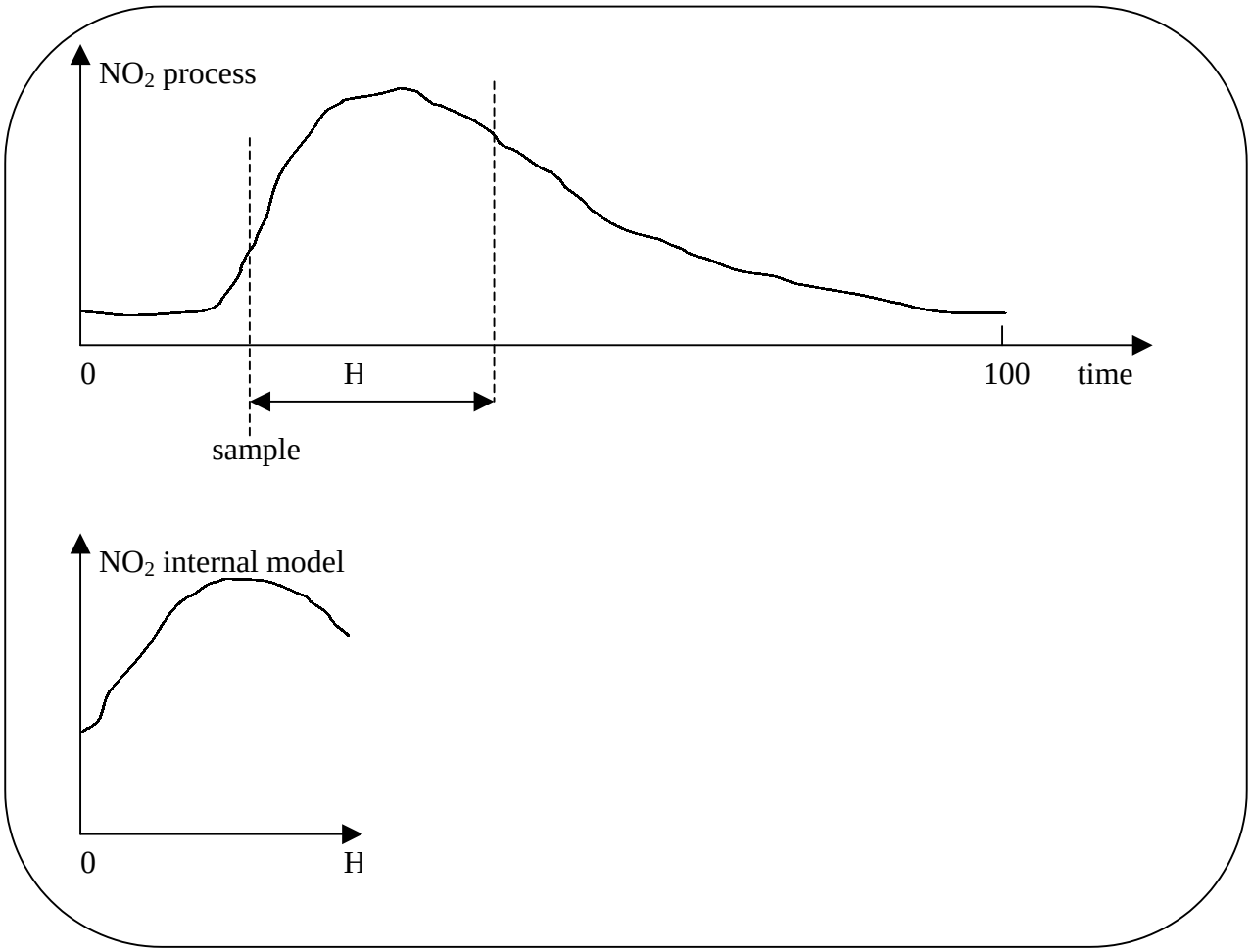

Figure 2 : Comparison internal model / process.
If the internal model is equal to the process, the 2 curves are the same on H

The figures 4 to 7 show the NO_2 of the internal model for the tests 4 to 1 (of Table 1), respectively, on the horizon H (20 h). It can be observed that for low load (figures 4 and 5), the NO2 of the internal model is nearly equal to the process one when, for high load (figures 6 and 7), it is 50 % greater. This is an advantage for the control because the overestimation of

NO_2 will under-estimate the controlled flow rate, which will reduce the risk of overflowing the constraint.

