Test Generation using Simulink Design Verifier (SLDV) and Reactis

This report consists of two parts:

(1) The first part reports our experience on formalizing the output instability behavior, and running SLDV and Reactis to violate it.

(2) The second part reports the incompatibly issues and shortcomings of SLDV and Reactis tools.

Part 1:
We first implemented the insatiability behavior using Simulink model in Figure 1. It computes the absolute differences of consecutive values of the input signal. That is, higher values at the output port indicate higher instability in the input signal.

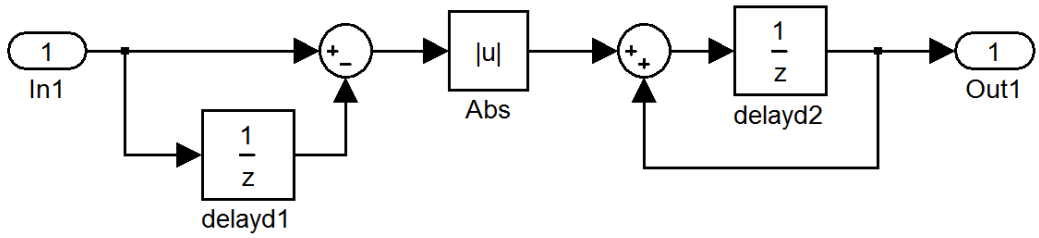

Figure 1. Formalization of the instability assertion

We then applied SLDV and Reactis to the model in Figure 2, where an input signal is coming to the asserted block capturing the instability property.

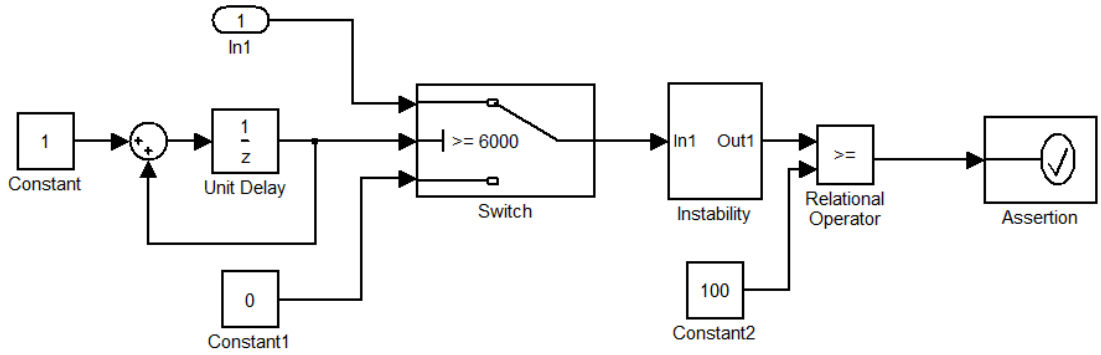

Figure 2. Simple model we used to check if SLDV and Reactis can violate the instability assertion.

The simulation time of the model in Figure 2 is 10 seconds and the simulation time step is 0.001 seconds. To replicate the same fault observed in practice, we tried to identify an occurrence of stability violation after 6000 time steps. That is why the threshold of Switch operation is set to 6000. Further the threshold of the instability assertion is set to 100, i.e., the threshold violated by the instability failure observed in practice.

We ran both SDLV and Reactis to check if they are able to find a test input that can violate the instability assertion. As shown in Figure 3, SLDV was not able to detect a violation of the assertion after almost 24 hours, even for this very small model.

