ParmVar – A tool for studying the variations of parameter estimates in StochSD models

Leif Gustafsson and Erik Gustafsson.

1. Introduction

ParmVar is a tool for statistical analysis of estimated parameters for stochastic models created for StochSD. The use of ParmVar is restricted to stochastic models producing the same expected output as their embedded deterministic models.

Introducing stochastics to the model implies that a study must include a number of simulation runs to analyse the results by statistical means.

The results from a model study can be of two kinds: **A)** statistical analysis of *model output* and **B)** statistical analysis of *parameter values* (*model input*). **Case A** is handled by **StatRes**, which analyses how output quantities vary for a given stochastic model. **Case B** is handled by **ParmVar** (Parameter Variations). ParmVar analyses how *estimated* parameters[1] vary depending on which simulation run (replication) is used to estimate the parameters of a given stochastic model.

After optimisation or parameter estimation (using e.g. Optim), you may ask how accurate the estimation of the parameters was. With other words you want confidence intervals around your parameter estimates. This can be achieved by using ParmVar on a stochastic version of the model used for the optimisation or parameter estimation performed with Optim.

Statistical analysis of parameter estimates is more complicated and computer intensive than statistical analysis of output quantities. Statistical analysis of output quantities requires *N* replications of the model. But single parameter estimation requires *M* replications with e.g. **Optim**. Thus, to obtain statistics from *N* parameter estimates require *N*M* replications. Functionally, **ParmVar** is a combination of **Optim** and **StatRes**. Based on *N* replications of a given stochastic model, the Optim part makes parameter estimation for each replication, where after the StatRes part performs statistical analysis of the *N* parameter estimates.

In Figure 1, the parameter estimation part is shown.

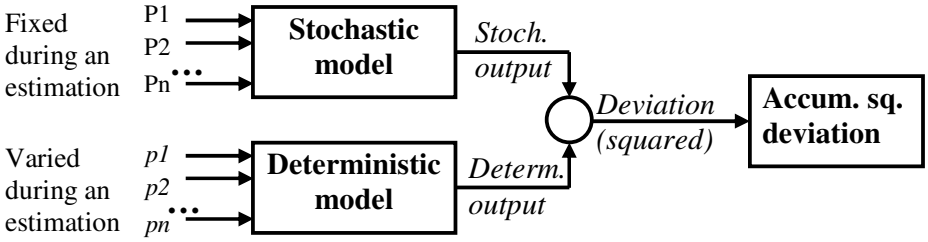

Figure 1. The parameter estimation part. The stochastic model (with fixed, already estimated parameters P1, P2, … Pn) represents the system under study. Although these parameters are fixed, the *stochastic model* can exhibit different behaviours (outputs). From the view of the *deterministic* model,

[1] Also initial values of compartments or stages can be regarded as a parameters.

an optimal fitting to the stochastic output will result in a set of parameter estimates (*p1', p2', … pn'*) that differs over the *N* simulation runs with the stochastic model. With *N* sets of estimates, it is time for the statistical analysis.

This approach contains an important condition *that is not always satisfied*. It requires that the stochastic model is expected to produce the same output as the deterministic model. This is *often* the case for two classes of models: Linear models and Models with large numbers; see [2]. Before using ParmVar, you should check (with e.g. StatRes) that the stochastic and its embedded deterministic models are consistent. To make it simple, we describe the problem with the help of a small example.

Further, the approach used in ParmVar could be improved in various ways. For example, it could include the use of co-variations between the studied parameters.

Example: Variations in parameter estimates in radioactive samples

Assume that a sample on average contains $X0_{Syst}$ radioactive atoms of a certain kind that decay with a time constant T_{Syst}. Each time we study such a sample we get a slightly different result because the number of radioactive atoms is only approximately $X0_{Syst}$ and the number of decays during a study time will only give a time constant of approximately the value T_{Syst}.

