SIMS 2007

The 48th Scandinavian Conference on Simulation and Modeling

Göteborg (Särö)

30–31 October, 2007

Conference Proceedings

Organized by

Linköping University

Lund University

Edited by

Peter Bunus, Dag Fritzson and Claus Führer

Published by

Linköping University Electronic Press
Linköping, Sweden, 2007

Linköping Electronic Conference Proceedings, No. 27
Linköping University Electronic Press
Linköping, Sweden, 2007

ISSN 1650-3686 (print)
ISSN 1650-3740 (online)
http://www.ep.liu.se/ecp/027/

Table of Contents

Additional Papers (not presented during the conference)

Preface

Scandinavian Simulation Society and was held on October 30-31, 2007 in Göteborg (Särö), Sweden. The Scandinavian Simulation Society (SIMS) has members in the four Nordic countries Denmark, Finland, Norway and Sweden. The SIMS history goes back to 1959 and is organised as a federation of the regional Scandinavian societies. SIMS is also a member society of EUROSIM - the Federation of European Simulation Societies.

The purpose of the annual SIMS conference is to cover broad aspects of modelling and simulation and scientific computation. Every year a special topic is considered, this year multibody dynamics has been the main focus of the conference.

Following the tradition of previous conferences in this series, the scientific program consisted of 2 tutorial sessions, 3 keynote addresses and 24 contributed talks, distributed over two days. The speakers covered a wide range of topics, including numerical issues in simulation, distributed simulation, co-simulation, modelling and simulation aspects of multibody systems, domain specific models and issues related to simulation environments design. Several speakers discussed applications in automotive, production of pharmaceuticals, heating systems and batch reactors.

The broad range of interests was exemplified by the topics selected for the keynote addresses:

- Gustaf Söderlind from the Centre for Mathematical Sciences Lund University, Sweden presented recent advances in adaptive time-stepping numerical algorithms.
- Pieter J. Mosterman, Senior Research Scientist at The MathWorks, Inc.USA gave an overview of tool infrastructure that is essential for the successful adoption of Model-Based Design at an enterprise-wide level and presented the new directions in modelling and simulation research.
- Sten Henriksson, a semi retired Senior Lecturer at the Computer Science Department of Lund University presented his personal recollections and reflections on the intellectual contributions of computers to science.

The conference was organized in cooperation with Linköping University (Department of Computer and Information Science), Lund University (Center for Mathematical Sciences) and the two simulation societies: Scandinavian Simulation Society and Swedish Simulation Society.

The SIMS 2007 Conference Program Chairs
Peter Bunus
Dag Fritzson
Claus Führer

SIMS 2007

The 48th Scandinavian Conference on Simulation and Modeling

Program Co-Chairs

Peter Bunus
Department of Computer and Information Science
Linköping University, Linköping, Sweden

Dag Fritzson
SKF, Göteborg and Department of Computer and Information Science
Linköping University, Linköping, Sweden

Claus Führer
Center for Mathematical Sciences
Lund University, Lund, Sweden

Programme Committee

Felix Breitenecker
Vienna Univ. of Technology
Dept. of. Analysis and Scientific Computation RG Mathematical Modelling
and Simulation Austria

Erik Dahlquist
Mälardalens Högskola, Sweden

Brian Elmegaard
Department of Energy Engineering, Technical University of Denmark

Hilding Elmqvist
Dynasim AB , Lund, Sweden,

Peter Fritzson
Department of Computer and Information Science
Linköping University, Sweden

Esko Juuso
University of Oulu, Finland

Kaj Juslin
VTT Technical Research Centre of Finland, Finland

Bernt Lie
Høgskolen i Telemark, Norway

Per Lidström
Lund University, Division of Mechanics, Department of Mechanical Engineering, Sweden

Automatic Parallelization of Mathematical Models Solved with Inlined Runge-Kutta Solvers

Håkan Lundvall and Peter Fritzson
PELAB – Programming Environment Lab, Dept. Computer Science
Linköping University, S-581 83 Linköping, Sweden
{haklu, petfr}@ida.liu.se

Abstract

In this work we report preliminary results of automatically generating parallel code from equation-based models together at two levels: Performing inline expansion of a Runge-Kutta solver combined with fine-grained automatic parallelization of the resulting RHS opens up new possibilities for generating high performance code, which is becoming increasingly relevant when multi-core computers are becoming common-place.

We have introduced a new way of scheduling the task graph generated from the simulation problem which utilizes knowledge about locality of the simulation problem. The scheduling is also done in a way that limits communication, to the greatest extent possible, to neighboring processors thus avoiding expensive global synchronization. Preliminary tests on a PC-cluster show speedup that is better than what was achieved in previous work where parallelization was done only at the equation system level.

Keywords: Modelica, automatic parallelization.

1. Background – Introduction to Mathematical Modeling and Modelica

Modelica is a rather new language for equation-based object-oriented mathematical modeling which is being developed through an international effort [5][4]. The language unifies and generalizes previous object-oriented modeling languages. Modelica is intended to become a *de facto* standard. It allows defining simulation models in a declarative manner, modularly and hierarchically and combining various formalisms expressible in the more general Modelica formalism. The multidomain capability of Modelica gives the user the possibility to combine electrical, mechanical, hydraulic, thermodynamic, etc., model components within the same application model.

In the context of Modelica *class libraries* software components are Modelica classes. However, when building particular models, components are *instances* of those Modelica classes. Classes should have well-defined communication interfaces, sometimes called ports, in Modelica called *connectors*, for communication between a component and the outside world. A component class should be defined *independently of the environment* where it is used, which is essential for its *reusability*. This means that in the definition of the component including its equations, only local variables and connector variables can be used. No means of communication between a component and the rest of the system, apart from going via a connector, is allowed. A component may internally consist of other connected components, i.e. *hierarchical* modeling.

To grasp this complexity a pictorial representation of components and connections is quite important. Such graphic representation is available as *connection diagrams*, of which a schematic example is shown in Figure 1 where a complex car simulation model is built in a graphical model editor.

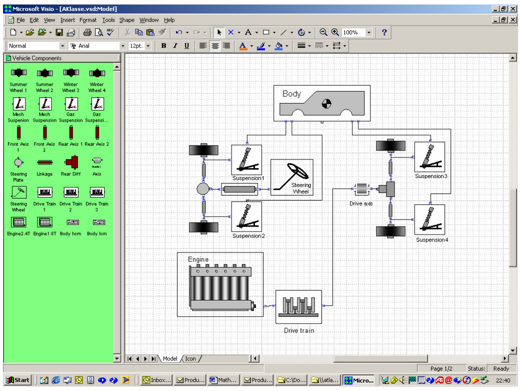


Figure 1. Complex simulation models can be built by combining readily available components from domain libraries.

To summarize, Modelica has improvements in several important areas:

- *Object-oriented mathematical modeling.* This technique makes it possible to create physically relevant and easy-to-use model components, which are employed to support hierarchical structuring, reuse, and evolution of large and complex models covering multiple technology domains.

- *Acausal modeling.* Modeling is based on equations instead of assignment statements as in traditional input/output block abstractions. Direct use of equations significantly increases re-usability of model components, since components adapt to the data flow context in which they are used. This generalization enables both simpler models and more efficient simulation. However, for interfacing with traditional software, algorithm sections with assignments as well as external functions/procedures are also available in Modelica.

- *Physical modeling of multiple application domains.* Model components can correspond to physical objects in the real world, in contrast to established techniques that require conversion to "signal" blocks with fixed input/output causality. In Modelica the structure of the model becomes more natural in contrast to block-oriented modeling tools. For application engineers, such "physical" components are particularly easy to combine into simulation models using a graphical editor (Figure 1).

2. The OpenModelica Open Source Implementation

In this work the OpenModelica software is used which is the major Modelica open-source tool effort.

The OpenModelica environment is the major Modelica open-source tool effort [3] consists of several interconnected subsystems, as depicted in Fig. 2. Arrows denote data and control flow. Several subsystems provide different forms of browsing and textual editing of Modelica code. The debugger currently provides debugging of an extended algorithmic subset of Modelica. The graphical model editor is not really part of OpenModelica but integrated into the system and available from MathCore without cost for academic usage. In this research project two parts of the OpenModelica subsystem is used.

- *A Modelica compiler subsystem*, translating Modelica to C code, with a symbol table containing definitions of classes, functions, and variables. Such definitions can be predefined, user-defined, or obtained from libraries.
- *An execution and run-time module.* This module currently executes compiled binary code from translated expressions and functions, as well as simulation code from equation based models, linked with numerical solvers.

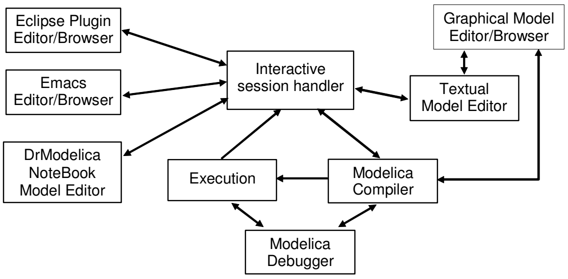

Eclipse Plugin Editor/Browser
Emacs Editor/Browser
DrModelica NoteBook Model Editor
Interactive session handler
Graphical Model Editor/Browser
Textual Model Editor
Execution
Modelica Compiler
Modelica Debugger

Figure 2. OpenModelica architecture.

3. Approaches to Integrate Parallelism and Mathematical Models

There are several approaches to exploit parallelism in mathematical models. In this section we briefly review some approaches that are being investigated in the context of parallel simulation of Modelica models.

3.1 Automatic Parallelization of Mathematical Models

One obstacle to parallelization of traditional computational codes is the prevalence of low-level implementation details in such codes, which also makes automatic parallelization hard.

Instead, it would be attractive to directly extract parallelism from the high-level mathematical model, or from the numerical method(s) used for solving the problem. Such parallelism from mathematical models can be categorized into three groups:

- *Parallelism over the method.* One approach is to adapt the numerical solver for parallel computation, i.e., to exploit parallelism over the method. For example, by using a parallel ordinary differential equation (ODE) solver for that allows computation of several time steps simultaneously. However, at least for ODE solvers, limited parallelism is available. Also, the numerical stability can decrease by such parallelization. See for example [13].
- *Parallelism over time.* A second alternative is to parallelize the simulation over the simulated time. This is however best suited for discrete event simulations, since solutions to continuous time dependent equation systems develop sequentially over time, where each new solution step depends on the immediately preceding steps.
- *Parallelism of the system.* This means that the modeled system (the model equations) is parallelized. For an ODE or DAE equation system, this means parallelization of the right-hand sides of such equation systems which are available in explicit form; moreover, in many cases implicit equations can

automatically be symbolically transformed into explicit form.

A thorough investigation of the third approach, automatic parallelization over the system, has been done in our recent work on automatic parallelization (fine-grained task-scheduling) of a mathematical model [1] [11], Fig 3.

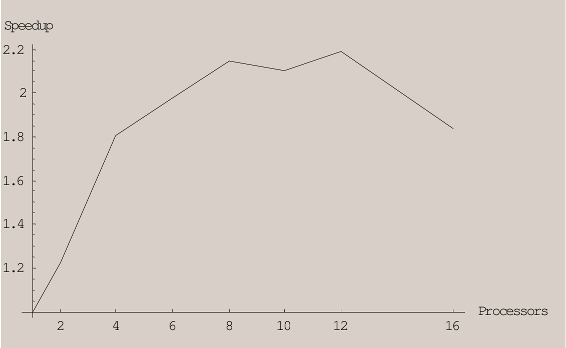


Figure. 3. Speedup on Linux cluster with SCI interconnect.

In this work we aim at extending our previous approach to inlined solvers, integrated in a framework exploiting several levels of parallelism.

3.2 Coarse-Grained Explicit Parallelization Using Computational Components

Automatic parallelization methods have their limits. A natural idea for improved performance is to structure the application into computational components using strongly-typed communication interfaces.

This involves generalization of the architectural language properties of Modelica, currently supporting components and strongly typed connectors, to distributed components and connectors. This will enable flexible configuration and connection of software components on multiprocessors or on the GRID. This only involves a structured system of distributed solvers/ or solver components.

3.3 Explicit Parallel Programming

The third approach is providing general easy-to-use explicit parallel programming constructs within the algorithmic part of the modeling language. We have previously explored this approach with the NestStep-Modelica language [7, 12]. NestStep is a parallel programming language based on the BSP (Bulk-Synchronous Parallel) model which is an abstraction of a restricted message passing architecture and charges cost for communication. It is defined as a set of language extensions which in the case of NestStepModelica is added to the algorithmic part of Modelica. The added constructs provide shared variables and process coordination. NestStepModelica processes run, in general, on different machines that are coupled by the NestStepModelica language extensions and runtime system to a virtual parallel computer.

4. Combining Parallelization at Several Levels

Models described in object oriented equation-based languages like Modelica give rise to large differential algebraic equation systems that can be solved using numerical DAE or ODE-solvers. Many scientific and engineering problems require a lot of computational resources, particularly if the system is large or if the right hand side is complicated and expensive to evaluate. Obviously, the ability to parallelize such models is important, if such problems are to be solved in a reasonable amount of time.

As mentioned in Section 3, parallelization of object oriented equation based simulation code can be done at several different levels. In this paper we explore the combination of the following two parallelization approaches

- Parallelization across the method, e.g., where the stage vectors of a Runge-Kutta solver can be evaluated in parallel within a single time step.
- •Fine grained parallelization across the system where the evaluation of the right hand side of the system equations is parallelized.

The nature of the model dictates to a high degree what parallelization techniques that can be successfully exploited.

We suggest that it is often desirable to apply parallelization at more than one level simultaneously. In a model where two parts are loosely coupled it can, e.g., be beneficial to split the model using transmission line modeling [8] and use automatic equation parallelization across each sub model. In this paper, however, we investigate the possibility of doing automatic parallelization across the equation system and the solver simultaneously. In previous work [1] automatic parallelization across the system has been done by building a task graph containing all the operations involved in evaluating the equations of the system DAE. In order to make the cost of evaluating each task large enough compared to the communication cost between the parallel processors that approach uses a graph rewriting system which merges tasks together in such a way that the total cost of computing and communicating is minimized. In that approach the solver is centralized and runs on one processor. Each time the right hand side is to be evaluated, data needed by tasks on other processors is sent and the result of all tasks is collected in the first process before returning to the solver. As a continuation of this work we now inline an entire Runge-Kutta solver in the task graph before scheduling of the tasks. A similar paper to this was recently published at the Eurosim 2007 conference in Ljubljana. In this paper we add some analysis of the test results gained so far.

Many simulation problems give rise to a DAE system consisting of a very large number of equations but where each equation only depends on a relatively small number of other equations.

Let $f = (f_1,...,f_n)$ be the right hand side of such a simulation problem and let f_i contain equations only depending on equations of components of indices in a range near *i*. This makes it possible to pipeline the computations of the resulting task graph, since evaluating f_i for stage *s* of the Runge-Kutta solver depend only on f_j of stage *s* for *j* close to *i* and on f_i of stage *s-1*. The largest distance between *i* and *j* defines the access distance of the system.

Systems that usually lead to short access distances include discretizations of different kinds. Examples of such systems are systems of partial differential equations, models of compressible fluids in pipes, or mechanical systems with flexible parts.

A task graph of a system where the right hand side can be divided into three parts, denoted by the functions f_1, f_2 and f_3 where f_i only depend on f_{i-1}, inlined in a two stage Runge-Kutta solver is shown in figure 4. In the figure *n_k* represent the state after the previous time step. We call the function f_i the blocks of the system. If we schedule each block to a different processor, let us say f_i is scheduled to p_i, then p_1 can continue calculating the second stage of the solver as p_2 starts calculating the first stage of f_2. The communication between p_1 and p_2 can be non-blocking so that if many stages are used communication can be carried out simultaneous to the calculations. In a shared memory system we only have to set a flag that the data is ready for the next process and no data transfer needs to take place.

The pipelining technique is described in [10]. Here we aim at automatically detecting pipelining possibilities in the total task graph containing both the solver stages and the right hand side of the system, and automatically generating parallelized code optimized for the specific latency and bandwidth parameters of the target machine.

It the earlier approach with task merging including task duplication the resulting task graph usually ends up with one task per processor and communication takes place at two points in each simulation step; initially when distributing the previous step result from the processor running the solver to all other processors and at the end collecting the results back to the solver.

When inlining a multi-stage solver in the task graph each processor only needs to communicate with its neighbor. In this approach however we cannot merge tasks as much since the neighbors of a processor depends on initial results to be able to start their tasks. Thus, instead of communicating a lot at the beginning and at the end, smaller portions are communicated throughout the calculation of the simulation step.

If the task graph of a system mostly has the property of having a narrow access distance, which is required for the pipelining, but only on a small number of places access components in more distant parts of the graph. The rewriting system could also make suggestions to the user about places in the model which would benefit from a decoupling using transmission line modeling if it can be done without loosing the physical correctness of the model.

The task rewriting system is build into the OpenModelica compiler previously mentioned.

5. Pipelining the task graph

Since communication between processors is going to be more frequent with this approach we want to make sure that the communication interferes as little as possible with the computation. Therefore, we schedule the tasks in such a way that communication taking place inside the simulation step is always directed in one direction. The used direction is of no importance, we chose to direct the communication from processors with lower rank to higher ranked processors. Each processor is given a rank from 0 to *p-1*, where *p* is the number of processors used. In this way the lower ranked processor is always able to carry on with calculations even if the receiving processor temporarily falls behind. At the end of the simulation step there is a phase where values required for the next simulation step are transferred back to lower ranked processors, but this is only needed once per simulation step instead of once for each evaluation of the right hand side. Furthermore this communication takes place between neighbors and not to a single master process which otherwise can get overloaded with communication when the number of processors becomes large.

6. Sorting Equations for Short Access Distance

One part of translating an acausal equation-based model into simulation code involves sorting the equations into data dependency order. This is done using Tarjan's algorithm [14] which also finds any strongly connected components in the system graph, i.e., a group of equations that must be solved simultaneously. We assign a sequence number to each variable, or set of variables in case of a strongly connected component, and use this to help the scheduler assign tasks that communicate much within the same processor. When the task graph is generated each task is marked with sequence number of the variable it calculates. When a system with *n* variables is to be scheduled onto *p* processors, tasks marked 1 through *n/p* is assigned to the first processor and so on.

Even though Tarjan's algorithm assures that the equations are evaluated in a correct order we cannot be sure that there is not a different ordering where the access distance is smaller. If for example two parts of the system is largely independent they can become interleaved in the sequence of equations making the access distance unnecessarily large. Therefore we apply an extra sorting step after Tarjan's algorithm which moves equations with direct dependencies closer together. This reduces the risk of two tasks with a direct dependency getting assigned to different processors.

As input to the extra sorting step we have a list of components and a matching defining which variable is solved by which equation. On component represent a set of equations that must be solved simultaneously. A component often includes only one equation. The extra sorting step works by popping a component from the head of the component list and placing them in the resulting sorted list as near the head of the sorted list as possible without placing it before a component on which it depends. See pseudo-code for the algorithm below.

The following data structures are used in the code:

`matching` A map from variables to the equations that solve them.

`componentList` The initial list of components.

`sortedList` The resulting sorted list of components.

```
set sortedList to an empty set of components
while componentList not empty
   set comp to  componentList.PopHead()
   set varSet to the set of all variables
       accessed in any equation of comp
   set eqSet to an empty set of equations
   for each variable v in varSet
       eqSet.insert(matching[v])
   end for

   Devide sortedList into left and right so
   that right is the largest suffix of
   sortedList where Intersection(right,eqSet)
   is empty.

   set sortedList to the concatenation of
       left, comp and right
end while
```

7. Scheduling

In this section we describe the scheduling process. We want all communication occurring inside the simulation step to be one-way only, from processors with lower rank to processors with higher rank. To achieve this we make use of information stored with each task telling us from which equation it originates and thus which variable's evaluation it is part of. We do this by assigning the tasks to the processors in the order obtained after the sorting step described in section 6.

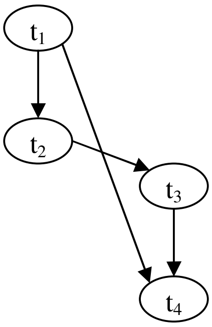
Tasks with variable number *0* through n_1 are scheduled to the first processor, n_1+1 through n_2 to the second and so on. The values of n_i are chosen so that they are always the variable number representing a state variable.

If we generate code for a single stage solver, e.g., Euler, this would be enough to ensure that backward communication only takes place between simulation steps, since the tasks are sorted to ensure no backward dependencies. This is not, however, the case when we generate code for multi-stage solvers. When sorting the equations in data-dependency order, variables considered known, like the state of the previous step are not considered, but in a later stage of the solver those values might have been calculated by an equation that comes later in the data-dependency sorting. This kind of dependency is represented by the dotted lines in figure 4. Luckily such references tend to have a short access distance as well and we solve this by adding a second step to the scheduling process.

For each processor *p* starting with the lowest ranked, find each task reachable from any leaf task scheduled to *p* by traversing the task graph with the edges reversed. Any task visited that was not already assigned to processor *p* is then moved to processor *p*. Tests show that the moved tasks do not influence the load balance of the schedule much.

When generating code for the individual processors there might be internal dependencies that dictate the order in which the tasks are laid out, which do not correspond to the order in which the results are needed by dependent tasks on other processors.

Assume t_1 and t_2 are assigned to processor p_1 and t_3 and t_4 are assigned to processor p_2. Furthermore, assume that there are dependencies between the tasks as shown in figure 5.



.

Figure 5. Data dependency graph.

In this case there is no sense in scheduling a send operation from t1 until t2 is also done, since no other processor can proceed with the result of t1 alone. Therefore all send operations are postponed until there is one that another processor actually might be waiting for. Then all queued up send operations are merged into a single message which reduces the communication overhead.

8. Measurements

In order to evaluate the gained speedup we have used a model of a flexible shaft using a one-dimensional discretization scheme. The shaft is modeled using a series of *n* rotational spring-damper components connected in a sequence. In order to make the simulation task computationally expensive enough, to make parallelization worth while, we use a non-linear spring-damper model.

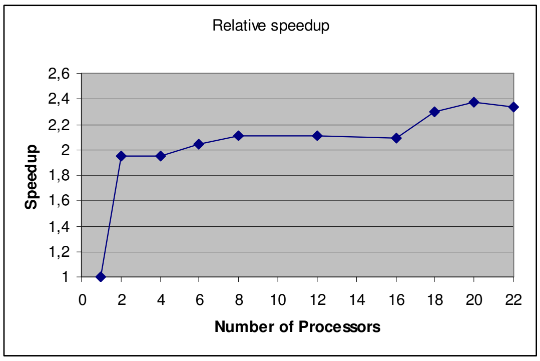


Figure 6. Relative speedup on PC-cluster

In these tests we use a shaft consisting of 100 spring-damper elements connected together. The same model has bee used when the task merging approach was evaluated in [1], which makes it possible to compare the results of this work to what was previously achieved.

The measurements were carried out on a 30-node PC cluster where each computation node is equipped with two 1.8 GHz AMD Athlon MP 2200+ and 2GB of RAM. Gigabit Ethernet is used for communication.

Figure 6 shows the results of the tests carried out so far. As can be seen the speedup for two processors is almost linear, but when the number of processors increase the speedup does not follow.

These preliminary results show very good speedup for two processors, but the increased speedup when using more processors is very modest. The maximum speedup, though, is slightly better compared to the task merging approach, at least for this test case.

We measured the time spent by the different processors sending messages as well as waiting for and receiving messages. And we can conclude that even though the amount of communication taking place between any two processors is largely invariant to the number of processors used, more time was spent in the send functions when the number of processors increased. This problem should be significantly reduced if the simulation were to run on a shared memory architecture using threads.

The second problem is that so far we have not considered how the tasks are scheduled within a processor. We are working on a scheduling algorithm to order the task internally in a way such that tasks depended upon by tasks on other processors are prioritized over other tasks.

It is also worth noting that for these kinds of models the communication need is invariant of the size of the problem. Doubling the number of spring dampers in the test model would not increase the need for communication, but the computation work would double, thus improving the computation to communication ratio.

9. Conclusion

To conclude we can se that for two processors the tests were very promising, but those promises were not fulfilled when the number of processors increased. If we compare to the previous results obtained with task merging in [1], though, we do not suffer from slowdown in the same way (see figure 3). Most likely this has to do with the fact that the communication cost for the master process running the solver increases linearly with the number of processors whereas in our new approach this communication is distributed more evenly among all processors.

10. Future work

In the nearest future we will profile the generated code to see were the bottlenecks are when ran on more than two processors and see if the scheduling algorithm can be tuned to avoid them. Also, tests must be carried out on different simulation problems to see if the results are general or if it differs much depending on the problem.

We also intend to port the runtime to run on threads in a shared memory setup. Since the trend is for CPU manufacturers to add more and more cores to the CPUs, it is becoming more and more relevant to explore parallelism in such an environment.

A runtime for the Cell BE processor is also planed. This processor has eight, so called, Synergistic Processing Elements (SPE) which do not actually share memory. Instead each SPE has its own local memory. Transfers to and from those local memories can be carried out using DMA without using any computation resources, so it should be possible to hide the communication latency during computation.

11. Acknowledgements

This work was supported by Vinnova in the Safe & Secure Modeling and Simulation project.

12. References

[1] Peter Aronsson. *Automatic Parallelization of Equation-Based Simulation Programs*. PhD thesis, Dissertation No. 1022, Dept. Computer and Information Science, Linköping University, Linköping, Sweden.

[2] Olaf Bonorden, Ben Juurlink, Ingo von Otte, and Ingo Rieping. The Paderborn University BSP (PUB) Library. *Parallel Computing*, 29:187–207, 2003.

[3] Peter Fritzson, Peter Aronsson, Håkan Lundvall, Kaj Nyström, Adrian Pop, Levon Saldamli, and David Broman. The OpenModelica Modeling, Simulation, and Software Development Environment. In *Simulation News Europe*,
44/45, December 2005. See also: http://www.ida.liu.se/projects/OpenModelica.

[4] Peter Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*, 940 pp., ISBN 0-471-471631, Wiley-IEEE Press, 2004. See also book web page:
www.mathcore.com/drModelica

[5] The Modelica Association. *The Modelica Language Specification Version 2.2*, March 2005. http://www.modelica.org.

[6] OpenMP Architecture Review Board. OpenMP: a Proposed Industry Standard API for Shared Memory Programming. White Paper, http://www.openmp.org/, October 1997.

[7] Joar Sohl. *A Scalable Run-time System for NestStep on Cluster Supercomputers*. Master thesis LITH-IDA-EX-06/011-SE, IDA, Linköpings universitet, 58183 Linköping, Sweden, March 2006.

[8] Kaj Nyström and Peter Fritzson. Parallel Simulation with Transmission Lines in Modelica. *In Proceedings of the 5th International Modelica Conference (Modelica'2006)*, Vienna, Austria, Sept. 4-5, 2006.

[9] Alexander Siemers, Dag Fritzson, and Peter Fritzson. Meta-Modeling for Multi-Physics Co-Simulations applied for OpenModelica. *In Proceedings of International Congress on Methodologies for Emerging Technologies in Automation (ANIPLA2006)*, Rome, Italy, November 13-15, 2006.

[10] Matthias Korch and Thomas Rauber. Optimizing Locality and Scalability of Embedded Runge-Kutta Solvers Using Block-Based Pipelining. *Journal of Parallel and Distributed Computing,* Volume 66 , Issue 3 (March 2006), Pages: 444 – 468.

[11] Peter Aronsson and Peter Fritzson. Automatic Parallelization in OpenModelica. In *Proceedings of 5th EUROSIM Congress on Modeling and Simulation*, Paris, France. ISBN (CD-ROM) 3-901608-28-1, Sept 2004.

[12]Christoph Kessler, Peter Fritzson and Mattias Eriksson. NestStepModelica: Mathematical Modeling and Bulk-Synchronous Parallel Simulation. *PARA-06 Workshop on state-of-the-art in scientific and parallel computing*, Umeå, Sweden, June 18-21, 2006.

[13]Thomas Rauber and Gudula Runger, Iterated Runge-Kutta Methods on Distributed Memory Multiprocessors. In *Proceedings of the 3rd Euromicro Workshop on Parallel and Distributed Processing*, pages 12--19, 1995

[14]Robert E. Tarjan, Depth First Search and Linear Graph Algorithms, *SIAM Journal of Computing*, 1, pages 146-160, 1972

How to communicate in a more efficient way over long distances instead of travelling

Erik Dahlquist
Malardalen University
erik.dahlquist@mdh.se

Abstract:

Today many long trips are made because of a need to meet physically. This is especially the case for business travels. If it would be possible to have a long distance contact where it felt like the counterpart was sitting next to you, the need for the actual travel would decrease significantly. This would save both time and energy. In this paper a discussion about technical possibilities to do this is discussed. A review is made over previous research related to the subject. Also ideas are presented on how this could be performed using a simulator model that drives a “mechanical, elastic head” from the information received on-line from a photo taken continuously by the counterpart.

Key words: long distance communication, remote, 3 D, animation, real time

Introduction:

The most energy consuming part of our lives may be energy for heating and cooling, but the fastest growing increase in energy usage today is for transportations. Many of these are concerning travels to see other people and different nice environments in other countries. These travels may be needed for sales purpose, or for other business communications. But many trips are for pure amusement. It is not enough to just communicate by telephone to replace a travel. To see a small “photo” of the one you talk to is better but not enough. What you want is to really feel like the person you talk to is sitting next to you, so that you can see all the movements in the eyes, the mouth, the hands and head. In this paper a possibility to use simulation models to move an artefact looking like your speaking partner as you speak to him or her.

Back ground:

There are some fantastic examples of what could be done on “remote communication”. In Hong-Kong there is a restaurant, where one wall is made out of glass, and “through” this glass wall you can communicate with persons sitting in a similar restaurant at Stanford University in San Francisco. It is almost like you were in the same room, at least if you use your imagination.

Many of us have been to IMAX theatres, and here we can see 3-D movies, that really make you feel like you were “in the middle of it”. The movie with the “crazy professor” is one example that many of us may have seen, where a hammer comes towards the audience, and everyone is moving towards the floor to really avoid it. This is a level of quality you would like to see. The question then is – how do we do it, and can we do it interactively?

A number of research groups have been working on “holography”, where photos are made out-of stereo photography. Examples of this can be seen in Leuski et al [2006] where an installation is made with a “hologram like” full scale human (“a robot”), that you communicate with. Here the focus is on making “the robot” communicating with you in an automatic way within a limited subject area, but still – it is a step towards “virtual humans”.

Zhang et al [1996] have studied “photomotion”. This means to produce 3-D pictures by varying from where the light comes, instead of having a “3-D camera”. By measuring the blackness strength in the same spot as the light source is moved, stereo effects can be achieved. This was further developed by e.g. Aaron Hertzman and Steven Seitz [2005]. They used a two light source system to produce the stereo photographs, but still moving the light sources.

For our purpose, to make a telephone call where you feel like talking to your friend in the same room, face animation is important. Most of the communication we perform except the direct words is produced by the face. Curio et al [2006] has developed a system for decomposition of facial motion to capture data using the Facial Action Coding System. The captured dynamic data are retargeted onto a 3-D face model. This gives movements that are very realistic. Subtle facial movements can be captured but still collecting sparse measurements. Vlasic et al [2005] has made an algorithm that handles especially the small facial movements. Blanz et al [1999] made a "morphable" model of facial movements in 3-D. Allen et al[2002] scanned bodies and made models of movements of the whole body from that. Kähler et al [2002] made anatomical head animations. Marschner et al [2000] made a procedure for how to make realistic facial animations, primarily for use in movies. Praun et al [2001] made a parameterization for facial animation.

Guenter et al [2004] made animation of facial movements by taking series of photos of persons who made different facial characters. They used six cameras. The characters were mapped with rubber markers, to get spots to link to. From this they could do general animations. Chang et al [2004] made facial recognition by used both visible and thermal images. Gotsman [2003] has presented a model of how to do spherically parameterized 3-D images.

Blanz [2006] showed how animation can be made of human faces in an automatic way from photos. Golovinskiy et al [2006] made a statistical modelling approach for detailed facial geometry. Sifakis [2006] simulated speech with a physical based facial muscle model.

Allen et al [2006] also have modelled in real time, which is of special interest with respect to our goal. Deng et al [2006] have done development for how to animate in interactive games, which is of the same complexity as what we want to do, although without a real human to interact with.

Givens et al [2003] and Min et al [2003] both have been working on face recognition. Givens has been using a PCA-method, Min a method with multiple gallery and probe images per person. In a survey made by Bowyer et al [2006] recognition is performed by matching models of the three dimensional shape of the face, either alone or in combination with matching corresponding two-dimensional intensity images.

A conclusion from all this is that there has been a lot of work on how to animate faces, production of 3-D images and similar, but most of it has been focused on putting this on a flat screen, and to use 3-D glasses to get the 3-D effect. Some of the methods have been focusing on getting data reductions, so that it could be possible to operate in real-time.

Still not much has been done on trying to project the figure on a screen that is shaped like a head, as we would like to do! I actually have not been able to find any reference on that.

Basic idea for the communication:

If we make a picture of a person in 3-D and analyse a number of key indices with respect to distances between eye centres, chin to nose to mouth, width of mouth etc, we can use this to produce a master from an elastic material formed like a head. This is attached to a body as well. By adapting the model description to a program with moveable arms inside the head, we can create a shape of the head that is similar to the actual persons head.

Now we use one camera or two coordinated cameras to film record the person you want to talk to. Also you are film recorded in the same way. The data representing changes in the physical geometry of the face are sent to the receiver. Now the camera image for both cameras are projected on the face of the "master", and coordinated to small movements of the arms to follow the movement of the lips (if these are not moved mechanically), the eyes and possibly some other parts of the face like eyebrows.

With respect to the arms and head the position of these can also be analysed and moved in the same way, to emphasise the body language.

The mouth is moving as a function of the words told, as the words are translated into physical movements. This is taught once for all for a certain language. This has already been shown by e.g. Sifakis et al [2006]. By doing this, the need to project the film exactly at the right position of the “face” may not be needed.

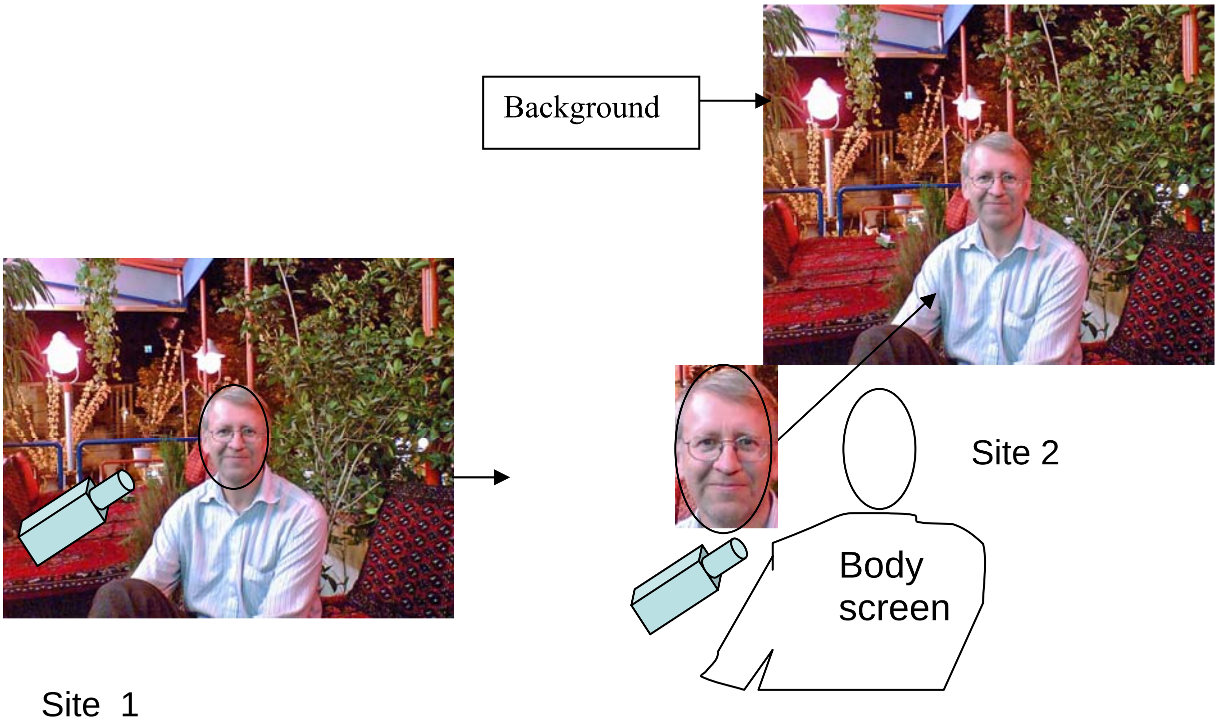


Figure 1. Photo is taken and projected on a body screen

A simulation model may be built in e.g. Modellica to give the set points and character of how to move the “arms” inside the “head” to position the tips of it at the right position. This includes using pulses and a number of servo motors. There is a known movement when moving the motors as a function of the number of pulses

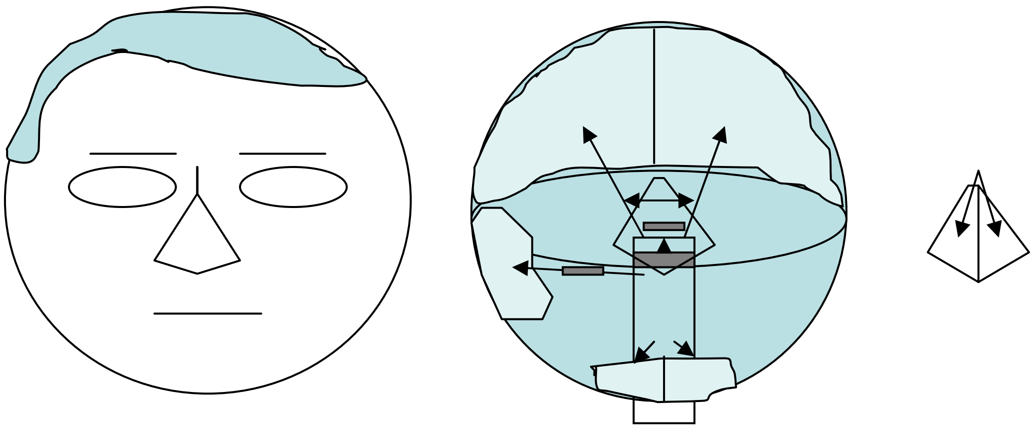


Figure 2. The elastic surface is formed by “muscles” made by servo motors to show facial movements

A couple of position meters are included to keep control of the distances, to avoid successive deviation of the arm positions.

How to use simulation models to produce set points for "mechanical muscles" for face and body:

The very quick movements are not adjusted inside the head, as it is enough to see the projected figures giving the movements. There is one positioning system inside the head, while there is another system for the movement of the actual head in different directions.

The simulation program is including the movements of the different actions – lips, eyes, head. The input is given from each photo and an image analysis that detects the differences from the first picture to the second, and calculates the step of movements. A filter function depresses some fast movements. The from the simulation model determined movements are transferred into actions of the servo motors. The different time constants of the different movements are adjusted by looking at the film and adjusting different constants in the equations.

The transfer of the signals can be performed with a good speed if broad band is used. By a suitable filter the signal noise can be reduced. The movements "inside the head" are a bit slower than the projected pictures at the outside. The movements of the arms are reduced by a factor compared to the "real", to avoid "jumpiness" in the movements.

The idea has been tested with a "pet animal" and servo-motors. The arm movements are principally similar to those performed by an industrial robot, where a model exist in Modellica (an ABB robot).

The head, or the "3-D screen", should have a "skin-type" surface and be elastic, so that the muscles can drag parts of the skin with internal "muscles" made with servo motors. It is spanned by plates attached at the top of the arms. The plates have skull/face-like characteristics. The skull can be widened as well as the chins… in relation to the wanted position. Adjustments are made when the deviation exceeds a certain level.

In the simplest version we have only one camera and one projector, and the 3-D effect is produced only by the "3-D screen". As the film is projected at a "face screen" we can get depth also without two cameras.

The CPU is centred in the body below the "head", and cables attached to the servo motors and the position meters. The camera(s) is (are) attached in a position fixed to the face. Quick movements of the head are filtered away. The focus of the cameras is projected towards a fixed position inside the nose.

The Modellica program is getting the input from the broad band data from the image pattern analysis of the face that should be modelled and performs a simulation for each time step, which gives input to the face "muscles" – the servo drives - to move. This will give a slight delay, but not more than can be acceptable.

For the spoken language a fixed algorithm may as an alternative be produced as an ANN, where each word is having a certain characteristic series of movements of the mouth. A pattern recognition algorithm is learning to identify a series of movements as one of a limited number of alternatives. This is then implemented as a series of fixed movements. The same goes for body language. If you move forward, this is identified and implemented "in the other end". In this way you can get an idea of the body language, although it may not be exactly the same.

Discussion and conclusions:

It would be interesting to combine a simulation model trained "by experience" and a physical model of "muscles" with moveable arms that are forced to move. Input to the model is data from images or photos sent to your recieving system after analysis in an image pattern system. This reduces the need for signals to be communicated to a reasonable level, for on-line applications. This can give a possibility to feel like communicating with someone almost "alive" close to you also at a far distance. This can give a possibility to replace travels, and thus save time and money, as well as energy.

We have not been implementing this yet, but have made a “facial robot” which is part of it. The intention is to proceed and build a “real set up” for testing the possibilities with the aim to have a fully interacting system sometime in the not too far off future.

References:

Allen B., Curless B., Popović.Z:, Articulated body deformation from range scan data, ACM Transactions on Graphics (TOG), v.21 n.3, July 2002

Allen B., Curless B., Popović.Z:, Hertzmann A.: Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis, Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, September 02-04, 2006, Vienna, Austria

Aleksey Golovinskiy , Wojciech Matusik , Hanspeter Pfister , Szymon Rusinkiewicz , Thomas Funkhouser, A statistical model for synthesis of detailed facial geometry, ACM Transactions on Graphics (TOG), v.25 n.3, July 2006

Bowyer KW., Chang K., Flynn P.: A survey of approaches and challenges in 3D and multi-modal 3D+ 2D face recognition - Computer Vision and Image Understanding, 2006 - Elsevier

Chang K., K. Bowyer, and P. Flynn, “Multi-Biometrics Using Facial Appearance, Shape, and Temperature,” Proc. Sixth IEEE Int'l Conf. Face and Gesture Recognition, pp. 43-48, 2004.

Craig Gotsman , Xianfeng Gu , Alla Sheffer, Fundamentals of spherical parameterization for 3D meshes, ACM Transactions on Graphics (TOG), v.22 n.3, July 2003

Curio C., Breidt M., Kleiner M., Vuong Q, Giese M., Bulthoff H.: Semantic 3D motion retargeting for facial animation. Proceedings of the 3rd symposium on Applied perception in graphics and visualization, July 28-29, 2006, Boston, Massachusetts

Deng Z. , Chiang P-Y. , Fox P.: Ulrich Neumann, Animating blendshape faces by cross-mapping motion capture data, Proceedings of the 2006 symposium on Interactive 3D graphics and games, March 14-17, 2006, Redwood City, California

Givens J. G. , R. Beveridge, B. Draper, and D. Bolme, “A Statistical Assessment of Subject Factors in the Pca Recognition of Human Faces,” Proc. Workshop Statistical Analysis in Computer Vision (CVPR), 2003., P. Flynn, and K. Bowyer, “Using Multiple Gallery and Probe Images Per Person to Improve Performance of Face Recognition,” Technical Report TR-03-7, Univ. of Notre Dame, 2003.

Givens G., R. Beveridge, B. Draper, and D. Bolme, “A Statistical Assessment of Subject Factors in the Pca Recognition of Human Faces,” Proc. Workshop Statistical Analysis in Computer Vision (CVPR), 2003.

Hertzmann A. and Seitz S.M.: Example-Based photometric stereo: Shape reconstruction with general varying BRDFs. IIIE Transactions on pattern analysis and machine intelligence, vol 27, no 8, pp 1254-1264 August 2005.

Kähler K., Haber J. , Yamauchi H., Hans-Peter Seidel, Head shop: generating animated head models with anatomical structure, Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, July 21-22, 2002, San Antonio, Texas

Leuski A.,Pair J., Traum D. ,McNerney P., Georgiou P., Patel R.; How to talk to a hologram. International Conference on Intelligent User Interfaces. Proceedings of the 11th international conference on Intelligent user interfaces Sydney, Australia , SESSION: Short papers
Pages: 360 - 362 , 2006

Praun E., Sweldens W., Schröder P.:, Consistent mesh parameterizations, Proceedings of the 28th annual conference on Computer graphics and interactive techniques, p.179-184, August 2001

Sifakis E. , Andrew Selle A., Robinson-MosherA. , Fedkiw R., Simulating speech with a physics-based facial muscle model, Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, September 02-04, 2006, Vienna, Austria

Stephen R. Marschner , Brian K. Guenter , Sashi Raghupathy, Modeling and Rendering for Realistic Facial Animation, Proceedings of the Eurographics Workshop on Rendering Techniques 2000, p.231-242, June 26-28, 2000

Vlasic D., Brand M., Pfister H., Popovic J.: Face transfer with multilinear models, International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2005 Papers,Los Angeles, California ,SESSION: Skin & faces ,Pages: 426 - 433 ,2005

Volker Blanz , Thomas Vetter, A morphable model for the synthesis of 3D faces, Proceedings of the 26th annual conference on Computer graphics and interactive techniques, p.187-194, July 1999

Volker Blanz, Computing human faces for human viewers: automated animation in photographs and paintings, Proceedings of the 8th international conference on Multimodal interfaces, November 02-04, 2006, Banff, Alberta, Canada

Zhang R., Tsal P-S., Shah M.: Photomotion, Computer vision and image understanding, Vol 63, No 2, March, pp 221-231, 1996.

Secure Distributed Co-Simulation over Wide Area Networks

Kristoffer Norling[1], David Broman[1], Peter Fritzson[1], Alexander Siemers[2], Dag Fritzson[2]

[1)] Department of Computer and
Information Science, Linköping University
SE-581 83 Linköping, Sweden
{x06krino,davbr,petfr}@ida.liu.se

[2)] SKF Engineering Research Centre
MDC, RKs-2, SE-415 50 Göteborg, Sweden
alexander.siemers@skf.com
dag.fritzson@skf.com

Abstract

Modeling and simulation often require different tools for specialized purposes, which increase the motivation to use co-simulation. Since physical models often are describing enterprises' primary know-how, there is a need for a sound approach to securely perform modeling and simulation. This paper discusses different possibilities from a security perspective, with focus on secure distributed co-simulation over wide area networks (WANs), using transmission line modeling (TLM). An approach is outlined and performance is evaluated both in a simulated WAN environment, and for a real encrypted co-simulation between Sweden and Australia. It is concluded that several parameters affect the total simulation time, where especially the network delay (latency) has a significant impact.

Keywords: Modeling, Co-Simulation, Transmission Line Modeling, Security, Data communication.

1. Introduction

The interest for modeling and simulation of complex physical systems, such as aircrafts and trains, has dramatically increased during the last decades. There exist both commercial modeling and simulation environments for specific domains, such as Adams [9] for mechanical systems, and specialized environments for certain application areas, e.g., SKF's BEAST [14,15] dedicated for bearing simulations. Moreover, new standardized acausal modeling languages for mixed-domains, e.g., Modelica [8] and VHDL-AMS [4], are gaining popularity in various applications, leading to a wide variety of models and specialized tool environments within an enterprise. These models and components are often dependent on each other and commonly need to be simulated together.

One technique that has proven to be both stable and efficient for simulating existing models is *co-simulation*, where sub-models are coupled together using *transmission line modeling (TLM)* [3,5,6,10]. The technique makes use of physically motivated time delays, enabling both numerically robust simulations and better performance using parallel processing.

Since modeling and simulation is becoming a common activity within enterprises, large amount of money and knowledge are invested into such models. Hence, models are becoming critical business assets describing primary know-how.

Within larger enterprises, different departments can be spread out over of the world, have special modeling competences, and model different parts of systems. Usually, there are defined confidentiality levels within an organization, requiring models to be protected against unauthorized disclosure. Furthermore, models need to be protected from modification by mistake and still being available for large scale reuse. Even if confidentiality issues with sub-contractors are today normally regulated by contracts, the need to protect and keep control of critical business assets is still vital.

Hence, in a co-simulation environment, it is important to have a sound approach for *secure modeling and simulation*, i.e., to create, modify, distribute, and simulate models in an acceptable manner according to the enterprise's security policy.

1.1 Approaches to Secure Modeling and Simulation

Information security can be divided into three aspects[1]:

- *Confidentiality*: protection against unauthorized disclosure of information.
- *Integrity*: protection against unauthorized creation, modification, or deletion of information.
- *Availability*: the assurance that authorized entities have access to correct information when needed.

[1] Even if most practitioners in the field of information security recognize these three aspects as fundamental, there is no exact border between e.g., robustness and availability. Here we treat robustness of the system as being part of the availability of simulation and model information.

The importance of these three areas for secure modeling and simulation will be outlined in the following section.

The obvious first phase in modeling and co-simulation is the modeling phase, which can be divided into three steps, each requiring different expertise [10,12]:

- Modeling of sub-models in specialized environments, e.g., Adams and BEAST models.
- Encapsulate sub-models and define interfaces.
- Design a *meta-model*, where the different sub-models are integrated and connected to each other.

It is of great importance to perform these steps in a secure manner. However, in this paper we have chosen to focus on how the models are *securely distributed and simulated*, leaving secure modeling as future research.

We have identified two general possible approaches for secure distribution and simulation:

- *Centralized simulation with secure model exchange*. The models are centralized and simulated at one location. The models must be encrypted to avoid unauthorized disclosure of the model content.
- *Secure distributed co-simulation.* The models are kept within their departments/companies and simulated locally. Simulation data is exchanged between the locations during simulation, making use of co-simulation technologies.

Both these approaches raise several questions regarding security aspects of confidentiality, integrity and availability, as well as practical simulation feasibility.

The former centralized approach uses traditional simulation techniques, where all models are locally available. To provide confidentiality and avoid disclosure of model content, the models need to be encrypted. This sounds initially as a sound approach, but even if the model was encrypted when sent to the other locations, it still needs to be decrypted at the centralized location. This means that the simulation environment needs to know the encryption key. Hence, the information is only *practically confidential*, i.e., it might be hard to get the model in clear text, but not theoretically impossible[2].

A second alternative to model encryption can be *model obfuscating*, i.e., to rearrange the models structure making it unreadable, without changing the model's behavior. Another variant is to distribute the models in another format, e.g., as binary executables.

A certain level of integrity protection, e.g., to achieve need-to-know access (least privilege), can be used by applying message authentication codes (MAC) on the models. However, the same problematic issues regarding keys are unfortunately applicable here as well. Moreover, the centralized approach cannot control the number of times a model is used or that it is not illegally redistributed.

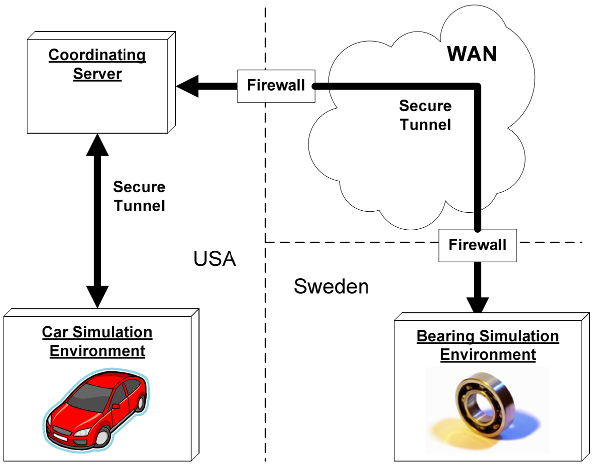


Figure 1. A scenario of secure distributed co-simulation distributed over long distances.

Figure 1 outlines a scenario of the second approach of distributed co-simulation, where a car model is simulated at a location in the USA and the bearing model is simulated in Sweden. To make this scenario possible, a co-simulation technique such as the one based on Transmission Line Modeling (TLM) can be used. Models are not sent between the different locations. Instead data describing states of the models' interfaces are exchanged between the locations during simulation, using a wide area network (WAN), e.g., the Internet. This scenario leads to several security observations:

- *Confidentiality*: The models are never exchanged between the parties and are therefore never exposed to the other party, i.e., confidentiality is kept using a black-box view of the model. If the TLM data sent between the simulation nodes is encrypted, unauthorized disclosure of the traffic is avoided from third parties. Furthermore, the permission to reuse models for simulation can be controlled, since the models never leave the original location.
- *Integrity:* Unauthorized modification or changes by mistake of models are avoided due to the fact that models are never sent between the locations. Note however that the meta-model needs to be defined and maintained at one location. Integrity protection of simulation traffic can be protected using standard MAC technique.

[2] In cryptography, it is never impossible to decrypt data without a key, but it can be *computationally very hard*. E.g., using a brute-force approach, it might take hundreds of years to decrypt a specific model.

- *Availability.* Co-simulation using TLM has been shown to provide numerically robust results. However, if the data communication link is not robust or is too slow, and data is not delivered in time, the simulation can be slowed down or terminated. Hence, performance, robustness, and total simulation time are critical factors concerning distributed co-simulation.

Both approaches have pros and cons. A centralized approach raises several problematic issues regarding true confidentiality and integrity of models. On the other hand, centralization can guarantee better communication networks and may therefore result in better performance.

The distributed approach is appealing due to the fact that the black-box characteristics give both better integrity and confidentiality properties of models. In this approach, computational resources from different locations can also be shared and used in a distributed manner. However, the extra overhead needed for protection of data sent during simulation and the cost of transmitting data traffic over large distances can naturally affect performance of the total simulation time.

1.2 Challenges and Contributions

We have chosen to focus on the second approach of *secure distributed co-simulation*, due to its promising properties of confidentiality and integrity. Hence, this paper will primarily deal with aspects of performance and robustness, with the following questions:

- Is it even practically possible to co-simulate stiff-bearing models in Sweden together with other mechanical models, which are simulated as far away as the USA or Australia?
- If it is possible, how much longer time does it take to simulate?
- Which parameters and factors affect the total simulation time? Which dependencies exist between these parameters? Are they linear or are there certain breakpoints?

The main contribution of this work is the described approach of secure distributed co-simulation together with conclusion draw from analysis of an experimental simulation test.

1.3 Paper Outline

The paper is structured as follows. Section 2 introduces the fundamental theory of Transmission Line Modeling (TLM) for co-simulation and the basic concepts of data communication. Parameters and factors that may affect the total simulation time are discussed. Section 3 outlines the experimental setup used for generating simulation data from a model reflecting the scenario described in the introduction. The physical model, simulation framework, deployment structure, and simulation tools are described. Section 4 presents the results from the experiment followed by analysis of how different parameters affect the duration of the total simulation time. Finally, section 5 states conclusions of the work.

2. Parameters Affecting the Total Simulation Time

In a distributed co-simulation environment, several parameters and factors can affect the total simulation time. First, we will consider the equations stating transmission line modeling. Second, different factors of the data communication link will be discussed.

2.1 Transmission Line Modeling

The theory of transmission line modeling (TLM) has evolved from the telegraph equations, which concern transmission of electrical signals over long wires.

The TLM method can also be used in other domains, such as mechanical systems where force and velocity are affecting a system. The main motivation to use TLM in such a system is that it describes physically relevant time delays. These delays allow different parts of the system, which are separated by TLM interfaces, to be simulated independently of each other's time steps. Hence, this technique enables the subsystems to be simulated in parallel. Moreover, since exchange of data is needed only between well-defined interfaces, it is also possible to co-simulate between different simulation environments and tools. The latter fact is used in SKF's co-simulation framework [10,12,13].

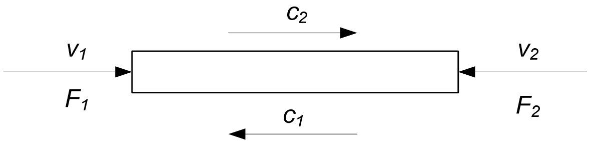


Figure 2. Delay line with wave variables c_1, c_2, velocity v_1, v_2, and reaction forces F_1, F_2.

Figure 2 outlines the main variables involved in a transmission line in a mechanical context. The velocity and reaction forces on each side of the line affect the wave variables c_1 and c_2.

The equations involved in the TLM connection are as follows [5,10]:

$$c_1(t) = F_2(t - T_{TLM}) + Z_c v_2(t - T_{TLM})$$
$$c_2(t) = F_1(t - T_{TLM}) + Z_c v_1(t - T_{TLM})$$
$$F_1(t) = Z_c v_1(t) + c_1(t)$$
$$F_2(t) = Z_c v_2(t) + c_2(t)$$

Here c_1 and c_2 denote the force waves, F_1 and F_2 reaction forces and v_1 and v_2 velocities. There are two parameters specified in the model: T_{TLM} indicating the delay time through the line and Z_c that is the characteristic impedance. These parameters must be assigned values, which are within physically acceptable boundaries. See [5,6,10] for a more detailed discussions about the physical aspects of the model.

From the equations above, we can clearly see that only old values of F_2 is needed to calculate F_1, and only old values of F_1 is needed to calculate F_2. The allowed length of this delay depends on the parameter T_{TLM}. Larger values of T_{TLM}, will allow the simulations to take larger time steps, which may result in shorter simulation time. Hence, T_{TLM} is an important parameter affecting the total simulation time.

2.2 Data communication

During the co-simulation a continuous flow of data packages containing TLM information are transmitted between the computation nodes. Before the data is transmitted over the WAN, it is being encrypted and message authentication codes (MAC) are created for integrity checks. When the data packet is received at the other node, it is being decrypted and forwarded to the simulation software.

There are two fundamental measurable characteristics of network performance [11].

- *Bandwidth* - the number of bits that can be transmitted over the network during a specific period of time. Normal measurement is in Bits/s.
- *Delay (or latency)* - the time period it takes for a very small message to be sent over the network. Delay is measured strictly in time, e.g., number of milliseconds.

A network always has these two characteristics, even though they can change over time, depending on routing and traffic load. This variation can lead to another important phenomenon called *jitter*, which is the variation of the delay for different network packets in a data stream. In the rest of the paper B_{WAN} denotes the bandwidth for the simulation environment and T_{WAN} the delay between the nodes.

Note that in this paper we will discuss two different time scales. The first one is the simulated time corresponding to the modeled physical process. Integration time steps, time stamps sent within the TLM framework, and the T_{TLM} parameter correspond to this time scale. The second time scale belongs to the real-time of the computation of the simulation result. The total simulation time and delays in the network, i.e., T_{WAN} belong to this scale.

When measuring delays in a system, the time is often referred to as the round-trip-time (RTT), which is the time it takes for the packet to travel to the destination and back. The delay of the system can be approximated to *RTT/2*.

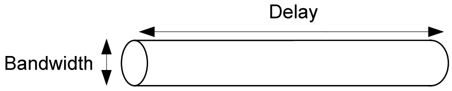


Figure 3. The delay × bandwidth product *P*.

Another central concept is *the delay × bandwidth product*, depicted in Figure 3. The formula is as follows:

$$P = B_{WAN} * T_{WAN}$$

The bandwidth can be seen as the diameter of a pipe, and the delay as the length of the pipe. Hence, the *P* represents the maximal natural queue of data located in the network.

Besides the TLM equations and the delay / bandwidth discussion above, there are of course several other factors that can affect the outcome. One such important factor is the load and balanced computation power among the nodes. Another factor is the integration step size / tolerance level used on the simulation nodes. Other potential requirements, such as encryption protocols and algorithms used, can both affect T_{WAN} and B_{WAN}. If the bandwidth is critical, lossless data compression techniques can be used to increase the bandwidth [1]. However, note that delay cannot be improved by data compression.

3. Experimental Setup

This section provides a detailed description of the environments surrounding the experiment, the simulation framework, and software used in the experiment.

3.1 Meta-Models and Components

A *component* is a model instance created in a specialized modeling and simulation tool (e.g., Adams or BEAST). A *meta-model* defines the interconnections between two or more components.

The objective of the experiment is to measure how secure communication over large distances will affect the meta-model simulation in terms of total simulation time. To model the scenario described in Figure 1, a highly simplified model is used instead of a real car model. We are interested in how the data communication factors, T_{WAN} (the delay) and B_{WAN} (the bandwidth), together with the TLM delay T_{TLM} affect the simulation time.

The meta-model used in the experiment is a double pendulum. As shown in Figure 4, the pendulum is constructed from three different components, two shafts, which both have TLM connection to a bearing.

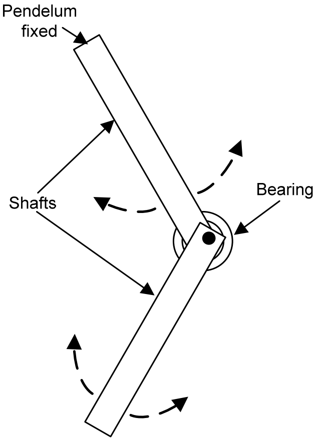


Figure 4. The bearing-shaft model to be simulated.

In terms of simulation time and resources, the two shaft components represent the less demanding components in our meta-model.

3.2 Simulation Framework

To conduct the experiment, we have used a centralized TLM co-simulation application referred to as the TLM manager. Figure 5 presents an overview of the TLM co-simulation framework.

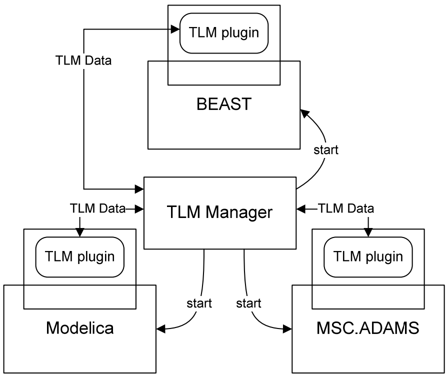


Figure 5. SKF's TLM system framework.

The TLM manager reads a meta-model specification defining the simulation components and the TLM interfaces in the meta-model.

The simulation components are simulated in a specialized simulation environment, such as Adams, BEAST, or Modelica. In order to make the framework functional, the specialized simulation environments need to be incorporated into the framework. The TLM manager and the specialized environment communicate through TLM plug-in's, as depicted in Figure 5.

During simulation, the simulated components interact by sending time-stamped data and delayed position and orientation data. The TLM data is transmitted to relevant node via the TLM manager. For a more detailed description of the framework, see [10].

3.3 Deployment Structure

The static deployment structure is depicted in Figure 6.

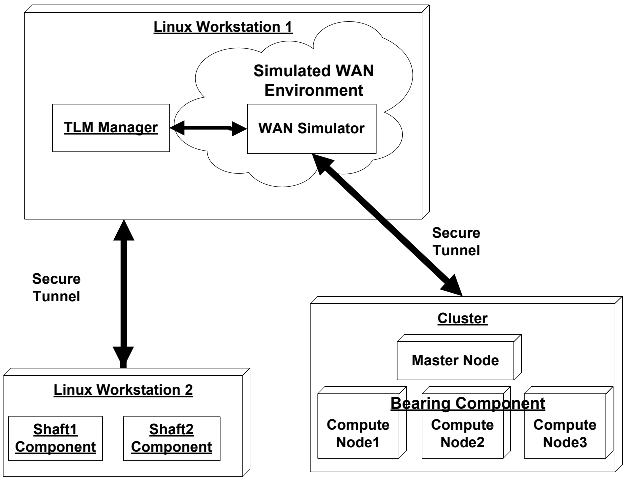


Figure 6. The deployment structure of the static structure of the system.

Two Linux workstations and a computer cluster are used for simulation of the components. The simulation is started from the workstation labeled Linux Workstation 1; this is also where the TLM-Manager and WAN simulator are instantiated.

In order to evaluate what impact delay and bandwidth in the WAN environment have on the simulation it is desirable to have full control of these parameters. This control is obtained through the WAN simulator. The WAN simulator is a software service that intercepts all data sent between the cluster and Linux Workstation 1. It will then apply the desired data communication factors and pass on the data to the correct destination. A more detailed overview of the WAN simulator will be given in the end of this section.

The shaft components are simulated using BEAST. These simulations are run on the workstation labeled Linux Workstation 2. The bearing component is simulated on the cluster using BEAST as well. The two tunnels are ensuring secure communication between the TLM manager and Linux Workstation 2 and between the TLM manager and the cluster. In the experimental setup SSH version 2 (SSHv2)[7] is used to create the secure tunnels. Another strategy would be to incorporate the security layer within the application. With such an approach TLS [2] may be a sound alterative.

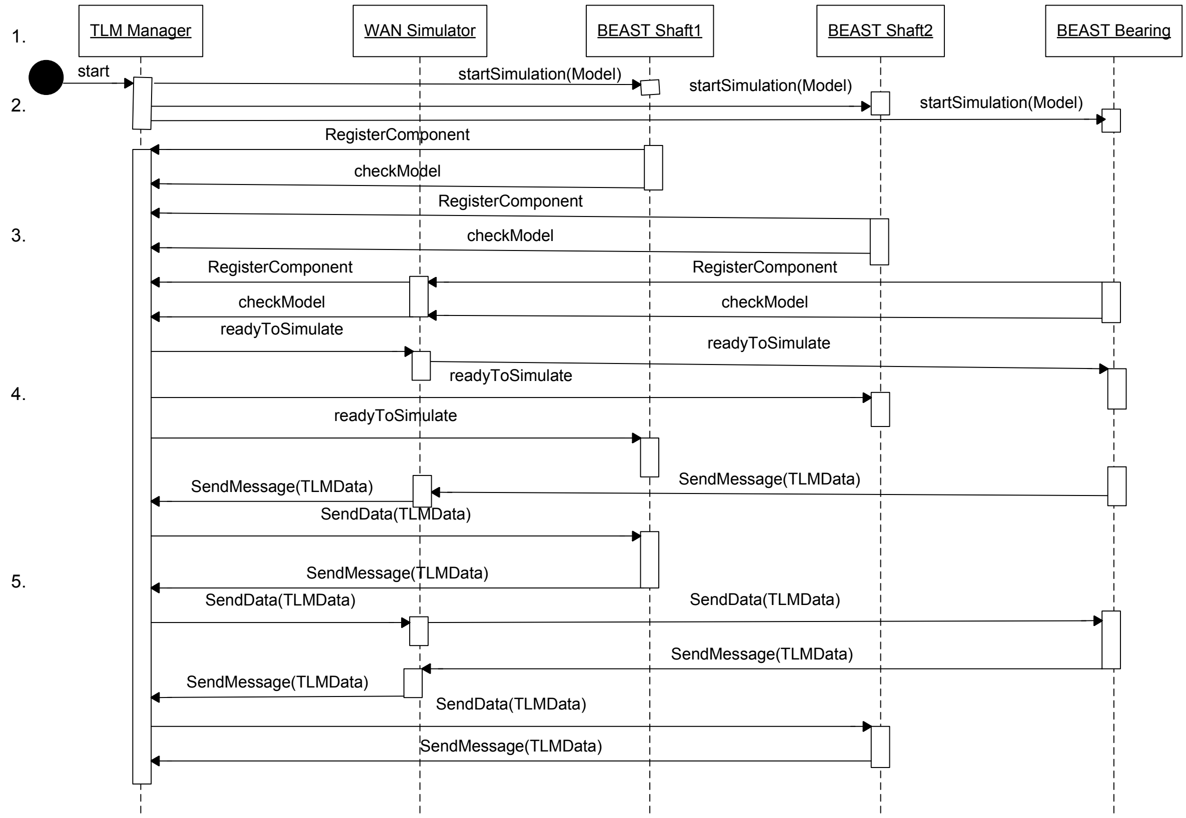


Figure 7. Sequence diagram showing basic information flow in the co-simulation environment.

3.4 Dynamic System Behavior

This section describes the interaction between the simulation components and the TLM manager. The sequence diagram in Figure 7 can be described as follows:

1. There are five different objects involved in the interaction during the simulation. To the right, there are the three simulation components, *BEAST Shaft1*, *BEAST Shaft2,* and *BEAST bearing*. The *TLM manager* is handling the data exchange between the components. Finally, the *WAN simulator* is intercepting and forwarding all calls between the bearing simulation component and the TLM manager.

2. The meta-model simulation is initiated by a script. The script will start the TLM manager on the local machine. It will also start the specialized simulation environments of all the simulation components.

3. Each simulation component has to register itself to the TLM manager. The component will also register all of its TLM interfaces. Once this has been accepted by the manager the simulation component will send a check model request to the manager. This is a request asking if the entire meta-model is ready to be simulated.

4. Once all simulation components and TLM interfaces are accounted for, the manager will reply to the simulation components that the meta-model is ready to be simulated. At this point the actual simulation will begin.

5. The simulation components will regularly send data for its TLM interfaces to the manager. The manager will then pass the data to the appropriate destinations. At the bottom of the diagram we can see how the bearing component through the TLM manager sends some TLM data directed to the shaft1 component. The WAN simulator intercepts this data, holds it for a while, and then forwards it to the TLM manager. The manager forwards the data to the shaft1 component that after some simulation time sends new data to the manager. This communication is not intercepted by the WAN simulator since the data is not passing through our simulated WAN environment. The manager will receive the new data from shaft1 and forward it to the bearing component. This message, from the manager to the bearing, will once again be intercepted by the WAN simulator, and then forwarded to the bearing component. This type of data exchange will continue until the simulation has finished.

For a more detailed specification of the interaction in the communication protocol, see [10].

3.5 WAN Simulator

The WAN simulator is a specific application designed for this experiment with the purpose of controlling the data communication (T_{WAN} and B_{WAN}) between the bearing simulation component and the TLM manager. The simulator captures all data sent between the TLM manager and the bearing component. The simulator consists of two queues, as depicted in Figure 8. One queue holds the data sent from the simulation component to the manager and one queue holds the data sent from the manager to the simulation component. The WAN simulator reads and writes data in the queues concurrently.

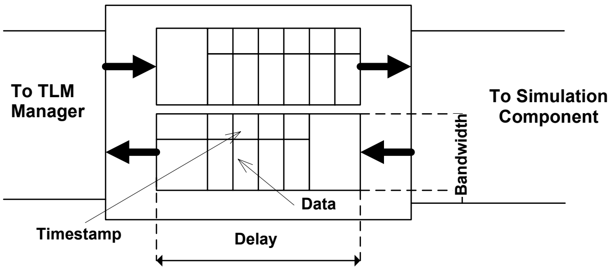


Figure 8. Conceptual outline of the WAN simulator.

To control the delay, the WAN simulator will, when capturing a data message, put a timestamp on it, before it is added to the queue. This data message will not be released from the queue until it has been held for the entire duration of the delay time.

For example, the simulation component sends some data to the TLM manager. The WAN simulator captures it and stamps it with the time of the capture. The message together with the timestamp is added to the last position in the queue of data waiting to be forwarded to the TLM manager. When the data reaches the first position in the queue it is next in line to be forwarded to the manager. The WAN simulator will then perform a check of the timestamp on the data. If the condition

$$T_{WAN} \leq (currentTime - timestamp)$$

is true, the data will be released from the queue and forwarded to the TLM manager. Else, the data will be held until enough time has passed to fulfill the condition. The second queue is working in the same way, but in the opposite direction.

Bandwidth may be controlled with the aid of the delay × bandwidth product, *P*, described in Section 2. The queues may hold up to *P* bits of data at any given time without risk of exceeding the desired bandwidth restrictions in respective communication direction. When a queue is filled, further data that wish to pass the WAN simulator will be delayed until data has been released from the queue and space has been made available.

4. Experiment Results and Analysis

In the following section we will present the experimental results produced by performing several test simulations in the experimental setup. The result is followed up by discussions and analysis of how different parameters affect the overall simulation time.

4.1 Experiment Results

The experiment was divided into two parts, where the first part used the experimental setup with WAN simulator according to Section 3. In the second part, co-simulation was performed over the Internet between Sweden and Australia.

Part 1 of the experiment was implemented by performing four different test sequences, where the data communication delay T_{WAN} was varied for each sequence. Figure 9 shows the four sequences for different values of T_{TLM}.

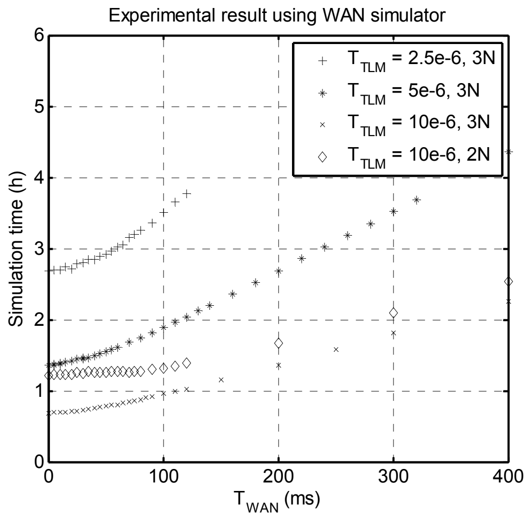


Figure 9. Plot of total simulation time in relation to the network delay T_{WAN}.

In the first three sequences, 3 computation nodes in the cluster (see Figure 6) were used, while the fourth sequence used only 2 nodes (denoted 3N and 2N in the figure). In addition to the values shown in this diagram, simulations were performed for T_{WAN} = 500 ms and T_{WAN} = 1000 ms for some of the sequences. These values are not shown in the diagram due to illustration reasons, but will be used in the analysis in the next section.

Part 2 of the experiment was performed without the WAN simulator, i.e., the same setup as depicted in Figure 6 was used, except that the TLM manager communicates directly to the cluster via an encrypted tunnel. Three computation nodes were used in the cluster for these experiments. The results are given in Table 1.

Type of simulation	T_{TLM} = 2.5e-6	T_{TLM} = 5e-6	T_{TLM} = 10e-6
LAN, No Encryption	120 min	62 min	31 min
LAN, Encryption	160 min	82 min	41 min
Increased time due to encryption	33 %	32 %	32 %
Round-trip simulation to University of Queensland, Australia	495 min	277 min	133 min
Increased time due to delays to Australia (round-trip)	312 %	344 %	332 %

Table 1. Simulation time when executed at the local area network (LAN) in Sweden, both with and without SSH tunnel encryption enabled. Round-trip simulation to and from Australia, including SSH encryption.

The table shows the increased time needed in percent using an encryption tunnel (SSH) compared to performing the simulation without it.

When executing the simulation via Australia, both the shafts and the bearing were simulated in Sweden, but the traffic between the cluster and the TLM manager was transmitted via a tunnel from Sweden to Australia and then back again. Hence, the experiment is actually performed for a distance twice as long as Australia-Sweden. Note also that the simulation time for just one way cannot be calculated simply by dividing in half.

While performing the simulations, the data throughput in the WAN simulator was being monitored and measured. Since the WAN simulator captures all data between the TLM manager and the bearing simulation component, it was measured that the data throughput did not normally exceed 35 Kbit/s during the TLM data exchange. However, when the simulations were initiated, a small peak in throughput of about 140 Kbit/s was noticed.

In the simulations performed in this experiment, no major robustness problems were discovered and all started simulation jobs were finished without unexpected termination.

All in all, the simulation experiment results presented in this section required approximately 250 hours of simulation time.

4.2 Discussion and Analysis

The main purpose of this work is to investigate if it is possible to co-simulate over long distances and which parameters that affect the total simulation time. In Table 1 it was shown that the simulation was indeed possible to perform, even for very long distances. We will now analyze and discuss the affected parameters and then finally compare the experimental data obtained using the WAN simulator with the simulation times resulting from the simulation via Australia.

Recall the diagram given in Figure 9 in the previous section. For small values of T_{WAN}, a non-linear behavior can be seen, while for larger values the curves have a more linear characteristic.

Consider Figure 10, where the values are plotted for T_{TLM} = 5e-6 and T_{TLM} = 10e-6, where $T_{WAN} \leq 250$ ms .

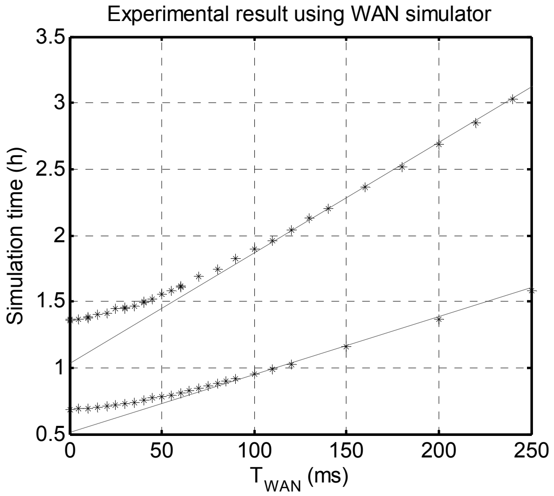


Figure 10. Simulation time with linear regression estimations. T_{TLM} = 5e-6 (the upper curve). T_{TLM} = 10e-6 (the lower curve). Both sequences are using 3 cluster nodes.

Linear regression using the least square method is used to adjust a line for $T_{WAN} \geq$ 80ms. Both measured values and the adjusted curves are plotted in the diagram. Note that all measured values above 80ms are used for finding the curve, including measurements 500 ms and 1000 ms.

In both curves, we can imagine a smooth breakpoint somewhere around 75ms, where the derivative becomes fixed. What can cause this breakpoint? One reasonable cause is that to the right of the breakpoint, one of the computation nodes is idle for some time while waiting on TLM data input, i.e., the delay causes the bottleneck of the system to be switched from one of the computation nodes to the delay value in the communication link. For this reason, we call this breakpoint the bottleneck breakpoint, denoted T_{BBP}. Hence, from the data shown so far, we clearly see that T_{WAN}, affects the simulation time significantly, especially above $T_{BBP.}$ The approximated formulas for all experiment sequences are given in Figure 11:

(1)	$f_{2.5e-6,3N}(t) = 7964{,}4 + 46{,}8t$
(2)	$f_{5e-6,3N}(t) = 3735{,}5 + 30{,}0t$
(3)	$f_{10e-6,3N}(t) = 1845{,}1 + 15{,}7t$
(4)	$f_{10e-6,2N}(t) = 2911{,}0 + 15{,}9t$

Figure 11. Linear approximations after using the least square method on measured simulation data. *f(t)* states simulation time in seconds and *t* represents T_{WAN} in ms.

All formulas in Figure 11 were approximated for T_{WAN} larger or equal to 80ms, except for the last case where $T_{WAN} \geq 110$ms, since the linear characteristic started at larger values. One interesting observation is that changing the computational power for the bearing calculation changes the total simulation time, but not the derivative for the linear part of the curve (15,7 ≈ 15,9).

Another important parameter in data communication is the bandwidth in the communication link. Does the limitations of the bandwidth affect the total simulation time? The WAN simulator tool has the capability to simulate a smaller bandwidth. However, for this model, we noted that only a small bandwidth of about 35Kbit/s was needed. Since leased lines normally have the capacity of several MBit/s, this throughput requirement can be negligible. However, for larger models with many TLM-interfaces, the bandwidth factor may be important. To draw any further conclusions about how larger models affect the bandwidth, more experiments must be conducted, which is suggested as future research.

There are of course several other factors that may affect the total simulation time, which we have not measured in this experiment. One such parameter is the time step or tolerance level of the numerical integration routine for the simulation.

In the experiments, SSH was used to protect the TLM data between the nodes. The encryption and hash computations do indeed affect the delay in the system. As we can see in Table 1, the effect is not negligible (around 32% increased simulation time). We have not made any comparisons in this experiment what the effect would be when applying different encryption algorithms, even if there could be expected to be differences. Note however that if the co-simulation is performed on a trusted WAN, e.g., leased private lines, it might be acceptable to disable data encryption to improve performance.

To be able to compare the simulated delays of T_{WAN} with delays in real world networks, a number of measurements were performed over the Internet using the standard `ping` application. The tests were performed both using a connection at Linköping University (LIU), and a best effort connection from Stockholm, Sweden. Since we know that the simulation time grows linear to T_{WAN}, we can approximate the simulation time for different cities. The measured delays and estimated simulation times are presented in Table 2.

For values below $T_{BBP} \approx 75$ms, estimations are taken from measured data and not the linear approximation. However, can we really expect that these approximations correspond to real distributed simulation? Since the experiments with the WAN simulator were made in an idealized environment, it can be expected that these results are optimistic.

University, Location	T_{WAN} (ms) Lkp	T_{WAN} (ms) Stock holm	Simulation time (min)
University of Trier, Germany	21	20	85
University of Cambridge, UK	19	19	85
MIT, Massachusetts, USA	57	56	96
Stanford, California, USA	96	87	108
University of Tokyo, Japan	142	143	134
University of Queensland, Australia	170	178	149

Table 2. Measured communication delays between Sweden and different locations all over the world. The optimistic approximated simulation times, using formula (2) in Figure 11, are given for T_{TLM} = 5e-6.

In the second part of the experiment, as given in Table 1, a real simulation via Australia was performed. A comparison between measured and approximated results is depicted in Table 3.

	T_{WAN}	T_{TLM} = 2.5e-6	T_{TLM} = 5e-6	T_{TLM} = 10e-6
Ping RTT	348 ms	*404 min*	*236 min*	*122 min*
Max, RTT, SSH	470 ms	*499 min*	*297 min*	*154 min*
Min, RTT, SSH	340 ms	*398 min*	*232 min*	*120 min*
Mean, RTT, SSH	405 ms	*473 min*	*265 min*	*137 min*
Measured, Australia	-	495 min	277 min	133 min
Difference Measured to Ping RTT	-	-18,3 %	-14,7 %	-8,4%
Difference Measured to Mean, RTT, SSH	-	-4,5%	-4,4 %	2,8%

Table 3. A comparison between approximated simulation time and real measured simulation time.

Note that all times are round-trip-times and not one-way delays. Simulation times given in italic indicate an approximated value, using (1)-(4) in Figure 11. Row 5 corresponds to the experiment in Table 1.

It was discovered that the actual delay during the simulation varied from 340 ms to 470 ms (row 2 and 3), indicating a certain jitter in the communication[3].

In the table we can see that the ping RTT gives an optimistic value, resulting in an 18% to 8% too short simulation time. However, if we calculate the mean (row 4), a closer approximation is achieved.

Using formula (2) in Figure 11, together with the measured simulation time of the round-trip simulation in Table 1, we can compute the total simulation time it would take to simulate the shafts in Australia and the bearing in Sweden, i.e., use the delay

[3] Note that these delays show the RTT for the TCP packets of the SSH tunnel, which does not include the delay of the actual encryption/decryption.

instead of the RTT. For T_{WAN} = 5e-6, an approximated simulation time is ≈ 170 min; a 174% increase in time compared to an unencrypted local simulation that took 62 minutes.

5. Conclusions

We have in this paper discussed different aspects of *secure modeling and simulation*. Focus was put on secure distribution and simulation and both a centralized and a decentralized approach were discussed and compared from a security perspective. The decentralized approach was argued to have benefits regarding confidentiality and integrity, while giving open questions about performance and robustness.

An experiment was performed where a simple double pendulum with a bearing was co-simulated between Sweden and Australia. No robustness issue was discovered during simulation, and the simulation time was increased from approximately 62 min to 170 min (≈ 174% time increase), compared to a local simulation without encryption of data traffic.

To investigate which parameters that affect the total simulation-time, four series of tests were performed using a simulated environment of the wide area network (WAN). The main findings can be summarized as follows:

- The *network bandwidth* between simulation nodes has little effect, since throughput of TLM data is small.
- The *network delay (latency)* between simulation nodes has great impact on the total simulation time. The growth is linear after a certain breakpoint. Consideration must be taken to *jitter* (delay variations over time).
- The *TLM delay* of the model affects the simulation time significantly. Larger delay gives shorter simulation time, since the solver is allowed to take longer time steps.

Finally, we would like to conclude that secure co-simulation over long distances seems to be both a practical and possible solution for secure distribution and simulation of models within, and potentially between, enterprises.

6. Acknowledgements

This research was funded by CUGS (National Graduate School in Computer Science), by SKF, and by Vinnova under the NETPROG Safe and Secure Modeling and Simulation on the GRID project.

References

[1] David Broman. (2001). "Lossless Data Compression - Methods for Achieving Better Performance in a Wireless VPN". Master thesis. LiTH-ISY-EX-3159. Linköping University. Sweden.

[2] T. Dierks and E. Rescorla. (2006). "The Transport Layer Security (TLS) Protocol. Version 1.1". RFC 4346.

[3] Dag Fritzson, Jonas Ståhl, and Iakov Nakhimovski. (2007). "Transmission line co-simulation of rolling bearing applications". In Proceedings of the 48th Conference on Simulation and Modelling (SIMS'07). Göteborg. Sweden. Linköping University Electronic Press.

[4] IEEE 1706.1 Working Group. (1999). "IEEE Std 1076.1-1999, IEEE Standard VHDL Analog and Mixed-Signal Extensions". IEEE Press. USA.

[5] Petter Krus. (1999). "Modelling of Mechanical Systems Using Rigid Bodies and Transmission Line Joints". Transactions of ASME. Journal of Dynamic Systems Measurement and Control. December 1999.

[6] Petter Krus, Arne Jansson, Jan-Ove Palmberg, Kenneth Weddfelt. (1990). "Distributed Simulation of Hydromechanical Systems". Presented at Third Bath International Fluid Power Workshop, Bath, UK 1990.

[7] S. Lehtinen and C. Lonvick. Ed., (2006). "The Secure Shell (SSH) Protocol Assigned Numbers". RFC 4250.

[8] Modelica Association. (2007). "Modelica - A Unified Object-Oriented Language for Physical Systems Modeling - Language Specification Version 3.0". Available from: www.modelica.org.

[9] MSC Software. (2007). "MD Adams - overview". http://www.mscsoftware.com/.Last Access:Oct 3, 2007

[10] Iakov Nakhimovski. (2006). "Contribution to the Modeling and Simulation of Mechanical Systems with Detailed Contact Analysis". PhD Thesis. Linköping University. Sweden.

[11] Larry L. Peterson and Bruce S. Davie. (2000). "Computer Networks - A Systems Approach". Sec. Ed. ISBN 1558605770. Morgan Kaufmann Publ. USA.

[12] Alexander Siemers and Dag Fritzson. (2007). "A Meta-Modeling Environment for Mechanical System Co-Simulation". In Proceedings of the 48th Conference on Simulation and Modelling (SIMS'07). Göteborg. Sweden. Linköping University Electronic Press.

[13] Alexander Siemers, Iakov Nakhimovski, and Dag Fritzson. (2005). "Meta-modelling of Mechanical Systems with Transmission Line Joints in Modelica". In Proceedings of the Fourth International Modelica Conference. Pages 177-182. Hamburg. Germany.

[14] Lars Erik Stacke and Dag Fritzson. (2001). "Dynamic behaviour of rolling bearings: simulations and experiments". Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology. Volume 215. Number 6. Pages 499-508. Professional Engineering Publishing

[15] Lars Erik Stacke, Dag Fritzson, and Patrik Nordling. (1999). "BEAST—a rolling bearing simulation tool". Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics. Volume 213. Number 2. Pages 63-71. Professional Engineering Publishing

Transmission line co-simulation of rolling bearing applications

Dag Fritzson, Jonas Ståhl and Iakov Nakhimovski
SKF Engineering Research Centre, Modelling & Simulation, Göteborg, Sweden
dag.fritzson@skf.com

Abstract

Connectivity, i.e., connecting different types of simulation tools, becomes increasingly important. The objective is to be able to do cost effective simulations of a large system by utilizing existing tools together. The TLM (Transmission Line Modelling) co-simulation technique is stable, does not have numerical problems, and is proven. It has potential for being a co-simulation technique of general usage. The aim of the paper is an evaluation of TLM technique for rolling bearing applications via a study of a grinding spindle model. The conclusion is that the results of TLM and non-TLM simulations are similar for a realistic application with rolling bearings and applicable for engineering purposes. The numerical stability and performance are satisfactory for the considered application.

Keywords: Co-simulation, TLM, bearing simulations, multi-body dynamics

1. Introduction

In the area of modelling and simulation of mechanical systems one can identify many different classes of models and corresponding tools. The specialization leads to different focus for different tools. For instance, consider the differences in focus for the equation-based multi-physics Modelica models [1], general multibody models in MSC.ADAMS [2], models with detailed contact definitions in SKF's BEAST (BEAring Simulation Tool) [3, 4], flexible components as modeled in FE tools [5, 6], etc.

There is no single universal tool that can be used to analyze all kind of problems with maximum efficiency and accuracy. One might say that every tool is optimized for a certain kind of tasks. This has lead to the creation of a large number of different specialized simulation frameworks. Large industrial models often use some special features of the particular framework. This makes the translation of models between different frameworks very complicated and costly.

The need to bring different components into a complete more tightly coupled simulation is therefore justified. This allows higher accuracy and preserves the investments in the components.

Co-simulation is one of the possible techniques of joining existing sub-models into a more complete model. The joining technique is very different depending on the particular application area (see, e.g., surveys in [7, 8, 9]). One method that was earlier used to enable closer interaction between such sub-models in a coupled simulation is transmission lines modelling (TLM). The TLM uses physically motivated time delays to separate the components in time and enable efficient co-simulation. The technique has proven to be stable and was implemented for coupling of different sub-systems [10, 11, 12, 13]. TLM is further described in Section 3.1. Therefore, it has a potential for being a co-simulation technique of general usage. The aim here was to investigate its suitability for rolling bearing applications. The evaluation was performed within the context of SKF BEAST.

2. Application cases

Several different application cases have been studied in order to evaluate the TLM. In the first two examples, two BEAST models were connected to each other through TLM.

2.1 Two bodies

This model was used for fundamental tests in order to investigate the basics of the concept. Two bodies are connected to each other as visualised in Figure 1.

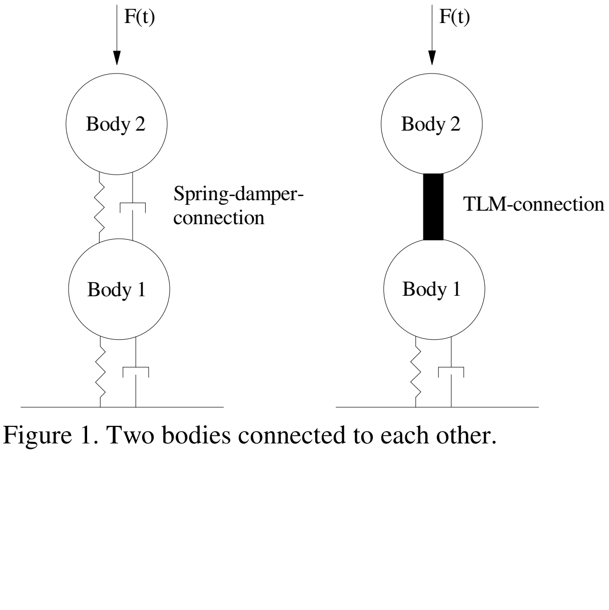


Figure 1. Two bodies connected to each other.

A spring and a damper connect Body 1 to the ground. The connection between the bodies was also initially represented by a spring and a damper, see Figure 1. Both constant and time-dependent forces were applied to Body 2. This complete model was compared to a model where the spring and the damper between the bodies were replaced by a TLM connection; see Figure 1. Same load conditions were applied in the two cases.

Static load cases give identical results. Differences in the dynamic cases were as expected. Since this model was used for fundamental tests only, no further results are presented in this paper.

2.2 Grinding spindle application

A more complex model compared to the previous one is the Grinding spindle application; see Figure 2-Figure 3. The shaft is supported in a housing by two bearings. A detailed model of a DGBB was used for one bearing. A simplified stateless model referred as "SPB" (Simplified Parametric Bearing) was used for the other bearing. The grinding wheel is attached to a hub that is mounted close to the end of the shaft. A circular mass represents a possible balancing unit. A pulley attached to the opposite end of the shaft drives the shaft. The DGBB-model consists of an inner ring, an outer ring, a cage and 11 balls. Figure 4-Figure 6 show all parts of the model separately. Altogether, the grinding spindle model consists of 20 bodies of which 14 represent the DGBB.

The stiffness of the SPB supporting the shaft was kept similar to the DGBB stiffness. The gravity was included.

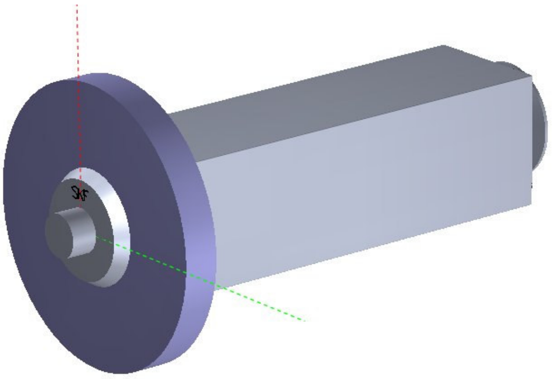


Figure 2. The grinding spindle application, view 1.

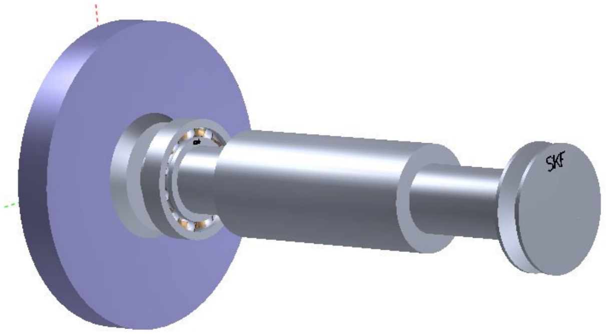


Figure 3. The grinding spindle application, view 2.

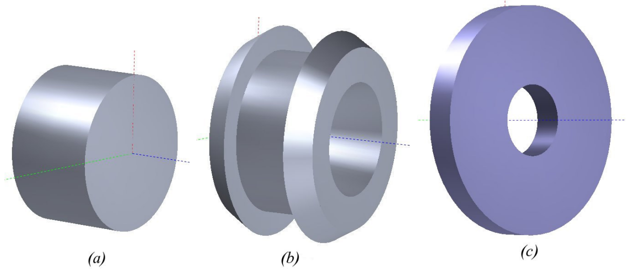


Figure 4. The balancing unit (a), the hub (b), and the grinding wheel (c).

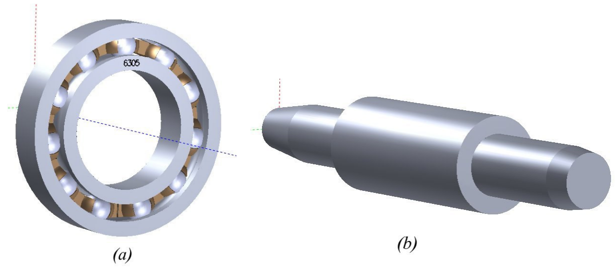


Figure 5. The DGBB (a) and the shaft (b).

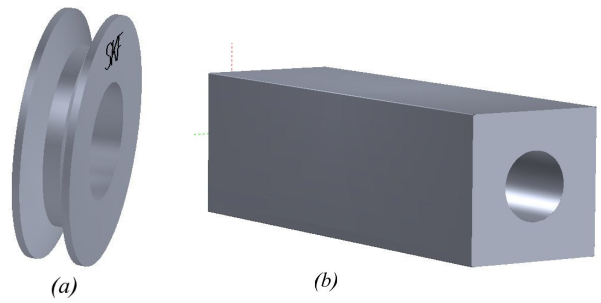


Figure 6. The pulley (a) and the housing (b).

2.2.1 Load conditions

In the studied load cases, the rotational speed of the shaft was 5000 rpm created via rotating "belt-forces" acting on the pulley. The bearings were pre-loaded axially with 100 N. A radial force of 1000 N was applied to the grinding wheel in order to simulate the grinding force. The loads and the speed were ramped up using a smooth step.

2.2.2 TLM-connections

To study the TLM-concept, some of the connections between the bodies in the original model were replaced by TLM-connections. A schematic view of the models is presented in Figure 7-Figure 8. In this case, both parts on each side of a TLM-connection were modelled with BEAST. However, there are no theoretical restrictions to the choice of software. The bearing model may be analysed with BEAST, and the surrounding parts like the shaft and the housing may be modelled in another multi-body simulation software like MSC.ADAMS. The parameters of the TLM-connections and the stiffness in the non-TLM connections had realistic physical properties. The results from the analyses of the Grinding spindle application are as previously mentioned, presented in Section 4.

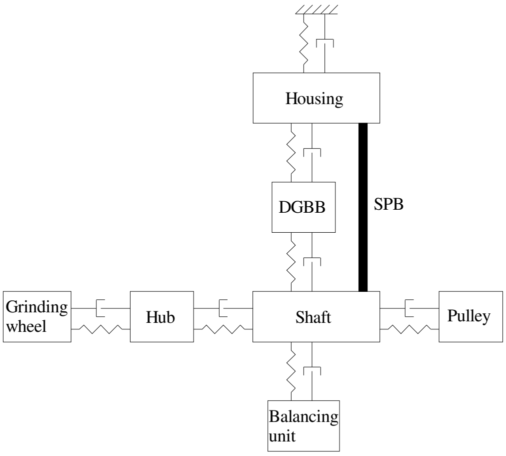


Figure 7. Schematic view of the original model.

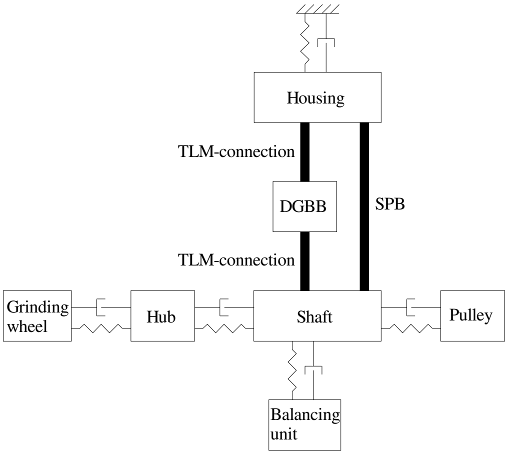


Figure 8. Schematic view of the model with TLM-connections.

3. Co-simulation techniques

Different kinds of co-simulation techniques exist, both the ordinary co-simulation techniques with exchange of state variables, and the Transmission Line Modelling co-simulation. The solution chosen for implementation in this study is the TLM concept. In the long term, the aim is to handle connections between simulation processes in different systems, such as between BEAST and some other multi-body software, by using TLM-technique. In the present investigations, BEAST/BEAST TLM connections were studied.

3.1 Transmission Line Modelling (TLM)

Transmission line modelling (TLM) has been used to model multibody and pneumatic systems with success, see references [11, 14, 15]. Since the TLM technique utilizes distributed simulators it can also be seen as a way to introduce co-simulation in a systematic way.

3.1.1 TLM Basics

Consider the system of two connected masses, see Figure 9.

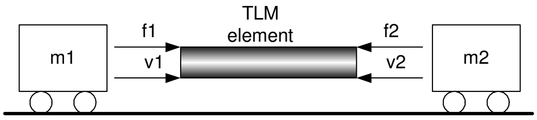


Figure 9. The TLM element communicates the forces and velocities between the masses as energy waves. The direction of the arrows shows the positive direction for energy pulses action on the TLM-element.

The TLM element is a mathematical method to decouple a system into two parts. In the case above, the two mass systems `m1` and `m2` have been decoupled and the TLM element will serve as the communication between these two systems. Some reasoning around how a real physical TLM element would work can motivate the mathematics of the TLM element.

Suppose the TLM element consisted of some elastic medium, of length *L*. A force impulse f_1 on the end of the medium will create a wave in the medium that propagates through the medium and reaches the other end. The time for this wave to reach the other end is the TLM delay time T_{TLM}. The main principle of the TLM element is that any action in one system will be propagated with a certain delay. Therefore, no direct impact (at the same time instance) between the two systems is possible. This decouples the simulation of the two systems.

Apart from the time delay the TLM element must propagate the correct information about the behaviour of the systems at the ends. By examining the PDE of the wave propagation assuming no energy loss in the TLM element, the TLM interaction can be formulated using the force and velocity at each end, and to form equations for how they interact. These equations turns out to be fairly simple since the details inside the TLM is not important just the "interface" at the ends.

$$\begin{aligned} f_1(t) &= f_2(t-T) + Z\nu_2(t-T) + Z\nu_1(t) \\ f_2(t) &= f_1(t-T) + Z\nu_1(t-T) + Z\nu_2(t) \end{aligned} \quad (1)$$

The most important property with the transmission line is the modelling of the time delay of the signals that enters a TLM element. This time delay can be used to separate the simulation into several parts that can be distributed over several different computers. In this way, it is possible to simulate the parts separately and communicate between the simulations at communication points. Using transmission line elements no numerical error is introduced, as opposed to the multi-rate integration scheme. The errors introduced are due to the separation of the model comes from modelling error instead. The TLM element is symmetric in regard to both separated simulations and no regard needs to be taken to which side is the fastest. The TLM element properties can be determined based on physical considerations only.

In the example above, an elastic medium was used to describe a time delay, but this can of course be anything that provides some reasonable time delay compared to the typical time constants in the simulations. Steel can therefore also be considered as a medium if the simulation time steps are in the same order as the time required for a sound wave to propagate through the steel material.

3.1.2 Benefits of TLM-element

The most obvious reason for using TLM is that it provides a structured way to connect to simulators in a co-simulation set-up. The TLM element does not introduce any numerical problems for the solvers. The price to pay is that a new component has to be introduced in the model. This component behaves as a medium for the communicating "wave" between the simulated processes. This therefore add a model error, but that error is always explicit and well defined compared to errors introduced "under the hood" due to numerical issues in the simulations.

In order to avoid the impact of the model error from the TLM-element it is always important to let the TLM-part represent a natural elasticity in the total system. Then the TLM-element can be tuned to adapt (or model) that elasticity as much as possible, making the model error as small as possible.

3.1.3 TLM-element formulas

Since the TLM element is a continuum, oscillation frequencies can be introduced into the simulation.

$$f_i = \frac{i}{2T_{\mathrm{TLM}}} \quad (2)$$

where T_{TLM} is the time delay factor, and *i* a positive integer. If this cause problems, a damping term can be added in the TLM formulation.

For a simple model of a steel beam, the spring constant of a TLM element can be computed as

$$k = \frac{EA}{L_0} \quad (3)$$

where *E* is Young's modulus, *A* is the cross section area and L_0 the length of the beam. The impedance Z_{F} also has a relation to the spring constant *k*, $Z_{\mathrm{F}} = k \cdot T_{\mathrm{TLM}}$. The impedance factor can then be formulated as a function of the area and length of the beam according to

$$Z_{\mathrm{F}} = \frac{EA\,T_{\mathrm{TLM}}}{L_0} \quad (4)$$

It can be shown that the TLM element also introduces a mass (*parasitic mass*) that can be viewed to be outside the simulated system. The total mass for the combined systems must therefore also include the parasitic mass of the TLM element in order to make, e.g., the energy conservation formulas correct. This mass depends on the impedance factor and the time delay factor

$$m_{\mathrm{p}} = Z_{\mathrm{F}} \cdot T_{\mathrm{TLM}} \quad (5)$$

This implies that if the impedance factor Z_{F} is increased, the parasitic mass will increase if the synchronization delay T_{TLM} is not decreased.

3.1.4 TLM-parameters

The TLM-parameters for the application case, i.e., grinding spindle (see section 2.2), were determined from real physical properties of a chosen connection volume, i.e. the part of the model that will be represented by a TLM-element and applying the formulas above. The chosen shape of the modelled volume depends on application. If a cube with a specific length, L_0, is assumed, then for the volume gives the stiffness according to Eq. (3) is:

$$k = \frac{EL_0^2}{L_0} = EL_0 \quad (6)$$

For rotation stiffness, one way is to assume two springs of stiffness $k/2$ spaced with distance L_0. The time constant T_{TLM} is computed from the specific length and the speed of sound in the material

$$T_{\mathrm{TLM}} = \frac{L_0}{v} \tag{7}$$

Using steel as the connection medium gives $E = 2.1 \cdot 10^{11}\ \mathrm{Pa}$, $v = 5180\ \mathrm{m/s}$, and $\rho = 7.8 \cdot 10^{3}\ \mathrm{kg/m^3}$.

Calculations like these give approximate values of the stiffness and the time delay of the TLM element. This gives a foundation for choosing these parameters. In the reference case for the analysis of the grinding spindle, the stiffness and time delay parameters were chosen to $k = 1 \cdot 10^{10}\ \mathrm{N/m}$ and $T_{\mathrm{TLM}} = 1 \cdot 10^{-5}\ \mathrm{s}$. Two other cases were analysed where the time delay was halved and doubled respectively, compared to the reference case, see Table 1.

