ASSOCIATION POUR LE DÉVELOPPEMENT DE L'ENSEIGNEMENT
ET DE LA RECHERCHE EN SYSTÉMATIQUE APPLIQUÉE

MELISSA

Predictive Control of biomass production in the Rhodobacter compartment of MELISSA

Contract ESA - ESTEC / ADERSA
Purchase Order n°151491 of 10/05/95

Memorandum of Understanding
ECT/FG/CB/95.205

Technical Note 28.1
Version 1 Issue 0

J.-J. LECLERCQ

August 1996

7, Bd du Maréchal Juin
B.P. 52
91371 VERRIERES-LE-BUISSON CEDEX
Téléphone : (1) 60 13 53 53
Télécopie : (1) 69 20 05 63
E-Mail : adersa@adersa.worldnet.net

ADERSA

CONTENTS

NOTATIONS

cE_b : E_b set point ($W.m^{-2}$)

cF_R : F_R set point ($W.m^{-2}$)

C_X : biomass concentration ($kg.m^{-3}$)

cpi : level i (i = 1 or 2) production rate set point (g/h)

cqi : level i (i = 1 or 2) flow rate set point (l/h)

dil : dilution rate (h^{-1})

dq : flow rate ratio (dimension less)

dt : control time period (h)

E_b : total radiant energy absorbed by the sphere photometer ($W.m^{-2}$)

F_R : mean incident radiant light energy flux ($W.m^{-2}$)

h_c : coincidence horizon (h)

p : biomass production rate (g/h)

q : flow rate (l/h)

TR : closed loop time response (h)

1. INTRODUCTION

The aim of this note is to test a control law to be applied to the production of biomass in the compartment II (rhodospirillum rubrum photobioreactor) of the MELISSA project.

The process simulator is based on the model for growth of rhodospirillum rubrum under light limitation in photobioreactors (developed at LGCB in Clermont-Ferrand by J.F. Cornet ; TN 23.4).

The control law is based on the above-mentioned model and widely inspired from the one of the compartment IV (developed at ADERSA by N. Fulget ; TN 24.1) the law structure is the same, only parameters are different.

The dimensions of the simulated reactor are those of compartment IV (Spirulina photobioreactor).

The studied system (fig.1) is composed of :

- the process (photobioreactor) whose inputs are the flow rate q and the light flux F_R, and whose outputs are the concentration biomass C_X and the production p ;
- the systems A and B (detailed in fig. 2 and 3) which include, each of them, a level 0 controller associated to an actuator (valve or volumetric pump for system A ; light energy intensity for system B) and to a sensor (flow meter for system A ; integrating sphere photometer for system B). These systems are supposed to have short dynamics compared to those of the process and the level 0 controllers are supposed perfect. So they are reduced to identity function, in the simulator :

  $q = cq1$ at each instant t
  $F_R = cF_R$

- the level 1 controller which works on the PFC (Predictive Functional Control) principles and computes the light flux F_R to control the production p versus the level 1 production set point cp1 ;
- the level 2 controller which computes the level 1 flow rate and production set points, cq1 and cp1, versus the corresponding level 2 set points, cq2 and cp2, taking into account the max and min concentration constraints and the flow variation dq (± 10 % of cq2). The set points cq2 and cp2 are fixed by the operator.

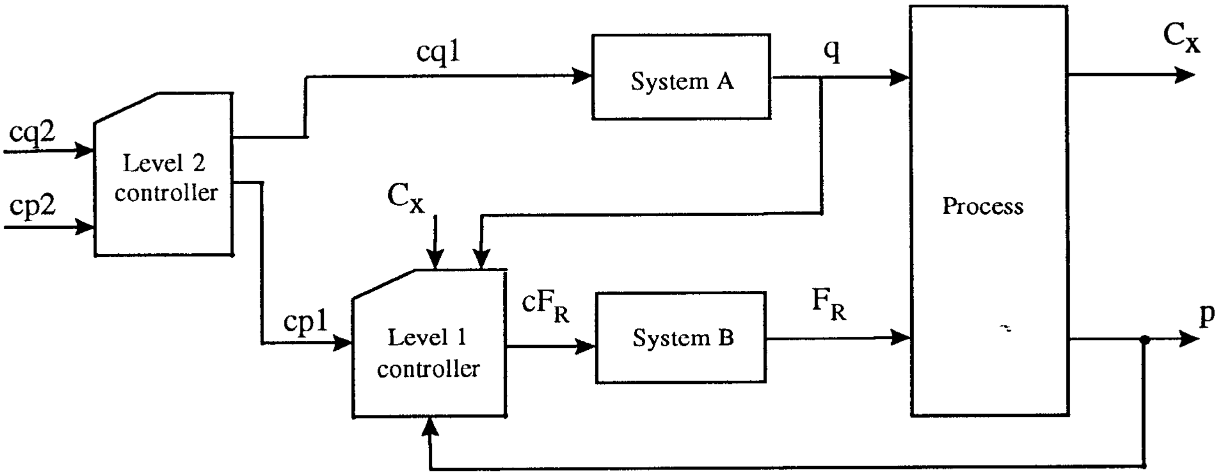


Fig 1 : Global scheme processus and levels 0 to 2 controllers

The systems A and B which contain level 0 controllers (generally PID) are detailed hereafter.