Now assume that we have *only one sample* that we can study. We can then fit a deterministic model that generates an exponential decay with the parameters X0 and T to describe the observations of the system under study. By tuning the parameters, *X0* and *T*, we can obtain a best fit of the model to the system under study. The parameter estimates of *X0* and *T* are of course not identical to $X0_{Syst}$ and T_{Syst}, but they are the best estimates we can get from a single experiment (or data set).

To judge the precision of the estimates *X0* and *T* it would be valuable to know how much *X0* and *T* would vary *if we could* repeat the system experiment a number of times and thus calculate standard deviations and confidence intervals for these estimates.

However, if we have the physical knowledge of initial value of the number of radioactive atoms varies over the samples and the nature of the process, we can make a *stochastic* model that replaces the system. Then we can perform repeated experiments on the stochastic model and for each of these experiments estimate *X0* and *T* by fitting our *deterministic* model to the stochastic one. Provided that this stochastic model is a good description (dynamically and stochastically) of the real system under study the estimates of the model variation around *X0* and *T* is about the same as the real variations would have been around $X0_{Syst}$ and T_{Syst}. This is what can be achieved with ParmVar.

The **model file** should then contain:
1) A physically trustworthy *stochastic model* of the system under study (that displays the variations in behaviour in repeated experiments).
2) A corresponding *deterministic model* with parameters (here *X0* and *T*) to be estimated.
3) An *objective function* that measures how well the deterministic model fits to the stochastic model for different parameter values. The model fitting procedure is a minimisation of this objective function.

The fitting procedure is performed by tuning the values of the deterministic model parameters so that this model behaves as similarly as possible to the behaviour of the stochastic model (for each experiment). By repeating this fitting procedure *N* time, we obtain a probability distribution of the studied parameters, *X0* and *T*. Thereafter, usual statistic methods are used to calculate variations and confidence intervals for the parameters, *T* and *X0*. In Section 3, we will demonstrate how this is performed in StochSD using ParmVar.■

Technically and functionally, ParmVar is composed of the simplex optimiser used in Optim combined with the statistical devises used in StocRes.

2. The user's interface of ParmVar and its features

The ParmVar user's interface is shown in Figure 2.

