

User manual for SMP-Tool beta v3.4

September 7, 2022

SMP-tool (Semi-Markov Process tool) is a Matlab application for analyzing Semi-Markov Processes (SMPs). This manual will cover how to install SMP-tool, how to create a model in Stateflow and how to analyze a model using SMP-tool. SMP-tool requires that Matlab with Simulink/Stateflow and Symbolic Math Toolbox is installed. The tool has been created and tested in Matlab version 2021b. Like the tool itself, the manual is currently in a beta version and might therefore miss or only have brief summarized descriptions for some functionalities. For questions and clarification, we encourage the reader to send an email to kaalen@kth.se.

Contents

1	Installation	3
2	Create a Stateflow Model	3
2.1	Specify States	3
2.2	Specify Transitions	4
2.2.1	Non-branching Transition	4
2.2.2	Branching Transition	4
2.2.3	Defining Parameters	5
2.3	Hierarchical Models	6
3	SMP-tool Overview	8
3.1	Open a Model	8
3.2	Introduction to the Graphical User Interface	8
3.2.1	Compare Results	10
4	Analysis of a SMP	11
4.1	General analysis settings	11
4.1.1	Analysis Time	11
4.1.2	Select model type	11
4.1.3	Plot Options	12
4.1.4	Relative Error	12
4.1.5	Sensitivity Settings	13
4.2	Transient Analysis	13
4.2.1	Transient Analysis with Monte Carlo Simulation	13
4.2.2	Transient Analysis with Numerical Solver	14
4.2.3	Transient Analysis of HSMP Models	15
4.3	Sensitivity Analysis	15
4.4	Steady State Analysis	17
4.5	Simulate Sample Path	19
4.6	Display State Transitions	19
5	SMP-tool 3: Parallel process model simulation	20
5.1	Parallel models	20
5.2	Guards	21
5.3	Events	21
5.4	Rules and limitations of parallel models, currently	22

1 Installation

Before installing SMP-tool, make sure that you have Matlab with Simulink/Stateflow, Symbolic Math Toolbox, and Statistics and Machine Learning Toolbox installed. Extract SMP-tool.zip into some arbitrarily chosen folder. Once extracted, simply run the mlappinstall-file titled SMP-tool followed by the file StartUpScript.m to install SMP-tool as a Matlab app.

2 Create a Stateflow Model

An example model, named “highwayPilot.slx”, is included with the SMP-tool download. This model shows what a model created for SMP-tool might look like. This model will be used as an example throughout the manual.

To create a SMP, the initial distribution, all states, all transitions, and the distribution of all transition timers must be known and specified in Stateflow. To create a new model in Stateflow simply start Matlab and type `sfnew` in the command window. In the new window that pops up, double-click on the yellow icon that shows two rectangles with rounded corners and arrows between them. This will open an empty Stateflow Chart. We divide specifying all properties of the SMP into two steps, specifying the states with initial distributions, and specifying transitions with timer distributions.

2.1 Specify States

Use the containers marked **State** (the icon is a rounded off rectangle) on the left-hand side in the Chart-view to represent each state. When the first state is added an arrow pointing to the state will be generated automatically. This arrow represents the initial transition and points towards the initial state. The arrow will only be considered by SMP-tool when performing a simulation of the system.

Replace **Name** in each state with for instance **S1**, **S2**, etc. The name of the states can be chosen to your own liking. Note that comments can later be added to each state in order to give a clear description of what each state represents. Next, let x_i be the probability that state i , denoted S_i , is the initial state of the process. In the field **Add state action** below the name field of each state, write $p = x_i$; **down_state** if the state is a down state and $p = x_i$ if the state is an up state of the process. Comments can be added after the previous specifications to a state by simply writing them after %, e.g. $p = x_i$ %This is a comment.. If nothing else is stated in a state, it is assumed that $x_i = 0$ and that the state is an up state. If no p is specified, the Stateflow initial transition will be used instead. To exemplify the state generation procedure described above, the states of Highway pilot model are illustrated in Figure 1.

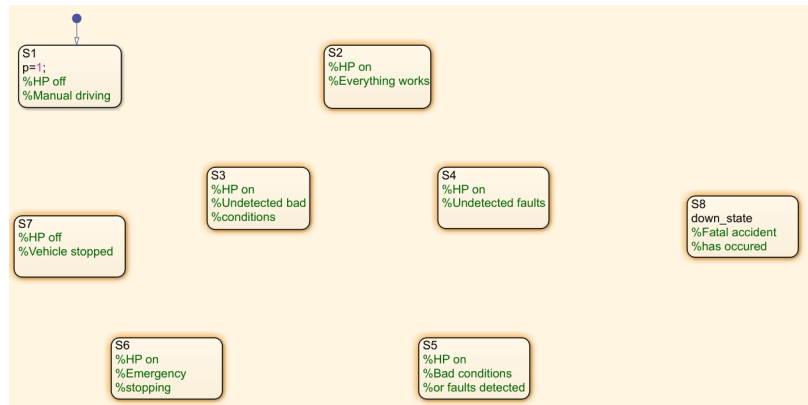


Figure 1: The states of the Highway pilot model.

2.2 Specify Transitions

When each state has been created, the next step is to add all possible transitions to the model.

2.2.1 Non-branching Transition

To create a non-branching transition from a state, right-click on the state and choose “Create Semi Markov Transition” (if no such choice appears go back to section 1 of this manual and run StartUpScript.m as specified there). The window presented in Figure 2 will appear. Choose the destination state of the transition and the distribution of the corresponding timer. Finally, give the transition its required value(s) and its (their) unit(s). Press “Create Transition” to create the transition.

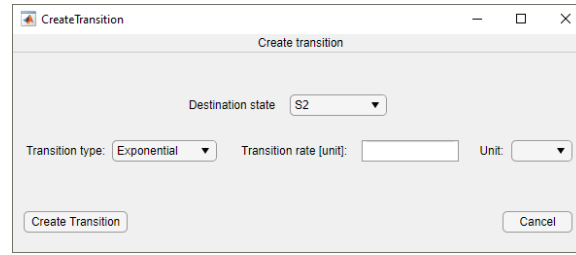


Figure 2: Window for creating a non-branching transition.

2.2.2 Branching Transition

A branching transition is a transition with several possible destination states that all share the same transition timer. Once the timer reaches zero, the destination state is chosen according to the probability of each of the possible destination states. A simple example model with a branching transition is illustrated in Figure 3.

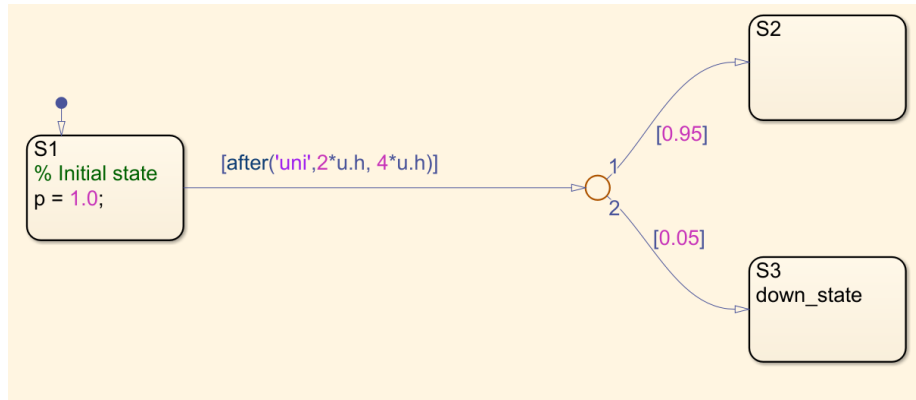


Figure 3: Simple example of a model with a branching transition.

