FILTREX Manual

Version 3.0 (R2015a)

A. Bouvier, J.P. Gauchi,
INRA, UR1404, F78352 Jouy-en-Josas, France

January 29, 2015

Contents

Chapter 1

Overwiew

1.1 Introduction

1.1.1 General framework

Many dynamic systems in different fields such as the life sciences, industry, economics and many others (see [Cappé *et al.* (2005)] for an overview) can be modelled by stochastic non-linear state-space models, i.e., hidden Markov chain models. Such systems are observed at any time t through a set of output variables y_t, while their dynamics are characterised by a vector of unobserved state variables x_t that evolves according to a given Markov transition probability distribution function often arising from an autoregressive state model. The output variables themselves evolve according to a given probability distribution function conditioned by the state variables and some parameters.
Several objectives can be considered on the basis of the knowledge of the system model and assumptions about all its random components, as well as on the basis of the successive output values. The first one is the identification of the system, i.e., the estimation at every time t of the probability distribution function of the state variables conditional on the observed values of the output variables up to time t, or when it exists, the estimation of its probability density function (pdf) $p_t(x|y_1, \ldots, y_t)$, as well as the estimation of all the unknown static model parameters θ. Inference about the model parameters or functional components of the model is often also to be considered, as is the preliminary question of statistical model comparison and state model choice. Such a dynamic model can be a valuable tool for prediction, in which case another question of interest would be the estimation at time t of the anticipated pdf of the state variables k-step ahead, $p_{t+k}(x|y_1, \ldots, y_t)$, especially when observed co-variables u_t are present in the state model. Finally, one can want to use some of these co-variables to control the evolution of the state variables according to some tracking or optimal control objectives.

1.1.2 Filtering approach

Filtering is the standard approach to the problem of estimation of the conditional probability distribution function of the state variables and that of the unknown parameters ([Bain and Crisan (2009)]). When the system is linear optimal estimators are provided by the Kalman filter (KF). In the nonlinear case, the extended Kalman filter (EKF) is often used by engineers but without definitive theoretical support and with frequent practical drawbacks because of the local validity of the corresponding approach that relies on successive model linearisations (see [Jazwinski (1970)] for more details about the KF and the EKF, and [Chen (1993)] for some improvements of the EKF). Global approximation methods have also been developed, with validity in the full state-space. Some rely on analytical approaches such as the Gaussian Sum Method ([Sorenson and Alspach (1971)]; [Šimandl and Královec (2000)]), whereas others rely on more accurate numerical approaches that approximate the state-space by systems of discrete points. Among them are the Point Mass method ([Kramer and Sorenson (1988)]; [Šimandl *et al.* (2006)] with its orthonormal grids. However, the most well-known in this second class of global numerical approaches are the sequential Monte Carlo methods, which have benefited from theoretical results (see, for example, the review of [Liu (2001)]). These so-called particle approaches first gave rise to filters based on sequential importance sampling (SIS) ([Akashi *et al.* (1975)]; [Davis (1981)]; [Kitagawa (1987)]), and then to SIS-R filters that include a resampling step to improve convergence, such as the well-known Bootstrap Filter ([Gordon *et al.* (1993)]) and the Interacting Particle Filter ([Del Moral *et al.* (1992)]; [Del Moral (1998)]; [Del Moral *et al.* (2001)]) (see [Doucet *et al.* (2001)] for a review). However, the discrete nature of the probability distribution approximations provided by the SIS and SIS-R filters ([Hürzeler and Künsch (1998)]), combined with some degree of inability of these filters to deal with small observation noise, did not completely eliminate long time divergence problems. Some state variable distribution regularisations were then introduced into the SIS-R algorithm ([Oudjane (2000)]; [Warnes (2001)]; [Musso *et al.* (2001)]) to provide a distribution approximation under the form of a probability density function. A convergent Regularized Interacting Particle Filter using convolution kernels was thus proposed by [Oudjane (2000)]) and [LeGland and Oudjane (2004)]. However, with the notable exception of [Del Moral and Jacod (2001)] who performed a regularisation of the output variable distribution, all the previous filters rely on the analytical knowledge of the probability distribution function of the output variables conditional on the state variables and the parameters, and on the tractability of the observation likelihood function, a requirement that reduced the practical range of these filters (it is worth saying however, that when one can obtain unbiased estimates of the resampling particle weights used in these filters, asymptotic exact inference may be possible. See [Doucet and Rosset (2006)] or [Fearnhead *et al.* (2008)])

1.1.3 Convolution particle filtering

To eliminate this limitation, a new generation of convergent particle filters has been proposed based on convolution kernel density estimation and on implicit regularisation of both state and output variable distribution estimates ([Rossi (2004)]; [Rossi and Vila (2005)]; [Rossi and Vila (2006)]; [Campillo and Rossi (2009)]. This nonparametric filtering or convolution particle filtering approach, thus makes it possible to deal with the frequent situation in which both state and output variable distributions are analytically unknown: one only needs to be able to simulate these variables at each time t. Interestingly, this convolution particle filtering can be compared with some recent Sequential Monte Carlo Approximate Bayesian Computation filtering algorithms ([Del Moral *et al.* (2012)]). These SMC-ABC algorithms are similar to special cases of this nonparametric filtering, through particular choices of the kernel functions involved ([Jasra *et al.* (2012)]). Moreover, this nonparametric particle filtering approach can be very efficiently combined with statistical methods that depend on the availability of these state and output variable distributions, to restore these methods when these distributions are not available. For example, nonparametric filtering has been used to consistently estimate the Bayes factor between two competing state-space models ([Vila and Saley (2009)] and Section A.5 for technical details) and to perform CUSUM parameter change detection tests ([Verdier *et al.* (2008)]) in these same models. Finally, contrary to the other filtering approaches, convolution particle filtering can itself be easily optimised by making it possible to determine a sequence of observation times of the output variables for a given experimental cost, that favours the convergence of the successive parameter estimates to their true values ([Gauchi and Vila (2011)], [Gauchi and Vila (2013)]).

1.2 Discrete time dynamic state-space model

1.2.1 A general model structure

We will consider dynamic systems that are assumed to obey hidden Markov chain models of the following form:

$$\left\{ \begin{array}{rcl} x_t & \backsim & Q_t(.|x_{t-1}, \theta) \\ y_t & \backsim & G_t(.|x_t, \theta) \end{array} \right. \tag{1.1}$$

in which $x_t \in I\!R^d$ and $y_t \in I\!R^s$ are vectors of unobserved state variables and observed output variables, respectively, and $\theta \in \Theta \subset I\!R^p$ is a vector of p unknown static parameters with given prior density p_0^θ. Q_t is a Markov transition probability distribution function with density q_t, often arising from a nonlinear autoregressive state model, $x_t = f_t(x_{t-1}, \theta, \varepsilon_t)$, in which ε_t is a vector of independent random variables (possibly noises) and f_t is a known Borel measurable function. G_t is an absolutely continuous probability distribution function with density g_t. Both f_t and G_t can be time-varying. The distribution G_t, the transition distribution Q_t (or the distribution of ε_t) are not necessarily known but can at least be

simulated. In some simpler situations, the output equation can also be given by a regression equation $y_t = r_t(x_t, \theta, \eta_t)$, in which r_t is a known Borel measurable function and η_t is a vector of random variables (possibly noises) that can at least be simulated.

1.2.2 A microbiological growth model

This very general type of state-space model can explain many dynamic behaviours encountered in real life such as, for example, pathogenic bacteria growth in a food medium.
One of the most well-known models in microbiology is the Baranyi-Roberts model ([Baranyi and Roberts (1995)]) whose discrete version (Euler scheme) is given in closed form ([Gauchi *et al.* (2009)]) by

$$x_{t+1} = \delta\ x_0\ \exp(\mu_{max}\ A_t)\ \frac{1}{B_t}\ (\mu_{max}\ \frac{dA_t}{dt} - \frac{dB_t}{dt}\ \frac{1}{B_t}) + x_t + \varphi_t \tag{1.2}$$

with $A_t = t + \frac{1}{\mu_{max}} \ln(\exp(-\mu_{max}\ t) + \exp(-\mu_{max}\ \lambda) - exp(-\mu_{max}\ t - \mu_{max}\ \lambda))$
and $B_t = 1 + \frac{\exp(\ \mu_{max}\ A_t)-1}{\frac{x_{max}}{x_0}}$
where:

- x_t is the state variable, i.e., the number of bacteria in the medium at time t.
- μ_{max}(maximum growth speed), λ (mean latency time), x_0 (minimum number of bacteria and x_{max} (maximum number of bacteria), form the parameter vector θ to be estimated.
- ϕ_t is a centered random Poisson variable and δ is a discretisation step.

The observed variable y_t is the number of bacteria colony-forming units (CFU) in a Petri dish (culture medium) from bacteria sampled at time t. Its probability distribution function $G_t(.|x_t, \theta)$ is the result of the interaction of several independent random phenomena: the spatial sampling in the primary medium at time t, a given number of successive samplings in several dilution tubes (with Poisson or aggregative spatial distribution assumptions), the successive volume sampling errors and dilution errors (assumed to be Gaussian) and, finally, the log-normal error counts in the Petri dishes (where sampled bacteria develop and form colonies on the culture medium). The corresponding probability distribution function G_t cannot be analytically characterised, but can be easily simulated.

Chapter 2

FILTREX Software

2.1 Presentation

FILTREX is a software for parametric identication, models comparison, and optimal sampling of experiments for complex microbiological dynamic systems by nonlinear filtering.

Parametric identification. This parametric identification concerns microbiological dynamic systems, based on primary models (growth or thermal inactivation models). It is build by implementing the new nonlinear particle technique using a convolution kernel approach mentioned at Section 1.1.3. Let us just recall here that for this efficient particle filtering procedure, the only a priori information needed for the parameters is their respective possible variation ranges. The coding of this functionality in FILTREX has been developped from an open source code of the convolution particle filter ([Choquet and Rossi (2005)]).

Several growth and inactivation models are provided in FILTREX and instructions are given in the Developer Guide (appendix B) to add more. The CV (coefficients of variation) can be estimated in addition to the model parameters. Default values are provided for the variation ranges of the parameters and check is made for their valid bounds. Some of them can be fixed to given values.

Dynamic comparison of two models with the Bayes factor. This second functionality computes the so-called Bayes Factor, for deciding which of two models better fits a given set of data (see Section 1.1.3). This Bayes Factor is the ratio of the respective marginal likelihood functions of the two competing models. It is not a genuine statistical test but it has been proved to be one of the best indices for comparing two nonlinear models. Its particle estimation in FILTREX does not need the knowledge of the model likelihoods as required by the usual statistical selection procedures (e.g. Akaïke criterion). FILTREX computes the Bayes factor all along the observation times.

Simulation of an optimal sequential sampling. Three technics are proposed:
(i) Construction of a Sobol-Saltelli method based approach at starting time of the

dynamics ([Saltelli (2002)],
(ii) Construction of a D-optimal design ([Donev and Atkinson (1988)]) at starting time of the dynamics using the Tornsey algorithm ([Droesbeke *et al.* (1997)]),
(iii) A more sophisticated and powerful technics: an on-line approach based on the SIVIP method ([Gauchi and Vila (2013)], [Gauchi and Vila (2011)]).

FILTREX simulates an optimal sampling of experiments times, for a given model, provided its parameter values (D-optimal design) or parameter range (Sobol-Saltelli and SIVIP methods), and a simulated observation dataset (SIVIP method).

Simulation of an observation dataset. Observation datasets can be simulated from a model and its parameters values, with constant or unconstant variance and uniform or normal sampling. The generated dataset can be used in the other tasks, in particular, in task `Simulation of an optimal sequential sampling` with SIVIP method.

2.2 Authors and contributors

FILTREX is developed in INRA, UR1404, F78352 Jouy-en-Josas, France. (INRA: Institut National de la Recherche Agronomique).
Authors and contributors are:

- *Project coordinator*
  J-P. Gauchi (INRA/UR1404-Jouy-en-Josas, France).

- *Scientific advisors*
  J-P. Vila (INRA/MISTEA-Montpellier, France),
  J-P. Gauchi (INRA/UR1404-Jouy-en-Josas, France),
  P. Del Moral (INRIA/Bordeaux University, France)

- *Main contributors to the source code (alphabetic order)*
  C. Bidot (INRA/UR1404-Jouy-en-Josas, France)
  A. Bouvier (INRA/UR1404-Jouy-en-Josas, France)
  R. Choquet (CNRS/CEFE, Montpellier, France)
  V. Rossi (PhD student 2002-2004, Montpellier University/INRA-ENSAM, France)

- *Secondary contributors to the source code (alphabetic order)*
  E. Atlijani (Technical trainee, 2009, INRA/MIA-Jouy-en-Josas, France)
  E. Maillot (Technical trainee, 2008, INRA/MISTEA-Montpellier, France)

2.3 Download

- It is a free Matlab software, under license GPL[1] >= 3.

[1] http://www.gnu.org/licenses/gpl-3.0.txt

- FILTREX has been tested with Matlab2012a and Matlab2013a. A C-compiler is needed for task `Simulation of an optimal sampling` unless the compiled version of FILTREX is used (see below).
- A compiled version is in progress on several platforms, which makes uncessary the need of Matlab software and C-compiler.
- Download from Web site :
  http://www3.jouy.inra.fr/miaj/public/logiciels/filtrex/welcome.html

Chapter 3

FILTREX Use: Getting started

3.1 Start FILTREX

Before the very first use, **you must compile the C-programs by running the Matlab**[1] **file** `compil.m`[2]. Then, and at all the other times, run the file `FILTREX.m`: a menu opens which offers the different tasks:

- `Parametric identification`
- `Dynamic comparison of two models with the Bayes factor`
- `Simulation of an optimal sequential sampling`
- `Simulation of an observation dataset`
- `What is FILTREX` (this gives some general information about the FILTREX package).

3.2 Observation dataset

Most of the FILTREX tasks require an observation dataset that should be provided in a file format `xls` or `csv`[3]. See the files in the folder EXAMPLES/REAL_DATA as patterns. Note that the time unit (day, hour or minute) has to be coded inside the label of the first column between parenthesis. Only the first character is taken into account. Valid values are: 'd', 'h' or 'm'. It is case insensitive.
Examples of valid labels:

```
"(d)" "time (h)" " t (mns)" "temps (m)"
```

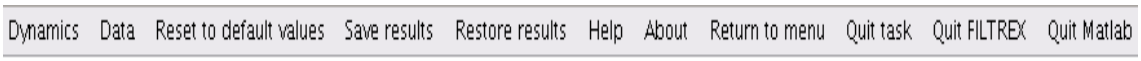
[1] http://www.mathworks.fr/

[2] Compilation is only required for running the task `Simulation of an optimal sampling` and when the compiled version of FILTREX is not used.

[3] Formats `xls` and `csv` are available in Microsoft Excel and OpenOffice

3.3 General layout

When a task is selected, a window opens, vertically divided into 2 parts: the left one is for input, the right one is for output. Each is made up of one or several pages or "panels". You switch from a panel to the following or preceding one by clicking the + or − token at the bottom of the pages.
The input panel is topped by a menu bar, whose most items are identical in all the tasks:



Dynamics. This button unfolds in several items:
`Model equation` opens a menu with the available models. Select the model to be studied.
`Help about model equations` opens a similar menu. Display PDF files showing the equation and general shape of the models.
`Fix/unfix parameters` and `Set parameters to the maximal range`. These two items appear only when a model is selected. Open a menu with the parameter names. Select the ones to fix/unfix or set to their maximal valid bounds.

Data. Browse your folders to select the observation dataset file (see Section 3.2).

Reset to default values. Whenever it is possible, default values are proposed for the input fields. This button resets them.

Save results. Backup FILTREX runs into a Matlab file.

Restore project. Restore the panels from a backup file. After a restoration, execution can be launched again, possibly in modifying some input.

Help. Display some pratical hints.

About. Display some general information about `FILTREX` package.

Chapter 4

Parametric identification

4.1 Input

The steps to enter a study are the following ones :

1. **select an observation dataset** by using the button `Data` of the top bar (see Section 3.3).

   When the observation dataset is chosen, some of its characteristics are displayed (see frame 3 in Fig. 4.1). Press the icon  to make a graphical window open with the plot of observations versus time.

2. **select a model** by using the button `Dynamics` of the top bar (see Section 3.3).

   You can try different parameter ranges, or, by using the submenu `Dynamics` of the top bar, set them to their maximal bounds and reset them to their default values.

3. **fill in the `TIME STEP` box** (see frame 1, Fig. 4.1) by the computational time step. The `proposition` is the least of the greatest common divisor between consecutive times in your observation dataset. The time step should be a divisor of this value because there should be an observation at each computational time.
   Note that smaller the time step is and larger is the maximal time in the observation dataset, the more there are computational times and more the run time is long.

4. **fix the other options**.

   All the other boxes of the input panel contain default values that can be modified at will within valid bounds.

   - `Seed` is the random seed. If its value is equal to `clock`, seed is the current time. Note that in this case, successive execution would not produce same results.
   - The coefficients of variation (CV) can be estimated by selecting `Estimate CV`. Some of them can then be fixed to given values by the top bar button `Dynamics -> Fix/unfix parameters`.

- The other options (see frame 2, Fig. 4.1) will be explained in Sections 4.4 and 4.3.

5. **launch the execution**. Execution is launched by a click on button `GO` at the bottom of the input panel. The button `GO` becomes then a `STOP` button. Click on it to stop the run.