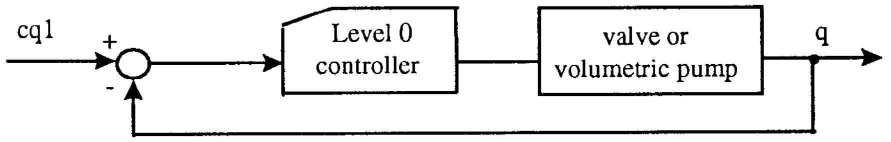


Figure 2 : System A

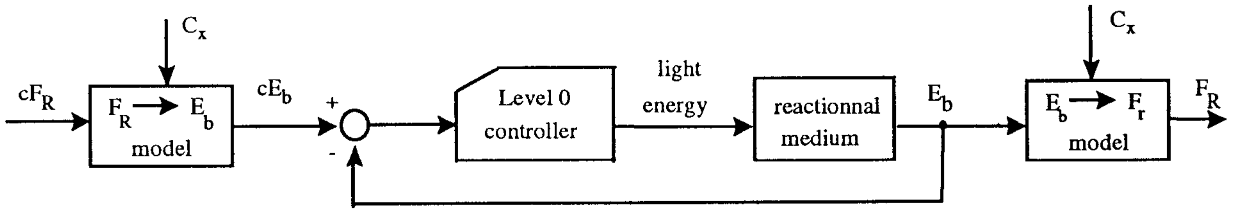


Figure 3 : System B

2. CONTROL STRCTURE

Full details are given in TN 24.1 about the levels 1 and 2 of the control. Here are a few recalls.

2.1. *Level 1 control*

This controller is built on the principles of PFC (Predictive Functional Control) :

- internal model ;
- reference trajectory to specify the future closed loop behavior ;
- manipulated variable structuration ;
- modeling error extrapolation.

A scenario strategy is used to adapt this control law to the present process which is non linear : several (generally 2) input protocols are applied to the model from which a prediction of the model output is inferred. The computation involves the integration of a differential equation : the Euler integration method is quite convenient for this purpose.

2.2. *Level 2 control*

This level checks the compatibility flow rate and production set points, fixed by the operator (or a upper level), with regard to variation flow rate and concentration constraints.

Algorithm : at each period of time, the following computations are made :

1 - maximum and minimum flow rate

q_max = cq2 (1+ dq)
q_min = cq2 (1 - dq)

2 - maximum and minimum production

p_max = q_max $*C_X$_max
p_min = q_min $*C_X$_min

3 - level 1 production set point

cp1 = max [p_min, min (p-max, cp2]

4 - level 1 flow rate set point

```
if $\left(\frac{cp1}{C_{x}\_max} > cq2\right)$ then
    $cq1 = \min\left(q\_max, \frac{cp1}{C_{x}\_max}\right)$
else if $\left(\frac{cp1}{C_{x}\_min} < cq2\right)$ then
    $cq1 = \max\left(q\_min, \frac{cp1}{C_{x}\_min}\right)$
else
    $cq1 = cq2$
```

3. SIMULATION RESULTS

The present control structure works on the compartment IV (Spirulina biophotoreactor) and has already been validated. The aim is now to check its extension to the compartment II and to tune the specific parameters.

3.1. *Control parameter tuning*

- control period : dt = 0,5 hour
- closed loop response time : TR = 12h
- coïncidence point : h_c = 2.5 hours (nhc=5 control periods)

3.2. *Parameters process*

They are summed up in the file comnl2.h.

Their values are not exactly those of the TN 23.4 (CORNET J.-F., 1996. Model parameters for growth of rhodospirillum rubrum under light limitation in photobioreactors) because the present simulations have been done before its issue. Here are the differences :

Parameter	Present study	TN 23.4	Unit
Ea	600	220	m^2/kg
Es	1330	480	m^2/kg

3.3. *Tests results*

The figures 4 to 8 show the simulations whose presentation is described in the following table.

Graph	Signal name	Type of line	Unit
1 (top of figure)	production • measure • level 2 setpoint • level 1 set point	 continuous -. dashed	g/h
2	light energy flux F_R	continuous	W/m^2
3	biomass concentration • measure • minimum or maximum constraint	 continuus -.	g/l
4	flow rate • measure • level 2 set point	 continuous -.	l/h

A white noise (amplitude : 0.02 g/l) is added to the concentration.

- **First test (figure 4)**

  The dilution rate is fixed to 0.01 h^{-1} (cq2 = 0.07 l/h). At time t=20h, a step (from 0.035 g/h to 0.045 g/h) is applied to the level 2 production set point cp2. As the concentration and the flow rate do not over run their limits, the level 2 control calculates the level 1 set points equal to the level 2 set points, all over the test :

  cq1 = cq2
  cp1 = cp2

  Immediately after the cp1 step, the max constraint of F_R is reached ans slows the dynamics of the production. The time response (30 h) depends on the amplitude step of cp1 and on its sense (positive or negative).

- **Second test (figure 5)**

  The test is nearly the same as in the previous figure, except that the cp2 step is negative, from 0.045 to 0.035 g/h. The production, which is not null even when F_R is minimum (10 W/m^2), goes down only because of dilution. The response time is 60h.

  These two tests (fig 4 and 5) illustrate the high non linear aspect of the process.

- **Third test (figure 6)**

  The dilution is higher : 0.02 h^{-1} (cq2 = 0.14 l/h). The result is opposite to the one of the two previous tests : the response time is higher for the positive step (at t=20h) than for the negative one (at t=70h), 30h and 20h, respectively. It is another illustration of the non linear aspect of the process.

- **Fourth test (figure 7)**

  The dilution rate is lower : its initial value is 0.004 h^{-1} (cq2 = 0.028 l/h). At time t=20h, a production set point step cp2 is applied from 0.04 g/h to 0.06 g/h.

  The level 2 control detects an incompatibility between level 2 set points and the constraints. First it computes a level 1 production set point cp1, taking into account the maximum flow rate and the maximum concentration (graph 3) : cp1 varies from 0.04 g/h to 0.0308 * 1.5 = 0.0462 g/h (graph 1). Then it allows a 10% increase of flow rate with regard to cq2 : the flow rate q rises from 0.028 l/h to 0.0308 l/h (graph 4).

