

SMP Tool Beta 3.9.2 Quickstart Guide

May 24, 2024

SMP Tool is a MATLAB application, currently in a beta version, for analysis of stochastic and time-dependent models. The supported modelling language, called Stochastic Stateflow (SSF), is an extension of a safe subset of Stateflow, with formally defined syntax and semantics [1].

SMP tool supports transient and sensitivity analysis of SSF models. Transient analysis of an SSF model finds the failure probability while sensitivity analysis indicates to which parameters in the model the failure probability is most sensitive. SMP Tool implements two solver methods. Firstly, a Monte Carlo simulation solver that gives statistical bounds on the failure probability. Secondly, an analytical solver that gives an upper bound on the failure probability.

This quickstart guide covers installation and analysis of a SSF model using SMP Tool. For questions and clarification, the reader is encouraged to send an email to kaalen@kth.se.

1 Installation

A prerequisite for installing SMP Tool is an installation of MATLAB 2023b with Simulink, Stateflow, Symbolic Math Toolbox, and Statistics and Machine Learning Toolbox. Note that SMP Tool has only been tested on Windows 10 64-Bit and MATLAB 2023b but may work on other operating systems and other MATLAB versions. To install SMP Tool extract `SMP_Tool_Beta_3.9.2.zip` and run the MATLAB application installation file `SMP Tool Beta 3.9.2.mlappinstall` and click *Install* in the dialog that appears. After installation SMP Tool can be launched from the *APPS* tab in MATLAB by finding it in the drop-down menu and clicking on it. When launched, the graphical user interface shown in figure 1 appears.

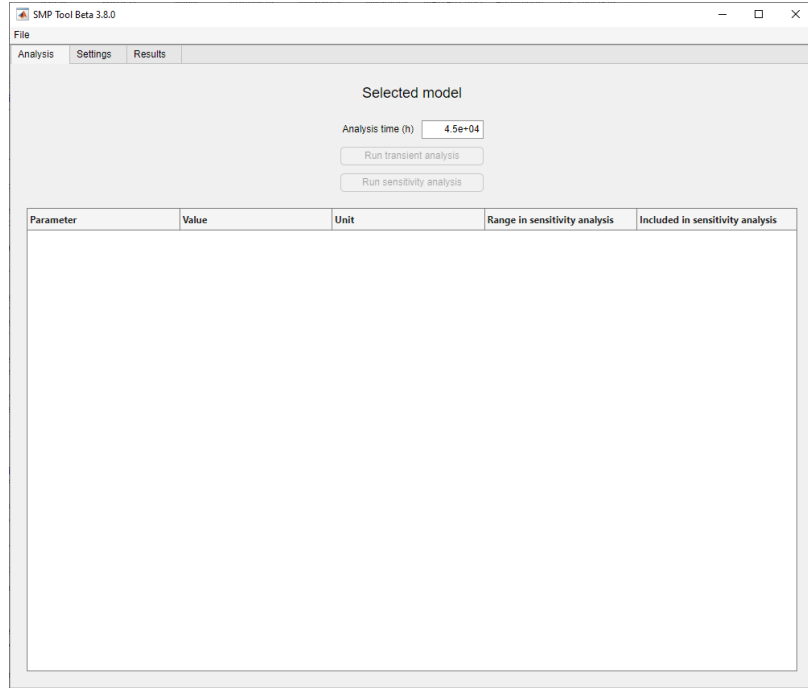


Figure 1: Graphical user interface for SMP Tool.

1.1 Open a Model

To load a model into SMP Tool, click the *File* button in the top left corner, see figure 2. To open the existing model *Gearbox_wheel_lock* found in *example.models* click *Open model*, and select the model in the dialog that appears. The model is an SSF model, which is described in [1], and exemplifies most of the SSF model syntax. The graphical user interface is updated as seen in figure 3.

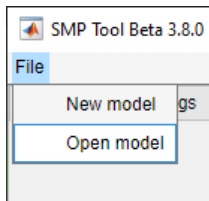


Figure 2: *File* menu from SMP Tool.