Table 1. Examples of TLM-parameters used in the grinding spindle case. The stiffness was $k = 1 \cdot 10^{10}\ \mathrm{N/m}$.

Case	m_{p} [kg]	Z_{F} [Ns/m]	T_{TLM} [s]
Ref.	1.0	$1.0 \cdot 10^{5}$	$1.0 \cdot 10^{-5}$
Low T_{TLM}	0.25	$0.5 \cdot 10^{5}$	$0.5 \cdot 10^{-5}$
High T_{TLM}	4.0	$2.0 \cdot 10^{5}$	$2.0 \cdot 10^{-5}$

Decreasing the stiffness is one way to decrease the parasitic mass. However, before this is done, the stiffness of the surrounding, serially connected, parts must be estimated in order to avoid decreasing the stiffness of the TLM-connection too much. Another action is to change the delay time. Increasing the delay time makes it possible for the solver to take larger time steps and thus decrease the execution time. However, the parasitic mass is a second order function of the delay time, which has a negative influence on the simulation results. A positive effect is that an increase of the delay time counteracts a decrease of the stiffness.

3.2 Related work

So far, co-simulations gained most acceptance for the mechatronic applications. Developers of control systems routinely utilize co-simulations to test the control algorithms. The most common environment that serves as a coupling framework is SimuLink [16]. Simulation systems that support multibody dynamics simulations like MSC.ADAMS [2] or Dymola [17] provide special modules for running such co-simulations. Other competitive environments like CosiMate [18] are also available. The central issue in such simulations is often real-time performance since the models are eventually used for the hardware-in-the-loop simulations. High Level Architecture (HLA) standard [19] is designed with the vision of distributed simulations in mind. The goal here is to simulate a complete environment with many different actors and tools. The main concerns are not simulation methods but instead protocols that allow individual actors to communicate with the environment. The focus of this paper is different. The intention is to couple transient simulation of mechanical components. The main issues in this case are stability and accuracy of the coupling method. Let us provide a brief survey of the methods used for coupling of dynamics simulations. The intention is to further motivate the choice of the TLM technique for the framework. Otherwise a wider recent survey can be found in [8]. The same paper provides a classification of some of the co-simulation strategies depending on the interface variables exchanged between the sub-models and the co-simulation coordinator. The problem with setting kinematic variables into an existing simulation codes is identified. This motivates the preference for the approaches that compute only reaction forces. The authors also classify the time stepping methods typically used in co-simulations. The parallel time stepping where the compatibility between the sub-systems is achieved at every global time step is identified as the most attractive. Glue code [20, 21] is a co-simulation framework developed for connecting MSC.ADAMS to other codes with discrete or continuous time stepping. It uses quadratic interpolation to pass information about ADAMS model to the other code and quadratic extrapolation for the force coming into the ADAMS model. The latter article introduces a concept of interface mass. The interface mass is simulated in the both coupled codes and gets different reaction forces in different simulations. The reaction forces are communicated between the tools. The coupling framework was successfully used in coupling an ADAMS model to a flight simulator. The problem with this coupling approach is the lack of study on the numerical stability of the method. That leads to the necessity for additional ’tweaking’ for each particular case. The paper [22] is mostly focused on fluid-structure interaction problems but the results can be applied for other kinds of co-simulations as well. The paper analyses different numerical coupling methods and a new block-Newton method is proposed. The discussion here is focused on global solution of implicitly coupled differential equations. The introduction of the global iterative solver requires a close interaction with the numerical solvers in the sub-models and may be hard to implement for existing simulation codes. A similar problem arises for the gluing algorithm presented in [8]. The algorithm relies on the information available at the subsystem interfaces but it requires a special evaluation of the coupling matrices. This, again, may be hard to implement in an existing environment. In [23] a control block, called Boundary Condition Coordinator, is used to minimize the difference in the coupled variables. The mismatch in the conditions does not have any physical meaning and should be seen as a numerical artifact. Additionally the paper does not address the problems related to the use of multi-step implicit solvers for the sub-systems. The TLM technique combines the following attractive properties:

- Very clear and small interface to the sub-models where kinematic data is only read and the response is introduced via an external reaction force and torque.
- Proven unconditional stability of the coupling method.
- Coupling parameters are physically motivated and no additional numerical error is introduced.

The outlined combination of characteristics makes the technique an attractive candidate for a co-simulation.

3.2.1 Example: Co-simulation of a bouncing ball

To illustrate the problem of co-simulation, a co-simulation of a system with two masses according to Figure 10 is carried out. The set-up is a ball (1 kg), bouncing on a floor with mass 100 kg suspended with a spring 1000 N/m. The bounce of the ball is modelled by a Hertz contact, i.e. the acting force on the body when hitting the floor is $10^{10}(-\Delta)^{3/2}$, where Δ is the geometrical intersection.

Figure 11 shows the plots of the simulation where each body is simulated with a separate simulator that is connected using a simple co-simulation technique. The integrator that has integrated the shortest time period will take a step and extrapolate the motion from the other body. The extrapolation uses a Hermite interpolation scheme, i.e. a third order formula interpolating the position and the derivate of the other body in the two previous integration points. The co-simulation was run with two different tolerances (absolute and relative tolerance of the solution vector) of the solver. The topmost plot in Figure 11 represents a tolerance of 10^{-6} and the lower plot represents a tolerance of 10^{-8}. The behaviour of the lower plot emerges because the simulation process for the floor extrapolates the ball position too far, computes the reaction force and moves the floor downwards. When the simulation process for the ball reaches the place where the floor should be, a little time after the extrapolated time that the floor computed, the floor has already started to move downwards. The ball process extrapolates the movement of the floor, and finds that it will not hit the floor this time and continues to fall. The floor process will then extrapolate the position of the ball again and will react too much. At some point the ball will hit the floor and continue to hit the floor several times on the way upwards. This behaviour is very hard to predict, since it did not occur with a lower tolerance, thus rendering this type of co-simulation useless for our purpose.

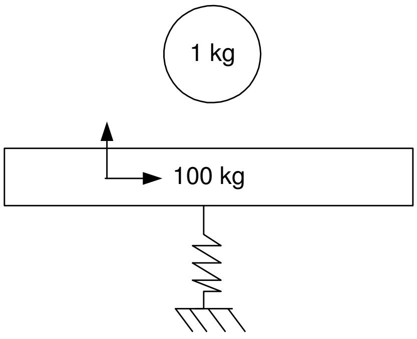


Figure 10. A ball with mass 1 kg is bouncing on a floor (mass of 100 kg) suspended with a spring 1000 N/m. Only the ball is exposed to gravity.

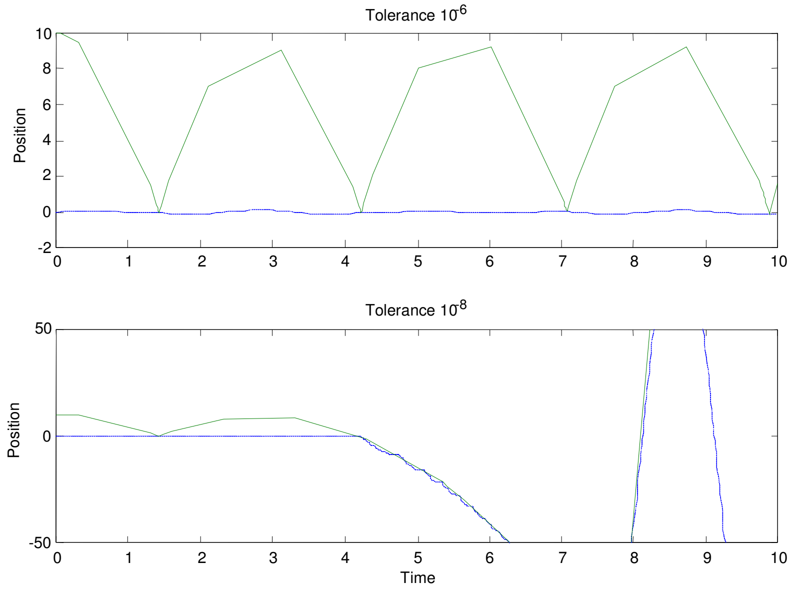


Figure 11. Co-simulation of the two bodies in Figure 17, which are interacting with a Hertz contact between them. The plots show the vertical position of the bodies where the small body (ball) is bouncing with highest altitude for tolerances 10^{-6} and 10^{-6} respectively.

4. Results from grinding spindle application case

A number of different parametric studies have been done for the grinding spindle case, described in section 2.2, and some results are presented here. Different combinations of TLM delay time, stiffness and damping parameters were used. The aim of the studies was a comparison between analyses with and without the TLM-connections. Note, that TLM elements are used in place of springs and dampers. The assumption that exactly the same results should be calculated in both models is misleading. Differences are caused by the presence of parasitic mass, see Eq. (5). However, this may actually be a more relevant description of the physical system, than mass less springs.

In Figure 12-Figure 14 views of the normal force between the inner ring and one of the balls are shown. This force is a fundamental parameter in rolling bearing analysis. The three diagrams represent different TLM-times, each compared with the non-TLM case. The normal forces are varying as the balls are passing through the loaded and unloaded zones. The variation of these zones is solely determined by the rotation of the shaft, which is a motion dominating the vibration of the different bodies in the model. Therefore, no significant difference between the cases could be identified.

The difference between the TLM-time affects as well the size of the time-steps the solver can take. Each TLM case is compared to a non-TLM case where the same time intervals have been used for the maximum time step.

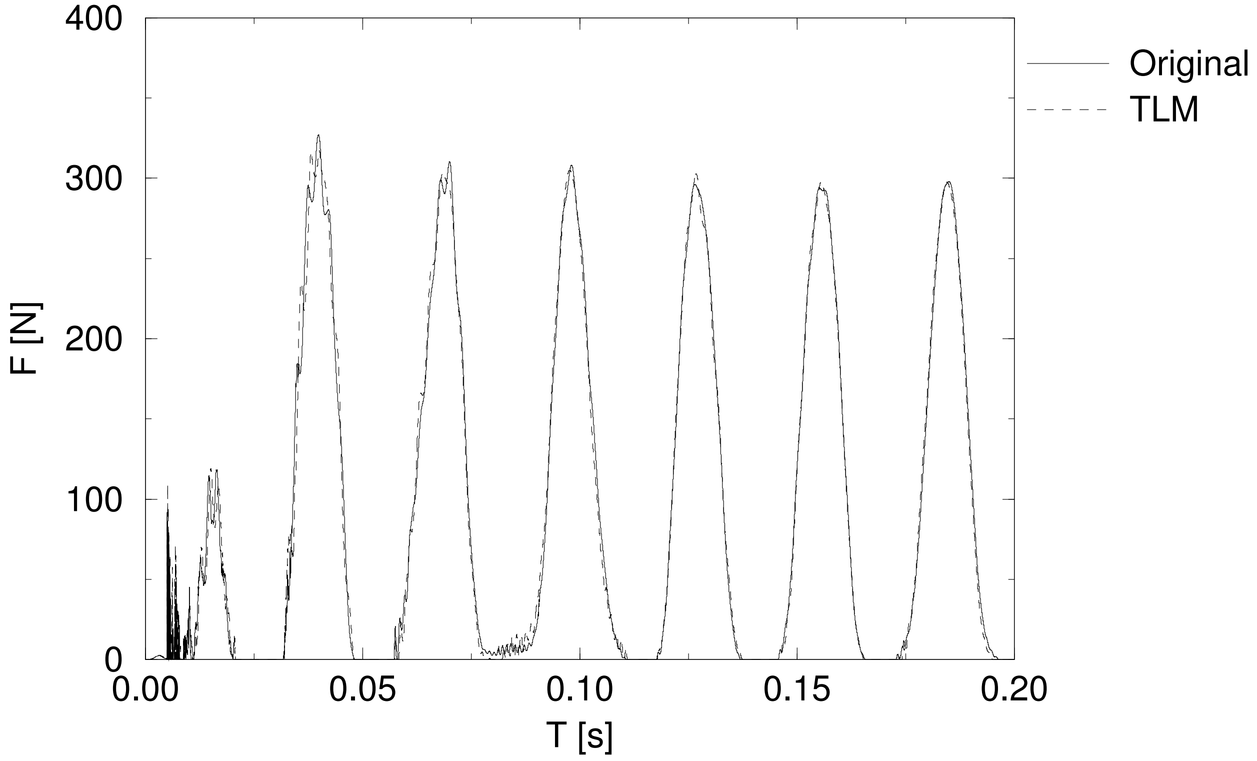


Figure 12. The normal force in the contact between the inner ring and a ball for $T_{\text{TLM}} = 1 \cdot 10^{-5}$ s .

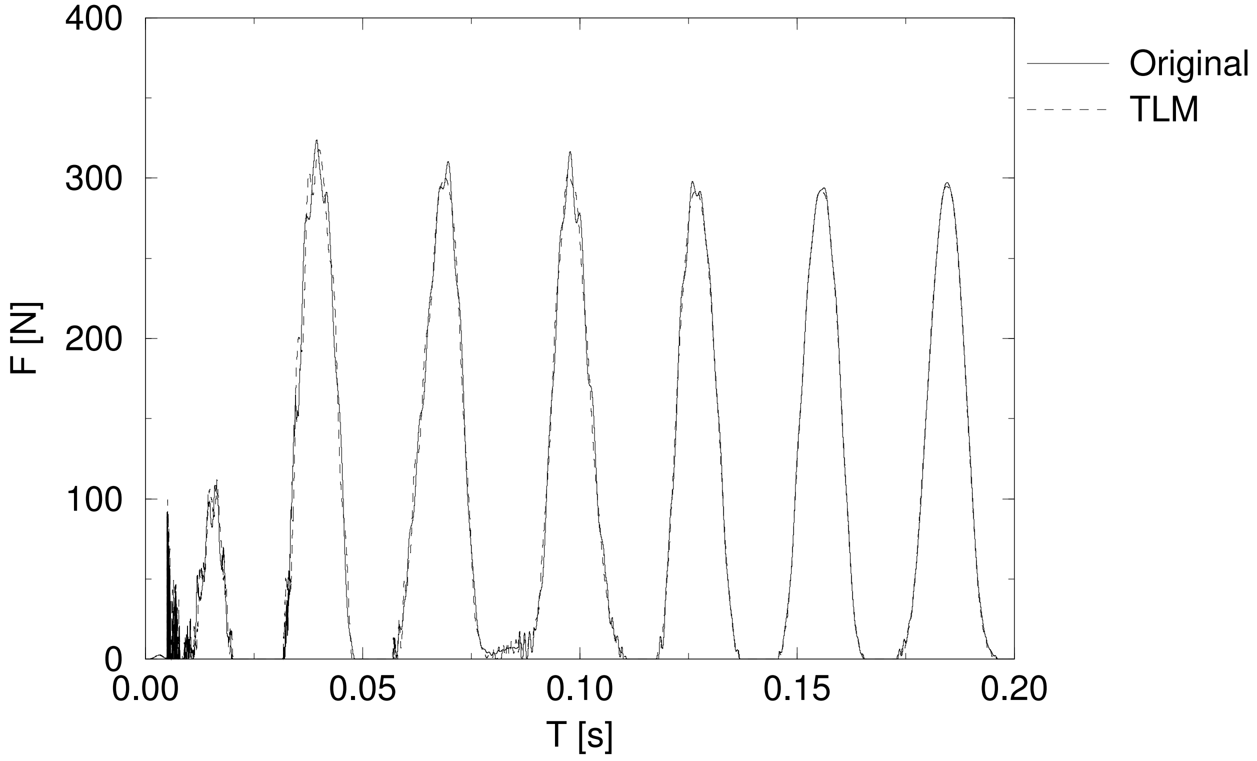


Figure 13. The normal force in the contact between the inner ring and a ball for $T_{\text{TLM}} = 0.5 \cdot 10^{-5}$ s .

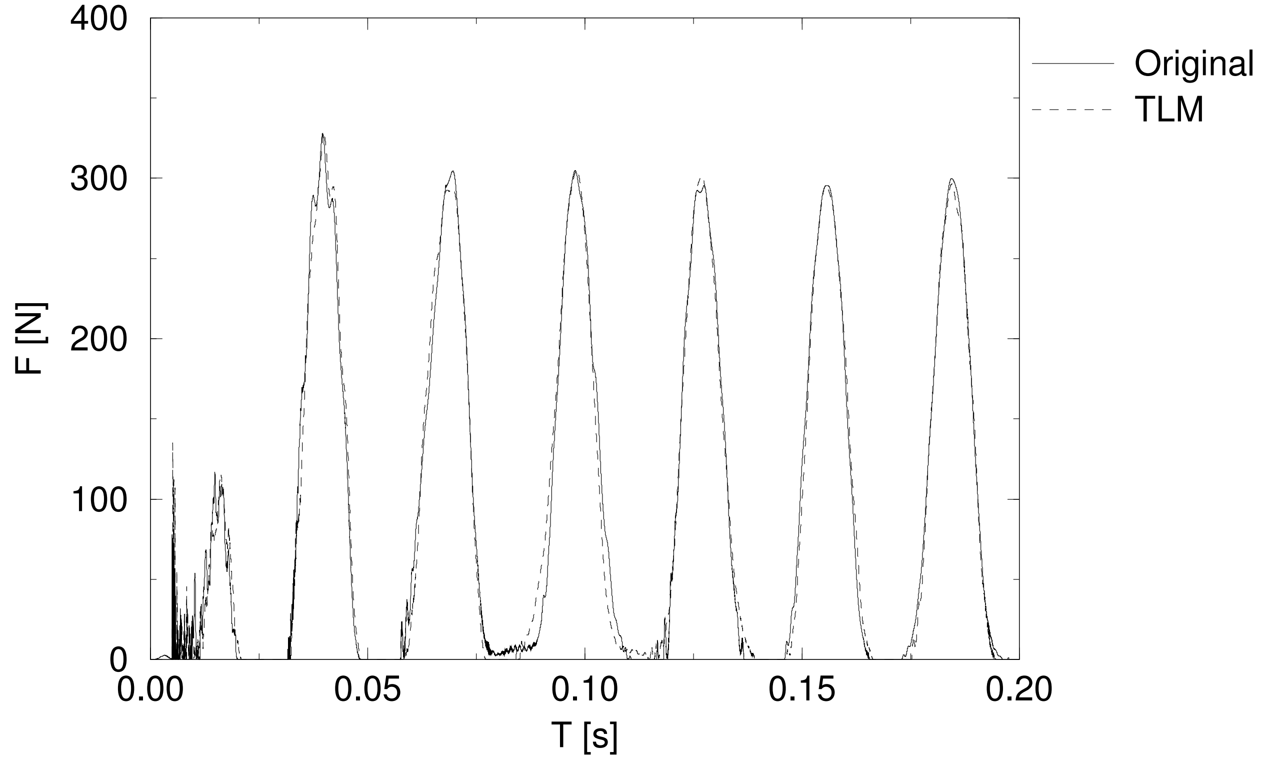
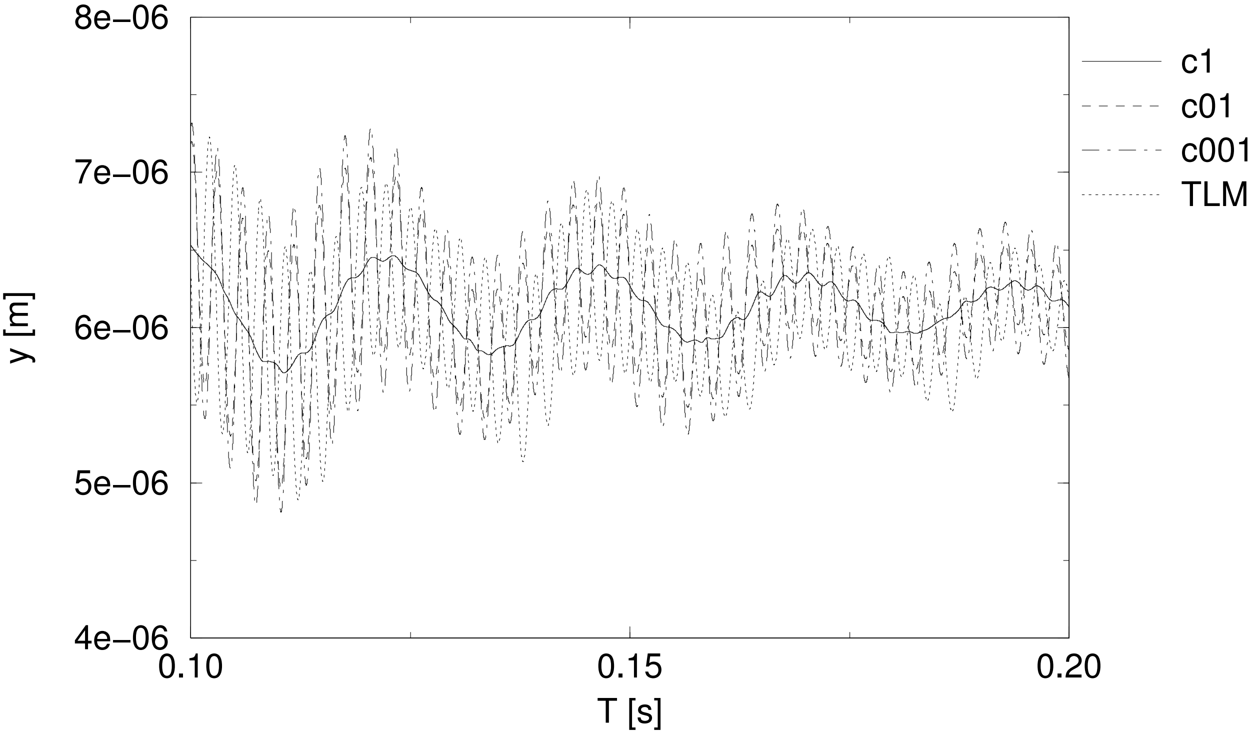


Figure 14. The normal force in the contact between the inner ring and a ball for $T_{\mathrm{TLM}} = 2 \cdot 10^{-5}\ \mathrm{s}$.

Another set of simulations was performed in order to find out how to choose the TLM parameters and how they influence the results compared to non-TLM simulations. Different combinations of the TLM delay time (T_{TLM}), the stiffness, and the damping were used. The damping parameter was only used for the non-TLM simulations since no corresponding parameter exist in a TLM-connection. The stiffness values were $k = 1 \cdot 10^{9}\ \mathrm{N/m}$ and $k = 1 \cdot 10^{10}\ \mathrm{N/m}$. The TLM delay times were $T_{\mathrm{TLM}} = 1 \cdot 10^{-5}\ \mathrm{s}$ and $T_{\mathrm{TLM}} = 2 \cdot 10^{-5}\ \mathrm{s}$. Finally, the relative damping coefficients used in the non-TLM simulations were $c = 1$, $c = 0.1$ and $c = 0.01$. Here $c = 1$ corresponds to the critical damping for an oscillating system with a mass of 1 kg. In these tests, the non-dimensional tolerance was 10^{-7}, which is a typical value for rolling bearing applications. The parasitic mass varies from 0.1 kg to 4.0 kg for the four different TLM cases.

Here, some diagrams are presented showing the results in terms of the positions of the shaft and the cage, with four graphs shown in the same diagram. The four graphs represent, for a certain combination of TLM delay time and stiffness, three non-TLM simulations (one for each damping coefficient) and one TLM simulation. The legends denoted `c1`, `c01` and `c001` represent the relative damping coefficients $c = 1$, $c = 0.1$ and $c = 0.01$ respectively.



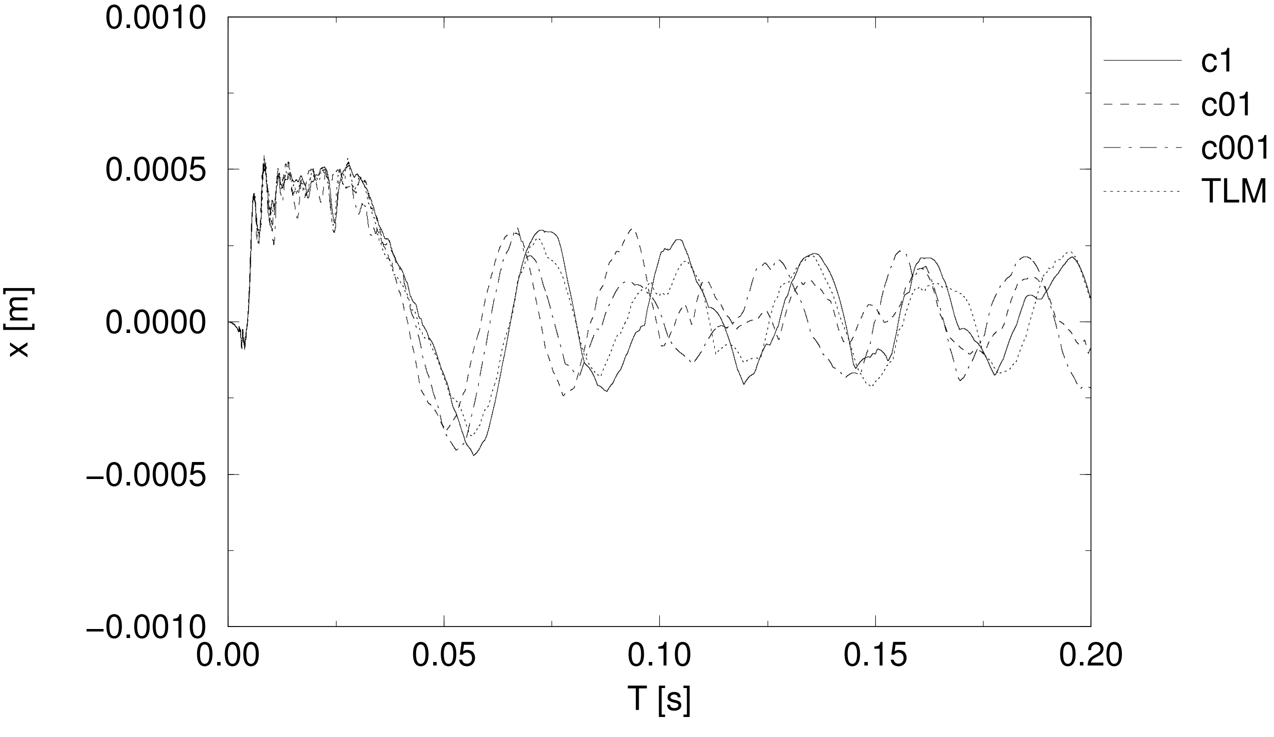
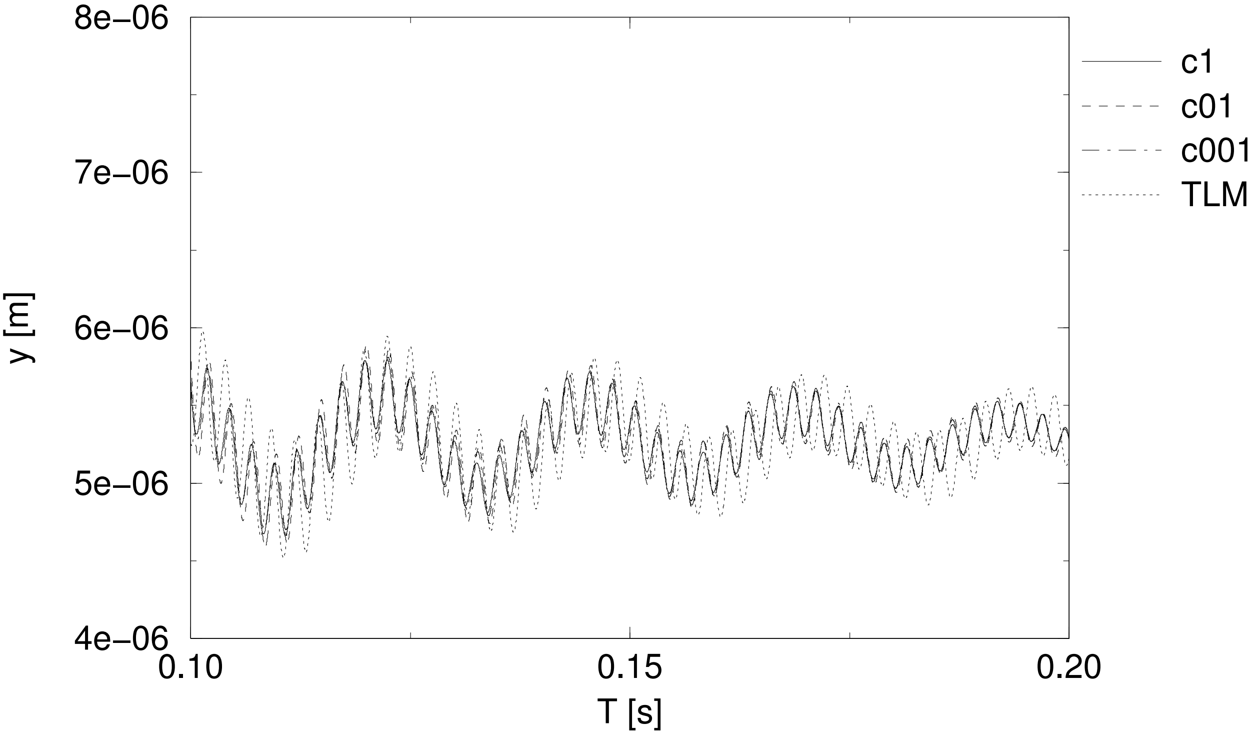


Figure 15. The position of the shaft in the horizontal (y) direction in the upper graph. The lower graph shows the vertical position of the cage. The load case were $k = 1 \cdot 10^9$ N/m and $T_{\mathrm{TLM}} = 1 \cdot 10^{-5}$ s .



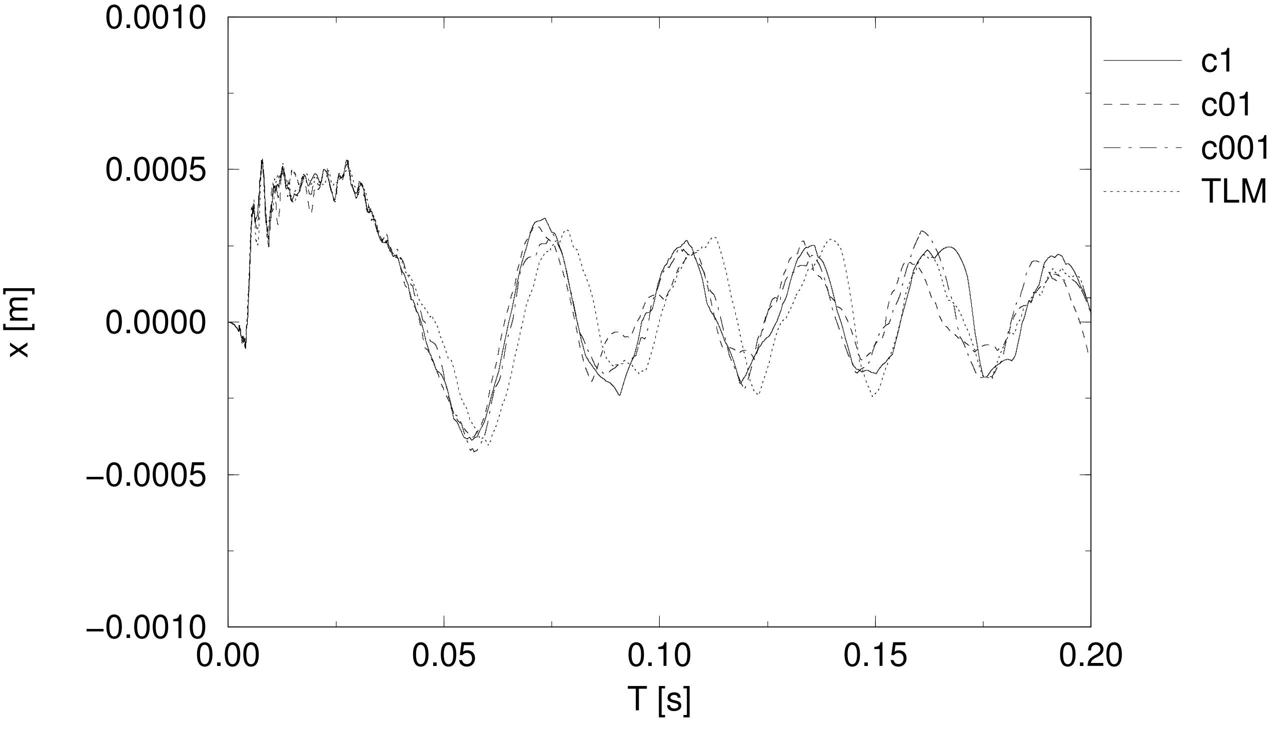
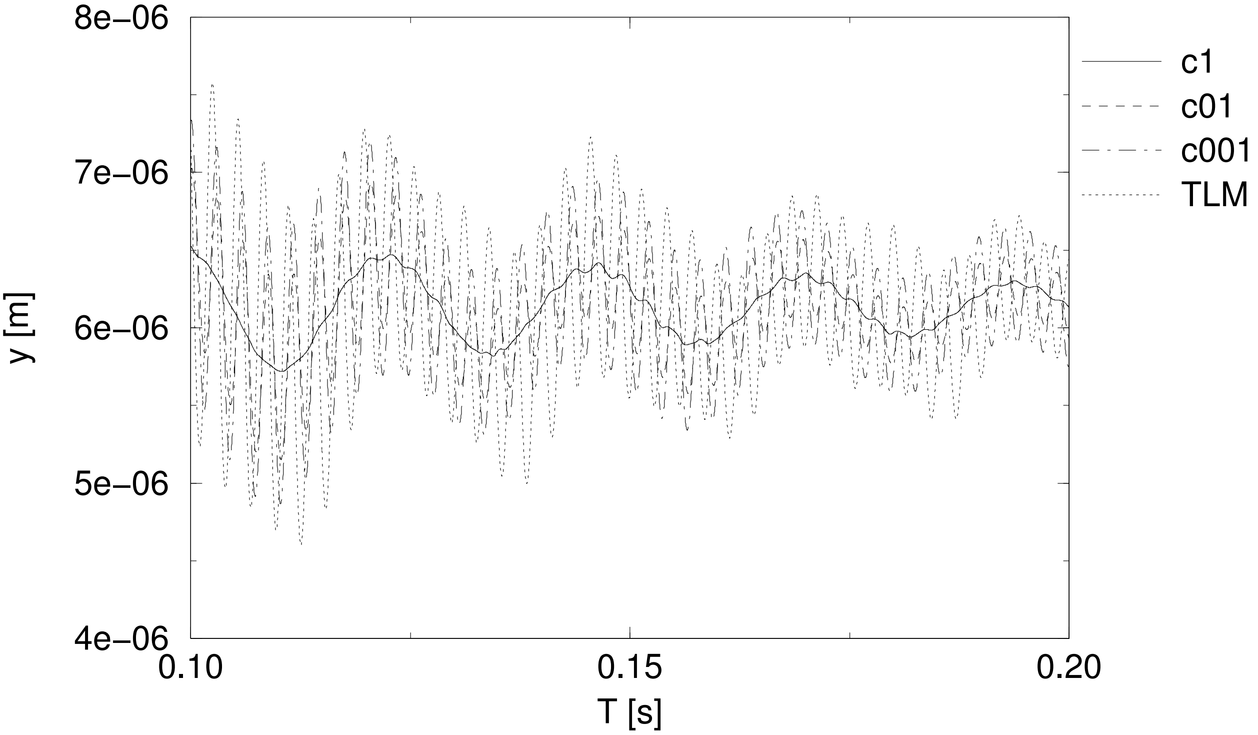


Figure 16. The position of the shaft in the horizontal (y) direction in the upper graph. The lower graph shows the vertical position of the cage. The load case were $k = 1 \cdot 10^{10}\ \mathrm{N/m}$ and $T_{\mathrm{TLM}} = 1 \cdot 10^{-5}\ \mathrm{s}$.



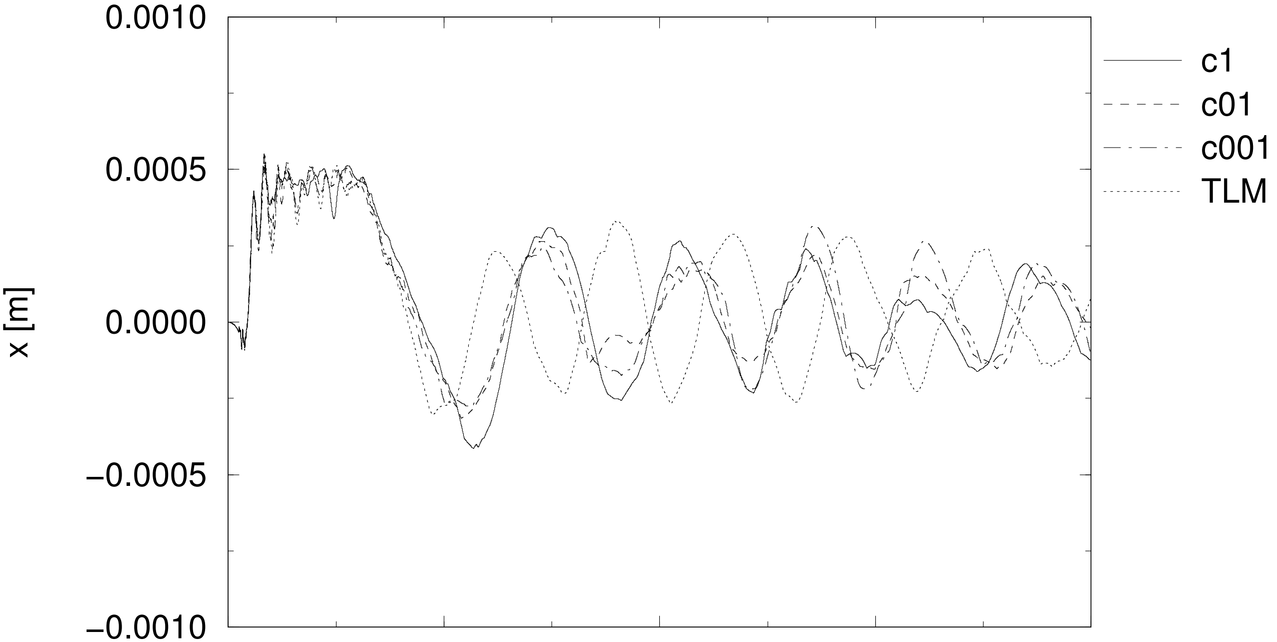
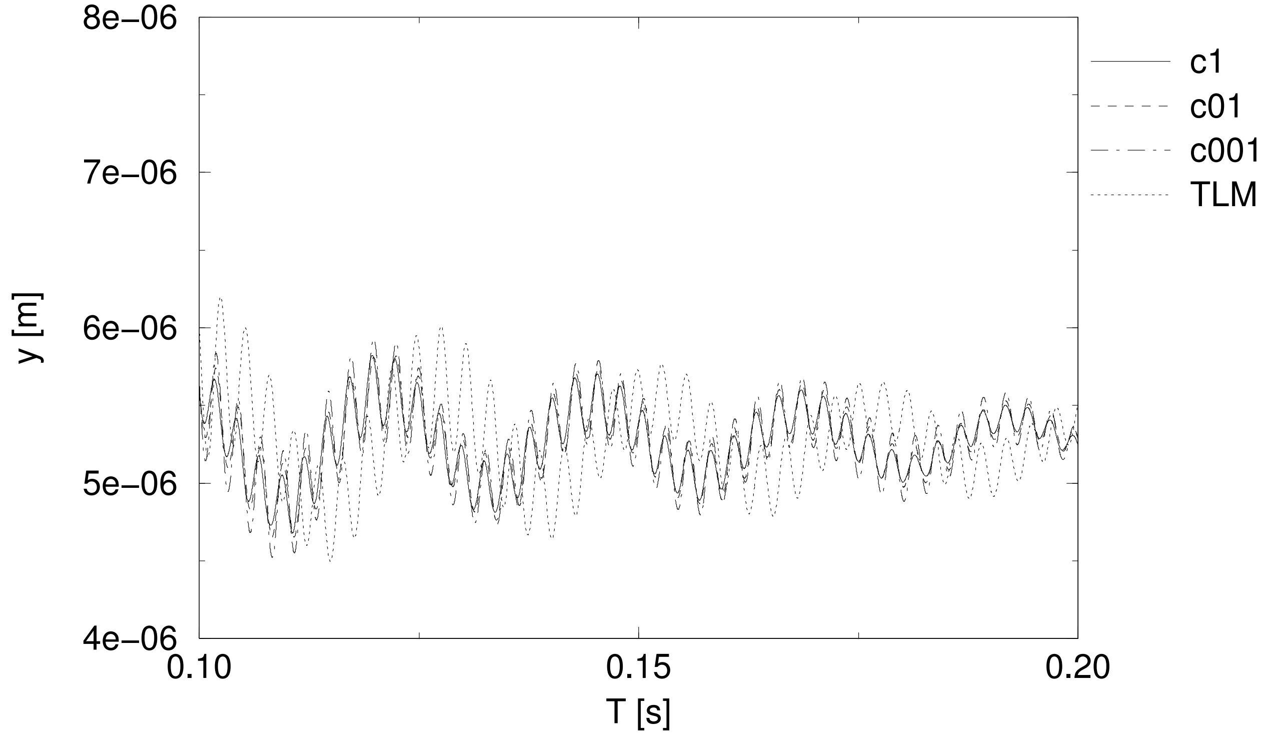


Figure 17. The position of the shaft in the horizontal (y) direction in the upper graph. The lower graph shows the vertical position of the cage. The load case were $k = 1 \cdot 10^9$ N/m and $T_{\mathrm{TLM}} = 2 \cdot 10^{-5}$ s.



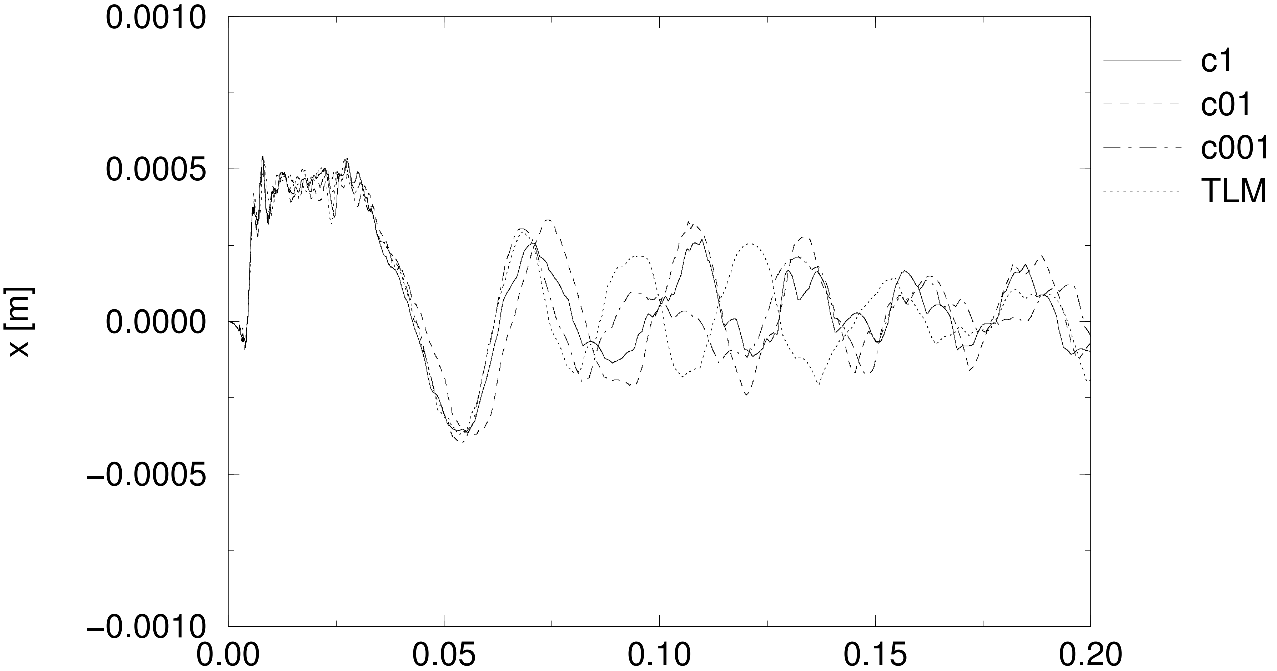


Figure 18. The position of the shaft in the horizontal (y) direction in the upper graph. The lower graph shows the vertical position of the cage. The load case were $k = 1 \cdot 10^{10}$ N/m and $T_{\mathrm{TLM}} = 2 \cdot 10^{-5}$ s.

The case with a stiffness of $k = 1 \cdot 10^{10}\ \mathrm{N/m}$ and a TLM delay time of $T_{\mathrm{TLM}} = 1 \cdot 10^{-5}\ \mathrm{s}$ is referred to as the reference case. Here the results are almost independent of the damping as clearly seen in Figure 16. The agreement between the non-TLM and the TLM analyses is good. If the stiffness is decreased, the agreement is still acceptable, at least if low damping is used. It is also natural that the oscillations increase if the stiffness of the system is decreased. An increased TLM delay time causes larger difference between the TLM analysis and the non-TLM analyses, which is especially visualised in the top diagram in Figure 18. However, this may again be explained with that it is two different systems from a physical point of view. A larger TLM delay time makes the parasitic mass larger and thus increases the difference between the two systems. It might be of some interest to increase the TLM delay time since the solver may then take longer steps and consequently decreases the calculation time. It is then possible to limit the negative influence on the results by decreasing the stiffness since this parameter counteracts the effect of an increased TLM delay time.

5. Numerical performance and stability

In this section the performance and stability of the grinding spindle test case will be analysed in detail.

In any study like this where there is a large number of parameters that might influence the results and a very large number of combinations, one can for practical reasons only investigate a subset of such combinations. A large number of simulations have been done, and below only a few of the most relevant ones are presented.

For practical reasons, all the simulations have been done in serial mode. Running serial instead of parallel BEAST calculations does not affect the analysis. All the necessary measurements that were carried out to approximately predict the behaviour in parallel model, led to this conclusion. For this application, a speedup of a factor 7-11 is expected.

When the whole application or system is simulated not using TLM, it is denoted “BRG+HS”. Using TLM means that two simulation processes communicate. One is the bearing sub-system denoted “BRG TLM” and the other the rest of the grinding spindle sub-system denoted “HS TLM”.

The parasitic mass of the TLM connection, given in the table captions, can be compared with the mass of the spindle shaft 27.8 kg, the housing 48.0 kg, the bearing outer ring 0.51 kg, and the bearing inner ring 0.33 kg.

The solver non-dimensional tolerance on state is $1.0 \cdot 10^{-7}$, which gives dimensional values of $8.334 \cdot 10^{-10}$ [m], $1.0 \cdot 10^{-7}$ [Euler parameters], $1.0 \cdot 10^{-5}$ [m/s], and $1.2 \cdot 10^{-3}$ [rad/s]. The tolerance is typical for a rolling bearing analysis.

Looking at the numerical performance for these TLM cases, the general conclusion is that the performance of the solver is similar for all cases, i.e., the maximum difference in number of RHS calls for the cases was less than 10 %. The case with a stiffness of $k = 1 \cdot 10^{10}\ \mathrm{N/m}$ and a TLM delay time of $T_{\mathrm{TLM}} = 2 \cdot 10^{-5}\ \mathrm{s}$ was the fastest. Consequently, the solver is relative unhindered to do its job. A further increase in $T_{\max}$ will give some performance increase in the order of 10 %.

For the non-TLM cases it is seen that the influence of a double $T_{\max}$ is in the order of 1-15 % fewer RHS calls and thus the solver is also here relative unrestricted and this is the least sensitive parameter. The most sensitive parameter is the damping; for the high stiffness the increase of RHS calls was about 90 % from critical (1.0) to low damping (0.01). Corresponding increase for the low stiffness was about 20 % from critical (1.0) to low damping (0.01).

Comparing the number of RHS calls for the TLM cases with the non-TLM cases, shows that the variation is much larger for the non-TLM cases that have values both lower and higher than the TLM cases. Compared to a typical TLM case, it is from minus 26 % to plus 42 %. It is the difference in physical properties of the models that give these results.

In Table 2-Table 3, two examples of TLM simulations compared with the medium non-TLM damping (0.1) are given. It serves as examples that the numerical performance and stability is good.

Table 2. Case with a stiffness of $k = 1 \cdot 10^{10}$ $\mathrm{N/m}$, a damping ratio of 0.1, with $T_{\mathrm{TLM}} = 1.0 \cdot 10^{-5}$ s, $T_{\max} = T_{\mathrm{TLM}} / 2.01$, $T_{\mathrm{sync}} = T_{\mathrm{TLM}} / 2$, and $T_{\mathrm{write}} = T_{\mathrm{TLM}}$. Parasitic mass is 1.0 kg.

Description	BRG+HS	BRG TLM	HS TLM
Number of steps	71042	72436	40205
Number of RHS calls	88321	88479	40347
Number of Jacobian calls	1253	1270	671
Failed convergence	4	7	0
Failed error test	3725	3175	1
Unsuccessful RHS calls/step %	10.1	8.51	0.005
Jacobian calls/step %	1.76	1.75	1.67
Time in solver %	0.42	0.0569	0.0122
Time in writing (with extra RHS) %	23.3	43.8	0.466
Time in RHS %	71.0	50.6	3.28
Time in Jacobian %	5.18	3.64	0.00374
Time in TLM com. %	0	1.75	99.45
Mean solver order	3.39	3.17	1.99

Table 3. Case with a stiffness of $k = 1 \cdot 10^{9}$ $\mathrm{N/m}$, a damping ratio of 0.1, with $T_{\mathrm{TLM}} = 2.0 \cdot 10^{-5}$ s, $T_{\max} = T_{\mathrm{TLM}} / 2.01$, $T_{\mathrm{sync}} = T_{\mathrm{TLM}} / 2$, and $T_{\mathrm{write}} = T_{\mathrm{TLM}}$. Parasitic mass is 0.4 kg.

Description	BRG+HS	BRG TLM	HS TLM
Number of steps	56995	68570	20129
Number of RHS calls	75217	88086	20234
Number of Jacobian calls	1028	1214	336
Failed convergence	6	6	0
Failed error test	3884	3930	2
Unsuccessful RHS calls/step %	13.2	11.1	0.02
Jacobian calls / step %	1.80	1.77	1.67
Time in solver %	0.481	0.0643	0.007
Time in writing (with extra RHS) %	17.2	42.6	0.21
Time in RHS %	76.6	52.7	0.02
Time in Jacobian %	5.71	3.78	0.002
Time in TLM com. %	0	0.837	99.7
Mean solver order	3.49	3.09	3.89

6. Discussion

The TLM delay time computed from physical properties is of the magnitude that it can influence the simulation performance. That means that the TLM input parameters needs to be selected with care, to avoid an increased simulation time. However, an increase can be compensated by that a TLM co-simulation is in fact a parallel computation.

The overall conclusion is that the performance of TLM and non-TLM simulations are similar for the grinding spindle application, and that the numerical stability and performance is very good.

7. Conclusions

Two test cases have been created, one simple two-body model used for debugging and fundamental testing, and one grinding spindle application used for evaluation of the TLM technique for rolling bearing applications.

The conclusion is that the results of TLM and non-TLM simulations are similar for a realistic application with rolling bearings and applicable for engineering purposes. The numerical stability and performance is good.

8. References

[1] Modelica standard. *http://www.modelica.org*.

[2] MSC Software Corporation homepage.
http://www.mscsoftware.com/.

[3] **Fritzson, D., Stacke, L.-E., and Nordling, P**. (1999) BEAST - a Rolling Bearing Simulation Tool. *Proc. Instn Mech Engrs*, Part K, 213, pp. 63-71, 1999

[4] **Stacke, L.-E. and Fritzson, D**. (2001). Dynamic behaviour of rolling bearings: simulations and experiments. *Proc. Instn Mech Engrs*, Part J, 213, pp. 499-508

[5] ABAQUS homepage. *http://www.abaqus.com/*.

[6] ANSYS solutions homepage.
http://www.ansys.com/.

[7] **Felippa, C.A., Park, K. C. and Farhat, C.**. (2001). Partitioned analysis of coupled mechanical systems. Invited Plenary Lecture, 4th World Congress in Computational Mechanics, Buenos Aires, Argentina, July 1998, expanded version in *Comp. Meth. Appl. Mechanics and Engineering.*, 190, 3247-3270

[8] **Hulbert, G.M., Ma, Z.-D., and Wang, J.** (2005). Gluing for Dynamic Simulation of Distributed Mechanical Systems. In Jorge A.C. Ambrosio, editor, Advances in *Computational Multibody Systems*, pages 69–94. Springer.

[9] **Veitl, A., Gordon, T., Van De Sand, A., Howell, M., Valasek, M., Vaculin, O., and Steinbauer, P.**. (1999). Methodologies for coupling simulation models and codes in mechatronic system analysis and design. *Vehicle System Dynamics Supplement*, 33:231–243.

[10] **Johns, P.B. and O'Brien, M.** (1980). Use of the transmission-line modeling (t.l.m.) method to solve non-linear lumped networks. *The Radio and Electronic Engineer*, 50(1/2): 59–70, Jan/Feb 1980.

[11] **Krus, P.** (1999). Modelling of Mechanical Systems Using Rigid Bodies and Transmission Line Joints. *Transactions of ASME Journal of Dynamic Systems Measurement and Control.*, Dec 1999.

[12] **Krus, P. and Jansson, A.** (1990). Distributed simulation of hydromechanical systems. In *Third Bath International Fluid Power Workshop*, Bath, UK.

[13] **Pulko, S.H., Mallik, A., Allen, R., and Johns, P.B.** (1990). Automatic Timestepping in TLM Routines for the Modelling of Thermal Diffusion Processes. *Int. Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 3:127-136, 1990.

[14] **Auslander, D.M.** (1968) Distributed System Simulation with Bilateral Delay-Line Models, *Trans ASME J. Basic Eng*, 1968, pp 195-200.

[15] **Krus, P.** (2000). Distributed Modelling for Simulation of Pneumatic Systems. Tech Report, Linköping University, 2000

[16] Mathworks homepage.
http://www.mathworks.com/.

[17] Dynasim and Dymola homepage.
http://www.dynasim.com/.

[18] Cosimate homepage.
http://www.tni-world.com/cosimate.asp/.

[19] **Kuhl, F., Weatherly, R. and Dahmann, J.** (2000). Creating Computer Simulation Systems. An Introduction to the High Level Architecture. Prentice Hall PTR, 2000

[20] **Elliot, A.S.** (2000). A Highly Efficient, General-Purpose Approach for Co-Simulation with ADAMS. MDI Inc, Nov 2000

[21] **Elliot, A.S.** (2002). Status Update on Advanced, General-Purpose Co-Simulation with ADAMS. MDI Inc, May 2002.

[22] **Matthies, H.G. and Steindorf, J.** (2002). Strong coupling methods (revised version). In Stuttgart Multifield Conference, Stuttgart, Germany, April 2002

[23] **Ben Gum, H., Asada, H., He, X.** (2002). Software Development of Co-Simulation of Algebraically Coupled Dynamic Subsystems without Disclosure of Proprietary Subsystem Models, IMECE, New Orleans, Louisiana, Nov 2002

Regularized, Stabilized, Variational Methods for Multibodies.

Claude Lacoursière
HPC2N/VRlab
and
Department of Computing Science
Umeå University
SE-901 87, Umeå, Sweden
claude@hpc2n.umu.se

September 30, 2007

Abstract

A time-discrete formulation of the variational principle of mechanics is used to construct a novel first order, fixed time step integration method for multibody systems subject to mixed constraints. The new stepper, coined Spook, includes physics motivated constraint regularization and stabilization terms. The stepper is proved to be stable for the case of linear constraints, for non-zero regularization and stabilization parameters. For fixed stabilization value, the regularization can be made arbitrarily small, corresponding to arbitrarily stiff penalty forces. The "relaxed" constraint formulation permits a separation of time scales so that stiff forces are treated as relaxed constraints. Constraint stabilization makes the stiff forces modeled this way strictly dissipative, and thus, the stepper essentially filters out the high oscillations, but is rigorously symplectic for the rest of the motion. Spook solves a single linear system per time step and is insensitive to constraint degeneracies for non-zero regularization. In addition, it keeps the constraint violations within bounds of $O(h^2)$, where h is the time step. Because it is derived from the discrete variational principle, the stepping scheme globally preserves the symmetries of the physical system. The combination of these features make Spook a very good choice for interactive simulations. Numerical experiments on simple multibody systems are presented to demonstrate the performance and stability properties.

1. Introduction

Integration methods that preserve the qualitative aspects of given sets of differential equations have captured the interest of numerical analysts over the last decade [13]. This investigative effort has shed much light on the mystery of surprisingly well behaved low order formulae widely used in the physical sciences, such as the Verlet [25], Shake, and Rattle methods of molecular dynamics [20], or the Newmark method in structural dynamics, to name just a few.

The recently developed theory of the *discrete least action* principle [22] provides an explanation for this good behavior. Indeed, all the aforementioned integration methods can be derived from the discrete least action principle. In turn, this means that they are guaranteed to be symplectic and to uniformly preserve all physical invariants withing bounds related to the time step. It is one thing to preserve energy, say, locally within $O(h^p)$, for step of size h and a method of order p, so that $|E_{k+1} - E_k| = O(h^p)$ at each step k. But it is yet more reassuring to have the global bound $|E_k - E(0)| = O(h^p)$ for all steps $k = 1, 2, \ldots$, where $E(0)$ is the known energy of the system. For the simplistic looking Verlet stepping scheme for instance, momentum is *exactly* preserved at machine precision, as long as the local stability requirements are met, irrespective of the time step. In addition, for closed physical systems with fixed energy, preservation of symmetries implies a *shadowing property*, meaning that computed trajectories intersect the analytical solution once per step [11, 15]. Therefore, the discrete samples of the trajectory might not be very accurate but they never deviate far from the exact motion, another good property. If one takes the view that the most fundamental aspects of physical motion are the symmetries, the discrete least action principle is the appropriate technique to use for constructing new integration schemes for physical systems.

Particularly in the context of real-time interactive applications, such as simulator training systems for instance, the time constraints on computations are uniformly stringent. In addition, inputs signals are not known in advance and the system configuration can be changed at arbitrary times. In consequence, low order, single step, strongly stable, fixed step size integration methods are best suited, and the discrete least action principle is the best tool to construct them.

This is the route followed in the present article in which a novel stepping scheme is constructed for constrained mechanical systems. In particular, the issues of constraint stabilization, constraint degeneracy, time scale separation are addressed directly. The same ideas were also applied to dry frictional contact problems in my PhD thesis [18] but due to space limitations, this is not covered in the present article.

Mechanical integrators are constructed directly from a discretization of a time integral of the Lagrangian, expressed in terms of discrete time samples of the systems generalized coordinates. For the low order methods, the stepping equation are three-terms recurrence relations which are in fact the necessary stationarity conditions dictated by the least action principle for conservative systems, or the Fourier-d'Alembert principle of vanishing virtual work for non-conservative ones. The latter principle expresses the equations of motion in weak form. In turn, the weak form provides a straight forward strategy for discretizing discontinuous phenomena such as frictional impacts and contacts, though this is not done here.

The rest of the paper is organized as follows. In § 2, the discrete least action principle is introduced and applied to mechanical systems with finite degrees of freedom. Both conservative and non conservative systems are covered, as well as both holonomic and nonholonomic kinematic constraints. In § 3, the novel physics based constraint regularization and stabilization scheme is presented. This is then linearized in § 4 to produce the Spook stepping scheme. The linear stability is investigated in § 5. The results of numerical experiments are presented in § 6. Conclusions and summary remarks are collected in § 7.

2. Discrete mechanics

Consider a finite-dimensional mechanical system with configuration space Q, generalized coordinates $q : \mathbb{R} \mapsto Q$, and generalized velocities $q : \mathbb{R} \mapsto TQ$, where TQ is the tangent bundle of Q. Introduce the mass matrix $M(q)$ so the kinetic energy $T : TQ \mapsto \mathbb{R}$ is a positive definite quadratic form $T(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q}$, and is positive definite. Finally, introduce the continuously differentiable potential energy $V : Q \mapsto \mathbb{R}$. The Lagrangian is then

$$\mathcal{L}(q, \dot{q}) = T(q, \dot{q}) - V(q) = \frac{1}{2}\dot{q}M(q)\dot{q} - V(q), \qquad (1)$$

and the action of a given trajectory $(q(t), \dot{q}(t)), t \in [t_0, t_1]$ is the functional

$$S[q] = \int_{t_0}^{t_1} \mathrm{d}s\mathcal{L}(q(s), \dot{q}(s)). \qquad (2)$$

The principle of least action [19] then states that for any infinitesimal variation δq satisfying all the constraints on the system and such that $\delta q(t_0) = \delta q(t_1) = 0$, then, $\delta S[q] = 0$. This means that $S[q]$ is stationary for the physical trajectory $q(t)$.

For a system subject to non-conservative or polygenic forces $f(q, \dot{q}, t)$, the Fourier-d'Alembert principle of virtual work applies, namely

$$\delta S[q] + \int_{t_0}^{t_1} \mathrm{d}s f^T \delta q(s) \leq 0, \qquad (3)$$

where the inequality holds in case Q is closed and bounded and $q(t)$ touches the boundary for $t \in [t_0, t_1]$. A particularly important form of polygenic forces can be written as

$$f(q, \dot{q}) = -\frac{\partial \mathfrak{R}(q, \dot{q})}{\partial \dot{q}}, \qquad (4)$$

where $\mathfrak{R} : Q \times TQ \mapsto \mathbb{R}$ is a generalized Rayleigh dissipation function. A simple example of that is the viscous drag force $f = -\gamma D\dot{q}$ where D is a constant square, symmetric, positive semi-definite real matrix. This corresponds to $\mathfrak{R} = \frac{\gamma}{2}\dot{q}^T D\dot{q} \geq 0$.

The general least action principle leads to the Euler-Lagrange equations of motion

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T} = f(q, \dot{q}, t), \qquad (5)$$

which is a set of nonlinear second order differential equations which can be solved with general methods. Observe here that (3) is a weak form of the equations of motion which can be used to analyze nonsmooth phenomena.

Instead of using a standard numerical method on (3), it is possible to discretize the action directly in terms of discrete samples along the trajectory, namely, $q_k = q(kh), k = 0, 1, \ldots, N$, where $h > 0$ is the fixed time step. Introduce the discrete Lagrangian

$$\mathbb{L}_d(q_0, q_1, h) = \int_0^h \mathrm{d}s\mathcal{L}(q(s), \dot{q}(s)), \qquad (6)$$

and the discrete action

$$\mathbb{S}_d(q_0,\ldots,q_N,h) = \sum_{k=0}^{N-1} \mathbb{L}_d(q_k,q_{k+1},h). \qquad (7)$$

The discrete action is then a multivariate function $\mathbb{S}_d(q_0,\ldots,q_N,h) : \otimes^{N+1}Q \mapsto \mathbb{R}$. For a conservative, a stationary point over the discrete trajectory $(q_0,q_1,\ldots,q_N)$ must have vanishing partial derivatives with respect to each of the discrete sample points q_k, which means

$$\begin{aligned}&\frac{\partial\mathbb{S}_d(q_0,\ldots,q_N,h)}{\partial q_k}\\ &\quad= \frac{\mathbb{L}_d(q_{k-1},q_k,h)}{\partial q_k} + \frac{\mathbb{L}_d(q_k,q_{k+1},h)}{\partial q_k}\\ &\quad= D_1\mathbb{L}_d(q_{k-},q_k,h) + D_2\mathbb{L}_d(q_k,q_{k+1},h) = 0, \qquad (8)\end{aligned}$$

where D_k is the operator yielding the partial derivative with respect to the kth argument. For a system subject to polygenic forces, first write

$$\begin{aligned}&\int_{t_k}^{t_{k+1}} \mathrm{d}s f^T\delta q =\\ &\qquad \delta q_k^T f^{(+)}(k,k+1) + \delta q_{k+1}^T f^{(-)}(k,k+1). \qquad (9)\end{aligned}$$

Then, the discrete Fourier-d'Alembert principle is

$$\begin{aligned}&\Bigg(\frac{\partial\mathbb{S}_d(q_0,\ldots,q_N,h)}{\partial q_k}+\\ &\qquad f^{(+)}(k-1,k)^T + f^{(-)}(k,k+1)^T\Bigg)\delta q_k = 0, \qquad (10)\end{aligned}$$

and after transposing, this reads

$$\begin{aligned}&D_1^T\mathbb{L}_d(q_k,q_{k+1},h) + D_2^T\mathbb{L}_d(q_{k-1},q_k,h)\\ &\qquad + f^{(+)}(k-1,k) + f^{(-)}(k,k+1) = 0. \qquad (11)\end{aligned}$$

Observe now that either (8) or (11) are discrete stepping equations defining a map $\Phi : Q\times Q \mapsto Q\times Q$ so that $(q_k,q_{k+1}) = \Phi(q_{k-1},q_k)$. The map Φ is called a either a variational or a discrete mechanical integrator [22]. The main benefit of this integration strategy is that the discrete trajectory is in fact symplectic. In addition, a discrete form of Noether's theorem applies to the discrete variational principle which means that physical symmetries are preserved by the discrete trajectories. For fixed time integration step, energy is not preserved exactly but oscillates within bounds of $O(h^2)$ of the energy of the continuous system. Because global invariants are preserved by the discrete integrators, the trajectory produced are more faithful to the physics, even at low integration order, than general methods with higher local accuracy but without global error bounds, such as standard Runge-Kutta or multistep methods [13].

There are several choices for defining $\mathbb{L}_d(q_k,q_{k+1},h)$ and $f^{(\pm)}$ but for the present paper, the following choices are made when there is a need to be specific

$$\begin{aligned}&q = q_0, \qquad \dot{q} = \frac{1}{h}(q_1-q_0) = v_1,\\ &\mathbb{L}_d(q_0,q_1,h) = \frac{1}{2h}(q_1-q_0)^T M(q_0)(q_1-q_0) - hV(q_0)\\ &\qquad f^{(+)}(0,1) = 0\\ &\qquad f^{(-)}(0,1) = hf(q_1,\frac{1}{h}(q_1-q_0),0). \qquad (12)\end{aligned}$$

Adding holonomic and nonholonomic constraints to this analysis is straight forward [22, 8]. Given holonomic constraints of the form $g(q) = 0$ and Pfaffian-form nonholonomic constraints $a(q,\dot{q}) = A(q)\dot{q} = 0$, the extensions read

$$\begin{aligned}&D_1^T\mathbb{L}_d(q_k,q_{k+1},h) + D_2^T\mathbb{L}_d(q_{k-1},q_k,h)+\\ &hG_k^T\lambda + hA_k^T\alpha + f^{(+)}(k-1,k) + f^{(-)}(k,k+1) = 0\\ &\qquad g(q_{k+1}) = 0\\ &\qquad A_{k+1}(q_{k+1}-q_k) = 0. \qquad (13)\end{aligned}$$

This result is easily understood as an application of the theory of constrained multivariate extrema.

Stepping formulae directly or closely related to (13) include SHAKE [24] and RATTLE [1]. These are in fact extensions of the Verlet [25] stepping scheme which is common in molecular dynamics [20]. Such schemes have been all but common in multibody dynamics except for some recent applications in computer graphics [12].

The constrained stepping equations (13) form a nonlinear system of equations which has to be solved to yield the updated kinematic state (q_{k+1},q_k), as well as the constraint forces, λ,α. Questions remain as to how much numerical precision is needed in solving (13) to preserve stability and the qualitative aspects of mechanical integrators. This is addressed to some extent in the next section.

3. Constraint regularization and stabilization

It is well-known that kinematic constraints of the form $g(q) = 0$ are in fact the limit of strong penalty terms of the form $U_\epsilon = \frac{1}{2\epsilon}g^Tg$, as $\epsilon \to 0$. Likewise, non-holonomic constraints can be understood as the limit of Rayleigh dissipation terms of the form $\frac{1}{2\gamma}\dot{q}^TA^T(q)A(q)\dot{q}$, as $\gamma \to 0$. The theory behind this fact is called *constraint realization*[23, 16, 17, 6, 7]. What is also well known is that introducing penalty forces without taking appropriate precaution against high frequency, low amplitude oscillations, leads to numerical catastrophe. For holonomic constraints, detailed analysis [23] demonstrates that though the trajectories $(q_\epsilon, \dot{q}_\epsilon)$ of the penalized system do converge uniformly to the trajectories of the constrained system, $(q, \dot{q})$, say, but the penalized constraint forces λ_ϵ can fail to converge to the constraint forces λ.

The remedy to these problems is to use implicit time integration for the constraint forces and to introduce sufficient damping to guarantee stability. The result is a "relaxed" constrained system for which accuracy requirements on the solution of the nonlinear stepping equations (13) is dramatically reduced.

Consider first the case of holonomic constraints $g(q) = 0$ with Jacobian $G = \partial g/\partial q$. Introduce the Lagrangian term

$$\mathcal{L}_\epsilon(q, \dot{q}, \lambda, \dot{\lambda}) = \frac{\epsilon}{2}\|\lambda\|^2 + \lambda^T g(q). \tag{14}$$

Considering λ as a dynamical variable, the Euler-Lagrange equations of motion of the augmented system $\mathcal{L}(q, \dot{q}) + \mathcal{L}_\epsilon(q, \dot{q}, \lambda, \dot{\lambda})$ are

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T} - G^T\lambda = 0 \\ \epsilon\lambda + g(q) = 0, \end{aligned} \tag{15}$$

and the standard constrained equations are recovered in the limit where $\epsilon \to 0$. To discretize this, put

$$\begin{aligned} \int_0^h \mathrm{d}s \left(\epsilon\lambda^2 + \lambda^T g(q)\right) = \\ \frac{h\epsilon}{8}\|\lambda_0 + \lambda_1\|^2 + \frac{h}{4}(\lambda_0 + \lambda_1)^T(g_0 + g_1), \end{aligned} \tag{16}$$

where $g_k = g(q_k)$. In the variational integration framework, this discretization corresponds to the implicit midpoint rule. After applying the discrete principle of least action and relabelling $\lambda = \frac{1}{4}(\lambda_{k+1} + 2\lambda_k + \lambda_{k-1})$, the discrete Euler-Lagrange equations read

$$\begin{aligned} D_1^T\mathbb{L}_d(q_k, q_{k+1}, h) + D_2^T\mathbb{L}_d(q_{k-1}, q_k, h) \\ + hG_k^T\lambda = 0 \\ \epsilon\lambda + \frac{1}{4}(g_{k+1} + 2g_k + g_{k-1}) = 0. \end{aligned} \tag{17}$$

In the limit where $\epsilon \to 0$, provided $g_0 = g_1 = 0$, the standard variational constrained equations are recovered. As shown further below however, this stepping scheme is at least linearly stable for $\epsilon > 0$. The λ variables are now regarded as the coordinates of a ghost point particle with zero mass. A short analysis reveals that if they did have finite mass, it would have to be negative to preserve physical stability, hence the term "ghost".

For nonholonomic constraints $a(q, \dot{q}) = A(q)\dot{q} = 0$, introduce the Rayleigh function

$$\mathfrak{R}\ a(q, \dot{q}, \alpha, \dot{\alpha}) = -\left(\frac{\gamma}{2}\|\dot{\alpha}\|^2 + \dot{\alpha}^T a(q, \dot{q})\right), \tag{18}$$

producing the force terms

$$\begin{aligned} f_q &= -\frac{\mathfrak{R}}{\partial \dot{q}^T} = -A^T\dot{\alpha}, \\ f_\alpha &= -\frac{\mathfrak{R}}{\partial \dot{\alpha}^T} = \gamma\dot{\alpha} + A(q)\dot{q}. \end{aligned} \tag{19}$$

This form of Rayleigh dissipation function has not been found in the literature. The reason for using $\dot{\alpha}$ instead of α itself, is that α can be treated like a ghost variable, subjected to the dissipative force $f_\alpha = -\partial\mathfrak{R}/\partial\dot{\alpha}^T$.

With the choices

$$f_q^{(-)}(0,1) = A_0(\alpha_1 - \alpha_0), \tag{20}$$

$$f_q^{(+)}(0,1) = 0, \tag{21}$$

$$f_\alpha^{(-)}(0,1) = \gamma(\alpha_1 - \alpha_0) + A_1(q_1 - q_0), \tag{22}$$

$$f_\alpha^{(+)}(0,1) = 0, \tag{23}$$

the discrete Euler-Lagrange equations become

$$\begin{aligned} D_1^T\mathbb{L}_d(q_k, q_{k+1}, h) + D_2^T\mathbb{L}_d(q_{k-1}, q_k, h) + A_k^T\alpha = 0 \\ \gamma\alpha + A_{k+1}(q_{k+1} - q_k) = 0, \end{aligned} \tag{24}$$

after defining $\alpha = \alpha_{k+1} - \alpha_k$, and the previously known variational stepping nonholonomic stepping scheme [8] is recovered in the limit where $\gamma \to 0$.

Combining these two results, holonomic constraints $g(q) = 0$ can be stabilized by introducing the dissipation term

$$\mathfrak{R}_g(q, \dot{q}, \lambda, \dot{\lambda}) = -\tau\left(\frac{\epsilon}{2}\|\dot{\lambda}\|^2 + \dot{\lambda}G(q)\dot{q}\right). \tag{25}$$

Reusing the definitions provided in (20–23), the resulting discrete Euler-Lagrange equations are now

$$\begin{aligned} D_1^T\mathbb{L}_d(q_k, q_{k+1}, h) + D_2^T\mathbb{L}_d(q_{k-1}, q_k, h) + hG_k^T\lambda = 0 \\ \epsilon\lambda + \frac{1}{4}\left(g_{k+1} + 2g_k + g_{k-1}\right) + \frac{\tau}{h}G_{k+1}(q_{k+1} - q_k) = 0, \end{aligned} \tag{26}$$

with the definition $\lambda = \frac{1}{4}(\lambda_{k+1} + 2\lambda_k + \lambda_{k-1}) + \frac{\tau}{h}(\lambda_{k+1} - \lambda_k)$. The reason for this labelling is that λ_k itself is not necessary to step the system forward and the aggregate variable is more convenient.

4. The Spook stepper

Consider a systems with constant mass matrix $M(q) = M$ with n degrees of freedom so $q, \dot{q} \in \mathbb{R}^n$. Add smooth, slowly varying potential energy $V(q)$. Subject this system to a combination of m_h holonomic constraints of the form $g_i(q) = 0, i = 1, \ldots, m_h$ with $m_h \times n$ Jacobians $G = \partial g/\partial q$, and m_n nonholonomic constraints $a_i(q), i = 1, \ldots, m_n$ so $a(q) = A(q)\dot{q} - w(t) = 0$, where $A(q)$ is a smooth $m_n \times n$ matrix, and $w : \mathbb{R} \mapsto \mathbb{R}^{m_n}$

is a smooth driving term. Add smooth non-potential forces discretized so that $f^{(-)}(k, k+1) = 0$ in (9). Then, choose a time step $h > 0$, as well as m_h pairs of positive scalars $\epsilon_i, \tau_i > 0, i = 1, \dots, m_n$, and a further set of m_n of positive scalars $\gamma_i > 0, i = 1, \dots, m_n$. Assemble these scalars in the following diagonal matrices

$$\begin{aligned}\Gamma &= \frac{1}{h}\operatorname{diag}(\gamma_1, \gamma_2, \dots, \gamma_{m_n}),\\ \Sigma &= \frac{4}{h^2}\operatorname{diag}(\frac{\epsilon_1}{1+4\frac{\tau_1}{h}}, \frac{\epsilon_2}{1+4\frac{\tau_2}{h}}, \dots, \frac{\epsilon_{m_h}}{1+4\frac{\tau_{m_h}}{h}}), \quad (27)\\ \Upsilon &= \operatorname{diag}(\frac{1}{1+4\frac{\tau_1}{h}}, \frac{1}{1+4\frac{\tau_2}{h}}, \dots, \frac{1}{1+4\frac{\tau_{m_h}}{h}}),\end{aligned}$$

Starting from the discretization provided in (12), and using the linear approximation of the constraints

$$\frac{1}{4}(g_{k+1} + 2g_k + g_{k-1}) \approx g_k + \frac{h}{4}G_k v_{k+1} - \frac{h}{4}G_k v_k, \quad (28)$$

the stepping equations become

$$\begin{bmatrix} M & -A_k^T & -G_k^T \\ A_k & \Gamma & 0 \\ G_k & 0 & \Sigma \end{bmatrix}\begin{bmatrix} v_{k+1} \\ \alpha \\ \lambda \end{bmatrix} = \begin{bmatrix} r \\ s \\ t \end{bmatrix},$$

$$\text{where} \quad \begin{bmatrix} r \\ s \\ t \end{bmatrix} = \begin{bmatrix} Mv_k - h\nabla V_k + hf^{(+)}(k-1, k) \\ w_k \\ -\frac{4}{h}\Upsilon g_k + \Upsilon G_k v_k \end{bmatrix}. \quad (29)$$

Note that with suitably chosen perturbation parameters, the system matrix is never singular which means that for this formulation, constraint degeneracy is not an issue. In addition, it is now possible to model all stiff forces as "relaxed" constraints. In that respect, potential forces lead naturally to relaxed holonomic constraints, and polygenic forces lead to relaxed nonholonomic constraints.

The case of three dimensional rigid bodies is special since then, the mass matrix $M(q)$ is configuration dependent. It is possible to produce corrections to the main Spook stepper defined in (29) using the variational formulations. When this is done, matrix M is replaced by a time dependent $\tilde{M}_k = M_k + O(h^2)$, and the forcing terms are also modified to account for the gyroscopic forces. This is discussed further in my thesis [18] and a complete specification is part of future work.

The matrix appearing on the left hand side of (29) is positive definite but non-symmetric, and it can easily be transformed to a symmetric but indefinite matrix of the type processed efficiently by well-known sparse matrix packages, such as Umfpack [9], for instance. Experience shows that it is a better idea to use Umfpack directly on the large matrix instead of forming the symmetric, positive definite Schur complement matrix

$$\begin{bmatrix} A & G \end{bmatrix} M^{-1}\begin{bmatrix} A^T \\ G^T \end{bmatrix} + \begin{bmatrix} \Gamma & 0 \\ 0 & \Sigma \end{bmatrix}, \quad (30)$$

since the latter can get quite dense in comparison to the original matrix.

5. Linear stability

Several schemes have been suggested for constraint stabilization in mechanical systems [5, 3, 2]. In addition to these, there are regularization schemes for treating the differential-algebraic equations of constrained multibody systems [14]. Though the simplest and most common stabilization scheme [5] does work in some cases, there is no optimal choice of stabilization parameters [2]. Other stabilization schemes [3, 2] are stable but are not motivated from physics and require additional processing. The standard regularization method for differential algebraic equations [14] amounts to replacing constraint equations $g(q) = 0$ with $\epsilon\dot{\lambda} + g(q) = 0$, which is non physical, or replacing (14) with $-\epsilon/2\|\dot{\lambda}\|^2 + \lambda^T g(q)$, leading to $\epsilon\ddot{\lambda} + g(q) = 0$, and thus adding unnecessary second order dynamics to the system. This is clearly not justified from the point of view of constraint realization theory.

The regularization and stabilization scheme of § 3 is motivated by physics however and does reduce to known methods in the limit where the parameters ϵ and τ vanish. It remains to show that the new stepping scheme is indeed stable.

To this end, consider a mechanical system with constant, real, symmetric, and positive definite mass matrix M, of dimension $n \times n$. Set the potential to $V(q) = 0$. Add constant linear homogeneous holonomic constraints of the form $g(q) = Gq = 0$, where G is an $m \times n$ real matrix. In reference to (26), redefine λ to absorb one factor of h, and introduce the nonnegative parameters $\theta = \tau/h$, $\psi = 1/(1+4\theta)$, and $\sigma = (4/h^2)\psi\epsilon$. Discretizing the basic Lagrangian as in (12), and introducing the discrete velocity $p_k = (1/h)(q_k - q_{k-1})$, the equations of motion now read

$$\begin{aligned} q_{k+1} - hp_{k+1} &= q_k \\ Mp_{k+1} - G^T\lambda &= Mp_k \\ \frac{h}{4}\psi^{-1}\sigma\lambda + \frac{1}{4}G(q_{k+1} + 2q_k + q_{k-1}) & \\ + h\theta Gp_{k+1} &= 0. \quad (31)\end{aligned}$$

After defining $x_k = Gq_k$, $y_k = hGp_k$, $S = GM^{-1}G^T$ and $S_\sigma = S + \sigma I$, where I is the identity matrix of suitable dimension, eliminating λ from the system (31) yields the reduced stepping equations

$$\begin{aligned} x_{k+1} - y_{k+1} &= x_k \\ y_{k+1} &= -4\psi S S_\sigma^{-1} x_k + (I - 4\theta\psi S S_\sigma^{-1}) y_k. \end{aligned} \tag{32}$$

Now, it is possible to use the polar decomposition to produce a factorization $G = URP^T$ so that U is an $m \times m$ orthonormal matrix, P is a $n \times n$ permutation matrix, and R is a $m \times n$ matrix with the same rank as G. With this factorization, $(U^T Gq)^T = (u_1, u_2, \ldots, u_r, 0, \ldots, 0)$. In addition, since $S = GM^{-1}G^T$ is symmetric and positive semidefinite, then, so is $\tilde{S} = U^T S U$ and so, there is an orthonormal matrix V, with $VV^T = I$, such that $V\tilde{S}V^T = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_r, 0, 0, \ldots)$. Finally, since matrices S and S_σ differ by a scalar multiple of the identity, they commute with each other and are simultaneously diagonalized by the orthogonal transform UV. Thus, changing variables in (32) to $u = UVx$ and $v = UVy$, the stepping equations become

$$\begin{aligned} & Bz_{k+1} = Az_k, \quad z = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} UV & 0 \\ 0 & UV \end{bmatrix} \\ & B = \begin{bmatrix} I & -I \\ I & 0 \end{bmatrix}, \quad A = \begin{bmatrix} I & 0 \\ -4\psi\Gamma & (I - 4\theta\psi\Gamma) \end{bmatrix} \\ & \Gamma = \operatorname{diag}(\frac{\lambda_1}{\lambda_1 + \sigma}, \frac{\lambda_2}{\lambda_2 + \sigma}, \ldots, \frac{\lambda_r}{\lambda_r + \sigma}, 0, \ldots, 0). \end{aligned} \tag{33}$$

The stationary recurrence defined in (33) can be written as $z_{k+1} = B^{-1}Az_k$ and this is stable provided all the eigenvalues of the matrix $B^{-1}A$ are within the unit circle in the complex plane. Because of the block diagonal structure of matrices A and B, the $2m$ eigenvalues of $B^{-1}A$ come in pairs, $\mu_\pm^{(i)}$, $i = 1, 2, \ldots, m$. Straightforward computations yield

$$\begin{aligned} \mu_\pm^{(i)} &= 1 - 2\gamma^{(i)}\psi(\theta+1) \pm 2\sqrt{(\gamma^{(i)}\psi(1+\theta))^2 - \gamma^{(i)}\psi}, \\ & \quad \text{for} \quad i = 1, 2, \ldots, r, \quad \text{and} \\ \mu_\pm^{(i)} &= 1, \quad \text{for} \quad i = r+1, \ldots, m. \end{aligned} \tag{34}$$

Simple analysis yields

$$\begin{aligned} |\mu_\pm^{(i)}| &< 1, \quad \text{when } i < r, \, , \quad \text{and} \\ \mu_\pm^{(i)} &= 1, \quad \text{when } i > r, \, \epsilon > 0, \text{ and } \theta > 0. \end{aligned} \tag{35}$$

With the choice of coordinates, $u_k^{(i)} = v_k^{(i)} = 0$ for $i > r$ and $k = 0, 1, 2, \ldots$. Therefore, $\|x_k\| = \|Gq_k\| \to 0$ and $\|y_k\| = \|hGp_k\| \to 0$ as $k \to \infty$ and so the linear homogeneous system is stable. We have thus proved the following.

Theorem 5.1. *For a discrete mechanical system with configuration space $Q = \mathbb{R}^n$, constant $n \times n$ mass matrix M that is real, symmetric and positive definite, subjected to m-dimensional linear homogeneous constraints $g(q) = Gq$, where matrix G is an $m \times n$ matrix of rank $r \le m$, the regularized and stabilized stepping scheme of (26) is stable provided $\epsilon > 0$ and $\tau > 0$.*

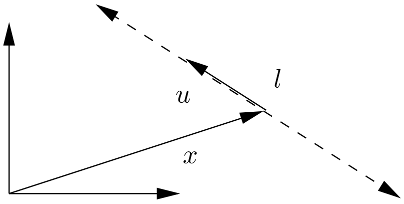


Figure 1: A two dimensional rigid rod.

Theorem 5.1 means in particular that the linearized stepping scheme can process degenerate constraints. There is strong numerical evidence that linear stability is preserved when the perturbation parameters ϵ and τ in (26) take different values for each individual constraint. However, analysis has not yielded strong enough bounds on the spectrum of the corresponding stepping matrix $B^{-1}A$ of (33) yet. Likewise, there are numerical indications that nonlinear stability is not guaranteed for cases where the constraint Jacobian varies quickly, i.e., when the curvature of the constraints is high.

6. Numerical experiments

The framework described above requires additional elements to correctly process three dimensional rigid bodies with non-homogeneous inertia tensors. To avoid these issues for the moment, consider the motion of a rigid rod as shown in Fig. 1. The coordinates for this consist of Cartesian coordinates for the center of mass $x \in \mathbb{R}^n$, and a unit vector $u \in \mathbb{R}^n$, with the constraint that $\|u\| = 1$. If the rod has length l and mass m, the inertia tensor is diagonal with value m for the first n entries, and $l^2/12$ for the remaining ones. Interconnected systems of these rods can be easily constructed in a scripting language such as Octave [10] or Matlab$^{\text{TM}}$, making it easy to use UMFPACK to process the linear algebra efficiently. The example considered here is that of a ladder of which one corner is attached to the origin, as shown in Fig. 2. This configuration has appeared in the literature previously [4] to test a strategy for constraint stabilization. This is provides an interesting test case for SPOOK. With the labelling provided in Fig. 2, the connections between the rods is as follows. Rods 1 and 2 have their tails fixed at the origin. Then, rod j is connected by its tail to the head of rod $j-3$ when $\mod(j, 3) = 2$, and to rod $j-2$ otherwise. Finally, rod j is connected head to head with rod $j+1$ for $j = 3k, k = 1, 2, \ldots, N$. If the total number of links is N, there are $n = 4 + 3(N-1)$ rigid rods in the system, $4N + 1$ connection constraints between

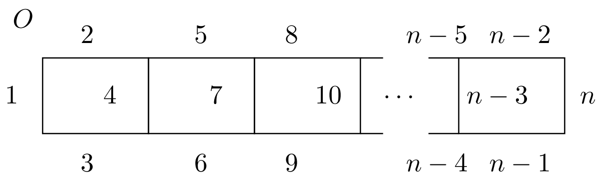


Figure 2: A square chain of rigid rods.

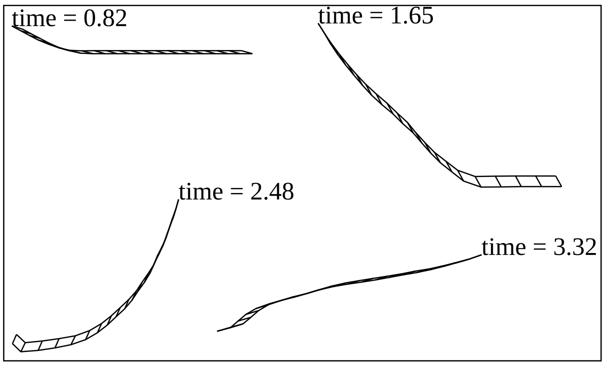


Figure 3: Animation of a rod with $N = 20$.

pairs of rods or between a rod and the origin, and n unit constraints to keep $\|u_i\| = 1$. Among the $4N + 1$ constraints, the number of closed kinematic loops is N. For the illustrations that follow, all links have mass 10 and length 1. Other runs were performed with variable masses and link lengths and the stability results were similar to what is reported here, as long as the mass ratios are less than 1000, approximately. Beyond that point, stability problems do surface. The target time step is kept at $h = 1/60$ in most cases, except when analyzing error convergence as in Fig. 6. This is because interactive applications that drive 3D graphics display are often tuned to that frequency. The stepper was found to be stable on this problem for h as large as $h = 1/20$ for $N = 20$.

The other parameters chosen here for the simulations are regularization $\epsilon = 10^{-8}$, and constraint stabilization $\tau/h = 2$. Experience with Spook stepper indicates that these parameters are good defaults for a wide range of problems, and rarely need to be tuned. Of course, when masses become very large and their inverses become comparable to ϵ, the regularization parameter must be reduced. A few steps of a simulation are shown in Fig. 3 where $N = 20$. This is a moderate problem size involving 427 variables in two dimensions.

An interesting aspect of the present problem is that as the ladder falls, whiplash develops at the end links after a few oscillations and some velocities become very large. Other fixed time step integration strategies such as Runge-Kutta fourth order with Baumgarte stabilization or plain Shake break down completely when whiplash occurs. As explained before, variable time-step is not convenient for real-time simulations and robustness against short burst of high velocity is therefore desirable. In fact, the plain linear version of Spook using (29) works best. Performing Newton-Raphson iterations does not improve the solution significantly in most cases, and when the velocities are too big, e.g., during whiplash, the "refined" solution was found to be worse than the linear approximation. This was evident from a sudden, large increase in energy.

The idea of using the ladder problem is to check that Spook can process this problem at fixed step, using a relatively large time step, but without loosing stability, even though local errors might be significant. In addition, many methods for handling loop closures in mechanical systems have complexity $O(m^2)$ where m is the number of loops. This is in fact what was reported of Mexxax [21] in previous work [4]. In that last article, the regularization technique was found to have linear complexity in both the total number of constraints and in the number of closed loop constraints. The same result is achieved here as seen from Fig. 7, in which the time taken by linear system solver is plotted as a function of system size. Thanks to the quality of Umfpack, this is linear as it should.

The energy does dissipate in this problem as seen in Fig. 4 for a chain of $N = 10$ links and another with $N = 100$ links. The reason for that is that constraint violations are damped. This damping is most severe when the chain passes the vertical position and gravity forces stretches all vertical links. However, constraint violation is kept within tight bounds as shown in Fig. 5, even for large system. It is true that for the ladder problem, with the formulation given above, most constraints are in fact linear, except for the unit length ones. On problems with predominantly nonlinear constraints, such as the simple pendulum for instance, the errors can be significantly larger.

Overall, the quality of the solution delivered by Spook is comparable to what was observed with other integrators. The numerical cost is less, however, since only one linear system is solved at each step. Stability is also good enough to use large constant step size. The error made on the constraint violation appears to be $O(h^2)$ as seen from Fig. 6.

7. Conclusion

The discrete variational principle provides excellent numerical integrator formulae for physical systems which globally preserve the fundamental symmetries. In addition, the technique for constructing new integration formulae is straight forward and *additive*, making it possible to mix and match different discretizations for different terms. In particular, separating fast oscillatory forces and strong damping forces from the rest of the physics is easy to achieve as described in § 3.

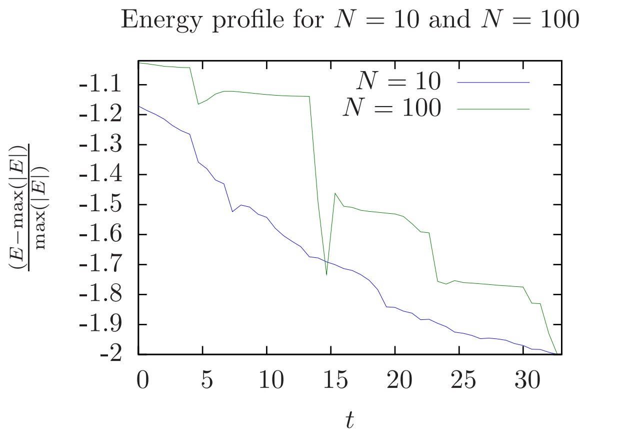


Figure 4: The energy as a function of time for $N = 10$ and $N = 100$.

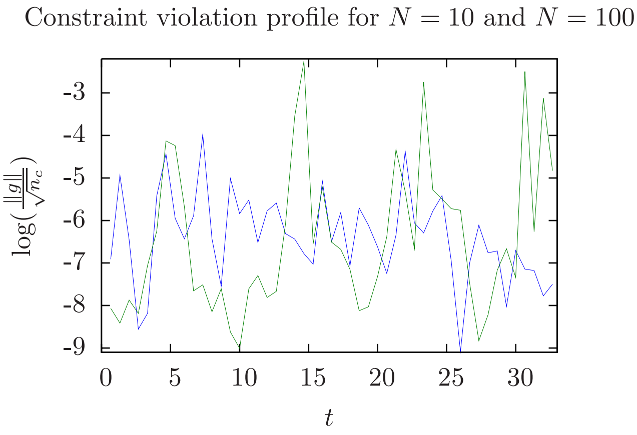


Figure 5: The root mean square norm of constraint violations for $N = 10$ and $N = 100$.

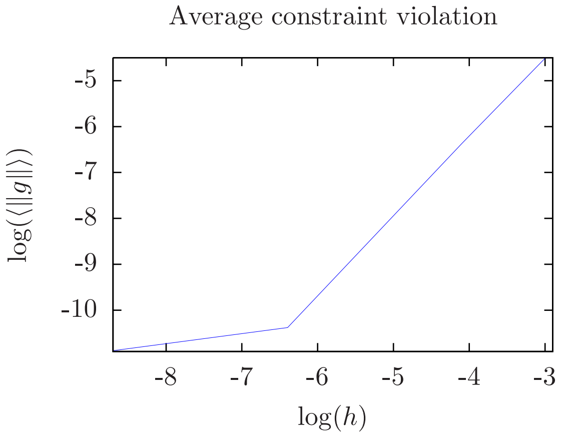


Figure 6: Average constraint violation $\langle\|g\|\rangle$ as function of time step size h.

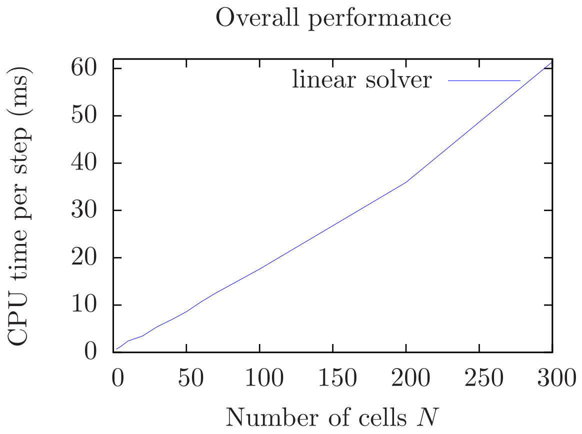


Figure 7: Overall performance of the stepper as a function of N. Average CPU time per step, in ms.

This is exactly what was done in the construction of SPOOK and it is very satisfying to see that indeed, it can integrate constrained mechanical systems reliably with very little tuning, as illustrated in § 6. This is not a general result for differential algebraic equations, however, since the constraint regularization technique presented here is limited strictly to the algebraic conditions appearing in mechanical systems.

At this time, SPOOK has not been implemented within a generic multibody dynamics library and it remains to be seen how robust it is on problems with strongly nonlinear constraints. Also not included in the present article are the subtle modifications required to treat the gyroscopic forces of generic three dimensional rigid bodies. Future work will address these issues.

8. Acknowledgements

This research was conducted using the resources of High Performance Computing Center North (HPC2N), and supported in part by the "Objective 1 Norra Norrlands" EU grant VISTA awarded to VRlab/HPC2N at Umeå University, and and by the *Swedish Foundation for Strategic Research* (SSF) under the frame program grant SSF-A3 02:128, and by Vinnova/SSF grant VINST#247.

References

[1] H. C. ANDERSEN, *RATTLE: A velocity version of the SHAKE algorithm for molecular dynamics calculations*, J. Comp. Phys., 52 (1983), pp. 24–34.

[2] U. ASCHER, H. CHIN, L. PETZOLD, AND S. REICH, *Stabilization of constrained mechanical sys-*

tems with DAEs and invariant manifolds, Mech. Struct. & Mach., 23 (1995), pp. 135–158.

[3] U. Ascher and P. Lin, *Sequential regularization methods for nonlinear higher index DAEs*, SIAM J. Scient. Comput, 18 (1997), pp. 160–181.

[4] ——, *Sequential regularization methods for simulating mechanical systems with many closed loops.*, SIAM J. Scient. Comput., 21 (1999), pp. 1244–1262.

[5] J. Baumgarte, *Stabilization of constraints and integrals of motion in dynamical systems*, Computer Methods in Applied Mechanics and Engineering, 1 (1972), pp. 1–16.

[6] F. A. Bornemann, *Homogenization in Time of Singularly Perturbed Mechanical Systems*, vol. 1687 of Lecture notes in mathematics, 0075-8434, Springer, Berlin, 1998.

[7] V. N. Brendelev, *On the realization of constraints in nonholonomic mechanics*, J. Appl. Math. Mech., 45 (1981), pp. 481–487.

[8] J. Cortés and S. Martínez, *Non-holonomic integrators*, Nonlinearity, 14 (2001), pp. 1365–1392.

[9] T. A. Davis, *Algorithm 832: UMFPACK — an unsymmetric-pattern multifrontal method*, ACM Transactions on Mathematical Software, 30 (2004), pp. 196–199.

[10] J. W. Eaton, *GNU Octave Manual*, 2.9.12 ed., 2007.

[11] R. E. Gillilan and K. R. Wilson, *Shadowing, rare events, and rubber bands - a variational Verlet algorithm for molecular-dynamics*, J. Chem. Phys., 97 (1992), pp. 1757–1772.

[12] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun, *Efficient simulation of inextensible cloth*, in SIGGRAPH '07: ACM SIGGRAPH 2007 papers, New York, NY, USA, 2007, ACM Press, p. 49.

[13] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, vol. 31 of Spring Series in Computational Mathematics, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong, 2001.

[14] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential Algebraic Problems*, vol. 14 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong, second revised edition ed., 1996.

[15] C. Kane, J. E. Marsden, M. Ortiz, and M. West, *Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems*, International Journal for Numerical Methods in Engineering, 49 (2000), pp. 1295–1325.

[16] A. V. Karapetian, *On realizing nonholonomic constraints by viscous friction forces and Celtic stones stability*, J. Appl. Math. Mech., 45 (1981), pp. 42–51.

[17] A. J. Kurdila and F. J. Narcowich, *Sufficient conditions for penalty formulation methods in analytical dynamics*, Computational Mechanics, 12 (1993), pp. 81–96.

[18] C. Lacoursière, *Ghosts and Machines: Regularized Variational Methods for Interactive Simulations of Multibodies with Dry Frictional Contacts*, PhD thesis, Department of Computing Science, Umeå University, Sweden, SE-901 87, Umeå, Sweden, June 2007.

[19] C. Lanczos, *The Variational Pinciples of Mechanics*, Dover Publications, New York, fourth ed., 1986.

[20] B. Leimkuhler and S. Reich, *Simulating Hamiltonian Dynamics*, vol. 14 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2004.

[21] C. Lubich, C. Engstler, U. Nowak, and U. Pöhle, *Numerical integration of constrained mechanical systems using MEXX*, Mech. Struct. & Mach., 23 (1995), pp. 473–497.

[22] J. E. Marsden and M. West, *Discrete mechanics and variational integrators*, Acta Numer., 10 (2001), pp. 357–514.

[23] H. Rubin and P. Ungar, *Motion under a strong constraining force*, Communications on Pure and Applied Mathematics, X (1957), pp. 65–87.

[24] J.-P. Ryckaert, G. Ciccotti, and H. J. C. Bendersen, *Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes*, J. Comp. Phys., 23 (1977), pp. 327–341. shake.

[25] L. Verlet, *Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules.*, Phys. Rev., 159 (1967), pp. 98–103.

Semiexplicit numerical integration by splitting with application to dynamic multibody problems with contacts

Klas Modin and Dag Fritzson

SKF Engineering Research Centre
MDC, RKs–2
SE–415 50 Göteborg, Sweden
E-mail: Klas.Modin@na.lu.se, Dag.Fritzson@skf.com

Claus Führer

Centre for Mathematical Sciences
Box 118, SE–221 00 Lund, Sweden
E-mail: claus@maths.lth.se

December 4, 2007

Numerical integration is considered for second order differential equations on the form

$$\ddot{q} = A(q, \dot{q}, t) + B(q, \dot{q}, t),$$

where A is significantly more expensive to evaluate than B, and B is stiff (highly oscillatory) in comparison with A. Examples of such problem are multibody problem with contact forces acting between bodies, and constraints formulated as penalty forces.

Based on the splitting $A + B$ of the acceleration field, a numerical integration algorithm, which is explicit in the A–part and implicit in the B–part, is suggested. Consistency and linear stability analysis of the proposed method is carried out.

Numerical examples with the proposed method is carried out for two simple test problems, and for a complex multibody model of a rotating ball bearing. Comparison with conventional implicit methods is given for each example. The results indicate that the proposed method is more efficient, in terms of number of evaluations of A, at the same accuracy level.

Contents

1. Introduction

We are interested in the numerical time integration of dynamic multibody systems. In particular, problems where bodies frequently come in contact with each other, which is modelled by complex force laws. The application we have in mind is simulation of rolling bearings, where contacts between bodies are present. Our aim is to design a numerical integrator that is particularly efficient for such problems.

In Section 2, a characterization of the governing equations is given. The objective is twofold: (i) to pin-point where the computational cost is high; (ii) to estimate typical frequencies in the solution. The specific character of the governing equations is then used as a basis in Section 3 in order to design a more efficient integrator. Further, in Section 4 the proposed integrator is analyzed in terms of consistency and linear stability. In Section 5 we discuss issues concerned with adaptive time-stepping. Lastly, in Section 6, we give numerical test examples, both for simple test problems and for a fully complex rolling bearing problem.

2. Characterization of the governing equations

In this section we give an overview of the formulation, and a characterization, of the governing differential equations that are to be numerically integrated. There are, of course, a number of choices on how to formulate the equations of motion for multibody systems. Primarily, the so called "floating frame of reference" is what we have in mind. In particular, the formulation used in the multibody simulation software BEAST, which is a tool for detailed transient analysis of rolling bearings and other machine elements, developed and maintained by SKF (www.skf.com). For a full specification of the governing equations, see [Nak06]. In this paper we consider only issues that are essential for the proposed integrator.

Let $\mathscr{Q} = \mathbb{R}^d$ be the configuration space of the multibody system. Further, let $q \in \mathscr{Q}$ denote the position coordinates. For rigid systems these are the centre of mass and the orientation of each body, relative to a fixed global coordinate system in Euclidean 3–space. In the case of elasiticity, generalized coordinates describing the deflection field of each body are also included in q. The governing equations are on the form

$$M(q,\dot{q})\ddot{q} = F(q,\dot{q},t)\,, \qquad q(0) = q_0\,, \qquad \dot{q}(0) = \dot{q}_0\,, \tag{1}$$

where the mass matrix $M(q,\dot{q})$ is a symmetric positive definite $d \times d$–matrix, and the force field $F(q,\dot{q},t)$ is a vector valued map corresponding to the forces acting on the system. Time is denoted t. Notice that (1) is a second order ordinary differential equation (ODE).

Constraints are taken into account by penalty force laws. Thus, we do not utilize the standard Lagrangian multiplier formulation, which is typically used for constrained mechanical systems. Notice that the penalty formulation implies that (1) is highly stiff. Thus, an implicit numerical integrator must be used.

Remark. The integrator suggested in this paper could easily be extended to governing equations formulated as a differential algebraic system using Lagrangian multipliers.

2.1. Computational costs

Solving (1) numerically with a standard ODE solver basically involves the following computations in order to evaluate the vector field:

- Compute the inverse of the mass matrix. This is a cheap operation, as the mass matrix is block diagonal (one block per body).
- Compute the constraint forces, which is also cheap.
- Compute "simple" non-stiff forces such as gravity and Coriolis forces.
- Compute contact forces. This task is heavily dominating the computational cost. Evaluation of each contact involves searching for intersecting surfaces. In the case of rolling bearings, the surface geometries of the bodies are highly complex. Further, for very detailed contact models (as in BEAST), tribological issues, such as oil film thickness in the contact, are also taken into account. See [SF01] for details on advanced contact modelling.