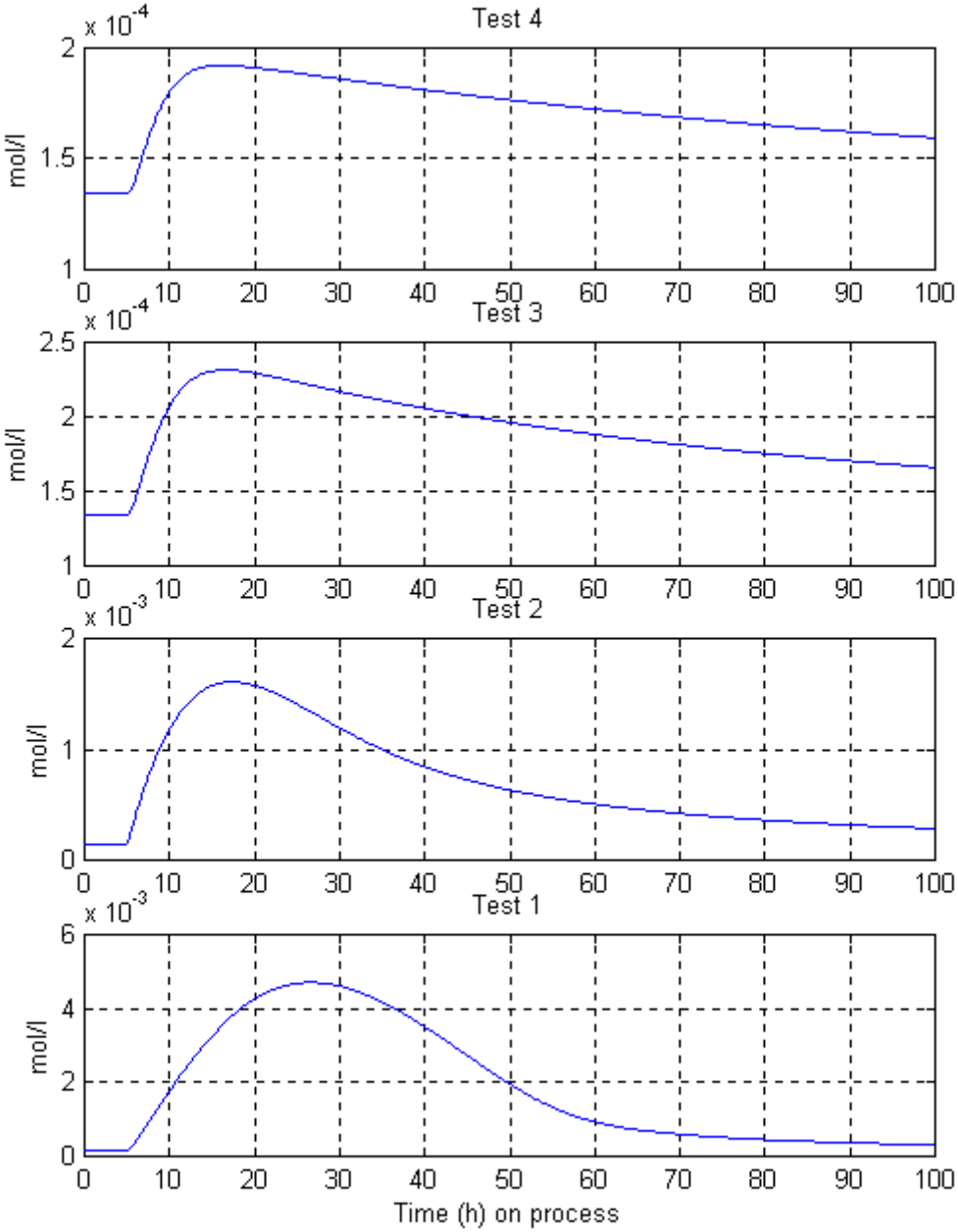

Figure 3 : NO2 of the process for the tests of Table 1

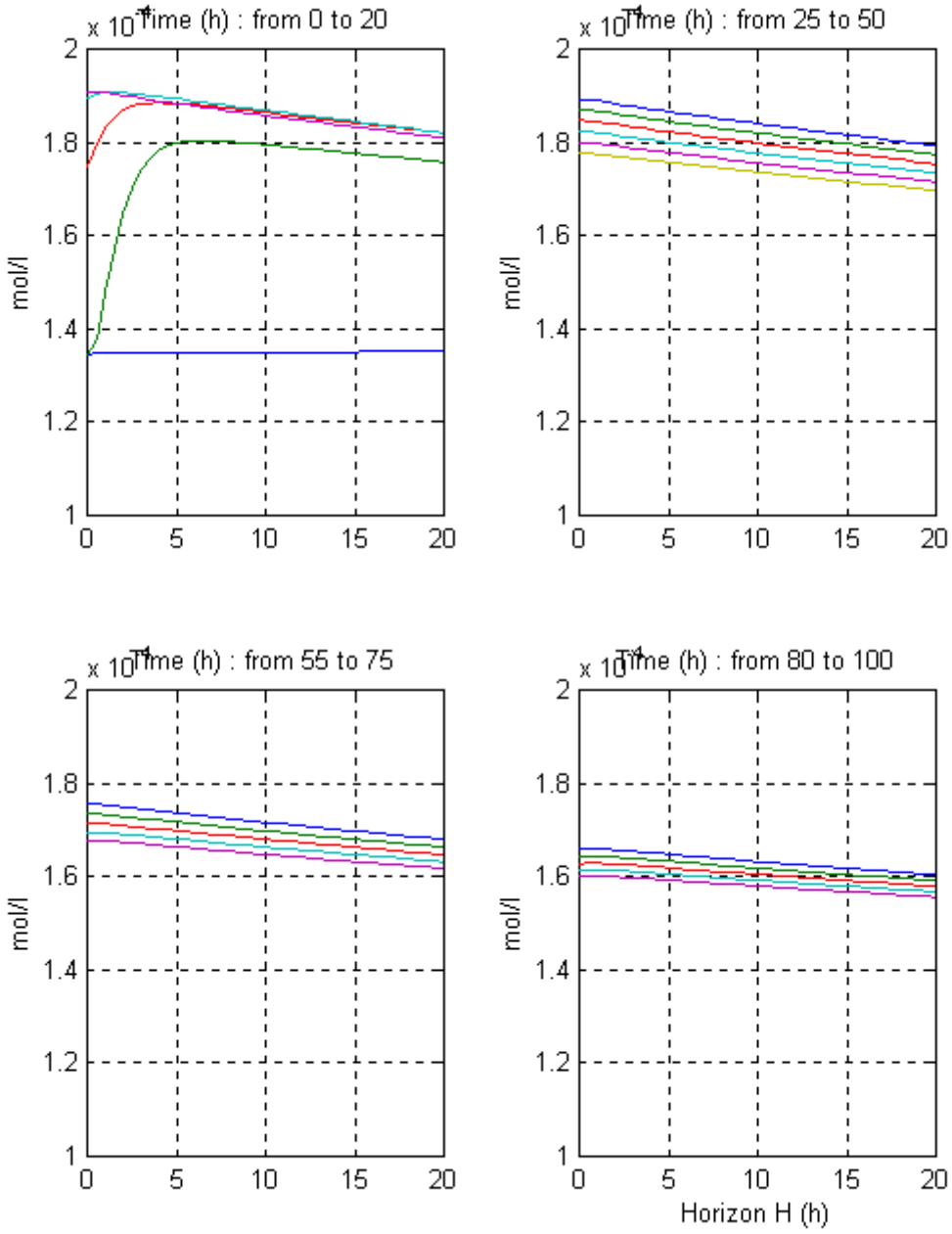

Figure 4 : NO2 of the internal model for the test 4

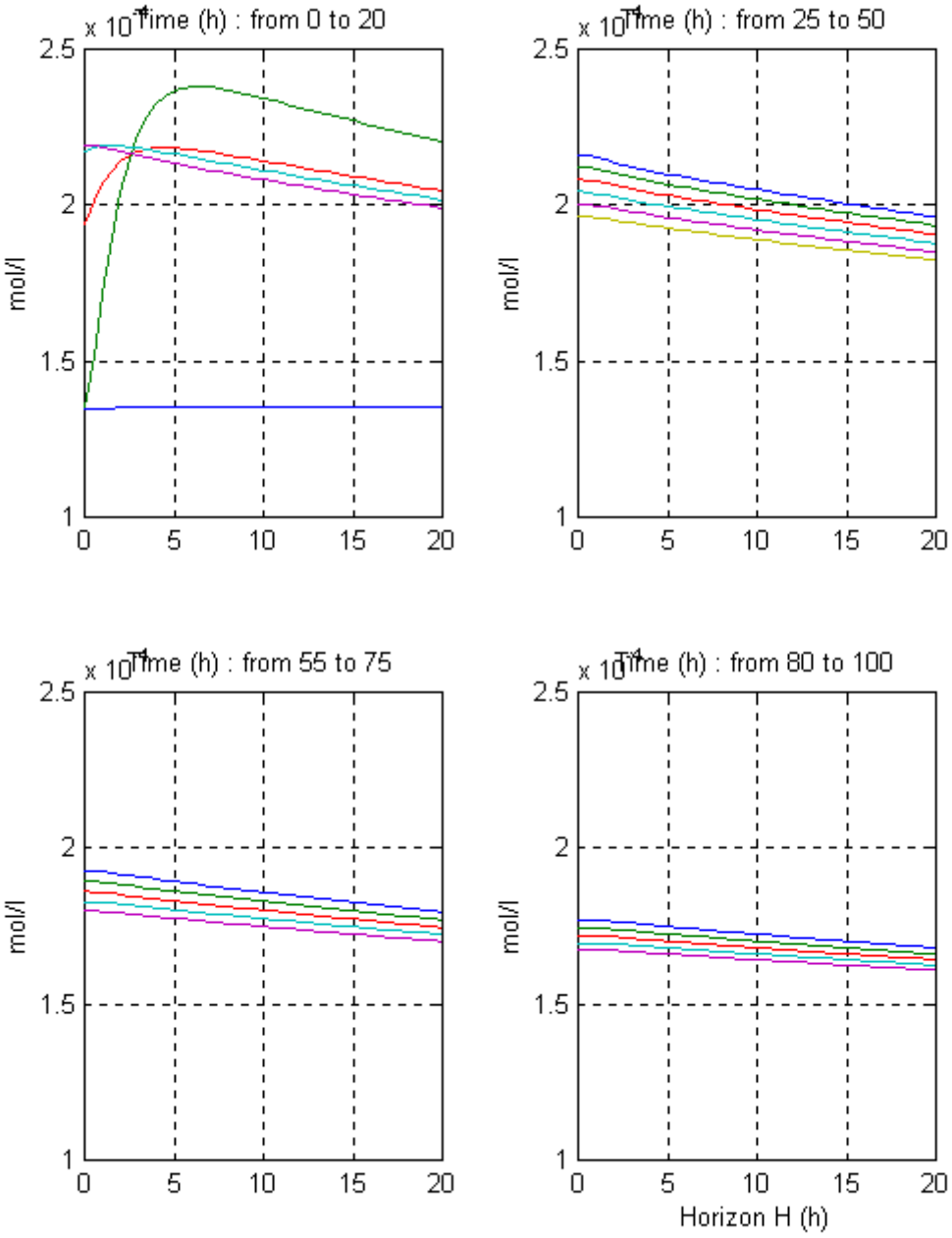

Figure 5 : NO2 of the internal model for the test 3

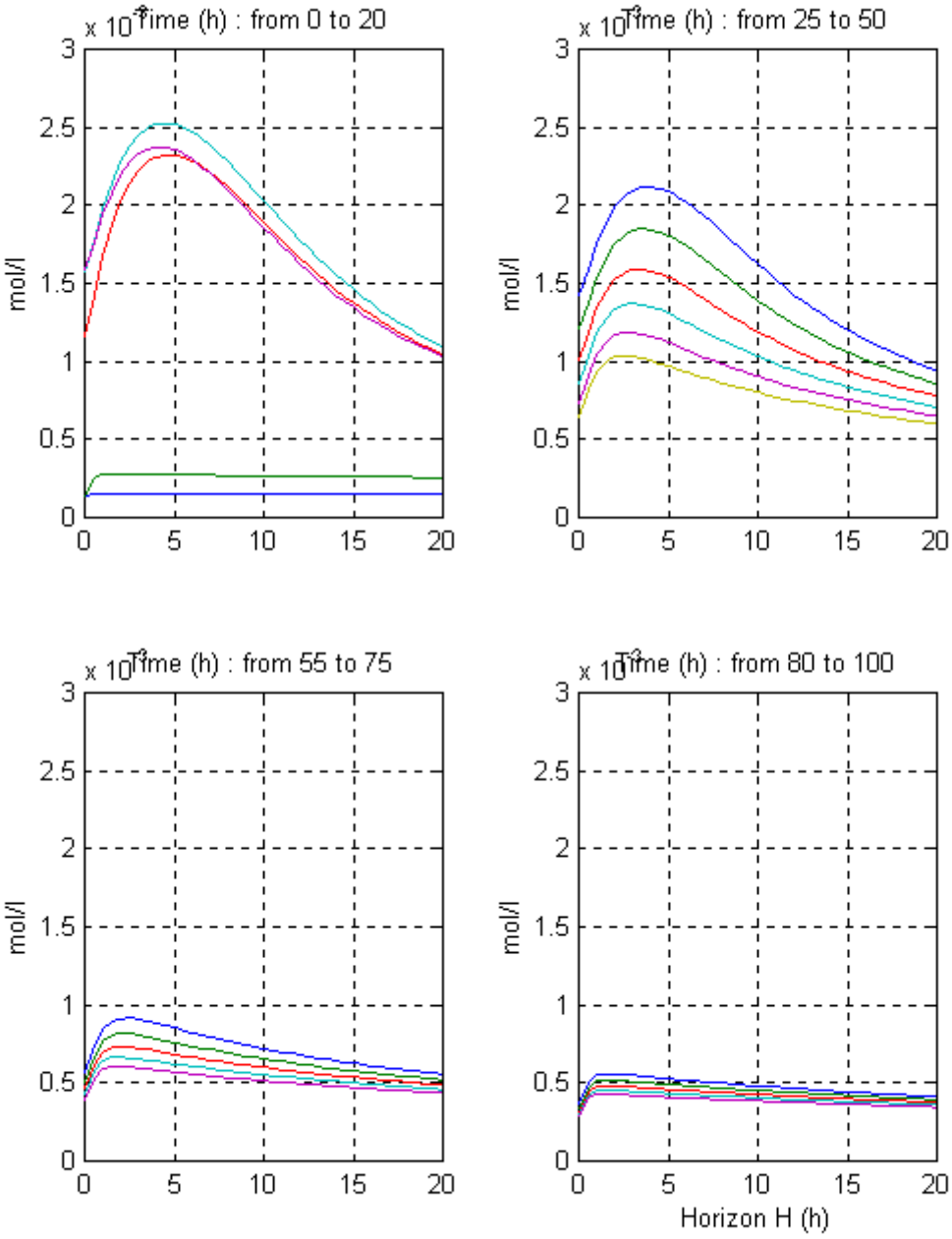

Figure 6 : NO2 of the internal model for the test 2

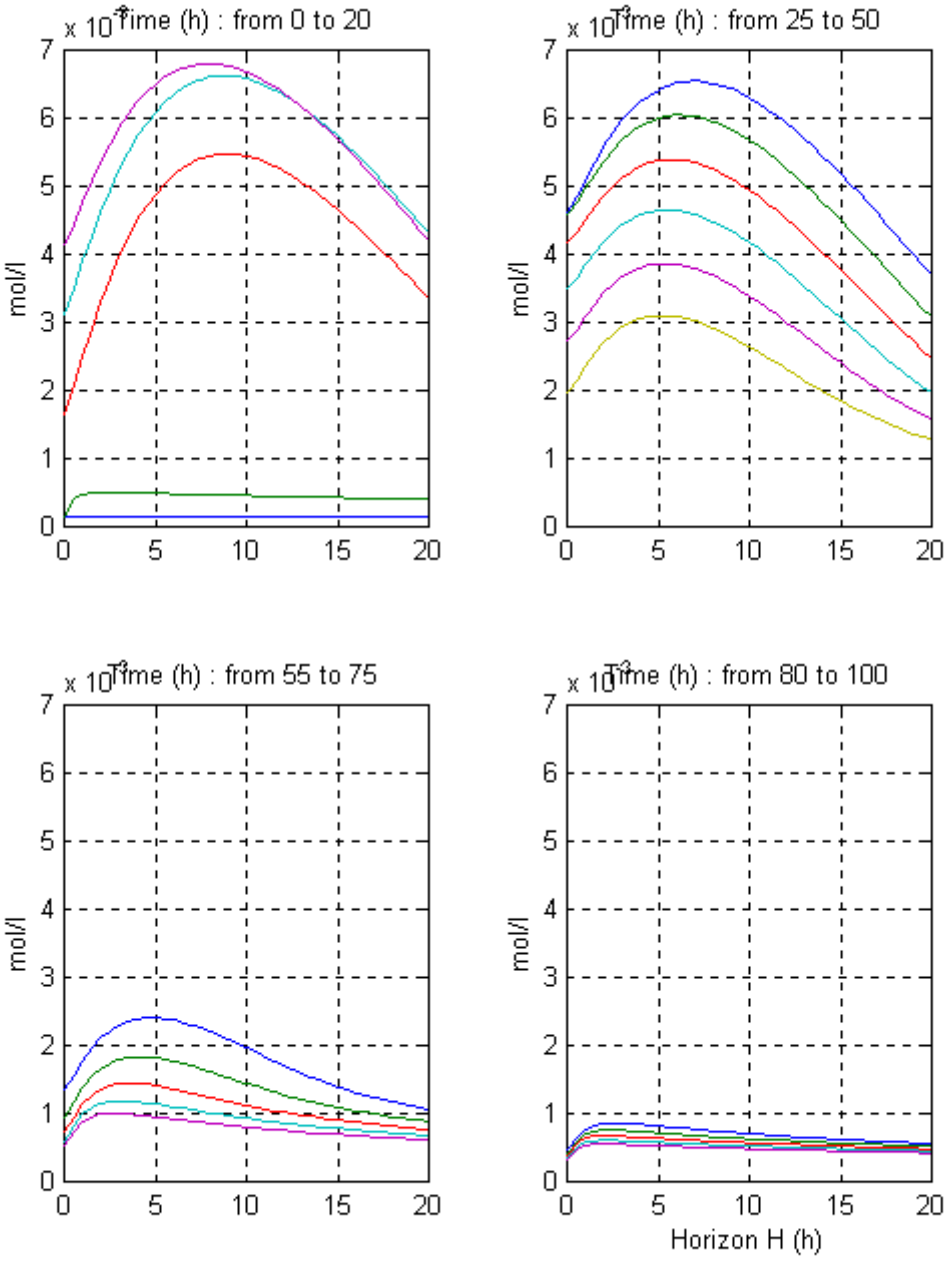

Figure 7 : NO2 of the internal model for the test 1

4. SIMULATOR OF THE CLOSED LOOP SYSTEM

The simulator of the closed loop system (process bound to the control) runs under Simulink® (figure 8).
The vectors **dG** and **dL** are the concentrations of substrates in gas and liquid streams, respectively, at column input.
Fint is the time profile of the flow rate requested by the operator (or the upper level control, if any).
spNO2t is the time profile of the max constraint of NO2 at column output.
The **process data** are all the data measured on the process and necessary to the control.

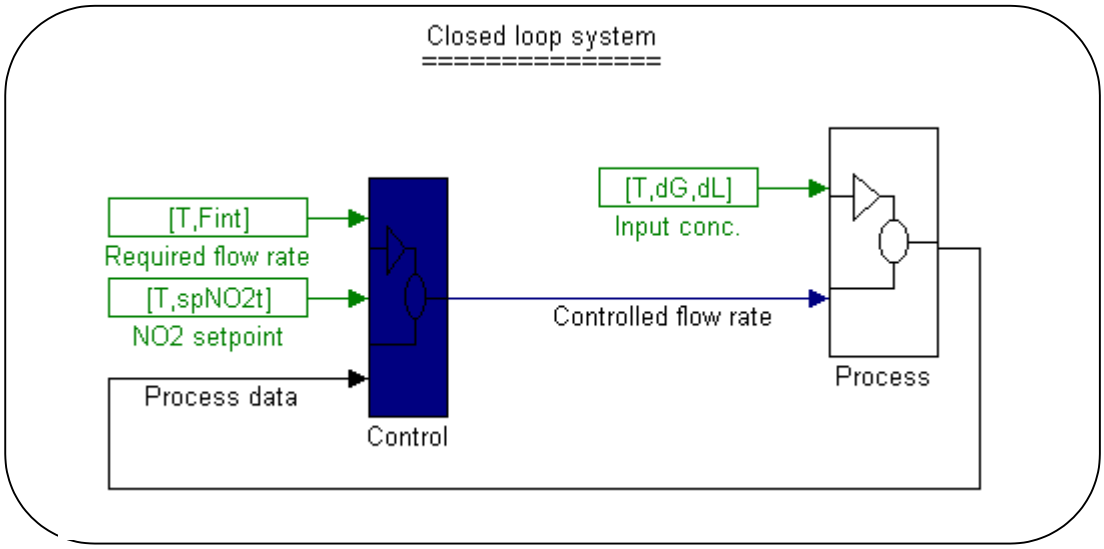

Figure 8 : Simulator of the closed loop system

5. EXAMPLES OF CONTROL SIMULATION

In the two following examples, the parameters of the process and the control are identical; there is no process/control mismatch (this aspect will be studied in future technical note). The parameters are detailed in the initialization file 'i_sim_3' given in annex 1.
In the 2 simulations, the max constraint of NO_2 is the same : $3\ 10^{-4}$ mol/l .
In the former the operator is supposed to apply a positive step of flow rate : the initial flow rate of .77 l/h is multiplied by 1.5 at time t=2 h.
In the later, a step of ammonia occurs at time t=2 h : initial concentration of total ammonia ($2.1\ 10^{-3}$ mol/l) multiplied by 1.5 .
All the other variables of the process are time constant.
The control has to manage the input flow rate of the column so that the max constraint of NO_2 is never overshot.