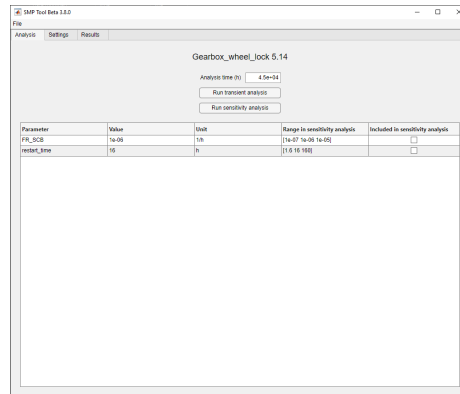


Figure 3: *Analysis* tab from SMP Tool.

1.2 Analyze a Model

Analyses are started under the *Analysis* tab seen in figure 3. Two analyses, transient analysis and sensitivity analysis, are available. The analysis time is also specified here, which is the time of interest in hours for the analyses. The default analysis time *45 000* hours is reasonable for this example.

Currently the analyses are performed using Monte Carlo simulations or analytically. The solvers can be selected under the *Settings* tab and *Solver* panel, see figure 4. Select both solvers.

The Monte Carlo simulation solver has one parameter that affects accuracy, the number of failure samples, which can be changed under the *Settings* tab in the field titled *Failure samples*, see figure 4. The default is sufficient for this example.

The analytical solver has two parameters that affects accuracy, which can be changed under the *Settings* tab in the fields titled *Regenerative path absolute tolerance* and *Laplace inverse relative tolerance*, see figure 4. The default is sufficient for this example.

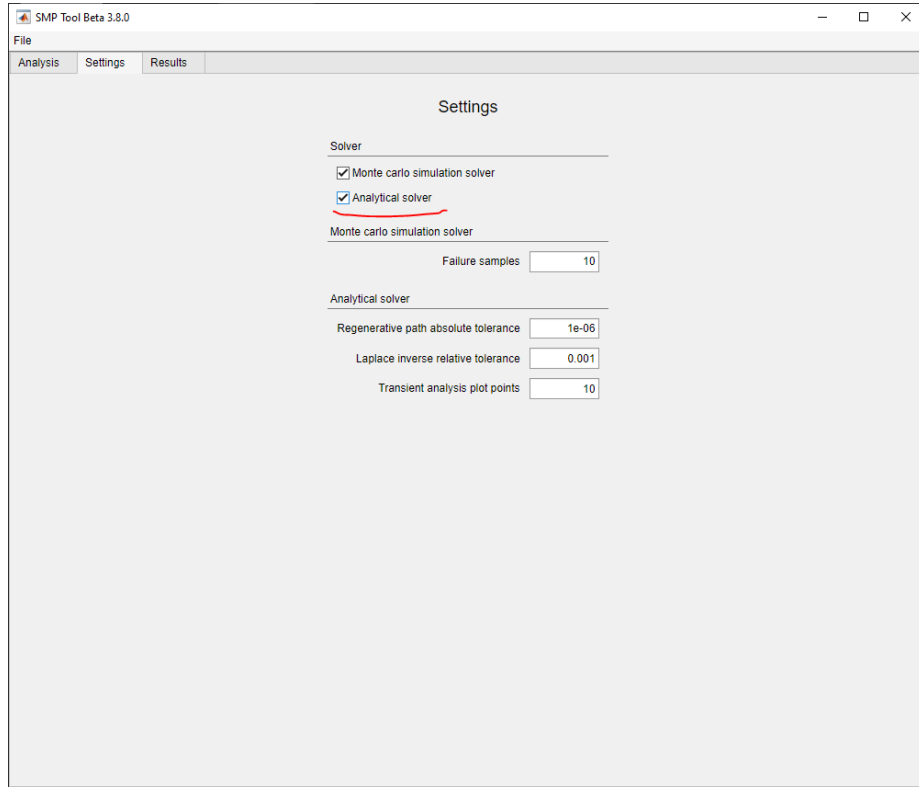


Figure 4: *Settings* tab from SMP Tool.

1.2.1 Transient analysis

Transient analysis calculates the failure probability for the model. To find this for the opened model *Gearbox_wheel_lock*, click *Run transient analysis* and wait for the analysis to complete. When completed, tabs showing the analysis results using both solver methods are opened under the *Results* tab, see figures 5 and 6. As seen in the figures the probability of failure during the analysis time is found to be around $1.2e-5$ with a certain confidence interval using the Monte Carlo simulation solver and at most $2.0e-5$ using the analytical solver.