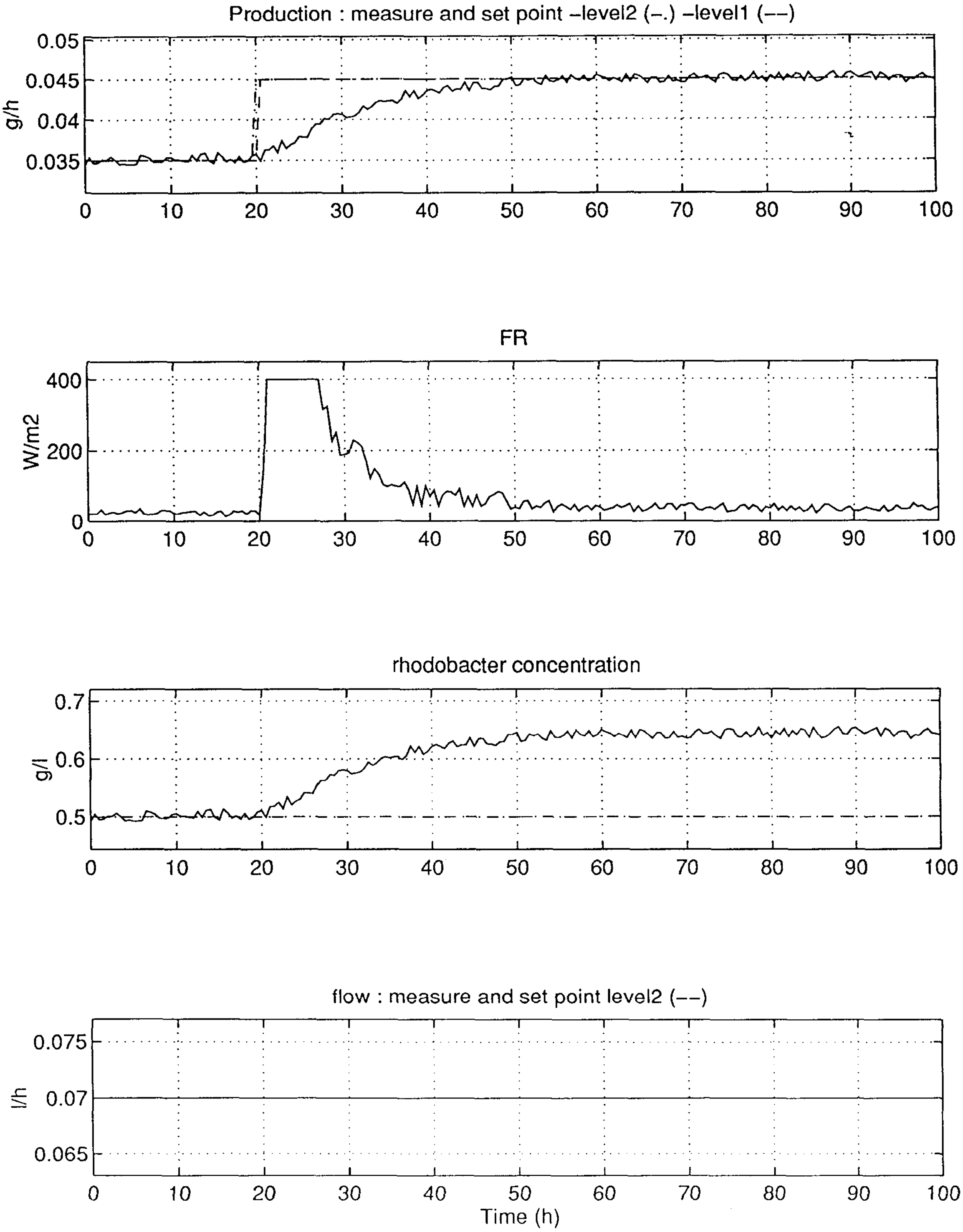
  The dynamics of the production is quick : the response time is less than 10 h.

- **Fifth test (figure 8)**

A step is applied to cp2 at time t=20h and then a step is applied to cq2 at time t=60h.

At time t=20h, all happens as in the test of figure 7 : the level 2 control computer cp1 versus the maximum concentration and raises the flow rate q with a 10% increase.

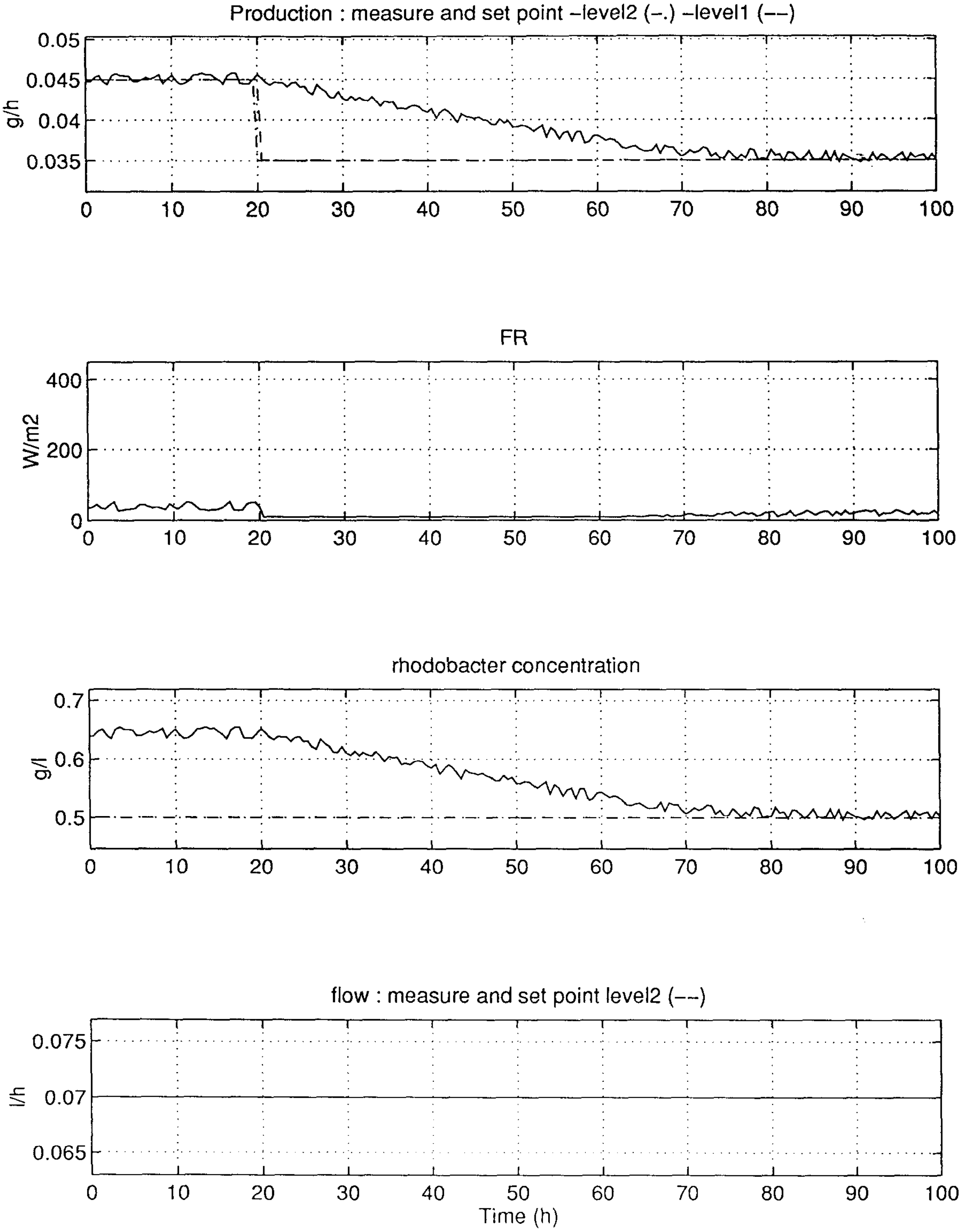
At time t=60h, the level 2 control sets the flow rate q to 0.04 l/h so that cp1 remains equal to cp2 ; the concentration stays at its maximum value : 1.5 g/l.



