

Towards Design Optimization with OpenModelica Emphasizing Parameter Optimization with Genetic Algorithms

Hubert Thieriot^a, Maroun Nemer^a, Mohsen Torabzadeh-Tari^b, Peter Fritzson^b, Rajiv Singh^c, John John Kocherry^c

^aCenter For Energy and Processes, MINES ParisTech, Palaiseau, France

^bPELAB Programming Environment Lab, Dept. Computer Science, Linköping University, SE-581 83, Sweden

^cEvonik Energy Services, Pvt. Ltd., Corporate Office, Noida 201 301, India

Abstract

One of the main goals when modeling a physical system is to optimize its design or configuration. Currently existing platforms are often dependent on commercial software or are based on in-house and special-purpose development tools. These two alternatives present disadvantages that limit sharing and reusability. The same assessment has partly motivated the origin of the Modelica language itself. In this paper, a new optimization platform called OMOptim is presented. Intrinsically linked with OpenModelica, this platform is mainly aimed at facilitating optimization algorithm development, as well as application use together with models. A first version is already available and three test cases of which one using respectively Dymola and two using OpenModelica are presented. Future developments and design considerations of OMOptim but also of related OpenModelica computation functions are also discussed.

Keywords: Optimization, model-based, parameter, genetic algorithm, Modelica, modeling, simulation

1. Introduction

Model-based product development is an approach where a computer-based model of the product is built and refined before the actual production, to reduce costs, increase quality, and shorten time-to-market. Optimization is often used to improve product quality or design. Several types of optimizations can be used with these goals in mind. This can either concerns parameter or configuration optimization (e.g. which selection of the best components or connection paths to use in a defined process). Some design tasks also need a dynamic optimization to benchmark different configurations. For the user but also for the developer of such algorithms, two main issues can be noticed. The first issue concerns the development platform itself. The developers can either use a commercial platform (e.g. MatLab connected

with a external simulator) or develop their own environment. The disadvantages of the first option are mainly the proprietary aspects of such tools which makes it harder to modify and extend, and also the involved license fees. The latter solution needs more development time and reduce exchange opportunity with other teams. Another important issue of model-based optimization lies in the computation time. Optimization applications often requires a large number of iterations and thus, a long time to give interesting results. This paper presents an initiative to limit these two main issues by developing an open-source optimization platform for OpenModelica (OMOptim) involving generation of efficient source code for multi-core computer architectures for increasing simulation performance.

1.1. Structure of the Paper

This paper first presents the context and motivation of the OMOptim development. A general review of optimization methods is then presented. The next sections successively describe the first version of OMOptim, an example of an application already

Email addresses: hubert.thieriot@mines-paristech.fr (Hubert Thieriot), maroun.nemer@mines-paristech.fr (Maroun Nemer), mohsen.torabzadeh-tari@liu.se (Mohsen Torabzadeh-Tari), peter.fritzson@liu.se (Peter Fritzson), r.singh@evonik-es.in (Rajiv Singh), jj.kocherry@evonik-es.in (John John Kocherry)

implemented and some concluding words about the intended future of this platform.

2. Requirements

The Center for Energy and Processes of Mines ParisTech school is involved in the CERES project [1] concerning industrial processes optimization. In this project, the best process technologies and heat recovery topology should be chosen simultaneously with mini-mum costs and environmental impacts. PELAB on the other hand is involved in the SSF Proviking EDOp project [2], concerning dynamic optimization for large industrial optimization problems, targeting both para-metric as well as dynamic optimization. This paper aims at building a bridge between these two projects with a common open source optimization platform. Thus, it should be ergonomic and efficient enough to use but also allow development of algorithms in the environment. A first version of this tool called OMOp-tim has been developed and is described below. Besides this goal, one critical issue will be the simulation (and thus) optimization time. Therefore, optimization algorithms but also the simulation tool efficiency should be very high. This paper briefly presents current and intended developments which go in this direction.

3. Optimization

This project aims at solving several different optimization problems, and in order to do this efficiently, a number of different solution techniques are required. Optimization problems can be classified according to several criteria e.g. existence of constraints, the nature of variables, and the nature of equations involved. A large number of optimization algorithms have been developed over the last decades to solve these different problems. One can roughly divide them in two families: gradient based methods and meta-heuristics algorithms.