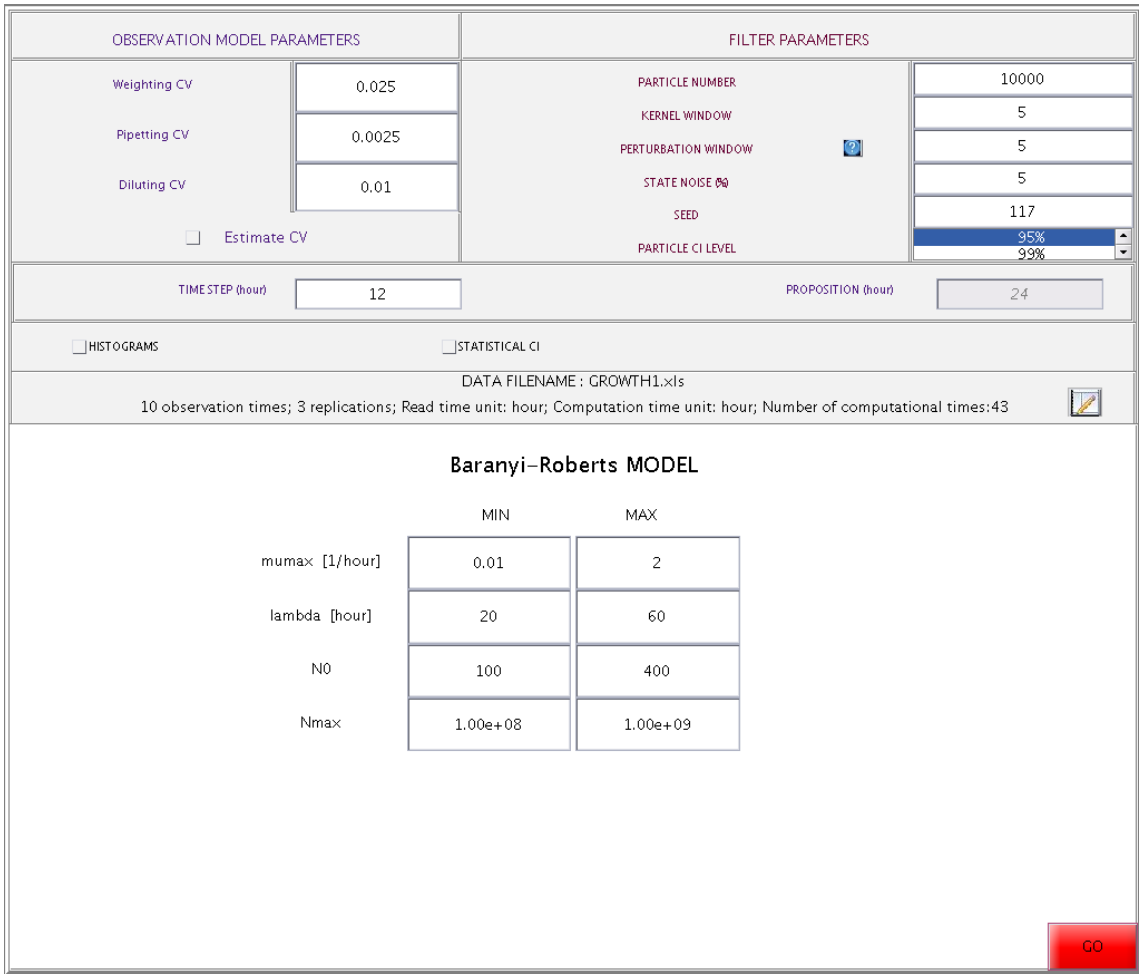


Figure 4.1: Input panel in task `Parametric identification`

4.2 Output

The results of the task `Parametric identification` are displayed on several panels. Panel number 1 displays the plot of parameter estimation and its confidence intervals versus observations.
Note that a click anywhere on a FILTREX plot opens a graphical Matlab window, you can save, modify or export.

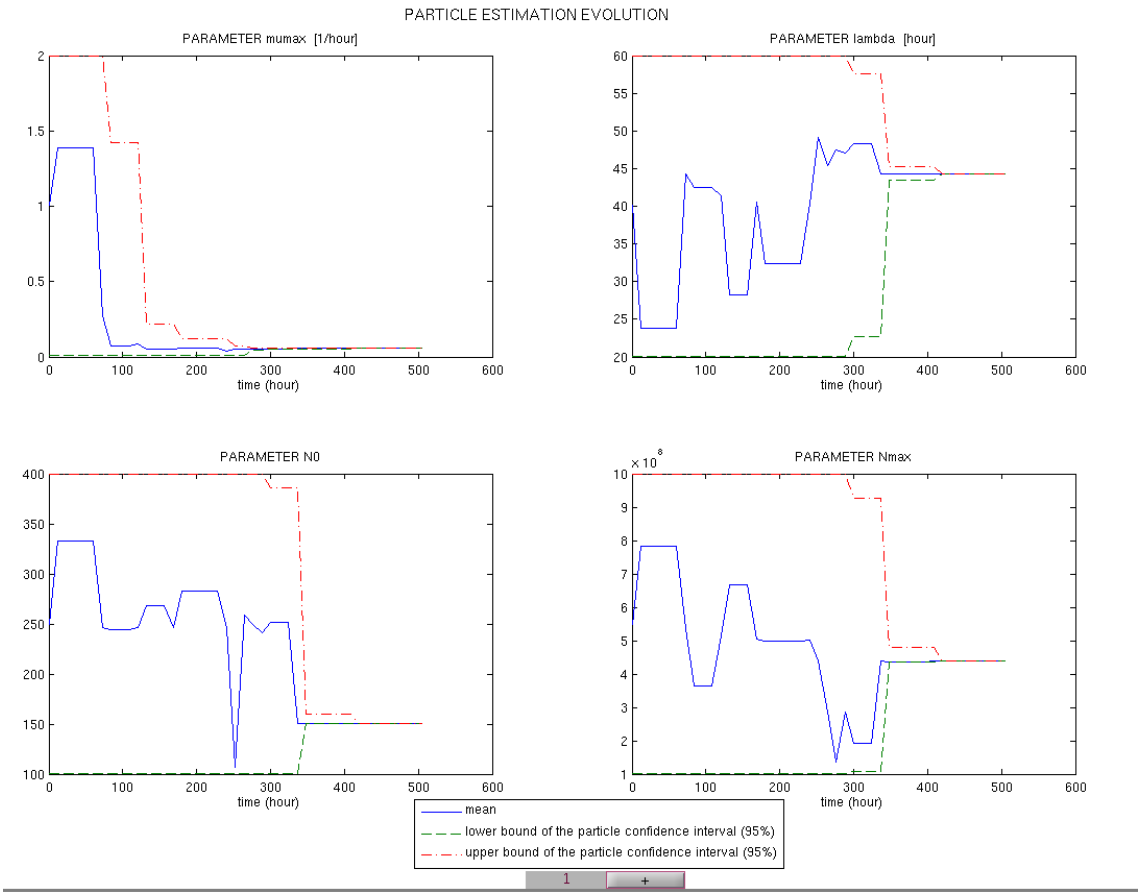


Figure 4.2: Output panel number 1 in task `Parametric identification`

Panel number 2 plots the data observations versus time and two kinds of results : first, the final filtering estimated dynamics, and secondly, the stepwise estimation of the state value and its confidence intervals.

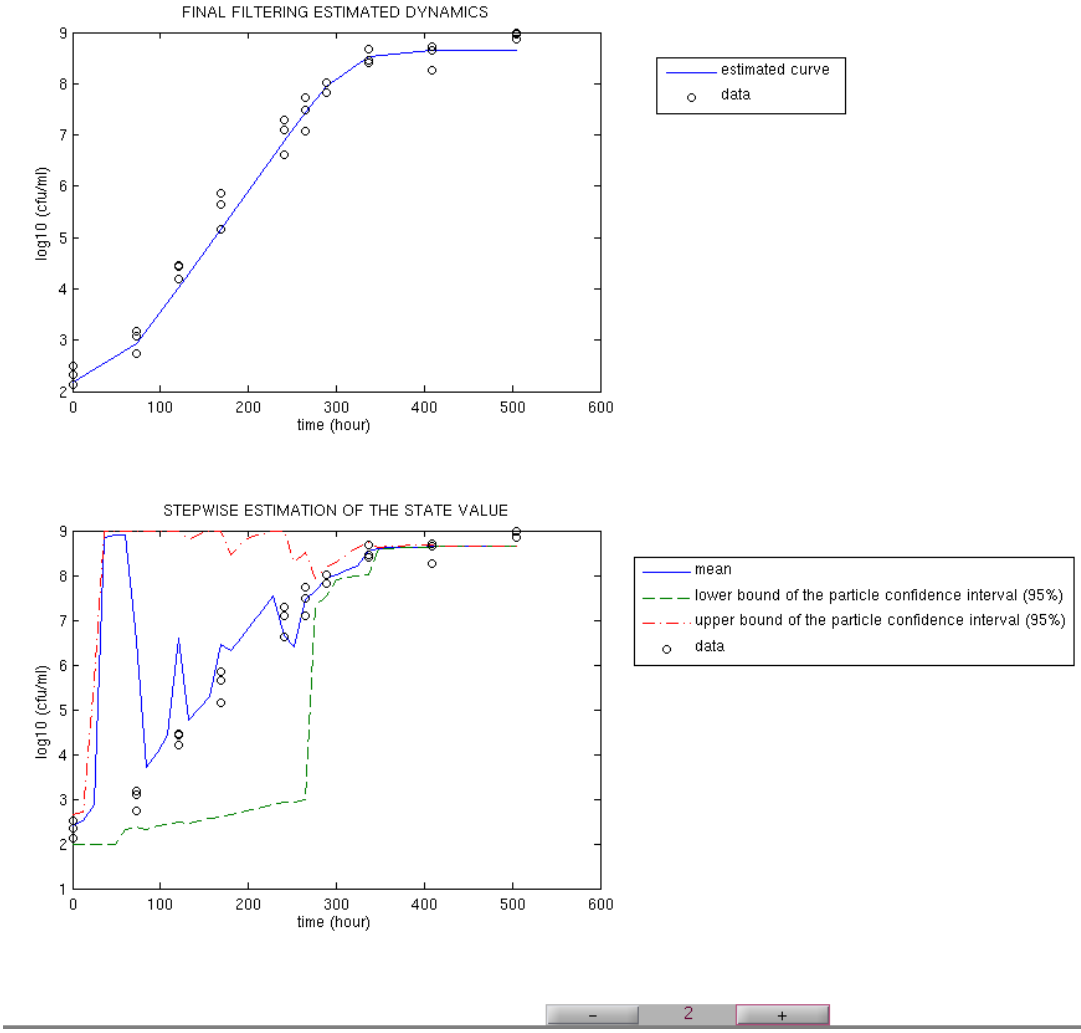
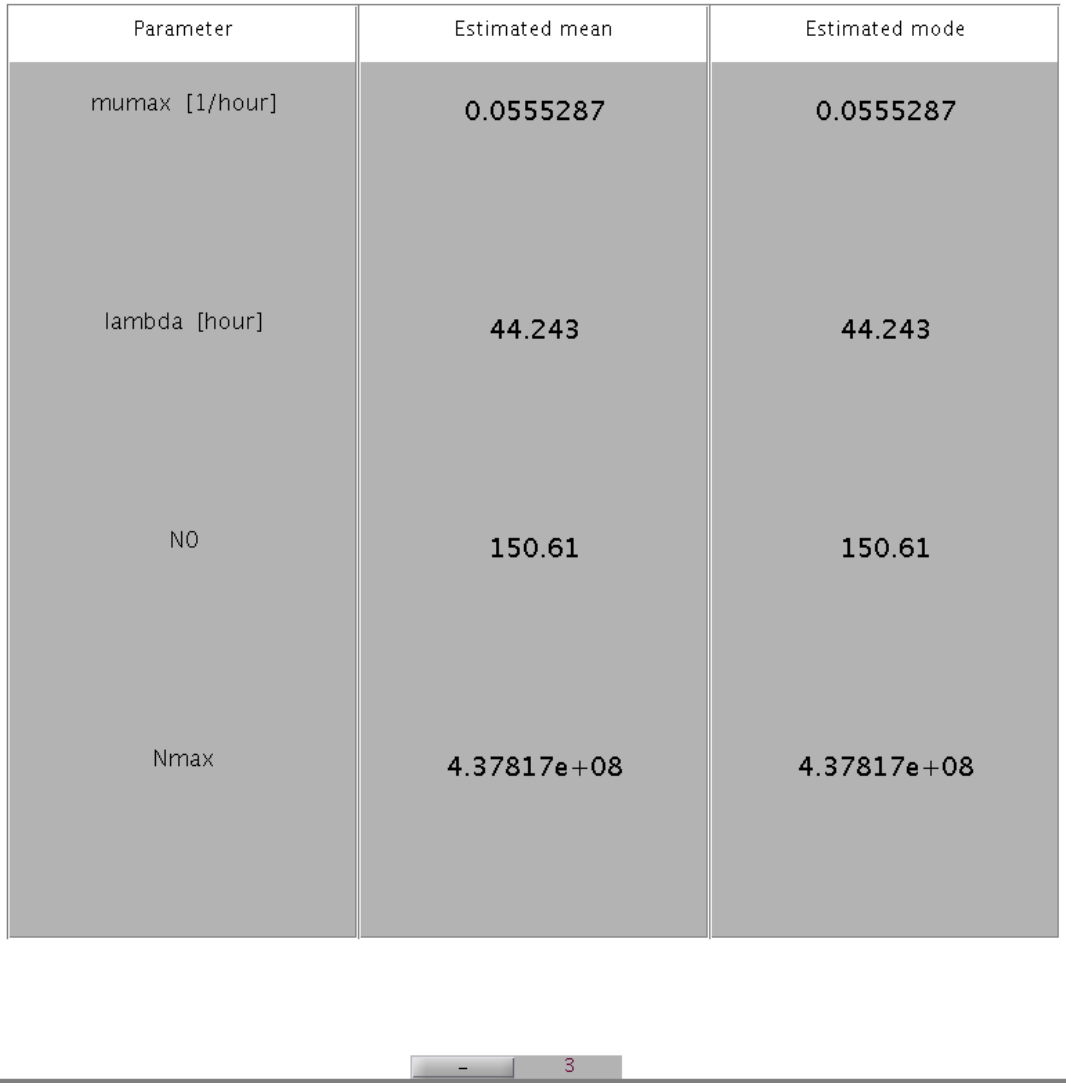


Figure 4.3: Output panel number 2 in task `Parametric identification`

Panel number 3 displays the mean and mode of the estimated values of the parameters.

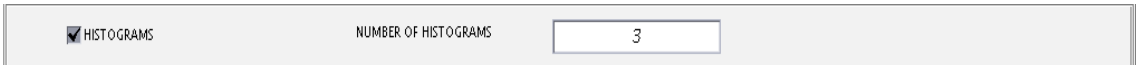


Parameter	Estimated mean	Estimated mode
mumax [1/hour]	0.0555287	0.0555287
lambda [hour]	44.243	44.243
N0	150.61	150.61
Nmax	4.37817e+08	4.37817e+08

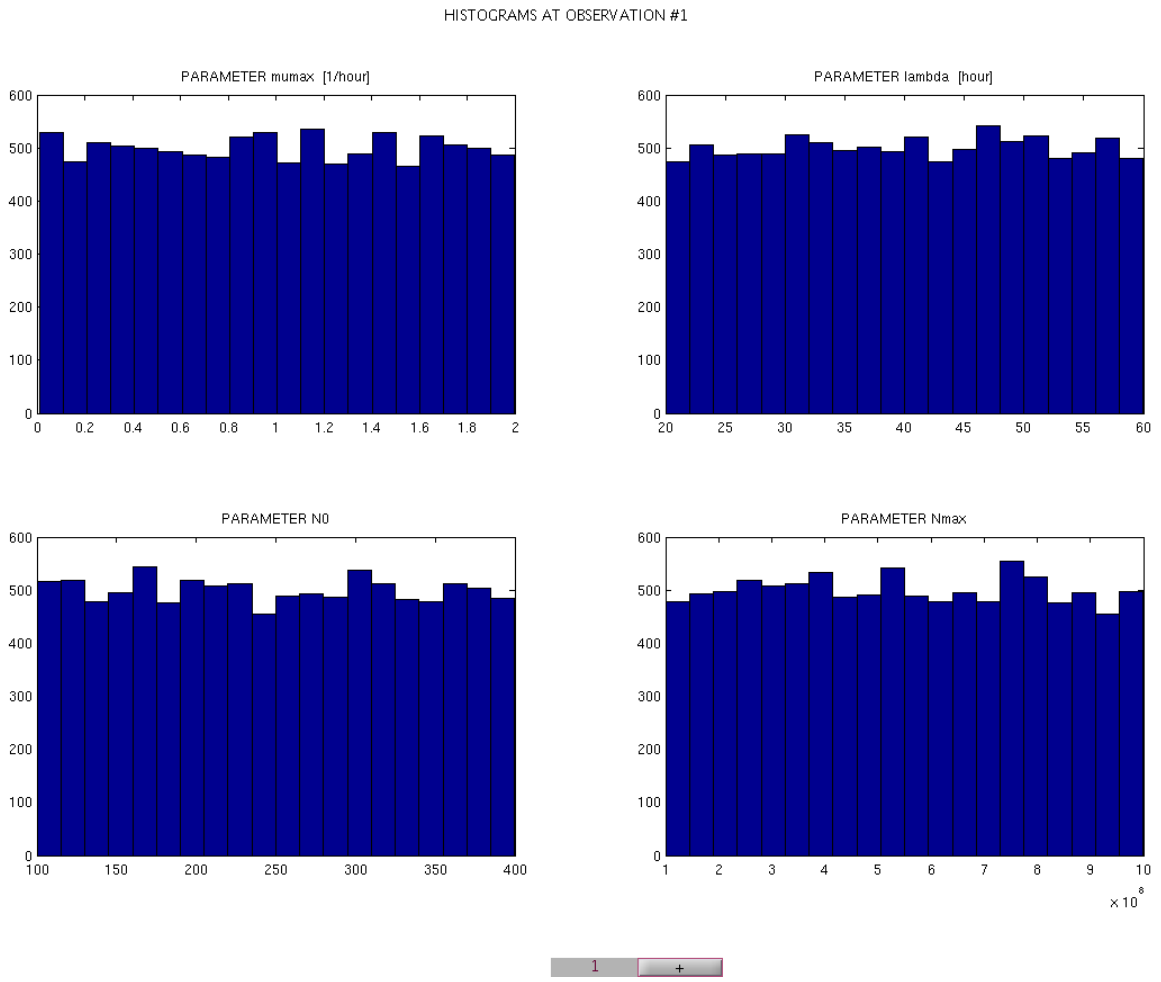
Figure 4.4: Output panel number 3 in task `Parametric identification`

