Breach Issues

# Wrong dimensions

At lines 786 and 790 of BreachRequirement.m we had to replace

B.SetParam(params\_sys,values(i\_sys,:));

by

B.SetParam(params\_sys,values(i\_sys));

and similarly at line 790.

# Hardcoded dimension for fmincon

At line 516 of BreachProblem.m we had to replace

(this.ub-this.lb).\*rand(3,1) + this.lb

by

(this.ub-this.lb).\*rand(length(this.ub),1) + this.lb.

It seems very strange to use rand(3,1) here. Does this mean fmincon is not fully supported?

# Missing arguments in STL\_Eval

In STL\_Eval, we call STL\_EvalThom in the case where ischar(phi). Should we not call STL\_Eval(...,args) instead with whatever args are passed?

# Considering the robustness of preconditions

In Eval (and Eval\_IO) in BreachRequirement and also in objective\_fn in the context of a falsification problem, it may or may not be necessary to take into account the precondition value in the global robustness value as in the following lines

% A BreachRequirement must return a single value

global\_val = min(min(traces\_vals));

global\_precond\_val = min(min(trace\_vals\_precond))

this.val = min([global\_val,global\_precond\_val]);

# Display-related error in Falsification

In BreachProblem class, the variable x\_best is not initialized and in the case where the formula robustness is always Inf then x\_best is undefined which will produce an error in the method DisplayBestResults. To fix this issue we had to initatialize it explicitely in the class declaration.

A similar error then happened when reading the attribute BrSetBest of the BreachProblem instance, which was empty. Luckily calling the method GetBrSetBest() instead fixes the issue. Perhaps the attribute BrSetBest needs to be made private.

# Broken example script

In test\_PlotRobustMap, should not simple\_spec.stl be ultimately replaced by AFC\_simple\_spec.stl? Note that this test script still goes into error even after this change.

# Definition domain of satisfaction signals

In STL\_EvalThom, I think the following is not correct:

time\_values1 = [time\_values1 time\_values1(end)+I\_\_\_(end)];

This seems to increase the size of the definition domain instead of shrinking it.

Should it not use -I\_\_\_(end) instead?

# Naming of Plotting functions

It is not immediately clear that PlotRobustSat, in addition to plotting, also takes care of computing the robust satisfaction value. Same remarks apply to other plotting functions. Perhaps different naming convention could indicate that some robustness computation is also involved?