The figures 9 and 10 show these 2 simulations. The disturbances (requested flow step of example 1 and ammonia load step of example 2) are plotted on the 2 bottom graphs.

The top graph shows the NO_2 process versus its max constraint which is never overshot.

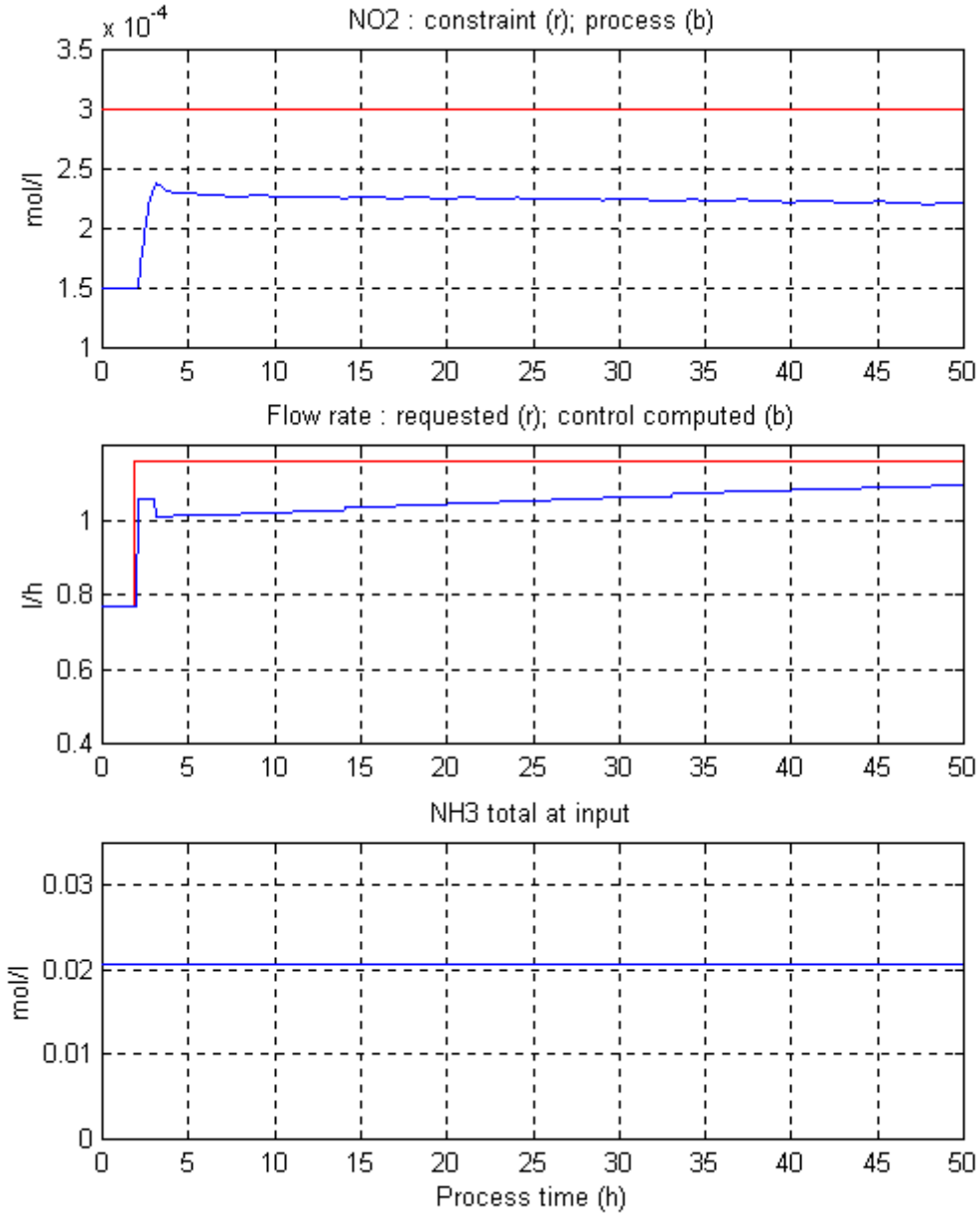

Figure 9 : Step of flow rate requested by operator

The example 2 (fig 10) shows clearly a bias : beyond time t=30 h, the controlled flow rate remains at constant distance (about 0.1 l/h) below the requested flow when the max constraint is not reach (and moreover decreasing). This phenomenon is due to the scenario method where the internal model can never be exactly identical to the process. This permanent distance will be cancelled by a bias estimator in the future version of the control.

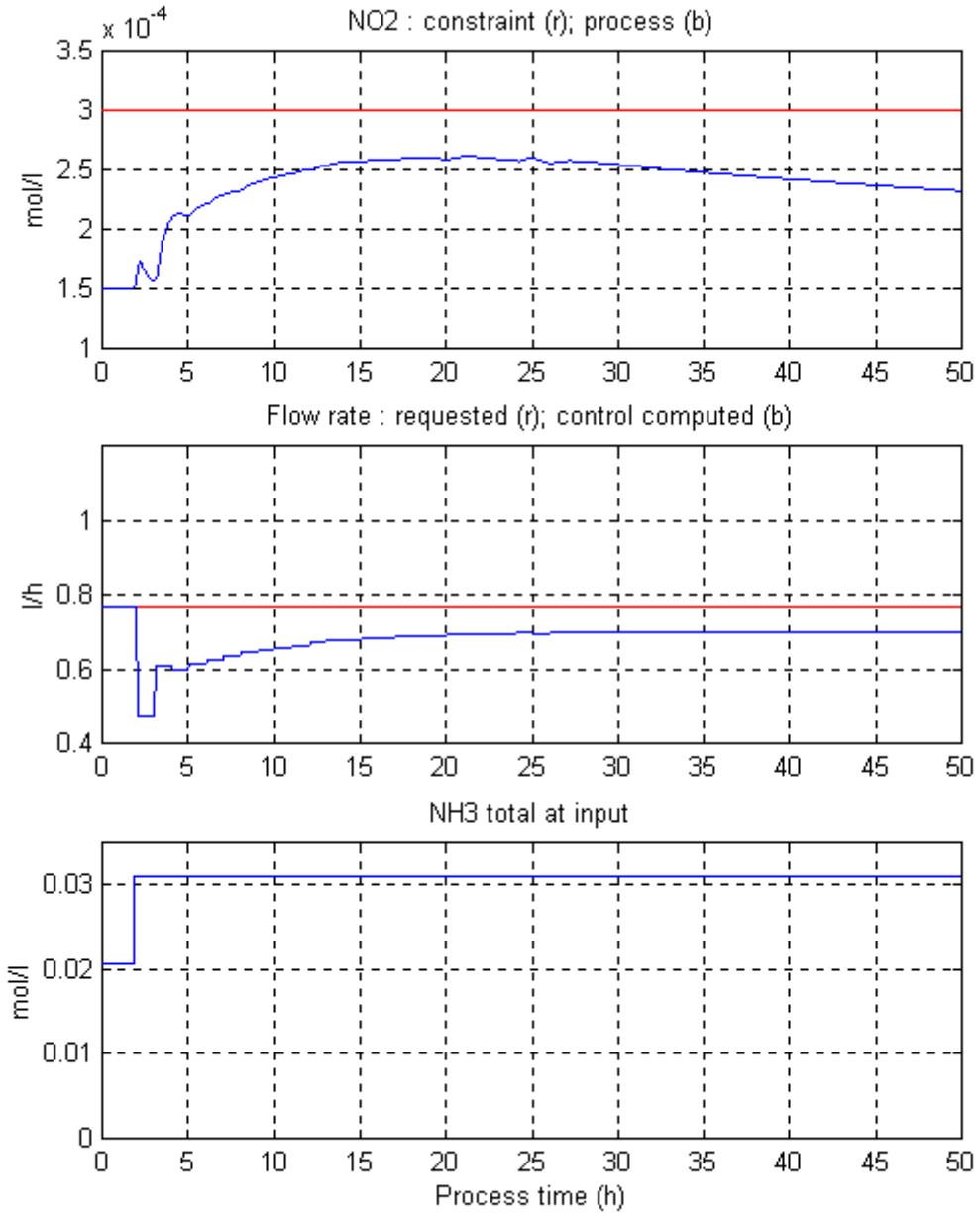

Figure 10 : Step of ammonia load

6. TOOLS FOR DEBUGGING

Due to the complexity of the software, a few tools have been developed to check its good functioning during its elaboration and will be of great interest for its future evolution.
One of these tools is the plotting of the NO_2 estimated versus the NO_2 simulated at column output (figure 11). A similar one concerns the biomass. Another computes the manipulated variable without taking into account the estimations (that are then replaced by measurements).

The software has been written in a modular way. The estimation and the computation of the manipulated variable (input flow rate) are done independently : each part can be changed or modified without consequence on the other (excepted however if the state system deduced from estimator were modified).

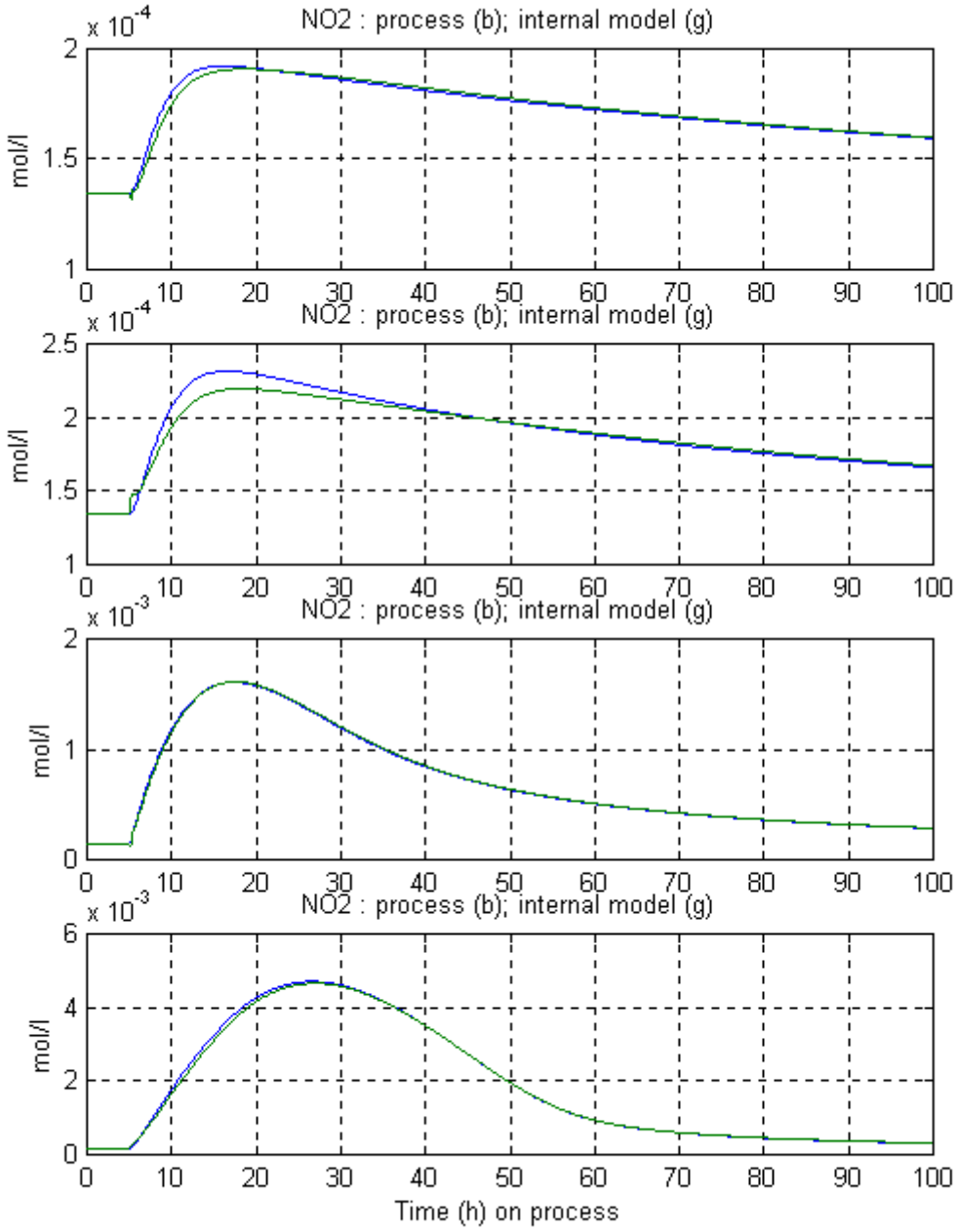

Figure 11 : Comparison NO_2 process (blue) / NO_2 estimator (green) for tests Table 1

7. CONCLUSION

The control is now built in Matlab language.
Before improving the software it is necessary to validate the estimator on which the internal model of the control lies.
The future work is the elaboration of the bias estimator and then the robustness study.
At the end the software will be translated in C language.