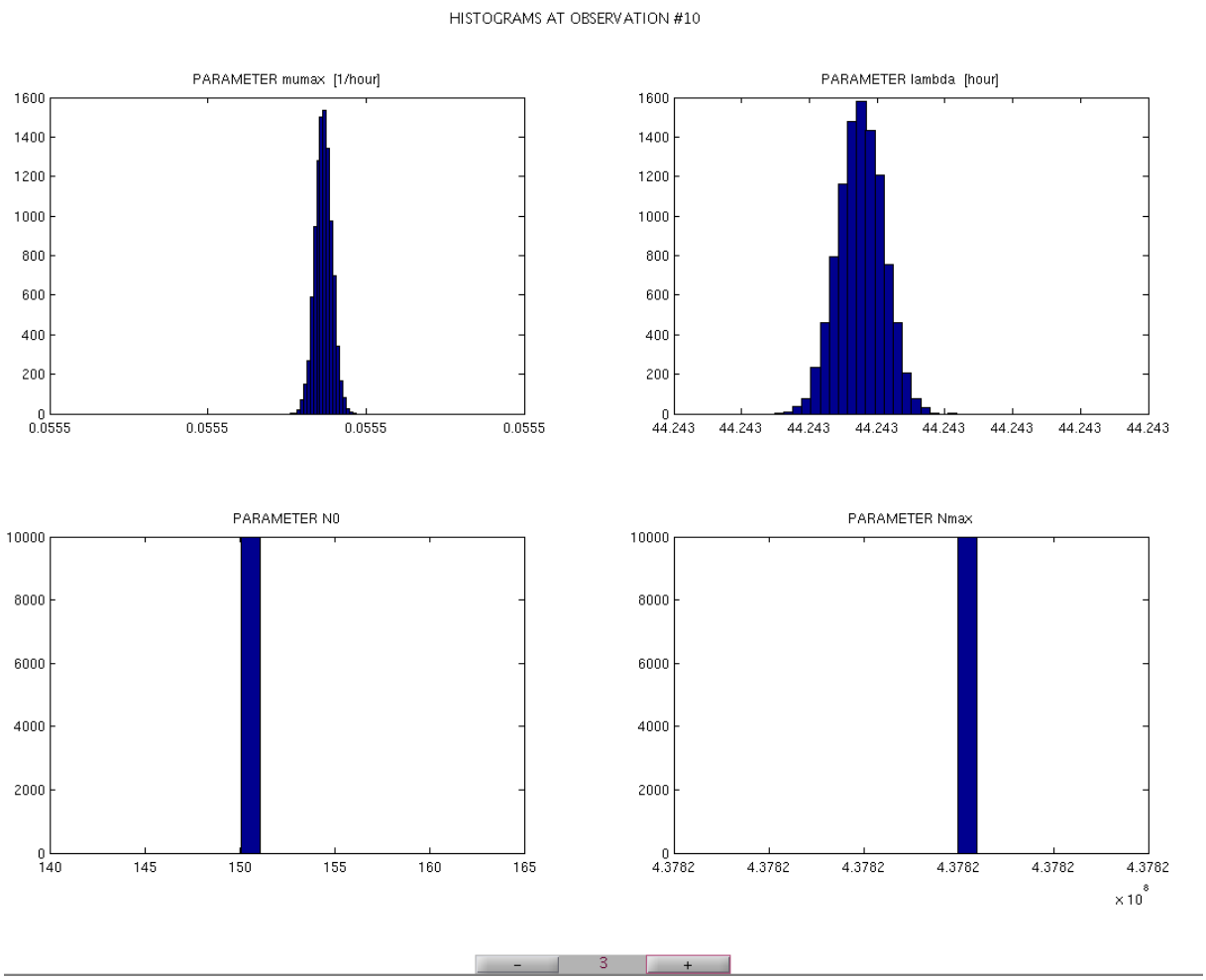
4.3 Histograms

The option `histograms` plots histograms for each observation and each parameter, of the `n` estimated values of the parameters, `n` stating for the number of particles.
When the option `histograms` is selected, frame 2 of the input panel (Fig. 4.1) is:



`Number of histograms` is the number of observations for which histograms of all parameters are plotted. Its upper limit is 20. If there are more observations, histograms will be plotted for the 4/5 first observations and for the 1/5 last ones.
Parameter histograms are the first output. Some examples:

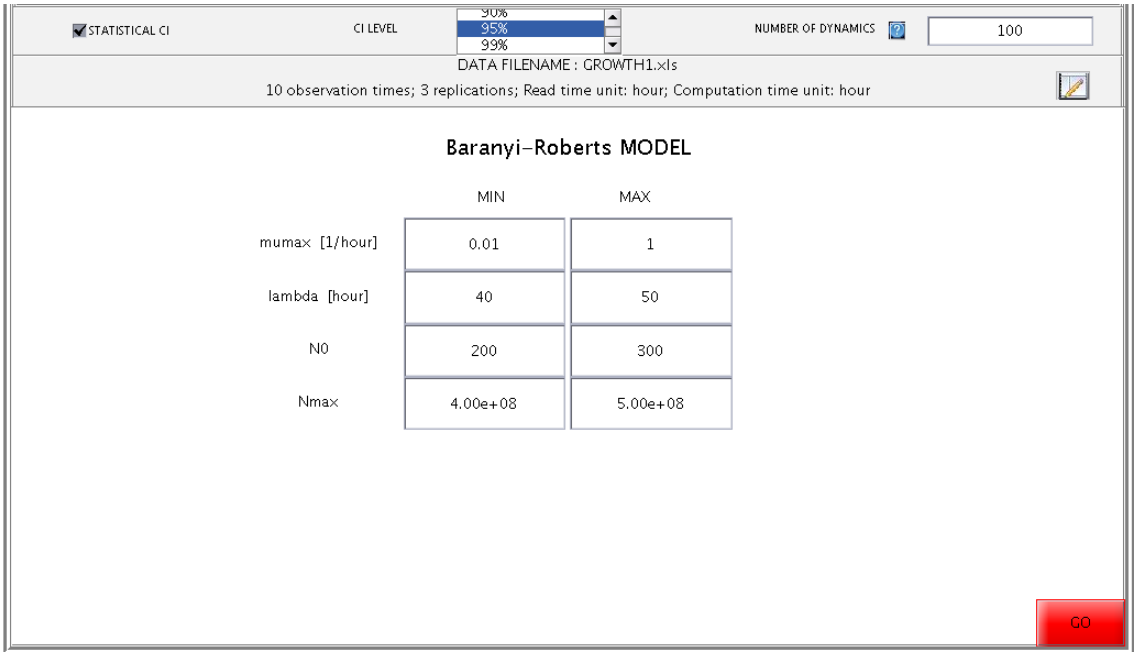




The following panels are similar to those in general case (Fig. 4.2, Fig. 4.3, Fig. 4.4).

4.4 Statistical confidence intervals calculation

The option `statistical CI` computes statistical confidence intervals of the estimated values of the parameters. When this option is selected, frame 2 and following of the input panel (Fig. 4.1) are:



Three CI levels are offered: 90%, 95%, 99 %
The first output panel displays histograms of the estimated values of the parameters. Note that histograms are only plotted when the number of dynamics is greater or equal to 10. The next panel displays numerical results: minimum, maximum, mean, standard deviation and statistical confidence intervals at the selected CI level of the estimated values of the parameters.

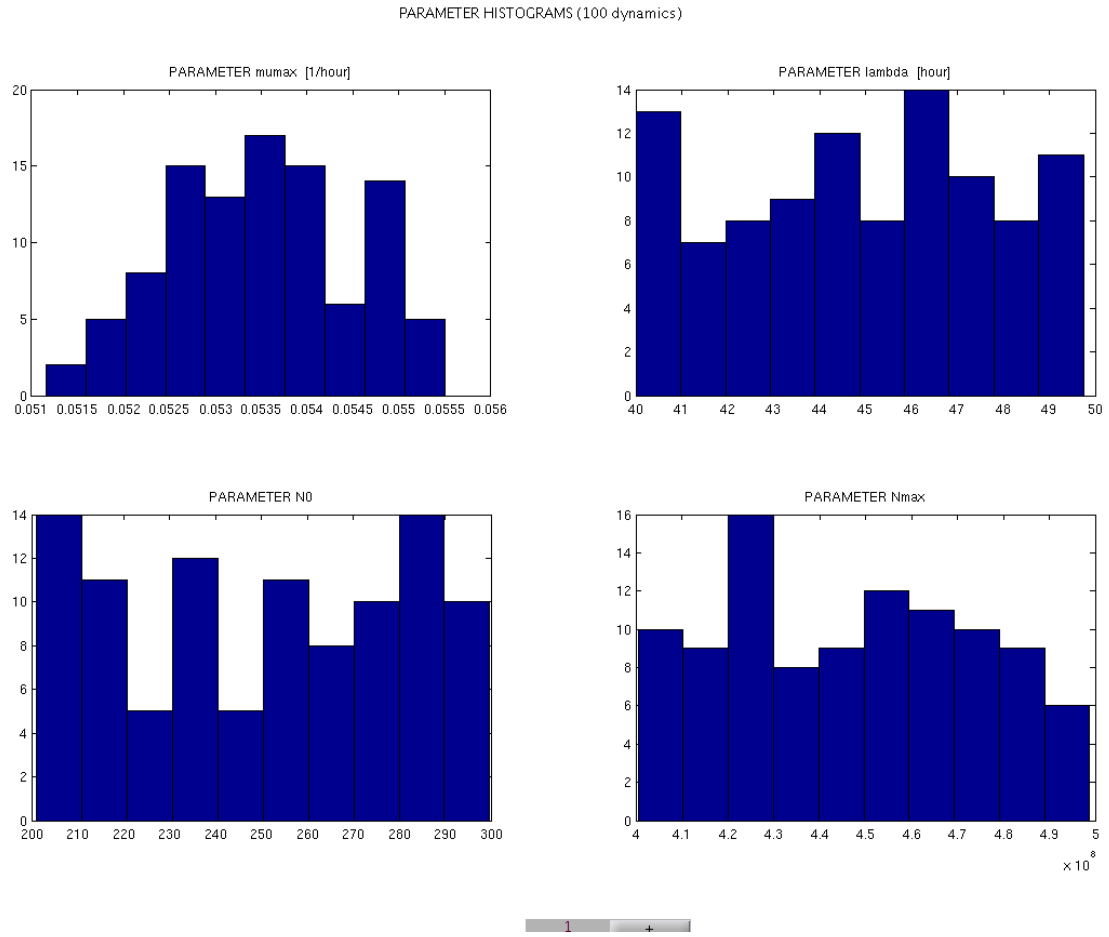


Figure 4.5: First output panel in task `Parametric identification` when CI are required

Parameters	Min	Max	Mean	STD	Statistical CI (95%)
mumax [1/hour]	0.05116	0.05549	0.05351	0.0009894	[0.05334;0.05367]
lambda [hour]	40.01	49.75	44.92	2.862	[44.45;45.4]
N0	201.1	299.7	250.4	30.6	[245.3;255.4]
Nmax	4.005e+08	4.988e+08	4.476e+08	2.682e+07	[4.431e+08;4.52e+08]

Figure 4.6: Numerical results in task `Parametric identification` when CI are required

Chapter 5

Dynamic comparison of two models with the Bayes factor

5.1 Input

The input panel looks like the one of the task `Parametric identification` (Fig. 4.1), but the top bar is slightly different:

Two models should be selected: the model used as numerator in the Bayes formulae, and the model used as denominator.
The following example compares models Rosso ([Rosso (1995)]) and Baranyi-Roberts ([Baranyi and Roberts (1995)]). The first input panel is relative to the numerator model (Fig. 5.1) and the second one to the denominator model (Fig. 5.2). CV, filter parameters and data characteristics are unique for both models.

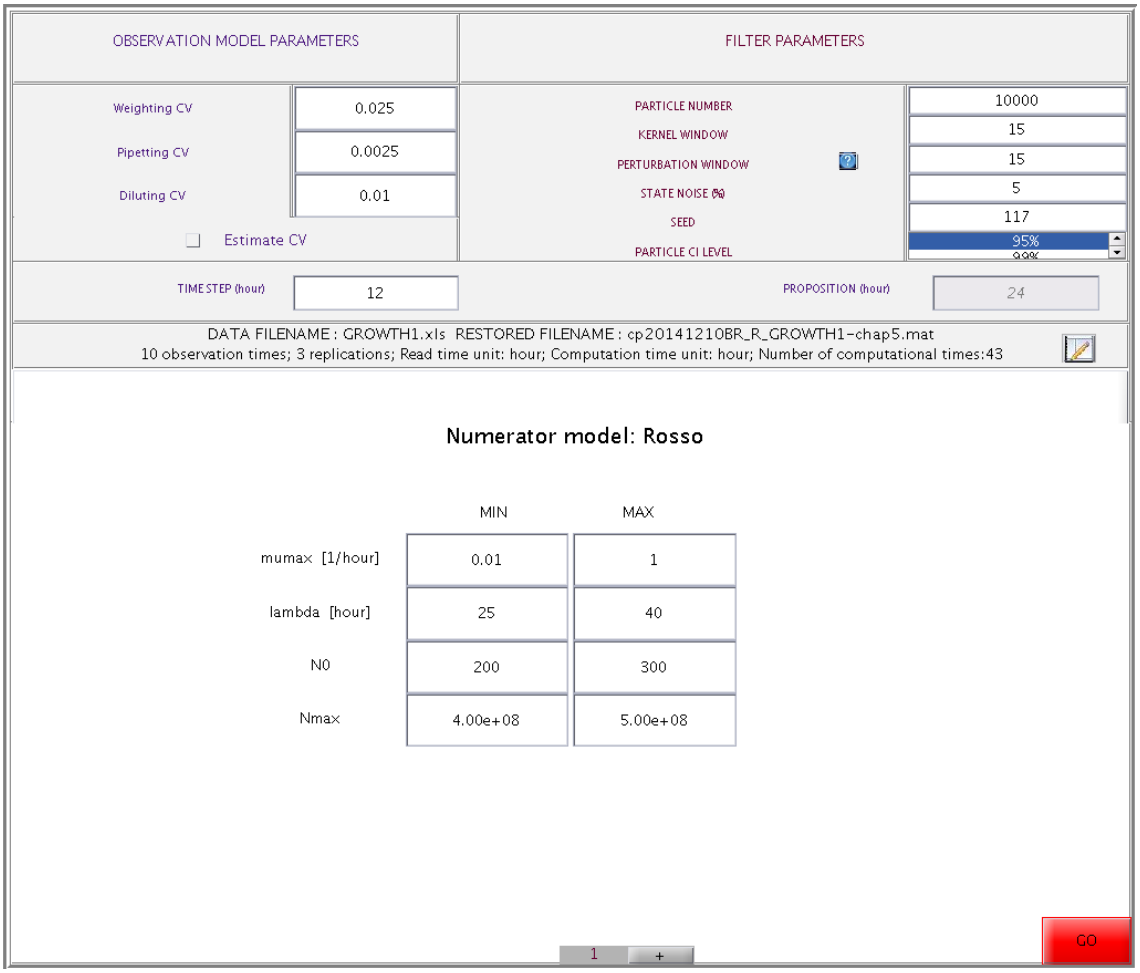


Figure 5.1: Input panel number 1 in task `Dynamic comparison of two models`.