To create a branching transition from a state, right-click on the state and choose “Create Semi Markov Branching Transition” (if no such choice appears go back to section 1 of this manual and run StartUpScript.m as specified there). The window presented in Figure 4 will appear. For each possible destination state, select the state from the drop-down menu and specify a probability. Press “Add destination” to add it to the table to the right. Select the distribution of the timer for this transition and specify the parameters for the chosen distribution. Finally, press “Create Transition”.

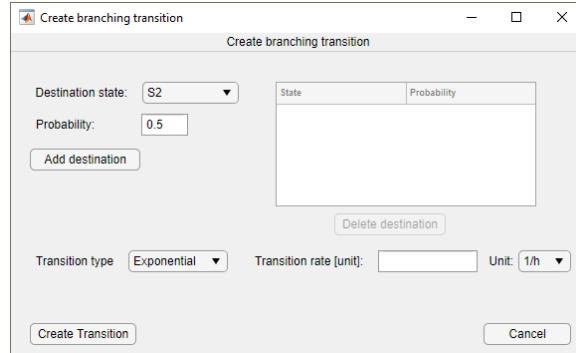


Figure 4: Window for creating a branching transition.

2.2.3 Defining Parameters

With SMP-tool version 2.0, and later, it is possible to create and manage parameters in the “Symbols” panel. Parameters can be used to define transitions, and some parameters are required for the model to be identified correctly by SMP-tool.

If you have followed along and created the model presented in figure 3 there will be a “Symbols” panel right-hand side, similar to what figure 5 shows. Here, the model’s symbols and parameters are collected. So far, the model has three parameters, `down_state`, `u`, and `p`. None of them are properly defined yet, which is why there is a warning sign next to their names. These three parameters are included in every model and should be specified as input data. To do this, press the button next to the warning sign (highlighted in yellow) for each parameter and selecting “Input Data” in the drop-down list.

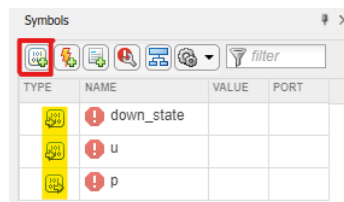


Figure 5: Symbols window.

To create a transition defined by one or more parameters, the parameters must first be created. Simply press the “Create Data” button, highlighted with a red box in figure 5. A new row will appear and by default it is set as “Local Data”. You can change the name of the newly created parameter and give it a value. The yellow exclamation mark will disappear as soon as the parameter is used in a transition or state.

An illustration of how parameters can be used to define a transition can be seen in figure 6. There, the model from figure 3 has been remade with a new branching transition. This branching transition has been defined with two new parameters, `DC_prob` and `uni_value`.

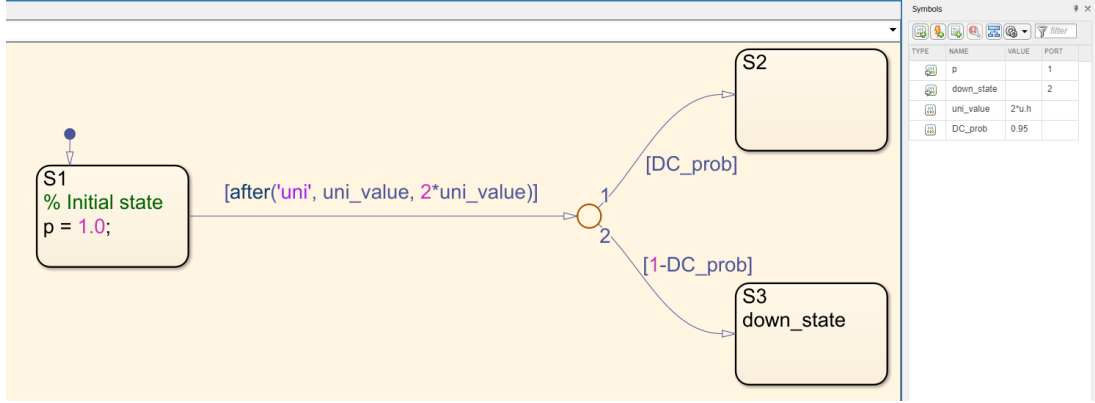


Figure 6: Example model with a branching transition and two specified parameters used to define the branching transition.

Once all states and transitions are set, save the Stateflow model. The previously mentioned example Stateflow model, the one bundles with the SMP-tool download, can be seen in figure 7. Note that this particular model does not contain any branching transitions nor any parameters.

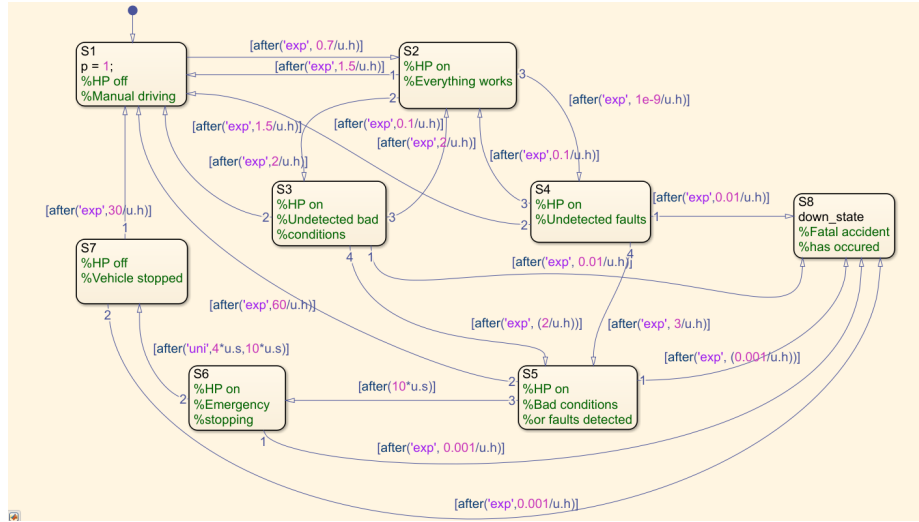


Figure 7: Stateflow model of Highway Pilot, the arcs of the transition arrows have been adjusted for readability after creation.

2.3 Hierarchical Models

SMP-tool supports the analysis of hierarchical semi-Markov process models (HSMP models). An hierarchical model is created by placing states and transitions inside another, hierarchical, state. Simply increase the size of the rectangle for the outer state (gives you more space to work with) and add states and transitions inside the outer state. To be able to enter a hierarchical state, a transition must be made to one of the inner sub-states. Unlike normal state transitions, a transition timer from an hierarchical state is not reset when transition within the states occurs. The process will exit the hierarchical state either when its inner process leaves the state, or its own transition timers fire. Figure 8 shows an example of a hierarchical state.