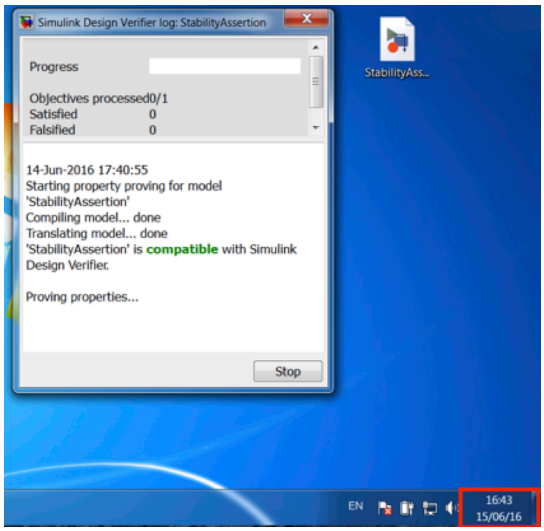

Figure 3. SLDV test generation report after 24 hours.

Reactis was also not able to detect the fault after 24 hours. Here is the Reactis test generation report after 24 hours.

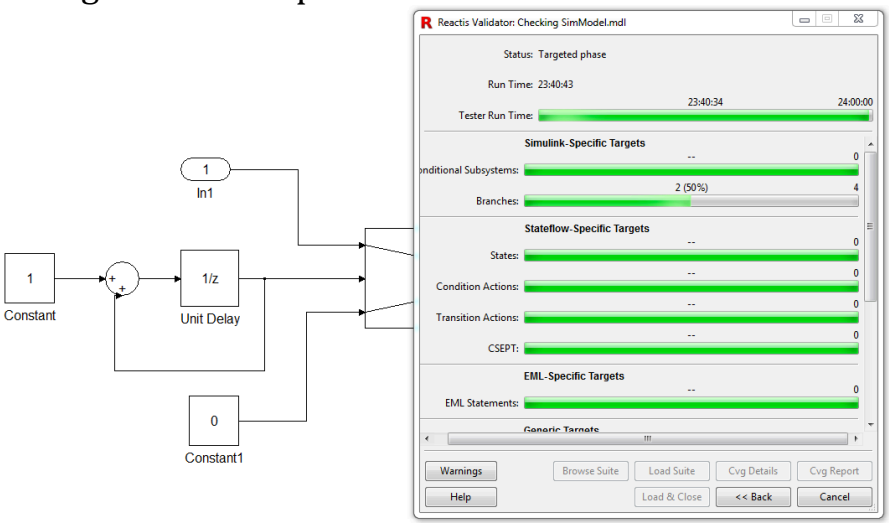

Figure 4. A snapshot of the Reactis test generation report after 24 hours

Part 2: Incompatibility Issues and Shortcomings

1) SLDV and Reactis are both incompatible with time-continuous blocks. Both SLDV and Reactis show an incompatibility error message when asked to generate test cases for this model as it includes a time-continuous integrator:

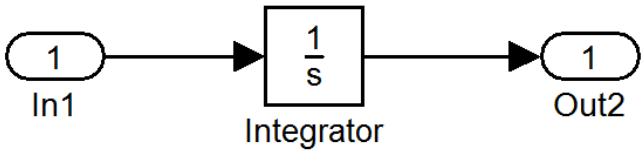

Figure 5. A simple Simulink model containing a time-continuous integrator

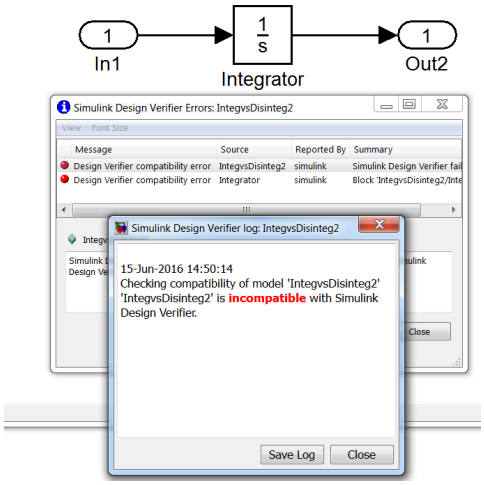

Figure 6. SLDV shows an error message when asked to generate tests for the Simulink model in Figure 5