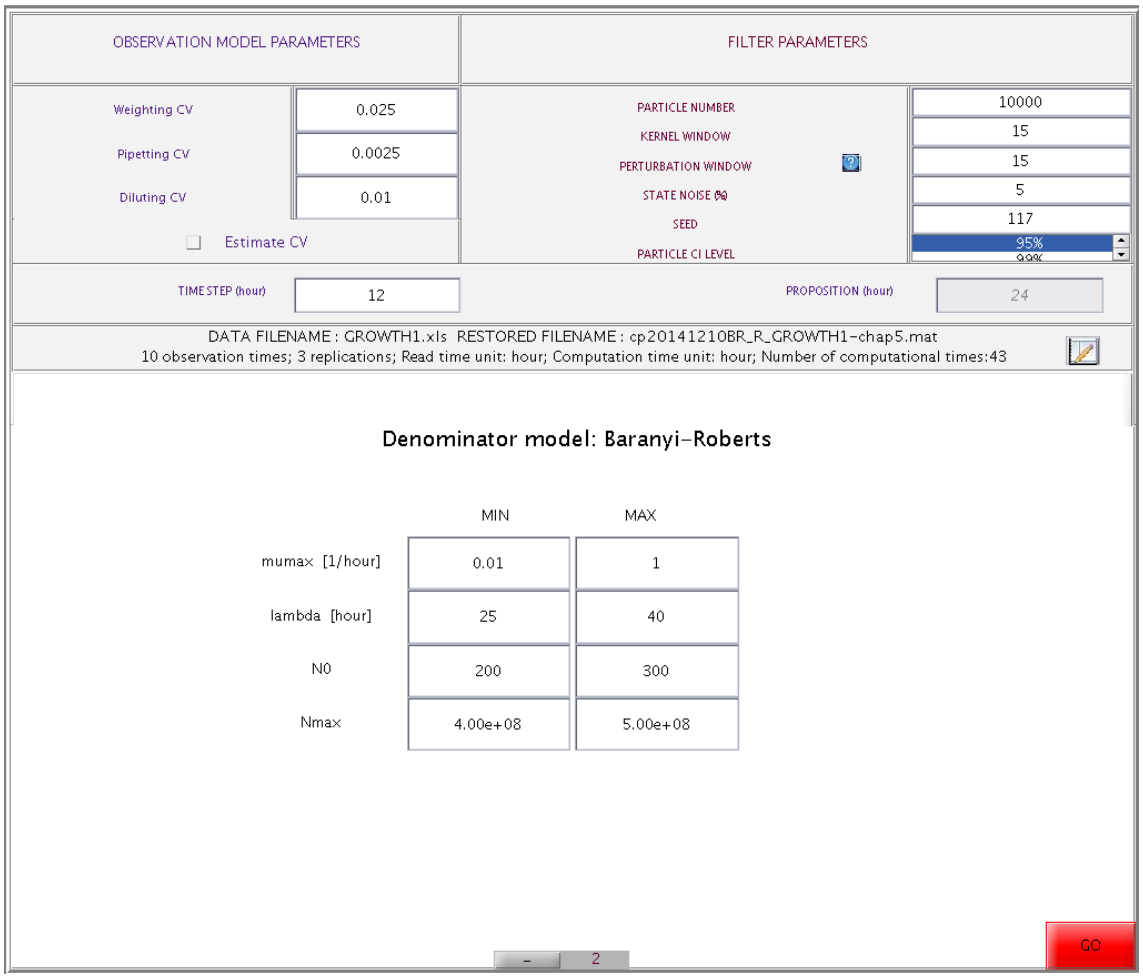


Figure 5.2: Input panel number 2 in task `Dynamic comparison of two models`.

5.2 Output

- The output panels number 1 to 6 display the results of parametric identification successively by each model (same output as in task `Parametric identification`, see Section 4.2).
- Panel number 7 displays, side by side for each model, the mean of the estimated values of the parameters (Fig. 5.3).
- The following panel is the plot of the parameter estimation, for both models (Fig. 5.4).
- Panel number 9 displays the plots of the final filtering estimated dynamics and the stepwise estimation of the state value versus time, for both models (Fig. 5.5).
- Panel number 10 displays the plot of the Bayes factor value versus time (Fig. 5.6).
- Last panel is the numerical results: marginal likelihoods and BF (Fig. 5.7).

Note: restoration of model comparison results does not restore panels 1 to 6.

Both models: ESTIMATED RESULTS

Parameter	Estimated mean for the numerator model	Estimated mean for the denominator model
mumax [1/hour]	0.0463566	0.0507455
lambda [hour]	34.1554	31.495
N0	283.265	284.801
Nmax	4.22443e+08	4.09401e+08

Figure 5.3: Output panel number 7 in task `Dynamic comparison of two models`.

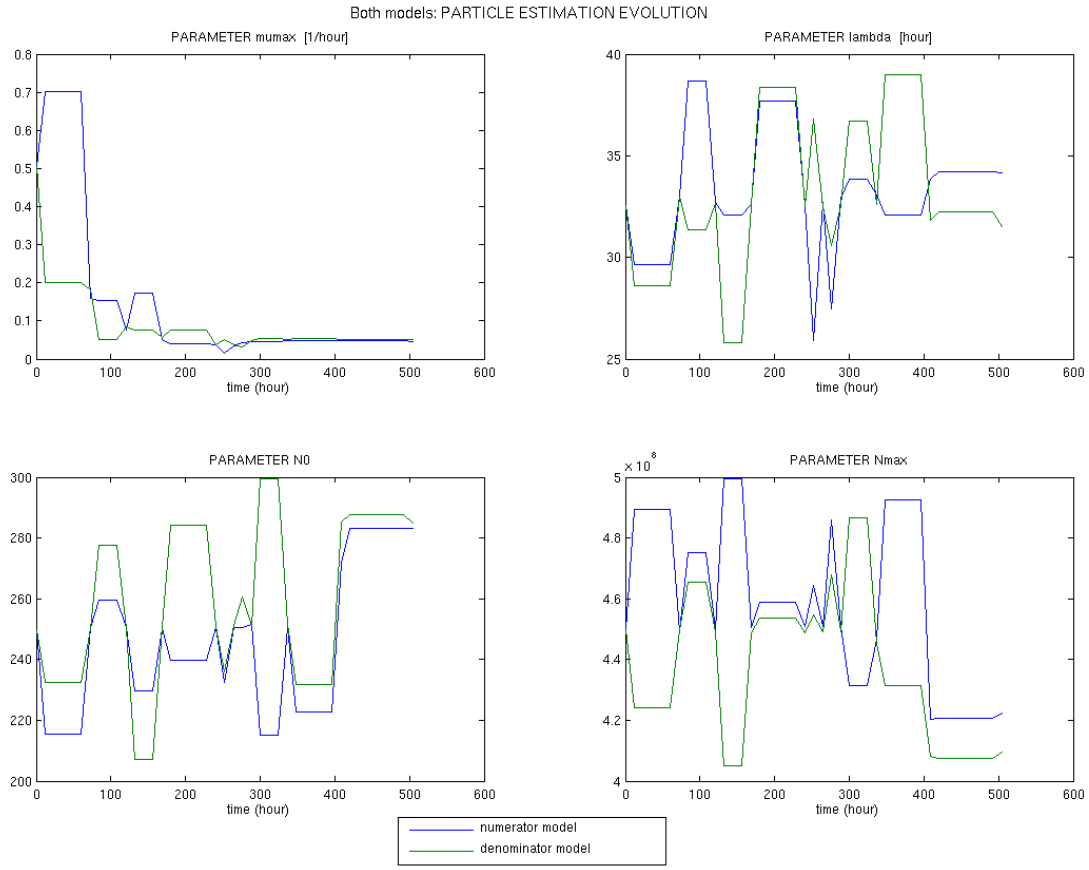


Figure 5.4: Output panel number 8 in task `Dynamic comparison of two models`.

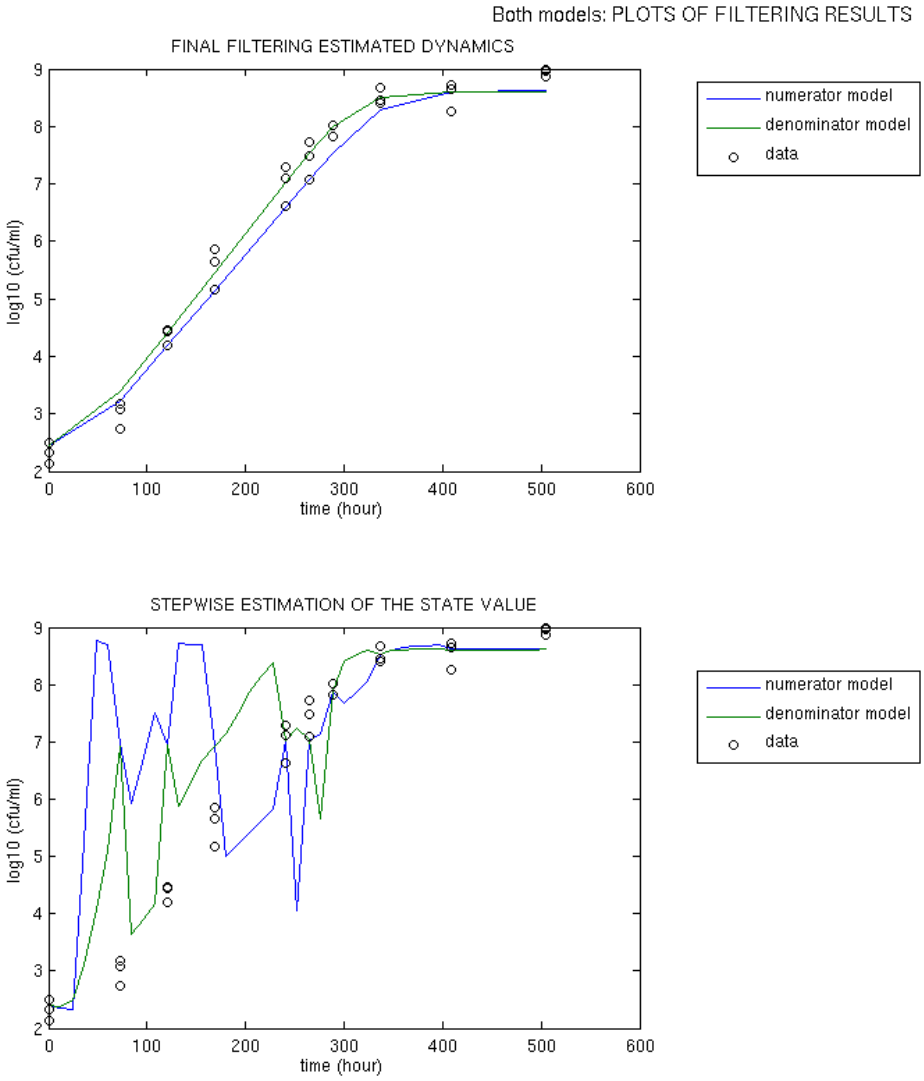


Figure 5.5: Output panel number 9 in task `Dynamic comparison of two models`.

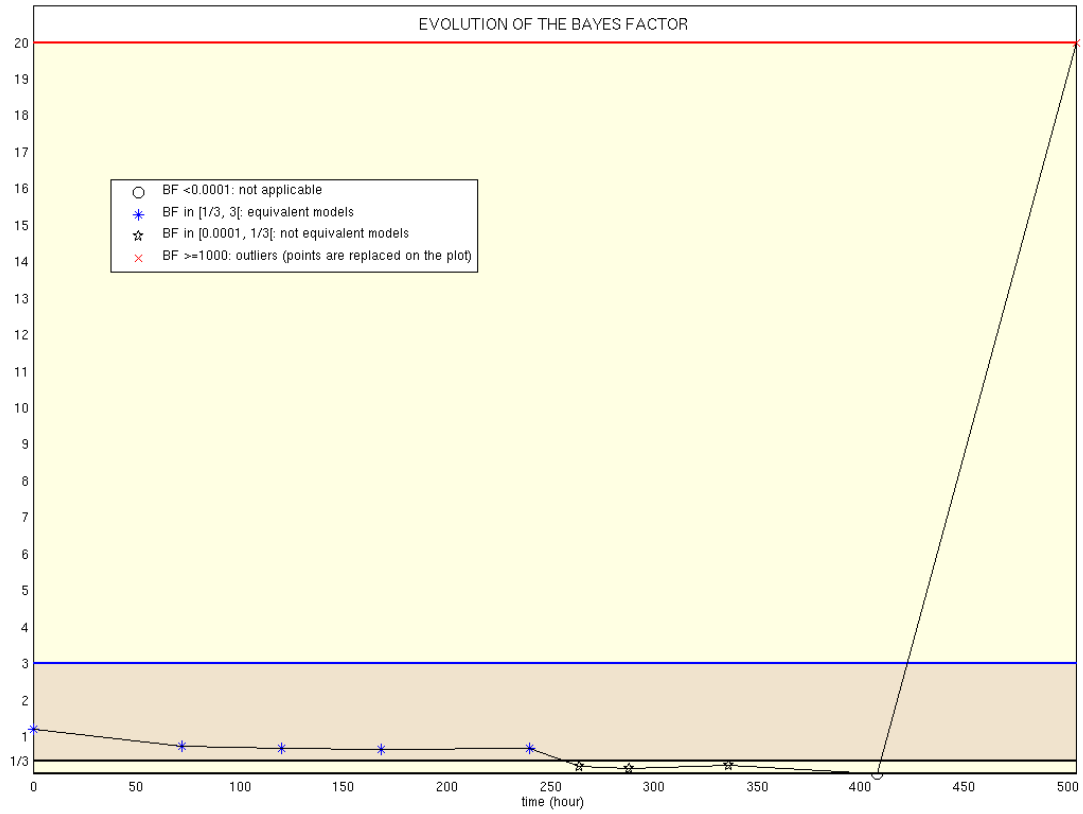


Figure 5.6: Bayes factor plot in task `Dynamic comparison of two models`.

Up to time 240, both models can be considered as equivalent: as shown by the plot of the final filtering estimated dynamics (first graph of Fig. 5.5), both the estimated curves fit the observations more or less as well. But from this time, the denominator model fits the observations better than the numerator model. Furthermore, the preceding differences are cumulated. The models are no more considered as equivalent: BF is less than $1/3$[1] (it would be greater than 3[1] if the numerator model fitted better than the denominator model). At the two last times, BF cannot be calculated because of numerical reasons.

[1]The BF ranges are given in [Kass and Raftery (1995)].

EVOLUTION OF THE MARGINAL LIKELIHOODS AND EVOLUTION OF THE BF

Time (hour)	Numerator marginal likelihood	Denominator marginal likelihood	BF
0	17.5079	14.6825	1.192
72	4.2091e-17	6.8403e-17	0.7337
120	1.267e-13	1.3876e-13	0.67
168	2.726e-10	2.8534e-10	0.6401
240	0.00029845	0.00028257	0.676
264	0.00012069	0.00040395	0.202
288	0.041836	0.063885	0.1323
336	0.00021258	0.00013079	0.215
408	6.2784e-30	2.084e-25	6.477e-06
504	1.4291e-59	1.7483e-86	6.477e+14

Figure 5.7: Output panel number 11 in task`Dynamic comparison of two models`.

Chapter 6

Simulation of an optimal sequential sampling

The three methods implemented in FILTREX (Sobol-Saltelli, D-optimal, SIVIP, see Section 2.1) can be used in a complementary or alternative way. The final output is a simulated optimal sampling of experiments times. These times can be saved in a file for later use, for example, to provide the simulation times in the task `Simulation of an observation dataset` (see Section 7).
.

6.1 Method Sobol-Saltelli

This method is preferred if only parameter ranges are known.

6.1.1 Input

Fig. 6.1 is an example of input panel with Baranyi-Roberts model selected.

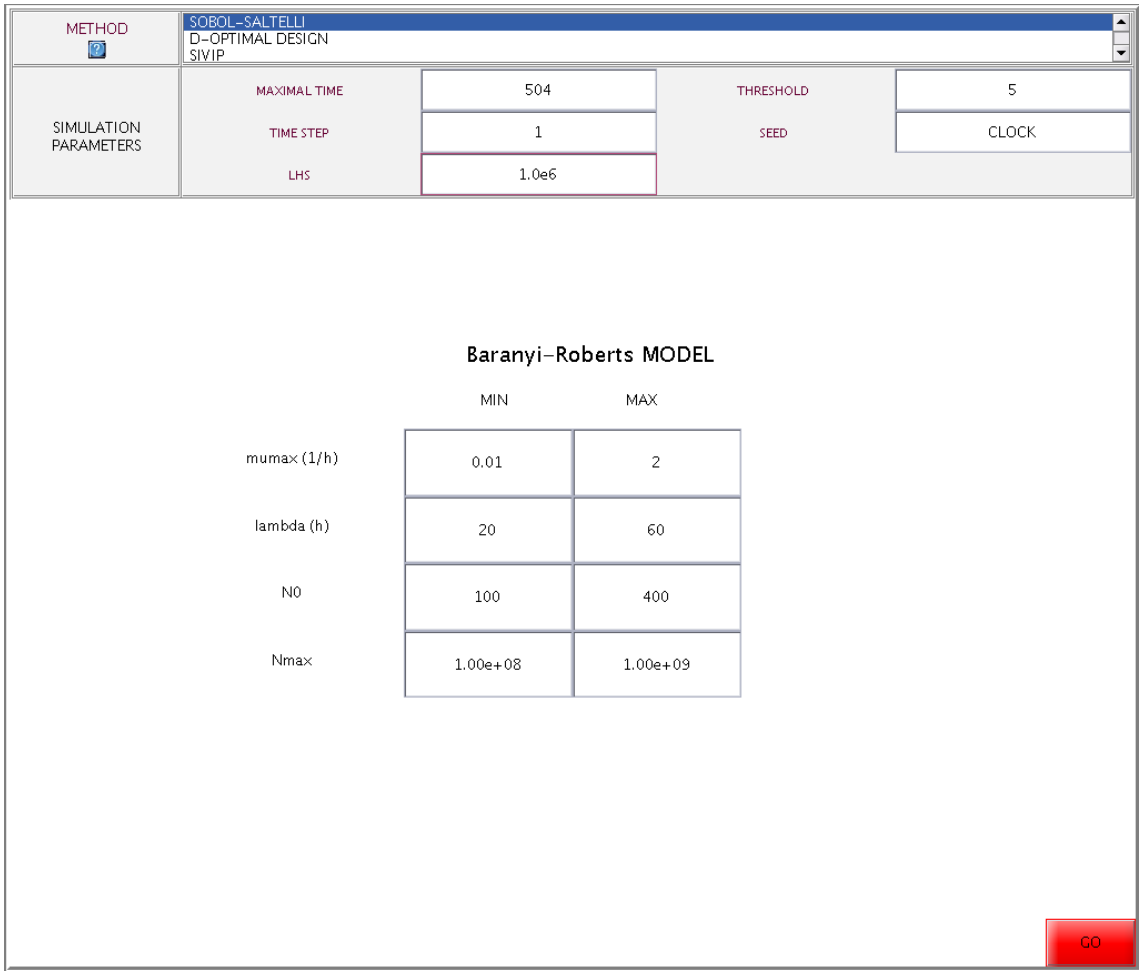


Figure 6.1: Input panel in task `Simulation of an optimal sequential sampling`, method Sobol-Saltelli.