Production control

dil = .01 (1/h)

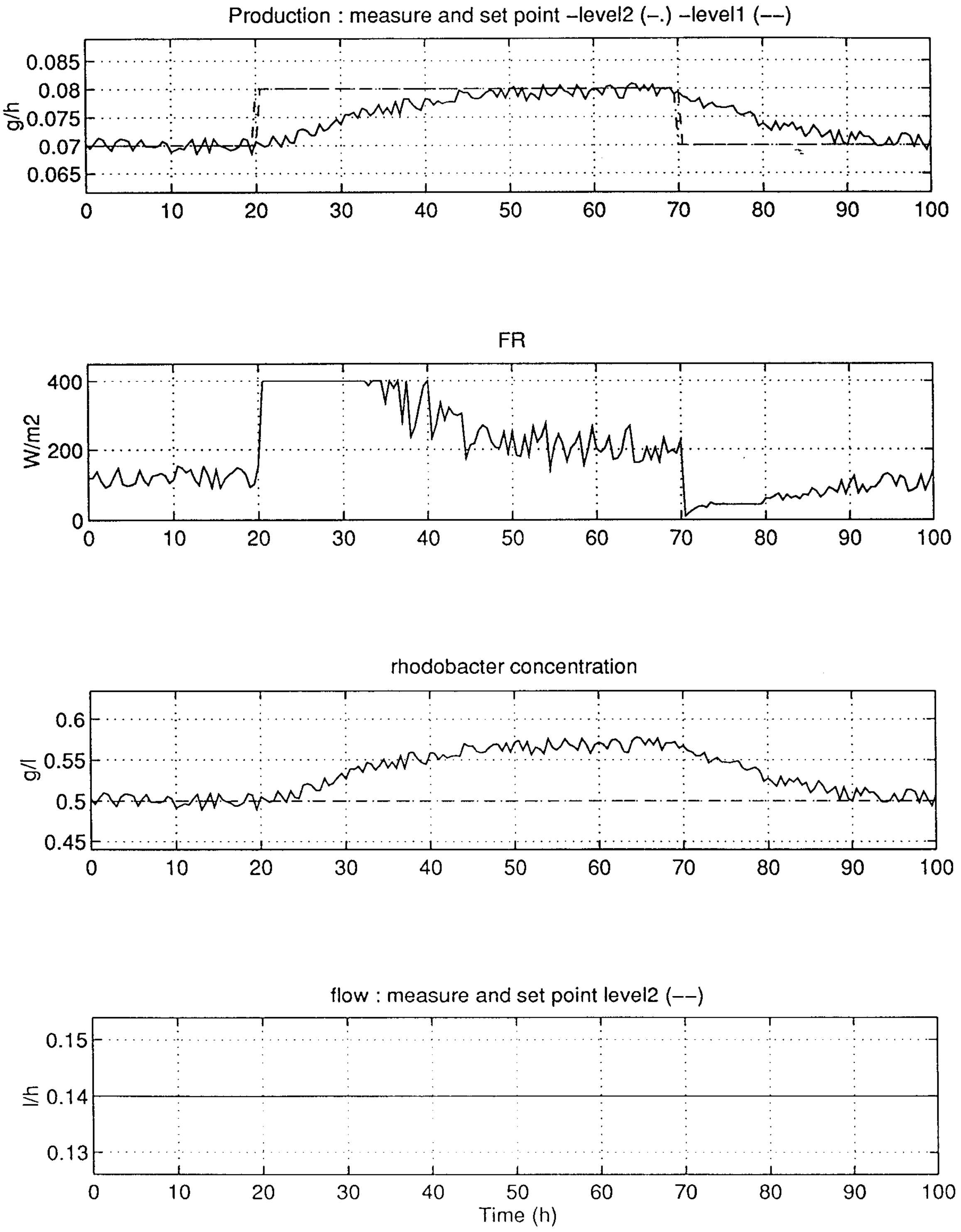
Figure 4



Production control

dil = .01 (1/h)

Figure 5



Production : measure and set point -level2 (-.) -level1 (--)

