VHM simulation with Cpp-runtime in OpenModelica

Anton de Villiers[*]

April 1, 2016

1 Introduciton

Franke *et al.* [1] investigated the exploitation of Cpp (C++) for Modelica code optimization. They claim that "a publically available application example demonstrates the achievements. CPU times obtained with the OpenModelica [2] C++ runtime are significantly faster than CPU times obtained with the C runtime or with Dymola."

We therefore decided to investigate this claim by focusing on the VHM. The first steps involve building OpenModelica with the `CppRuntime` flag enabled.

2 Results

Some preliminary numerical tests were performed with the following parameters:

- stopTime = 100
- tolerance = 1e-6
- numberOfIntervals = 500

We investigated the DASSL solver with the C runtime. For the C++ runtime, we investigated the following solvers:

- DASSL
- CVode
- IDA

An Intel(R) Core(TM) i7-4710MQ CPU @ 2.50GHz computer containing 8 processors and 16GB RAM, using operating system Linux Ubuntu 14.04 was used for the tests.

Graphical results are shown in Figure 2.1 and Figure 2.2.

A breakdown of the execution times are provided in Table 2.1. Notice that the compilation times are larger for the C++ runs as opposed to the C run, but the simulation times are shorter for the C++

[*] HealthQ Technologies, Office 9, First Floor, The Woodmill Lifestyle, Vredenburg Road, Devon Valley, Stellenbosch, 7600, South Africa

runs. The DASSL and CVode solvers seem to provide the quickest execution times for the VHM in C++ runtime.

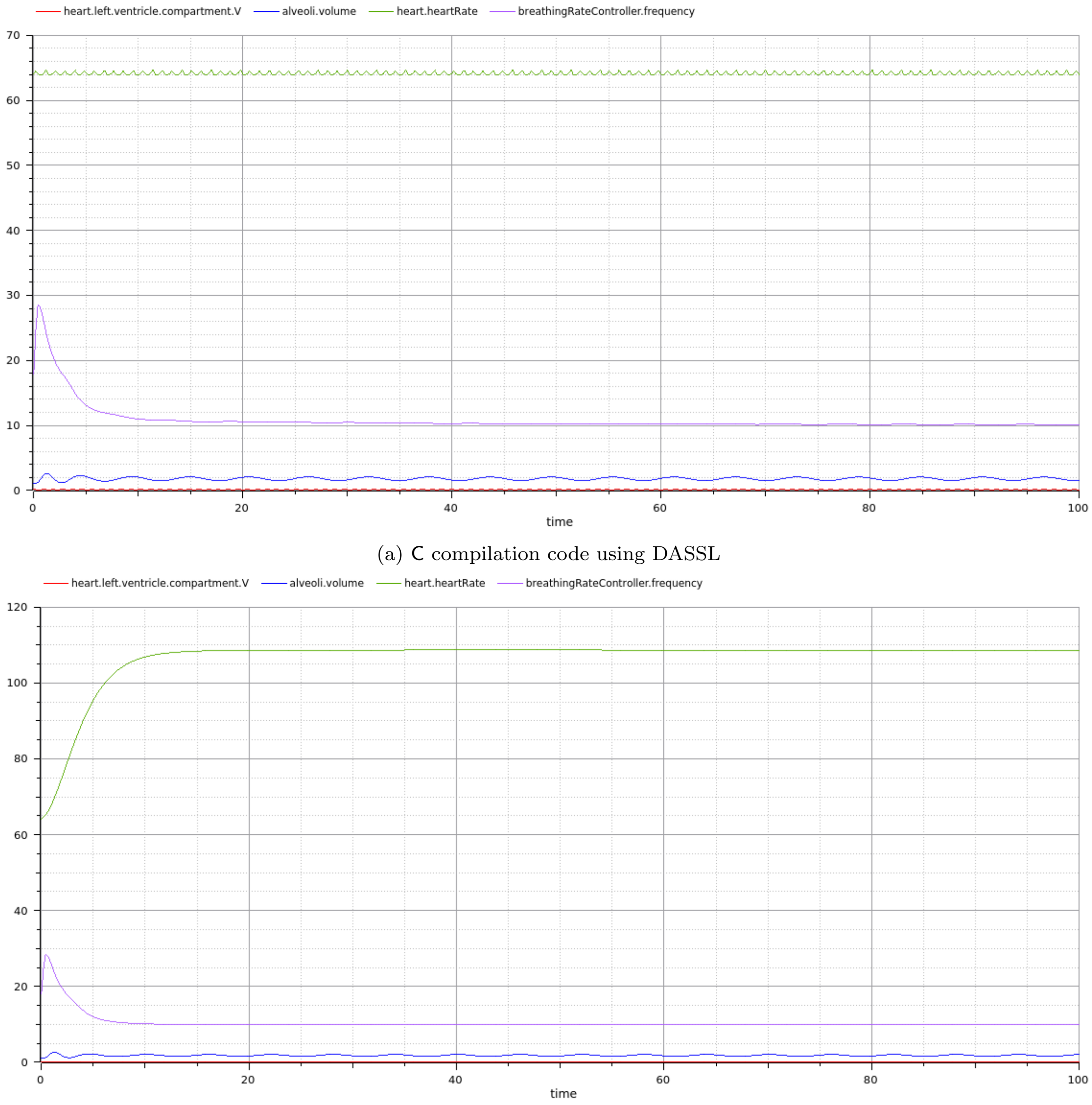
Runtime	Solver	Times	
C	DASSL	timeFrontend	1.605435356
		timeBackend	2.04504742
		timeSimCode	0.328685344
		timeTemplates	0.799143101
		timeCompile	5.007660417
		timeSimulation	31.422863225
		timeTotal	41.208929966
C++	DASSL	timeFrontend	1.505189705
		timeBackend	1.982867962
		timeSimCode	0.3326616499
		timeTemplates	2.469315036
		timeCompile	21.191103299
		timeSimulation	4.864816006
		timeTotal	32.346041009
C++	CVode	timeFrontend	1.521981099
		timeBackend	1.997048266
		timeSimCode	0.320210118
		timeTemplates	2.463317947
		timeCompile	21.14042106
		timeSimulation	4.865331074
		timeTotal	32.308397227
C++	IDA	timeFrontend	1.513984494
		timeBackend	1.993646609
		timeSimCode	0.33445055699
		timeTemplates	2.501393176
		timeCompile	21.10962507
		timeSimulation	13.587780533
		timeTotal	41.040962892

