MELiSSA TECHNICAL NOTE

Memorandum of Understanding 1907105/NL/CP

Université Blaise Pascal LGCB
Laboratoire de Génie Chimique et Biochimique

TECHNICAL NOTE: 83.1

DYNAMIC MODELLING OF A COUPLED MELiSSA CREW - COMPARTMENT C4A WITH MATLAB/SIMULINK

prepared by/*préparé par* Laurent Poughon

reference/*réference* ESA/ESTEC CONTRACT
issue/*édition* 1
revision/*révision* 1
date of issue/*date d'édition* 16/04/2007
status/*état* Final
Document type/*type dedocument* Technical Note
Distribution/*distribution* MELiSSA

CONFIDENTIAL DOCUMENT

APPROVAL

Title *titre*	DYNAMIC MODELLING OF A COUPLED MELiSSA CREW - COMPARTMENT C4a WITH MATLAB/SIMULINK	issue *issue* 1	revision *revision* 1

author *auteur*	Laurent Poughon	date *date*	16/04/ 2007

approved by *approuvé by*	date *date*

CHANGE LOG

reason for change /*raison du changement*	issue/*issue*	revision/*revision*	date/*date*
Final release	1	1	16/4/2007

CHANGE RECORD

Issue: 1 Revision: **ERROR! REFERENCE SOURCE NOT FOUND.**

reason for change/*raison du changement*	page(s)/*page(s)*	paragraph(s)/*paragraph(s)/responses*
If the dynamic modelling is used for simulating and controlling the crew compartment, the time scale must be less to the lethal/adverse timescale. It means that we need at least 10sec dynamic resolution or a back-up system that is independent of the loop. Perhaps it is a good idea to discuss this a bit further	15, 3.2.1	The crew+C4a model proposed here is without any gas supplies for the crew. Then it doesn't include the lethal/adverse level (i.e. an emergency in case of failure or oxygen quick depletion, CO2 quick accumulation). A 10s dynamic resolution is linked with a control level with mechanical/physical support system, what is of course a requirement for a life support system devoted to gas regeneration.

reason for change/*raison du changement*	page(s)/*page(s)*	paragraph(s)/*paragraph(s)/responses*
The assumption of constant pressure is acceptable, however, it must come along with an acceptable set range for co2 and O2 partial pressure Certainly, this is a stiff mathematical system but exactly what we need in reality. Could you please comment on the way of getting total and partial pressures correct?	22, 3.3.4	On a mathematical point of view the problem is not very complex. The problem is rather that the flows should be represented as partial matter flows (in moles or mass), associated to physical parameters (temperature, pressure) (report to conclusion). This require a refurbishing of the models. The other point will be the control system used for pressures and volumes, which will be probably associated to sink/supply subsystems.
It doesn't show steady state for CO2, please discuss in more detail what causes and consequences it has	24, graph 4	Steady-state is reached, but it is right that it is not obvious in figures 8. (it is more obvious for simulations over 4-5 days instead of a simulation over 3days)
Please provide bars for Photosim if possible	38, graph 4	EPS values are not directly given by photosim
Please elaborate on the impact of flow rates in terms of simulation and design	39, first paragraph	In terms of design, this supposes the addition of other subsytems. In principle errors are acceptable for short term simulations (1-3weeks) of a closed system. For open systems, this assumption is often used, with low impacts on mass balances calculations
Please clarify why a rectangular reactor design was used	44, 6.1	Because its is the type of reactor developed for Biorat.

TABLE OF CONTENTS

1 INTRODUCTION

The objective of this work is to start the dynamic modelling of the complete MELiSSA loop. At this time, modelling of the loop is limited to a steady-state mass and elements (CHONSP) balance. This tool allows

- to compute an average recycling efficiency of the loop,
- to identify critical point in the architecture and in the mass flows distribution in the loop,
- to estimate roughly volumes of the bioreactors, assuming several characteristic rates (as organic load, NH_3 load, productivity rates).

In order to have a complete tool for simulating the loop, it is necessary to have a dynamic model which will include mass transfer limitation, transient behaviour (all compartments didn't have the same time constants). A dynamic model will be much more complex to establish and required to set up correctly each compartment (an under-dimensioned compartment will lead to failed the system) and to include the control algorithms (which are required if we want to close the loop)

As a first step it was decided to build a model of a gas coupled Crew-C4a compartment. The C4a compartment is the most well-known MELiSSA compartment at this time (at least on a modelling point of view), and the gas coupled system was experimentally tested in the "Biorat" framework. This work is also used to check the ability to translate an existing dynamic model (i.e. PHOTOSIM) under Matlab/Simlink and to check the ability to build a functional gas coupled system under Simulink.

In this document are presented the development of a crew dynamic model, the adaptation of PHOTOSIM as a dynamic model under Matlab/Simulink and a simple control law for managing the coupled system. Each model of the subsystems is checked independently and then coupled to model a crew-C4a system.

2 CREW- C4A SYSTEM DESCRIPTION

2.1 Obectives

The aim of this work is to model and simulate a gas coupled system between a crew and the C4a MELiSSA compartment (Figure 1). It is quite a "Biorat" system. This is a first step in the dynamic modeling of the MELiSSA loop which would allow at least identifying possible problems and bottlenecks in the Matlab/Simulink approach for modeling such closed systems. Three subsystems will be studied for achieving this objective:

- ➔ The crew subsystem, which dynamic model needs to be developed.
- ➔ The C4a MELiSSA compartment subsystem, which is adapted from the current dynamic and steady-state models.
- ➔ A control subsystem to manage the coupling between the two other subsystems.

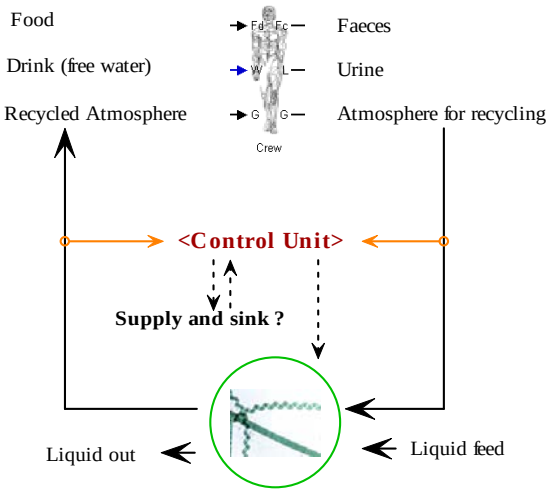


Figure 1 Principle of the Crew-C4a gas loop closed system

As for the complete mass balance steady-state model of the MELiSSA loop [4] Matlab/Simulink software is used for developing the models, the subsystems (as Simulink S-block objects) and for solving the system built.

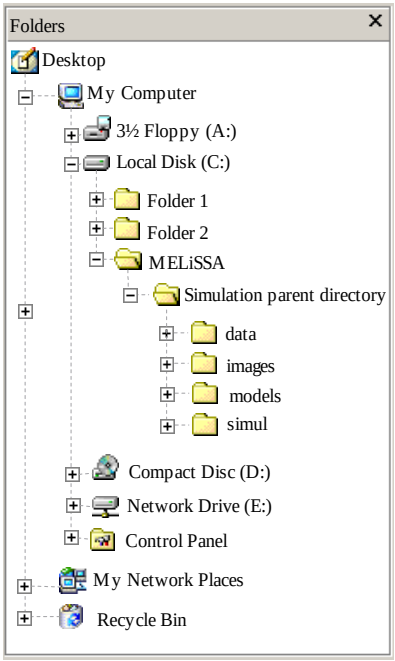
2.2 Simulink model – structure and files

Principles of Matlab/Simulink were already presented in TN 79.2 [4]. For the dynamic models, the Simulink S-function, which are developed, are not restricted to the definition of an output from the knowledge of an input (principles of mass balanced model) but include also a system of ordinary differential equations (ODE) or /and a discrete time dependant system (which acts

as an update of the current state of the system simulated). The continuous (ODE) and discrete systems are numerically solved to predict the dynamic behaviour of the system modelled.

The models presented here and released with the Technical Note were developed with Matlab/Simulink 13-R3. The solver used is ODE15s with the parameter reported in Figure 12. The models of the subsystem will be detailed in the next chapters, but we will give here the file structure chosen (and then required) for the modelling and the simulation of the Crew-C4a system.

The files (Table 2) for the modelling and the simulation are sorted and stored in one off the directories listed in Table 1. This architecture must be respected for using the models. All models must be opened with Simulink and started from the *"**../models**/"* directory, which also must be the current directory for running a simulation of the model (if it is not the current directory simulation failed and an error message indicating that a file cannot be found should be displayed in the Matlab command line).
The simulation directory and its sub-directory must be added to the "Simulink path". Note that this may be incompatible with the path added for the mass balance MELiSSA loop model [4] which may need to be removed.



The "simulation parent directory" can have any kind of name , as well as the upper folders in the architecture. The file can alos be installed on other hard drive letters (D:, E:,….)

The folders in the "simulations parent directory, as well as the files they contain MUST'NT be renamed, moved or deleted. The 4 folders for the model cbuilding and simulation are :

Data : here are stored the intermediate data files (used and created during a simulation) and the configuration text files

Images : here are stored the pictures used in the S-block of simulink models

Models : here are located all S-function files of the simulink S-block , and all sripts and functions associated to the models developed.

Simul: here are the simulink models. They are the files that should be opened for the simulation. **This MUST be the current directory** of Matlab/Simulink when running a simulation.

Table 1 : Directories architecture

file	details

Crew subsystem	
../simul/subsystem_crew.mdl	**Simulink Model for the crew susbsystem**
../data/crew_def.txt	Text file for the definition of the crew
../data/planning.txt	Text file for the activity planning of the crew
../models/crew_respiration_dyn_001.m	S-function : models for the calculation of respiration rate of the crew and the gas balance on the crew cabin.
../models/EER_dyn_004.m	S-function : models for the calculation of the Energy Expenditure Rate of the crew as a function of its activity
C4a subsystem	
../simul/subsystem_C4a.mdl	**Simulink Model for the C4a susbsystem**
../models/C_IVa_004.m	S-function : models for the C4a MELiSSA photobioreactor.
../models/composition_spiru.m	This function calculate the compostion the the total biomass of Spirulina for a given light flux, according to.Table 18
../models/wil_reacteur.m	This function calculate the length (or radius) of the reactor at wich the radiant light energy reach the compensation point (fixed at $1W/m^2$).it returns the L_{ut} or R_{ut} value (report to 4.1.2)
../models/wiv_reacteur.m	This function is the returns is the expression that must be integrated to calculate the mean volumetric specific growth rate (report to 4.1.2)
Common files	
./models/load_compounds.m	This script is included in the S-function. It provides the sorted list, compostion and molar mass of the 23 compounds taken in to account in the system.
./models/load_partitions.m	This script is included in the S-function. It provides the values of the partition coefficients for the gas/liquid equilibrium.
../data/???????.mat	They are the storage files for some intermediate calculus, exchange of data and results and checking of programs.
Integrated systems	
../simul/subsystem_C4a_control.mdl	**Simulink Model for a coupled C4a + simple control law**
../simul/MELISSA_5_4a_dyn_005.mdl **../simul/MELISSA_5_4a_dyn_006.mdl**	**Simulink Model for the gas coupled system CREW + C4a**
../models/control_v2.m	S-function : model for the control law.

Table 2 : Files involed in the models for the crew-C4a coupled system.

1) water	9) CO2 + ionic forms	17) CH4
2) NH3 + ionic forms	10) Acetic Acid	18) Inert gas
3) H2SO4 +ionic forms	11) Propionic acid	19) Organic Matter
4) H3PO4 + ionic forms	12) Butyric Acid	20) Nitrosomonas Biomass
5) HNO3 + ionic forms	13) Valeric Acid	21) Nitrobacter Biomass
6) HNO2 + ionic forms	14) Caproic Acid	22) Rhodobacter biomass
7) Urea	15) N2	23) Spirulina Biomass
8) O2	16) H2	

Table 3 : Sorted list of the compounds involved

3 CREW COMPARTMENT

The Melissa modelling approach is a "process engineering" approach, i.e. each operation (compartment, unit operation,…) is studied on a basis of material balances, including coupling between biological and physical phenomena. This is a classical approach for chemical and biological controlled processes. This approach is more difficult to apply to a "human" (or any other animal").

Up to now, the steady-state model used in Melissa simulations is a variable mass balance equation supposed to be an average of input/output mass flows on a man (TN 79.1). The stoichiometric coefficients were calculated in order to achieve the elemental C,H,O,N,S,P balance on the reaction with constraints on the food composition and on the faeces composition. The food itself is defined in order to fit nutrionnal requirements for human.

$$[CHONSP]Food + O2$$

$$\downarrow$$

$$CO2 + Urea + [CHONSP]Faeces + Water + H3PO4 + H2SO4 + NH3 + H2 + CH4$$

$$LiquidWater \rightarrow PerspirationWater$$

Switching from this steady-state description to dynamic inputs/outputs mass flow model of the crew is not a simple task.
First if the gas flow (i.e. respiration, perspiration) can be taken as a continuous process, flows rates and composition are highly dependant of the activities which are not predictable, even if it is possible to plan activities periods (sleep, rest, physic exercises,…). The called "solid" and "liquid" mass flows are discrete events and if inputs (lunch) can be planned it is obvious that outputs flows can not be planned.
Secondly, the mass and elemental balances must be checked, considering that the body mass can also change (especially in microgravity conditions) and can differ from one man to another.

Looking at human models leads often to find ground-based simulation models of weightlessness for reproducing some of environmental conditions encountered during spaceflight (microgravity). Their objectives are linked to the observation of medical parameters and the study of physiological adaptation [2].
The mass flow analysis is often treated as an element of the nutritional analysis, itself being a part of the study of metabolic requirements [1,2]. This approach was chosen to establish a mass flow model for a man.

3.1 Energy metabolism

The definition of metabolism is based upon the body's use of energy. The human body requires a supply of chemical energy for survival. The sources of this energy are:

- the dietary macronutrients – carbohydrate , fat and proteins – which are the bulk of the dietary (consumed) dry matter
- the micronutrients – vitamins, minerals - which are necessary for proper functioning of enzyme involved in energy metabolism and in other key metabolic functions, but do not by themselves provide chemical energy to the body.

As a consequence when looking at energy metabolism, it is sufficient to focus and restrict studies only on macronutrient. But it must be kept in mind that micronutrients remain important on a nutritional point of view.

3.1.1 HUMAN ENERGY EXPENDITURE

The processes of energy metabolism in the human body are highly regulated even if acute changes in dietary energy intake (metabolic inputs - Figure 2) do alter energy expenditure to a small degree.

3.1.1.1 Nutrition and energy

The body's ability to use all three macronutrients to meet its energy needs allows for high flexibility in the composition of the diet. There are however limits on the percentage of dietary energy that can be derived from each macronutrient.

Proteins: a minimum level of high–quality proteins is required to maintain nitrogen balance and prevent excessive loss of body protein (muscle homeostasis) present recommendation are for a protein intake that is **12% to 15%** of total calories consumed and reflects an animal-to-plant ratio of approximately 60:40 [2]. The mix of animal and plant sources is to ensure adequate intake of all essential amino acids. High level of proteins will increase the urinary urea production as a result of the disposal of excess nitrogen.

Fats: to reduce the health risks (cardiovascular diseases) associated to high fat diet, about 30% to 35% of calories should be provided as lipids, mainly triglyceride [2]. The distribution among fatty acyl moieties for polyunsaturated:monounsaturated:saturated is 1:1.5 to 2:1. A relative high level of fat must be maintained for several reasons, one of them being fats provide an important component of taste and thus increase the palatability of food.

Carbohydrates: from general recommendation adults should obtain approximately half their caloric intake as carbohydrate (in order to not increase the uptake of proteins and fat above recommendation level) from diverse sources so that most carbohydrate are complex, digestible form i.e. starch from cereal products. Less than 10% of the total (<5% of calories) should be provided as simple sugar (such as sucrose).

In Table 4 to Table 7 are compiled some data for diet and energy metabolism from various sources. More or less, it can be observed that recommendations for crew are applied in the diverse spaceflight mission. In table 4 it is interesting to compare the estimated Total Energy Expenditure and the energy intake. In short mission (shuttle), it can be noticed an unbalance in the energy (intake<expenditure) while in long mission (Skylab) intake and expenditure are balanced. Skylab results are more representative of future and planned long term mission
At this time the comparison between space and Earth energy requirements concludes that there are not dramatically differences [2]. Energy expenditure and hence energy requirements to maintain energy balance are similar to what they are on Earth. Energy requirement can therefore be estimated using the predictive equation that have been developed for one g. gravity. These are the World Health Organization (WHO) requirements for human involved in moderate activity, which can be estimated as [2]:

Man
TEE = 1.7 BMR
With BMR (in MJ:d) = 0.063 *body weigth (kg) + 2.896 for 18 to 30 years
BMR (in MJ:d) = 0.062 *body weigth (kg) + 2.036 for 30 to 60 years

Woman
TEE = 1.6 BMR
With BMR (in MJ:d) = 0.048 *body weigth (kg) + 3.653 for 18 to 30 years
BMR (in MJ:d) = 0.034 *body weigth (kg) + 3.538 for 30 to 60 years

macronutrient	RQ	Energy (kJ/g)	Needs (g/d)	Needs (% diet)
Carbohydrates	1.0	17.22	300-600	50-60
Proteins	0.8	17.22	50-150	10-18
Fat	0.7	39.06	50-300	20-40
European diet	0.8	-	-	-

Table 4: Nutritional requirement for the three macronutrients. The respiratory quotient is a characteristic of the oxidation of each of the macronutrient class: $RQ = \frac{CO_2}{O_2}$

	Reference.	[a]	[b]	[c]	[d]	[d]	[e]
	Body mass (kg)	70,3	-	-	50	80	
	needs kcal/j	2700	2800	3021	3632	5362	3000
Daily needs							
Carbohydrates	g/d	383		420,6	495,3	735,8	415
Protein	g/d	105		126,2	152,4	226,4	118
Fat	g/d	56		92,7	114,3	169,8	97
Total	g/d	544	618	639,5	762	1132	630

Table 5 Nutritional constraints and energy expenditure. [3]. [a] Krause M.V. et al., 1984. ; [b] Gustan E. Vinopal T. Boeing Aerospace Co, 1982. ; [c] Sauer. R.L., 1985. [d] Condran M.J., 1989. ; [e] Toki A.et al., 1994.

	Vostok		Voskhod		Soyuz		Skylab	
	g/d	%	g/d	%	g/d	%	g/d	%
Protein	100	14,9	150	17,2	139	20,4	111+/-18	15,5
Fat	120	40,3	130	33,5	88	29	86+/-14	27
Carbohydrate	300	44,8	430	49,3	345	50,6	412+/-60	57,5
Energy Expenditure (kcal/day)	2680		3490		2728		2865+/-307	

Table 6 Diet and average energy expenditure estimated in past spaceflight missions [3].

Vostok	**Gemini**	**Salyut**	**Space Shuttle**				**Slykab**			
TEE	TEE	TEE	TEE		E intake		TEE		E intake	
1914	2201	2297	2703	+/-22%	2177	+/-21%	2727	+/-13%0	2679	3%
			2799	+/-16%			2871	+/-3%0	2943	18%
							2990	+/-5%0	2943	2%

Table 7 average energy expenditure(TEE) estimated by variousmethods and Energy intake estimated from crew diet in past spaceflight missions [2]. Energy in kcal/d. +/- SD

3.1.1.2 Components of energy expenditure

The Total Energy Expenditure (TEE) is the energy needed to support the human body and activity. Energy is supplied by diet (metabolic inputs – Figure 2) to balance the TEE taking into account also that nutrient are not completely metabolized (metabolic output). Then about 9% of energy provided by diet is recovered in faeces and urine [7].
Even if it is measured the interpretation of TEE often requires information on its components: Basal Metabolic Rate (BMR), Thermic Effect of Meals (TEM) and physical Activity Energy Expenditure.

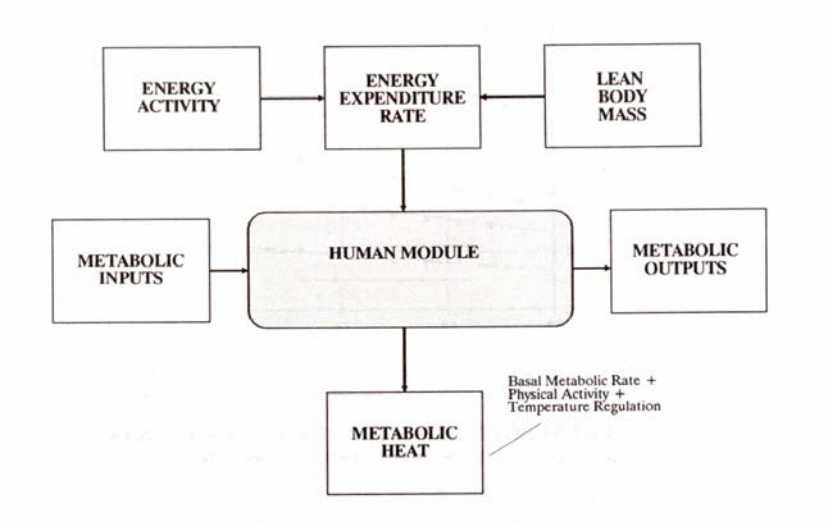


Figure 2 The schematic of human expenditure rate [3].

Basal Metabolic Rate : It represents the energy cost for the biochemical and physiological processes that maintain life. Because it is difficult to meet the stringent requirement for measuring BMR (12 to 15 hour fast at rest and at thermal neutrality) a Resting Metabolic Rate (RMR), what is is simply the rate of energy expended by an individual at rest, is usually measured. RMR closely approximate BMR after a time of the meal longer than 6 hour. RMR is the largest component of TEE accounting for 50 to 70%.Thus small changes in RMR affect TEE. RMR is altered by:

- Nutrition : overfeeding increases RMR 5% to 10% and underfeeding reduces RMR to 10% to 20%
- Sleeping decrease RMR to 5% and anxiety increases RMR to 5% to 10%
- RMR is affected by ambient temperature. Below 27°C, there 2% increases by degree.

Thermic Effect of Meal: it is the smallest component of TEE accounting for 6% to 10%. More than 60% of TEM is considered as obligatory as it is the direct result of the metabolic cost of the processing and storing macronutrients of meal. TEM is often assumed to equal 10% of the intake (i.e. roughly equivalent to 10% of the sum RMR and AEE)

Activity Energy Expenditure: it is the most variable component of AEE, comprising 20% to 40%. It directly depends on the intensity of frequency and duration of physical activity. It is also dependant of the weight of the individual and then is affected by microgravity conditions.

3.1.2 HUMAN METABOLISM MASS FLOWS

A description of human inputs and outputs mass flows (i.e. metabolic inputs/outputs) was summarized in Figure 3. The diagram presents the elements required to meet the metabolic need of an individual and the waste produced.

This diagram can be use as a support for developing a "human mass flow" model. Even considering here human metabolism as a "black box", it illustrates the problems of the quantity and of the quality of compounds involved. At a mass balance point of view, it is difficult to model the mass flows as they can be continuous (respiration) discrete (feed, drink,..) and the body mass can be variable with the time. Moreover as complex compounds are involved it add a difficulty when expecting an elemental balanced flow model.
Moreover, even the continuous flows are not constant flows as they are function of the environmental conditions and of the energy expenditure rate, itself linked to the activity and the metabolism of the individual.

At the end it is difficult to establish a reliable and predictable model of human mass/elemental flow.

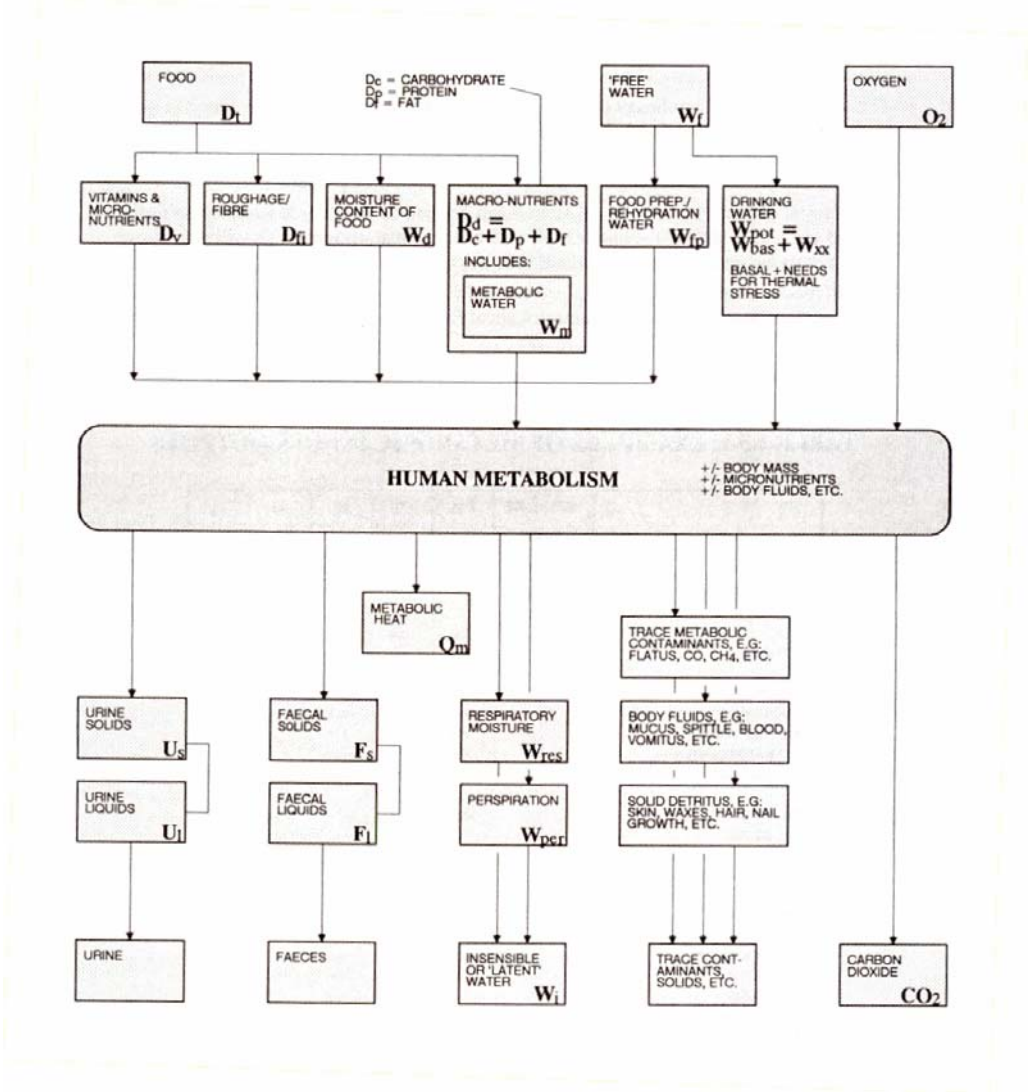


Figure 3 : Summary of metabolic inputs and outputs . [3] developed from Condran et al. [1]

3.1.3 ACTIVITY AND ENERGY RELATIONS

It is obvious that a heavy link exists between human mass flow and human metabolism (Figure 2). As well human metabolism and energy expenditure are linked. Thus relations must exist between human energy expenditure and mass flow. Some of these relations have been studied by Condran et al. 1989 [1], and reported in the Life Support and Habitability Manual ESA reference document [3].