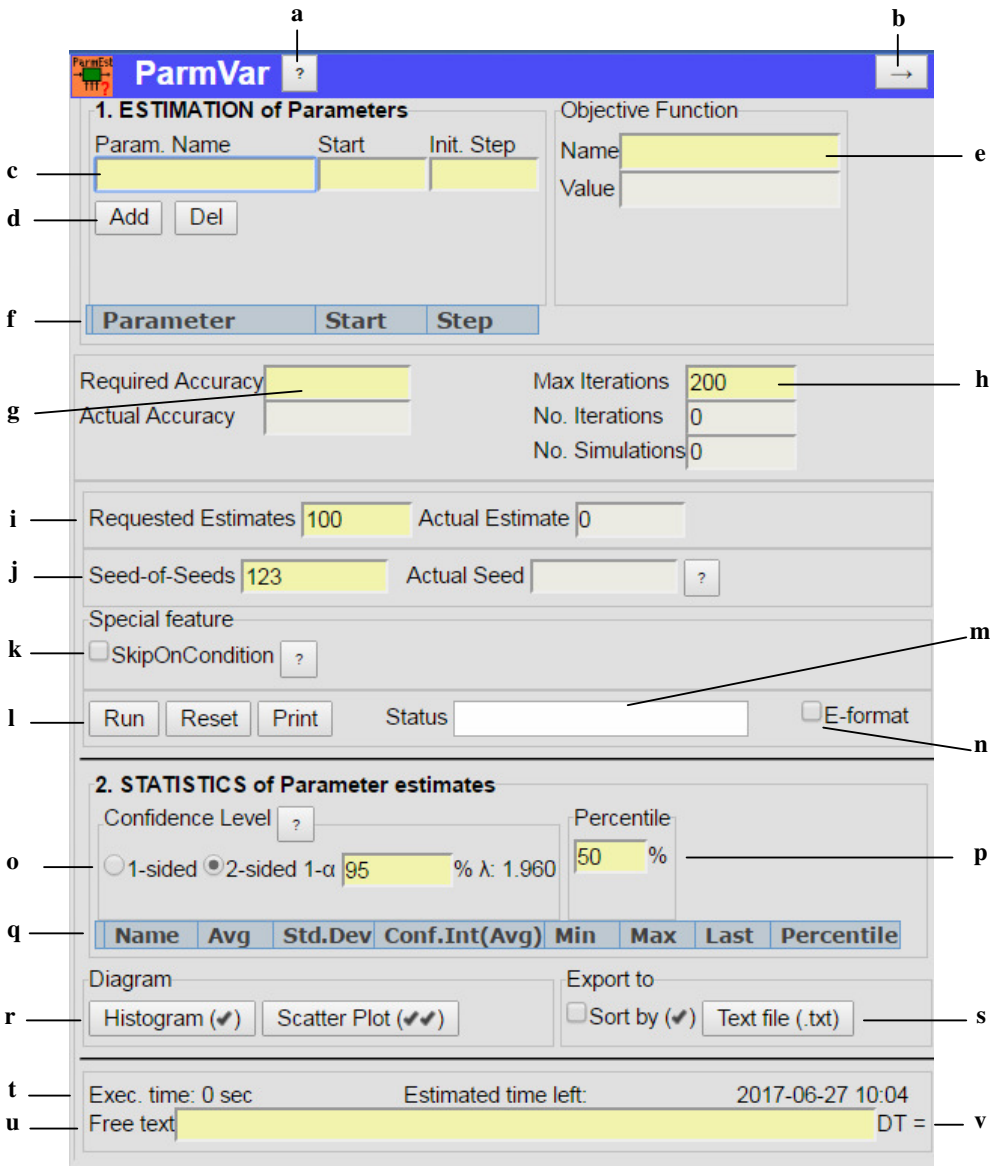

Figure 2. The ParmVar user's interface.

Figure 2 shows, the user's interface as described in more detail below. (An overall rule is that you may/can only specify the conditions by pressing buttons and filling in textbox fields with a light yellow colour.) To understand this interface, it must be kept in mind that the underlying model file is composed of three parts: **1)** The stochastic model, 2) The deterministic model, and 3) An objective function describing the cumulated squared differences between the deterministic model and the stochastic model.

a) On the top bar, you find a Help button (**?**) that briefly describes how to use ParmVar.

b) To the right you find the button (→) that minimises ParmVar.

c) Each parameter in the deterministic model to be fit must be specified by 'Param. Name', 'Start' value and 'Init. Step'. The 'Start' value tells ParmVar where to start the parameter search and 'Init. Step' tells how much the parameter is (initially) changed in each direction in the parameter space.

d) Press the **Add** button for each specified parameter. This will check if the specified parameter exists in the model and, if so, add it to the Parameter grid. The **Del** button enables removal of already specified parameters.

e) The Name of the Objective function in the model file must be specified. The resulting Value is delivered from the minimising fitting process (and should thus *not* be filled in by you).

f) In the 'Parameter grid' you will see the specified and added parameters, their initial values and initial step sizes. The very first field of the Parameter grid can be empty or hold a check mark '✓'. This is automatically inserted for the last added item and cleared by the **Run** button. You may check *several* parameters. The check mark is needed for several purposes. For example the **Del** button operates on checked rows only. A check-marked parameter can also, after the estimation process, be studied in a Histogram or in a Scatter Plot (requires two checked parameters).

g) Below the Parameter grid you present the breaking conditions by specifying the 'Required Accuracy' of the fitting process. In the field below you can follow the development of the 'Actual Accuracy' during the fitting process.

h) 'Max Iterations' will stop the fitting process if it requires more than the specified number of iterations (in a simplex optimiser, each iteration takes one to several simulations). Also the number of performed simulations is shown.

i) In 'Requested Estimates' you specify the number of Parameter Estimates you want to create (default: 100). In the 'Actual Estimate' you can see the number of parameter estimation rounds performed so far.

j) During a parameter estimation round, the parameters in the deterministic model are tuned during a large number of simulation in order to make the deterministic model's behaviour resemble that of the stochastic model. It is then important that the stochastic

model behaves the same during each simulation. This is accomplished by using the same **seed** in the model.