8. REFERENCES

LECLERCQ J.-J. : "Nitrite and Biomass predictors of the nitrifying compartment – Phase I : Mathematical inferred variable". Contract ESTEC n° 12924/98/NL/MV, Contract change notice n^{o} 1 of May 1999, TN 48.1.

POUGHON L. "Review of models and basis of a dynamic structured model of the nitrifying compartment". ESTEC contract PRF 151739, February 1996, TN 27.1.

POUGHON L. "Description of the nitrifying column model and first simulations". ESTEC contract PRF 151739, May 1996, TN27.2

RICHALET J. 1993. Pratique de la Commande Prédictive. HERMES Edition.

9. ANNEX 1 : PARAMETERS OF THE PROCESS

Initialization file : 'i_sim_3' :

```
%**************************************************************************
%       Nitrifying column                                                 *
%       Version 2.4     May 2002                                          *
%                                                                         *
%       State system according to TN 44.2                                 *
%          . dissociation of CO2 and NH3                                  *
%          . column = NB+2 CST's in series                                *
%                                                                         *
%       Modifications in March 2000                                       *
%          . vector KlNs and KlNb moved into scalar                       *
%          . Monod type law for maintenance in 'irate.m'                  *
%                                                                         *
%       i_sim_3.m   Initialization of the process model                   *
%                                                                         *
%**************************************************************************

arret = 0;

% Column parameters
%------------------
    % 1.Parameters of the previous column (until TN 48.2)
        NB = 5;         % number of stirred tanks in part B (fixed bed)
        if (typtst == 0 | typtst == 10 | typtst == 11 | typtst == 12), NB=1; end
        VA = 1.48;      % volume of part A (l)
        VB = 6.17;      % volume of part B (l)
        VC = 0.45;      % volume of part C (l)
        epsL = .33;     % volume ratio of liquid
        epsG = .04;     % volume ratio of gas
        epsT = epsG + epsL;
```

```
        Gin = .03*60;   % incoming gas volumetric rate (l/h)
        Fin = .0028*60; % incoming liq volumetric rate (l/h)
        RG = 1*99.0;    % recycling ratio of gas
        RL = 1*6.42;    % recycling ratio of liquid
        fG = 0;         % ratio of backward flow in a tank (gas phase)
        fL = 0;         % ratio of backward flow in a tank (liquid phase)

    % 2.Parameters of the column for TN 54.2
        Vmin = 31.7;    % (l) volume min of the part B (fixed bed)
        Vmin = 31.7/5.5;% (l) volume min of the part B for NO3 production = .012 mol/h
(october 2001)
        coef_V=Vmin/epsL/VB;% multiplicative coef. of the vol of previous column
        VA = VA*coef_V; % adapted volume of part A (l)
        VB = VB*coef_V; % adapted volume of part B (l)
        VC = VC*coef_V; % adapted volume of part C (l)
        Gin = Gin*coef_V;% adapted incoming gas volumetric rate (l/h)
        Fin = Fin*coef_V;% adapted incoming liq volumetric rate (l/h)
        % modif. of the L and G flow rates to fit the flow of Spirulina Compart
        coef_L = max(0,(1+RL)*Fin/Fin_3 - 1);
        Fin = Fin_3;                  % l/h
        coef_G = max(0,(1+RG)*Gin/Gin_4 - 1);
        Gin = Gin_4;       % l/h
        % modif. of the recycling ratio RG to keep constant '(1+RG)*Gin'
        RG = RG*coef_G;
        % modif. of the recycling ratio RL to keep constant '(1+RL)*Fin'
        RL = RL*coef_L;

        VnB = VB / NB;  % volume of an equivalent stirred tank

        Temp = 303;  % K (TN35.1 ADERSA)
        pH = 8;      % TN35.1 ADERSA
% general parameters
% ------------------
        NG = 3;         % number of bi-phases substrates (in the L and G phases)
        NL = 4;         % number of mono-phase substrates in the liquid phase
        NS = 2;         % number of strains of bacteria (Ns and Nb)
        NX = NG+NL+2*NS;% nb of coef of state vector Xi for any tank
        NO = NX+NG;     % nb of outputs for any tank
        NI = 2*NG + NL;
        NV = NI * (NB + 2);

% kinetic parameters
% ------------------
        Nls = 4;                        % nb of limiting substrates
        Nis = 2;                        % nb of inhibitory substrates

        %          Ns         Nb
        mumax = [5.7e-2 ;   3.6e-2];    % max specific growth rate (1/h)
        maint = [3.38e-3;   7.92e-3];   % maintenance coeffcient (1/h*(mol/gbio))

        KlNs  = [5.05e-6           ;    % limiting substrate for growth : O2   (mol/l)
                 6.625e-5          ;    % limiting substrate for growth : NH3  (mol/l)
                 1e-10            ];    % no limitation with HCO3- for growth  (mol/l)

        KlNb  = [           1.7e-5 ;    % limiting substrate for growth : O2   (mol/l)
                            3.6e-4 ;    % limiting substrate for growth : NO2- (mol/l)
                            1e-10 ];    % no limitation with HCO3- for growth  (mol/l)

        In =    [1e30       1e30   ;    % inhibitory substrate : NO2- (mol/l)
                 1e30       1e30]  ;    % inhibitory substrate : NO3- (mol/l)

        KmNs  = KlNs(1:2,1);         % limiting substrate for maintenance : O2; NH3   (mol/l)
        KmNb  = KlNb(1:2,1);         % limiting substrate for maintenance : O2; NO2-  (mol/l)

% stoechiometric parameters
% -------------------------
                                        % inverse of yield of biosynthesis :
                                        % (mol substr./mol biomass)
        Yx1  =  [-5.4269   -6.5106 ;    % O2
                 -1        -1      ;    % HCO3-
                 -4.5341   -0.1994 ;    % NH3
                 4.3347    -15.1714;    % NO2-
```

```
                0          15.1714;    % NO3-
                -0.0089   -0.0089 ;    % HPO4--
                -0.0035   -0.0035];    % SO4--
              % 4.3099    -0.0248 ;    % H+   (unused for simplification)
              % 1           1     ];    % HO-  (unused for simplification)

                                        % inverse of yield of maintenance :
                                        % (mol substr./mol maintenance substr.)
        Ym1  =  [-1.5      -0.5    ;    % O2
                 0          0      ;    % HCO3-
                 -1         0      ;    % NH3
                 1         -1      ;    % NO2-
                 0          1      ;    % NO3-
                 0          0      ;    % HPO4--
                 0          0     ];    % SO4--

        % change of unit of Yx1 :
        Scompn = [1     1.6147 .3906 .1994 .0035 .0089]; % stoechio composition biomass nitri
        M_nitri = Scompn*Matom;
        Yx1 = Yx1 / M_nitri; % (mol substrate / g biomass)

        Yx(1,1) = 1/Yx1(3,1);   % yield for (global) growth of Ns (g X/mol NH3)
        Yx(2,1) = 1/Yx1(4,2);   % yield for (global) growth of Nb (g X/mol NO2-)

% gas/liquid transfer parameters
% ------------------------------
        KLa = 1*[51; 51; 500];    % O2 CO2 NH3   (1/h)
% Physico-chemical constants
% --------------------------
        [kpart,Kab,ksi_bid] = i_physic(Temp,pH);
% partition coefficients
        kpartN = kpart([2,4,6],1);   % O2 CO2 NH3 (T = Temp K,non ionic form)
% dissociation coefficient (for NH3)
        Kb = Kab(3,1);          % basicity cst at T = 'Temp' K
        Ke = Kab(4,1);          % ionic product at T = 'Temp' K
        KNH3 = Kb*10^(-pH)/Ke;  % [NH4+] = KNH3 * [NH3]solvated
% dissociation coefficient for CO2
        KCO2 = Kab(1,1)/(10^(-pH))*(1+Kab(2,1)/(10^(-pH))); % [HCO3-]+[CO3--] = KCO2 *
[CO2]solvated
% dissociation coefficient (for H3PO4)
% ------------------------------------
        Ka = 6.166e-8;          % (at T=298K, from TN 27.2,LGCB)
        xx = 1e-8/Ka;   KPO4 = 1+xx;       %[PO4]total = f([HPO4=])
% dissociation vector for the biphase substrates
% ----------------------------------------------
        Kdis = [0; KCO2; KNH3]; % O2  CO2  NH3
% liquid/gas thermodynamical equilibrium constants
% ------------------------------------------------
        VM = 22.4 * 303 / 273;  % molar volume (l/mol)
        alpha = kpartN / 55.56 / VM;
% Computation of the state system of the nitrifying compartment
%--------------------------------------------------------------
    [Ae_3,Be_3,Ce_3,De_3,E_3] = stasys_3(NG, NL, NB, Gin, Fin, RG, RL, fG, fL, ...
                                         VA, VnB, VC, epsL, epsG, epsT, ...
                                         alpha, Kdis, KLa);
% Initialization of vector of indices
%------------------------------------
    iO2  = 1:NG+NL:(NB+1)*(NG+NL)+1; % indices of O2 L in the state vector
    iCO2 = iO2  + 1;                 % indices of HCO3- in the state vector
    iNH3 = iCO2 + 1;                 % indices of NH3 L in the state vector
    iNO2 = iNH3 + 1;                 % indices of NO2 in the state vector
    iNO3 = iNO2 + 1;                 % indices of NO3 in the state vector
    iPO4 = iNO3 + 1;                 % indices of PO4 in the state vector
    iSO4 = iPO4 + 1;                 % indices of SO4 in the state vector
    iSub = 1:(NG+NL)*(NB+2);         % ind. of subs. in state vector
    iXNs = 1 : 2*NS : 2*NS*(NB+1) + 1;
    iXNs = iXNs + (NG+NL)*(NB+2); % ind. of act. biomass Ns in the state vector
    iXNb = iXNs + 1;              % ind. of act. biomass Nb in the state vector
    iXag = (NG+NL)*(NB+2) + 1 :(NG+NL+2*NS)*(NB+2); % ind. of active and global
    indG0 = 1:NG;          indG = indG0;
    indL0 = NG+1:2*NG+NL; indL = indL0;
    for ii = 1:NB+1
```

```
      indG = [indG, indG0+ii*NI];   % ind. of Gas concen. in the output vector
      indL = [indL, indL0+ii*NI];   % ind. of Liq concen. in the output vector
    end
% Noise
%------
        % 1. Weight vector of the noise on the derivative of the state vector
        WX0 = 0*5*[0    0    1e-6   1e-6    1e-6  0    0];
        WX = WX0;
        for ii = 1:NB+1
          WX = [WX, WX0];
        end

        % 2. Weight vector of the noise on the output vector
        %WYG0 = 0*5*[0    0    0 ];
        %WYL0 = 0*5*[0    0    1e-5/(1+KNH3)   0    1e-5  0    0];
        %WYG = WYG0;
        %WYL = WYL0;
        %for ii = 1:NB+1
        %  WYG = [WYG, WYG0];
        %  WYL = [WYL, WYL0];
        %end
        WYG = zeros(1, NG*(NB+2));
        WYL = zeros(1, (NG+NL)*(NB+2));
        if typtst == 0 % noise added on part B of the column where measurements are supposed
to be done
          WYG0 = 0*2*[0    0    0 ];
          WYL0 = 0*2*[0    0    1e-5/(1+KNH3)   0    1e-5  0    0];
          WYG(1,NG*NB+[1:NG]) = WYG0;
          WYL(1,(NG+NL)*NB+[1:NG+NL]) = WYL0;
        else % noise added on part C of the column
          WYG0 = 0*5*[0    0    0 ];
          WYL0 = 0*5*[0    0    1e-5/(1+KNH3)   0    1e-5  0    0];
          WYG(1,NG*(NB+1)+[1:NG]) = WYG0;
          WYL(1,(NG+NL)*(NB+1)+[1:NG+NL]) = WYL0;
        end
% Checking the mass balance
%--------------------------
        % 1. Matrices of stoechiometric composition of each compounds :
        % atom : C         N         S         P
        MSG = [  0         0         0         0       ;  % O2  Gas
                 1         0         0         0       ;  % CO2 Gas
                 0         1         0         0       ]; % NH3 Gas
        MSL = [  0         0         0         0       ;  % O2
               1+Kdis(2)   0         0         0       ;  % CO2 L and ionic forms
                 0      1+Kdis(3)    0         0       ;  % NH3 L and ionic forms
                 0         1         0         0       ;  % NO2
                 0         1         0         0       ;  % NO3
                 0         0         0         1       ;  % HPO4
                 0         0         1         0      ];  % SO4
        MSS = [Scompn(:,[1,4,5,6]);Scompn(:,[1,4,5,6])];  % biomass Ns and Nb
        MSS = MSS ./ ([M_nitri;M_nitri]*ones(1,4));       % CNSP atom/g biomass
        MSSt = Scompn ./ (M_nitri*ones(size(Matom')));    % CHONSP atom/g biomass
        % 2. matrices of volume of tanks
        Vg = [ VA*epsG/epsT;
               VnB*epsG*ones(NB,1);
               VC*epsG/epsT];
        Vl = [ VA*epsL/epsT;
               VnB*epsL*ones(NB,1);
               VC*epsL/epsT];
% Saving the specific variables of the Nitrifying compartment
% ============================================================
NL_3 = NL;     NG_3 = NG;     NS_3 = NS;
NB_3 = NB;     NX_3 = NX;     NO_3 = NO;     NI_3 = NI;     NV_3 = NV;
Kdis_3 = Kdis;
Fin_3 = Fin; Gin_3 = Gin; RG_3 = RG;     RL_3 = RL;
fG_3 = fG;     fL_3 = fL;     VA_3=VA;       VnB_3=VnB;     VB_3=VB;  VC_3=VC;
epsL_3=epsL; epsG_3=epsG; epsT_3=epsT;
WX_3 = WX;     WYG_3 = WYG; WYL_3 = WYL;
iO2_3 = iO2; iCO2_3 = iCO2; iNH3_3 = iNH3; iNO2_3 = iNO2;
iSub_3 = iSub; iXNs_3 = iXNs; iXNb_3 = iXNb; iXag_3 = iXag;
KlNs_3 = KlNs; KlNb_3 = KlNb; KmNs_3 = KmNs; KmNb_3 = KmNb;
mumax_3=mumax; maint_3=maint;
```

```
Yx_3 = Yx;      Yx1_3 = Yx1; Ym1_3 = Ym1;
indG_3 = indG; indL_3 = indL;
MSG_3=MSG;      MSL_3=MSL;      MSS_3=MSS;      MSSt_3=MSSt;
Vl_3 = Vl;      alpha_3=alpha; KLa_3=KLa;

if arret
  disp(' *=*=* Nitri : initialization aborted *=*=*')
  break
end
disp(' *** Nitri - End of initialization ***')
```