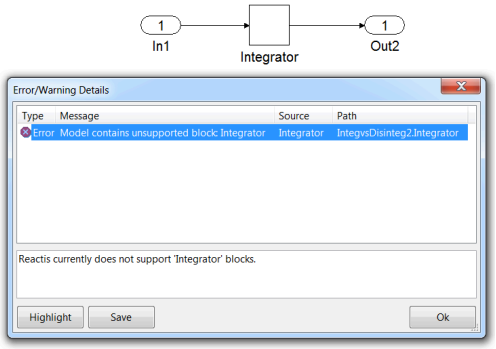

Figure 7. Reactis shows an error message when asked to generate tests for the Simulink model in Figure 5

2) SLDV is not applicable to Simulink models containing non-linear floating-point operations such as trig and square root functions, but

Reactis seems to be applicable to them. We used the model in Figure 8, which contains a sine block, to see if these tools can generate coverage-adequate tests that cover both branches of the switch block in the model.

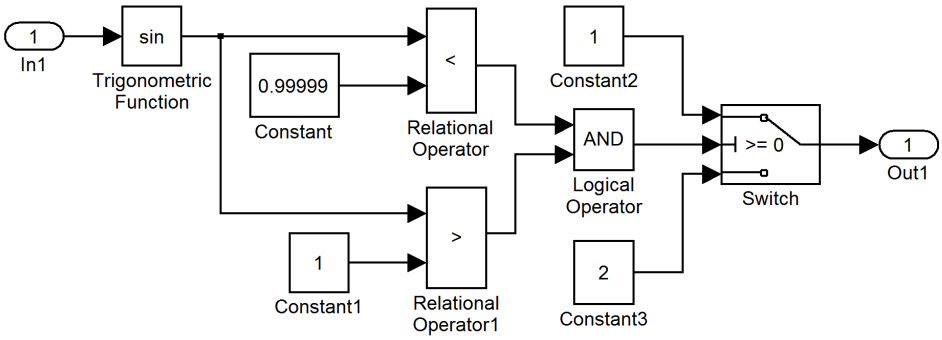

Figure 8. A Simulink model containing a sine block

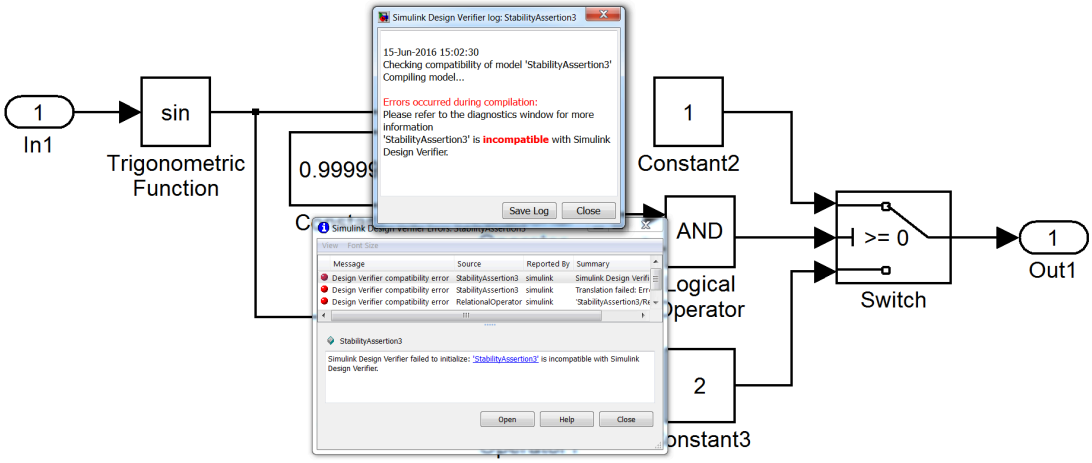

Figure 9. SLDV shows an error message when asked to generate tests for the Simulink model in Figure 8

SLDV shows an incompatibility error message when asked to generate tests for the model in Figure 8. Unlike SLDV, Reactis does not show an error message. However, it cannot generate test cases that cover all branches in the models, i.e., both outputs of the switch block, even after 12 hours. Figure 10 shows the test generation report of Reactis after 12 hours.