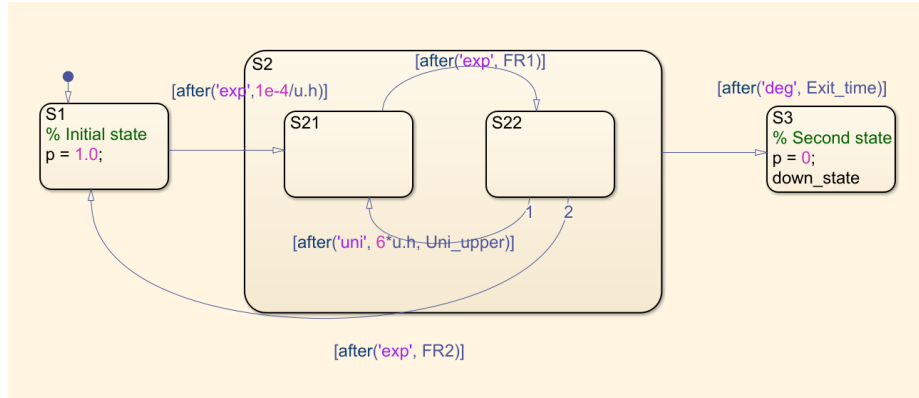


Figure 8: An example HSMP model.

3 SMP-tool Overview

When SMP-tool has been installed, it can be found and launched from the APPS tab in Matlab. In the APPS section there is a drop-down menu that contains all the installed apps. Find SMP-tool and press to launch it. When launched, the main user interface displayed in figure 9 will appear.

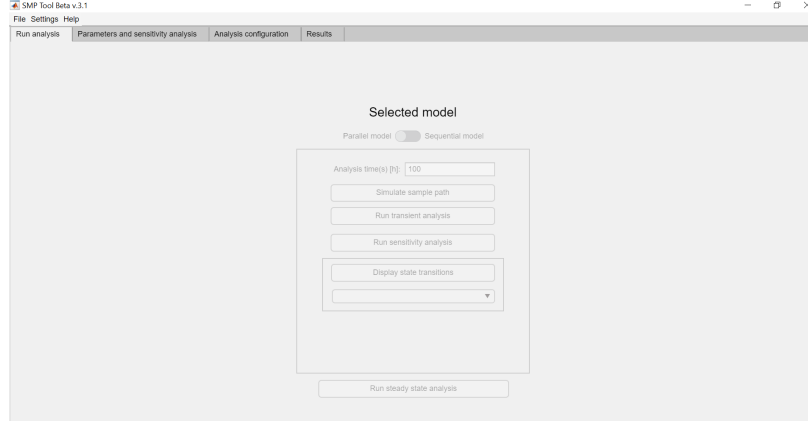


Figure 9: Graphical user interface for SMP-tool.

Before we get into all the different tabs of SMP-tool, let us look at how to load a model into SMP-tool.

3.1 Open a Model

To load a model into SMP-tool, press the “File” button in the top left corner, see figure 9. A drop-down menu will appear where you can choose to open a new model, open an existing model, or exit the app. To open an existing model, hover over “Open” and then press “Existing model”, see figure 10. After that, simply brows to the location of your model and open it. If you have opened the model before it will be available under “Recent Models”. If you choose a new model, an empty Stateflow model will be opened. Design your model, save it and load it into SMP-tool as described previously.

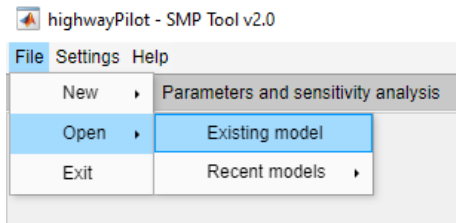


Figure 10: The files menu of SMP-tool.

3.2 Introduction to the Graphical User Interface

Figure 11-13 present the three main tabs of SMP-tool. We will now give a brief overview of each tab. A more detailed description will follow later.

The “Run analysis” tab seen in figure 11 is where all different analyses are started. They will be described in more detail in section 4. The analysis times are also specified here, but they only affects the analyses that are inside the same box as the analysis times’ field, as can be seen in figure 11.

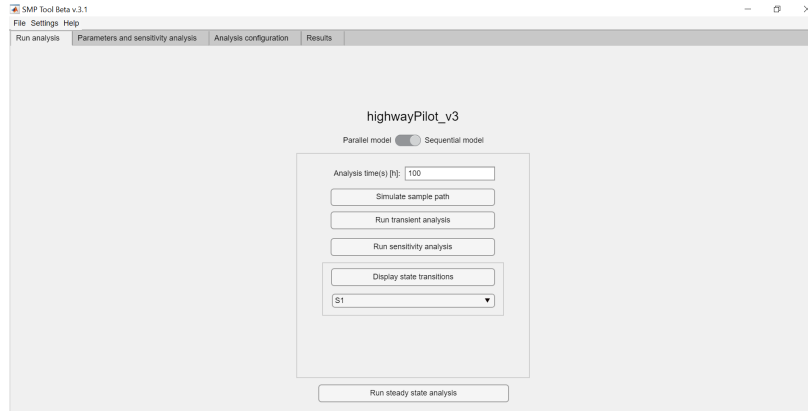


Figure 11: “Run analysis” tab from SMP-tool.

Under “Parameters and sensitivity analysis”, shown in figure 12, the parameters of the model can be changed. It is also possible to add or delete parameters from the table, as well as change the unit or the name. Different choices for sensitivity analysis can be made from this tab and the ranges of the parameters for the sensitivity analysis can be changed. Here you must also choose which parameters that should be included in the sensitivity analysis. In the lower corner there are two buttons that makes it possible to export and import parameters, for instance to save or load custom parameter ranges. Note that the parameters used here has to be specified in the Stateflow model.

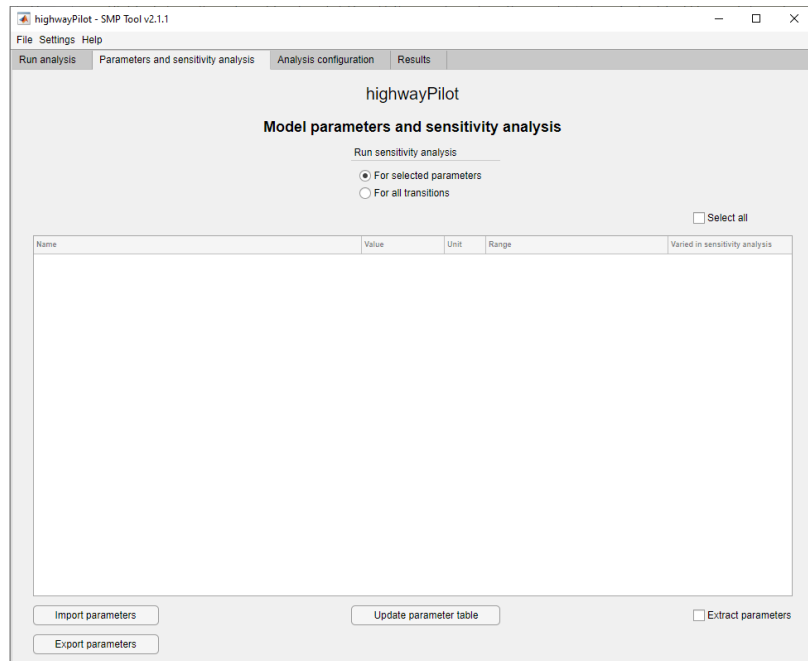


Figure 12: “Parameters and sensitivity analysis” tab from SMP-tool.

Finally, the “Analysis configuration” tab shown in figure 13 houses settings for the solver, inversion algorithm etc. It is also possible to change some plot settings.