In order to be able to separate the computationally intensive force evaluations from less costly evaluations, we rewrite (1) as

$$M(q,\dot{q})\ddot{q} = F^A(q,\dot{q},t) + F^B(q,\dot{q},t)\,, \tag{2}$$

where F^A are the contact forces plus "simple" non-stiff forces (from now on we refer to F^A as contact forces), and F^B are the constraint penalty forces. Multiplying from the left with the inverse of the mass matrix we get a second order ODE written on standard form

$$\ddot{q} = A(q,\dot{q},t) + B(q,\dot{q},t) \quad \text{or shorter} \quad \ddot{q} = C(q,\dot{q},t)\,. \tag{3}$$

The maps A and B correspond to accelerations due to contact forces and constraint forces respectively. $C = A + B$ is the total acceleration field. The field A is much more expensive to compute than B. This "splitting formulation" of the governing equations will be utilized by the integrator algorithm described in Section 3.

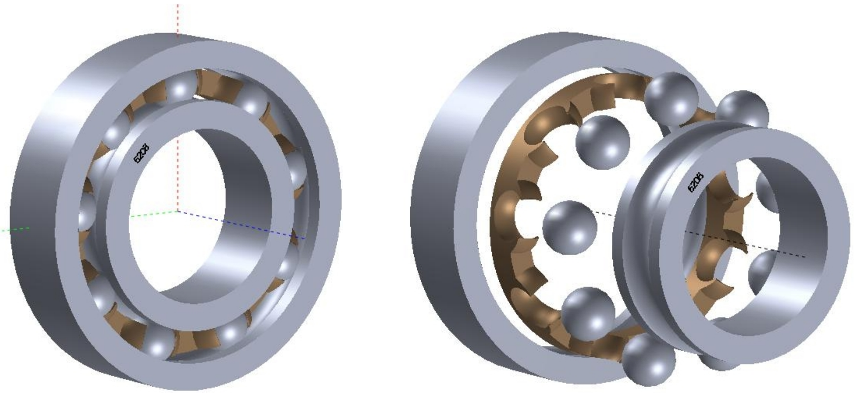


Figure 1: A common ball bearing. To the right in exploded view. The bodies in the model are: (1) an outer ring; (2) a cage; (3) nine balls; (4) an inner ring. All elements are steel, except the cage which is plastic.

Notice that (3) is a non-autonomous differential equation, i.e., the right hand side depends explicitly on time t. From now on we rewrite the governing equations as an autonomous first order system, evolving on the extended phase space $\mathscr{P} = T\mathscr{Q} \times \mathbb{R} = \mathbb{R}^{2d+1}$, equiped with coordinates $\boldsymbol{z} = (\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$. That is, we write (3) as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \\ t \end{pmatrix} = \begin{pmatrix} \dot{\boldsymbol{q}} \\ A(\boldsymbol{q},\dot{\boldsymbol{q}},t) + B(\boldsymbol{q},\dot{\boldsymbol{q}},t) \\ 1 \end{pmatrix} \quad \text{or shorter} \quad \dot{\boldsymbol{z}} = X(\boldsymbol{z})\,. \tag{4}$$

The phase flow corresponding to (4) is denoted φ^h. That is, φ^h is a map $\mathscr{P} \to \mathscr{P}$, depending on the time step length h, such that $\varphi^h(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), t) = (\boldsymbol{q}(t+h), \dot{\boldsymbol{q}}(t+h), t+h)$.

2.2. Estimated frequencies

Our next objective is to estimate the frequencies, or time scales, in the system due to A and B. In order to do so we consider a simplified model of a ball bearing, see Figure 1.

We begin with typical frequencies due to contact forces, i.e., due to A. The stiffness in a steel–steel contact (e.g. between a ball and the outer ring) is typically about $k = 10^8\,\mathrm{N/m}$. The damping and friction forces are small, so we neglect them in this simple analysis. The mass of a ball is about $m = 0.01\,\mathrm{kg}$. Thus, a typical translational frequency is about $(\sqrt{k/m})/(2\pi) \approx 1.6 \cdot 10^4\,\mathrm{Hz}$. The smallest moment of inertia of the outer ring is about $J = 3 \cdot 10^{-5}\,\mathrm{kg \cdot m^2}$ and its radius about $3 \cdot 10^{-2}\,\mathrm{m}$. This gives a typical rotational frequency of about $5 \cdot 10^4\,\mathrm{Hz}$. From experience with the BEAST software, using a standard implicit ODE solver, it is known that the time step length, for bearings like in Figure 1, typically is about $10^{-6}\,\mathrm{s}$. This time step is small enough to fully capture the dynamics of A.

In order to get an accurate result, the stiffness (and damping) of the constraint forces must be set much higher than the stiffness of the “real physical” forces. From our point of view, it means that the time scales of B must be much smaller than those of A. For the bearing case described it is suitable to choose the stiffness and damping of the penalty forces so that the frequencies of B are about $10^7\,\mathrm{Hz}$. As the time step $10^{-6}\,\mathrm{s}$ is not small enough to resolve such frequencies, it is essential to use an implicit integrator.

3. Integrator algorithm

From the previous section it is clear that the governing equations are stiff due to constraint forces, so for the time step lengths we have in mind an implicit integrator is needed in order to achieve stability. From an efficiency point of view, an implicit integrator is, per time step, more expensive than an explicit dito, because at each step a non–linear root finding problem needs to be solved iteratively. In particular, this implies that A and B are evaluated several times per time step. Furthermore, the Jacobian needs to be computed for an implicit method (which is severely expensive for A in our case), as it is used by the root finding algorithm (typically some variant of Newton's method).

The frequency analysis of the previous section indicates that the dynamics of A can be resolved accuratelly with an explicit integrator for the time step lengths we have in mind. However, it is not so for B. One possibility is of course to choose a much smaller time step, but that would be inefficient, as to many time steps are needed.

Our approach is to treat A explicitly and B implicitly. The notion is that B is "responsible" for the stiff character of the system, so in order to have stability for long time steps it is enough if only this part is handled implicitly. The idea is that the number of evaluations of A should be on par with that of explicit methods, whereas B can be evaluted more frequently without any significant increase in the computational cost.

An obvious possibility is to use so called splitting methods (see [MQ02]). That is, to consider an explicit method Φ_h^A for the equation $\ddot{\boldsymbol{q}} = A(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$ and an implicit method Φ_h^B for $\ddot{\boldsymbol{q}} = B(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$, and then utilized a composition of the two methods, e.g. $\Phi_h^A \circ \Phi_h^B$ or $\Phi_{h/2}^B \circ \Phi_h^A \circ \Phi_{h/2}^B$. However, this approach is not so good in our case. Indeed, as Φ_h^A does not take constraint forces into account, the solution would at each step drift away a little from the constraint manifold, and then be "forced back" towards it by Φ_h^B. As Φ_h^B does not exactly project onto the constraint manifold, high frequency $\mathcal{O}(h)$ oscillations would thus appear in the solution.

We suggest the following discretization of the governing equations (4)

$$\begin{aligned} \boldsymbol{q}_{n+1} &= \boldsymbol{q}_n + h\dot{\boldsymbol{q}}_n + \frac{h^2}{2}\Big(A_{n+\alpha} + B_{n+\beta}\Big) \\ \dot{\boldsymbol{q}}_{n+1} &= \dot{\boldsymbol{q}}_n + h\Big(A_{n+\alpha} + B_{n+\beta}\Big) \\ t_{n+1} &= t_n + h\,, \end{aligned} \tag{5}$$

where $\alpha, \beta \in [0,1]$ are method parameters and

$$\begin{aligned} A_{n+\alpha} &= A(\boldsymbol{q}_n + h\alpha\dot{\boldsymbol{q}}_n, \dot{\boldsymbol{q}}_n, t_n + \alpha h) \\ B_{n+\beta} &= B\big((1-\beta)\boldsymbol{z}_n + \beta\boldsymbol{z}_{n+1}\big)\,, \quad \boldsymbol{z}_n = (\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n, t_n)\,. \end{aligned}$$

Notice that $A_{n+\alpha}$ is independent of $\boldsymbol{q}_{n+1}$ and $\dot{\boldsymbol{q}}_{n+1}$ for all α. Thus, it only needs to be evaluated once per time step, and the Jacobian of A is not needed. This is what we mean by "treating A explicitly".

The method (5) is a blend of the Störmer–Verlet method, the explicit Euler method, and the implicit midpoint rule. The correspondence is as follows.

Condition	**Method**
$B = 0$, $\alpha = 1/2$, and A independent of $\dot{\boldsymbol{q}}$	Störmer–Verlet
$B = 0$ and A independent of $\boldsymbol{q}$	explicit Euler (for the $\dot{\boldsymbol{q}}$ part)
$A = 0$ and $\beta = 1/2$	implicit midpoint rule

4. Consistency and linear stability analysis

In this section we analyse the order of consistency and the linear stability of the proposed method (5). We begin with the consistency analysis. Next, the stability analysis, which is split into two parts. First the explicit component of the algorithm is analysed separately, i.e. the case $B = 0$. Based on these results, stability of the full algorithm, with constraint forces present, is thereafter analysed.

4.1. Order of consistency

Let $\Phi_h : \mathscr{P} \to \mathscr{P}$ be the numerical flow of (5), i.e., the map defined by

$$\Phi_h(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n, t_n) = (\boldsymbol{q}_{n+1}, \dot{\boldsymbol{q}}_{n+1}, t_{n+1}) \,.$$

The local error map is given by $E_h = \Phi_h - \varphi^h$. Consistency of Φ_h with respect to φ^h means that $E_h(\boldsymbol{z}) = \mathcal{O}(h^2)$ as $h \to 0$ for all $\boldsymbol{z} \in \mathscr{P}$. Concering the principle term of E_h we have the following result:

Lemma 1. *The $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and t components of the local error map E_h fulfills*

$$E_h(\boldsymbol{z}) = \begin{pmatrix} \frac{1}{6}h^3\Big((3\alpha-1)\big(\partial_{\boldsymbol{q}}A(\boldsymbol{z})\dot{\boldsymbol{q}} + \partial_t A(\boldsymbol{z})\big) - \partial_{\dot{\boldsymbol{q}}}A(\boldsymbol{z})C(\boldsymbol{z}) + (3\beta-1)\partial_{\boldsymbol{z}}B(\boldsymbol{z})X(\boldsymbol{z})\Big) + \mathcal{O}(h^4) \\ \frac{1}{2}h^2\Big((2\alpha-1)\big(\partial_{\boldsymbol{q}}A(\boldsymbol{z})\dot{\boldsymbol{q}} + \partial_t A(\boldsymbol{z})\big) - \partial_{\dot{\boldsymbol{q}}}A(\boldsymbol{z})C(\boldsymbol{z}) + (2\beta-1)\partial_{\boldsymbol{z}}B(\boldsymbol{z})X(\boldsymbol{z})\Big) + \mathcal{O}(h^3) \\ 0 \end{pmatrix} .$$

Proof. Compare Taylor expansions in h of $\Phi_h(\boldsymbol{z})$ and $\varphi^h(\boldsymbol{z})$. □

Using the lemma we immediatelly obtain the following result:

Theorem 2. *The method Φ_h, defined by* (5)*, has the following order of consistency properties.*

- *It is consistent for all α, β.*
- *It is second order accurate in position variables $\boldsymbol{q}$ for all α, β.*
- *If $\alpha = \beta = 1/2$ and A is independent of $\dot{\boldsymbol{q}}$, then it is second order accurate in all variables.*

Remark. In our application, the friction and damping in the contacts are very small in comparison to the stiffness. That is, $||\partial_{\dot{q}}A(\boldsymbol{z})||$ is small in comparison to $||\partial_{q}A(\boldsymbol{z})||$. Thus, we have "almost" second order accuracy for $\alpha = \beta = 1/2$.

4.2. Stability analysis when $B = 0$

In absence of constraint forces, i.e., when $B = 0$, the method Φ_h is fully explicit. Thus, it has a bounded stability region in terms of the step size h. We carry out a linear stability analysis for the scalar test equation given by

$$\ddot{q} = A(q, \dot{q}) = -kq - c\dot{q} \,, \qquad k, c \in \mathbb{R} \,. \tag{6}$$

Recall that stability of the method Φ_h means that $\lim_{n\to\infty} \Phi_h^n(\boldsymbol{z})$ is bounded. The result is as follows.

Theorem 3. *The stability region of the method* (5) *applied to* (6) *(with $B = 0$) is given by*

$$\Omega = \Big\{(hc, h^2k) \in \mathbb{R}^2 \setminus \{(0,0), (2-4\alpha, 4)\}\,;\, h^2k \geq 0, hc \leq 2 - \alpha h^2 k, hc \geq (1/2 - \alpha)h^2 k\Big\} \,. \tag{7}$$

See Figure 2 for an illustration.

Proof. The numerical flow map Φ_h is linear for the test equation (6), i.e., it can be written

$$\Phi_h(q,\dot{q}) = R\begin{pmatrix} q \\ \dot{q} \end{pmatrix}, \qquad \text{with} \quad R = R(h,k,c) \in \mathbb{R}^{2\times 2}. \tag{8}$$

Thus, $\lim_{n\to\infty} \Phi_h^n(q,\dot{q}) = \lim_{n\to\infty} R^n \begin{pmatrix} q \\ \dot{q} \end{pmatrix}$. This expression is bounded for all $(q,\dot{q})$ if and only if the eigenvalues of R satisfy the root condition, i.e., they lie on or within the unit circle and if on the unit circle they are simple. Written out, the characteristic equation $\det(R - \mathrm{Id}\lambda) = 0$ is

$$1 - hc + \frac{1}{2}h^2k(1-2\alpha) - \big(2 - hc - \frac{1}{2}h^2k(1+2\alpha)\big)\lambda + \lambda^2 = 0. \tag{9}$$

The solutions, i.e., the two eigenvalues, are

$$\lambda_\pm = 1 - \frac{hc}{2} - \frac{h^2k(1+2\alpha)}{4} \pm \frac{1}{4}\sqrt{\left(2hc + h^2k(1+2\alpha)\right)^2 - 16h^2k} \quad . \tag{10}$$

The eigenvalues are equal (non-simple) and lie on the unit circle for $(hc, h^2k) \subset \{(0,0),(2-4\alpha,4)\}$. They are unequal (simple) and and one of them lie on the unit circle at $\partial\overline{\Omega} \setminus \{(0,0),(2-4\alpha,4)\}$. Furthermore, inside Ω the are both strictly inside the unit circle, and outside Ω at least one of them is strictly outside the unit circle. □

From (10) in the proof of Theorem 3 we get the following result, which is a discrete analog to what is known as critical damping.

Corollary 4. *The numerical solution is oscillatory in*

$$\Omega^{osc} = \left\{(hc, h^2k) \in \Omega\,;\; hc < 2h\sqrt{k} - \frac{1}{2}h^2k(1+2\alpha)\right\}$$

and non-oscillatory in $\Omega \setminus \Omega^{osc}$. *See Figure 2 for an illustration.*

Remark. The oscillation condition (critical damping) for the exact flow φ^h of (6) is given by

$$\omega^{osc} = \left\{(hc, h^2k)\,;\; hc < 2h\sqrt{k}\right\}.$$

Notice that for $\alpha > -1/2$ we always have $\Omega^{osc} \subset \omega^{osc}$. This means that critical damping in the numerical flow is "reached too fast" as c is increased from zero. Curiously, the choice $\alpha = -1/2$ gives exactly the correct critical damping.

4.3. Full stability analysis

We now extend the analysis to the linear test equation

$$\ddot{q} = A(q,\dot{q}) + B(q,\dot{q}) = -k_A q - c_A\dot{q} - k_B q - c_B\dot{q}, \tag{11}$$

with $A(q,\dot{q}) = -k_A q - c_A\dot{q}$ and $B(q,\dot{q}) = -k_B q - c_B\dot{q}$. This test equation is then discretized by the proposed scheme (5). Again, the numerical flow map is linear, i.e.,

$$\Phi_h(q,\dot{q}) = S\begin{pmatrix} q \\ \dot{q} \end{pmatrix}, \qquad \text{with} \quad S = S(h,k_A,c_A,k_B,c_B) \in \mathbb{R}^{2\times 2}. \tag{12}$$

Hence, our objective is to study the magnitude of the eigenvalues of the matrix S. The following result connects the stability analysis carried out in the previous section (the case $B = 0$) to the current case.

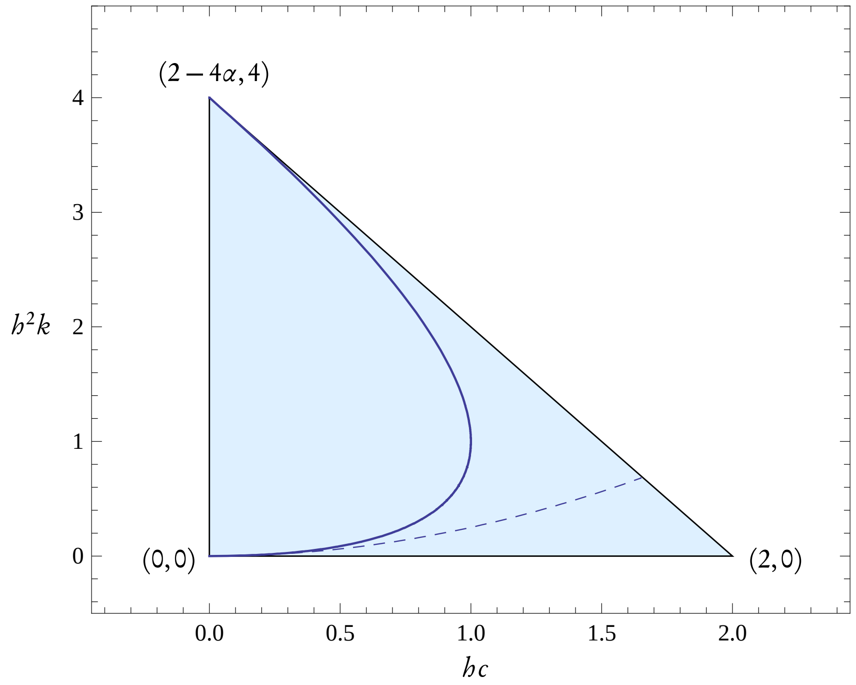


Figure 2: Stability region of the method (5) applied to (6) illustrated in the (hc, h^2k)–plane. The region is triangular, with corners marked. The full drawn curve within the region gives the "critical damping" condition, i.e., to the left of this curve the numerical solution is oscillatory and to the right it is non–oscillatory. The dashed curve gives the critical damping condition of the exact flow.

Lemma 5. *The transformation*

$$hc \quad \longleftrightarrow \quad \frac{2h(c_A+c_B)}{2+2\beta hc_B+\beta h^2k_B} \tag{13a}$$

$$h^2k \quad \longleftrightarrow \quad \frac{2h^2(k_A+k_B)}{2+2\beta hc_B+\beta h^2k_B} \tag{13b}$$

$$\alpha \quad \longleftrightarrow \quad \frac{\alpha h^2k_A+\beta h^2k_B}{h^2k_A+h^2k_B} \tag{13c}$$

takes the characteristic equation (9) *into the characteristic equation* $\det(S-\mathrm{Id}\lambda)=0$.

Proof. Substitute (hc,h^2k,α) in (9) by the transformation (13). The resulting equation coincides with the characteristic equation $\det(S-\mathrm{Id}\lambda)=0$. □

From Theorem 3 we know for which (hc,h^2k,α) the eigenvalues of R in (8) fulfill the root condition. Thus, using Lemma 5, we can investigate the root condition for the eigenvalues of S in (12). The following result, which asserts that the stability of the "explicit part" is not affected by the "implicit part", is then obtained:

Theorem 6. *For* $\beta \geq 1/2$ *the method* (5) *applied to* (11) *is stable for all* $k_B, c_B \geq 0$ *if* $(hc_A,h^2k_A)\in\Omega$.

Proof. From Theorem 3 it follows that the root condition is fulfilled for the roots of (9) if

(i) $h^2k \geq 0$,

(ii) $hc+\alpha h^2k \leq 2$,

(iii) $(1/2-\alpha)h^2k-hc \leq 0$.

Using Lemma 5, the root condition for the roots of $\det(S-\mathrm{Id}\lambda)=0$ are fulfilled if the conditions obtained by substituting (13) in (i)–(iii) are fulfilled. Thus, our objective is to investigate (i)–(iii) after the substitution (13).

(i) Trivial because (13b) is always non-negative, so the condition is always true.

(ii) After substitution the condition becomes

$$\frac{2hc_A+2\alpha h^2k_A+2hc_B+2\beta h^2k_B}{2+2\beta hc_A+\beta h^2k_B} \leq 2\,.$$

For $\beta \geq 1/2$ the left hand side is a decreasing function of both c_B and k_B. Thus, the left hand side is maximal when $c_B=k_B=0$, which corresponds to the case $B=0$.

(iii) After substitution the condition becomes

$$\frac{h^2k_A(1-2\alpha)-2hc_A+h^2k_B(1-2\beta)-2hc_B}{2+2\beta hc_A+\beta h^2k_B} \leq 0\,.$$

Again, for $\beta \geq 1/2$ the left hand side is a decreasing function of both c_B and k_B, and the maximum at $c_B=k_B=0$ corresponds to the case $B=0$. □

Remark. The stability result in Theorem 6 is the best possible, because for $B=0$ it replicates Theorem 3, and for $A=0$ it gives unconditional stability (cooresponding exactly to the classical θ–method with $\theta=\beta$).

5. Adaptivity

In order to increase the efficiency of the integration process it is important to introduce adaptive time stepping. There are various ways of doing this. It is out of scope of this paper to discuss any of them in full detail, but we mention two techniques.

The classical approach is to estimate the local error l^{err} at each time step, and then to consider the control objective $l^{err} \approx tol$ for some user specified tolerance level tol, see [Söd02, Söd03] for details.

Another approach, which is typically used in conjunction with geometric integration, is to introduce a Sundman transformation of the governing equations, which is a dynamic time transformation. Let s be a strictly positive real valued function on the phase space $\mathscr{P}$, called a scaling function, and introduce a new independent variable τ defined dynamically by $\mathrm{d}/\mathrm{d}\tau = s(z)\mathrm{d}/\mathrm{d}t$. The governing equations (4) then transform into

$$\frac{\mathrm{d}z}{\mathrm{d}\tau} = s(z)X(z)\,. \tag{14}$$

Solutions to (14) correspond to time stretched solutions of (4). Thus, equidistant steps ε in the τ–domain correspond to variable steps $h = h(z) = s(z)\varepsilon$ in the physical time domain. The easiest way to use this approach in conjunction with the proposed method (5) is to set $h = s(z_n)\varepsilon$ at each step $n \to n+1$. For other, more intricate, techniques that also conserve the geometric properties of the flow (e.g. energy for conservative systems), see [MF06, HS05, HLW06].

5.1. Choice of control objective

As mentioned above, in classical ODE solvers the local error is estimated at each time step and is used as step size control objective. We suggest another choice based instead on the stability condition.

In Section 4 we found that for A only dependent on q and t (i.e. no damping), the linear stability condition is $h^2\sigma(\partial A(z)/\partial q) \leq 4$. Thus, if an estimate $\sigma^{est} \approx \sigma(\partial_q A(z))$ is available, then a feasible step size control objective is $h^2\sigma^{est} \approx tol \leq 4$.

For the proposed algorithm (5), an estimate of $h\alpha\partial_q A(z_n)\dot{q}$ is given by $A_{n+\alpha} - A_n$. Thus, the quantity

$$\sigma^{est} = \frac{||A_{n+\alpha} - A_n||}{h\alpha||\dot{q}||} \tag{15}$$

gives an estimate of the stiffness in the direction of the flow. Notice that (15) is a function of z_n, i.e., $\sigma^{est} = \sigma^{est}(z)$ is a function on the phase space $\mathscr{P}$. Thus, the corresponding scaling function is given by $s(z) = 1/\sqrt{\sigma^{est}(z)}$. Hence, the Sundman transformation technique (14) may be used in conjunction with this control objective. Furthermore, from Lemma 1 it is evident that for $\alpha \neq 1/2$ this choice corresponds to keeping the principle relative local error term constant for velocity variables.

6. Numerical examples

In this section we present numerical examples of the proposed algorithm (5) applied to: (1) a simple linear problem consisting of two harmonic oscillators; (2) a non-linear pendulum problem in Cartesian coordinates; and (3) a complex multibody ball bearing problem. The last example is carried out in the multibody environment BEAST, where the method has been implemented.

6.1. Harmonic oscillators

The problem describes two particles, both with mass 1, moving on the real line. Between the two particles there is a linear spring with stiffness 10^3 and damping 10^2. One of the particles is attached to a spring with stiffness 1 and no damping. The governing equations are

$$\ddot{q} = -\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} q - 10^3 \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} q - 10^2 \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \dot{q} \,.$$

In terms of (3), we split the acceleration field so that the first term corresponds to $A(z)$, and the two last terms to $B(z)$. Thus, A corresponds to the weak spring, and B to the stiff spring "constraining" the two particles to stick together. We are interested in resolving the dynamics in A, but not that in B. The frequencies in the system are $1/(4\pi)$ Hz (due to the weak spring) and $(1/\pi)\cdot 10^3$ Hz (due to the stiff spring). As initial data we choose $q_0 = (1.0, 1.1)$ and $\dot{q}_0 = (0,0)$.

Numerical simulation with the method (5) is carried out for the constant step size $h = 1$, and method parameters $\alpha = 1/2$ and $\beta = 0.8$. A comparison is given with the classical θ–method, with $\theta = 1/2$ and $\theta = 0.8$. This method is fully implicit in both A and B, and thus requires many more evaluations of A (which we "pretend" to be expensive).

The results in Figure 3 show that, although the θ–method is more expensive per time step, it is less accurate. This is due to the fact that $\alpha = 1/2$ corresponds to a symplectic integrator for the A–part (the explicit part), which is known to have superior accuracy for conservative systems. The θ–method with $\theta = 1/2$ (i.e. the implicit midpoint rule) is also symplectic, but with this choice the highly oscillatory dynamics is not damped out correctly.

6.2. Non-linear pendulum

The problem is a pendulum expressed in Cartesian coordinates $q = (q^x, q^y)$. The length and mass of the pendulum is 1. Thus, a constraint is given by $||q||^2 - 1 = 0$. This constraint is modeled by as stiff spring. The governing equations are

$$\ddot{q} = g - 10^4(||q||^2 - 1)q \,,$$

where the gravity is given by $g = (0, -1)$. We choose A as the first term and B as the second term.

Initial conditions are given by $q_0 = (1.01, 0)$ and $\dot{q}_0 = (0,0)$. Numerical simulation is carried out for the constant step size $h = 0.01$, and method parameters $\alpha = 1/2$ and $\beta = 0.6$. A comparison is given with the θ–method, with $\theta = 1/2$ and $\theta = 0.6$.

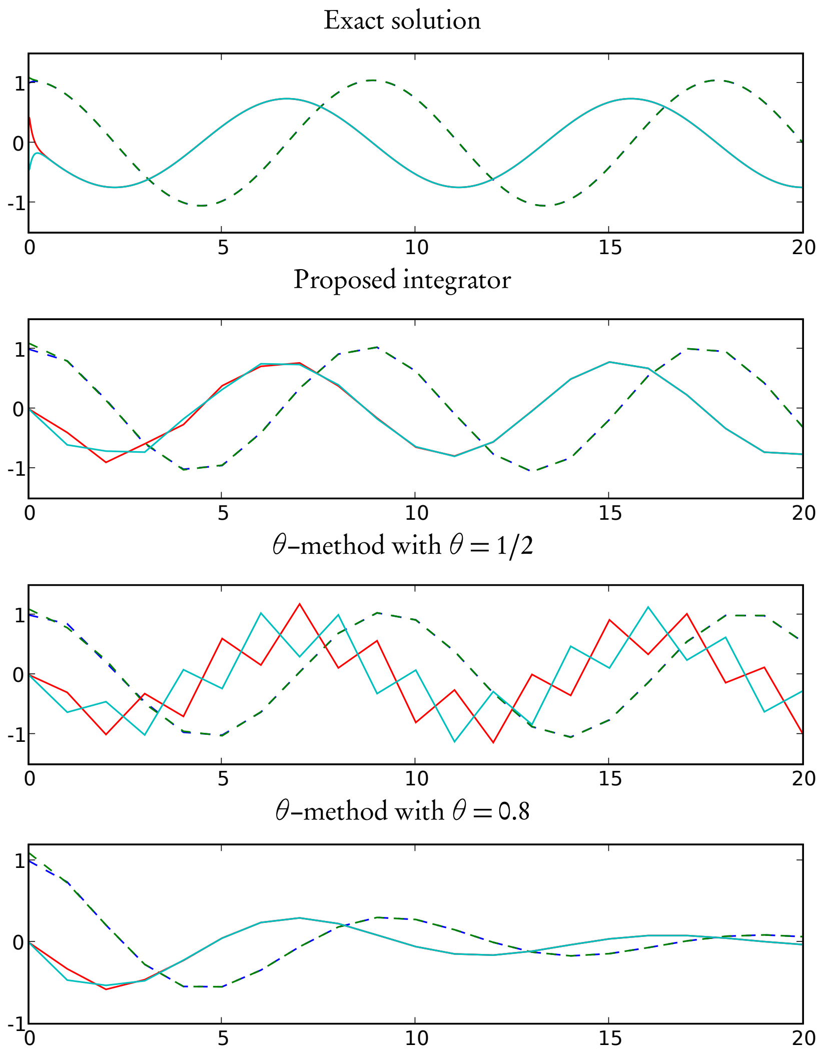
The plots in Figure 4 show the error in the variable q^x. It is small for the $\theta = 1/2$ method, but the solution there contains high oscillations due to the constraint forces, which are not damped out correctly (as in the previous example). Figure 5 shows the constraint error, i.e., the quantity $||q||^2 - 1$. Furthermore, these small oscillations cause the Newton solver to require significantly more iterations, which means more evaluations of A (which we pretend to be expensive).

6.3. Complex ball bearing

This example consists of the ball bearing model illustrated in Figure 1. The outer ring is held fixed, and the inner ring is rotated with 10^4 revolutions per minute. Further, the inner ring is loaded axially with a constant force of 10^3 N.

Simulations of the system is carried out within the software package BEAST with: (i) the proposed integrator with $\alpha = 1/2$, $\beta = 0.8$ and constant step size $h = 10^{-6}$ s; (ii) a standard implicit BDF–solver with adaptive time steps (CVODE, see [Hin]). The plots in Figure 6 show: the contact forces between the first ball and the outer ring; between the first ball and the cage; and the angular velocity of the cage. The results are nearly identical. Since these variables are highly sensitive (especially contact forces on the cage) high similarity between the two simulations indicate that the accuracy is about the same.

Statistics from the two simulations are given in the table below.



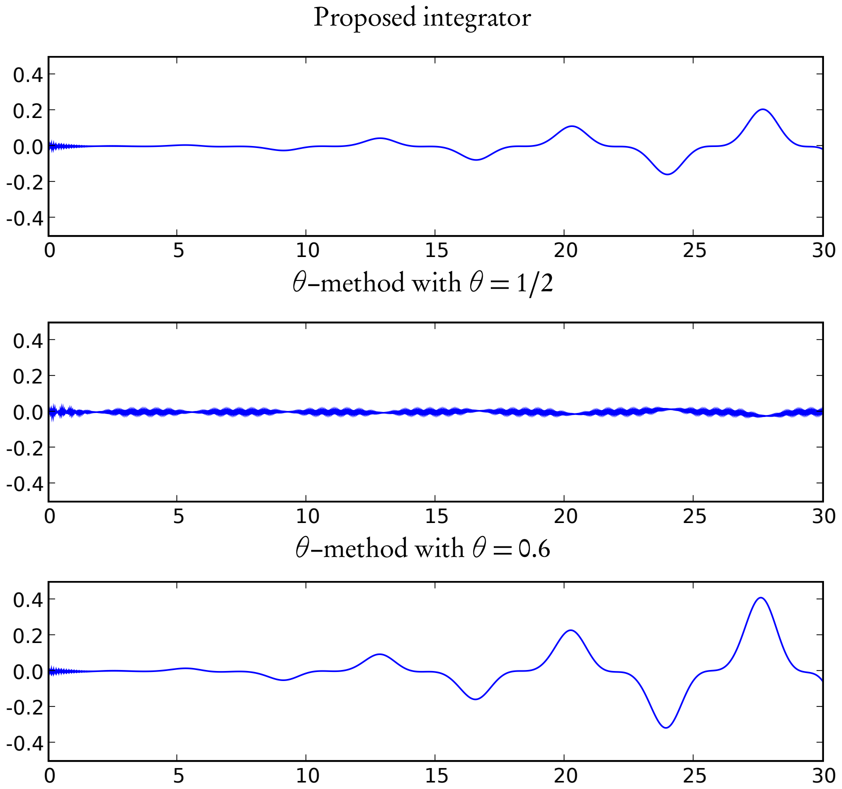
Exact solution

Proposed integrator

θ–method with $\theta = 1/2$

θ–method with $\theta = 0.8$

Figure 3: Numerical results for the test problem in Section 6.1. The full drawn curves are the position variables q and the dashed curves are the velocity variables $\dot{q}$. The upper graph shows the exact solution. Notice that the θ method gives a less accurate result than the proposed method (both for $\theta = 1/2$ and $\theta = 0.8$), even though it is more expensive in terms of evaluations of A.

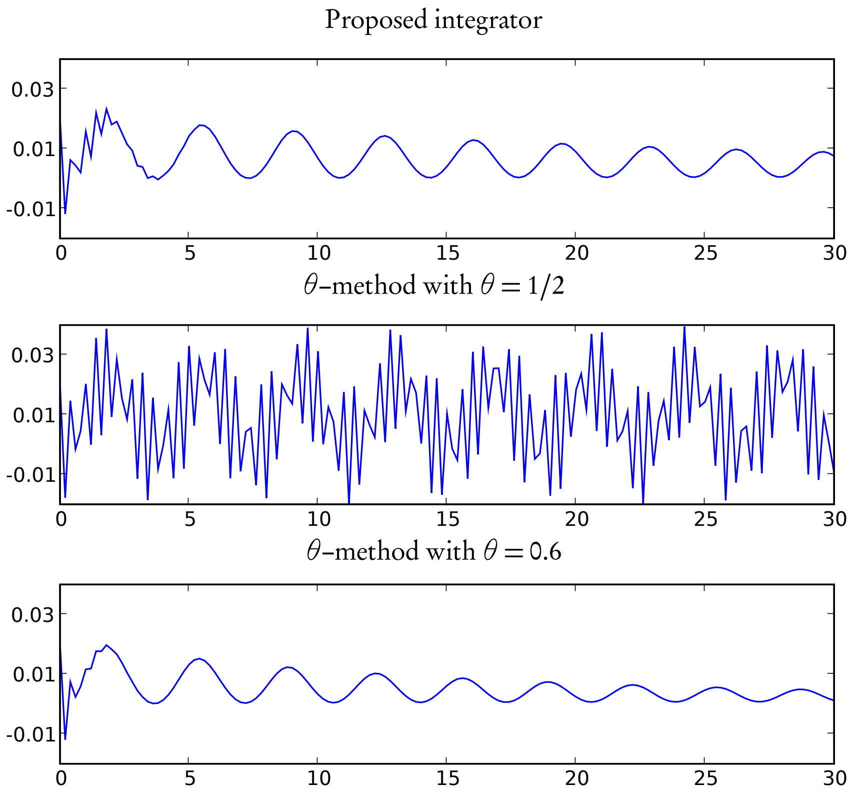


Proposed integrator

θ–method with $\theta = 1/2$

θ–method with $\theta = 0.6$

Figure 4: Numerical results for the test problem in Section 6.2. Global error in the q^x variable. Notice that, although the error is small for the $\theta = 1/2$ method, the solution is highly oscillatory.



Proposed integrator

θ–method with $\theta = 1/2$

θ–method with $\theta = 0.6$

Figure 5: Numerical results for the test problem in Section 6.2. Error in the constraint $||q|| - 1$.

	mean h	force evaluations
BDF–solver	$6.8 \cdot 10^{-7}$ s	1588[1]
proposed method	$1 \cdot 10^{-6}$ s	600

Thus, we gain a factor of about $1588/600 \approx 2.6$ in efficiency. We estimate that the gain will increase if the proposed solver is implemented with adaptive time steps. Further, for rigid models a Jacobian evaluation is relatively cheap (12 force evaluations are needed). With flexible bodies it is much more expensive (12+2n_f force evaluations with n_f number of flexible states), so the potential efficiency gain for models with flexible bodies is promising.

7. Conclusions

A new numerical integrator specifically designed for problems of the type described in Section 2 (e.g. multibody problems with contact forces between bodies) have been proposed. Contrary to standard methods for such problems, the proposed integrator requires only one evaluation of the contact forces per time step, and no contact Jacobians.

Consistency and stability analysis of the proposed integrator have been carried out, and a control objective for adaptive step size implementations has been proposed, based on the stability condition.

Numerical examples show that the proposed integrator is more efficient (in terms of number of contact force evaluations) in comparison with standard implicit integrators.

[1] 1288 force evaluations from iterations plus 25 Jacobian evaluations. Each Jacobian require 12 force evaluation, so in total $1288 + 25 \cdot 12 = 1588$.

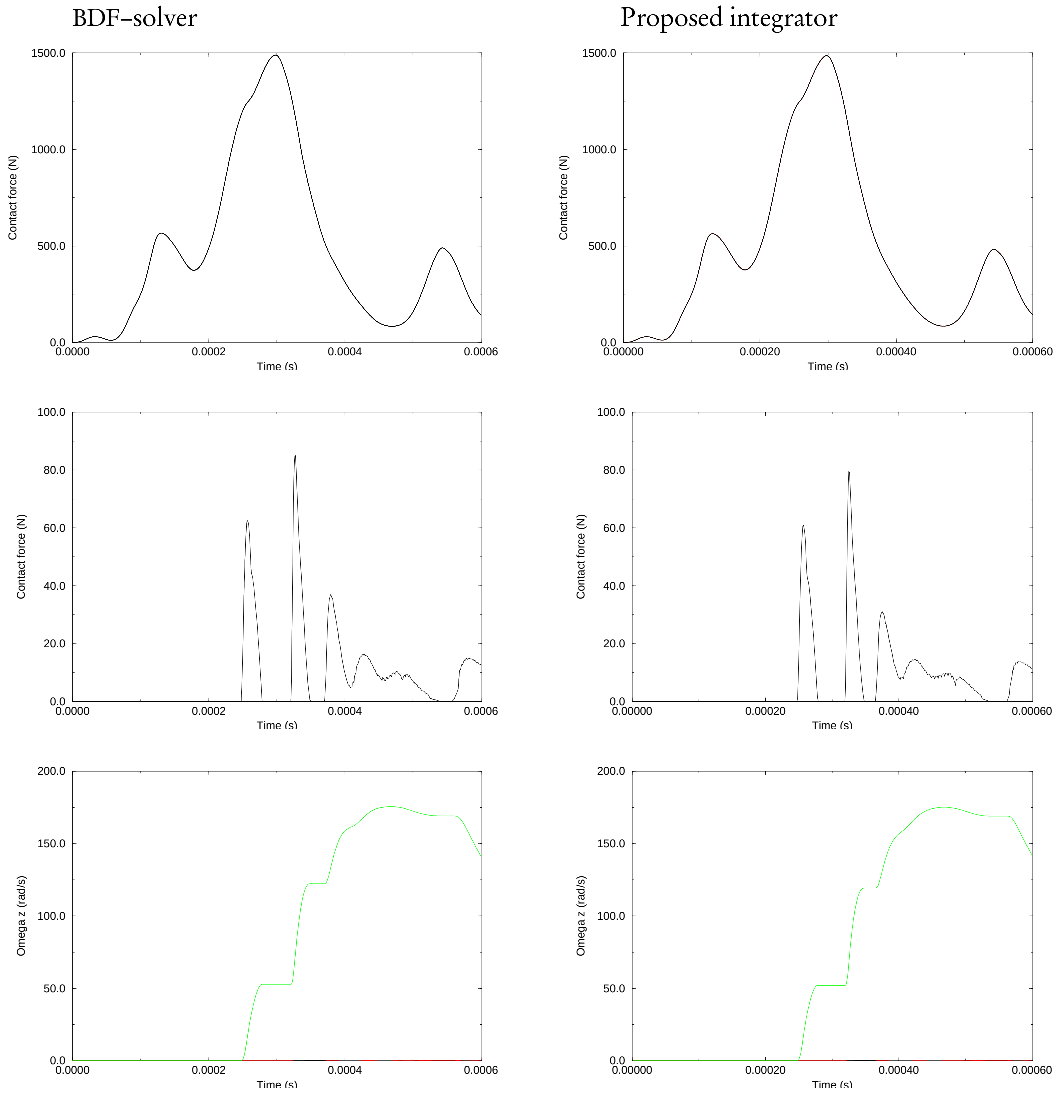


Figure 6: Numerical results for the test problem in Section 6.3. In the left column a standard adaptive BDF–solver is used. In the right column the proposed method is used. The first row is the contact force between one of the balls and the outer ring. The second row is the contact force between one of the balls and the cage. The third row is the angular velocity of the cage.

References

[Hin] A. Hindmarsh. *CVODE open-source software*. `www.llnl.gov/CASC/sundials/`.

[HLW06] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. Structure-preserving algorithms for ordinary differential equations.

[HS05] Ernst Hairer and Gustaf Söderlind. Explicit, time reversible, adaptive step size control. *SIAM J. Sci. Comput.*, 26(6):1838–1851 (electronic), 2005.

[MF06] Klas Modin and Claus Führer. Time-step adaptivity in variational integrators with application to contact problems. *ZAMM Z. Angew. Math. Mech.*, 86(10):785–794, 2006.

[MQ02] Robert I. McLachlan and G. Reinout W. Quispel. Splitting methods. *Acta Numer.*, 11:341–434, 2002.

[Nak06] I. Nakhimovski. *Contributions to the Modeling and Simulation of Mechanical Systems with Detailed Contact Analyses*. PhD thesis, Linköpings Universitet, Linköping, 2006.

[SF01] L-E Stacke and D. Fritzson. Dynamical behaviour of rolling bearings: simulations and experiments. *Proc. Instn. Mech. Engrs.*, 215:499–508, 2001.

[Söd02] Gustaf Söderlind. Automatic control and adaptive time-stepping. *Numer. Algorithms*, 31(1-4):281–310, 2002. Numerical methods for ordinary differential equations (Auckland, 2001).

[Söd03] Gustaf Söderlind. Digital filters in adaptive time-stepping. *ACM Trans. Math. Software*, 29(1):1–26, 2003.

MULTIBODY MODELLING AND DYNAMICAL ANALYSIS OF THE WORM GEAR DRIVE SYSTEMS

Rostyslav Stolyarchuk
State S&R Institute of Information Infrastructure, Lviv, Ukraine
rostyks@yahoo.com

Abstract

The presented work is currently focusing on the dynamic analysis of a worm gear drive mechanical system by multibody contact simulation of a force constraint problem and efficiency analysis. Such systems can be modelled by a Two-Sided Wedge Mechanism (TSWM)as Translational Designed Dynamical Model where worm gearmesh friction contact is exhibited on a plane by two slope active contact lines between conjugated wedge-like rigid inertial bodies via the presence of backlash. This model has been developed to simplify visualization of internal contact conditions and force/reactions distributions resembling a screw-like worm gear mesh. The different dynamical regimes of motion (called as "tractive" and "inverse-tractive") are resulted with respect to the type of a Force Transfer Function between reduced dynamic reactions and appropriate force constraints. The transition from one regime to another physically means instantaneous lose and re-establishing contact line due to the backlash which is resulted in discontinuous event at the force contour/flow level. The mathematical problem formulations have been resulted in the residual ODE form with unpredictable discontinuities in the right-hand side within the set of the Force Transfer Functions. An additional Switching function for control and numerical realization of motion in different regimes has been included. The problem has been tested by MATLAB ODE solver with added switching conditions that incorporated into continuous ODE system. The reduced model parameters of a real-life Motor Operated Valve with the worm gearbox are used in the numerical simulation.

Keywords: Multibody contact systems, internal force contour, friction, force transfer function, switching function and logic, discontinuities.

1. Introduction

The Worm gear reducers are essential elements in the mechanical drive systems including vehicles, tracks, lifts, cranes, valves etc. The dynamically different regimes of motion which dealing with qualitative level of internal force/power flow and efficiency is very actually for the worm gear drive design relies on specific conditions of a sliding friction contact and backlash between gear components. By involves the multibody contact model approach and partitioning co-ordinate method [1], an assembly motion of a conjugated rigid and inertial bodies is acceptable for gear dynamics. Such design dynamical systems with structural-variable properties on the force constraints level can be classified as discontinuous dynamical systems with respect to unpredictable time events causing by system transitions from one dynamic regime of motion to another. On theoretical level, the Discontinuous systems is well discussed in [2,3] and its numerical realization has been developed in [4] by using switching logic approach based on incorporated monitoring tools into the ODE systems in the form of switching functions.

The internal dynamic loads (reduced reactions) and appropriate varying relationships at the Worm Gear Mesh are represented by compatible set of the Force Transfer Functions. Such force/load parameters are the main object of simulation and qualitative system analysis for diagnostic as well. The corresponding Two-Sided Wedge Mechanism as dynamical translational analogue model of the Worm Gear Drive has been developed which can satisfied all possible regimes of motion. The computer–aided formulation of the mathematical model with embedded switching conditions for numerical simulation of motion with discontinuous kernel is discussed. The discontinuities are treated separately on the force constraints level as time events and deals with jumps within the Force Transfer Functions.

2. Design Model

2.1 Description and assumptions

The Design Dynamical Model of the Worm Gear Drive system that modeled by a Two-Sided Wedge Mechanism (TSWM) shown in fig 1. The model can be classified as translational and dynamically structural variable Multibody system with respect to the contact type and appropriate force constraints. The worm gear mesh contact is exhibited on a plane model by two slope active lines (a-a) and (b-b) between conjugated wedge-like rigid and inertial bodies via the presence backlash. This model has been developed to simplify visualization of variable internal force distributions according to the sliding friction contact conditions resembling a screw-like worm gear mesh and to separate motion into two different dynamical regimes called as "tractive" and "inverse-tractive" corresponds to relationships between internal force components (reactions) and contact conditions.

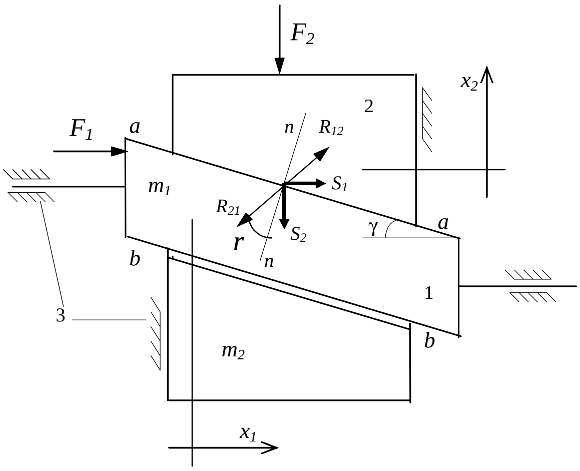


Figure 1. Two-Sided Wedge Mechanism. The Translational Multibody Model of the Worm Gear Drive System

The assumptions have been accepted as follows.

The inertial bodies m_1 and m_2 are realized an assembly motion in generalized rectangular co-ordinate x_1, x_2 with relative motion to each other via one of two slope contact surface-line (a-a) or (b-b) through presence of negligible clearance. The absolute bodies' motion along co-ordinate directions are restricted by ideal support guides 3. The interconnected mass are divided on a driving mass m_1 and driven mass m_2 respectively. They can be derived by following reduced formulas: $m_1 = I_{\sum 1} / R_1^2$, $m_2 = I_{\sum 2} / R_2^2$, where $I_{\sum 1}$ - summarized moment of inertia of a drive elements, which connected to the worm shaft, $I_{\sum 2}$ - summarized moment of inertia of the driven elements, which are connected to the wheel shaft, R_1, R_2 - radiuses on the pitch cylinders of the worm and worm wheel respectively.

The negligible clearance between TSWMs bodies identifies the worm gear mesh backlash and instantaneous transitions during contact-case modifications.

The contact conditions at the TSWM are subjected to the pure sliding friction law resembling to the worm gear contact.

2.2 External forces

The generalized forces F_1 and F_2 (fig.1), which applied to the mass m_1, m_2 respectively and acts along the coordinate directions $0x_1, 0x_2$ are represented an external force contour components.

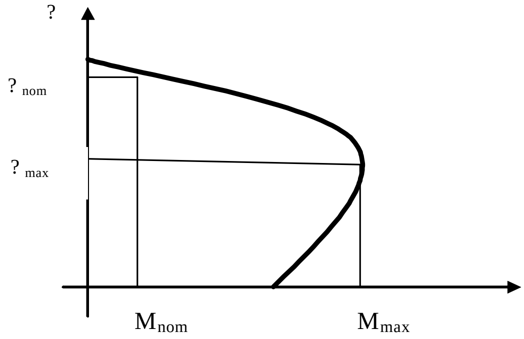


Figure 2. AC motor torque/speed characteristics

For the real-life Motor Operated Drive systems, the force F_1 can be expressed according to the start AC motor characteristic shown on fig.2 by nonlinear function

$$F_1 = \frac{M(w)}{R_1} = \frac{2F_{\max}}{\frac{V_0 - \dot{x}_1}{V_0 - V_{\max}} + \frac{V_0 - V_{\max}}{V_0 - \dot{x}_1}}, \qquad (1)$$

where $V_0 = w_0 R_1$ - synchronous linear velocity; $V_{\max} = w_{\max} R_1$ - maximal linear velocity; $\dot{x}_1$ - current linear velocity of driving body 1; $F_{\max} = M_{\max} / R_1$ - maximal motor force; $w_0, w_{\max}, M_{\max}$ - recommended constant motor parameters; R_1 - worm pitch radius.

The external resistant force $\vec{F}_2$, which is applied to the driven body 2 in the opposite direction to the body motion and used as negative value ($F_2 < 0$), has to be constant over the observed period of time.

2.3 Kinematic constraints

In general, the contact at the wedge mechanism and appropriate worm gear mesh undergoes holonomic constraints with constant kinematic ratio. The constraint equation on position level $x_1 = x_1(\text{t})$, $x_2 = x_2(\text{t})$ according to the contact geometry via slope surface between moving bodies can be written as follows:

$$x_2(t) = x_1(t)tgg \text{ or } x_2(t) - x_1(t)tgg = 0\,, \qquad (2)$$

where g - constant slope angle, which constitutes a geometrical parameter.

Differentiating (2) gives joint velocities and accelerations of the moving parts as corresponding constraint equations

$$\dot{x}_2(t) = \dot{x}_1(t)tgg\ , \qquad (3)$$

$$\ddot{x}_2(t) = \ddot{x}_1(t)tgg\ . \qquad (4)$$

A sliding/relative velocity V_s acts along the slope contact line between the moving bodies is shown on the sketch Fig.3a resembling to the worm gear mesh velocity scheme shows in fig.3b and resulting in

$$V_s = \frac{\dot{x}_{1`}(t)}{\cos g} = \frac{\dot{x}_2(t)}{\sin g}\,, \qquad (5)$$

where $\dot{x}_1(t) = V_1, \dot{x}_2(t) = V_2$ denotes linear velocities in the worm gear mesh.

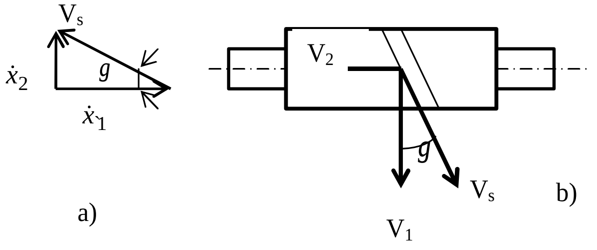


Figure 3. Sketch plane of the distribution of the linear velocities at the TSWM and the worm gear mesh

2.4 Internal force contour and constraints

The reduced reactions $\vec{S}_1$ and $\vec{S}_2$ in fig.1 acting at arbitrary contact point via the wedge slope between the conjugated bodies are represented the components of an internal force contour of the TSWM and respectively modeled the tangential forces in the worm gear mesh. They are resulted from projections of internal reactions R_{12} and R_{21} $(\vec{R}_{12} = -\vec{R}_{21})$ on appropriate coordinate directions of the moving wedge components. In the Fig.4a and Fig.4b have been summarized the graphical results of the force distribution according to the active contact line (a-a) and (b-b) respectively, which based on kinetostatic analysis. We can make the following observations from the fig.4a,b.

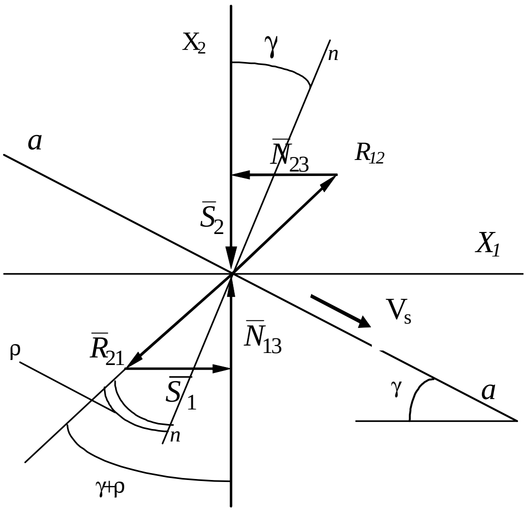


fig.4a Force plane within (a-a) contact case

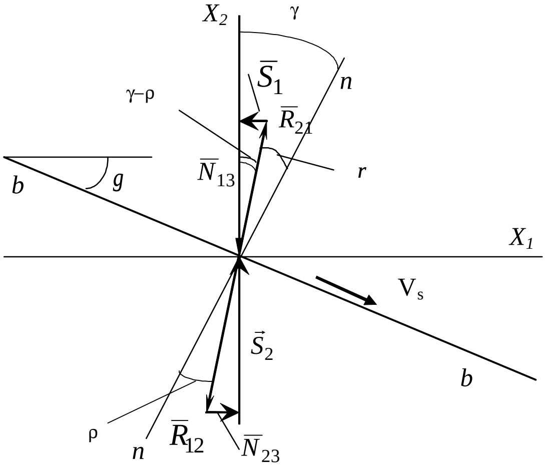


Fig.4b Force plane within (b-b) contact line

1). The internal reduced reactions $\vec{S}_1$ and $\vec{S}_2$ represents the resultant dynamical loads in the TSWM and identifies relationships between tangential forces F_{t1} and F_{t2} in the worm gearing.

2).The values of reduced reactions S_1 and S_2 are unknown system's parameters and can obtained through simulation. Nevertheless, the relationships between reduced forces $\vec{S}_1$ and $\vec{S}_2$ can be derived from the force planes shown on fig.4 (a,b) and have established the force constraint equation in the form

$$\begin{aligned} &S_1 = y_{\,j} S_2, (j = 1{,}2) \\ &y_1 = -tg(g + r), y_2 = -tg(g - r) \end{aligned}, \qquad (6)$$

where indices $j = \overline{1{,}2}$ denote the corresponding type of a Force Transfer Function for each of two qualitatively different regimes of motion associated with relationships between lead angle g and friction angle

r . Hereby the following remarks needs for regimes treatment are introduced.

The first regime of motion can be called as "tractive" regime within active contact line (a-a) that undergoes a direct internal force flow from body 1 to body 2 when $\vec{S}_1 > 0, \vec{S}_2 < 0$. The corresponding absolute value of the FTF (6)$|y_1|$ and the Force Amplification Factor (FAF) $c_{j=1} = \frac{1}{|y_1|} = \frac{1}{tg(g + r)}$ >1 (7) can be given for qualitative regime estimation and analysis.

The second regime can be called as "inverse-tractive" dynamical regime dealing with motion via the active contact line (b-b) when the reversed internal force flow from body 2 to body 1 undergoes following force conditions $\vec{S}_1 < 0, \vec{S}_2 > 0$. The absolute value $|y_2| = |tg(g - r)|$ and corresponding Force Amplification Factor (FAF) $c_{j=2} = \frac{1}{|y_2|} = \frac{1}{tg(g - r)}$ >>1 (8) are given for regime qualitative estimation.

The "inverse-tractive" regime is unfavorable before the “tractive” regime of motion in the sense of efficiency, wear, durability and contact strength, especially, for a bronze worm wheel with respect to steel hardness worm.

The qualitative level of power losses can be evaluated for each regime by Efficiency factor written as:

for the "tractive" regime

$$h = \frac{S_1\dot{x}_1}{S_2\dot{x}_2} = \frac{tgg}{tg(g + r)}, \qquad (9)$$

for the "inverse-tractive" regime

$$m = \frac{S_2\dot{x}_2}{S_1\dot{x}_1} = \frac{tg(g - r)}{tgg}. \qquad (10)$$

2.5 Approximation of the Friction angle

The sliding friction angle r at a contact zone of the TSWM and appropriate worm gearmesh have been obtained from experimental measurements as function against relative velocity V_s and given by table form that reported in [5].

These table data can be analytically defined by approximation formula and written as

$$r(V_s) = \left(aV_s^{\,b} + c\right)^{-1} \qquad (11)$$

or by using one of model state variable $\dot{x}_1$ for the driving mass 1, we can rewrite (11) as

$$r(\dot{x}_1) = 1/\left(a\left(\dot{x}_1 / \cos g\right)^b + c\right), \qquad (12)$$

where a,b,c - constant coefficients of approximation. Those coefficients have been evaluated by Newton's approximation method [5] depends on contact gear materials, which are summarized at the table 1 and used below for the simulation test example.

Table 1

Group of contact materials	Coefficients a,b,c (11)
I*	?=0,256;b=0,591;c=0,151
II*	a=0,174;b=0,642;c=0,140
III*	a=0,161;b=0,535;c=0,088

*Group I.Worm: hardened steel HRC>48. Wheel: Bronze (Sn 6-7%, Ni 1-2%)

**Group II. Worm: hardened steel 48>HRC>32. Wheel: Bronze(Sn 6-10%

***Group III.Worm: hardened steel HRC>48. Wheel: Bronze (Fe 1%)).

3. Mathematical Model

The dynamics equations of motion including active internal forces and complement constraint equations are resulted in DAEs form

$$\begin{aligned} m_1\ddot{x}_1 &= F_1 - S_1, \\ m_2\ddot{x}_2 &= F_2 - S_2, \end{aligned} \qquad (13)$$

$$\begin{aligned} \ddot{x}_2 &= \ddot{x}_1 tgg, \\ S_1 &= S_2 y_j, (j = \overline{1,2}). \end{aligned} \qquad (13a)$$

$y_1 = tg(g + r), y_1 = tg(g - r)$ All

The mathematical problem formulation results in residual ODEs form with unpredictable discontinuityes in the right-hand side within FTFs together with corresponding initial conditions and implementation of non-linear functions $r = r(\dot{x}_1)$ (12) and $F_1 = F_1(\dot{x}_1)$ (1) are constitutes a full mathematical model resulted in

$$\ddot{x}_1(t) = \frac{F_1 - F_2 y_{\,j}}{m_1 - m_2 y_{\,j}\,tgg}, (y_{\,j} = \{y_{\,j}, j = 1{,}2\}),$$

$$x_1(0) = x_{10},\ \dot{x}_1(0) = \dot{x}_{10}. \quad (14)$$

Where

$$y_{\,1} = -tg(g + r(\dot{x}_1)), y_{\,2} = -tg(g - r(\dot{x}_1)),$$

$$r(\dot{x}_1) = 1/\left(a\left(\dot{x}_1 / \cos g\right)^b + c\right) \qquad .$$

$$F_1 = \frac{2F_{\max}}{\dfrac{V_0 - \dot{x}_1}{V_0 - V_{\max}} + \dfrac{V_0 - V_{\max}}{V_0 - \dot{x}_1}}$$

This mathematical model (14) of the TSWM describes an assembly motion in different regimes within the space of independent state coordinates $x_1, \dot{x}_1$. For numerical realization by a time-stepping integration method (ODE solver) this problem needs an additional Switching Function $j(t)$ that has been constructed on acceleration level and introduced into following control block

$$y_{\,j} = \begin{cases} y_{\,1}\ if\ \ j(t) > 0 & for\ the\ tractive\ regime \\ y_{\,2}\ if\ \ j(t) < 0 & inverse - tractive\ regime. \end{cases}$$

$$j(t) = \ddot{x}_1^{\,own}(t) - \ddot{x}_2^{\,own}(t)\frac{1}{tgg}, \qquad (15)$$

where

$\ddot{x}_1^{\,own} = \dfrac{F_1(\dot{x}_1)}{m_1}$ - is a so-called the “own” accelerations of body 1;

$\ddot{x}_2^{\,own} = \dfrac{F_2}{m_2}$ - “own” accelerations of body 2.

The expressions of the “own” accelerations illustrate its dependence only from external active forces and reduced masses however its independence from internal reactions.

The positive values of SF $j(t) > 0$ physically guarantee a direct internal force flow with FTF $y_{\,1}$ and assembly motions of TSWM in “tractive regime”. The negative values of the SF $j(t) < 0$ means assembly motion in reversed internal force flow with FTF $y_{\,2}$ in the “inverse-tractive” regime.

Jump discontinuities within $y_{\,j} = \{y_{\,1}, y_{\,2}\}$ can occurs at the time event $t = t$ when have been satisfied zero equality $j(t) = \ddot{x}_1^{\,own}(t) - \ddot{x}_2^{\,own}(t)\dfrac{1}{tgg}$ =0 and internal forces are equal zero $S_1(t) = S_2(t) = 0$. The time event t is called as a Switching Point [3] and physically deals with the instantaneous loose and re-established contact at TSWM and consequently in the worm gear mesh.

4. Simulation results

The problem have been tested using resulted formulas for internal reduced forces $S_1(t), S_2(t)$ written as

$$S_1(t) = F_1(\dot{x}_1) - m_1\ddot{x}_1,\ S_2(t) = S_1(t)/y_{\,j}, j = \overline{1{,}2}$$

and FTFs in the time history according to derived state variables $x_1(t), \dot{x}_1(t), \ddot{x}_1(t)$.

In order to use one of MATLAB ODE solver, we need to convert the 2nd order ODE (14) into the 1st order ODEs by defining some new variables $y_1 = x_1,\ y_2 = \dot{x}_1$.

Using these new variables, our original mathematical model with added switching function and control conditions are written as follows:

$$\dot{y}_1 = y_2,$$

$$\dot{y}_2 = \frac{F_1 - F_2 y_{\,j}}{m_1 - m_2 y_{\,j}tgg}, \qquad y_{\,j} = \begin{cases} y_{\,1}\ if\ \ j(t) > 0, \\ y_{\,2}\ if\ \ j(t) < 0, \end{cases}$$

$$y_{\,1} = -tg(g + r(y_2)), y_{\,2} = -tg(g - r(y_2)),$$

$$r(y_2) = 1/(a(y_2 / \cos g)^b + c),$$

$$V = (V_0 - y_2)/(V_0 - V_{\max})$$

$$F_1 = 2F_{\max}/(V + 1/V)$$

$$j(t) = F_1 / m_1 - F_2 /(m2 \cdot tgg)$$

Implementation of the above model into Matlab function m-file named `cpm1.m`. is shown below. The function name must agree with the file name. Modal variables declared as global or constant.

```
function dy=cpm1(t,y)
global m1 m2 pk vc1 vk1 gam a b c
F2
V=(vc1-y(2))/(vc1-vk1);
F1=2*pk/(V+1./V);
fi=F1/m1-F2/m2/tan(gam);
ro=1/(a*(y(2)/cos(gam))^b+c);
ro=ro*atan(1)/45;
if sign(fi)==1
 lam=-(tan(gam+ro));
else lam=-(tan(gam-ro)); end
```

```
dy1=y(2);
dy2=(F-lam*F2)/(m1-
m2*lam*tan(gam));
dy = [dy1;dy2];
```

Solving the equation of motion (14) by MATLAB solver ODE23s with zero initial conditions and integration time interval from 0 to o,14 s shown below in code as fragment of the script m. file

```
y0=[0 0];% initial conditions in vector form
t0=0;tfinal=0.14;% time interval
[t,y]=ode23s('cpm1',time,y0,options
);
```

The model parameters are put in the table 2, which are taken for the Motor Operated Valve with worm gear reducer. The AC motor technical characteristics: P=7,5 kW, n_0=3000 rpm; n_n=2900 rpm, M_{max}/M_n=2,2 ,s_{max}=17%;I_r= 0.0069 kgm^2. Worm gear parameters: reduction ratio u=40:1; diameter of worm d_1=50 mm, Contact gear materials - group III.

Table 2

Model data	Program code	Value range
m_1	m1	11.8 kg
m_2 (var)	m2	[100; 200,300]kg
F$_2$ (var)	F2	[-5000;-10000;-15000] N
g	gam	4,76 grad
$F_{\max}$	pk	2178 N
V$_0$	vc1	7.85 m/s
V$_{\max}$	vk1	6.5 m/s
a	a	0.161
b	b	0.535
c	c	0.088

On fig.5,6,7,8 have been plotted the simulation results of $S_1(t), S_2(t)$ versus time with denoted peak values S_1^m, S_2^m for the driven mass equal m_2=200 kg and external resistant force F_2 = -10000 N., F_2 = -15000 N.

In general, the dynamic response has to be determined by means of peak values S_1^m, S_2^m with calculation of the dynamical factors K_{d1}, K_{d2} according to the expressions

$$K_{d1} = \left|S_1^m\right| / F_{t1} \text{ and } K_{d2} = \left|S_2^m\right| / F_{t2} \text{ , (15)}$$

where F_{t1}, F_{t2} are the values of tangential forces in ??? worm gear mesh taken from static force analysis.

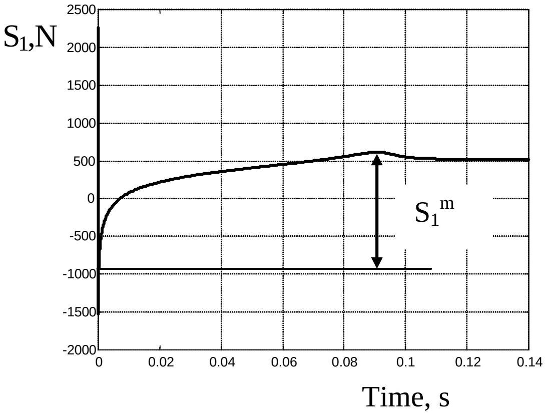


Figure 5. Plot of $S_1(t)$ with peak value S_1^m =1500 N for F$_2$=-10000 N

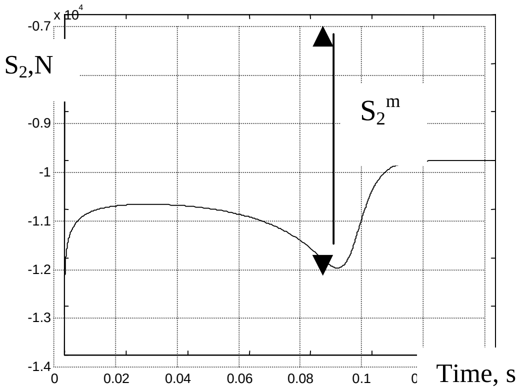


Figure 6. Plot of $S_2(t)$, peak value S_2^m =-12100 N for F$_2$=-10000 N

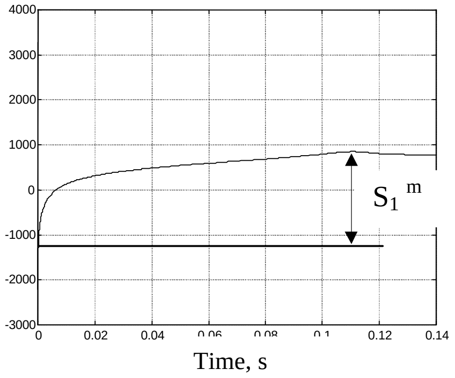


Figure 7. Plot of $S_1(t)$ with peak value S_1^m =2000 N for F2=-15000 N.

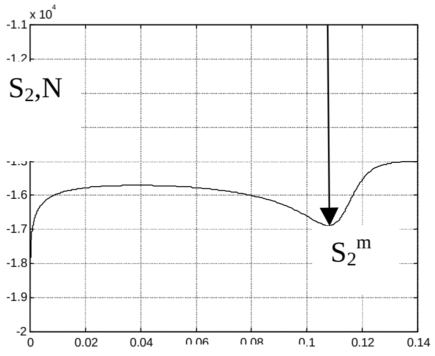


Figure 8. Plot of $S_2(t)$, negative peak value S_2^m = -16900 N for F2=-15000 N.

The simulation results are summarized in Table 3

Table 3

Test case	Model variables m_2 [kg], F_2 [N]	S_1^m, [N]	K_{d1} (15)	S_2^m, [N]	K_{d2} (15)
1	100 kg; -5000 N;	800	0,81	-6300	0,76
2	100 kg; -10000 N;	1400	1,4	- 11300	1,37
3	100 kg; -15000 N;	1800	1,84	- 16500	2,0
4	200 kg; -5000 N;	900	0,91	-7600	0,92
5	200 kg; -10000 N;	1500	1,50	- 12100	1,36
6	200 kg; -15000 N	2000	2.01	- 16900	2,0

We can observed from the table 3 that an increase in the resistant force F_2 is claim increasing dynamical factor in the range K_d = [0,76 ; 2,0]. The force F_2 has the more influence on dynamical coefficient then increase in the driven mass m_2.

The presented hereby results shows that the worm gear drive system of MOV for all parameter range is realized only “tractive” regime of motion when S_1>0 and $S_2 < 0$ that means dynamic behavior of system in direct transmitted energy and force flow.

5. Conclusions

New formulations for the simulation of dynamics of Worm gear drive systems have been described in this work. The developed TSWM and appropriate mathematical model as ODE with discontinuous right-hand side includes sliding friction contact, modeled by friction angle and the actual lead angle of worm geometry with the variation of the force constraints and efficiency. The variation of contact and Force Transfer Functions between moving parts of multibody system just requires to perform a dynamic simulation including two-dimensional switching logic and appropriate switching function on acceleration level. The model allows implementation of motor drive characteristics as appropriate external force functions which depends from state velocity as well.

These model help engineers to easily identify and effectively solve the root causes of an engineering problem within a worm gear system.

Acknowledgements

The support of the Royal Swedish Academy of Engineering Sciences and Prof. Erik Dahlquist from MDH Malardalen University are gratefully acknowledged.

References

[1] Nikravesh PE. (1988). “Computer-Aided Analysis of Mechanical Systems”. Prentice-Hall, Englewood Clifs. NJ. USA.

[2] Leine R.I.,Van Campen D.H., Van De Vrande B.L. (2000). “Bifurcation in Nonlinear Diacontinuous Systems”. In J.Nonlinear Dynamics, 23: pp.105-164. Kluwer Academic Publishers.

[3] Aizerman M.A., Piatnitsky Ye.S. Fundamentals of the theory of discontinuous systems I,II. Automatics and telemechanics.. No.7 p.33-47, No.8.p.39-61.1974. (in Russian)

[4] Eich Soellner E., Fuhrer C., (2002). “*Numerical Methods in Multibody Dynamics”.* Lund, Corrected reprint of Teubner Stuttgart.

[5] Rostyslav Stolyarchuk.. (2000). “Approximation of non-stationary sliding friction characteristic in the worm gear “. In Journal News of Ternopil State Technical University. Vol.5,No3, pp. 129-134..(Ternopil, Ukraine)

Aspen HYSYS Simulation of CO_2 Removal by Amine Absorption from a Gas Based Power Plant

SIMS2007 Conference, Gøteborg, October 30-31st 2007

Lars Erik Øi
Telemark University College, Norway
lars.oi@hit.no

Abstract

A simplified combined cycle gas power plant and a MEA (monoethanol amine) based CO_2 removal process have been simulated with the process simulation tool Aspen HYSYS. The thermodynamic properties are calculated with the Peng Robinson and Amines Property Package models which are available in Aspen HYSYS. The adiabatic efficiencies in compressors, gas turbines and steam turbines have been fitted to achieve a total thermal efficiency of 58 % in the natural gas based power plant without CO_2 removal. The efficiency is reduced to about 50 % with CO_2 removal. The CO_2 removal in % and the energy consumption in the CO_2 removal plant are calculated as a function of amine circulation rate, absorption column height, absorption temperature and steam temperature. With CO_2 removal of 85 %, heat consumption is calculated to 3.7 MJ/kg CO_2 removed, close to a literature value of 4.0 MJ/kg CO_2.

Keywords: Aspen HYSYS, CO_2 removal, monoethanol amine, absorption.

1. Introduction

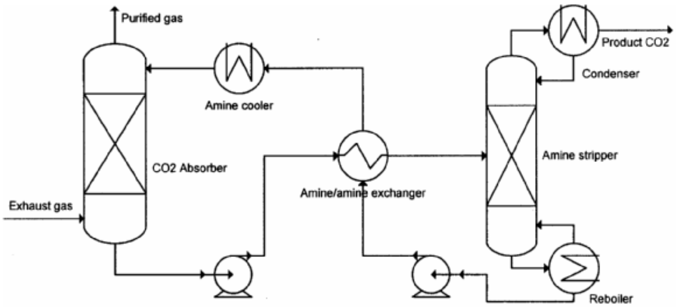


Figure 1. Principle for CO_2 removal process based on absorption in amine solution.

The possibility of removing CO_2 from fossil fuel based power plants has got increased interest due to environmental reasons. The most actual method for CO_2 removal is by absorption in an amine based solvent followed by desorption. The principle of the removal process is shown in figure 1. The simplest and most used amine for CO_2 removal is MEA (monoethanol amine). This removal process has a high consumption of thermal energy, and one of the main aims of improvement is to minimize this energy consumption. In the case of CO_2 removal from a combined cycle gas power plant, it is natural to cover the heat requirement for CO_2 stripping with steam from the power plant. A possible steam outlet from a typical combined cycle process is shown in figure 2.

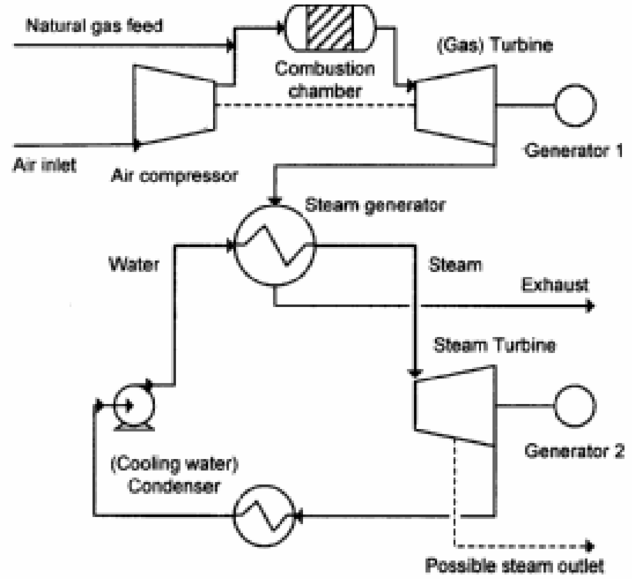


Figure 2. Principle of combicycle power plant

Because testing at large scale is so expensive, it is natural to use process simulation to evaluate such processes. There are however few literature references on process simulation of CO_2 removal from exhaust gases at atmospheric pressure. There are some journal articles [1-4], and 3 of them have used the process simulation program Aspen Plus. AspenTech bought the program HYSYS from HyproTech in 2002, and in 2006 the program name was changed to Aspen HYSYS. The last reference [4] uses a Fortran code to simulate the process. An important advantage with using a process simulation program for such calculations, is that the available models for thermodynamic properties can be used. Aspen Plus has an MEA property insert model [5] which was used in all the references [1-3]. Aspen HYSYS has an Amines Property Package [6]. Within the Amines Property Package, one of the two models, Kent Eisenberg or Li-Mather, can be selected. Even though Aspen HYSYS is probably the most used process simulation program in the world, there has not been found any journal references on CO_2 removal from atmospheric exhaust gas using this program. There is however much literature on CO_2 removal from natural gas at high pressure. Other process simulation programs containing amine packages, are ProVision and ProMax.

At Telemark University College, Hysys has been used since 2000 to simulate CO_2 removal from gas based power plants. Most of this work has been Bachelor and Master student projects with the author of this paper as supervisor. Major challenges in the simulation of CO_2 absorption and desorption processes, are the description of thermodynamics and absorption efficiency, convergence and total energy or cost optimization.

The purpose of this paper is to present and discuss the use of Aspen HYSYS for simulating CO_2 removal from atmospheric exhaust by amine absorption, and to simulate the total effect of efficiency reduction in a combined cycle gas power plant.

2. Available models

2.1 Absorption and reaction mechanisms

The details of the mechanisms of CO_2 absorption into an amine solution in an absorption column are quite complex. There are many references about the chemistry involved in the process, and many references and models comprising mass transfer mechanisms and chemical reaction kinetics. Reviews are written by Danckwerts and Sharma [7] and Versteeg [8].

First, CO_2 has to be transported from the gas to the liquid surface, and then it is absorbed in the liquid solution. The gas liquid interface area a (in m^2/m^3) and liquid holdup h (in m^3/m^3) are main parameters in describing such mechanisms.

In the liquid, the CO_2 may react chemically with other components. The following reactions are normally assumed to take place when CO_2 reacts in a primary amine like MEA in an aqueous solution. In the case of MEA ($NH_2C_2H_2OH$), R is C_2H_2OH.

$$CO_2 + NRH_2 \rightarrow RH_2^+NCOO^- \qquad (1)$$

$$RH_2^+NCOO^- + NRH_2 \rightarrow RH_2NCOO^-NRH_2^+ \qquad (2)$$

According to equation (1) and (2), two moles of MEA are necessary to absorb one mole of CO_2.

A simple overall description of the combined absorption and reaction process is simply

$$CO_2\ (gas) \rightarrow CO_2\ (absorbed) \qquad (3)$$

The removal of CO_2 is not 100 %. The % CO_2 removal is limited both by low absorption and reaction rates and by the equilibrium conditions.

If the kinetics in the reactions should be calculated, more details about the intermediate reactions in equation (1) and (2) should be included. This is done in the MEA property insert model in Aspen Plus.

The simulation program Aspen HYSYS is mainly based on equilibrium calculations. In that case, equation (3) is sufficient to calculate the absorption process.

2.2 Equilibrium models

The concentration of CO_2 in the gas may be expressed by the partial pressure p_{CO2} (eg. in bar) and the concentration in the liquid may be expressed by C_{CO2} (eg. in mole/m^3).

The equilibrium between the concentrations of CO_2 in a gas and a liquid may be represented as a function

$$p_{CO2} = f(C_{CO2}) \qquad (4)$$

This expression might be a function of temperature, pressure and concentrations of the components in the solution. There are many models available to describe this function.

In Aspen HYSYS, gas/liquid equilibrium for a component (i) is normally calculated using k-values defined by the equation

$$K_i = y_i/x_i \qquad (5)$$

where y_i and x_i are the mole fractions of (i) in the gas and liquid phase. For general purpose use, equation of state models like SRK (Soave Redlich Kwong) and PR (Peng Robinson) are often used. Aspen HYSYS recommends Peng Robinson [9]. Peng Robinson is regarded to be suitable to handle systems containing hydrocarbons, water, air and combustion gases, the typical components in a natural gas based power plant.

Traditional equation of state models are not regarded to be suitable for non-ideal liquid systems. An amine solution is an electrolytic system and also comprises chemical reactions. This is not expected to be well described with traditional equations of state.

Within the Amines Property Package in Aspen HYSYS, one of the two models, Kent Eisenberg [10] or Li-Mather [11], can be selected. The models are quite complex, but in principle they are models to describe the equilibrium of the CO_2 concentrations in the gas and the liquid (equation 4).

Aspen Plus has an electrolytic package to calculate liquid systems containing ions. Different electrolyte equilibrium models can be used. Using an MEA property insert model, equilibrium models can be combined with reaction kinetic models, including rate expressions of chemical reactions like equation (1) and (2).

2.3 Column models in simulation programs

A CO_2 absorption column is a unit where gas flows up and liquid (eg. an amine solution) flows down. CO_2 is transfered from the gas phase to the liquid phase where it reacts with the amine solution. The gas and liquid phases are made to get in contact by the help of column plates or random or structured packing.

The CO_2 stripping column also has plates or packing, and this column also has a reboiler at the bottom to provide heating, and a condenser at the top to provide cooling.

The traditional way to model such columns is by using equilibrium stages. One plate can be calculated assuming equilibrium between the CO_2 concentration in the gas and liquid leaving the plate. In a packed column, a certain height of packing can be modeled as one equilibrium stage.

The equilibrium stage model can be refined by introducing a stage efficiency. Murphree efficiency for stage number n is defined by

$$E_M = (y-y_{n-1}) / (y^*-y_{n-1}) \qquad (6)$$

where y is the mole fraction of CO_2 in the gas leaving the stage, y_{n-1} is the mole fraction leaving the stage below, and y* is the mole fraction CO_2 in equilibrium with the liquid leaving the stage. This is illustrated in figure 3.

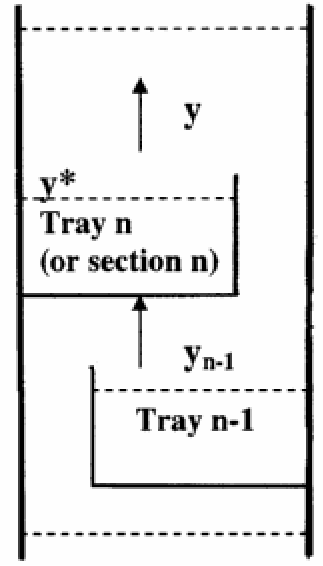


Figure 3: Definition of Murphree efficiency

Most process simulation programs have models for implementing Murphree efficiency in a column model. The Amines property package in Aspen HYSYS has a special estimation method for predicting this Murphree efficiency. This is based on the work of Tomkej [12]. This model is based on experience with CO_2 removal from high pressure natural gas. In Aspen Plus it is also possible to specify these efficiencies explicitly in an absorption or distillation column model.

Aspen Plus has a column model (RateFrac) which can include kinetic rate expressions in chemical reactions at each stage. In the references [1-3] the MEA property insert model in Aspen Plus is used in a column model.

2.4 Column convergence

To converge a column model in a process simulation program, all the equations describing the equilibrium and gas and liquid flows must be solved for each calculation stage. Including rate expressions for chemical reactions complicates the calculations further. This often leads to convergence problems.

The column model in Aspen HYSYS has a default set of convergence criterias, and a default set of calculation parameters. Different calculation models are also available. The Inside-Out algorithm is default, and a Modified Hysim Inside-Out algorithm is also available. A damping parameter for the column iteration is adjustable, and the damping can be specified to be adaptive.

2.5 Flowsheet convergence

All process simulation programs are based on modules for calculating different unit operations like heat exchangers, pumps, distillation columns etc. Process simulation programs are traditionally divided into either sequential modular or equation based programs. In a sequential modular program, the in-streams of each calculation module must be known prior to the calculation, and the out-streams are the result of the module calculation. The programs Aspen Plus and ProVision are sequential modular. Equation based simulation programs can be able to calculate in-streams based on out-streams. Aspen HYSYS is an equation based simulation program.

However, also in Aspen HYSYS, the column models are based on specified in-streams. Because of this, flowsheets with columns in practice have to be calculated in a modular sequential manner.

In many cases, it is of interest to calculate in-streams also to columns. This can be done by iteration methods. In the case of recycle streams, the flowsheet can be solved including recycle blocks. A recycle block compares the in-stream to the block with the out-stream from the block in the former iteration.

In the case of convergence problems in a column model, recycle iterations complicate the calculations further. In some cases, a recycle block will not converge due to parameters of minor interest. An example of such a parameter is the concentration of a trace component. In such cases, a possibility is to iterate manually on the main parameter (eg. the CO_2 concentration) by replacement, and accept the errors in the parameters of minor importance.

3. Calculations

3.1 Power plant simulation

Figure 2 shows the principle of a combined cycle power plant based on combustion of natural gas. A real plant will be much more complex with many heat exchangers, recycle pipes and tanks to optimize the overall efficiency of the power plant. However, a process as in figure 2 is practically realistic, but it will not have an optimum overall efficiency.

The main purpose of the power plant simulation is to make a simple but realistic model to evaluate the influence of heat for CO_2 removal on the overall power plant efficiency.

A simplified combined cycle 400 MW gas based power plant has been simulated with Aspen HYSYS. Pure methane is used as natural gas, air is 79 % nitrogen and 21 % oksygen, 100 % combustion is assumed, and traditional temperatures and pressures are used in the process. The Peng Robinson model has been used for the thermodynamic properties in the power plant. Specifications for the calculation are listed in table 1. A flowsheet of the Aspen HYSYS model is presented in figure 4. This model has been developed during the Master Thesis work of Kristin Vamraak [13] and Bjørn Moholt [14].

Inlet air/gas temperatures	25 ºC
Inlet natural gas pressure	30 bar(a)
Combustion temperature	1500 ºC
Steam high pressure	120 bar(a)
Steam medium pressure	3.5 bar(a)
Steam low pressure	0.07 bar(a)
Pressure to stack	1.01 bar(a)
Stack temperature	100 ºC

Table 1: Specifications for Power Plant Simulation

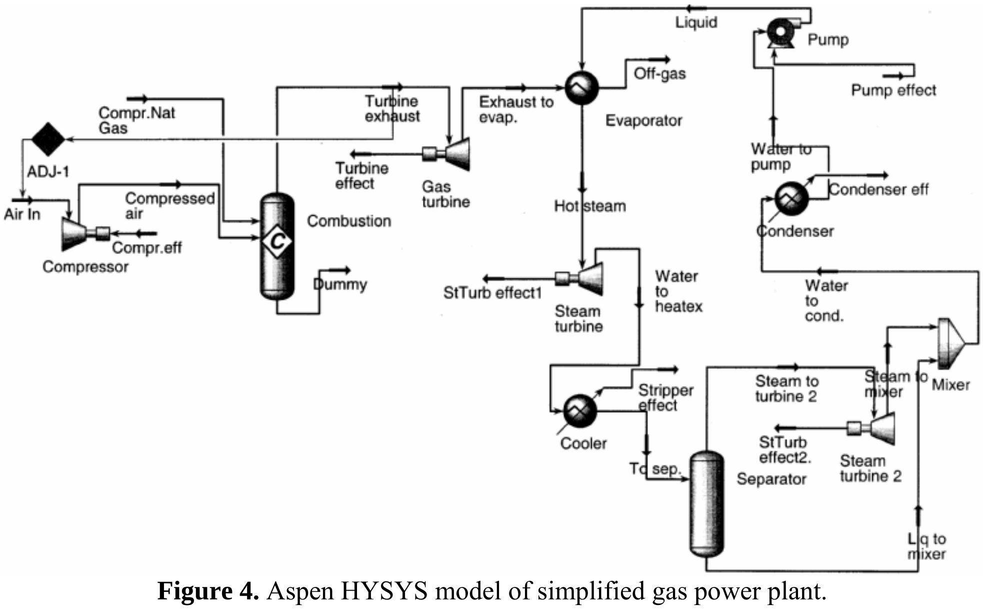


Figure 4. Aspen HYSYS model of simplified gas power plant.

To have a physically possible prosess, the flue gas temperature (680 ºC from gas turbine to exhaust at 100 ºC) has to be higher than the steam temperature through all the steam heat exchanger. This results in a maximum superheated steam temperature of 540 ºC (at 120 bar). The chosen pressure is not optimized.

The efficiencies in compressors, gas turbines and steam turbines have been fitted to achieve a total thermal efficiency of 58 % (which is traditional) in the gas based power plant. The total efficiency is calculated as the turbine effects (minus compressor and pump) divided by the lower heating value of natural gas. The compressor efficiency (adiabatic) was adjusted to 90 %, the expander part of the gas turbine (expander) and the steam turbines were adjusted to 85 % (also adiabatically). These values are high compared to actual efficiencies for such equipment. A real power plant would be more energy optimized, and have lower equipment efficiencies.

In this part of the power plant simulation, the steam delivery (for CO_2 removal) is set to zero.

3.2 Simulation of CO_2 removal base case

An absorption and desorption process for CO_2 removal with an aqueous MEA solution has been simulated. The exhaust gas from the power plant model is used as the feed to this model. The absorption column is specified with 10 stages each with a Murphree efficiency of 0.25. (An estimated HETP (Height Equivalent to a Theoretical plate) of 4 meter, is about equivalent to 0.25 efficiency for each meter of packing.) Traditional concentrations, temperatures and pressures are used in the base case simulation. The thermodynamics for this mixture is described by an Amines Property Package available in Aspen HYSYS. The Kent Eisenberg [10] model is selected in the Amines Property Package. Specifications for the calculation are listed in table 2. The Aspen HYSYS CO_2 removal model is presented in figure 5. Different versions of this model have been developed in several student projects. The version in figure 5 is based on a Master Project work by Trine Amundsen [15].

Inlet gas temperature	40 ºC
Inlet gas pressure	1.1 bar(a)
Inlet gas flow	85000 kmole/h
CO_2 in inlet gas	3.73 mole-%
Water in inlet gas	6.71 mole-%
Lean amine temperature	40 ºC
Lean amine pressure	1.1 bar(a)
Lean amine rate	120 000 kmole/h*)
MEA content in lean amine	29 mass-% *)
CO_2 in lean amine	5.5 mass-% *)
Number of stages in absorber	10
Murphree efficiency in absorber	0.25
Rich amine pump pressure	2 bar
Heated rich amine temperature	104.5 ºC *)
Number of stages in stripper	6 (3 + 3)
Murphree efficiency in stripper	1.0
Reflux ratio in stripper	0.3
Reboiler temperature	120 ºC
Lean amine pump pressure	2 bar
Minimum deltaT in heat exch.	10 ºC
*) In first iteration	

Table 2: Specifications for Base Case CO_2 removal

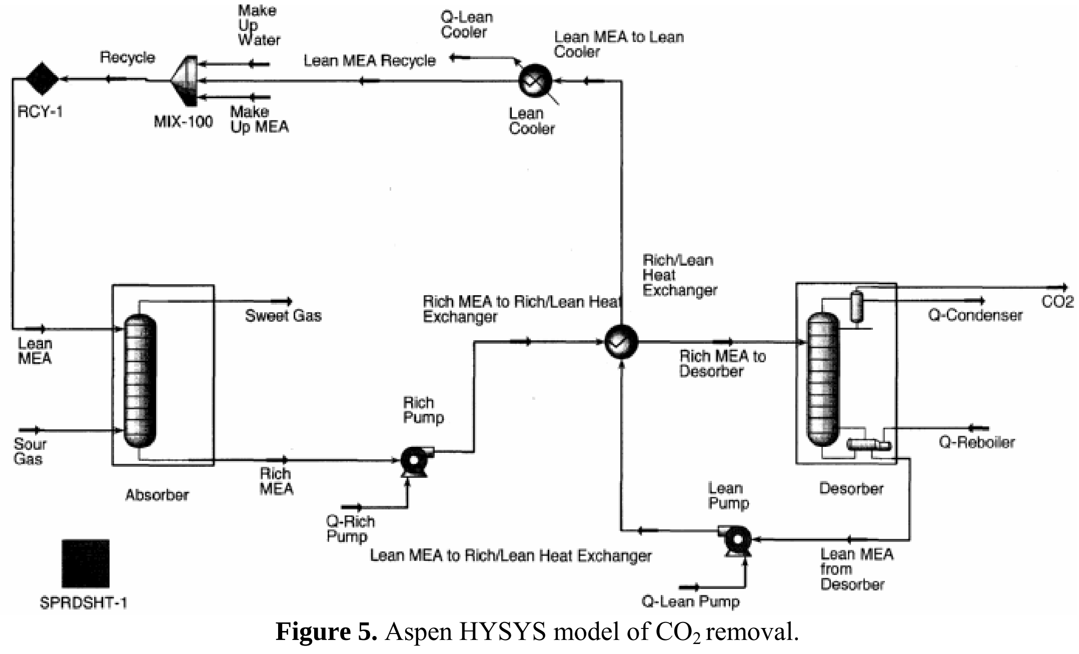


Figure 5. Aspen HYSYS model of CO_2 removal.

85 % CO_2 removal can be specified in the process. The Kent Eisenberg equilibrium model has been compared with the Li-Mather equilibrium model [11]. The CO_2 removal calculated by Aspen HYSYS was reduced from 85 to 82 %, and the heat consumption was reduced from 3.65 to 3.4 MJ/kg CO_2.

3.3 CO_2 removal sensitivity calculations

3.3.1 Variables held constant

The model has been used to evaluate the effects of changing the most important parameters.

In most of the calculations, the CO_2 removal and the stripping heat consumption were calculated, while keeping all the other parameters in table 2 constant. From a calculation viewpoint, this is probably the simplest.

Another possibility had been to keep the % CO_2 removal constant, and calculate the heat duty and the necessary column height. This would give the possibility to optimize the trade-off between operation cost (due to heat consumption) and capital cost (due to column height).

In the cases where the default Inside-Out algorithm did not converge, the Modified Hysim Inside-Out algorithm with adaptive damping was tried to obtain convergence.

3.3.2 Circulation rate

The effect of increased circulation rate, is that the removal grade increases. The results of the simulations are shown in figure 6. A minimum calculated steam consumption is calculated to 3.62 MJ/kg CO_2 removed.

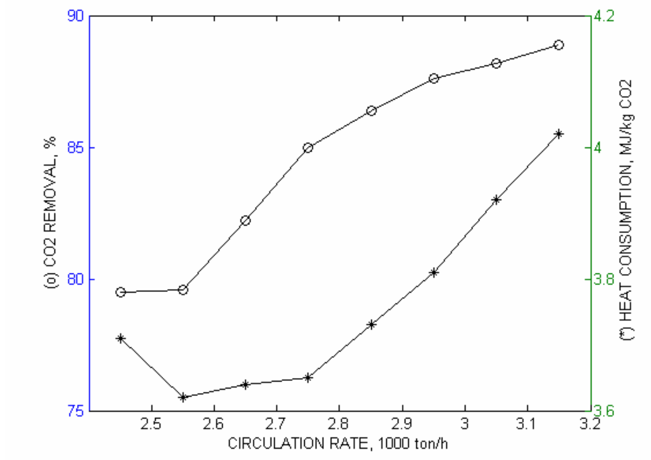


Figure 6: Circulation rate dependence

3.3.3 Number of absorption stages

The height of the absorption column is varied by changing the number of stages. The Murphree stage efficiency for CO_2 is kept constant at 0.25. The height can also be changed by varying the stage efficiency. As expected, removal grade increases and heat requirement decreases with increased column height. The result is shown in figure 7. The calculation diverges using more than 12 stages in the column.

Hysys has calculated estimated E_M (equation 6) for a plate, and the resulting E_M for a plate varied between 0.08 and 0.13.

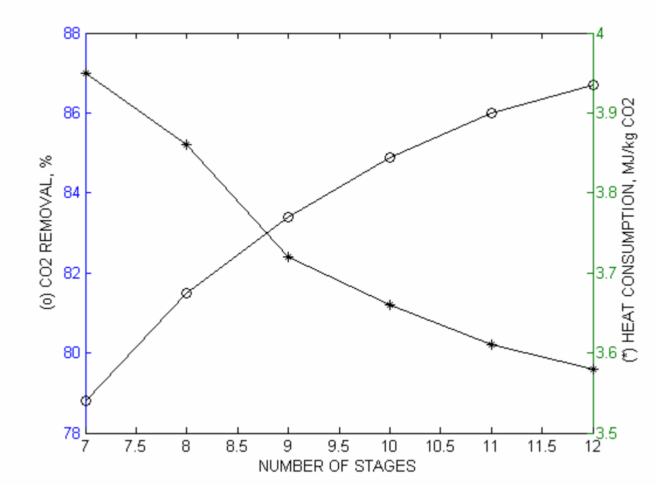


Figure 7: Number of stages dependence

3.3.4 Absorption temperature

An increase in gas and liquid inlet temperature leads to reduced absorption at equilibrium. Simulation results based on a constant stage efficiency are shown in figure 8. In practice, a higher temperature will give a higher absorption and reaction rate, but the equilibrium results will not show this effect.

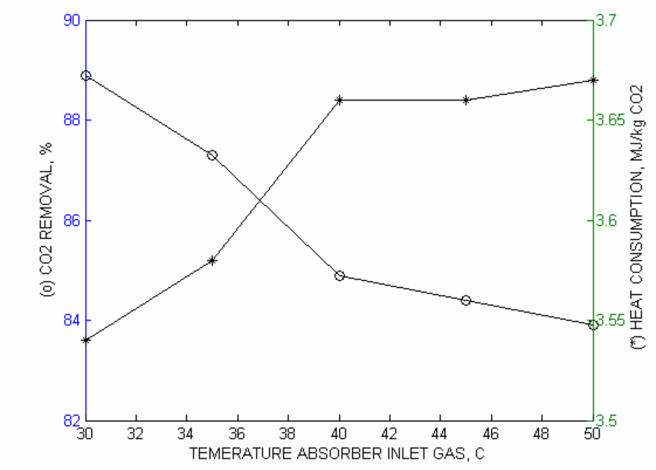


Figure 8: Absorber temperature dependence

3.3.5 Absorption pressure

The absorption pressure is set to atmospheric pressure at the outlet and atmospheric pressure plus pressure drop at the inlet. The pressure inlet in the base case was 1.1 bar. In the case of a pressure drop of 0.5 bar, the % CO_2 removal increased to 87.1 % and the energy consumption was reduced to 3.59 MJ/kg CO_2 removed.

3.3.6 Reboiler temperature

Increased reboiler temperature gives purer amine solution and better CO_2 removal efficiency. However, amine degradation problems arise above 120 ºC. The temperature was varied between 118 and 121 ºC. It was difficult to get converged solutions outside this range. At 121 ºC, an outside range warning was given. The results up to 120 ºC are shown in figure 9.

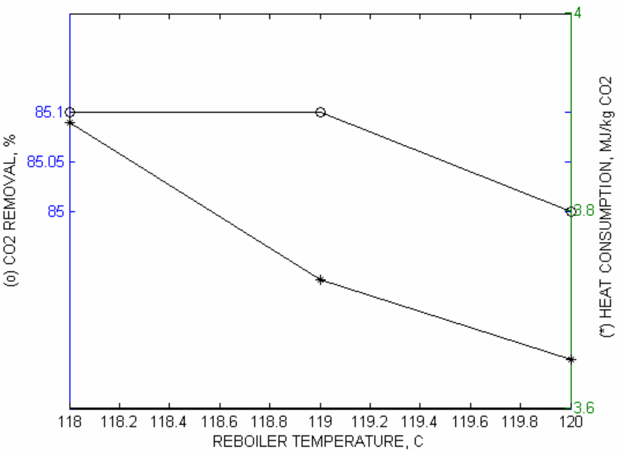


Figure 9: Reboiler temperature dependence

3.3.7 Stripper pressure

The stripper pressure was specified to 2 bar(a) in the calculation. It was very difficult to get a converged solution at other pressures. A solution with a warning (outside range) was achieved with a pressure at 1.9 bar(a).

3.3.8 Minimization of heat consumption

There have been performed many Aspen HYSYS calculations at different conditions. One aim is to reduce the heat duty as much as possible. The lowest heat consumption calculated was 3.39 MW/kg CO_2 removed. The CO_2 removal efficiency was then 93.8 %. This was obtained with gas temperature at 30 ºC, 16 absorption stages, stripper pressure 2.0 bar and reboiler temperature at 200 ºC. At lower temperatures or more absorption stages, the calculation did not converge.

3.4 Power plant efficiency reduction due to CO_2 removal

The steam consumption in the CO_2 removal process is delivered from the gas power plant as shown in figure 2, and the total thermal efficiency is reduced. Steam at 3.5 bar is delivered at 139 ºC. When the base case heat consumption was used, the total efficiency was reduced from 58 % to 53 %. If the efficiency loss due to exhaust gas fans and circulation pumps was included, the resulting total efficiency would be approximately 50 %. If energy for CO_2 compression and condensing was to be delivered from the power plant, the total efficiency would be below 50 %.

The calculated alternative with the lowest duty was with 3.39 MJ/kg CO_2 removed. This gives a reduced reduction in total efficiency of 0.4 % (%-points).

The effect of a possible lower temperature for heat stripping can be calculated. The medium steam pressure can be reduced slightly. The same duty is used, and the total effect should give a slight increase. This effect is so marginal, that it is within the uncertainty of the calculations.

4. Discussion and Conclusion

4.1 Accuracy

With the same specifications, the calculated results vary slightly dependent on initial values. The accuracy in the calculations is normally within 1 % (absolute) in the CO_2 removal and a few percent (absolute) in heat consumption. This accuracy can probably be improved with tighter convergence limits.

The uncertainty due to equilibrium is probably higher. This is indicated by the CO_2 removal changing with 3 % (%-points) when changing equilibrium model from Kent Eisenberg to Li-Mather. The uncertainty in the power plant calculation due to the Peng Robinson equation of state has not been checked, but is expected to be less than the uncertainty in the amine models.

4.2 Evaluation of the simplified process

The calculated CO_2 removal process is a simplified process. Heat losses and some pressure losses are neglected. A real process contains more equipment, pipes and valves, and all this equipment also have heat losses and pressure losses.

A real process with MEA will probably include a water wash section to reduce MEA emissions and a reclaimer unit to recover MEA from thermally de-generated MEA. These additional units will increase the energy consumption.

There are possibilities to reduce the energy consumption in the process by different stream configurations (eg. split stream). These possibilities will however normally increase the investment.

The calculated heat consumption of 3.65 MJ/kg CO_2 is regarded as realistic. This is slightly lower than normally found in literature, eg. Desideri [1] which has a list of references with values mostly above 4.0 MJ/kg CO_2. If water wash is included in the highest value, this can explain the difference. The lowest calculated value of 3.39 MJ/kg CO_2 is regarded as an optimistic value. It might be regarded as a potential value for improvements. The cost optimum heat consumption will probably be a trade-off between investment and operational cost.

4.3 Convergence problems

There are many problems with convergence in these calculations. The problems normally occur in the absorption or stripping column. One problem is numerical. It is found that the Modified Hysim Inside-out algorithm with adaptive damping gives the best convergence.

If there are too many stages specified in the columns, they tend to diverge. That is traditional for column stage calculations in typical process simulation tools.

There is also a problem with the range for the model. The Kent Eisenberg model is limited to below 30 weight-% MEA and below 120 ºC. The simulation is calculated, but the program gives a warning.

4.4 Parameter variations

Different parameters have been varied to calculate the effects and to make a tool for optimizing the process.

The only parameter that is varied so that an optimum is found, is the recirculation rate. In figure 6 it is shown that a minimum heat consumption (of 3.62 MJ/CO_2) is achieved at a lean amine circulation rate of 2550 ton/h.

When the number of stages is increased, the performance of the process increases, but the calculation tends to diverge. However, when the column diverges due to too many stages, the % CO_2 removal and heat consumption is probably close to a maximum.

A reduction in absorption temperature leads to improved performance according to figure 8. According to literature, a temperature of about 40 ºC is recommended. At lower temperature, the reaction rates will be too slow. The calculations have been performed with constant column stage efficiencies of 0.25. The original Aspen HYSYS calculates efficiencies are temperature dependent. A temperature dependent stage efficiency is probably necessary to find an optimum absorption temperature.

It is not difficult to calculate the effects of absorption pressure drop. There is a trade-off between CO_2 removal efficiency and cost of column height and pressure drop.

Increased reboiler temperature gives purer amine solution and better performance. The optimum temperature is a trade-off between improved CO_2 removal and degradation of the amine solution. Aspen HYSYS limits the calculation to 120 ºC as in the base case. The resulting lean loading (mole CO_2/mole amine) in the liquid to the absorption column is 0.26. This is close to 0.25 as reported as an optimum by Alie [3].

The stripper pressure is an adjustable parameter. It is difficult to get a converged calculation with a stripper pressure far away from 2 bar with 120 ºC. According to Freguia [2], the pressure should be between 1.5 and 2 bar. It is not obvious whether the convergence difficulties are due to physical limitations, or if it is a numerical problem. The pressure must be consistent with the pressure of the amine solution leaving the reboiler. The problems of limited range of reboiler temperatures and stripper pressures are probably related.