6.1.2 Output

Fig. 6.2 is the output panel issued by Fig. 6.1 execution. The percentages of the total sensivity index (TSI) of each parameter are plotted versus time. The simulated optimal times are the successive optima noted by squares on the curves and listed below the plot.

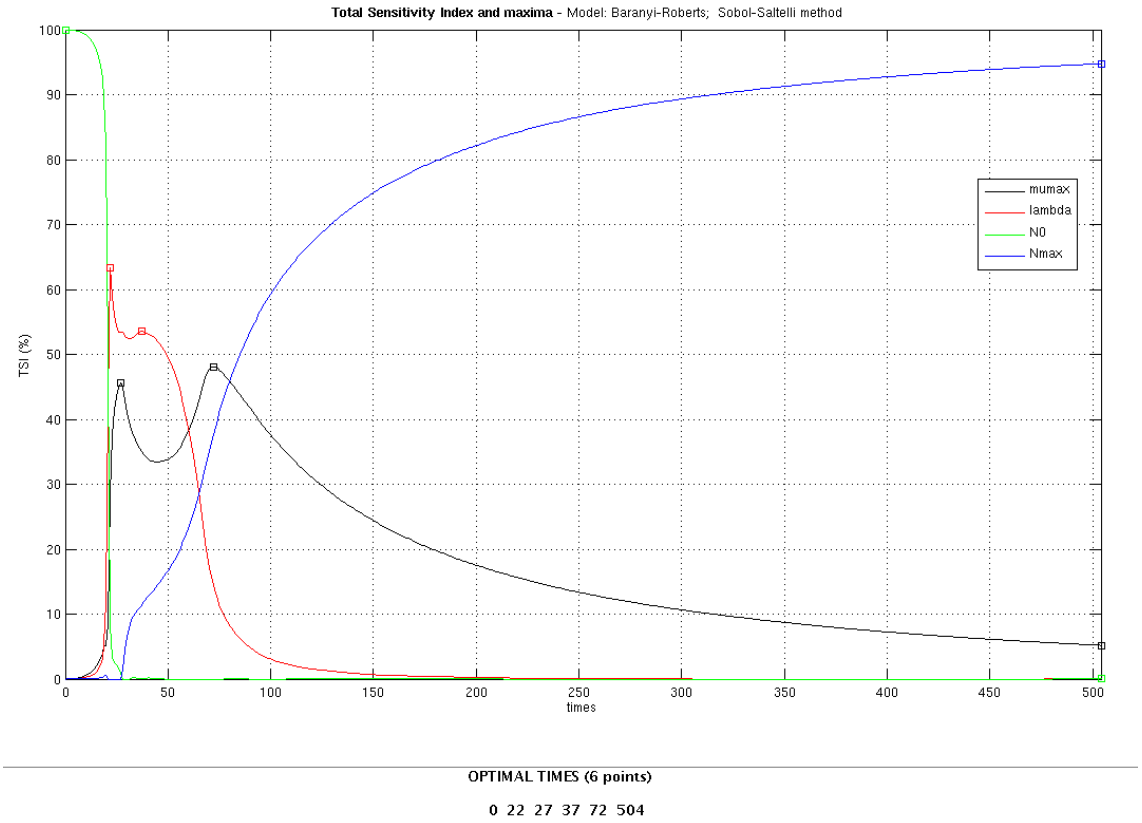


Figure 6.2: Output panel in task `Simulation of an optimal sequential sampling`, method Sobol-Saltelli.

At bottom of Fig. 6.2, the simulated optimal times where the future data must be observed are given.

6.2 Method D-optimal design

This method is preferred if good guess parameter values are known.

6.2.1 Input

Fig. 6.3 is an example of input panel with Baranyi-Roberts model selected.

Figure 6.3: Input panel in task `Simulation of an optimal sequential sampling`, method D-optimal

6.2.2 Output

Fig. 6.4 and Fig.6.5 are the output panels produced by execution of Fig. 6.3.
On Fig.6.5, lapses of constant variance are considered as plateaus when they are equal or greater than the `minimal lapse of constant variance`. Note that experiment support points can be interactively added inside the plateaus.
The simulated optimal times are noted by squares on the curve and listed below the plot.

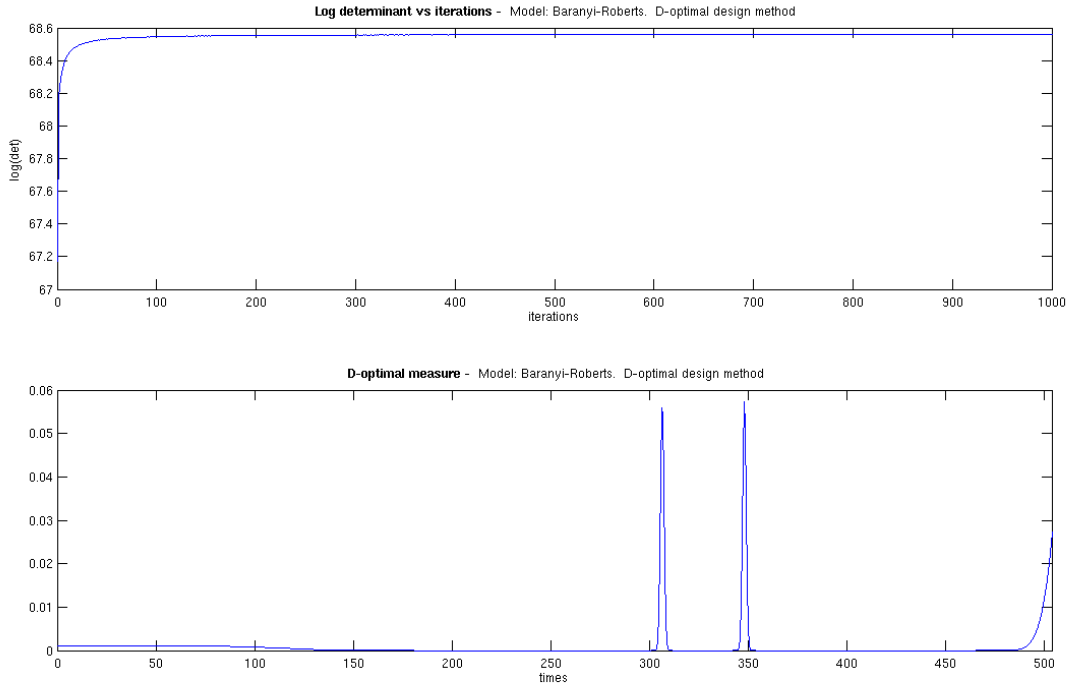


Figure 6.4: Output panel number 1 in task `Simulation of an optimal sequential sampling`, method D-optimal

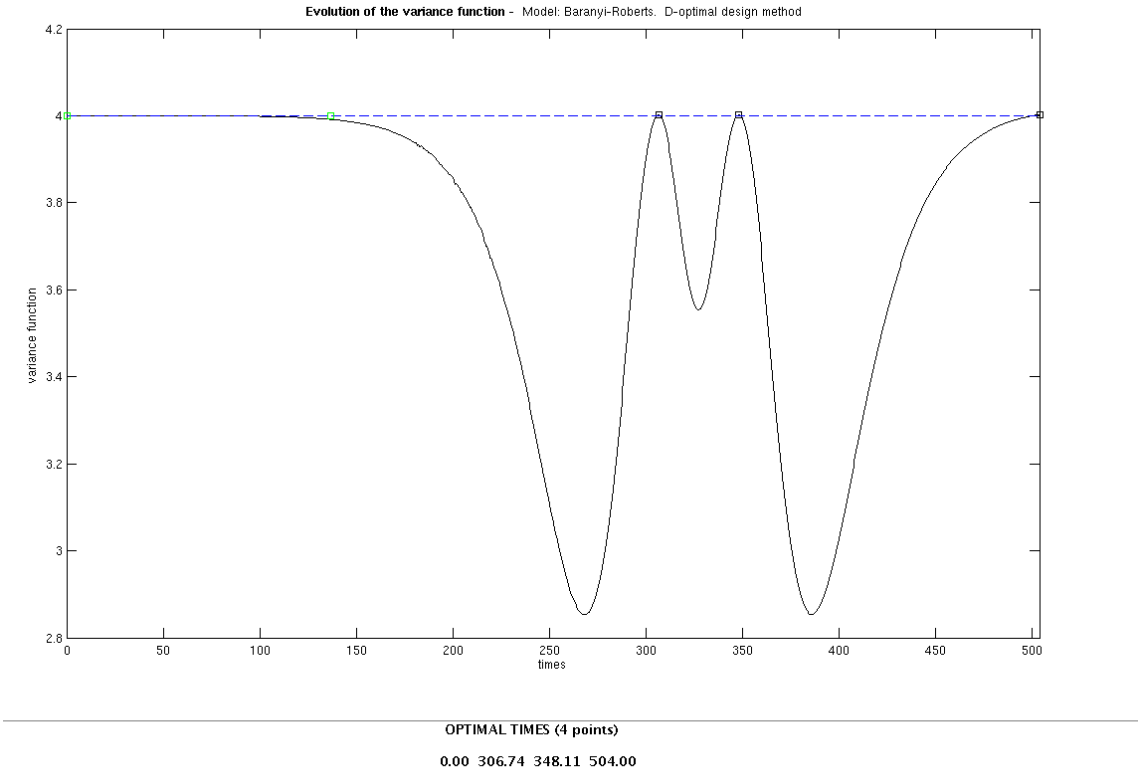


Figure 6.5: Output panel number 2 in task`Simulation of an optimal sequential sampling`, method D-optimal

At bottom of Fig. 6.5, the simulated optimal times where the future data must be observed are given.

6.3 Method SIVIP

This is a more sophisticated method because the optimal times are dynamically determined depending on the preceding observations.

6.3.1 Input panel

Fig. 6.6 is an example of input panel with Baranyi-Roberts model selected. An observation dataset is required for this method. Here, it has been previously simulated by the task `Simulation of an observation dataset` (see Section 7).
Unlike the task `Parametric identification`, no time step is required here. The computational time step is the least of the greatest common divisors between consecutive data times (i.e the value of the `proposition` in the task `Parametric identification`).

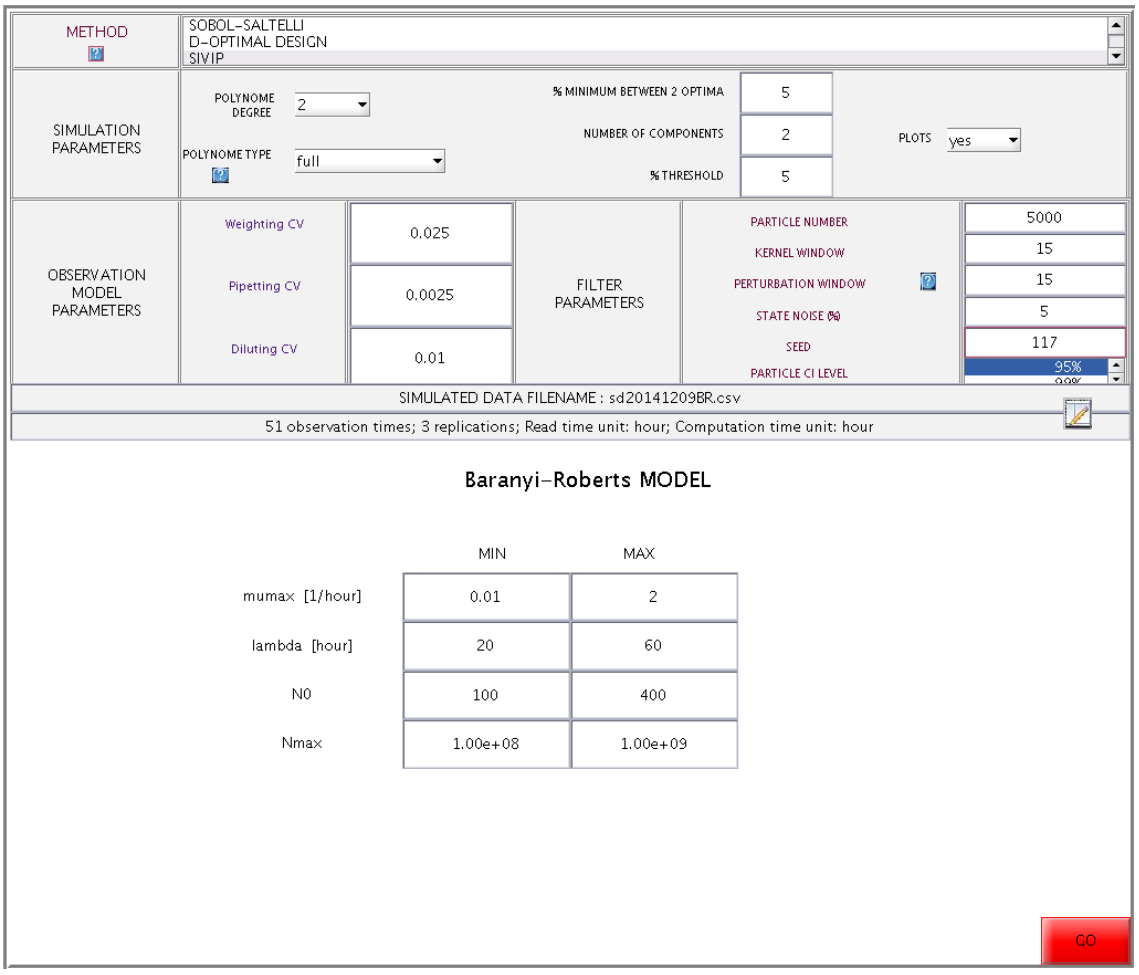


Figure 6.6: Input panel in task `Simulation of an optimal sequential sampling`, method SIVIP

6.3.2 Output

For every optimal time throughout the period, and for each parameter, the percentage of the total sensitivity index (TSI) ([Gauchi and Vila (2011)], [Gauchi and Vila (2013)]) and the explanation percentage ($R^2\%$) of the polynome function of the model parameters (`mumax`, `lambda`, ...) are plotted as calculation goes along. Each time, except for the last one, the next optimal time is noted by a square on the TSI plots : it is located at the next greatest TSI value (Fig. 6.7 and 6.8). At last, all the simulated optimal times are displayed (Fig. 6.9).

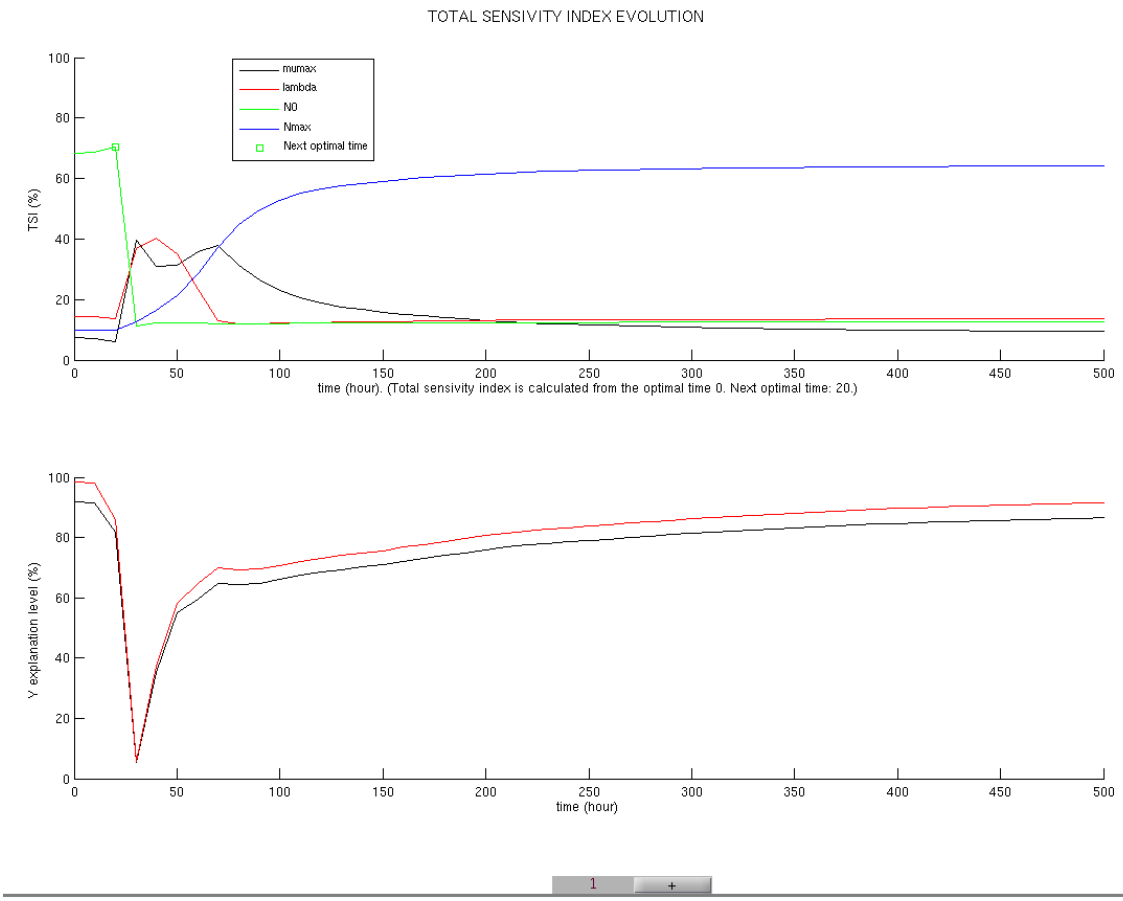


Figure 6.7: Output panel number 1 in task `Simulation of an optimal sequential sampling`, method SIVIP

