

Prototype P3.14

Compiler support for uncertainty computations

Adrian Pop (LiU), Peter Aronsson (MaCo) and Peter Fritzson (LiU)

December, 2012

.....

Summary

The objectives and results of this task and deliverable are:

- Add support for uncertain variables and distributions to the OpenModelica compiler
- Add connection from OpenModelica compiler to OpenTURNS for computation with uncertainties

The OpenModelica compiler now supports declaration of uncertain variables that can have different uncertainty distributions.

New functionality has been implemented to automatically generate from the Modelica models:

- A wrapper for connection to OpenTURNS
- Simulation models tailored for the OpenTURNS connection
- A Python script that can be ran in OpenTURNS for performing uncertainties analyses

Additional functionality has been implemented to run automatically from OpenModelica the generated Python script in the OpenTURNS framework and return the results.

Furthermore, the OpenModelica generated simulation code has been tailored so it can be efficiently run in parallel on several processors by the OpenTURNS framework. This benefits both on the speed of the uncertainties analyses performed on the OpenTURNS framework but also on the speed of any other type of Monte Carlo simulation or parameter sweeps.

Uncertainties in Modelica

OpenModelica now has support for declaration of uncertain variables and their distributions. The Distribution is a built-in type in the compiler. Variables have additional attributes that are tailored for uncertainties: distribution and uncertain (enumeration type Uncertain.given, Uncertain.refine and Uncertain.sought). An example is given below.

```

model CantileverBeam
  "Model from: http://doc.openturns.org/openturns-latest/html/ExamplesGuide/cid1.xhtml#cid1"
  parameter Distribution distribution =
    Distribution("Beta", {0.93,3.2,28000000.0,48000000.0}, {"r","t","a","b"});
  parameter Distribution distribution =
    Distribution("LogNormal", {30000,9000,15000}, {"mu","sigma","gamma"});
  parameter Distribution distributionL =
    Distribution("Uniform", {250,260}, {"a","b"});
  parameter Distribution distributionI =
    Distribution("Beta", {2.5,4.0,310.0,450.0}, {"r","t","a","b"});
  Real y(uncertain=Uncertainty.sought);
  parameter Real F(distribution=distributionF,
    uncertain=Uncertainty.given)=300;
  parameter Real E(distribution=distributionE,
    uncertain=Uncertainty.given)=3000000000.0;
  parameter Real L(distribution=distributionL,
    uncertain=Uncertainty.given)=250;
  parameter Real I(distribution=distributionI,
    uncertain=Uncertainty.given)=4e-06
  Correlation correlation[1];
algorithm
  correlation:={Correlation(L, I, -0.2)};
equation
  y=(F*L*L*L)/(3.0*E*I);
end CantileverBeam;
  
```

The CantileverBeam is an example from OpenTURNS documentation that has been modeled in OpenModelica using the new uncertainties support. The variables with `Uncertainty.sought` are the outputs from the Modelica model and the variables for which the uncertainty should be computed. The variables with `Uncertainty.given` are inputs to the Modelica model and they can be varied by OpenTURNS with the given distribution. The variables with `Uncertainty.refine` are both input/output.

OpenTURNS connection in OpenModelica

A script to generate the OpenTURNS interface files and to run the generated script in OpenTURNS for the CantileverBeam is given below:

```
loadFile("TestModels/OpenTURNSTests.mo");

pyscript :=
  buildOpenTURNSInterface(
    OpenTURNSTests.CantileverBeam,
    "CantileverBeam_template.py"); getErrorString();

"run python script";
otlog := runOpenTURNSPythonScript(pyscript); getErrorString();
"read OpenTurns output log";
readFile(otlog); getErrorString();
```

The new OpenModelica API that handles the connection to OpenTURNS is given by two new functions:

- `buildOpenTURNSInterface` – function that builds the OpenModelica model, OpenTURNS wrapper and the OpenTURNS python script (by populating the given template with the information from the Modelica model).
- `runOpenTURNSPythonScript` – function that runs the uncertainties analysis described in the python script in the OpenTURNS framework

A very simple `CantileverBeam_template.py` to be populated by the OpenModelica with the information from the Modelica model is:

```
from openturns import *
from math import *

<%distributions%>
<%correlationMatrix%>
<%collectionDistributions%>
<%inputDescriptions%>

# Use that function defined in the wrapper
# Create a NumericalMathFunction from wrapper
deviation = NumericalMathFunction("<%wrapperName%>")

dim = deviation.getInputDimension()
print 'Dimension of input to modelica model: '
print dim
print '\n'
```

```
x = NumericalPoint (dim)
x[0] = 10.
x[1] = 20.
x[2] = 30.
x[3] = 30.
print x
y = deviation (x)
print y
```

OpenModelica inserts inside the text template below only the parts given inside “<% %>”, i.e, distributions, correlation matrix, etc. For example this part of the template:

```
<%distributions%>
<%correlationMatrix%>
<%collectionDistributions%>
<%inputDescriptions%>
```

is replaced by:

```
# 4 distributions from Modelica model
distributionE = Beta(0.93,3.2,28000000.0,48000000.0)
distributionF = LogNormal(30000.0,9000.0,15000.0, LogNormal.MUSIGMA)
distributionL = Uniform(250.0,260.0)
distributionI = Beta(2.5,4.0,310.0,450.0)

RS = CorrelationMatrix(4)
RS[2,3] = -0.2

# Initialization of the distribution collection:
collectionMarginals = DistributionCollection()
collectionMarginals.add(Distribution(distributionE,"E"))
collectionMarginals.add(Distribution(distributionF,"F"))
collectionMarginals.add(Distribution(distributionL,"L"))
collectionMarginals.add(Distribution(distributionI,"I"))

"E", "F", "L", "I"
```

The result of running the generated python script inside OpenTURNS is:

```
call pyot from OpenTURNS with script: CantileverBeam_template-
OpenTURNSTests.CantileverBeam.py
"c:/adrpo/openmodelica/uncertainties/OpenTURNSTests.CantileverBeam" -noemit
  -override E=10,F=20,L=30,I=30 -output y
    -> time=1,y=600
    ->         600
Dimension of input to Modelica model:
4

[10,20,30,30]
[600]
Welcome to OpenTURNS version 1.0
```

One can see that the OpenModelica generated simulation code is run by OpenTURNS with the given inputs and the computed output is returned. Similarly, much more complex statistical analyses can be run on the OpenModelica model via the Python script.

Conclusions

The OpenModelica compiler has been extended to support uncertain variables and their distributions. A connection to the OpenTURNS framework for computing with uncertainties has implemented. Complex analyses on the uncertain variables can be performed on the Modelica model via the OpenTURNS connection using Python scripts. All the needed parts from the Python script are generated automatically from the Modelica model and an OpenTURNS wrapper used for execution of the model is automatically generated. Fast parallel execution of the Modelica models has been implemented to shorten the analysis execution time.

Publications Referred to in this Document

1. Daniel Bouskela, Audrey Jardin, Zakia Benjelloun-Touimi, Peter Aronsson, Peter Fritzson “*Modelling of Uncertainties with Modelica*”, 8th International Modelica Conference, Dresden, Germany, 2011