For the next parameter estimation round a new seed is required. This means that we for *N* parameter estimation rounds will need *N* seeds. ParmVar therefore has its own random number generator (RNG1) that delivers these *N* seeds to the models RNG2. ParmVar's generator is initiated by a **Seed-of-Seeds** (default value 123). See Figure 3.

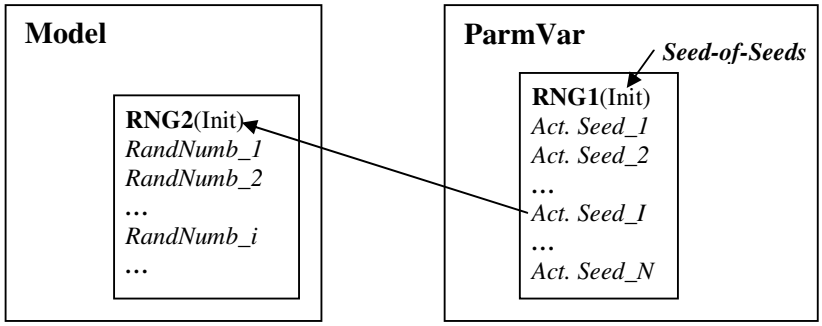

Figure 3. The Seed mechanism.

k) Checking the 'SkipOnCondition' box enables you to use a variable with the name [SkipOnCondition] that must be included in the model. This variable should define a condition. If this condition is TRUE at the end of a simulation run, this run will be skipped and replaced by a new run. This is handy if you for some reason want to exclude certain outcomes e.g. runs where nothing of interest happens. (See the StatRes manual for an example of using this feature.)

l) With the buttons: **Run**, **Reset**, and **Print** you can control the execution and print the results:
- **Run**] (actually a combined **Run/Halt/Continue** button) starts the simulation run sequence. **Halt** halts the run sequence after a full estimation round. When the number of 'Requested Estimates' is performed the specified process is complete. By increasing 'Requested Estimates' you may press this button again (which is now labelled **Continue**.)
- **Reset** resets the specifications to the state before **Run** was last pressed.
- **Print** prints the form as is.

m) The status during and after the estimation process ('RUNNING, 'HALTED, 'DONE') is presented in the Status field to the right of the buttons.

n) To the far right you also find the checkbox 'E-format'. Default is unchecked which gives maximal numerical information as a decimal number. If 'E-format' is checked the results are presented as: ±X.XX E±XX (where E±XX stands for $10^{\pm XX}$).

o) The 'Confidence Level' frame allows you to specify 1- or 2-sided confidence intervals for the Average estimates, and in the '1-α' textbox you can specify the Confidence Level in per cent. (Default 2-sided interval and 95 %.) The corresponding λ-values are automatically calculated. (You may also change these options after the simulations.) A Help button (**?**) gives the theoretical background.

p) In the 'Percentile' frame you can specify any percentile between 0 and 100 %. Default is 50% which means the median value. (Also afterwards you may change the percentile.)

q) In the Results grid you will see statistics of the *N* estimates of each parameter. For a parameter, X, you will see: Name (X), Av(X), Std.Dev(Avg(X)), Min(X), Max(X), Lasr(X) and Percentile(X) from the *N* fitting rounds.
By setting the Percentile to e.g. 2.5% and 97.5% you will obtain the lower and the upper limits of a 95% confidence interval for the variations of the estimated parameters (not for their averages).

r) The two buttons: **Histogram (✓)** and **Scatter Plot (✓✓)** present the results of specified parameters graphically. The number of check marks on these buttons tells how many parameters have to be checked. To display a Histogram, one parameter is required, and to display a Scatter plot two parameters are required. The check mark is included in the first column of the Results grid.

s) To analyse the results further with other tools, there is a device to export the results to a text file (*.txt). By checking '□ Sort by (✓)', you decide if the data should be arranged in the order of the estimation rounds (Unsorted) or in ascending order (Sorted by the checked parameter). You can then send all data to a specified text file. The model-file name, date and time etc. are then sent to a Text file (*.txt), followed by the data results of all parameters in columns. If 'Sorted by' is checked, the data of this checked parameter will be presented in ascending order. All parameters from each run are presented together on the same row in the text file.

t) The 'Execution Time', 'Estimated time left' for the process and Date & Time are also displayed.

u) There is a Free text field for your notations.

v) The time-step (DT) used in the StochSD model is automatically documented.