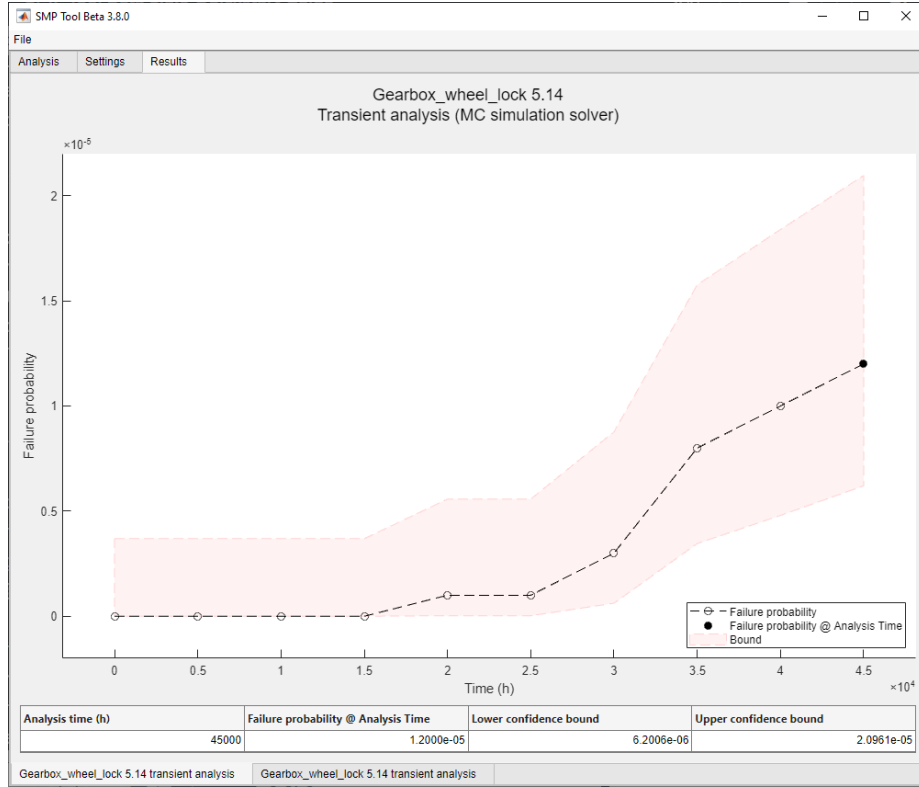


Figure 5: Result from a transient analysis of the *Gearbox_wheel_lock* model using Monte Carlo simulation solver. The *Failure samples* was 10 and the analysis time was 45000 hours.