3.1. Gradient based methods

The gradient based family contains numerical linear and non linear programming methods. These algorithms require substantial gradient information and are often used to improve a solution near a starting point. Applied on simple models, they offer an efficient way to find global optimum. However, many

engineering optimization problems are highly non-linear and present several optima. Such problems create numerical difficulties (like discontinuities) for this family algorithms and result can depend on initial point defined by the user.

3.2. Meta-heuristic algorithms

Meta-heuristic algorithms present a common characteristic: they combine rules and randomness to imitate natural phenomena. Within such methods, derivative computation is unnecessary. Most developed methods are evolutionary algorithms and genetic algorithms which are based on biological evolution formulation [3] but also tabu search, which reproduces animal behavior [4]. Simulated annealing is another meta-heuristic method based on physical annealing process [5].

3.2.1. Genetic algorithms and evolution strategies

A genetic algorithm (GA) is based on natural evolution and reproduces its main operations: reproduction, crossover and mutation. The initial theory has been proposed by Holland [6] and Goldberg [3] among others. An individual is represented by a genome which contains values of decisive parameters. For each individual, fitness values are calculated; these fitness values correspond to the objectives we want to minimize or maximize. A population is initially created by assigning random values to decisive parameters for each individual. New generations are created by combination of parents and innovation is introduced by mutation step. At each generation, a selection operation is followed which keep only the best individuals according to the fixed objectives but also following diversity parameters. Evolution strategies mainly differ from Genetic algorithms (GAs) in parameters coding: while GAs use binary coding and operations, evolution strategies use real coded parameters [7]. By extension, evolution strategies are often called genetic algorithms. These methods have largely been applied to estimate parameter values which minimize one or several objectives. It is indeed independent of problem type and can be applied to constrained or unconstrained problems, can have discrete or continuous variables, can follow one or several objectives and can be applied to linear or non-linear problems. Evolution strategies and more generally meta-heuristic algorithms present several advantages. First, they can be applied to complex engineering problems. They also do not need any

particular initialization point and are therefore independent of it. Finally, they tend to escape local optimum problems (e.g. with highly discontinuous problems). However, for linear and simple non-linear problems, linear or non-linear programming methods are much more suited and efficient (especially because of specific formulation and gradient information).

4. OMOptim 0.9

4.1. Goals

OMOoptim intends to be a platform where different families of optimization algorithms can be implemented and linked with the OpenModelica simulator but also with other tools e.g. using FMI (Functional Mock-up Interface) [8]. Figure 1 illustrates its high-level design concept.

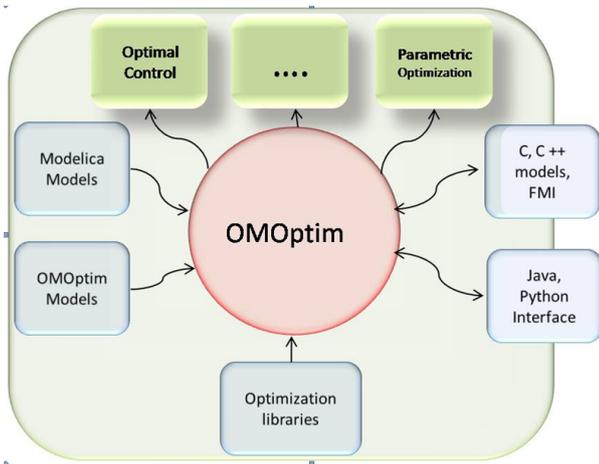


Figure 1: Top-level conceptual view of the OMOptim model-based optimization tool in OpenModelica.

4.2. Implementation

A first version of OMOptim including a graphical user interface has been developed in C++ and already tested on several use-cases (cf. Section 5). This version uses the OpenModelica API to read and eventually modify the model through the Corba communication protocol [9].

This version can only run meta-heuristic optimization methods since at this time, it does not have access to information about derivatives, even though OpenModelica can produce such information. As previously stated, only input variables specification and output variables reading are needed for such methods. Specifying input variables and reading results

is done using input and output text files. To implement meta-heuristic algorithms, an efficient and adapted framework has been used (ParadisEO library [10]). OMOptim already includes several genetic algorithms, e.g. NSGA2, SPEA2 [11] or self-adaptive versions [12].

4.3. User interface

At the same time, a GUI has been developed allowing graphical selection of optimization variables, parameters and objectives (Figure 2) but also reading results.