10. ANNEX 2 : SOFTWARE OF THE CONTROL

10.1. Link between the different routines

i_con				
scon	estim1			
	con_3	integ_im	im_nitri	irate
				stasysim
			im_nitr2	irate
				stasysim

10.2. Listings

10.2.1. Initialization programme of control and estimation : 'i_con'

```
%****************************************************************
%       Control and estimators                                  *
%       Version 3.2     May 2002                                *
%                                                               *
%       Control model according to TN 64.4                      *
%       Estimator model according to TN 48.1                    *
%                                                               *
%       Parameters of the control and estimators                *
%                                                               *
%****************************************************************

%>============================================================================
% CONTROL
%>============================================================================
% Simulation parameters
dt_c = 1;   % (h) sampling period of control.
tmin_c=dt_c;
tmax_c=dt_c;
prec_c=1e-2;  % necessary accuracy to avoid numerical unstability in Nitri
dt_s = .5;  % (h) sampling period of scenario simulation.
if (typtst == 0 | typtst == 10 | typtst == 11)
  H_c = 10; % (h) horizon of the scenario simulation of the internal model
else
  H_c = 20; % (h) horizon of the scenario simulation of the internal model
end
T_s = 0:dt_s:H_c;
% Max flow rate increment for scenario method
stepmax = .2; % (l/h) maximum step of the algorithm in 'con_3'
% Initialization of vector of indices
iO2_c  = 1;           % indices of O2 L in the state vector
iCO2_c = iO2_c  + 1;  % indices of HCO3- in the state vector
iNH3_c = iCO2_c + 1;  % indices of NH3 L in the state vector
iNO2_c = iNH3_c + 1;  % indices of NO2 in the state vector
iNO3_c = iNO2_c + 1;  % indices of NO3 in the state vector
```

```
iPO4_c = iNO3_c + 1;  % indices of PO4 in the state vector
iSO4_c = iPO4_c + 1;  % indices of SO4 in the state vector
iSub_c = 1:NG_3+NL_3; % ind. of subs. in state vector
iXNs_c = NG_3+NL_3+1; % ind. of act. biomass Ns in the state vector
iXNb_c = iXNs_c + 1;  % ind. of act. biomass Nb in the state vector
iXag_c = NG_3+NL_3+1 :NG_3+NL_3+2*NS_3; % ind. of active and global in the state vector
% Parameters of the simplified column (internal model)
NV_c = NI_3;
Kdis_c = Kdis_3; % Dissociation constant
KlNs_c = KlNs_3;
KlNb_c = KlNb_3;
KmNs_c = KmNs_3;
KmNb_c = KmNb_3;
Yx_c = Yx_3;
Yx1_c = Yx1_3;
Ym1_c = Ym1_3;
indG_c = indG_3;
indL_c = indL_3;
RG_c = RG_3;
RL_c = RL_3;
fG_c = fG_3;
fL_c = fL_3;
epsL_c = epsL_3;
epsG_c = epsG_3;
epsT_c = epsG_c + epsL_c;
alpha_c = alpha_3;
KLa_c = KLa_3;
Gin_c = Gin_3; % Gas flow rate
Fin_c = Fin_3;
VG_c = VB_3*epsG_c;  % volume of gas of the simplified column
VL_c = VB_3*epsL_c;  % volume of liq of the simplified column
mumax_c = mumax_3;
maint_c = maint_3;
[A_c,B_c,C_c,D_c,E_c] = stasysim(NG_3, NL_3, Gin_c, Fin_c, VG_c, VL_c, ...
                                 alpha_c, Kdis_c, KLa_c);
%>=========================================================================================
% ESTIMATORS
%>=========================================================================================
dt_e = .1;  % (h) sampling period of estimators.
            % For model 3, dte=.1 h is necessary for the integrative method (otherwise,
unstable)
%>-----------------------------------------------------------------------------
% PARAMETERS OF ESTIMATORS
% 1_ Estimators type 1 & 3
  VA_e = VA_3;
  VB_e = VB_3;
  VC_e = VC_3;
  VG_e = VA_e*epsG_c/epsT_c + VB_e*epsG_c + VC_e*epsG_c/epsT_c;  % volume of gas of the
equivalent column
  VL_e = VA_e*epsL_c/epsT_c + VB_e*epsL_c + VC_e*epsL_c/epsT_c;  % volume of liq of the
equivalent column
% 2_ Estimator type 3 only
  RLm = RL;
  RGm = RG;
  gamma = .5;  % volumetric ratio first part / total column
  VG1 = gamma*VG_e;  % volume of gas of the first part of the column
  VL1 = gamma*VL_e;  % volume of liq of the first part of the column
  VG2 = VG_e - VG1;  % volume of gas of the second part of the equivalent column
  VL2 = VL_e - VL1;  % volume of liq of the second part of the equivalent column
%>-----------------------------------------------------------------------------
% CONSTANTS OF ESTIMATORS 1 & 3
% 1_ Coefficients of the linear system binding rate and biomass concentration
  sigma11 = Yx1(3,1) * mumax_c(1) + Ym1(3,1) * maint_c(1); % for NH3 and Nitrosomonas
  sigma12 = Yx1(3,2) * mumax_c(2) + Ym1(3,2) * maint_c(2); % for NH3 and Nitrobacter
  sigma21 = Yx1(5,1) * mumax_c(1) + Ym1(5,1) * maint_c(1); % for NO3 and Nitrosomonas
  sigma22 = Yx1(5,2) * mumax_c(2) + Ym1(5,2) * maint_c(2); % for NO3 and Nitrobacter
  sigma31 = Yx1(4,1) * mumax_c(1) + Ym1(4,1) * maint_c(1); % for NO2 and Nitrosomonas
  sigma32 = Yx1(4,2) * mumax_c(2) + Ym1(4,2) * maint_c(2); % for NO2 and Nitrobacter
% 2_ Coefficients of the linear system binding rates of NH3 NO3 and NO2
  den = sigma11*sigma22 - sigma12*sigma21;
  beta1 = (sigma22*sigma31 - sigma21*sigma32)/den;
  beta2 = (sigma11*sigma32 - sigma12*sigma31)/den;
```

```
  delta11 = sigma22/den;          delta12 = -sigma12/den;
  delta21 =-sigma21/den;          delta22 =  sigma11/den;
%>-------------------------------------------------------------------------
% Compensation term
  compens = 0*(-.002); % mol/l
```

10.2.2. Main programme of control and estimation : 'scon'

```
function [sys,x0]=scon(tn,x,u,flag, ...
            dt_c,dt_e,u0_c,H,T_s,NL,NG,NS,NI,NV, ...
            iSub,iXNs,iXNb,iXag, ...
            KlNs,KlNb,KmNs,KmNb,mumax,maint, ...
            Yx,Yx1,Ym1,VG_c,VL_c,VL_e,alpha,Kdis,KLa,Gin,stepmax, ...
            beta1,beta2,delta11,delta12,delta21,delta22,compens,ind_3e, ...
            dimx_e,dimx_c,dimi_c)

% SCON      S-Function to manage the calling time of the nitrite control (with
SIMULINK)
%                                                    the nitrite estimator
% Synopsis
%       [sys,x0]=scon(tn,x,u,flag,dt, ...)
%
% Parameters
%       dt      sampling period of the control
%       u0      initial inputs (to compute the initial value of x)
%       The other parameters are defined in file 'i_con.m'
%
% Input 'u' (column vector)
%       u(1)                          (l/h) liquid flow rate
%       u(1+[1:NG])                   (mol/l) conc. in the Gas input flow
%       u(1+NG+[1:NG+NL])             (mol/l) conc. in the Liquid input flow
%       u(1+2NG+NL+[1:NG,NG+[2:NL]])(mol/l) conc. of O2 CO2 NH3 NO3 PO4 SO4 in the
L output flow
%       u(3NG+2NL+1)                  (l/h)   required flow rate
%       u(3NG+2NL+2)                  (mol/l) NO2 setpoint (or NO2 max constrain)
%
% State
%       (mol/l)  forecast max of NO2 and temporal NO2 on horizon H
%
% Output
%       (mol/l)  forecast max of NO2 and temporal NO2 on horizon H
%
% Version 1.0  May 2002

global tnc
global coefil3
global itest_e % to be cancelled in the C programme

%> Array sizing and initial conditions   ------------------------------------------------
if flag==0,
  [m, n] = size(T_s);
  sys = [
    0                   % continuous states
    dimx_e+dimx_c+dimi_c  % discrete states for estimator and control
    dimx_e+dimx_c+dimi_c  % outputs for estimator and control
    dimi_c                % inputs for estimator and control
    0                   % discontinuous ...
    0                   % direct feedthrough
  ];
  % Initialisation of x of estimator
    x0e = estim1([],u0_c(ind_3e),flag,dt_e,VL_e,Kdis(NG),alpha(NG),Gin,beta1,beta2, ...
              delta11,delta12,delta21,delta22,compens);
  % Initialisation of x of control
```

```
    [temps,x0c] = con_3(...
            [u0_c;x0e([3,5,6])], H, T_s, NL, NG, NS, NI, NV, ...
            iSub, iXNs, iXNb, iXag, ...
            KlNs, KlNb, KmNs, KmNb, mumax, maint, Yx, Yx1, Ym1, ...
            VG_c, VL_c, alpha, Kdis, KLa, Gin, stepmax, VL_e);
  % Initialisation of time between to successive runnings of control
  tnc = dt_c;
  % Initialisation of x of the S_function
  x0 = [x0e;
        x0c;
        u0_c];

%> Discrete state  x(n+1)   ------------------------------------------------------------
elseif abs(flag)==2,
  if abs( round(tn/dt_e)-(tn/dt_e) ) < sqrt(eps),
    % Filtering the signals
    %ufil3 = coefil3*ufil3 + (1-coefil3)*u;
    ufil3 = coefil3*x(dimx_e+dimx_c+[1:dimi_c]) + (1-coefil3)*u;
    %> Computation of the state x(n+1)at each sampling period (estimator)
    % Inputs of the estimator
    %       ue(1)    (l/h) liquid flow rate
    %       ue(2)    (mol/l) total NH3 conc. in the liquid input flow
    %       ue(3)    (mol/l) NO2 conc. in the liquid input flow
    %       ue(4)    (mol/l) NO3 conc. in the liquid input flow
    %       ue(5)    (mol/l) total NH3 conc. in the liquid output flow
    %       ue(6)    (mol/l) NO3 conc. in the liquid output flow
    %ue = u(ind_3e);
    %xe = x(1:dimx_e)
    syse = estim1(x(1:dimx_e),ufil3(ind_3e),flag,dt_e,VL_e,Kdis(NG),alpha(NG),Gin,beta1,beta2,
...
              delta11,delta12,delta21,delta22,compens);
    sys = [syse; x(dimx_e+[1:dimx_c]); ufil3];
    if ~itest_e % if 0 --> checking of estimator alone (without control)
      if tnc >= dt_c
        %> Computation of the state x(n+1)at each sampling period (control)
        tn
        tic
        % syse([3,5,6]) ==> estimation of NO2 according to the integrative method
        % syse([4,5,6]) ==> estimation of NO2 according to the Laplace transfer
        [temps,sysc] = con_3(...
          [ufil3;syse([3,5,6])], H, T_s, NL, NG, NS, NI, NV, ...
          iSub, iXNs, iXNb, iXag, ...
          KlNs, KlNb, KmNs, KmNb, mumax, maint, Yx, Yx1, Ym1, ...
          VG_c, VL_c, alpha, Kdis, KLa, Gin, stepmax, VL_e);
        toc
        sys(dimx_e+[1:dimx_c]) = sysc;
        tnc = 0;
      end
      tnc = tnc + dt_e + eps;
    end

  else
    sys = x;
  end

%> System outputs   ----------------------------------------------------------------
elseif flag==3,
  sys =  x;

%> Instant of the next call   ------------------------------------------------------------
elseif flag==4,
  ns = tn/dt_e;            % number of simulations
  %sys = dt_e * (1  + floor(ns + 1e-13*(1+ns)));
  sys = dt_e * (1  + floor(ns + eps*(1+ns)));

%> ------------------------------------------------------------------------------------
else
  sys = [];
end
```