4.5 Aspen HYSYS compared to other tools

There are other tools to simulate such processes. Other commercial process simulation tools like ProVision, ProMax or Aspen Plus can be used to simulate the process in a similar way. Aspen Plus also has the possibility to calculate rate expressions on an ideal mixing stage (simulating a column plate). This has the advantage of taking into account the reaction rates for different reactions simultaneously. It is possible (but difficult) to include rate expressions of absorption (transport of CO_2 from the gas to the liquid phase) in such a model. It is also a question whether this kind of a mixing stage model is a good model for continuous countercurrent operation as in structured packing.

The presented Aspen HYSYS model is based on a specified Murphree efficiency for each stage (or height of packing). It is possible to make this efficiency a function of rate expressions for the absorption rate and the reaction rates.

It is of course possible to simulate CO_2 removal processes without using commercial process simulation programs. It is however necessary to include at least one reliable and robust equilibrium calculation model and one robust column model. It is difficult to compete with the commercial process simulation programs in these two matters. The commercial programs also normally have very good input and output facilities.

A problem with the commercial process simulation programs, at least from an academic point of view, is that some of the models of interest are not documented accurately.

4.6 Further development

There are available calculation models for the design of CO_2 removal plants based on MEA. The largest uncertainty is probably connected to the absorber stage efficiency. In the case of a structured packing column, this gives a large uncertainty in the necessary packing height. An improvement here will be of great value.

In the case of other amines than MEA, the limitation is in the equilibrium model or in the uncertainty in the equilibrium model parameters.

In the case of mixed amines, the importance of the reaction rates increases. Such models are probably best treated in a rate based model like RateFrac in Aspen Plus. It is however possible that a practical approach when using Aspen HYSYS, is to perform estimation of Murphree efficiency outside the process simulation tool.

4.7 Conclusion

The CO_2 removal model developed in Aspen HYSYS is useful for evaluating the effects of changing amine circulation rate, absorption column height, absorption temperature and reboiler temperature. The reduced total efficiency in a gas based power plant can be calculated.

The developed models can be developed further to improve amine absorption processes for CO_2 removal. The combination of one model for a power plant and another for CO_2 removal can be useful for total energy optimization and total cost optimization.

5. Acknowledgements

These models are the result of much trial and error work done by my students during many years. Thanks to all of you!

References

[1] Desideri, U., Alberto, P. (1999). Performance modelling of a carbon dioxide removal system for power plants, Energy Conversion & Management 40, 1899-1915.

[2] Freguia, S., Rochelle, G.T. (2003). Modelling of CO_2 Capture by Aqueous Monoethanolamine, AIChE Journal, 49, 1676-1686.

[3] Alie, C., Backham, P., Croiset, E., Douglas, P.L. (2005). Simulation of CO_2 capture using MEA scrubbing: a flowsheet decomposition method, Energy Conversion & Management 46, 475-487.

[4] Tobiesen, F.A., Svendsen, H.F., Hoff, K.A. (2005). Desorber energy consumption amine based absorption plants. International Journal of Green Energy, 2, 201-215.

[5] Aspen Plus, AspenTech, Cambridge Mass., 2006.

[6] Aspen HYSYS 2004.2, AspenTech, Calgary 2007.

[7] Danckwerts, P.V., Sharma, M.M. (1966). The absorption of Carbon Dioxide into solutions of alkalis and amines, The Chemical Engineer, oct. 244-280.

[8] Versteeg, G.F., Van Dijck, L.A.J., Van Swaaij, W.P.M. (1996). On the kinetics between CO_2 and alkanolamines both in aqueous and non-aqueous solutions. An overview, Chem. Eng. Comm. 144, 113-158.

[9] Peng, D., Robinson, D.B. (1976). A New Two-Constant Equation of State, Ind. Eng. Chem. Fundam., 15 No.1, 59-64.

[10] Kent, R.L., Eisenberg, B. (1976). Better data for Amine Treating, Hydrocarbon Processing, 55, No.2, 87-90.

[11] Lee, I.J., Otto, F.D., Mather, A.E. (1975). Solubility of Mixtures of Carbon Dioxide and Hydrogen Sulfide in 5.0 N Monoethanolamine Solution, J. Chem. Eng. Data, 20, 161-163.

[12] Tomkej, R.A., Otto, F.D. (1986). Improved Design of Amine Treating Units by Simulation using Personal Computers, World Congress III of Chemical Engineering, Tokyo, Sept. 21-25.

[13] Vamraak, K. (2004). Energiforbruk ved CO_2-fjerning fra gasskrafverk, Master Thesis, Telemark University College.

[14] Moholt, B. (2005). Simulering av CO_2-fjerning med aminer, Master Thesis, Telemark University College.

[15] Amundsen, T. (2007). CO_2-renseanlegg i Aspen HYSYS, Master Project, Telemark University College.

SIMS 2007

Modelling of Electrical Faults in Induction Machines Using Modelica®

Dietmar Winkler Clemens Gühmann
Technische Universität Berlin
Department of Electronic Measurement and Diagnostic Technology
{Dietmar.Winkler|Clemens.Guehmann}@TU-Berlin.de

Abstract

This paper presents the modelling of electric faults in induction machines using Modelica® as modelling language. The primal application for such fault simulations is the ever increasing application of electric machines in cars. In this paper you will find a short introduction of how to model an electrical machine and their most common failures. The modelling is done in the phases reference frame rather than in a field-oriented reference frame. The reason for this is given in this paper. The simulation models work and simulation results are given. The verification of the simulation results with a motor test-bench is still an outstanding task at which we are currently working on.

Keywords: *Modelica, electrical machines, modelling, fault simulation, field-oriented control*

1 Introduction

It has been ten years now since Toyota released its first version Prius I. Since then the number of hybrid electric vehicles has been continuously growing which in turn means that a new ever growing market for electric drives has emerged. This new marked pushed the development efforts in various field of engineering. For one, different control strategies for the electric drive are needed in order to use the electric power as efficient as possible. Thus being able to save as much petrol as possible when driving from A to B [1].

The application of electric drives in cars also raised various safety-related issues. There lies a great interest in investigating the possible impact of failures of electric machines on the cars driving behaviour. Because of the (in most hybrid architectures) rigid connection of the electric drive with the cars drivetrain, electrical failures result in a direct mechanical impact via torque changes. Depending on the type of hybridisation (e.g., micro-, mild-, or full-hybrid) the effect might vary from an inconvenient torque loss to a full traction-loss. The impact can even become quite critical if we consider the use of in-wheel-motors which might even lead to accidents.

With the help of modelling and simulation of electrical machines some of the possible dangers can be investigated and appropriate control strategies can be developed to minimise the failure-impacts. This work will show the development of induction machine models using the object-oriented modelling language *Modelica®* (for more on modelling with Modelica® see [2, 3].

2 Mathematical fundamentals of electrical induction machines

The classic approach to describe an electrical machine is by the use of the so called *equivalent phase circuit diagram.* In the following the machine equations shall be explained taking an asynchronous induction machine as an example. There are however much similarities to the synchronous machine which shall not be investigated further in this short paper (see for example [4]). Figure 1 represents the equivalent circuit for one phase of the electrical machine. It consists of resistances representing the ohmic losses in the field windings (R_s and R_r) and the stray inductance ($L_{s\sigma}$ and $L_{r\sigma}$) in the stator and the rotor. The rotor and stator are coupled via the mutual inductance (L_m).

The standard machine equations for the rotor and sta-

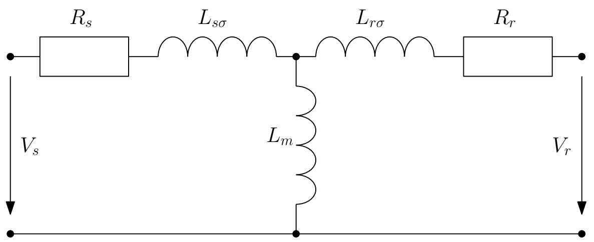


Figure 1: Equivalent circuit diagram of electrical induction machine

tor voltages can be written as:

$$\vec{v}_s = R_s \vec{i}_s + L_{s\sigma}\frac{\mathrm{d}\vec{i}_s}{\mathrm{d}t} + \frac{\mathrm{d}\vec{\Psi}_{ms}}{\mathrm{d}t} \tag{1}$$

$$\vec{v}_r = R_r \vec{i}_r + L_{r\sigma}\frac{\mathrm{d}\vec{i}_r}{\mathrm{d}t} + \frac{\mathrm{d}\vec{\Psi}_{mr}}{\mathrm{d}t} \tag{2}$$

where Ψ_{mx} are the flux vectors and defined as

$$\vec{\Psi}_{ms} = \mathbf{L_{ss}} \cdot \vec{i}_s + \mathbf{L_{sr}}(\gamma) \cdot \vec{i}_r \tag{3}$$

$$\vec{\Psi}_{mr} = \mathbf{L_{rs}}(\gamma) \cdot \vec{i}_s + \mathbf{L_{rr}} \cdot \vec{i}_r \tag{4}$$

In the equations (3) and (4) you can see the four inductance matrices. Each of these have the dimension $(m \times m)$ where m stands for the number of phases (three in most cases). $\mathbf{L_{ss}}$ and $\mathbf{L_{rr}}$ represent the self-inductance matrices for the stator and the rotor, respectively.

$$\mathbf{L_{ss}} = \begin{bmatrix} L_m & L_{sm} & L_{sm} \\ L_{sm} & L_m & L_{sm} \\ L_{sm} & L_{sm} & L_m \end{bmatrix} \tag{5}$$

and

$$\mathbf{L_{rr}} = \begin{bmatrix} L_m & L_{rm} & L_{rm} \\ L_{rm} & L_m & L_{rm} \\ L_{rm} & L_{rm} & L_m \end{bmatrix} \tag{6}$$

If the the machine is symmetrically built than the matrices can be simplified further by providing the equations:

$$L_{sm} = L_{rm} = L_m \cdot \cos(\frac{2}{m}\pi)$$

This also means that stator and rotor self-inductance would be identically.

We still got the two matrices $\mathbf{L_{sr}}(\gamma)$ and $\mathbf{L_{rs}}(\gamma)$ left. These represent the mutual coupling between the stator and the rotor side and vice versa. Unfortunately the coupling effect depends on the rotor position γ. For a three phase machine ($m = 3$) this gives:

$$\mathbf{L_{sr}}(\gamma) = [\mathbf{L_{rs}}(\gamma)]^T \tag{7}$$

$$= \begin{bmatrix} \cos(\gamma) & \cos(\gamma + \frac{2}{3}\pi) & \cos(\gamma - \frac{2}{3}\pi) \\ \cos(\gamma - \frac{2}{3}\pi) & \cos(\gamma) & \cos(\gamma + \frac{2}{3}\pi) \\ \cos(\gamma + \frac{2}{3}\pi) & \cos(\gamma - \frac{2}{3}\pi) & \cos(\gamma) \end{bmatrix} \tag{8}$$

2.1 Field-oriented control (FOC)

At this stage normally the field-oriented control is introduced. The idea of the field-oriented control is to transform the three (or m) phases into an orthogonal system consisting only of two resulting space vectors (dq-frame). Depending on the application these vectors can then be aligned for example to the stator, or the rotor flux. This gives the benefit that instead of having to calculate three (or m) time variant interdependent stator currents, for example, only two currents have to be calculated. These can also be controlled independently and are constant during steady state operations. This technique can improve the simulation speed considerably. Since all the transformation matrices are quite complex and are not the focus of this paper, the field-oriented control is not described any further. The interested reader can find various papers and books on this topic. Two of theses are for example [4] and [5].

2.2 Reason for not using FOC

As much as the field-oriented control can improve the simulation speed, the simulation models itself become quite abstract. When it comes to simulating some physical errors of the machine in order to gain some information on the impact such an error would have, field-oriented models are not the first choice.

Example: Let's take an example to explain the drawback of field-oriented control for error simulation. In an induction machine we have got a short-circuit of one or more stator windings. This would result in a different stator resistance and inductance. Since in our field-oriented model only consists of the transformed and combined components we than have to also transform the stator error from the (let's say) three phase system to the orthogonal system. But this also means that our machine model is not symmetrical anymore and we have to take care that all previous transformations from the three phase system to the orthogonal system are still valid (some actually require symmetrical systems as a prerequisite). Finally we end up with an orthogonal model which is unsymmetrical and therefor has time depending variables even for steady state operations. So for this kind of error simulation the benefit of reduced simulations times is not present any more and moreover we still have the abstract representation of the error values in the orthogonal system.

Because of this reason, in this paper the modelling is done in the m-phase domain rather then the orthogonal system. Mind you, the control of the electrical machine would still be done field-oriented. Just the model is not transformed. This also corresponds with the real life where we got an field-oriented controller giving control signals to an inverter-bridge which in turn then provides the m-phase voltages and currents.

3 Machine models in Modelica®

The free available *Modelica Standard Library* (MSL[6]) already contains basic machine models[7]. Unfortunately these models are only suitable for modelling symmetrical electrical machines by using dq-frame-oriented parameters. An electrical fault in one of the phase windings (stator or rotor) leads however to unbalanced machine equations which are not used by the basic machine models of the MSL. So different models have to be developed to take the unbalanced behaviour into account. This leads to a more complicated model than the normally used field-oriented dq-model. Such models of unbalanced electrical machines are presently part of the ongoing development of a free and open-source Modelica® library called `freeFOClib`[8] (short for free Field-Oriented Control library) at our department.

When developing the new models for error simulation it was desirable to use as much from the standard components from the MSL in order to keep the library as slim as possible. For the modelling of an asynchronous induction machine with squirrel-cage in a three-phase system all components which are not rotor-angle dependent could be modelled using standard components from the `MultiPhase` sub-library of the MSL. Also when no errors in the squirrel-cage part are introduced than the standard field-oriented model of the squirrel cage can be used. In Figure 2 a screen shot from the graphical representation in Dymola (modelling and simulation tool of Dynasim AB, Sweden) of such a system is depicted.

All the rotor angle dependent matrices are modelled in Modelica code within the model `airGap3Phase`.

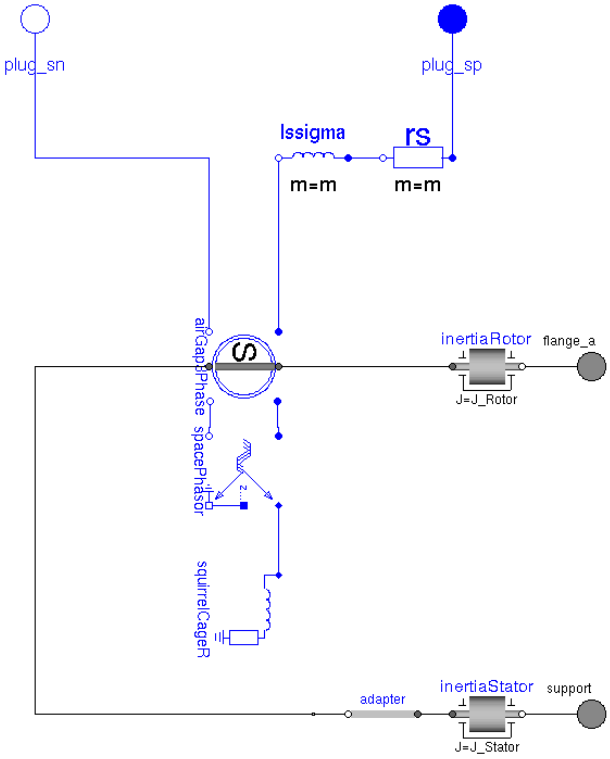


Figure 2: Graphical model representation of a 3-phase asynchronous machine with squirrel cage

It follows an excerpt of the source code of the `airGap3Phase` model which is used to build the matrices in equation (5) trough to equation (8):

```
Lss = fill(fill(Lsm, 3), 3) -
diagonal(fill(Lsm - Lm, 3));
Lrr = Lss;
Lsr = Lm .* {{cos(gamma),cos(gamma -
2/3*pi),cos(gamma + 2/3*pi)},
      {cos(gamma +
2/3*pi),cos(gamma),cos(gamma - 2/3*pi)},
             {cos(gamma -
2/3*pi),cos(gamma + 2/3*pi),cos(gamma)}};
Lrs = transpose(Lsr);
```

The mutual fluxes and the resulting voltages are then calculated by:

```
 // mutual fluxes:
psi_ms = Lss*i_s + Lsr*i_r;
psi_mr = Lrs*i_s + Lrr*i_r;
// mutual voltages:
v_ms = der(psi_ms);
v_mr = der(psi_mr);
```

4 Fault simulation

After a brief introduction of how the the machines are modelled in a *m*-phase system we will now come to that actual application, i.e. the fault simulation. At first let us see which different fault scenarios are likely
Typical examples are:

- open-circuit of a stator phase (e.g., a connecting cable is broken)
- short-circuit phase to ground (e.g., insulation failure because of mechanical damage)
- short-circuit of one or more phase windings (e.g., insulation failure because of thermal stress within the stator or rotor)

Theses faults would then again have an influence on the:

- mechanic torque (e.g. which has a direct impact on the drive train behaviour in a hybrid electric vehicle)
- emerging voltages and currents (which might damage the power electronic devices supplying the electric motor)

In order to generate these faults we need an interface to our model to trigger the fault. The first two faults (open-circuit and short circuit) can be achieved by introducing an electrical switch which then closes or opens the corresponding phase connection. Since our model is modelled in the physical phase domain the implementation of this is pretty straight forward.
However the third fault (short-circuit of one or more phase windings) requires a change in the air-gap model. To be more specific, the way the inductance matrices are built has be altered. In our we have chosen to use an input variable into which the faulty inductance matrices can be fed. The switch from a normal to a faulty inductance is realised by a time-triggered ramp function. The slope of the inductance change can be configured as well as the switching time. It is also possible to reduce the normal inductance value by a specific amount (say 10% for example). Furthermore in a real application there would be some kind of guidance control to avoid any damage to the machine or inverter. Very simply speaking, this would be some kind of fuse. In the fault-models of the `freeFOClib` also fuses are implemented.

4.1 Open-circuit fault

Here now comes an example simulation of a synchronous machine having a open-circuit fault. This could occur if a connecting cable comes loose. In Figure 3 you can see the three phase currents and the mechanical torque depicted over time just before and after the connection of one phase was opened at the time of two seconds.

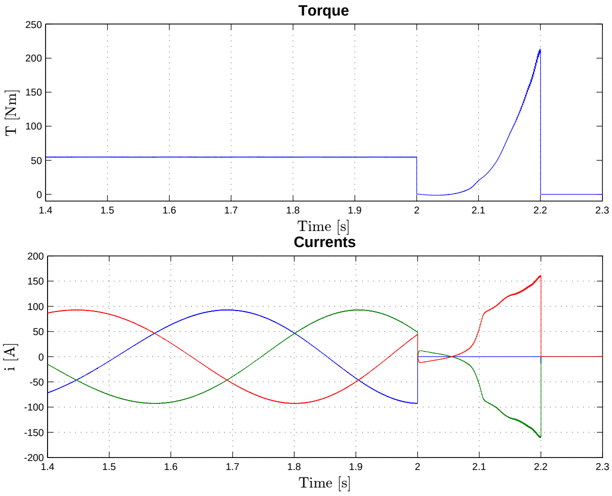


Figure 3: Fault scenario: open-circuit of one stator phase

As you can see, the torque drops instantaneously and but the controller try to keep the torque output until at some point the fuses for the remaining phase currents get triggered.

4.2 Short-circuit fault

This kind of fault can occur when the insulation of the connection cables is somehow damaged. In Figure 4 again you can see the three phase currents and the mechanical torque depicted over time just before and after one phase winding connects to ground at the time of two seconds.
Again the controller tries to keep the torque at a constant level but can only do so for a certain amount of time until the the fuse finally gets triggered.

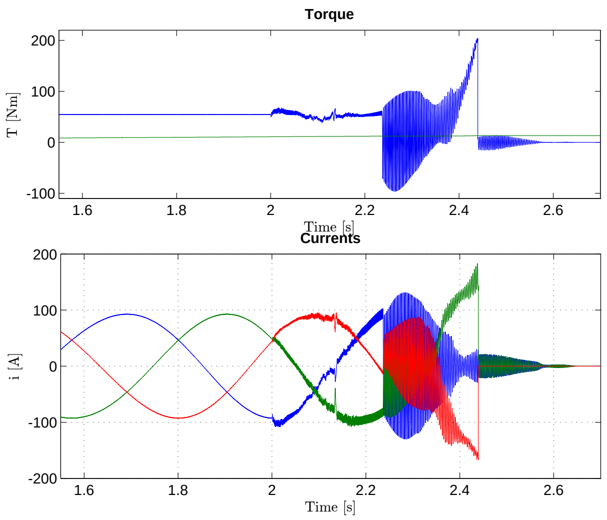


Figure 4: Fault scenario: short-circuit of one stator phase to ground

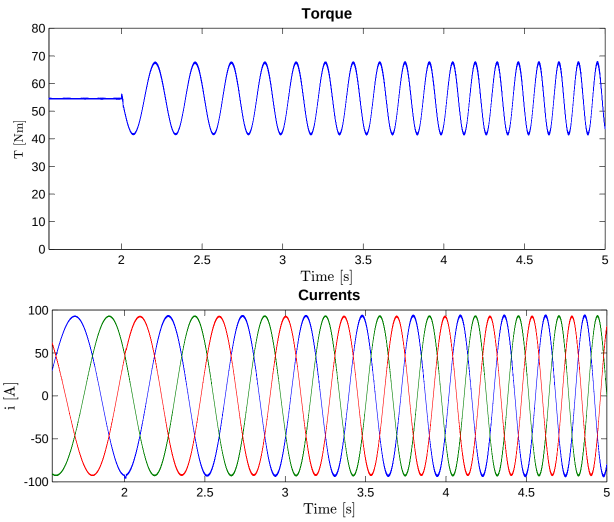


Figure 5: Fault scenario: short-circuit of phase windings

4.3 Short-circuit of windings

This kind of fault can occur when the insulation between the phase windings is damaged (e.g., overtemperature, overload, etc.). This behaviour is modelled by reducing the inductance abruptly by 20 percent. In Figure 5 you can see the three phase currents and the mechanical torque over time just before and after the connection of one phase was opened at the time of two seconds.

As you can see this time the change in the currents is not that drastic. However since the field-oriented control now calculates the wrong control values the torque starts to oscillate.

5 Conclusions

In this paper a the modelling and simulation of electric faults in electrical machines using Modelica® was presented. The reader could get a brief introduction into the topic of modelling electrical induction machines in general. By using the object-oriented language Modelica® it is possible to keep the new models compact by making use of the inheritance mechanism in connection with the free available Modelica Standard Library. Extensive usage of inheritance also means that the maintenance is simpler than if one bit of code is used in several places. We presented some ways of how to implement the different fault scenarios and also included some simulation results.

6 Future work

The developed models of unbalanced electrical machines are going to be a part of a free and open-source Modelica® library called `freeFOClib`[8] (short for free Field-Oriented Control library). The first official release is due in spring of 2008. The `freeFOClib` will also contain all components which are necessary for a field-oriented control (hence the name). The idea is to provide a Modelica® library which gives the possibility to model a modern electrical drive system and still leaves the user the freedom to look at the underlying code and perhaps even contribute improvements to `freeFOClib`.

A pure estimate of the simulated values might give some clues on the general behaviour during faults. However to actually use the simulation to gain useful information (e.g., for programming a controller for the power electronics) we need more than just estimates. So the simulation model has to be verified by doing real measurements using a real electric motor. At the moment we setting up a test-bench system consisting of a asynchronous induction machine of a currently available hybrid electric vehicle and an electrical load machine. Although we hoped to be able to include some measurement results from the test-bench already, unfortunately it was not feasible for this paper. Still we are working hard on the further development of the library and the verification with the test-bench system. For any news on the `freeFOClib` library see `www.freefoclib.org`. On that site also a mailinglist is available to keep you automatically up to date.

References

[1] D. Winkler and C. Gühmann, "Hardware-in-the-Loop simulation of a hybrid electric vehicle using Modelica/Dymola," in *The 22nd International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium & Exposition* (E. . Secretariat, ed.), (Yokohama, Japan), pp. 1054–1063, EVS, Japan Automobile Research Institute, 23-28 October 2006.

[2] P. A. Fritzson, *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. No. 0-471-471631, John Wiley & Sons, Januar 2004.

[3] M. Tiller, *Introduction to Physical Modeling with Modelica*. Kluwer International Series in Engineering & Computer Science, Kluwer Academic Publishers, 2001.

[4] D. Novotny and T. Lipo, *Vector Control and Dynamics of AC Drives*. Clarendon Press Oxford, 1996.

[5] W. Leonhard, *Control of Electrical Drives*. Springer, 3rd ed., 2001.

[6] Modelica Association, *Modelica® - Free library from the Modelica Association*, version 2.2.2 ed., 31st August 2007.

[7] C. Kral and A. Haumer, "Modelica libraries for dc machines, three phase and polyphase machines," in *Proceedings of the 4th International Modelica Conference* (G. Schmitz, ed.), pp. 549–558, Modelica Association, March 7-8 2005.

[8] D. Winkler, E. Bakhach, F. Döring, and S. Rinderer, "`freeFOClib` - A free Field-Oriented Control library for Modelica." unreleased, see `www.freefoclib.org` for any news. Due in, Spring 2008.

Reduction of Finite Element Models of Complex Mechanical Components

Håkan Jakobsson
Research Assistant
hakan.jakobsson@math.umu.se

Mats G. Larson
Professor Applied Mathematics
mats.larson@math.umu.se

Gabriel Granåsen
Research Assistant

Department of Mathematics and Mathematical Statistics,
Umeå University, S-901 87 Umeå, Sweden

Abstract

There is an increasing interest in using detailed finite element models of individual components in multibody simulation with detailed contact analysis. Due to the large amount of degrees of freedoms in such models we need to develop reduced models with fewer degrees of freedoms that still capture the flexible body dynamics with sufficient accuracy. In this paper we investigate a static load case on a geometrically complex mechanical component. We use modal basis functions to describe the global deformations of the component and certain Krylov-modes to capture a localized load and to provide accurate coupling to the modal basis functions. We investigate how the overall accuracy depends on the number of modal basis functions and the number of Krylov-modes.

1. Introduction

Dynamics simulation involving complex finite element models is computationally demanding despite the rapid development of computational resources. To reduce computational cost, model reduction techniques are commonly employed. There are several classical model reduction techniques. See for instance [1] for a detailed description on the use of model reduction in structural dynamics. See also [2] for an account of model reduction in context of multibody dynamics.

Commonly reduction techniques are based on the assumption that a deformable object in a simulation is affected by boundary loads at a fixed number of specified points only, where the object is connected to its surroundings through so called coupling elements, e.g., springs, dampers and joints. In the general multibody setting this may however not be the case. Here objects may in addition be affected by contact forces originating from dynamic collisions with other objects. These collisions may occur at arbitrary points, and the contact forces may be moving, e.g., a rolling bearing simulation.

This paper is the first in a series documenting a research effort towards finding new model reduction techniques that may be applied in the general multibody setting with geometrically complex deformable objects. The initial aim is to identify suitable correction modes that can be computed on the fly, or in a post processing step, in order to improve the convergence of a modal based reduction in a continuous manner. To this end we investigate in this paper static deformation of a geometrically complex elastic body with a localized boundary load. We use eigenmodes to capture low frequency global effects, and compute certain Krylov-modes to capture high frequency effects stemming from the boundary load. The Krylov-modes we generate are L_2-orthogonal to the eigenmodes per construction, thus removing the computational overhead associated with orthogonalization against these. We illustrate the effect of the Krylov-modes in a numerical study. N.b. Krylov-modes are commonly used in Krylov subspace methods for the iterative solution of large, sparse, linear systems of equations. Typically such methods seek approximate solutions to $m \times m$ linear system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ in subspaces $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{b}) = \text{span}\,\{\boldsymbol{A}^{i-1}\boldsymbol{b}\}_{i=1}^{k}$, of $\mathbf{R}^m$ cf. [3]. These are natural subspaces to seek approximate solutions in. In fact, if $\boldsymbol{A}$ is invertible it follows that $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b} \in \mathcal{K}_n(\boldsymbol{A}, \boldsymbol{b})$, where n is the degree of the minimal polynomial of $\boldsymbol{A}$.

The paper is organized as follows. In Section 2 we outline the theory associated with modal based model reduction techniques and define the specific Krylov-modes we use in our computations. In Section 3 we perform a numerical study and discuss some aspects of mesh refinement. In Section 4 we summarize the results and outline the continuation of these works.

2. Theoretical Investigation

2.1 The Equations of Linear Elasticity

We consider the equations of linear elasticity in two dimensions: find the vector displacement $\boldsymbol{u} = [u_i]_{i=1}^{2}$ and the symmetric stress tensor $\boldsymbol{\sigma} = [\sigma_{ij}]_{i,j=1}^2$ such that

$$\begin{aligned} -\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}) &= \mathbf{0}, \quad x \in \Omega, \\ \boldsymbol{\sigma} &= 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) + \lambda(\nabla \cdot \boldsymbol{u})\boldsymbol{I} \quad x \in \Omega, \\ \boldsymbol{u} &= \mathbf{0}, \quad s \in \Gamma_D, \\ \boldsymbol{n} \cdot \boldsymbol{\sigma} &= \boldsymbol{h}, \quad s \in \Gamma_N, \end{aligned} \tag{1}$$

where Ω is a closed subset of $\mathbf{R}^2$, Γ_D and Γ_N are parts of the boundary, $\partial\Omega$, such that $\Gamma_D \cup \Gamma_N = \partial\Omega$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Furthermore, $\boldsymbol{\varepsilon}(\boldsymbol{u}) = [\varepsilon_{ij}]_{i,j=1}^2$ is the strain tensor with components

$$\varepsilon_{ij}(u) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{2}$$

$\nabla \cdot \boldsymbol{\sigma} = \left[\sum_{j=1}^2 \partial\sigma_{ij}/\partial x_j\right]_{i=1}^2$, $\boldsymbol{I} = [\delta_{ij}]_{i,j=1}^2$ with δ_{ij} the Kronecker delta, $\boldsymbol{h}$ is a given traction load, and $\boldsymbol{n}$ is an outward pointing unit normal. Lastly, λ and μ are the Lamé parameters satisfying $0 < \mu_1 < \mu < \mu_2$ and $0 < \lambda < \infty$. In the case of plane strain we have $\lambda = E\nu/((1+\nu)(1-2\nu))$ and $\mu = E/(2(1+\nu))$, where E is Young's elastic modulus and ν Poisson's ratio.

2.2 The Finite Element Method

The weak formulation of (1) reads: Find $\boldsymbol{u} \in \mathcal{V}_{\mathbf{0}} = \{\boldsymbol{v} \in H^1[\Omega]^2 \, : \, \boldsymbol{v}|_{\Gamma_D} = \mathbf{0}\}$ such that

$$a(\boldsymbol{u}, \boldsymbol{v}) = b(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in \mathcal{V}_{\mathbf{0}}, \tag{3}$$

where the linear forms $a(\cdot,\cdot)$, and $b(\cdot)$ are defined by

$$\begin{aligned} a(\boldsymbol{v}, \boldsymbol{w}) = 2(\mu\boldsymbol{\varepsilon}(\boldsymbol{v}) : \boldsymbol{\varepsilon}(\boldsymbol{w})) \\ + (\lambda\nabla \cdot \boldsymbol{v}, \nabla \cdot \boldsymbol{w}), \end{aligned} \tag{4}$$

$$b(\boldsymbol{v}) = (\boldsymbol{h}, \boldsymbol{v}). \tag{5}$$

In order to set up the finite element method, let $\mathcal{T}^h = \{T\}$ be subdivision of Ω into triangular elements T, and let $\mathcal{V}_{h,\mathbf{0}} = \{\boldsymbol{v} \in H^1(\Omega)^2 \, : \, \boldsymbol{v}|_T \in \mathbf{P}(T)^k, \boldsymbol{v}|_{\Gamma_D} = \mathbf{0}\}$, with $k = 1$ or $k = 2$, be an N-dimensional subspace to $[H^1(\Omega)]^2$. In the finite element method we seek an approximate solution $\boldsymbol{u}_h \in \mathcal{V}_{h,\mathbf{0}}$ such that

$$a(\boldsymbol{u}_h, \boldsymbol{v}) = b(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in \mathcal{V}_{h,\mathbf{0}}. \tag{6}$$

Given a basis $\{\boldsymbol{\varphi}_j\}_{j=1}^N$ to $\mathcal{V}_{h,\mathbf{0}}$, this problem is equivalent to an $N \times N$ linear system of equations

$$\boldsymbol{K}\hat{\boldsymbol{u}}_h = \hat{\boldsymbol{b}}, \tag{7}$$

with elements

$$K_{ij} = a(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i), \quad i, j = 1, \ldots, N, \tag{8}$$

$$b_i = b(\boldsymbol{\varphi}_i), \quad i = 1, \ldots, N. \tag{9}$$

2.3 The Model Reduction

In 3D dynamics simulation with flexible bodies accurate approximation of deformation and stresses comes with a significant computational cost. The system of equations corresponding to (7) is likely to contain a very large number of degrees of freedoms and require extensive computational resources to solve. To reduce the computational cost model reduction techniques are commonly employed.

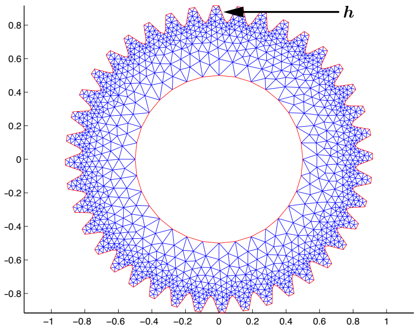


Figure 1: Localized force acting on domain boundary.

We investigate a static two-dimensional example. Consider therefore the situation illustrated in Figure 1, where a localized load acts on the boundary of a geometrically complex domain, a two-dimensional gear wheel with 40 teeth. Even in 2D, the system (7) will contain a significant number of degrees of freedoms and a model reduction is justified. This means that we must try to find a subspace $\mathcal{V}_{h,R} \subset \mathcal{V}_{h,\mathbf{0}}$ such that $n = \dim \mathcal{V}_{h,R} \ll \dim \mathcal{V}_{h,\mathbf{0}}$ and such that $\|\boldsymbol{u}_{h,R} - \boldsymbol{u}_h\|_{H^1(\Omega)^2}$ is small, where $\boldsymbol{u}_{h,R} \in \mathcal{V}_{h,R}$ satisfies

$$a(\boldsymbol{u}_{h,R}, \boldsymbol{v}) = b(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in \mathcal{V}_{h,R}. \tag{10}$$

It is obvious that it is hard to construct a general such space $\mathcal{V}_{h,R}$. If we however exploit our knowledge of $\boldsymbol{h}$ we can tune $\mathcal{V}_{h,R}$ according to the specific load case.

Therefore, let $\mathcal{V}_{h,R} = \mathcal{V}_{h,TM} \oplus \mathcal{V}_{h,K}$, where $\mathcal{V}_{h,TM} \subset \mathcal{V}_{h,\mathbf{0}}$ is a space spanned by a subset of the eigenvectors of a, and $\mathcal{V}_{h,K} \subset \mathcal{V}_{h,\mathbf{0}}$ a Krylov-space. The space $\mathcal{V}_{h,TM}$ is chosen as to capture global deformations, and the space $\mathcal{V}_{h,K}$ is chosen to capture localized high-frequency effects in a neighborhood around the applied boundary load.

We first consider the space $\mathcal{V}_{h,TM}$. Since a is a symmetric bilinear form on $\mathcal{V}_{h,\mathbf{0}}$, the eigenmodes corresponding to the eigenvalue problem

$$a(\boldsymbol{e}_h, \boldsymbol{v}) = \lambda \int_\Omega \boldsymbol{e}_h \boldsymbol{v} \, d\boldsymbol{x}, \quad \forall \boldsymbol{v} \in \mathcal{V}_{h,0}, \tag{11}$$

form an orthogonal basis in $\mathcal{V}_{h,\mathbf{0}}$. Since we want to capture only the low-frequent, global behavior of the body with eigenmodes, we let $\mathcal{V}_{h,TM}$ be the space spanned by the truncated modal basis (cf. [1]), $\{\boldsymbol{e}_h^i\}_{i=1}^m$, where $\boldsymbol{e}_h^i$, $i = 1, \ldots, m$ are the eigenmodes corresponding to the m smallest eigenvalues of a.

Now, the eigenvalue problem (11) is equivalent to a linear system of equations

$$\boldsymbol{K}\hat{\boldsymbol{e}}_h = \lambda \boldsymbol{M}\hat{\boldsymbol{e}}_h, \tag{12}$$

where $\boldsymbol{M}$ is the mass matrix. We find the low-frequency component, $\boldsymbol{u}_{h,TM}$, of our reduced solution by solving the $m \times m$ system of equations

$$\boldsymbol{V}_{h,TM}^T \boldsymbol{K} \boldsymbol{V}_{h,TM} \hat{\boldsymbol{u}}_{h,TM} = \boldsymbol{V}_{h,TM}^T \hat{\boldsymbol{b}}, \tag{13}$$

for its coefficients $\hat{\boldsymbol{u}}_{h,TM} = [u_{1,TM}, \ldots, u_{m,TM}]^T$, where $\boldsymbol{V}_{h,TM}$ is an $N \times m$ matrix with the m most low-frequent eigenmodes from (12) in its columns.

In order to capture the local effects we introduce the space $\mathcal{V}_{h,K}$, defined by the $N \times k$ coefficient matrix $\boldsymbol{V}_{h,K}$, with column vectors

$$\hat{\boldsymbol{c}}_j = \left[(\boldsymbol{M}^{-1}\boldsymbol{K})^j \boldsymbol{M}^{-1} \boldsymbol{r}_{TM} \right]_{j=0}^k, \tag{14}$$

where $\boldsymbol{r}_{TM}$ is the residual stemming from the eigenmode approximation. That is,

$$\boldsymbol{r}_{TM} = \hat{\boldsymbol{b}} - \boldsymbol{V}_{h,TM} \hat{\boldsymbol{u}}_{h,TM}, \tag{15}$$

where $\hat{\boldsymbol{u}}_{h,TM}$ is the solution to the system (13). The Krylov-modes thus generated are per construction orthogonal to the truncated modal basis. This approach can be compared with the so called modal acceleration method, cf. [1], where in fact the generating operator is the inverse to the operator in (14) above.

We find the high-frequency component, $\boldsymbol{u}_{h,K}$, of the reduced solution by solving the $k \times k$ system of equations

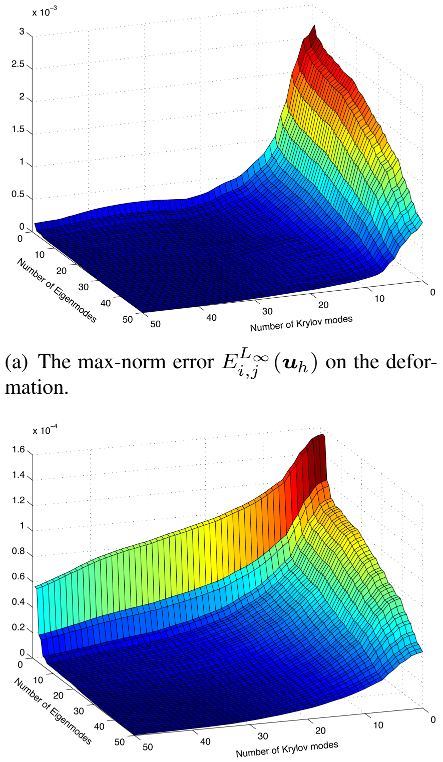
$$\boldsymbol{V}_{h,K}^T \boldsymbol{K} \boldsymbol{V}_{h,K} \hat{\boldsymbol{u}}_{h,K} = \boldsymbol{V}_{h,K}^T \hat{\boldsymbol{b}}, \tag{16}$$

for the coefficients $\hat{\boldsymbol{u}}_{h,K} = \{u_{1,K}, \ldots, u_{k,K}\}$, and our reduced solution is $\boldsymbol{u}_{h,R} = \boldsymbol{u}_{h,TM} + \boldsymbol{u}_{h,K}$.

3. Numerical Investigations

3.1 Numerical Results

To illustrate the ideas described above we present the following numerical investigation of how the accuracy in the computation depends on the number of eigenmodes and the number of Krylov-modes in the reduction basis. The computational domain is depicted in Figure 1. We apply the localized force as illustrated in the same figure. We use the elastic modulus $E = 1.0$, and Poisson's ratio $\nu = 0.29$.



(a) The max-norm error $E_{i,j}^{L_\infty}(\boldsymbol{u}_h)$ on the deformation.

(b) The L_2 error $E_{i,j}^{L_2}(\boldsymbol{u}_h)$ on the deformation.

Figure 2: Errors on the deformation for $i = 1, \ldots, 50$ eigenmodes, and $j = 0, \ldots, 50$ Krylov-modes, respectively.

Let $\mathcal{V}_{h,R(i,j)}$ be the reduced space spanned by the first i eigenmodes and the first j Krylov-modes respectively, and let $\boldsymbol{u}_{h,R(i,j)}$ be the corresponding solution. We compute the maximum-norm errors

$$\begin{aligned} E_{i,j}^{L_\infty}(\boldsymbol{u}_h) =& \|\boldsymbol{u}_{h,R(i,j)} - \boldsymbol{u}_h\|_{L_\infty(\Omega)}, \\ & i = 1, \ldots, 50,\ j = 0, \ldots, 50, \end{aligned} \tag{17}$$

and the L_2-errors

$$\begin{aligned} E_{i,j}^{L_2}(\boldsymbol{u}_h) =& \|\boldsymbol{u}_{h,R(i,j)} - \boldsymbol{u}_h\|_{L_2(\Omega)}, \\ & i = 1, \ldots, 50,\ j = 0, \ldots, 50, \end{aligned} \tag{18}$$

between a reference FEM solution and a set of reduced solutions. The results from these computation are depicted in Figure 2. It is clear from the graphs that the truncated modal basis alone is not able to resolve the load case in a satisfactory way, and that the error is reduced effectively by the addition of Krylov-modes to the basis. It is also clear from graph (b) that it is necessary to keep some eigenmodes in the basis in order to capture global effects. Let $\sigma_{h,R(i,j)}^v$ and σ_h^v denote the von Mises stresses corresponding to the reduced solution and the

FEM solution respectively. The maximum-norm errors

$$E^{L_2}_{i,j}(\sigma^v_h) = \|\sigma^v_{h,R(i,j)} - \sigma^v_h\|_{L_\infty(\Omega)}, \quad i = 1,\dots,50,\ j = 0,\dots,50, \tag{19}$$

in the von Mises stresses, are plotted in Figure 3. It is clear from this figure that the Krylov-modes significantly enhances the von Mises approximation. In Figure 4, we see the von Mises stresses corresponding to the reference solution $\boldsymbol{u}_h$, a reduced solution $\boldsymbol{u}_{h,R(20,10)}$ and a reduced solution $\boldsymbol{u}_{h,R(30,0)}$ respectively.

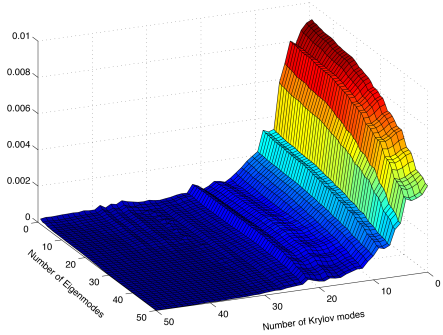


Figure 3: The maximum-norm error $E^{L_\infty}_{i,j}(\sigma^v_h)$ in the von Mises stresses, for $i = 1,\dots,50$ eigenmodes, and $j = 0,\dots,50$ Krylov-modes, respectively.

3.2 Aspects of mesh refinement

By subtracting (6) from (3) we get the Galerkin orthogonality:

$$a(\boldsymbol{u} - \boldsymbol{u}_h, \boldsymbol{v}) = 0, \quad \forall \boldsymbol{v} \in \mathcal{V}_h. \tag{20}$$

The analog Galerkin orthogonality for the reduced solution is attained by subtracting (10) from (6):

$$a(\boldsymbol{u}_h - \boldsymbol{u}_{h,R}, \boldsymbol{v}) = 0, \quad \forall \boldsymbol{v} \in \mathcal{V}_{h,R}. \tag{21}$$

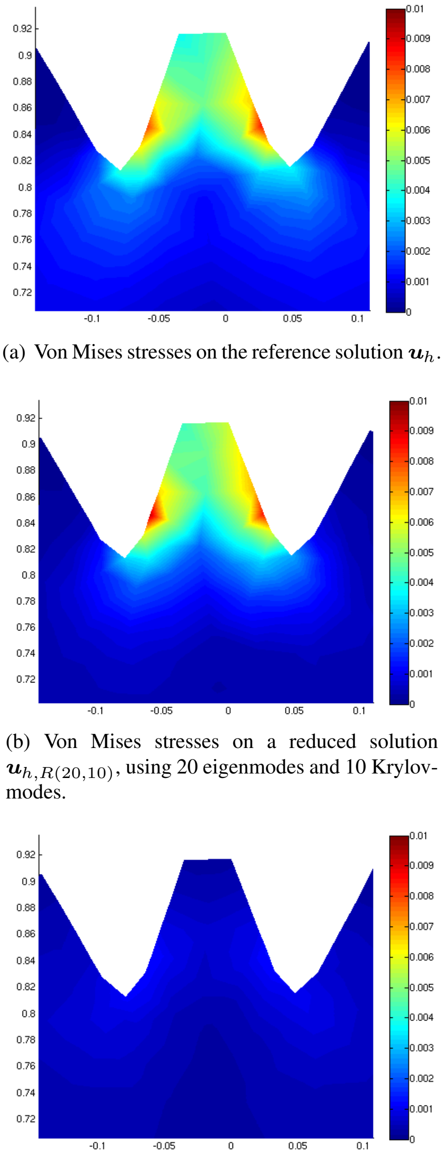
From (20) and (21) now follows

$$\|\boldsymbol{u} - \boldsymbol{u}_{h,R}\|^2_E = \|\boldsymbol{u} - \boldsymbol{u}_h\|^2_E + \|\boldsymbol{u}_h - \boldsymbol{u}_{h,R}\|^2_E, \tag{22}$$

were $\|\cdot\|_E$ denotes the energy norm, defined by

$$\|\boldsymbol{w}\|^2_E = a(\boldsymbol{w}, \boldsymbol{w}), \quad \forall \boldsymbol{w} \in \mathcal{V}_0. \tag{23}$$

Looking at (22) it's apparent that given a fixed set of reduction modes, the left hand side will be approximately constant (compared to the variation in h). The second term on the right hand side will hence grow larger as h tends to zero. This means that a greater number of reduction modes is needed if the geometry is complex and the mesh has fine-scale features.



(a) Von Mises stresses on the reference solution $\boldsymbol{u}_h$.

(b) Von Mises stresses on a reduced solution $\boldsymbol{u}_{h,R(20,10)}$, using 20 eigenmodes and 10 Krylov-modes.

(c) Von Mises stresses on a reduced solution $\boldsymbol{u}_{h,R(30,0)}$, using 30 eigenmodes and 0 Krylov-modes.

Figure 4: Von Mises stresses on the reference and reduced solutions.

In order to see how this manifests itself in the numerical results we compute the von Mises stresses on a reference FE solution $\boldsymbol{u}_h$ and a reduced solution $\boldsymbol{u}_{h,R(20,10)}$ with 20 eigenmodes and 10 Krylov-modes on a mesh locally refined as in Figure 5.

The Mises stresses around the loaded tooth of the reference FEM solution and the reduced solution can be seen in Figure 6. Compare this with the corresponding result on the coarse mesh in Figure 4. On the refined mesh it is apparent that 20 eigenmodes and 10 Krylov-modes in the reduction space is not enough to capture FE solution. We see that the effect of the Krylov-modes here is somewhat too local.

Now, the addition of more Krylov-modes to the basis will not improve the reduction significantly since the locality of the Krylov-modes will increase with k, and we know that the computation of more eigenmodes is expensive. Research is currently being undertaken in order to identify an effective approach to resolve this matter.

3.3 Notes on the Implementation

We have implemented the above computations in MATLAB [4] using the Standard Galerkin [5] method on a triangular mesh with linear basis functions. For fast Krylov-mode generation we have used a lumped mass matrix in the computations. The Krylov-modes are generated using the Arnoldi Modified Gram-Schmidt algorithm [3]. The eigenvalue problem is solved using ARPACK [6] through the MATLAB command `eigs`.

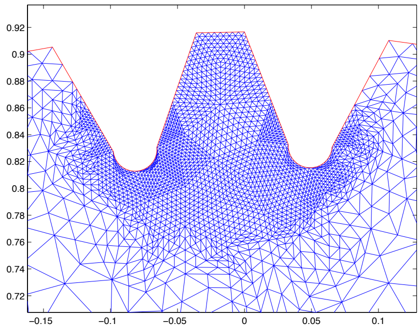
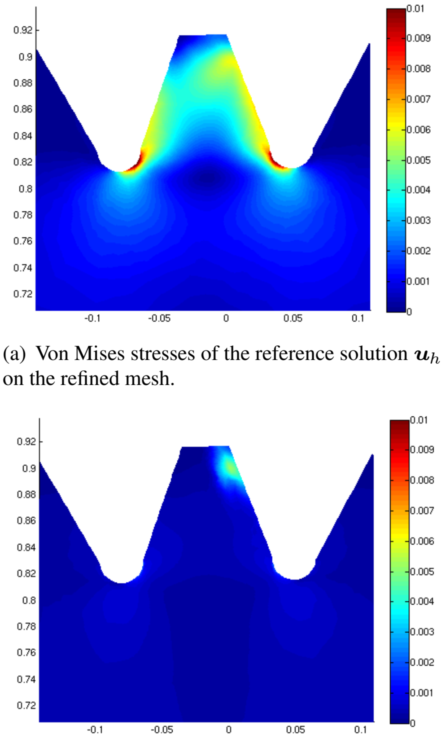


Figure 5: The mesh is refined locally around the loaded tooth.

4. Conclusions and Outlook

We have investigated the use of certain Krylov-modes in a modal based model reduction to capture high-frequency effects from a localized boundary load on a two-dimensional geometrically complex elastic body. We have presented numerical evidence showing that the Krylov-modes significantly enhances the reduction on a coarse mesh, and we have illustrated the effect of mesh refinement.

Current research is aimed at the investigation of other methods to capture localized loads. In the near future we shall begin the evaluation of these methods in 3D. Planned research include error analysis and the construction of adaptive reduction algorithms.



(a) Von Mises stresses of the reference solution $\boldsymbol{u}_h$ on the refined mesh.

(b) Von Mises stresses on a reduced solution $\boldsymbol{u}_{h,R(20,10)}$ on the refined mesh, using 20 eigenmodes and 10 Krylov-modes.

Figure 6: Von Mises stresses on the reference and reduced solutions on the refined mesh.

References

[1] Zu-Qing Qu. *Model Order Reduction Techniques with Applications in Finite Element Analysis*. Graduate Texts in Mathematics. Springer, London, Berlin, Heidelberg, 2004.

[2] S. Dietz. *Vibration and Fatigue Analysis of Vehicle Systems Using Component Modes*, volume 401. VDI Verlag, 1999.

[3] Yousef Saad. *Iterative Methods for Sparse Linear Systems, 2nd edition*. SIAM, 2003.

[4] The MathWorks, Inc. MATLAB. http://www.mathworks.com/products/matlab/.

[5] Alexandre Ern and Jean-Luc Guermond. *Theory and Practice of Finite Elements*. Graduate Texts in Mathematics. Springer, New York, Berlin, Heidelberg, 2004.

[6] ARPACK. http://www.caam.rice.edu/software/ARPACK/.

Eclipse Support for Design and Requirements Engineering Based on ModelicaML

Adrian Pop, Vasile Băluță, Peter Fritzson
Programming Environments Lab
Department of Computer and Information Science
Linköping University,
SE-581 83 Linköping, Sweden
{adrpo,x07vasba,petfr}@ida.liu.se

Abstract

In order to support the development of complex products, modeling tools and processes need to support co-design of software and hardware in an integrated way. Modelica is the major object-oriented mathematical modeling language for component-oriented modeling of complex physical systems and UML is the dominant graphical modeling notation for software. The ModelicaML UML profile integrates Modelica and UML to support engineering of whole products. In this paper we present the Eclipse ModelicaML implementation and integration with the MDT Eclipse plugin, with emphasis on requirements support.

Keywords: Requirements, Modeling and Simulation, Modelica, Eclipse, UML, SysML, ModelicaML.

1. Introduction

The development in system modeling has come to the point where complete modeling of systems is possible, e.g. the complete propulsion system, fuel system, hydraulic actuation system, etc., including embedded software can be modeled and simulated concurrently. This does not mean that all components are dealt with down to the very smallest details of their behavior. It does, however, mean that all functionality is modeled, at least qualitatively.

In this paper, the implementation of the ModelicaML UML profile for Modelica in Eclipse is presented, with emphasis on requirement diagrams.

1.1 SysML

The Unified Modeling Language (UML) has been created to assist software development processes by providing means to capture software system structure and behavior. This has evolved into the main standard for Model Driven Development [5].

The System Modeling Language (SysML) [4] is a graphical modeling language for systems engineering applications. SysML was developed by systems engineering experts, and was adopted by OMG in 2006. SysML is built on top of UML and tailored to the needs of system engineers by supporting specification, analysis, design, verification, and validation of a broad range of systems and system-of-systems.

The main goal behind SysML is to unify and replace different document-centric approaches in the system engineering field with a single systems modeling language. A single model-centric approach improves communication, assists managaging complex system design and allows its early validation and verification.

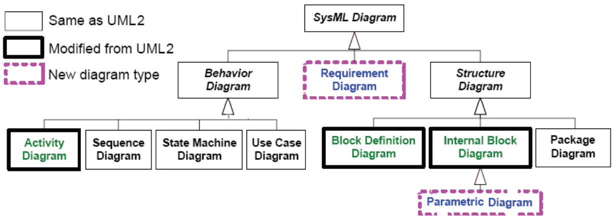


Figure 1. SysML diagram taxonomy.

The taxonomy of SysML diagrams is presented in Figure 1. For a full description of SysML see (SysML, 2006) [4]. The major SysML extensions compared to UML are:

- Requirements diagrams support requirements presentation in tabular or in graphical notation; allow composition of requirements; and support traceability, verification and "fulfillment of requirements".
- Block diagrams extend the Composite Structure diagram of UML2.0. The purpose of this diagram is to capture system components, their parts and connections between parts. Connections are handled by means of connecting ports which may contain data, material or energy flows.
- Parametric diagrams help perform engineering analysis such as performance analysis. Parametric diagrams contain constraint elements, which define mathematical equations, linked to properties of model elements.
- Activity diagrams show system behavior as data and control flows. Activity diagrams are similar to Extended Functional Flow Block Diagrams, which

are already widely used by system engineers. Activity decomposition is supported by SysML.

- Allocations are used to define mappings between model elements: For example, certain Activities may be allocated to Blocks (to be performed by the block).

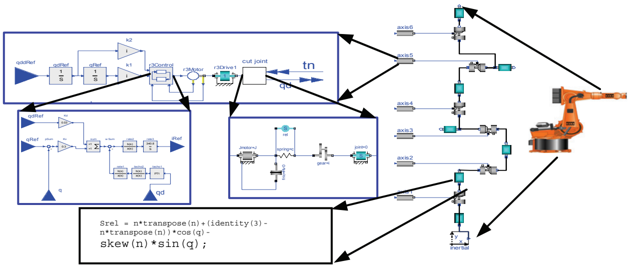
1.2 Modelica

Modelica [2][3] is a modern language for equation-based object-oriented mathematical modeling primarily of physical systems. Several tools, ranging from open-source such as OpenModelica [1], to commercial like Dymola [11] or MathModelica [10] support the Modelica specification.

The language allows defining models in a declarative manner, modularly and hierarchically and combining various formalisms expressible in the more general Modelica formalism. The multidomain capability of Modelica allows combining electrical, mechanical, hydraulic, thermodynamic, etc., model components within the same application model. In short, Modelica has improvements in several important areas:

- Object-oriented mathematical modeling. This technique makes it possible to create model components, which are employed to support hierarchical structuring, reuse, and evolution of large and complex models covering multiple technology domains.
- Physical modeling of multiple application domains. Model components can correspond to physical objects in the real world, in contrast to established techniques that require conversion to “signal” blocks with fixed in-put/output causality. In Modelica the structure of the model naturally correspond to the structure of the physical system in contrast to block-oriented modeling tools.
- Acausal modeling. Modeling is based on equations instead of assignment statements as in traditional input/output block abstractions. Direct use of equations significantly increases re-usability of model components, since components adapt to the data flow context in which they are used.

A component may internally consist of other connected components, i.e., hierarchical modeling as in Figure 2.



```
Srel = n*transpose(n)+(identity(3)-
n*transpose(n))*cos(q)-
skew(n)*sin(q);
```

Figure 2. Hierarchical model of an industrial robot.

1.3 SysML vs. Modelica

The System Modeling Language (SysML) has been proposed to unify different approaches and languages used by system engineers into a single standard. SysML models may span different domains, for example, electrical, mechanical and software. Even if SysML provides means to describe system behavior like Activity and State Chart Diagrams, the precise behavior can not be described and simulated without complex transformations and additional information provided for SysML models. In that respect, SysML is rather incomplete compared to Modelica.

Modelica was also created to unify and extend object-oriented mathematical modeling languages. It has powerful means for describing precise component behavior and functionality in a declarative way. Modelica models can be graphically composed using Modelica connection diagrams which depict the structure of designed system. However, complex system design is more that just a component assembly. In order to build a complex system, system engineers have to gather requirements, specify system components, define system structure, define design alternatives, describe overall system behavior and perform its validation and verification.

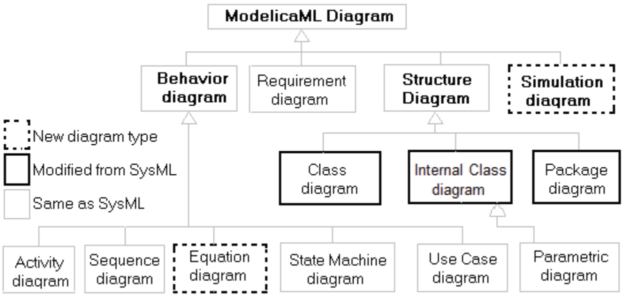


Figure 3. ModelicaML diagrams overview.

2. The ModelicaML UML profile

Before we present the Eclipse integration, we briefly describe the ModelicaML profile.

With respect to SysML, ModelicaML reuses, extends and provides several new diagrams. The ModelicaML diagram overview is shown in Figure 3. Diagrams are grouped into four categories: Structure, Behavior, Simulation and Requirement. The description of the ModelicaML profile is presented in [8], [16] and [17]. The most important properties of the ModelicaML profile are outlined in the following:

- The ModelicaML profile supports modeling with all Modelica constructs and properties i.e. restricted classes, equations, generics, variables, etc.
- Using ModelicaML diagrams it is possible to describe most of the aspects of a system being designed and thus support system development process phases such as requirements analysis, design,

implementation, verification, validation and integration.

- The profile supports mathematical modeling with equations since equations specify behavior of a system. Algorithm sections are also supported.
- Simulation diagrams are introduced to model and document simulation parameters and simulation results in a consistent and usable way.
- The ModelicaML meta-model is consistent with SysML in order to provide SysML-to-ModelicaML conversion.

2.1 Package Diagrams

ModelicaML extends SysML packages in order to support Modelica packaging features, in particular: package inheritance, generic packages, constant declaration within a package, package "instantiation" and renaming import (see [2] for Modelica packages details).

A snapshot of the Modelica Standard Library hierarchy is shown in Figure 4 using UML notation. Package nodes in the hierarchy are connected via the package containment link as in the example in Figure 4.

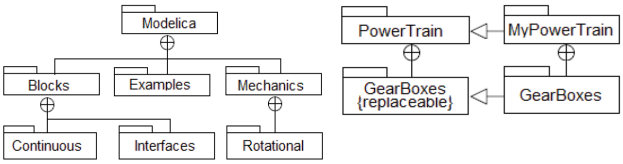


Figure 4. Package hierarchy modeling.

2.2 Modelica Class Diagrams in ModelicaML

Modelica uses restricted classes such as class, model, block, connector, function and record to describe a system. Modelica classes have essentially the same semantics as SysML blocks specified in [4] and provide a general-purpose capability to model systems as hierarchies of modular components. ModelicaML extends SysML blocks by defining features which are relevant or unique to Modelica.

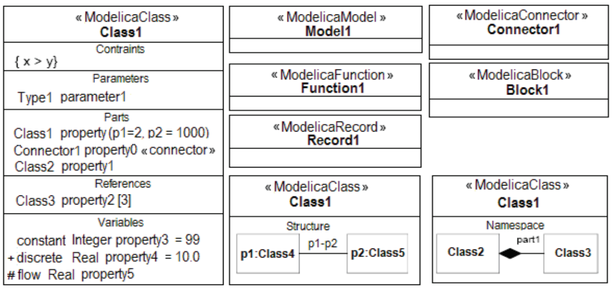
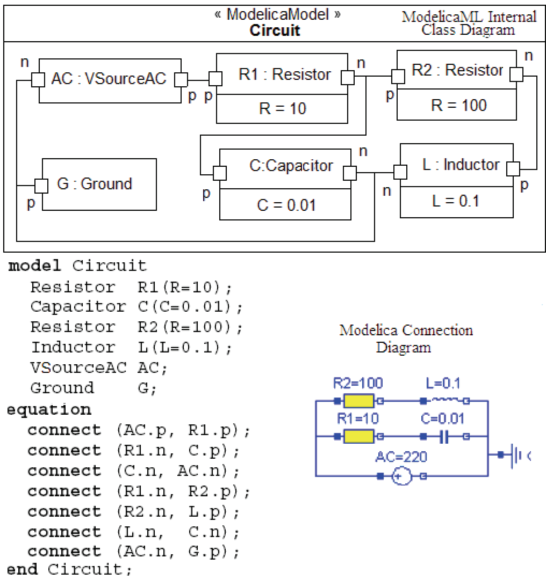


Figure 5. ModelicaML Class Diagrams.

The purpose of the Modelica Class Diagram is to show features of Modelica classes and relationships between classes. Additional kind of dependencies and associations between model elements may also be shown in a Modelica Class Diagram. For example, behavior description constructs – equations, may be associated with particular Modelica Classes. The detailed description of structural features of ModelicaML and graphical notation of ModelicaML class definitions is shown in Figure 5. Each class definition is adorned with a stereotype name that indicates the class type it represents. The ModelicaML Class Definition has several compartments to group its features: parameters, parts, variables. Some compartments are visible by default; some are optional and may be shown on ModelicaML Class Diagram with the help of a tool. Property signatures follow the Modelica textual syntax and not the SysML original syntax, reused from UML. A ModelicaML/SysML tool may allow users to choose between UML or Modelica style textual signature presentation.

The ModelicaML Internal Class Diagram is based on the SysML Internal Block Diagram. The Modelica Class Diagram defines Modelica classes and relationships between classes, like generalizations, association and dependencies, whereas a ModelicaML Internal Class Diagram shows the internal structure of a class in terms of parts and connections. The ModelicaML Internal Class Diagram is similar to the Modelica connection diagram, which presents parts in a graphical (icon) form. An example Modelica model presented as an Internal Class diagram is shown in Figure 6



```
model Circuit
  Resistor   R1(R=10);
  Capacitor  C(C=0.01);
  Resistor   R2(R=100);
  Inductor   L(L=0.1);
  VSourceAC  AC;
  Ground     G;
equation
  connect (AC.p, R1.p);
  connect (R1.n, C.p);
  connect (C.n, AC.n);
  connect (R1.n, R2.p);
  connect (R2.n, L.p);
  connect (L.n,  C.n);
  connect (AC.n, G.p);
end Circuit;
```

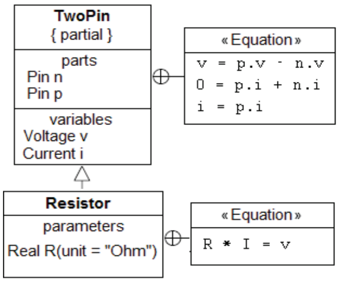
Figure 6. ModelicaML Internal Class Diagram vs. Modelica Connection Diagram.

Usually Modelica models are presented graphically via Modelica connection diagrams (Figure 6, bottom). Such diagrams are created by the modeler using a graphic connection editor by connecting together components from available libraries. Since both diagram types are used to compose models and serve the same purpose, we briefly compare the Modelica connection diagram to the ModelicaML Internal Class Diagram. The main advantage of the Modelica connection diagram over the Internal Class Diagram is that it has better visual comprehension as components are shown via domain-specific icons known to application modelers. Another advantage is that Modelica library developers are able to predefine connector locations on an icon, which are related to the semantics of the component.

2.3 Equation Diagrams

SysML defines Constraint blocks which specify mathematical expressions, like equations, to constrain physical properties of a system. Constraint blocks are defined in the Block Definition diagram and can be packaged into domain-specific libraries for later reuse. There is a special diagram type called Parametric Diagram which relates block parameters with certain constraints blocks. The Parametric Diagram is included in ModelicaML without any modifications.

The Modelica class behavior is usually described by equations, which also constrain Modelica class parameters, and have a domain-specific usage. SysML constraint blocks are a less powerful means of domain model description than Modelica equations, which also include certain kinds of equations which cannot be modeled using constraint blocks, i.e.: if-, for-, and when- equations.



```
partial class TwoPin
  Pin p, n;
  Voltage v;
  Current i;
equation
  v = p.v - n.v;
  0 = p.i + n.i;
  i = p.i;
end TwoPin;

class Resistor
  extends TwoPin;
  parameter Real R(unit = "Ohm");
equation
  R * I = v;
end Resistor;
```

Figure 7. Equation modeling example with a ModelicaML Class Diagram.

2.4 Simulation Diagram

ModelicaML also introduces a new diagram type, called Simulation Diagram (Figure 8), used for simulation experiment modeling. Simulation is usually performed by a simulation tool which allows parameter setting, variable selection for output and plotting. The Simulation Diagram can be used to store any simulation experiment, thus helping to keep the history of simulations and its results. When integrated with a modeling and simulation environment, a simulation diagram may be automatically generated.

The Simulation Diagram provides facilities for simulation planning, structured presentation of parameter passing and simulation results. Simulations can be run directly from the Simulation Diagram. Association of simulation results with requirements from a domain expert and additional documentation (e.g. by: Note, Problem Rationale text boxes of SysML) are also supported by the Simulation Diagram.

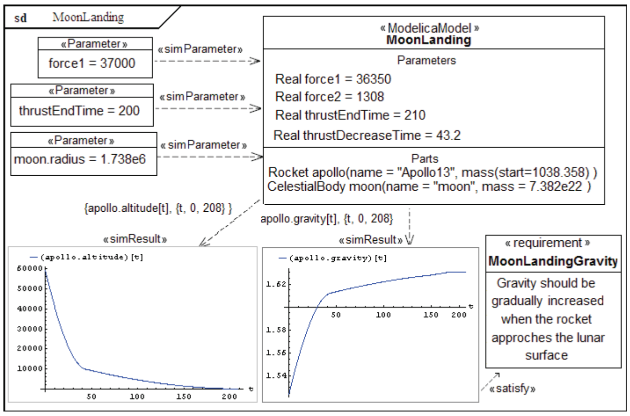


Figure 8. Simulation Diagram example.

2.5 Requirement Diagrams

Requirement diagrams have been included in ModelicaML from SysML without any modification. Requirement Diagrams have several attributes: ID, Level, Status, Name and Description. Requirements support hierarchical modeling, i.e., more specific requirements can derived from more general ones.

Requirements can be linked to any other ModelicaML element via `satisfies` or `satisfiedBy` relations. Any tool implementing the ModelicaML profile can be used to build, query, trace, and manage requirements.

The Modelica language does not support a standard requirements representation. Since the ModelicaML profile supports requirements, we need a way to save these requirements in a Modelica file. Requirements can be represented in Modelica in several ways which we will describe in detail in the next section.

3. ModelicaML in Eclipse

Eclipse [14] is an open source framework for creating extensible integrated development environments (IDEs). One of the goals of the Eclipse platform is to avoid duplicating common code that is needed to implement a powerful integrated environment for the development of software. By allowing third parties to easily extend the platform via the plugin concept, the amount of new code that needs to be written is decreased. For the development of our prototype we have used several Eclipse frameworks:

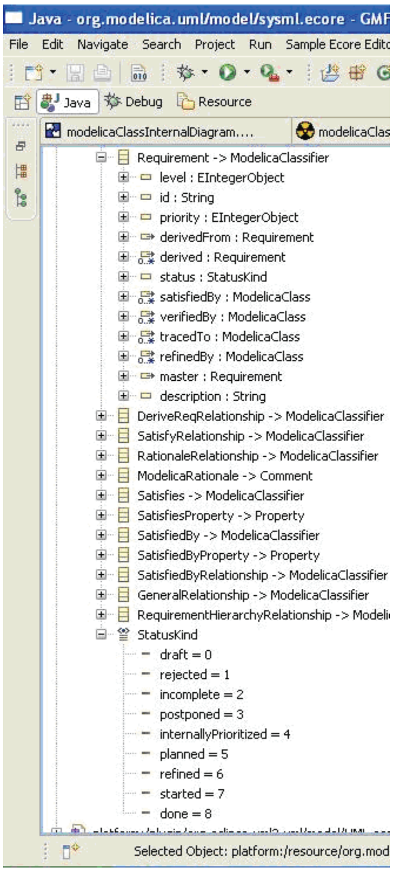
- EMF – Eclipse Modeling Framework [18] is an Eclipse framework for building domain-specific model implementations. The EMF implementation is based on the Meta Object Facility (MOF) standard and implements the "Essential MOF" (EMOF) part of a standard. EMF is used by the GMF and UML2 frameworks.
- GMF – Graphical Modeling Framework [20] provides a generative component and runtime infrastructure for developing graphical editors based on EMF and GEF [19]. GMF consists of tooling, gen-

erative and runtime parts, depends on the EMF and GEF frameworks and also on other EMF related tools.

- The UML2 Eclipse meta-model implementation. The UML2 Eclipse project is an EMF based implementation of a UML2 meta-model for the Eclipse Platform to support development of UML modeling tools. The UML2 project doesn't aim to provide any graphical modeling or diagram interchange capabilities as it only implements UML abstract syntax. UML2 Tools focus on editing capabilities.

3.1 The ModelicaML GMF Model

The Eclipse editor is created from a GMF model of the ModelicaML profile. The model describes the existing elements and their properties. As an example, in Figure 9 we present the Requirement Diagram element and its properties.


Java - org.modelica.uml/model/sysml.ecore - GMF
File Edit Navigate Search Project Run Sample Ecore Edito
Java
Debug
Resource
modelicaClassInternalDiagram....
modelicaClas

```
Requirement -> ModelicaClassifier
    level : EIntegerObject
    id : String
    priority : EIntegerObject
    derivedFrom : Requirement
    derived : Requirement
    status : StatusKind
    satisfiedBy : ModelicaClass
    verifiedBy : ModelicaClass
    tracedTo : ModelicaClass
    refinedBy : ModelicaClass
    master : Requirement
    description : String
DeriveReqRelationship -> ModelicaClassifier
SatisfyRelationship -> ModelicaClassifier
RationaleRelationship -> ModelicaClassifier
ModelicaRationale -> Comment
Satisfies -> ModelicaClassifier
SatisfiesProperty -> Property
SatisfiedBy -> ModelicaClassifier
SatisfiedByProperty -> Property
SatisfiedByRelationship -> ModelicaClassifier
GeneralRelationship -> ModelicaClassifier
RequirementHierarchyRelationship -> Modelic
StatusKind
    draft = 0
    rejected = 1
    incomplete = 2
    postponed = 3
    internallyPrioritized = 4
    planned = 5
    refined = 6
    started = 7
    done = 8
Selected Object: platform:/resource/org.mod
```

Figure 9. ModelicaML GMF Model (Requirements)

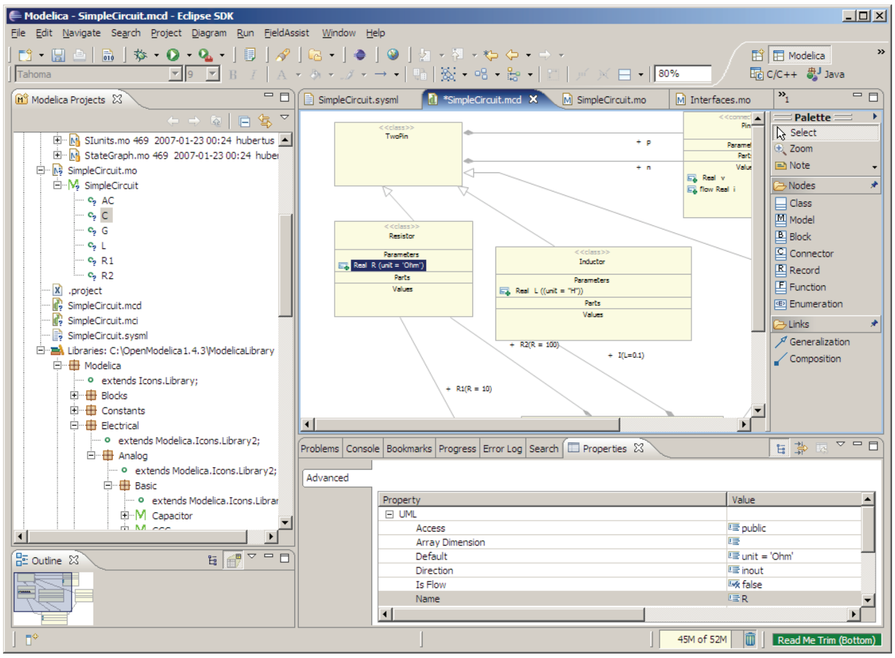
From the GMF model an editor that supports common operations on that model is automatically generated. The generated code can be extended to deal with issues specific to ModelicaML.

3.2 The ModelicaML Eclipse Editor

The Modelica Development Tooling (MDT) [7] Eclipse plugin is part of the OpenModelica system [1] and provides an environment for working with Modelica projects. The following features are available within the editor:

- Browsing support and wizards for creating Modelica projects, packages, Modelica Standard Libr.
- Syntax color highlighting, syntax and semantic checking.
- Code assistance for packages and function calls
- Support for MetaModelica meta-programming extensions to standard Modelica
- Debugging support for Modelica and MetaModelica algorithmic code.

We have extended the MDT plugin with a design view to facilitate ModelicaML integration. The ModelicaML integrated design environment where SysML/ ModelicaML diagrams are created is shown in Figure 10. It consists of a diagram file browser (left), diagram editor (middle), tool palette (right), properties editor (bottom) and a diagram outline (bottom left).



Property	Value
UML	
Access	public
Array Dimension	
Default	unit = 'Ohm'
Direction	inout
Is Flow	false
Name	R

Figure 10. ModelicaML Eclipse based design environment with a Class diagram.

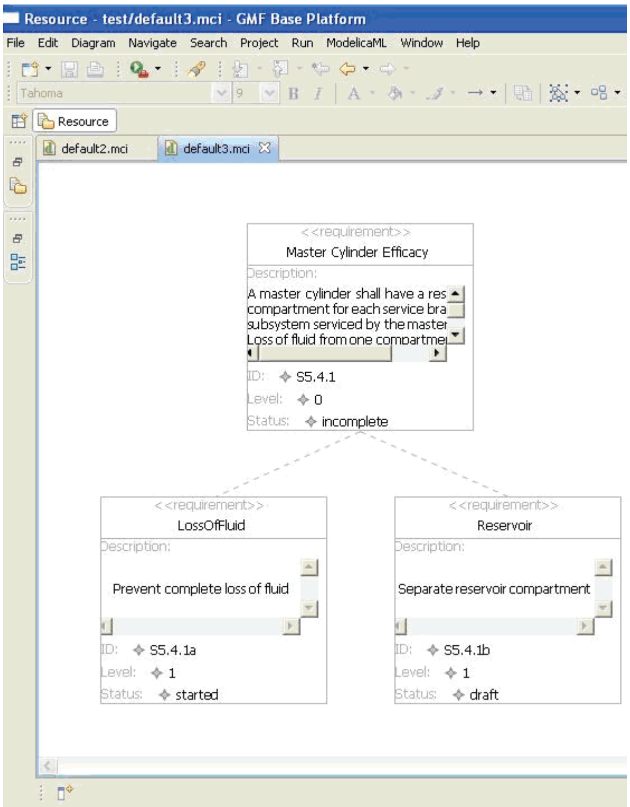
The Project Browser lists all Modelica files. The Diagram Editor is a tool where diagrams can be created and graphical elements laid out. It has the following graphical features: Graphical elements like Modelica Class, Model, or Requirement can be picked up from a Tool palette and created in a Diagram editor pane in a drag-and-drop way. Elements in a palette are grouped by Standard tools (zooming, note, etc), Nodes and Links elements. The tool palette for Modelica Class and Internal Class diagram contains different sets of elements. The Property Editor can be used for changing the properties of the object selected in the diagram editor pane. Property elements vary depending on the type of chosen object.

The ModelicaML diagrams can be automatically generated from Modelica source files. The integrated tool can also generate Modelica source code from ModelicaML diagrams. However, the implementation of the Modelica code generation and ModelicaML diagram generation is, at the moment, in an experimental stage. In the current implementation ModelicaML diagrams are saved both in Modelica form and also the XMI dialect written to XML files. Further work is needed to save diagram position information within Modelica source code as annotations.

3.3 Modeling with Requirements

The ModelicaML/MDT Eclipse environment supports modeling with requirements. The following functionality is available in the development environment:

- Hierarchies of requirements can be created.
- Requirements can be traced during the development process.
- Requirements can be queried with respect to any of their attributes.



<<requirement>>
Master Cylinder Efficacy
Description:
A master cylinder shall have a res
compartment for each service bra
subsystem serviced by the master
Loss of fluid from one compartme
ID: S5.4.1
Level: 0
Status: incomplete

<<requirement>>
LossOfFluid
Description:
Prevent complete loss of fluid
ID: S5.4.1a
Level: 1
Status: started

<<requirement>>
Reservoir
Description:
Separate reservoir compartment
ID: S5.4.1b
Level: 1
Status: draft

Figure 11. Modeling with Requirement Diagrams.

Examples of modeling with Requirement Diagrams are presented in Figure 11 and in the Appendix.

3.4 Representing Requirements in Modelica

While the default storage of ModelicaML diagrams is within XML files, our goal is to be able to save this information within normal Modelica files. Having this information within Modelica code would make it accessible to other Modelica and UML tools. Using this information tools could provide additional functionality. For example Modelica tools could display the inheritance hierarchy of a library, display the requirements for a specific class, etc.

To find the best way to encode the ModelicaML diagram information within Modelica we have experienced with several ways of encoding the Requirement Diagrams.

Using Modelica Annotations

A requirement could be saved as an annotation in the following way (the top requirement from Figure 11):

```
type RequirementStatus =
  enumeration(Incomplete, Draft, Started);

annotation(
  Requirement(
    id="S5.4.1",
    level=0,
    status=RequirementStatus.Incomplete,
    name="Master Cylinder Efficacy",
    description="A master cylinder…"));
```

The problem with Modelica annotations is that they can only be present at specific places within code and they are usually tied to a class definition. Because requirements are usually cross cutting is impossible to represent all requirements as such annotations. Another problem using annotations is the representation of hierarchies of requirements. Linking of a class with a requirement via a `satisfy` relation could be possible using Modelica `extends`, but would be cumbersome.

Creating a new Restricted Class: requirement

A requirement could be saved as a standard `class` marked with an `isRequirement` annotation or alternatively using a new restricted class in the following way (the diagrams from Figure 12 in the Appendix):

```
requirement R1
  String name="Master Cylinder Efficiency";
  String id="S5.4.1";
  Integer level=0;
  RequirementStatus status=
     RequirementStatus.Incomplete;
  String description=”A master cylinder
                     shall have…”;
end R1;
```

Now, we can use `extends` over requirements to build hierarchies:

```
requirement R2
  extends R1;
  String name"Loss Of Fluid";
  String id="S5.4.1a";
  Integer level=1;
  RequirementStatus status=
     RequirementStatus.Started;
  String description="Prevent complete
                     loss of fluid";
...
end R2;
```

To link requirements to Modelica elements one can use annotations:

```
model BreakSystem
  annotation(satisfy=R1);
...
end BreakSystem;
```

We believe that the best way to encode requirements within Modelica would be to create a new restricted class. Using this new class, requirements can be fully modeled in Modelica.

We will propose in the Modelica Association to introduce the `requirement` restricted class into the Modelica specification. In Modelica, the requirement class could also have equation or algorithm sections that impose constraints to be verified against the class linked with the requirement.

4. Conclusion and Future Work

In this paper we have presented the integration and implementation of the ModelicaML profile in Eclipse. UML Statecharts and Modelica have previously been combined, see e.g. [9][15]. SysML has already been adopted for system on chip design [13] evaluated for code generation [14], and extended with bond graphs support [12].

The support for Modelica in ModelicaML allows precise definition, specification, and simulation of physical systems. Modelica provides the means for defining behavior for SysML block diagrams while the additional modeling capabilities of SysML provides additional modeling and specification power to Modelica (e.g. requirements and inheritance diagrams, etc).

Ongoing and future work targets further development and improvement of the Eclipse-based ModelicaML integrated development environment as a part of our Modelica Development Tooling (MDT).

5. Acknowledgements

This work was supported by Vinnova in the Safe & Secure Modeling and Simulation project, by SSF in the RISE project and by the CUGS graduate school at Linköping University.

References

[1] P Fritzson, P Aronsson, H Lundval, K Nyström, A Pop, L Saldamli, and D Broman. The OpenModelica Modeling, Simulation, and Software Development Environment. In Simu-lation News Europe, 44/45, Dec 2005. http://ww.ida.liu.se/projects/OpenModelica.

[2] Peter Fritzson. Principles of Object-Oriented Modeling and Simulation with Modelica 2.1, 940 pp., Wiley-IEEE Press, 2004. See also: http://www.mathcore.com/drmodelica/

[3] The Modelica Association. The Modelica Language Specification Version 2.2.

[4] OMG, System Modeling Language, (SysML), http://www.omgsysml.org

[5] OMG : Guide to Model Driven Architecture: The MDA Guide v1.0.1

[6] Eclipse.Org, http://www.eclipse.org/

[7] A Pop, P Fritzson, A Remar, E Jagudin, and D Akhvlediani. OpenModelica Development Environment with Eclipse Integration for Browsing, Modeling, and Debugging. In Proc of the Modelica'2006, Vienna, Austria, Sept. 4-5, 2006.

[8] David Akhvlediani. Design and implementation of a UML profile for Modelica/SysML. Final Thesis, LITH-IDA-EX--06/061—SE, April 2007.

[9] J.Ferreira, J. Estima, Modeling hybrid systems using Statechars and Modelica". 7th IEEE Intl. Conf. on Emerging Technologies and Factory Automation, October 1999, Spain.

[10] MathCore, MathModelica http://mathcore.com

[11] Dynasim, Dymola, http://dynasim.com

[12] Skander Turki, Thierry Soriano, A SysML Extension for Bond Graphs Support, Proc. of the International Conference on Technology and Automation (ICTA), Greece, 2005

[13] Yves Vanderperren, Wim Dehane, SysML and Systems Engineering Applied to UML-Based SoC Design, Proc. of the 2nd UML-SoC Workshop at 42nd DAC, USA, 2005.

[14] Yves Vanderperren, Wim Dehane, From UML/SysML to Matlab/Simulink, Proceedings of the Conference on Design, Automation and Test in Europe (DATE), Munich, 2006.

[15] André Nordwig. Formal Integration of Structural Dynamics into the Object-Oriented Modeling of Hybrid Systems. ESM 2002: 128-134.

[16] Adrian Pop, David Akhlevidiani, Peter Fritzson, Towards Unified System Modeling with the ModelicaML UML Profile, EOOLT'2007 - 1st International Workshop on Equation-Based Object-Oriented Languages and Tools, July 29-August 3, 2007, Berlin, Germany.

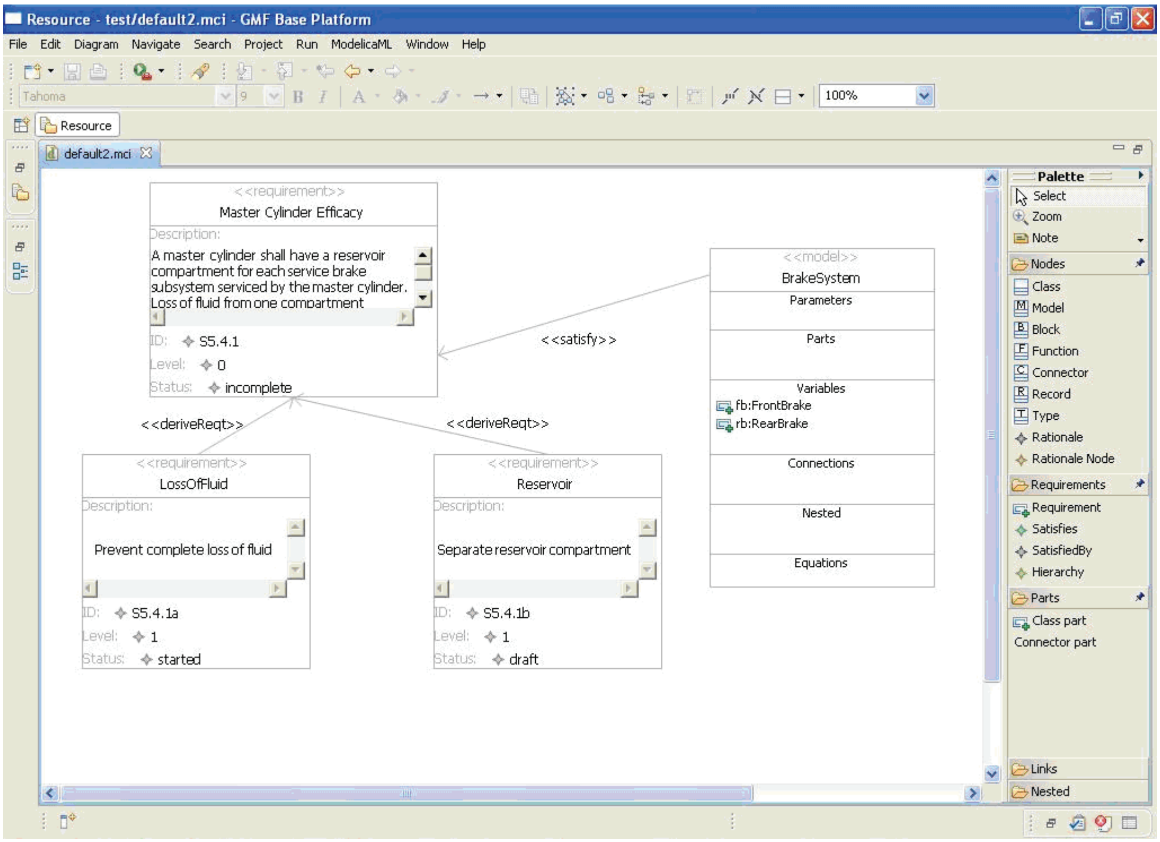
[17] Adrian Pop, David Akhvlediani, Peter Fritzson: Integrated UML and Modelica System Modeling with ModelicaML in Eclipse, 11th IASTED International Conference on Software Engineering and Applications (SEA 2007), November 19-21, 2007, Cambridge, MA, USA

[18] Eclipse.Org, Eclipse Modeling Framework (EMF), http://www.eclipse.org/emf/

[19] Eclipse.Org, Graphical Editing Framework (GEF), http://www.eclipse.org/gef/

[20] Eclipse.Org, Graphical Modeling Framework (GMF), http://www.eclipse.org/gmf/

Appendix

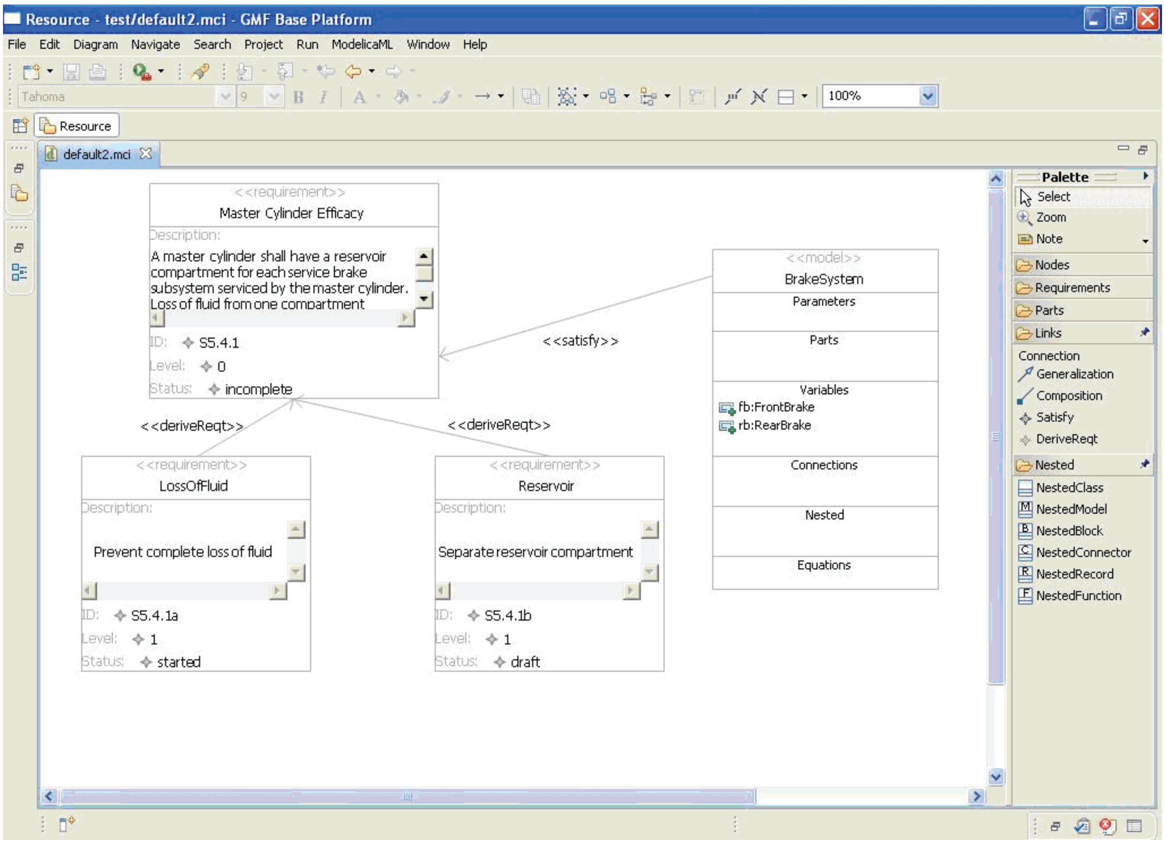


<<requirement>>
Master Cylinder Efficacy
Description:
A master cylinder shall have a reservoir compartment for each service brake subsystem serviced by the master cylinder. Loss of fluid from one compartment
ID: S5.4.1
Level: 0
Status: incomplete

<<requirement>>
LossOfFluid
Description:
Prevent complete loss of fluid
ID: S5.4.1a
Level: 1
Status: started

<<requirement>>
Reservoir
Description:
Separate reservoir compartment
ID: S5.4.1b
Level: 1
Status: draft

Figure 12. Modeling with requirements (Requirements palette).



<<requirement>>
Master Cylinder Efficacy
Description:
A master cylinder shall have a reservoir compartment for each service brake subsystem serviced by the master cylinder. Loss of fluid from one compartment
ID: S5.4.1
Level: 0
Status: incomplete

<<requirement>>
LossOfFluid
Description:
Prevent complete loss of fluid
ID: S5.4.1a
Level: 1
Status: started

<<requirement>>
Reservoir
Description:
Separate reservoir compartment
ID: S5.4.1b
Level: 1
Status: draft

Figure 13. Modeling with requirements (Connections).

Intelligent modelling of a fluidised bed granulator used in production of pharmaceuticals

Esko K. Juuso
Control Engineering Laboratory, Department of Process and Environmental Engineering,
P.O.Box 4300, FI-90014 University of Oulu, Finland
esko.juuso@oulu.fi

Abstract

The aim of dynamic modelling and simulation is to improve the control of the fluidised bed granulator. Modelling and simulation was done on the basis of data collected from several test campaigns. Several modelling methodologies have been compared in Matlab-Simulink environment. A solution based on dynamic linguistic equation models was chosen. The main input variables are humidity difference between incoming and outgoing air, temperature difference between inflowing air and granule and the rate of inflowing air. The final output is the estimated granule size but the overall models contains also dynamic models for temperature and humidity. The simulator combines several models which are specific to the operating conditions. According to the results, the spraying and drying processes included short-duration periods. Extension to fuzzy LE models provides useful information about uncertainties of the forecasted granulation results. The complexity of the models is increased only slightly with the new system based on the extension principle and fuzzy interval analysis.

Keywords: fluidised bed granulator, linguistic equations, dynamic modelling, fuzzy set systems.

1. Introduction

Powder particles are agglomerated though granulation processes due to interparticle bonds caused by the addition of a granulation liquid. The handling of the starting materials is facilitated and further processing (e.g. tabletting) becomes more secure [1]. Granulation usually refers to processes whereby aggregates with sizes ranging from approximately 0.1 to 2.0 mm are produced by agitation of moistened powder. Compression characteristics are improved, and handling of powders become easier because of less dust, less adhesion with hydroscopic materials. [2]

During the granulation process a three-phase system of solid, liquid and gas is established. The system will reduce its free energy by formation of liquid bridges between the particles. By the liquid bridges cohesive forces are established which may cause agglomeration and consolidation of the agglomerates in so far as they can resist the disruptive forces. The outcome depends on the interactions between apparatus, process and compositional variables and the properties of the powder. [3, 4, 5]

Airflow rate, temperature and humidity of the inlet air and the addition rate and droplet size of the granulating liquid are critical input variables. Temperature and humidity measurements of the process air are the most important parameters for monitoring heat and mass transfer. However, the inlet air humidity cannot usually be specified accurately because the seasonal variations in the process air humidity are difficult to control entirely. The actual effect of different humidity levels of the inlet air on the various fluid bed process parameters have been studied in [6]. Properties of the particles are also important. Effects of primary particle surface wettability by a binder solution on the rate of agglomeration were investigated in [7].

A physically-based mathematical model for the description of particle wetting and of temperature and concentration distribution in fluidized bed spray-granulation is presented in [8]. The bed mass and particle diameter growth in discontinuous granulation are taken into account and the two-dimensional calculation of the temperature and concentration distributions were carried out for the steady, continuous fluidized bed spray-granulation.

The physical changes in the beginning of spraying process are fast, because the weight of granules increases rapidly, which requires also that the amount of the inlet air has to be increased significantly so that fluidising would continue. This part of the process was the most difficult part to model. Correspondingly during the first few minutes of drying, the surface drying proceeds quickly until the balance is found. According to samples, the size of granules continued to grow for a while even the drying phase was started.

The aim of dynamic modelling and simulation is to improve the control of the fluidised bed granulator. Modelling is based on linguistic equation (LE) approach introduced in [9]. LE approach has been used in various applications [10, 11]. Dynamic LE models have provided accurate prediction and good performance in continuous processes, e.g. a lime kiln and a solar collector field [10]. A set of interactive intelligent systems can be combined with other modelling and simulation methodologies to build practical simulators for industrial processes [12].

For granulation process, dynamic modelling and simulation is necessary. Dynamic LE modelling was started in 2000 in cooperation with Helsinki University and Orion [13]. Dynamic models are well suited for forecasting the granulation result [13, 14]. The research equipment used in this project was a bench-scale fluidised bed granulator (Glatt WSG 5) shown in Figure 1. Modelling and simulation was done on the basis of data collected from test campaigns based on experimental design.

This paper presents more details of the solution and extends the models to uncertain environment.

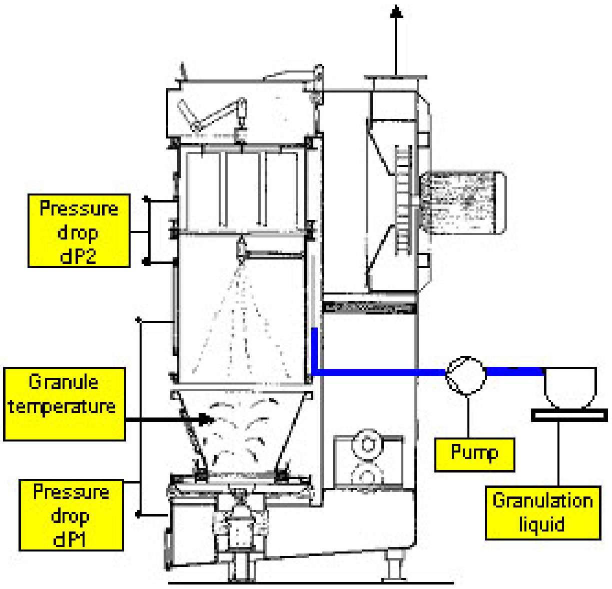


Figure 1. A bench-scale fluidised bed granulator.

2. Measurements

The granulation process shown in Figure 1 has three main phases:

- mixing to get granules homogeneous,
- spraying with PVP granulation liquid, and
- drying with warm air.

The model formulation (batch size 3500 g) consisting of verapamil hydrochloride, microcrystalline cellulose and lactose monohydrate was applied. Polyvinylpyrrolidone was used as a binder. The temperature of the drying air was 60 ^{o}C.

To eliminate the granules escape from granulator, filters were needed to shake every 100 seconds. The shaking takes 10 seconds, and meanwhile the process flows are off. For proper modelling it was essential to eliminate the effect of shaking as well as possible. More stable data and better modelling was achieved by median and moving average method.

Testing data was collected from 38 batches, 27 batches were used to training and the rest of the 11 to testing. The design of experiments for the test batches R1-11 presented in Table 1 includes three levels (high, normal and low) for the feed rate and the pressure of the granulation liquid. To confirm the functionality there were three repeated batches in the normal conditions.

Table 1. The design of experiments for the batches.

Test	Feed rate of granulation liquid [l/min]	Feed pressure of granulation liquid [bar]
R1	100	1.5
R2	125	1.5
R3	100	2
R4	125	2.5
R5	100	2
R6	125	2
R7	112.5	1.5
R8	112.5	2.5
R9	112.5	2
R10	112.5	2
R11	112.5	2

Numerous variables are known to affect the fluid bed process and the final granules. During the test campaigns, on-line measurements of more than 40 variables were collected with sampling time of one second. Air flow rate, temperature and humidity of the inlet air and the addition rate of the granulation are important variables. The instrumentation is described in [15].

Particle size analyses of intermediate and product granules and bulk factor were done off-line. Particle size analysis introduces two challenges: (1) it is based on samples, and there cannot be too many of them, and (2) particle size has always a distribution. The data driven modelling was based on the average particle size, and interpolation based on nonlinear regression was used to obtain additional points required for the dynamic modelling.

Seasonal variation in humidity is considerable and that will cause the changes in the water amount of the incoming air. The incoming air humidity should be included to the input variables or another possibility is to make a pre-moistening to a constant moisture value in the beginning of granulation

3. Nonlinear modelling

Nonlinear models are needed in modelling of the granulation process. Various statistical and intelligent methodologies have been compared in this project.

3.1 Statistical modelling

Response surface methodology combines linear terms with interaction and quadratic terms to calculate one output variable y from multiple input variables x_j:

$$\begin{aligned} y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \ldots \\ + b_{12} x_1 x_2 + b_{13} x_1 x_3 + b_{23} x_2 x_3 + \ldots \\ + b_{11} x_1^2 + b_{22} x_2^2 + b_{33} x_3^2 + \ldots \end{aligned} \qquad (1)$$

The number of parameters increases very fast with the number of variables.

Statistical models have been used for interpolating the granule size to obtain data for dynamic modelling. There are considerable differences between the batches (Figure 6).

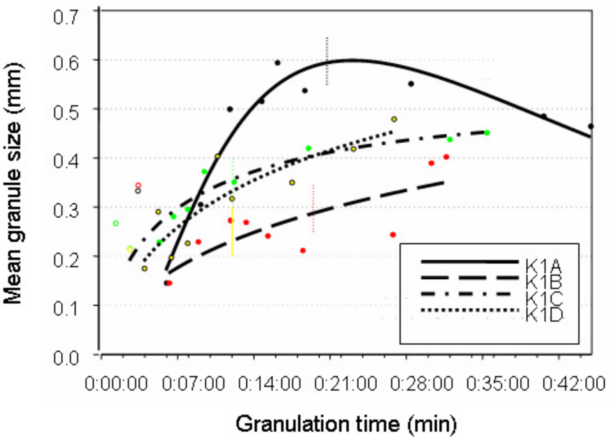


Figure 2. Interpolation of the granule size from 8 samples, starting point presumed to been 0.05 mm.

Dynamic statistical modelling is widely used in system identification. For parametric models, the output at time t is computed as a linear combination of past inputs and past outputs in such a way that the output at time t can depend on the signals at many previous time instants chosen according to appropriate time delays [16]. The number of delayed inputs and outputs is usually referred as the model order(s). The simplest model, ARX model, is usually written

$$y(t) + A_1 y(t-1) = A_2 u(t-n_k) + e(t), \qquad (2)$$

where A_1 and A_2 are model coefficients, *y(t)* and *y(t-1)* state variables and *u(t-n$_k$)* input variable delayed with n_k time steps. State-space models are widely used for combining effects of several input variables.

Various structures based on ARMAX, output error and Box Jenkins with different orders of the respective polynomials have been compared. Nonlinear models are needed, i.e. higher orders were needed in the parametric models, and the state-space models were insufficient.

3.2 Fuzzy modelling

Fuzzy set theory was first presented by Zadeh [17] to form a conceptual framework for linguistically represented knowledge. Extension principle is the basis generalisation of the arithmetic operations if the inductive mapping *F(x)* is a monotonously increasing function, e.g. *F(x)=x^2* in Figure 3. These results can be combined by applying fuzzy interval analysis in fuzzy arithmetic [18].

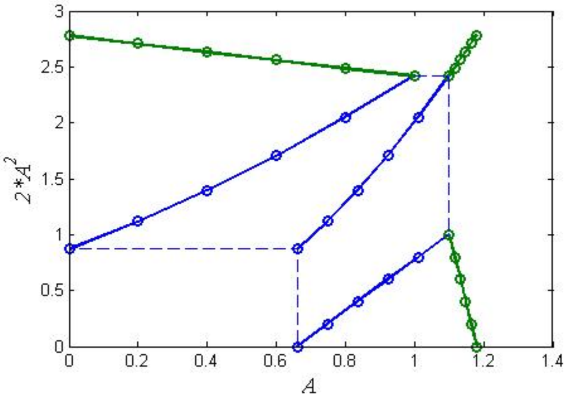


Figure 3. Fuzzy extension of a power function.

Linguistic fuzzy models [19], where both the antecedent and consequent are fuzzy propositions, suit very well to qualitative description of the process as they can be interpreted by using natural language, heuristics and common sense knowledge. The key idea is to use membership functions for both the inputs x and the outputs y. These functions can be defined by expert knowledge or by experimentation. The input-output mapping is realized by the fuzzy inference mechanism equipped with conversion interfaces, fuzzification and defuzzification. The approximate reasoning is based on T-norms and T-conorms [19].

The Takagi-Sugeno (TS) fuzzy modelling method was proposed by Takagi and Sugeno as a framework for generating fuzzy if–then rules from numerical data [20]. A TS fuzzy model consists of a set of fuzzy rules, each describing a local linear input–output relationship:

$$\begin{aligned} &If\ x_1\ is\ A_{i1}\ and \ldots and\ x_n\ is\ A_{in}\ then\ y_i = a_i x + b_i, \\ &i = 1,2,\ldots,K, \end{aligned} \qquad (3)$$

where A_{i1}, ..., A_{in} are fuzzy sets defined in the antecedent space and y_i is the rule output of the model. K denotes the number of rules. The results of the rules are usually combined as a weighted average where the weights are obtained from the fulfilment of the rules.