FR

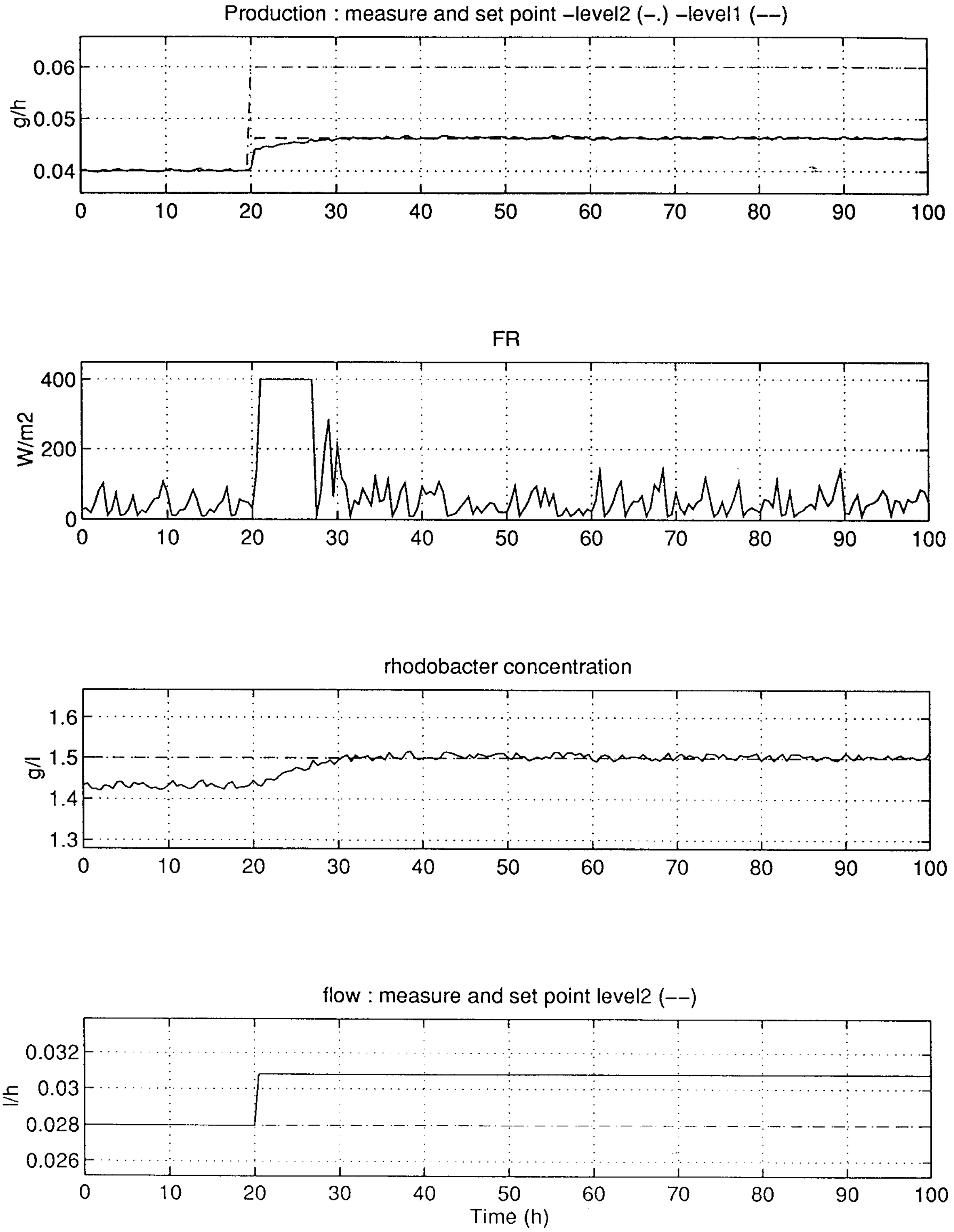
rhodobacter concentration

flow : measure and set point level2 (--)

Production control

dil = .02 (1/h)