3.1.3.1 Energy Expenditure and activity

Table 8 and Table 9 illustrate the variation of the energy expenditure as function of the activity. The activity planning affects the daily Total Energy Expenditure and then affects the nutritional requirements (report to 3.1.1.1), which are discrete events on a mass flow point of view. Activity affects also directly the continuous mass flow rate (respiration, perspiration) or energy flow rates (thermal exchanges). Such tables are useful if we want to translate human activity into a variable (hereafter Energy Expenditure) that can be use to correlate activity with mass flows.

Crew Activity	Energy Expenditure (kcal/h)	Duration	(h/days)
Sleep	73	7,5	
Pre et Post-sleepl	75	0,5	
Leisure activities	77	2,5	
Personal Hygiene	150	0,5	
Lunch	85	1,5	
Exercise	250	0,5	
Material Maintenance	145	3	
Laboratory activities	150	8	
Extra vehicular activity (EVA)	250		6
Space suit (taking/removing)	275		0,5
EVA transfer	225		0,5
EVA Preparation	200		1
Energy Expenditure kcal/day		2740	3490
Average Energy Expenditure (1/3 EVA)		2990	

Table 8 : Activity and Energy Expenditure (Life Support and Habitability manual, 1990 [3]).

Activity	a (1)	b (2)	Activity details
Exercise	6.4	1.4	Combination of 50% running at a pace of 8.785 km.h-1 ; 25% stationary cycling ; 25% weight lifting
Work	3.5	0.09	Similar to electrical bench type
Leisure	2.5	-0.09	Equivalent to playing billards
Sleep	1.5	-8.5	Rest ; lying at ease

Table 9 : Energy Expenditure Rate coefficients. (EER = a LBm + b) LBm is the Lean Body mass in kg., EER is the Energy Expenditure Rate in kcal.h^{-1} [1] - (1) in kcal.person^{-1}. h^{-1}. kg^{-1} ; (2) in kcal .personn^{-1}.h^{-1}. ** The lean body mass is basically The mass of the body minus the fat (storage lipid) – See Annex 1.**

3.1.3.2 Energy Expenditure and mass flows

Oxygen / Carbon dioxide

The amount of oxygen consumed and of the carbon dioxide produced depend on total energy expenditure primarily and to a lesser extend on composition of the diet (i.e. demand in oxygen for the oxidation of the three classes of macro-nutrient is not the same – see Table 4). It can be outlined that in the steady-state crew model developed for the MELiSSA loop model (TN

79.2), the two effects are implicitly taken into account, the yields O_2/diet and CO_2/diet being calculated from a stoichiometric equation in which the diet composition is variable, and the quantity consumed or produced being dependant on the diet quantity which is calculated from an average total energy expenditure.
Even if basal oxygen demand increases in a non linear way with body weight, the physical activity, which is the main variable for energy expenditure, is also the main variable affecting the oxygen demand and the carbon dioxide production. In Table 10 are reported the relations proposed by Condran et al. [1] for the oxygen consumption rate and the carbon dioxide production rate , as function of the Energy Expenditure Rate (EER). It can be noticed that the respiratory quotient is not the same for the different activities as substrate used are also different: carbohydrates for exercise (muscular activity) , fat and basal metabolic synthesis for sleep.

Activity	ao	bo	ac	bc	RQ
Exercise	0.28	0.1	0.39	0	1
Work	0.29	0	0.34	0	0.85
Leisure	0.3	-0.1	0.31	0.1	0.75
Sleep	0.31	0	0.29	0	0.7

Table 10 : Oxygen (ao,bo) and carbon dioxide (ac, bc) coefficients [1]. Compounds expressed as mass flows ($g.p\text{-}h^{-1}$) function of EER (in $kcal.p\text{-}h^{-1}$). = a.EER + b

Diet

The total food (Figure 3 : macronutrient, fibre, micronutrient, moisture) is mainly composed of macronutrients (report to 3.1.1.1). The total macronutrient mass flow can be calculated from the total daily energy expenditure rate (Table 4) if the relative ratios of the macronutrients are defined and fixed. But it must be noticed that this approach is theoretical as diet is defined from a more complex nutritional approach and that recipes cannot be defined with a strict fixed composition.
Fibres are compounds that are considered as not digestible and then are not use to produce energy. As they are not metabolised they can be considered as an "inert material" regarding on the "human metabolism" and must be recovered in faeces. Implicitly this means that faeces mass flow is at lest equal to the fibre content of the diet.
Micronutrients flows rates are negligible even if they are important on a nutritional point of view.
Moisture is not a negligible flow, but is highly dependant on the food used (fresh, cooked, dehydrated/re-hydrated) and is difficult to estimate in absence of detailed menu for a diet.

Water

There is several sources of water: the moisture content of food, the water used in recipices and food preparation, the drinking water used to cover basal metabolic needs and to compensate thermal stress. The free water (drinking+food preparation) is assumed as a function of the Energy Expenditure Rate [1], even if it is obvious that for food preparation water is more linked to the recipices than to EER.

At least, the water sources must fit the physiological needs. These needs are estimated in the range 1.8 – 2.8 $kg.p^{-1}.d^{-1}$ [3]. For the design of the Colombus ECLSS, a lost of 2.28 kg by perspiration, of 1.5 kg by urine and 0.09 kg by faeces were considered, leading to at least 3.87 $kg.p^{-1}.d^{-1}$ of water required for physiological needs [5].
The water sinks are urine (at least 0.3kg but in normal condition about 1.5 $kg.p^{-1}.d^{-1}$), faeces (about 0.1 $kg.p^{-1}.d^{-1}$) and latent water due to perspiration and respiration moisture. These two last losses are highly depending on activity and thermal environment as they are related to temperature control of the body.

Faeces and urine (solids - minerals)
Faecal solids mass flow are assumed to be proportional to some residual percentage of dry food plus the total quantity of the fibre (non digestible).
Considering some data faeces solids represent between 3.75% [3] and 4.29% [6] of the dry food. But in these calculations, "dry food' and faeces include fibre, and then the values are not residual percentage of the "digestible" food in faeces. A study of a fibre free diet leads to a faeces wet weight of 86+/-25 $g.d^{-1}$ (15+/-2 $g.d^{-1}$ of dry solids) [7] while a similar study with a less digestible diet based largely on dried and processed foods gives 138+/-17 $g.d^{-1}$ of wet faeces and 41+/-1 $g.d^{-1}$ of dry solids.

Urine solids are mainly urea (30-50 $g.d^{-1}$) and inorganic ions or salts (10 – 20 $g.d^{-1}$) [7]. Urine solid can, as faeces solid, be considered proportional to the dry food, but such relation must be carefully considered as urine solids (urea and ammonia salts) are linked to the N-metabolism and then are affected by diet quality (especially for N-rich food).

Waste	[kg p^{-1} d^{-1}]	Waste	[kg p^{-1} d^{-1}]
Carbon dioxide	1.00	Water for hygiene	12.58
Perspiration	2.28	Hygiene (solids)	0.01
Urine (solids)	0.06	Latent water (food preparation)	0.04
Urine (liquids)	1.50	Latent water (experiments)	0.454
Transpiration (solids)	0.02	Latent water (whashing)	0.06
Faeces (solids)	0.03	Water (washing)	11.90
Faeces (water)	0.09	Washing (solids)	0.08
Metabolic water	0.354	Charcoal (filters)	0.059
"water flush"	0.5	Containers	0.454
Latent water (hygiene)	0.43	Other waste	0.817

Table 11 : Waste produced by crew activities [5].

3.1.4 PROBLEM OF THE APPLICATION OF THE THEORETICAL FORMULA FOR EER CALCULATION

Before the use of the data previously presented, it is important to check their consistency.

First we can check if the daily average Energy Expenditure Rate calculated using the WHO relation (report to 3.1.1.1) and the one calculated from the sum of the EER for various activities are of the same order of magnitude. An example is given in Table 12 for an individual of 70kg and a standard activity planning.

Crew member					
weight	70	kg	Lean Body Mass method of James (LBM1)	55,3	kg
age	30	years	Lean Body Mass method of Hume (LBM2)	51,1	kg
height	170	cm			
EER (WHO)	**2971**	**kcal/day**			

		EER (kcal/h)			RQ
Activity	duration(h)	from Table 9 and LBM1	from Table 9 and LBM2	From Table 8	From Table 10
sleep	8	74	68	73	0,7
leisure	4	138	128	77	0,75
work	10	194	179	150	0,85
physic	2	355	329	250	1
Total	24	**3795**	**3503**	**2892**	**0,80**

Table 12 : Daily average EER for an individual of 70kg calculated using different methods. The Lean Body Mass is calculated using relation given in Annex.

Secondly we check the Respiratory Quotient, which is of great importance in coupling crew with C4a. The Human average RQ reported range from 0.8 to 0.9, but it is often considered that the average should be closed to 0.8. This is consistent with the value obtained from the sum of the different activities (Table 12). But considering the RQ predicted from the food composition, (Table 13), it would be probably difficult to fit accurately at the same time RQ calculated from EER and mass balance on food assimilation (i.e. metabolism).

	Mass %	C-mol	Mass %	C-mol	Mass %	C-mol
Proteins	18%	0,0088	15%	0,0072	15%	0,0072
Fat	14%	0,0090	20%	0,0128	40%	0,0256
Carbohydrates	68%	0,0226	65%	0,0217	45%	0,0150
QR food	**0.90**		**0.89**		**0.82**	

Table 13 : Average RQ calculated for complete oxidation of macrocomponents, assuming proteins=casein, fat=tristearin and carbohydrate=glucose.

Conclusion : **1** - If the daily EER predicted by the WHO equation is closed to the usually cited values for space mission, the EER calculated from activities is up to 128% higher, but is consistent with the highest values reported. In Table 12, the main discrepancies seems linked to the EER for physical activity and leisure. The method for calculating the Lean body mass

has also a non negligible effect. It is possible that the definition of the activities is not always the same (leading to various values of EER). Relations of Table 9 are interesting but need probably deeper investigation to be used accurately. Then, if using activities to predict EER, the value taken from Table 8 seems more accurate at this time.

2 – The influence of the food composition is not taken into account in the relation predicting the RQ from the EER. This may lead to inconsistencies when trying to establish a mass balance for the complete human metabolism (i.e. Food metabolized to CO_2 and waste)

3.2 A dynamic mass flow model for a crew

3.2.1 THE PROBLEM OF MASS BALANCE AND ELEMENTAL BALANCE

Data and "models" for a crew are mainly daily mass flow and energy expenditure. Such results are useful for developing steady-state models (such as for the current MELISSA loop Matlab/Simulink model [4] but are more difficult to use for a dynamic approach, especially if it is intended to simulate and control mass flow at hour or minute level.
If for solid and liquid, approximation of models on the basis of daily dynamic can be done, it is not possible for the gas (if we expect to have consistent model). The crew atmosphere must be controlled with dynamics at least at minute level (accumulation of carbon dioxide is quickly lethal).

The problem is complicated when expecting an elemental balance as it supposes to fit stoichiometrically (report to 3.1.4):

- the metabolism with consistent complex macromolecules, operated in a discrete mode
- a continuous respiration rate of variable intensity
- and a variable body mass and composition

Moreover, even if it is planned, the crew activity is not strictly predictable, and then the metabolic activity is also not strictly predictable.

3.2.2 CREW MODEL PRINCIPLES

Considering the different elements previously listed, the following approach for the development of a dynamic model for the crew was chosen.

> **First we choose to use the Energy Expenditure Rate as the basis** of all mass flow calculation. The EER is itself calculated from correlations between EER and the crew members activity.
>
> **Secondly we consider the two human functions "respiration" and "food assimilation"** (or general human metabolism) separately. The input/outputs of "respiration" are continuous flows while for the "metabolism" the inputs/outputs should be discretes (eating, drinking, ..)

In order to simplify the problem, the compounds involved (metabolic inputs of Figure 3) are restricted to water, food (proteins, fat, carbohydrate, fibers), O_2, CO_2, Faeces (proteins, fat, carbohydrate, fibers), urea, minerals (NH_3,$H2SO_4$,$H3PO_4$), and trace gas (H_2,CH_4), as in the previous MELiSSA loop study [4].

The principles for a crew model are presented in Figure 2. From the definition of the crew members and of their activity planned for 24h, the hourly Energy Expenditure Rate of each member can be calculated. These calculations can be done by two methods, the first is based on the relation of the Table 9, and the second is based on the values reported in Table 8. Accordingly to the observation of chapter 3.1.4, the second method seems more reliable at this time.

From hourly EER, can be calculated:

- The daily average EER, which can be used to determine the quantity of food required to fit the EER requirement
- The oxygen consumption rate and the carbon dioxide production rate using relations of Table 10. This can be considered as the "respiration" function of the crew, which gives mass flows for the gas phase.

The food and water daily input mass flows, associated to the cumulated oxygen consumed and carbon dioxide produced over 24h are used to establish a mass balance (i.e. mass and elements over an average of one day are neither accumulated nor lost). But within the 24h period, variations due to respiration are assumed to be associated to the body mass and composition variation. This mass balance can be considered as the "metabolism" of the crew member (i.e. digestion, energy production for activities,…) and leads to produce the solid (faeces) and liquid (urine) waste.

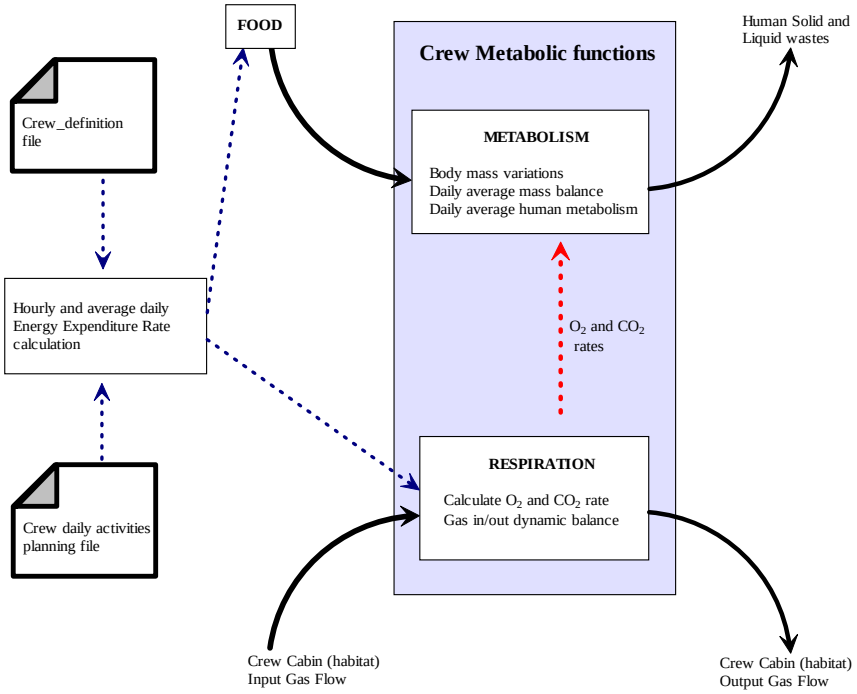


Figure 4 : Principles for a crew/human model

3.3 Crew simulink model description

3.3.1 CREW SUBSYSTEM OVERVIEW

For the current work the model was restricted to the "respiration" part of the complete crew model. The metabolism will be required when C1 compartment will be modeled and for a complete MELiSSA loop.
The Simulink model for the crew (in *file* ***subsystem_crew.mdl***) is presented in Figure 5.

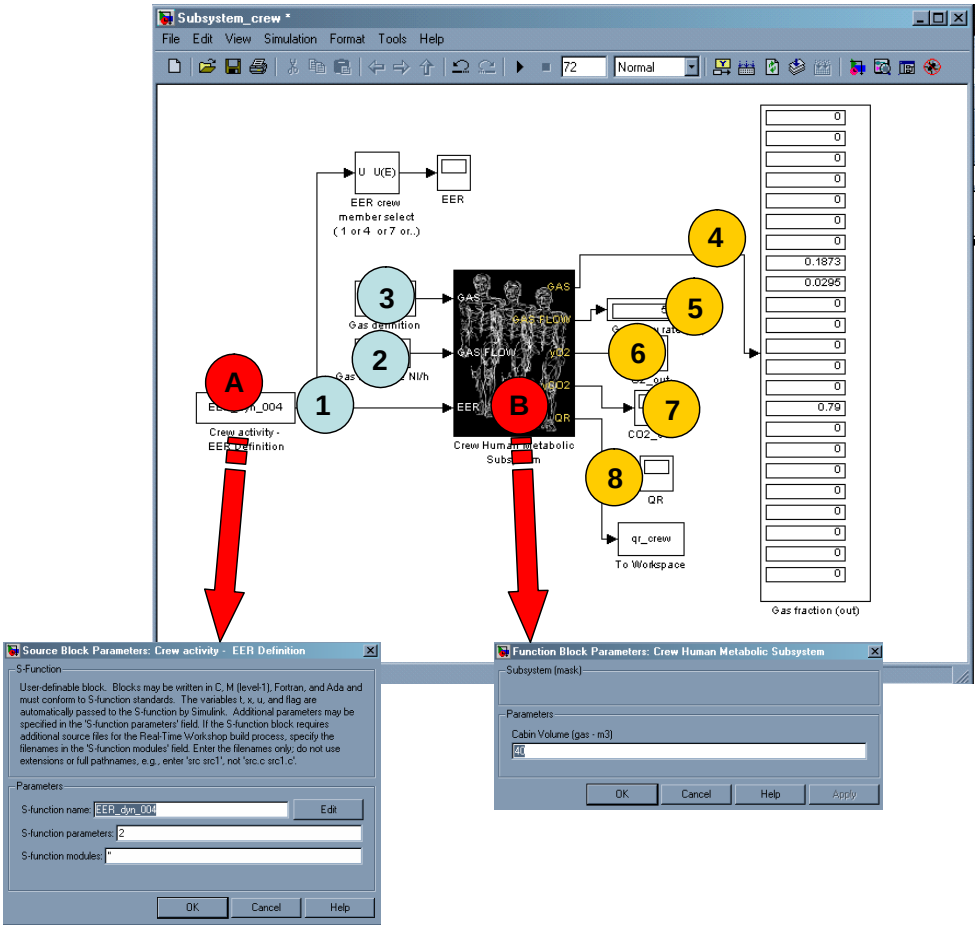


Figure 5 : Crew subsystem under Matlab/Simulink – A] is the block associated to the calculation of the EER with a dialog box allows manipulation of the parameters. B] is the subsystem associated to the crew metabolic activity. It only includes at this time the "respiration" function. A dialog box allows manipulation of parameters 1] is the hourly EER (kcal/h) and average daily EER (kcal/d) of each crew member. 2] is the gas flow rate (in Nl/h). 3] is the gas composition in molar fraction for the 23 compounds. 4] is the gas composition in molar fraction at cabin output. 5] is the output gas flow rate in Nl/h. 6] is the O_2 molar fractions in the gas output (i.e. in the cabin). 7] is the CO_2 molar fraction in the gas output (i.e. in the cabin). 8] is the respiratory quotient.

3.3.2 CONFIGURATION FILES AND STORAGE FILES

As presented in Figure 4, there are several files associated to the "crew model". These files are stored in the ***"../data/"*** directory (report to 2.2).

3.3.2.1 Configuration files

These file can be (must be) modified for simulating different scenarii.

crew_def.txt : text file containing the definition of the crew. The file must have the format detailed in Table 14. This file is used to define each crew member (weight, height, age) and identify each member by a number. These data are required to calculate the Lean Body mass (Annex 1) and may be required in some correlations. This file fixes also the crew size (number of members).

Definition of a crew of several members (each row corresponds to a member). The first column member identification number				Example for a crew of 2 member			
Member N°	weight(kg)	height(cm)	age (years)	1	70	170	30
Member N°	weight(kg)	height(cm)	age (years)	2	80	190	40
Member N°	weight(kg)	height(cm)	age (years)				
……							

Table 14 : Format of the crew_def.txt file.

planning.txt : text file containing the planning activity of each crew member for a 24h day. Note that the same planning is considered for each day. The file must have the format detailed in Table 15. It is assumed that the activities listed are sorted for each member from 0h to 23h in a day.

Definition of crew members daily activities. Each row details an activity for a crew member.			Example for a crew of 2 member
Duration (h)	activity number	crew member number	8 1 1
Duration (h)	activity number	crew member number	1 2 1
Duration (h)	activity number	crew member number	3 4 2
…….			…..
Duration (h)	activity number	crew member number	1 2 2

Table 15 : Format of the planning.txt file. The code for the crew activity is 1=sleep ; 2=rest/Leisure ; 3=physical ; 4= work. As defined in Table 9. The crew number is the one defined in crew_def.txt (Table 14).

3.3.2.2 Storage files

These files are used by models algorithms for storing some intermediates calculus and exchanging results within Simulink blocks and models. It is not the most efficient way for exchanging data (disk write/disk read is a relatively slow process), but it is the simplest one. These files have a Matlab format and mustn't be edited or changed.

Calculs_crew.mat : contains the planning data read in the ***planning.txt*** file and the EER values corresponding to each planned activities of the crew members. This file is created each time the simulation is started.

Calculs_respiration.mat : contains data such as RQ and gas flow rate calculated in the respiration model.

3.3.3 EER BLOCK AND MODEL (EER_DYN_004.M – ANNEX 2)

This is the subsystem A] in Figure 5. This is a block for the S-function ***eer_dyn_004.m*** stored in the ***"../model/"*** folder (report to 2.2).

Aims of the function : the aims of the function is to load the definition files crew_def.txt and planning.txt and to establish for each crew member their "agenda" for one 24hours/day. Using correlations between activity and EER (report to 3.1.3.1), the hourly EER and the daily average EER of each member can be calculated. These values are used as information fluxes for further crew subsystems.

Algorithm of the model: The algorithm of the function is detailed in Figure 6. Two methods can be used to calculate EER function of the activity. The first one based upon relations of Table 9, with the methods of Hume for the calculation of the Lean Body Mass, the second with fixed value for activities dependant EER (Table 12).

Parameters of the function: One parameter must be given which is the code of the methods selected for the calculation of the hourly EER for each activity. The code number is 1 or 2 and is given through the dialog box (Figure 5 – A]).
Code 1 is for the use of relations of Table 9, with a Lean Body mass calculated by Hume relation (Annex 1)
Code 2 is for using the values fixed in Table 12.

Inputs/outputs of the S-block: There are no inputs on this block. It works as a "signal generator", giving to further crew S-blocks the values of the current hourly EER of each member. The size of the output is function of the number of members in the crew (then is defined by the crew_def.txt file). The output is sorted by crew members, for which 3 information are available:

- the current hourly EER (in kcal/h),
- the average daily EER (in kcal/day),
- the code for the current activity of the crew member (1=sleep; 2=rest/leisure ; 3 = physical ; 4=work).

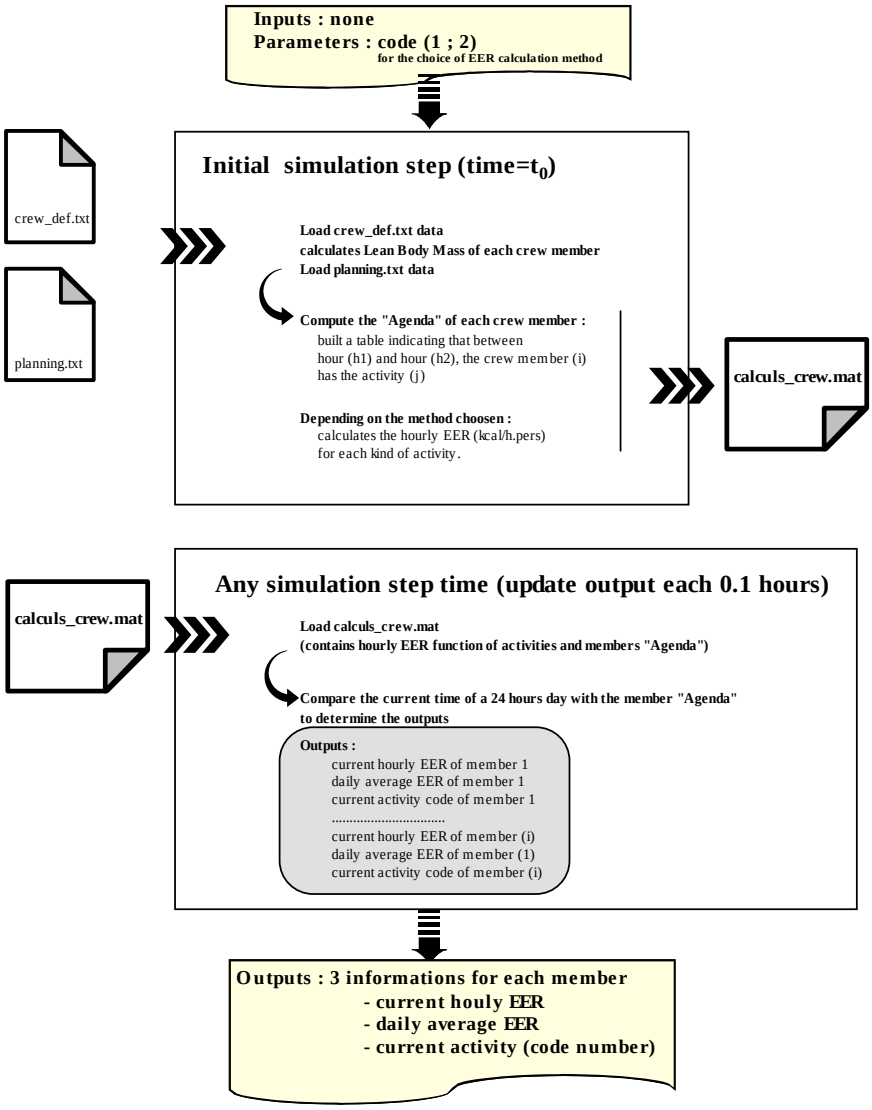


Figure 6 : Algorithm of the EER crew member calculation (eer_dyn_004.m S-function). Note that the update period of 0.1h can be changed in the initialization subroutine of the function.