Fuzzy relational models [21], which allow one particular antecedent proposition to be associated with several different consequent propositions, can be regarded as generalizations of the linguistic fuzzy models. Each element of the relation represents the degree of association between the individual reference fuzzy sets defined in the input and output domains, i.e. all the antecedents are tied to all the consequents with different weights.

Dynamic fuzzy models can be constructed on the basis of state-space models, input-output models or semi-mechanistic models [22]. In the state-space models, fuzzy antecedent propositions are combined with a deterministic mathematical presentation of the consequent. The most common structure for the input-output models is the NARX (Nonlinear AutoRegressive with eXogenous input) model, in which the input and output values are chosen as in the ARX model according to appropriate system orders. The regressor vector consists of a finite number of past inputs and outputs [23]. This structure is directly used for multiple input, single output (MISO) systems. Multiple input, multiple output (MIMO) systems can be built as a set of coupled MISO models.

3.3 Neural modelling

Artificial neural networks consist of neurons

$$y_i = F(\sum_{j=1}^{m} w_{ij} p_j + B_i) \qquad (4)$$

where w_{ij} is the weight factor of the element p_j in the input vector of the neuron i, and B_i a scalar bias. For the input layer, the elements are usually normalised values of the variables x_j, $j=1,2,\ldots,m$.

Neurofuzzy systems use fuzzy neurons to combine the weight factors and the inputs. The activation function is handles as a function in the extension principle.

Dynamic ANN models are based on similar structures as the dynamic fuzzy models: simple structures, e.g. NARX structures, can constructed by taking delays into account in the input vector p in (2). A dynamic ANN model can be realised by a static feedforward network and an external feedback connection [23]. Another possibility is to use recurrent networks, e.g. Elman networks} are two-layer feedforward networks, with the addition of a feedback connection from the output of the hidden layer to its input [24]. This feedback path allows Elman networks to learn to recognize and generate temporal patterns, as well as spatial patterns. The weight factors w_{ij} can also depend on time.

4. LE modelling

Data-driven steady state modelling is normally used in linguistic equation (LE) modelling [11]. Dynamic structures extend the models to dynamic simulation, and in this paper uncertainty of the results is handled with fuzzy arithmetics.

4.1 Steady state LE modelling

Linguistic equation models consist of two parts: interactions are handled with linear equations, and nonlinearities are taken into account by (membership definitions) [10]. The output is obtained by

$$x_{out} = f_{out}\left(- \frac{\sum_{j=1, j\neq out}^{m} A_{ij} f_j^{-1}(x_j) + B_i}{A_{iout}} \right) \qquad (5)$$

Where parameters A_{ij}, $j=1,\ldots,m$, and B_i are the interaction coefficients of the linguistic equation i. Nonlinear scaling is based on membership definitions f_j and corresponding inverse functions f_j^{-1}. This model corresponds to the neural model (4) if the normalization is replaced by the nonlinear scaling.

In the LE models, the nonlinear scaling is performed twice: first scaling from real values to the interval [-2, 2] before applying linguistic equations and then scaling from the interval [-2, 2] to real values after applying linguistic equations (Figure 4). The linguistic level of the input variable j is calculated the inverse functions of the polynomials [11].

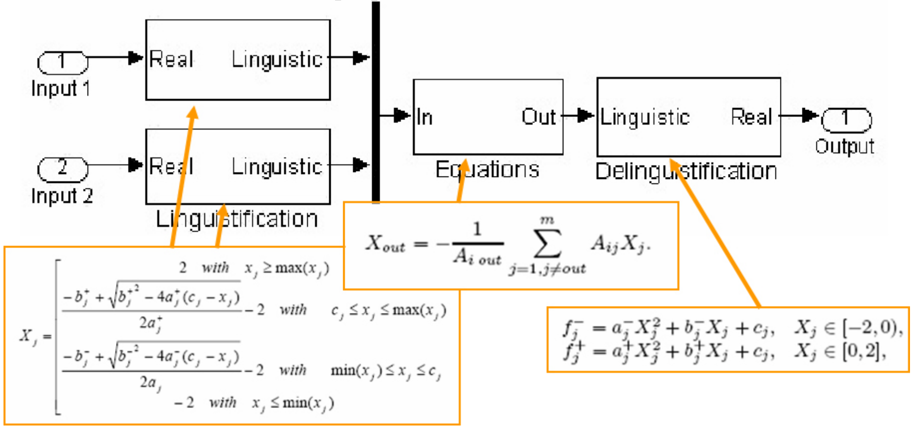

1
Input 1
Real
Linguistic
2
Input 2
Real
Linguistic
Linguistification
In
Out
Equations
Linguistic
Real
Delinguistification
1
Output
$X_{out} = -\frac{1}{A_{i\,out}} \sum_{j=1, j\neq out}^{m} A_{ij} X_j.$
$X_j = \begin{cases} 2 & \text{with } x_j \geq \max(x_j) \\ \frac{-b_j^+ + \sqrt{b_j^{+2} - 4a_j^+(c_j - x_j)}}{2a_j^+} - 2 & \text{with } c_j \leq x_j \leq \max(x_j) \\ \frac{-b_j^- + \sqrt{b_j^{-2} - 4a_j^-(c_j - x_j)}}{2a_j} - 2 & \text{with } \min(x_j) \leq x_j \leq c_j \\ -2 & \text{with } x_j \leq \min(x_j) \end{cases}$
$f_j^- = a_j^- X_j^2 + b_j^- X_j + c_j, \quad X_j \in [-2, 0),$
$f_j^+ = a_j^+ X_j^2 + b_j^+ X_j + c_j, \quad X_j \in [0, 2],$

Figure 4. A steady state LE model for two inputs and one output.

4.2 Dynamic LE modelling

The basic form of the linguistic equation (LE) model is a static mapping in the same way as fuzzy set systems and neural networks, and therefore dynamic models will include several inputs and outputs originating from a single variable. External dynamic models provide the dynamic behaviour. The models are developed for a defined sampling interval in the same way as in various identification approaches [16]. However, the LE simulators normally use variable time step integrators.

Rather simple input-output LE models, where the old value of the simulated variable and the current value of the control variable as inputs and the new value of the simulated variable as an output, can be used since nonlinearities are taken into account by membership definitions. To use integration methods available in the simulation software, a difference of the output is calculated (Figure 5).

Nonlinear scaling reduces the order of the model, i.e. the number of input and output signals needed for modelling of nonlinear systems. Need for higher order models can be tested by applying classical identification with different polynomial degrees to the data after scaling with membership definitions. For the default LE model, all the degrees of the polynomials in parametric models become very low, i.e. all the parametric models become the same, ARX model shown in (2).

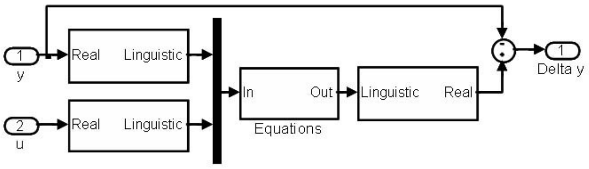


Figure 5. Dynamic LE model of Δy.

Several artificial neural networks have been compared for expanding the linear models, but these more complex model structures do not provide any considerable improvement to the results obtained by the basic LE models, i.e. a linear activation function can be chosen in (4) if the nonlinear scaling described above is used for the input variables.

Changing operating conditions can be taken into account by modifying membership defintions and/or interaction coefficients of the LE models. Linguistic fuzzy models can be used for selecting submodels. This approach is used for selecting the appropriate submodels for spraying and drying.

Also structures used Takagi-Sugeno type fuzzy models can be used if the interaction coefficients depend clearly on the input variables.

4.3 Fuzzy LE modelling

Universal approximators for fuzzy functions can be constructed as extension principle extensions of continuous real-valued functions which continuously map fuzzy numbers into fuzzy numbers [18, 25]. LE models can extended to fuzzy inputs with this approach if the membership definitions and the corresponding inverse functions, are replaced by corresponding extension principle extensions of these functions presented in Figure 4.

The argument of the function f_{out} in (1) is obtained by fuzzy arithmetics. Here the calculations are based on interval analysis which has been widely used in physics for handling measurement errors. In this methodology, measurement values are assumed to be on intervals whose lengths depend on the accuracy of the measurements. The interval analysis is used for estimating the intervals of calculated variables [26]:

$$\begin{aligned}
&[a,b]+[c,d]=[a+c,b+d],\\
&[a,b]-[c,d]=[a-d,b-c],\\
&[a,b]\cdot[c,d]=[\min(ac,ad,bc,bd),\max(ac,ad,bc,bd)],\\
&[a,b]\big/[c,d]=[a,b]\cdot\left[\frac{1}{d},\frac{1}{c}\right] \; if \;\; 0\notin[c,d],
\end{aligned} \qquad (6)$$

where intervals [a, b] and [c, d] are arbitrary real intervals.

The original interval analysis does not include any gradual approach, but the methodology can be generalized to horizontal membership functions (Figures 3 and 6) by applying interval analysis on each α-cut separately. The number of α levels should be increased when the fuzziness of the input increases.

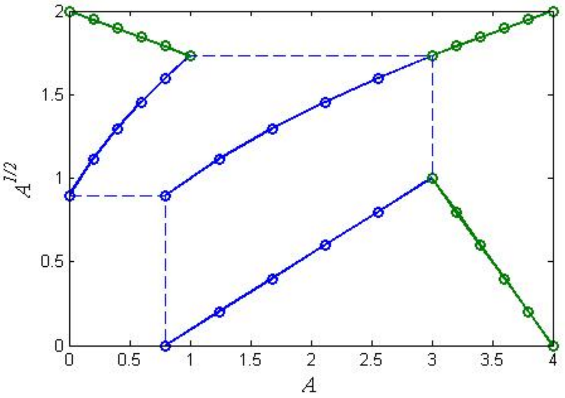


Figure 6. Fuzzy extension for a square root function.

Only addition and subtraction are needed if the interaction coefficients are crisp. The only fuzzy part in the linguistification function is the term which includes the input x_j (Figure 4). The extension principle is used to obtain a square root of the fuzzy number (Figure 6). The width and location of the input fuzzy number are modified by the parameters of the scaling functions.

The output fuzzy numbers of these blocks are limited to the range [-2, 2].

In the equation block presented by (2), the state variable *y(t-1)* and the input variable *u(t-n_k)*, which are both fuzzy numbers, are multiplied by crisp numbers – A_1 and A_2, respectively, if the model is crisp. The sum of the resulting fuzzy numbers is the new state variable *y(t)* in the range [-2, 2].

In the delinguistification block, the terms X_j^2 can be obtained by the extension principle (Figure 3) or by multiplication $X_j \cdot X_j$ with fuzzy interval analysis. The resulting fuzzy number and the original fuzzy number X_j multiplied by crisp numbers shown in Figure 4 are then added to the crisp number c_j to obtain the fuzzy output.

Fuzzy LE models with fuzzy inputs can be constructed by using fuzzy multiplication and division as well since the parameters a_j^-, b_j^-, a_j^+, b_j^+ and c_j are all fuzzy numbers. Fuzzy extension of the classical interval analysis suits very well also to these calculations. However, the result becomes naturally more uncertain when fuzzy models are used.

Results of the fuzzy interval analysis have always maximal uncertainty as it takes the worst case. A negative associations between the input variables reduces the uncertainty considerably. In the calculations, this can be taken into account by using own membership functions for the upper and lower parts of the value range.

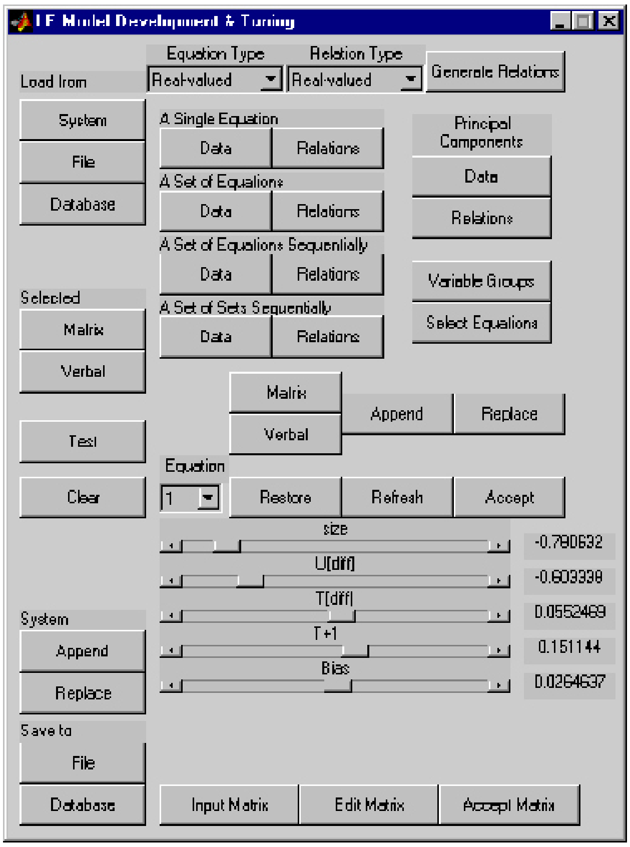
5. Dynamic simulator

Data was separated to three main processes: mixing, spraying and drying. Modelling for mixing area has not done so far because of insignificant changes in the humidity and the airflow. Thus physical knowledge of mixing process is not well known. The aim of mixing step is to make a homogeneous batch.

In the beginning of spraying the physical changes were very rapid and that part of process has been the most difficult area to model. The weight of granules increase rapidly, i.e. the inlet air has to be increased significantly to maintain fluidising. Correspondingly during the first few minutes of the drying, the surface drying proceeds quickly until the balance is founded. The granule growth may still continue for some time in the beginning of the drying phase.

The overall model for the spraying and drying phases consists of three models:

- temperature,
- humidity, and
- granule size.



Equation Type: Real-valued
Relation Type: Real-valued
Equation: 1

size -0.780632
U[diff] -0.603338
T[diff] 0.0552469
T+1 0.151144
Bias 0.0264637

Figure 7. Model development and tuning in FuzzEqu Toolbox.

Output variables were the temporary value of the granule temperature, the new value of humidity difference and the new estimated value of the granule size, correspondingly. The dynamic submodels have similar structures as shown in Figure 5 and model specific variables:

- The new granule size depends on the current granule size and two other variables, temperature difference and humidity difference.
- The granule temperature depends on airflow (F_{in}), humidity difference between inlet and outlet air (U_{diff}) and temperature between granule and inlet air (T_{diff}).
- The humidity depends on T_{diff}, U_{diff} and granule temperature.

The distribution of the particle size is based on the fuzzy extension principle, i.e. the membership function of the particle size is computed in each time step from the uncertain input values by using the dynamic LE model as a function. In this way the uncertainty of the model is not forgotten in the analysis.

The system is able to select automatically the best submodel during the granulation process and move gradually from one submodel to another when the process proceeds by fuzzy methods.

The LE models have been developed and tuned in the FuzzEqu Toolbox: the LE model in Figure 7 is a model of the granule size. Membership definitions have been developed from the data: a batch specific example is shown in Figure 8.

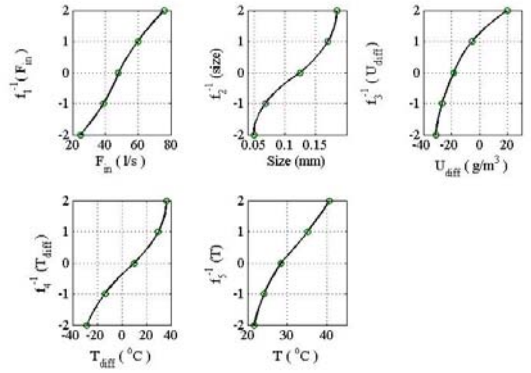


Figure 7. Membership definitions.

6. Results and discussion

Testing data was collected from 38 batches, 27 batches were used to training and the rest of the 11 to testing. Stable data for modelling was obtained by filtering. The modelled and simulated results were compared with experimental data. According to the modelling and simulation results, the most representative input variables were airflow in (F_{in}), humidity difference between inlet and outlet air (U_{diff}) and temperature between granule and inlet air (T_{diff}). In Simulink model, also other input variables were used but the interaction coefficients of these variables are rather small.

Modelling is aimed to help in estimating the granule size while processing since the analysis result of samples is not available on line. The granulation goes through considerably different routes depending on the operating conditions. By controlling the interaction coefficients of the variables the model worked well also in rapidly changing areas.

The temperature of granules varied mainly from 20 to 60 ^{o}C. The data of the batches were shared to several subperiods, and the membership definitions were made for every area. In the linguistic equation model, the variables are scaled between –2 and +2. The value range of the variables must be wide enough to guarantee the applicability to the modelling.

Fuzzy modelling is a reasonable extension as also the granule size has always a distribution rather than a single value. This distribution changes with time, and the result becomes more and more uncertain when the prediction horizon increases. Negative associations between T_{diff} and U_{diff} alleviate this problem slightly.

The first results show that the complexity of the models is increased only slightly with the new system based on the extension principle and fuzzy interval analysis. This study will be extended to the complete data set as it provides a lot of useful additional information about the granulation process. In future, the results will be compared to the measured distributions of the granule size. The goal is to develop more general models, i.e. the membership definitions will be developed from the complete data set related to the batches of the verapamil granulations.

Partly the uncertainty is caused by uncontrolled process conditions, e.g. seasonal variation in humidity is considerable and that will cause the changes in the water amount of the incoming air. The incoming air humidity should be included to the input variables or another possibility is to make a pre-moistening to a constant moisture value in the beginning of granulation. Later a humidifying system has been included, and this enables high and fluctuating humidity of the process air.

7. Conclusions

The data based modelling succeeded well after the main process stages were divided into sub stages corresponding to shorter time periods. The interaction of the main variables was improved by using fuzzy modelling. Extension to fuzzy LE models provides useful information about uncertainties of the forecasted granulating results. The complexity of the models is increased only slightly with the new system based on the extension principle and fuzzy interval analysis. Associations between input variables were useful in reducing the uncertainty of the final result.

References

[1] J. Rantanen, S. Lehtola, P. Rämet, J.-P. Mannermaa,.- and J. Yliruusi (1998). ”On-line monitoring of moisture content in an instrumented fluidized bed granulator with a multi-channel NIR moisture sensor”, Powder Technology, Vol. 99, 163-170.

[2] M. E. Aulton (1992), “Pharmaceutics: The science of dosage form design”, Churchill Livingstone, New York.

[3] R. Bergman, M.E. Johansson, T. Lundstedt, E. Seifert and J. Åberg (1988). “Optimization of a granulation and tabletting process by sequential design and multivariate analysis”, Cheometrics and Intelligent Laboratory Systems, Vol 44, 271-286.

[4] E. A. Colburn and R. C. Rowe (1996). “Modelling and optimization of a tablet formulation using neural networks and genetic algorithms”, Pharmaceutical Technology Europe, Vol 8, 46-55.

[5] H. G. Kristensen (1995). “Advances in pharmaceutical sciences”. Academic Press, 221-271.

[6] T. Lipsanen, O. Antikainen, H. Räikkönen, S. Airaksinen and J. Yliruusi (2007). ”Novel description of a design space for fluidised bed granulation”, Int. J. Pharmaceut.(2007), doi:10.1016/j.ijpharm.2007.05.051

[7] F. Thielmann, M. Naderi, M. A. Ansari, F. Stepanek (2007)." The effect of primary particle surface energy on agglomeration rate in fluidised bed wet granulation", Powder Technology (2007), doi:10.1016/j.powtec.2006.12.015

[8] S. Heinrich and L. Mörl (1999). "Fluidized bed spray granulation—A new model for the description of particle wetting and of temperature and concentration distribution", Chemical Engineering and Processing 38 (1999) 635–663.

[9] E. K. Juuso and K. Leiviskä. Adaptive expert systemsfor metallurgical processes. In S.-L. Jämsä-Jounela and A. J. Niemi, editors, Expert Systems in Mineral and Metal Processing, Proceedings of the IFAC Workshop, Espoo, Finland, August 26-28, 1991, IFACWorkshop Series, 1992, Number 2, pages 119–124, Oxford, UK, 1992. Pergamon.

[10] E. K. Juuso (1999). "Fuzzy Control in Process Industry: The Linguistic Equation Approach". In: Verbruggen, H. B., H.-J. Zimmermann and R. Babuska, editors, Fuzzy Algorithms for Control, International Series in Intelligent Technologies, pp. 243-300. 1999, Kluwer, Boston.

[11] E. K. Juuso (2004). "Integration of intelligent systems in development of smart adaptive systems. International Journal of Approximate Reasoning, 35:307–337.

[12] E. K. Juuso (2004). "Modelling and simulation with intelligent methods". In White paper of theVirtual Institute for Simulation (Sim-Serv): www.simserv.com. Sim-Serv, 2004. 17 pp.

[13] T. Mäki, and E. K. Juuso (2000). "Multiple Model Dynamic Simulation of a Fluidised Bed Granulator with Linguistic Equations". Proceedings of TOOLMET 2000 Workshop, Oulu, April 13-14, 2000, pp. 78-89, Oulu. Oulun yliopistopaino.

[14] T. Mäki, E. Juuso and K. Leiviskä (2004). "Fuzzy Modelling and Dynamic Simulation of a Fluidized Bed Granulator". In Proceedings of AFNC 2004- The 2nd IFAC Workshop on Advanced Fuzzy/Neural Control. September 16-17, 2004, Oulu, Finland, pp. 133-138.

[15] J. Rantanen (2000). "Near-Infrared Reflectance Spectroscopy in the Measurement of Water as a Part of Multivariate Process Monitoring of Fluidised Bed Granulation Process", Dissertationes Biocentri Viikki Universitatis Helsingiensis 21/2000, pp. 46.

[16] L. Ljung (1999). "System Identification - Theory for the User". Prentice Hall, Upper Saddle River, N.J., 2nd edition.

[17] L. A. Zadeh (1965). "Fuzzy sets", Information and Control, 8:338-353.

[18] J. J. Buckley and Y. Hayashi (1999). "Can neural nets be universal approximators for fuzzy functions?", Fuzzy Sets and Systems, 101:323–330.

[19] D. Driankov, H. Hellendoorn, and M. Reinfrank (1993). "An Introduction to Fuzzy Control", Springer, Berlin, Germany.

[20] T. Takagi and M. Sugeno (1985). "Fuzzy identification of systems and its applications to modelling and control", IEEE Trans. Syst., Man, & Cybern. 15(1):116--132.

[21] W. Pedrycz (1984). "An identification algorithm in fuzzy relational systems", Fuzzy Sets and Systems 13:153--167.

[22] R. Babuska, M. Setnes, U. Kaymak and H. B. Verbruggen (1997). "Fuzzy Modelling: a Universal and Transparent Tool". In: Yliniemi, L. \& Juuso, E. (eds.), Proceedings of TOOLMET'97- Tool Environments and Development Methods for Intelligent Systems, Oulu, April 17-18, 1997}. Oulun yliopistopaino, Oulu, pp. 1-27.

[23] R. Babuska and H. Verbruggen (2003). "Neuro-fuzzy methods for nonlinear system identification", Annual Reviews in Control 27:73--85.

[24] J. L. Elman (1990). "Finding structure in time", Cognitive Science 14:179--211.

[25] J. J. Buckley and T. Feuring (2000). "Universal approximators for fuzzy functions", Fuzzy Sets and Systems, 113:411–415, 2000.

[26] R. E. Moore (1966). "Interval Analysis". Prentice Hall, Englewood Cliffs, NJ.

A Meta-Modeling Environment for Mechanical System Co-Simulations

Alexander Siemers
Dept. of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping - Sweden
alesi@ida.liu.se

Dag Fritzson
SKF Engineering Research Centre
MDC, RKs-2
SE-415 50 Göteborg, Sweden
Dag.Fritzson@SKF.com

Abstract

A general approach for modeling of mechanical system co-simulations is presented that is built upon the previously defined general framework for TLM co-simulations and co-simulation meta-modeling [1].

Co-simulation is one technique for coupling different simulators into one coherent simulation. Existing co-simulation applications are often capable of interconnecting two specific simulators where a unique interface between these tools is defined. However, a more general solution is needed to make co-simulation modeling applicable for a wider range of tools. Any such solution must also be numerical stable and easy to use to be applicable by a larger group of people.

In this work the concept of meta-modeling is applied to mechanical co-simulation. Several tool-specific simulation models can be integrated and connected by means of a meta-model, where the meta-model defines the physical interconnections of these models.

A general meta-modeling process is described that represents the basis for this work. A meta-modeling language (MML) has been defined to support the modeling process and store the meta-model structure. Besides elements for physical interconnections, etc., the language also defines graphical elements that can be used for meta-model visualization. All proposed solutions are general and simulation tool independent.

A fully functional modeling environment has been created to make meta-modeling applicable. The modeling environment supports easy encapsulation and integration of simulation tool-specific models. Each simulation tool implements a single, well defined co-simulation interface. All interfaces implement a numerically stable method for force/moment interaction. The presented environment features a graphical user interface for co-simulation modeling with support for three dimensional visual representation of the co-simulation model including all its components.

1. Motivation

In the area of modeling and simulation of mechanical systems one can identify many different classes of models and corresponding tools, each optimized for a certain kind of analysis. Typically each model is defined in some specific language together with some modeling and simulation tool that can perform a transient simulation of it. The examples of such models are equation-based multi-physics Modelica [7] models, multi-body models in MSC.ADAMS [21], models with detailed contact definitions in SKF's BEAST [8], control systems in Simulink, flexible models as modeled in finite element (FE) tools, etc. In reality the different systems represented by these models are often physically tightly coupled and interdependent. Combining the different models into a single, coupled, simulation enables thus for a more complete and exact system analysis while at the same time preserving the investments in these models.

The general question is: *How to model co-simulations and how to make co-simulation modeling generally usable*. Typical co-simulations interconnect two specific simulators where a unique interface between these tools is defined. The interconnected, tool specific, models and the co-simulation set-up require expert knowledge. To make co-simulation modeling adaptable to wider range of tools and usable by a larger group of people more general solutions are needed. Any such solution must be adaptable to many existing simulation tools and easy to use from a user perspective.

Environments for modeling of interacting concurrent components typically use two dimensional block diagram and interface connection models, e.g., the model-based Integrated simulation framework [12] (MILAN) whose user interface is based on the generic modeling environment [13] (GME), Cosimate [18], and Ptolemy II [11]. This is sometimes called data flow modeling or actor oriented design (AOD) [16]. In such systems the so called actors communicate by means of message passing through well defined interfaces. A similar approach for co-simulation meta-modeling has been presented in [1]. However, the three dimensional topological structure is important for modeling of mechanical systems. A fully three-dimensional view of the co-simulation meta-model including all external-models is therefore desirable, and has to our knowledge not been previously developed.

2. Meta-Modeling

Generally Meta-Modeling is semantic or domain specific modeling [19] [20] [14], where a meta-model is the semantic information model that describes the vocabulary of a specific modeling domain. In the context of this work a meta-model defines the physical interconnections of different external models, where an external model is a model defined in some specific language together with some modeling and simulation tool that can perform a simulation of it. By interconnecting several external models an (extended) overall structure of the system model is created. The meta-model defines the semantics of each external model and their interconnections within this structure.

A meta-modeling process has been defined that forms the basis of this work. The process can be divided into three steps:

- *External model design* in a specialized environment. Each external model in the co-simulation is modeled separately in its specific environment.

- *Model integration* into the meta-modeling environment. First of all the external-interfaces are defined in the external models. Subsequently, all external models need to be encapsulated for integration into the meta-model. Startup methods, interface names, and possibly geometry data must be specified.

- *Meta-Model design* in the meta-model editor. The different external models are integrated into the meta-model and connect with each other. Global simulation parameters need to be defined as well.

The process is also shown in Figure 1. There are different levels of expertise required in the different phases of the modeling process. All work in a specialized simulation environment requires a high level of expertise about this environment. The meta-model design phase, on the other hand, requires less expertise of any of the specialized environments and is based on a simple easy-to-use meta-model editor. The level in between requires little expert knowledge, i.e., to add the external interfaces to the external models. This phase can be performed by the external model designer, the meta-model designer, or a combination of both.

2.1. Meta Model Representations

One important aspect concerns the different representations of the meta-model throughout the modeling and simulation process. The following meta-model representations have been identified in this work:

- *Language representation.* This is the fundamental representation used to store the meta-model and to communicate meta-model information.

- *File representation.* This is a mapping of the meta-model hierarchy into a file and directory structure. This is needed to store and organize external models, i.e., files needed to execute and visualize an external model.

- *Visual representation.* This is used during meta-modeling for visual model verification and improved usability. Typical visual representations are:
  - *Hierarchical view* of the meta-model structure.
  - *Connection view*, showing the connections between the external models.
  - *System view*, showing a three dimensional representation of the complete system including all external models.

- *Simulation representation.* This describes where the different external models are executed and how the communication between them is organized.

Most fundamental is the language representation that is needed as input for the other representations, see Figure 2. It is a textual representation that contains language elements to support the file representation and the different visual representations.

To keep the meta-model portable is is important to separate the system-setup from the meta-model itself. Instead, the meta-model simulator must be adapted to the

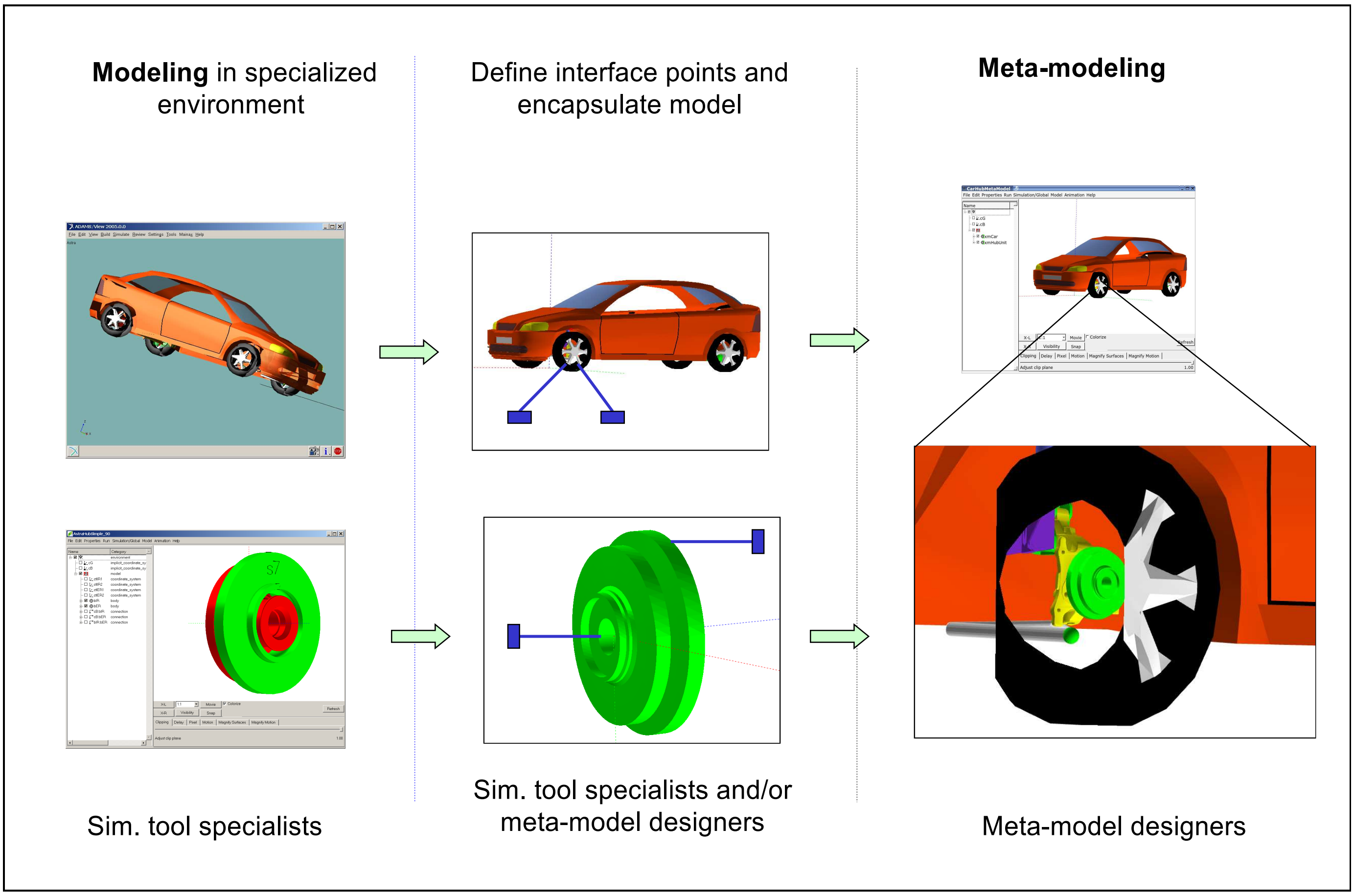

Modeling in specialized environment
Define interface points and encapsulate model
Meta-modeling
Sim. tool specialists
Sim. tool specialists and/or meta-model designers
Meta-model designers

Figure 1. External models from specialized tools are encapsulated and integrated into the meta-model.

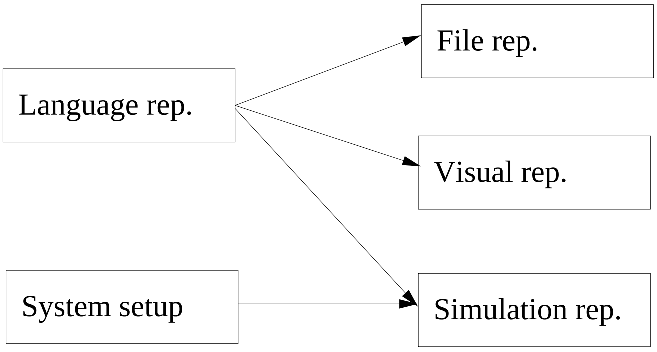


Figure 2. Dependencies between the different Meta-Model representations.

specific system-setup. The simulator reads the meta-model and creates the simulation representation based on the system-setup.

2.2. Meta Modeling Language

The basic concept of a meta-model is the definition of external models and the connections between them. External models use external interfaces to communicate forces and moments. Note that there is no limitation to the type of external interfaces in the meta-model, but this work focuses on mechanical system modeling where forces and moments are of interest.

A language representation with strict notation and grammar is needed to represent and store the meta-model and to share meta-model information between different tools and people. The meta-modeling language (MML) has been defined for this purpose. In [1] an XML-based meta-modeling language was presented. The language has later been redesigned to meet the following requirements:

- Remove XML-tags to improve readability
- Add hierarchical constructs to allow for hierarchical meta-models

- Allow for geometrical elements for improved meta-model visualization

MML is a domain specific language [17] designed for co-simulation meta-modeling primarily in the mechanical system domain. The aim was to create a simple and restricted language that allows simulation experts to solve problems involving different simulation models by means of a meta-model.
Meta-Models are organized in a hierarchical fashion building the meta-model tree. These are classes of objects (here called categories) each containing data and representing a certain functionality. In detail the following categories and hierarchical composition rules exist in MML:

- *Environment* is the top most node in the Meta-Model. It contains the Meta-Models inertial system and the root Model. Global co-simulation parameters are specified here, these are: model name, simulation start time, and simulation end time.

- *Models* are used as intermediate level nodes in the Meta-Model. Models define different hierarchical sub-levels. They contain other Models, XModels, and Connections.

- *XModels* are encapsulated external simulation models. Each XModel (external-model) represents a participant in the co-simulation. They contain: information about the simulation files and startup methods, necessary coordinate transformations, one or more control-points, and one or more XGeometries.

- *Coordinate-systems*, also called *control-points*, represent the external interfaces in the meta-model. They contain an interface name and information about the motion of the interfaces coordinate-system relative to the external models local inertial system.

- *Implicit-coordinate-systems* define the meta-models global inertial system and the external model local inertial systems.

- *XGeometry* contains the external-models graphical information for 3D visualization.

- *A Connection* connects exactly two XModels. A connection is a container for all the ties between any two XModels.

- *Tie.* Each tie interconnects exactly two control-points. They store necessary connection parameters used by the manager of the co-simulation.

Here is an outline of the MML file for a “car with hub” model:

```
environment type=Environment
  def time=0 unit="s";
  def StartWriteTime=0 unit="s";
  def timeEnd=2.5 unit="s";
  model type=SModel name=""
    xmodel type=XModel name=AdamsCar
      def StartCommand="StartTLMAdams";
      def SimulationFile="CarTLM.acf";
      def ExtraSimFiles="CarTLM.bin,
                         CarTLM.cmd";
      def cX‘R[cG][cG]={0, 0, 0} unit="m";
      def cX‘phi[cG]={0, 0, 0} unit="rad";
      def cX‘phiSequence[cG]=0;
      coordinate_system type=XCtlPoint
                        name=ctTLMknuckle
        def TLMPortName="M17305495";
        def R[cX]={0,
                   0.7342,
                   0.2624 }
                   unit="m";
        def phiSequence[cX]=0;
        def phi[cX]={3.1415926536,
                     1.5707963268,
                     1.5707963268}
                     unit="rad";
      end coordinate_system;
      ...
      xgeometry type=XGeometry name=xs1
        def CADFileName="Car.wrl";
      end xgeometry;
    end xmodel;
    ...
    connection type=XModelXModelConnection
               from=AdamsCar to=xmHubUnit
      tie type=TLMTie
          from=ctTLMknuckle to=ctbER‘ctl2
        def delay=2.0e-5;
        def Zf=1.0e4;
        def Zfr=100;
        def alpha=0.01;
      end tie;
      ...
    end connection;
  end model;
end environment;
```

3. Meta-Modeling Environment

A meta-modeling environment has been designed to support the whole modeling process in all its phases. The environment is simulation tool independent, easy to use, and supports model verification throughout the whole modeling process.

The method that is used to enable stable interaction between dynamic models in the co-simulation is *transmission line modeling* (TLM) [2]. TLM uses physically motivated time delays to separate the models in time and enable efficient co-simulation. Only TLM connections are currently supported by the meta-modeling environment because the TLM method, when used correctly, gives numerical stability [5] [4] [3] [6].

In earlier work meta-models were designed using a graphical Modelica editor and stored as Modelica models [1]. Here this work has been carried on to improve meta-model simulation pre- and post-processing, i.e., fully three dimensional model editor and meta-model animations. For practical reasons the realization has been based on the tools of an existing simulation environment, called BEAST [8].

The meta-model editor (MME) features three dimensional representations of all meta-model components, see Figure 3, plus hierarchal view of the complete model-tree. External models can be imported into the meta-model. All co-simulation parameters, i.e., start and stop time, maximum time-step, and TLM related parameters are stored in the meta-model.

Consider modeling a vehicle, see Figure 3. MSC.ADAMS seems to be a “standard” tool within this field. BEAST on the other hand allows for detailed modeling of hub units. Correct location of the hub-unit in the car wheel can be verified visually.

A meta-model is typically created in three steps:

1. Import or add the required *external models*.
2. Add *connections* between any two *external models* that are physically interconnected.
3. Add the *TLM-Ties* between any two *control-points* (TLM-interfaces) that build a physical connection. Ties are automatically placed inside the *connections*.
4. Set co-simulation and TLM specific parameters in the meta-model.

3.1. External Models

The TLM framework [1] defines a strict interface for all external models. This is important for meta-modeling because all external models can be treated in a general way. Each external model needs at least a startup method, i.e., the name of the specific simulation tool it belongs to, a simulation input file (or model file), and position and name of all TLM interface points. External models can be created (or encapsulated) using a small platform independent application that can easily be distributed to all model owners. Those who own a certain simulation model can encapsulate and verify the external model at their location before distributing and sharing it with others. The same application can be started from within MME to directly encapsulate models inside MME.

Correct identification and placement of the TLM interface points within the meta-models inertia system is important. This data can be manually extracted from the simulation model. A more convenient and safe solution is to automatically read the information through the TLM-Interface. This requires to start the simulation of the particular external model, read the values, and exit the simulation again. The TLM-framework [1] has been extended to support request of TLM interface data. The data request has been automated in MME. Another important aspect of this feature is the possibility for verification of correct model execution during external model encapsulation.

3.2. Interface Alignment

External models exchange force and motion in the TLM interface points. A correct meta-model requires that any two connected TLM interface points have the same start position and orientation relative the meta-models global inertial system. In some cases the different modeling tools use a different local inertial system. This leads to misalignment in the connected interfaces and thus to errors in the meta-model. To avoid redesign of the external model in its specific modeling environment MME allows for aligning any two selected interface points automatically. The hub-unit in Figure 3, for instance, was originally not located in the wheel of the car but centered in the global inertial system. By aligning two of the connected interface points the hub-unit was automatically positioned correctly. The calculated transformation is stored in the XModels (here the hub-unit) specific TLM-plug-in and applied before any data exchange.

4. Verification

A pendulum with three connected shafts has been designed as the test meta-model, see Figure 4. Each shaft is modeled as a separate external model.

The external models for the shafts have been created for Simulink using SimMechanics, MSC.ADAMS, and BEAST. The model is kept simple to avoid errors in

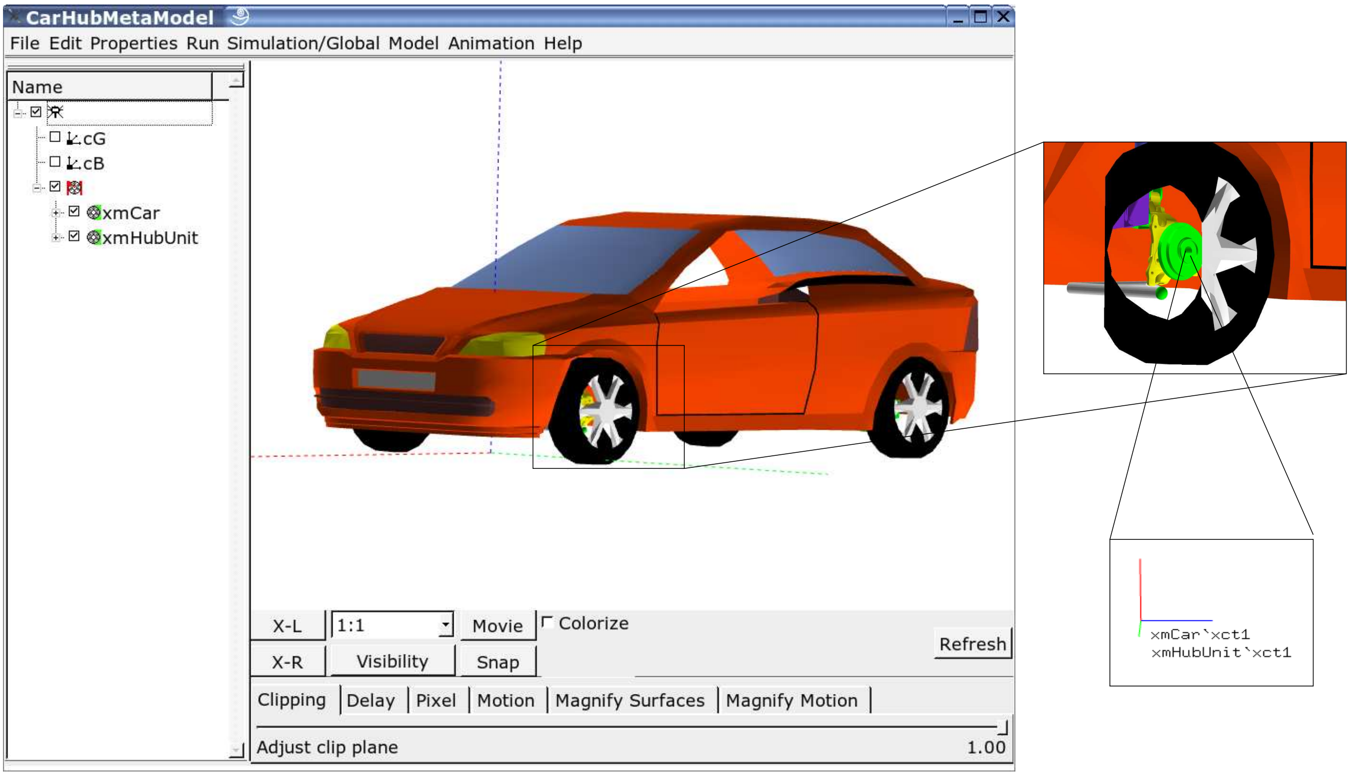


Figure 3. *The meta-model of a* MSC.ADAMS *car model with integrated* BEAST *hub-unit model (green). Correct placement of all components including TLM interfaces is automated and can be verified within the meta-model editor.*

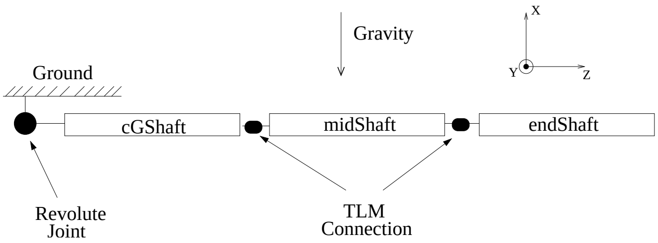


Figure 4. Three shaft pendulum with TLM connections between the shafts and a revolute joint connected to the ground.

the simulation tool-specific models, e.g., the Simulink model. All shafts have the same properties in all simulation tools. Meta-models with different combinations of the shafts, e.g., BEAST-Simulink and BEAST-MSC.ADAMS, have been tested to verify meta-model simulation and meta-model editor.

5. Conclusion

The complete meta-model process for mechanical co-simulation modeling has been presented. It provides the basis for a general meta-modeling approach for mechanical system co-simulation.

A meta-modeling language (MML) has been defined. MML is human readable and editable so that models can be designed, edited, and verified using a standard text editor. The textual representation also simplifies parsing and translation from and to other modeling languages. MML allows hierarchical models to express hierarchical containment relations.

A meta-modeling environment has been presented that features simplicity and model verification throughout the whole modeling process. It contains a meta-model editor with support for fully three-dimensional representation of the complete meta-model. Calculation of necessary transformation of external-models relative to the meta-models global inertial system is automated to avoid time consuming changes of the simulation models. External model integration and interface specification is automated. The environment fully supports the complete meta-model process in all its phases.

References

[1] Siemers, A. and Nakhimovski, I. and Fritzson, D., *Meta-modelling of Mechanical Systems with Transmission Line Joints in Modelica*, Proceedings of the 4th International Modelica Conference, Hamburg, Germany, 2005

[2] Nakhimovski, I., *Contributions to the Modeling and Simulation of Mechanical Systems with Detailed Contact Analysis*, Dissertation No. 1009, Linköpings universitet, Sweden, 2006

[3] Krus, P. and Jansson, A, *Distributed Simulation of Hydromechanical Systems*, 'Third Bath International Fluid Power Workshop', Bath, UK 1990

[4] Krus, P., *Modelling of Mechanical Systems Using Rigid Bodies and Transmission Line Joints*, Transactions of ASME, Journal of Dynamic Systems Measurement and Control. Dec 1999

[5] Johns, P. B. and O'Brien, M., *Use of the transmission-line modelling (t.l.m.) method to solve non-linear lumped networks*, The Radio and Electronic Engineer, 1980.

[6] Pulko, S.H. and Mallik, A. and Allen R., and Johns, P.B., *Automatic Timestepping in TLM Routines for the Modelling of Thermal Diffusion Processes*. Int. Journal of Numerical Modelling: Electronic Networks, Devices and Fields, 1990.

[7] Fritzson, P. *Object-Oriented Modeling and Simulation with Modelica 2.1*, Wiley-Interscience, 2004

[8] Stacke, L-E. and Fritzson, D. and Nordling, P., *BEAST—a rolling bearing simulation tool*, Proc. Instn Mech. Engrs, part K, Journal of Multi-body Dynamics, 1999,

[9] *High Level Architecture (HLA) - Framework and Rules*, IEEE Standard for Modeling and Simulation, IEEE 1516-2000, Institute of Electrical and Electronics Engineers, May 2000

[10] Kuehl, F. and Weatherly, R. and Dahmann, J., dahmann *Creating Computer Simulation Systems, An Intruduction to the High Level Architecture*, Prentice Hall PTR, 1999

[11] Center for Hybrid and Embedded Software Systems (CHESS) in the Department of Electrical Engineering and Computer Sciences of the University of California at Berkeley, *http://ptolemy.eecs.berkeley.edu/ptolemyII*.

[12] Agrawal, A. and Bakshi, A. and Davis, J. and Eames, B. and Ledeczi, A. and Mohanty, S. and Mathur, V. and Neema, S. and Nordstrom, G. and Prasanna, V. and Raghavendra, C. and Singh, M., *MILAN: A Model Based Integrated Simulation Framework for Design of Embedded Systems*, Workshop on Languages, Compilers, and Tools for Embedded Systems (LCTES 2001), Snowbird, Utah, June 2001.

[13] Ledeczi, A. and Maroti, M. and Bakay, A. and Karsai, G. and Garrett, J. and Thomason, C. and Nordstrom, G.and Sprinkle, J. and Volgyesi, P. *The Generic Modeling Environment*, Proceedings of WISP'2001, May 2001, Budapest, Hungary.

[14] Lara, J. and Vangheluwe, H., *Using AToM^3 as a Meta-CASE Tool*, 4th International Conference on Enterprise Information Systems, Ciudad Real, Spain

[15] Larsson, J. and Johansson, B. and Krus, P. and Sethson, M., *MODLITH: A Framework Enabling Tool-Independent Modelling and Simulation*, European Simulation Symposium 2002, Dresden, Germany

[16] Lee, E., *Model-Driven Development - From Object-Oriented Design to Actor-Oriented Design*, Extended abstract of an invited presentation at Workshop on Software Engineering for Embedded Systems: From Requirements to Implementation (a.k.a. The Monterey Workshop) Chicago, Sept. 24, 2003

[17] van Deursen, A. and Klint, P. and Visser, J., *Domain-Specific Languages: An Annotated Bibliography*, SIGPLAN Notices, volume 35, number 6, pages 26-36, 2000

[18] TNI Software, Cosimate - Heterogeneous co-simulation in distributed environments, *http://www.tni-software.com*

[19] Community site for meta-modeling and semantic modeling, *http://www.metamodel.com*.

[20] MetaCase, MetaEdit+ DSM environment, *http://www.metacase.com*

[21] MSC-Software, MSC.ADAMS - interactive motion simulation software, *http://www.mscsoftware.com*

State/Parameter Estimation of a Small-scale CHP model

Juan I. Videla
Telemark University College
juan.videla@hit.no

Bernt Lie
Telemark University College
bernt.lie@hit.no

Abstract

The state/parameter estimation problem is studied for a small-scale ICE CHP model. Three main groups of estimators with significant performance and complexity differences are analyzed: the Extended Kalman Filter (EKF) as an extension of the classical Kalman filter, the generalized unscented Kalman filter (UKF) that uses the unscented transformation, and particle filtering like the particle filter with resampling (PFr) and the Ensemble Kalman Filter (EnKF)

The internal combustion engine is modeled as a mean-value engine model connected with a static generator model and the heat recovery circuit is modeled with two lumped heat exchanger models, one for the coolant circuit and the other for the exhaust gases. The coolant circuit is connected with the engine through a lumped inner engine thermal model.

Experimental data sets are artificially generated to test the different estimators. Dynamic parameters of the mean-value engine model are identify when the CHP model is simulated in open loop. Additionally, relevant heat transfer coefficients of the heat recovery circuit are monitored when the model is simulated in closed loop.

1. Introduction

There is a growing interest in the use of small-scale and micro cogeneration systems in the residential sector to produce both useful thermal energy and electricity. A review and comparison of these systems presented in [1] indicates that internal combustion engine (ICE) based combined heat and power (CHP) units are the most suitable products available today in the market for the residential sector. However, they do need regular maintenance and servicing, and most of them operate with simple high level control policies (e.g. on/off). Advanced controllers (e.g. optimization based techniques) and better monitoring techniques can be used with an online state/parameter estimator to reduce the operating costs.

It is of interest to study the state/parameter estimation problem of a small-scale CHP unit with the purposes of control, maintenance, and identification.

Online parameter identification of a CHP unit can be achieved using recursive state/parameter estimators. Fast and slow varying parameters tracking is of interest [2] not only to identify the physical parameters but also to detect failures and components degradation. Additional difficulties for the estimation problem include: nonlinearities, a limited number of noisy measurements, unmodeled dynamics, and simplifying assumptions.

For linear systems with normally distributed process and measurement noise the optimal recursive estimator is the Kalman filter [3]. State estimation for nonlinear systems is considerably more difficult and admits a wider variety of suboptimal solutions [4]. Three main groups of techniques with significant performance and complexity differences are analyzed: classical nonlinear extensions of the Kalman filter like the Extended Kalman Filter (EKF), unscented filtering with the generalized Unscented Kalman Filter (UKF), and particle filtering like the particle filter with resampling (PFr) and the Ensemble Kalman Filter (EnKF). The parameters are directly estimated using the parameter state-augmented approach and the discrete version of the estimators is implemented.

A simple CHP model structure of a small-scale spark ignition (SI) ICE based cogeneration system is taken [5]. The main subsystems are: the SI ICE engine, the generator, and the outer loops heat recovery circuits.

The paper is organized as follows: first a description of the CHP model including its main assumptions is given. Then, the recursive nonlinear state/parameter estimators algorithms and their tuning parameters are described. In section 4, the implementation details and the estimation results are presented. Finally, the conclusions with a general comparison of the nonlinear estimators for this problem are given.

2. Model description

In [5,6] a small-scale SI ICE based cogeneration model with a simple dynamic engine model is presented. The CHP main subsystems are the reciprocating spark ignition (SI) internal combustion engine (ICE), the generator, and the outer loop heat recovery circuits.

Table 1. MVEM and generator models

Throttle Valve:

$$A_{\mathrm{th}} = \frac{\pi d_{\mathrm{th}}^2}{4}\left(1 - \frac{\cos\left(\alpha_{\mathrm{th},0}+\left(\frac{\pi}{2}-\alpha_{\mathrm{th},0}\right)u_{\mathrm{th}}\right)}{\cos\alpha_{\mathrm{th},0}}\right) + A_{\mathrm{th},0}$$

$$\text{if } \frac{p_{\mathrm{ab}}}{p_{\mathrm{mn}}} < \frac{1}{2} \quad \dot{m}_{\mathrm{th}} = C_{\mathrm{th}} A_{\mathrm{th}} \frac{p_{\mathrm{ab}}}{\sqrt{R_\beta \vartheta_{\mathrm{ab}}}} \frac{1}{\sqrt{2}}$$

$$\text{else } \ \dot{m}_{\mathrm{th}} = C_{\mathrm{th}} A_{\mathrm{th}} \frac{p_{\mathrm{ab}}}{\sqrt{R_\beta \vartheta_{\mathrm{mn}}}} [2\frac{p_{\mathrm{mn}}}{p_{\mathrm{ab}}}(1 - \frac{p_{\mathrm{mn}}}{p_{\mathrm{ab}}})]^{\frac{1}{2}}$$

Intake Manifold:

$$\vartheta_{\mathrm{mn}} = \vartheta_{\mathrm{ab}} + \Delta\vartheta_{\mathrm{mn}}$$

$$\frac{d}{dt} p_{\mathrm{mn}} = \frac{R_\beta \vartheta_{\mathrm{mn}}}{V_{\mathrm{mn}}} \left[\dot{m}_{\mathrm{th}} - \dot{m}_\beta\right]$$

Cylinders air mass flows:

$$\dot{m}_{\mathrm{ex}} = \dot{m}_\beta + \dot{m}_\varphi$$

$$\dot{m}_{\mathrm{ex}} = \frac{p_{\mathrm{mn}}}{R_\beta \vartheta_{\mathrm{mn}}} \frac{V_d}{4\pi} \eta_{\mathrm{vl}} \omega_{\mathrm{en}}$$

$$\dot{m}_\beta = \lambda \sigma_0 \dot{m}_\varphi$$

$$\eta_{\mathrm{vl}} = \eta_{\mathrm{vl},1}(\omega_{\mathrm{en}}) \eta_{\mathrm{vl},2}(p_{\mathrm{mn}})$$

$$\eta_{\mathrm{vl},1}(\omega_{\mathrm{en}}) = \left(\gamma_0 + \gamma_1 \omega_{\mathrm{en}} + \gamma_2 \omega_{\mathrm{en}}^2\right)$$

$$\eta_{\mathrm{vl},2}(p_{\mathrm{mn}}) = \frac{V_c + V_d}{V_d} - \frac{V_c}{V_d}\left(\frac{p_{\mathrm{ex}}}{p_{\mathrm{mn}}}\right)^{\frac{1}{\kappa_\beta}}$$

Cylinders torque generation:

$$T_{\mathrm{en}} = p_{\mathrm{meb}} \frac{V_d}{4\pi}$$

$$p_{\mathrm{meb}} = \eta_{\mathrm{td}}\left(\omega_{\mathrm{en}}\right) p_{\mathrm{me}\,\varphi} - p_{\mathrm{mef}}\left(\omega_{\mathrm{en}}\right) - p_{\mathrm{mepg}}$$

$$\eta_{\mathrm{td}} p_{\mathrm{me}\,\varphi} = \left(\eta_0 + \eta_1 \omega_{\mathrm{en}} + \eta_2 \omega_{\mathrm{en}}^2\right) H_l \frac{\dot{m}_\varphi 4\pi}{\omega_{\mathrm{en}} V_d}$$

$$p_{\mathrm{mef}} = \left(\beta_0 + \beta_1 \omega_{\mathrm{en}} + \beta_2 \omega_{\mathrm{en}}^2\right) \frac{4\pi}{V_d}$$

$$p_{\mathrm{mepg}} = p_{\mathrm{ex}} - p_{\mathrm{mn}}$$

Cylinders energy balance:

$$\dot{H}_{\mathrm{ex}} = \dot{H}_\beta + \dot{H}_{\Delta c} - \dot{W}_{\mathrm{en}} - \dot{W}_{\mathrm{pg}} - \dot{Q}_f - \dot{Q}_{\mathrm{cg,cw}}$$

$$\dot{Q}_{\mathrm{cg,cw}} = (\delta_0 + \delta_1 p_{\mathrm{meb}} + \delta_2 p_{\mathrm{meb}}^2 + \delta_3 p_{\mathrm{meb}}^3) \frac{\omega_{\mathrm{en}}}{4\pi}$$

$$\dot{W}_{\mathrm{en}} = \omega_{\mathrm{en}} T_{\mathrm{en}}$$

$$\dot{W}_{\mathrm{pg}} = \omega_{\mathrm{en}} p_{\mathrm{mepg}} \frac{V_d}{4\pi}$$

$$\dot{H}_{\Delta c} = H_l \frac{\dot{m}_\varphi 4\pi}{\omega_{\mathrm{en}} V_d}$$

Cylinders angular velocity:

$$\frac{d}{dt}\omega_{\mathrm{en}} = \frac{1}{I_{\mathrm{en}}}\left[T_{\mathrm{en}} - T_{\mathrm{ld}}\right]$$

Generator:

$$T_{\mathrm{ld}} = \frac{P_{\mathrm{ec}}}{\omega_{\mathrm{en}} \eta_{\mathrm{ec}}} P_{\mathrm{ec}}$$

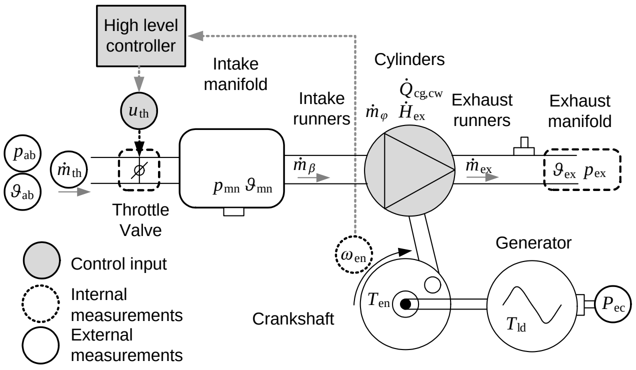


Figure 1. MVEM and electrical generator

The SI ICE is modeled as a lumped parameter mean-value engine model [7–9] and the generator is considered as a simple static model (see Fig. 1). It is assumed that the engine is controlled to operate at its stochiometric ratio with a fixed ignition angle equal to the value of the engine maps. The volumetric and thermodynamic efficiencies are approximated using second order polynomials of the angular velocity and the intake manifold pressure (for details see [10]). The heat flow rate transferred from the cylinder gases into the cylinder walls, $\dot{Q}_{\mathrm{cg,cw}}$, is also approximated using a second order polynomial of the current mean effective pressure (i.e. load) and the angular velocity.

The heat recovery system (see Fig. 2) connects the engine inner thermal model with a secondary circuit (e.g. a buffer tank). The engine cooling and the exhaust gases heat recovery circuits are coupled with the secondary system using two simple heat exchanger models. The pipes connecting the different components are assumed adiabatic and the pumps are operated at constant mass flow rates, that is, $du_{\mathrm{p1}}/dt = 0$ and $du_{\mathrm{p2}}/dt = 0$.

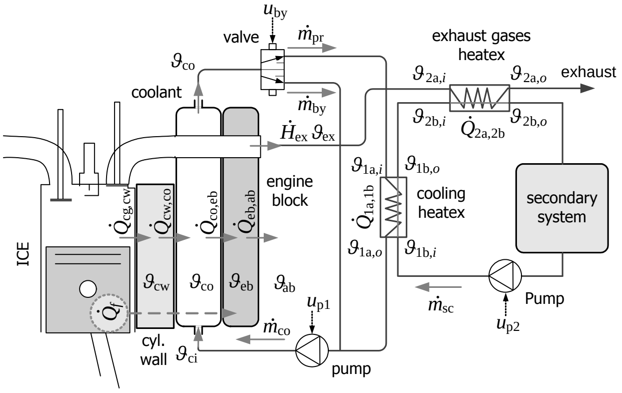

u_{by}
valve
$\dot{m}_{\mathrm{pr}}$
exhaust gases heatex
ϑ_{co}
$\vartheta_{2\mathrm{a},i}$
$\vartheta_{2\mathrm{a},o}$
exhaust
coolant
$\dot{m}_{\mathrm{by}}$
$\vartheta_{2\mathrm{b},i}$
$\dot{Q}_{2\mathrm{a},2\mathrm{b}}$
$\vartheta_{2\mathrm{b},o}$
$\dot{H}_{\mathrm{ex}}$ ϑ_{ex}
$\vartheta_{1\mathrm{a},i}$
$\vartheta_{1\mathrm{b},o}$
engine block
$\dot{Q}_{1\mathrm{a},1\mathrm{b}}$
cooling heatex
secondary system
$\dot{Q}_{\mathrm{cg,cw}}$
$\dot{Q}_{\mathrm{cw,co}}$
$\dot{Q}_{\mathrm{co,eb}}$
$\dot{Q}_{\mathrm{eb,ab}}$
$\vartheta_{1\mathrm{a},o}$
$\vartheta_{1\mathrm{b},i}$
ICE
ϑ_{cw}
ϑ_{co}
ϑ_{eb}
ϑ_{ab}
u_{p1}
$\dot{m}_{\mathrm{sc}}$
Pump
$\dot{Q}_f$
$\dot{m}_{\mathrm{co}}$
u_{p2}
cyl. wall
ϑ_{ci}
pump

Figure 2. HRC and engine inner thermal model

A parameterized models of the CHP unit can be seen in table 1 for the MVEM and the electrical generator, and in table 2 for the heat recovery circuit and the engine inner thermal model. The model notation is summarized in table 8.

3. Nonlinear estimation

3.1 Model equations

The CHP model can be written in the general discrete nonlinear state space form:

$$\begin{aligned} x_k &= f_{k-1}\left(x_{k-1}, u_{k-1}, w_{k-1}\right) \\ y_k &= h_k\left(x_k, v_k\right) \end{aligned} \tag{1}$$

where $f_{k-1} : \mathbb{R}^{n_x+n_u+n_w} \mapsto \mathbb{R}^{n_x \times 1}$ is the discrete state function, $x_k \in \mathbb{R}^{n_x \times 1}$ is the discrete state vector,

Table 2. HRC models

Inner Engine Thermal Model:
$\frac{d}{dt}\vartheta_{\mathrm{cw}} = \frac{1}{c_{\mathrm{cw}} m_{\mathrm{cw}}}(\dot{Q}_{\mathrm{cg,cw}} - \dot{Q}_{\mathrm{cw,co}})$
$\frac{d}{dt}\vartheta_{\mathrm{co}} = \frac{1}{c_{\mathrm{co}} m_{\mathrm{co}}}[c_{\mathrm{co}}\dot{m}_{\mathrm{co}}(\vartheta_{\mathrm{ci}} - \vartheta_{\mathrm{co}}) + \dot{Q}_{\mathrm{cw,co}} - \dot{Q}_{\mathrm{co,eb}}]$
$\frac{d}{dt}\vartheta_{\mathrm{eb}} = \frac{1}{c_{\mathrm{eb}} m_{\mathrm{eb}} + c_{\mathrm{eo}} m_{\mathrm{eo}}}(\dot{Q}_f + \dot{Q}_{\mathrm{co,eb}} - \dot{Q}_{\mathrm{eb,ab}})$
$\dot{Q}_{\mathrm{cw,co}} = \alpha_{\mathrm{cw,co}} A_{\mathrm{cw,co}}[\vartheta_{\mathrm{cw}} - \frac{1}{2}(\vartheta_{\mathrm{ci}} + \vartheta_{\mathrm{co}})]$
$\dot{Q}_{\mathrm{co,eb}} = \alpha_{\mathrm{co,eb}} A_{\mathrm{co,eb}}[\frac{1}{2}(\vartheta_{\mathrm{ci}} + \vartheta_{\mathrm{co}}) - \vartheta_{\mathrm{eb}}]$
$\dot{Q}_{\mathrm{eb,ab}} = \alpha_{\mathrm{eb,ab}} A_{\mathrm{eb,ab}}(\vartheta_{\mathrm{eb}} - \vartheta_{\mathrm{ab}})$
$\dot{Q}_f = \frac{1}{4\pi} p_{\mathrm{mef}} V_d \omega_{\mathrm{en}}$
Cooling circuit:
$\dot{m}_{\mathrm{co}} = \dot{m}_{\mathrm{by}} + \dot{m}_{\mathrm{pr}}$
$\dot{m}_{\mathrm{by}} = \varpi_0 u_{\mathrm{by}}$
$\dot{m}_{\mathrm{co}} = \xi_{\mathrm{p1,0}} + \xi_{\mathrm{p1,1}} u_{\mathrm{p1}} + \xi_{\mathrm{p1,2}} u_{\mathrm{p1}}^2$
$\frac{d}{dt}\vartheta_{\mathrm{1a}} = \frac{1}{c_{\mathrm{1a}} m_{\mathrm{1a}}}[c_{\mathrm{1a}}\dot{m}_{\mathrm{1a}}(\vartheta_{\mathrm{1a},i} - \vartheta_{\mathrm{1a},o}) - \dot{Q}_{\mathrm{1a,1b}}]$
$\frac{d}{dt}\vartheta_{\mathrm{1b}} = \frac{1}{c_{\mathrm{1b}} m_{\mathrm{1b}}}[c_{\mathrm{1b}}\dot{m}_{\mathrm{1b}}(\vartheta_{\mathrm{1b},i} - \vartheta_{\mathrm{1b},o}) + \dot{Q}_{\mathrm{1a,1b}}]$
$\dot{Q}_{\mathrm{1a,1b}} = \frac{1}{2}\alpha_{\mathrm{1a,1b}} A_{\mathrm{1a,1b}}[(\vartheta_{\mathrm{1a},i} + \vartheta_{\mathrm{1a},o}) - (\vartheta_{\mathrm{1b},i} + \vartheta_{\mathrm{1b},o})]$
Exhaust circuit:
$\frac{d}{dt}\vartheta_{\mathrm{2a}} = \frac{1}{c_{\mathrm{2a}} m_{\mathrm{2a}}}[c_{\mathrm{2a}}\dot{m}_{\mathrm{2a}}(\vartheta_{\mathrm{2a},i} - \vartheta_{\mathrm{2a},o}) - \dot{Q}_{\mathrm{2a,2b}}]$
$\frac{d}{dt}\vartheta_{\mathrm{2b}} = \frac{1}{c_{\mathrm{2b}} m_{\mathrm{2b}}}[c_{\mathrm{2b}}\dot{m}_{\mathrm{2b}}(\vartheta_{\mathrm{2b},i} - \vartheta_{\mathrm{2b},o}) + \dot{Q}_{\mathrm{2a,2b}}]$
$\dot{Q}_{\mathrm{2a,2b}} = \frac{1}{2}\alpha_{\mathrm{2a,2\,b}} A_{\mathrm{2a,2b}}[(\vartheta_{\mathrm{2a},i} + \vartheta_{\mathrm{2a},o}) - (\vartheta_{\mathrm{2b},i} + \vartheta_{\mathrm{2b},o})]$
Secondary Circuit:
$\dot{m}_{\mathrm{sc}} = \xi_{\mathrm{p2,0}} + \xi_{\mathrm{p2,1}} u_{\mathrm{p2}} + \xi_{\mathrm{p2,2}} u_{\mathrm{p2}}^2$

$u_{k-1} \in \mathbb{R}^{n_u \times 1}$ is the discrete input, $w_{k-1} \in \mathbb{R}^{n_w \times 1}$ is the discrete process noise vector, $h_k : \mathbb{R}^{n_x + n_v} \mapsto \mathbb{R}^{n_x \times 1}$ is the discrete output function, $v_k \in \mathbb{R}^{n_v \times 1}$ is the discrete measurement noise vector, $y_k \in \mathbb{R}^{n_y \times 1}$ is the output vector, and k is the time index. The noise vector sequences $\{w_{k-1}\}$ and $\{v_k\}$ are assumed Gaussian, white, zero-mean, uncorrelated, and have the known covariance matrices $Q_k \in \mathbb{R}^{n_x \times n_w}$ and $R_k \in \mathbb{R}^{n_y \times n_v}$.

3.2 Augmented states

The augmented state space approach can be directly used to simultaneously solve the state and the parameter estimation problem (e.g. see [11]). An augmented state space representation is formulated by adding the vector of parameters to be estimated $\theta_k \in \mathbb{R}^{n_\theta \times 1}$ as new states:

$$\begin{aligned} \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} &= \begin{bmatrix} f_{k-1}\left(x_{k-1}, u_{k-1}, w_{k-1}^{(x)}\right) \\ \theta_{k-1} + T_s w_{k-1}^{(\theta)} \end{bmatrix} & (2) \\ y_k &= h_k\left(x_k, v_k\right) & (3) \end{aligned}$$

where T_s is the sampling time step, $w_{k-1}^{(x)} \in \mathbb{R}^{n_w^{(x)} \times 1}$ is the process noise vector that affects the original states, and $w_{k-1}^{(\theta)} \in \mathbb{R}^{n_w^{(\theta)} \times 1}$ is the process noise vector that affects the added parameter states. The noise vector sequences $\{w_{k-1}\}$ and $\{v_k\}$ are assumed Gaussian, white, zero-mean, uncorrelated, and have the known covariance matrices $Q_k = \mathrm{blkdg}(Q_k^{(x)}, Q_k^{(\theta)}) \in \mathbb{R}^{(n_x + n_\theta) \times (n_w^{(x)} + n_w^{(\theta)})}$ and $R_k \in \mathbb{R}^{n_y \times n_v}$.

$$\begin{aligned} w_k &\sim \mathcal{N}(0, \mathrm{blkdg}(Q_k^{(x)}, Q_k^{(\theta)})) \\ v_k &\sim \mathcal{N}(0, R_k) \end{aligned}$$

During the propagation step the augmented states corresponding to parameters θ_k are considered equal to the previous time step θ_{k-1} with some additive process noise $w_{k-1}^{(\theta)}$. If it is assumed that the parameters do not change at all, then there is no process noise vector $w_{k-1}^{(\theta)}$, but for the more general case of time-varying parameters (or where component degradation may happen), the value of $Q_k^{(\theta)}$ will be given by the admissible range of variation of θ_k. During the measurement update step the parameter values are corrected.

For notation simplicity, in the estimators algorithms that follow, the augmented state vector is referred as x_k, the state augmented function (2) is referred as f_{k-1}, and the augmented process noise vector is referred as w_{k-1}.

3.3 Nonlinear Recursive Estimators

The nonlinear estimation problem can be formulated as a recursive Bayesian estimation problem with a propagation and a measurement update steps. This is the optimal way of predicting a state probability density function (pdf) $p(x_k)$ for any system in state space representation with process and measurement noise[1].

Assuming that the initial state pdf $p(x_0)$, the process noise pdf $p(w_{k-1})$, and the measurement noise pdf $p(v_k)$ are known, a recursive solution of the estimation problem can be found using first the Chapman-Kolmogorov equation to calculate the a priori pdf for the state x_k based on the last measurement y_{k-1} (propagation step)

$$p(x_k|y_{k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|y_{k-1}) dx_{k-1} \qquad (4)$$

where $p(x_k|x_{k-1})$ can be calculated from the state function f_{k-1} and the pdf of the process noise w_k.

And then, the Bayes rule to update the pdf of the state x_k with the new measurement y_k (measurement update)

$$p(x_k|y_k) = \frac{p(y_k|x_k)\, p(x_k|y_{k-1})}{\int p(y_k|x_k)\, p(x_k|y_{k-1})\, dx_k} \qquad (5)$$

where $p(y_k|x_k)$ is available from our knowledge of the output function h_k and the pdf of v_k, $p(x_k|y_{k-1})$ is known from (4). Although the initial state pdf

[1] Markov process of order one

$p(x_0)$, the process noise pdf $p(w_{k-1})$, and the measurement noise pdf $p(v_k)$ are needed to solve the recursive Bayesian estimation, no specific statistical distribution is required.

The recursive relations (4) and (5) used to calculate the posterior pdf $p(x_k|y_k)$ are a conceptual solution and only for very specific cases can be solved analytically. In general, approximations are required for practical problems. Three main groups of suboptimal techniques with significant performance and computational cost differences are used to approximate the recursive Bayesian estimation problem: the classical nonlinear extensions of the Kalman filter, the unscented filtering techniques, and the particle filtering approaches.

3.3.1 Extended Kalman Filters

Table 3. EKF algorithm

```
Initialization:
$\hat{x}_{0|0} = E(x_0)$
$P_{0|0} = E[(x_0 - \hat{x}_{0|0})(x_0 - \hat{x}_{0|0}^T)]$
for $k = 1, 2, \ldots$
Propagation step:
(a priori covariance estimate)
$F_{k-1} = \left.\frac{\partial f_{k-1}}{\partial x_{k-1}}\right|_{\hat{x}_{k-1|k-1}}$   $L_{k-1} = \left.\frac{\partial f_{k-1}}{\partial w_{k-1}}\right|_{\hat{x}_{k-1|k-1}}$
$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T$
(a priori state-output estimate)
$\hat{x}_{k|k-1} = f_{k-1}(\hat{x}_{k-1|k-1}, u_{k-1}, 0)$
$\hat{y}_{k|k-1} = h_k(\hat{x}_{k|k-1}, 0)$
Measurement update:
(Kalman gain calculation)
$H_k = \left.\frac{\partial h_k}{\partial u_k}\right|_{\hat{x}_{k|k-1}}$   $M_k = \left.\frac{\partial h_k}{\partial v_k}\right|_{\hat{x}_{k|k-1}}$
$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + M_k R_k M_k^T)^{-1}$
(a posteriori state-covariance estimate)
$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1})$
$P_{k|k} = (I - K_k H_k) P_{k|k-1}$
```

The discrete EKF is probably the most used sequential nonlinear estimator nowadays. It was originally developed as a nonlinear extension by Schmidt [12] upon the seminal work of Kalman [3] Based on the Kalman filter, it assumes that the statistical distribution of the state vector remains Gaussian after every time step[2] so it is only necessary to propagate and update the mean and covariance of the state random variable x_k. The main concept is that the estimated state (i.e. estimated mean of x_k) is sufficiently close to the true state (i.e. true mean of x_k) so the nonlinear state/output model equations can be linearized by a truncated first-order Taylor series expansion around the last estimated state.

The discrete algorithm can be seen in table 3. In general, it works for many practical problems but no general convergence or stability conditions can be established[3] and its final performance will depend on the specific case study. For highly nonlinear models with unknown initial conditions the EKF assumptions may prove to be poor and the filter may fail or have a poor performance. The main tunning parameters are the estimator covariance matrices Q_k and R_k.

3.3.2 Unscented Kalman Filters

Table 4. UKF algorithm

```
Initialization:
$L = n_x + n_w + n_v$, $\lambda = \alpha^2(L + \kappa) - L$
$\gamma = \sqrt[2]{L + \lambda}$, $\Lambda_m^0 = \lambda/(\lambda + L)$
$\Lambda_c^0 = \lambda/(\lambda + L) + (1 - \alpha^2 + \beta)$
  for $i = 1, 2, \ldots, 2L$
  $\Lambda_m^i = (2(\lambda + L))^{-1}$, $\Lambda_c^i = \Lambda_m^i$
$\hat{x}_{0|0} = E(x_0)$
$P_{0|0} = E[(x_0 - \hat{x}_{0|0})(x_0 - \hat{x}_{0|0})]$
for $k = 1, 2, \ldots$
Propagation step:
(sigma points propagation)
$\tilde{P}_{k-1|k-1} = \mathrm{blkdiag}(P_{k-1|k-1}, Q_k, R_k)$
$\tilde{x}_{k-1|k-1}^0 = [(\hat{x}_{k-1|k-1})^T, 0_{1\times n_W}, 0_{1\times n_v}]^T$
  for $i = 1, 2, \ldots, L$
  $\tilde{x}_{k-1|k-1}^i = \tilde{x}_{k-1|k-1}^0 + \gamma\,\mathrm{chol}(\tilde{P}_{k-1|k-1}, i)$
  $\tilde{x}_{k-1|k-1}^{i+L} = \tilde{x}_{k-1|k-1}^0 - \gamma\,\mathrm{chol}(\tilde{P}_{k-1|k-1}, i + L)$
  $\tilde{x}_{k|k-1}^{(x)i} = f_{k-1}(\tilde{x}_{k-1|k-1}^{(x)i}, u_{k-1}, \tilde{x}_{k-1|k-1}^{(w)i})$
  $\tilde{y}_{k|k-1}^i = h_k(\tilde{x}_{k|k-1}^{(x)i}, \tilde{x}_{k-1|k-1}^{(v)i})$
(a priori state-output estimate)
$\hat{x}_{k|k-1} = \sum_{i=0}^{2L} \Lambda_m^i \tilde{x}_{k|k-1}^{(x)i}$
$\hat{y}_{k|k-1} = \sum_{i=0}^{2L} \Lambda_m^i \tilde{y}_{k|k-1}^i$
(a priori state covariance estimate)
$\tilde{e}_{x,k|k-1}^i = (\tilde{x}_{k|k-1}^{(x)i} - \hat{x}_{k|k-1})$
$P_{k|k-1} = \sum_{i=0}^{2L} \Lambda_c^i (\tilde{e}_{x,k|k-1}^i)(\tilde{e}_{x,k|k-1}^i)^T$
Measurement update:
(Kalman gain calculation)
$\tilde{e}_{y,k|k-1}^i = (\tilde{y}_{k|k-1}^i - \hat{y}_{k|k-1})$
$P_y = \sum_{i=0}^{2L} \Lambda_c^i (\tilde{e}_{y,k|k-1}^i)(\tilde{e}_{y,k|k-1}^i)^T$
$P_{xy} = \sum_{i=0}^{2L} \Lambda_c^i (\tilde{e}_{x,k|k-1}^i)(\tilde{e}_{y,k|k-1}^i)^T$
$K_k = P_{xy} P_y^{-1}$
(a posteriori state-covariance estimate)
$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1})$
$P_{k|k} = P_{k|k-1} - K_k P_y K_k^T$
```

The unscented Kalman filter was originally developed

[2] this assumption is in general not true for nonlinear systems

[3] except for some special cases [13]

by Julier and Uhlman [14–17].

In the unscented Kalman filters instead of approximating the nonlinear state/output functions, it is the probability distribution what is approximated. Basically, a set of points, called sigma points, are generated to match the state mean and state covariance of the probability distribution of the last estimated state, then they are propagated through the nonlinear function. The projected points are used to approximate the first two moments (i.e. the a priori estimated state and state covariance) that are necessary during the measurement update step. This filter normally outperforms the previously presented EKF. Its more general form has a higher computational cost but it does not require the calculation of any Jacobian matrices (i.e. derivatives). The algorithm can be seen in table 4.

The tunning parameters of the UKF are also the estimator process and measurement noise covariance matrices and the scalar parameters $\{\alpha, \kappa, \beta\}$: α determines the spread of the sigma points around the last estimate, and β value depends on the type of distribution assumed (for more details about their values see [17])

3.3.3 Particle Filters

The particle filter was first proposed by Metropolis and Wiener in the 40's but until the the 80's its implementation was not practical [13]. The particle filter is a statistical, brute-force approach to estimation that works by approximating the recursive Bayesian estimation problem. Its fundamental idea is to represent a pdf by a set of samples called particles.

During the propagation step, the particles that approximates $p\left(x_{k-1}|y_{k-1}\right)$ from the last iteration are projected through the known dynamic model to obtain an approximation of $p\left(x_k|y_{k-1}\right)$. Then during the measurement update, each particle likelihood q_i is calculated using the measurement y_k, the known output function and the measurement noise known pdf to approximate the likelihood $p\left(y_k|x_k\right)$. If the measurement noise pdf $p\left(v_k\right)$ is normal, then the likelihood function for each particle is proportional to

$$\begin{aligned} q_i &= p\left(y_k|x^i_{k|k-1}\right) && (6)\\ e^i_k &= y_k - h_k\left(x^i_{k|k-1}\right) \\ &\propto \frac{1}{2\pi^{m/2}\left|R\right|^{1/2}}\exp(-0.5(e^i_k)R^{-1}(e^i_k)^T) && (7) \end{aligned}$$

Then this expression is normalized to assure that the sum of all likelihoods is equal to one

$$\tilde{q}_i = \frac{q_i}{\sum_{j=1}^{N} p\left(y_k|x^j_{k|k-1}\right)} \quad (8)$$

Table 5. PF algorithm

```
Initialization:
(initial particles set)
for $i = 1, 2, \dots, N$
  $x^i_{0|0} \sim \mathcal{N}(x_0, P_0)$
for $k = 1, 2, \dots$
Propagation step:
(particles state-output propagation)
for $i = 1, 2, \dots, N$
  $x^i_{k|k-1} = f_{k-1}(x^i_{k-1|k-1}, u_{k-1}, w^i_{k-1})$
  $y^i_{k|k-1} = h_k(x^i_{k|k-1}, v^i_{k-1})$
Measurement update:
(relative likelihood for each particle)
  $q_i = p(y_k|x^i_{k|k-1})$
  $\tilde{q}_i = q_i / \sum_{i=1}^{N} q_i$
(resampling with replacement step)
$\Pr\left(x^j_{k|k} = x^i_{k|k}\right) = \tilde{q}^i$
(a posteriori state-covariance estimates)
$\hat{x}_{k|k} = (N)^{-1}\sum_{i=1}^{N}\tilde{q}_i x^i_{k|k}$
$P_{k|k} = \sum_{i=1}^{N}\tilde{q}_i(x^i_{k|k} - \hat{x}_{k|k})(x^i_{k|k} - \hat{x}_{k|k})^T$
```

Using the resampling with replacement step a new particle set is generated from the old one likelihood weights. The new particle set approximates the $p\left(x_k|y_k\right)$. From this pdf the two first central moments are computed [4]. The new particle set is used in the next iteration. For details about the algorithm see table 5.

It is clear that as the number of particles goes to infinity, the quality of the approximations is increased and the approximated pdf approaches the real distribution $p\left(x_k|y_k\right)$. The particle filter has several advantages: it can be applied to any nonlinear non-gaussian estimation problem, it approximates the whole pdf and not only a few estimators, it easy to implement and it normally works. Its main disadvantage is clearly its high computational cost. For more details see [18,19].

The EnKF can be considered also a particle filter. It uses an ensemble (i.e. particle set) during the propagation step but the classical Kalman measurement update equations (instead of using the resampling with replacement approach) during the measurement update step. The covariances matrices P_{xy} and P_y obtained from the propagation of the ensemble elements through the nonlinear state-space are used to calculate the Kalman gain K_k. The a posteriori ensemble is calculated from the Kalman gain matrix and an artificially generated measurement particle set that is normally distributed with mean equal to the last measurement y_k and covariance equal to R_k. The a posteriori ensemble is used to calculate the a posteriori state

[4] to be consistent with the other implemented filters.

Table 6. EnKF algorithm

```
Initialization:
(initial ensemble)
for $i = 1, 2, \dots, N$
    $x^i_{0|0} \sim \mathcal{N}(x_0, P_0)$
for $k = 1, 2, \dots$
Propagation step:
(ensemble propagation)
for $i = 1, 2, \dots, N$
    $x^i_{k|k-1} = f_{k-1}(x^i_{k-1|k-1}, u_{k-1}, w^i_{k-1})$
    $y^i_{k|k-1} = h_k(x^i_{k|k-1}, v^i_{k-1})$
(estimated state-output propagation)
$\hat{x}_{k|k-1} = (N)^{-1} \sum_{i=1}^{N} x^i_{k|k-1}$
$\hat{y}_{k|k-1} = (N)^{-1} \sum_{i=1}^{N} y^i_{k|k-1}$
(covariance calculation)
$e^i_{x,k|k-1} = (x^i_{k|k-1} - \hat{x}_{k|k-1})$
$P_{k|k-1} = (N-1)^{-1} \sum_{i=1}^{N} (e^i_{x,k|k-1})(e^i_{x,k|k-1})^T$
Measurement update:
(Kalman gain calculation)
$e^i_{y,k|k-1} = (y^i_{k|k-1} - \hat{y}_{k|k-1})$
$P_y = (N-1)^{-1} \sum_{i=0}^{N} (e^i_{y,k|k-1})(e^i_{y,k|k-1})^T$
$P_{xy} = (N-1)^{-1} \sum_{i=0}^{N} (e^i_{x,k|k-1})(e^i_{y,k|k-1})^T$
$K_k = P_{xy} {P_y}^{-1}$
(state-out-covariance update)
$x^i_{k|k} = x^i_{k|k-1} + K_k((y_k + v^i_k) - y^i_{k|k-1})$
$\hat{x}_{k|k} = (N)^{-1} \sum_{i=1}^{N} x^i_{k|k}$
$P_{k|k} = P_{k|k-1} - K_k P_y K_k^T$
```

and covariance estimate, and it is used for the next filter iteration of the algorithm. For details about the algorithm see table 6. The EnKF was originally developed by [20] to overcome the curse of dimensionality in large scale problems (i.e. weather data assimilation). It is suggested in the literature [21] that ensembles (i.e. particle sets) of 50 to 100 are often adequate for systems with thousands of states but no conclusive work has been done on this direction.

Beside the estimator process and measuremnet noise covariance matrices the other tunning parameter for this two filters is the number of particles/ensemble elements.

4. Results

The CHP model is written in Modelica and compiled in Dymola into a stand-alone executable file called Dymosim. The different estimators are implemented in Matlab from where Dymosim is sequentially called during the propagation step to project the state vector in the estimators algorithms. The parameter state vector θ_k is directly propagated within the Matlab code so the original CHP model does not need to be modified to include the parameter dynamic equations.

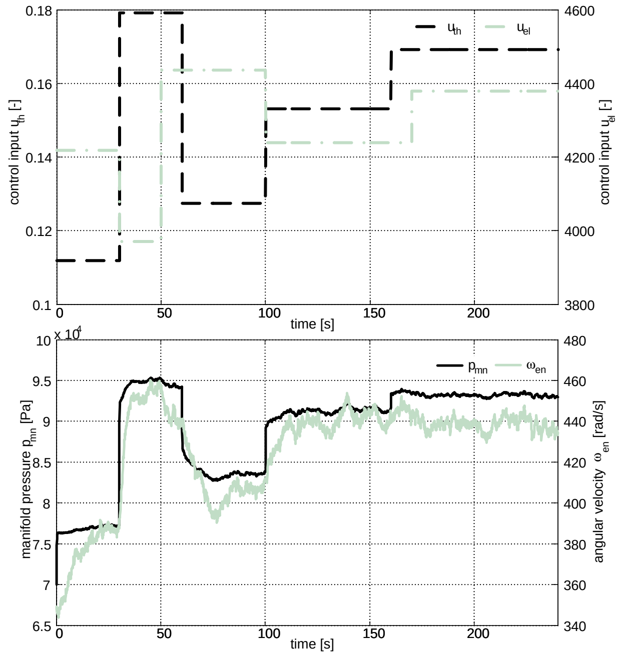


Figure 3. MVEM throttle and electrical load inputs (above) and measured outputs (below).

Because the HRC does not affect the performance of the MVEM model wrt. the mechanical work generated[5] the estimation problem can be divided into two estimation problems for the MVEM and the HRC subsystems. For the two cases it is assumed that the inputs of the system are perfectly known. Due to the lack of experimental measurements, simulated data sets are generated using normally distributed additive white process and measurement noise.

For the MVEM model the parameters to be estimated are $\{V_{\mathrm{mn}}, I_{\mathrm{en}}\}$. The measured outputs are the manifold pressure p_{mn} and the angular velocity ω_{en}. The MVEM is operated in open loop, the inputs are the throttle control input u_{th} and the electrical load control input u_{el} (see Fig. 3). Other model parameters are assumed to be known.

The estimators are simulated for 240 s with a sampling time of 0.1 s. Because the transient response is relevant to identify these parameters, step-like input sequences with high frequency content are used. The same estimator initial state, measurement and process noise covariance matrices are used for all the filters.

The discrete Jacobian matrix is approximated from the jacobian given by Dymosim during its linearization. The UKF parameters are $\{\alpha, \kappa, \beta\} = \{1\,\mathrm{E}-2, 0, 2\}$, the EnKF is simulated for $N = \{30, 50, 100\}$ and the PF for $N = \{100, 200\}$. The estimated parameters

[5] This is true for the simplified model presented.

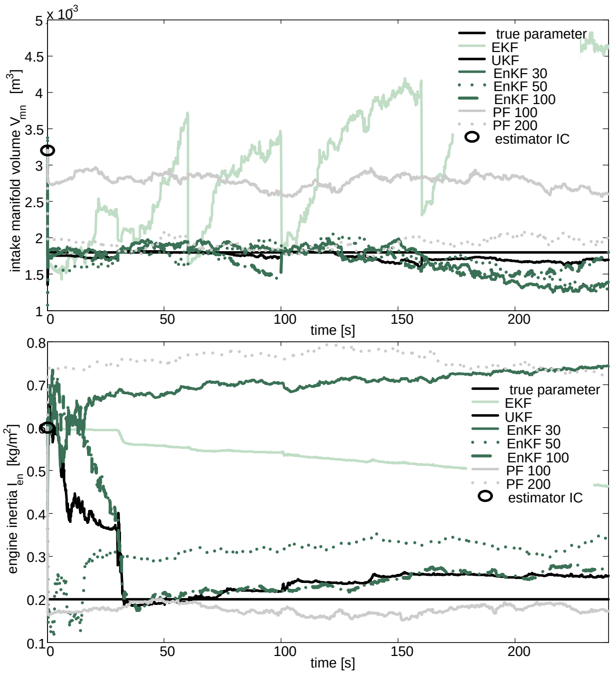


Figure 4. Estimated intake manifold volume V_{mn} (above) and engine inertia I_{en} (below)

$$x_0 = [p_{\mathrm{mn}}, \omega_{\mathrm{en}}, V_{\mathrm{mn}}, I_{\mathrm{en}}]^T = [7\,\mathrm{E}\,4, 350, 1.8\,\mathrm{E}\,{-2}, 0.2]^T$$
$$x_{0|0} = [7.9\,\mathrm{E}\,4, 470, 3.2\,\mathrm{E}\,{-2}, 0.6]^T$$
$$Q_k = \mathrm{diagblk}\,[\mathrm{diag}([3\,\mathrm{E}\,3{,}1\,\mathrm{E}\,2]), \mathrm{diag}([1\,\mathrm{E}\,{-8}{,}1\,\mathrm{E}\,{-4}])]$$
$$R_k = \mathrm{diag}([6\,\mathrm{E}\,3{,}2\,\mathrm{E}\,2])$$

for each filter are shown in Fig. 4. The EKF has a poor performance with a very noisy parameter estimation. A zig-zag behavior is seen for the V_{mn} parameter when the inputs change abruptly, this is a clear consequence and limitation of its derivative based formulation.The UKF performs very well staying around the value of the true parameters. The EnKFs as expected, increase their accuracy as the number of elements in the ensemble is increased. The PF has suprisingly a poor performance, an increased number of particles improves the estimation of one parameter but degrades the estimation of the other one. Probably due to the sampling empoverishment the filter stays trapped and an increased number of particles do little to remedy this situation, a roughening algorithm should probably solve this problem.

For the HRC models the parameters to be estimated are $\{\alpha_{\mathrm{cw,co}}, G_1 = \alpha_{\mathrm{1a,1b}} A_{\mathrm{1a,1b}}, G_2 = \alpha_{\mathrm{2a,2b}} A_{\mathrm{2a,2b}}\}$. Other HRC model parameters are assumed known. The measured outputs are $\{\vartheta_{\mathrm{co}}, \vartheta_{\mathrm{2a}}, \vartheta_{\mathrm{2b}}, \vartheta_{\mathrm{1b}}\}$. The HRC is operated in closed loop with the thermal control loop regulating the coolant ouput temperature at 95 °C. The MVEM model is simulated operating under closed loop wrt. angular velocity with the MVEM parameters are assumed known. The control input is the electrical load u_{el}. Fouling and a sudden change in G_1 is artificially generated to see the tracking capabilities of the filters.

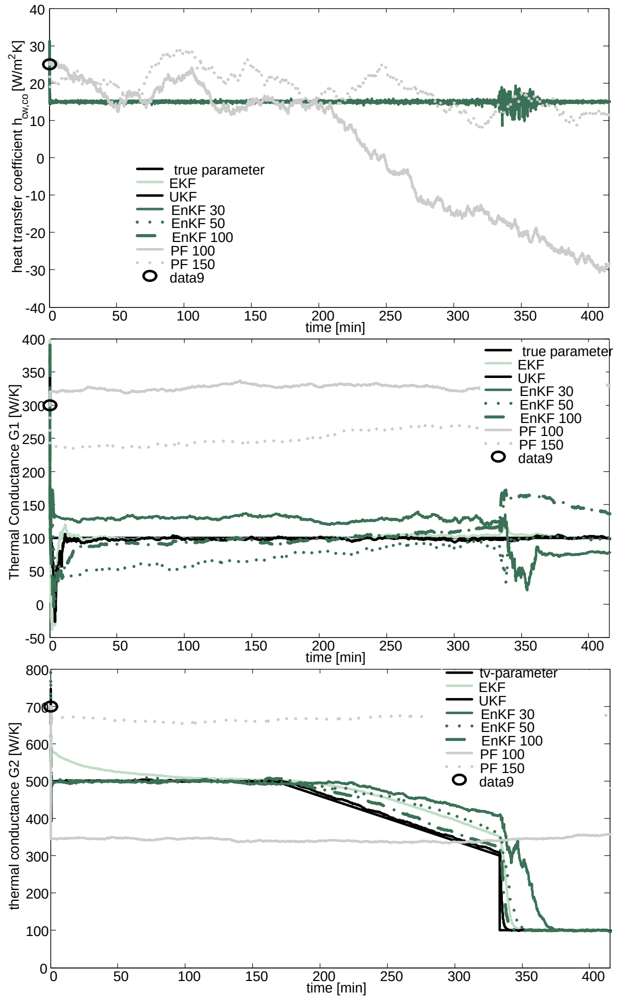


Figure 5. HRC estimated parameters: heat transfer coefficient $\alpha_{\mathrm{cw,co}}$ (above), G_1 (middle), and G_2 (below).

The estimators are simulated for 6.9 h with a sampling time of 60 s. The same estimator initial state, measurement and process noise covariance matrices are used for all the filters.

$$x_0 = [\vartheta_{\mathrm{cw}}, \vartheta_{\mathrm{co}}, \vartheta_{\mathrm{eb}}, \vartheta_{\mathrm{2a}}, \vartheta_{\mathrm{2b}}, \vartheta_{\mathrm{1a}}, \vartheta_{\mathrm{1b}}, \alpha_{\mathrm{cw,co}}, G_1, G_2]^T =$$
$$=[573, 293, 293, 323, 323, 323, 323, 15, 100, 500]^T$$
$$x_{0|0} = [353, 363, 313, 333, 303, 273, 273, 25, 300, 700]^T$$
$$Q_k = \mathrm{dgblk}\,[\mathrm{eye}(7) * 5\,\mathrm{E}\,{-3}, \mathrm{eye}(7) * 1\,\mathrm{E}\,{-3}]$$
$$R_k = \mathrm{diag}(\mathrm{eye}(4) * 5\,\mathrm{E}\,{-2})$$

The UKF parameters are $\{\alpha, \kappa, \beta\} = \{1\,\mathrm{E}\,{-2}, 0, 2\}$, the EnKF is simulated for $N = \{30, 50, 100\}$ and the PF is simulated for $N = \{100, 150\}$ particles. The estimated parameters for each filter are shown in Fig. 5. The EKF has a good performance with a similar response to the UKF for the first two parameters $\{\alpha_{\mathrm{cw,co}}, G_1\}$. For the third parameter G_2 the EKF converges slower than the UKF to the true time-varying parameter. But it outperforms the EnKF with ensembles of size less than 100 elements. The EnKF 100 and the UKF show a comparable performance for all the estimated parameters. The PF again shows a poor performance that can again be explained by the sampling empoverishment.

5. Conclusions

The recursive parameter estimation problem is analyzed for a general CHP model. Some relevant model parameters are estimated using simulated data sets. The best estimator performance is achieved by the UKF for the presented cases. The EKF performs well but some undesirable effects appear as a consequence of the use of the derivatives in its formulation. The EnKF performs similarly to the EKF for moderate size ensembles (i.e. less than 100) reaching a performance similar to the UKF for ensembles larger than 100 elements. The PF performs poorly but this can be probably fix by adding a roughnening algorithm in its formulation [19].

The computational cost of the estimators increases considerably from the EKF to the PF because of the number of projection required for every estimator iteration. Additionally, the CHP model is run from a Dymosim executable file and this slows down significantly the computational performance of the estimators (i.e. the computational time required for every estimator interation). Dymosim uses a slow file input/output interface and consequently exhaustive Montecarlo simulation analysis of the performance of the different estimators are prohibitively slow.

Many aspects concerning the online parameter estimation problem depends on the many characteristics of the real system: the number of uncertain state/parameters, the magnitude of uncertainty, the functional dependence of outputs on the uncertain state/parameters, the quality of output measurements, and the knowledge of system inputs [22]. So our future work will be focus on the identification of the CHP model from real experimental data. Data sets with a limited number of measurements from a CHP unit manufacturer may be available in a short time.

It is also of future interest to run the estimators using a the CHP model implemented in Matlab to speed up the iteration time and to run exhaustive analysis of the estimators.

Table 7. Estimators Notation

symbol	description
(Common notation)	
$k \in \mathbb{N}^+$	time index
$\hat{x}_{k\|k-1} \in \mathbb{R}^{n_x \times 1}$	a priori x estimate
$\hat{x}_{k\|k} \in \mathbb{R}^{n_x \times 1}$	a post. x estimate
$\hat{y}_{k\|k-1} \in \mathbb{R}^{n_y \times 1}$	a priori y estimate
$\hat{y}_{k\|k} \in \mathbb{R}^{n_y \times 1}$	a post. y estimate
$P_{k\|k-1} \in \mathbb{R}^{n_x \times n_x}$	a priori P estimate
$P_{k\|k} \in \mathbb{R}^{n_x \times n_x}$	a post. P estimate
$K_k \in \mathbb{R}^{n_x \times n}$	kalman gain
$\hat{x}_{0\|0} \in \mathbb{R}^{n_x \times 1}$	initial x estimate
$P_{0\|0} \in \mathbb{R}^{n_x \times n_x}$	initial P estimate
(extended Kalman filters)	
$F_{k-1} \in \mathbb{R}^{n_x \times n_x}$	jacob. f_{k-1} wrt. x_{k-1}
$L_{k-1} \in \mathbb{R}^{n_x \times n_w}$	jacob. f_{k-1} wrt. w_{k-1}
$H_k \in \mathbb{R}^{n_y \times n_x}$	jacob. h_k wrt. x_k
$M_k \in \mathbb{R}^{n_y \times n_v}$	jacob. h_k wrt. v_k
(unscented Kalman filters)	
$L \in \mathbb{N}^+$	num. aug. states
$\alpha, \beta, \kappa, \gamma, \lambda \in \mathbb{R}$	filter parameters
$\Lambda_m^i \in \mathbb{R}$	ith σ-pt. mean weight
$\Lambda_c^i \in \mathbb{R}$	ith σ-pt cov. weight
$\tilde{x}_{k\|k}^i \in \mathbb{R}^{Lx1}$	ith σ-pt
$\tilde{x}_{k\|k}^{(x)i} \in \mathbb{R}^{n_x \times 1}$	ith σ-pt x elements
$\tilde{x}_{k\|k}^{(w)i} \in \mathbb{R}^{n_w \times 1}$	ith σ-pt w elements
$\tilde{x}_{k\|k}^{(v)i} \in \mathbb{R}^{n_v \times 1}$	ith σ-pt v elements
$\tilde{P}_{k\|k} \in \mathbb{R}^{LxL}$	a post. aug. P estimate
$P_y \in \mathbb{R}^{n_y \times n_y}$	cov. y vector
$P_{xy} \in \mathbb{R}^{n_x \times n_y}$	crosscov. xy vectors
(particle filters and ensemble KF)	
$N \in \mathbb{N}$	num. particles
$x_{k\|k-1}^i$	ith a prior. particle
q^i	ith particle likelihood
$\tilde{q}^i$	normalized q^i

References

[1] H. I. Onovwiona and V. I. Ugursal, "Residential cogeneration systems: review of the current technology," *Renewable and Sustainable Energy Reviews*, pp. 1–43, 2004.

[2] G. E. Hovland, T. P. von Hoff, E. A. Gallestey, M. Antoine, D. Farruggio, and A. D. B. Paice, "Nonlinear estimation methods for parameter tracking in power plants," *Control Engineering Practice*, vol. 13, pp. 1341–1355, 2005.

[3] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

Table 8. CHP model notation

$u_{x=\{\mathrm{th,by,p1,p2}\}}$	x control signal $[-]$
$p_{x=\{\mathrm{ab,mn,ex}\}}$	x pressure $\left[\mathrm{N/m^2}\right]$
$\vartheta_{x=\{\mathrm{ab,mn,ex}\}}$	x temperature $[\mathrm{K}]$
$\dot{m}_{x=\{\mathrm{th,en},\beta,\varphi,\mathrm{by,co,pr,sc}\}}$	x mass flow rate $[\mathrm{kg/s}]$
$p_{me[x]=\{\mathrm{pg},b,\varphi,f\}}$	x mean effec.press.$[\mathrm{N/m}]$
$T_{x=\{\mathrm{en,ld}\}}$	x torque $[\mathrm{N\,m}]$
$\dot{H}_{x=\{\mathrm{en,ex}\}}$	x enthalpy flow rate $[\mathrm{W}]$
$\dot{Q}_{[x],[y]}$	heat flow rate from
$x=\{\mathrm{cg,cw,co,eb,1a,2a}\}$ $y=\{\mathrm{cw,co,eb,2a,2b,ab}\}$	x to y $[\mathrm{W}]$
$\dot{W}_{\mathrm{en}}$	work rate $[\mathrm{W}]$
P_{en}	electrical power $[\mathrm{W}]$
ω_{en}	angular velocity $[\mathrm{rad/s}]$
$V_{x=\{\mathrm{mn},d,c\}}$	x volume in $\left[\mathrm{m^3}\right]$
$\eta_{x=\{\mathrm{td,ec,vl}\}}$	x efficiency $[-]$
$c_{x=\{\mathrm{cw,co,eb,oi,1a,1b,2a,2b}\}}$	x spc.heat cap.$[\mathrm{J/kg\,K}]$
$A_{x=\{\mathrm{cw,co,eb,oi,1a,1b,2a,2b}\}}$	x area $\left[\mathrm{m^2}\right]$
$m_{x=\{\mathrm{cw,co,eb,oi,1a,1b,2a,2b}\}}$	x mass $[\mathrm{kg}]$
$\alpha_{x=\{\mathrm{cw,co,eb,oi,1a,1b,2a,2b}\}}$	x heat-tf.coef.$\left[\mathrm{W/m^2\,K}\right]$
$\beta_i, \delta_i, \gamma_i$	$i$ths $p_{\mathrm{mef}}, \dot{Q}_{\mathrm{cg,cw}}, \eta_{\mathrm{vl}}$ coef.
I_{en}	flywheel inertia $\left[\mathrm{kg/m^2}\right]$
$A_{\mathrm{th}}, \alpha_{\mathrm{th}}, d_{\mathrm{th}}, C_{\mathrm{th}}, \Delta\vartheta_{\mathrm{mn}}$	throttle valve parameters
$R_\beta, \kappa_\beta, \lambda, \sigma, H_l$	air-fuel parameters

subscript notation:

1a	hx1 side a	ec	electrical	td	thermdyn.
1b	hx1 side b	en	engine	th	throttle
2a	hx2 side a	eo	eng. oil	vl	volumet.
2b	hx2 side b	ex	exhaust	β	air
ab	ambient	mn	manifold	φ	fuel
by	bypass	ld	load	b	brake
cg	cyl. gas	p1	pump 1	c	compress
ci	coolant in	p2	pump 2	d	displmnt.
co	coolant out	pg	gass loss	f	friction
cw	cyl. wall	pr	primary	l	lower
eb	eng. block	sc	second.		