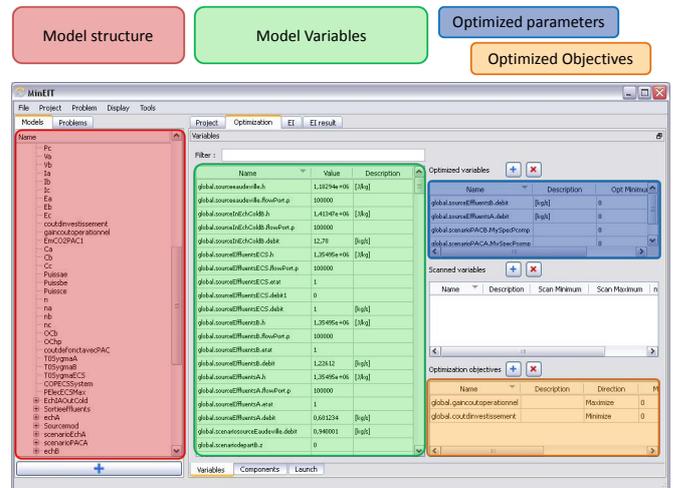


Figure 2: Parameters and objectives selection in the OMOptim optimization problem definition.

5. Test cases

Three test cases are presented here. The first uses Dymola as a simulation tool on an industrial application, but still uses OpenModelica to access the model structure. The second shows a small example application with OpenModelica. The third uses OpenModelica on an industrial application and an optimization module which is currently executed separately, but will be integrated with OpenModelica. As previously stated, meta-heuristic algorithms can interact with simulation tool using only input and output files. Thus, it is possible to interact with most simulation tools. However, in the future, all OMOptim algorithms may not be compatible with other simulators than OpenModelica (cf. section 6)

5.1. Heat-pump application using Dymola for simulation

A first application has been done which concerns a multi heat-pump system in a food industrial process [13]. This system consists of three heat-pumps used to heat-up solutions of the process. These three heat-pumps are connected to a heat-recovery stream. The model integrates dynamic items e.g. hot water tank emptying and filling during simulation (cf. Figure 3).

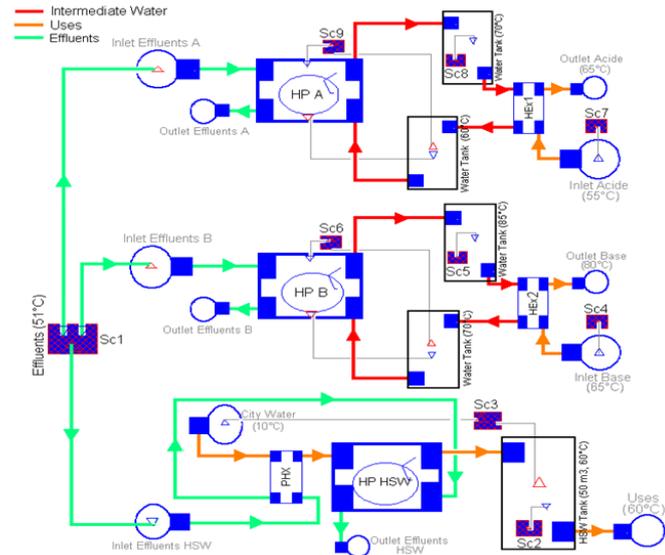


Figure 3: Modelica model of an industrial process being optimized.

The optimization consists in finding optimal flow repartitioning of the heat-recovery stream but also optimal powers of these heat-pumps, including the possibility to disable one or several heat-pumps. Two objectives are considered in this optimization: decreasing operational cost and investment cost. An auto-adaptive genetic algorithm has been developed for this study in OMOptim [13][12]. This genetic algorithm includes standard deviation of each genome parameter in the genome itself of the genetic algorithm. Therefore, the variation amplitude between each generation is itself submitted to modification and selection.

OMOptim allows the user to obtain several optimal configurations according to the two objectives followed i.e. investment and operating cost. Moreover, a sensitivity analysis has been performed to analyze the impact of CO2 carbon tax on optimum configurations (Figure 4).

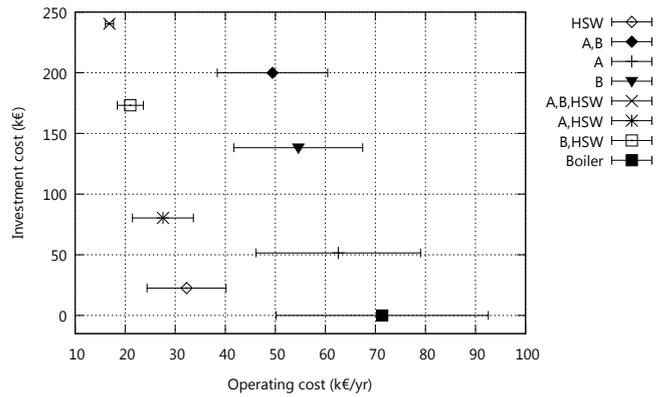


Figure 4: Investment and operation costs for optimal configurations (horizontal bars correspond to carbon tax variation sensitivity).