3. An example for ParmVar (continued)

Return to the Example of a radioactive sample in Section 1. By observing the decays occurring over time, we could represent the decay process in a time graph. But since we can only count the decays during a certain time period, we don't know the exact values of the initial number of atoms $X0_{syst}$ and of the decay time constant T_{syst}.

Knowing the physics behind the studied system, we can build a *deterministic* model of the decay process and use **Optim** to estimate the values of the initial number of atoms as $X0_{syst}$ and the time constant as T_{syst} from counting the decays of the observed sample. Here we will assume that we then obtained the estimates of $X0_{syst}$=100 and T_{syst}=12.00, respectively.

Now we also want a measure of *the precision* of the estimates of $X0_{syst}$ and T_{syst}. If we could repeat the physical experiment *N* times, we could also estimate *the variations* in the estimates of $X0_{syst}$ and T_{syst}. But when this is not possible, the next best way is to construct a *stochastic model* of the physical system under study and do the experiments on this model.

Therefore, we construct a *stochastic model* of the system and transfer our estimates $X0_{syst}$ and T_{syst} to the corresponding parameters $X0_{stoch}$ and T_{stoch} of this model. Since this model is stochastic it will behave differently each time it is simulated. What we here are interested in is how the stochastic model varies in repeated simulations. Or more precisely, what effect would these variations have on new estimates of $X0_{stoch}$ and T_{stoch}?

By including both the *stochastic model* with the fixed estimated values $X0_{stoch}$ and T_{stoch} and a *deterministic model* with the same structure and undecided parameters X0 and T, we can repeat model experiments a large number of times. Now we use ParmVar to fit the outcome of the deterministic model to the outcome of the stochastic model in order to obtain a new and independent estimates of $X0_{stoch}$ and T_{stoch}. After doing this *N* times, we get *N* new results that are then analysed statistically to obtain e.g. confidence intervals for the original estimates of $X0_{stoch}$ (=100) and T_{stoch} (=12). The StochSD model file for ParmVar is shown in Figure 4 (where we now replace the subscript '$_{stoch}$' with 's'):

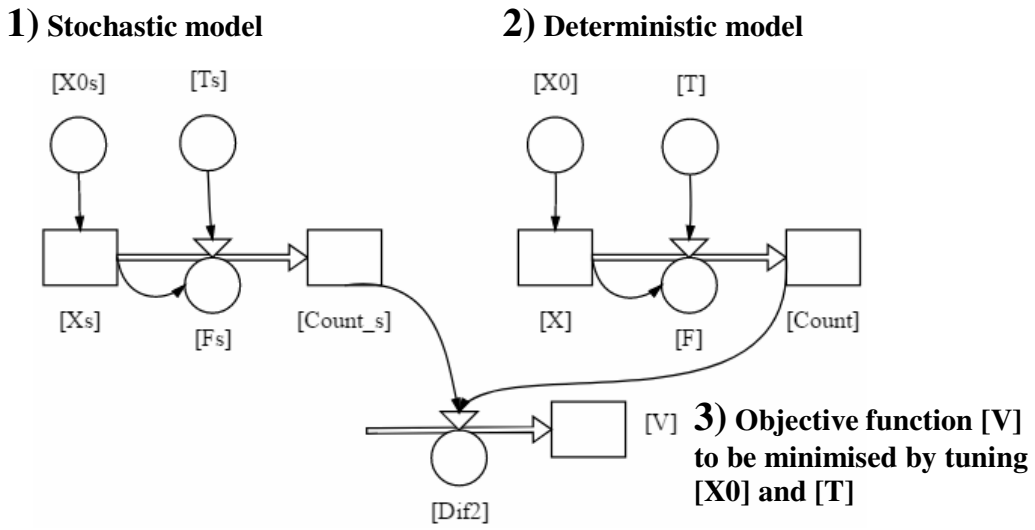

Figure 4. A model file for ParmVar to analyse the uncertainty of previous estimates $X0_{syst}$=100 and T_{syst}=12 from an observation of a physical sample. This is performed by studying the variations of a stochastic model of the sample where the parameters [X0s] and [Ts] are assigned the previously estimated values 100 and 12.