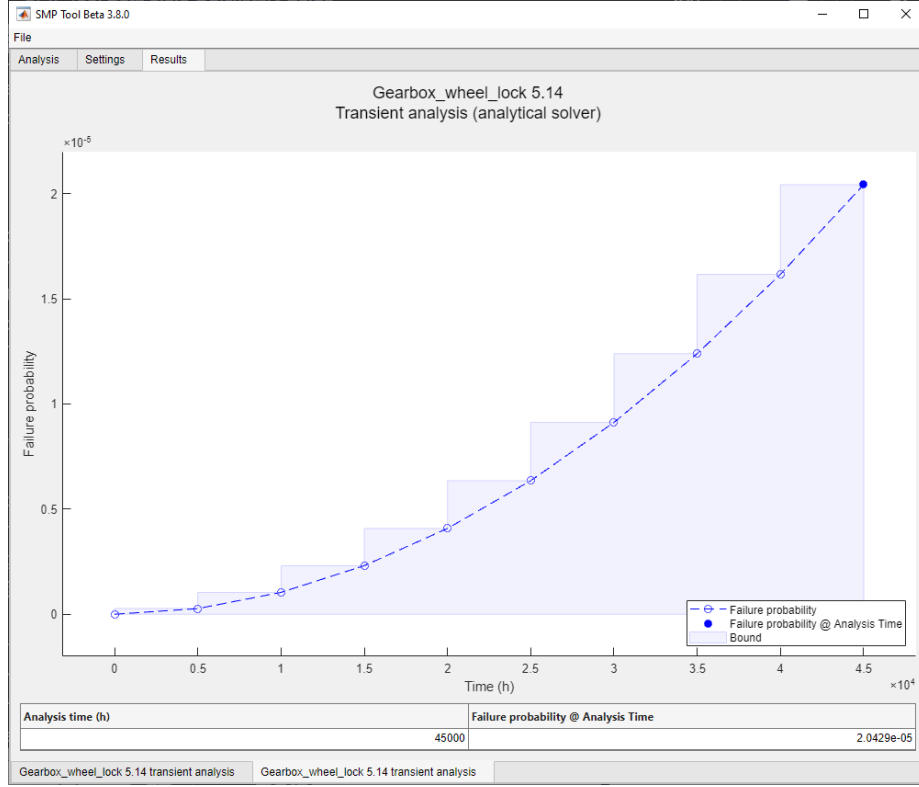


Figure 6: Result from a transient analysis of the *Gearbox_wheel_lock* model using analytical solver. The *Regenerative path absolute tolerance* was $1e-6$, the *Laplace inverse relative tolerance* was $1e-3$ and the analysis time was 45000 hours.

1.2.2 Sensitivity Analysis

Sensitivity analysis calculates failure probability for the model where each specified parameter is varied one at a time. A sensitivity analysis gives a clear view over which parameters affect the failure probability the most and the least.

The specific parameters and their ranges can be changed under the *Analysis* tab, see figure 7. Select the parameters *FR_SCG* and *restart_time*, their default ranges are sufficient.

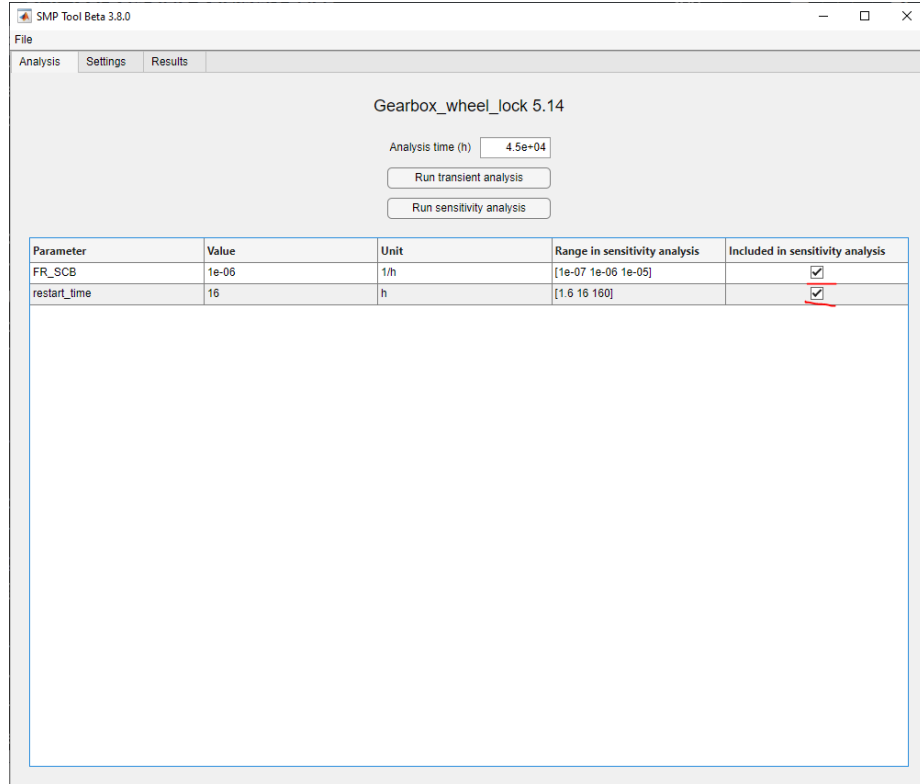


Figure 7: *Parameters and sensitivity analysis* tab from SMP Tool.

To perform a sensitivity analysis for the parameters FR_{SCG} and $restart_time$ for the opened model *Gearbox_wheel_lock*, click *Run sensitivity analysis* and wait for the analysis to complete. When completed, tabs showing the analysis results using both solver methods are opened under the *Results* tab, see figures 8 and 9. Note that the results are split into sub-tabs, one for each parameter. As seen in the figures the parameter $restart_time$ doesn't affect the overall failure probability much when it is varied between the specified values while FR_{SCG} does.