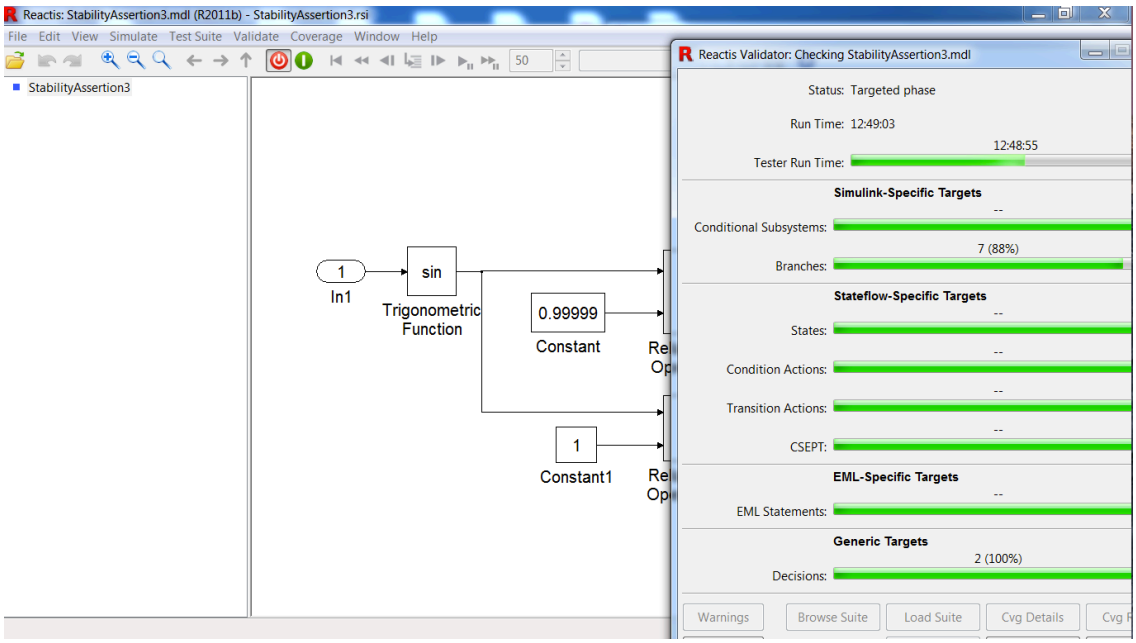

Figure 10. SLDV test generation report for the model in Figure 8 after 12 hours

3) As explained in the help documents of SLDV and Reactis, they both have limitations for testing S-Functions (library code and system functions) used in Simulink models. We tried to test the code generation models from our industry partner that include many S-Functions, with SLDV and Reactis. SLDV was not applicable to the S-Functions used in the models from our industry partner, and shows an incompatibility error message, as follows:

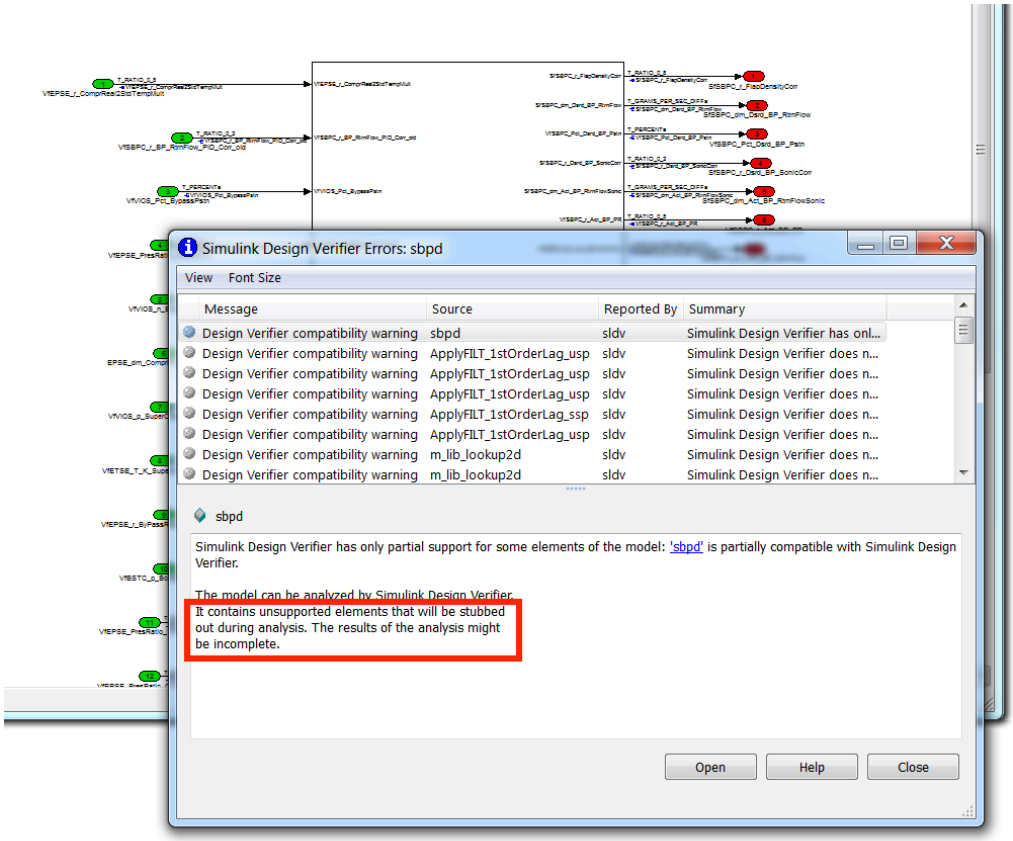

Figure 11. SLDV test generation error

Reactis did not show an incompatibly error message, however, was not able to generate test cases for our models and shows a run-time error related to the S-functions used in the model during test generation. Note that the same model with the same S-Function executed perfectly fine in Matlab.

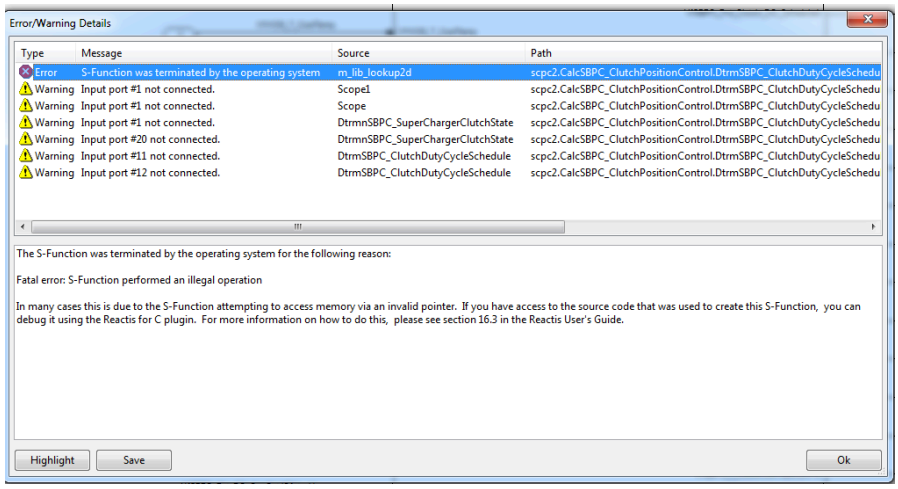

Figure 12. Reactis test generation error