As seen from figure 4, the model file contains 1) a stochastic model of the radioactive decay process, 2) a deterministic model of it and 3) an objective function comparing the recorded number of delays.

During each parameter estimation round, the parameters *X0* and *T*, are tuned to minimise the difference between the deterministic and stochastic models behaviours in a least square sense. Technically, for each time-step this difference is squared and accumulated into the objective function *V*, whose value is returned to ParmVar after each simulation run.

The stochastic model will behave differently in the *N* parameter estimation rounds depending on the value of the seed (delivered from ParmVar). This gives *N* new estimates picked up by the parameters *X0* and *T*.

Finally, these *N* estimates are statistically presented in terms of Average, Standard Deviation, Confidence interval of the average, Min, Max, Last and Percentile.

In a **StochSD** model the code is:

The equations of the **stochastic model** are:

Initialisation phase:

[X0s] = RandPoisson(100)	(Previously estimated value by Optim.)
[Ts] = 12	(Previously estimated value by Optim.)
[Counted_s] = 0	

Run phase:

[Xs] = [Xs] – DT()*[Fs]	
[Fs]=RandPoisson(DT()*[Xs]/[Ts])/DT()	(Or shorter: [Fs]=PoFlow([Xs]/[Ts]).)
[Counted_s] = [Counted_s] + DT()*[Fs]	

The equations of the **deterministic model** are:

Initialisation phase:

[X0] = *initial guess*
[T] = *initial guess*
[Counted] = 0

Run phase:

[X] = [X] – DT()*[F]
[F] = [X]/[T]
[Counted] = [Counted] + DT()*[F]

The **objective function**, [V], accumulates the squared differences between [*Counted_s*] and [*Counted*] for each time-step (after initialisation by [V] = 0):

[Dif2]= ([Counted_s] – [Counted])^2.
[V] = [V] + [Dif2]

In Figure 5, the results of *N =100* estimation rounds are presented.