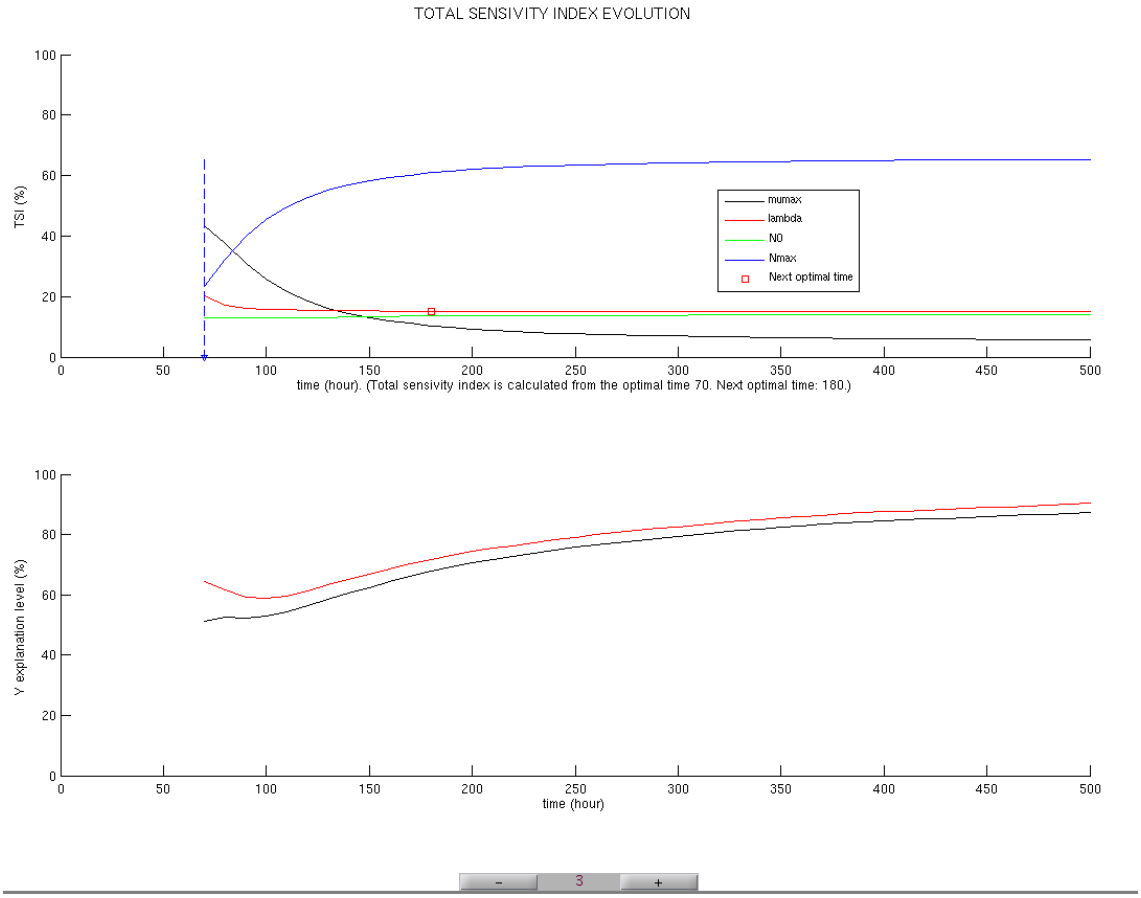


Figure 6.8: Output panel number 3 in task `Simulation of an optimal sequential sampling`, method SIVIP

OPTIMAL TIMES

Model: Baranyi–Roberts. SIVIP method; Number of particles: 5000; Time unit: hour
Number of selected optimal times: 10

time#1	0
time#2	20
time#3	70
time#4	180
time#5	230
time#6	300
time#7	360
time#8	410
time#9	450
time#10	500

Figure 6.9: Output panel in task `Simulation of an optimal sequential sampling`, with SIVIP method: optimal times

Chapter 7

Simulation of an observation dataset

7.1 Function

This task simulates an observation dataset according to a model and its parameters values. The simulated dataset can be stored in a file for further analysis.

7.2 Input

The input panel of this task is Fig. 7.1.

- **Frame 1**, Fig. 7.1
  - `Observation times` are the times where an observation should be simulated. You can either,
    - enter the times through the FILTREX interface. Select `KEYBOARD` : a window opens in which you type in the times,
    - previously store the times in a file. Select `FILE` : browse your folders and select the file in which are stored the times (ASCII text file, a value per line),
    - make the times regularly spaced generated. Select `EQUIDISTANT` : "`Number of times`" values will be generated between 0 and "`Maximal time`" (`Maximal time` and `Number of times` are boxes in frame 2).
- **Frame 2, Fig. 7.1**
  - `Maximal time` and `Number of times` are input in case of equidistant times. In the other cases, they are automatically filled in according to the given times.
  - `Number of replications` is the required number of replications of each observation.

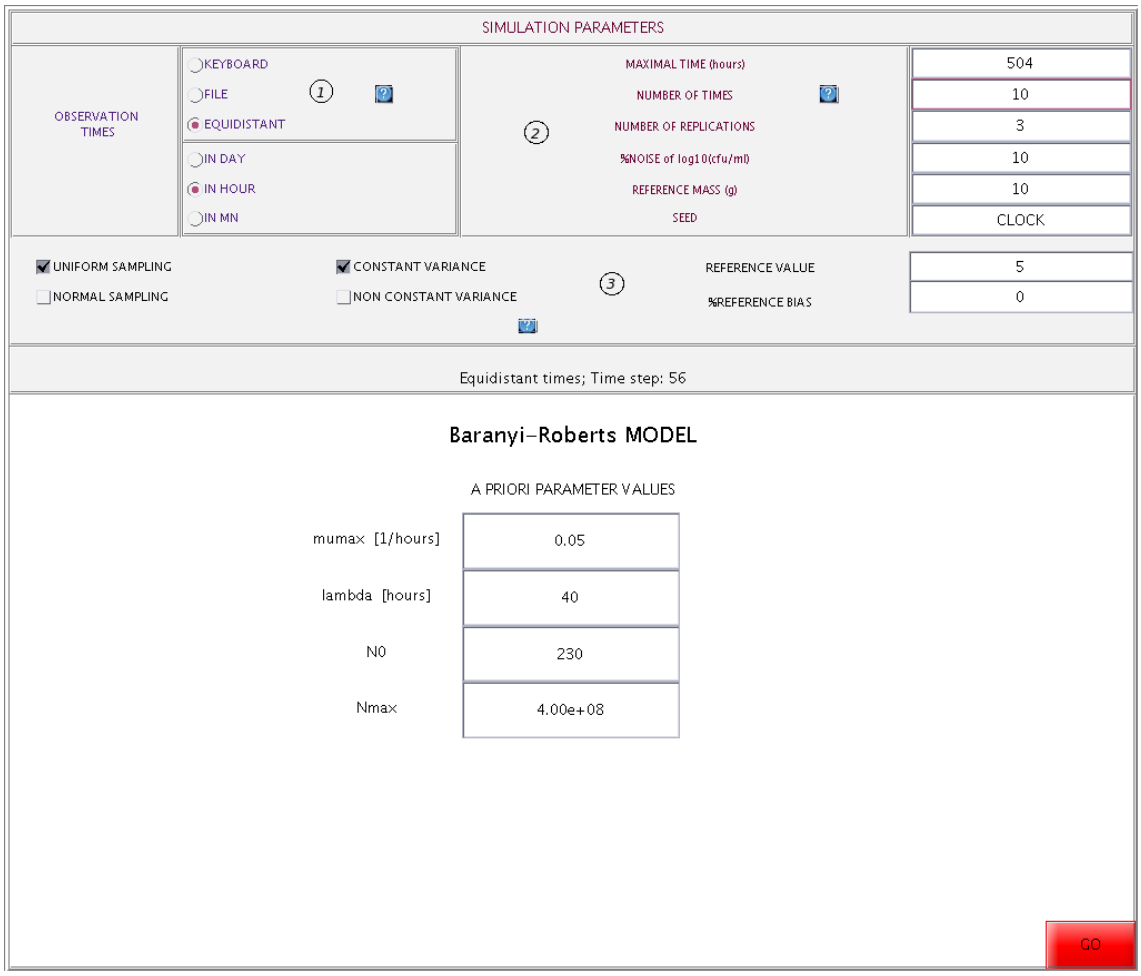


Figure 7.1: Input panel in task `Simulation of an observation dataset`

- `%NOISE of the log10(CFU/ml)` is used to calculate the gap between the replications and the theorical observation.
- `Reference mass` is the amount of analyzed quantity (second column in the observation dataset files, see Section 3.2). Its value is fixed for all the observations.
- `Seed` is the random seed used when generating the replications.

- **Frame 3, Fig. 7.1 or Fig. 7.2**

  - `Uniform sampling`. The gap between replications and theorical observations is calculated from uniformly distributed pseudorandom integers. Their upper bound depends of `%noise` and `reference value` (constant variance) or `power coefficient` (nonconstant variance).

    When there is bias, a quantity proportional to `%reference bias` (constant variance) or `% bias` (nonconstant variance) is added to the gap.

Figure 7.2: Frame 3 of input panel in task `Simulation of an observation dataset`, when non constant variance is selected

– `Normal sampling.` The gap between replications and theorical observations is then random numbers from the normal distribution with mean parameter 0 and standard deviation proportional to `%noise` and `reference value` (constant variance) or `power coefficient` (nonconstant variance). The gap is modified as above when there is bias.

Note : there is no biais when `%reference bias` or `% bias` is equal to 0.

7.3 Output

Theorical observations and simulated data are plotted. Example Fig.7.3.

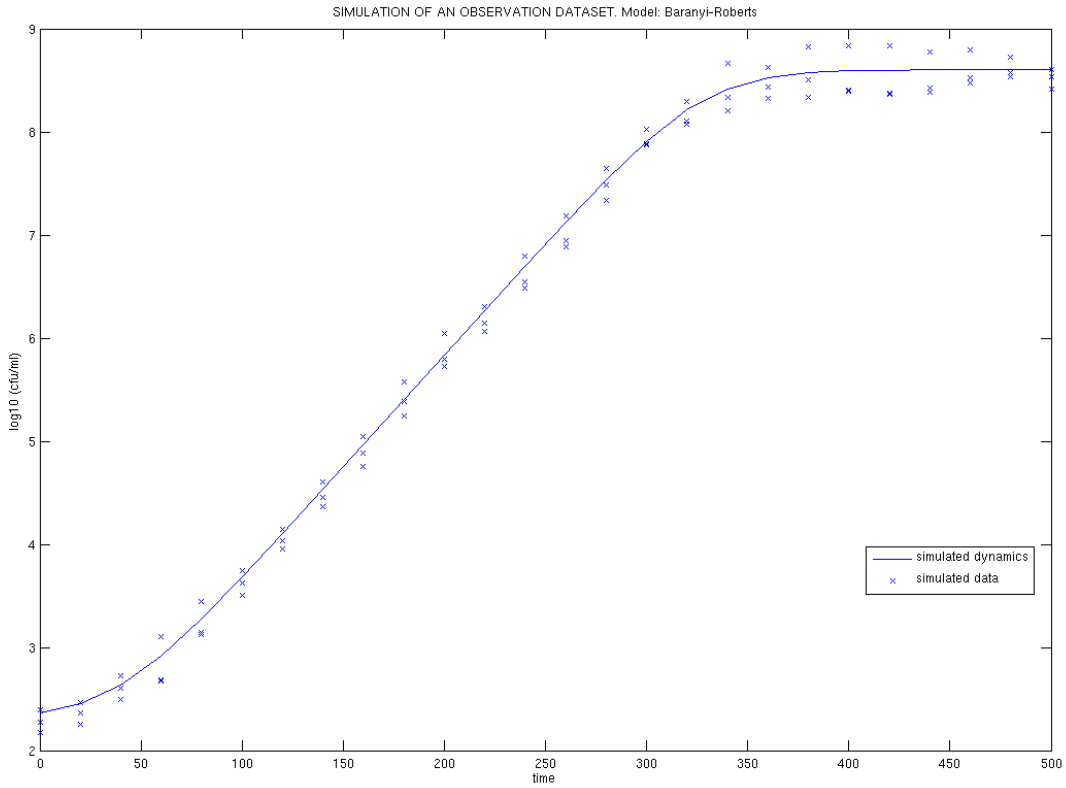


Figure 7.3: Simulated dataset with Baranyi-Roberts model, uniform sampling and constant variance

The simulated dataset is stored in a file by using the `Save -> Save observations` button located in the top bar. This file is in the format described at Section 3.2 and so can be used as input in the other tasks.

Chapter 8

References

[Akashi *et al.* (1975)] Akashi, H., Kumamoto, H. and Nose, K. 1975. Application of Monte Carlo Methods to Optimal Control for Linear Systems under Measurement Noise with Markov Dependent Statistical Property. Int. Journal on Control, 22(6), 821-836.

[Bain and Crisan (2009)] Bain, A., Crisan, D. 2009. Fundamentals of stochastic filtering. Springer, New York, USA.

[Baranyi and Roberts (1994)] Baranyi J. and Roberts T.A. 1994. A dynamic approach to predicting bacterial growth in food. International Journal of Food Microbiology, 23, 277-294.

[Baranyi and Roberts (1995)] Baranyi, J., Roberts, T. A. 1995. Mathematics of predictive food microbiology. Int. Journal of Food. Microbiology, 26, 199-218.

[Bartolucci *et al.* (2006)] Bartolucci, F., Scaccia ,L., Mira, A. 2006. Efficient Bayes factor estimation from the reversible jump output, Biometrika, 93(1), 41-52.

[Bidot *et al.* (2009)] Bidot, C. and Gauchi, J.-P. and Vila, J.-P. 2009. Programmation MATLAB : du filtrage non linéaire par convolution de particules pour l'identification et l'estimation d'un système dynamique microbiologique. Rapport technique 2009-3. INRA, UR341 Mathématiques et Informatique Appliquées, F-78350 Jouy-en-Josas, France.

[Campillo and Rossi (2009)] Campillo, F., Rossi, V. 2009. Convolution particle filter for parameter estimation in general state-space models. IEEE Trans. Aero. Elec. Sys., 45, 1063-1072.

[Cappé *et al.* (2005)] Cappé, O., Moulines, E., Rydén, T. 2005. Inference in Hidden Markov Models. Springer, New York, USA.

[Chen (1993)] Chen, G. 1993. Approximate Kalman Filtering. World Scientific, Approximations and Decompositions, Vol. 2.

[Choquet and Rossi (2005)] Choquet, R. and Rossi, V. (2005). Routines pour le filtrage particulaire. Rapport CEFE-CNRS

[Davis (1981)] Davis, M. 1981. New Approach to Filtering nonlinear Systems. IEEE Proceedings, Part D, 128(5), 166-172.

[Dawid (1984)] Dawid, A. P. 1984. Statistical theory: The prequential approach, Journal of the Royal Statistical Society, Series A, 147, 278-292.

[Dean *et al.* (2011)] Dean, T.A., Singh, S.S., Jasra, A., Peters, G.W. 2011. Parameter estimation for hidden Markov models with intractable likelihoods. arXiv:1103.5399v1.

[Del Moral *et al.* (1992)] Del Moral, P., Rigal G., Salut G. 1992. Estimation et commande optimale non linéaire: un cadre unifié pour la résolution particulaire. Technical report 2, LAAS/CNRS, contrat DRET-DIGILOG, Toulouse.

[Del Moral (1995)] Del Moral, P. 1995. Nonlinear filtering using random particles. Theory Probab. Appl, 40(4), 690-701.

[Del Moral (1998)] Del Moral, P. 1998. A uniform convergence theorem for the numerical solving of the nonlinear filtering problem. Journal of Applied Probability, 35(4), 873-884.

[Del Moral and Jacod (2001)] Del Moral, P., Jacod, J. 2001. Interacting Particle Filtering With Discrete Observation. In Sequential Monte Carlo Methods in Practice Ed. Doucet A., de Freitas, N., Gordon., N., Statistics for Engeering and Information Science, Springer, 43-75.

[Del Moral *et al.* (2001)] Del Moral, P., Jacod, J., Protter, P. 2001.The Monte-Carlo method for filtering with discrete-time observations. Probab Theory Relat. Fields, 120, 346-368.

[Del Moral (2004)] Del Moral, P. 2004. Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications, Springer, New York, USA.

[Del Moral *et al.* (2012)] Del Moral, P., Doucet, A., Jasra, A. 2012. An adaptive sequential Monte Carlo Method for approximate Bayesian computation. Statistics and Computing, 22 (5), 1009-1020.

[Donev and Atkinson (1988)] Donev, A.N. and Atkinson, A. C. 1988. An Adjustment Algorithm for the Construction of Exact D -Optimum Experimental Designs. Technometrics, 30 (4), 429-433.

[Doucet *et al.* (2001)] Doucet, A., de Freitas, N., Gordon, N. 2001. Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science, Springer, New York, USA.

[Doucet and Rosset (2006)] Doucet, A., Rousset, M. 2006, in discussion on the paper by Beskos *et al.*, Journal of the Royal Statistical Society, series B, 68(3), 333-382.

[Droesbeke *et al.* (1997)] Droesbeke, J.-J., Fine, J., Saporta, G. 1997. Plans d'experiences: applications à l'entreprise. Editions Technip.

[Ellouze *et al.* (2010)] Ellouze, M., Gauchi, J.P., Augustin, J.C. 2010. Global sensitivity analysis applied to a contamination assessment model of Listeria monocytogenes in cold smoked salmon at consumption. Risk Analysis, 30 (5), 841-852.

[Fearnhead *et al.* (2008)] Fearnhead, P., Papaspiliopoulos, O., Roberts, G.O. 2008. Particle filters for partially observed diffusions. Journal of the Royal Statistical Society, series B, 70(4), 755-777.

[Gauchi *et al.* (2009)] Gauchi, J.P., Bidot, J.C., Augustin, J.C., Vila, J.P. 2009. Identification of complex microbiological dynamic systems by nonlinear filtering. 6^{th} International Conference on Predictive Modelling in Foods, Washington, USA.