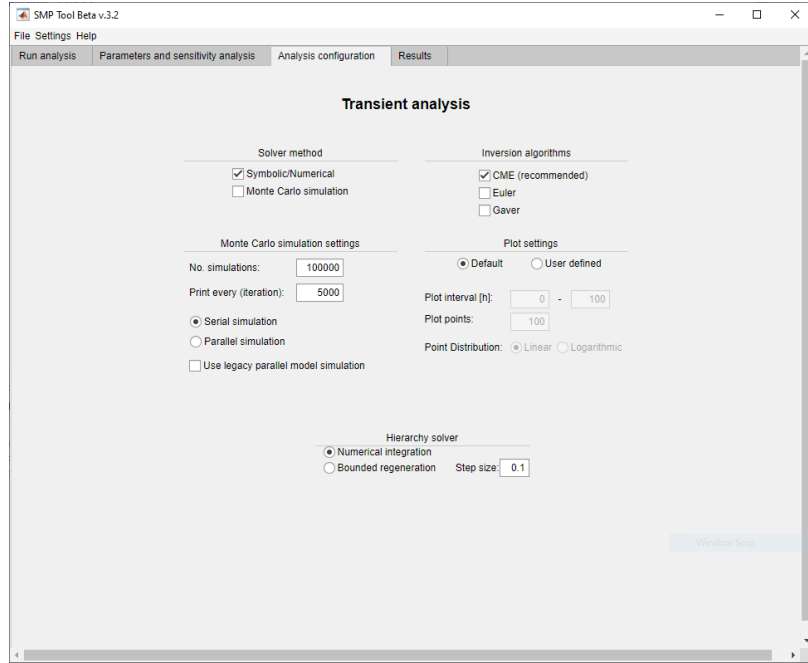


Figure 13: “Analysis configuration” tab from SMP-tool.

3.2.1 Compare Results

It is possible to compare the results from two or more different runs. To be more specific, only the results from transient analyses can be compared to each other (regardless of solver method). To compare results, tick the “incl.” tic box in the bottom right corner to include the data from that result tab in the comparison. Then navigate to the result tab where you want the comparison to happen and press “compare”. To remove the comparison, simply press “remove compare”.

4 Analysis of a SMP

SMP-tool can perform several types of analyses. To get a better understanding of the theory behind these analyses and to get clear definitions of some quantities, the reader is recommended to study [1], [2] and [3]. Before we go into more detail on each type of analysis, we will briefly go over some general settings.

4.1 General analysis settings

This section introduces some settings for the different types of analyses. Note that some settings do not apply to all types of analyses.

4.1.1 Analysis Time

The analysis time, which is the time in hours that the model should run for, affects transient analysis and simulation of sample path. Multiple times can be specified by separating the times with comma, see figure 14. The given analysis time is not taken into consideration for steady state analysis.

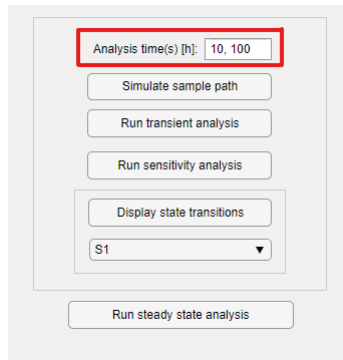


Figure 14: How to specify several analysis times at once.

4.1.2 Select model type

The switch shown in figure 15a indicates to the user if the model is parallel or sequential. When a model is loaded into the tool, the switch will automatically be set to the type that suits the loaded model.

If a sequential model is loaded in the tool and is updated by the user in Stateflow to be a parallel model, the switch will be set to parallel when the user presses any button in the tool. If the user instead converts a parallel model into a sequential one the user will only get a warning since it is possible to analyze a sequential model with the "Parallel model" switch option selected, but not the other way around. If the model is parallel or the user selects the "Parallel model" switch option several buttons will be disabled, see figure 15b. Read more about parallel models in section 5.

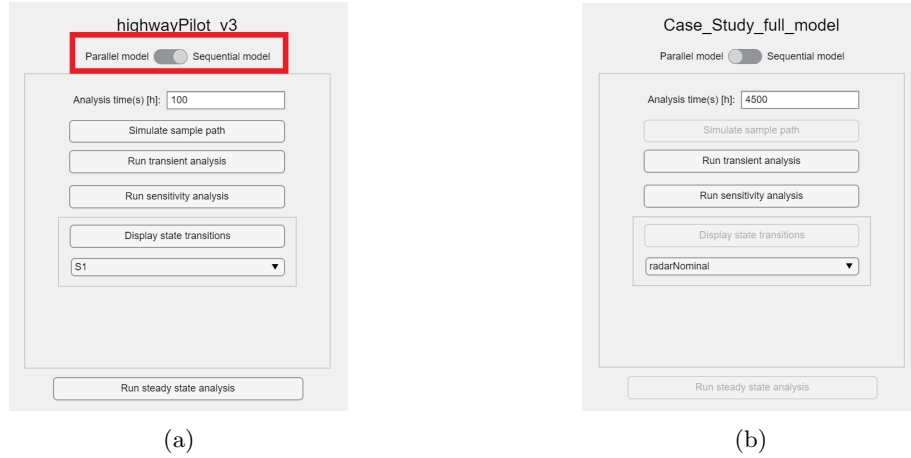


Figure 15: Switch that shows if the model is parallel or sequential. Figure 15a shows how the GUI looks like for a sequential model and figure 15b shows how the GUI looks like for a parallel model.

4.1.3 Plot Options

It is possible to change the plot interval and the number of points as well as the distribution of the points used to generate the plotted line. These settings only affect the result from a transient analysis. The settings are found under “Analysis configuration” in the “Plot settings” section, see figure 16. Note that if a logarithmic plot distribution is used, the lower plot interval must be strictly greater than zero.

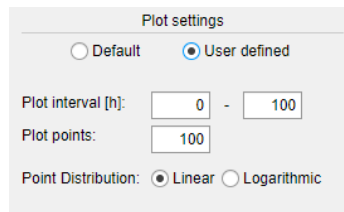


Figure 16: The different plot settings available to customize the resulting plots.

4.1.4 Relative Error

Under “Settings” and “General settings” it is possible to specify the relative error, see figure 17. This sets the magnitude for the relative error of the System failure probability, i.e. the relative error of $1 - R(t)$ where $R(t)$ denotes the reliability. The relative error applies to transient analysis with symbolic solvers, see section 4.2.2.

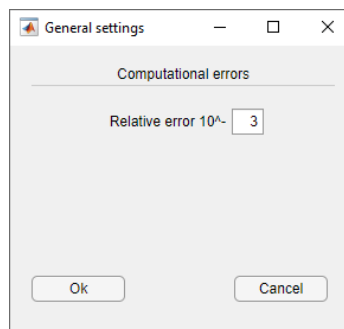


Figure 17: Window where a relative error can be specified.

4.1.5 Sensitivity Settings

The default range for parameters and transition parameters, described more thoroughly in section 4.3, can be changed from the sensitivity settings window. These settings can be found as shown in figure 18a.

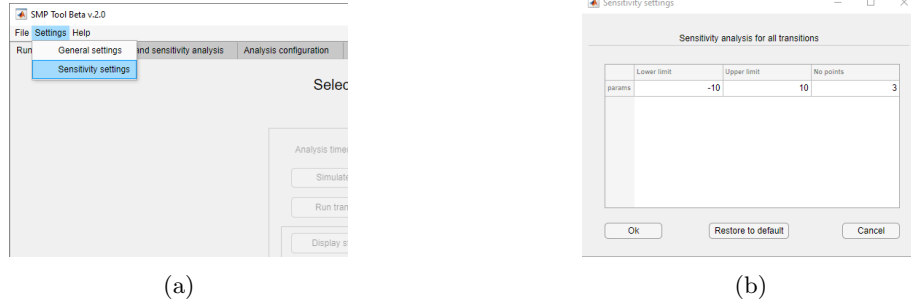


Figure 18: Figure 18a shows how to find the sensitivity settings and figure 18b shows the sensitivity settings window.