[4] A. Gelb, J. F. Kasper, R. A. Nash, C. F. Price, and A. A. Sutherland, *Applied Optimal Estimation.* The M.I.T. Press, 1974.

[5] J. I. Videla and B. Lie, "Simulation of a small scale si ice based cogeneration system in modelica/dymola," in *SIMS conference*, Helsinki, Finland, September 2006.

[6] ——, "Simulation of a bio-ethanol ice chp system," in *3rd International Green Energy Conference*, Sweden, 2007.

[7] D. Cho and J. K. Hedrick, "Automotive powertrain modeling for control," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 111, no. 4, pp. pp.586–576, 1989.

[8] E. Hendricks and S. C. Sorenson, "Mean value modeling of spark ignition engines," *SAE Technical Paper Series*, no. 900616, 1990.

[9] E. Hendricks and T. Vesterholm, "Analysis of mean value si engine models," *SAE Technical Paper Series*, no. 920682, pp. 1–19, 1992.

[10] L. Guzzella and C. H. Onder, *Introduction to Modeling and Control of Internal Combustion Engine Systems.* Springer, 2004.

[11] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*, ser. CRC applied mathematics and nonlinear science series. Chapman & Hall, 2000.

[12] S. F. Schmidt, *Application of State-Space Methods to Navigation Problems*, c.t. leondes ed. Academic Press, New York, San Francisco, London, 1966, vol. 3, pp. 293–340.

[13] D. Simon, *Optimal State Estimation – Kalman, H_∞, and Nonlinear Approaches.* Hoboken, New Jersey: John Wiley & Sons, Inc., 2006.

[14] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of the 1995 American Control Conference*, Seattle, WA, 1995, pp. 1628–1632.

[15] S. Julier and J. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," tech. rep., RRG, Dept. of Engineering Science, University of Oxford, Nov 1996, Tech. Rep., 1996.

[16] ——, "A new extension of the Kalman filter to nonlinear systems," in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997.

[17] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation (invited paper)," in *Proceedings of the IEEE*, vol. 92(3). IEEE Institute of Electrical and Electronics, 2004, pp. 401–422.

[18] T. B. Schön, "Estimation of nonlinear dynamic systems, theory and applications," Ph.D. dissertation, Linköpings Studies in Science and Technology, Sweden, 2006.

[19] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter– Particle Filters and Tracking Applications.* Artech House, 2004.

[20] G. Evensen, "The ensemble kalman filter: Theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, pp. 343–367, 2003.

[21] S. Gillijns, O. B. Mendoza, J. Chandrasekar, B. L. R. D. Moor, D. S. Bernstein, and A. Ridley, "What is the ensemble kalman filter and how well does it work?" in *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, USA, June 2006.

[22] R. F. Stengel, *Optimal Control And Estimation.* John Wiley & Sons, Inc., 1986.

Using an adaptive FEM to determine the optimal control of a vehicle during a collision avoidance manoeuvre

Karin Kraft and Stig Larsson
Mathematical Sciences, Chalmers University of Technology and Göteborg University
karin.kraft@chalmers.se
stig@chalmers.se
Mathias Lidberg
Applied Mechanics, Chalmers University of Technology
mathias.lidberg@chalmers.se

Abstract

The optimal manoeuvering of a vehicle during a collision avoidance manoeuvre is investigated. A simple model where the vehicle is modelled as point mass and the mathematical formulation of the optimal manoeuvre are presented. The resulting two-point boundary problem is solved by an adaptive finite element method and the theory behind this method is described.

Keywords: Vehicle dynamics, collision avoidance manoeuvre, optimal control, boundary value problem, adaptive finite element method.

1 Introduction

Historically, active safety systems for vehicles are designed to ensure that the driver can steer and brake the vehicle. Automatic controls are being incorporated in conventional safety systems such as ESC with the ability to minimise driver errors. It is important to evaluate how such systems perform in various situations. For this purpose the American National Highway Traffic Safety Administration has proposed a test called "sine with dwell" to evaluate the performance of a car during a collision avoidance manoeuvre. In such a manoeuvre the driver of a vehicle tries to avoid an object that suddenly appears in front of the vehicle [1].

This article is a theoretical investigation of how to combine braking and steering to perform a collision avoidance manoeuvre in an optimal way. The optimisation function has two goals. The primary objective is to achieve a vehicle trajectory distance to avoid collision. If the primary objective cannot be met, then a secondary objective is to minimise the final velocity of the unavoidable collision. This is an optimal control problem, since we want to find controls and states which minimise a quantity subject to constraints consisting of a dynamical system.

Methods for solving optimal control problem can be classified as either a *direct* or an *indirect* approach [4]. The direct approach approximates the dynamical system and then looks for a solution, such that the objective function is minimised. The indirect approach determines the necessary conditions for optimality, and then seeks their solution. Taking the indirect approach means that we have to derive the adjoint equations and optimality conditions explicitly. However, we use this approach because the indirect approach in combination with the finite element method gives us the possibility to control the error in the numerical solution of the optimality conditions over the entire interval. We believe that this is important for an efficient solver. Our first attempts in this direction are described in the present work. The most common numerical methods for solving optimal control problems based on either a direct or an indirect approach are multiple shooting or collocation methods [4]. However, in this work we use an adaptive finite element method similar to the one in [8] to solve the necessary optimality conditions that arise in an indirect approach. In the presented adaptive finite element method we derive an a posteriori error estimate which is used as a basis for error control and adaptive mesh refinement. Since we estimate the error over the entire time interval we can use the computational power where it is best needed. This gives us the ability to choose the level of modelling for the FEM solver and we also believe that we will be able to solve optimal control problems for more advanced vehicle models.

2 The collision avoidance manoeuvre

A traffic situation that presents a safety risk is defined for the investigation. A vehicle is driven on a plane homogeneous surface. There is an obstacle in front of the vehicle. How shall the driver manoeuver the vehicle in the best way in order to avoid collision and, if that is not possible, minimise the collision severity? In Figure 1 we show a picture of the steering in a collision avoidance manoeuvre. The driver performs the avoidance manoeuvre by braking and steering simultaneously. The manoeuver starts at time $t = 0$, at a distance a from the obstacle and with velocity U_0. After the manoeuvre the car hits or passes the obstacle at time T, with velocity U_T and at distance b from the original track.

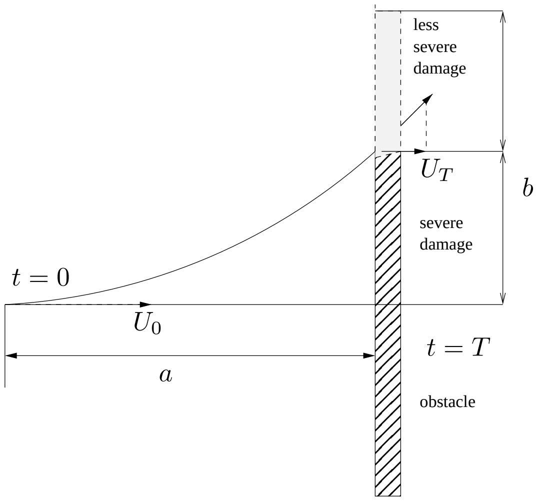


Figure 1: The collision avoidance manoeuvre

We know that the higher speed the vehicle has at the time of collision, the more severe the accident. Therefore we want to determine the best braking and steering strategy to avoid collision or minimise the speed perpendicular to the object at impact. This optimisation problem can be formulated as follows: given the manoeuvre distances a and b determine the braking and steering strategy that minimises the final velocity component U_T.

3 Point mass vehicle dynamics model

The driver controls the braking and steering but it is the friction forces acting on the car tyres that makes the vehicle move in a certain direction. For our purposes, the dynamics of the vehicle due to these forces can be modelled as a point mass [10]. We introduce the X-axis as the direction of the original track and the Y-axis as the axis perpendicular to the X-axis. The equations of planar motion for the vehicle then become

$$\begin{aligned} \ddot{X} &= -\mu g \cos(\beta), \\ \ddot{Y} &= \mu g \sin(\beta), \end{aligned} \tag{1}$$

where β is the angle between the X-axis and the sum of the forces between the tyres and the road, μ is the friction coefficient and g is the gravitational acceleration.

4 Optimal control theory for the collision avoidance manoeuvre

4.1 State-space formulation

To derive the necessary conditions of optimality, the final speed optimal control problem is formulated in state space by transforming differential equations (1) to first order differential equations. The equations of planar motion for the vehicle then become

$$\dot{z} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{U} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} U \\ V \\ -\mu g \cos(\beta) \\ \mu g \sin(\beta) \end{bmatrix}, \tag{2}$$

where U and V are the velocities in the X and Y directions, respectively.

We want to minimise the speed at the time of the accident in order to reduce the damage. Therefore we formulate an optimal control problem: Find the state $z(t) \in \mathbb{R}^n$ and control $\beta(t) \in \mathbb{R}^m$ which fulfill the following minimisation problem

$$\begin{aligned} \min \quad & \mathcal{J}(z,\beta) = c^{\mathrm{T}} z(T) \\ \text{s.t.} \quad & \dot{z}(t) = f(z,\beta), \\ & J_0 z(0) = z_0, \quad J_T z(T) = z_T. \end{aligned} \tag{3}$$

Here J_0 and J_T are diagonal matrices with zeroes or ones on the diagonals and f is given by the right hand side of (1) and $c^{\mathrm{T}} = (0,0,1,0)$.

Since this problem has a free terminal time we transform the time interval $t \in [0,T]$ into a normalised time interval $\tau \in [0,1]$ by introducing the new independent variable

$$\tau = \frac{t}{T}, \tag{4}$$

rewrite the equations in (3) for the new variable τ and add the trivial equation $\dot{T} = 0$. This results in a problem of the form (3) but with a fixed time interval.

4.2 Necessary conditions of optimality

Introducing the Hamiltonian,

$$H = \lambda^{\mathrm{T}} f(z,\beta),$$

and then applying variational calculus [6] to (3) leads to the following necessary conditions of optimality.

The optimal solution $(z^*(t), \lambda^*(t), \beta^*(t))$ fulfills the optimality conditions

$$\dot{z} = \frac{\partial H}{\partial \lambda} = f(z), \tag{5}$$

$$\dot{\lambda} = -\frac{\partial H}{\partial z} = -\left(\frac{\partial f}{\partial z}\right)^{\mathrm{T}} \lambda, \tag{6}$$

$$0 = \frac{\partial H}{\partial \beta} = \left(\frac{\partial f}{\partial \beta}\right)^{\mathrm{T}} \lambda, \tag{7}$$

the boundary conditions

$$J_0 z(0) = z_0, \quad J_T z(T) = z_T, \tag{8}$$

and the transversality conditions

$$(J - J_0)\lambda(0) = \lambda_0, \quad (J - J_T)\lambda(T) = \lambda_T, \tag{9}$$

where λ_0 and λ_T are obtained from $\mathcal{J}$. We note here that $x_0 \in R(J_0)$ and $x_T \in R(J_T)$ which means that the components of the adjoint variable λ that have boundary values are the ones complementary to the components of x that have boundary values. To simplify the problem we assume that the optimality condition (7) can be solved explicitly for β^*.

4.3 Reformulating the boundary value problem into standard form

General purpose software for treating boundary value problems for ordinary differential equations usually requires the problem to be reformulated into standard form [3]. We make this conversion by joining the states z and the costates λ into a new variable $x \in \mathbb{R}^d$ for $d = 2n$, and then redefining f by merging the right hand sides of (5) and (6). The resulting system is a two point boundary value problem with fixed time interval and separated linear boundary conditions,

$$\begin{aligned} &\dot{x} = f(x), \\ &I_0 x(0) = x_0, \;\; I_T x(1) = x_T, \end{aligned} \tag{10}$$

where $\dot{x}$ denotes the derivative of x with respect to the new independent variable τ.

5 An adaptive finite element method

5.1 Weak formulation

In this section we derive an adaptive finite element method. It consists of the discretisation of the problem with definitions of the right function spaces and an a posteriori error estimate. We start with the so called weak formulation. To obtain the weak formulation we multiply (10) by a test function $v \in V = C^1([0,T])$, integrate over the interval $[0,T]$ and the weak formulation of the problem is: Seek $x \in V$ such that

$$\begin{aligned} &I_0 x(0) = x_0, \;\; I_T x(T) = x_T, \\ &F(x,v) = \int_0^T (\dot{x} - f(x), v)\, dt = 0, \;\; \forall v \in V, \end{aligned} \tag{11}$$

where $(\cdot,\cdot)$ is the Cartesian scalar product in $\mathbb{R}^d$.

5.2 Discretisation of the problem

The problem in (11) is an infinite dimensional problem which we discretise as follows to get a finite problem. We discretise the time axis and introduce the trial and test spaces as follows.

- Mesh: $0 = t_0 < t_1 < t_2 < \ldots < t_N = T$, $h_n = t_n - t_{n-1}$ and $I_n = (t_{n-1}, t_n)$.
- Trial space: $W_h = \mathbb{R}^d \times \{w : w|_{I_n} \in P^0(I_n)\} \times \mathbb{R}^d$, discontinuous piecewise constant functions.
- Test space: $V_h = \{v : v|_{I_n} \in P^1(I_n) \cap C^0([0,T])\}$, continuous piecewise linear functions.

The notation $P^k(I_n)$ refers to the $\mathbb{R}^d$-valued polynomials of degree k on the interval I_n. We also introduce the left and right limits $w_n^{\pm} = \lim_{t \to t_n^{\pm}} w(t)$, and jumps $[w]_n = w_n^+ - w_n^-$. The two factors $\mathbb{R}^d$ in W_h contain the boundary values w_0^- and w_N^+. Now our finite element problem can be stated: Find a function $X \in W_h$ which fulfills

$$\begin{aligned} &I_0 X_0^- = x_0, \;\; I_T X_N^+ = x_T, \\ &F(X,v) = \sum_{n=1}^{N} \int_{I_n} (\dot{X} - f(X), v)\, dt \\ &\qquad + \sum_{n=0}^{N} ([X]_n, v_n) = 0, \;\; \forall v \in V_h. \end{aligned} \tag{12}$$

Here the definition of the form F from (11) has been extended to include the contributions from the jump terms which appear since we use discontinuous trial functions. Since the trial space consists of piecewise constant functions, $\dot{X} = 0$. Hence, (12) results in a system of $(N+2)d$ equations that have to be solved, more precisely, d boundary conditions and $(N+1)d$ equations. With boundary conditions at both ends, the equations are coupled and thus we cannot use time stepping and therefore the equations in the system have to be solved simultaneously.

5.3 An a posteriori error estimate

An adaptive finite element method gives us the possibility to control the error in the numerical solution.

In order to derive an a posteriori error estimate we introduce ϕ as the solution to the adjoint problem to (10) with data functional G. We want to construct an equation for the error, $e = X - x$ where $e \in W = \mathbb{R}^d \times \{w|_{I_n} : w \in C^1(I_n)\} \times \mathbb{R}^d$, the difference between the real and the computed solution. The details of the a posteriori error estimate are given below.

5.3.1 Proof of the error estimate

We subtract (11) from (12),

$$\begin{aligned} &F(X,v) - \underbrace{F(x,v)}_{=0,\forall v\in V} \\ &= \sum_{n=1}^{N} \int_{I_n} (\dot{X} - \dot{x} - (f(X) - f(x)), v)\, dt \\ &\quad + \sum_{n=0}^{N} ([X-x]_n, v_n). \end{aligned} \tag{13}$$

Since f is nonlinear we linearise $f(X)-f(x)$ by rewriting it as follows

$$\begin{aligned} &f(X) - f(x) \\ &= \int_0^1 \frac{d}{d\theta} f(\theta(X(t)-x(t)) + x(t))\, d\theta \\ &= \underbrace{\int_0^1 Df(\theta(X(t)-x(t)) + x(t))\, d\theta}_{=A(t)}(X(t)-x(t)). \end{aligned}$$

Inserting this in (13) we get

$$\begin{aligned} &F(X,v) \\ &= \sum_{n=1}^{N} \int_{I_n} (\dot{X} - \dot{x} - (f(X) - f(x)), v)\, dt \\ &\quad + \sum_{n=0}^{N} ([X-x]_n, v_n) \\ &= \sum_{n=1}^{N} \int_{I_n} (\dot{e} - A(t)e, v)\, dt \\ &\quad + \sum_{n=0}^{N} ([e]_n, v_n), \quad \forall v \in V. \end{aligned} \tag{14}$$

Since (14) is linear in both e and v we introduce a bilinear form to simplify the notation. The bilinear form B is defined as

$$\begin{aligned} B(w,v) &= \sum_{n=1}^{N} \int_{I_n} (\dot{w} - A(t)w, v)\, dt + \sum_{n=0}^{N} ([w]_n, v_n) \\ &+ (I_0 w_0^-, v_0) - (I_T w_N^+, v_N), \quad w \in W, v \in V, \end{aligned} \tag{15}$$

Now we can write the equation for the error (14) with the bilinear form as

$$\begin{aligned} &e \in W \\ &B(e,v) = F(X,v), \ \ \forall v \in V. \end{aligned} \tag{16}$$

Partial integration of (15) gives us the backward form of the bilinear form

$$\begin{aligned} B(w,v) &= \sum_{n=1}^{N} \int_{I_n} (\dot{w} - A(t)w, v)\, dt \\ &\quad + \sum_{n=0}^{N} ([w]_n, v_n) \\ &\quad + (I_0 w_0^-, v_0) - (I_T w_N^+, v_N) \\ &= \sum_{n=1}^{N} \int_{I_n} (w, -\dot{v} - A(t)^{\mathrm{T}} v)\, dt \\ &\quad - (w_0^-, (I - I_0)v_0) + (w_N^+, (I - I_T)v_N), \\ &\quad w \in W, v \in V. \end{aligned} \tag{17}$$

This suggests the dual problem with arbitrary data functional G

$$\begin{aligned} &\phi \in V \\ &B(w,\phi) = G(w), \ \ \forall w \in W. \end{aligned} \tag{18}$$

We put $v = \phi$ in (16) and $w = e$ in (18) to obtain

$$G(e) = B(e,\phi) = F(X,\phi), \tag{19}$$

that is

$$\begin{aligned} G(e) &= B(e,\phi) \\ &= F(X,\phi) = \sum_{n=1}^{N} \int_{I_n} (\dot{X} - f(X), \phi)\, dt \\ &\quad + \sum_{n=0}^{N} ([X]_n, \phi_n). \end{aligned} \tag{20}$$

Subtracting a Lagrange node interpolant $\tilde{\phi} \in V_h$ from ϕ in the right hand side of (20) using (12) gives us

$$G(e) = \sum_{n=1}^{N} \int_{I_n} (\dot{X} - f(X), \phi - \tilde{\phi})\, dt + \sum_{n=0}^{N} ([X]_n, \phi_n - \tilde{\phi}_n).$$

Hence,

$$
\begin{aligned}
|G(e)| \leq {} & \Big| \sum_{n=1}^{N} \int_{I_n} (\dot{X} - f(X), \phi - \tilde{\phi}) \, dt \Big| \\
& + \Big| \sum_{n=0}^{N} ([X]_n \, , \phi_n - \tilde{\phi}_n) \Big| \\
\leq {} & \underbrace{\sum_{n=1}^{N} \int_{I_n} \|\dot{X} - f(X)\| \|\phi - \tilde{\phi}\| \, dt}_{I} \\
& + \underbrace{\sum_{n=0}^{N} \| \, [X]_n \, \| \|\phi_n - \tilde{\phi}_n\|}_{II} .
\end{aligned} \tag{21}
$$

Now we have the basis for an error estimate, but we want the method to be symmetric, meaning that we want each interior node to contribute to the error estimate on both sides of the node. To do this we rewrite the last term II in (21) as follows

$$
\begin{aligned}
\sum_{n=0}^{N} \| \, [X]_n \, \| \|\phi_n - \tilde{\phi}_n\| = {} & \| \, [X]_0 \, \| \|\phi_0 - \tilde{\phi}_0\| \\
& + \frac{h_1}{h_1 + h_2} \| \, [X]_1 \, \| \|\phi_1 - \tilde{\phi}_1\| \\
& + \sum_{n=2}^{N-1} \Big(\frac{h_n}{h_n + h_{n+1}} \| \, [X]_n \, \| \|\phi_n - \tilde{\phi}_n\| \\
& \quad + \frac{h_n}{h_n + h_{n-1}} \| \, [X]_{n-1} \, \| \|\phi_{n-1} - \tilde{\phi}_{n-1}\| \Big) \\
& + \frac{h_N}{h_{N-1} + h_N} \| \, [X]_{N-1} \, \| \|\phi_{N-1} - \tilde{\phi}_{N-1}\| \\
& + \| \, [X]_N \, \| \|\phi_N - \tilde{\phi}_N\|.
\end{aligned} \tag{22}
$$

At this stage we introduce the notation $\|v\|_{I_n} = \max_{I_n} \|v\|$ for the maximum norm of a function on an interval. Now we note that $\phi - \tilde{\phi} \in V$ is continuous and the following estimates $\|\phi_n - \tilde{\phi}_n\| \leq \|\phi - \tilde{\phi}\|_{I_n}$ and $\|\phi_n - \tilde{\phi}_n\| \leq \|\phi - \tilde{\phi}\|_{I_{n+1}}$ hold. Using this we can estimate the last term in (22) by

$$
\begin{aligned}
& \sum_{n=0}^{N} \| \, [X]_n \, \| \|\phi_n - \tilde{\phi}_n\| \\
& \leq \left(\| \, [X]_0 \, \| + \frac{h_1}{h_1 + h_2} \| \, [X]_1 \, \| \right) \|\phi - \tilde{\phi}\|_{I_1} \\
& + \sum_{n=2}^{N-1} \Big(\frac{h_n}{h_n + h_{n+1}} \| \, [X]_n \, \| \\
& \quad + \frac{h_n}{h_n + h_{n-1}} \| \, [X]_{n-1} \, \| \Big) \|\phi - \tilde{\phi}\|_{I_n} \\
& + \left(\frac{h_N}{h_{N-1} + h_N} \| \, [X]_{N-1} \, \| + \| \, [X]_N \, \| \right) \|\phi - \tilde{\phi}\|_{I_N} .
\end{aligned} \tag{23}
$$

The term I in (21) (where $\dot{X} = 0$) can be estimated as follows

$$
\begin{aligned}
& \sum_{n=1}^{N} \int_{I_n} \|\dot{X} - f(X)\| \|\phi - \tilde{\phi}\| \, dt \\
& \quad \leq \sum_{n=1}^{N} h_n \|\dot{X} - f(X_n^-)\|_{I_n} \|\phi - \tilde{\phi}\|_{I_n} .
\end{aligned} \tag{24}
$$

Collecting the estimates (24) and (23) we now have

$$
\begin{aligned}
|G(e)| \leq {} & \sum_{n=1}^{N} h_n \|\dot{X} - f(X)\|_{I_n} \|\phi - \tilde{\phi}\|_{I_n} \\
& + \left(\| \, [X]_0 \, \| + \frac{h_1}{h_1 + h_2} \| \, [X]_1 \, \| \right) \|\phi - \tilde{\phi}\|_{I_1} \\
& + \sum_{n=2}^{N-1} \Big(\frac{h_n}{h_n + h_{n+1}} \| \, [X]_n \, \| \\
& \quad + \frac{h_n}{h_n + h_{n-1}} \| \, [X]_{n-1} \, \| \Big) \|\phi - \tilde{\phi}\|_{I_n} \\
& + \Big(\frac{h_N}{h_{N-1} + h_N} \| \, [X]_{N-1} \, \| \\
& \quad + \| \, [X]_N \, \| \Big) \|\phi - \tilde{\phi}\|_{I_N} .
\end{aligned}
$$

According to [5] we have the following error bound for the interpolant, $\tilde{\phi}$,

$$
\left\| \phi - \tilde{\phi} \right\|_{I_n} \leq C h_n \int_{I_n} \left\| \ddot{\phi} \right\| \, dt.
$$

With

$$
\begin{aligned}
\mathcal{R}_1 = {} & h_1 \|\dot{X} - f(X)\|_{I_1} \\
& + h_1 \| \, [X]_{n-1} \, \| + \frac{h_1}{h_1 + h_2} \| \, [X]_1 \, \|,
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{R}_n = {} & h_n \|\dot{X} - f(X)\|_{I_n} \\
& + \frac{h_n}{h_n + h_{n+1}} \| \, [X]_n \, \| \\
& + \frac{h_n}{h_n + h_{n-1}} \| \, [X]_{n-1} \, \|, \quad n = 2, \ldots, N - 1,
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{R}_N = {} & h_N \|\dot{X} - f(X)\|_{I_N} + \\
& \frac{h_N}{h_{N-1} + h_N} \| \, [X]_{N-1} \, \| + \| \, [X]_n \, \|,
\end{aligned}
$$

and

$$
\mathcal{I}_n = C h_n \int_{I_n} \left\| \ddot{\phi} \right\| \, dt,
$$

we can write the error estimate as

$$
|G(e)| \leq \sum_{n=1}^{N} \mathcal{R}_n \mathcal{I}_n, \tag{25}
$$

where $e = X - x$ is the error and $\mathcal{R}_n$ is essentially the residual, $\dot{X} - f(X)$, expressing how well the differential equation is satisfied by the numerical solution. The weights $\mathcal{I}_n$ depend on the solution to the adjoint problem, ϕ, and express the sensitivity of the error quantity $G(e)$ to the local residuals. The functional G is chosen to be the quantity in which we want to measure the error, for example, $G(e) = \frac{e}{\|e\|}$. The error resulting from approximate nonlinear equation solver is small compared to the error resulting from the discretisation and is therefore neglected in this estimate. Checking which intervals give large contributions to the error estimate (25), we can refine the intervals where the contributions are large and vice versa. Using (25) we obtain an adaptive procedure where we refine those intervals that give large contributions to the estimate and vice versa, see below.

5.4 Implementation

The finite element discretisation of (10) results in the system (12) to be solved. Since we have boundary conditions at both ends it is a coupled system of equations that we have to solve simultaneously. The system is also nonlinear and the nonlinearity is handled using a damped Newton method [7] to extend the convergence region. The initial guess is decided by a homotopy process [3]. Once a solution is calculated the error estimate above is computed. Then we apply the criterion

$$\sum_{n=1} \mathcal{R}_n \mathcal{I}_n \leq \delta,$$

where δ is a given tolerance. If the error is too large compared to the tolerance an iteration is made over the intervals and the mesh is refined where the error is large. New nodes are inserted according to the *principle of equidistribution*, that is, we want to insert nodes such that the contribution to the error is the same from each time interval. A new solution is calculated on the refined mesh and so on until the solution has reached the desired accuracy. In theory the mesh can also be coarsened but we have not implemented this.

The error estimate is dependent of the solution to the dual problem. We need to approximate the unknown data $G(e) = \frac{e}{\|e\|}$ to solve the dual problem numerically. We do this by a Richardson extrapolation using twice the number of nodes. Then the dual problem is solved with the finite element method.

The solver is a prototype solver and it has been implemented in Matlab. More information regarding the implementation can be found in [2].

6 Results

The indirect approach to our optimal control problem results in a boundary value problem. This makes it possible to compare our new approach to the boundary value solver `bvp4c` in Matlab [11]. In Table 1 we can see the results from various choices of the manoeuvre distances a and b. We have used the same initial velocity $u_0 = 90$ km/h and constant friction μ for all cases. We can see that the FEM code is almost always about three times slower than `bvp4c` but it always uses fewer nodes. There is also a remarkable case where `bvp4c` solves the problem in about 30 seconds and with 3415 nodes compared to 1.3 seconds and 21 nodes for the finite element solver. The problem becomes difficult to solve but our FEM solver performs well, maybe due to the adaptivity. There is also one problem that the finite element solver can solve but `bvp4c` cannot.

In some cases where `bvp4c` finds a solution the FEM solver seems to compute the wrong one, maybe by missing a singularity. On the other side of the singularity it continues on another solution. Some results about existence and uniqueness of solutions to boundary value problems can be found in [7] and [9]. This aspect of our solver is something that we have to investigate further.

Figure 2 and 3 show some results from the case with initial velocity $u_0 = 90$ km/h (25 m/s) and the manoeuvre distances $a = 50$ m and $b = 8$ m. We see in the figures that the solutions from the FEM solver and `bvp4c` coincide. The final velocity is 53.32 km/h from `bvp4c` and 53.37 km/h from the FEM solver.

		FEM	FEM	`bvp4c`	`bvp4c`
a [m]	b [m]	CPU [s]	nodes	CPU [s]	nodes
40	6	2.54	15	0.27	25
50	9	1.59	90	0.36	41
50	8	0.61	10	0.24	28
50	5	0.52	10	0.27	37
50	3	1.45	10	-	-
60	8	1.45	21	18.39	1287
60	6	1.46	10	0.63	81
60	5	3.45	10	3.06	286

Table 1: Performance of the FEM solver and `bvp4c` measured in CPU time and number of nodes for different combinations of manoeuvre distances.

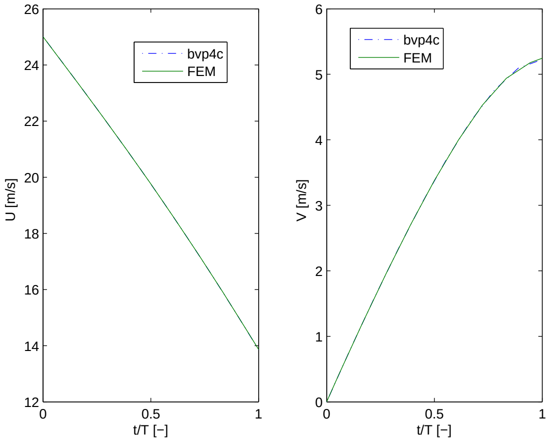


Figure 2: The optimal velocities in the X and Y-directions for the manoeuvre distances $a = 50$ m and $b = 8$ m.

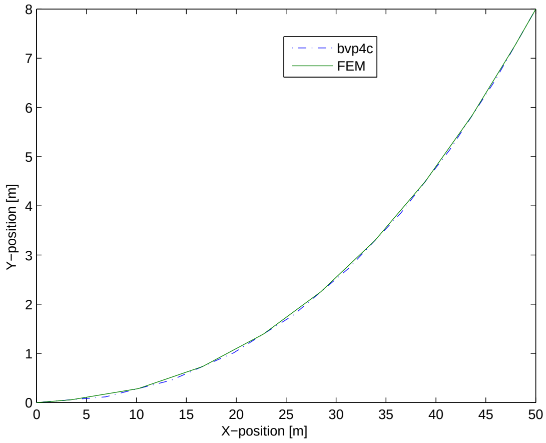


Figure 3: The position of the vehicle when it is manoeuvered in the optimal way for the manoeuvre distances $a = 50$ m and $b = 8$ m.

7 Conclusion

In this article we have presented an adaptive finite element method for solving optimal control in vehicle dynamics. Modelling the vehicle as a point mass, we obtain a system of ordinary differential equations which is solved, together with the constraints imposed by the manoeuvre, using the adaptive finite element method. With this approach, we can control the error and concentrate our resources to the most sensitive parts of the computations.

We have compared the finite element method to the Matlab solver `bvp4c` and found that there are at least some cases where our solver outperforms `bvp4c`. It is noteworthy that in all studied cases, our method uses fewer nodes to find the same solution. At the moment the finite element solver is slower than `bvp4c`, but up to this point no extra effort has been put in optimising the code. Thus, it is expected that the computation time can be reduced by a more efficient implementation. Further, these comparisions are very preliminary, since the accuracies of both methods depend on error tolerances that are not directly comparable. We are not sure that the settings are equal. Still, since the quality of the solutions have been similar throughout our computations, we feel confident that our comparision is reasonable.

We have also noted that our solver is sensitive to the initial guess. If we give the solver a poor initial guess for some component, it may fail to solve the problem or show a dramatical increase in computation time. To make the solver more useful we have to make it more robust to poor initial guesses.

The model used in this article may look simple. However, when it comes to evaluation of combined steering and braking versus braking and then steering, the behaviour of this model gives insights into the behaviour of the more realistic vehicle models and manoeuvres we will consider. In our future work we also intend to compare the performance of our indirect approach to the direct approach.

References

[1] National Highway Traffic Safety Administration, *Laboratory Test Instruction for FMVSS 126 Stability Control Systems*, `http://www.nhtsa.dot.gov`.

[2] C. Andersson, *Solving optimal control problems using FEM*, Master's thesis, Chalmers University of Technology, 2007.

[3] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, first ed., Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1988.

[4] J. T. Betts, *Survey of numerical methods for trajectory optimization*, AIAA J. Guidance Control Dynam. **21** (1998), 193–207.

[5] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, second ed., Springer-Verlag, New York, 2002.

[6] A. E. Bryson, Jr and Y. Ho, *Applied Optimal Control*, revised printing ed., Hemisphere Publishing Corporation, Washington, D.C, 1975.

[7] P. Deuflhard and F. Bornemann, *Scientific Computing with Ordinary Differential Equations*, Springer-Verlag New York Inc., New York, 2002.

[8] D. J. Estep, D. H. Hodges, and M. Warner, *Computational error estimation and adaptive error control for a finite element solution of launch vehicle trajectory problems*, SIAM J. Sci. Comput. **21** (1999), 1609–1631.

[9] H. B. Keller, *Numerical Methods for Two-Point Boundary-Value Problems*, Dover Publications, Inc., New York, 1992.

[10] B. Schmidtbauer, *Sväng och bromsa samtidigt (in Swedish)*, Teknisk Tidskrift (1971).

[11] L. F. Shampine, J. Kierzenka, and M. W. Reichelt, *Solving Boundary Value Problems for Ordinary Differential Equations in Matlab using* `bvp4c`, Tech. report, MathWorks, 2000, `http://www.mathworks.com`.

Towards Run-time Debugging of Equation-based Object-oriented Languages

Adrian Pop and Peter Fritzson
Programming Environments Laboratory
Department of Computer and Information Science
Linköping University
{adrpo, petfr}@ida.liu.se

Abstract

The development of today's complex products requires advanced integrated environments and modeling languages for modeling and simulation. Equation-based object-oriented declarative (EOO) languages are emerging as the key approach to physical system modeling and simulation. The increased ease of use, the high abstraction and the expressivity of EOO languages are very attractive properties. However, these attractive properties come with the drawback that programming and modeling errors are often hard to find. In this paper we propose an integrated framework for run-time debugging of equation-based modeling languages. The framework integrates classical debugging techniques with special techniques for debugging EOO languages and is based on graph visualization and interaction. The debugging framework targets the Modelica language.

Keywords: Run-time Debugging, Modeling and Simulation, Graph Visualization and Interaction, Eclipse.

1. Introduction

The development in system modeling has come to the point where complete modeling of systems is possible, e.g. the complete propulsion system, fuel system, hydraulic actuation system, etc., including embedded software can be modeled and simulated concurrently. This does not mean that all components are dealt with down to the very smallest details of their behavior. It does, however, mean that all functionality is modeled, at least qualitatively.

Such advanced development of today's complex products requires integrated environments and modeling languages for modeling and simulation. Equation-based object-oriented declarative (EOO) languages are the key approach to physical system modeling and simulation. The increased ease of use, the high abstraction and the expressivity of EOO languages are very attractive properties. However, these attractive properties come with the drawback that programming and modeling errors are often hard to find. This paper proposes an integrated framework for run-time debugging of equation-based modeling languages. The framework integrates classical debugging techniques with special techniques for debugging EOO languages and is based on graph visualization and interaction.

The paper is structured as follows: Section 2 presents the background of EOO languages and their current debugging techniques. Section 3 presents our debugging idea, whereas Section 4 presents the debugging framework. Conclusions and future work are presented in Section 5.

2. Background and Related Work

In this section we briefly introduce the Modelica language which is the target language for our debugging framework and describe the current debugging methods for EOO languages.

2.1 Modelica

Modelica [4][5] is a modern language for equation-based object-oriented mathematical modeling of primarily physical systems. A non-profit organization named "Modelica Association" was founded the year 2000 for further development and promotion of Modelica. Several tools implement the Modelica specification, ranging from open-source such as OpenModelica [3], to commercial implementations like Dymola [8] and MathModelica [7].

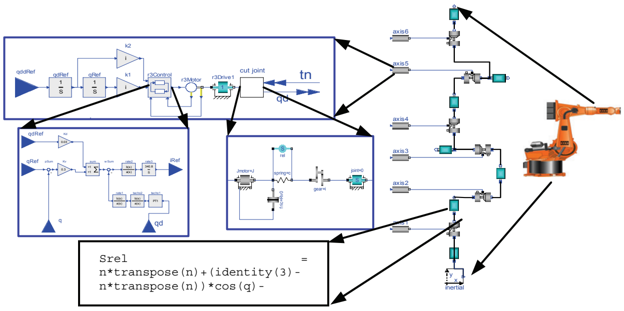
The language allows defining models in a declarative manner, modularly and hierarchically. The multi-domain capability of Modelica allows combining electrical, mechanical, hydraulic, thermodynamic, etc., model components within the same application model. In short, Modelica has improvements in several important areas:

- *Object-oriented mathematical modeling*. This technique makes it possible to create model components, which are employed to support hierarchical structuring, reuse, and evolution of large and complex models covering multiple technology domains.

- *Acausal modeling*. Modeling is based on equations instead of assignment statements as in traditional input/output block abstractions. Direct use of equations significantly increases re-usability of model components, since components adapt to the data flow context in which they are used. Interfacing with traditional software is also available in Modelica.
- *Physical modeling of multiple application domains*. Model components can correspond to physical objects in the real world, in contrast to established techniques that require conversion to "signal" blocks with fixed input/output causality. In Modelica the structure of the model naturally correspond to the structure of the physical system in contrast to block-oriented modeling tools such as Simulink. For application engineers, such "physical" components are particularly easy to combine into simulation models using a graphical editor.

Hierarchical system architectures can easily be described with Modelica thanks to its powerful component model. The Modelica component model includes the following three items: a) components, b) a connection mechanism, and c) a component frame-work. Components are connected via the connection mechanism realized by the Modelica system, which can be visualized in connection diagrams. The component frame-work realizes components and connections, and ensures that communication works over the connections.

For systems composed of acausal components with behavior specified by equations, the direction of data flow, i.e., the causality is initially unspecified for connections between those components. Instead the causality is automatically deduced by the compiler when needed. Components have well-defined interfaces consisting of ports, also known as connectors, to the external world. A component may internally consist of other connected components, i.e., hierarchical modeling.. Figure 1 shows hierarchical component-based modeling of an industry robot.



```
Srel                                    =
n*transpose(n)+(identity(3)-
n*transpose(n))*cos(q)-
```

Figure 1. Hierarchical model of an industrial robot, including components such as motors, bearings, control software, etc. At the lowest (class) level, equations are typically found.

2.2 Debugging techniques for EOO languages

In the context of debugging declarative equation-based object-oriented languages both the static and the dynamic (run-time) aspects have to be addressed.

The static aspect of debugging EOO languages deals with inconsistencies in the underlying system of equations:

1. *Overconstrained* system: the number of variables is smaller than the number of equations, which means that some equations have to be removed when solving the system of equations.
2. *Underconstrained* system: the number of variables is larger than the number of equations, which means that more equations have to be added in order to solve the system of equations.

The dynamic (run-time) aspect of debugging EOO languages addresses run-time errors that may appear due to faults in the model:

1. *model configuration*: when parameters values for the model simulation are incorrect.
2. *model specification*: when the equations that specify the model behavior are incorrect.
3. *algorithmic code*: when the functions called from equations return incorrect results.

Methods for both static and dynamic (run-time) debugging of the EOO languages have been proposed earlier [1][2]. With the new Modelica 3.0 language specification, the static debugging of Modelica presents a rather small benefit, since all models are required to be balanced. All models from already checked libraries will already be balanced; only newly written models might be unbalanced.

In the context of dynamic (run-time) aspect of debugging of EOO languages, [1] proposes an automated algorithmic debugging solution in which the user has to provide a correct diagnostic specification of the model which is used to generate assertions at runtime. Moreover, starting from an erroneous variable value the user explores the dependent equations (a slice of the program) and acts like an "oracle" to guide the debugger in finding the error.

In this paper we present a different approach that does not require the user to write diagnostic specifications of the model. Our method is based the integration between graph visualization/interaction and execution-based debugging of algorithmic code.

3. Proposed Debugging Method

In this section we present our run-time debugging method. The proposed integration within a general debugging framework for EOO languages is presented in the next section.

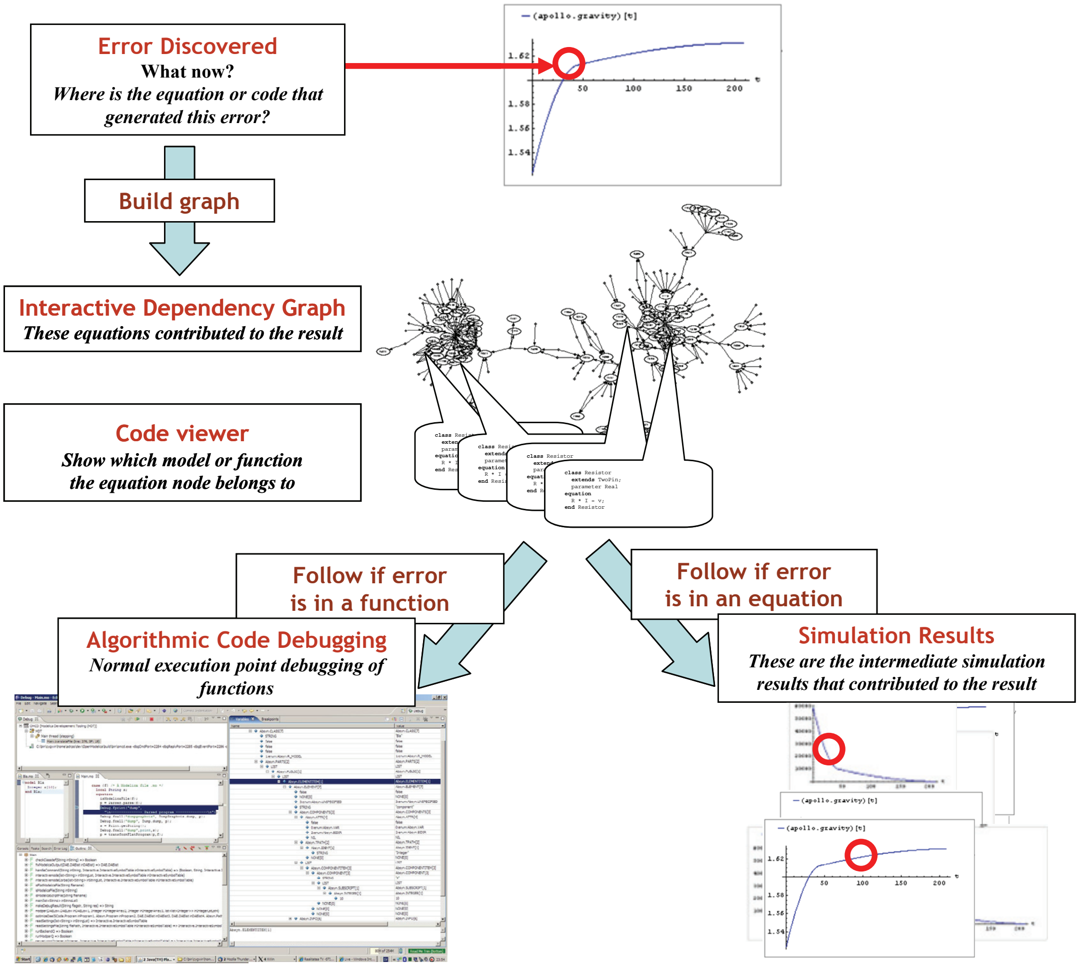

Error Discovered
What now?
Where is the equation or code that generated this error?
Build graph
Interactive Dependency Graph
These equations contributed to the result
Code viewer
Show which model or function the equation node belongs to
Follow if error is in a function
Follow if error is in an equation
Algorithmic Code Debugging
Normal execution point debugging of functions
Simulation Results
These are the intermediate simulation results that contributed to the result
(apollo.gravity)[t]
1.62
1.58
1.56
1.54
50
100
150
200
t
class Resistor
extends TwoPin;
parameter Real
equation
R * I = v;
end Resistor

Figure 2. Debugging approach overview.

3.1 Run-time Debugging Method

Our method partly follows the approach proposed in [1]. However, our approach does not require the user to write diagnostic specifications of models. Also, the approach we present here can also handle the debugging of algorithmic code using classic debugging techniques [6].

The overview of our debugging strategy is presented in Figure 2. In short, our run-time debugging method is based on the integration of the following:

1. Graph visualization and interaction.
2. Presentation of simulation results and modeling code.
3. Mapping of errors to model code positions.
4. Execution-based debugging of algorithmic code.

In the following we present a possible debugging session.

During the simulation phase, the user discovers an error in the plotted results. The user marks either the entire plot of the variable that presents the error or parts of it and starts the debugging framework. The debugger presents an (IDG) interactive dependency graph (the dynamic program slice with respect to the variable with the wrong value) where nodes consist of all the equations, functions, parameter value definitions, and inputs that were used to calculate the wrong variable value. The variable with the erroneous value is displayed in a special node which is the root of the graph. The interactive dependency graph contains two types of edges:

1. *Calculation dependency edges*: the directed edges labeled by variables or parameters which are inputs (used for calculations in this equation) or outputs (calculated from this equation) from/to the equation displayed in the node.
2. *Origin edges*: the undirected edges that tie the equation node to the actual model which this equation belongs to.

The user interacts with the dependency graph in several ways:

- *Displaying simulation results* through selection of the variables (or parameters) names (edge labels). The plot of a variable is shown in a popup window. In this way the user can quickly see if the plotted variable has erroneous values.

- *Classifying a variable* as having wrong values: addition of the variable to the set of variables with wrong values.
- *Classifying an equation as correct* eliminates the equation node from the graph and builds a new graph based on the inputs of the correct equation node.
- *Building a new dependency graph* based on the new set of variables with wrong values (classified variables) or by modifying the equations or parameter values nodes.
- *Displaying model code* by following origin edges.
- *Invoking the algorithmic code debugging subsystem* when the user suspects that the result of a variable calculated in an equation which contains a function call is wrong, but the equation seems to be correct.

Using these interactive dependency graph facilities the user can follow the error from manifestation to origin.

Our debugging method can also start from multiple variables with wrong values with the premise that the error might be at the confluence of several dependency graphs.

4. Run-time Debugging Framework

In this section we present the first prototype of the debugging framework based on the proposed method from the previous section. In this paper the debugging framework is limited to error tracking of a single variable with wrong results.

4.1 Translation in the Debugging Framework

The debugging framework is closely related to the translation process. The translation process from the modeling language down to simulation code is presented in the following. The Modelica translation process has several stages (Figure 3):

- *Parser* – breaks the model down into tokens and builds the abstract syntax tree. (not in Figure 3)
- *Translator (Flattening and elaboration)* – reports the errors and flatten the model hierarchy and applying modification.
- *Analyzer* – analyses the system of equations and sorts the equations in the order they need to be solved
- *Optimizer* – optimizes the sorted system of equations
- *Code Generator* – generates C code linked with the simulation runtime and solvers.
- *C Compiler* – compiles the generated C code to an executable
- *Simulation* – the executable is executed to generate the simulation results.

As one cans see, the translation process is complex and most of the transformations performed on the models are destructive. For debugging purposes all the transformations performed in each stage needs to be recorded to be able to point the errors to the user using the high level Modelica code.

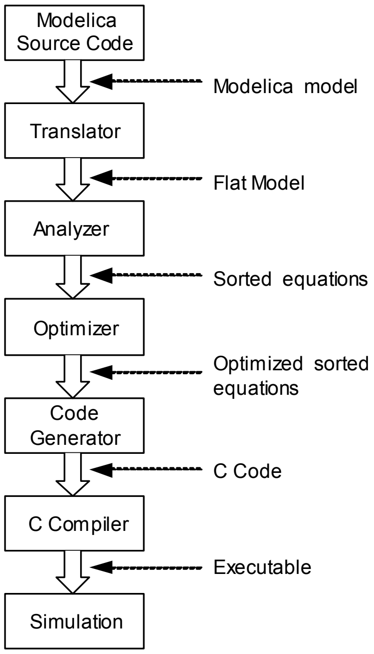


Figure 3. Translation stages from Modelica code to executing simulation.

The debugging framework alters the Modelica translation stages by introducing means to map (and save such mapping) each transformed model element back to its origin as presented in Figure 4.

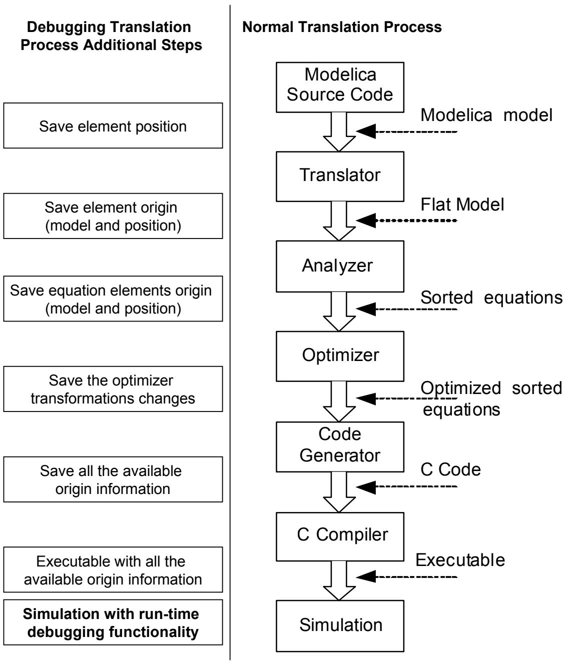
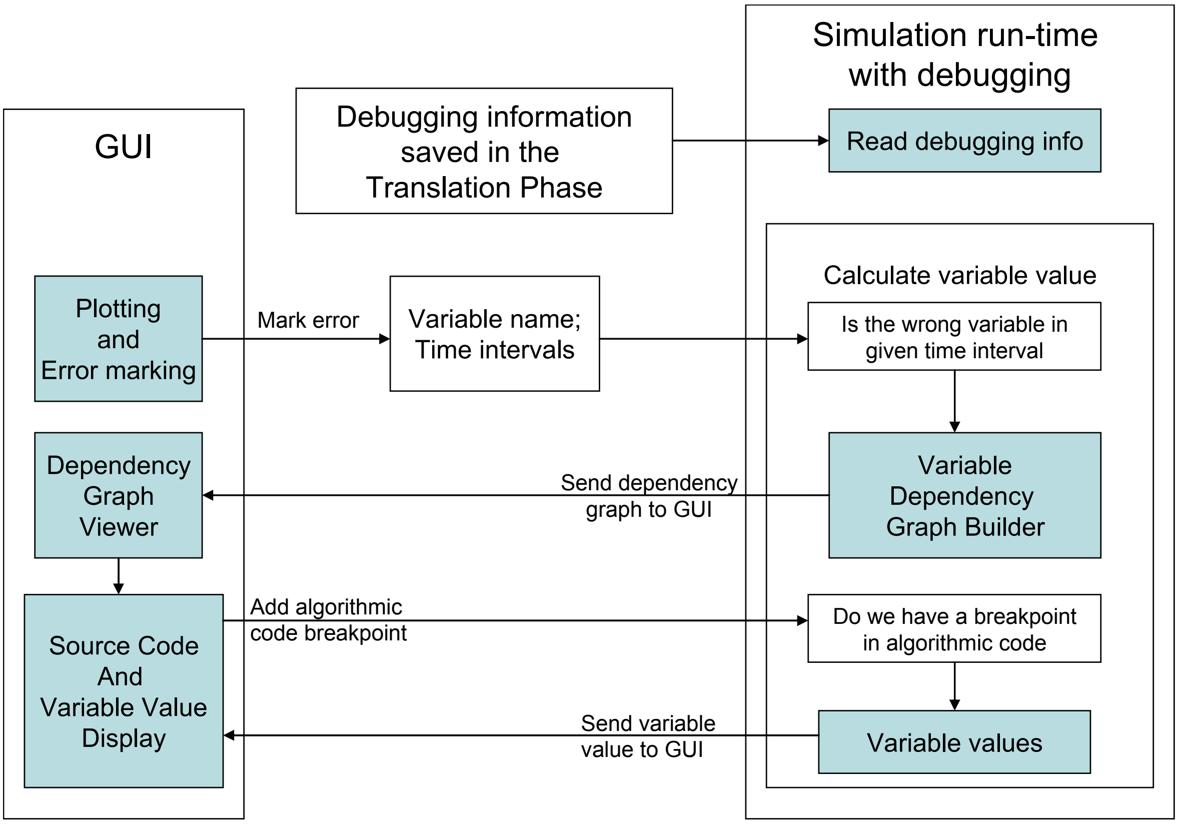

Debugging Translation Process Additional Steps
Normal Translation Process
Save element position
Save element origin (model and position)
Save equation elements origin (model and position)
Save the optimizer transformations changes
Save all the available origin information
Executable with all the available origin information
Simulation with run-time debugging functionality
Modelica Source Code
Modelica model
Translator
Flat Model
Analyzer
Sorted equations
Optimizer
Optimized sorted equations
Code Generator
C Code
C Compiler
Executable
Simulation

Figure 4. Translation stages from Modelica code to executing simulation with additional debugging steps.



GUI
Plotting and Error marking
Dependency Graph Viewer
Source Code And Variable Value Display
Debugging information saved in the Translation Phase
Mark error
Variable name; Time intervals
Send dependency graph to GUI
Add algorithmic code breakpoint
Send variable value to GUI
Simulation run-time with debugging
Read debugging info
Calculate variable value
Is the wrong variable in given time interval
Variable Dependency Graph Builder
Do we have a breakpoint in algorithmic code
Variable values

Figure 5. Run-time debugging framework overview.

The additional origin information needed by the debugging framework is saved by the debugging translation process within a file: `debug-info.xml`. The debug file is read by the simulation run-time only when needed.

If an error appears in the simulation results, the user can mark the variable with the wrong value and the error time interval(s) on the simulation plot. The simulation with run-time debugging functionality is then invoked with the error information.

4.2 Debugging Framework Overview

The run-time debugging framework overview is presented in Figure 5. The figure presents the interaction between the components of the graphical user interface (GUI) and the components of the simulation run-time with debugging. Typically, the user debugging starts at the end of a simulation when the user observes the erroneous behavior of a plotted variable value. The user marks the variable name and the time interval and invokes the debugging functionality. The simulation run-time with debugging is then invoked with the user selection as input.

In the next section we detail the debugging framework components.

4.3 Debugging Framework Components

The debugging framework has several components which deal with the user interaction (GUI part) and the handling of the debugging information (simulation run-time part). The information saved during the translation process also plays an important role in the debugging framework.

4.3.1 Plotting and Error Marking

This GUI component shows the values of a variable during simulation time. The component has special functionality which helps the user to mark an error on the plot using the mouse. The user markings are encoded as a variable name and time intervals. After marking the error, the user invokes the debugging functionality with this marking.

4.3.2 Dependency Graph Viewer

The dependency graph viewer is a GUI component that displays an interactive graph. The graph is given by the dependency graph builder component. The graph shows the calculated variable name and value, the equation in which this value was calculated and all the additional data (parameters, equation blocks, etc) which was used to calculate this value.

In this implementation the user has limited graph interaction possibilities. When the user double clicks on a graph node or edge, the origin of the selected element (variable, equation or parameter) is computed from the debugging information and the Source Code and Variable Value Display component is shown presenting the original source code element.

4.3.3 Source Code and Variable Value Display

The source code display is handled by this component. Also, the user can set breakpoints on the algorithmic code within this view. If the runtime reaches a breakpoint, the execution breaks and the variable values from this model can be examined.

4.3.4 Dependency Graph Builder

The most complex component of the debugging framework is the dependency graph builder. This component starts from a variable name and builds the dependency graph for that variable based on the debugging information saved in the translation phase.

The constructed graph is based on the Block Lower Triangular Dependency Graph (BLTDG) which is computed from the Block Lower Triangular form by considering the data dependencies. The calculation of the BLTDG is presented in detail in [2]. The constructed graph contains also additional information regarding the origin of each involved element.

4.4 Implementation status

Currently we are working at the integration of the debugging framework components.

The debugging framework is developed in Eclipse as a set of plugins that integrate our existing OpenModelica Modelica Development Tooling (MDT) [6] (for code browsing and algorithmic code debugging) with graph visualization and interaction libraries. The OpenModelica Compiler [3] has been adapted to produce the additional debugging information, the dependency graph and the simulation results.

5. Conclusions and Future Work

In this paper we present an integrated run-time debugging framework for EOO languages based on graph visualization and interaction.

We argue that such debugging framework will ease both the run-time debugging and the understanding of EOO languages.

We are aware that the scalability of our method might be an issue and we plan to research different filtering techniques for pruning the dependency graph.

Our short term goal is to finalize the prototype implementation of the proposed debugging framework, evaluate it and report experience on debugging a set of selected models, and release it as part of the OpenModelica Development Environment.

6. Acknowledgements

This work was supported by Vinnova in the Safe & Secure Modeling and Simulation project, by SSF in the RISE project and by the CUGS and Proviking graduate schools at Linköping University.

References

[1] Bunus Peter and Peter Frizson. (2003). Semi-Automatic Fault Localization and Behavior Verification for Physical System Simulation Models. In Proceedings of the 18th IEEE International Conference on Automated Software Engineering. Montreal, Canada, October 6-10, 2003)

[2] Bunus Peter. Debugging Techniques for Equation-Based Languages. PhD Thesis. Department of Computer and Information Science, Linköping University, 2004.

[3] Peter Fritzson, Peter Aronsson, Håkan Lundvall, Kaj Nyström, Adrian Pop, Levon Saldamli, and David Broman. *The OpenModelica Modeling, Simulation, and Software Development Environment*. In Simulation News Europe, 44/45, Dec 2005. http://ww.ida.liu.se/projects/OpenModelica.

[4] Peter Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*, 940 pp., ISBN 0-471-471631, Wiley-IEEE Press, 2004. See also: http://www.mathcore.com/drmodelica/

[5] The Modelica Association. The Modelica Language Specification Version 2.2. http://www.modelica.org

[6] Adrian Pop, Peter Fritzson, Andreas Remar, Elmir Jagudin, and David Akhvlediani. *OpenModelica Development Environment with Eclipse Integration for Browsing, Modeling, and Debugging*. In Proc of the Modelica'2006, Vienna, Austria, Sept. 4-5, 2006.

[7] MathCore,*MathModelica* http://www.mathcore.com

[8] Dynasim AB, *Dymola*, http://www.dynasim.com

Dynamic Model of a Bubbling Fluidized Bed Boiler

Tuomas Kataja,
Tampere University of Technology
Institute of Automation and Control
P.O. Box 692, FI-33101 TAMPERE
tuomas.kataja@tut.fi

Yrjö Majanne
Tampere University of Technology
Institute of Automation and Control
P.O. Box 692, FI-33101 TAMPERE
yrjo.majanne@tut.fi

Abstract

A dynamic model for a high-volatile solid fuel fired bubbling fluidized bed boiler is presented. The model consist of an air-flue gas model which includes a furnace model describing combustion in a bubbling fluidized bed and a model for a water-steam circuit describing heat transfer from hot flue gases to water and steam. The versatile furnace model takes account of quality parameters of fuel so that the effects of moisture, particle size, heat value, and the amount of volatiles can be simulated. The model is based on the first principles mass, energy, and momentum balances. Results from validation of the model against a bubbling fluidized bed boiler process data are presented. The validation showed that the model can describe the dynamics and static gains of the process very well.

Keywords: Dynamic boiler model, bubbling fluidized bed combustion.

1. Introduction

In the modern power generation the dynamic performance of power plants plays a very important role. This is because of the stringent demands of productivity and deregulation of energy markets. E.g. the increasing number of uncontrolled wind turbines connected to the power system disturbs the power balance which must be compensated by other controlled power plants. Also in process industry unit sizes of steam consuming processes, like paper machines have been increased causing bigger load disturbances for industrial power plants.

The most typical way to get information about the dynamic properties of the boiler is to perform test runs to determine e.g. thermal inertia, storage capacity, and load change rate. However, it is quite common that it is not possible to perform all the useful test runs because of the economic, productive, or safety reasons.

The other way to study the dynamic properties of the boilers is simulation. Mathematical modeling of boilers has been interested researchers already for decades [Maffezzoni, 1992]. Some of the pioneering works in this field are the works of Chien [Chien et al, 1958] and Profos [Profos, 1962]. The reported boiler models can be divided into two categories; fairly complicated models and simpler models derived for some limited purposes. The complicated models typically include dozens of nonlinear dynamic equations, static equations, variables and parameters. Some examples of those complicated models can be found e.g. in [Cori and Busi, 1977] and [McDonald and Kwatny, 1970]. In the simpler models many dynamic equations are neglected and many variables are not included. These models are typically applied to control purposes (design and implementation of model based control methods). Examples of this type of models are e.g. [Åström and Bell, 1998, 2000], [Cheres, 1990] and [de Mello, 1991]. Most of these models are focused on the modeling of the water-steam cycle of the boiler, heat transfer from the furnace and the flue gas duct to the water-steam circuit, thermo hydraulics, flow dynamics, etc.

In the boiler models the phenomena in the furnace have been typically left for the minor consideration. The heat power released in the furnace is often handled as a simple first order transfer function from the fuel power demand. With this kind of approach it is not possible to take account the effect of fuel quality parameters to the operation of the boiler. However, especially with biomass fired boilers the quality of fuel may vary remarkably influencing on the performance and usability of the boiler. There is a lot of research work going on about the modeling of combustion of different type of fuels in different type of combustors, e.g. [Scala, 2002] and [Galgano, 2005]. However, the combustion models developed there are seldom connected with the dynamic boiler models. Some reasons for this may be that the combustion models require heavy computing and also it is the different research groups working with combustion models and boiler models and there is a lack of information transfer between these research areas.

The bubbling fluidized bed boiler model presented in this paper is developed in a research project with a boiler manufacturer Metso Power Inc. A moderately complicated combustion model is connected with the water-steam cycle model resulting a dynamic boiler model taking account also physical phenomena existing in the bubbling fluidized bed combustion. The model structure is described and some simulation results and future plans are depicted.

2. Boiler Model

A dynamic model for the high-volatile solid fuel fired bubbling fluidized bed boiler consists of air-flue gas cycle (air preheaters, furnace, and heat exchangers) and water-steam cycle (water preheaters, drum, evaporator, super heaters, and steam attemperation). In addition steam pressure, steam temperature, and feed water control loops are included in the model. The simulator is built with MathWorks Inc's MATLAB/Simulink software. The model is based on the mass, energy, and momentum balances together with constitutional equations.

2.1 Air - Flue Gas Cycle

The furnace is divided into four segments according to the different combustion zones: the bed, the splashing region, the secondary zone, and the freeboard. The structure of the furnace model is shown in fig. 1. The fluidized bed is modeled according to the two-phase theory. In the theory the flow rate through the emulsion phase is equal to the flow rate for the minimum fluidization. Any flow in excess of that required for minimum fluidization appears as bubbles in the separate bubble phase [Toomey and Johnstone, 1952], [Oka, 2004], [Yang, 2003]. The bed is considered isothermal where two phases have uniform temperature. Minimum fluidization velocity is adopted from [Wen and Yu, 1966]. An average bubble size is assumed throughout the bed and calculated according to [Darton *et al.,*1977].

It is assumed that fuel is fed above the bed and it will spread uniformly and immediately across the cross-section of the bed. In the bed section, according to the two-phase theory, combustion takes place in the emulsion phase. Coarse char and fine char particles are burnt there. In the model the coarse char particles can also burst to the secondary zone of the furnace and drop back into the bed. The fine particles can elutriate out from the furnace [Galgano, 2005] and [Scala, 2002]. The elutriation constant for fine char particles is calculated according to [Tasirin and Geldart, 1998].

The rate of combustion is calculated by taking into account diffusive resistance to coarse char particles and kinetic resistance to elutriable size fine particles. Combustion rates depend also on the density and the diameter of the particle and oxygen concentration in the emulsion phase [Borman, 1998]. The fragmentation of fuel particles accelerates the combustion [Raiko, 2005]. The primary fragmentation occurs immediately when the particles arrive at the bed. Attrition and fragmentation by percolation increase the amount of the elutriable size fine char particles. For high volatile biomass, the generation of the fines is proportional to the coarse char combustion rate at the particle surface [Salatino, 1998]. For biomass the average diameter of the fine particles is assumed to be constant, 100 μm [Galgano, 2005]. Devolatilization occurs in the bed section and volatile components burn in the upper zone of the furnace. Devolatilization yields carbon monoxide, carbon dioxide, and hydrogen. The rate of devolatilization is modeled by a simple correlation considering diameter of the fuel particle and temperature of the fluidized bed [de Diego, 2003], [Scala, 2002].

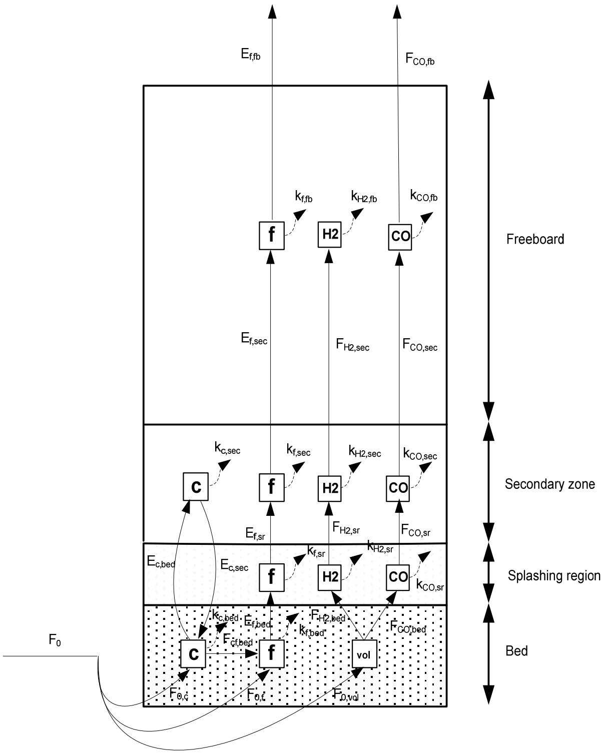


Figure 1. Structure of the furnace model. F_0 = fuel feed, **C** = coarse char balance equations, **f** = fine char particles balance equations, **vol** = volatile matter balance equations, **H2** = hydrogen balance equations, **CO** = carbon monoxide balance equations, E = entrainment of char coal from one zone to another zone, F = mass flow between model zones, k = burning rate of the material.

Heat transfer into the walls in the bed section is efficient. The heat transfer coefficient is modeled using the packet-renewal model [Mickey and Fairbanks, 1955], [Basu, 2006]. In the model the packets stay in contact with the wall surface short time and then sweep back to the bed. While the packet is in contact with the wall, the unsteady state heat transfer takes place between the packet and the surface.

Inert sand particles are ejected from the surface of the bed into the splashing region. The mass flow is calculated according to [Pemberton and Davidson, 1986]. Gas flows leaving the bubble and emulsion phases are assumed to be instantaneously mixed at a beginning of the splashing region [Okasha, 2007]. A clustering of bed particles is neglected [Benoni, 1994]. The upper edge of the splashing region is assumed to be at the same level with a refractory wall of the furnace and the lower edge depends on the bed height. In the splashing region, all available oxygen reacts with carbon monoxide and hydrogen. Combustion of elutriated fines is assumed to be neglected. Heat transfer is modeled by the packet-renewal model.

Secondary air is brought to the secondary zone where the combustion of elutriated fines and volatiles takes place. Tertiary air is brought to the freeboard where some of the elutriated fines and volatiles are burned. Part of the elutriated fines and carbon monoxide can exit the furnace but hydrogen will burn out. In the secondary zone and the freeboard sections heat is transferred to the walls by radiation.

The furnace model is based on the time-depend ordinary differential mass and energy balance equations. The mass balances of fine char particles and oxygen are written in all furnace zones. It is assumed that average velocity of the fine particles equals to gaseous components. In addition in the bed section the mass balances are written for coarse char particles, volatile matter, and bursting sand particles. Energy balances are written for the hot coarse char particles and the bed including sand particles, fine char and gaseous components. It is assumed that the fine char particles are in the same temperature with the bed. The mass balance of the coarse char particles is also defined in the secondary zone.

Table 1. *Mass and energy balances of the bed and the splashing region*

Balances of the bed:

Mass balance for the fine char particles:

$$\frac{d}{dt}W_{f,bed} = F_{0,f} + G - k_{f,bed}W_{f,bed} - k_{el} \tag{1}$$

Mass balance for the coarse char particles:

$$\begin{aligned}&\frac{d}{dt}W_{c,bed} = F_{0,c} + E_c^{in} - G - \\ &k_{c,bed}W_{c,bed} - E_c^{out}\end{aligned} \tag{2}$$

Mass balance for the volatile matter:

$$\frac{d}{dt}W_{vol,bed} = F_{0,vol} - k_d W_{vol,bed} \tag{3}$$

Mass balance for the oxygen in the emulsion phase:

$$\begin{aligned}&Ah_{mf}\varepsilon_{mf}\frac{d}{dt}C_{O2}^{bed} = Au_{mf}C_{O2}^{in} + \delta_B AK_{be}\Omega - \\ &\frac{1}{M_C}\left(k_f W_{f,bed} + k_c W_{c,bed}\right) - Au_{mf}C_{O2}^{bed}\end{aligned} \tag{4}$$

Mass balance for the sand:

$$\frac{d}{dt}W_{s,bed} = E_{s,sr} - E_{s,bed} \tag{5}$$

Energy balance for the coarce char particles:

$$\begin{aligned}&c_C\frac{d}{dt}\left[\left(W_{c,bed} + W_{vol,bed}\right)\left(T_c - T_{ref}\right)\right] = F_{0,c}c_c\left(T_0 - T_{ref}\right) - Gc_c\left(T_c - T_{ref}\right) + \\ &k_{c,bed}\frac{W_{c,bed}}{M_C}M_{O2}c_{O2}\left(T_{bed} - T_{ref}\right) - k_{c,bed}\frac{W_{c,bed}}{M_C}M_{CO2}c_{CO2}\left(T_c - T_{ref}\right) - \\ &k_d W_{vol,bed}\Delta H_{vol} + H_v F_0\left(T_0 - T_{vap}\right) - H_v F_0 l + k_{c,bed}W_{c,bed}\Delta H_C - \\ &\frac{6\left(W_{c,bed} + W_{vol,bed}\right)}{\rho_c d_c}\left[\alpha_{c,bed}\left(T_c - T_{bed}\right) + \sigma\varepsilon_{eff}\left(T_c^4 - T_{bed}^4\right)\right]\end{aligned} \tag{6}$$

Energy balance for the bed sand and fine char particles:

$$\begin{aligned}&\frac{d}{dt}\left[\left(W_{f,bed}c_f + W_s c_s\right)\left(T_{bed} - T_{ref}\right)\right] = F_{0,f}c_f\left(T_0 - T_{ref}\right) + \\ &Gc_c\left(T_0 - T_{ref}\right) - k_{el}c_f\left(T_{bed} - T_{ref}\right) + w_{g,ap}c_g\left(T_0 - T_{bed}\right) + \\ &k_c\frac{W_c}{M_C}M_{CO2}c_{CO2}\left(T_{bed} - T_{ref}\right) + \frac{k_f}{M_C}M_{CO2}c_{CO2}\left(T_{bed} - T_{ref}\right) - \\ &\frac{k_f}{M_C}M_{O2}c_{O2}\left(T_{bed} - T_{ref}\right) + E_{s,sr}c_s\left(T_s - T_{bed}\right) + \\ &k_d W_{vol,bed}c_{vol}\left(T_{vol} - T_{bed}\right) + \\ &H_v F_0 c_{H2O}\left(T_{vap} - T_{bed}\right) + k_f W_{f,bed}\Delta H_f - Q_{bed} + \\ &\frac{6\left(W_{c,bed} + W_{vol,bed}\right)}{\rho_c d_c}\left[\alpha_{c,bed}\left(T_c - T_{bed}\right) + \sigma\varepsilon_{eff}\left(T_c^4 - T_{bed}^4\right)\right] + \\ &\sigma A\phi_{bedsr}\left(T_{sr}^4 - T_{bed}^4\right)\end{aligned} \tag{7}$$

Balances of the splashing region:

Mass balance of the fine char particles:

$$\frac{d}{dt}W_{f,sr} = k_{el} - k_{f,sr}W_{f,sr} - \frac{u_{sr}}{h_{sr}}W_{f,sr} \tag{8}$$

Mass balance of the hydrogen:

$$Ah_{sr}\frac{d}{dt}C_{H2}^{sr} = k_d W_{V,bed}\frac{x_{H2}}{M_{H2}} - Ah_{sr}k_{H2,sr} - Au_{sr}C_{H2}^{sr} \tag{9}$$

Mass balance of the carbon monoxide:

$$Ah_{sr}\frac{d}{dt}C_{CO}^{sr} = k_d W_{V,bed}\frac{x_{CO}}{M_{CO}} - Ah_{sr}k_{CO,sr} - Au_{sr}C_{CO}^{sr} \tag{10}$$

Mass balance of the oxygen:

$$Ah_{sr}\frac{d}{dt}C_{O2}^{sr} = N_{O2}^{in} - \frac{k_f W_{f,sr}}{M_C} - \frac{Ah_{sr}}{2}\left(k_{CO,sr} + k_{H2,sr}\right) - Au_{sr}C_{O2}^{sr} \tag{11}$$

Mass balance of the water vapor:

$$Ah_{sr}\frac{d}{dt}C_{H2O}^{sr} = N_{H2O}^{in} + Ah_{sr}k_{H2,sr} - Au_{sr}C_{H2O}^{sr} \tag{12}$$

Mass balance of the sand:

$$\frac{d}{dt}W_{s,sr} = E_{s,bed} - \frac{E_{s,sr}}{\tau_s} \tag{13}$$

Energy balance of the sand:

$$W_{s,sr}c_s\frac{d}{dt}T_s = \chi\alpha_{c,sr}\tau_s\left(T_{sr} - T_s\right) + W_{s,sr}c_s\left(T_{bed} - T_s\right) \tag{14}$$

Energy balance of the splashing region:

$$\begin{aligned}&\frac{d}{dt}\left[\left(W_{H2}c_{H2} + W_{CO}c_{CO} + W_f c_f + W_{s,sr}c_s\right)\left(T_{sr} - T_{ref}\right)\right] = w_{g,bed}c_{g,bed}\left(T_{bed} - T_{sr}\right) + \\ &k_{el}W_{f,bed}c_f\left(T_{bed} - T_{ref}\right) - u_{sp}\frac{W_{f,sr}}{h_{sr}}c_f\left(T_{sr} - T_{ref}\right) + \frac{k_f}{M_C}M_{CO2}c_{CO2}\left(T_{sr} - T_{ref}\right) - \\ &\frac{k_f}{M_C}M_{O2}c_{O2}\left(T_{sr} - T_{ref}\right) + k_{CO,sr}Ah_{sr}M_{CO2}c_{CO2}\left(T_{sr} - T_{ref}\right) + \\ &k_{H2,sr}Ah_{sr}M_{H2O}c_{H2O}\left(T_{sr} - T_{ref}\right) - k_{CO,sr}Ah_{sr}M_{CO}c_{CO}\left(T_{sr} - T_{ref}\right) - \\ &k_{H2,sr}Ah_{sr}M_{H2}c_{H2}\left(T_{sr} - T_{ref}\right) - \frac{1}{2}\left(k_{H2,sr} + k_{CO,sr}\right)Ah_{sr}M_{O2}c_{O2}\left(T_{sr} - T_{ref}\right) + \\ &\chi\alpha_{c,sr}\left(T_s - T_{sr}\right) + k_f W_{f,sr}\Delta H_f + \left(k_{CO,sr}\Delta H_{CO} + k_{H2,sr}\Delta H_{H2}\right)Ah_{sr} - \\ &Q_{sr} + \sigma A\left[\phi_{bedsr}\left(T_{bed}^4 - T_{sr}^4\right) + \phi_{bedsec}\left(T_{sec}^4 - T_{sr}^4\right)\right]\end{aligned} \tag{15}$$

In the splashing region the mass and energy balances of the sand particles are defined. Hydrogen, carbon monoxide, and the water vapor mass balances and fine char and gas energy balances are written for the splashing region, the secondary zone, and the freeboard. In table 1 is written mass and energy balance equations for the bed and splashing region sections. Similar are written for the secondary zone and the freeboard sections.

In the heat exchange section of the air – flue gas cycle the heat transfer into the super heaters, feed water preheaters, and air preheaters is calculated. Primary, secondary, and tertiary airs are heated by the air preheaters. In the air preheater model the mass and energy balances of the air and the energy balance of the preheater metal mass are calculated. A schematic diagram of the air-flue gas model is shown in fig. 2.

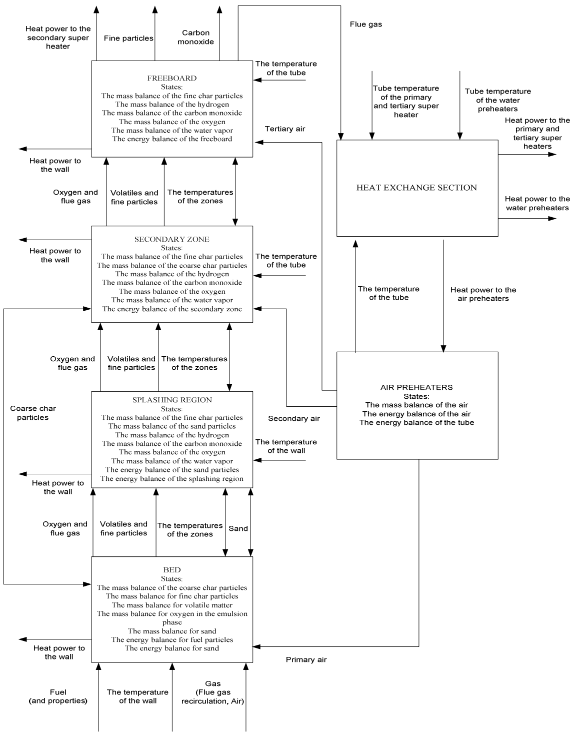

Heat power to the secondary super heater
Fine particles
Carbon monoxide
Flue gas
The temperature of the tube
FREEBOARD
States:
The mass balance of the fine char particles
The mass balance of the hydrogen
The mass balance of the carbon monoxide
The mass balance of the oxygen
The mass balance of the water vapor
The energy balance of the freeboard
Tertiary air
Tube temperature of the primary and tertiary super heater
Tube temperature of the water preheaters
Heat power to the primary and tertiary super heaters
HEAT EXCHANGE SECTION
Heat power to the water preheaters
Heat power to the wall
Oxygen and flue gas
Volatiles and fine particles
The temperatures of the zones
SECONDARY ZONE
States:
The mass balance of the fine char particles
The mass balance of the coarse char particles
The mass balance of the hydrogen
The mass balance of the carbon monoxide
The mass balance of the oxygen
The mass balance of the water vapor
The energy balance of the secondary zone
The temperature of the tube
Heat power to the wall
The temperature of the tube
Heat power to the air preheaters
Oxygen and flue gas
Volatiles and fine particles
The temperatures of the zones
Coarse char particles
SPLASHING REGION
States:
The mass balance of the fine char particles
The mass balance of the sand particles
The mass balance of the hydrogen
The mass balance of the carbon monoxide
The mass balance of the oxygen
The mass balance of the water vapor
The energy balance of the sand particles
The energy balance of the splashing region
Secondary air
The temperature of the wall
AIR PREHEATERS
States:
The mass balance of the air
The energy balance of the air
The energy balance of the tube
Heat power to the wall
Oxygen and flue gas
Volatiles and fine particles
The temperatures of the zones
Sand
BED
States:
The mass balance of the coarse char particles
The mass balance for fine char particles
The mass balance for volatile matter
The mass balance for oxygen in the emulsion phase
The mass balance for sand
The energy balance for fuel particles
The energy balance for sand
Heat power to the wall
Primary air
Fuel (and properties)
The temperature of the wall
Gas (Flue gas recirculation, Air)

Figure 2. A schematic diagram of the air-flue gas model.

2.2 Water - Steam Cycle

In the evaporation circuit steam is generated inside the vertical tubes located on the furnace walls. Water-steam mixture flows up in a two-phase flow to the drum for phase separation [Adam & Marchetti, 1999].

In the model the evaporation circuit is divided into sixteen segments. Each wall is modeled separately and every wall is divided into four segments according to the furnace segments heights. It is assumed that the water-steam mixture is saturated. The energy balances are written for the water-steam mixture and for the furnace wall tubes. Energy balances for the refractory elements are also written for the two lowest segments of the evaporation circuit. The steam quality and the density of the water-steam mixture are calculated in every segment. Heat flux from the tube wall to the water-steam mixture is calculated by using an experimental equation [Ordys 1994].

Usually the heat flow from the furnace to the evaporator is assumed to be uniform [Åström 2000], [Kim 2005], but in this model the heat flow is divided into each evaporator segment (16 pc.) according to the temperature of the furnace and the heat transfer coefficient in each zone.

Water-steam mixture in the drum is divided into water and steam sub-volumes interacting with each other. The mass balance equation is applied to both sub-volumes and the energy balance is applied to water sub-volume [Kim, 2005], [Ordys, 1994], [Åström, 2000]. From the drum steam is directed to the super heaters. Saturated water in the drum is mixed with feed water coming from the water preheaters. The incoming feed water is directed to the downcomers. In the naturally circulating drum type boilers density difference between the water in the downcomers and the two-phase mixture in the evaporation tubes is the driving force of the natural circulation.

The modeled boiler is equipped with three super heaters and two feed water preheaters. The super heaters are used to raise the live steam temperature. The temperature is controlled by steam attemperating sprays. The feed water preheaters are used to transfer low grade flue gas heat to feed water. The heat exchanger models are based on the mass, energy, and momentum balances. Heat is transferred into exchangers from the heat exchange section of the air-flue gas model. A schematic diagram of the water-steam model is shown in fig 3 and main mass and energy balance equations of the water - steam cycle is written in table 2.

Table 2. *Mass and energy balances of the water – steam cycle.*

<u>*Balances of the evaporation circuit section:*</u>

Energy balance of the refractory wall in the bed section:

$$m_{rf} c_{rf} \frac{d}{dt}(T_{rf}) = Q_{rf} - Q_{cond} \tag{16}$$

Energy balance of the evaporation circuit in the one refractoty area section:

$$V_{rf} \frac{d}{dt}(\rho_{wv} h_{wv,out}) = Q_{cond} + w_{wv,in} h_{wv,in} - w_{wv,out} h_{wv,out} \tag{17}$$

Energy balance of the evaporation circuit in the one tube area section:

$$V_t \frac{d}{dt}(\rho_{wv} h_{wv,out}) = Q_{in,ev} + w_{wv,in} h_{wv,in} - w_{wv,out} h_{wv,out} \tag{18}$$

Balances of the drum section:

Mass balance of the drum water:

$$\frac{d}{dt}(m_{dr}) = w_e + (1-x)w_{ev} - w_{dc} - w_{ec} \tag{19}$$

Energy balance of the drum water:

$$\frac{d}{dt}(m_{dr}h_w) = w_e h_e + (1-x)w_{ev}h_w - w_{dc}h_w - w_{ec}h_v \tag{20}$$

Mass balance of the drum steam:

$$\frac{d}{dt}(V_v \rho_v) = xw_{ev} - w_v \ - w_{ec} \tag{22}$$

Balances of the super heater section:

Energy balance of the steam in the one superheater section:

$$V_{sh}\frac{d}{dt}(\rho_v h_{v,out}) = Q_{in,sh} + w_{v,in}h_{v,in} - w_{v,out}h_{v,out} \tag{23}$$

Energy balance of the tube in the one superheater section:

$$m_{sh}c_{sh}\frac{d}{dt}(T_{sh}) = Q_{sh} - Q_{in,sh} \tag{24}$$

Mass balance of the steam in the one superheater section :

$$\frac{d}{dt}(w_{v,in}) = \frac{w_{v,out} - w_{v,in}}{\tau_{sh}} \tag{25}$$

Balances of the feed water preheater section:

Energy balance of the feed water in the one feed water preheater section:

$$V_e\frac{d}{dt}(\rho_e h_{e,out}) = Q_{in,e} + w_{e,in}h_{e,in} - w_{e,out}h_{e,out} \tag{26}$$

Energy balance of the tube in the one feed water preheater section:

$$m_e c_e \frac{d}{dt}(T_e) = Q_e - Q_{in,e} \tag{27}$$

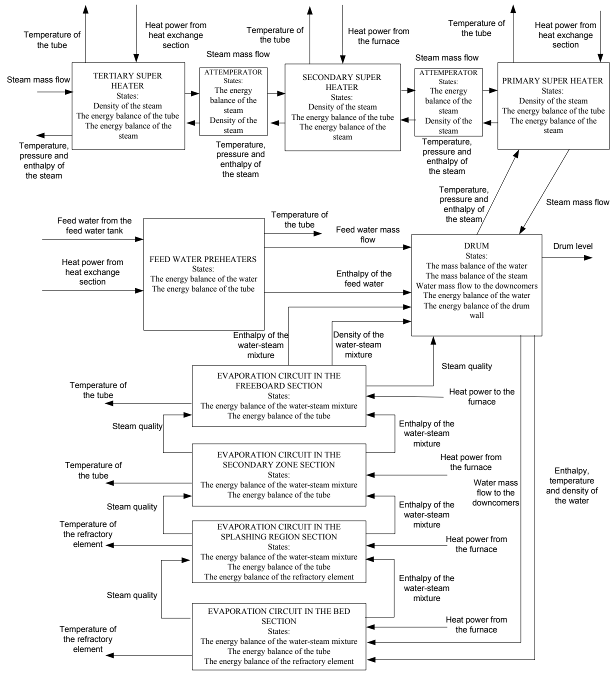

Temperature of the tube
Heat power from heat exchange section
Steam mass flow
Temperature of the tube
Heat power from the furnace
Temperature of the tube
Steam mass flow
Heat power from heat exchange section
Steam mass flow
TERTIARY SUPER HEATER
States:
Density of the steam
The energy balance of the tube
The energy balance of the steam
ATTEMPERATOR
States:
The energy balance of the steam
Density of the steam
SECONDARY SUPER HEATER
States:
Density of the steam
The energy balance of the tube
The energy balance of the steam
ATTEMPERATOR
States:
The energy balance of the steam
Density of the steam
PRIMARY SUPER HEATER
States:
Density of the steam
The energy balance of the tube
The energy balance of the steam
Temperature, pressure and enthalpy of the steam
Temperature, pressure and enthalpy of the steam
Temperature, pressure and enthalpy of the steam
Temperature, pressure and enthalpy of the steam
Steam mass flow
Feed water from the feed water tank
Heat power from heat exchange section
FEED WATER PREHEATERS
States:
The energy balance of the water
The energy balance of the tube
Temperature of the tube
Feed water mass flow
Enthalpy of the feed water
DRUM
States:
The mass balance of the water
The mass balance of the steam
Water mass flow to the downcomers
The energy balance of the water
The energy balance of the drum wall
Drum level
Enthalpy of the water-steam mixture
Density of the water-steam mixture
Steam quality
Temperature of the tube
EVAPORATION CIRCUIT IN THE FREEBOARD SECTION
States:
The energy balance of the water-steam mixture
The energy balance of the tube
Heat power to the furnace
Steam quality
Enthalpy of the water-steam mixture
Temperature of the tube
EVAPORATION CIRCUIT IN THE SECONDARY ZONE SECTION
States:
The energy balance of the water-steam mixture
The energy balance of the tube
Heat power from the furnace
Water mass flow to the downcomers
Enthalpy, temperature and density of the water
Steam quality
Enthalpy of the water-steam mixture
Temperature of the refractory element
EVAPORATION CIRCUIT IN THE SPLASHING REGION SECTION
States:
The energy balance of the water-steam mixture
The energy balance of the tube
The energy balance of the refractory element
Heat power from the furnace
Steam quality
Enthalpy of the water-steam mixture
Temperature of the refractory element
EVAPORATION CIRCUIT IN THE BED SECTION
States:
The energy balance of the water-steam mixture
The energy balance of the tube
The energy balance of the refractory element
Heat power from the furnace

Figure 3. A schematic diagram of the water-steam model.

3. Comparisons with plant data and simulation tests

The model is parameterized using a construction data and measured process data from a 260 MW$_{\text{th}}$ bubbling fluidized bed boiler. The validation of the model showed that the model can describe the dynamics and the static gains of the process very well. The model is tested using measured process data as an input to the model and then compared the computed signals with the measured outputs. Figure 4 depicts the measured and simulated results of the drum pressure and the live steam pressure.

Figure 5 depicts simulated and measured fuel mass flows. In the model milled peat and bark are used as a fuel. Measured fuel mass flow is estimation from real fuel mass flow. Also the heat value and moisture of the fuel cannot be measured. Still results are quite close to each other.

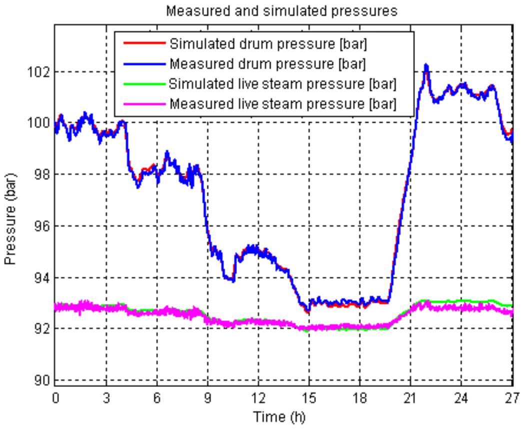


Figure 4. Measured and simulated drum and live steam pressures.

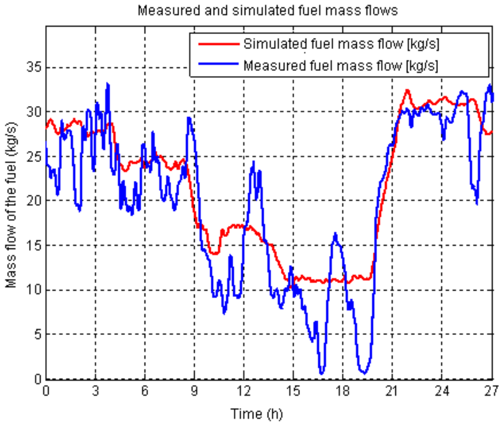


Figure 5. Measured and simulated fuel mass flows.

The model is used to study the dynamic behavior of the boiler process. The points of interest are e.g. the thermal inertia and the steam storage capacity of the boiler. Knowledge about the behavior of these properties gives useful information for the process and control design. The dynamic model helps process developers to answer questions like how long a boiler can generate steam in case of a boiler trip or what will happen in the process during a black out situation?

Figure 6 depicts the boiler dynamic response when the fuel mass flow rate is increased 10 %. From the simulation results e.g. the time delay and the time constant of the boiler can be solved. In the test control loops are switched off.

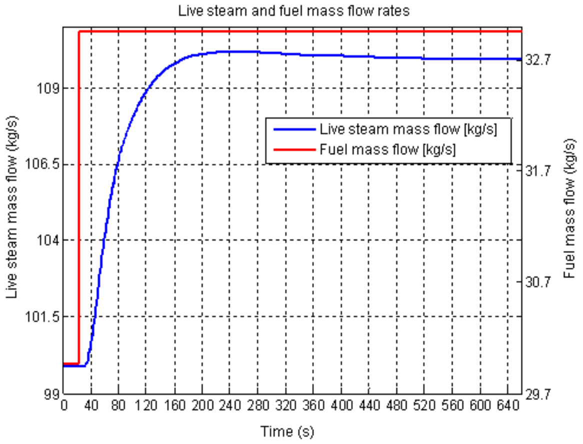


Figure 6. The response of the boiler to the 10 % fuel mass flow rate step change.

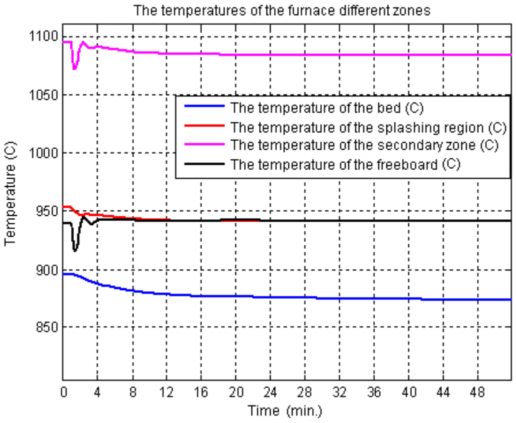


Figure 7. The step responses of the furnace temperatures in different zones when increasing the moisture of the fuel 10%.

In fig. 7 the dynamic behavior of the different zones of the furnace is studied when increasing the moisture content of the fuel 10 %. The control loops are switched on. The temperatures of the bed and the splashing region are dropped slowly due to the thermal capacity of sand.

4. Conclusions and future plans

A dynamic model of the bubbling fluidized boiler is presented. The model is based on physical parameters for the boiler and can quite easily scale to represent any drum type BFB boiler. The model consists of the air-flue gas model and the water-steam circuit model. In the air-flue gas model furnace model takes into account important processes which effects on combustion dynamics like fuel particle fragmentation, attrition and volatile matter segregation and post combustion in the upper zones. In the water-steam circuit model the dynamics of the steam generation in the evaporation circuit, separation of the steam in the drum section and super heating of the steam ar investigated. Also feed water preheater dynamics are studied.

The model is nonlinear and agrees well with experimental data. The model helps to study the dynamic behavior of the boiler process.

The future plans of the modeling project are to test the developed model by modeling three different size bubbling bed boilers and compare the simulated results with the measured data. Also the user interface of the simulator will be developed to make the build up of the simulator faster. There is also another project going on with Metso Power to develop a similar simulation model for a circulating fluidized bed boiler.

Nomenclature

A	Cross-sectional area of the furnace [m^2]
c_x	Specific heat of the component *x* [kJ/kgK]
C_x^y	Concentration of component *x* in section *y*
d_x	Average diameter of component *x* [m]
E_x^z	Entrainment rate of the component *x* to the direction *z* [kg/s]
$F_{0,x}$	Fuel feed mass flow rate of the component *x* [kg/s]
G	Fine particles generation constant [kg/s]
h_{mf}	Height of the bed in the minimum fluidization velocity [m]
H_v	Moisture content of the fuel [-]
h_x	Enthalpy of component *x* [kJ/kg]
h_y	Height of the section *y* [m]
ΔH_x	Heat of combustion of component *x* [kJ/kg] or [kJ/kmol]
K_{be}	Mass transfer coefficient between dense phase and bubbles [1/s]
k_d	Devolatilization constant [1/s]
k_{el}	Fines elutriation mass flow [kg/s]
$k_{x,y}$	Combustion rate of component *x* in section *y* [s^{-1}]
l	Latent heat [kJ/kg]
M_x	Molecular weight of component *x* [kg/kmol]
m_y	Mass of the section *y* [kg]
N_x^z	Quantity mass flow of component *x* to direction *y* [mol/s]
$Q_{in,y}$	Heat power from the tube to the fluid in the section *y* [W]
Q_y	Heat power to the wall in the section *y* [W]
T_x	Temperature of the component *x* [K]
u_{mf}	Minimum fluidization velocity [m/s]
u_y	Gas velocity in section *y* [m/s]
V_y	Volume of the section *y* [m^3]
$W_{x,y}$	Mass of component *x* in section *y* [kg]
w_y	Mass flow from section *y* [kg/s]
x	Steam quality [-]
x_x	Mass fraction of component *x* [-]
$\alpha_{x,y}$	Heat transfer coefficient of component *x* in section *y* [W/Km^2]
δ_B	Bubble fraction in the bed [-]
ε_{eff}	Effective emissivity [-]
ε_{mf}	Voidage at minimum fluidization [-]
ρ_x	Density of the component *x* [kg/m^3]
σ	Stefan-Boltzmann constant [W/m^{2}K^4]
τ_x	Residence time of component *x*
ϕ_{xy}	View factor related to region *x* and *y* [-]
χ	Total exposed surface area of ejected inert particles [m^2]
Ω	Coefficient between dense phase and bubbles [-]

Subscripts and superscripts

ap	Air preheater
bed	Bed
C	Carbon
c	Coarse char particle
dr	Drum
CO	Carbon monoxide
CO2	Carbon oxide
cond	Conduction
d	Devolatilization
dc	Downcomer
e	Feed water preheater
ec	Evaporation in the drum
ev	Evaporation circuit
f	Fine char particle
g	Gas
H2	Hydrogen
H2O	Water in the combustion

in	Inlet direction
O2	Oxygen
out	Outlet direction
ref	Reference
rf	Refractory wall
s	Sand
sh	Super heater
sec	Secondary zone
sr	Splashing region
t	Tube
v	Steam
vap	Vaporization
vol	Volatile matter
w	Water
wv	Water-steam mixture
0	Initial

References

Adam EJ & Marchetti L. (1999). Dynamic simulation of large boilers with natural recirculation. Combust Chem Eng 1999;23:1031–40.

Basu, P. (2006). Combustion and gasification in fluidized beds. Boca Raton, CRC Press. 473 p.

Benoni, D., Briens, C. L., Baron, T., Duchesne, E., & Knowlton, T. M. (1994). A procedure to determine particle agglomeration in a fluidized bed and its effect on entrainment. Powder Technology, Vol 78, pp. 33-42.

Borman, G.L. & Ragland, K,W. (1998). Combustion engineering. Singapore, McGraw-Hill Inc. 613 p.

Cheres, E. (1990). Small and medium size drum boiler models suitable for long term dynamic response. IEEE Transactions on Energy Conversion, Vol. 5, No. 4, pp. 686–692

Chirone, R., Marzocchella, A., Salatino, P., Scala, F. (1999) Fluidized bed combustion of high-volatile solid fuels: An assessment of char attrition and volatile matter segregation, Proceedings of the 15th International Conference on Fluidized Bed Combustion, Savannah, May 16–19, 1999, The American Society of Mechanical Engineers, Paper No. FBC99-0021.

Chien, K. L., Ergin, E. I., Ling, C., &Lee, A. (1958). Dynamic analysis of a boiler. Transactions of ASME, 80, 1809–1819.

Cori, R., & Busi, T. (1977). Parameter identification of a drum boiler power plant. Proc. 3rd Power Plant Dynamics, Control and Testing Symposium, Knoxwille, Tennessee, September 7-9, 1977.

Darton, R.C., La Nauze, J-F., Davidson, J.F. & Harrison, D. (1977). Bubble growth due to coalescence in fluidized beds. Transactions of the Institution of Chemical Engineers, Vol 55, pp. 274-280.

de Diego, L.F., Garcia-Labiano, F., Abad, A., Gayan, P. & Adanez, J. (2003). Effect of moisture content on devolatilization times of pine wood particles in a fluidized bed. Energy & Fuels, Vol. 17, No.2, pp. 285-290.

Galgano, A., Salatino, P., Crescitelli, S., Scala, F. & Maffettone, P.L. (2005). A model of the dynamics of a fluidized bed combustor burning biomass. Combustion and flame Vol.140, No.4, pp. 271–284.

Kim, H. & Choi, S. (2005). A model on water level dynamics in natural circulation drum-type boilers. Heat and mass transfer, Vol. 32, pp. 786–796.

Maffezzoni, C. (1992). Issues in modeling and simulation of power plants. In Proceedings of IFAC symposium on control of power plants and power systems, Vol. 1, pp. 19–27

McDonald, J. P., & Kwatny, H. G. (1970). A mathematical model for reheat boiler–turbine–generator systems. In Proceedings of IEEE. PES winter power meeting, New York. Paper 70 CP221-PWR.

de Mello, F. P. (1991). Boiler models for system dynamic performance studies. IEEE Transactions on Power Systems, Vol 6, No. 1, pp. 753–761

Mickey, H.S. & Fairbanks, D.F. (1955). Mechanism of heat transfer to fluidized beds. AICHE J, Vol.1, pp. 374–384.

Oka, S. (2004). Fluidized bed Combustion. New York, Marcel Dekker, Inc. 590 p.

Okasha, F. (2007). Modeling combustion of straw-bitumen pellets in a fluidized bed. Fuel Processing Technology, Vol 88, pp. 281-293.

Ordys, A.W., Pike, A.W., Johnson, M.A., Katebi, R.M. & Grimble M.J. (1994). Modelling and Simulation of Power Generation Plants. London, Springer-Verlag. 311 p.

Pemberton, S. T., & Davidson, J. F. (1986). Elutriation from fluidized beds—I. Particle ejection from the dense phase into the freeboard. Chemical Engineering Science, Vol. 41, pp. 243-251.

Profos, P. (1962). Die Regelung von Dampfanlagen. Berlin: Springer.

Raiko, R., Saastamoinen, J., Hupa, M. & Kurki-Suonio, I. Poltto ja palaminen. 2nd ed. Jyväskylä 2002, Teknillistieteelliset akatemiat. 744 p.

Salatino, P., Scala, F. & Chirone, R. Proc. of 27th Symp (Int.) on Combust., Combustion Institute, Pittsburgh (PA) in press (1998)

Scala, F. & Chirone, R. (2006). Combustion and attrition of biomass in fluidized bed. Energy and fuels, Vol. 20, pp. 91-102.

Scala, F. & Salatino, P. (2002). Modelling fluidized bed combustion of high-volatile solid fuels. Chemical engineering science, Vol. 57, pp. 1175-1196.

Scala, F., Salatino, P. & Chirone, R. (2000). Fluidized bed combustion of biomass char (robinia pseudoacacia). Energy & Fuels, Vol. 14, No.4, pp. 781-790.

Tasirin, S.M & Geldart, D. (1998). Enrainment of FCC from fluidized beds – a new correlation for the elutriation rate constants $K_{i\infty}$. Powder Technology, Vol. 95, pp. 240-247.

Toomey, R.D. & Johnstone, H.F. (1952). Gaseous fluidization of solid particles. Chem Eng Prog, Vol. 48, pp. 220–226.

Wen, C.Y. & Yu, Y.W. (1966). Mechanics of fluidization. Eng. Progr. Symp. Series 100-125.

Yang, W-C. (2003). Handbook of fluidization and fluid-particle systems. New York. Marcel Dekker, Inc. 861 p.

Åström, K. J., & Bell, R. (1988). Simple drum-boiler models. In IFAC international symposium on power systems, modelling and control applications. Brussels, Belgium.

Åström, K. J., & Bell, R. (2000). Drum boiler dynamics. Automatica, 36, pp. 363–378

Modelling Space Heating Systems Connected to District Heating in Case of Electric Power Failure

Per-Olof Johansson, M Sc, Patrick Ljunggren, M Sc, Janusz Wollerstrand, Ass. professor
Lund University, Department of Energy Sciences, Division for Efficient Energy Systems
Per-Olof.Johansson@vok.lth.se

Abstract

Recent year's extensive power failures have put focus on the importance of secure local production and distribution of energy. Since district heating (DH) is the dominating heating system in Scandinavia it is of great importance to investigate the possibility for buildings connected to DH to receive heat during an electric power failure.