[Gauchi *et al.* (2010)] Gauchi, J.P.,Lehuta, S., Mahévas, S. 2010. Optimal Sensitivity Analysis under Constraints: Application to fisheries. Sixth International Conference on Sensitivity Analysis of Model Output. 2(6), 7658-7659.

[Gauchi and Vila (2011)] Gauchi, J.P., Vila, J.P. 2011. Optimal sequential sampling design for improving parametric identification of complex microbiological dynamic systems by nonlinear filtering. 7^{th} International Conference on Predictive Modelling of Food Quality and Safety, Dublin, Ireland.

[Gauchi *et al.* (2011)] Gauchi, J.P., Vila, J.P., Bidot, C., Atlijani, E., Coroller, L., Augustin, J.C., Del Moral, P. 2011. FILTREX: a new software for identification and optimal sampling of experiments for complex microbiological dynamic systems by nonlinear filtering. 7^{th} International Conference on Predictive Modelling of Food Quality and Safety, Dublin, Ireland.

[Gauchi *et al.* (2012)] Gauchi, J.-P. and Bidot, C. and Bouvier, A. and Vila, J.-P. and Coroller, L. and Augustin, J.-C. and Del Moral, P. 2012. FILTREX: Un logiciel convivial pour la microbiologie alimentaire prévisionnelle. Modélisation dynamique de la croissance ou décroissance de populations bactériennes. Poster aux Journées 2012 des microbiologistes INRA.

[Gauchi and Vila (2013)] Gauchi, J.-P. and Vila, J.-P. 2013. Nonparametric particle filtering approaches for identification and inference in nonlinear state-space dynamic systems. Statistics and Computing, 23, 523-533.

[Gauchi *et al.* (2013)] Gauchi, J.-P. and Bidot, C. and Bouvier, A. and Vila, J.-P. and Coroller, L. and Augustin, J.-C. 2013. A New User-friendly Software for Parametric Identification, Model Comparison and Optimal Sequential Sampling of Experiments of Complex Microbiological Dynamic Systems by Nonlinear Filtering. Poster and demo at International Conference on Predictive Modelling in Food, 16-20 September 2013, Paris, France.

[Gordon *et al.* (1993)] Gordon, N.J., Salmond, D.J. and Smith, A.F.M. 1993. Novel approach to nonlinear non-Gaussian Bayesian state estimation. IEE Proceedings-F, 140(2), 107-113.

[Hürzeler and Künsch (1998)] Hürzeler M. and Künsch, H.R. 1998. Monte Carlo Approximations for General State-Space Models.Journal of Computational and Graphical Statistics, 7(2), 175-193.

[Jasra *et al.* (2012)] Jasra, A., Singh, S.S., Martin, J.S., McCoy, E. 2012. Filtering via approximate Bayesian computation. Statistics and Computing, 22 (6), 1223-1237.

[Jazwinski (1970)] Jazwinski, A.H. 1970. Stochastic processes and filtering theory. Academic Press, London, GB.

[Jeffreys (1961)] Jeffreys, H. 1961. Theory of Probability, Oxford University Press, Oxford, GB.

[Kass and Raftery (1995)] Kass, R.E, Raftery, A.E. 1995. Bayes factors. J. Amer. Statist. Assoc., 90, 773-795.

[Kitagawa (1987)] Kitagawa, G. 1987. Non-Gaussian State-Space Modeling of Nonstationary Time Series. Journal of the American Statistical Association, 82(400), 1032-1041.

[Kramer and Sorenson (1988)] Kramer, S.C., Sorenson, H.W., 1988. Recursive Bayesian estimation using piece-wise constant approximations. Automatica, 21, 789-801.

[Lai (1998)] Lai, T.L. 1998. Information bounds and quick detection of parameter changes in stochastic systems. IEEE Trans. Inform. Theory, 44, 2917-2929.

[LeGland and Oudjane (2004)] LeGLand F., Oudjane, N. 2004. Stability and Uniform Approximation of Nonlinear Filters using the Hilbert Metric, and Application to Particle Filters. The Annals of Applied Probability, 14(1), 144-187.

[Liu and Chen (1998)] Liu, J.S. and Chen, R. 1998. Sequential Monte Carlo Methods for Dynamic Systems. Journal of the American Statistical Association, 93(443), 1032-1044.

[Liu (2001)] Liu, J.S. 2001. Monte Carlo strategies in scientific computing. Springer-Verlag, New York, USA.

[Lorden (1971)] Lorden, G. 1971. Procedures for reacting to a change in distribution. Ann. Math. Stat., 42, 1897-1908.

[Magni (2009)] Magni, L, Raimondo, D.M., Allgöwer, F. (Eds.). 2009. Nonlinear model predictive control. Springer-Verlag, New York, USA.

[Musso *et al.* (2001)] Musso, C., Oudjane, N., LeGland, F. 2001. Improving Regularised Particle Filters. In Sequential Monte Carlo Methods in Practice, Eds: Doucet A., de Freitas N., Gordon N., Statistics for Engineering and Information Science, Springer-Verlag, New York, USA, 247-271.

[Oudjane (2000)] Oudjane, N. 2000. Stabilité et approximations particulaires en filtrage non linéaires, application au pistage. PhD thesis, Université de Rennes I, France.

[Page (1954)] Page, E.S. 1954. Continuous inspection schemes. Biometrika, 41, 100-115.

[Parzen (1962)] Parzen, E. 1962. On estimation of a probability density function and mode. Ann. Math. Statist., 33, 1065-1076.

[Robert *et al.* (2011)] Robert, C., Cornuet, J.M., Marin, J.M., Pillai, N.S. 2011. Lack of confidence in ABC model choice. in arXiv: 1102.4432.

[Rodriguez (2007)] Rodriguez-Fernandez, M., Kucherenko, S., Pantelides, C., Shah, N. 2007. Optimal experimental design based on global sensitivity analysis. 17^{th} European Symposium on Computer Aided Process Engineering.

[Rossi (2004)] Rossi, V. 2004. Filtrage non linéaire par noyaux de convolution. Application à un procédé de dépollution biologique. Thèse de Docteur en science, ENSAM.

[Rossi and Vila (2005)] Rossi, V. and Vila, J.P. 2005. Approche non paramétrique du filtrage de système non linéaire à temps discret et à paramètres inconnus, C.R. Acad. Sci., Ser I (340), 759-764, Paris, France.

[Rossi and Vila (2006)] Rossi, V. and Vila, J.P. 2006. Nonlinear filtering in discrete time: A particle convolution approach. Inst. Stat. Univ. Paris, L, 3, 71-102.

[Rosso (1995)] Rosso, L. 1995. Modélisation et microbiologie prévisionnelle: élaboration d'un nouvel outil pour l'agro-alimentaire. Thèse de Doctorat en Science, Université Claude Bernard - Lyon I, France.

[Rosso *et al.* (1996)] Rosso L., Bajard S., Flandrois J.P., Lahellec C., Fournaud J., Veit P. 1996. Differential growth of Listeria monocytogenes at 4 and 8°C: Consequences for the shelf life of chilled products. Journal of Food Protection, 59, 944-949

[Šimandl and Královec (2000)] Šimandl, M. and Královec, J. 2000. Filtering, prediction and smoothing with Gaussian sum representation. In Proceedings of the IFAC 12^{th} Symposium on System Identification. Santa Barbara, USA.

[Šimandl *et al.* (2006)] Šimandl, M., Královec, J., Söderström, T. 2006. Advanced point-mass method for nonlinear state estimation. Automatica, 42, 1133-1145.

[Saltelli (2002)] Saltelli, A. 2002. Making best use of model evaluations to compute sensitivity indices. Computer Physics Communications, 145(2), 280-297.

[Sorenson and Alspach (1971)] Sorenson, H.W. and Alspach, D;L. 1971. Recursive Bayesian estimation using Gaussian sums. Automatica, 7, 465-479.

[Verdier *et al.* (2008)] Verdier, G., Hilgert, N., Vila, J.P. 2008. Optimality of CUSUM rule approximations in change-point detection problems. Application to nonlinear state-space systems. IEEE Trans. on Inf.Theory, 54, 5102-5112.

[Vila and Saley (2009)] Vila, J.P., Saley, I. 2009. Estimation de facteurs de Bayes entre modèles dynamiques non linéaires à espace d'état. C.R. Acad. Sci., Ser. I, 347, 429-434, Paris, France.

[Vila *et al.* (2009)] Vila, J.-P. and Gauchi, J.-P. and Bidot, C. 2009. Identification de systèmes dynamiques microbiologiques complexes par filtrage nonlinéaire. Actes des 41èmes Journées de Statistique, Bordeaux, France.

[Vila (2011)] Vila, J.P. 2011. Nonparametric multi-step prediction in nonlinear state space dynamic systems. Statistics and Probability Letters, 81, 71-76.

[Vila and Gauchi (2011)] Vila, J.P, Gauchi, J.P. 2011. Predictive control of stochastic non-linear state space dynamic systems: a particle nonparametric approach. Technical Report, UMR MISTEA.

[Vila (2012)] Vila, J.P., 2012. Enhanced consistency of the Resampled Convolution Filter. Statistics and Probability Letters, 82, 786-797.

[Warnes (2001)] Warnes, G.R., 2001. The Normal Kernel Coupler: An adaptave Markov Chain Monte Carlo method for efficiently sampling from multi-modal distributions. Technical Report 39, Departement of Statistics, University of Washington, USA.

Appendix A

Technical details on FILTREX mathematics

A.1 Introduction to a nonparametric particle filtering approach

As previously mentioned, the main aim of filtering is to estimate the probability distribution functions of x_t and that of the unknown parameters θ, conditional on the past values of the output variables up to time t, from a Bayesian point of view. We will assume that these distributions are absolutely continuous and will attempt to estimate their densities, $p_t(x, \theta|y_1, \ldots, y_t)$, $p_t(x|y_1, \ldots, y_t)$, $p_t(\theta|y_1, \ldots, y_t)$ and possibly the unknown parameter values, even if the distribution $G_t(.|x, \theta)$ is analytically unknown (but allows simulations of y_t from x and θ values).
The nonparametric particle approach we consider for this purpose has already been completely described in [Rossi (2004)] and [Rossi and Vila (2005)], [Rossi and Vila (2006)] for two of these nonparametric filters, with full proofs of convergence. We refer to these papers for details and extensions ([Vila (2011)], [Vila (2012)]). We will restrict the present description to the principles of the Resampled-Convolution filter (R-CF), the most efficient filter in this family.

Assumptions
Let

- p_0^x be the known probability density of the state variable vector at time $t = 0$.
- p_0^θ be a given prior density for $\theta \in \Theta$, non zero for θ^*, the true unknown values of the parameters.
- $y_{1:t} = (y_1, \ldots, y_t)$ (notation).

A.2 The R-CF Algorithm: an overview

Let us first note that the unknown static parameters θ can be considered as special state variables by just introducing the parameter invariance, $\theta_t = \theta_{t-1}$ i.e. $\theta_t \sim \delta_{\theta_{t-1}}(.)$, as a new state equation into model (1.1), where $\delta_{\theta_{t-1}}(.)$ is the Dirac probability measure charging θ_{t-1}.
Let us note also that the conditional density $p_t(x_t, \theta_t|y_{1:t})$ to be estimated, is such that

$$p_t(x_t, \theta_t|y_{1:t}) \propto \int_{\mathbb{R}^{t(d+p)}} \Big[\prod_{j=1}^{t} g_j(y_j|x_j, \theta_j) q_j(x_j|x_{j-1}, \theta_{j-1}) \mathbb{1}_{\theta_{j-1}}(\theta_j)\Big] p_0^x p_0^\theta dx_0, \ldots, dx_{t-1} d\theta_0, \ldots, d\theta_{t-1} \tag{A.1}$$

with similar expressions for its marginals $p_t(x_t|y_{1:t})$ and $p_t(\theta_t|y_{1:t})$.
Let $K^x_{h_n}, K^\theta_{h_n}$ and $K^y_{h_n}$ be symmetric, positive kernel functions of dimension d, p and s, respectively, with common scalar window-width parameter h_n (this last assumption could be relaxed). $K^y_{h_n}$ will be taken as: $K^y_{h_n}(w) = \frac{1}{h_n^s} K^y(\frac{w}{h_n})$, $w \in \mathbb{R}^s$, where $K^y(.)$ is a basic Parzen-Rosenblatt kernel ([Parzen (1962)]) of dimension s, so that $K^y(.)$ is bounded, positive, symmetric, $\lim_{||w||\to\infty} ||w||^s K^y(w) = 0$ and $\int K^y d\lambda = 1$, where λ is the Lebesgue measure. *Idem* for $K^x_{h_n}$ and $K^\theta_{h_n}$, with respect to Parzen-Rosenblatt kernels $K^x(.), K^\theta(.)$ of dimension d and p, respectively.
A given number n of particles are simulated at each time t, according to the following recursive scheme that obeys the classical two steps of filtering:

- $t = 0$: for $i = 1, \ldots, n$, let $\bar{x}_0^i \sim p_0^x$, $\bar{\theta}_0^i \sim p_0^\theta$.
- $t > 0$:
  - Prediction step: simulation of n particles.
    For $i = 1, \ldots, n$
    - if $t = 1$: let $x_1^i \sim q_1(.|\bar{x}_0^i, \bar{\theta}_0^i), \theta_1^i = \bar{\theta}_0^i$, $y_1^i \sim g_1(.|x_1^i, \theta_1^i)$.
    - if $t > 1$: let $(\bar{x}_{t-1}^i, \bar{\theta}_{t-1}^i) \sim p_{t-1}^n(x, \theta|y_{1:t-1})$, and $x_t^i \sim q_t(.|\bar{x}_{t-1}^i, \bar{\theta}_{t-1}^i)$, $\theta_t^i = \bar{\theta}_{t-1}^i$, $y_t^i \sim g_t(.|x_t^i, \theta_t^i)$.
  - Updating step (Bayes Formula kernel approximations):
    Estimation of the conditional probability densities and expectations

$$p_t^n(x,\theta|y_{1:t}) = \frac{\sum_{i=1}^n K_{h_n}^y(y_t - y_t^i) \times K_{h_n}^\theta(\theta - \theta_t^i) \times K_{h_n}^x(x - x_t^i)}{\sum_{i=1}^n K_{h_n}^y(y_t - y_t^i)} \quad (A.2)$$

$$p_t^n(x|y_{1:t}) = \frac{\sum_{i=1}^n K_{h_n}^y(y_t - y_t^i) \times K_{h_n}^x(x - x_t^i)}{\sum_{i=1}^n K_{h_n}^y(y_t - y_t^i)} \quad (A.3)$$

$$p_t^n(\theta|y_{1:t}) = \frac{\sum_{i=1}^n K_{h_n}^y(y_t - y_t^i) \times K_{h_n}^\theta(\theta - \theta_t^i)}{\sum_{i=1}^n K_{h_n}^y(y_t - y_t^i)} \quad (A.4)$$

$$\hat{x}_t^n = \frac{1}{n}\sum_{i=1}^n \bar{x}_t^i \quad \text{and} \quad \hat{\theta}_t^n = \frac{1}{n}\sum_{i=1}^n \bar{\theta}_t^i \quad (A.5)$$

- $t = t + 1$

- Go back to step t.

Remark 1 *From a practical point of view, the generation of the n particles $(\bar{x}_t^i, \bar{\theta}_t^i) \sim p_t^n(x,\theta|y_{1:t})$ at time $t+1$, does not require formal simulations according to the just estimated density (A.2): These particles can be easily obtained by sampling from the population of n particles $\{(x_t^i, \theta_t^i)\}$ according to a multinomial with weights $\left\{\frac{K_{h_n}^y(y_t - y_t^i)}{\sum_{i=1}^n K_{h_n}^y(y_t - y_t^i)}\right\}$, followed by addition of random perturbations according to the density corresponding to the kernel $K_{h_n}^x K_{h_n}^\theta$.*

Remark 2 *Formula (A.2) results from the kernel estimation of the density $p_t(x,\theta|y_{1:t}) = \frac{p_t(x,\theta,y_t|y_{1:t-1})}{p_t(y_t|y_{1:t-1})}$, by noticing that the n particles $\{x_t^i, \theta_t^i, y_t^i\}$ are conditioned by $y_{1:t-1}$. Similar remarks apply to (A.3) and to (A.4).*

Remark 3 *The previous conditional density estimate $p_t^n(x,\theta|y_{1:t})$, can be seen as an SMC-based estimate of the following approximation of the true conditional density $p_t(x,\theta|y_{1:t})$:*

$$\hat{p}_t(x,\theta|y_{1:t}) \;=$$

$$\Big[\int_{I\!R^{2d+2p+s}} p_{t-1}(x_{t-1},\theta_{t-1}|y_{1:t-1})q_t(\check{x}_t|x_{t-1},\theta_{t-1})\mathbb{1}_{\theta_{t-1}}(\check{\theta}_t)g_t(\check{y}_t|\check{x}_t,\check{\theta}_t)$$
$$K^y_{h_n}(y_t-\check{y}_t)K^x_{h_n}(x-\check{x}_t)K^\theta_{h_n}(\theta-\check{\theta}_t)dx_{t-1}d\theta_{t-1}d\check{x}_t d\check{\theta}_t d\check{y}_t\Big]\Big/$$
$$\Big[\int_{I\!R^{2d+2p+s}} p_{t-1}(x_{t-1},\theta_{t-1}|y_{1:t-1})q_t(\check{x}_t|x_{t-1},\theta_{t-1})\mathbb{1}_{\theta_{t-1}}(\check{\theta}_t)g_t(\check{y}_t|\check{x}_t,\check{\theta}_t)$$
$$K^y_{h_n}(y_t-\check{y}_t)dx_{t-1}d\theta_{t-1}d\check{x}_t d\check{\theta}_t d\check{y}_t\Big] \tag{A.6}$$

- *with* $x_0 \sim p^x_0(.),\ \theta_0 \sim p^\theta_0(.)$ *and* $x_1 \sim q_1(.|x_0,\theta_0),\ \theta_1=\theta_0$.