In figure 18b the sensitivity settings window is presented. The values in the three different columns can be changed by the user. The number of points is simply the number of points in the range, distributed linearly over the interval. The upper and lower limit set a percentage value for how much the parameter value should vary. The default values of "Lower limit = -10, Upper limit = 10, and No points = 3 gives, for a parameter p , the range $[0.9p \quad p \quad 1.1p]$.

4.2 Transient Analysis

Transient analysis calculates the reliability, availability and system failure probability. Definitions and examples of this can be found in [1]. Transient analysis is an appropriate method to analyze models with an absorbing state to determine the probability of termination within a certain time. For models with no absorbing state, the result will indicate the probability of first reaching a down state.

There are two different ways to perform a transient analysis with SMP-tool. The first option is to use Monte Carlo simulation and the second option is to use a numeric solver. The options can be found under "Analysis configuration" and then select the desired option under "Solver method", see figure 13. Note that both options can be selected. This will be treated as two separate analyses, the first with a numerical solver and the second with Monte Carlo simulation.

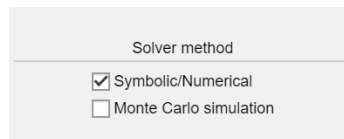


Figure 19: The different solvers available for transient analysis.

4.2.1 Transient Analysis with Monte Carlo Simulation

Transient analysis with Monte Carlo simulation simulates the model a repeated number of times. The specific number of times can be changed under "Analysis configuration" in the field next to "No. simulations". Simulation is done by sampling timers from the given transition distributions, and terminate either when a state marked as down state or absorbing state is reached, or when the accumulated time reaches the selected time limit. Reliability, probability distribution function (PDF) and system failure probability is plotted in one figure. Figure 20 shows the result for the highway pilot model.

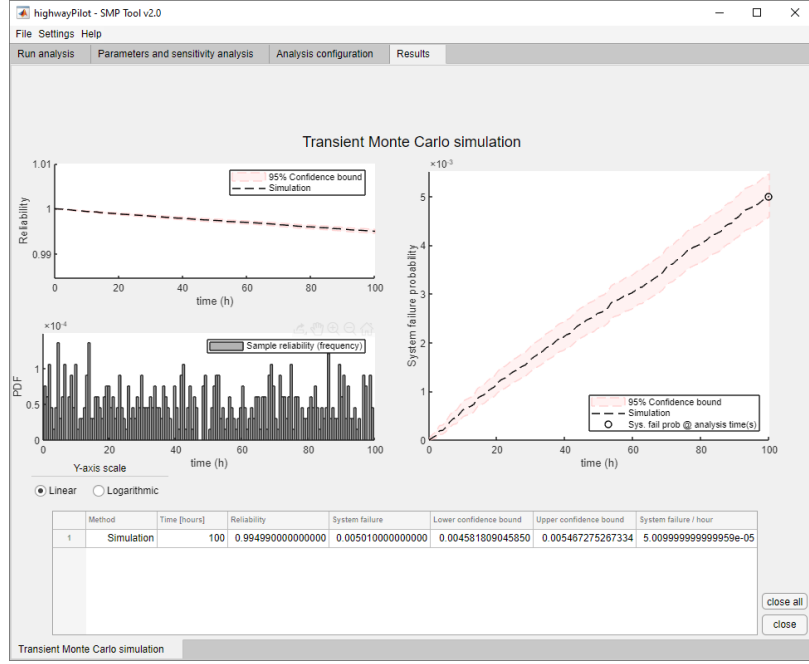


Figure 20: Result from a transient analysis with Monte Carlo simulation of the highway pilot model. The number of simulations used was 100000 and the analysis time was set to 100h.

4.2.2 Transient Analysis with Numerical Solver

SMP-tool uses a Laplace-inversion algorithm for the transient analysis with the numerical solver. It is possible to specify which algorithm to use under “Inversion algorithm” in the “Analysis configuration” tab. Multiple methods can be selected for the purpose of comparison.

Figure 21 shows a transient analysis result for the highway pilot model where the numerical solver was used.

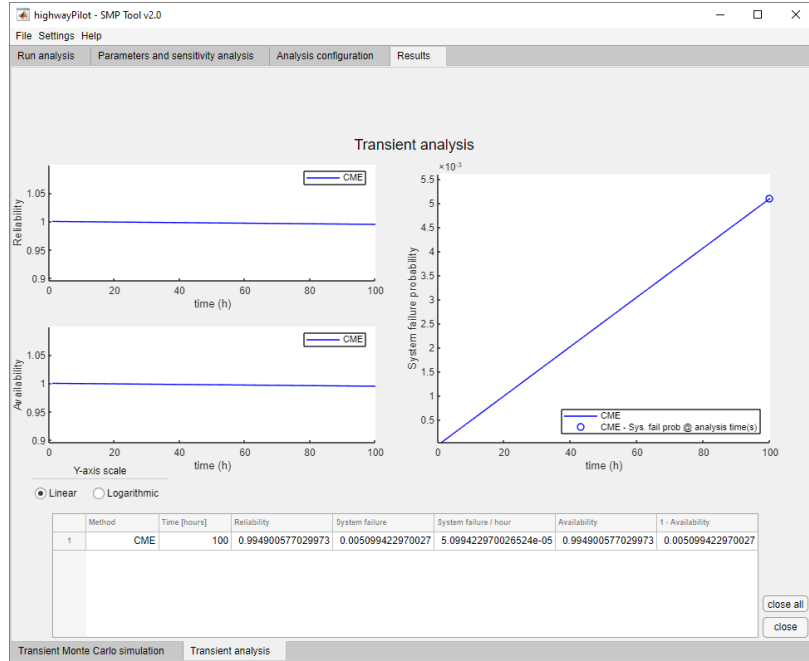


Figure 21: Result from a transient analysis with numerical solver for the highway pilot model. The CME inversion algorithm was used and the analysis time was set to 100h.

4.2.3 Transient Analysis of HSMP Models

Hierarchical models can be analyzed either with Monte Carlo simulation, or with numerical analysis. There are two different analysis methods for hierarchical models, *numerical integration* and *bounded regeneration*, either of which can be selected under “Hierarchy solver” in the “Analysis configuration” tab, see figure 13. Bounded regeneration is a more accurate method in case the hierarchy in the model has a finite possible number of transitions before the hierarchical state is exited. In case of unbounded regeneration, the approximate method of numerical integration is preferred.

4.3 Sensitivity Analysis

For a sensitivity analysis, each parameter of the model is varied one at a time, and for each value of each parameter the failure probability is calculated. This results in as many graphs as there are parameters, where each graph plots how the failure probability varies with the value of that parameter. A sensitivity analysis gives a clear view over which parameters affect the failure probability the most and the least. The specified analysis time decides after what time the failure probability is calculated. Similarly to a transient analysis it is possible to set the relative error as described in section 4.1.4.

There are two different sensitivity analyses available in SMP-tool. The first alternative, called “For selected parameters”, studies all the parameters of the model. The second alternative, called “For all transitions”, varies all defining values of the transitions that are not defined by parameters. In the case of the highway pilot model, which has zero parameters, the first alternative does not apply. The second alternative on the other hand suits this model. The choice for sensitivity analysis is available under the “Parameters and sensitivity analysis” tab. There, it is also possible to specify the range for each parameter, which will be used for the sensitivity analysis. The default range varies the parameter value $\pm 10\%$.