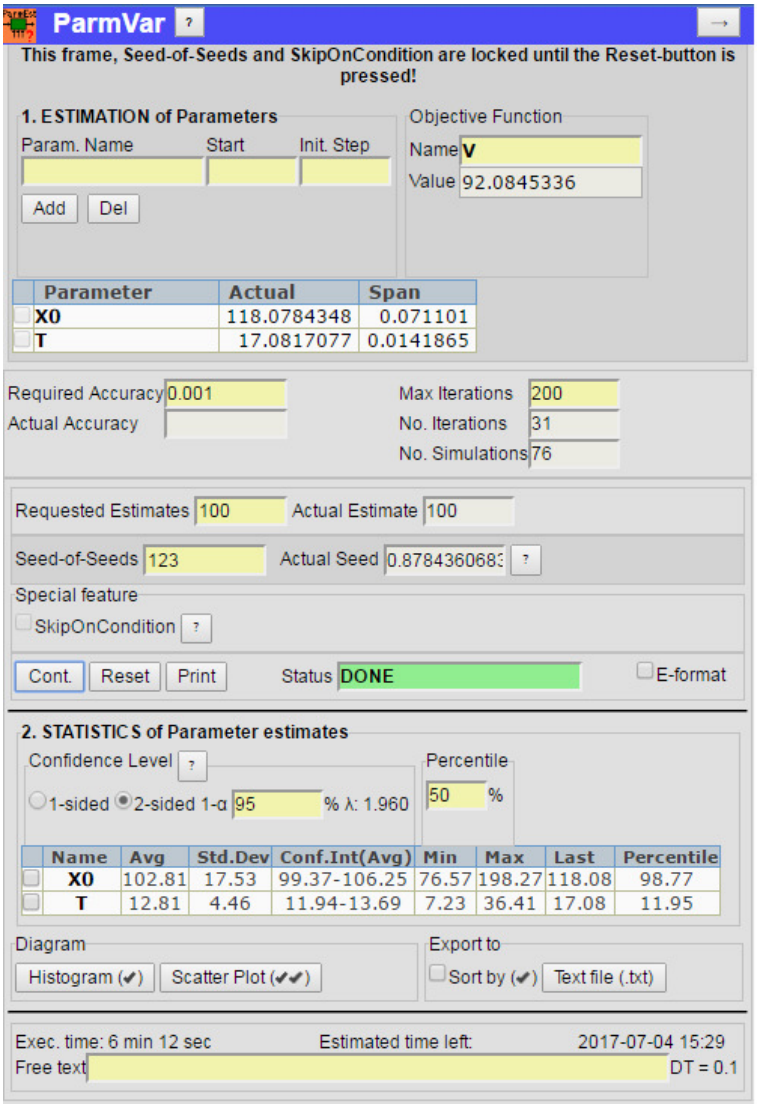

Figure 5. The results after *100* parameter estimation rounds of the radioactive decay model.

From Figure 5, we see the results from the *100* rounds of parameter estimation. The Average estimates [*X0*] = *102.81* and [*T*] = *12.81*. Thus we got small deviations from the original estimates [*X0s*]=*100* and [*Ts*]=*12* from the observation of the sample.

The Standard deviation for [*X0*] was *17.53* and for [*T*] it was *4.46*. The 2-sided Confidence Intervals on 95% confidence level of the Average estimates are *99.4 – 106.3* for [*X0*] and of *11.9 – 13.7* for [*T*].

As you may have observed, there is a difference between the original estimates from a sample and from using a deterministic model to estimate the parameters of the stochastic model ($X0_{syst} - X0_{avg} = -2.81$ and $T_{syst} - T_{avg} = 0.81$). This is probably because the estimating procedure will introduce some bias and also random deviations. However, you can stick to the original estimates $X0_{syst} = 100$ and $T_{syst} = 12.00$ and adjust the confidence intervals around these values by subtracting 2.81 and 0.81, respectively.

In the figure, Min, Max and the median (50:th percentile) are also shown.

A 2-sided Confidence Intervals on 95% confidence level a *of the parameter estimates* (not of the estimated averages) could also be obtained by specifying the Percentile to 2.5% and 97.5%, respectively and reading the Percentile for each parameter. This gave *79.8 – 143.2* for [*X0*] and of *7.9 – 23.14* for [*T*]. (Not shown in Figure 5.)

You can also see a histogram of [X0] and [T] or a scatter plot of [X] versus [T].

At the bottom of the form, you see that it took about 6 minutes of execution time (with a simulation time-step = 0.1 time unit from time 0 to 20 time units, and executed on a 2.2 GHz PC).

4. References

- Gustafsson L. Poisson simulation - A method for generating stochastic variations in continuous system simulation. Simulation, May 2000.

- Gustafsson, L. and Sternad, M. When can a Deterministic Model of a Population System Reveal What Will Happen on Average? *Mathematical Biosciences*, **243**, 28-45, 2013.

- Gustafsson, L. and Sternad, M. (2017) The full potential of Continuous System Simulation modelling, OJMSi, 2017. http://xxx ???

- Law A.M. and Kelton W.D. *Simulation Modeling and Analysis.* McGraw-Hill Inc, New York, 1991, 1982.

- Press W.H., Flannery B.P., Teukolsky S.A. and Vetterling W.T. *Numerical Recipes in Pascal – The Art of Scientific Computing.* Cambridge, UK. Cambridge University Press, 1989. (The book is also available in FORTRAN or C.)

5. Responsibility

The user is fully responsible for the use of this product. The producer and the supplier of this code take no responsibility for the use or functioning of ParmVar.