By using computer models, buildings' heating systems can be simulated, and by comparing the results with field studies the model can be evaluated. The model shows good resemblance with the field study.

By using a model the influence of different parameters can be studied. The influence of, e.g., changed primary supply temperature and outdoor temperature can be studied.

Keywords: Power failure, District heating, Computer modelling, Space heating system

Nomenclature

Variables

A	Area, m^2
c_p	Specific heat capacity, J/kgK
g	Gravity force, m/s^2
h	Height, m
k_v	Flow capacity, kg/(s·Pa$^{1/2}$)
$LMTD$	Logarithmic mean temp. diff., °C
m	Mass, kg
$\dot{m}$	Mass flow, kg/s
n	Radiator exponent
p	Pressure, Pa
$\dot{Q}$	Heat flow, W
t	Time, s
T	Temperature, °C
U	Overall heat transfer coeff., W/m^{2}K
α	Heat transfer coeff., W/m^{2}K
ρ	Density, kg/m^3

Subscripts

br	Branch
eq	Equivalent
exp	Experimental
hor	Horizontal
i	Indoor
int	Internal
N	Number of branches
nat	Natural circulation
o	Outdoor
p	Primary
r	Return
rad	Radiator
s	Secondary
s	Supply
sim	Simulated
vent	Ventilation
ver	Vertical
w	Wall
∞	Stationary values (after long time)

1. Introduction

In case of a large-scale power failure many important functions in the society are interrupted. Space heating is such an important and critical function. Obviously, heating systems using electricity (either directly or indirectly via e.g. a heat pump) cannot operate. Systems using oil, natural gas or pellets will not work either since electric power is demanded for the operation of burners.

District heating (DH) is the dominating heating system in Scandinavia. It comprises central heat production and distribution by hot water circulated in a grid constituted of buried pipes. The concept allows for making use of waste heat, e.g. from industries or thermal power stations.

Buildings connected to DH are also assumed not to be functioning in case of a power failure. Even if the DH supplier has back-up power in order to maintain production and distribution of heat, the buildings are generally assumed not to be able to receive any heat when the house-internal heating system is dependent on electric power to pump and control equipment. Recent studies have, however, proved that this does not have to be the case.

Extensive power failures in recent years have produced large efforts on the possibility to produce and distribute electric power locally during a breakdown on the national power grid, so-called island operation. Many power stations can produce both heat and electric power, so-called cogeneration or CHP (combined heat and power). In this case, the DH grid functions as cooler to the power generating process. To be able to produce electricity during island operation, CHP stations still need cooling. If it is still possible to use the DH grid as a cooler, the power station could still operate. However, then the end users must be able to receive district heat during a power failure.

The society can also benefit if heat supplies is functioning to an extent as large as possible. If, e.g., evacuation of citizens could be avoided or at least delayed, this could benefit not at least elderly people and persons receiving health care.

The objective of our study is to investigate what will happen in buildings connected to DH in case of a power failure. Is it possible, if the DH network still is functioning, that natural circulation will arise in modern space heating systems, and to what extent? Natural circulation was used many years ago, before pumps were introduced in space heating systems, and is based on the difference in density between hot and cool water. I.e., hot water from the heat source will rise in the vertical supply pipes and then, when heat is emitted in the radiators, the cold water will descend back to the heat source.

Computer simulations are a useful and necessary tool and good complement to field experiments in the process to establish how different heating systems and buildings are able to receive heat.

2. Objective of the study

In the study, a number of buildings of various sizes, age and type have been examined through practical field studies. To be able to study different outer circumstances and the influence of different parameters, e.g. the outdoor temperature, the field studies have been complemented with computer simulations using MATLAB's toolbox Simulink. The simulations are a useful instrument when estimating how much district heat that could be distributed during a power failure. It is also possible to study the influence from various parameters that might be difficult to capture in reality, e.g. the possible gain from an increased DH supply temperature.

The objective of the paper is to show possibilities to simulate natural circulation in space heating systems and the dynamic influence on the indoor air temperature. The agreement between results from simulations and field experiments will be discussed.

3. Space heating systems connected to DH

Figure 1 below describes a waterborne radiator space heating system connected to a DH grid. The water from the DH grid (primary water) is led into a heat exchanger (HEX) where the radiator water (secondary water) is heated. The secondary supply temperature, T_{ss}, is regulated by a controller which in turn regulates the DH flow passing through the HEX. The set point for T_{ss} is based on the current outdoor temperature and the building's time constant (a sudden change in outdoor temperature should not momentarily be compensated for due to the thermal inertia of the building shell). The circulation in the radiator system is achieved by an electric pump. In order to receive proper indoor temperature in the whole building, valves are used to balance the flow between risers and radiators in different parts of the system.

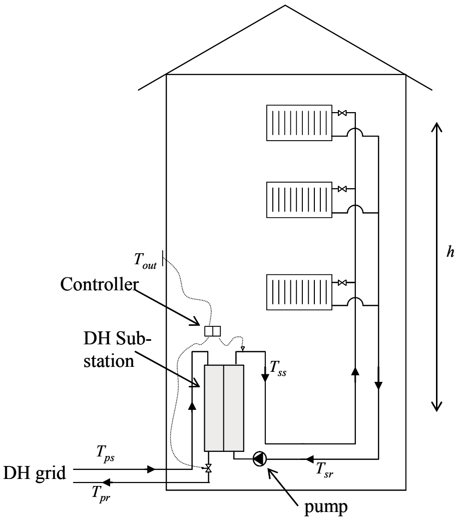


Figure 1 A building with space heating system connected to a DH grid

4. Computer modelling

The building model is based on existing models of HEX, control equipment, actuators, valves, connection pipes (taking into account pressure drop, heat loss and time delay), radiators and building. The theory and function of these components have been described in detail by a number of authors, for example [1], [2] and [4].

In this work, these components have been put together in order to constitute a complete building model with DH substation, space heating system and building shell. To be able to meet the objective of this work, the model has also been modified and improved on several points. The space heating system has been extended to comprise four risers and three storeys, making it possible to study the heat distribution in the building during natural circulation. The flow distribution with pump operation is built on a method based on an analogy to Kirchoff's circuit laws. In case of natural circulation the problem must however be attacked in a different way, which will be described in section 4.1. In order to study the dynamics of the indoor air temperature, in case of a complete or partial break in heat supply, the modelling of the building shell has been modified, which will be described in section 4.2.

The overall structure of the complete model is shown in Figure 2 below.

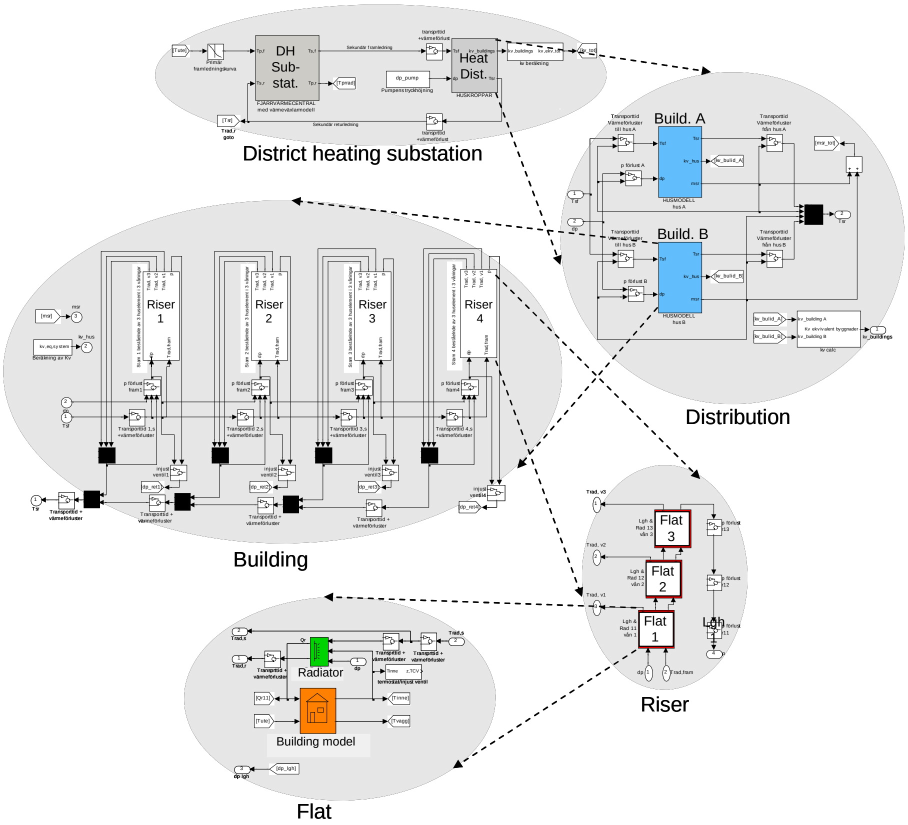


Figure 2 Overview of the computer model of the system including substation, space heating system and building

4.1 Basic parts of the model

The first step in the energy transfer from the DH grid to the building takes place in the DH substation where heat is transferred via a HEX. A control valve adjusts the DH flow in order to achieve a correct outgoing temperature on the secondary side of the HEX, i.e. the radiator supply temperature. A traditional way to handle the dynamic thermal behaviour of the HEX is to divide it into a number of sections. For each section energy balances can be stated for both primary and secondary side flows, and for the wall separating them. After differentiating the following equations are obtained:

$$\frac{\partial}{\partial t}\left(T_{s,out}\right)=\frac{1}{m_s\cdot c_{p,s}}\Big[\dot{m}_s\cdot c_{p,s}\left(T_{s,in}-T_{s,out}\right)-\alpha_s\cdot A\left(\frac{T_{s,in}+T_{s,out}}{2}-T_{wall}\right)\Big] \quad (1)$$

$$\frac{\partial}{\partial t}\left(T_{p,out}\right)=\frac{1}{m_p\cdot c_{p,p}}\Big[\dot{m}_p\cdot c_{p,p}\left(T_{p,in}-T_{p,out}\right)-\alpha_p\cdot A\left(\frac{T_{p,in}+T_{p,out}}{2}-T_{wall}\right)\Big] \quad (2)$$

$$\frac{\partial}{\partial t}\left(T_{wall}\right)=\frac{A}{m_{wall}\cdot c_{p,wall}}\left[\alpha_p\left(\frac{T_{p,in}+T_{p,out}}{2}-T_{wall}\right)\right.$$
$$\left.-\alpha_s\left(\frac{T_{s,in}+T_{s,out}}{2}-T_{wall}\right)\right] \tag{3}$$

The equations are easily implemented in Simulink. The temperature profile is assumed to be linear, while the theoretically correct profile is logarithmic .The logarithmic mean temperature difference, *LMTD* is defined as:

$$LMTD=\frac{\left(T_{p,s}-T_{s,s}\right)-\left(T_{p,r}-T_{s,r}\right)}{\ln\left(\dfrac{T_{p,s}-T_{s,s}}{T_{p,r}-T_{s,r}}\right)} \tag{4}$$

As mentioned, an arithmetic mean temperature difference has been used. The advantage with the later is that the heat transfer can be directed in both ways, i.e. from primary to secondary side and vice versa. Such situations can occur for short periods in the radiator HEX. This means that we must use a sufficient number of sections in the model. On the other hand, an increasing number of sections will increase the computational time. A suitable choice is to divide the HEX into 3-5 sections, [5]. The method involves a number of assumptions such as uniform temperature in each section, no conduction in the water flow direction or in the length direction of the HEX wall. The heat transfer coefficient, α, is approximated to be a function of the flow only, ignoring minor influence from temperature dependant quantities. The thermal resistance in the wall material can be ignored due to the thinness of the plate. All these assumptions ease the mathematical description, but investigations still have proved a good resemblance to real data. [1], [7].

The piping system model includes three features, pressure drop, heat loss and time delay. The pressure drop due to friction in pipes, valves, radiators and HEX is essential when calculating the flows in the system. The time delay could be of some interest at normal (pump) operation, while the heat loss is of minor interest. While simulating natural circulation, however, all of these features become essential. The driving force from natural circulation is rather small, leading to a substantially smaller circulation flow and, consequently, a low flow velocity. Especially when studying larger buildings, transportation times and heat losses can be considerable and essential for the operation of the system.

The pressure loss calculations will be discussed later on in connection to flow calculations.

The next level of heat transfer in the system takes place in the radiators. Basically, the heat transfer here works in the same way as in the HEX. The model can, however, be simplified on some points. On both sides of the HEX the medium is water, while in the radiator we have water on one side and air on the other. The air has a substantially lower heat transfer coefficient than water and is the absolutely dominating resistance in the radiator heat transfer. The overall heat transfer number, *U*, can therefore be assumed to be constant. We are interested in calculating the return temperature and the emitted heat from the radiator. The air temperature can be regarded as constant in the calculations. In the radiator, we can expect that the heat transfer always will be directed from the radiator water to the air. Therefore, we now can use the *LMTD* calculation (as shown in (4)) and do not have to divide the radiator into sections. We then arrive with the following two equations:

$$\frac{\partial}{\partial t}\left(T_{rad,out}\right)=\frac{1}{m_{rad}\cdot c_{p,rad}}\left[\dot{m}_s\cdot c_{p,s}\left(T_{rad,in}-T_{rad,out}\right)\right.$$
$$\left.-U_{rad}\cdot A\cdot LMTD^n\right] \tag{5}$$

$$\dot{Q}_{rad}=\dot{m}_s\cdot c_{p,s}\left(T_{rad,in}-T_{rad,out}\right) \tag{6}$$

Note that the *LMTD* is raised to the exponent *n*. This is an approximation to the fact that only part of the heat transfer from the radiator is due to convection, the rest is due to radiation. The contribution from radiation (which normally amounts to 30-50 per cent of the heat transfer) is also a function of the temperature difference between the radiator's surface and the surrounding walls. The complex relation, which also includes physical quantities, can be simplified by including the radiation in (5) by introducing the so-called radiator exponent, *n* [6].

A common method to model the dynamics of the building's heat losses is described for example in [1] and [3]. The basic idea is more or less the same as with the HEX and the radiator. The wall is treated as a homogenous material with a homogenous temperature described as:

$$\frac{\partial}{\partial t}\left(T_w\right)=\frac{1}{m_w\cdot c_{p,w}}\left[\alpha_i\cdot A\left(T_i-T_w\right)-\right.$$
$$\left.\alpha_o\cdot A\left(T_w-T_o\right)\right] \tag{7}$$

With the wall temperature known the heat flow from the indoor air to the wall can be calculated from:

$$\dot{Q}_w=\alpha_i\cdot A\left(T_i-T_w\right) \tag{8}$$

The indoor air temperature can finally be calculated from:

$$\frac{\partial}{\partial t}\left(T_i\right)=\frac{1}{m_i\cdot c_{p,i}}\left[Q_{rad}+Q_{int}-Q_w-Q_{vent}\right] \tag{9}$$

Q_{int} is internally generated heat from humans and electric equipment like TV's, refrigerators etc. Q_{vent} is the unavoidable heat loss from ventilation of the indoor air.

The heat carrier from the HEX to the radiator is the circulating water in the system. The flow depends on the available pump head and the flow resistance in the system. For fully turbulent flows and limited temperature variations, the relation between flow and differential pressure in a component can be approximated as:

$$\dot{m} = k_v \sqrt{\Delta p} \tag{10}$$

It is reasonable to assume that the flow is turbulent, even when dealing with natural circulation. If a laminar flow should occur, the pressure drop will be so small, in absolute numbers, that the influence will be negligible. The factor k_v describes the flow capacity of a component (to be precise, the flow capacity at 1 bar differential pressure). A k_v value can be calculated for all components, a pipe section, valve (which has a variable k_v value depending on the opening degree), HEX or radiator, as long as the pressure loss for a specific flow is known. For a system with many components (connected in parallel or series), equivalent k_v values can be calculated from:

$$k_{v,eq,parallel} = \sum_{i=1}^{N} k_{v,i}\ , \quad k_{v,eq,series} = \frac{1}{\sqrt{\sum_{i=1}^{N}\left(\frac{1}{k_{v,i}}\right)^2}} \tag{11}$$

Let us now consider the hydronic radiator system as a circuit diagram, see Figure 3. This analogy to electric circuit laws is described in [4] for a domestic hot water system. For branch i, k_v values are indicated for supply and return pipes to the branch and to the radiator, respectively, and for the radiator itself, including its control valve which can be controlled manually or by a thermostatic head. In fact, every branch in the model consists of three storeys, although only one is indicated in the figure. However, each branch in turn comprises a system just like the one shown in the figure.

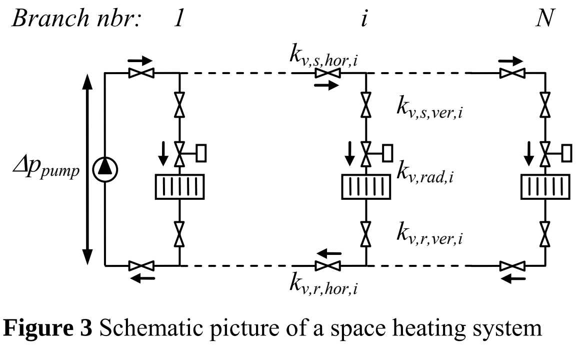


Figure 3 Schematic picture of a space heating system

Equation (11) can now be used to simplify the system by replacing flow capacities, in series or in parallel, by equivalent capacities which in turn can be simplified. The procedure is shown in Figure 4 below.

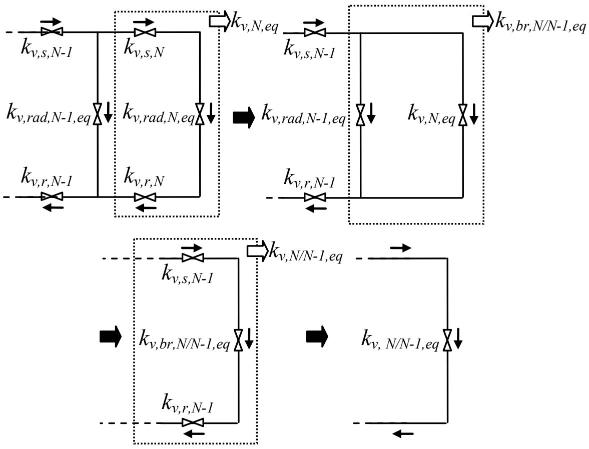


Figure 4 Simplification of the circuit scheme of the space heating system

Finally, we arrive with just one equivalent k_v value for the whole system, and (10) can be used to calculate the total circulation flow, $\dot{m}_s$, in the system. Then, the flow in each part of the system can be calculated by once again going through the system, but now in the opposite direction. We know the available differential pressure for the first branch (after the pump) and the equivalent k_v value and (10) gives us the flow through branch 1. We can then carry on by subtracting the pressure loss from branch 1 and repeat the calculation for branch 2, and so on.

Nowadays, the use of speed controlled pumps is a rule rather than an exception. In the model the pump pressure head is therefore assumed to be constant.

4.2 Improvements and expansion of the model

In this work, the model is used to facilitate the studies on how space heating systems connected to DH will behave without electric power. In order to do that the model needs to be modified. The first step is to introduce the driving force from natural circulation.

4.2.1 Natural circulation

This mechanism is based on the difference in density between hot and cold water. Many years ago, before pumps were introduced, heating systems were designed to operate with natural circulation. The differential pressure originating from natural circulation can be derived from:

$$\Delta p_{nat} = g \cdot h\left(\rho_r - \rho_s\right) \tag{12}$$

The height, h, in the system is measured between the heat source (HEX) and the heat sink (radiator). This force is of course present also in a pump operated system, but is negligible compared to the pump's pressure head, and can be neglected. When designing a natural circulation system, large pipe diameters must be used in order to minimize pressure drops. As seen in (12) a high building and a high supply temperature support natural circulation. The question is how this works in a system designed for pump operation.

The problem is that the Kirchoff method will have limitations when the pump head is lost. As a first approach, the differential pressure from the pump can be substituted with a calculated differential pressure from (12), based on supply and return temperatures at the HEX and the system's height. This is, however, a simplified model, since the differential pressure actually arises at every riser in the system, i.e., where the height actually is situated. When testing natural circulation in large buildings, it turns out that some risers, situated far from the HEX, might not obtain any circulation, or it might take a long time until circulation arises. Also, the magnitude of the circulation tends to be exaggerated when the temperature difference at the HEX is used for calculations in all risers. During natural circulation, the flow velocity is drastically reduced which means that heat losses will be significant and the supply temperature at the risers will be more or less reduced, a fact that gives a reduced driving force from natural circulation.

The mechanism for natural circulation could be compared to having a small "pump" at every riser. In our case, the model has four risers which mean that we have four such "pumps" or sources for differential pressure. With four sources instead of one, we must consider four circuits when calculating the flow distribution. The flow in each circuit is calculated individually and the total radiator flow corresponds to the sum of the four circuit flows. The difference between calculating flows in the system driven by pump or by natural circulation, respectively, is shown in Figure 5.

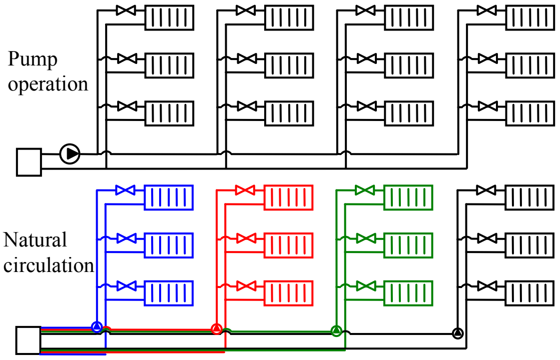


Figure 5 Difference between pump and natural circulation regarding flow calculations

Regarding the implementation of natural circulation in Simulink, a smooth way to handle the to operational modes, pump or natural circulation, is to introduce a control variable which has the value 1 as long as the power supply is OK (pump operation) and 0 when the power fails (natural circulation). The control variable will then govern the switching between the operational modes and, consequently, flow calculations. The control equipment is also affected by a power failure, and this is easily taken care of using the control variable to freeze the position of the DH valve in its current position.

4.2.2 Building shell model

The model of the building shell showed in section 4.1 has proved to be sufficient when modelling the operation of space heating systems during "normal" conditions or static calculations, i.e. where the transient course of the indoor air temperature is of minor interest. However, dealing with power failures must be considered as rather extreme conditions. The existing model showed problems using the conventional building model where the wall is modelled as one section with a homogeneous temperature. Let us consider an example: The outdoor temperature is -10°C and the indoor temperature is 20°C. The temperature throughout the wall will then be (20 – (-10)) / 2 = 5°C. If the heat supply is cut off (or substantially decreased) the indoor air temperature will drastically approach the wall temperature, 5°C. This is not realistic since the wall will have a temperature gradient where the inside surface is just a few degrees cooler than the indoor air temperature.

Figure 6 below shows data from an experiment carried out in a small office at our department. The radiator and ventilation in the room was shut of and an electric radiator, equipped with a timer function, was installed. Several temperatures were measured in the room, among them indoor and outdoor air, the temperature on the inside wall and on the inner surface of the outer wall. The figure shows what happens when the heat supply is cut of. During the test the outdoor temperature was about 5°C.

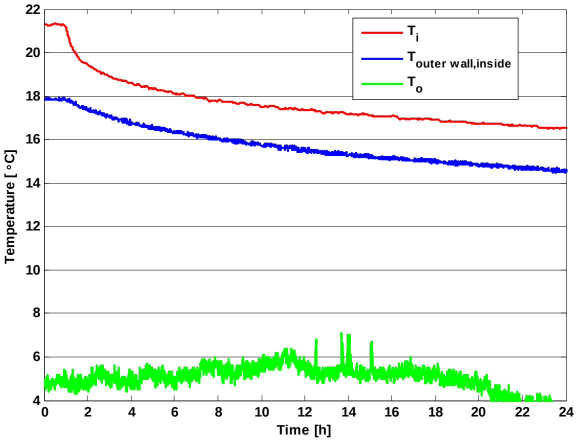


Figure 6 Results from an experiment where the heat supply to a room is cut off

As can be seen, the indoor temperature initially approaches the inside surface temperature of the outer wall. Figure 7 below shows a simulation corresponding to the experiment above. The figure comprises results from, on one hand, the existing model (the curve indicated with "◊") and, on the other, the modified model (the curve indicated with "o"). With the one-section model the wall temperature lies exactly between the indoor and the outdoor temperature. As already mentioned, the indoor temperature drastically approaches the wall temperature and estimates to 7°C already after 24 hours.

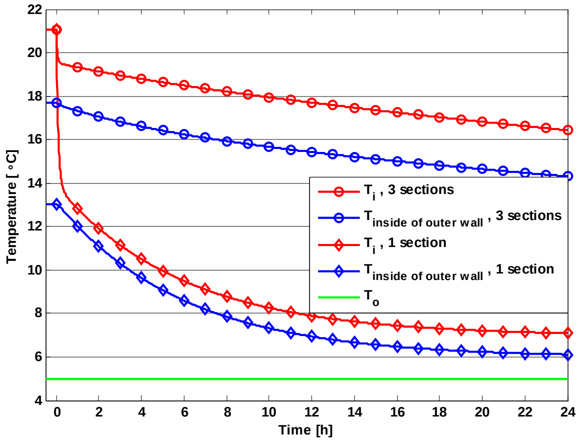


Figure 7 Simulation using the one-section and the three-section building wall model, respectively, where the heat supply to a room is cut off

To improve the modelling of the indoor temperature the building model has been modified according to Figure 8. To the left is the simple, one-section, model and to the right is the new, three-section, model. By using more sections, different wall materials can be used. Here, the two innermost sections are assumed to be made of concrete, which has a high conductivity but a large mass, and the outermost section is the insulation, which has a low conductivity but a small mass. Both materials are essential. The concrete will assure that we have an inside wall temperature relatively close to the indoor temperature and make sure that the building has a realistic thermal inertia. The insulation will assure that we get the correct heat loss and temperature drop through the wall. The choice to use three sections is a compromise; more sections give higher accuracy but longer computational times.

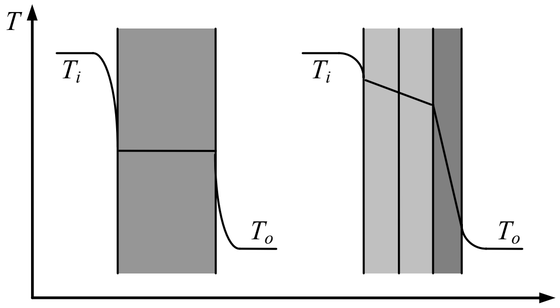


Figure 8 Schematic picture of the one-section (left) and the three section (right) wall model

Now, using the new three-section model, the simulation was run once again. The simulation result, already shown in Figure 7, is now much more in accordance with the results of the real experiment (Figure 6).

4.3 Assumptions for the model

Regarding the model, a number of assumptions have to be made in order to limit the complexity of the model and the computational time.

Most of the assumptions are already discussed, such as *LMTD* versus arithmetic mean value calculation in the HEX model and number of sections in the wall model. Another essential assumption is that there is no heat transfer between the flats. Without this assumption the model would be extremely complex and demand large computational power. The consequence of the assumption is that if the radiator flow is unbalanced the indoor temperature in the flats will not be accurate due to the fact that heat conduction through the walls between the flats is neglected. However, the average temperature in the building would still be correct. This is only relevant if the flow is unbalanced for a long time since heat transfer between the flats is very slow.

The supply temperature in the DH grid is a function of the outdoor temperature. In this study the temperature profile for Malmö is used for the simulations, see Figure 9.

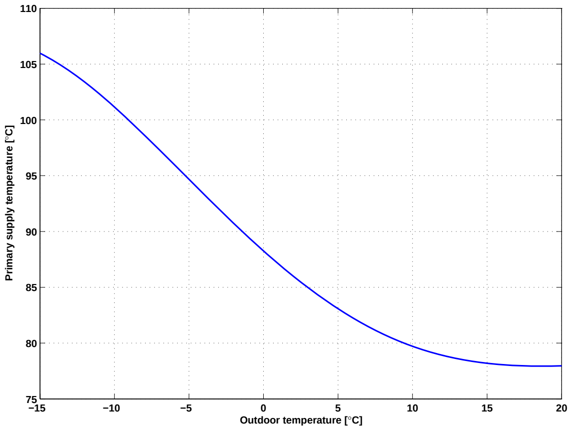


Figure 9 The supply temperature in the DH grid

5. Results

In order to evaluate the accuracy of the model, field studies and simulations with similar outer conditions have been compared. The model has thereafter been used for parameter studies.

5.1 Model evaluation

For evaluation of the model a comparison between a real field study and simulations are made in Figure 10. The studied object is a building located in Malmö. The building was built in 1952 and consists of 20 residential flats distributed on three floors. Before the time 0 in the diagram, the operation is normal and we have 100 per cent circulation flow and heat output. At the time 0, a power failure occurs and the circulation flow decreases drastically. As a consequence, the radiator supply temperature increases substantially. Finally, we end up with a substantially higher temperature drop in the space heating system, which leads to a rather high heat load even with natural circulation.

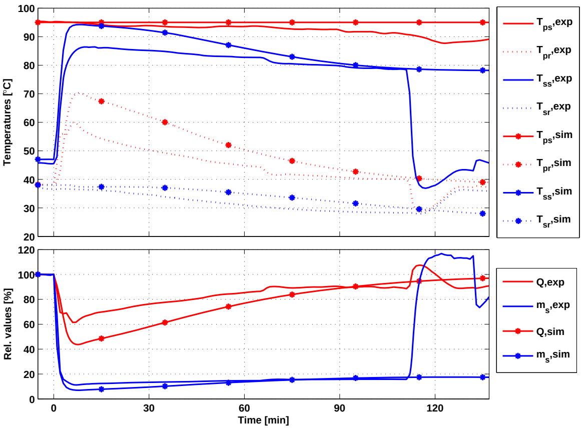


Figure 10 Comparison between field study and simulation

As seen, the simulation shows good resemblance with the field experiment. Even if there is some difference in the transient phase good resemblance is shown both initially (before the power failure) and when the natural circulation has come to a steady state.

Possible explanations to the deviation are the estimated time constant for the building and the space heating system (water volume, pipe lengths etc.) and variation in primary supply temperature.

The model now makes it possible for example to investigate long time effects of a power failure for the building. In Figure 11 a power failure lasting for three days is simulated for the same building.

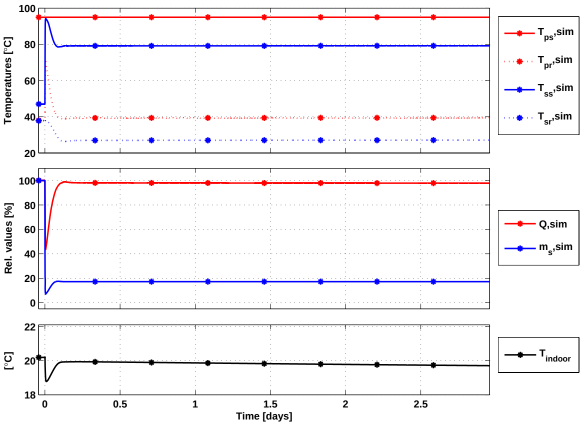


Figure 11 Simulation of a power failure lasting for three days

The figure shows the obvious fact that the natural circulation, once started, will go on at the same level after reaching a steady state. The mean indoor temperature is also represented in the figure.

5.2 Parameter variation

In the modelling world it is possible to investigate the influence of different parameters and outer conditions.

In case of a power failure the DH company can adjust the primary supply temperature level. As described in section 4.2.1 the magnitude of the natural circulation flow is dependent on the temperature drop in the radiator system. In Figure 12 steady state results (the state after infinite time) from simulations with different primary supply temperatures are shown at different outdoor temperatures. In the upper diagram the relative heat load (natural circulation compared with normal operation) is shown. The lower diagram shows the mean indoor temperature.

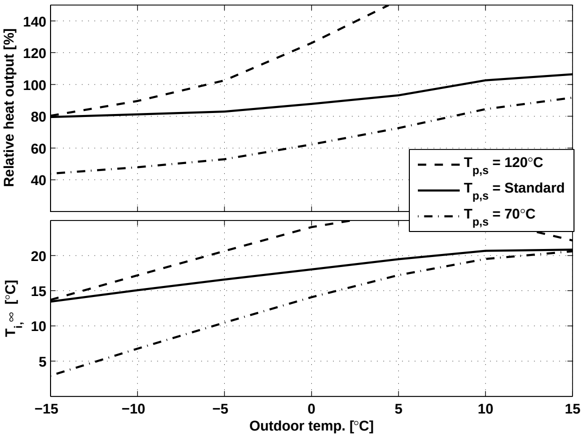


Figure 12 Variation of primary supply temperature at various outdoor temperatures at steady state conditions

The figure shows that by increasing the primary supply temperature the relative heat load increases, especially for outdoor temperatures above -10°C. At low outdoor temperatures the effects of increasing the primary supply temperature to 120°C is small due to the already high supply temperature.

The next diagram, Figure 13, shows the dynamic indoor temperature at three different outdoor conditions. Also in this figure the influence of changed primary supply temperature is shown.

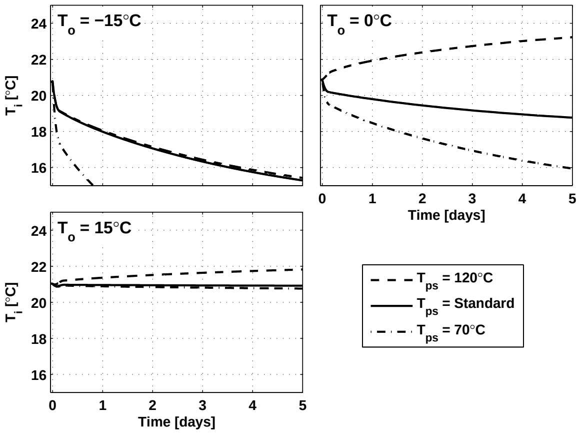


Figure 13 Dynamic indoor temperature at different outdoor temperatures and variation of primary supply temperature

5.3 Conclusions

A complete model of a building with a hydronic space heating system connected to DH has been built up. Parts of the model have been improved and developed to make it possible to simulate not only normal operation using a pump but also natural circulation during a power failure. By comparing the results from simulations with field studies good resemblance can be shown. A model makes it possible to study a building when changing input parameters, e.g. outdoor temperature. It is also possible to simulate the expected indoor temperature dynamic as well as at steady state in case of a power failure.

The model will constitute an important tool in our further studies regarding secure heat distribution during power failure.

6. Acknowledgements

The study was funded by the Swedish Energy Agency, the Swedish District Heating Association, the municipality of Malmö, E.ON Värme Sverige AB and MKB Fastighet AB.

References

[1] Gummérus, P. (1989). "Analys av konventionella abonnentcentraler i fjärrvärmesystem" Doctoral thesis, Chalmers University of Technology. (Gothenburg, Sweden, 1989).

[2] Hjorthol, E. M. (1990). "Optimization of Design Values in District Heating Substations by System Simulation" Doctoral thesis, Norges Tekniske högskole. (Trondheim, Norway, 1990).

[3] Näslund, M. (1959). “On the design of residential condensing gas boilers” Doctoral thesis, Lund University. (Lund, Sweden, 1997).

[4] Persson, T. (2004). "Dynamic Modeling of a Domestic Hot Water System Using Simulink" In Proceedings of the 45th International Conference of Scandinavian Simulation Society (SIMS 2004). (Copenhagen, Denmark, 2004).

[5] Persson, T. (2005). "District Heating for Residential Areas with Single-Family Housing" Doctoral thesis, Lund University. (Lund, Sweden, 2005).

[6] Trüschel, A. (1999). “Värmesystem med luftvärmare och radiatorer” Chalmers University of Technology. (Gothenburg, Sweden, 1999).

[7] Wollerstrand, J. (1993). “Fjärrvärme-abonnentcentraler med genomströmningsberedare för tappvarmvatten” Licentiate thesis, Lund University. (Lund, Sweden, 1993).

Assessment of Power Transformer cooler with FEMLAB

S. Firouzifar, J, Mahmoudi
M.D.H University of Vasteras , Sweden
sfirouzifar@yahoo.com

Abstract- The significance of overload in power system has increased substantially due to of consumption and ambient temperature rise. An important consideration when evaluating the impact of overload is their on power system ambient tem-and load. Transformers are major components in power systems. The increase useful lifetime and hence abnormal temperature rise.

Existing standards give a procedure to determine the capability of an existing transformers subject to overload problem and ambient temperature rise based on conservative assumptions .in addition, the temperature oil rise of power transformer due to overloading and ambient temperature are estimated based on power and cooling system, and the average daily or monthly temperatures to which a transformer would be subjected while in service. it is the purpose of this research of this effort to quantify the decreased life time due to overloading and the corresponding temperature rise in transformers. This is accomplished using a 2-D FEMLAB Model adapted for cooling simulation in power transformers. [20-24].

Keyword: Transformer, Overload, temperature, Ambient, cooling Femlab

NOMENCLATURE

q : Temperature ($°C$)
HX: Heat exchanger
$q_h(t)$: The windings hot-spot temperature
$q_a(t)$: The ambient temperature
$\Delta q_h(t)$: The hot-spot temperature rise, depending on the load, nameplate rating and type of cooling of the transformer
T : is given degrees Celsius (° c)
M/s: Meter per second
U: fluid velocity
P: pressure field
ρ : density
h : Viscosity
C: specific heat capacity
Q: heat power per unit volume
2-D: two dimensions

Introduction

LOading of power transformers at their nameplate capacity does not always match their loading capacity. In [1], an effort to determine loading capability of power Transformer based on their maximum windings temperature and not on their rating nameplate capacity, various simulation thermal models have been developed up to now to determine the windings hot-spot temperature .This temperature is being used not only for the calculation of the permissible overloading but for the aging of the power transformers as well [2, 3, 4]. IEEE Thermal aging of power transformers is a result of their insulation deterioration .This deterioration is cumulative, depending on the temperature and the corresponding time interval, this is why the variation of the windings hot-spot temperature vs. time should be taken into account, and not only its maximum value, to determine transformer ageing. This variation is given from the expression:

$\theta h\,(t)= \theta a\,(t) + \Delta\theta h\,(t)$…….. (1)

Rise in electric power consumption increased the power transformer loads by %16.3 during year 2006 in provinces of Mazandaran. Over loading generates high temperature within the power transformers. Over loading and over heating of power transformers combined with high ambient temperature caused long black outs and major equipments damages in province of Mazandaran during summer of 2006.

Maximum allowable loading and overloading are the most important parameters for the power transformers. Depending on how long a power transformer is under a certain load or overloaded and ambient temperature rise they generates a considerable amount of heat internally. The internal heat generated in turn reduces the life of the power transformer insulation this reduces the life of the power transformer. Rise in internal temperature depends on amount of load, the time that the transformer is under load or amount of over load and the ambient temperature.

This paper shares general and practical information concerning observation of heat generation in power transformers due to overloading. It evaluates the effect of ambient temperature on over heated power transformers. To carry out these tasks IEC-354 standard were used. Data collected from regional meteorological center was used to evaluate the effect of the ambient temperature on the power transformer performance.

The ageing rate of power transformers determined using IEC & BS standard and guidelines for loading of the oil filled power transformers.

1-overload and ambient temperature

1-1-2-Loading beyond nameplate Rating

Corresponding Author; S. Firouzifar PhD student at Mdh University,Dept. of public technology, Malardalen University, With the exception of generator-transformers .the load imposed on transformers varies between a higher level within the day and during the year .the most critical limitation in the loading of a transformer is the temperature reached in the hottest area of the winding, named Hot-spot temperature, Every effort should be made to determine this temperature with accuracy .As the size of transformer increases. The hot-spot temperatures are more difficult to determine correctly.

With loading values beyond the nameplate rating, are not exceeded for current I, hot-spot temperature *q h* and metallic parts in contact with insulating materials and top-oil temperature.

The loading of the transformers beyond the nameplate rating cause the aging of the nameplate rating cause the aging of the insulation materials and the loss of the life expectancy, which is discussed in the next parts . As shown in table (1) below, overloading is the biggest problem in M.R.E.C. [19]

Table (1): transformers overload in (M.R.E.C) sub-station [19]

row	No. of transformers	transformers 63/20(kv)	transformers 230/63(kv)	transformers 400/230(kv)
1	No. of transformer in iran	2304	467	95
2	No.of transformer in mazendran	130	20	8
3	No. of transformer in more than nominal load mazendran	61%	65%	76%
4	No. of transformer over load more then(120%)	37%	34%	27%

Fig [1]: A power transformer in a 63/20 KV substation, West of Mazandaran

1-2- Ambient temperature in M.R.E.C

Regional meteorological center in Mazandaran province have total of 23 Synoptic stations, 183 rain evaluation stations and 27 climatology stations. These stations collect meteorological data from environment hourly bases. Meteorological stations in Ghaemshar, Ramsar, Noshar, Babolsar, Gorgan and Gonbad store the meteorological data of past 25 years from Mazandaran province.

Ambient temperature is an important factor in determining the load capability of a transformer since the temperature rise for any load must be added to the ambient to determine operating temperature. Transformer ratings are based on a 24h average ambient of 20°c. whenever the actual ambient can be measured ,such ambient should be averaged over 24h ,and then used in determining the transformer's temperature and loading capability of the ambient air temperature seen by a transformer and the air in contact with in radiators or heat exchangers.

It is often necessary to predict the load that a transformer can safely carry at some future time in an unknown ambient. The probable ambient temperature for any month may be approximated from data in reports prepared by the national or local atmospheric authority for the sections of the country where the transformer is located. [8, 9]

At different locations on earth, the average of variations in the ambient temperatures over a day, month or the whole

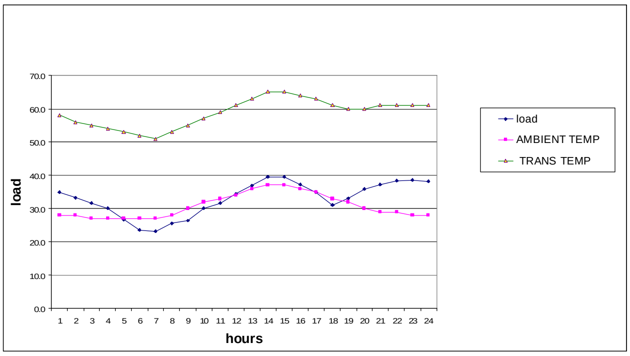


Fig [2]: Load, ambient temperature and transformer temperature M.REC substation

Year varies due to the rotation of the earth on its own axis in

24h and around the sun in a year. The temperature variations over a day are plotted for the most of the world, but in Mazandaran state the variation observed are found close to sinusoidal.

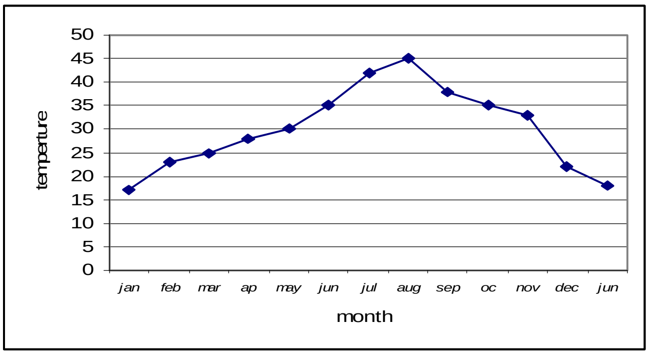


Fig [3]: Typical monthly ambient variation

That will appear as 365 sinusoidal ripples on the yearly sinusoidal variations which is very bad news for M.R.E.C.

1-3- Effect of ambient temperature on useful life of the power transformers

The most important parameters in computation of temperature build up in power transformers dielectric are the ambient temperature and the cooled airflow from the cooling system. Unfortunately, when ordering a power transformer most companies indicate the maximum absolute temperature as ambient temperature.

They believe that loading the ordered power transformers with maximum absolute temperature and nominal power is possible during the predicted lifetime of the power transformer. Based on IEC-354 standards, if hot spot temperature of the wire winding hot spot at the time of the loading is 98^{o}C regardless of ambient temperature the relative aging of the power transformer is one.

For every 10^{o}C increase in wire winding hot spot temperature within limit of $98^{o}\text{C} < Y_h < 140^{o}\text{C}$, either caused by ambient temperature or by loading conditions the aging rate of the power transformer doubles. Oil temperature and wire winding hot spot temperature of power transformers should not exceed 105^{o}C and 140^{o}C respectively. In order to derive the aging rate of the power transformers to one, power transformers must operate within the average yearly ambient temperature with specified nominal loading. When ambient temperature drops below average yearly ambient temperature the aging rate also drops blow one. In this condition it is possible to load the power transformers beyond the specified nominal loading so long as the wire winding hot spot temperature does not exceed more than 98^{o}C. If ambient temperature is higher than average yearly ambient temperature transformer load must be reduced to keep the wire winding hot spot temperature at 98^{o}C.

If the above loading pattern during a year is not acceptable. If it is desired to keep the loading of the power transformer at nominal during all seasons of a year; Then in hours of a year when ambient temperature is lower than average yearly ambient temperature the aging rate of the transformers are less than one and vice versa. This gives the total aging rate of one for the year that the power transformer load kept at nominal constantly. In another word over loading a power transformer operating in its designed environment accelerate its aging rate. It is not recommended by any standards including the IEC. Power transformers designed to operate at higher ambient temperature than their geographic location have capacity to accept load more than their stated nominal load.

For power transformers to live for 30 to 40 years given they operate at 98^{o}C which is equivalent to 20^{o}C at nominal loads, IEC-354 standard gives the allowable loading limit in per unit for power transformers in different ambient temperatures according to their nominal power [1]. Loading guideline is for power transformers according to IEC-354/1991 for hot spot temperature less than 140^{o}C. Fig 2&3 presents the temperature variation of the different area for provinces of Mazandaran

2-.Technique for simulation of temperature [20-24]

The principal factor in determining the thermal rating, and thus the loading capability and ambient temperature of transformers, was the average rise of the winding. This temperature rise was determined by simulation and the change when in oil temperature when the oil velocity was increasing to the transformer. The temperature rise was changed when it was discovered that the hottest-spot winding temperature gives a better indication of how the load affects the life expectancy of a transformer [20-23].

However, the hottest-spot temperature cannot be measured directly because of the hazard of inserting temperature sensors directly into the windings. As a result, numerous methods have been developed to calculate the hottest-spot temperature, but these are very complicated. The winding and oil temperature of a transformer are measured with a fiber optic temperature measurement system. Although the sensors are not located directly in the hottest-spot of the winding, significant results can be obtained. Assuming a conventional ONAN transformer design according standard using cellulose insulation and mineral oil also of life table can be constructed for a rang of overloads and varying periods of timeout overload. Table (4),

Table 4, loss of life in power transformer[25]

load % rated	Hot spot temp, °c at amb.20° c	Top oil temp .°c	% Time at over load		
			5	10	20
100	98	76	0	0	0
110	104	78	1.5	2.5	5
120	110	85	2.5	5	10
130	116	87	5	10	20
140	122	94	10	20	>40
150	127	98	20	>40	>40
			loss of life in years over normal Power transformers life of 29 years.		

3- Cooling transformers

The transformer is normally cooled by natural convection (via the ambient air) or forced convection (a fan attached to The wall of the radiator)

The efficiency of the air – side heat transfer in a transformer radiator is a primary consideration when determining the best heat exchanger for a particular application. Further more, air–side heat convection is typically the limiting factor affecting heat performance of a heat exchanger. The convective heat transfer equation is proportional to surface area of the fins or bodies subjected to the airflow.

In order to improve the modeling of the natural convection problem, an independent geometrical model of the surrounding air will be created and considered separately. The continuity of mass flow, temperature and heat flux a long the interface between transformer and air will be enforced by an iterative procedure. This procedure will allow us to calculate and then prescribe local heat fluxes to the external walls of the transformer. The numerical results obtained with the numerical model will compared with experimental measurements .On the basis of hydrodynamic theory and of data obtained by simulated with FEMLAB a method will be presented for calculating the overall flow and its components through the transformer Radiator. Following a hydraulic calculation .It is possible to suggest suitable design adjustments to render the temperature - rise increments, thus achieving optimum utilization of the circulating oil. The calculated oil velocities in the transformer radiator permit the heat - transfer coefficients to be determined reliably; the oil/air temperature gradients can be established from the known rates of oil flow through the ducts of the coolers.

The optimal configuration will be approached by determining the optimal radiator geometry. In brief, optimal means to have a batter heat exchanger that those are currently available in the market. In addition to high efficiency and thermal performance issues such as manufacturing cost, ease of manufacturing, as well as weight and size reduction Of the HX must be taken into consideration.

4- Prepared program in FEMLAB software [24]

Using the equations and data collected from ambient temperature changes, overloading data collected from the power transformers and also power transformer specifications, a computer program was developed to simulate the hot spot location and the oil temperature rise.

The model uses two stationary application modes to simulate the problem:

1- Non-Isothermal flow

2- General heat transfer

"It simulates non –isothermal flow with the Navier-stokes equations that describe the fluid velocity and pressure field, in this case density and viscosity are temperature dependent"

$$\rho(u.\nabla)u = \nabla.[-PI + \eta[(\nabla u + (\nabla u)] - (2\eta/3)(\nabla.u)T] + \rho g \quad \nabla.(\rho u) = 0 \quad \ldots\ldots. (6)$$

The General heat transfer application mode is based on a general energy balance:

$$\nabla . (-k\nabla T) = Q - \rho\, Cp\, u. \nabla T \quad \ldots\ldots\ldots \quad (7)$$

Following are the temperature dependencies of viscosity, density and specific heat capacity in this model (this information comes from the producer of the transformer oil)

$*P = 875.6 - 0.63\, T$

$*y = \rho 10^{-4.726 - 0.0091\, T}$

$*Cp = 1960 + 4.005T$

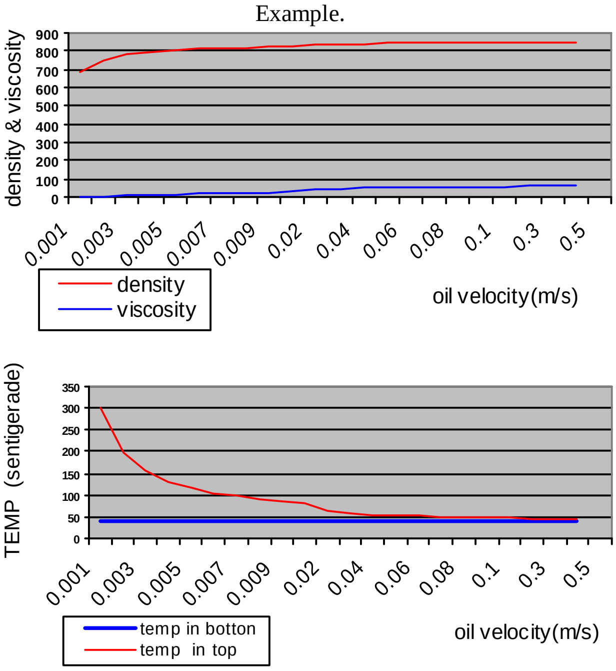
1-4.Design approche

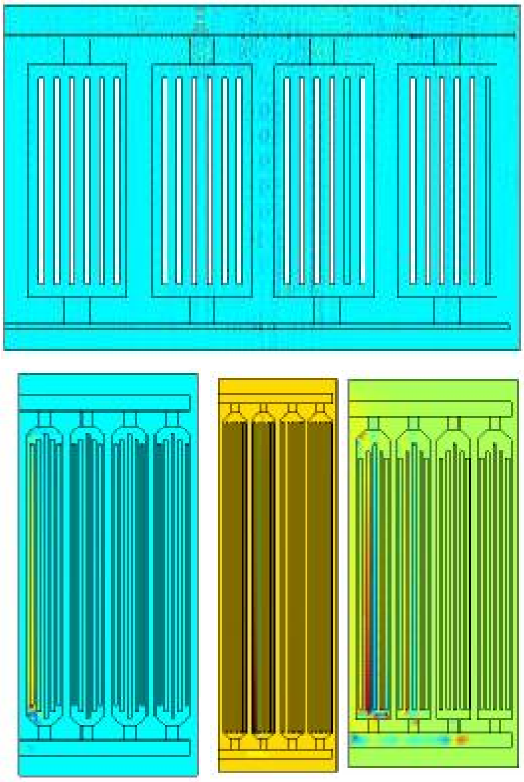
Using the equations and data collected from ambient temperature changes, overloading data collected from the power transformers and also power transformer specifications, a computer program was developed to simulate the hot spot location and the oil temperature rise.

The developed program helped to find a suitable cooling system for overloaded power transformers for nine areas in Mazandaran province of Iran. This program helps us to distinguish the kind of the applicable cooling in different ambient conditions and transformer load. For this case we clarify the cooling kind for transformers, sub transmission and transmission substation with the entering ambient temperature and kind of insulation in order to make us able to calculate aging insulation rate; for example; the hot spot is obtained at 98 $^{\circ C}$ and aging rate is obtained at 0.092. With consideration to plot temperatures a function of oil velocity, for the simplified model used in this example the temperature characteristics is plotted in Fig [4] as a function of oil velocity in cooling. Oil temperature will be decreased and oil velocity started of 0.001 [m/s] and continue until 0.4 [m/s]. Proposed stop at

0.2 [m/s] that maybe, practicing beholds relay at over than 0.2[m/s] oil velocity .with decreased oil temperature in power transformer modify viscosity and density. Shown in fig [4].By using various simulation software's such as FEMLAB , we can reach the provided primary design. For preventing the reduction of power transformers normal life we have to use novel cooling system. {According to Figures [5, 6, and 7], Transformer cooling [2] This has been developed to provide an alternative, Non-invasive Solution to the problem of overload and ambient temperature rise .rather than significantly altering the morphological level of the novel cooling system. Novel cooling utilizing the base of parametric understanding of the phenomenon, together with recent advances in morphological level and software level of simulation in FEMLAB environment.[2].changes only the operation of the system .[3,4]with fig [8] could modify exist cooling system .

Fig [4]: plot of temperature as a function of oil velocity for the Example.





Fig[8]: topic of simulated radiators
With FEMLAB software

1-5. Boundary conditions-General heat transfer [24]

There is a power transformer module in FEMLAB software.. We used of FEMLAB Environment. and put boundary conditions General heat transfer in software FEMLAB. This program set according to table (3). of course whit consideration to IEC- 354/standard.

1- From the metaphysics menu select General heat transfer

2- From the physics list choose 1, then in the boundary condition
Edit field select axial symmetry.

3- Select boundaries, and assign them the temperature boundary Condition. In the to edit field for temperature enter T-0
(Note that the value of T-0 is given in degrees Celsius).

4- Select boundaries, and assign them the convective flux
Boundary condition.

5- Select boundaries, and assign them the thermal insulation Boundary condition. They click "Ok"

options/constants		
name	**expression**	**value**
T_0$^{\circ C}$	105	105
eta -air	60-6+4e8*2018e-6	6.00E-06
K-air	10000	10000
Cp-air	1006	1006
R-air	2000	2000
Po	1.00E+05	1.00E+05
Q-SOUR	1	1
kfl	0.125	0.125
ks	383	383
Vo(M/s)	0.008	0.008
Qs	1	1
rho-air	p/(R-air*T-air)	2.5
P	100000	1.00E+05
V-air(M/s)	10	10
T-air $^{\circ C}$	20	20

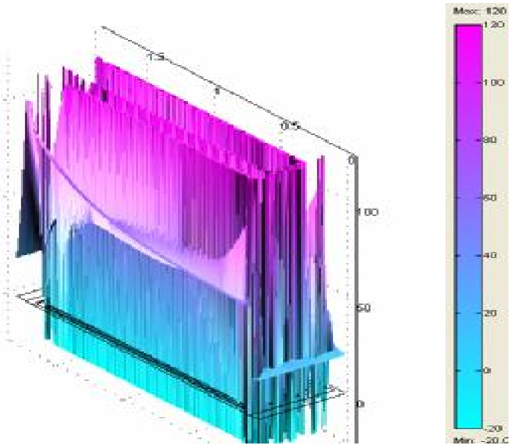


Fig [6]: simulation surface: temperature Height: temperature of power transformer cooling In FEMLAB

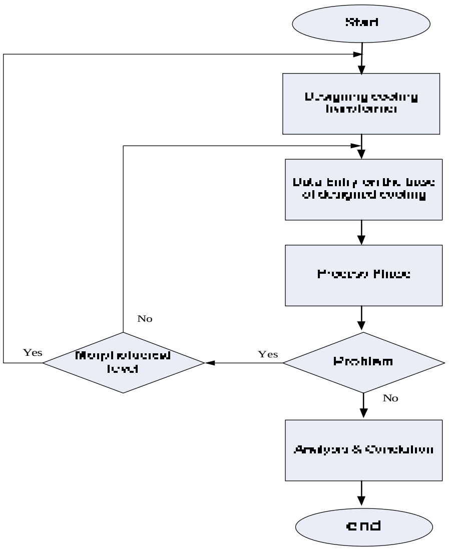


Fig [7]: simulation of power transformer cooling
In FEMLAB

CONCLUSION

Overloading the power transformers according to IEC-354 standard especially in emergency situations for long or short period of time at high ambient temperature can weaken the power transformers and reduce its effective life.

It can generally be said that one of the most important reasons early aging of transformers insulation system, is the problem of over current and overheat due to the increase of environmental temperature.

By using the FEMLAB simulation software, can somewhat improve the existing cooling system to prevent insulation early aging in transformers.

With the help of temperature simulation in FEMLAB software, a design can be implemented so that we may have heat without early aging of the insulation.

1. Than, we can conclude that the simulation project will help remarkably the transformer cooling system

2. Considering the kind of material the morphological level and the dimensions of Radiators can totally modify %17 the cooling system.

3. If more studies are carried out on the design of cooling system a cooling system can be designed to modify %45 the situation. In this case the existing cooling system should be changed to a greater extent.

Acknowledgment

The authors grate fully acknowledge the contribution of H. Mohammadian for his advice on the original version of this paper, and the director of MREC dispatching center for his helpful assistance for giving the data of power transformers used in this paper .

References

[1] –V. M. Montsinger ," loading transformer by temperature", AIEE Trans,Vo1.49,April 1930,pp293-297.

[2]–E .T. Norris,"Loading of power transformer", Proc, IEE, Vol. 114, no.2, February 1967, pp 228-232.

[3] –M .G. Carrothers, E. T. Morris,"Thermal rating of transformers ", proc, IEEE, Vol. 116, no.9, September 1961, pp 1564-1570.

[4]-B.D. Lahoti, D. E. Flowers," Evaluation of transformer loading above nameplate rating ", IEEE trans. Vol . PAS-100, no.4, April 1981, pp 1989 -1998.

[5]-J. Nejedly , "The loading of Transformer Unit in Power Plants and Network with Regard to Ambient Temperature ", CIRGE 1978 Session, Paper 12-01

[6]-B. J. Conway, D.W, MeMullen, A. J. Peat,J. M. Scofield," Loading of Substation Electrical Equipment With Emphasis on Thermal Capability Part I; principles" ,IEEE Trans., Vol. PAS-98 ,no.4, July/August 1979,pp 1394-1402.

[7]-I. S. Benko. D. O. Craghead, P. Q. Nelson," Loading of Substation Electrical Equipment With Emphasis on Thermal Capability Part II Application", IEEE Trans

[8]-B. J. Conway, D.W, MeMullen, A. J. Peat,J. M. Scofield," Loading of Substation Electrical Equipment With Emphasis on Thermal Capability Part I; principles" ,IEEE Trans., Vol. PAS-98 ,no.4, July/August 1979,pp 1394-1402.

[9]-I. S. Benko. D. O. Craghead, P. Q. Nelson," Loading of Substation Electrical Equipment With Emphasis on Thermal Capability Part II Application", IEEE Trans.Jacques Aubin Gesyprotec, Pointe – clarre, (Quebec), Canada.

[10]– Transformer (BHARA HEAVY Electrical LIMITED)

[11]- Heat sink auxiliary transformer cooler (Unifin) http://www.unifin.com/heatsink.htm

[12]- IEC publication 76-1" power transformer, part: 1 General", Second Ed...1993

[13]- IEC publication 76-2 "power transformer part2: temperature Rise " Second Ed..1993

[14]- D.HShroff and A.W. Stan net,"A Review of paper Aging in power transformers" IEE proceeding Vol.132, pt.c, No.6 Nov.1985, pp 312-319

[15]- K.Karsai, D.Kerenyi and L.Kiss, "large power transformers" book, Elsevier. Amsterdam, second Ed.1995

[16]- C.A. bozzini, " transformer Aging Diagnosis by Means of Measurements of theDegree of Polymerization Results of new Experiments CIGRE paper 12-08 1968

[17]- IEC publication 354 "Guide to loading of oil-immersed power transformer 1991, BS, 7735 1994

[18]- IRANTRANSFO company technical Documents.

[19]- MAZREC. Dispatching centers Technical reports . Website: www. Mazrec .co .ir

[20]- Golestan Province Dispatching center Documents.

[21] - MAZREC. Transmission Deputy. System Studies Affair.

[22]- VDE 0536/30"Belastbarkeit Von" oil Transformers" .

[23]- book" Transformer" Bharatheavy Electrical Limited Bhopal" New Delhi: tata McGraw-hill 2003"

[24]-"FAMLAB 3.1i" with MATLAB software consol Metaphysics 3.2 modeling guide .

[25] – "Evaluation of transformer overloading capability "Sarat C. Tripathy*,y and Erkki Lakervi

Department of Electrical and Communication Engineering, Helsinki University of Technology, P.O. Box 3000, 02015 Hut, EUROPEAN TRANSACTIONS ON ELECTRICAL POWER Euro. Trans. Electr. Power 2005; 15: 455–464

Published online 11 April 2005 in Wiley Inter Science www.interscience.wiley.com). DOI: 10.1002/etep.59

[26]- "Economic Benefit and Risk Evaluation of Power Transformer Overloading "Robert Ch6nier and Jacques Aubin, GE Syprotec, Pointe-Claire, (Qu4bec), Canada

[27] – "Power transformer overloading" A. Kalas D. 'Tsanakas

University of I'atras Power Systems 1,aboratory Patras. Giecce A. Sferides

Public Power Corpoiatiori Department lor (tic Maintenance of Trarismissioii Lines tind Substations Thessaloniki. Greecc

[28] – "Assessment of Overload Capacity of Power Transformers by On-line Monitoring Systems"S. Tenbohlen, T. Stirl ALSTOM Schorch Transformatoren GmbH, Rheinstr.73, 41065 Monchengladbach, Germany transforrner.monitorin g@tde.alstom.com

[29] L. W. Pierce, “Transformer design and application consideration for nonsinusoidal

Load currents,” IEEE Trans. on Industry Applications, vol.32, no.3

1996, pp.633-645,

[30] S. A. Stigant and A. C. Franklin, The J & P Transformer Book. 10th ed. New

York: Wiley, 1973

[31] J. Lammeraner and M. Stafl, Eddy Currents, London: Iliffe, 1966

[32] R. L. Stool, The Analysis of Eddy Currents. Clarendon, 1974,

[33] O. W. Anderson, “Transformer leakage flux program based on finite element

Method,” IEEE Trans Power Apparatus and systems, vol. PAS-92, 1973, pp.682-

689.

[34] D. Pavlik, D. C. Johnson, and R. S. Girgis, “Calculation and reduction of stray

And eddy losses in core form transformers using a highly accurate finite element

modelling technique,” IEEE Trans Power Delivery, vol. 8, no. 1 Jan. 1993, Pp.239-245, References 114

[35] "EVALUATION OF POWER SYSTEM HARMONIC EFFECTS ON TRANSFORMERS"Hot Spot Calculation and Loss of Life Estimation

Doctoral Dissertation Helsinki University of Technology Department of Electrical and Communications Engineering Power Systems and High Voltage Engineering

Asaad A. Elmoudi TKK Dissertations 24 Espoo 2006

[36] IEC 354 Standard Publications: 1991, Loading Guide for Oil-Immersed Power

Transformers,

[37] IEEE standard C57.91-1995, Loading Guide for Mineral-Oil-Immersed

Immersed Transformers,

[38] Working Group 12-09, “Heat Run Test For Power Transformers” Electra,

no.129, pp.37-39,

Biographies

Sohrab.Firouzifar: was born in Damavand in Tehran, IRAN on July 9, 1959. He graduated from the institute of technology abbaspor of Tehran (M.S), and PHD student in MDH University of Vasteras Sweden.

He is director standard and quality control in Mazandaran regional electrical company Iran north, and member IEEE in IRAN .he did publish several papers in Iran for EX. International Power system conference and distribution system conference about expert system in transmission substation & power transformers utilization, he was director technical office for 10 years, and was director R&D in M.R.E.C for 7 years.

Ontology Approach for Co-Use of 3D Plant Modelling and Large Scale Process Simulation

Marko Luukkainen, Tommi Karhela
VTT Technical Research Centre of Finland, P.O.Box 1000, FIN-02044 VTT
marko.luukkainen@vtt.fi

Abstract

Traditionally plant design- and process simulation applications are separated, but increasing use of simulation to support design and validation drives to find means for integration. We propose an ontology-based method for combining 3D plant modelling, large scale process simulation, and visualization. The use of ontologies allows clear distinction of a plant model and a simulation model, which enables multiple simulation models and simulation types to be used with the same plant model. Another contribution is that the same user interface and same visualization techniques can be used with different simulators.

Keywords: 3D Industrial plant modelling, process simulation, visualization, ontology

1. Introduction

Use of simulation has increased in process industry. Plants are getting more complicated, and need for advanced methods for complex system design is growing. Today it is common that when new plant is designed, designs are tested and verified using simulators. Typical simulators contain their own user interfaces and their own data structures, and transforming modelling data to simulator compatible form is seldom automatic. Usually a plant model has to be redone for the simulator. This makes the use of simulators more laborious and more error prone. In typical case, various systems in plant are modelled with different software, which further increases the risk of error.

Currently, the most common way to create a model for process simulation is to use two dimensional diagrams. At the same time, nearly all new plants and changes to old plants are designed with some 3D modelling software. Combining 3D modelling with simulation combines advantages from both: Two-dimensional diagrams cannot show physical properties of the modelled process, but 3D models are similar to real world objects. 3D-modelling comes especially handy when sizes of objects are needed by the simulator. In 2D-diagrams numbers must be given explicitly, whereas in 3D-modelling all dimensional values are already in the modelled process.

Because simulators are used for testing and verifying the functionality of the modelled process, it is more practical to include simulation within the plant modelling software. This would allow an engineer to test his design earlier and improve his understanding of the modelled process and its behaviour. The improved understanding leads to better design practices and results in savings in time and money. In understanding the process and its behaviour, visualization is important. Animations and other visualization techniques can improve perception of simulation results over text-based numeric information or separate 2D graphs.

Ontology-based knowledge representation has gained popularity because of its flexibility and expressive power. Here using the ontology-based approach has critical advantages: 3D modelling user interface does not have to be bound to any particular simulator nor any particular simulator type. Instead, simulators can be linked to 3D model's data structures, and the same visualization package can be used with many different simulators and the same 3D model with multiple simulators. This is required, because different simulators are used in different stages of the modelling process: First mass and energy balances are simulated using steady state simulation; later dynamic simulation can be used for various purposes, including testing plant's behaviour in changing conditions, and in operator training.

2. Related work

Gruber[1] uses ontologies for knowledge sharing. Uschold and Gruninger [2] use them for communication, inter-operability, and systems engineering. According to Guarino [3], ontologies can be used as database components, user interface components, and application program components either development time or run-time. Ontologies have become popular in several scientific areas including Artificial Intelligence, Computational Linguistics, and Database Theory.

Kalogerakis, Christodoulakis, and Moumoutzis [4] present an interoperable framework for integration of virtual reality scenes and semantic information. Their system uses several different semantic mappings between information model and graphical model. For instance, “equivalence”-relation describes that every NURBS curve represents a membrane or a polygon mesh represents a component in a specific shell. Linking graphics to domain concepts works with domain independent mapping relations and intermediate mapping classes, which allow interpreting relationships always the same way. Park [5] uses ontologies to combine dynamic models used in simulation to geometric models of the phenomena being modelled, calling the methodology as integrative multimodeling. With integrative multimodelling, the user could bind geometric models to simulation model, and then visualize the simulation. Lehtonen and Karhela [6] combine process simulation and active diagrams with ontologies. The system used ontologies and rules to generate simulation model from graphical model (diagram), and then visualizes the simulation with animations.

Rohrer [7] shows how visualization provides better means to understand behaviour of simulated model. The main reason is that the models of manufacturing simulations tend to be complex and counter intuitive, but visualization is a natural way to communicate a lot of information between computer and its user. Quick et al. [8] develops a system for manufacturing simulations and visualizations. The system contains a library of reusable models of equipment and animation methods. The user could use those to create manually or in some cases automatically a visualization model for the simulation. Animations are bound to event-based simulation and it could visualize for example movements of a forklift.

3. Ontologies and mappings

Before we can create an ontology for 3D plant modelling, we have to choose base ontology; the ontology that describes concepts, like classes and instances, basic concepts that are needed in information modelling. Instead of using existing ontology languages, we have developed our own language called Layer0. Layer0 is similar to OWL Full (Web Ontology Language) [9], but while OWL uses open-world definitions, where ontologies can contribute new concepts to other ontologies, Layer0 uses closed world semantics. This enables more stable environment, where concepts remain the same. Other Layer0 specific features are separation of real-time data from modelling data, which is used with simulation engines.

Our purpose is to distinct graphical model of a plant from its simulation models. At the same time we need a way to describe 3D graphical models of plant equipment and their animations. Instead of solving these separately, it is beneficial to use common representation for 3D graphics. We use X3D [10] specification as a basis for our ontology. Using ontology gives better ways to describe relationships between objects and properties than X3D; therefore its concepts are not just copied to ontological format. Basic design principle in 3D modelling ontology is that it supports only those features that are necessary, and other features can be added by creating new ontologies. This goes according to the ontology design principles [1], where minimal ontological commitment is considered as good feature that enhances re-usability of ontologies. Therefore created base ontology for 3D graphics contains only minimal set of the features that X3D supports.

Created 3D graphics ontology is used as basis for CSG modelling ontology (Constructive Solid Geometry) and 3D plant modelling ontology. CSG modelling is used for describing shape of equipment, and while it restricts freedom for 3D modelling compared to polygon modelling or B-Rep modelling (Boundary representation), it provides all necessary features for equipment modelling.

Similarly as 3D graphics ontology, concepts of 3D plant modelling ontology are based on previous taxonomies. Here we use combination of names and concepts from commercial plant modelling products and from ISO 15926 [11,12], but in very reduced form. For example, ISO 15926 defines thousands of concepts that are related to process industry, but using all of them would complicate our design without giving any real benefits. The plant modelling ontology is extended from the 3D modelling ontology, giving it 3D modelling capabilities.

Since the main purpose 3D plant modelling is to act as a user interface to process simulation, the simulation model must be generated according to the 3D model. Our purpose is not to create our own simulator, but instead, in the long run, use existing simulators, for instance simulators developed at VTT [13]. The principle here is that an ontology, which describes data structures of a simulator, is created for each used simulator, and then, with mappings, system will automatically generate simulation model when the user creates 3D model. As general, mappings can be divided to one-to-one mappings, where single concepts are linked to each other. For example diameter of pipes in 3D model is linked to diameter in simulation model. More difficult cases are one-to-many and many-to-many mappings, which link a group of concepts to each other. One good example is that when 3D model is linked to simulation model, multiple pipelines in 3D could be described as single connection in simulation model. More detailed discussion can be found in [14].

4. Visualizations

The most intuitive way to visualize behaviour of equipment is either mimicking their natural behaviour or to use colours to represent some information about them.

For instance, visualizing tank fill with an animation that changes liquid's height inside the tank is natural to every user and requires no further explanation (Figure 1). Sometimes this is not enough, for example, while status of a valve can be visualized by animating the position of its handle, the handle will most likely be too small to be seen from long distances. Because understanding of the behaviour of the simulated process requires that some part of the process can be seen at the same time, it may be more practical to visualize state of the valve by changing its colour. Since the colour of equipment can visualize multiple different properties, the user must be responsible for binding the animation to a simulation value. Animations bound to natural features are different, for instance, the fluid height changing animation is only useful for visualizing fluid height.

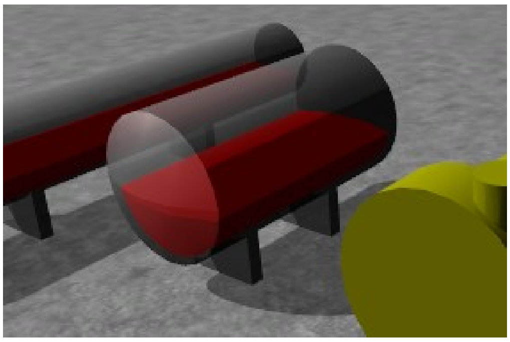


Figure 1. 3D model of a tank visualizes liquid level.

Animating just equipment will not bring complete visualization of a process plant: also flows inside pipes are in critical part. Therefore we need means to visualize flows. Similarly as with equipment, flows may have multiple properties that the user may want to visualize. The most common of them is probably mass flow, but other properties, like temperature and pressure are also important. Various techniques exists for flow visualization in 2D and in 3D [15], but most of them are not usable in visualizing a flow in pipe, which is essentially 1D flow in 3D space. Only techniques that we considered useful were glyphs and particle animations. Glyphs have proven useful in depicting spatially complex, multivariate data, because their graphical properties, such as colour and size, can be bound to visualized data. They can be seen in weather forecasts in TV, where arrows indicate wind direction and temperature. Particle animations are probably more realistic visualization of flow, since flow velocity can directly be mapped to particle velocity, it requires constant updates of positions of particles, which can be computationally demanding.

Figure 2 shows overall data model, including a 3D plant model, references from plant model to equipment models in a library, and a simulation model generated from the plant model. The figure also shows how animations can be chosen for each piece of equipment and how their animation can be bound to simulation values.

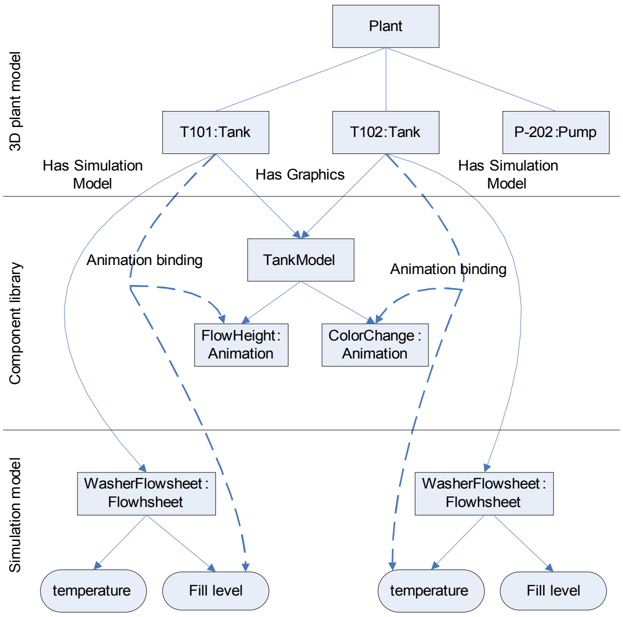


Figure 2. Overall data model. Model is divided to 3D plant model, component library containing animatable models of equipment, and simulation model.

5. Implementation Environment

Simantics is ontology based modelling and simulation platform developed (and is still under development) by Semantics Models research team at VTT. Design principles and history of the platform can be found in [16]. One of the key concepts of Simantics platform is separation of modelling data, simulation algorithms, and real-time data access. With this separation, the platform is capable of supporting multiple different models and multiple different simulation algorithms, with their own data structures, at the same time.

Simantics platform is multi-user modelling environment, hence it consist of two components: client application and server (Figure 3). Server, called ProCore, is decentralized, versioning triple storage, where all client applications are connected. Client application, called ProConf, is built on top of Eclipse RCP (Rich Client Platform) [16].

To support modelling, Simantics uses semantic graph, and its base ontology is previously described Layer0. It also contains set of foundation ontologies that provide basic concepts required in simulation and modelling, but were separated from Layer0. These include math, equations, units, structural modelling, and mapping. Math-ontology contains currently only complex numbers, Units-ontology contains units and conversion functions between them, and it allows user interface to use any units that user wants to use without breaking editors or simulators. Equations-ontology is extensible mechanism for handling mathematical equations. For instance, it is used in 3D modelling for parameterizing equipment models; A tank may have diameter and length properties that user can change and geometric representation of the tank is changed accordingly. Structural modelling ontology provides mechanism for reusing instance data of models, while providing unique property values for re-used parts. It is critical part for reducing required storage space, especially with simulation models, which typically contain complex property definitions.

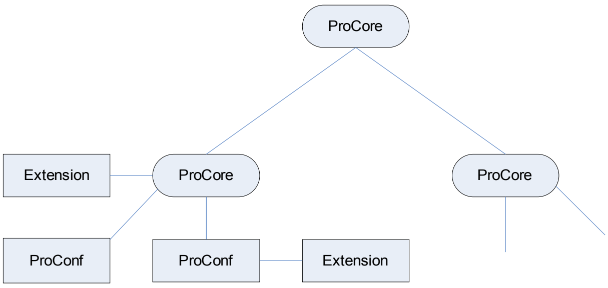
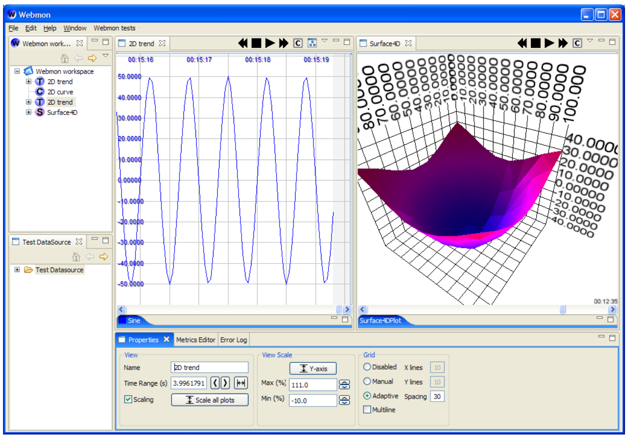


Figure 3. Architecture of Simantics Platform

Eclipse, which is used for user application, is plug-in based; core components contain only OSGi-based plug-in managers, which are capable of loading and unloading plug-ins when needed. Whole Eclipse-based application is just a set of plug-ins, which use each other's services to provide functionality that user needs. Plug-ins can use services of other plug-ins two ways: either extending other plug-in and its classes, or adding new functionality to existing plug-in using Extension / Extension Point mechanism. For ontology based programming Eclipse's plug-in architecture suits perfectly because the editors related to an ontology can be loaded only if the ontology is used, which decreases memory footprint of the program, when the user needs only few of the all available ontologies to do his tasks. An other way, how Eclipse's plug-in architecture is used, is that when an ontology extends concepts of another ontology, it can provide code that can handle those extended concepts by using the Extension Point mechanism.

For simulators, platform provides Extension mechanism. Extensions are client applications to ProCore, similarly as ProConf, but they are not tied to Eclipse and do not contain user interface. The most common use for the extension mechanism is simulators; extensions can be either registered to ProCore and it is able to launch them when needed, or extensions can be attached as part of local ProConf installation. Simulation data transfer between extensions, ProCore, and ProConf is done with Value Sets, real-time data access interface of Simantics platform. Value Sets are separated from triple database, and therefore are capable of transferring much larger amount of data.

Since ProConf is used for simulation, data visualization capabilities are one of required features of such platform. ProConf contains data visualization package called Webmon (Web technology based process data monitoring tool). Webmon is extensible visualization package, where new visualizations and data connectivity capabilities can be added as plug-ins. Currently implemented visualizations are 2D trend, binary signal, 2D curve, 3D trend, 3D surface, and 4D surface (Figure 4). Webmon can retrieve data from OPC XML-DA servers [17], import and export ASCII-based files, typical in automation systems, import data from and export data to historian server. Similarly capability to read data from Simantics' ValueSets has been added to Webmon, which allows it to visualize any simulation that is run in Simantics environment.



View
Name: 2D trend
Time Range (s): 3.9961791
☑ Scaling
Scale all plots

View Scale
Y-axis
Max (%): 111.0
Min (%): -10.0

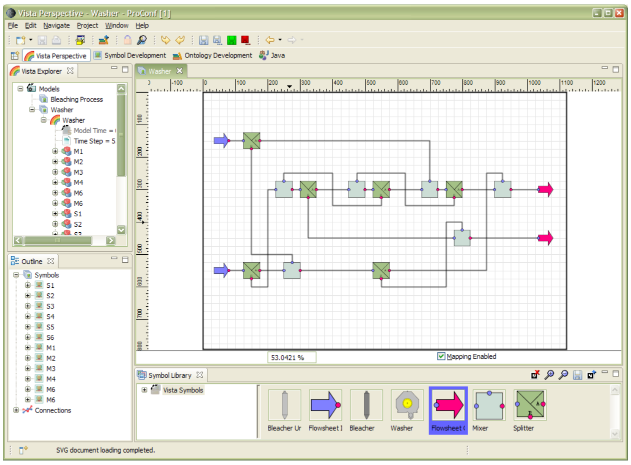
Grid
○ Disabled X lines 10
○ Manual Y lines 10
● Adaptive Spacing 30
□ Multiline

Figure 4. Data visualization component of Simantics.

6. An Example Case: Bleaching Line

As a practical use case, Process Editor is used for modelling a bleaching line that is part of a pulp mill. Own ontology is developed for this case. Ontology contains equipment that is needed to model the process (bleaching tower and washer) and mappings that generated simulation model from 3D plant model. Simulation model itself is divided to multiple ontologies, one describing basic flowsheet, one describing concepts of multiphase chemistry, and one specific to this simulation case, dubbed as Vista ontology. Name Vista comes from the name of project where multiphase chemistry simulation algorithms were developed [18].

Before mapping between 3D plant model and simulation model could be created, simulation models had to be linked to equipment models. While simulation model of a bleacher were implemented with a custom, algorithm [19], simulation of a washer was handled with more generic model. Therefore its simulation model had to be modelled with diagramming tools (Figure 5) before ontology for the test case could be created. The actual simulation model is generated by cloning flowsheets attached to equipment, and then linking ports in flowsheets to nozzles when the nozzles are connected to each other with pipelines. As a summary, one piece of equipment is always mapped to one flowsheet, and one pipeline to one flow.



Model Time =
Time Step = 5

SVG document loading completed.

Figure 5. Multiphase chemistry model of a washer modelled with ProConf's diagramming tools [20].

Figure 6 shows visualization of the bleaching line, where particle effects are used for visualizing flows in pipes: colour of particles show cleanliness of the pulp and velocity of the particles is mapped to mass flow in the pipes. Effect of the first reactor can bee seen right from the image: particles are red when they enter the reactor and particles coming out of the reactor are violet. Difference of mass flow rates is much harder to see, even when pulp's mass flow is 190 Kg/s and water's mass flow is 295 Kg/s.

The problem of particle effects shows in the figure; particles are very hard to see at large distances. Another problem, which cannot be seen in the figure, is frame rate dependency of the particle effects: When computer cannot render visualization fast enough, not only seeing velocity of mass flow is hard, but also seeing direction of mass flow is impossible. This problem was hit when visualization was used in a laptop that used power saving mode, and its graphics processing capability was significantly reduced.

While particle effects give rather vague information about the simulation, text based monitors, which appear in the figure, can show precise numerical data. When the user enables monitors, simulation data from equipment that user is pointing is shown. Monitors can be configured, and here it is set to show mass flow rate, pH, and Kappa value of flows.

Test case showed that implemented animation features are not enough for all cases. The simulation model of reactors can provide Kappa in multiple points of the reactor. Simple colour change animation can change the colour of the whole object, and cannot create a colour gradient over the surface of model. Now Webmon's 2D curve and 3D trend plots were the only way to visualize kappa value's change inside reactors.

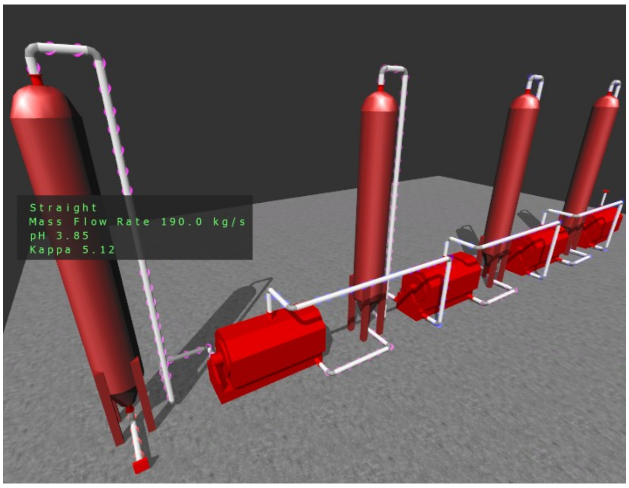


Figure 6. 3D model of a bleaching line with visualization of simulation results.

7. Future work

One of the main challenges for ontology based simulation environment in the future is scalability. As the example case of bleaching line shows, ontology based approach creates large amount of triples even for simple models. The future vision for Simantics platform is to use it as a user interface for VTT's own process simulation solvers, Apros and Balas. This sets a challenging goal for scalability: currently Apros has been used for simulating nuclear and conventional power plants and paper and pulp mills. These models may contain hundreds, if not thousands of equipment, and many kilometres of piping. When all that is described in ontological format, database will contain several hundred million triples. Therefore the example test case is just a tiny fraction of what the system should be capable of handling.

Work for this has already been started. Unnecessary relations have been removed from Layer0, and that reduces amount of triples needed by the models. Caching algorithms of ProConf have been re-implemented: Now it contains only one cache and it can also release cached triples. There has also been discussion of packing properties; instead of storing all relations, structured property is stored using arrays. This would reduce amount of triples especially with simulation models. Also clustering capability for ProCore has been implemented. This allows distribution of triples to multiple servers, reducing workload of one server.

If the scalability issue for triples can be solved, the next limit will be graphics. Designed caches of ProConf will allow it to handle larger models that will fit to the computer's memory. When a whole industrial plant is modelled, that is also required from graphical editors. When user is modelling the plant, it is more practical to filter unnecessary objects from the view, but when whole plant level simulation is visualized, the case is bit different. It depends on the user and what he wants to do: he may want to simulate the whole plant but is interested in only one section of it. Then similar filtering can be used, but when the user wants to just browse the plant and see everything, dynamic loading and unloading of graphical models is required.

8. Conclusions

The industry uses process simulators e.g. for process and automation design, control system testing and for training and support of operators.. Modelling and designing applications are currently separate from the simulation applications, and that leads to manual input of configuration data to simulator. Combining the design application and simulators into a single package provides means to tackle this problem. The designer can use simulators without extra effort and without information loss in the translation process.

3D industrial plant modelling has been widely accepted by the industry, and most of the new plants are designed with some 3D modelling software. When compared to traditional 2D process simulation tools, 3D graphics provide more lifelike view of the plant that benefits designers and end users.

We studied how 3D industrial plant modelling, process simulation and visualization could be combined with ontologies. Ontology-based approach was proven useful mechanism to break coupling of the user interface and the actual simulator. The visual model is separated from the simulator model, but since both models are in the same database, simulation can be configured through the 3D model. Therefore different simulators can be easily added without changing anything in 3D modelling and visualization.

Used visualization techniques provide means to better understand the behaviour of a simulated process. When compared to traditional text based monitors and graphical trends, animations are more intuitive and faster to interpret. When more precise information is needed, older methods are better; animations should not be seen as replacement, but as additional way to help the designer.

Easy extendibility comes at its cost: semantic triple model takes large amount of space, and requires lot of caching to work fast. This creates huge memory requirements and scalability of the method is an issue that represents a future challenge for the whole Simantics platform.

References

[1] Gruber, T. R. 1995. Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum.-Comput. Stud. 43, 5-6 (Dec. 1995), pp. 907-928.

[2] Uschold, M. and Gruninger, M. 1996. Ontologies: principles, methods, and applications, In Knowledge Engineering Review, 11(2), 1996, pp. 93-155.

[3] Guarino, N. 1998. Formal Ontology and Information Systems. Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy. 1st. IOS Press. pp. 3-15.

[4] Kalogerakis, E., Christodoulakis, S., Moumoutzis, N., 2006. Coupling Ontologies with Graphics Content for Knowledge Driven Visualization. Proceedings of the IEEE Virtual Reality Conference 2006 (IEEE VR '06), pp.43-50, Virginia, USA, 25-28 March 2006

[5] Park, M., 2005. Ontology-based Customizable 3D Modeling for Simulation. PhD Thesis. University of Florida.

[6] Lehtonen, T. and Karhela, T. 2006. Ontology approach for building and visualising process simulation models using 2d vector graphics. In Esko Juuso, editor, SIMS 2006 Proceedings of the 47th Conference on Simulation and Modeling, pages 141–146. Finnish Society of Automation, SIMS — Scandinavian Simulation Society, September 2006.

[7] Rohrer, M. W. 2000. Seeing is believing: the importance of visualization in manufacturing simulation. In Proceedings of the 32nd Conference on Winter Simulation (Orlando, Florida, December 10 - 13, 2000). Winter Simulation Conference. Society for Computer Simulation International, San Diego, CA, 1211-1216.

[8] Quick, J. M., Zhu, C., Wang, H., Song, M., and Müller-Wittig, W. 2004. Building a virtual factory. In Proceedings of the 2nd international Conference on Computer Graphics and interactive Techniques in Australasia and South East Asia (Singapore, June 15 - 18, 2004). S. N. Spencer, Ed. GRAPHITE '04. ACM Press, New York, NY, 199-203.

[9] McGuinness, D.L. and van Harmelen, F. 2004. OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004. [Cited 15 Apr 2007]. Available at: http://www.w3.org/TR/owl-features/

[10] X3D Specifications. [Cited 10 Mar 2007]. Available at: http://www.web3d.org/x3d/specifications/

[11] ISO IS 15926-2:2003: Industrial automation systems and integration – Integration of life-cycle data for oil and gas production facilities – Part2: Data Model, 2003. International Organization for Standardization, Geneva, Switzerland.

[12] ISO IS 15926-4:2004: Industrial automation systems and integration – Integration of life-cycle data for oil and gas production facilities – Part4: Initial Reference Data, 2004. International Organization for Standardization, Geneva, Switzerland.

[13] Juslin, K., Paljakka, M. 1997. APROS - A Multifunctional Modelling Environment. Proceedings of the 2nd CSNI Specialist Meeting on Simulators and Plant Analysers, Espoo, Finland.

[14] Villberg, A. 2007. Design Challenges of an Ontology-Based Modeling and Simulation Environment. Master's thesis, Helsinki University of Technology, To be published in 2007.

[15] Hansen, C.D., Johnson, C.R., 2005. The visualization handbook. Elsevier, Burlington, MA. 962 pages. ISBN 0-12-387582-X

[16] Eclipse – an open development platform. [Cited 10 Mar 2007]. Available at: http://www.eclipse.org/

[17] OPC Foundation [Cited 10 Mar 2007]. Available at: http://www.opcfoundation.org/

[18] Räsänen, E., 2003. Modelling ion exchange and flow in pulp suspensions. PhD (Tech.) Thesis. VTT Processes. Espoo. 62 p. + app. 111p. VTT Publications : 495.

[19] Brink, A. Lindberg, D., Hupa, M., Louhenkilpi, S., Wang, S., Fabritius, T., Riipi, J., Härkki, J., Kangas, P., Koukkari, P., Lilja, R., Pajarre, R., Penttilä K., Kankkunen A., Järvinen, M., Fogelholm, C-J., Bergström, F., Eriksson, K. 2007. Multi-phase Chemistry in Process Simulation MASIT04 (VISTA). In Eija Alakangas and Pekka Taskinen, editors, MASI Technology Programme 2005-2009, Yearbook 2007, pages 35-46. Technology Review 207/2007. Tekes. ISSN 1239-758X, ISBN 978-952-457-367-2. Available at http://akseli.tekes.fi/opencms/opencms/OhjelmaPortaali/ohjemat/MASI/fi/Dokumenttiarkisto/Viestinta_ja_aktivointi/Julkaisut/Yearbook__2007_final_www.pdf [Cited 10 Sep 2007]

[20] Lehtonen, T. 2007. Ontology-Based Diagram Methods in Process Modelling and Simulation. Master's thesis, Helsinki University of Technology, May 2007.

Modeling and simulation of a multi phase semi-batch reactor

Anna Nyström
Mathematical Sciences, Chalmers University of Technology and
Mathematical Sciences, Göteborg University
anno@chalmers.se

Abstract

The operation of an industrial semi-batch reactor, in which the bulk chemical EHEC, ethyl hydroxyethyl cellulose, is produced, is studied and simulated. In the reactor a strongly exothermic polymerization reaction takes place followed by a slightly exothermic reaction, and we want to minimize the duration of the operation of the process. Various operational as well as quality and safety related constraints have to be met during the batch. The complete process model, derived from measurements, first principles, and reasoning about effects on molecular level, is stated. The model includes heat and mass balances of the reactor, a pressure model, models of PID controllers, the jacket and the condenser. Technical limitations, for instance maximal and minimal jacket temperature changes due to limitations in the heat exchanger, have been modeled as constraints.

The equations have been implemented in SIMULINK, MATLAB and the model predicts the process variables rather well over time. During the first reaction, the model is not able to reproduce the jacket temperature to the desired accuracy, but the other variables have acceptable predictions. An optimization problem is formulated, wherein the total batch time is minimized under the constraints of the differential algebraic equation system and other constraints originating from the process, for instance limited pump capabilities.

As a first step in optimizing the operation of the process, a series of simulations has been performed in order to decrease the total batch time. It is concluded that a 10 % shorter batch time than today is possible if the quality is discarded, and a 5 % shorter batch time can be reached while using the existing requirements for the quality.

Keywords: Semi-batch reactor, simulation, optimization

1 Introduction

1.1 Optimization of batch reactors

In a batch process, the materials are loaded, the process is initiated, and after the reactions are completed, the products are removed. Hence, the conditions within the process are changing. The technology for making a given product is contained in the product recipe that is specific to that product [18]. The recipes are typically based on heuristics and experience. By the term semi-batch we means processes in which some parts are continuous but others are of batch type. For instance, in semi-batch operation a gas of limited solubility may be fed in gradually as it is used up.

Batch reactors are popular in practice because of their flexibility with respect to the duration of the chemical reaction and to the kinds and amounts of reactions that they can process. Generally batch processes are less safe, both for people and the environment, and the variations are smaller than in continuous processes [12]. In industry, batch and semi-batch reactors are often used in the production of fine chemicals, specialties, polymers and other high value products. Batch reactors are typically used when production volumes are low, when there are many processing steps, when isolation is required for reasons of sterility or safety, and when the materials involved are difficult to handle. The plants are often small and flexible, and the raw material and the products are expensive, but they can also be used in large volumes. They are primarily employed for relatively slow reactions of several hours duration, since the down time for filling, emptying and cleaning the equipment may be about one hour.

Chemical processes are modeled dynamically using differential algebraic equations (DAEs), consisting of differential equations that describe the dynamic behavior of the system, such as mass and energy balances, and algebraic equations that ensure physical and thermodynamic relations. Batch systems are difficult to study numerically due to the fact that steady state is never reached. In addition, chemical processes are typically nonlinear. In order to improve their performance and safety conditions, batch reactors generally require knowledge about the dynamic behavior, for instance through a mathematical description of the kinetics. The development and validation of detailed dynamic models are often quite expensive, why there is, in contrast to continuous processes which have been rigorously studied, a limited availability of detailed dynamic models. For bulk chemicals, the cost of developing models is rarely taken. Instead, the operators use experience to adjust the process periodically. Verwater-Lukszo [18] addresses this issue.

Starting as a technology in applications, mainly through operations research in the sense of optimizing complex systems and phenomena, optimization gradually became an area of academic interest. In the process systems engineering, on the contrary, it has evolved from an academic interest into a technology that has and continues to make a significant impact in industry. The increased competition in industry makes process optimization a natural choice for reducing production costs, improving product quality and reducing product variability. Often the objective in optimization of batch and semi-batch reactors is economic in nature, for instance reducing operational costs. Typically, operational decisions such as temperature and feed flow rates are determined from the optimization problem, and various operational constraints are considered. The practice of, and optimization challenges in, batch chemical industry is addressed by Bonvin et al. [4]. Overviews of the research on optimization of batch reactors until 1998 can be found in Rippin [14] and Bonvin [3]. Generally, the industry has a limited acceptance for optimization based techniques for the determination of operational profiles: to develop and validate detailed dynamic models is often considered too expensive to be motivated.

1.2 Optimal control problems

A general continuous optimal control problem, with the control variable u, and state variable x, is written as follows:

$$\text{minimize} \qquad f = \int_0^{t_f} L(x(t), u(t), t)dt + \phi(x(t_f)), \tag{1a}$$

$$\text{s.t.} \qquad h(\frac{dx}{dt}(t), x(t), u(t), t) = 0, \tag{1b}$$

$$g(x(t), u(t), t) \leq 0, \tag{1c}$$

$$x(t) \in X, u(t) \in U, \tag{1d}$$

$$x(0) \in X_0, x(t_f) \in X_f, \tag{1e}$$

$$t \in [0, t_f]. \tag{1f}$$

Minimum-time problems, which is the type of optimization problem we have in this work, are optimal control problems in which it is required to go from some initial state to some terminal state in a minimum amount of time.

$$minimize \qquad t_f \tag{2a}$$

$$s.t. \, for \; t \geq 0 \quad h(\dot{x}(t), x(t), u(t), t) = 0 \tag{2b}$$

$$g(x(t), u(t), t) \leq 0 \tag{2c}$$

$$x \in X, u(t) \in U \tag{2d}$$

$$x(0) \in X_0, x(t_f) \in X_f \tag{2e}$$

Optimization problems can be classified in terms of continuous or discrete variables. For continuous problems an important distinction is whether the problem is differentiable or not. Another important distinction, for both types of problems, is whether the problem is convex or nonconvex, since the latter may give rise to multiple local minima.

1.3 Dynamic optimization methods

A continuous optimal control problem, Equation (1), can be solved either by Calculus of variations (indirect methods) or by applying some level of discretization that converts the problem into a discretized problem (direct methods) [2]. If the optimization will be performed in combination with an existing simulator, a direct sequential method is the first choice.

The variational approach, resulting in indirect methods, is based on the solution of the first order necessary conditions for optimality, that are obtained from Pontryagin's Maximum Principle [13], which results in a Two Point Boundary Value Problem (TPBVP). This TPBVP can be solved with different approaches, including single shooting, invariant embedding, multiple shooting, or some discretization method such as collocation on finite elements or finite differences [6].

In direct methods, the problem is parameterized by a finite number of parameters, transforming the continuous optimization problem to a discretized optimization problem. These methods use NLP solvers, and can be divided into two groups, sequential and simultaneous strategies. A general drawback is that the quality of the solution depends on the parametrization. In simultaneous methods all variables are discretized, and the dynamic equation is solved implicitly, simultaneously with the optimization problem. This results in large nonlinear optimization problems, which require specialized methods. The simultaneous methods couple the DAE to the optimization problem, and the DAE is solved only once at the optimal point. An advantage of these methods is that they are applicable to general problems. The methods are advantageous for problems with path constraints and also for problems where instabilities occur for a range of inputs, since they are able to suppress unstable nodes by enforcing the appropriate boundary condition. A disadvantage is the need to solve large nonlinear problems. A review of simultaneous methods can be found in [6]. Examples of methods that can be used to solve the NLP are Successive Quadratic Programming (SQP) [7], single shooting, multiple shooting [10] and direct shooting method (which is a bridge between direct and indirect methods), invariant embedding, global orthogonal collocation, orthogonal collocation on finite elements [5, 19], and moving finite elements [2].

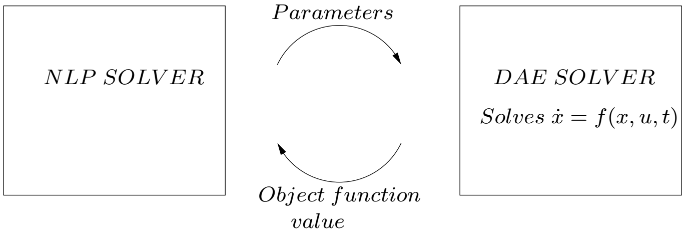


Figure 1: In sequential methods, each iteration the NLP solver sends values of the control parametrization to the DAE solver, which solves the equation $\dot{x} = f(x, u)$. This produces a value of the objective function which is used by the NLP solver to find the optimal parameters in the control parametrization.

In sequential methods, also called Control Vector Parametrization (CVP) [1], the control vector u is discretized (piecewise constant, piecewise linear or piecewise polynomial), and the dynamic equation $\dot{x} = f(x, u)$ is solved explicitly in each optimization step, with well-known integration, which means that also a non-optimal solution is feasible, Figure 1. Advantages of these methods are that they can handle rather large problems without large scale optimization techniques, and that a non-optimal solution is feasible. These methods run into problems, if the optimization algorithm requires gradient information, because standard DAE solvers are not usually written to provide parametric sensitivities of the solution, or, if provided, they might not be accurate enough for highly nonlinear models.

1.4 Objectives

In this study, a bulk chemical is produced in a semibatch reactor. We want to show that also small models can be used to gain knowledge when optimizing a semibatch process. In our model, two PID- regulators are included, in addition to mass and energy balances. Our semibatch process works reasonably well today. However, the set point temperature and set point pressure profiles are more or less based on experience, and measurements during batches show that with the existing equipment, these profiles are quite difficult to follow. Also, measurements on the resulting product implies that the prevailing settings gives a product with larger variations of the properties between batches than wanted. If the temperature can be better controlled the variation between batches can be decreased, with a more homogeneous product as a result. In this work, which is a first step in optimizing the operation of the process, we minimize the total batch time, without changing the slopes in the set point temperature profile or set point pressure profile and keeping the existing equipment.

We handle the dynamics of the cooling system by adding constraints on the dynamics and we assume that the control system is fast enough in order to implement the new trajectories. We do not have access to sensitivity information for the objective function and the constraint states with respect to the free optimization variables, as Abel et al. [1] do. Compared to Khuu et al. [9], in which the reactor content is preheated to the reactor temperature before EO is added, we include the heating in our study. Also we have limited access to reaction kinetics.

In this work we focus on a sequential method for optimization.

2 The model

2.1 Process description

In the reactor alkali cellulose is reacted with ethylene oxide (EO) and gaseous ethyl chloride (EC) under pressure and increased temperature, to produce EHEC, (Ethyl HydroxyEthyl Cellulose), a thickener in water-based systems. The reactions are described by:

$$\begin{aligned}
\mathrm{Cell-OH+OH^-} &\rightarrow \mathrm{Cell-O^-+H_2O} \\
\mathrm{Cell-O^-+nEO} &\rightarrow \mathrm{Cell-O-(EO)_n^-} \\
\mathrm{Cell-O-(EO)_n^-+EC} &\rightarrow \mathrm{Cell-O(EO)_n-Et+Cl^-} \\
\mathrm{Cell-O^-+EC} &\rightarrow \mathrm{Cell-O-Et+Cl^-}
\end{aligned}$$

The EO is added to the reactor as a liquid and vaporizes. In our process, the distribution of phases inside the reactor is not known. Possibly three phases are present, and it is also not known in which phase the reactions take place.

The temperature in the reactor is controlled by two separate systems; a condenser and a jacket, cf. Figure 2. The water in the jacket is circulated through a cold water inlet, through a heat exchanger to the reactor jacket, passing by an outlet where surplus water can be removed to keep pressure constant on the way back to the cold water inlet. The flow through the jacket system, F_j, is kept constant at a high level with a pump. In the heat exchanger steam is used as heating agent and the temperature of the cold, so called raw, water varies slightly during the year. The flow in the jacket is much higher than in the condenser. Two PID-regulators control the temperature of the jacket flow and the flow through the condenser, and the temperature and the pressure in the reactor is measured.