3.3.4 RESPIRATION BLOCK AND MODEL (CREW_RESPIRATION_DYN_001.M)

This S-block is a part of the crew subsystem B] in Figure 5. You have to "look under the mask of the subsystem" to access this S-block. But the subsystem is designed in order to be manipulated directly through its inputs/outputs and subsystem dialog box ((Figure 5 – B]). The crew respiration S-block associated to the S-function ***crew_respiration_dyn_001.m*** is stored in the ***"../model/"*** folder (report to 2.2).

Aims of the function: the function predicts the oxygen consumption rate and the carbon dioxide production rate of the crew from the hourly EER of each crew member. For this purpose it uses the relations presented in Table 10. With these rates the function calculates the variation of the composition of the cabin atmosphere. The balance implies gas input (defined in terms of composition and flow rates) and a cabin volume which is assumed "perfectly mixed".

Algorithm of the model: The algorithm of the function is detailed in Figure 7. This is a dynamic model which is built as a gas balance model on the crew cabin. For each compound, the dynamic equation is of the form :

$$\frac{dyi}{dt} = \frac{y_i^{input}.G_{input}. - y_i\, G_{output} + rx_i}{total\_gas\_mole\_in\_cabin}$$

y : molar gasfraction

Ginput and Goutput are the molar flows rates. It is assumed that Ginput=Goutput

Total_gas_mole_in_cabin is the number of mole in the cabin. It is assumed that this number is constant (i.e. no pressure variation)

rx_i is the reaction rate of compound i (they are reduced here to the oxygen and CO_2 rates)

Note that even if the EER input is a discrete data (non continuous value), the model is a continuous mass flow model. The hypothesis of a constant molar flow rate in input and output is obviously false (RQ≠1) and would be improved in further developments. At this time the hypothesis is taken for practical reasons (direct linking between crew and C4a), reducing the number of variables in the system.

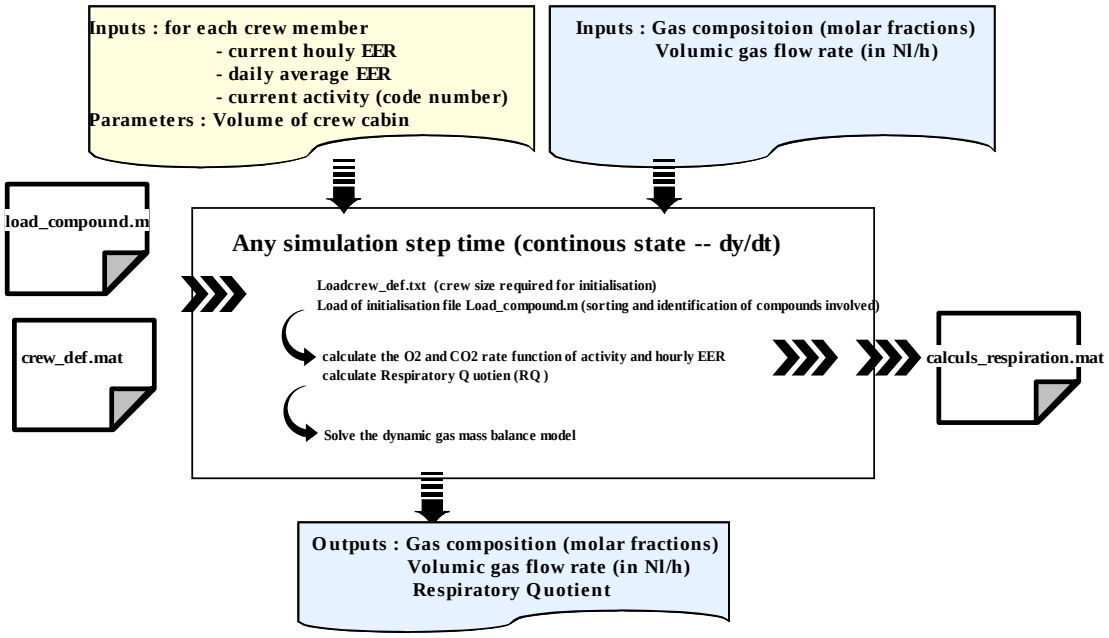


Figure 7 : Algorithm of the EER crew member calculation (eer_dyn_004.m S-function)

Parameters of the function : the cabin (gas) volume (in m^3) must be given. It gives the total moles of gas in the system and has a non-negligible impact on the output gas fraction (dilution effect).

Inputs/outputs of the S-block: There are two kinds of inputs.
The first is the information about the crew EER and activity. This input is produced by the previous EER S-block (3.3.3) (Figure 5 – 1]).
The second kind is for the gas flow input on the crew cabin. It includes the gas composition, in molar fraction (Figure 5 – 3]) and the volumetric gas flow, in Nl/h, (Figure 5 – 2]).
The outputs are the gas flow at the output of the cabin: gas composition and gas volumetric flow rate in Nl/h (Figure 5 – 4] and 5]). The average current Respiratory Quotient of the whole crew is also obtained (Figure 5 – 8]). From the gas composition flux are also extracted the O_2 molar fraction (Figure 5 – 6]) and the CO_2 molar fraction (Figure 5 – 7]) which can be further use as "measurement" values.

3.4 Tests and results of subsystem_crew.mdl

The Matlab /Simulink model for the crew subsystem (restricted to the respiration function) was tested for a crew of two members. Member definition and activity planning were detailed in Table 16.

Member definition (crew_def.txt)		**Member activity planning (planning.txt)**
Crew member 1		
Weight Height Age	70kg 170 cm 30 year	8 hours of sleeping (code activity 1) 1 hour of rest/leisure (code activity 2) 3 hours of work (code activity 4) 1 hour of physical exercise (code activity 3) 2 hours of rest/leisure (code activity 2) 5 of work (code activity 4) 2 of rest/leisure (code activity 2) 1 of physical exercise (code activity 3) 1 of rest/leisure (code activity 2)
Crew member 2		
Weight Height Age	80kg 190 cm 40 year	8 hours of sleeping (code activity 1) 1 hour of rest/leisure (code activity 2) 3 hours of work (code activity 4) 1 hour of physical exercise (code activity 3) 2 hours of rest/leisure (code activity 2) 5 of work (code activity 4) 2 of rest/leisure (code activity 2) 1 of physical exercise (code activity 3) 1 of rest/leisure (code activity 2)

Table 16 : Crew definition for tests

In Figure 8 are reported the EER predicted for the crew activity and the evolution of the gas fraction in the cabin for a cabin volume of $40m^3$ and a fresh air flow rate of 5 m^3/h. For EER calculation, the second method (parameter=2 ; see chapter 3.3.3) was selected.

In Figure 9 can be compared the differences between the two methods for EER calculation. The shape is not changed, but the EER values are different (also affecting the O_2 and CO_2 fraction).

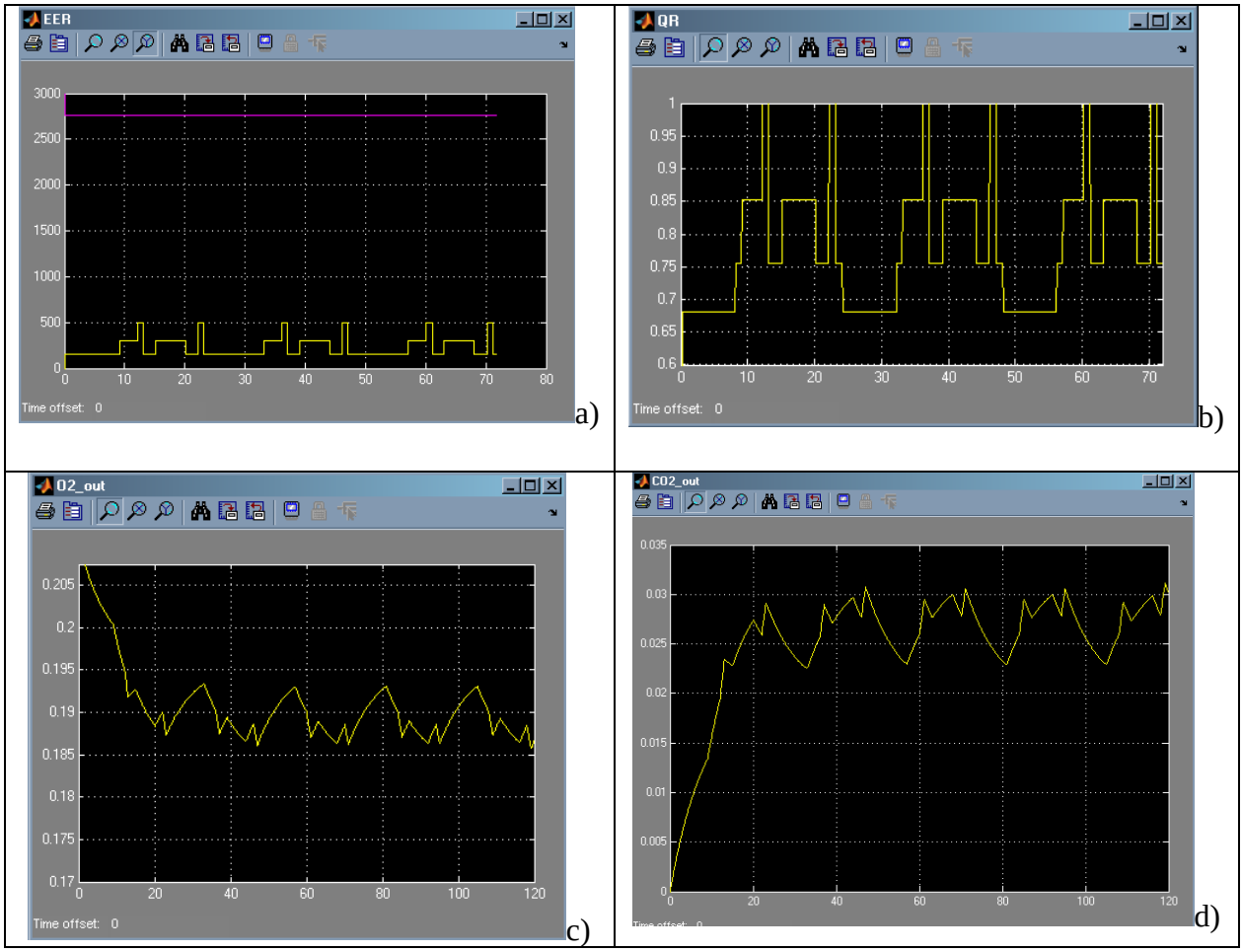


Figure 8 : simulation curves for the crew subsystem. a] hourly (yellow curve) and daily average EER. b] RQ (hourly value, corresponding to various activities). c] O_2 molar fraction; d] CO_2 molar fraction.

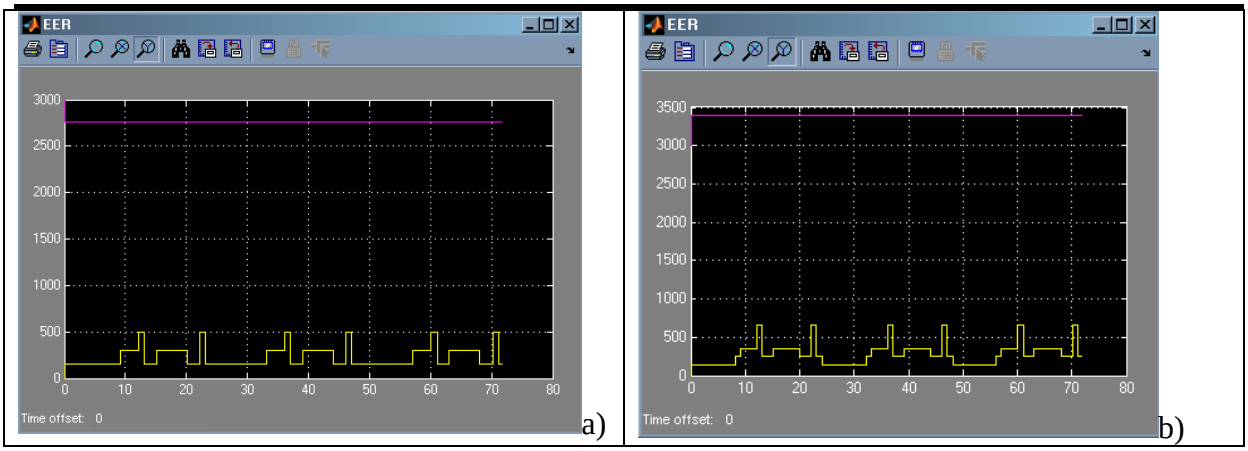


Figure 9 : EER predicted by method 1 (function of Lean Body Mass) and method 2 (fixed EER for each activity)

REMARKS :

1 – the computing step time is about 0.1 hour, as this is the period fixed for updating the discrete events (i.e. the activity of the crew).

2 – considering the gas fractions, the sum at the output is not equal to 1. This is due to the assumption that output flow rate is equal to input flow rate. This is required to be changed in improved versions of the subsystem.

4 C4A PHOTOBIOREACTOR

The C4a MELiSSA compartment is one of the most well-known compartments of the MELiSSA loop. It was completely defined and validated as a single process. The dynamic model was developed by J. F. Cornet and released as PHOTOSIM software ([8] - TN 19.2). This is this model that was translated and adapted for developing a Matlab/Simulink S-function.

4.1 Photosim model

4.1.1 BASIS OF THE PHOTOBIOREACTOR MODEL

The models developed by Cornet ([8], [9], [10]) assume the reactor is perfectly mixed. The models consider that the main limiting factor is the light. Thus the model is first based on a law predicting the light radiant energy distribution in the photobioreactor and its coupling with the micro-organism growth rate. The rate of all compounds involved (O_2, biomass,etc..) are further linked to the light dependant rate by the stoichiometric yields, then the model is implicitly mass-balanced.
In addition to the light limitation, a limitation due to the gas/liquid transfert is taken into account in the model. It is important if we consider that the carbon source is the gas CO_2.
Another physical process should be taken into account: the pH. This is a critical point especially if the carbon source is the bicarbonate form of the dissolved CO_2. But the prediction of the pH evolution is a complex task as lot of compounds are involved in the pH calculation.

4.1.2 LIGHT DEPENDANT GROWTH RATE MODEL

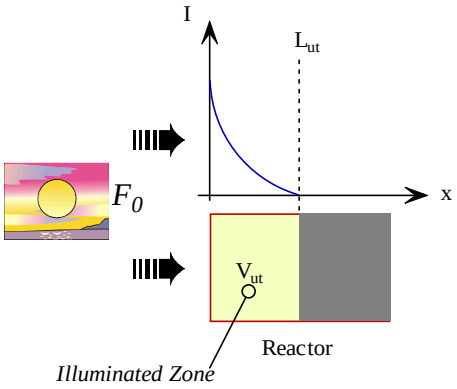
The prerequisite of the model is to establish the radiant light energy profile. The profile is obviously dependant on the reactor shape, on the lightning and on the medium (i.e. the biomass concentration in the reactor). The Matlab/Simulink model proposed here includes the updated 3 models of PHOTOSIM for the following 3 reactors:

- ➔ Rectangular - one side illuminated ;
- ➔ Rectangular - 2 opposite sides illuminated;
- ➔ Cylindrical with radial lightning.

The relations given for the calculation of a mean volumetric specific growth rate can be applied to any kind of growth rate (total biomass, active biomass, exopolysaccharide,…)

4.1.2.1 Rectangular photobioreactor – single light

The profile of the Light radiant energy I_Σ (in W/m^2) in the photobioreactor is given by the following relation ([9]):



$$\frac{I_x}{Fo} = 2\frac{(1+\alpha).e^{-\delta(x-L)} - (1-\alpha).e^{\delta(x-L)}}{(1+\alpha)^2.e^{\delta} - (1-\alpha)^2.e^{-\delta}}$$

$$\alpha = \sqrt{\frac{Ea}{Ea+Es}} \quad \text{and} \quad \delta = \alpha.(Ea+Es).C_{XT}$$

L is the length of the reactor (in m)
L_{ut} is the length of the reactor which is illuminated (in m) (corresponding to a volume V_{ut}). L_{ut} is obtained by solving the equation for $\frac{I_{Lut}}{Fo} = \frac{1}{Fo}$ (i.e. the compensation point)
F_0 is the incident radiant light flux (in W/m^2)
Ea and Es are respectively the mean coefficient (in m^2.kg^{-1}) of absorption and scattering.
C_{XT} is the total biomass concentration (kg/m^3)

The calculation of the light dependant mean volumetric specific growth rate in the reactor is then:

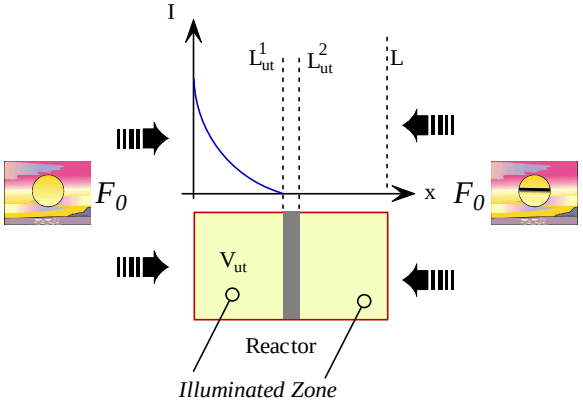
$$\left\langle \mu^{XT} \right\rangle = \mu^{XT}_{\max} \cdot \frac{1}{L} \int_0^{Lut} \frac{I_x}{I_\Sigma + K_{XT}} dx$$

with $\mu^{XT}_{\max}$ the maximum growth rate for XT (in h^{-1})

K_{XT} the saturation constant for light (in W/m^2)

4.1.2.2 Rectangular photobioreactor – dual light

The profile of the Light radiant energy I_Σ (in W/m^2) in the photobioreactor is given by the following relation ([9]):



$$\frac{I_x}{Fo} = 2\frac{\left[(1+\alpha)+(1-\alpha).e^{-\delta.L}\right]e^{-\delta.x} - \left[(1+\alpha)+(1-\alpha).e^{\delta.L}\right]e^{\delta.x}}{(1+\alpha)^2.e^{\delta} - (1-\alpha)^2.e^{-\delta}}$$

$$\alpha = \sqrt{\frac{Ea}{Ea+Es}} \quad \text{and} \quad \delta = \alpha.(Ea+Es).C_{XT}$$

L is the length of the reactor (in m)
L^1_{ut} and $L^2_{ut} = L - L^1_{ut}$ are the abscises in the reactor which is illuminated (in m). L^1_{ut} is obtained by solving the equation for $\frac{I_{Lut}}{Fo} = \frac{1}{Fo}$ (i.e. the compensation point)
F_0 is the incident radiant light flux (in W/m^2)
Ea and Es are respectively the mean coefficient (in m^2.kg^{-1})of absorption and scattering.
C_{XT} is the total biomass concentration (kg/m^3)

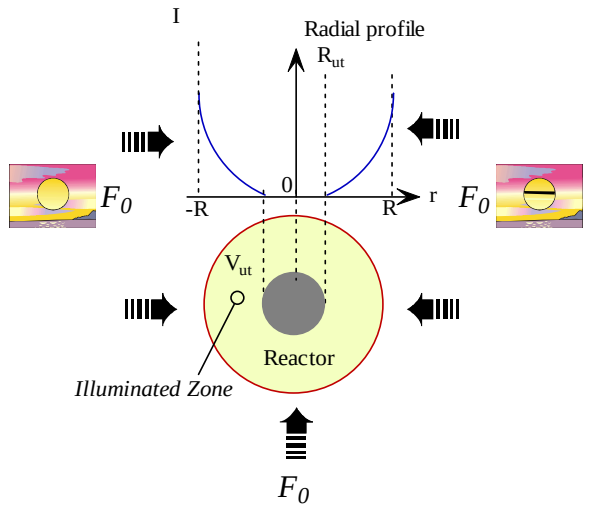
The calculation of the light dependant mean volumetric specific growth rate in the reactor is then:

$$\left\langle \mu^{XT} \right\rangle = \mu_{\max}^{XT} . \frac{1}{L} \left[\int_0^{L_{ut}^1} \frac{I_x}{I_x + K_{XT}} dx + \int_{L_{ut}^2}^{L} \frac{I_x}{I_x + K_{XT}} dx \right]$$

with $\mu_{\max}^{XT}$ the maximum growth rate for XT (in h^{-1})
K_{XT} the saturation constant for light (in W/m^2)

4.1.2.3 Cylindrical photobioreactor – radial light

The profile of the Light radiant energy I_r (in W/m^2) in the cylindrical photobioreactor with an uniform radial light is given by the following relation (updated from [8] – [11]):



$$\frac{I_r}{Fo} = 2 \frac{Bessel(0,r).\delta.r}{Bessel(0,r).\delta.R + \alpha.Bessel(1,r).\delta.R}$$

$$\alpha = \sqrt{\frac{Ea}{Ea + Es}} \quad \text{and} \quad \delta = \alpha.(Ea + Es).C_{XT}$$

L is the length of the reactor (in m)
R_{ut} is the radius of the reactor which is illuminated (in m). R_{ut} is obtained by solving the equation for $\frac{I_{Rut}}{Fo} = \frac{1}{Fo}$ (i.e. the compensation point)
F_0 is the incident radiant light flux (in W/m^2)
Ea and Es are respectively the mean coefficient (in $m^2.kg^{-1}$)of absorption and scattering.
C_{XT} is the total biomass concentration (kg/m^3)

The calculation of the light dependant mean volumetric specific growth rate in the reactor is then:

$$\left\langle \mu^{XT} \right\rangle = \mu_{\max}^{XT} . \frac{1}{\pi . R^2} \int_{Rut}^{R^1} 2.\pi.r. \frac{I_r}{I_r + K_{XT}} dr$$

with $\mu_{\max}^{XT}$ the maximum growth rate for XT (in h^{-1})
K_{XT} the saturation constant for light (in W/m^2)

4.1.3 BIOLOGICAL (STOICHIOMETRIC AND KINETIC) MODEL OF *ARTHROSPIRA PLATENSIS*

Initially, the biological model is a two stoichiometric equations (2 kinetics) model [8]. A single equation with light dependant stoichiometric coefficients was further developed for the mass balance model of the loop in which kinetic coupling of the previous 2 equation wasn't possible [4].

4.1.3.1 Two stoichiometric equations model

In the model, the biomass (Total biomass, XT) is taken as the sum of two synthesized compounds, each of them having their own stoichiometric equation and their own kinetic : the active biomass (XA=proteins+lipids+carbohydrate+nucleic acids) and the exopolysaccharide (EPS).
The mass balance expression of the two stoichiometric equations is fixed while the rates of the active biomass (XA) and of EPS are function of the volumetric specific growth rate, which was previously detailed.
Equations and parameters relatives to this biological model are reported in Table 17. It must be outlined that this biological model is this used in PHOTOSIM.

Active Biomass

$$CO_2 + 0.673\ H_2O + 0.192\ HNO_3 + 0.005\ H_2SO_4 + 0.006\ H_3PO_4 \xrightarrow{rXA} CH_{1.566}O_{0.405}N_{0.192}S_{0.005}P_{0.006} + 1.444\ O_2$$

$$\langle r_{XA} \rangle = \langle \mu^{XA} . \rangle . C_{XA} . \prod_{\text{substrate i}} \frac{C_{\text{substrate i}}}{KS_{\text{Substrate i}} + C_{\text{Substrate i}}}$$

μ_{max}^{XA} =0.073 h^{-1} K_{XA} =20 W/m^2 Ea=150 m^2. kg^{-1} Es=200m^2.kg^{-1}

Exopolysaccharide

$$CO_2 + 0.81\ H_2O + 0.015\ H_2SO_4 \xrightarrow{rEPS} CH_{1.65}O_{0.95}S_{0.015} + 0.96\ O_2$$

$$\langle r_{EPS} \rangle = \langle \mu^{EPS} . \rangle . C_{EPS} . \prod_{\text{substrate i}} \frac{C_{\text{substrate i}}}{KS_{\text{Substrate i}} + C_{\text{Substrate i}}}$$

μ_{max}^{EPS} =1.852 h^{-1} K_{XA} =750 W/m^2 Ea=150 m^2. kg^{-1} Es=200m^2.kg^{-1}

Other compounds

$\langle r_{XT} \rangle = \langle r_{XA} \rangle + \langle r_{EPS} \rangle$ Mean growth rate of the total Biomass XT

$\langle r_{\text{sCompound i}} \rangle = Y_{\text{Compound i/XA}} . \langle r_{XA} \rangle + Y_{\text{Compound i/EPS}} . \langle r_{EPS} \rangle$ Mean rate of a compound i

$Y_{\text{Compound i/XA}}$ is the yield mass Compound i / mass XA predicted by the stoichiometric equation

$Y_{\text{Compound i/EPS}}$ is the yield mass Compound i / mass EPS predicted by the stoichiometric equation

Table 17 : Kinetic of the biological model for the growth of *Arthrospira Platensis*

4.1.3.2 Single stoichiometric equation model

This one stoichiometric equation model was established in TN 17.3 and further used in the steady-state mass balance model of the complete loop [4]. The stoichiometric coefficients of the equation are computed at each time iteration as they are dependant on the radiant light flux. The equation gives directly the total biomass (XT). The kinetics associated to this equation is the same as this for the active biomass (Table 19).

	% mass (TN 17.3)	CHONSP
Protein	% mP=0,96 (-0,1067 Fo + 66,088)	$CH_{1,526}O_{0,327}N_{0,2496}$

Lipides	% mF=9,6	$CH_{1,714}O_{0,204}$
Carbohydrates	% mC=0,96 (100 - % mP - % mF- % mE)	$CH_{1,670}O_{0,711}$
DNA	% mAN=0,04 (100 - % mE)	$CH_{1,273}O_{0,701}N_{0,393}$
Exopolysaccharid (Carbohydrate equivalent)	% mE=0,110 Fo + 9,028	$CH_{1,650}O_{0,950}$
Total Biomass composition (XT)	**100%**	**$[CHONSP]_{Biomass\_Sp}$**

Table 18 : Spirulina biomass composition function of Fo , the light intensity (W/m^2). The calculation of $[CHONSP]_{Biomass\_Sp}$ is carried out by the function composition_sp.m. located in "../model/" directory.