5.2. A Linear actuator application using OpenModelica

The model here consists of a linear actuator with a spring damped stopping [14, p. 583]. The model configuration is presented on Figure 5.

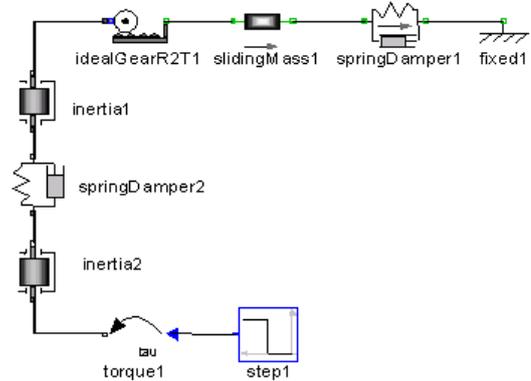


Figure 5: Linear actuator model

A reference response is generated considering a first order system. This response is defined by a first order ODE : $0.2 * \dot{y}_{ref}(t) + y_{ref}(t) = 0.05$. The optimization consists in making the resulting linear actuator behavior be as close as possible to this reference response. To achieve this, the damping parameters d_1 and d_2 of both spring dampers are considered as free variables to be determined by the optimization algorithm. The objective function corresponds to the integral of square deviation along simulation time T : $f(d) = \int_0^T (y(t) - y_{ref}(t))^2 dt$.

With obtained parameters ($d_1 = 4.90$ and $d_2 =$

19.88), the behavior suits the reference response well (cf. Figure 6). These results were obtained in less than five minutes on a standard Intel Core2 Duo @ 2.53 GHz.

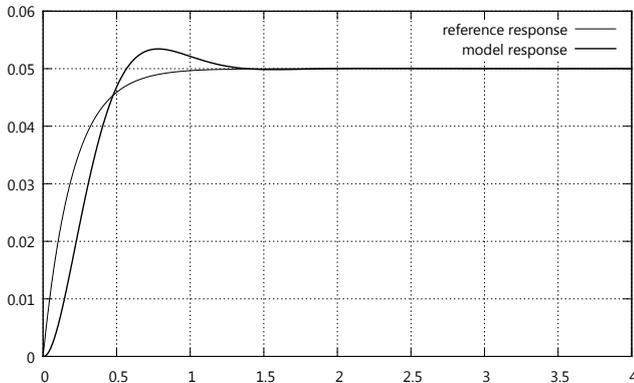


Figure 6: Model and reference responses after optimization of damper parameters

5.3. A dynamic optimization using an external SQP module

This third test case concerns a power plant regulation. It is only described very briefly here - a more detailed presentation is planned in a future paper. This application has been run using an external Sequential Quadratic Programming (SQP) optimization module.

In power plants, the main steam temperature control regulates the spray (attemperator) flow rate. Precise modeling of super heater dynamics and improving the quality of control of the superheated steam temperature is essential to improve the efficiency of the Boiler. In addition to this, the physical constraints of the turbine blades are also met using this control strategy. This control methodology is based on an adaptive prediction of the steam temperature trends. The architecture of the newly developed control system is similar to that of conventional boiler but the temperature feedback is given from the model instead of a sensor as shown in Figure 7.

A simple heat exchanger model is adapted to model the first stage of super heater regarding steam temperature, steam flow and flue gas temperature as measurements. This resulted into a set of algebraic differential equations which captured the behavior of the super heater along with the attemperator.