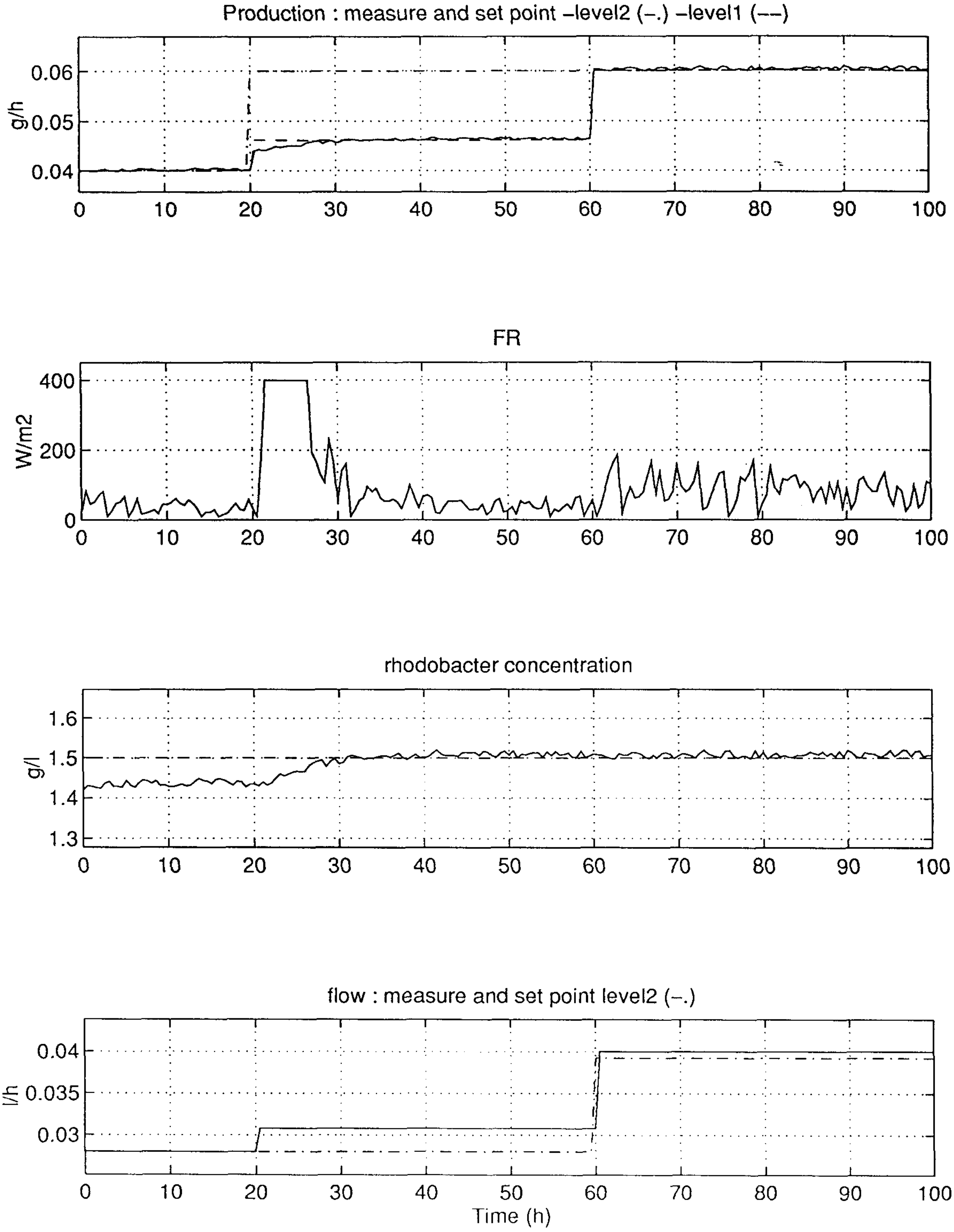
Figure 6



Production control

dil = .004 (1/h)

Figure 7



Production control

dil variation at time = 60 : from .004 to .0056 (1/h)

Figure 8

4. CONCLUSION

The present control law based on the PFC principles is checked in simulation, and is quite convenient to the rhodospirillum rubrum photobioreactor, highly non linear process.

It can now be validated on the real process.

The parameters of the regulator are summed up in the file 'comnl2.h' in annex (do not forget to give to Ea and Es their right values according to the Technical Note 23.4).

REFERENCES

BINOIS C. Automatisation d'un écosystème artificiel utilisé comme système de support vie. Première intéraction modèle/système de contrôle. CNAM Thesis. September 1994.

CORNET J.-F., 1996. Modelling of the MELISSA artificial ecosystem. Model parameters for growth of Rhodospirillum rubrum under light limitation in photobioreactors. ESA contract PRF 141315, Technical Note 23.4.

CORNET J.-F., DUSSAP C.G., GROS J.-B., 1993a. Modelling of physical limitations in photobioreactors. Adaptation of the light energy transfer model to cylindrical geometries. ESA contract PRF 130-820, Technical Note 19.1.

CORNET J.-F., DUSSAP C.G., GROS J.-B., 1993b. Modelling of physical limitations in photobioreactors. Modelling of exopolysaccharide synthesis in cultures of *Spirulina platensis*. ESA contract PRF 130-820, Technical Note 19.2.

CORNET J.-F., DUSSAP C.G., GROS J.-B., 1993c. Modelling of physical limitations in photobioreactors. Applications to simulation and control of the Spirulina Compartment of the MELISSA artificial ecosystem. ESA contract PRF 130-820, Technical Note 19.3.

FULGET N., 1994. MELISSA, first approach of Model Based Predictive Control of Spirulina compartment. ESA contract PRF 132-443, Technical Note 21.2.

FULGET N. 1995. MELISSA. Study for the non linear Model Based Predictive Control of Spirulina compartment using knowledge model. ESA contract PRF 142356, Technical Note 24.1.

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

ANNEX

- scomnl.m : main control programme (Matlab® language)
- mx_comnl2.c : interface programme from Matlab® to C language
- comnl2.c : levels 2 and 1 control routine (C language)
- funcalc.c : specific calculation functions
- comnl2.h : control model parameters
- proto2.h : declaration of functions

```
  function [sys,x0]=scomnl(tn,x,u,flag,dt,vol_rhodo,prod0,qe0)

% SCOMNL        S-Function d'un regulateur PFC (utilisation avec SIMULINK).
%
% Synopsis
%       [sys,x0]=scomnl(tn,x,u,flag,dt,vol_rhodo,prod0,qe0)
%
% Parameters
%       dt      periode de commande.
%       vol_rhodo       volume du reacteur 2 (rhodobacter)
%       prod0   consigne initiale de production niveau 2
%       qe0     consigne initiale de debit niveau 2
%
% Entrees
%       u(1)    consigne production niveau 2
%       u(2)    consigne debit niveau 2
%       u(3)    concentration mesuree
%       u(4)    flux lumineux mesure
%       u(5)    debit mesure
%
% Sorties
%       s(1)   FR
%       s(2)   consigne production niveau 1
%       s(3)   consigne debit niveau 1
%
%       L'utilisation de cette fonction suppose l'existence du Mex-File
%       MX_COMNL2
%

% ADERSA  Imm. 6  15-DEC-94
%         modifie le 5-JUIL-96 par JJL

%> Taille des parametres et Conditions initiales   ----------------------------

global flinit Fr0

if flag==0,
   sys = [
        0       % continuous states
        3       % discrete states
        3       % outputs
        5       % inputs
        0       % discontinuous ...
        0       % direct feedthrough
        ];

%       Calcul de la valeur stationnaire de Fr
%       --------------------------------------
%       Initialisation de la precision relative
        prec = 1e-2;
%       Initialisation du point de depart de la procedure iterative
        c0 = prod0 / qe0;
        xx0 = 40;
        yy0 = prod0 / vol_rhodo;
        epsilon = yy0 * prec;
        deltax = 10;
%       Calcul par procedure iterative
        Fr0 = caliter(c0, xx0, yy0, deltax, epsilon);
        if Fr0 == Inf
                Fr0 = 10;
        end

   x0 = [prod0 Fr0 qe0];
   flinit=0;
```

```
%> Etats Discrets  x(n+1)  --------------------------------------------------
elseif abs(flag)==2,
  if abs( round(tn/dt)-(tn/dt) ) < sqrt(eps),
   %> Entree
    consp = u(1);
    consq = u(2);
    cx    = u(3);
    frmes = u(4);
    qemes = u(5);

   %> Calcul de la commande
   %  Dans l'etat actuel du regulateur, l'argument 'flinit' de 'mx_comnl2'
   %  est inutile.
    if flinit == 0 ,
       frmes = Fr0;
       [fr, cons_p1, cons_q1] = mx_comnl2( consp , cx , consq , ...
                                         frmes , qemes , flinit);
       flinit=1;
    else
       [fr, cons_p1, cons_q1] = mx_comnl2( consp , cx , consq , ...
                                         frmes , qemes , flinit);
    end

   %> Sortie de la fonction: vecteur d'etat x
    sys = [cons_p1 fr cons_q1];

  else
    sys = x;
  end

%> Sorties du systeme   -------------------------------------------------------
elseif flag==3,
   sys =  x;

%> Instant du prochain appel  ------------------------------------------------
elseif flag==4,
   ns = tn/dt;            % nombre de simulation
   sys = dt * (1  + floor(ns + 1e-13*(1+ns)));

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