10.2.3. Estimator : ‘estim1’

```
%************************************************************
%       Biomass and Nitrite estimators  (model 1)           *
%       Version 3.2     May 2002                            *
%                                                           *
%       Estimator model according to TN 48.1                *
%                                                           *
%       The model of the estimator is a mere first order    *
%       in which the biomass is constant throughout the volume*
%                                                           *
%       Estimation is computed by 2 methods :               *
%       .1 integration (result in variables 'Xi' & 'Xic')   *
%       .2 Laplace transfer according to TN 48.1 p.18       *
%          (result in variables 'Xi2' & 'Xi2c')             *
%                                                           *
%************************************************************
% . time period of the predictor : between 0.1 and 1 hour   *
% . [NO2-] at input of column is assumed null               *
%************************************************************
function x_out = estim1(x,u,flag,dt,VL,Kdis_h,alpha_h,Gin,beta1,beta2, ...
                  delta11,delta12,delta21,delta22,compens)

% Inputs
% ------
    Fin  = u(1);              % (l/h) liquid flow rate
    dL_h = u(2)/(1+Kdis_h);   % (mol/l) NH3_total at column input
    dL_i = u(3);              % (mol/l) NO2 at column input
    dL_a = u(4);              % (mol/l) NO3 at column input
    Xb(1,1)= u(5)/(1+Kdis_h); % (mol/l) NH3_total at column output
    Xb(2,1)= u(6);            % (mol/l) NO3 at column output
    %> Derivative of measurement of NH3 and NO3 at column output
    if flag == 0, % initialization (assumption of steady state)
      dXb = zeros(size(Xb));
    else
      dXb = (Xb-x(7:8)) / dt;
    end

% If 'Fin' null, estimation is frozen
% -----------------------------------
    if Fin <= 0, x_out=x; break, end
% Estimation of biomass and nitrite
% ---------------------------------
    coef = 1+Kdis_h+alpha_h*Gin/Fin;
    coef1 = beta1*coef;
    % Liquid time constant (for the whole column)
    tauL1 = Fin/VL;
    alph = exp(-dt*tauL1);
    % 1. Computation of the (NH3 and NO3-) variation rate
    %r1 = (tau_h*dXb(jj,1) + Xb(jj,1) - G1_h*dL_h(jj) - G2_h*dG_h(jj)) ...
    %     / G3_h;                                    % [NH3] rate
    r1 = (dXb(1) + (Xb(1)-dL_h)*tauL1)*coef;  % [NH3] rate
    r2 = dXb(2) + (Xb(2) - dL_a)*tauL1;       % [NO3] rate

    % 2. Mean productive Ns & Nb biomass concentration in the column
    Cx(1,1) = delta11*r1 + delta12*r2;
    Cx(2,1) = delta21*r1 + delta22*r2;

    % 3. production rate of nitrite in last tank of the column
    r3 = beta1*r1 + beta2*r2; % [NO2] rate

    % 3. concentration of nitrite in last tank of the column (Laplace transfer)
    e = -(dL_i + beta2*dL_a + coef1*dL_h);
    if flag == 0, % initialization (assumption of steady state)
      s1=e; % init. of the first order (static gain equal to 1)
    else
      s1 = x(12);
    end
    s1 = alph*s1 + (1-alph)*e;
    s2 = beta2*Xb(2) + coef1*Xb(1);
```

```
    Xi2 = s1 + s2;
    Xi2c = max(0,Xi2 + compens);  % Concentration >= 0
    % 4. concentration of nitrite in last tank of the column (integration of the variation
rate of NO2)
    if flag == 0, % initialization (assumption of steady state)
      Xi=Xi2; % init. of the integration
    else
      Xi = x(1) + (-x(1)*tauL1 + r3)*dt;
    end
    Xic = max(0,Xi + compens);  % Concentration >= 0
    % Saving in x
    %       Index Unit    Definition
    x_out=[
      Xi;   % 1   (mol/l) [NO2] estimated in the liquid output flow; integrative method
      Xi2;  % 2   (mol/l) [NO2] estimated in the liquid output flow; Laplace transfer
      Xic;  % 3   (mol/l) x(1) with bias compensation
      Xi2c; % 4   (mol/l) x(2) with bias compensation
      Cx;   % 5,6 (g/l)   estimated Ns & Nb productive biomass concentration
      Xb;   % 7,8 (mol/l) Xb at previous time n
      r1;   % 9   (mol/h) r1 = indirect measurement of variation rate of NH3
      r2;   % 10  (mol/h) r1 = indirect measurement of variation rate of NO3
      r3;   % 11  (mol/h) r1 = indirect measurement of variation rate of NO2
      s1;   % 12  (mol/h) s1 = internal variable of Laplace transfer at previous time n
      s2];  % 13  (mol/h) s2 = internal variable of Laplace transfer
```

10.2.4. Main programme of control : 'con_3'

```
%*************************************************************
%       Nitrite production control                           *
%       Version 1.0     May 2002                             *
%                                                            *
%       Internal model according to TN 64.4                  *
%                                                            *
%*************************************************************
function [T_s,cs_out] = con_3(...
            u, H, T_s, NL, NG, NS, NI, NV, ...
            iSub, iXNs, iXNb, iXag, ...
            KlNs, KlNb, KmNs, KmNb, mumax, maint, Yx, Yx1, Ym1, ...
            VG, VL_c, alpha, Kdis, KLa, Gin, stepmax, VL_e)

global ut_c

% Input 'u' (column vector)
%       u(1)                        (l/h) liquid flow rate
%       u(1+[1:NG])                 (mol/l) conc. in the Gas input flow
%       u(1+NG+[1:NG+NL])           (mol/l) conc. in the Liquid input flow
%       u(1+2NG+NL+[1:NG,NG+[2:NL]])(mol/l) conc. of O2 CO2 NH3 NO3 PO4 SO4 in the L output
flow
%       u(3NG+2NL+1)                (l/h)   required flow rate
%       u(3NG+2NL+2)                (mol/l) NO2 setpoint (or NO2 max constrain)
%       u(3NG+2NL+3)                (mol/l) estimated conc. of NO2 at column output
%       u(3+3NG+2NL+[1:NS])         (mol/l) estimated conc. of biomass at column output
% Output 'cs_out' (column vector) : concentration of NO2 (scenario on horizon H)
% The parameters of the simplified column (H, NL ...) are defined in file 'i_con.m'

% Simulation or general parameters
tdeb=0;
tfin=H;
T = [tdeb, tfin];
n1 = max(size(T_s));
cs1 = zeros(n1,1);
nu = max(size(u));
% Iterative algorithm parameters
cntmax = 20; % max counter of iterations
epsabs = .01; %absolute accuracy
% NO2 setpoint
spNO2 = u(nu-3);
% Computation of the limiting terms 'a1' & 'a2'
CB0 = u(1+2*NG+NL+[1:NG])./(1+Kdis); % conc. of biphasic compounds in part B of the simplified
column
xNO2 = u(nu-2); % NO2 estimation
```

```
xO2 = CB0(1); % O2 measurement at column output
xNH3 = CB0(NG);% molecular NH3 at column output
a1 = xO2 .* xNH3 ./ (KlNs(1) + xO2) ./ (KlNs(2) + xNH3);
a2 = xO2 .* xNO2 ./ (KlNb(1) + xO2) ./ (KlNb(2) + xNO2);
% State vector
CX0 = u(nu-1:nu) ./ [a1;a2] * VL_e/VL_c; % estimated active biomass Ns & Nb
X0 = [CB0;                  % conc. of biphasic compounds in part B of the simplified column
      xNO2;                 % conc. of NO2 estimated in part B of the simplified column
      u(1+3*NG+NL+[1:NL-1]); % conc. of NO3 PO4 SO4 in part B of the simplified column
      CX0;                  % conc. of active biomass
      CX0];                 % conc. of total biomass (set to conc. of active biomass)

% Process inputs
dG = (u(1+[1:NG]) * ones(size(T)))';
CL0 = u(1+NG+[1:NG+NL])./[1+Kdis; ones(NL,1)];
dL = (CL0 * ones(size(T)))';
utp = [T' dG dL]; % partial inputs
%> Computation of the interval for dichotomy method
%> ------------------------------------------------
x1 = u(1);   % init of input of the non linear function
if x1 <= 0, x1=stepmax/10; end
idichoto=0;
for cnt = 1:cntmax
  Fint =  (x1 * ones(size(T)))';
  ut_c = [utp Fint]; % complete inputs

  [temps,cs] = integ_im( ...
    ut_c, tdeb, tfin, ...
    X0, NL, NG, NS, NI, NV, ...
    iSub, iXNs, iXNb, iXag, ...
    KlNs, KlNb, KmNs, KmNb, mumax, maint, Yx, Yx1, Ym1, ...
    VG, VL_c, alpha, Kdis, KLa, Gin);

  cs1 = interp1(temps,cs(:,NV-3),T_s); % concentration of NO2 (scenario on horizon H)
  dy = cs1(2) - cs1(1); % approximate derivative at origin
  if dy >= 0
    extr1 = max(cs1);  % extremum is the max of the non linear function
  else
    extr1 = min(cs1);  % extremum is the min of the non linear function
  end
  y1 = spNO2 - extr1; % distance to the setpoint
  if cnt == 1
    if y1>= 0
      way = 1;
    else
      way = -1;
    end
  else
    if y1*y2 <=0, idichoto=1; break, end % the interval [x1 x2] is found
  end
  x2=x1;  y2=y1;  extr2=extr1; cs2=cs1;  % saving the later point
  x1 = x1 + stepmax*way;
  if x1 <= 0, x1=stepmax/10; end
  if (abs(x1-x2)<=2*eps)
    disp('*** No interval found in con_3 ***')
    break
  end
  cnt = cnt+1;
end
cnt1 = cnt;
if (cnt >= cntmax)
  disp('*** No interval found in con_3 ***')
end

%> Running dichotomy method
%> ------------------------
if idichoto
  for cnt = 1:cntmax
    x3 = (x1+x2)/2;
    Fint =  (x3 * ones(size(T)))';
    ut_c = [utp Fint]; % complete inputs
```

```
    [temps,cs] = integ_im( ...
      ut_c, tdeb, tfin, ...
      X0, NL, NG, NS, NI, NV, ...
      iSub, iXNs, iXNb, iXag, ...
      KlNs, KlNb, KmNs, KmNb, mumax, maint, Yx, Yx1, Ym1, ...
      VG, VL_c, alpha, Kdis, KLa, Gin);

    cs3 = interp1(temps,cs(:,NV-3),T_s); % concentration of NO2 (scenario on horizon H)
    dy = cs3(2) - cs3(1); % approximate derivative at origin
    if dy >= 0
      extr3 = max(cs3);  % extremum is the max of the non linear function
    else
      extr3 = min(cs3);  % extremum is the min of the non linear function
    end
    y3 = spNO2 - extr3; % distance to the setpoint
    if (y1*y3 < 0)
      x2 = x3;
      extr2 = extr3;
      cs2 = cs3;
    else
      x1 = x3;
      y1 = y3;
      extr1 = extr3;
      cs1 = cs3;
    end
    if (abs(x2 - x1) < epsabs)
      break
    end
  end
  if (cnt >= cntmax)
    idichoto=0;
    disp('*** No convergence in con_3 ***')
  end
end
if idichoto
  % Manipulated variable = min(x1, required flow rate)
  mv = min([x1, x2, u(nu-4)]);
  %mv = min([x1, x2]);
else
  if way == 1
    mv = min([10*stepmax, u(nu-4)]);
  else
    mv = min([stepmax/10, u(nu-4)]);
  end
end
% Output vector
if x1 <= x2
  cs_out = [mv;             % manipulated variable applied to the process
            [x1; x2];       % interval of the manipulated variable
            [extr1; extr2];% interval of the extremeum
            cnt1;cnt;       % counters of iterations
            cs1; cs2];      % forecast temporal NO2 on horizon H, depending on x1 and x2
else
  cs_out = [mv;             % manipulated variable applied to the process
            [x2; x1];       % interval of the manipulated variable
            [extr2; extr1];% interval of the extremeum
            cnt1;cnt;       % counters of iterations
            cs2; cs1];      % forecast temporal NO2 on horizon H, depending on x1 and x2
end
```