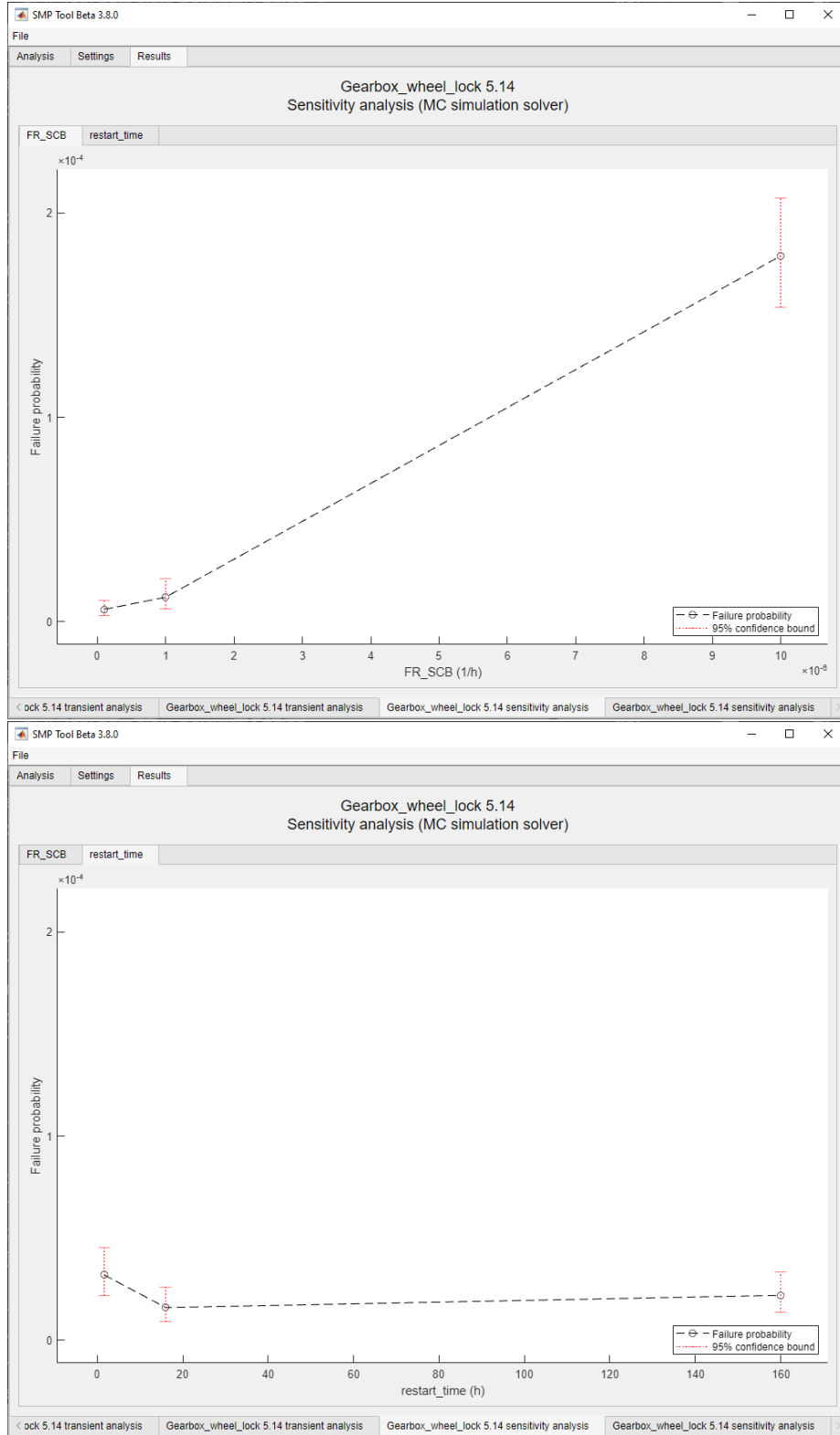


Figure 8: Result from a sensitivity analysis of the *Gearbox_wheel_lock* model using Monte Carlo simulation solver. The *Failure samples* was 10, the analysis time was 45000 hours and the parameters *FR_SCG* and *restart_time* were varied $\pm 900\%$.