```
/* ---   MX_COMNL2.C   ---------------------------------------------------------
   Fonction:
      mex_fonction permettant le calcul de la commande d'un regulateur PFC
      pour le compartiment 2 (Rhodobacter)
   Synopsis:
      [FR,CONS_PROD1,CONS_QE1]=MX_COMNL2(CONS_PROD2,CX,QE2,FRMES,QEMES,FLAG)
   Description:
      Calcul des consignes de production et debit de niveau 1, CONS_PROD1
      et CONS_QE1, a partir des consignes recues par le niveau 2, CONS_PROD2
      et QE2
      Calcul la commande courante FR a partir de
      la mesure de concentration CX, de la consigne CONS_PROD1, du
      debit mesure QEMES et du flux lumineux estime FRMES
*/

/* modifie le 5-JUIL-96 par JJL */

#include "mex.h"
#include "comnl2.h"
#include "proto2.h"

/* Arguments d'entree/sortie */
#define PC_IN   prhs[0]
#define YP_IN   prhs[1]
#define QC_IN   prhs[2]
#define FR_IN   prhs[3]
#define QM_IN   prhs[4]
#define FL_IN   prhs[5]
#define FR_OUT  plhs[0]
#define CONS_P_OUT  plhs[1]
#define CONS_Q_OUT  plhs[2]

mexFunction(nlhs, plhs, nrhs, prhs)
int nlhs, nrhs;
Matrix *plhs[], *prhs[];
{
   double    *yp , *c , *d , *fr , *frmes , *qemes , *fl , *cons_p1 , *cons_q1 ;
   double varsortie[3];

        /* Test sur le nombre d'arguments */
           if (nrhs<6 | nrhs>6)
      {
         mexErrMsgTxt("MX_COMNL2 a besoin de 6 arguments d'entree.");
      }
      else if (nlhs <3 | nlhs>3)
      {
         mexErrMsgTxt("MX_COMNL2 a besoin de 3 arguments de sortie.");
      }

   /* Creation de matrice pour les arguments de sortie */
      FR_OUT = mxCreateFull( 1 , 1 , REAL );
      CONS_P_OUT = mxCreateFull( 1 , 1 , REAL );
      CONS_Q_OUT = mxCreateFull( 1 , 1 , REAL );

        /* Affectation des pointeurs pour les parametres */
           fr = mxGetPr( FR_OUT );
           cons_p1 = mxGetPr( CONS_P_OUT );
           cons_q1 = mxGetPr( CONS_Q_OUT );
           yp = mxGetPr( YP_IN );
           c   = mxGetPr( PC_IN );
           d   = mxGetPr( QC_IN );
           frmes = mxGetPr( FR_IN );
           qemes = mxGetPr( QM_IN );
           fl = mxGetPr( FL_IN );
```

```
        /* Appel de la fonction COMNL2 */
            comnl2 ( *c , *yp , *d , *frmes , *qemes , *fl , varsortie );
            *fr = varsortie[0];
            *cons_p1 = varsortie[1];
            *cons_q1 = varsortie[2];
}
```

```
#include "comnl2.h"
#include "math.h"
#include "proto2.h"

/*
   COMNL2.C   Algorithme du regulateur non lineaire.
   ESA - MELISSA - RHODOBACTER

   Date creation pour compartiment Spiruline : 15-DEC-94
   Modifie le 5-JUIL-96 par JJL
*/


/* ---  COMNL2     -------------------------------------------------------
   Fonction:
      Equations du regulateur non lineaire du compartiment 2 (Rhodobacter)
   Synopsis:
      COMNL2(CONS_PROD2,CX,CONS_QE2,FRMES,QEMES,FR,CONS_PROD1,CONS_QE1)
   Description:
      Calcul des consignes de production et debit de niveau 1, CONS_PROD1
      et CONS_QE1, a partir des consignes recues par le niveau 2, CONS_PROD2
      et CONS_QE2
      Calcul la commande courante FR a partir de
      la mesure de concentration CX, de la consigne CONS_PROD1, des mesures
      de debit QEMES et du flux lumineux FRMES
*/
void comnl2(cons_prod2,cx,cons_qe2,frmes,qemes,indic,varsortie)

double   cons_prod2, cx, cons_qe2, frmes, qemes, indic , varsortie[3];
{
/* declaration des variables internes */
   double   prod, dil, prod_ref, delfr;
   double   fr , fr1, fr2, prod1, prod2;
   double   qe_max , qe_min , prod_max , prod_min ;
   double   cons_prod1 , cons_qe1;

   prod = cx*cons_qe2;