A SQP optimizer is used to calculate the spray flow, driven by an objective function to find the least

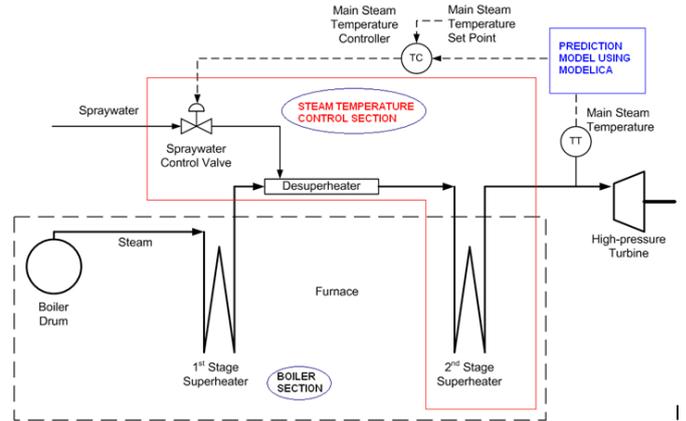


Figure 7: Advanced steam temperature control strategy for power plant

square error between the predicted and set point of steam temperature for a defined control horizon. Dynamic constraints are considered for spray and metal temperatures to consider the metal strains.

The first results are promising. However, this function is a separate module that is not yet integrated with the available version of OMOptim. This integration is planned in the near future.

6. Future work

6.1. OMOptim Structure Evolution

OMOptim intends to become an attractive framework to develop and execute optimization algorithms for Modelica users. To achieve this, its structure should be flexible enough to address the needs of many different kinds of optimization. The structure should also provide an efficient and ergonomic way to develop special-purpose algorithms including sharing and usage. Like a Modelica library, it would be pertinent and useful to list available optimization algorithms in libraries sharable within the Modelica community. Moreover, the structure should be able to support the combination of several algorithms working together. It should for example be used to apply a meta-heuristic optimization function with an objective function computed from another function (e.g. the objective could itself be the result of a sensitivity analysis). In some cases, it should also be possible to create new algorithms by graphically connecting existing optimization modules like in component-based modeling.

6.2. Hybrid Optimization

Meta-heuristic optimization algorithms can be coupled with local search functions [15]. This combination intends to combine advantages of both families. Meta-heuristics allow spreading populations over a large domain and thus limit the risk of obtaining a local optimum solution. Local search functions can lead to a faster convergence and to more precise results (e.g. [16] or [17]).

To achieve hybrid optimization implementation, a stronger link with OpenModelica should be built. In particular, gradient information should be communicated to optimization methods. The first developments in this direction are currently under way.

6.3. Dynamic optimization

Dynamic optimization requires modifying model parameters while performing the simulation. This functionality assumes the development an interface between OpenModelica and OMOptim while the former is computing. First trials have been done in this direction, using the new online interactive simulation facility of OpenModelica. More specifically, integration of a Sequential Quadratic Programming optimizer within OMOptim is planned in the near future (cf. section 5.3).

6.4. Parallelization for Efficient Computation

Applying parallelism and parallel compilation techniques at many levels of the problem, from problem formulation to inlining the solver and software pipelining, is being addressed in this project [18]. The constraints of the optimization problem can often be handled in parallel. In this case large system models can be restructured to smaller sub-system models. The PELAB research group at Linköping University has a long tradition of handling the compilation process in parallel, optimizing it, and adapting it for multi-core architectures. Some recent encouraging results [19] about using GPU architectures instead of CPU caused PELAB to invest in a two-teraflop (peak) Nvidia Fermi GPU that will be used in this project. Another step is to extend the support of efficient event-handling in parallelized code in order to also handle hybrid models.

6.5. Optimization Performance Profiling and Debugging

One current disadvantage of using high-level equation based languages [14] as well as other high-level

simulation tools is the poor support for performance profiling and debugging. This will be even more pronounced when an engineer wants to trace the reason to why an optimization is too slow or has failed. There exists a substantial expertise at PELAB regarding debugging and traceability technology in integrated environments. We are planning to use this as a basis for a profiling feature in the optimization platform that is needed for tracing the causes of problems bottle-necks in the model.

7. Related Work

7.1. jModelica

The current Modelica language does not include formulating optimizations problems. However, a language extension called Optimica [20] has been developed by JModelica (www.jmodelica.org). JModelica offers an efficient platform for dynamic optimization and works in close collaboration with the model since it has an integrated Modelica compiler.

7.2. Dymola optimization library

The Dymola commercial tool from Dassault Systems [21], Dymola has its own optimization library, containing genetic algorithms. Another product from Dassault Systems is Isight [22] that supports process flow optimization with genetic algorithms. The main disadvantage of these two products is their closeness.

7.3. Meta-heuristic algorithms

Several tools may link meta-heuristic optimization methods to different simulators. One can cite OptiY [23], modeFrontier [24], Isight [22], or GenOpt [25]. They propose a rich list of implemented algorithms and can be used with nearly all simulators (all these tools interact with simulation software using input file modification and output file reading). Excepting GenOpt, all these softwares are commercial.