10.2.5. Gathering several methods of integration : ‘integ_im’

```
%************************************************************
%       Integration of the internal model on horizon H      *
%       Version 1.0     May 2002                            *
%                                                           *
%       Internal model according to TN 64.4                 *
%                                                           *
%************************************************************
function [temps,cs] = integ_im( ...
            ut, tdeb, tfin, ...
```

```
            X0, NL, NG, NS, NI, NV, ...
            iSub, iXNs, iXNb, iXag, ...
            KlNs, KlNb, KmNs, KmNb, mumax, maint, Yx, Yx1, Ym1, ...
            VG, VL, alpha, Kdis, KLa, Gin)

global A_c B_c C_c D_c E_c
global Fin1_c
global integ_prog  % to be cancelled in the C language programme

if integ_prog == 1
  % Integration parameters
  tmin=1e-3; % 4.04.00
  tmax=.2;
  prec=1e-6;  % necessary accuracy to avoid numerical unstability in Nitri
  options(1) = prec;
  options(2) = tmin;
  options(3) = tmax;
  options(4) = 1;        % automatic choice gear/adams
  % Integration algorithm
  % It has been noted after the implementation of the steady state (March 2001) that :
  % . 'Runge-Kutta 5' of Simulink in 'g_sim.m' is no more convenient (negative concentration)
  % . 'rk45' of Matlab in 'sim.m' is acceptable but is numerically noisy
  % . 'Adams/Gear' seems to be the best when called from Simulink in 'g_sim.m'
  % . 'gear' with 'options(4) = 1' seems to be the best when called from Matlab in 'sim.m'
  [temps,xx,cs] = gear('im_nitri',[tdeb tfin], [], options, ut, ...
    X0, NL, NG, NS, NI, NV, ...
    iSub, iXNs, iXNb, iXag, ...
    KlNs, KlNb, KmNs, KmNb, mumax, maint, Yx, Yx1, Ym1, ...
    VG, VL, alpha, Kdis, KLa, Gin);
%
elseif integ_prog == 0
  dt_E = .01; % (h) sampling period of Euler method
  temps = [tdeb:dt_E:tfin]';
  [m, n] = size(temps);
  [sys,x0]=im_nitri([],[],[],0, ...
    X0, NL, NG, NS, NI, NV, ...
    iSub, iXNs, iXNb, iXag, ...
    KlNs, KlNb, KmNs, KmNb, mumax, maint, Yx, Yx1, Ym1, ...
    VG, VL, alpha, Kdis, KLa, Gin);
  x = x0; % initial state
  cs = zeros(n,sys(3));
  u = interp1(ut(:,1), ut(:,1+[1:sys(4)]), temps);
  for ii = 1:m
    for flag = [3, 1]
      % Computation of outputs 'sys'
      [sys,xdummy]=im_nitri(temps(ii),x,u(ii,:)',flag, ...
        X0, NL, NG, NS, NI, NV, ...
        iSub, iXNs, iXNb, iXag, ...
        KlNs, KlNb, KmNs, KmNb, mumax, maint, Yx, Yx1, Ym1, ...
        VG, VL, alpha, Kdis, KLa, Gin);
      if flag == 3
        cs(ii,:) = sys; % process outputs
      else
        x = x + sys'*dt_E; % integration of state
        ind = find(x <=0);
        if ~isempty(ind), x(ind) = zeros(size(ind)); end
        ind = find(x <=0);
        if ~isempty(ind),
          disp(['Negative concentration in integ_im at point ii=',num2str(ii)'])
          keyboard
          x(ind) = zeros(size(ind));
        end
      end
    end
  end
  %
elseif integ_prog == 2
  prec=1e-6;  % necessary accuracy to avoid numerical unstability in Nitri
  [temps,xx] = ode45('im_nitr2',tdeb,tfin,X0,prec);
  % Matrices of the state system
  Fin = ut(1,NI+2);
  if Fin ~= Fin1_c  % in order to divide the running time by 5
```

```
    [A_c,B_c,C_c,D_c,E_c] = stasysim(NG, NL, Gin, Fin, VG, VL, alpha, Kdis, KLa);
    Fin1_c = Fin;
    disp('Y State system renewal')
  end
  [mu, nu] = size(ut);
  u = (interp1(ut(:,1),ut(:,2:nu-1),temps));
  % subtrates
  sys = xx(:,iSub)*C_c' + u*D_c';
  % output : substrates and active and global biomass :
  cs = [sys, xx(:,iXag)];
end
```

10.2.6. Integration by Matlab Sub System function (Sfunction of Matlab) : 'im_nitri'

```
%*****************************************************************
%       Nitrifying column (internal model of control)            *
%       Version 1.0     May 2002                                 *
%                                                                *
%       State system according to TN 64.4                        *
%       Matrices of the state system are time variable           *
%                                                                *
%       im_nitri.m : Computation of the derivative vector        *
%                                 and output vector              *
%                    for Matlab S_function                       *
%                                                                *
%*****************************************************************

function [sys,x0]=im_nitri(tn,x,u,flag, ...
                           X0, NL, NG, NS, NI, NV, ...
                           iSub, iXNs, iXNb, iXag, ...
                           KlNs, KlNb, KmNs, KmNb, mumax, maint, Yx, Yx1, Ym1, ...
                           VG, VL, alpha, Kdis, KLa, Gin)

global A_c B_c C_c D_c E_c
global Fin1_c

% IM_NITRI        S-Function for simulation of the simplified nitrifying column (NB=1)
%
% Synopsis
%       [sys,x0]=im_nitri(tn,x,u,flag,X0, ...)
%
% Parameters
%       X0      initial state vector
%       The other parameters are defined in file 'i_con.m'
%
% State vector x
%       Length of the state vector : NG+NL+2*NS
%         with : NG = number of bi-phases substrates (present in Gas and Liquid)
%                NL = number of mono-phase substrates (present in Liquid only)
%                NS = number of strains (Nitrosomonas and Nitrobacter)
%                NE = NG + NL
%                NU = NE * NB
%                NI = 2*NG + NL
%                NV = NI * NB
%                NB = 1
%       The state vector is composed of the concentrations of the compounds
%       of the simplified column (one tank) in liquid phase only.
%       Indices of the compounds in the state vector :
%         For a tank i (i = 1)
%         Indice                          Compound
%         (i-1)*NE + 1                    O2    in Liquid phase
%         (i-1)*NE + 2                    CO2   in Liquid phase
%         (i-1)*NE + NG                   NH3   in Liquid phase
%         (i-1)*NE + NG + 1               NO2
%         (i-1)*NE + NG + 2               NO3
%         (i-1)*NE + NG + 3               HPO4
%         (i-1)*NE + NG + NL              SO4
%         NU + 2*(i-1)*NS + 1             Xv_Ns  active Nitrosomonas biomass
%         NU + 2*(i-1)*NS + NS            Xv_Nb  active Nitrobacter biomass
%         NU + 2*(i-1)*NS + NS + 1        Xt_Ns  global Nitrosomonas biomass
```

```
%         NU + 2*(i-1)*NS + NS + NS       Xt_Nb  global Nitrobacter biomass
% Inputs
%       The input vector is composed of
%       . the concentrations of the compounds of the incoming liquid and gas flows;
%       . and of the liquid flow rate
%       Indices of the compounds in the inputs vector :
%         Indice     Compound
%         1          O2    in the incoming gas flow (flow rate : Gin)
%         2          CO2   in the incoming gas flow (flow rate : Gin)
%         3          NH3   in the incoming gas flow (flow rate : Gin)
%         NG + 1     O2    in the incoming liquid flow (flow rate : Fin)
%         NG + 2     CO2   in the incoming liquid flow (flow rate : Fin)
%         NG + 3     NH3   in the incoming liquid flow (flow rate : Fin)
%         2*NG + 1   NO2   in the incoming liquid flow (flow rate : Fin)
%         2*NG + 2   NO3   in the incoming liquid flow (flow rate : Fin)
%         2*NG + 3   HPO4  in the incoming liquid flow (flow rate : Fin)
%         2*NG + 4   SO4   in the incoming liquid flow (flow rate : Fin)
%         NI + 1     Fin (l/h)
%
% Outputs
%       The output vector is composed of the concentrations of the compounds
%       of the column in gas and liquid phases.
%       Indices of the compounds in the state vector :
%         For a tank i (i=1 and NB=1)
%         Indice                          Compound
%         (i-1)*NG + 1                    O2    in Gas phase
%         (i-1)*NG + 2                    CO2   in Gas phase
%         (i-1)*NG + 3                    NH3   in Gas phase
%         NG*NB + (i-1)*NE + 1            O2    in Liquid phase
%         NG*NB + (i-1)*NE + 2            CO2   in Liquid phase
%         NG*NB + (i-1)*NE + 3            NH3   in Liquid phase
%         NG*NB + (i-1)*NE + NG + 1       NO2
%         NG*NB + (i-1)*NE + NG + 2       NO3
%         NG*NB + (i-1)*NE + NG + 3       HPO4
%         NG*NB + (i-1)*NE + NG + 4       SO4
%         NV + 2*(i-1)*NS + 1             Xa_Ns  active Nitrosomonas biomass
%         NV + 2*(i-1)*NS + 2             Xa_Nb  active Nitrobacter biomass
%         NV + 2*(i-1)*NS + NS + 1        Xg_Ns  global Nitrosomonas biomass
%         NV + 2*(i-1)*NS + NS + 2        Xg_Nb  global Nitrobacter biomass
%

%> Sizes array and Initial conditions ----------------------------------------
if flag==0,
   sys = [
        NG+NL+2*NS  % continuous states
        0           % discrete states
        2*NG+NL+2*NS% outputs
        NI+1        % inputs
        0           % discontinuous ...
        0           % direct feedthrough
        ];
   x0 = X0;

%> Continuous state (computation of derivatives) -----------------------------
elseif abs(flag)==1,
  % The concentrations cannot be negative
    ind = find(x<0);
    x(ind) = zeros(size(ind));

  % Computation of the matrix J (notation of TN 44.2)
  %-------------------------------------------------
    [mui,mai,rmu] = irate(x,1,2,3,4,KlNs,KlNb,KmNs,KmNb,mumax,maint);
    J = Yx1*diag(mui(:,1)) + Ym1*diag(mai(:,1));
    %dim = NG+NL;
    %J = zeros(dim*(NB+2),NS*(NB+2));
    %ind1 = 1:dim;
    %ind2 = 1:NS;
    %for ii = 1:NB+2
    %  Ji = Yx1*diag(mui(:,ii)) + Ym1*diag(mai(:,ii));
    %  J(ind1,ind2) = Ji;
    %  ind1 = ind1 + dim;
    %  ind2 = ind2 + NS;
```

```
    %end

  % Computation of growth rate of active and global biomass
  %-------------------------------------------------------
    mua = mui + diag(maint.*Yx) * (1-rmu); % specific growth rate of active bio.
    Cx = [x(iXNs); x(iXNb)];  % concentration of active biomass Ns and Nb
    rX = mua .* Cx;  % growth rate of active biomass Ns and Nb
    rg = mui .* Cx;  % growth rate of global (active + decay) biomass Ns and Nb

  % Computation of the derivative of the state vector
  %--------------------------------------------------
    Fin = u(NI+1);
    if Fin ~= Fin1_c  % in order to divide the running time by 5
      [A_c,B_c,C_c,D_c,E_c] = stasysim(NG, NL, Gin, Fin, VG, VL, ...
                                       alpha, Kdis, KLa);
      %disp(['X State syst comput in im_nitri Fin,Fin1_c: ',num2str(Fin),' ;
',num2str(Fin1_c)])
      Fin1_c = Fin;
    end
    % subtrates :
    sys(iSub) = A_c*x(iSub) + B_c*u(1:NI) + E_c*J*Cx;

    % active and global biomass :
    sys(iXag) = [rX; rg];

%> Outputs ----------------------------------------------------------------------
elseif flag==3,
  % The concentrations cannot be negative (if they were ==> problem to analyse)
    ind = find(x<0);
    if ~isempty(ind)
      disp(['Negative concentration in im_nitri at time t=',num2str(tn),' hour'])
      keyboard
    end

  % Computation of the output vector
  %---------------------------------
    Fin = u(NI+1);
    if Fin ~= Fin1_c  % in order to divide the running time by 5
      [A_c,B_c,C_c,D_c,E_c] = stasysim(NG, NL, Gin, Fin, VG, VL, ...
                                       alpha, Kdis, KLa);
      %disp(['Y State syst comput in im_nitri Fin,Fin1_c: ',num2str(Fin),' ;
',num2str(Fin1_c)])
      Fin1_c = Fin;
    end
    % subtrates :
    sys(1:NV) = C_c * x(iSub) + D_c * u(1:NI);

    % active and global biomass :
    sys(iXag + NG) = x(iXag);

%> ------------------------------------------------------------------------------
else
   sys = [];
end
```