/* algorithme de commande de niveau 2 */
qe_max = cons_qe2*(1+dq);
qe_min = cons_qe2*(1-dq);
prod_max = qe_max*cx_max;
prod_min = qe_min*cx_min;
cons_prod1 = max(prod_min,min(prod_max,cons_prod2));
cons_qe1 = cons_qe2;
if (cons_prod1/cx_max > cons_qe2 )
   {
   cons_qe1 = min(qe_max,cons_prod1/cx_max);
   }
if (cons_prod1/cx_min < cons_qe2 )
   {
   cons_qe1 = max(qe_min,cons_prod1/cx_min);
   }

   dil = cons_qe1/vol;
   prod = cx * qemes;

/* trajectoire de reference */
   prod_ref = cons_prod1 - pow(lambda,nhc)*(cons_prod1 - prod);

/* commande precedente mesuree */
   fr = frmes;

/* premier scenario */
```

```
   fr1 = fr;
   prod1 = model2(cx,fr1,dil);

/* deuxieme scenario */
   delfr = dfr*sign(cons_prod1 - prod);
   fr2 = fr1 + delfr;
   prod2 = model2(cx,fr2,dil);

/* calcul de fr */
   fr+ = (prod_ref - prod1)/(prod2 - prod1)*delfr;

/* contraintes sur fr */
   fr = max(fr_min,min(fr_max,fr));

/* ecriture des arguments de sortie */
   varsortie[0] = fr;
   varsortie[1] = cons_prod1;
   varsortie[2] = cons_qe1;
   return;
}
/* ---  MODEL2  ----------------------------------------------------------------
   Fonction:
         integration du modele
   Synopsis:
         MODEL2(CX,FR,DIL,PROD)

*/
double model2(cx,fr,dil)

double cx, fr, dil ;

{
   double v, dv, vout , prod;
   int k;

   v = cx;
   for (k=1 ; k <= nhc; k++)
       {
       dv = dercx2(v,fr,dil);
       vout =v + dt *dv;
       v=vout;
       }
   prod=vout*dil*vol;
   return (prod);
}


/* --- DERCX2 -----------------------------------------------------------------

   Fonction:
         calcul de la derivee de cx
   Synopsis:
         DERCX2(cx,fr,dil,dvt);

*/
double dercx2(cx,fr,dil)

double cx, fr, dil;

{
double  dcxdt;
double  alpha, delta, pij, pijz;
double z, rx;
```

```
alpha = sqrt(Ea/(Ea+Es));
delta = (Ea+Es)*alpha*RT*cx;

pij = 0.;
for (z=jstep/2; z<=1-jstep/2; z+=jstep)
    {
    pijz = fr/z*2*cosh(delta*z)/(cosh(delta)+alpha*sinh(delta));
    if (pijz>=Fmin)
       {
       pij+ = 2*z*pijz/(Kj+pijz)*jstep;
       }
    }
rx = muM*pij*cx*wiv;

dcxdt = -dil*cx + rx ;
return (dcxdt);
}
```

```
#include "math.h"

/* ---   MIN.C   --------------------------------------------------------------
   Fonction:
         Minimum de deux valeurs.
   Synopsis:
         X=MIN(Y,Z)
*/
double min( x1 , x2 )

double x1 , x2;
{
   double  x;
   x = (x1 < x2) ? x1 : x2;
   return( x );
}

/* ---   MAX.C   --------------------------------------------------------------
   Fonction:
         Maximum de deux valeurs.
   Synopsis:
         X=MAX(Y,Z)
*/
double max( x1 , x2 )

double x1 , x2;
{
   double  x;
   x = (x1 > x2) ? x1 : x2;
   return( x );
}

/* ---   SIGN.C   --------------------------------------------------------------
   Fonction:
         signe d'une valeur.
   Synopsis:
         X=SIGN(Y)
*/
double sign( y )

double y;
{
   double  x;
   x = (y < 0) ? -1. : 1.;
   return( x );
}
```

```
/*
   Nom : comnl2.h

   Fonction : Coefficients de la commande du compartiment 2 (Rhodobacter)

   Date : 21-MAY-96
   Modified 5-JUL-96
*/

#define dt          0.5       /* control period (in h ) */
#define nhc         5.        /* coincidence point (in dt) */
#define lambda      0.88      /* reference trajectory dynamic */
#define dfr         5.        /* radiant flux increment (in W/m2)  */
#define fr_min      10.       /* min constraint on FR (in W/m2) */
#define fr_max      400.      /* max constraint on FR (in W/m2) */
#define dq          .1        /* flow variation (dimensionless) */
#define cx_min      0.5       /* min constraint on CX (in g/l) */
#define cx_max      1.5       /* max constraint on CX (in g/l) */
#define vol         7.        /* reactor volume */

/*  not used : */
/* #define zpc */                 /* biotic mass fraction of phycocyanins */
/* #define zch */                  /* biotic mass fraction of clorophylles */
/* #define zg  */                  /* biotic mass fraction of glycogen */
/* #define za  */                  /* resultant biotic mass fraction of the
                                      light harvesting antenna */

#define Ea          600.      /* global absorption mass coefficient (m2/kg) */
#define Es          1330.     /* global scattering mass coefficient (m2/kg) */
#define RT          .048      /* radius of the cylindrical bioreactor */
#define Kj          7.        /* half saturation constant for radiant energy
                                 avalaible (W/m2) */
#define muM         .17       /* maximum growth rate (1/h) */
#define wiv         .52       /* working illuminated volume */
#define Fmin        1.        /* minimum radiant energy flux (W/m2) */
#define jstep       .01       /* integrative dimensionless radius */
```

```
void comnl2( );
double model2( );
double dercx2( );
double sign( );
double min( );
double max( );
```