To be able to demonstrate both alternatives for sensitivity analysis a modified version of the highway pilot is analyzed, see figure 22. The only difference compared to the original version in figure 7 is that some values have been swapped for parameters defined with the same value.

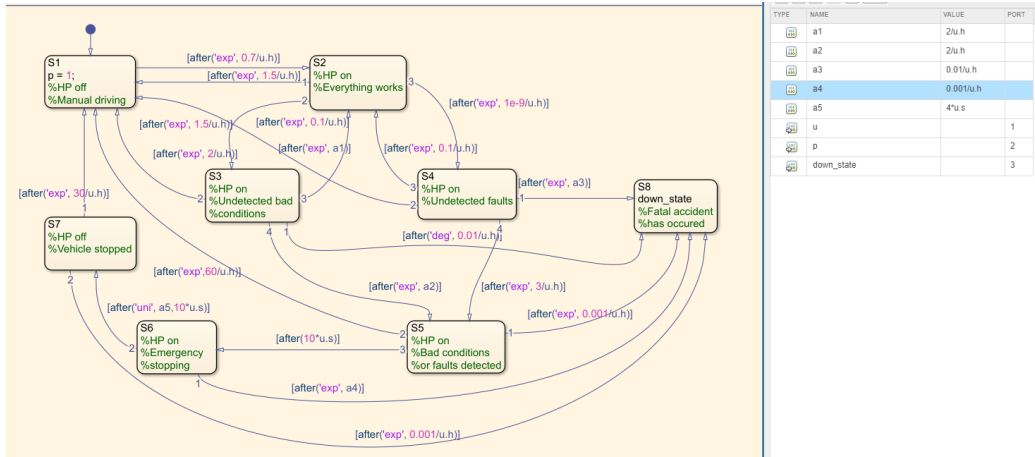


Figure 22: A slightly modified version of the highway pilot model. Compared to the original version, presented in figure 7, some values have been swapped for parameters defined with the same value.

With the model from figure 22 loaded into SMP-tool we can take a look at the “Parameters and sensitivity analysis” tab, see figure 23. First, notice that all parameters show up in the table together with a default range of $\pm 10\%$. Secondly, highlighted in red is the selector for if the parameters or the transition values should be varied.

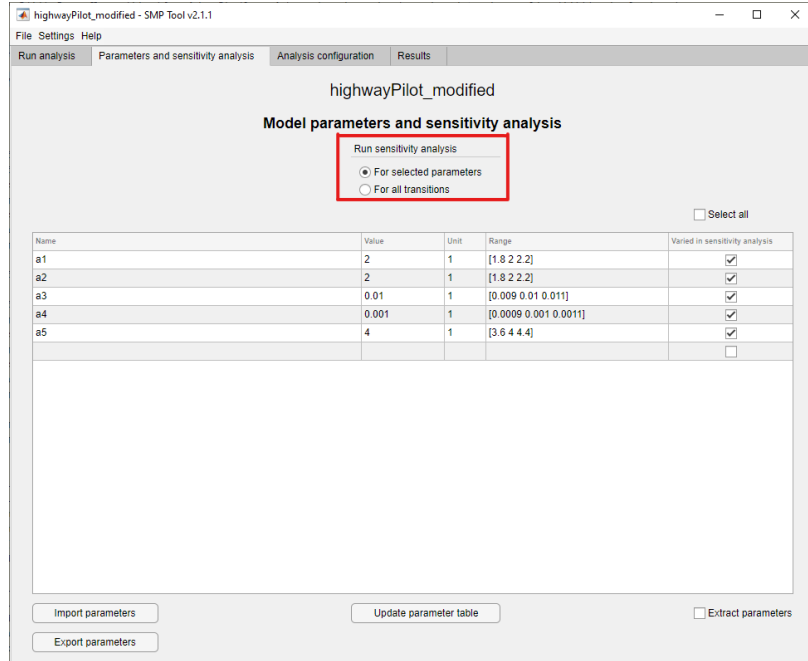


Figure 23: An overview of the settings for sensitivity analysis with a modified version of the highway pilot model.

A result from a sensitivity analysis of all parameters of the modified version of the highway pilot model from figure 22 is presented in figure 24. All parameters have been varied with $\pm 10\%$. Note that the result is split into sub-tabs, one for each parameter.

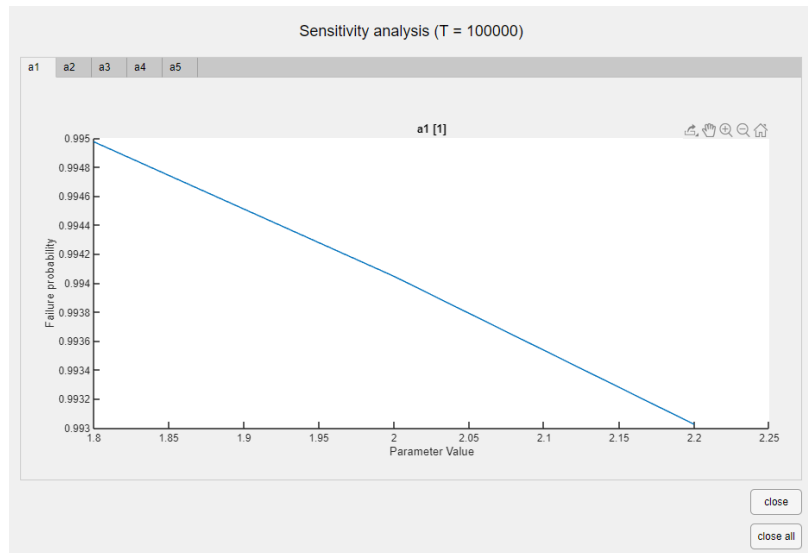


Figure 24: Sensitivity analysis of the modified highway pilot model from figure 22. All parameters have been varied with $\pm 10\%$.

From version 3.0 and onward it is also possible to perform a sensitivity analysis using Monte carlo simulations. Using this method, SMP-tool performs a transient analysis with the Monte Carlo solver for each of the specified parameters with the specific range values. The result from this analysis will show a plot with the failure probability and confidence interval for the simulation at the given analysis time for all of the different parameter values. An example can be seen in figure 25.

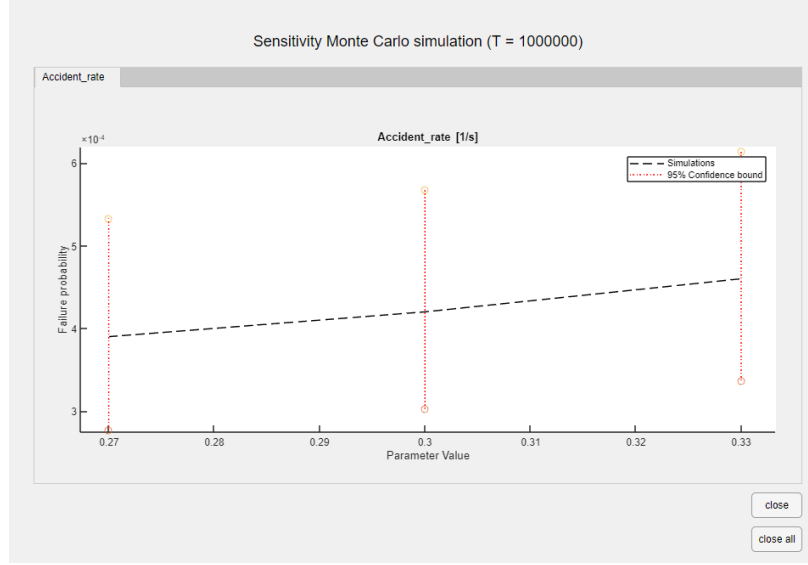


Figure 25: Sensitivity analysis of the modified redundant steering model given as a test model in the tool, in which the Monte Carlo solver has been used. The red lines shows the limits of the 95% confidence intervals.