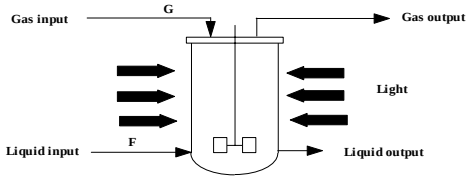
1 CO2 + ? HNO3 + ? H2SO4 + ?H3PO4 → ? $[CHONSP]_{Biomass\_Sp}$ + ? O2 +? H2O
$\langle r_{XT} \rangle = \langle \mu^{XT} . \rangle . C_{XA} . \prod_{\text{substrate i}} \frac{C_{\text{substrate i}}}{KS_{\text{Substrate i}} + C_{\text{Substrate i}}}$
$\mu_{\max}^{XT}$ =0.073 h^{-1} K_{XT} =20 W/m^2 Ea=150 m^2. kg^{-1} Es=200m^2 kg^{-1}
$\langle r_{\text{sCompounde i}} \rangle = Y_{\text{Compound i/XA}} . \langle r_{XA} \rangle + Y_{\text{Compounde i/EPS}} . \langle r_{EPS} \rangle$ Mean rate of a compound I
$Y_{\text{Compound i/XT}}$ is the yield mass Compound i / mass XT predicted by the stoichiometric equation

Table 19 : Kinetic of the 1-stoichiometry biological model for the growth of *Arthrospira Platensis*. "?" are unknown stoichiometric coefficients that can be calculated knowing $[CHONSP]_{Biomass\_Sp}$ (Table 18). The equation is solved in the S-function.

4.1.4 EXPRESSIONS OF DYNAMIC MODEL OF THE PHOTOBIOREACTOR

The two previous chapters detail only the dynamic growth model for *Arthrospira Platensis*. As presented in TN 39.1 [12], dynamic model of a bioreactor requires also to consider the hydrodynamics and the physical reactions (at least pH equilibria, gas/liquid equilibria).

Assuming a perfectly mixed reactor, **the hydrodynamic equations** are :



$$V_L . \frac{dC_{Si}\big|_L}{dt} = F.C_{Si}^{in}\big|_L - F.C_{Si}\big|_L + V_L . \phi_{Si}\big|_{GL} + V_L . \phi_{Si}\big|_{\text{Re}ac}$$

$$V_G . \frac{dC_{Si}\big|_G}{dt} = G.C_{Si}^{in}\big|_G - G.C_{Si}\big|_G - V_L . \phi_{Si}\big|_{GL}$$

$\phi_{Si}\big|_{GL}$: Transfer rate between the gas phase and the liquid phase for the compound Si in the reactor (mol.L.h^{-1})

$\phi_{Si}\big|_{REAC}$: reactionnal term relative to compound Si in reactor (mol.L.h^{-1})

$C_{Si}\big|_P$: Concentration of compound Si in the phase P (liquid=L; gas=G) of reactor (mol.L^{-1}).

F: Liquid flow rate ($L.h^{-1}$) and G: Gas flow rate ($NL.h^{-1}$).

The flow rate can be different in input and in output, leading to a variation of the volumes V_L and V_G , and of gas pressure. But it is assumed in the current model that gas and liquid flow rates are constants. This assumption will have a non negligible effect on the gas fraction as it then assumed that the total number of moles in the gas output is equal to the one in the gas input. This approach was chosen in the objective to link directly C4a and crew. With a gas flow rate which may varies the coupling of two compartment will be more complex.

Biological reactions and hydrodynamic equations

The term representing the exchanges between the biological phase and the liquid phase in the biological reactor is noted $\phi_{Si}^{n}\big|_{REAC}$. Assuming no transfer limitation between the two phases leads to:

$$\phi_{Si}^{n}\big|_{REAC} = \langle r_{Si} \rangle \text{ or } \langle r_X \rangle$$

Ionic forms and pH equilibrium

pH and ionic forms of a compound are an important element, as well in mass balance model as in the dynamic model.

In the previous mass balance model [4], the pH equilibrium was taken into account only for compounds existing both in liquid and in gas phase by modifying the partition coefficients of the compounds concerned [13]. This approach reduces the number of compounds to take into account as only the non ionic forms are considered. This approach implicitly assumes also that the acid/base reaction is "immediate" so that if a form is consumed there is no limitation due to the acid/base reaction.

A more realistic approach consists to directly consider in the different reactions involved in a process, both the ionic and the non ionic forms. This requires to deal with equilibria constants and to consider a rate for the acid/base reactions (or to solve at the same time linear and differential system, what is more complex on a numerical point of view).

We choose the first approach as it will result in a simpler model.

Gas-liquid mass transfer

The gas liquid equilibrium is an important aspect to take into account in dynamic modeling of bioreactors. The dynamic of the transfer from the gaseous compounds to the liquid phase can be expressed by:

$\phi_{Si}^{n}\big|_{GL} = K_L a\big|_{Si} \cdot (C_{Si}^{*}\big|_{L} - C_{Si}\big|_{L})$ where

$C_{Si}^{*}\big|_{P}$ is the saturation concentration of the substrate (i) in the liquid phase.

$K_L a\big|_{Si}$ is the volumetric transfer rate of the compound Si between the gas and the liquid phase

Note that this expression is different when considering "evaporation" (i.e. a liquid compound in equilibrium with the gas phase) : $\phi_{Si}^{n}\Big|_{GL} = -K_L a\big|_{Si} \cdot (C_{Si}^{*}\Big|_{G} - C_{Si}\Big|_{G})$

The saturation concentration is calculated from the partition coefficient $k_{Si} = \frac{y_{Si}}{x_{Si}}$. For most of the compounds in the MELiSSA loop, the partition coefficients have been already determined in TN 23.1. These coefficients are characteristic of the thermodynamic equilibrium of each compound. As demonstrated in TN 17.3 , this partition coefficient can be defined as a function of the temperature, the pressure and the pH. For the Matlab/Simulink model, the partition coefficients of the compounds are fixed in the script **"../models/load_partition.m" (**Table 20**).** It is considered that :

k_{Si}=0 means the compound is only in the liquid phase
k_{Si}>1e10 means the compound is only in the gas phase
k_{Si}<1 means the compound is mainly in the liquid phase (such as water)
k_{Si} >1 means the compound is mainly in the gas phase (such as O_2)

```
ki(i_eau)=exp((18.3036-3816.44/(-46.13+Temp+273.15)))/(760*Press); Antoine law
ki(i_o2)=4200; % O2
ki(i_co2)=2000; % CO2
ki(i_inert)=1e10; %
ki(i_h2)=1e10; %
ki(i_n2)=1e10; %
ki(i_ch4)=1e10;
```

Table 20 : Partition coefficients used in the current model.

On a numerical point of view, the gas-liquid transfer dynamic can be a limiting step in the numerical treatment of the biological models. The simplification of models, based on Laplace transformed equations, developed by J.J. Leclercq (TN 35.2) could be a useful way to skip such problems.

4.2 C4a simulink model description (C4a_subsystem.mdl)

The Matlab/Simulink model of the C4a MELiSSA susbsystem is saved in file **C4a_subsystem.mdl** (Figure 10). The core of the subsystem is the dynamic model of the photobioreactor (**C4a_dyn_004.m**).

4.2.1 MATLAB/SIMULINK C4A SUBSYSTEM OVERVIEW

The standalone model of the subsystem is presented in the following Figure 10. Input and outputs of the photobioreactor were designed here in order to be compatible with the crew and control subsystems.

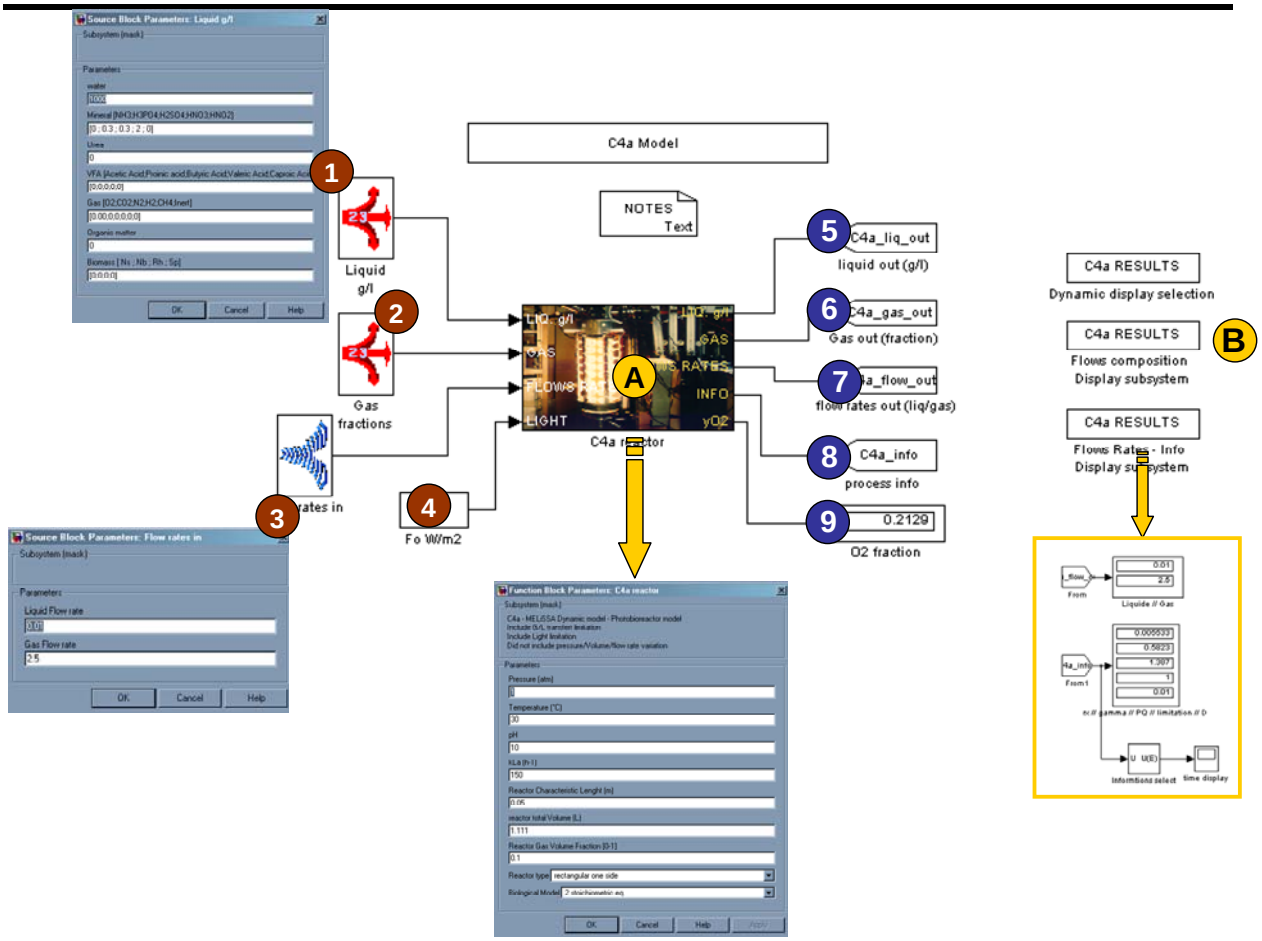


Figure 10 : C4a subsystem under matlab/Simulink – A] is the block associated to the photobioreactor model a dialog box allows manipulation of the parameters. B] are the subsystems for displaying the results of the simulation. 1] is a dialog box for the set-up of the input liquid flow (concentrations in g/l). 2] is is a dialog box for the set-up of the input gas flow (in molar fraction). 3] is a dialog box for the set up of the flows rates. 4] is the set-up of the light radiant energy in W/m^2. 5] is the output liquid compostion in g/l. 6] is the output gas composition in molar fraction. 7] are the output flow rates. 8] are a set of relevant information from the simulation: the biomass productivity; the gamma value ; the substrate limitation factor and the dilution rate. 9] is the output O_2 gas fraction.

4.2.2 C4A MODEL (C4A_DYN_004.M)

C4a_dyn_004.m located in '**../models/**' directory is the file containing the S-function for the C4a photobioreactor dynamic model. It is associated to the C4a reactor S-block of Figure 10.

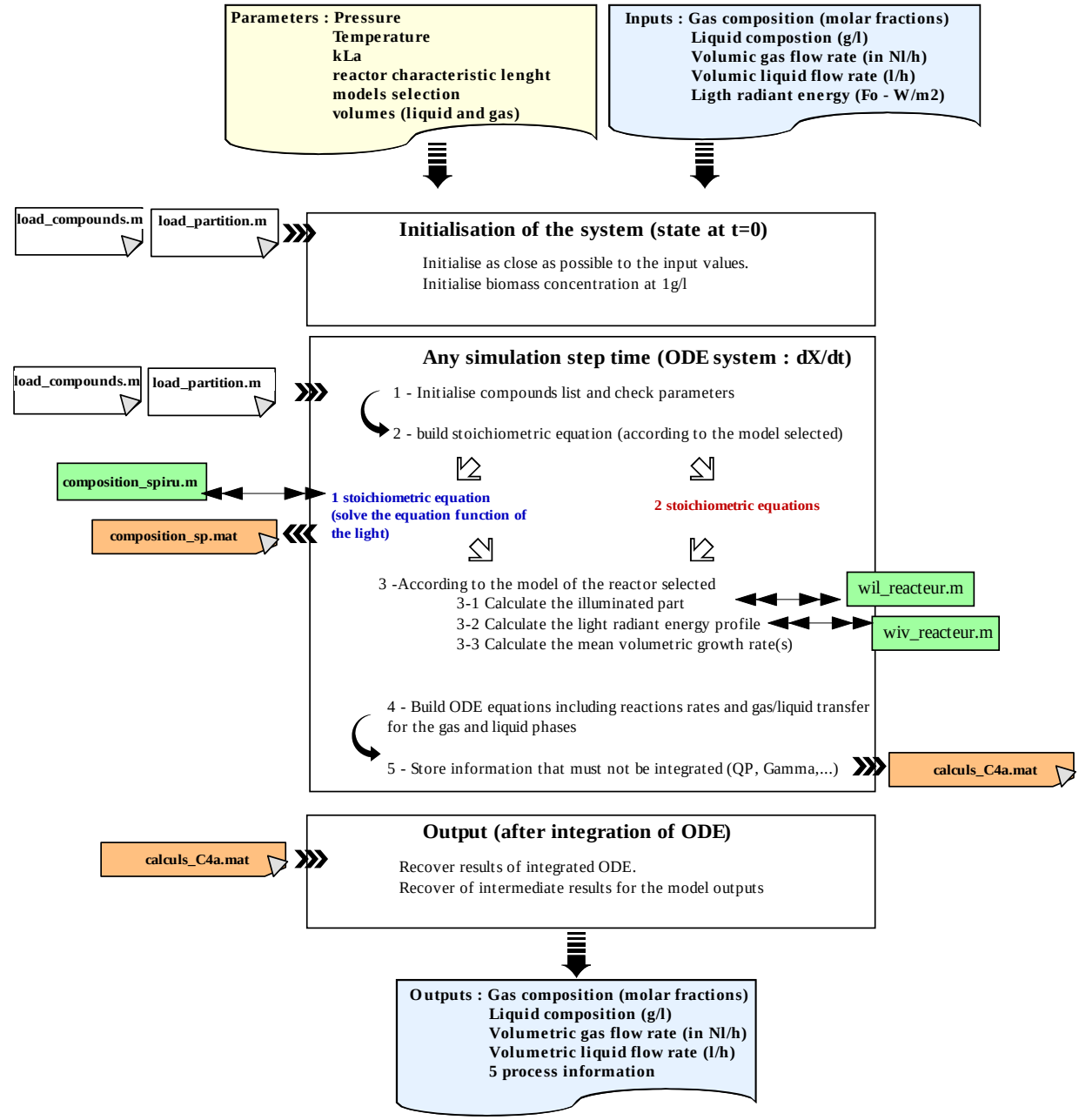


Figure 11 : Overview of the algoritm of the S-function model for C4a photbioreactor (C4a_dyn_004.m). Green squares are function called by the S-function. Gold squares are storage files in matlab format.

Aims: The function builds the differential equations system (report to 4.1.4) modelling the photobioreactor for the *Arthospira Platensis* growth. This system is established from the different options selected (reactor type and biological model). It takes into account the gas/liquid equilibrium, but the acid/base equilibrium is only considered by adjusting the partition coefficient for the gas/liquid thermodynamic equilibrium as in the previous steady-state mass balance models (report to 4.1.4). The S-function contains only a differential

equations system (continuous dynamic system) which is solved to give at each iteration time the state (compounds concentrations and fractions) of the reactor.

Algorithm: The algorithm of the function is detailed in Figure 11. The Input flows are first translated in moles for both gas and liquid for homogeneity. The biological and light radiant profiles (i.e. mean volumetric growth rate) are computed depending on the options selected. From the biological model (stoichiometric yields) and the mean volumetric growth rate are computed all the reaction rate to put in the hydrodynamic equations. The system obtained is solved by Matlab/Simulink to compute the reactor outputs. Note that some data are loaded from files, that the model calls also several subroutines for repetitive calculations and that some intermediate results of the dynamic calculation are stored in data files for further use.

Parameters Most of the parameters are managed through the dialog box under Matlab/Simulink (Figure 10). They are :

- ➔ Pressure (in atm) – default : 1atm
- ➔ Temperature (in °C) – default : 30°C
- ➔ k_La (volumetric transfer coefficient in h^{-1}) – default $50h^{-1}$
- ➔ Characteristic length of the reactor (length or diameter of reactor in m) – default 1 m
- ➔ Total volume of the reactor (in L) -- default 1L
- ➔ Volume fraction of gas in the reactor – default 0.1
- ➔ Choice between the 2 biological models
- ➔ Choice between the 3 photobioreactors

Some are manipulated directly in source codes :

- ➔ the saturation constant for substrates limitation ($KS=1e^{-6}$ mol/l by default) is fixed in the source code of the model (**C4a_dyn_004.m**)
- ➔ the partition coefficient for the gas/liquid equilibrium are set-up in the script **load_partition.m**.
- ➔ the list, composition and molar mass of the compounds are set-up in the script **load_compounds.m**.

Inputs/outputs : Inputs and outputs of the subsystem are both mass fluxes and information.
The gas composition flow, in molar fraction (Figure 10 – 1,2]) and the liquid composition flow (in g/l) are vectors of size equal to the number of compounds involved sorted as fixed in the configuration file ***"../models/load_compounds.m***". In the current version this is a list of the 23 compounds previously used in the steady-state model of the loop [3]. You can notice that for this purpose specific S-block with a dialog box has been developed.
The flow rates were given as a single input (vector of 2 values). They are given in L/h for the liquid and in Nl/h for the gas (i.e. flow measurement assuming T=0°C and P=1 atm). At this time the flow rates at the output are assumed equal to the input.
The light flux (in W/m^2) applied on the reactor is taken as an input, because it is a manipulated variable in the subsystem.

The oxygen molar gas fraction is an output of the subsystem, which may be used as a controlled variable.

Some information are recovered from the calculation (Figure 10 – 8]) and can be checked in the results S-block (Figure 10 – B]). They are:

- ➔ The total biomass productivity, r_X, in $g.l^{-1}.h^{-1}$.
- ➔ The ratio between illuminated and total length for the reactor $\gamma = \frac{L_{ut}}{L}$ or $\gamma = \frac{R_{ut}}{R}$
- ➔ The photosynthetic Quotient $PQ = \frac{r_{O2}}{r_{CO2}}$
- ➔ The substrate limitation factor (between 0 and 1 , 1 meaning "no limitation")
- ➔ The liquid dilution rate.

It can be noticed that the information recovered can be changed by modifying the outputs in the model source code ***C4a_dyn_004.m.***

Files associated: Several files are associated either as results storage files or as required functions called by the model. They are listed and briefly explained in Table 21.

file	details
../models/composition_spiru.m	This function calculate the composition of the total biomass of *Spirulina* for a given light flux, according to.Table 18
../data/composition_sp.mat	This is the storage file for the composition of *Spirulina*. The results are saved under a cell and structure data type
../models/wil_reacteur.m	This function calculate the length (or radius) of the reactor at which the radiant light energy reaches the compensation point (fixed at $1W/m^2$). It returns the L_{ut} or R_{ut} value (report to 4.1.2)
../models/wiv_reacteur.m	This function is the expression that must be integrated to calculate the mean volumetric specific growth rate (report to 4.1.2)
./models/load_compounds.m	This script is included in the S-function. It provides the sorted list, composition and molar mass of the 23 compounds taken in to account in the system.
./models/load_compounds.m	This script is included in the S-function. It provides the values of the partition coefficients for the gas/liquid equilibrium.
../data/calculs_C4a.mat	This is the storage file for some intermediate calculus done for the ODE system that could be useful for checking the process simulation.

Table 21 : Files involed in the the S-function *C4a_dyn_004.m* for the C4a model.

4.3 Tests and results

Several functional tests were done to check the C4a standalone subsystem. Excepting in specific situations where gas/liquid transfer and/or substrate limitation occur there is no numerical problems for solving the model under Matlab/Simulink. The solver used is ODE15s with the parameter reported in Figure 12. Error tolerance can be changed in cases of numerical problems at the price of computing time.

Remark: The solver reset method should be set to "robust" to avoid numerical failure

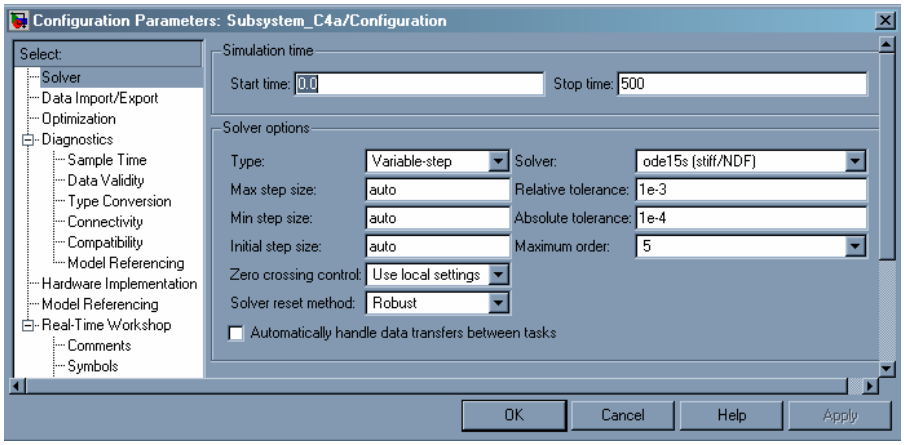


Figure 12 : Screen capture of the parameters for the numerical solver under Matlab/Simulink.

4.3.1 MATLAB/SIMULINK VS. PHOTOSIM

In order to check the validity of the model we have compared the steady state values obtained after 500h of simulation with the results obtained with PHOTOSIM for a single side illuminated reactangular reactor of 0.05m length (Table 22).
Under Matlab/Smulink the reactor was operated in non limiting substrate and gas/liquid transfert conditions. Its parameters were:

- total volume = 1.11 l (i.e. liquid volume = 1 l)
- gas volume fraction = 0.1
- pH=10
- Pressure = 1 atm
- temperature = 30°C.

The comparison (Figure 13) shows that the results obtained with the 2 stoichiometric equations for the biological model are identical to the ones obtained with PHOTOSIM. This is an attempted result as in such a scenario the two models are the same. With the 1 variable stoichiometric equation, differences are observed. It may be interesting to wonder if the correlation established for the variation of the biomass composition with the light may be improved.

Test	D (h^{-1})	Fo (W/m^2)	Cx (g/l)	r_X $g.l^{-1}.h^{-1}$	PQ	Ro_2 ($mol.l^{-1}$ h^{-1})	γ
test 1	0,02	20	0,255	0,0051	1,39	2,98E-04	1
test 2	0,01	20	0,55	0,0055	1,39	3,22E-04	0,587
test 3	0,005	20	1,1	0,0055	1,39	3,22E-04	0,29
test 4	0,04	100	0,36	0,0144	1,344	7,90E-04	1
test 5	0,01	100	1,599	0,016	1,344	8,86E-04	0,346
test 6	0,05	300	0,526	0,0263	1,289	1,35E-03	1
test 7	0,01	300	2,88	0,028875	1,289	1,49E-03	0,278

Table 22 : Configuration of the simulations test with PHOTOSIM for a a single side illuminated rectangular reactor of 0.05m length, and results obtained for the steady-state

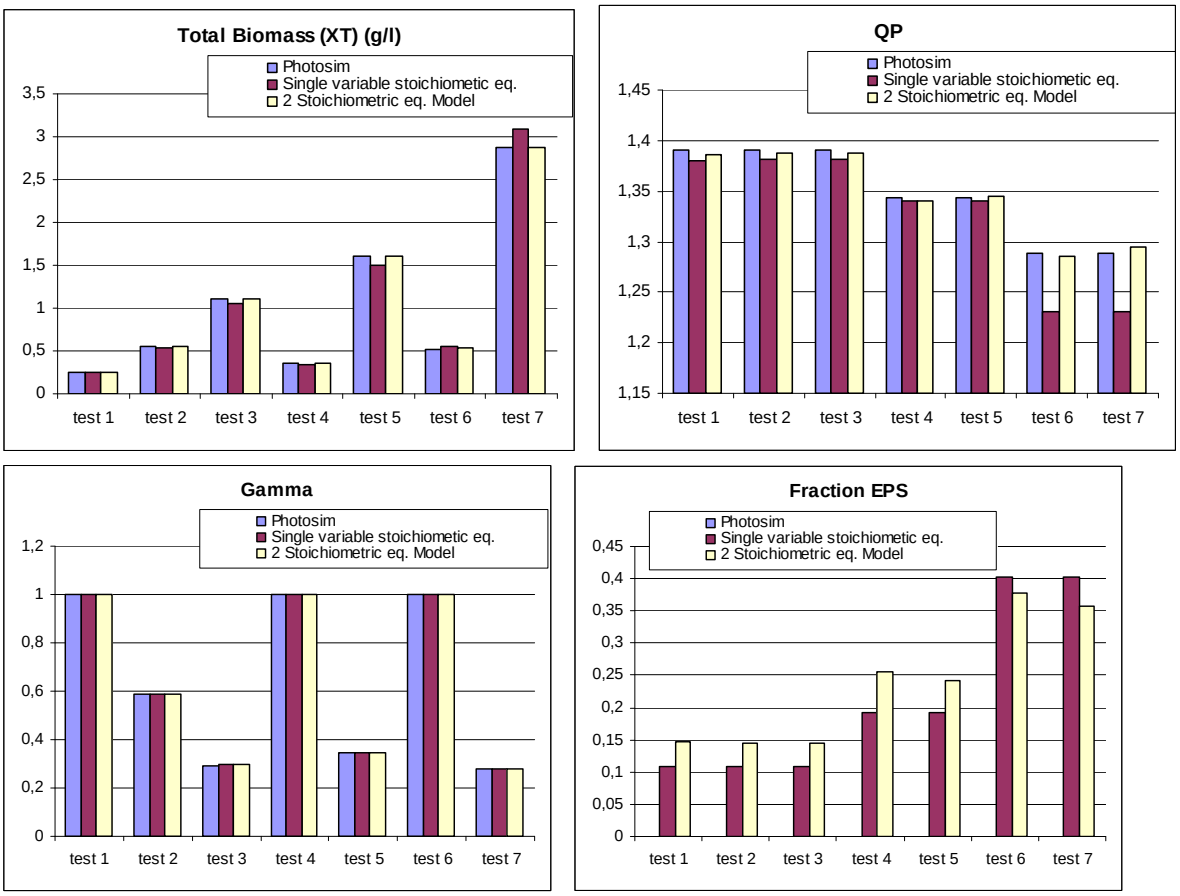


Figure 13 : Comparison model for various steady-state conditions between PHOTOSIM and Matlab/Simulink with one single variable stoichiometric equation or two fixed stoichiometric equations for the biological model.