Table 2.1: Simulation times in seconds.

References

[1] Franke R, Walther M, Worschech N, Braum W & Bachmann B, 2015, *Model-based control with FMI and a C++ runtime for Modelica*, Proceedings of the 11th International Modelica Conference, Versailles, France, pp. 339–347.

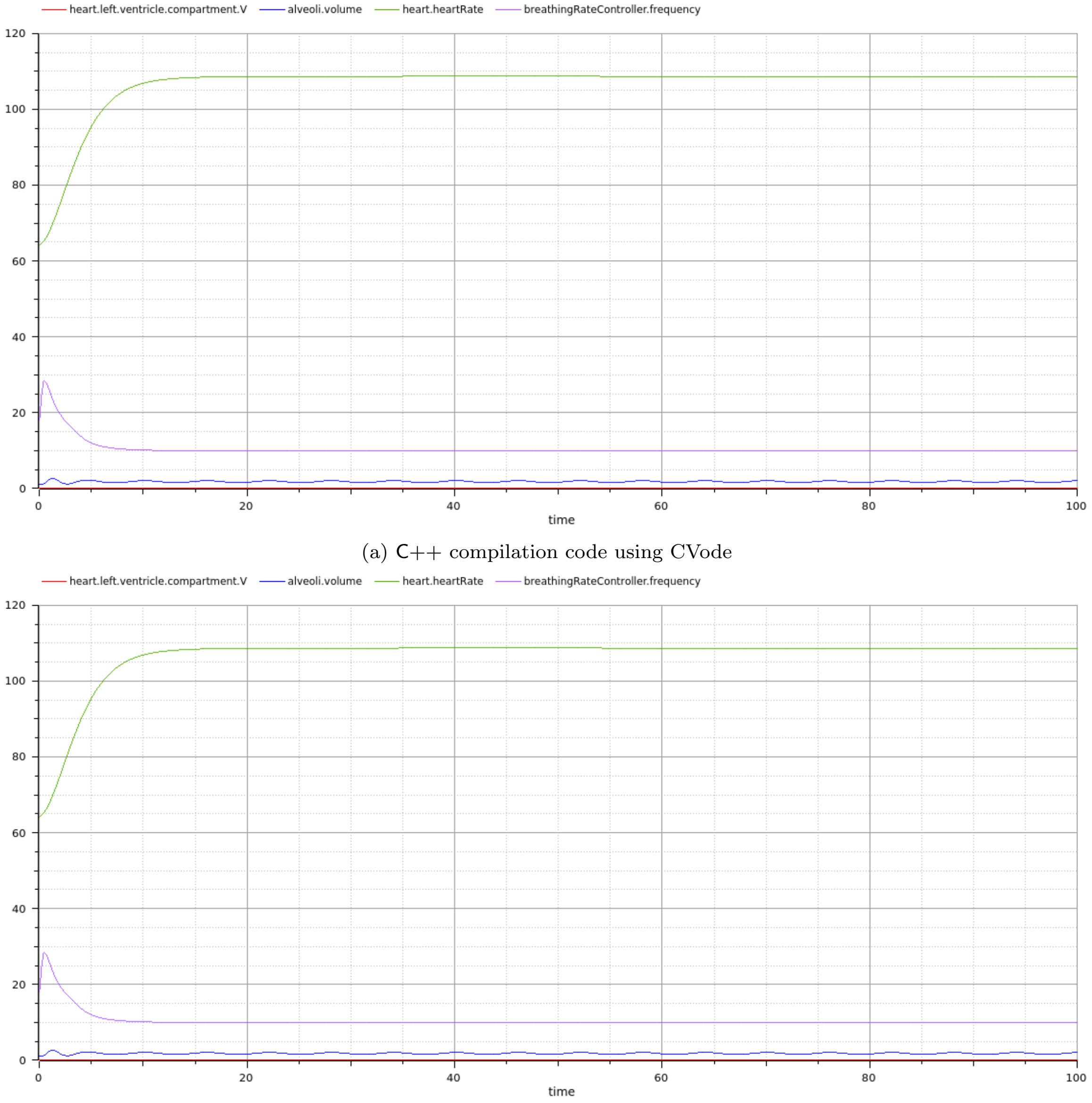
[2] OpenModelica, 2016, *Open Source Modelica Consortium*, [Online], Cited 15th March 2016, Available from `https://openmodelica.org/`



(a) C compilation code using DASSL

(b) C++ compilation code using DASSL

Figure 2.1: Numerical results of the DASSL solvers in (a) C and (b) C++ for the VHM.



(a) C++ compilation code using CVode

(b) C++ compilation code using IDA

Figure 2.2: Numerical results of the (a) CVode and (b) IDA solvers in C++ for the VHM.