Similar remarks can be done for the pdf estimates $p^n_t(x|y_{1:t})$ *and* $p^n_t(\theta|y_{1:t})$.

Remark 4 *This convolution particle filter algorithm can be compared with recent Sequential Monte Carlo approaches that sample from Approximate Bayesian Computation approximations of the successive target densities of interest ([Dean et al. (2011)]). This theoretical and numerical comparison is outside the scope of the paper and is referred to in another article to come. Let us just say here following [Jasra et al. (2012)], that replacing the kernel functions* $K^x_{h_n}$ *and* $K^\theta_{h_n}$ *with Dirac masses, and* $K^y_{h_n}$ *with* $\mathbb{1}_{A_{\epsilon,y_t}}$ *where* A_{ϵ,y_t} *is an ABC* ϵ*-controlled acceptance region, leads to a well-known SMC-ABC algorithm ([Del Moral et al. (2012)]). In the present approach, the kernel functions are not adapted with such an* ϵ *but with the bandwidth parameter* h_n *to ensure convergence of the pdf estimates for any* $t>1$ *as* n *grows to infinity, as shown in the following two theorems.*

A.2.1 L_1 a.s. convergence properties of the R-CF filter

Theorem 5 *For any* $t>1$*, if the pdf* $p_t(y|y_{1:t-1})$ *is continuous and strictly positive at* y_t*, if there exists* $M>0$ *such that* $p_t(y|x_t,\theta)\le M$*, and if* $Var[x_t,\theta_t|y_{1:t}]$ *exists and is bounded, then*

$$\left\{\begin{array}{l}\lim_{n\to\infty}\frac{nh_n^{s+d+p}}{\log n}=\infty\\ h_n^s=O(n^{-\alpha/2}),\\ 0<\alpha<1\end{array}\right. \Longrightarrow \quad \begin{array}{l}\lim_{n\to\infty}\|p^n_t(x,\theta|y_{1:t})-p_t(x,\theta|y_{1:t})\|_{L1}=0 \quad a.s.\\ \lim_{n\to\infty}\|p^n_t(x|y_{1:t})-p_t(x|y_{1:t})\|_{L1}=0 \quad a.s.\\ \lim_{n\to\infty}\|p^n_t(\theta|y_{1:t})-p_t(\theta|y_{1:t})\|_{L1}=0 \quad a.s.\\ \lim_{n\to\infty}\left|\hat{x}^n_t-\mathbb{E}[x_t|y_{1:t}]\right|=0 \quad a.s.\\ \lim_{n\to\infty}\left|\hat{\theta}^n_t-\mathbb{E}[\theta_t|y_{1:t}]\right|=0 \quad a.s.\end{array}$$

with $\big\|\Phi(z)\big\|_{L_1}=\int\big|\Phi(z)\big|dz$*, for an integrable function* $\Phi(z)$.

Proof: [Rossi (2004)]; [Rossi and Vila (2005)], [Rossi and Vila (2006)].

Remark 6 *Bounds for the corresponding expected L_1-discrepancies have also been characterised under some additional assumptions on the joint pdf $p_t(x,\theta,y|y_{1:t})$ and the kernel functions $K^x_{h_n}, K^\theta_{h_n}$ and $K^y_{h_n}$. Under these assumptions, for any $t>1$:*

$$E\Big[\|p^n_t(x,\theta|y_{1:t}) - p_t(x,\theta|y_{1:t})\|_{L1}\Big] \leq (2^t-1)\Big[O_t(h_n) \;+\; O_t\big((nh_n^{s+d+p})^{-1/2}\big)\Big] \qquad \text{(A.7)}$$

Similar bounds exist for $\|p^n_t(x|y_{1:t}) - p_t(x|y_{1:t})\|_{L1}$ and $\|p^n_t(\theta|y_{1:t}) - p_t(\theta|y_{1:t})\|_{L1}$.

A.2.2 Almost sure punctual convergence of the R-CF filter

Theorem 7 *For any $t>1$, if the pdf $p_t(y|y_{1:t-1})$ is continuous and strictly positive at y_t and if there exist $M_1>0$ such that $p_t(x|x_{t-1},\theta)\leq M_1$, $M_2>0$ such that $p_t(y|x_t,\theta)\leq M_2$ and $M_3>0$ such that $p_t(\theta|y_{1:t})\leq M_3$, then*

$$\begin{cases} h_n \;=\; O\big(n^{-\beta/(s+d+2p)}\big) \\ 0<\beta<1/2 \end{cases} \Longrightarrow \quad \begin{array}{lll} \lim_{n\to\infty} p^n_t(x|y_{1:t}) \;=\; p_t(x|y_{1:t}) & a.s. & (A.9) \\ \lim_{n\to\infty} p^n_t(\theta|y_{1:t}) \;=\; p_t(\theta|y_{1:t}) & a.s. & (A.10) \end{array}$$

Proof: [Vila (2012)].

A.3 Optimisation of the nonparametric particle filtering

To use the R-CF filter on a given time interval $[0,\tau]$, the output observation times do not need to be consecutive but can be restricted to a subset $t_1<t_2<\ldots<t_j<t_{j+1}<\ldots<t_{_H}$ with $t_1\geq 1$ and $t_{_H}\leq\tau$. The filter convergence properties are maintained if between t_j and t_{j+1}, the n particles $\{x^i,\theta^i\}$ are simply updated according to: $x^i_{t+1}\sim q_{t+1}(.|x^i_t,\theta^i_t)$, $\theta^i_{t+1}=\theta^i_t$, $i=1,\ldots,n$, for $t_j\leq t\leq t_{j+1}-1$. At time t_{j+1} a new observation $y_{t_{j+1}}$ is available and density estimates $p_{t_{j+1}}(x,\theta|y_{t_1},\ldots,y_{t_{j+1}})$, $p_{t_{j+1}}(x|y_{t_1},\ldots,y_{t_{j+1}})$ and $p_{t_{j+1}}(\theta|y_{t_1},\ldots,y_{t_{j+1}})$ are estimated as usual, and the filtering process continues.

At any time t during the filtering process, all the information about the unknown parameters θ is provided by their probability density function, conditional on the past observed output values, $p_t(\theta_t|y_{t_1},y_{t_2},\ldots,y_{t_T})$, with t_T being the last observation time before time t. Given a filtering time interval $[0,\tau]$ and a given number H of observation times, it is then possible to determine the sequence of observation times $0<t_1<t_2<\ldots<t_{_H}\leq\tau$ that optimises the parameter identification in some way, for example, by providing the thinnest conditional density function. From an information theory point of view, this optimal design problem can be restated in terms of output sensitivity with respect to the parameters. It is then possible to look for the sequence $t_1,\ldots,t_{_H}$ which maximizes some output sensitivity criterion. However, in the case of on-line filtering, experimenters often

prefer to follow a safe step-by-step adaptive approach rather than a full batch approach on a given interval $[0, \tau]$. This led us to consider a sequential optimal time design procedure on a sliding horizon.

A.4 On-line time optimal design algorithm

This approach relies on other sensitivity indices that are easier to compute: the recently developed total sensitivity indices, known as TdSI-VIP, typically with $d = 2$ ([Ellouze *et al.* (2010)] and [Gauchi *et al.* (2010)]), with values between 0 and 100 %, summing to 100 ([Gauchi *et al.* (2010)] for a full definition and method of computation). These indices will be referred to as SIVIP in the text. In this approach, after an observation has been recorded, the SIVIP are systematically computed for all next possible future times of observation. Such indices are computed for each parameter, leading to a bundle of p sensitivity curves. The next optimal observation time is then chosen as the time where the first maximum occurs, regardless of the curve. After the new observation has been recorded, a new set of SIVIP for all next possible future observation times is computed, giving a new bundle of sensitivity curves. The next observation time is then determined as previously, and so on. See the illustration given in chapitre 6.3.2 for the Baranyi-Roberts model filtering.

A.5 A nonparametric particle estimation of a Bayes factor

The Bayes factor ([Jeffreys (1961)]; [Kass and Raftery (1995)]) is one of the most efficient tools to discriminate between two competing models. However, its estimation can be troublesome and is now generally performed through Monte Carlo Markov Chain (MCMC) procedures ([Bartolucci *et al.* (2006)] for a review). Moreover, these procedures rely on the knowledge of the output variable probability density function of each model and are generally intractable when these densities are not available, as in the case of the nonlinear state-space models that we consider here. As we will see, the nonparametric particle filtering approach can easily overcome this drawback and leads to a consistent estimate of the Bayes factor (BF) between two state-space models. This nonparametric particle BF estimation as been described in [Vila and Saley (2009)].

A.5.1 The Bayes factor: an overview

Let M_1 and M_2 be two competing models and Θ_1 and Θ_2 their respective parameter spaces. Given an observation set $Y = y_{t_1}, \ldots, y_{t_H} = y_{t_1:t_H}$, the BF between M_1 and M_2 is defined as:

$$B_{12} = \frac{p_1(Y)}{p_2(Y)} \quad \text{with} \quad p_i(Y) = \int_{\Theta_i} p_i(Y|\theta)p_i(\theta)d\theta \quad i = 1, 2 \tag{A.11}$$

where $p_i(Y|\theta)$ and $p_i(\theta)$ are the likelihood function and the parameter prior density for model M_i, $i = 1, 2$. $p_i(Y)$ is the marginal likelihood of model M_i. As can be seen, it is also the normalisation constant of the posterior density $p_i(\theta|Y)$.
The decision rule is to select model M_1 when $B_{12} >> 1$, and model M_2 when $B_{12} << 1$. See [Kass and Raftery (1995)] for an interpretative scale of this ratio.

A.5.2 A non-likelihood-based BF estimation

This estimation relies on independent nonparametric estimations of both terms of the ratio B_{12}. According to [Dawid (1984)], let us first note that for each model:

$$p(Y) = p(y_{t_1})\Pi_{j=1}^{j=H-1} p(y_{t_{j+1}}|y_{t_1:t_j}).$$

Under the assumptions of Theorem 3.1, as n tends to infinity, it can be shown ([Vila (2012)], [Vila and Saley (2009)]) that an almost sure convergent nonparametric particle estimator of $p(y_{t_{j+1}}|y_{t_1:t_j})$ is given by:

$$p^n(y_{t_{j+1}}|y_{t_1:t_j}) = \frac{1}{n}\sum_{i=1}^{n} K_{h_n}^{y}(y_{t_{j+1}} - y_{t_{j+1}}^{i}),$$

in which the $y_{t_{j+1}}^{i}, i = 1, \ldots, n$ are particles generated as in the R-CF algorithm.
Under the same assumptions, for each model, an a.s. nonparametric convergent estimator of its marginal likelihood, $p(Y)$, is then given by:

$$p^n(Y) = p^n(y_{t_1})\Pi_{j=1}^{j=H-1} p^n(y_{t_{j+1}}|y_{t_1:t_j})$$

Now let $B_{12}^n = \frac{p_1^n(Y)}{p_2^n(Y)}$, where $p_1^n(Y)$ and $p_2^n(Y)$ are the marginal likelihood nonparametric particle estimators of model M_1 and model M_2, respectively.

Theorem 8 *Under the assumptions of Theorem 3.1, as n tends to infinity, B_{12}^n is an a.s. convergent nonparametric particle estimator of the Bayes factor B_{12} between model M_1 and model M_2.*

Proof: [Vila and Saley (2009)].

Remark 9 *This convolution particle filtering estimate of the Bayes factor relies on a sufficient summary statistics - the data themselves - and is not penalised by a lack of confidence similar to that which usually impairs ABC Bayes factor estimates ([Robert et al. (2011)]).*

See in Section 5.2 an application to microbiological model comparison between the Baranyi-Roberts model ([Baranyi and Roberts (1995)]) and the Rosso model ([Rosso (1995)]).

A.6 Conclusion

The probability distribution functions of the output variables of a dynamic system modelled by a hidden Markov chain or a nonlinear state-space model are frequently unknown. This limits the use of powerful statistical approaches of system identification such as particle filtering, as well as that of parameter inference methods (e.g., likelihood ratio test) or model comparison methods such as likelihood-based informational criteria (AIC, BIC, etc.) and MCMC-based Bayes factor estimation. To restore the interest of these methods in this context as much as possible, we considered combining them with a recent nonparametric particle filtering approach. We showed, in particular, how a Bayes factor, one of the most powerful model discrimination tools, can be consistently estimated by using this nonparametric approach. Other statistical issues such as optimal design of observation times, state variable probability density predictions and parameter change detection tests have also benefited from this coupling in the same context.

Appendix B

Developer Guide

B.1 Package structure

- Package structure is described in file `Contents.m`.
- Installation, way of use, available examples are described in file `ReadMe.txt`
- Changes in versions are explained in file `Changes.txt`
- Identification, authors and contributors are cited in the file `DOC/aboutFILTREX.txt`

B.2 Checking installation

Actual and expected results can be compared by executing programs located in `RUNTESTS`.

- **Check task *Parametric Identification***

  1. The program `RUNTESTS/testIdentification.m` launches the task on previously saved projects (files in `EXAMPLES/PROJECTS/IDENTIFICATION` whose name include "test") and compares actual results and saved results. The output of this comparison appears on the command window. Checking process may be time consuming.
  2. The files located in the directory `RUNTESTS/RESULTTESTS` contain the actual results. They can be destroyed if no difference is found. Otherwise, restore them on one hand, restore the corresponding reference files, on the other hand, and compare the output.

- **Check task *Dynamics Comparison of two Models with the Bayes Factor*** Same procedure as above, by replacing `RUNTESTS/testIdentification.m` by `RUNTESTS/testCompar.m`.

- **Check task *Simulation of an Optimal Sequential Sampling***
  Same procedure as above, by replacing `RUNTESTS/testIdentification.m` by `RUNTESTS/testStrat.m`. Warnings of type:
  "The time X is too close to the preceeding one ..."
  can be ignored.

B.3 Make a change

Each change in the source code must be described in the file `Changes.txt`. Version number must be increased in the files `FILTREX.m` and `DOC/aboutFILTREX.txt` (Careful: the version is indicated twice in this file). Eventually, add your name into the list of contributories.

B.3.1 Model parameter valid or default values

Modify the file `SRC/ROUTINES/init_Basemodeles.m` (see paragraph 4, in Section B.4.1).
For example, to modify the valid bounds of the parameter `mumax` in the model `BaranyiRoberts`, change the value of `DF.BaranyiRoberts.mumax`. To modify its default values, change the value of `init.BaranyiRoberts.mumax`.

B.3.2 CV valid or default values

Modify the file `SRC/ROUTINES/init_CVs.m`. The valid bounds are `CVs.DF` and the default values are `CVs.valparamS`.

B.3.3 Filter parameter valid or default values

Modify the file `SRC/ROUTINES/init_Filtre.m`. The valid bounds are `Filtre.DF` and the default values are `Filtre.valparamS`.
The parameters of the task *Simulation of an Optimal Sequential Sampling* are defined in the file `SRC/STRATEGY_SIMULATION/private/init_Simul.m`. The valid bounds are `Simul.DF`. The default values are `Simul.valparamD`.

B.3.4 Default folder of the user files

In the file `startup.m`, the variable `mydir` is set to the default top folder of DATA and PROJECTS. It is initialized to EXAMPLES. To distinguish your own files from the provided ones, set it to USER.

B.4 Make an addition

B.4.1 Add a model

1. **Create the Matlab program**
   Create a file `<modelname>.m` in `SRC/DYNAMICS`: it should be the program which calculates the model and its derivative relatively to time. The file `BaranyiRoberts.m` can be a pattern.

   Note:

   - if the model returns $\ln(N_t)$ (neperien logarithm of N_t), `<modelname>` should begin with "Ln";
   - if the model returns $log_{10}(N_t)$ (decimal logarithm of N_t), `<modelname>` should begin with "Log";
   - if necessary, modify the file `SRC/ROUTINES/model_decroit`: the function should return 1 when the model is decreasing, 0 otherwise.

2. **Create the C program**
   Create the C program which calculates the model in a file `C<modelname>.c` located in `SRC/STRATEGY_SIMULATION/C`. The file `CBaranyiRoberts.c` can be a pattern. No need to calculate derivative.

   Note: this file is required for simulating an optimal sequential sampling, only.

3. **Create the pdf model description**
   In `DOC/DYNAMICS`, create a pdf file, named `<modelname>.pdf` which lays out the model equation and plots its general form. It is intended to help the user to choose a model when he clicks on the button *Dynamics > Help about Equation model*.

   Note: if no such file is found, a message is issued but without consequence on execution.

4. **Define default and valid range parameter values**
   In file `SRC/ROUTINES/init_Basemodeles.m`, add the model and its parameters. For each parameter, you have to indicate:

   - the range of its valid values in `DF.<modelname>.<parametername>`
   - the range of its default values in `init.<modelname>.<parametername>`
   - its default values, when it is fixed in `valparamS.<modelname>.<parametername>`
   - its noise, in `initbruit.<parametername>`
   - its default unit, when it is time dependent, in `type.<parametername>`

5. **Define the valid range of response value**
   Model output can be restricted inside given bounds when simulating an optimal sequential sampling, by Sobol-Saltelli method.
   See file `SRC/STRATEGY_SIMULATION/private/init_modeles_repvalides.m`.

B.4.2 Add a task

1. Create the Matlab programs in a subfolder of `SRC`
2. Add the title of the task into the choice menu of `FILTREX.m`
3. Add a call to the task main program into `SRC/GO.m`