4.3.2 IMPORTANT REMARKS

It may be useful to list here some of the remarks done on the C4a Matlab/Simulink model. These remarks can help to analyze possible numerical problems when using the model and to investigate solutions for improving the model.

The flow rates. It is assumed that they are unchanged between input and output. This assumption can be accepted for the liquid phase as flow variation is mainly linked to the water evaporation, which can be reduced by use of condenser (i.e. equivalent in the model to the use of the condenser temperature for the calculation of the partition coefficient instead of the

reactor temperature). This is less true for the gas as volume is more sensitive to the moles variation. Then this assumption introduces errors in the mass balance which may be increased in the closed system. It will be necessary to improve this point in future models, but this will require to manage the gas flow rate by external supply or sink or pressurized buffer tank.

The volume and gas pressure. It is obvious that volumes and gas pressure in the reactor are linked to the flow rates. Assuming constant volume and pressure (as constant flow rates) may leads to accumulation of error for high closure systems.

The pH. It must be noticed that usually the pH of a reactor is maintained by the addition of an acid or of a base. The volume quantities added of base and acid used to maintain the pH are rarely included in dynamic models as the volumes added are generally low. But the influence in closed loop may become non-negligible. Moreover it may lead to mineral accumulation. It must be outlined that the prediction of pH is not a simple task.

The CO_2 source. The CO_2 for the *Spirulina* growth can be introduced under two forms: in the gas (CO_2) or in the liquid (Bicarbonate). In principle, in the MELiSSA objective, CO_2 should be introduced in the reactor by the gas, what implies that the gas/liquid transfer and equilibria should play an important role. Moreover the CO_2 forms are dependant on the CO_2 partial pressure (i.e. linked to the total pressure) and to the pH.

On a numerical point of view the most critical point concerns the substrate limitations (which may be complicated by a gas/liquid transfer limitation in the case of CO_2). By default the saturation constant is fixed to 10^{-6} mol/l. When a substrate concentration reaches this value a numerical bottle neck can appear (high reduction of integration step, computing error due to inaccurate tolerances). Such problems have been already highlighted [12].

5 CONTROL SUBSYTEM

A closed loop requires a control system in order to maintain the system in nominal conditions. Considering a closed gas loop between the crew and the C4a subsystems, the objectives of the control system are:

- ➔ To maintain the oxygen level to 21% (molar fraction)
- ➔ To maintain the CO_2 below 3000 ppm (0.3%)
- ➔ To maintain a constant gas flow rate , and pressure in the subsystems

For each of the controlled variable it must exist a manipulated variable which have an effect on the controlled variable.
With the assumptions in the model of constant pressure and flows rates, the control of these variables is not considered at this time.
It is not possible (or only in a small range) to control at the same time the CO_2 and the oxygen only with the C4a compartment. This will suppose to be **able to control accurately the PQ** of *Spirulina* (Figure 13).

We limit then here the control to the oxygen fraction. The controlled oxygen fraction can be :

- ➔ The fraction in the crew cabin (i.e. output of the cabin gas flow)
- ➔ The fraction at the input of the cabin (i.e. output of the C4a compartment).

5.1 Simple control system : minimum/maximum

Our objective is not to develop a control algorithm, then we have only implemented a simple control law which compare for a periodic step the controlled variable with its consign and which set-up the manipulated variable to a minimum value when control>consign and to a maximum value when control<consign.

A S-block for Matlab/Simulink was built (Figure 14), based upon the S-function (Annex 7) coding for the control law (**../data/control_v2.m)**. Parameters of the control system are set-up through the dialog box. Input and output must be a single value (not a vector).

5.2 Test of the control with a variable O_2 input on C4a

The control subsystem was tested by controlling the light radiant energy flux on the photobioreactor (C4a subsystem) in order to maintain a fraction of 21% of O_2 in the reactor, considering a variable O_2 fraction in the gas input. The coupling is presented in Figure 15, which is a screen capture of the "**subsystem_C4a_control/mdl**" Matlab/Simulink model

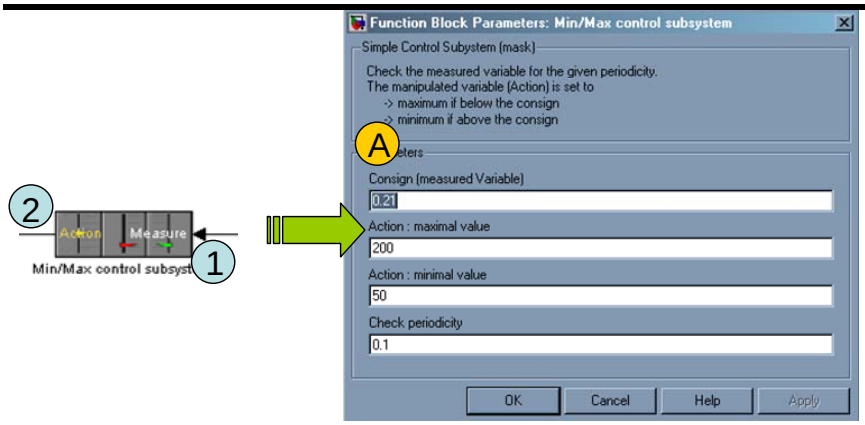


Figure 14 : S-block for simple maximum/minimum action for control of a variable. 1] is the value of the controlled variable (i.e. O2 fraction) ; 2] is the value of the manipulated variable (i.e. the light radiant enerfy) ; A] is the dialog box of the S-block to set-up the maximum and minimum of the action, the consign and the periodic step for comparison to the consign.

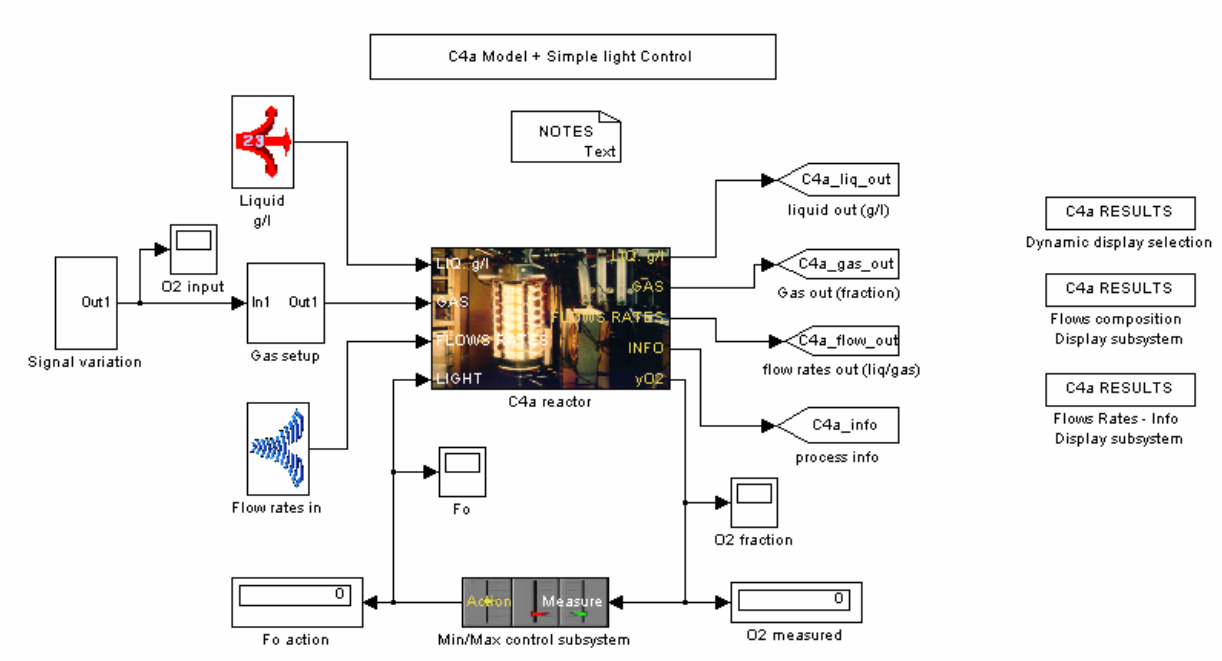
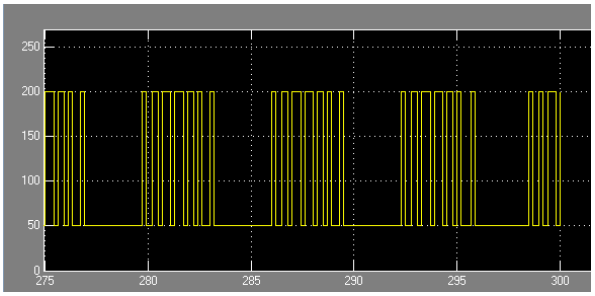


Figure 15: Model of a C4a susbsytem with a control loop for the light. The controlled variable is the O2 fraction in the reactor.

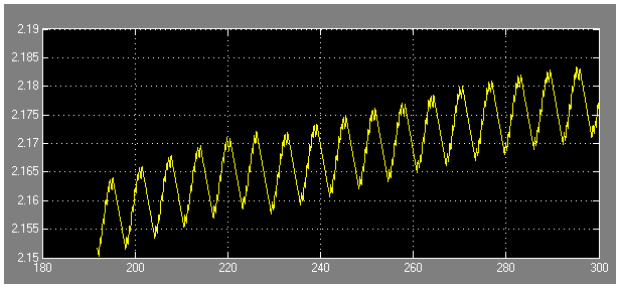
In Figure 16 are presented the results for the following operating conditions of the system:

- ➔ Continuous gas and liquid flows. Gas flow rate=3NL/h and liquid flow rate=0.02 L/h
- ➔ In liquid concentrations are 1kg/l of water , 1g/l H_3PO_4, 1 g/l H_2SO_4 , 5g/l HNO_3. For the growth substrates, the values were chosen in order to be non limiting.
- ➔ In gas the molar fractions are 0.79 for N_2 and 0.02 for CO_2 (i.e. 2%).

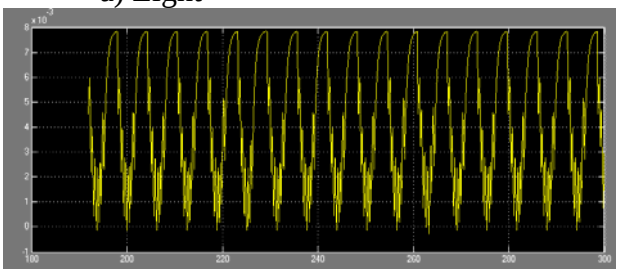
- ➔ The oxygen gas fraction is a sinusoidal function oscillating between 0.18 and 0.2 with a frequency of 6.3h.
- ➔ The initial concentration of the reactor was 2g/l of *Spirulina* (near steady-state concentration). The reactor is a dual light reactor (length of 0.05 m) modeled with the 2 stoichiometric equations for the biological model. The reactor has a total volume of 2.22 l and a gas fraction of 10%. k_La is set-up to $150h^{-1}$, temperature to 30°C and pressure to 1atm.
- ➔ The control law use a period of 0.1 hour, and the maximum and minimum for the light are respectively 50 W/m^2 and 200 W/m^2



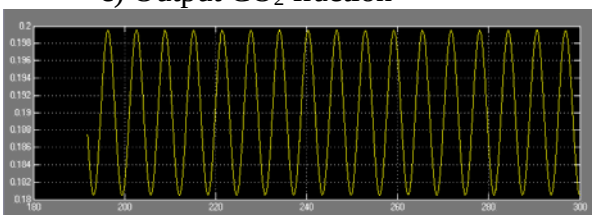
a) Light



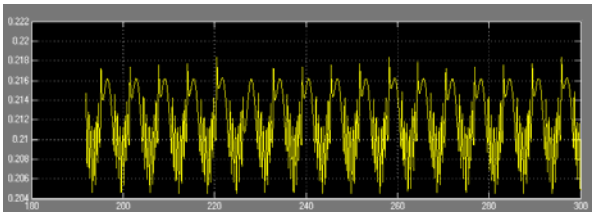
b) biomass concentration (g/l)



c) Output CO_2 fraction



d) Oxygen input fraction



e) Oxygen output fraction

Figure 16 : Results of control of the C4a subsystem

If it is obvious that the control law is too simple to obtain an accurate control of the oxygen fraction to 0.21, the results indicate that the response with the change of the light intensity is

very rapid and that even if the simulating can take a while, the system can be operated in that control mode.

Remarks : To be used, the system must be consistent. This means that the control law as well as the photobioreator configuration allow to reach the consign fixed. If, even at lowest light, the photobioreactor leads to produce more oxygen than needed or if a substrate limitation occurs, the control will fail. This raw control law has only one objective: to roughly maintain the system around the consign. It is not a reliable control law for deeper investigations of controlled loop.

6 OVERALL SYSTEM CREW+C4A-GAS COUPLED LOOP

We have built and tested the 3 subsystems required for a gas coupled Crew-C4a controlled system.
A prerequisite of the coupling is to set-up a consistent system. Table 22 can give an idea of the range of the oxygen production rate by the photobioreactor. These values must be compared to the oxygen consumption rate for a man ranging between 25-60 g/h and an average of about 40 g/h (depending on its physical activity).

6.1 Overview of the system under Matlab/Simulink

The coupling of the crew with the C4a was done in three steps:

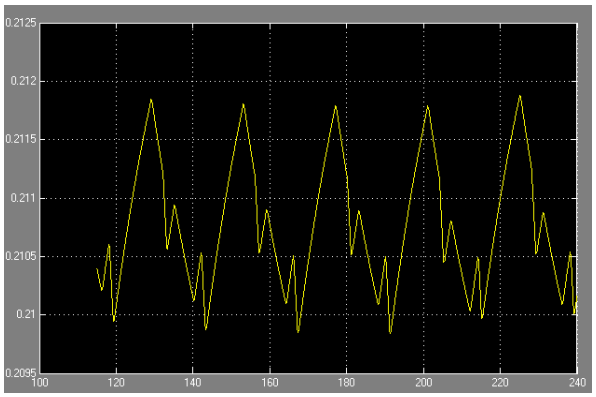
1. Gas output from the crew cabin, which is vented with fresh air, is directly used as a gas input to the C4a photobioreactor operated with a constant maximal light radiant energy flux. This step allows to check if the scenario allows the gas output of the C4a reactor to fit the requirement to become the input of the crew compartment, i.e. to satisfy the input constraints. In this step, the system is not closed and not controlled.
2. This linked system, always open for the gas loop is completed by the control system for the C4a compartment, with the objective to have a gas output of 21% oxygen. This step is for checking the ability to oscillate around the consign (Figure 16).
3. The third step is the coupling of the system in a closed gas loop.

We will not detail all these steps here, and only the third step will be presented in the section 6.2.

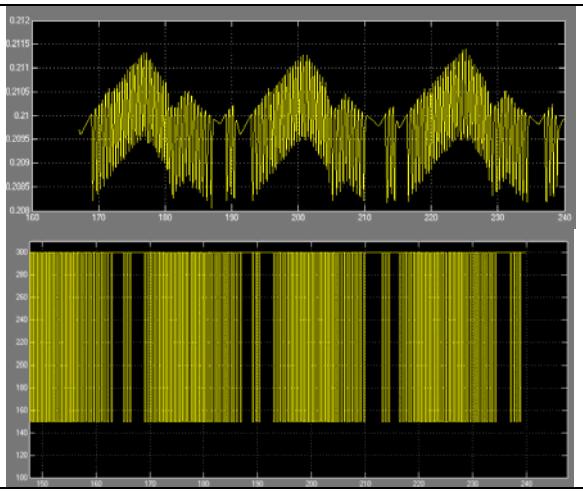
The crew is composed of one single member, which activities and characteristics are those reported in Table 16. The cabin volume was set up to 20 m^3 and the gas flow rate through the system was set-up to 3000 NL/h and the liquid flow rate to 10L/h..
The photobioreactor is a rectangular 2-side light of 0.05 m length and of a total volume of 555.55 L (500 L liquid) i.e. equivalent to a square of 3.3m length. The maximum light radian energy is set to 300 $W.m^{-2}$.

The first step was performed successfully. With this configuration, it was possible to be above 21% of oxygen fraction at the output of the C4a. This test has required to introduce a "switch block" between the crew and the C4a subsystem in order to link the two compartments only after 48 hours of operation. This "switch" avoids linking the crew with C4a compartment during the transient state of the crew compartment (Figure 8).

In the second step, the control subsystem was used to manage the light flux on the C4a compartment in order to maintain the oxygen gas fraction at the output of the reactor to 21%. The system built is quite the same as the one presented in Figure 15. Note that here we choose to control the output to 21% oxygen because it is the value of the fresh air used for venting the crew. A comparison of controlled/uncontrolled for the open system is presented in Figure 17.



Step 1 : oxygen fraction at C4a output with uncontrolled C4a reactor – light flux = 300W.m^{-2}



Step 2 : oxygen fraction at C4a output with controlled light flux (min=150 W.m^{-2} – max = 300 W.m^{-2})

Figure 17 : Comparison of the system for the step 1 and step 2 operating conditions

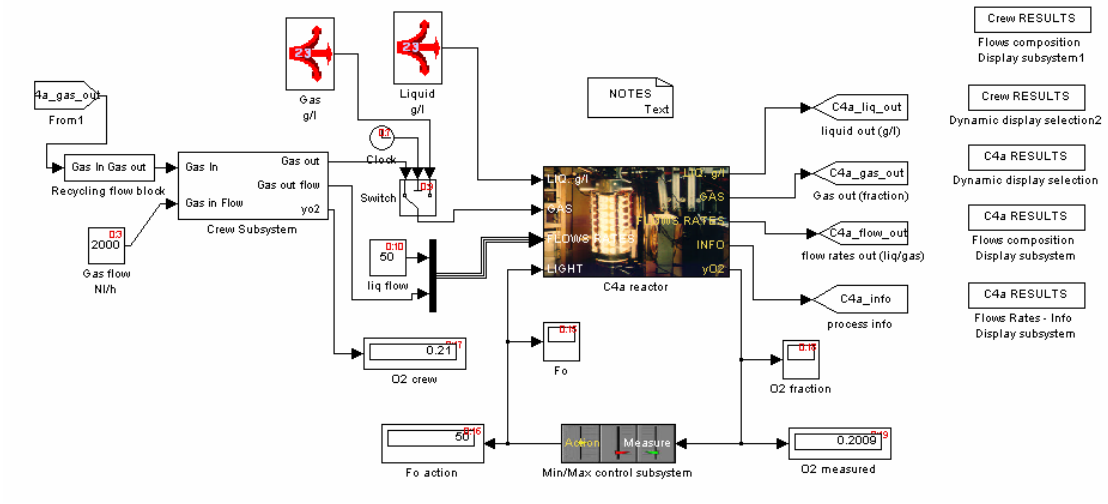
6.2 Operational test

The final test consists in the linking of the gas output of C4a compartment with the gas input of the Crew, without any kind of sink or supply. As can be seen in Figure 18, a new block called "recycling flow block" was added. This block is composed of :

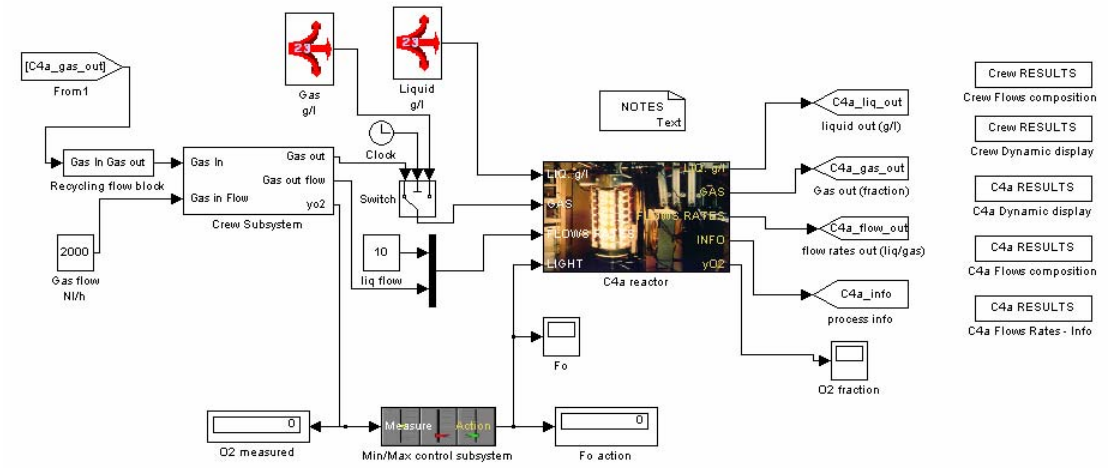
- ➔ An initializing block to set up the recycling flow at start-up of the simulation. The initial fractions can be given through the interface which opens when selecting the object.
- ➔ A delay block. This object is required to by-pass the numerical bottleneck due to the algebraic loop created by directly recycling the C4a output to the input of the crew.

The simulation can take a while, but the computing time remains acceptable. The system can be built with two designs. In a first one, the control is applied at the level of the C4a output (Figure 18 – a]), while in the second, the control is applied at the level of the Crew output (i.e. to control atmosphere), what seems to be a more realistic approach of the system (Figure 18 – b]). The two models are respectively **MELiSSA_5_4a_dyn_005.mdl** and **MELiSSA_5_4a_dyn_006.mdl**.

The results are presented in Figure 19.



a]



b]

Figure 18 : Two configurations for the control system: a] Control of the gas fraction at the C4a output MELiSSA_5_4a_dyn_005.mdl . b] Control of the crew atmosphere MELiSSA_5_4a_dyn_006.mdl.

In the two cases it can be seen the accumulation of CO_2 in the gas (up to 5% after 200 h). It is a result that was theoretically attempted as the PQ of *Spirulina* (1.3) and the RQ of the crew members (0.8) cannot compensate each other. On the other hand it is possible to maintain the oxygen in the desired range.

The control of the crew atmosphere to 21% of oxygen with the control subsystem use is more difficult mainly because of the mean residence time in the crew cabin (t=7.5h) compared to the one of the photobioreactor (0.018h) and because photobioreactor for this test is a little under-sized: the light is most of the time at its maximum value and the average fraction of O_2 in the cabin is about 20.8 % instead of 21% , i.e. the control system fails.

This last remark leads to suggest that a table giving the oxygen productivity by liter of reactor for given light, dilution rate and reactor design may be useful to avoid inaccurate simulation due to badly scaled design.

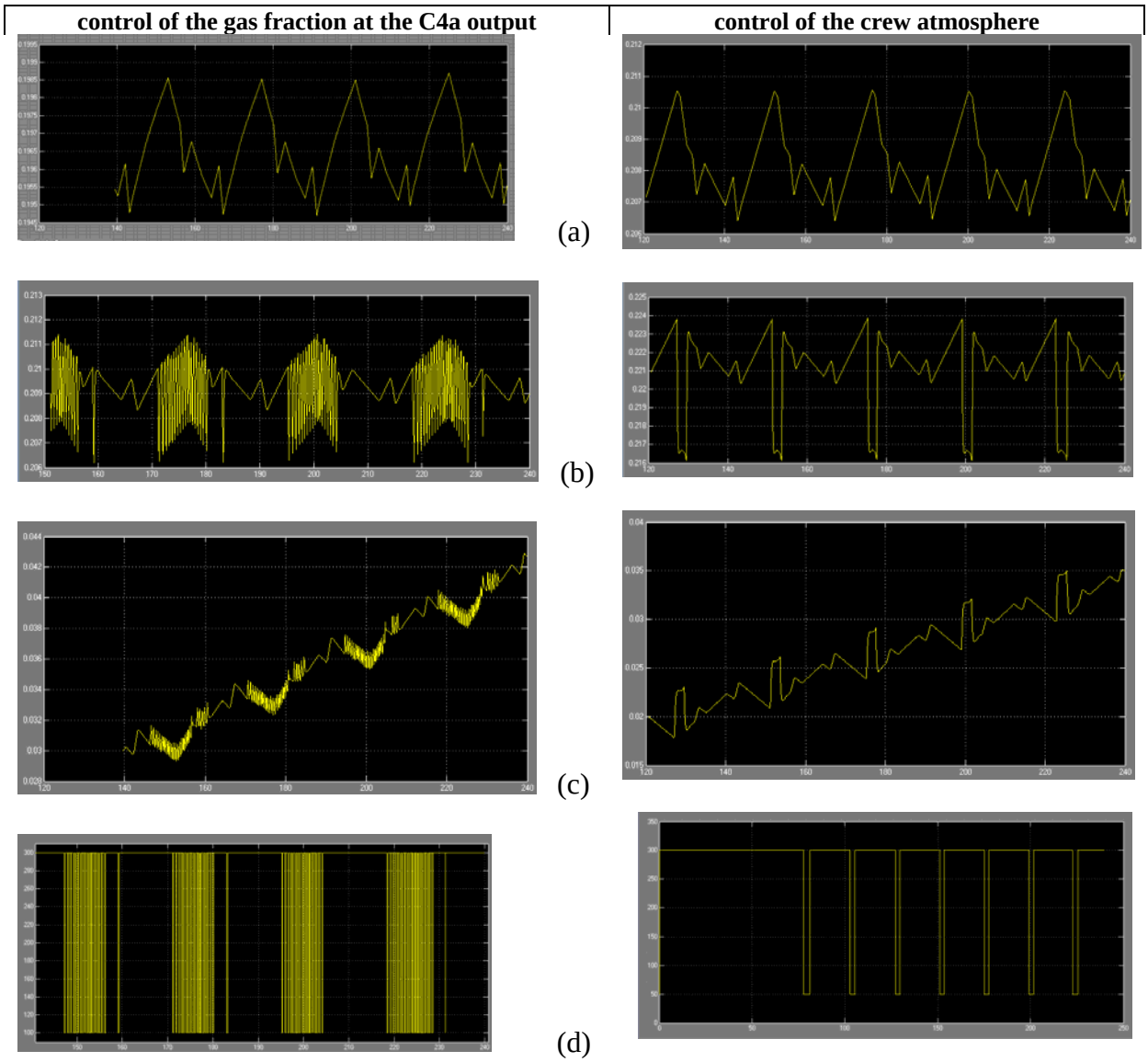


Figure 19 : Results of the two configurations of a gas closed loop crew-C4a. (a) – oxygen fraction at crew output. (b) – oxygen fraction at C4a output. (c) – CO2 fraction at C4a output. (d) light flux on C4a.