7.4. What should OMOptim offer

OMOptim aims to offer OpenModelica users an extension opening new opportunities. Especially, it intends to be a shared and open platform where scientists could develop optimization algorithms and apply them to Modelica models.

OpenModelica has been chosen for its opening and its substantial development rhythm. Also, OpenModelica supports symbolic differentiation which allows robust and advanced numerical methods, very

useful in optimization problems. This could be especially useful for the development of hybrid algorithms (cf. section 6.2).

Parallelism is also an intended development direction. Applying parallelism and parallel compilation techniques at many levels of the problem, from heuristic simulation repartition to inlining the solver and software pipelining, is being addressed in this project. For example, population based meta-heuristic optimization methods present high parallel scalability.

Concerning dynamic optimization or components/connections that change during simulations, the Modelica language doesn't yet support structural dynamism, i.e. changes in the causality during simulations. However, with a little relaxation of this requirement the environments would be much flexible and better suited for optimization tasks [26].

References

- [1] Ceres project : www.ecelear.com/ecelear_i_15/.
- [2] Edop project : <http://openmodelica.org/index.php/research/omoptim/edop>.
- [3] D. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison-wesley, 1989.
- [4] F. Glover, Future paths for integer programming and links to artificial intelligence, *Comp. Oper. Res.* 13 (1986) 533–549.
- [5] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, *Science* 220 (1983).
- [6] J. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI, 1975.
- [7] T. Bck, H. Schwefel, An overview of evolutionary algorithms for parameter optimization, *Evolutionary computation* 1 (1993) 1–23.
- [8] Itea2, modelisar : www.itea2.org.
- [9] Openmodelica system documentation, available on <http://www.openmodelica.org>.
- [10] A. Liefooghe, M. Basseur, L. Jourdan, E.-G. Talbi, *Paradiseo-moeo: A framework for evolutionary multi-objective optimization*, 2007.
- [11] E. Zitzler, M. Laumanns, L. Thiele, Spea2: Improving the strength pareto evolutionary algorithm, in: *EUROGEN*, Citeseer, 2001, pp. 95–100.
- [12] K. Deb, H. Beyer, Self-adaptation in real-parameter genetic algorithms with simulated binary crossover, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, Citeseer, 1999, pp. 172–9.
- [13] R. Murr, H. Thieriot, A. Zoughaib, D. Clodic, Multi-objective optimization of a multi water-to-water heat pump system using evolutionary algorithm, Submitted to *Applied Energy*. ()
- [14] P. Fritzson, *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*, Wiley-IEEE Press, 2004.
- [15] A. Hopgood, L. Nolle, A. Battersby, Hybrid genetic algorithms: A review (2006) –.
- [16] M. Fesanghary, M. Mahdavi, M. Minary-Jolandan, Y. Alizadeh, Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems, *Computer methods in applied mechanics and engineering* 197 (2008) 3080–3091.
- [17] S. Katare, A. Bhan, J. Caruthers, W. Delgass, V. Venkatasubramanian, A hybrid genetic algorithm for efficient parameter estimation of large kinetic models, *Computers & chemical engineering* 28 (2004) 2569–2581.
- [18] H. Lundvall, P. Fritzson, Automatic parallelization using pipelining for equation-based simulation languages, in: *In proceedings of the 14th Workshop on Compilers for Parallel Computing (CPC'2009)*, Zurich, Switzerland.
- [19] P. stlund, K. Stavker, P. Fritzson, Parallel simulation of equation-based models on cuda-enabled gpus, Submitted to *EuroPar* (2010).
- [20] J. Akesson, K.-E. Arzen, M. Gafvert, T. Bergdahl, H. Tummescheit, Modeling and optimization with optimica and jmodelica.org–languages and tools for solving large-scale dynamic optimization problems, *Computers & Chemical Engineering* 34 (2010) 1737–1749.
- [21] Dassault systemes, www.3ds.com.
- [22] Isight product page : <http://www.simulia.com/products/isight.html>.
- [23] Optiy, multidisciplinary analysis and optimization, <http://www.optiy.eu>.
- [24] Modefrontier, <http://www.modefrontier.com>.
- [25] Genopt, <http://simulationresearch.lbl.gov/go/>.
- [26] H. Nilsson, G. Giorgidze, Exploiting structural dynamism in functional hybrid modeling for simulation of ideal diodes, in: *Eurosim 2010*.