4.4 Steady State Analysis

Steady state analysis calculates the probability of being in each state as time goes to infinity. Due to the nature of a steady state analysis, this is a suitable analysis for models without absorbing states, that is states with no outgoing transitions. If a steady state analysis is performed on a model with one or several absorbing states the resulting probability will simply be 1 distributed among all the absorbing states. SMP-tool's steady state analysis also calculates the "Total probability", which is the probability of being in a down state.

To be able to present a realistic example of a steady state analysis we present a modified version of the highway pilot model, see figure 26. Compared to the original version presented in figure 7, an exponentially distributed transition from S8 to S1 with rate 0.01h has been added. This small adjustment makes S8 a non-absorbing state.

4.5 Simulate Sample Path

The option “Simulate sample path” simulates the model only once, but for a long period of time. This is used to determine the steady state distribution of models with no absorbing state. This method also displays the sample path of the simulation, which can be used on models with an absorbing state to generate an example of how the system behaves before it terminates. The analysis time determines the total simulation time. Figure 28 shows a result of the highway pilot model with an analysis time of 100h.

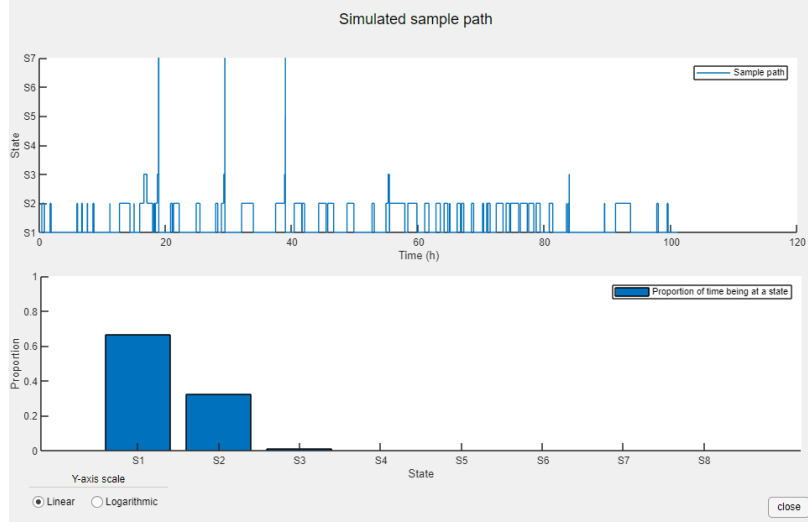


Figure 28: Example simulation of sample path done with the highway pilot model. Analysis time was set to 100h.

4.6 Display State Transitions

SMP-tool can provide information about each state in a visual form. This is done by selecting a state and pressing “Display state transitions”, see figure 29.

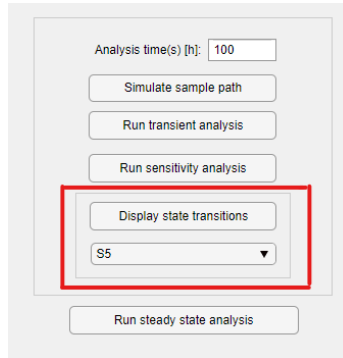


Figure 29: The figure shows where in the GUI the “Display state transitions” can be run and where the state for that is chosen.

The chart from “Display state transitions” can be used to visualize how the sojourn time $U_n(t)$, the semi-Markov kernel corresponding to some transition $Q_{ij}(t)$, the holding time given some transition $H_{ij}(t) = P(U_n \leq t \mid X_{n+1} = j, X_n = i)$, and the cumulative distribution function of the timer for some transition $F_{ij}(t)$, differ and relate to each other. For more information about these quantities, the reader is recommended to read [1]. The computed quantities can then be used to determine whether the semi-Markov process behaves according to the expertise of the user as well as

according to any possible given facts about the system of interest. As an example, figure 30 shows the result chart of the state transitions for S5 of the highway pilot model.

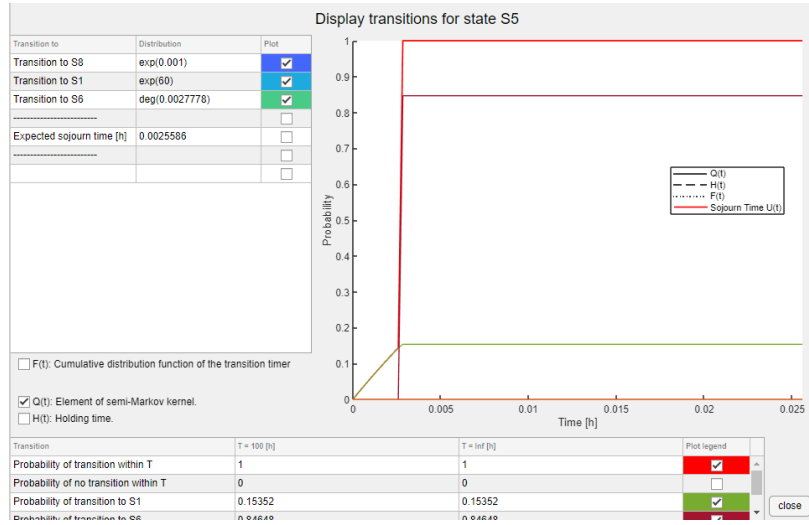


Figure 30: Information window for state S5 of the highway pilot model.

5 SMP-tool 3: Parallel process model simulation

SMP-tool 3.0 introduced a new way to model, using parallel states, which can be used to model processes that contain simultaneous sub-processes. Analysis of these models can only be done by Monte Carlo simulation.

5.1 Parallel models

A parallel model can be created by right clicking in any Stateflow-chart and changing the decomposition to "Parallel (AND)". States placed in the chart represent parallel processes. Each parallel state in the chart behaves independently much like a HSMP. Local times and transition occur in the same way as SMP, however, parallel states can communicate with both *guards* and *events*. Much like SMP:s, time is advanced by state jumps. With parallel states, however, it is possible for different parallel states to have their transition fire simultaneously. In this case, the global *parallel state execution order* will determine in which parallel state transition is to be taken first. Locally, inside each parallel state, if two or more transitions from the same state have the same time to fire, the *transition execution order* determines which transition to take. The global execution order is set by right clicking on a parallel state and choosing its execution order. Similarly, local execution order between transitions is set by right clicking on a transition.