10.2.7. Integration by Matlab ODE function (Ordinary Differential Equation): 'im_nitr2'

```
%*************************************************************
%       Nitrifying column (internal model of control)         *
%       Version 1.0     May 2002                              *
%                                                             *
%       State system according to TN 64.4                     *
%       Matrices of the state system are time variable        *
%                                                             *
%       im_nitr2.m : Computation of the derivative vector     *
%                                  and output vector          *
%                    for 'ode' Matlab function                *
%*************************************************************
```

```
function dx=im_nitr2(tn,x)

global A_c B_c C_c D_c E_c
global Fin1_c
global ut_c NL_3 NG_3 NS_3 NI_3 ...
       iO2_c iCO2_c iNH3_c iNO2_c iSub_c iXNs_c iXNb_c iXag_c ...
       KlNs_c KlNb_c KmNs_c KmNb_c mumax_c maint_c Yx_c Yx1_c Ym1_c ...
       VG_c VL_c alpha_c Kdis_c KLa_c Gin_c

% IM_NITR2        Differential equation of the simplified nitrifying column (NB=1)
%
% Synopsis
%       [sys,x0]=im_nitri(tn,x,u,flag,X0, ...)
%
% Parameters
%       X0      initial state vector
%       The other parameters are defined in file 'i_con.m'
%
% State vector x
%       Length of the state vector : NG+NL+2*NS
%         with : NG = number of bi-phases substrates (present in Gas and Liquid)
%                NL = number of mono-phase substrates (present in Liquid only)
%                NS = number of strains (Nitrosomonas and Nitrobacter)
%                NE = NG + NL
%                NU = NE * NB
%                NI = 2*NG + NL
%                NV = NI * NB
%                NB = 1
%       The state vector is composed of the concentrations of the compounds
%       of the simplified column (one tank) in liquid phase only.
%       Indices of the compounds in the state vector :
%         For a tank i (i = 1)
%         Indice                          Compound
%         (i-1)*NE + 1                    O2    in Liquid phase
%         (i-1)*NE + 2                    CO2   in Liquid phase
%         (i-1)*NE + NG                   NH3   in Liquid phase
%         (i-1)*NE + NG + 1               NO2
%         (i-1)*NE + NG + 2               NO3
%         (i-1)*NE + NG + 3               HPO4
%         (i-1)*NE + NG + NL              SO4
%         NU + 2*(i-1)*NS + 1             Xv_Ns  active Nitrosomonas biomass
%         NU + 2*(i-1)*NS + NS            Xv_Nb  active Nitrobacter biomass
%         NU + 2*(i-1)*NS + NS + 1        Xt_Ns  global Nitrosomonas biomass
%         NU + 2*(i-1)*NS + NS + NS       Xt_Nb  global Nitrobacter biomass
% Inputs
%       The input vector is composed of
%       . the concentrations of the compounds of the incoming liquid and gas flows;
%       . and of the liquid flow rate
%       Indices of the compounds in the inputs vector :
%         Indice     Compound
%         1          O2    in the incoming gas flow (flow rate : Gin)
%         2          CO2   in the incoming gas flow (flow rate : Gin)
%         3          NH3   in the incoming gas flow (flow rate : Gin)
%         NG + 1     O2    in the incoming liquid flow (flow rate : Fin)
%         NG + 2     CO2   in the incoming liquid flow (flow rate : Fin)
%         NG + 3     NH3   in the incoming liquid flow (flow rate : Fin)
%         2*NG + 1   NO2   in the incoming liquid flow (flow rate : Fin)
%         2*NG + 2   NO3   in the incoming liquid flow (flow rate : Fin)
%         2*NG + 3   HPO4  in the incoming liquid flow (flow rate : Fin)
%         2*NG + 4   SO4   in the incoming liquid flow (flow rate : Fin)
%         NI + 1     Fin (l/h)
%
% Outputs
%       The output vector is composed of the concentrations of the compounds
%       of the column in gas and liquid phases.
%       Indices of the compounds in the state vector :
%         For a tank i (i=1 and NB=1)
%         Indice                          Compound
%         (i-1)*NG + 1                    O2    in Gas phase
%         (i-1)*NG + 2                    CO2   in Gas phase
%         (i-1)*NG + 3                    NH3   in Gas phase
%         NG*NB + (i-1)*NE + 1            O2    in Liquid phase
```

```
%         NG*NB + (i-1)*NE + 2            CO2   in Liquid phase
%         NG*NB + (i-1)*NE + 3            NH3   in Liquid phase
%         NG*NB + (i-1)*NE + NG + 1       NO2
%         NG*NB + (i-1)*NE + NG + 2       NO3
%         NG*NB + (i-1)*NE + NG + 3       HPO4
%         NG*NB + (i-1)*NE + NG + 4       SO4
%         NV + 2*(i-1)*NS + 1             Xa_Ns  active Nitrosomonas biomass
%         NV + 2*(i-1)*NS + 2             Xa_Nb  active Nitrobacter biomass
%         NV + 2*(i-1)*NS + NS + 1        Xg_Ns  global Nitrosomonas biomass
%         NV + 2*(i-1)*NS + NS + 2        Xg_Nb  global Nitrobacter biomass
%
[m, n] = size(ut_c);
u = (interp1(ut_c(:,1),ut_c(:,2:n),tn))';
%> Continuous state (computation of derivatives) ------------------------------
  % The concentrations cannot be negative
    ind = find(x<0);
    x(ind) = zeros(size(ind));

  % Computation of the matrix J (notation of TN 44.2)
  %-------------------------------------------------
    [mui,mai,rmu] =
irate(x,iO2_c,iCO2_c,iNH3_c,iNO2_c,KlNs_c,KlNb_c,KmNs_c,KmNb_c,mumax_c,maint_c);
    J = Yx1_c*diag(mui(:,1)) + Ym1_c*diag(mai(:,1));
    %dim = NG+NL;
    %J = zeros(dim*(NB+2),NS*(NB+2));
    %ind1 = 1:dim;
    %ind2 = 1:NS;
    %for ii = 1:NB+2
    %  Ji = Yx1*diag(mui(:,ii)) + Ym1*diag(mai(:,ii));
    %  J(ind1,ind2) = Ji;
    %  ind1 = ind1 + dim;
    %  ind2 = ind2 + NS;
    %end

  % Computation of growth rate of active and global biomass
  %-------------------------------------------------------
    mua = mui + diag(maint_c.*Yx_c) * (1-rmu); % specific growth rate of active bio.
    Cx = [x(iXNs_c); x(iXNb_c)];  % concentration of active biomass Ns and Nb
    rX = mua .* Cx;  % growth rate of active biomass Ns and Nb
    rg = mui .* Cx;  % growth rate of global (active + decay) biomass Ns and Nb

  % Computation of the derivative of the state vector
  %-------------------------------------------------
    Fin = u(NI_3+1);
    if Fin ~= Fin1_c  % in order to divide the running time by 5
      [A_c,B_c,C_c,D_c,E_c] = stasysim(NG_3, NL_3, Gin_c, Fin, VG_c, VL_c, ...
                                       alpha_c, Kdis_c, KLa_c);
      Fin1_c = Fin;
      %disp('X State system comput')
    end
    % subtrates :
    dx1 = A_c*x(iSub_c) + B_c*u(1:NI_3) + E_c*J*Cx;

    % active and global biomass :
    dx2 = [rX; rg];

    % state derivative
    dx = [dx1; dx2];
```

10.2.8. Instantaneous production rate of biomass : 'irate'

```
%*****************************************************************
%       Nitrifying column and Estimators                         *
%       Version 2.1     June 2000                                *
%                                                                *
%       State system according to TN 44.2                        *
%                                                                *
%                                                                *
%       irate.m : instantaneous production rate of biomass       *
%                                                                *
%*****************************************************************
```

```
function [mui,mai,rmu] = irate(x,iO2,iCO2,iNH3,iNO2,KlNs,KlNb,KmNs,KmNb,mumax,maint)

% x   : state vector
% mui : limited specific growth rate of active biomass Ns or Nb for each tank
% mai : limited specific maintenance rate of active bio. Ns or Nb for each tank

  % Computation of rmu = mu/mumax :
  xx = x(iO2).*x(iCO2);
  x1 = xx.*x(iNH3);
  x2 = xx.*x(iNO2);
  a1 = x1 ./ (KlNs(1) + x(iO2)) ./ (KlNs(2) + x(iNH3)) ./ (KlNs(3) + x(iCO2));
  a2 = x2 ./ (KlNb(1) + x(iO2)) ./ (KlNb(2) + x(iNO2)) ./ (KlNb(3) + x(iCO2));
  rmu = [a1, a2]';

  % Computation of the limited specific global growth rate 'mui'
  mui = diag(mumax) * rmu;

  % Computation of the limited maintenance rate 'mai'
  x1 = x(iO2).*x(iNH3);
  x2 = x(iO2).*x(iNO2);
  a1 = x1 ./ (KmNs(1) + x(iO2)) ./ (KmNs(2) + x(iNH3));
  a2 = x2 ./ (KmNb(1) + x(iO2)) ./ (KmNb(2) + x(iNO2));
  rma = [a1, a2]';
  mai = diag(maint) * rma; % limited maintenance rate (06.06.00)
```

10.2.9. State system of the internal model : ‘stasysim’

```
%*****************************************************************
%       Nitrifying column control                                *
%       Version 1.0     May 2002                                 *
%                                                                *
%       State system according to TN 64.4                        *
%                                                                *
%       stasysim.m : Computation of the state system             *
%                    of the internal model of the control        *
%                                                                *
%*****************************************************************

function [As,Bs,Cs,Ds,Es] = stasysim(NG, NL, Gin, Fin, VG, VL, ...
                                     alpha, Kdis, KLa)

ii = 2; % index of the fixed bed of the column of the internal model (to be
        % compatible with 'transbi' which is also called for the column of the process)
qL = Fin;
qG = Gin;

tauL1 = qL/VL * ones(NL,1);                    % inverse of tauL
% Computation of teta1 teta2 beta1 beta2
[G1,G2,G3,alpha4,alpha5,tau,arret] = transbi(VG,VL,qG,qL,KLa,Kdis,alpha,ii);
As = [
  -diag(1./tau), zeros(NG,NL);
  zeros(NL,NG),  -diag(tauL1)];
Bs = [
  diag(G2./tau), diag(G1./tau), zeros(NG,NL);
  zeros(NL,NG),  zeros(NL,NG),  diag(tauL1) ];
Cs = [
  diag(alpha4),  zeros(NG,NL);
  eye(NG),       zeros(NG,NL);
  zeros(NL,NG),  eye(NL)     ];

Ds =  diag([alpha5; zeros(NG + NL,1)]);

Es =  diag([G3./tau; ones(NL,1)]);

if arret == 1, disp('*** Subroutine transbi called by stasysim ***'), end
```