HealthQ
TECHNOLOGIES

QSS Solver and LIQSS2

AP de Villiers

30 May 2016

QSS Solver Stats

- Some stats
  - 891 documents in total
  - 59 C files
  - 79 Cpp files
  - 142 header files
  - Many makefiles, scripting file extentions and .so library files as well
- SVN Repo creation date: 2012-03-18
- QSS Solver version 3.1
- 114 MB in size
- Active commits on repo

Why we emulate QSS Solver?

- We want to incorporate the LIQSS method into OpenModelica
- The QSS literature does not provide enough algorithmic details
- We can use the code of QSS Solver to understand the in-depth detail of LIQSS methods
- We gain access to additional info that we require related to specific models
- We can use the research to write a LIQSS2 solver in OpenModelica.

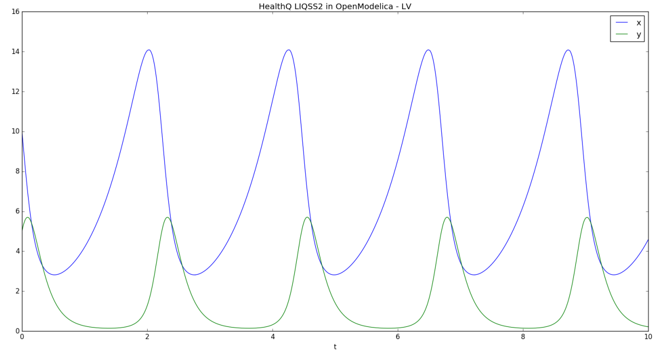
Why is this not a simple process?

The main reasons include the following:

- No comments are present in QSS Solver
- Variable abbreviations are often used
- Most of the procedures in the code are not explained/described in the QSS literature at all.
- The templating section of code has not been studied at all.
- Function pointers make it difficult to resolve actual function calls.

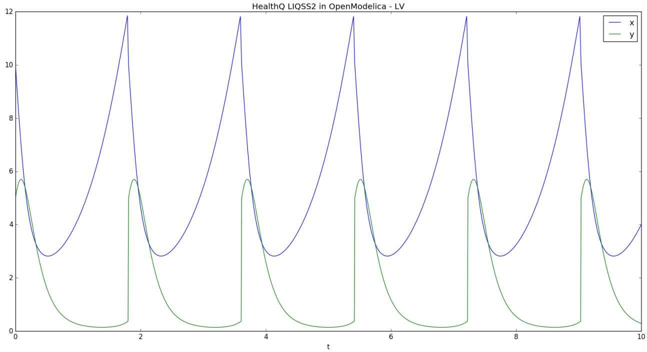
What we have done to date

- LIQSS2 in OpenModelica (Lotka-Volterra)



What we have done to date

- LIQSS2 in OpenModelica with simple events (Lotka-Volterra)



Comparing these results

- For this Lotka-Volterra model, there is very little difference between the model with events and the one without events, as the state variables are all merely reinitialized during the event.
- The simulation thus restarts the simulation in the same manner during each simulation.
- No root finding was done in the code for this kind of event in QSS Solver.
- The timing of when state variables must be updated is simple, since we have a simple model.

Moving on to more detailed models

- We looked at the bouncing ball example
- In our model the bouncing of the ball is irregular.
- We then looked at how QSS Solver addresses the model.
- Some interesting things were noticed:
    - The model created a .c file with many more variables created than in the Lotka-Volterra model.
    - Limited understanding of these variables and there importance.
    - The iterative algorithmic implementation has changed – a different algorithm is used.
- Graph based operations found:
    - In the Bouncing ball example, the update times of the state variables may abruptly change according to a function "SC_update".
    - This function often does not influence the outcome of the simulation, but it has an effect.
    - This function makes use of a graph based structure to recompute the update times.
    - This is not a trivial function — it contains many operations and calculations.
    - The reason for the usage of this function is unclear.

Challenges

- We go into the code of QSS Solver and find the sections of code where our model(s) have an effect.
- The reasoning and meaning needs to be deciphered — we need to understand the math and implementation considerations.
- We need to map the method into OpenModelica's framework.
- We need to make sure we get the same simulation results as QSS Solver in OpenModelica.

Findings

- The variables (number of variables) created in the .c file change according to the model in the .mo file.
- The iterative algorithmic implementation changes according to how the model is dissected.
- Events are incorporated in the .c file from the .mo file.
- The reasoning behind many of the variables are still unclear.
- The operations of certain functions and their impact are not clear and have not been explored.

- General:
  - The info of LIQSS in the literature is vastly different from the implementation in QSS Solver.
  - The reason for the existence of QSS Solver is becoming clear. This answers our question of why LIQSS solvers have not been implemented in OpenModelica.
  - QSS Solver is a big body of work and requires time to work through.

Other considerations

- Symbolic differentiation of second order derivatives — Martin
- Event handling
- Understanding/establishing what can be ignored (for now)