7 CONCLUSIONS

A dynamic model for the crew respiration was established, based on the members activities and their Energy Expenditure Rate.
A dynamic model for the C4a compartment (Spirulina photobioreactor) was developed from the previous PHOTOSIM model.
The two subsystems were linked for the gas phase introducing a simple control to maintain the oxygen gas fraction either in the crew cabin or in the bioreactor.

The coupled system can be simulated with Matlab/Simulink® using an approach comparable to the one used to model and simulate the complete loop. It seems then feasible on a technical point of view to use the software for the simulation of a dynamic MELiSSA loop. The computing time may be more or less important in the gas coupled system, mainly because of the discrete functioning of the control law.

The simulations seem consistent with previous simulations obtained with PHOTOSIM and with the attempted results for the coupled system. The coupled system is well–controlled, at the condition of a scaling of the system allowing the control, even if the control is a very simple law. This is not a predictive control law which should be more accurate. It is then possible to maintain the O_2 fraction constant, but the difference in the PQ and the QR of the two subsystems leads to CO_2 accumulation.
The model built is not for the sizing or scale-up. It is designed here only to simulate the behaviour of a defined system, assuming it is well-designed.

Some improvements are required.
For the C4a dynamic model, the gas flow rate, pressure and/or volume variation should be taken into account to have a more realistic view of the system. Instead of separate flow composition and flow rates, what is a more manageable view of the system, it will be better to directly work with partial flow rate for each compound, even if it is more difficult to manage. The pH variation is also an important point, especially if we expect to estimate the acid/base volume to add for maintaining the pH of the reactor. The partition coefficients model should also be revised to introduce the correlations established in TN 17.3.
The crew model should be completed with a "metabolic model" to complete the mass balance of the compartment. This should be done before addition of compartment 1.

This document should be released with the source code of the Matlab/Simulink models

8 REFERENCES

[1] Condran M.J. et al. 1989. "Amodel of human metabolic mass flow rates for an engineered closed ecosystem". SAE PAPER 891486

[2] Wade H. W. and Schoeller D.A. "Nutrition in Spaceflight and weightlessness Models" CRC Press. 1999

[3] Life support and habitability manual – Volume 1. ESA PSS-03-406 Issue 1 – August 1991.

[4] TN 79.2 : L. Poughon. "MELiSSA Loop Mass Balance Modelling with Matlab® / Simulink." Memorandum of Understanding TOS-MCT/2002/3161/In/CL . 2005

[5] Heinerwandel K. et Kring G. 1988. ECLSS for pressurized modules - From Spacelab to Columbus Space Thermal and life Support Systems. ESA SP-288. Pp. 45-50.

[6] Waleh Ahmad, Kanevsky V., Nguyen T.K. Upadhye R. and Wydeven T. 'Impact of diet o th design of waste processors in celss' ESA-SP 324 Vol 2. - 4th European Symposiium on Space Environmental Control Systems Florence-Italy . 1991

[7] Calloway D.H. "Basic Data for planning LSS" Chapter 1 – Incomplete reference source

[8] TN 19.2 CORNET J.F., DUSSAP C.G., GROS J.B. "Modeling of physical limitations in photobioreactors. Application to simulation and control of the Spirulina compartment of the MELISSA artificial ecosystem", July 1993, MELISSA Contract PRF13O820. 1993

[9] Cornet J.F. Dussap C.G. Gros J;B. Bionois C. Lasseur C. 1995. A simplified monodimensionnal approach for modeling coupling between radiant light transfert and growth kinetics in photobioreactors. Chem. Eng. Sci. 50(9) pp 1489-1500

[10] Cornet J.F., Dussap C.G. Gros J.B. 1998. Kinetics and energetics of photosynthetic microorganisms in photobioreactors. Adv. Biochem. Eng./Biotech. 59. pp 155-224.

[11] Cornet J.F. Procédés limités par le transfert de rayonnement en milieu hétérogéne – Etude des couplages cinétiques et énergétiques dans les photobioréacteurs par une approche thermodynamique. Habilitation à diriger les recherches, Université Blaise Pascal, Clermont Ferrand, France – To be appeared in 2006-2007.

[12] TN 39.2 Poughon L.. "Towards a dynamic model of the MELiSSA loop", Nov. 1998, Memorandum of Understanding ECT/FG/MMM/97.012. ESA/ESTEC P.O. 161.031. 1998

[13] L. Poughon MELiSSA Technical note 17.3. Contract ESA-ESTEC 8125/88/NL/FG - CCN4 " MELiSSA: simulation and modelling: *Spirulina* modelling ". 1994

ANNEX 1 : LEAN BODY MASS CALCULATION

The formula for lean body mass using the method of James[1,2] is:

Lean Body Weight (men) = (1.10 x Weight(kg)) - 128 x ($Weight^2/(100 \times Height(m))^2$)
Lean Body Weight (women) = (1.07 x Weight(kg)) - 148 x ($Weight^2/(100 \times Height(m))^2$)

An alternate formula for lean body mass using the method of Hume[3] is

For men over the age of 16: lean body mass in kilograms = (0.32810 * (body weight in kilograms)) + (0.33929 * (height in centimeters)) - 29.5336

For women over the age of 30: lean body mass in kilograms = (0.29569 * (body weight in kilograms)) + (0.41813 * (height in centimeters)) - 43.2933

These formulas are trusted and highly scientific, based on various types of measurements of human body composition, including dual energy X-ray absorptiometry (DEXA). However, remember that they are based on "averages". They predict the lean body weight "average" of a group of people with similar height and weight.

But you, as an individual, might have more muscle than others in your group, or bigger internal organs, or denser bones, or any number of factors that make you a little different than others. So even the lean body weight formulas are estimates.

References:

1. Hallynck TH Soep HH et al. Should clearance be normalised to body surface or to lean body mass? Br J Clin Pharmacol. 1981; 11: 523-526.

2. James WPT. Research on obesity. London. Her Majesty's Stationery Office.

3. Hume R. Prediction of lean body mass from height and weight. J Clin Path. 1966; 19

ANNEX 2 : eer_dyn_004.m – hourly EER computing for a crew

```
function [sys,x0,str,ts] = eer(t,x,u,flag,eer_calcul_method)
%=============================================================================
% Fonction EER : S-fonction pour simulink
%   Bibliotheque SIMULINK-MELISSA
%
%   Sky June-2005
%   Version 0.0.4
%   Initialise la definition de l'equipage , les planning d'activité
%   del'equipage et leur demande energetique par periode de 24 heures
%   Calcul le signal EER de chaque membre d'equipage
%
%   1// mdlOutputs : Définit le Vecteur de sortie du compartiment en
%   fonction de l'etat apres traitement des ODE et des opérations discretes
%
%   2// mdlDerivatives :
%
%   3// mdlUpdates : mise a jour de l'etat - fonction discrete
%
%   Parametres entrant :
%               eer_calcul_method : 1=method EER=function( Lean Body mass)
%                                   2=method EER=fixed value for each
activity
%
%
%=============================================================================
%   Entrées :
%
%   Sorties : 3

%  Vecteur d'etat (travail et integration) : 3

%=============================================================
switch flag
  case 0
    [sys,x0,str,ts] = mdlInitializeSizes(eer_calcul_method); %
Initialization

  case 1
    sys = mdlDerivatives(t,x,u,eer_calcul_method); % Calculate derivatives

  case 2
    sys = mdlUpdates(t,x,u,eer_calcul_method); % Update discrete states

  case 3
    sys = mdlOutputs(t,x,u,eer_calcul_method); % Calculate outputs

  case {4, 9} % Unused flags
    sys = [];

  otherwise
    error(['unhandled flag = ',num2str(flag)]); % Error handling
end
% End of dsfunc.



%=============================================================================
% Initialization
%=============================================================================
function [sys,x0,str,ts] = mdlInitializeSizes(eer_calcul_method)
% Call simsizes for a sizes structure, fill it in, and convert it
% to a sizes array.



% STEP 1 ---------------------------------------------------------------------
%------------------------------------------- INITIALISING - DATA LOADING

%------------------------------------------------ Load Crew definition
% the crew is defined in the file crew_def
% the file has the format :
%       crew number    mass (kg)   height (cm)   age (year)
%           1           70          170             30
%           2           80          185             36
%           3           71          175             32

crew_definition=load('../data/crew_def.txt');
crew_number=size(crew_definition,1); % number of line = number of crew

%---------------------------------------- Lean Body MASS (LBM) - Men
%                                        Method of Hume (report to
%                                        TN83.2
%   For men over the age of 16:
%       lean body mass in kilograms = (0.32810 * (body weight in
kilograms)) + (0.33929 * (height in centimeters)) - 29.5336
%   For women over the age of 30:
%       lean body mass in kilograms = (0.29569 * (body weight in
kilograms)) + (0.41813 * (height in centimeters)) - 43.2933

for crew_i=1:crew_number
    if ((crew_definition(crew_i,1)>0) &&
(crew_definition(crew_i,1)<=crew_number))

LBM(crew_definition(crew_i,1))=0.32810*crew_definition(crew_i,2)+(0.33929*c
rew_definition(crew_i,3))-29.5336;
```

```
    else
        disp ('EER_dyn_xxx : Error for crew identification')
        stop
    end
end

%-----------------------------------------------------load planning matrix :
%   currenly assumed all crew member have the same activity at the same
time
%   - column 1 : duration of activity (starting at day time undefined) but
%                sum of duration = 1 day
%   - column 2 : activity code
%                   1 = sleep
%                   2 = rest/leisure
%                   3 = physical activity
%                   4 = work
%   - column 3 : crew index (must be compatible with the number of the crew
%                         defined in crew_def.txt=
planning_matrix=load('../data/planning.txt');


%-------------------------------------- Setting of the activities of the
%                                       crew member form the planning and
%                                       calculation of the EER
time_activ=zeros(size(planning_matrix,1),4);
for crew_i=1:crew_number
    switch eer_calcul_method
        case 1
        %-----------------------------------------Energy Expenditure rate
in
        %                                           kcal/h.person for
activities
        %       Currently specific of each crew member - from LBM formula
        %       Reference data : TN 83.2
        % :::: Sleep :::::
        EER_activity(1,crew_i)=1.5*LBM(crew_i)-8.5;

        % :::: Rest ::::
        EER_activity(2,crew_i)=2.5*LBM(crew_i)-0.09;

        % :::: physical activity ::::
        EER_activity(3,crew_i)=6.4*LBM(crew_i)+1.4;

        % :::: work ::::
        EER_activity(4,crew_i)=3.5*LBM(crew_i)+0.09;

        case 2
        %-----------------------------------------Energy Expenditure rate
in
        %                                           kcal/h.person for
activities
        %       Currently specific of each crew member - from fixed values
        %       Reference data : TN 83.2
        % :::: Sleep :::::
        EER_activity(1,crew_i)=75;

        % :::: Rest ::::
        EER_activity(2,crew_i)=77;

        % :::: physical activity ::::
        EER_activity(3,crew_i)=250;

        % :::: work ::::
        EER_activity(4,crew_i)=150;

    end
    %-----------------------------------------Checking planning activities
    %   sum of activities == 24h
    %   defintition of day time in hours of start/end of each activity
    % time_activ : each line == 1 activity
    %   column 1 : time of start
    %   column 2 : time of end
    %   column 3 : crew member identification
    %   column 4 : code of activity
    eer_day(crew_i)=0;
    time_i(crew_i)=0;
    initial_activ(crew_i,1)=0;

    for i=1:size(planning_matrix,1)
        if (planning_matrix(i,3)==crew_i) % selection of palaning of crew_i
            if (initial_activ(crew_i,1)==0)

initial_activ(crew_i,1)=EER_activity(planning_matrix(i,2),crew_i);
                initial_activ(crew_i,2)=planning_matrix(i,2);
            end
            time_activ(i,1)=time_i(crew_i);
            time_activ(i,3)=crew_i;
            time_activ(i,4)=planning_matrix(i,2);
            time_activ(i,2)=planning_matrix(i,1)+time_i(crew_i);

eer_day(crew_i)=eer_day(crew_i)+EER_activity(planning_matrix(i,2))*planning
_matrix(i,1);
            time_i(crew_i)=time_activ(i,2);
        end
    end

    % At this point, activity are normaly defined for each crew member
    % checkin if planning correspond to a 24h/day
    if (time_i(crew_i)~=24 )
```

```
        for i=1:size(planning_matrix,1)
            if (time_activ(i,3)==crew_i) % selection of palaning of crew_i
                time_activ(i,[1 2])=time_activ(i,[1
2]).*(24/time_i(crew_i));
            end
        end
        eer_day(crew_i)=eer_day(crew_i).*(24/time_i(crew_i));
    end

end

%-------------------------------------------Sauvegarde des resultats de
%                                          l initialisation des activité
%                                          de l'equipage et de leur EER
save '../data/calculs_crew.mat' eer_day EER_activity time_activ crew_number
initial_activ;

% STEP 2 ------------------------------------------------------------------
%---------------------------------- Initialisation des tailles des matrices
%---------------------------------- pour les entrée/sorties du S-block
    sizes = simsizes;
    sizes.NumContStates  = 0; % Pour simulation continue (mdlderivaties)
    sizes.NumDiscStates  = 3*crew_number;
    sizes.NumOutputs     = 3*crew_number; % Sortie de la fonction
    sizes.NumInputs      = 0 ; % Entrée de la fonction
    sizes.DirFeedthrough = 0;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);

% Initialize the discrete states.
    for crew_i=1:crew_number
        xo((crew_i-1)*3+1)=EER_activity(initial_activ(crew_i,2),crew_i);
        x0((crew_i-1)*3+2)=3000;
        x0((crew_i-1)*3+3)=initial_activ(crew_i,2);
    end

    str = [];   % Set str to an empty matrix.
    ts  = [0.1, 0];    % sample time: [period, offset] - pour continu tout
a 0

%=============================================================================
% End of mdlInitializeSizes.
%=============================================================================



%=============================================================================
% mdlDerivatives
%=============================================================================
function sys = mdlDerivatives(t,x,u,eer_calcul_method)
%-------------------------------------------------------------------
% Entrée :  t - valeur du temps
%           u - flux de matiére en entrée (taille 2*nombre de corps)
%           x : vecteur d'etat correspondant à u pour l'etat du reacteur
%
%
%
% Parametres :
%
% Sorties : sys ==  derivée du vecteur d'etat
%-------------------------------------------------------------------
sys = [];
% End of mdlDerivaties.



%=============================================================
% Update the discrete states
% Creation du "signal " EER
%=============================================================
function sys = mdlUpdates(t,x,u,eer_calcul_method)

if (exist('../data/calculs_crew.mat')==2)
    load('calculs_crew.mat');
else
    disp('ERROR : calcul_crew.mat missing')
end


%=============================================================================
% GENERATING THE EER SIGNAL OF THE CREW
% t is the current simulation time
% t_day is the current time in one day (between 0 and 24)
% sys(i+1) is the output signal : kcal/day.crew.member
% sys(i+2) is the daily EER for the crew member
% sys(i+3) is the code of the curent activity of the crew member
%=============================================================================
t_day=(t/24-floor(t/24))*24;

for crew_i=1:crew_number
    % initialising for each member to first activity of the day
    i=1;
    sys((crew_i-1)*3+1)=EER_activity(initial_activ(crew_i,2),crew_i);
    sys((crew_i-1)*3+3)=initial_activ(crew_i,2);

    while (i<=size(time_activ,1))
        if (t_day>=time_activ(i,1) && t_day<time_activ(i,2))
            if (time_activ(i,3)==crew_i)
                sys((crew_i-
1)*3+1)=EER_activity(time_activ(i,4))*crew_number;
```

```
                sys((crew_i-1)*3+3)=time_activ(i,4);
            end
        end
        i=i+1;
    end
    sys((crew_i-1)*3+2)=eer_day(crew_i);

end
% End of mdlUpdate.



%=============================================================
% Calculate outputs - MODELE DISCRET
%=============================================================
function sys = mdlOutputs(t,x,u,eer_calcul_method)
%-------------------------------------------------------------
% entrées  - u : flux en entrée = aucun
%           - t : temps (inusité en Régime permanent)
%           - x : vecteur d'etat
%
% Sorties sys(i)
%
% Parametres :  Lumiere en W/m2
%
% Auteur : Sky
%          Version 0.0.1 - mai 2005
%-------------------------------------------------------------
sys=x;  % la sortie correspond a l'etat discret du systeme

%=============================================================================
% End of mdlOutputs.
```

ANNEX 3 : crew_respiration_dyn_001.m – hourly EER computing for a crew

```
function [sys,x0,str,ts] = crew_respiration(t,x,u,flag,cabin_volume)
%=========================================================================
% Fonction respiration membre equipage : S-fonction pour simulink
%   Bibliotheque SIMULINK-MELISSA
%
%   Sky Mai-2005
%   Version 0.0.2
%
%   1// mdlOutputs : Défini le Vecteur de sortie du compartiment en
%   fonction de l'etat apres traitement des ODE et des opérations discretes
%
%   2// mdlDerivatives :
%
%   3// mdlUpdates :
%
%   Parametres entrant :
%
%
%
%=========================================================================
%   Entrées :
%
%   Sorties : 3

%  Vecteur d'etat (travail et integration) : 3

%=================================================================
switch flag
  case 0
    [sys,x0,str,ts] = mdlInitializeSizes; % Initialization

  case 1
    sys = mdlDerivatives(t,x,u,cabin_volume); % Calculate derivatives

  case 2
    sys = mdlUpdates(t,x,u,cabin_volume); % Update discrete states

  case 3
    sys = mdlOutputs(t,x,u,cabin_volume); % Calculate outputs

  case {4, 9} % Unused flags
    sys = [];

  otherwise
    error(['unhandled flag = ',num2str(flag)]); % Error handling
end
% End of dsfunc.



%=========================================================================
% Initialization
%=========================================================================
function [sys,x0,str,ts] = mdlInitializeSizes
% Call simsizes for a sizes structure, fill it in, and convert it
% to a sizes array.
load_compounds;
load('../data/calculs_crew.mat');
if (exist('../data/calculs_respiration.mat')==2)
    delete('../data/calculs_respiration.mat');
end
%---- Initialisation des tailles des matrices
    sizes = simsizes;
    sizes.NumContStates  = 23; % Pour simulation continue (mldderivaties)
    sizes.NumDiscStates  = 0;
    sizes.NumOutputs     = 25; % Sortie de la fonction
    sizes.NumInputs      = 23+1+3*crew_number ; % Entrée de la fonction
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);

% Initialize the discrete states.
%       en g/l  pour liquide
%       en fraction pour gaz
    x0 = zeros(sizes.NumContStates,1);
    x0(i_o2)=0.21;
    x0(i_n2)=0.72;


    str = [];   % Set str to an empty matrix.
    ts  = [0, 0];    % sample time: [period, offset] - pour continu tout a
0

%=========================================================================
% End of mdlInitializeSizes.
%=========================================================================



%=========================================================================
% mdlDerivatives
%=========================================================================
function sys = mdlDerivatives(t,x,u,Vol_cabin)
%-------------------------------------------------------------
```

```
% Entrée :  t - valeur du temps
%           u - flux de matiére en entrée (taille 2*nombre de corps)
%           x : vecteur d'etat correspondant à u pour l'etat du reacteur
%
%
%
% Parametres :
%
% Sorties : sys ==  derivée du vecteur d'etat
%-------------------------------------------------------------------
%===================================================================
% INITIALISATION
%===================================================================

%------------------------------------------------------ setting
%                                                       configuration of
cabin
%Vol_cabin=40e3; % in Liters
Press=1;
Temp=25;

load_compounds;
    PM=[12 1 16 14 32 31]*M_C;
    PM(i_om)=100;
    PM(i_ns)=100;
    PM(i_nb)=100;
    PM(i_rh)=100;
    PM(i_sp)=100;

moles_GPN=1.013e5*1e-3/8.314/(25+273.15); % moles dans 1 L Gaz parfait ,
unitées normales
moles_GReac=Press*1.013e5*1e-3/8.314/(Temp+273.15); % moles dans 1 L Gaz
parfait , condtions du reacteur

G_in=u(24);
G_out=G_in;

%------------------------------------------------------ recovery of EER for
%                                                       each crew member

% :: recovery of all data of the crew
load('../data/calculs_crew.mat');


fraction=u([1:23]);

for crew_i=1:crew_number
    EER_rate(crew_i)=u(n_compounds+(crew_i-1)*3+1+1);
    crew_activity(crew_i)=u(n_compounds+(crew_i-1)*3+3+1);
end

%-----------------------------------------------O2/CO2 rate in
%                                                mol/h.crew for activities
%                                                Reference data : TN 83.2
rx([1:n_compounds])=0;
for crew_i=1:crew_number
    switch crew_activity(crew_i)
        case 1
        % :::: Sleep :::::
        O2_rate=(0.31*EER_rate(crew_i)-0)/32;
        CO2_rate=(0.29*EER_rate(crew_i)-0)/44;

        case 2
        % :::: Rest ::::
        O2_rate=(0.3*EER_rate(crew_i)-0.1)/32;
        CO2_rate=(0.31*EER_rate(crew_i)+0.1)/44;

        case 3
        % :::: physical activity ::::
        O2_rate=(0.28*EER_rate(crew_i)+0.1)/32;
        CO2_rate=(0.39*EER_rate(crew_i)-0)/44;

        case 4
        % :::: work ::::
        O2_rate=(0.29*EER_rate(crew_i)-0)/32;
        CO2_rate=(0.34*EER_rate(crew_i)-0)/44;
    end

    rx(i_o2)=rx(i_o2)-O2_rate;
    rx(i_co2)=rx(i_co2)+CO2_rate;
end
%===================================================================
% RESPIRATION RATES (in moles/h) AS FUNCTION OF EER rate (in kcal/h)
% Reference TN 83.2
%===================================================================
sys([1:n_compounds])=0;
for i=1:n_compounds
    gas_in(i)=fraction(i)*moles_GPN*G_in; % moles gas/h en entrée
    gas_out(i)=x(i)*moles_GPN*G_out; % moles gas/h en entrée
    gas_reac(i)=x(i)*moles_GReac*Vol_cabin; % moles gas dans réacteur à t
    sys(i)=(gas_in(i)-gas_out(i)+rx(i))/(moles_GReac*Vol_cabin); % En
supposant nbre mole constant-->
                                                                  %
variation affectant seulement le debit
end

if (rx(i_o2)~=0)
    QR=rx(i_co2)/rx(i_o2);
else
    QR=0;
```

```
end

save '../data/calculs_respiration.mat' G_out QR;
% End of mdlDerivaties.



%=============================================================
% Update the discrete states
% Creation du "signal " EER
%=============================================================
function sys = mdlUpdates(t,x,u,cabine_volum)
%=============================================================================
% INITIALISING - DATA LOADING
%=============================================================================
sys=[];

% End of mdlUpdate.



%=============================================================
% Calculate outputs - MODELE DISCRET
%=============================================================
function sys = mdlOutputs(t,x,u,cabin_volume)
%-------------------------------------------------------------
% entrées  - u : flux en entrée = aucun
%           - t : temps (inusité en Régime permanent)
%           - x : vecteur d'etat
%
% Sorties sys(i)
%
% Parametres :  Lumiere en W/m2
%
% Auteur : Sky
%          Version 0.0.1 - mai 2005
%-------------------------------------------------------------
sys=x;  % la sortie correspond a l'etat discret du systeme
if (exist('../data/calculs_respiration.mat')==2)
    load('../data/calculs_respiration.mat');
    sys(24)=G_out;
    sys(25)=-QR;
else
    sys(24)=0;
    sys(25)=0;
end

%=============================================================
% End of mdlOutputs.
```