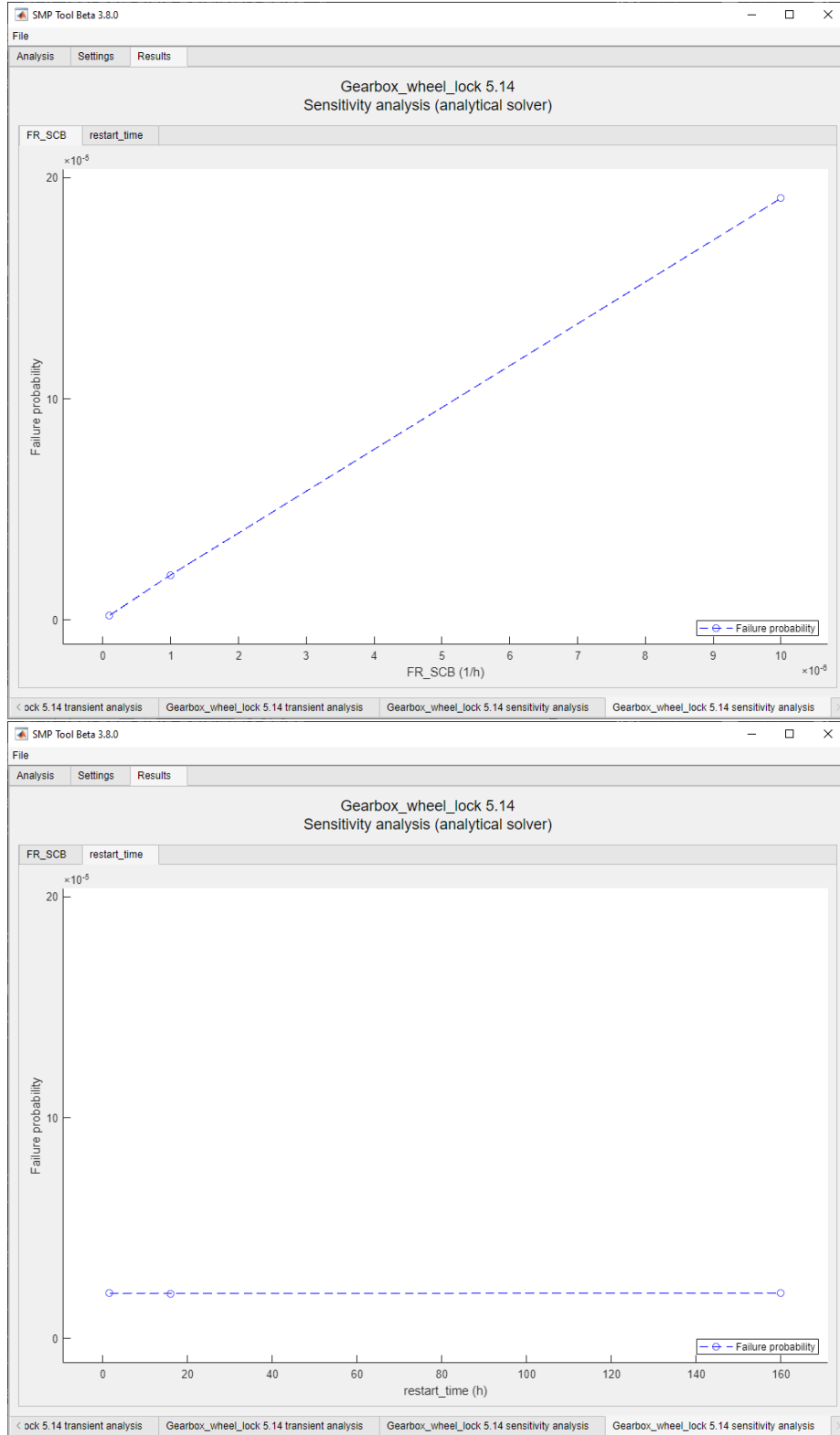


Figure 9: Result from a sensitivity analysis of the *Gearbox_wheel_lock* model using analytical solver. The *Regenerative path absolute tolerance* was $1e-6$, the *Laplace inverse relative tolerance* was $1e-3$, the analysis time was 45000 hours and the parameters FR_SCG and $restart_time$ were varied $\pm 900\%$.

References

- [1] Stefan Kaalen et al. “A Stochastic Extension of Stateflow”. In: *Proceedings of the 2022 ACM/SPEC on International Conference on Performance Engineering*. 2022, pp. 211–222.