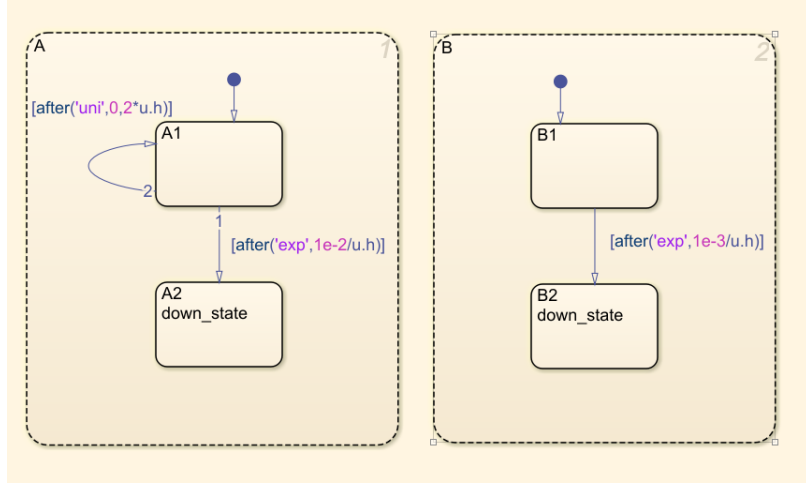


Figure 31: A simple parallel model. The number in the upper right corner of each parallel state represents global execution order. The number next to the transitions in state A represents the local transition execution order

5.2 Guards

Guards, using stateflow syntax `in(state_name)`, allows a condition to be set for a transition to be able to execute. The condition `in(state_name)` is true if the state `state_name` is currently active e.g. in another parallel state. The `after(distribution)` operator is true any time after the sampled time in `distribution`, which is sampled once the source of the transition is entered. If no `after()` condition is set, the transition is set to be instantaneous when the guard conditions are met. Conditions on several guards and distributions can be made by writing logic formulas with `~` (negation), `&&` (and) and `||` (or).

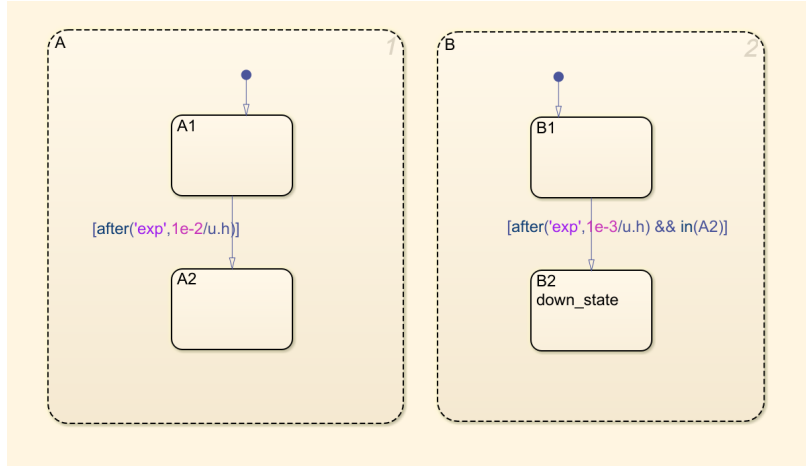


Figure 32: A guarded parallel model. The transition from B1 to B2 can only be taken if its `after()` timer has reached zero, and the transition from A1 to A2 has occurred.

5.3 Events

Another feature of parallel models is communication with events. A transition can optionally broadcast an event or to be conditioned by them. An event broadcast is defined on a transition string by placing it in the transition action braces `/{event}`, after the condition brackets. Event conditions are defined by placing an event in front of the condition brackets or several separated with `||`. For sake of clarity, it is recommended to define the events as an "local event" in the symbols panel.

When a transition with a broadcast event event is taken, the event is immediately executed in accordance with transition action semantics in Stateflow. Event-transitions can themselves trigger new events. Event transitions are always instantaneous, and will interrupt the current parallel state order to execute any possible condition event in parallel state order. When all transitions have been executed (including subsequent events), the parallel state order will continue from the next component. Event transitions can also be guarded by state-conditions or by an `after()` condition, although if a transition triggers an event, neither its source state or destination state will be active for guard conditions. Furthermore, transitions conditioned by an event can only occur when they are broadcast, such that any guard condition must also be true at that time. For `after()` conditions, this means that the time remaining must be 0 at the time of the event.

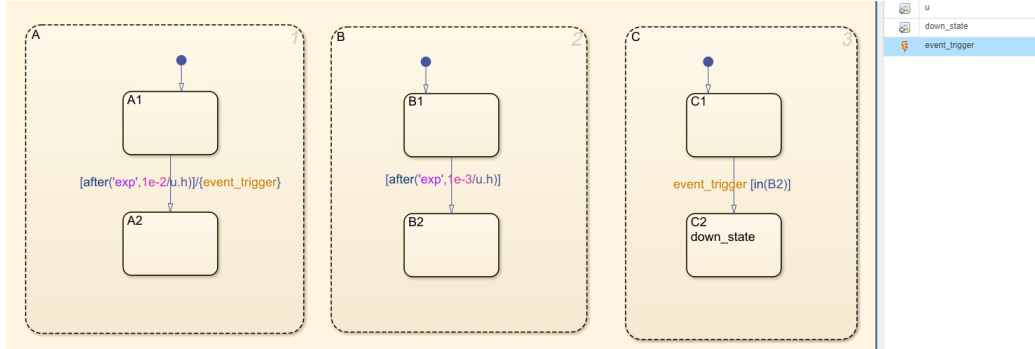


Figure 33: A parallel model with events. The transition from A1 to A2 will broadcast the event, after which the condition event transition in C will be immediately fired, but only if its guard condition is true at the time of the event.

5.4 Rules and limitations of parallel models, currently



Figure 34: Complete transition input string

- The distribution specified in the `after()` condition must be either the degenerate distribution `deg`, uniform distribution `uni`, exponential distribution `exp` or lognormal distribution `Lognormal`. The degenerate distribution represents a deterministic, constant time t . It can be specified either by `after('deg', t)`, or simply `after(t)`.
- Unlike `makedist`, the exponential distribution is defined by its *rate parameter* λ , rather than the mean value $\frac{1}{\lambda}$.
- Only one event can be broadcast per transition.
- Transitions between parallel states are not allowed.
- A transition can contain several conditions events, separated by *or* (`||`). *and* (`&&`) is not allowed since events are executed one by one.
- Analysis of parallel models can only be done via Monte Carlo simulation.
- Sample path simulation, steady state analysis and state transitions plots are not available for parallel models.
- Sensitivity analysis using the alternative “For all transitions” is not available using the new parallel model simulation backend introduced in v3.2.

- Nested junctions, junctions with no outgoing transitions and junctions with multiple ingoing transitions are not allowed using the new parallel model simulation backend introduced in v3.2.
- Transitions out of down states, their children or their ancestors are currently not allowed using the new parallel model simulation backend.
- Simulation using the new parallel model simulation backend is significantly slower on other platforms than 64-bit Windows.

References

- [1] S. Kaalen, M. Nyberg, and C. Bondesson. “Tool-Supported Dependability Analysis of Semi-Markov Processes with Application to Autonomous Driving”. In: *2019 4th International Conference on System Reliability and Safety (ICSRS)*. 2019, pp. 126–135. DOI: 10.1109/ICSRS48664.2019.8987701.
- [2] Stefan Kaalen and Mattias Nyberg. “Branching Transitions for Semi-Markov Processes with Application to Safety-Critical Systems”. In: *Model-Based Safety and Assessment*. Ed. by Marc Zeller and Kai Höfig. Cham: Springer International Publishing, 2020, pp. 68–82. ISBN: 978-3-030-58920-2.
- [3] Stefan Kaalen, Mattias Nyberg, and Olle Mattsson. “Transient Analysis of Hierarchical Semi-Markov Process Models with Tool Support in Stateflow”. In: *Quantitative Evaluation of Systems*. Ed. by Alessandro Abate and Andrea Marin. Cham: Springer International Publishing, 2021, pp. 105–126. ISBN: 978-3-030-85172-9.