ANNEX 4 : C4a_dyn_004.m – Spirulina photobioreactor

```
function [sys,x0,str,ts] =
C_IVa(t,x,u,flag,Temp,Press,pH,kla,gdr_reac,volume_t,frac_gas,type_reac,typ
e_bio)
%=========================================================================
% Fonction C_IVa : S-fonction pour simulink
%   Bibliotheque SIMULINK-MELISSA
%
%   Sky Mai-2005
%   Version 0.0.6
%
%   1// mdlOutputs : Défini le Vecteur de sortie du compartiment en
%   fonction de l'etat apres traitement des ODE et des opérations discretes
%
%   2// mdlDerivatives : Modèle régime dynamique (ode) pour Compartiment
IVa
%   (Spiruline) de MELiSSA
%
%   Parametres entrant :
%
%              Temp :  température en °C
%              Press : Pression en atm
%              pH : ph - fixé (supposé sous controle)
%              kla : en h-1; echange G/L
%              gdr_reac : grandeur caracteristique du réacteur
%                         (epaisseur ou rayon) en metre
%              volume_t : volume utile du réacteur en L
%              frac_gas : fraction de volume occupé par le gaz
%                         supposée fixe
%              type reac : 1 - rectangulaire eclairé d'un coté
%                          2 - rectangulaire eclairé de 2 cotés
%                          3 - cylindrique - eclairage radial
%
%              type_bio : 1  - 1 stoichiométrie, variable avec lumiere (TN
17.3)
%                         2  - 2 stoichiométries fixes (variabilité
globale liée à
%                         la cinétique)
%
%=========================================================================
%   Entrées : 2*n+2
%           les n_composés sous forme liquide - masse
%           les n_composés sous forme gaz -     fraction
%           débit de liquide                -     Vol/temps
%           débit de gaz                          Nm3/temps
%           Lumiere  (oblige de le faire passer par les variables)  en W/m2
%
%   Sorties : 2*n+7
%           les n_composés sous forme liquide - masse
%           les n_composés sous forme gaz -     fraction
%           débit de liquide                -     Vol/temps
%           débit de gaz                          Nm3/temps
%           resultats intermédiaires de calculs  info 1 : rx total
%                                                info 2 : gamma
%                                                info 3 : facteur
limitation substrat
%                                                info 4 : volume liquide
%                                                info 5 : taux de dilution
%                                                (modifiable dans la
%                                                routine mdloutput)
%
%  Vecteur d'etat (travail et integration) : 2*n+8
%           Meme distribution que le vecteur de sortie, avec
%           en 2*n+8 mis le calcul intermédiaire correspondant a la
%           quantité d'eps dans la biomasse (avec modele biologique 2)
%=========================================================================
switch flag
  case 0
    [sys,x0,str,ts] = mdlInitializeSizes; % Initialization

  case 1
    sys =
mdlDerivatives(t,x,u,Temp,Press,pH,kla,gdr_reac,volume_t,frac_gas,type_reac
,type_bio); % Calculate derivatives

  case 2
    sys =
mdlUpdates(t,x,u,Temp,Press,pH,kla,gdr_reac,volume_t,frac_gas,type_reac,typ
e_bio); % Update discrete states

  case 3
    sys =
mdlOutputs(t,x,u,Temp,Press,pH,kla,gdr_reac,volume_t,frac_gas,type_reac,typ
e_bio); % Calculate outputs

  case {4, 9} % Unused flags
    sys = [];

  otherwise
    error(['unhandled flag = ',num2str(flag)]); % Error handling
end
% End of dsfunc.



%=========================================================================
% Initialization
```

```
%=========================================================================
function [sys,x0,str,ts] = mdlInitializeSizes
% Call simsizes for a sizes structure, fill it in, and convert it
% to a sizes array.

%---- Recuperation de la configuration donnant le nombre de corps
%---- load_compounds : nombres de corps et leur indices
%---- load partition :  equilibre gaz/liquide pour 25°C 1atm
    load_compounds;
    Temp=25;
    Press=1;
    load_partitions;
if (exist('../data/calculs_c4a.mat')==2)
    delete('../data/calculs_c4a.mat');
end


%---- Initialisation des tailles des matrices
    sizes = simsizes;
    sizes.NumContStates  = 2*n_compounds+2+5+1; % Pour simulation continue
(mdlderivaties)
    sizes.NumDiscStates  = 0;
    sizes.NumOutputs     = 2*n_compounds+2+5; % Sortie de la fonction
    sizes.NumInputs      = n_compounds*2+2+1 ; % Entrée de la fonction
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);

% Initialize the discrete states.
%       en g/l  pour liquide
%       en fraction pour gaz
    x0 = zeros(sizes.NumContStates,1);
    x0(i_eau)=1000;
    x0(i_sp)=1 ; %initialisation pour biomasse spiruline
    x0(length(x0))=0.01*x0(i_sp); % initialisation EPS de spiruline
    x0(i_hno3)=1;
    x0(i_h2so4)=1;
    x0(i_h3po4)=1;
    x0(i_o2+n_compounds)=0.2009;
    x0(i_n2+n_compounds)=0.79;
    x0(i_co2+n_compounds)=0.001;
    x0(i_co2)=x0(i_co2+n_compounds)/ki(i_co2)*55.55*44;
    x0(i_o2)=x0(i_o2+n_compounds)/ki(i_o2)*55.55*32;
    x0(i_eau+n_compounds)=x0(i_eau)*ki(i_eau)/55.55/18;

    str = [];   % Set str to an empty matrix.
    ts  = [0, 0];    % sample time: [period, offset] - pour continu tout a
0

%=========================================================================
% End of mdlInitializeSizes.
%=========================================================================



%=========================================================================
% mdlDerivatives
% Return the derivatives for the continuous states.
% MODELE DYNAMIQUE PHOTOBIOREACTEUR (Culture Spiruline)
%=========================================================================
function sys = mdlDerivatives(t,x,u,Temp,Press,pH,kla, ...
                              gdr_reac,volume_t,frac_gas,type_reac,type_bio)
%-------------------------------------------------------------
% Entrée :  t - valeur du temps
%           u - flux de matiére en entrée (taille 2*nombre de corps)
%               --> 1 à nombre de corps :phase liquide [g/l]
%               --> 2*nombre de corps : phase gaz  [molar fraction]
%               --> 1+2*n_compound : debit liquide l/h
%               --> 2+2*n_compounds = débit gaz Normaux-Litres /h
%               --> 1+2+2*n_compounds = Lumiere  (oblige de le faire passer
par les variables)  en W/m2
%
%           x : vecteur d'etat correspondant à u pour l'etat du reacteur
%
%
%
% Parametres :
%                Temp :  température en °C
%                Press : Pression en atm
%                pH : ph - fixé (supposé sous controle)
%                kla : en h-1; echange G/L
%                gdr_reac : grandeur caracteristique du réacteur
%                            (epaisseur ou rayon)
%                volume_t : volume total utile du réacteur
%                frac_gas : fraction de volume occupé par le gaz
%                            supposée fixe
%                type reac : 1 - rectangulaire eclairé d'un coté
%                            2 - rectangulaire eclairé de 2 cotés
%                            3 - cylindrique - eclairage radial
%
%                type_bio : 1  - 1 stoichiométrie, variable avec lumiere (TN
17.3)
%                           2  - 2 stoichiométries fixes (variabilité
globale liée à
%                           la cinétique)
%
% Sorties : sys ==  derivée du vecteur d'etat
% Vecteur d'etat (travail et integration) :  x  == 2*n+8
%           Meme distribution que le vecteur de sortie, avec
%           en 2*n+8 mis le calcul intermédiaire correspondant a la
```

```
%           quantité d'eps dans la biomasse (avec modele biologique 2)
%-----------------------------------------------------------------

%--------Corps en jeu (liste triée dans l'ordre de stockage dans les flux
de matiere)
%   1) eau
%   2) nh3 et ses formes ioniques (distribution fonction du pH)
%   3) h2so4 et ses formes ioniques (distribution fonction du pH)
%   4) h3po4 et ses formes ioniques (distribution fonction du pH)
%   5) hno3 et ses formes ioniques (distribution fonction du pH)
%   6) hno2 et ses formes ioniques (distribution fonction du pH)
%   7) urée
%   8) o2
%   9) co2 ses formes ioniques (distribution fonction du pH)
%   10) acetate
%   11) propionate
%   12) butyrate (iso assumé equivalent)
%   13) valerate (iso assumé equivalent)
%   14) caproate (iso assumé equivalent)
%   15) n2
%   16) h2
%   17) ch4
%   18) Gaz inerte dissous
%   19) matiere_organique_mixée (flux normalisé)
%   20) biomasse nitrosomonas
%   21) biomasse nitrobacter
%   22) biomasse rhodobacter
%   23) biomasse spiruline totale (active + eps)
%   ----------------------------------------------------

warning off MATLAB:fzero:UndeterminedSyntax;

%================================================================
% 0) initialisations des données
%================================================================
sys=zeros(size(x));

%-----------------------------------------------CORPS ET PROPRIETE
%                                              DEFINTION DES INDICES
TABLEAU DES CORPS
%                                              VOIR FICHIER LOAD_COMPOUNDS
load_compounds;     %--definition corps, indices, M_C,
load_partitions;    %--PB de l'équilibre gaz/liquide
moles_GPN=1.013e5*1e-3/8.314/(25+273.15); % moles dans 1 L Gaz parfait ,
unitées normales
moles_GReac=Press*1.013e5*1e-3/8.314/(Temp+273.15); % moles dans 1 L Gaz
parfait , condtions du reacteur
lumiere=u(1+2+2*n_compounds);

%-----------------------------------------------VALEUR PAR DEFAUT
%                                                    DES PARAMETRES MAL
%                                                    CONDITIONNES
if (lumiere<0 || lumiere >1000)
    disp (['error light=' num2str(lumiere) 'reset to default' ]);
    lumiere=100; %Defaut eclarement 100 W/m2
end
if (Temp<-273.15 || Temp >373)
    Temp=25+273.15 %Defaut Temperature = 25°C
end
if (Press<0 || Press >10)
    Press=1 %Defaut Pression = 1 atm
end
if (frac_gas<=0 || frac_gas >=1)
    frac_gas=0.1 %Defaut fraction de vomume de gaz = 10%
end
if (pH<1 || pH >13)
    pH=10 %Defaut pH=10
end
if (gdr_reac<=0 )
    gdr_reac=1 %Defaut
end
if (volume_t<=0 )
    volume_t=1 %Defaut : 1 L
end
if (kla<0 )
    kla=50 %Defaut
end

%check parameters
%disp (['light=' num2str(lumiere) ' pH=' num2str(pH) ' Temp=' num2str(Temp)
' react=' num2str(type_reac) ' bio=' num2str(type_bio)])



%=========================================================================
% 1/ Bilan massique - Equation stoichiometrique - calcul des coefficients
%           2 : 1 equation biomasse active + 1 equation EPS
%           1 : 1 Equation globale variable en fonction de la lumiere
%=========================================================================
switch type_bio
case 1
    %-----------------------------------------------1 STOICHIOMETRIE
    % reactions pour Production de biomasse
    %    pour prendre en compte la variation de composition on résoud à
chaque
    %    fois plutot que de fixer les coefficients stoichiometriques
    %      reaction : 1 CO2 + O2 + +NH3 + H2SO4 +H3PO4 + H2O
    %                                  --->
    %                 biomasse SPIRU
```

```
    %
    %    pour resoudre ce système il suffit de fixer un coefficient
(Biomasse)

    % Chargement de la composition de spiruline fonction de la lumiere dans
la structure "spiru."
    Spiru=composition_spiru(lumiere);
    save('../data/composition_sp.mat' , 'Spiru');

    % Calcul des masses molaire standard
    % On rajoute les composition et masses molaire pour les corps
specifique
    % organique de composition variable
    % Seule biomasse Spiruline est importante. (corps 23)
    % Pour les autres corps on ne rempli pas MC et PM=100
    PM=[12 1 16 14 32 31]*M_C;
    PM(i_om)=100;
    PM(i_ns)=100;
    PM(i_nb)=100;
    PM(i_rh)=100;
    M_C(:,i_sp)=[Spiru.C ; Spiru.H ; Spiru.O ; Spiru.N ; Spiru.S ;
Spiru.P];
    PM(i_sp)=Spiru.PM;

    Acc=zeros(n_compounds,1);
    Acc(i_sp,1)=1; % Accumulation Biomasse

    M1=zeros(n_compounds,n_compounds);
    M1([1:6],:)=M_C;
    M1(7,i_nh3)=1; % nh3 =0
    M1(8,i_hno2)=1; % no2=0
    M1(9,i_uree)=1; % urée=0
    M1(10,i_inert)=1; % inerte=0
    M1(11,i_om)=1; % OM=0
    M1(12,i_ns)=1; % Ns=0
    M1(13,i_nb)=1; % Nb=0
    M1(14,i_rh)=1; % Rhodo=0
    M1(15,i_prop)=1; % propionate=0
    M1(16,i_buty)=1; % butyrate=0
    M1(17,i_val)=1; % valerate=0
    M1(18,i_capro)=1; % caproate
    M1(19,i_n2)=1; % N2
    M1(20,i_h2)=1; % h2
    M1(21,i_ch4)=1; % ch4
    M1(22,i_acetate)=1; % acetate
    M1(23,i_sp)=1; % Biomasse spiru totale (active + eps)

    coeff_R1=pinv(M1)*Acc; % Calcul des coeffcients stoichiometriques
reaction 1
    eps_fraction=Spiru.exopoly.percent;


case 2
    %-----------------------------------------------2 STOICHIOMETRIES FIXEES
    % 1 stoechiométrie production de biomasse active
    %    CO2 + 0.673 H2o + 0.192 HNO3 + 0.005 H2SO4 + 0.006 H3PO4
    %          --> CH(1.566)O(0.405N(0.192)S(0.005)P(0.006) + 1.444 O2
    % 1 stoechiometrie production EPS (exopolysaccharide)
    %    CO2 + 0.673 H2o + 0.015 H2SO4
    %          --> CH(1.65)O(0.95)S(0.015) + 0.96 O2
    % D'aprés TN 19.3
    %
    % ICI i_sp correspond a la biomasse totale (actif+eps)
    %     n_compounds+1 <=> length(x): dernier indice du vecteur d'etat
correspond a l'eps
    %
    coeff_R1=zeros(n_compounds+1,1); % coefficient reaction 1 : biomasse
active
    coeff_R2=zeros(n_compounds+1,1); % coefficient reaction 2 :
exopolysaccharide

    if (t==0)
        %--Initialisation du système a partir d'une compostion obtenue par
        %--la fonction compostion_sp(lumiere). on modifie les données pour
        %--stocker dans spiru.C .. et spiru.PM les données relatives a la
        %--biomasse active
        Spiru=composition_spiru(lumiere);

        Spiru.prot.percent=0;
        Spiru.lip.percent=0; %
        Spiru.exopoly.percent=(9.028735632+0.109770115*lumiere)/100*0.96;
        Spiru.carb.percent=0; %
        Spiru.arn.percent=0.0;
        Spiru.adn.percent=0.0;
        Spiru.C=1;
        Spiru.H=1.566;
        Spiru.O=0.405;
        Spiru.N=0.192;
        Spiru.S=0.005;
        Spiru.P=0.006;

Spiru.PM=12*Spiru.C+Spiru.H+16*Spiru.O+14*Spiru.N+32*Spiru.S+31*Spiru.P;
        save('../data/composition_sp.mat' , 'Spiru');

        %x(length(x))=Spiru.exopoly.percent*x(i_sp); % Initialisation pour
eps
    end

    load('../data/composition_sp.mat');
```

```
    %masses molaires de corps en jeu
    PM=[12 1 16 14 32 31]*M_C;
    PM(i_om)=100;
    PM(i_ns)=100;
    PM(i_nb)=100;
    PM(i_rh)=100;
    PM(i_sp)=Spiru.PM;
    PM(n_compounds+1)=Spiru.exopoly.PM;

    coeff_R1(i_sp)=1;  %Attention ici i_sp correspond a la fraction
biomasse active
    coeff_R1(i_co2)=-1;
    coeff_R1(i_o2)=1.444;
    coeff_R1(i_hno3)=-0.192;
    coeff_R1(i_h3po4)=-0.006;
    coeff_R1(i_h2so4)=-0.005;
    coeff_R1(i_eau)=-0.673;

    coeff_R2(n_compounds+1)=1; %Attention ici i_sp correspond a la fraction
eps
    coeff_R2(i_co2)=-1;
    coeff_R2(i_o2)=0.96;
    coeff_R2(i_h2so4)=-0.015;
    coeff_R2(i_eau)=-0.81;

    eps_fraction=x(length(x))/x(i_sp);
end




%==========================================================================
% 2) MODELES DYNAMIQUES ODE POUR CULTURE EN PHOTOBIOREACTEUR
%       type de reacteurs :
%           1 :  rectangulaire - eclairé d'un coté
%           2 : rectangulaire - eclairé des 2 cotés
%           3 : cylindrique - eclaré radialement
%==========================================================================

%-----------------------------------------------------PROPRIETES LUMIERE POUR
%                                                    SPIRULINE
%
% K_xa : W/m2 = Constante saturation pour biomasse active dans modele
lumiere
% K_eps : W/m2 = Constante saturation pour exopolysacch dans modele lumiere
% Ea : m2/kg = coefficient d'absorption massique moyen sprectrique
% Es : m2/kg = coefficient de diffusion massique moyen sprectrique
% b          = fraction d'energie rayonnante retrodiffusée
% mu_xa : 1/heure = taux de croissance biomasse active
% mu_eps : 1/heure = vitesse de production d'exopolysaccahride

% Modele 2 flux généralisé - 2006
%   Ea=162;     %coefficient d'absorption massique moyen sprectrique
%   Es=636;     %coefficient de diffusion massique moyen sprectrique
%   b=0.03;     %fraction d'energie rayonnante retrodiffusée

% Parametres modele shuster / JF. Cornet - TN 19.2 -1992/1995
Ea=150;
Es=200;
b=0.5;
K_xa=20;
K_eps=750;
mu_xa=0.073;
mu_eps=0.3;
ks_default=1e-7;

%-----------------------------------------------------CALCULS INITIAUX
%                                                    PARAMETRES MODELE
LUMIERE
para_alpha=sqrt(Ea/(Ea+2*b*Es));
para_delta=para_alpha*(Ea+2*b*Es)*x(i_sp)*(1-eps_fraction);  % biomasse
active
                                                                % uniquement

%-----------------------------------------------------PARAMETRES REACTEUR
Volume=volume_t; % Volume réacteur en L
Volume_L=Volume*(1-frac_gas); %VOlume de liquide
Volume_G=Volume*(frac_gas);
F_liq=u(1+2*n_compounds);
G_gas=u(2+2*n_compounds);

%-----------------------------------------------OPTIONS SUR FONCTIONS DE
%                                                    CALCULS
options_fzero = optimset('Display','off','TolX',1e-8); % pour execution de
fzero


switch type_reac
case 1
    %-----------------------------------------------REACTEUR RECTANGULAIRE
    %                                                ECLAIRAGE UNE FACE


    % Calcul de la longueur eclairee L (Localisation de la racine par
FZERO)
    L =
fzero('wil_reacteur',0,options_fzero,gdr_reac,lumiere,para_alpha,para_delta
,type_reac);
```

```
    if L>gdr_reac
        L=gdr_reac;
    end;
    gamma_L=L/gdr_reac; % factuer de limitation par la lumiere

    % Calcul de la vitesse specifique selon le modèle biologique----------
    switch type_bio
    case 1    %----> Modele 1 réaction stoichiometrie variable avec la
lumiere
        % Calcul de l'integrale de volume d'absorption du rayonnement
        J =
quadl('wiv_reacteur',0,L,[],[],gdr_reac,lumiere,para_alpha,para_delta,K_xa,
type_reac);

        %Calcul de la vitesse specifique de reference rx (masse biomasse /
unitées de temps : h-1)
        rx=mu_xa*x(i_sp)*J/gdr_reac;
        rx_px=rx;
        rx_eps=0;
        rx_act=0;

        % Prise compte de la limitation substrat :  S/Ks+S
        for i=1:n_compounds
            if abs(coeff_R1(i))<100*eps
                coeff_R1(i)=0;
            end
            if coeff_R1(i)<0 % prise en compte de la limitation substrat :
defaut KS=10[-8]
                if x(i)<=ks_default/50
                    %disp(['Warning limiting : ' num2str(i)])
                    rx=0;
                else
                    rx=rx*(x(i))/(x(i) + ks_default);
                end
            end
        end

    case 2    %----> Modele 2 reaction à 2 stoichiometries fixes
        % Calcul de l'integrale de volume d'absorption du rayonnement pour
        % biomasse active et eps
        J_act =
quadl('wiv_reacteur',0,L,[],[],gdr_reac,lumiere,para_alpha,para_delta,K_xa,
type_reac);
        J_eps =
quadl('wiv_reacteur',0,L,[],[],gdr_reac,lumiere,para_alpha,para_delta,K_eps
,type_reac);

        %Calcul vitesse specifique biomasse active rx_act
        rx_act=mu_xa*(x(i_sp)-x(length(x)))*J_act/gdr_reac;
        %Calcul de la vitesse specifique de eps
        rx_eps= mu_eps*(x(i_sp)-x(length(x)))*J_eps /gdr_reac;
        %%Calcul de la vitesse specifique de biomasse totale
        rx=rx_act+rx_eps;
        rx_px=rx;

        % Prise compte de la limitation substrat au sens large :  S/Ks+S
        rx_xa_lim=1;
        rx_eps_lim=1;
        for i=1:n_compounds
            if (coeff_R1(i)<0 ) % prise en compte de la limitation substrat
: defaut KS=10[-8]
                if x(i)<=ks_default/50
                    %disp(['Warning limiting : ' num2str(i)])
                    r_xa_lim=0;
                else
                    r_xa_lim=rx_xa_lim*(x(i))/(x(i) + ks_default);
                end
            end
            if (coeff_R2(i)<0) % prise en compte de la limitation substrat
: defaut KS=10[-8]
                if x(i)<=ks_default/50
                    %disp(['Warning limiting : ' num2str(i)])
                    rx_eps_lim=0;
                else
                    r_eps_lim=rx_eps_lim*(x(i))/(x(i) + ks_default);
                end
            end
        end

        rx=rx_act*rx_xa_lim+rx_eps*rx_eps_lim;

    otherwise  % ----> Modele inconnu
       disp('ERROR - BIOLOGICAL TYPE UNKNOWN');
       rx=0;
    end



case 2
    %-------------------------------------------------REACTEUR RECTANGULAIRE
    %                                                  ECLAIRAGE 2 FACES

    % Calcul de la longueur eclairee L (Localisation de la racine par
FZERO)
    % comme on eclaire des 2 coté si il existe une racine, elle est double
: L1 et L2
    % et symétrique par rapport au centre
    L =
fzero('wil_reacteur',0,options_fzero,gdr_reac,lumiere,para_alpha,para_delta
,type_reac);
```

```
    if (isfinite(L)==0)
        L=gdr_reac;
    end
    if L>gdr_reac/2
        L=gdr_reac/2;
    end;
    L1=L;
    L2=gdr_reac-L1;
    gamma_L=(2*L1)/gdr_reac;

    % Calcul de la vitesse specifique selon le modèle biologique
    switch type_bio
    case 1    %----> Modele 1 réaction stoichiometrie variable avec la
lumiere

        % Calcul de l'integrale de volume d'absorption du rayonnement
        J1 =
quadl('wiv_reacteur',0,L1,[],[],gdr_reac,lumiere,para_alpha,para_delta,K_xa
,type_reac);
        J2 =
quadl('wiv_reacteur',L2,gdr_reac,[],[],gdr_reac,lumiere,para_alpha,para_del
ta,K_xa,type_reac);
        J=J1+J2;

        %Calcul de la vitesse specifique de reference rx (masse biomasse /
unitées de temps : h-1)
        rx=mu_xa*x(i_sp)*J/gdr_reac;
        rx_px=rx;

        % Prise compte de la limitation substrat au sens large :  S/Ks+S
        for i=1:n_compounds
            if abs(coeff_R1(i))<100*eps
                coeff_R1(i)=0;
            end
            if coeff_R1(i)<0 % prise en compte de la limitation substrat :
defaut KS=10[-8]
                if x(i)<=ks_default/50
                    %disp(['Warning limiting : ' num2str(i)])
                    rx=0;
                else
                    rx=rx*(x(i))/(x(i) + ks_default);
                end
            end
        end
        rx_px=rx;
        rx_eps=0;
        rx_act=0;

    case 2    % ----> Modele 2 reaction à 2 stoichiometries fixes
        % Calcul de l'integrale de volume d'absorption du rayonnement
        J1_act =
quadl('wiv_reacteur',0,L1,[],[],gdr_reac,lumiere,para_alpha,para_delta,K_xa
,type_reac);
        J2_act =
quadl('wiv_reacteur',L2,gdr_reac,[],[],gdr_reac,lumiere,para_alpha,para_del
ta,K_xa,type_reac);
        J_act=J1_act+J2_act;
        J1_eps =
quadl('wiv_reacteur',0,L1,[],[],gdr_reac,lumiere,para_alpha,para_delta,K_ep
s,type_reac);
        J2_eps =
quadl('wiv_reacteur',L2,gdr_reac,[],[],gdr_reac,lumiere,para_alpha,para_del
ta,K_eps,type_reac);
        J_eps=J1_eps+J2_eps;

        %Calcul vitesse specifique biomasse active rx_act
        rx_act=mu_xa*(x(i_sp)-x(length(x)))*J_act/gdr_reac;
        %Calcul de la vitesse specifique de eps
        rx_eps= mu_eps*x(length(x))*J_eps /gdr_reac;
        %%Calcul de la vitesse specifique de biomasse totale
        rx=rx_act+rx_eps;
        rx_px=rx;

        % Prise compte de la limitation substrat au sens large :  S/Ks+S
        rx_xa_lim=1;
        rx_eps_lim=1;
        for i=1:n_compounds
            if (coeff_R1(i)<0 ) % prise en compte de la limitation substrat
: defaut KS=10[-8]
                if x(i)<=ks_default/50
                    %disp(['Warning limiting : ' num2str(i)])
                    r_xa_lim=0;
                else
                    r_xa_lim=rx_xa_lim*(x(i))/(x(i) + ks_default);
                end
            end
            if (coeff_R2(i)<0) % prise en compte de la limitation substrat
: defaut KS=10[-8]
                if x(i)<=ks_default/50
                    %disp(['Warning limiting : ' num2str(i)])
                    rx_eps_lim=0;
                else
                    r_eps_lim=rx_eps_lim*(x(i))/(x(i) + ks_default);
                end
            end
        end

    otherwise  % ----> Modele inconnu
       disp('ERROR - BIOLOGICAL TYPE UNKNOWN');
       rx=0;
```

```
    end

case 3
    %-----------------------------------------------REACTEUR cyclindrique
    %                                               ECLAIRAGE RADIAL
    % Calcul de la longueur (rayon) eclairee L à partir du centre
    %(Localisation de la racine par FZERO)
    % comme on eclaire radialement . 0=centre du reacteur , E=rayon du
    % reacteur
    L =
fzero('wil_reacteur',gdr_reac,options_fzero,gdr_reac,lumiere,para_alpha,par
a_delta,type_reac);
    if (isfinite(L)==0)
        L=0;
    end
    if L>gdr_reac
        L=gdr_reac;
    end;
    if L<0
        L=0;
    end

    gamma_L=(gdr_reac-L)/gdr_reac;

    % Calcul de la vitesse specifique selon le modèle biologique
    switch type_bio
    case 1    %----> Modele 1 réaction stoichiometrie variable avec la
lumiere

        % Calcul de l'integrale de volume d'absorption du rayonnement
        J =
quadl('wiv_reacteur',L,gdr_reac,[],[],gdr_reac,lumiere,para_alpha,para_delt
a,K_xa,type_reac);

        %Calcul de la vitesse specifique de reference rx (masse biomasse /
unitées de temps : h-1)
        rx=mu_xa*x(i_sp)*J/(pi*gdr_reac^2);
        rx_px=rx;

        % Prise compte de la limitation substrat au sens large :  S/Ks+S
        for i=1:n_compounds
            if abs(coeff_R1(i))<100*eps
                coeff_R1(i)=0;
            end
            if coeff_R1(i)<0 % prise en compte de la limitation substrat :
defaut KS=10[-8]
                if x(i)<=ks_default/50
                    %disp(['Warning limiting : ' num2str(i)])
                    rx=0;
                else
                    rx=rx*(x(i))/(x(i) + ks_default);
                end
            end
        end
        rx_px=rx;
        rx_eps=0;
        rx_act=0;

    case 2    % ----> Modele 2 reaction à 2 stoichiometries fixes
        % Calcul de l'integrale de volume d'absorption du rayonnement
        J_act=
quadl('wiv_reacteur',L,gdr_reac,[],[],gdr_reac,lumiere,para_alpha,para_delt
a,K_xa,type_reac);
        J_eps=
quadl('wiv_reacteur',L,gdr_reac,[],[],gdr_reac,lumiere,para_alpha,para_delt
a,K_eps,type_reac);

        %Calcul vitesse specifique biomasse active rx_act
        rx_act=mu_xa*(x(i_sp)-x(length(x)))*J_act/(pi*gdr_reac^2);
        %Calcul de la vitesse specifique de eps
        rx_eps= mu_eps*x(length(x))*J_eps /(pi*gdr_reac^2);
        %%Calcul de la vitesse specifique de biomasse totale
        rx=rx_act+rx_eps;
        rx_px=rx;

        % Prise compte de la limitation substrat au sens large :  S/Ks+S
        rx_xa_lim=1;
        rx_eps_lim=1;
        for i=1:n_compounds
            if (coeff_R1(i)<0 ) % prise en compte de la limitation substrat
: defaut KS=10[-8]
                if x(i)<=ks_default/50
                    %disp(['Warning limiting : ' num2str(i)])
                    r_xa_lim=0;
                else
                    r_xa_lim=rx_xa_lim*(x(i))/(x(i) + 1e-6);
                end
            end
            if (coeff_R2(i)<0) % prise en compte de la limitation substrat
: defaut KS=10[-8]
                if x(i)<=ks_default/50
                    %disp(['Warning limiting : ' num2str(i)])
                    rx_eps_lim=0;
                else
                    r_eps_lim=rx_eps_lim*(x(i))/(x(i) + ks_default);
                end
            end
        end

    otherwise  % ----> Modele inconnu
```

```
        disp('ERROR - BIOLOGICAL TYPE UNKNOWN');
        rx=0;
    end


otherwise
    disp('ERROR - REACTOR TYPE UNKNOWN');
    rx=0;
end

%================================================================
% système d'equation ode; incluant equilibre Gaz Liquide
% ATTENTION : (Notes importante)
%           u = [] entrée [mass/L]
%           x = vecteur d'état du système = [masse/L] dans le réacteur à t
%           sys = x' == variation de masse du systeme à t
%
%  A developper - idée de travailler uniqueme sur quantité et pas
%  concentration - puis retransformer en concentration ensuite
for i=1:n_compounds

    % Conversions des unités :
    %       liquide : de g/L ---> mole
    %       Gas     : de fraction ---> mole

    gas_in(i)=u(i+n_compounds)*moles_GPN*G_gas; % moles gas/h en entrée
    gas_out(i)=x(i+n_compounds)*moles_GPN*G_gas; % moles gas/h en entrée
    gas_reac(i)=x(i+n_compounds)*moles_GReac*Volume_G; % moles gas dans
réacteur à t
    liq_in(i)=u(i)/PM(i)*F_liq; % mole liq/h en entrée
    liq_out(i)=x(i)/PM(i)*F_liq; % mole liq/h en entrée
    liq_reac(i)=x(i)/PM(i)*Volume_L; % mole liq à t dans réacteur

    switch type_bio
    case 1
        %Réaction phase liquide : rxi en [mole i] /unité temps
        rxi=coeff_R1(i)/coeff_R1(i_sp) * rx/PM(i_sp)*Volume_L;

    case 2
        %reaction totale (biomasse active + eps) en mole /temps
        rxi= (coeff_R1(i)/coeff_R1(i_sp) * rx_act*rx_xa_lim/PM(i_sp) + ...
            coeff_R2(i)/coeff_R2(n_compounds+1) * rx_eps*rx_eps_lim
/PM(n_compounds+1) ) *Volume_L;

        if (i==i_sp)
            rxi=0 ; % pour i_sp on considere la biomasse totale - cas
traité en dehors de la boucle
        end

    end

    if ki(i)==0 % forme liquide uniquement
            Ech_GL=0;
    elseif ki(i)>=1e10 % Forme gaz uniquement
            Ech_GL=-rxi; % Equivalent a reaction immediate dans le gaz, et
le reste est inchangé
                         % Risque de bug si le corps est en entrée dasn le
                         % liquide car transfert non pris en compte
     else    % Equilibre G/L
           if ki(i)>1 % Gas-->Liquid
                C_etoile=x(i+n_compounds)*55.555/ki(i);
                n_etoile=C_etoile*Volume_L;
                % noms_c(i)
                %disp(['c* ' num2str(C_etoile/PM(i)) ' mol/l']);
                Ech_GL=kla*(n_etoile-liq_reac(i));

            else %Liquid --> gas
                C_etoile=ki(i)*(x(i)/PM(i)/55.555)*moles_GReac;
                n_etoile=C_etoile*Volume_G;
                Ech_GL=-kla*(n_etoile-gas_reac(i));

           end
     end

     if (i==i_o2)
         ro2=rxi; %vitesse molaire production o2
     end
     if (i==i_co2)
         rco2=rxi; %vitesse molaire production Co2
     end

     sys(i)=(liq_in(i)-liq_out(i)+rxi+Ech_GL)*PM(i)/Volume_L;
%==(dn/dt)/volume*MM
     sys(i+n_compounds)=(gas_in(i)-gas_out(i)-
Ech_GL)/(moles_GReac*Volume_G); % En supposant nbre mole constant-->

% variation affectant seulement le debit
end

% Equation pour l'exopolysaccahride et biomasse totale (modele 2)
% ATTENTION SUPPOSE AUCUNE ENTRE
if (type_bio==2)
    sys(length(x))=( 0-x(length(x)))*F_liq/Volume_L + rx_eps*rx_eps_lim ;
    sys(i_sp)=(u(i_sp)-x(i_sp))*F_liq/Volume_L
+(rx_act*rx_xa_lim+rx_eps*rx_eps_lim) ;
    sys(i_sp+n_compounds)=0;
    eps_fraction=x(length(x))/x(i_sp);
end

% Systeme pour prise en compte de la variation des debits volumiques Gaz
```

```
% (en NL/h) et Liquide en L/h
sys(1+2*n_compounds)=0;
sys(2+2*n_compounds)=0;
sys(1+2+2*n_compounds)=0;
sys(2+2+2*n_compounds)=0;
sys(3+2+2*n_compounds)=0;
sys(4+2+2*n_compounds)=0;
sys(5+2+2*n_compounds)=0;

deps=sys(length(x));
xeps=x(length(x));


%disp( ['Productivity=' num2str(rx) '  gamma=' num2str((L/gdr_reac))  '
Lim=' num2str(rx/rx_px)  '   Volume Liq=' num2str(Volume_L) '  D='
num2str(F_liq/Volume_L) ] );
save '../data/calculs_c4a.mat' rx gamma_L rx_px eps_fraction Volume_L F_liq
xeps deps rx_eps rx_act rx ro2 rco2;
% End of mdlDerivatives.



%=============================================================
% Update the discrete states
%=============================================================
function sys =
mdlUpdates(t,x,u,Temp,Press,pH,kla,gdr_reac,volume_t,frac_gas,type_reac,typ
e_bio)
sys = [];
% End of mdlUpdate.



%=============================================================
% Calculate outputs - OUtputs calculated from state vector of derivaties
%=============================================================
function sys =
mdlOutputs(t,x,u,Temp,Press,pH,kla,gdr_reac,volume_t,frac_gas,type_reac,typ
e_bio)
%-------------------------------------------------------------
% entrées  - u(i) Flux de matiere en entrée en unité masse / unité temps
%            IL FAUT LES MEME CORPS EN GAZ ET LIQUIDE (SIMPLIFIE LE
TRAITEMENT)
%            ORGANISES DANS LE MEME ORDRE
%            LES CORPS SONT DEFINIS DANS UN SCRIPT INDEPENDANT DONNANT :
%                 LEUR NOM
%                 LEUR COMPOSITION
%                 LEUR MASSE MOLAIRE
%
%          - t : temps (inusité en Régime permanent)
%          - x : vecteur d'etat
%
% Sorties sys(i) FLUX de matiere en sortie en unité masse / unité temps
%
% Parametres :  Lumiere en W/m2
%
%               Temp :  température en °C
%               Press : Pression en atm
%               pH : ph - fixé (supposé sous controle)
%               kla : en h-1; echange G/L
%               gdr_reac : grandeur caracteristique du réacteur
%                          (epaisseur ou rayon) en metre
%               volume_t : volume utile du réacteur en L
%               frac_gas : fraction de volume occupé par le gaz
%                          supposée fixe
%               type reac : 1 - rectangulaire eclairé d'un coté
%                           2 - rectangulaire eclairé de 2 cotés
%                           3 - cylindrique - eclairage radial
%
%               type_bio : 1  - 1 stoichiométrie, variable avec lumiere (TN
17.3)
%                          2  - 2 stoichiométries fixes (variabilité
globale liée à
%                          la cinétique)
%
% Auteur : Sky
%          Version 0.0.2 - Janvier 2005
%-------------------------------------------------------------
%-------------------------------------------------------------
%-------------------------------------------------------------
%
%sys = [];
%disp(' ')
%disp('Operating C IVa v. 0.0.2 ----------------------------------------
----------------------');
sys=zeros(length(x)-1,1);  % length -1 car une variable de travail en plus
par rapport a la sortie
sys([1:46],1)=x([1:46],1); % la sortie est le resultat de l'integration des
EDO (fonction mdlderivatives)
                           % x= integration ODE + update discrete
sys(47)=u(47); % Débit liquide sortie = débit liquide entrée
sys(48)=u(48); % Début gaz sortie = débit gaz entrée


if (exist('../data/calculs_c4a.mat')==2)
    load('../data/calculs_c4a.mat');
    %disp( ['FInal : Productivity=' num2str(rx) '  gamma='
num2str((L/gdr_reac))  '  Lim=' num2str(rx/rx_px)  '   Volume Liq='
num2str(Volume_L) '  D=' num2str(F_liq/Volume_L) ] );
    sys(49)=rx;
```

```
    sys(50)=gamma_L;
    if (rco2~=0)
        sys(51)=abs(ro2/rco2);
    end
    if rx_px>0
        sys(52)=rx/rx_px;
    else
        sys(52)=-999;
    end
    sys(53)=F_liq/Volume_L;
end

%-------------------------------------------------Checking output
%                                                  consistency
%for i=1:46
%    if sys(i)<0
%        sys(i)=0;
%        %disp(['C4a ERROR  --> index ' num2str(i) ' negative output'])
%    end
%end

%disp('Stop Operating C IVa v. 0.0.2 ------------------------------------
----------------------');
%==========================================================================
% End of mdlOutputs.
```

ANNEX 5 : Functions for Light radiant energy profile calculation

```
function y = wil_reacteur(x,E,F0,para_alpha,para_delta,type_reac)
%-------------------------------------------------------------------------
% L. Poughon
% Mars 2005 - Version 0.0.1
%
% Definition de la fonction dont on cherche la racine
% donnant la longueur utile eclairee (WIL) pour un reacteur
% type_reac = 1 -> rectangulaire eclairage sur une face
%             2 -> rectangulaire eclairé sur 2 faces
%             3 -> cylindrique eclairé radialement
% On defini le point de compensation à 1W/m2
%
% Entrée :
%   x : longeur à la source de lumiere (entre 0 et E) (en m)
%   E : longueur totale que traverse le flux lumineux (en m)
%   F0 : flux de lumiere (W/m2)
%   para_alpha : sqrt(Ea/(Ea+2*b*Es))
%   para_delta : sqrt(Ea*(Ea+2*b*Es))*biomasse
%   type_reac : code du reacteur (1,2,3)
%
% Sorties : y(x) = J(x)/F0 - 1/F0
%           on obtiendra WIL=x pour y(x)=0
%
% References : JF. Cornet - TN 19.2 - MELISSA Project
%              Modifié pour le Modele cyclindrique -
%                       expression par fonction de bessel
% -------------------------------------------------------------------------

switch type_reac
    case 1 % rectangulaire eclairage sur une face
        y=2.* ...
          ((1+para_alpha).*exp(-para_delta.*(x-E))-(1-
para_alpha).*exp(para_delta.*(x-E))) ...
          ./((1+para_alpha).^2.*exp(para_delta.*E)-(1-para_alpha).^2.*exp(-
para_delta.*E)) ...
          -1./F0;

    case 2 %rectangulaire eclairé sur 2 faces
        y=2.* ...
          (exp(para_delta*x)*( (para_alpha+1) + (para_alpha-1)*exp(-
para_delta*E)) ...
           + exp(-para_delta*x)*( (para_alpha-1) +
(para_alpha+1)*exp(para_delta*E)) ) ...
          ./((1+para_alpha).^2.*exp(para_delta.*E)-(1-
para_alpha).^2.*exp(-para_delta.*E)) ...
          -1./F0;

    case 3 % cylindrique eclairage radial
        y=2 .* besseli(0,para_delta*x) ...
            ./(besseli(0,para_delta*E) + para_alpha .* besseli(1,para_delta
.* E)) ...
            - 1./F0;

    otherwise % erreur
        disp('Erreur - fonction wi_l - reacteur non conforme')
end
```

```
function y = wiv_reacteur(x,E,F0,alpha,delta,K,type_reac)
%-------------------------------------------------------------------------
% L. Poughon
% Mars 2005 - version 0.0.1
%
% Fonction a integrer donnant mu sur le volume utile
% eclaire du reacteur (WIV).
% Le volume utile eclaire peut etre defini comme le
% volume pour lequel l'energie local disponible est
% superieure au point de compensation (1 W/m2 pour
% Spiruline) (cf. fonction WIL).
% Profil de l'energie lumineuse disponible en
% fonction de l'epaisseur du milieu et de la concen-
% tration en biomasse dans le reacteur.
% type_reac = 1 -> rectangulaire eclairage sur une face
%             2 -> rectangulaire eclairé sur 2 faces
%             3 -> cylindrique eclairé radialement
% On defini le point de compensation à 1W/m2
%
% Entrée :
%   x : longeur à la source de lumiere (entre 0 et E) (en m)
%   E : longueur totale que traverse le flux lumineux (en m)
%   F0 : flux de lumiere (W/m2)
%   para_alpha : sqrt(Ea/(Ea+2*b*Es))
%   para_delta : sqrt(Ea*(Ea+2*b*Es))*biomasse
%   K : constante limitation par la lumiere (corps-specifique) : W/m2
%   type_reac : code du reacteur (1,2,3)
%
% Sorties : y(x) = J(x)= I(x) *F0
%           on obtiendra WIL=x pour y(x)=0
%
% References : JF. Cornet - TN 19.2 - MELISSA Project
%              Modifié pour le Modele cyclindrique -
%                       expression par fonction de bessel
% -------------------------------------------------------------------------
```

```
switch type_reac
    case 1 % rectangulaire eclairage sur une face
        y=2.*F0.*...
            ((1+alpha).*exp(-delta.*(x-E))-(1-alpha).*exp(delta.*(x-E))./
...
            ((1+alpha).^2*exp(delta.*E)-(1-alpha).^2.*exp(-delta.*E));
        y=y./(K+y);
    case 2 %rectangulaire eclairé sur 2 faces
        y=2.*F0.*...
            (exp(delta*x)*( (alpha+1) + (alpha-1)*exp(-delta*E)) ...
                  + exp(-delta*x)*( (alpha-1) + (alpha+1)*exp(delta*E)) )
...
            ./((1+alpha).^2.*exp(delta.*E)-(1-alpha).^2.*exp(-delta.*E));
         y=y./(K+y);
    case 3 % cylindrique eclairage radial
        if (x==0)
            x=eps*10;
        end
        y=   2.*F0.* besseli(0,delta.*x) ...
            ./  (besseli(0,delta.* E) + alpha .* besseli(1,delta .* E));
         y=2.*pi .*x .* y./(K+y);
    otherwise % erreur
        disp('Erreur - fonction wiv - reacteur non conforme')
end
```

ANNEX 6 : Functions for compounds defintion

```
% LOAD_COMPOUNDS : script
%   version 1.0
%   sky Janv 2005
%
%
% FICHIER COMMUN A TOUS LES MODELES : IDENTIFIE LE NOMBRE ET LA LISTE DES
% CORPS EN JEU ET LEUR PROPRIETEES
%   1) eau
%   2) nh3 et ses formes ioniques (distribution fonction du pH)
%   3) h2so4 et ses formes ioniques (distribution fonction du pH)
%   4) h3po4 et ses formes ioniques (distribution fonction du pH)
%   5) hno3 et ses formes ioniques (distribution fonction du pH)
%   6) hno2 et ses formes ioniques (distribution fonction du pH)
%   7) urée
%   8) o2
%   9) co2 ses formes ioniques (distribution fonction du pH)
%   10) acetate
%   11) propionate
%   12) butyrate (iso assumé equivalent)
%   13) valerate (iso assumé equivalent)
%   14) caproate (iso assumé equivalent)
%   15) n2
%   16) h2
%   17) ch4
%   18) Gaz inerte dissous
%   19) matiere_organique_mixée (flux normalisé)
%   20) biomasse nitrosomonas
%   21) biomasse nitrobacter
%   22) biomasse rhodobacter
%   23) biomasse spiruline

%NOMBRE DE CORPS
n_compounds=23;


% noms_c(i) MATRICE DES NOMS - dans l'ordre des flux de matiere
% i_xxxx INIDICES(i) des corps dans les flux de matieres entre les elements
% M_C Matrice de compostion CHONSP des corps - composition en colonne

i_eau=1;   % eau
noms_c(i_eau)={'EAU'};
M_C(:,i_eau)=[0
          2
          1
          0
          0
          0];

i_nh3=2;   % nh3 et ses formes ioniques (distribution fonction du pH)
noms_c(i_nh3)={'NH3'};
M_C(:,i_nh3)=[  0
                3
                0
                1
                0
                0];

i_h2so4=3; % h2so4 et ses formes ioniques (distribution fonction du pH)
noms_c(i_h2so4)={'H2SO4'};
M_C(:,i_h2so4)=[  0
                2
                4
                0
                1
                0];

i_h3po4=4; % h3po4 et ses formes ioniques (distribution fonction du pH)
noms_c(i_h3po4)={'H3PO4'};
M_C(:,i_h3po4)=[  0
                3
                4
                0
                0
                1];

i_hno3=5;  % hno3 et ses formes ioniques (distribution fonction du pH)
noms_c(i_hno3)={'HNO3'};
M_C(:,i_hno3)=[  0
                1
                3
                1
                0
                0];

i_hno2=6;  % hno2 et ses formes ioniques (distribution fonction du pH)
noms_c(i_hno2)={'HNO2'};
M_C(:,i_hno2)=[  0
                1
                2
                1
                0
                0];

i_uree=7;  % urée
noms_c(i_uree)={'UREA'};
```

```
M_C(:,i_uree)=[  1
                 4
                 1
                 2
                 0
                 0];

i_o2=8;    % o2
noms_c(i_o2)={'O2'};
M_C(:,i_o2)=[  0
                 0
                 2
                 0
                 0
                 0];

i_co2=9;   % co2 ses formes ioniques (distribution fonction du pH)
noms_c(i_co2)={'CO2'};
M_C(:,i_co2)=[  1
                 0
                 2
                 0
                 0
                 0];

i_acetate=10; %acetate
noms_c(i_acetate)={'ACETATE'};
M_C(:,i_acetate)=[  2
                 4
                 2
                 0
                 0
                 0];

i_prop=11;  % propionate
noms_c(i_prop)={'PROPIONATE'};
M_C(:,i_prop)=[  3
                 6
                 2
                 0
                 0
                 0];

i_buty=12;  % butyrate (iso assumé equivalent)
noms_c(i_buty)={'BUTYRATE'};
M_C(:,i_buty)=[  4
                 8
                 2
                 0
                 0
                 0];

i_val=13;   % valerate (iso assumé equivalent)
noms_c(i_val)={'CAPROATE'};
M_C(:,i_val)=[  5
                10
                 2
                 0
                 0
                 0];

i_capro=14; % caproate (iso assumé equivalent)
noms_c(i_capro)={'CAPROATE'};
M_C(:,i_capro)=[  6
                12
                 2
                 0
                 0
                 0];

i_n2=15;    % n2
noms_c(i_n2)={'N2'};
M_C(:,i_n2)=[  0
                 0
                 0
                 2
                 0
                 0];

i_h2=16;    % h2
noms_c(i_h2)={'H2'};
M_C(:,i_h2)=[  0
                 2
                 0
                 0
                 0
                 0];

i_ch4=17;   % ch4
noms_c(i_ch4)={'HNO2'};
M_C(:,i_ch4)=[  1
                 4
                 0
                 0
                 0
                 0];

i_inert=18; % Gaz inerte - Assimilé N2 (formule obligatoire ou pb de masse
molaire)
noms_c(i_inert)={'INERT GAS'};
```

```
M_C(:,i_inert)=[  0
                  0
                  0
                  2
                  0
                  0];

i_om=19;     % matiere_organique_mixée (flux normalisé)
noms_c(i_om)={'O.M. Std'};
M_C(:,i_om)=[  0
                  0
                  0
                  0
                  0
                  0];

i_ns=20;     % biomasse nitrosomonas
noms_c(i_ns)={'Ns. Sp.'};
M_C(:,i_ns)=[  0
                  0
                  0
                  0
                  0
                  0];

i_nb=21;     % biomasse nitrobacter
noms_c(i_nb)={'Nb. Sp.'};
M_C(:,i_nb)=[  0
                  0
                  0
                  0
                  0
                  0];

i_rh=22;     % biomasse rhodobacter
noms_c(i_rh)={'Rh. Sp.'};
M_C(:,i_rh)=[  0
                  0
                  0
                  0
                  0
                  0];

i_sp=23;     % biomasse spiruline
noms_c(i_sp)={'Sp. Sp.'};
M_C(:,i_sp)=[  0
                  0
                  0
                  0
                  0
                  0];
```

```
% LOAD_PARTITIONS : script
%   version 1.0
%   sky Janv 2005
%
%  Pre-requis :
%       1/ le script LOAD_COMPOUNDS doit avoir été executé
%       2/ les indices et tailles des matrices doivent etre compatibles
%       avec le script LOAD_COMPOUNDS
%
%  Execution :  valeurs de coefficients de partages entre gaz et
%  liquide pour tous les corps données par ki(i_xxx)=yi/xi.
%  Liquide uniquement :  ki=0
%  Gaz uniquement :      ki=1e10
ki=zeros(n_compounds,1);


if ( (exist('Temp','var')~=0) && (exist('Press','var')~=0) )
    ki(i_eau)=exp((18.3036-3816.44/(-46.13+Temp+273.15)))/(760*Press);
    ki(i_o2)=4200; % equilibre pour O2
    ki(i_co2)=2000; % CO2
    ki(i_inert)=1e10; %
    ki(i_h2)=1e10; %
    ki(i_n2)=1e10; %
    ki(i_ch4)=1e10; %
else
      disp('Warning : No Temperature or Pressure defined - G/L partition
coefficient failed');
      disp('LOAD_PARTITION function error');
end
```

ANNEX 6 : S-Function of the control box (control_v2

```
function [sys,x0,str,ts] = control(t,x,u,flag,consign ,maxi_act
,mini_act,period )
%=============================================================================
% Fonction control simple : S-fonction pour simulink
%
%   Sky June-2005
%   Version 0.0.1
%
%   1// mdlOutputs : calul de la valeur de la variable d'action
%
%   2// mdlDerivatives : Néant
%
%   3// mdlUpdates : Néant
%
%   Parametres entrant :
%               consign = valeur de la consigne (comparaison avec mesure)
%               maxi_act= valeur maximale pour l'action (variable
manipulée)
%               mini_act= valeur minimale pour l'action (variable
manipulée)
%               period = periodicité des mises a jour de la verification
%               mesure/consigne
%
%
%
%=============================================================================
%   Entrées :
%
%   Sorties : 3

%  Vecteur d'etat (travail et integration) : 3

%=============================================================
switch flag
  case 0
    [sys,x0,str,ts] = mdlInitializeSizes(consign,maxi_act,mini_act,period);
% Initialization

  case 1
    sys = mdlDerivatives(t,x,u,consign,maxi_act,mini_act); % Calculate
derivatives

  case 2
    sys = mdlUpdates(t,x,u,consign,maxi_act,mini_act); % Update discrete
states

  case 3
    sys = mdlOutputs(t,x,u,consign,maxi_act,mini_act); % Calculate outputs

  case {4, 9} % Unused flags
    sys = [];

  otherwise
    error(['unhandled flag = ',num2str(flag)]); % Error handling
end
% End of dsfunc.



%=============================================================================
% Initialization
%=============================================================================
function [sys,x0,str,ts] =
mdlInitializeSizes(consign,maxi_act,mini_act,period)
% Call simsizes for a sizes structure, fill it in, and convert it
% to a sizes array.



%---- Initialisation des tailles des matrices
    sizes = simsizes;
    sizes.NumContStates  = 0; % Pour simulation continue (mldderivaties)
    sizes.NumDiscStates  = 1;
    sizes.NumOutputs     = 1; % Sortie de la fonction
    sizes.NumInputs      = 1 ; % Entrée de la fonction
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);

% Initialize the discrete states.

    x0 = [mini_act];

    %--- Check period consistency
    if period<=0
```

```
        period=0.1;
    end

    str = [];   % Set str to an empty matrix.
    ts  = [period, 0];    % sample time: [period, offset] - pour continu
tout a 0

%=============================================================================
% End of mdlInitializeSizes.
%=============================================================================



%=============================================================================
% mdlDerivatives
%=============================================================================
function sys = mdlDerivatives(t,x,u,consign,maxi_act,mini_act)
%-------------------------------------------------------------
% Entrée :  t - valeur du temps
%           u - flux de matiére en entrée (taille 2*nombre de corps)
%           x : vecteur d'etat correspondant à u pour l'etat du reacteur
%
% Parametres :
%
% Sorties : sys ==  derivée du vecteur d'etat
%-------------------------------------------------------------------
sys = [];
% End of mdlDerivaties.



%=============================================================
% Update the discrete states
%=============================================================
function sys = mdlUpdates(t,x,u,consign,maxi_act,mini_act)
if u(1)>consign
    sys(1)=mini_act;
else
    sys(1)=maxi_act;
end

% End of mdlUpdate.
```

```
%=============================================================
% Calculate outputs
%=============================================================
function sys = mdlOutputs(t,x,u,consign,maxi_act,mini_act)
%-------------------------------------------------------------
% entrées  - u : flux en entrée = aucun
%           - t : temps (inusité en Régime permanent)
%           - x : vecteur d'etat
%
% Sorties sys(i)
%
% Parametres :
%               consign = valeur de la consigne (comparaison avec mesure)
%               maxi_act= valeur maximale pour l'action (variable
manipulée)
%               mini_act= valeur minimale pour l'action (variable
%               manipulée)
%
% Auteur : Sky
%          Version 0.0.1 - June 2005
%-------------------------------------------------------------
sys=x;



%=====================================================================
% End of mdlOutputs.
```