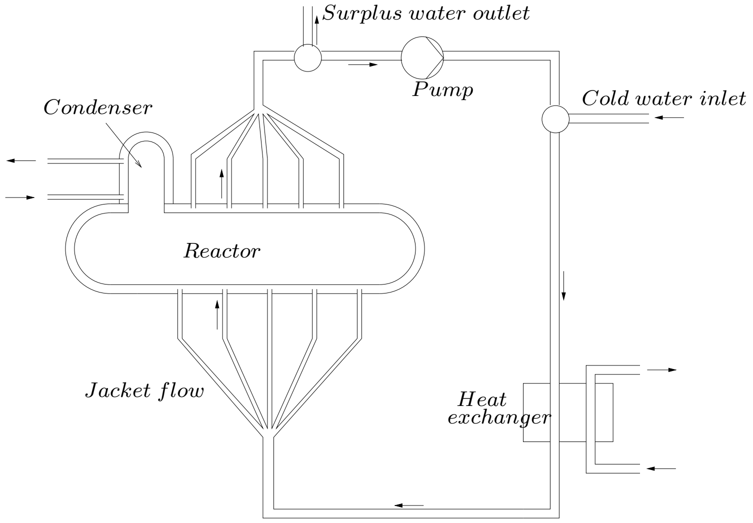


Figure 2: A process schematic, showing the jacketed reactor with the condenser to the left. The jacket system consists of the reactor jacket, outlet, pump, inlet and heat exchanger.

2.2 Model equations

The complete process model, derived from measurements, first principles, and reasoning about effects on molecular level, is stated below. All variables are scaled and uncertainties are not included in the model. The overall mass balance over the reactor, with units kg:

$$m_{tot}(t) = \sum_{\substack{All \\ Substances\ i}} m_i(0) + \int_0^t \dot{m}_{EO}(\tau)\, d\tau. \tag{3}$$

The overall heat balance, with units kJ/s, over the reactor, assuming $C_{P,tot}$ is constant:

$$\begin{aligned}\frac{\partial}{\partial t}(m_{tot}(t) C_{P,tot} T_r(t)) &= Q_{in}(t) - Q_{out}(t) + Q_r(t) \\ &= Q_{in}(t) - Q_j(t) - Q_c(t) + Q_r(t),\end{aligned} \tag{4}$$

where heat content in the EO-addition is:

$$\begin{aligned}Q_{in}(t) = \frac{\dot{m}_{EO}(t)}{M_{EO}} &[\Delta H_{vap}^{EO}(T_0) \\ &+ C_{P,EO}^{gas}(T_r)T_r(t) - C_{P,EO}^{gas}(T_0)T_0],\end{aligned}$$

with

$$\begin{aligned}\Delta H_{vap}^{EO}(T) &= A(1 - \frac{T}{B})^C, \\ C_{P,EO}^{gas}(T) &= a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4.\end{aligned}$$

The heat transfer Q_j from material inside the reactor to the jacket water is (the units are kJ/s):

$$Q_j = k_j A_j (T_r(t) - \overline{T}_j(t)),$$

and to the condenser average temperature:

$$Q_c = U_c(t) A_c (T_r(t) - T_{c,in}(t)),$$

where

$$U_c(t) = \frac{1}{\frac{1}{k_c A_c} + \frac{1}{2F_c(t)\rho_w C_{P,w}}}.$$

The reaction heat Q_r is given by

$$Q_r(t) = \Delta H_{EO} r_{EO}(t) + \Delta H_{EC} r_{EC}(t),$$

with the reaction rates r_i:

$$\begin{aligned}\frac{\partial c_{\text{EO}}}{\partial t} &= -\, r_{EO'} = -k_{EO}(T) c_{\text{EO}}, \\ \frac{\partial c_{\text{EC}}}{\partial t} &= -\, r_{EC'} = -k_{EC}(T) c_{\text{EC}} c_{\text{RO}^-}, \\ \frac{\partial c_{\text{RO}^-}}{\partial t} &= -\, r_{RO^-{}'} = -k_{RO^-}(T) c_{\text{EC}} c_{\text{RO}^-}.\end{aligned}$$

Here the reaction constants k_i are given by Arrhenius equation $k_i(T(t)) = A_i exp(-\frac{E_i}{T(t)})$.

We include two PID-regulators, each described by the following equation, with all variables expressed in percents:

$$u = K\left(r - y + \frac{1}{T_I}\int (r - y)\, dt - T_D \frac{dy}{dt}\right). \tag{5}$$

In the condenser, $u = F_c/F_{c,Max}$, $y = P_r/P_{Max}$ and $r = P_{set}/P_{Max}$ and in the jacket $u = T_j/T_{Max}$, $y = T_r/T_{Max}$ and $r = T_{set}/T_{Max}$.

In order to fully describe the system we also need an expression for the reactor pressure:

$$\begin{aligned}P_r &= P_{EO} + P_{EC}, \\ P_{EO} &= P_{EO}^o(T)(\alpha_1 n_{EO}(t) + \beta_1), \\ P_{EC} &= P_{EC}^o(T)(\alpha_2 n_{EC}(t) + \beta_2)(\gamma_1 \lambda_{EO}(t) + \delta_1),\end{aligned}$$

where $P_i^o(T_r)$, the vapor pressure for species i, and $\lambda_{EO}(t)$, describing the progress of the EO-reaction, are given on the form

$$P_i^o(T_r) = 10^{(A+\frac{B}{T+C})},$$
$$\lambda_{EO}(t) = \frac{n_{EO,reacted}(t)}{n^{max}_{EO,reacted}}.$$

The set point curves for temperature and pressure are shown in Figure 3, as well as the original EO-profile.

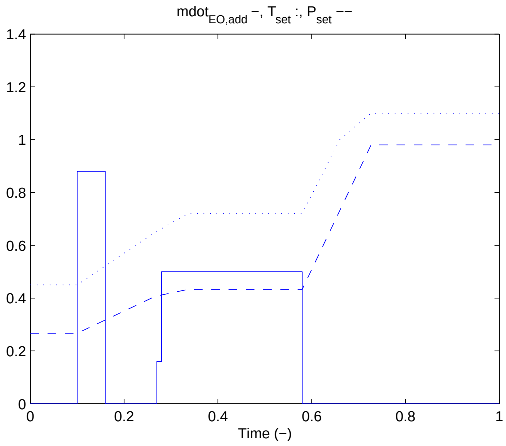


Figure 3: The figure shows the set point curves for temperature T_{set}, and pressure P_{set}, and the original EO-addition $\dot{n} = mdot_{EO,add}$.

This system is simulated in SIMULINK, with the result shown in Figures 4 and 5. As can be seen, all simulated curves but the jacket temperature follow the real curves quite well. The simulated jacket temperature rises in the beginning, as does the real jacket temperature, but then, during the time interval of 0.3–0.7 time units, it does not fall far enough compared to the real jacket temperature. This error, in turn, affects all the other variables in this time interval: the simulated reactor temperature is somewhat lower than the real one, the simulated pressure is somewhat higher than the real one, and there are twitches in the simulated concentrations c_{EO} and c_{EC} which are not present in the real concentrations. This error is possibly due to two major things: first, in the simulations we use constant PID parameter values instead of time variable ones, and second, the heat transfer between the reactor content and the jacket water is deliberately modeled with rather low accuracy. In this work, the focus has been on making a simple model to use as a start for further modeling work, and this is why we have chosen to use these approximated values. The relation between the effects of the approximations is difficult to say anything about without further information. In the future, these things should be looked further into.

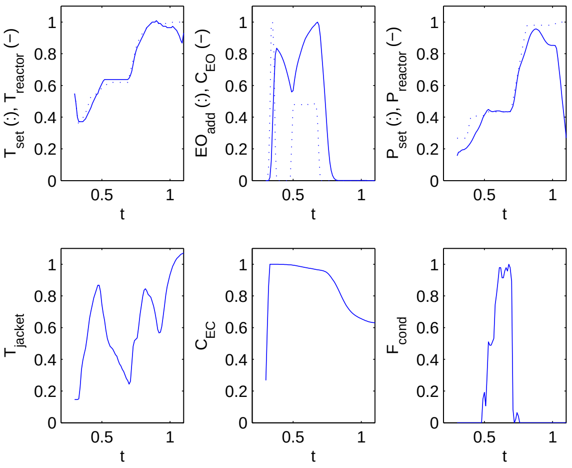


Figure 4: The figure shows real data from the process.

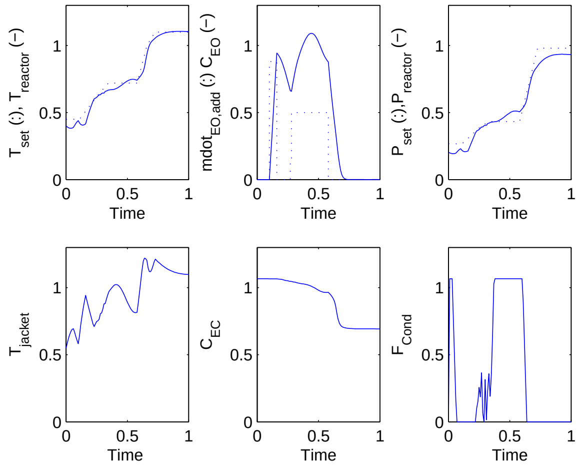


Figure 5: The figure shows a simulation in Simulink using the model equations above. As can be seen, all simulated curves but the jacket temperature follow the real curve quite well.

2.3 Optimization problem

We want to minimize the total batch time and hence use the complete model above to formulate the dynamic minimum time problem, Equation (2). It has the form

$$\begin{array}{ll} \text{minimize} & t_f, \\ & \\ \text{subject to} & h(\dot{x}(t), x(t), u(t), t) = 0, \\ & g(x(t), u(t), t) \leq 0, \\ & x \in X, u(t) \in U, \\ & x(0) \in X_0, x(t_f) \in X_f, \\ & t \geq 0. \end{array}$$

In this study, we consider the reactions starting with the addition of ethylene oxide, including the heat transfer from the reactor, but not loading, preheating, cooling and emptying. We define the end of the batch as the time t_f characterized by the concentration of EC dropping below a limit c^0_{EC}. We optimize the operation by finding an optimal way of adding the ethylene oxide to the reactor, $\dot{m}_{EO}(t)$.

In this formulation, the system equations h are given by the model equations above, summarized as

$$\begin{aligned}
\dot{T}_r &= f_1(T_r, n_{EO}, n_{EK}, n_{RO^-}, T_j, F_c, \dot{m}_{EO}, t),\\
\dot{T}_j &= f_2(T_r, n_{EO}, n_{EK}, n_{RO^-}, T_j, \dot{m}_{EO}, t),\\
\dot{F}_c &= f_3(T_r, n_{EO}, n_{EK}, n_{RO^-}, F_c, \dot{m}_{EO}, t),\\
\dot{n}_{EO} &= f_4(T_r, n_{EO}, t, \dot{m}_{EO}),\\
\dot{n}_{EK} &= f_5(T_r, n_{EK}, n_{RO^-}, t),\\
\dot{n}_{RO^-} &= f_6(T_r, n_{EK}, n_{RO^-}, t),
\end{aligned}$$

with the constraints g (both technical and natural)

$$\begin{aligned}
&0 \le T_r,\\
&c \le T_j \le C,\\
&m \le \frac{\partial T_j}{\partial t} \le M,\\
&0 \le F_c \le D,\\
&l \le \frac{\partial F_c}{\partial t} \le L,\\
&0 \le n_{EO} \le n_{EO,max},\\
&0 \le n_{EK} \le n_{EK,max},\\
&0 \le n_{RO^-} \le n_{RO^-,max}.
\end{aligned}$$

The function $\dot{m}_{EO}(t)$ is parameterized by dividing the batch time into $J = 150$ equal time elements and assuming the flow rate to be constant in each of these elements. To prevent too fast changes in flow rate (which originates from limited capability of pumps etc.), two adjacent elements are not allowed to differ more than a preset value, α:

$$\begin{aligned}
\dot{m}_{EO}(t) = m_j, \quad t \in [t_{j-1}, t_j),\ j = 1, \ldots, J\\
|m_j - m_{j-1}| \le \alpha, \quad j = 2, \ldots, J.
\end{aligned}$$

To prevent too high temperatures and too large temperature gradients, which are not allowed in the real process, but are accepted by MATLAB in simulations, a penalty is added to the objective function:

$$f_{obj} = t_f + P. \tag{6}$$

As a first step, used in the simulations below, a discontinuous penalty is used, namely

$$P(T_r) = \begin{cases} p, & |T_{set} - T_r| \ge \beta,\\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $p \gg 0$. It should be mentioned that when performing an automatic optimization, a continuous penalty is preferred, for instance the following

$$P(T_r) = p \max\{0, |T_{set} - T_r| - \beta\}. \tag{8}$$

The penalty P is added in those simulations where the control system does not manage to keep the reactor temperature close enough to the set point temperature T_{set}. The constant β is determined by the ability of the PID controllers to control the reactor temperature and using a large value means that we rely on the jacket to be good at controlling the reactor temperature. Since a DAE system is just a model of the real system, it contains errors, and the constant β is also used as an additional error control of the reactor heat transfer and the control system. If the error in the control system and/or the reactor heat transfer is large and the simulated system is able to control the reactor temperature, a small value of the constant β in Equations (7) and (8) decreases the risk of escalating reactor temperature in the real reactor.

3 Simulations

In this study, the focus has been on finding a mathematical model for the batch time, and initially not on the quality of the end product. Still, we need to use information about the end product quality to determine which profiles are acceptable: if a large portion of the EO is added prior to the wanted reaction temperature is reached, this may result in a worsened product quality. Below, this is judged by the user. Since, prior to this study, there is a lack of knowledge about how the quality is affected by different temperature profiles, adding this to the mathematical model would have been to comprehensive. The acceptable size of the portion of EO added prior to the wanted reaction temperature and how the quality is affected by the reaction temperature are interesting future research topics. To get a proper objective function, the quality should be added to the optimization problem, for instance as a penalty in the objective function just as large temperature gradients above.

Below we use the objective function defined by Equation (6), including a discontinuous penalty according to Equation (7) with $p = 10$. This means that an objective function value above 10 has been penalized. For the original EO profile in Figure 4, this objective function value is $f_{obj} = 0.61$.

A series of simulations have been run, and in each step we change as little as possible, to clearly notice the effects. We compare each simulation with the original profile used in the process today. All simulations have in common that the slopes and the plateau values of the set point curves T_{set} and P_{set} are the same as in the original profile. The starting time for the second set point rise is determined by the end of the EO addition. A reason for not allowing changes in slopes for the set point curves, is that if the reactor temperature (or pressure) rises too fast, there is a potential risk for escalating reactor temperature, which is not wanted; it is even dangerous. Of course, varying the set point curves would be interesting to include in a future optimization with a more accurate model.

We have relaxed the starting times for the first rise of the set point curves by delaying the start of the EO addition, and have run the following series of simulations with piecewise constant EO-profiles:

1. Using an EO-profile with only one plateau, at a starting level being the same as the second plateau in the original profile, varying the delay and level.
2. Using the original EO-profile with two plateaus, varying the levels proportionally and varying the delay.
3. Using a profile with three plateaus varying only the delay.

Table 1: Objective function values for different simulations, with one, two or three plateaus and varying levels and delays.

	One plateau				**Two plateaus (original profile)**									**Three**
Delay	**Level rise** $l = \frac{New\ levels}{Original\ levels}$				**Level rise** $l = \frac{New\ levels}{Original\ levels}$									**plateaus**
(time units)	1.00	1.20	1.60	2.00	1.00	1.10	1.20	1.30	1.40	1.50	1.60	1.70	1.75	
0.00	0.57	0.56	10.58	10.55	0.61	0.60	0.58	0.58	0.58	0.58	0.58	0.58	10.58	0.55
0.05	0.60	0.57	10.62	10.58	0.65	0.63	0.62	0.62	0.62	0.61	0.61	0.60		0.60
0.10	0.63	0.60	10.62	10.62	0.69	0.68	0.66	0.66	0.66	0.64	0.64	0.62		0.63
0.15	0.67	0.64	0.64	10.65	0.74	0.72	0.71	0.71	0.69	0.68	0.68	0.66		0.65
0.20	0.71	0.68	0.68	10.69	0.78	0.77	0.75	0.75	0.73	0.72	0.72	0.70		0.69
0.25	0.76	0.72	0.72	10.73	0.83	0.81	0.80	0.79	0.78	0.76	0.76	0.74		

4 Results

The different objective values for combinations of variations in delay and variations in levels are shown in Table 1. Using a profile with one plateau as in Figure 6, it is seen that for those simulations in which the penalty is not activated, the batch time decreases with increasing levels, which is natural since higher concentrations of EO in the reactor results in a higher reaction rate according to the reaction rates. But increasing the EO-addition too much results in a too large heat production for the control system to handle, which is seen in that the objective function value is penalized and $f_{obj} > 10$. For each level rise, increasing the delay naturally increases the batch time. For the two smallest level rises (1.00 and 1.20) some of the batch times are smaller than the original profile. However, here the quality must be taken into account, since a portion of the EO is added prior to the reactor temperature reaches the wanted reaction temperature.

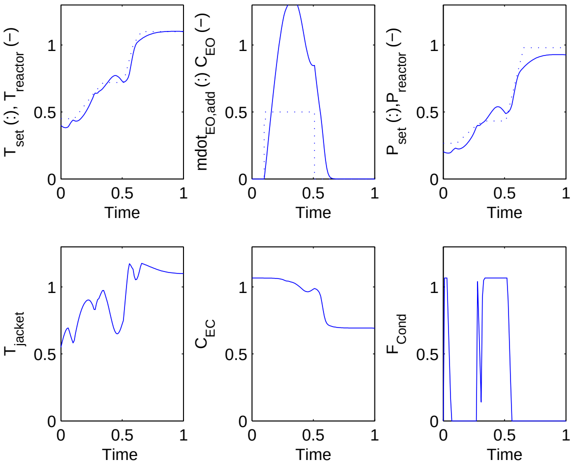


Figure 6: The figure shows a simulation where EO is added at a constant level being the same as the second plateau level in the original profile, with no delay.

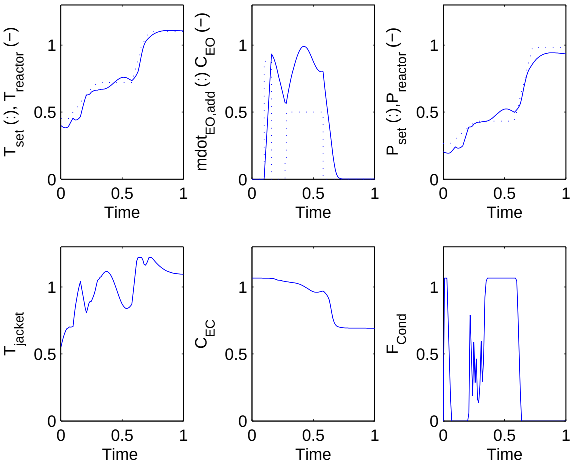
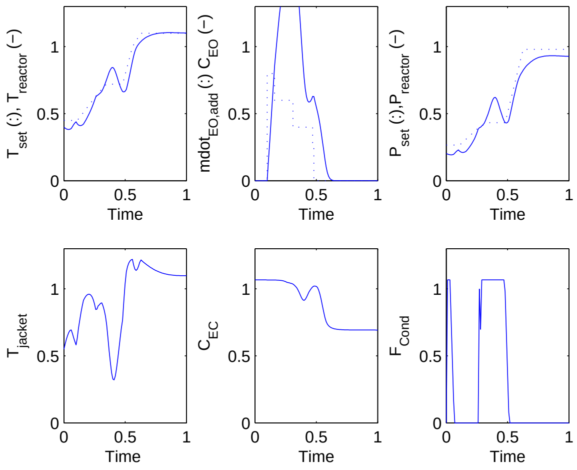


Figure 7: The figure shows a simulation with original EO profile, but where the EO addition is delayed 0.05 time units. The corresponding objective function value is $f_{obj} = 0.65$.

Using variations of the original EO-profile with two plateaus, it can be seen in Table 1, that using a 20 % rise in the plateau levels, with the delay = 0.00, it is possible to shorten the batch time to 0.58 time units, which is an improvement of 5 % compared to the original profile. As mentioned above, in these simulations we have kept the original set point curves. According to these simulations, with no delay, nothing is gained by raising the levels further. We conclude that the batch time increases when delaying the original profile. Figure 7 shows a simulation using the original profile with 0.05 time units delay, with the objective function value $f_{obj} = 0.65$.

Using the profile with three plateaus, again keeping the original set point curves, as in Figure 8, it is possible to decrease the batch time, with delays up to 0.05 time units, which correspond to 10 %. However, also here one must keep the quality in mind; in order to to determine whether these profiles give acceptable quality, more research is needed.



Figur 8: The figure shows a simulation in which the EO-profile has tree plateaus and no delay.

Summarizing, we have used piecewise constant EO profiles with up to three constant plateaus and varying levels, and simulations show that a 5 % increase in total batch time is possible, using a profile with two plateaus as in the original, but with 20 % higher levels and no delay.

Even a 10 % shorter batch time than today is possible, using a profile with three different plateau levels. However, in this profile, a large portion of the EO is added prior to when the wanted reaction temperature is reached, which may result in a worsened end product quality. In order to decide which profiles are acceptable, more research about the effect of the reaction temperature used on the end product quality is needed.

5 Future work

When optimizing the operation of a semi-batch reactor, an overall point of view is advantageous, since the decisions are of an economic character: mainly the costs of raw material, the production, and the end product quality. This should be kept in mind when performing an optimization like in this work. Shortening the batch time with for instance 10 % may in the end not decrease the total cost with more than 0.5 %.

This work is the first step in optimizing the operation of a semi-batch reactor. Steps planned to be followed include improving the process model, extending the optimization problem, performing the optimization with the aid of CVP and testing the optimal solution in the real process. We also intend to look further into other optimization techniques, such as simultaneous methods and optimization under uncertainty.

Work on improving the process model has already started, especially on achieving better values for the heat capacity of the reactor content $C_{P,tot}$. In the future, also a more complete model of the jacket system is to be added to the optimization problem. Several parameters in the model used in this work are uncertain to different degrees. Including uncertainty into the process model gives a different type of optimization problem, which demands other optimization techniques than discussed in this study [8, 11, 15, 16, 17].

As a first extension of the optimization problem, we recommend that the quality of the end product is included in the objective function of the optimization problem, following a proper mathematical modeling of this entity. From the series of simulations performed in this work, it is hard to tell which profile is the best. It is highly probable that further investigations, and especially automatic optimization, can show profiles resulting in even shorter batch times.

In an extended optimization problem, both the temperature and the pressure set point curves should be used as control variables and hence optimized over, together with the addition of the EO. The impact of the different set point curves on the total batch time in optimization is interesting to study, since the batch time depends on the reaction rates, which in turn strongly depend on the temperature. The pressure is strongly connected to the temperature due to the gas phase present and the chosen working point.

Performing a complete optimization to find a solution to the dynamic optimization problem, using for instance the control vector parametrization approach for the solution, can give useful, and perhaps unexpected, profiles. In parallel to improving the model, we intend to use the existing model when using CVP to optimize the operation of the process.

We already know that our existing model contains uncertainty, and the sensitivity of input parameters must be further investigated before the optimal solution is tested in the real process. An investigation into these uncertainties and their effects may ultimately produce a new and more robust optimization formulation of the problem. This work is also ongoing in parallel to the above mentioned work.

6 Acknowledgements

This work has been done in cooperation with Akzo Nobel Functional Chemicals, Stenungsund, Sweden. The financial support is greatly acknowledged.

Nomenclature

m_i	=	mass of substance i, kg
M_{EO}	=	molar mass for EO, kg/mol
T_r	=	average reactor temperature, K
T_0	=	temperature of added EO, K
$\overline{T}_j$	=	average jacket temperature, K
$T_{c,in}$	=	temperature of water into condenser, K
F_c	=	water flow through the condenser, m^3/s
P_r	=	reactor pressure, Pa
$\dot{n}$	=	molar flow of EO into the reactor, $mole/min$
n_i	=	moles of substance i in the reactor, $mole$
c_i	=	concentration of substance i in the reactor, $mole/m^3$
k_j	=	heat transfer coefficient from reactor content to jacket water, $W/m^2, K$
k_c	=	heat transfer coefficient from reactor content to condenser water, $W/m^2, K$
U_c	=	overall heat transfer coefficient from reactor content to condenser water, $W/m^2, K$
A_j	=	jacket area, m^2
A_c	=	condenser area, m^2
A_i	=	constant in Arrhenius equation for reaction i
E_i	=	activation energy in Arrhenius equation for reaction i
r	=	set point
y	=	measured variable
u	=	controlled variable
K	=	proportional constant for regulator
T_I	=	integral constant for regulator
T_D	=	derivative constant for regulator

References

[1] O. Abel, A. Helbig, W. Marquardt, H. Zwick, and T. Daszkowski. Productivity optimization of an industrial semi-batch polymerization reactor under safety constraints. *Journal of Process Control*, 10:351–362, 2000.

[2] L. T. Biegler, A. M. Cervantes, and A. Wächter. Advances in simultaneous strategies for dynamic process optimization. *Chemical Engineering Science*, 57:575–593, 2002.

[3] D. Bonvin. Optimal operation of discontinuous reactors - A personal view. *Journal of Process Control*, 8:355–368, 1998.

[4] D. Bonvin, B. Srinivasan, and D. Ruppen. Dynamic optimization in the batch chemical industry. *Aiche Symposium Series*, 326:255–273, 2002.

[5] A. M. Cervantes and L. T. Biegler. A stable elemental decomposition for dynamical process optimization. *Journal of Computational and Applied Mathematics*, 120:41–57, 2000.

[6] A. M. Cervantes and L. T. Biegler. Optimization strategies for dynamic systems. In C. Floudas and P. Pardalos, editors, *Encyclopedia of Optimization*, volume 4, pages 216–227. Kluwer, New York, 2001.

[7] F. A. D'Angelo, L Brunet, P Cognet, and M. Cabassud. Modelling and constraint optimisation of an aromatic nitration in liquid-liquid medium. *Chemical Engineering Journal*, 91:75–84, 2003.

[8] I. Datkov, G. M. Ostrovsky, L. E. K. Achenie, and Y. M. Volin. Process optimization under uncertainty when there is not enough process data at the operation stage. *Optimization and Engineering*, 7:249–276, 2006.

[9] S. M. Khuu, J. A. Rodriguez, J. A. Romagnoli, and K. F. Ngian. Optimisation and control of an industrial surfactant reactor. *Computers & Chemical Engineering*, 24:863–870, 2000.

[10] D. B. Leineweber, A. Schäfer, H. G. Bock, and J. P. Schlöder. An efficient multiple shooting based SQP strategy for large-scale dynamic process optimization, Part II: Software aspects and applications. *Computers & Chemical Engineering*, 27:167–174, 2003.

[11] Z. K. Nagy and R. D. Braatz. Open-loop and closed-loop robust optimal control of batch processes using distributional and worst case analysis. *Journal of Process Control*, 14:411–422, 2004.

[12] R. H. Perry and D. W. Green. *Perry's Chemical Engineers' Handbook*. McGraw-Hill, New York, 7 edition, 1997.

[13] V. Pontryagin, V. Boltyanskii, R. Gamkrelidge, and E. Mishchenko. *The Mathematical Theory of Optimal Processes*. New York: Interscience Publishers Inc., 1962.

[14] D. W. T. Rippin. Simulation of single- and multiproduct batch chemical plants for optimal design and operation. *Computers & Chemical Engineering*, 7:137–156, 1983.

[15] B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki. Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty. *Computers & Chemical Engineering*, 27:27–44, 2002.

[16] P. Terwiesch, M. Agarwal, and D. W. T. Rippin. Batch unit optimization with imperfect modelling: A survey. *Journal of Process Control*, 4:238–258, 1994.

[17] P. Terwiesch, D. Ravemark, B. Schenker, and D. W. T. Rippin. Semi-batch process optimization under uncertainty: Theory and experiments. *Computers & Chemical Engineering*, 22:201–213, 1998.

[18] Z. Verwater-Lukszo. A practical approach to recipe improvement and optimization in the batch processing industry. *Computers in Industry*, 36:279–300, 1998.

[19] V. M. Zavala, A. Flores-Tlacuahuac, and E. Vivaldo-Lima. Dynamic optimization of a semi-batch reactor for polyurethane production. *Chemical Engineering Science*, 60:3061–3079, 2005.

Identification of a best thermal formula and model for oil and winding 0f power transformers using prediction methods

Kourosh Mousavi Takami Jafar Mahmoudi
Mälardalen University, Västerås, sweden
Kourosh.mousavi.takami@mdh.se jafar.mahmoudi@mdh.se

Abstract

System identification is about building models from data. A data set is characterized by several pieces of information: The input and output signals, the sampling interval, the variable names and units, etc. Similarly, the estimated models contain information of different kinds, estimated parameters, their covariance matrices, and model structure and so on. In this paper we collected Temperature of oil and winding in 230/63kv transformer of SARI Substation and considered the winding temperature for input in the model and oil temperature for out put. After that calculated their data by MATLAB software and get a new model with the good best fit for the heat transfer from core and winding to oil. For verification of were calculated results, has been simulated the process in COMSOL Software.

Index: best fit, simulation, identification, error, model

1. Introduction

The system identification problem is to estimate a model of a system based on the observed input-output data. Several ways to describe a system and to estimate such descriptions exist.

This paper provides a brief account of the most important approaches.

The procedure to determine a model of a dynamic system from observed input-output data involves three basic ingredients:

- The input-output data
- A set of candidate models (the model structure)
- A criterion to select a particular model in the set, based on the information in the data (the identification method)

The typical identification process consists of stages where you iteratively select a model structure, compute the best model in the structure, and evaluate this model's properties. This cycle can be itemized, as follows:

1) Design an experiment and collect input-output data from the process to be identified.

2) Examine the data. Polish the data by removing trends and outliers, and select useful portions of the original data. You can also apply filters to the data to enhance important frequency ranges.

3) Select and define a model structure (a set of candidate system descriptions), within which a model is to be found.

4) Compute the best model in the model structure according to the input-output data and a given criterion for goodness of fit.

5) Examine the properties of the model obtained. If the model is good enough, then stop; otherwise go back to step 3 to try another model structure. You can also try other estimation methods (step 4), or work further on the input-output data (steps 1 and 2).

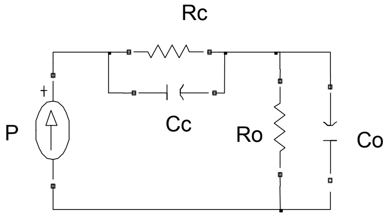
For step 2, the System Identification offers routines to plot the data, filter the data, and remove trends in the data, as well as to resample and reconstruct missing data.

For step 3, there are a variety of nonparametric models, the most common black-box input-output and state-space structures, as well as general tailor-made linear state-space models in discrete and continuous time.

For step 4, general prediction error (maximum likelihood) methods, as well as instrumental variable methods and subspace methods, are offered for parametric models, while basic correlation and spectral analysis methods are used for nonparametric model structures.

For examining the models in step 5, many functions are provided to compute and present frequency functions, poles, and zeros, as well as to simulate and predict with the model. There are also functions for transforming between continuous-time and discrete-time model descriptions.

In figure 1 a conventional thermal model is represented.



Rc: thermal resistance between core and oil

Ro: thermal resistance between oil and ambient

Cc: thermal capacitance core and winding

Cc: thermal capacitance oil

P: heat generation in core and winding per watt

Figure 1: Model of heat generation and transfer in power transformer in the above and so tank and windings&cores of power transformer, in the below.

The constructed thermal model is employed to predict the bottom-oil temperature θ_{oil-in} and top-oil temperature $\theta_{oil-out}$ that can be expressed by a set of ordinary, first-order differential equations. After comparing the matrix of the thermodynamics equations with the basic electric principles, the computation is undertaken using the following differential equations:

$$Pdt = G\theta dt + c_p md\theta \qquad (1)$$

or $$P = G\theta + {}^{Cd\theta}\!/_{dt} \qquad (2)$$

where P is the heat power as a vector, G is the heat conductance matrix, C is the heat capacitance matrix, θ is the temperature rise vector, in is the mass of a specific object, cp is the specific heat capacity, and d is the differential operator.

2. Experimental Data

A thermal test have performed on 230/63/20 kv, 250MVA power transformers, in SARI substation, Iran. The result of test have used for input data of calculation and simulations.

Table 1: transformer data from IEEE guide 1995[17]

Description	Value	
No Load (W)	78,100	
Pdc losses (W)	411,786	
Eddy losse (W)s	41,200	
Nominal Voltage	118 kv	230kv
Pdc at HST location	467	527
Eddy current losses at HST	309(0.65pu)	157(0.3pu)
p.u. height to winding HST	1	1
Temperature Rise ºC		
Rated top oil rise	38.3	
Rated top duct oil rise	38.8	
Rated hot spot rise	58.6	50.8
Rated average winding rise	41.7	39.7
Rated bottom oil rise	16	
Initial top oil	38.3	
Initial top oil duct	38.3	
Initial average winding	33.2	
Initial bottom oil	28	
Initial hot spot	38.3	
Transformer components weights (kg)		
Mass of core and oil assembly	172,200	
Mass of tank	39,700	
Mass of oil	37,887	

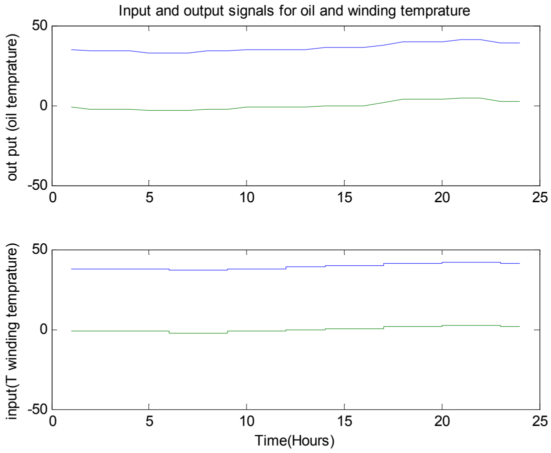


Figure2: Measured winding and oil temperature of 230/63/20kv SARI transformer no.1 (TV).

That group's of data that could not been available and hasn't been on nameplate or manual service catalog of power transformer has used by Table 1 data's, and they present the necessary data for calculation and simulations.

Table2: Temperature of oil and winding in 230/63/20kv SARI Substation at 21Feb. 2006

Temprature of oil and winding in 230/63kv Pounel Substation at 21FEB. 2006								Hours
T2				T1				
oil	winding			oil	winding			
	HV	LV	TV		HV	LV	TV	
39	40	41	36	35	42	41	38	1
38	39	40	36	34	41	40	38	2
38	39	40	36	34	41	40	38	3
38	39	40	36	34	40	39	38	4
37	38	39	36	33	38	38	38	5
37	38	39	36	33	37	37	37	6
36	37	38	36	33	36	37	37	7
38	38	38	36	34	38	37	37	8
39	38	38	36	34	39	38	38	9
40	39	39	35	35	40	38	38	10
40	39	39	35	35	40	39	38	11
41	40	41	35	35	41	39	39	12
41	40	41	36	35	41	40	39	13
42	41	42	37	36	42	41	40	14
42	41	42	37	36	42	41	40	15
42	41	42	37	36	42	41	40	16
43	42	43	38	38	43	42	41	17
43	43	44	39	40	45	44	41	18
44	44	44	39	40	45	44	41	19
45	45	45	40	40	45	44	42	20
45	45	45	40	41	46	45	42	21
44	45	45	40	41	47	45	42	22
44	44	44	39	39	47	44	41	23
44	45	44	39	39	46	44	41	24
45	45	45	40	41	47	45	42	MAX
36	37	38	35	33	36	37	37	MIN

Table3: Temperature of oil and winding in 230/63/20kv SARI Substation at 25 July 2006

Temprature of oil and winding in 230/63kv Pounel Substation at 25 July 2006								Hours
T2				T1				
oil	winding			oil	winding			
	HV	LV	TV		HV	LV	TV	
61	61	63	57	56	63	62	59	1
61	61	62	57	56	62	61	59	2
61	60	62	57	56	61	60	58	3
60	60	60	56	56	61	59	58	4
60	59	59	56	56	60	58	57	5
60	59	59	55	56	60	58	57	6
60	59	58	55	56	60	58	57	7
61	60	58	56	57	60	59	58	8
61	60	59	56	57	61	59	58	9
61	60	59	57	57	61	59	58	10
62	60	60	57	58	61	60	59	11
62	61	60	58	58	62	60	59	12
62	61	61	58	58	62	61	60	13
63	61	62	58	59	63	61	60	14
63	62	62	59	59	63	62	61	15
63	62	63	59	59	63	62	61	16
63	62	64	59	59	63	62	61	17
63	62	64	59	59	63	62	61	18
63	62	64	59	59	63	62	61	19
63	62	64	59	59	63	62	61	20
63	62	64	59	59	63	62	61	21
63	62	63	59	59	63	62	60	22
62	62	63	58	58	63	62	59	23
61	62	62	58	58	63	62	58	24
63	62	64	59	59	63	62	61	MAX
60	59	58	55	56	60	58	57	MIN

3. Theories and Model definition

3.1General Input-Output Models

A general input-output linear model can be written for a single-output system with an input u and output y, as follows:

$$A(q)\ y(t) = [B_i(q)/F_i(q)]\ u_i(t-nk_i) + [C(q)/D(q)]\ e(t)$$

Where u_i denotes input number i. There is an implied summation over all the inputs in the above expression.

A, B_i, C, D, and F_i are polynomials in the shift operator (z or q). The general structure is defined by giving the time delays nk and the orders of the polynomials (i.e., the number of poles and zeros of the dynamic model from u to y, as well as of the noise model from e to y).

Most often the choices are confined to one of the following special cases:

ARX: $A(q)\ y(t) = B(q)\ u(t-nk) + e(t)$

ARMAX: $A\ (q)\ y\ (t) = B\ (q)\ u\ (t-nk) + C\ (q)\ e\ (t)$

OE: $y(t) = [B(q)/F(q)]\ u(t-nk) + e(t)$ (Output Error)

BJ: $y(t) = [B(q)/F(q)]\ u(t-nk) + [C(q)/D(q)]\ e(t)$ (Box-Jenkins)

The basic state-space model in innovations form can be written as:

$$X\ (t+1) = A\ x\ (t) + B\ u\ (t) + K\ e\ (t)$$

$$Y\ (t) = C\ x\ (t) + D\ u\ (t) + e\ (t)$$

Note that A(q) corresponds to poles that are common between the dynamic model and the noise model (useful if noise enters system "close to" the input). Likewise, $F_i(q)$ determines the poles that are unique for the dynamics from input number i, and D(q) determines the poles that are unique for the noise.

These are all special cases of general linear input-output models. They correspond to linear difference equations relating the input to the output under various noise assumptions.

A prediction error/maximum likelihood method is used to estimate the coefficients of the polynomials by minimizing the size of the error term, e(t), in the above expressions. Several options govern the minimization procedure.

3-2.The ARX Model

The ARX model is a linear difference equation that relates the input u(t) to the output y(t) as follows:

$$y(t) + a_1\ y(t-1) + \ldots + a_{na}\ y(t-na) = b_1\ u(t-nk) + \ldots + b_{nb}\ u(t-nk-nb+1)$$

The structure is thus entirely defined by the three integers na, nb, and nk. na is the number of poles, nb+1 is the number of zeros, and nk is the pure time delay (the dead time) in the system. For a sampled data control system, typically nk=1 if there is no dead time. For multi-input systems nb and nk are row vectors, where the i^{th} element gives the order/delay associated with the i^{th} input.

3-3. Model Output

With using of Temperature data's that has collected by performing of a test on oil and winding of a 230/63/20kv Transformer in SARI Substation, the winding temperature for input in the model and oil temperature for out put has been used in calculation and simulations.

In the next sections the results of calculation in MATLAB that models has been estimated in many cases, and simulation in COMSOL would has been described.

3-3-1.State-space model

x(t+Ts) = A x(t) + B u(t) + K e(t)

y(t) = C x(t) + D u(t) + e(t)

```
A =
        x1         x2          x3          x4
x1   1.4537     0.97779    -0.88746     0.79613
x2  -0.92075    0.39492     1.4887     -0.7313
x3   0.80456   -0.35831     0.051135    0.81162
x4  -0.020607  -0.074647   -0.27479    -0.62393

B =
            u1
     x1    2.1079
     x2   -0.94638
     x3   -1.3377
     x4   -0.10788

C =
      x1       x2        x3        x4
y1  1.1812  0.92276   0.49652  -0.75747

D =
             u1
     y1      0

K =
            y1
     x1    3.3074
     x2   -4.4271
     x3    2.5696
     x4    0.14731

x(0) =
     x1    73.586
     x2   -80.235
     x3    34.517
     x4   -6.1225
```

- Estimated using N4SID from table 1 data
- Loss function 7.51265 and FPE 37.5632
- Sampling interval: 1

3-3-2.Discrete-time IDPOLY model (ARX):

A(q)y(t) = B(q)u(t) + e(t),

A(q) = 1 - 1.098 (± 0.3115) q^-1 + 0.4869 (± 0.4678) q^-2 - 0.6212 (± 0.4525)q^-3 + 0.5517 (± 0.3386) q^-4

B(q) = 0.3419 (± 0.4319) q^-1 - 0.346 (± 0.5567) q^-2 + 0.1409(± 0.5868)q^-3 + 0.1606 (± 0.4587) q^-4

- Estimated using ARX
- Loss function 0.425452 and FPE 0.850903
- Sampling interval: 1

3-3-3.Process model with transfer function

```
             K
G(s) = ---------- * exp (-Td*s)
          Tp1*s+1
```

With K = 7771.1 ± 7.3368e^(17)

Tp1 = 9.4869e ^ (5) ± 9.6618e^ (19)

Td = 30 ± 1.5629e^ (11)

- Estimated using PEM from data set of table1.
- Loss function 2.14608*e^(-025) and FPE 2.75925*e^ (-025)

3-3-4.Process model with transfer function with zero

```
             Tz*s+1
G(s) = K * ---------- * exp (-Td*s)
            Tp1*s+1
```

With K = -0.037547

Tp1 = 0.0010076

Td = 30

Tz = 34183

- Estimated using PEM from table 1
- Loss function 4.20726*e^(-29) and FPE 5.89016*e^(-29)

3-3-5.Discrete-time IDPOLY model (ARAMAX):

y(t) = [B(q)/F(q)]u(t) + e(t)

F(q) = 1 - 1.81 (± 0.05219) q^-1 + 0.905 (± 0.04991) q^-2

B(q) = 0.2859 (± 0.1169) q^-1 - 0.1988 (± 0.1111) q^-2

- Estimated using PEM
- Loss function 0.57436 and FPE 0.829632
- Sampling interval: 1

4- Result and discussions

The plots have shown the simulated (predicted) outputs of selected models. The models are fed with inputs from the Validation Data set, whose output is plotted in black (in white on a black background).

The plot takes somewhat different forms depending on the character of the validation data. This could be Time domain data.

Then the simulated or predicted model output is shown together with the measured validation data (figure3).

In all the cases, the percentage of the output variations that is reproduced by the model is displayed at the corner of the plots. The higher number means the better model.

The precise definition of the fit is:

FIT = [1 – NORM(Y – YHAT)/NORM(Y – MEAN(Y))]*100

Where Y is the measured output and YHAT is the simulated/predicted model output. The time span over which the fit is measured can be changed under the Options sub-menu Customized time span for fit.

There are sub-menus under the Options menu, which allow you to choose between simulated and predicted model output.

There are also options to show measured and model outputs together or to show the difference between them.

An intuitive interpretation of a K-step ahead predicted output is to see it as a result of a simulation, using the actual input that was started at the correct (measured) output level K samples earlier. K is called the prediction horizon.

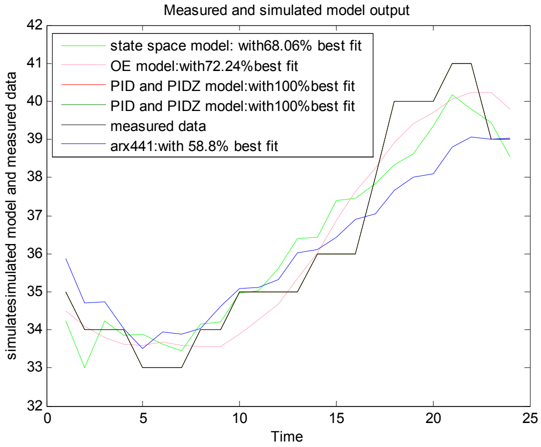


Figure3: Measured and simulated winding and oil temperature model of 230/63/20kv SARI transformer no.1 (TV).

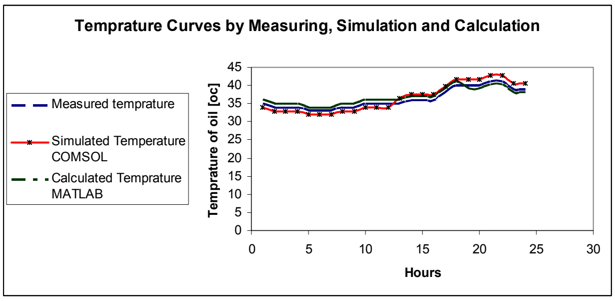


Figure 4: Comparison Measured, Simulated and calculated of Oil Temperatures.

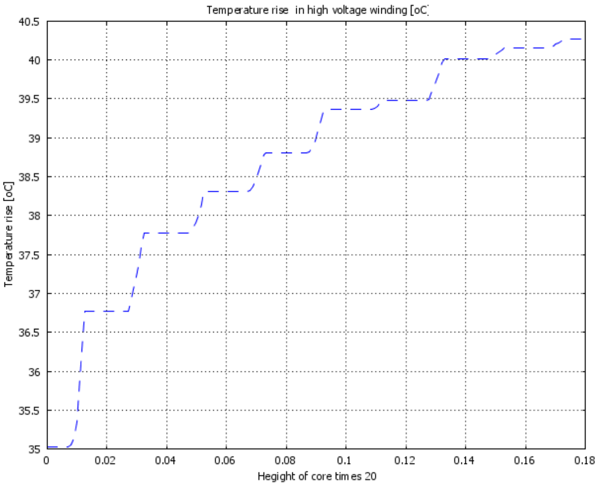


Figure5: simulated HV winding temperature of 230/63kv SARI transformer no.1 (TV), in COMSOL software.

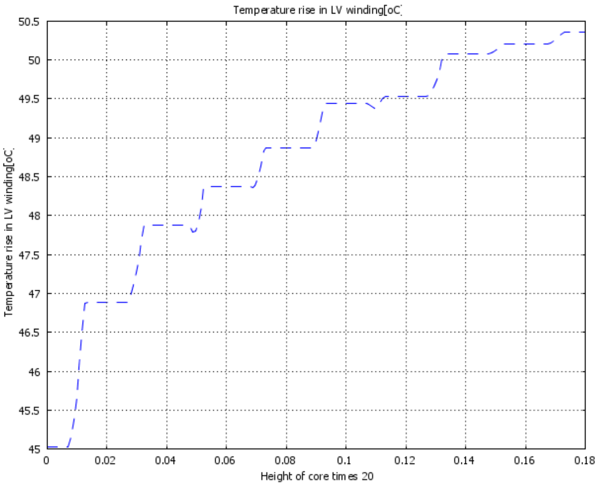


Figure6: simulated LV winding temperature of 230/63kv SARI transformer no.1 (TV), in COMSOL software.

By simulation in COMSOL that is shown in figures 5 and 6 it can be seen that the temperature of LV windings is higher than HV windings, because the LV carry out the higher current, this results has a good fits with measured data's. and so figure 6 is shown that the measured, simulated and calculated data has a good best fits.

5- Conclusion

A power transformer thermal model has been developed based on the analogy between thermal dynamics and electric circuits. The proposed thermal model can calculate continuously temperatures of the main parts of an ONAN/OFAF cooled power transformer under various ambient and load conditions. The model parameters can be obtained by the estimations search only based on the on-site measurements, instead of the experimental methods and off-line tests.

In the top of contexts we can see that the Process model of transfer function with zero is a best method for get a model for heat transfer from winding to oil and oil to ambient. It has a good best fit with highest coordination. It is shown in figure 3.

It is shown in figure3. of course with conversion of discrete to continuous, the data versus of continuous would obtain.

In figure 4 shown that the tolerances of simulation and calculation with measured data are less than 4%.
It shown that new model has a good best fit.

References

1. Dennis, J.E., Jr., and R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall, Englewood Cliffs, N.J., 1983.

2. A. J. Oliver: "Estimation of transformer winding temperatures and coolant flows using general network method", IEE Proc., Vol. 127, Pt. C, No. 6, November 1980

3. VanOverschee, P., and B. DeMoor, Subspace Identification of Linear Systems: Theory, Implementation, Applications, Kluwer Academic Publishers, 1996.

4. Verhaegen, M., "Identification of the deterministic part of MIMO state space models," Automatica, Vol. 30, pp. 61-74, 1994.

5. Larimore, W.E., "Canonical variate analysis in identification, filtering and adaptive control," In Proc. 29th IEEE Conference on Decision and Control, pp. 596-604, Honolulu, 1990.

6. Kourosh Mousavi Takami, Evaluation of oil in over 20 year's old oil immersed power transformer, Mazandaran University, May 2001.

7. Kourosh Mousavi Takami, Advanced Transformer Monitoring & Diagnostic Systems and thermal assessment with robust software's, research presentation, Water and power University, March 2007, Tehran, Iran

8. Kourosh Mousavi Takami, Hot Spot identification and find a best thermal model for large scale power transformers, April 2006, KTH University, Stockholm, Sweden.

9. M. K. Pradhan and T. S. Ramu, `` Prediction of Hottest Spot Temperature (HST) in Power and Station Transformers``, IEEE Transaction on power delivery,vol.18, NO.4,October2003

10. Kourosh Mousavi Takami, Hassan Gholinejad, Jafar Mahmoudi, Thermal and hot spot evaluations on oil immersed power Transformers by FEMLAB and MATLAB software's, IEEE Conference, Int. Conf. on Thermal, Mechanical and Multiphysics Simulation and Experiments in Micro-Electronics and Micro-Systems, EuroSimE 2007, London, 17 April 2007, pp 529-534.

11. Kourosh Mousavi Takami, Jafar Mahmoudi, Evaluation of Large Power Transformer Losses for green house gas and final cost reductions, 3rd IGEC conference, Sweden, June 18, 2007.

12. Kourosh Mousavi Takami, Jafar Mahmoudi, A novel device (oil spraying system) for local cooling of hot spot and high temperature areas in power transformers, 3rd IGEC conference, Sweden, June 19, 2007.

13. Kourosh Mousavi Takami, Jafar Mahmoudi, Thermal evaluation and energy saving with loss reduction in core and winding of power transformers, 3rd IGEC conference, Sweden, June 19, 2007.

14. Kourosh Mousavi Takami, Jafar Mahmoudi, A new apparatus for mitigating the hot spot problem in large power transformers using Ants algorithm, IEEE PES PowerAfrica 2007 Conference and Exposition Johannesburg, South Africa, 18 July 2007

15. FEMLAB V2.3, Electromagnetics Module. Comsol, 2002.

16. Lates L. V., Electromagnetic calculation of Transformers and Reactors: Moscow ENERGY, 1981 (In Russian), p.313.

17. IEEE Loading Guide for Mineral Oil Immersed Transformer, C57.91, pp. 18–19, 46–53, 1995.

Kourosh Mousavi Takami was born in Sari, Mazandaran,Iran . He received the B.S.c. degree in electric power engineering from the Iran University of Science and Technology (IUST) Tehran, Iran, Oct1995 and the M.Sc. degree in electric power engineering from the Engineering Faculty of Mazandaran University, Iran in 2002. Currently, he is PhD student at Mälardalen University in Sweden since 2005. He has so over ten years experience in power system design and installations. His research interests include Optimization and simulation of heat generation and transfer in the core and winding of power transformers; diagnostic testing and condition monitoring of power equipments, and application of fuzzy and Ants algorithm to condition monitoring of power equipments.

Jafar Mahmoudi was born in Tehran, Iran. He received the B.Sc., M.Sc. Degree in Sharif University and PhD degrees from KTH University, Stockholm, Sweden. Currently, he is a Professor with the Department of Public Technology Engineering in MdH University, Västerås, and Sweden.His major research focus is development of new technology and methods for industrial energy optimization with special focus on heat and mass transfer. He has years of theoretical & experimental- experience on this. He also has a broad technical background encompassing thermodynamic, numerical methods and modeling (CFD computation) as well as materials science. This in combination with his industrial experience has served as a solid basis to build upon in expanding his research activities and focusing on relevant and current industrial issues.

Over the last 10 years his focus has been on the practical and industrial application of the above mentioned methods, an effort conducted in a large number of industrial projects. In this, his teaching experience has proved invaluable.

Simulation and optimization of high power super heater reflectors usable in electrical furnaces for heat loss reduction

Kourosh Mousavi Takami
TDI (P.O. Box: 13445686, Sharif institute of technology (TDI), Tehran, Iran) researcher and Ph.D. student in Mälardalen University,
Kourosh.mousavi.takami@mdh.se

Abstract

Super heater elements concentrated in very high and clean power at temperature up to 1800 ºc is needed for aluminium melting, oil drying equipment using for power transformers and same that material. There fore, always it is need to minimize power consumption and increase effectiveness at power reflector customers. This means to minimize heat losses in the High Power Reflector, and to optimize heat performance with respect to output power and temperature distribution. This can be done by altering the size and form of the insulating ceramic fiber to have as much power as possible 'reflected' out from the module. It can also be done by altering the size and form of the element .It has been done and the task solved mainly by heat transfer simulations, using COMSOL and MTLAB software's. To minimize the number of different designs (and thus the number of calculations) were studied; it is also done that statistical methods for experimental design used to determine which designs should be used in the calculations. When an optimal solution has been found, it has been tested at factory's facilities, with two reference designs.

Keywords: reflector, simulation, element.

1. Introduction

Molybdenum disilicide (MoSi2) are resistance type heating elements made of a dense ceramic-metallic material which can produce furnace temperatures approaching 1800°C. Although more expensive than traditional metallic elements, MoSi2 elements are known for their longevity due in part to a protective quartz layer that forms on the surface of the element "hot zone" during operation.

The MoSi2 elements are made to industry standard established resistance values, dimensions and geometries, and therefore are interchangeable with other manufacturers of molybdenum disilicide elements.

The $MoSi_2$ elements are graded in many maximum temperature ranges, same 1700°C (3090°F) and 1850°C (3270°F). Elements are available in many shapes and forms (see examples shown below).

Benefits of $MoSi_2$ elements include high temperature operation (in air), long element life and the ability in some cases to replace any failed elements hot.

Customers of The MoSi2 elements enjoy two key added benefits – first, unparalleled price and second, experienced engineering support. Anyone can sell MoSi2, but knowing how to design the system is where MoSi2 excels. MoSi2 systems must be engineered for customer to get the best performance and optimum service life. The engineers can assist in:

analyzing the relationship between the furnace temperature, element temperature and the element surface load selecting the element surface load according to the furnace construction, atmosphere and operating temperature choosing the most suitable element size and style for customer application designing a new furnace with MoSi2 or converting from gas or oil fired to a MoSi2 based electric heating system determining if one of MoSi2 Power Pack systems can work for application.

MoSi2 heating element is used in the high temperature under oxidizing atmosphere. It will form the SiO2 film which can keep the element from being melted. During the oxidizing process, the SiO2 protecting film is formed again when the element continues to be used. The MoSi2 heating element must not be used in the temperature between 400ºC and 700ºC for a long time; otherwise the element will be cremated under the strong oxidizing function in the low temperature.

The resistance of MoSi2 heating element increases fast along with the temperature going up. This means that when the elements are connected to a constant voltage, the power will be higher at lower temperatures and will be graduadlly reduce with increasing temperature, thus shortening the time for the furnace to reach operating temperature. Furthermore, as the power of the elements decreases, the danger of overheating will be reduced. Under the normal condition, the elements resistance does not change along with used time generally. So the old and the new element can be used together. Element length, diameter and distance between elements- element with reflector wall and bottom; depth and dimensions of reflector, reflector material (material of surface, insulation and back of reflector), and design of reflector, power and sealed system are important in temperature distribution and so losses. Low losses and have high temperature efficiency in 20 centimeter in top of element is the aim of this optimization.

2. Problem definition

It's necessary to create optimum heating solutions for each power reflectors.

One of manufacturer same as RATH and KANTHAL many heating solutions is called Superthal, which consist of vacuum formed ceramic fiber shapes with an integrated Kanthal Super ceramic ($MoSi_2$) heating element. The Super ceramic heating elements can be operated at element temperatures up to 1850°C.

The Superthal High Power Reflector is a compact fiber insulated modular heater with so-called 'cubic' Kanthal Super elements, see Fig. 1. It is designed for a power of up to 110kW/m^2 at 1650°C.

Multiple units can easily be joint together in different configurations e.g. in rows or squares. The Superthal High Power Reflectors are used wherever a concentrated high power at temperatures of up to 1650°C is needed, for instance in single billet heating, aluminum melting furnaces or ladle heaters.

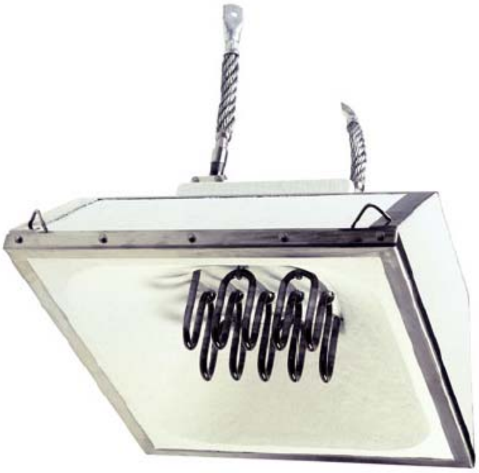


Figure 1. A Kanthal Superthal High Power Reflector

A reflector dimension that has been used for simulation is: 600×600mm in wide and so 230mm in depth, voltage and current are 66v and 605 Amp. at 1650 ºC respectively; power density is 110 w/m^2 and finally element type is 12/24mm.

2.1 Description of task

There is always a need to minimize power consumption and increase effectiveness at customers. In this case it means to minimize heat losses in the Power Reflector, and to optimize heat performance with respect to output power and temperature distribution. This can be done by altering the size and form of the insulating ceramic fiber to have as much power as possible 'reflected' out from the module. It can also be done by altering the size and form of the element.

It is suggested that the task is solved mainly by heat transfer simulations, using e.g. some FEM software and so COMSOL and MTLAB software's.

To minimize the number of different designs (and thus the number of calculations) that need to be studied, it is also suggested that statistical methods for experimental design are used to determine which designs should be used in the calculations.

When an optimal solution is reached, this will be tested at Kanthal's facilities, together with one or two reference designs.

3. Material properties

Thermal conductivity (W/m K), heat capacity (J/kg K) and resistivity (Ωm) are shown in figures 2,3 and 4 and so in this element emissivity is ε = 0.7

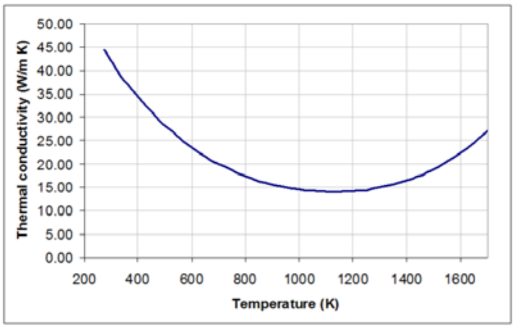


Figure 2. Thermal conductivity (W/m K)versus temperature

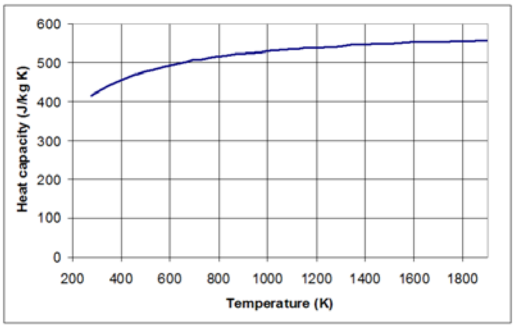


Figure 3. Heat capacity (J/kg K) versus temperature

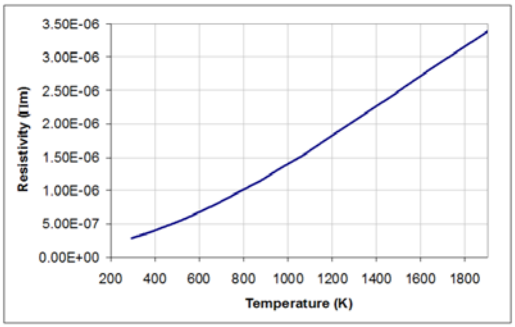


Figure 4. resistivity (Ωm) versus temperature

Table 1. Element for 12mm hot zone diameter and fiber panel characterizes

Element	**Min**	**Max**	
Number of shanks	4	25	4, 6, 12, 16, **20**, 25
Hot zone length, Le (mm)	100	300	
Terminal length, Lu (mm)			150 mm for 12 mm hot zone.
Center-to-center distance, a (mm)	35/45	60/80	Depends on the diameter of the hot zone (9 or 12 mm).
Hot zone diameter, d (mm)	9	12	12mm
Terminal diameter, c (mm)	18	24	For 12 mm hot zone, the terminal is 24 mm.
Surface load (W/cm^2)	10	40	
Current (A)	200	700	605
Plug and Fiber panel			
Material quality	1400	1700	1700
Size			600*600*230mm
Shape			To be determined by sim.
Heat transfer coefficient			100w/m^{2}k^4
Emissivity			0.7
Coefficient of radiation			5.67w/m^2°C

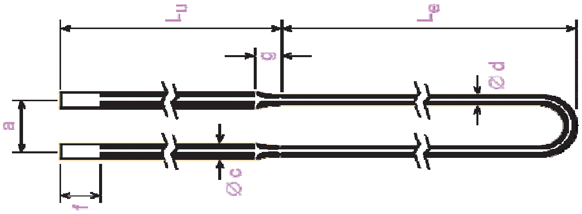


Figure 5. Element

3.1 1600°C VACUUM FORMED CERAMIC FIBER PRODUCTS

Vacuum formed reflector exhibit excellent thermal conductivity, good compressive strength and exceptional thermal stability. This combination of physical properties allows reflector products to be used in hot face applications at temperatures up to 1500°C (2732°F) as well as efficient back-up insulation. By surface-rigidizing, heating elements are able to be effectively mounted on the board or shape.

Table 2. Technical data of reflectors

Classification/Service Temperature		
	1600°C/1500°C	2912°F/2732°F
Chemical Composition		
	65% Al2O3	34% SiO2
Density	300 kg/m3	18.6 lb/ft3
LOI (@1000°C, 6h)		
	KVS 161 4%	KVS 164 0%
Linear Thermal Shrinkage (KVS 161) 24 hours:		
@ 1300°C (2372°F) - 0.5%		
@ 1400°C (2552°F) -1.0%		
@ 1500°C (2722°F) -1.5%		
[“-“ = shrinkage “+” = expansion]		
Compressive Strength		0.15 MPa
(@ 10 % compression)		21.9 lb/in2
Cold Bending Strength (MOR)		1 Mpa
(3 Point Bend)		146 lb/in2
Specific Heat	0.92 J/g·°C	0.22 BTU/lb·°F

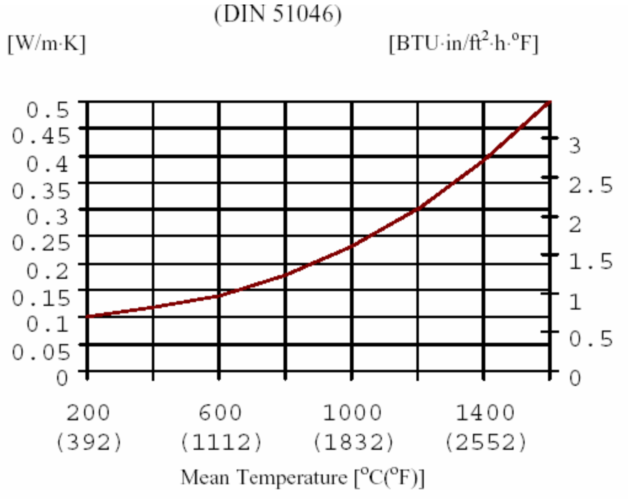


Figure 6. Thermal conductivity of ceramic reflectors

4. The Heat Equation

The mathematical model for heat transfer by conduction is the heat equation:

$$\rho C \frac{\partial T}{\partial t} + \nabla \cdot (-K\nabla T) = Q$$

Quickly review the variables and quantities in this equation:

T is the temperature.

ρ is the density.

C is the heat capacity.

C_p is the heat capacity at a constant pressure.

C_v is the heat capacity for a constant volume.

k is the thermal conductivity.

Q is a heat source or heat sink.

For a steady-state model, temperature does not change with time and the first term containing ρ and C vanishes. If the thermal conductivity is anisotropic, k becomes the thermal conductivity tensor k:

$$k = \begin{bmatrix} k_{xx} \, k_{xy} \, k_{xz} \\ k_{yx} \, k_{yy} \, k_{yz} \\ k_{zx} \, k_{zy} \, k_{zz} \end{bmatrix}$$

To model heat conduction and convection through a fluid, the heat equation also includes a convective term. this formulation in the General Heat Transfer application mode as:

$$\rho C_p \frac{\partial T}{\partial t} + \nabla \cdot (-K\nabla T + \rho C_p T u) = Q \qquad (1)$$

Where u is the velocity field, this field can either be provided as a mathematical expression of the independent variables or calculated by a coupling to a momentum balance application mode such as Incompressible Navier-Stokes or Non-Isothermal Flow.

The heat flux vector is defined by the expression within the parentheses in the equation above. For transport through conduction and convection this equation yields:

$$q = -K\nabla T + \rho C_p T u$$

Where: q is the heat flux vector. If the heat transfer is by conduction only, q is determined by: $q = -K\nabla T$

Heat capacity refers to the quantity that represents the amount of heat required to change one unit of mass of a substance by one degree. It has units of energy per mass per degree. This quantity is also called specific heat or specific heat capacity.

4.1- Boundary Conditions

We use two different types of boundary conditions, the Dirichlet type and the Neumann type. The Dirichlet type boundary condition is used to set a temperature on the boundary:

$$T = T_0$$

While the Neumann type condition is used to set the heat flux on the boundary:

$$-n \cdot q = q_0 \qquad (2)$$

Where:

q is the heat flux vector.

n is the normal vector of the boundary.

q_0 is inward heat flux, normal to the boundary.

The Heat Transfer Module uses the following, more general and formulation of Equation 2 :

$$-n \cdot q = q_0 + h(T_{\inf} - T) \qquad (3)$$

This formulation allows you to specify the heat flux in terms of an explicit heat flux q_0 and a heat transfer coefficient, h, relative to a reference temperature, $T_{\inf}$.

The thermal insulation condition is obtained by setting $q_0 = 0$. This is also the case for symmetry boundaries.

For heat transfer problems involving radiation, additional terms are added on the right side of Equation 3 to represent the radiative heat flux. This is covered in the next section.

4.2- Radiative Heat Transfer

So far we have considered heat transfer by means of conduction and convection. The third mechanism for heat transfer is radiation. Thermal radiation denotes the stream of electromagnetic waves emitted by a body that has a certain temperature. Here we will study the theory behind the radiative heat transfer process that occurs on the surface of a body.

The radiosity is the sum of the reflected radiation and the emitted radiation:

$$J = \rho G + \varepsilon \sigma T^4 \qquad (4)$$

The net radiative heat flux into the body is obtained by computing the difference between the irradiation and the radiosity:

$$q = G - J \qquad (5)$$

Using Equation 4 and Equation 5 , we eliminate J and obtain the following expression for the net inward heat flux into the body:

$$q = (1-\gamma)G - \varepsilon \sigma T^4 \qquad (6)$$

Besides the assumption that the surface is opaque, we also assume that the surface is diffusive-gray. For diffusive-gray surfaces, we can assume the following for the reflectivity and the emissivity:

$$\alpha = 1 - \gamma = \varepsilon \qquad (7)$$

By inserting Equation 7 into Equation 6 we eliminate the reflectivity from Equation 6 and express the net radiative heat flux q in the inward direction as:

$$q = \varepsilon (G - \sigma T^4) \qquad (8)$$

The expression above denotes the net radiative heat flux into a boundary.

We distinguish between two different types of radiative heat transfer: surface-to-ambient radiation and surface-to-surface radiation. Equation 8 holds for both radiation types. However, the irradiation term, G, is different for surface-to-ambient and surface-to-surface radiation. The irradiation term and the resulting radiative heat flux for both radiation types are derived in the sections below.

4.3- SURFACE-TO-AMBIENT RADIATION

For surface-to-ambient radiation, assume the following:

The ambient surroundings in view of the surface have a constant temperature Tamb.

The ambient surroundings have the properties of a black body, that is, emissivity and absorptivity equal to 1 and reflectivity equal to zero.

These assumptions allow us to explicitly express the irradiation on the surface as:

$$G = \sigma T_{amb}^4 \qquad (9)$$

By inserting Equation 9 into Equation 8 we obtain the net inward heat flux for surface-to-ambient radiation:

$$q = \varepsilon \sigma (T_{amb}^4 - T^4) \qquad (10)$$

4.6- Electromagnetic equations:

To derive the equation system this mode solves, start with Ampère’s law,

$$\nabla \times H = J + \tfrac{\partial D}{\partial t} = \sigma E + \sigma V \times B + J^e + \tfrac{\partial D}{\partial t} \qquad (11)$$

Now assume time-harmonic fields and use the definitions of the potentials, $B = \nabla \times A,\ E = -\nabla V - \tfrac{\partial A}{\partial t}$, and combine them with the constitutive relationships $B = \mu_0 (H + M)$ and $D = \varepsilon_0 E + P$ to rewrite Ampère’s law as:

$$\begin{aligned}&(j\omega\sigma - \omega^2 \varepsilon_0) A + \nabla \times (\mu_0^{-1} \nabla \times A - M) \\ &- \sigma V \times (\nabla \times A) + (\sigma + j\omega\varepsilon_0) \nabla V = J\end{aligned} \qquad (12)$$

In the 2D in-plane case there are no variations in the z-direction, and the electric field is parallel to the z-axis. Therefore you can write ∇V as −ΔV/L where ΔV is the potential difference over the distance L.

Now simplify these equations to:

$$\begin{aligned}&-\nabla .(\mu_0^{-1} \nabla A_z - \begin{bmatrix} -M_y \\ M_x \end{bmatrix}) + \sigma V . \nabla A_z \\ &+ (j\omega\sigma - \omega^2 \varepsilon_0) A_z = \sigma \tfrac{\nabla V}{L} + J_\varphi^e + j\omega P_\varphi\end{aligned} \qquad (13)$$

Where: σ electrical conductivity, V_{loop} loop potential, J_φ^e external current density, P_φ electric polarization, A_z magnetic potential z component and v is velocity.

With using finite element mode performs this transformation to avoid singularities on the symmetry axis.

The relevant interface condition is $n_2 \times (H_1 - H_2) = J_\varphi$

The natural boundary condition fulfills this equation if the surface current vanishes. We can transform the Neumann condition of this PDE into:

$$-n.[(\mu_0^{-1}\nabla A_z - \begin{bmatrix}-M_y\\ M_x\end{bmatrix})_1 - (\mu_0^{-1}\nabla A_z - \begin{bmatrix}-M_y\\ M_x\end{bmatrix})_2$$
$$= -n\times[(\mu_0^{-1}\nabla\times A - M)_1 - (\mu_0^{-1}\nabla\times A - M)_2]$$
$$= -n\times(H_1 - H_2) = 0 \qquad (14)$$

Joule Equations:

$$-\nabla.(\delta\nabla V - J^e) = Q \quad \text{and} \quad \delta = 1/(\rho_0(1+\alpha(T-T_0)))$$

$$n.(J_1 - J_2) = \delta(V - V_{ref})/d \qquad (15)$$

Where: V is potential is around of 66 volts, δ is conductivity at reference temperature and d is thickness.

5. Simulation in COMSOL software

The simulations of temperature distribution have done on base of many cases. Element length, diameter and distance between elements- element with reflector wall and bottom; depth and dimensions of reflector, reflector material (material of surface, insulation and back of reflector), and design of reflector, power and sealed system are important in temperature distribution and so losses. Low losses and have high temperature efficiency in 20 centimeter in top of element is the aim of this simulation. Necessary data for simulation has been given in table 1,2 and figures 2,3,4 and 6.

6. Result an discussion

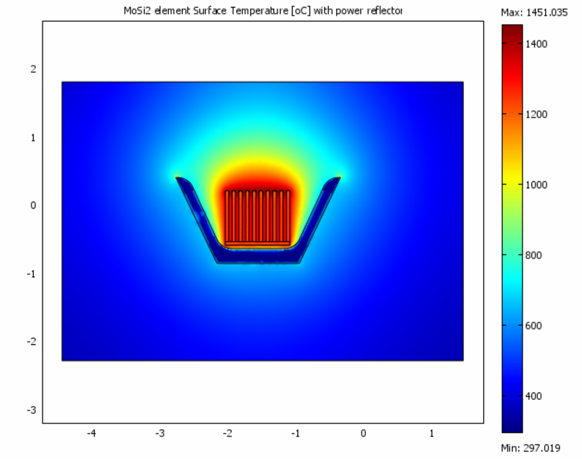


Figure 7. Simulation of temperature distribution in bowel form reflector in 500A current

Table 3. Simulation results

Parameter	Descriptions			
Emistivity	0.7	0.85	0.7	0.85
Reflector shape	cave	Cave	circle	circle
a[mm]	80	80	80	80
L$_e$[mm]	200	200	200	200
Depth[mm]	230	230	230	230
Element tmp.[ºk]	1900	1900	1900	1900
Simulated tmp.[ºk] on 20cmre	1720	1745	1795	1820

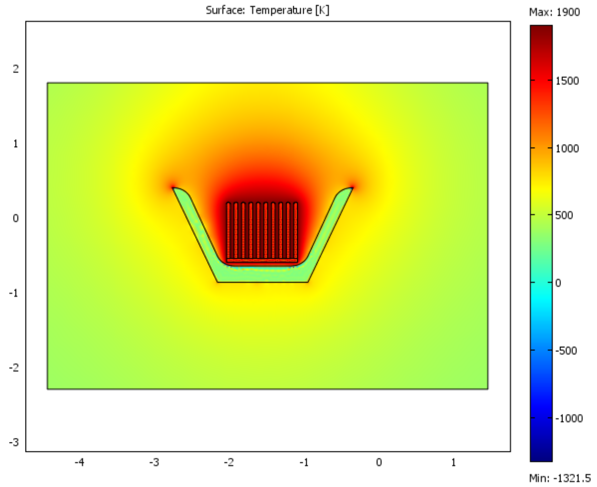


Figure 8. Simulation of temperature distribution in bowel form reflector in 605 A

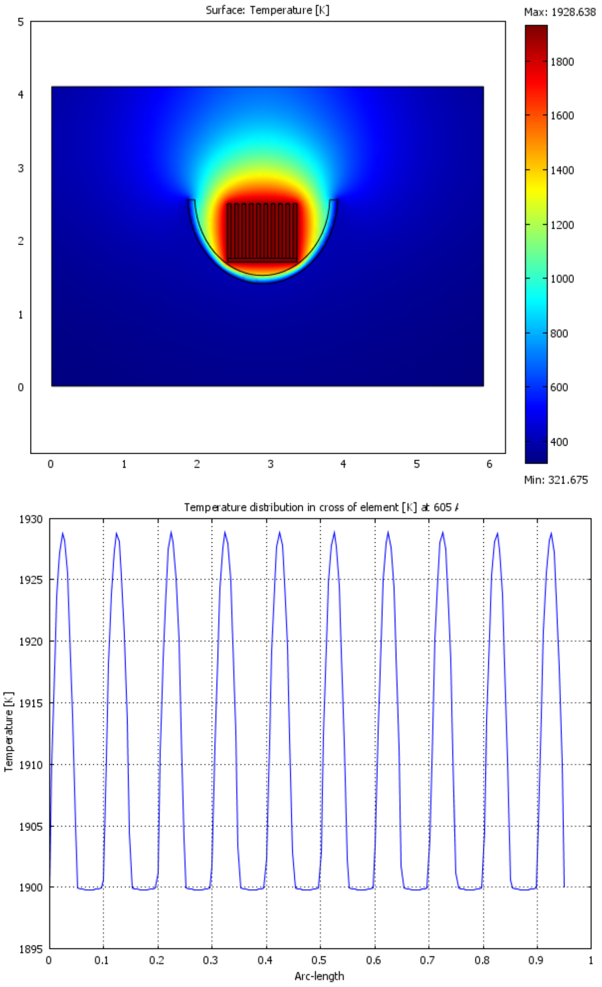


Figure 9. Simulation of temperature distribution in circular

form reflector in 605 A current and the Temperature distribution on element is drown in down figure

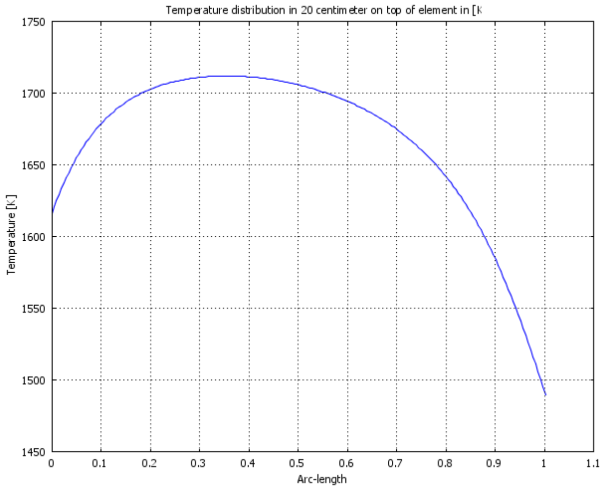


Figure 10. Simulation of temperature distribution in bowl form reflector in 605 A current at 20 centimeter on the top of element

In figures 7-10 the temperature distribution in bowl and circular reflector forms are shown. The summary results have written in table3. In these results illustrated that circular reflector on his fireplace has a high efficiency on temperature profile. And so in figure 11- 13 electrical field and temperature distribution are shown.

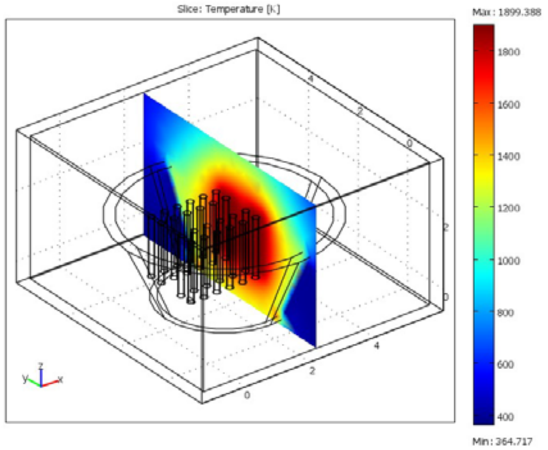


Figure 11. Simulation of temperature in Cone form reflector on 605 A current

7. Conclusion

These simulations indicate that the emissivity of the work piece has to be considered in addition to its thermal mass. Necessary distance, between elements, to counteract the effect of the electromagnetic force on elements must be kept; in this case, with increasing in the surface loading and so element length, the distance must be increased. Accessible area for element installation, element dimension and element location has shown that has a responsible effect on temperature distribution. In elliptical and circular reflectors, if 20 cm line will have located in the fireplace, on that's case, it will have higher efficiency.

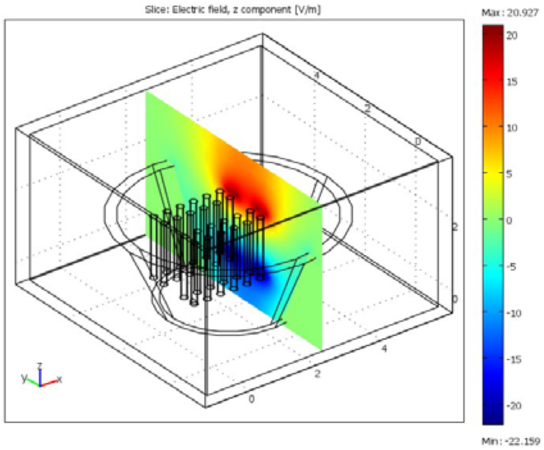


Figure 12. Simulation of Z direction electrical field in Cone form reflector on 605 A current

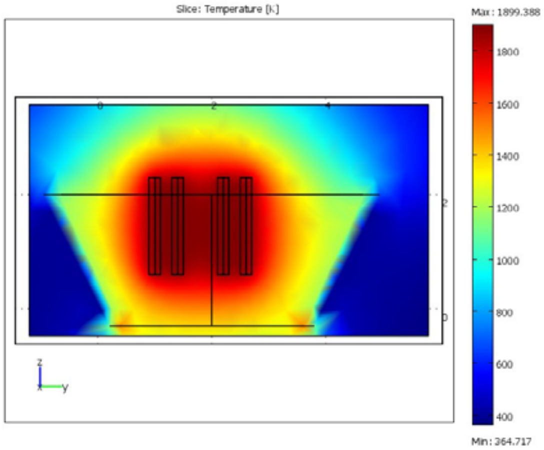


Figure 13. Simulation of temperature in Cone form reflector on 605 A current, 2D of figure 11

References

1. KANTHAL super electric heating element hand book
2. COMSOL Multiphysics version 3.3a software
3. Dennis, J.E., Jr., and R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall, Englewood Cliffs, N.J., 1983.

4. Jafar Mahmoudi, Mathematical modelling of fluid flow, heat transfer and solidification in a strip continuous casting process``, International Journal of Cast Metals Research IJC604.3d 20/1/06 13:22:12, The Charles worth Group, Wakefield +44(0)1924 369598- Rev 7.51n/W (Jan 20 2003).

Kourosh Mousavi Takami was born in Sari, Mazandaran,Iran . He received the B.S.c. degree in electric power engineering from the Iran University of Science and Technology (IUST) Tehran, Iran, Oct1995 and the M.Sc. degree in electric power engineering from the Engineering Faculty of Mazandaran University, Iran in 2002. Currently, he is PhD student at Mälardalen University in Sweden since 2005. He has so over ten years experience in power system design and installations. His research interests include Optimization and simulation of heat generation and transfer in the core and winding of power transformers; diagnostic testing and condition monitoring of power equipments, and application of fuzzy and Ants algorithm to condition monitoring of power equipments.

Simulation of energy in the building and design a new intelligent building with controllable and wise devices

Kourosh Mousavi Takami S.M.Esmail Mousavi Takami
TDI researcher and PhD student in MDH M.Sc in architect
Kourosh.mousavi takami@mdh.se architecttakami@yahoo.com

Abstract

Building energy simulation is important for the study of energy efficiency in buildings. An energy saving control system of lightning, heating and variable-air-volume air conditioner in intelligence building is simulated in this paper. It reaches good control effect and energy efficiency by making the best of the advantages of intelligence building. In the regulating period, the lightning lux, heating flux and air volume is decided by means of feed forward control. The previous turning off period is determined in the way of penalty function. It has been used in a sample building for developing building energy standards and analysing energy consumption and conservation measures of buildings.

Keywords: neural network, building, intelligent

1. Introduction

Intelligent building is a building with intelligence. The air condition (AC) system exhausts most energy in the building, so it is necessary to adopt a control method to get the optimum energy efficient by making the best of the resources in the intelligence building. The air system is a complex system with characters such as: too many factors, nonlinear, long delaying, strong inertia etc. And the accurate mathematics model is hard to make out, so the performance of classical control theory can hardly be satisfying. Thanks to the ability of intelligent building, such as measuring, recording and analyzing the data, some advanced methods can be used to achieve both good control effect and ideal efficiency.

Energy management and control system (EMCS) technology has evolved over the past three decades from pneumatic and mechanical devices to direct digital controls (DDC) or computer based controllers and systems. Today's EMCS systems consist of electronic devices with microprocessors and communication capabilities and utilize widespread use of powerful, low cost microprocessors and standard cabling communication protocols.

This paper discusses functions and capabilities of a typical building/facility EMCS and its simulation. Most basic functions and some intermediate and advanced functions described in this paper are generally required to operate buildings/facilities efficiently.

1.1. Intelligent buildings - control theory

The essence of Building Management Systems and Intelligent Buildings is in the control technologies, which allow integration, automation, and optimisation of all the services and equipment that provide services and manages the environment of the building concerned.

Programmable Logic Controllers (PLC's) formed the original basis of the control technologies.

Later developments, in commercial and residential applications, were based on 'distributed-intelligence microprocessors'.

The use of these technologies allows the optimisation of various site and building services, often yielding significant cost reductions and large energy savings. There are numerous methods by which building services within buildings can be controlled, falling broadly into two method types:

- Time based - providing heating or lighting services, etc., only when required, and
- Optimiser Parameter based - often utilising a representative aspect of the service, such as temperature for space heating or illuminance for lighting.

1.2. Heating - time-based control

Time-based controls can be used to turn on and off the heating system (and/or water heating) at pre-selected periods (of the day, of the week, etc). Optimiser Parameters: whatever the conditions, the controls make sure the building reaches the desired temperature when occupancy starts.

- Heating - optimizer parameter-based (temperature) control examples
- Temperature control: protection against freezing or frost protection generally involves running heating system pumps and boilers

when external temperature reaches a set level (0°C).

- Compensated systems: will control flow temperature in the heating circuit relative to external temperature. This will give a rise in the circuit flow temperature when outside temperature drops.
- Thermostatic radiator valves: these sense space temperature in a room and throttle the flow accordingly through the radiator or convector to which they are fitted.
- Proportional control: involves switching equipment on and off automatically to regulate output.
- Other methods can include thermostats, occupancy sensing PIR's (passive infra-red sensors), and manual user control.

1.3. lighting control methods

Different control systems exist, again time-based control and optimiser parameter-based where a level of illuminance or particular use of lighting is required.

- Zones: lights are switched on corresponding to the use and layout of the lit areas, in order to avoid lighting a large area if only a small part of it needs light.
- Time control: to switch on and off automatically in each zone to a preset schedule for light use.
- Passive Infra-Red (PIR) Occupancy sensing: In areas which are occupied intermittently, occupancy sensors can be used to indicate whether or not anybody is present and switch the light on or off accordingly.
- Light level monitoring: this consists of switching or dimming artificial lighting to maintain a light level measured by a photocell.

1.4. building management systems and intelligent buildings - energy savings

Until recent years, energy efficiency has been a relatively low priority and low perceived opportunity to building owners and investors. However, with the dramatic increase and awareness of energy use concerns, and the advances in cost-effective technologies, energy efficiency is fast becoming part of real estate management, facilities management and operations strategy. The concepts are also now making significant inroads into the domestic residential house building sectors.

For lighting, energy savings can be up to 75% of the original circuit load, which represents 5% of the total energy consumption of the residential and commercial sectors.

Energy savings potential from water heating, cooling, or hot water production, can be up to 10%, which represents up to 7% of the total energy consumption of the domestic residential and commercial sectors.

Experiences from studies in Austria suggest potential heating and cooling energy savings are up to 30% in public buildings. Even allowing for the fact that buildings used in the study may have been those with particularly high energy usage, the figure is an impressive one. (Source: EU2 Analysis and Market Survey for European Building Technologies in Central & Eastern European Countries - GOPA)

Intelligent Buildings and Building Management Systems technologies contribute directly to the reduction in energy use, in commercial, industrial, institutional and domestic residential sectors.

In short, Intelligent Buildings and suitably applied Building Management Systems are good for the environment.

Careful interpretation is required. In the UK, adoption of controls technologies into the new build and major refurbishment sectors is relatively high: Estimates a few years ago of the UK market for Building Management Control Systems for new build and major refurbishment, all sectors, suggest market adoption of (as at 1994 - Source UK1 An Appraisal of UK Energy RTD, ETSU -1994):

- Heating controls 70%.
- Hot water system controls 90%.
- Air conditioning controls 80%.

However according to European Commission as many as 90% of all existing buildings have inapplicable or ineffective controls, many of which require complete refurbishment of control systems.

Moreover conventional control systems stop short of automated Intelligent Buildings full capabilities. A significant human element is required for optimal effective operation even if control systems correctly specified and installed.

Given typical installations and equipment there is often a difficulty for building occupants (residential) or managers (commercial) to operate them correctly. Usage and correct operation are vital for effective results.

Education of users; improved systems-design user-friendliness, and the provision of relevant instructions and information are all critical to enable theory to translate into practice, and for potential effectiveness and savings to be realised.

1.5. Practical benefits

Energy-effective systems balance a building's electric light, daylight and mechanical systems for maximum benefit.

Enhanced lighting design is more than an electrical layout. It must consider the needs and schedules of occupants, seasonal and climatic daylight changes, and its impact on the building's mechanical systems.

1.6. Lighting systems

Adding daylight to a building is one way to achieve an energy-effective design. Natural daylight 'harvesting' can make people happier, healthier, and more productive. And with the reduced need for electric light, a great deal of money can be saved on energy. Nearly every commercial building is a potential energy saving project, where the electric lighting systems can be designed to be dimmed with the availability of daylight. Up to 75% of lighting energy consumption can be saved. In addition, by reducing electric lighting and minimizing solar heat gain, controlled lighting can also reduce a building's air conditioning load.

1.7. Mechanical systems

The HVAC system and controls, including the distribution system of air into the workspaces, are the mechanical parts of buildings that affect thermal comfort. These systems must work together to provide building comfort. While not usually a part of the aesthetics of a building, they are critical to its operations and occupant satisfaction.

The number one office complaint is that the workplace is too hot. Number two is that it's too cold.

Many people cope by adding fans, space heaters, covering up vents, complaining, conducting 'thermostat wars' with their co-workers, or simply leaving the office. Occupants can be driven to distraction trying to adjust the comfort in their space. Improper temperature, humidity, ventilation, and indoor air quality can also have significant impacts on productivity and health. When we are thermally comfortable we work better, shop longer, relax, breathe easier, focus our attention better.

In order to provide a comfortable and healthy indoor environment the building mechanical system must:

- Provide an acceptable level of temperature and humidity and safe guard against odours and indoor air pollutants.
- Create a sense of habitability through air movement, ventilation and slight temperature variation.
- Allow the occupant to control and modify conditions to suit individual preferences.

2. Simulations:

Thermal problem detection a model for integrating a rigorous thermal simulation with computational reasoning is discussed. This model is used to build an intelligent computer-aided system that assists designers throughout the design process. The model uses rigorous hierarchical thermal simulation modules linked to several databases. In addition, Artificial Intelligence Techniques are used to build a multilevel reasoníng structure for both the initial and the final design. Neural networks are utilized for the initial design stage problem detection where problems are incomplete. The neural networks are integrated with a multi-knowledge reasoning structure that utilizes the blackboard framework. This structure is used for problem detection in the intermediate to final design stages and for advice throughout the design process.

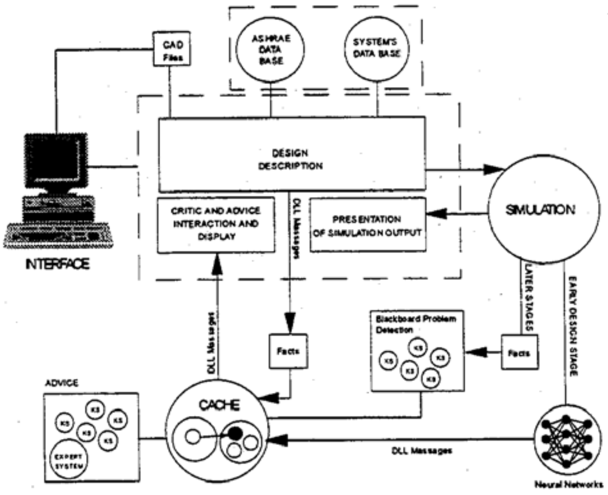


Figure 1: the system model

The system consists of four major components that are organized in an integrated framework. These components are: 1. the graphical user interface 2. The simulation program 3. The data base 4. The intelligent agents; figure (1).

The interface in this system is built to provide flexible tools to propose the design problem and to navigate and interact with the system's output and intelligent agents. This way accomplished by providing a graphical user interface that performs all necessary criteria in one environment. The interface allows the user to define the thermal design problem, start a detailed simulation, navigate the outcome, interact with the intelligent agents and based on this, allow alterations to the design. The interface is designed to allow integration with other CAD software.

The simulation in this system uses Transfer Function Method algorithm (TFM) for analysis.

The intelligent structure in the system consists of the problem detection and advise agents. The agents in the system use the blackboard model supported by a multi-knowladge statistical reasoning framework. The statistical reasoning supports modelling the uncertainty that is present within the reasoning. The control in the intelligent agent is based on a combination of a backward and forward reasoning and backtracking. Currently the problem detection agent is being revised to include an additional model that uses the neural network as its reasoning structure to support problem detection in incomplete designs. Both models are being connected to the multi-knowledge intelligent advisory structure.

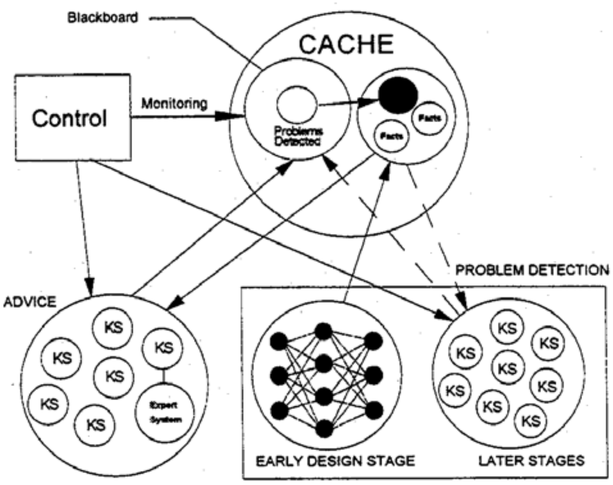


figure2: intelligent agent's structure

2.1. Control of Air Condition System

An energy saving control system of variable-air-volume air conditioner in intelligence building is formulated in this section. It reaches good control effect and energy efficiency by making the best of the advantages of intelligence building. During the cooling preparation period, neutral works is used. In the regulating period, the air volume is decided by means of feed forward control. The previous turning off period is determined in the way of penalty function.

In this paper, the predictive control of cooling preparation period of AC system in intelligence building by means of neutral network is expressed. BP network has the strong ability of adapting and self-studying. It suits to the control of those nonlinear or indefinite systems. The arithmetic could approach any nonlinear mapping relation and has strong deducing ability. So it suits to the cooling preparation predictive control. The preparation period Top is mainly determined by the outdoor temperature x I , initial indoor temperature x2, volume of rooms x3 and the power capability of the air conditioner x4.

$$T_{op} = f(x_1, x_2, x_3, x_4) \quad (1)$$

For a given building, as x3 and x4 are constant, so only x1 and x2 are considered. This object is a typical MIS0 nonlinear system. The precious mathematics model can scarcely be worked out in traditional way.

However, it can be solved by neutral BP algorithm. The Network is made up of 3 layers, the input layer, the hidden layer, and the output layer. The node number n1 of input layer is 2, and n3 of out layer is 1. n2 of the hidden layer is 7, which is determined by experience equation:

$$n_2 = \sqrt{n_1 + n_3 + a} \qquad a \in N(1,10) \quad (2)$$

As the BP algorithm has the shortcoming of slow convergence speed, so a variable studying step amendment is adopted:

$$\omega(k+1) = \omega(k) + \alpha(k) \cdot D(k) \quad (3)$$

$$\alpha(k) = 2^{\lambda} \cdot \alpha(k-1) \quad (4)$$

$$\lambda = \operatorname{sgn}[D(k)D(k-1)] \quad (5)$$

$$D(k) = -\partial E(k) / \partial \omega(k) \quad (6)$$

Where ω is the weight; α is the studying step; D is the minus grade of k time; E is the error, defined as the function of input and output sample x and d:

$$E = \frac{1}{2} \sum_{i=1}^{4} (d_i - x_i)^2 \quad (7)$$

The program is written in VC, and is trained by the data from a physical simulation system. The simulation system consists of a control subsystem, a given room and air conditioner. The studying data come from different conditions in this system, which meet the requirement of BP network. 70 sets of date are used to train the weight and valve of network. And 20 sets of date are used to detect it.

The result is proved to be good, shown in Fig.3 In practice, all the training date required can be got from the data sampling and storing system of intelligent building for the self-studying of BP network. And the predicted cooling preparation period can be figured out by it, and then air conditioner would turn on according to it.

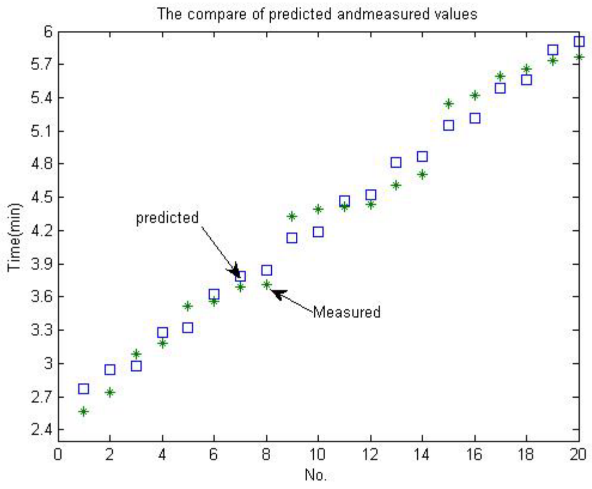


Figure3: The compare of predicted and Measured values

2.2. Feed-forward

Control in Regulating Period The main aim of AC control during regulating period is the temperature. The comfortable temperature scope is 18-28 degree, and the temperature in the buildings should be kept in it. The energy exhausted W by AC is approximately in proportion to the temperature deviation:

$$W = \int k(T_0 - T(t)).dt = (T_0 - T_{AV})\Delta t \quad (8)$$

Where To is the initial temperature, T(t) is inner temperature, TAv is the average temperature.

According to the statistics data, compared with 25 degree, if the inner average temperature is kept at 20 degree in summer, the electricity energy could be saved about 35%.

So, if the intelligence of the building is fully utilized, such as all kinds of sensors and controllers, the temperature can be regulated below the high limit, which can achieve ideal energy efficiency.

Feedback control can only reduce the error after it occurred. However, due to the time lag of the AC system and the delay of the control, the temperature would wave up and down like curve 11. So, an energy-saving method by means of feed forward control is presented, the main idea of which is to predetermine the lead time of control actions to reduce the overshoot and fluctuation of the controlled temperature.

Feed forward control is different from feedback control.

It makes decision according to the disturbance. When the disturbance occurs in a system, the forward control outputs the control value according to it and in the end to removing it.

If the arithmetic and the parameters are correct, the high control resolution and little fluctuation could be got easily, shown in Fig4. It can make an intelligent building “know” the trend of heat changing through the gate management system and energy statistic system, just like a man. In practical system, both feedback control and feed forward control is used to get good quality in the regulation of temperature in air conditioning system.

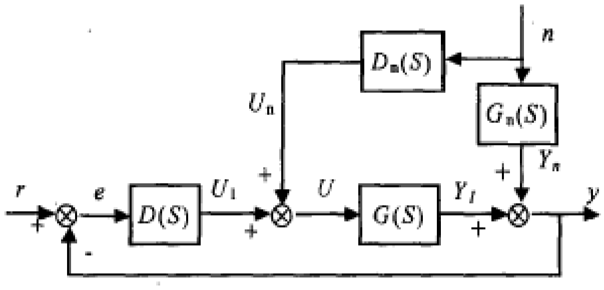


Fig4: Feed-forward Control

If disturbance N(S) could be measured, so the error Y,(S)caused by it would be eliminated by the feed-forward loop DII(S).

D,(s)G(s) + G,(s) = 0 (9)

So the feed-forward control unit should be:

If both forward and backward feedback control are used, the controller can eliminate the fluctuation caused by the measurable disturbance, and can reduce the interference immeasurable. Also the feedback can provide the function to detect the result of control. The forward and backward feedback controller can improve the control resolution greatly, and get the satisfactory control effect and energy efficiency. A digital simulation system by MATLAB is adopted to test the control of variable air volume condition system. As shown in Fig5, curve 2 has the accurate feed-forward function, where there is no error; curve 3’s feed-forward function is not accurate to the disturbance, where there is little error; and curve 1 is the pure feedback function, where there is great error.

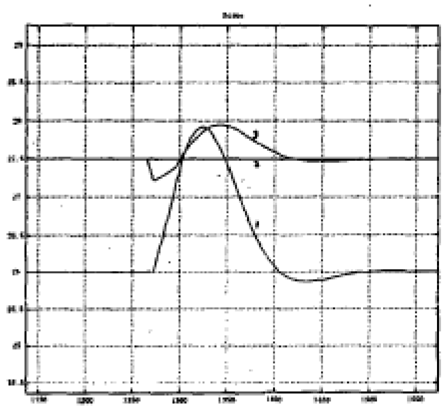


Fig.5 Simulation Results

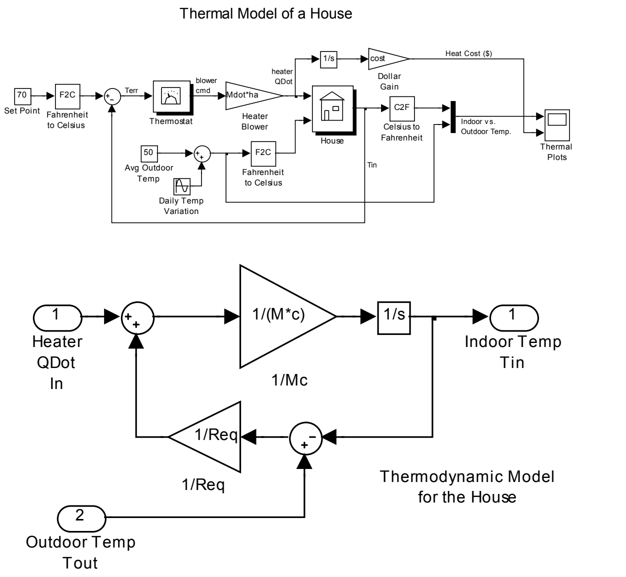


Figure6: thermal model of intelligent building for heat cost calculation.

Figure 6 is run in MATLAB software, SIMULINK environment. It can monitor the temperature of building by online mode.

3. Conclusion

Ultimately, the tools developed in this research will go beyond saving energy — they could also help building operators respond to electricity grid emergencies and to money-saving opportunities when energy costs are high. Thermal design contains many variables and constrains that must be taken into construction. These variables and constrains increase as the design evolves and make it more difficult to detect potential problems in the design.

In this paper we found a new method for thermal simulation and problem detection in the building and so have suggested a intelligent building for energy saving.

The prediction data has had a good fit with measured data’s.

Finally, for network controlling we suggest a internet-based control systems to improve the energy efficiency of buildings (IBEC). IBECS is a practical networking system that takes advantage of a building's existing IT infrastructure to control off-the-shelf lighting components and other building equipment through the internet.

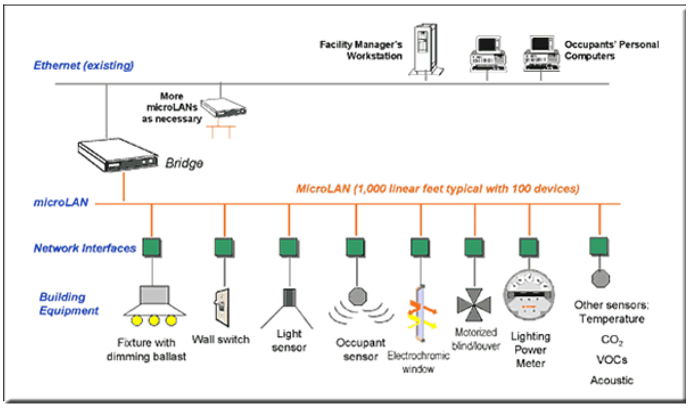

Ethernet (existing)
Facility Manager's Workstation
Occupants' Personal Computers
More microLANs as necessary
Bridge
microLAN
MicroLAN (1,000 linear feet typical with 100 devices)
Network Interfaces
Building Equipment
Fixture with dimming ballast
Wall switch
Light sensor
Occupant sensor
Electrochromic window
Motorized blind/louver
Lighting Power Meter
Other sensors: Temperature
CO_2
VOCs
Acoustic

Figure7: IBECS is a practical networking system that takes advantage of a building's existing IT infrastructure to control off-the-shelf lighting components and other building equipment through the internet.

References

[l] M.C. Jean. Scjenf$c Busis of Air Conditioning, London: Applied Science Publisher, 1981, pp.15-103

[2] M. Aucevic. “Energy Efficient Office Building Design”, ASHRAE Journal, vol.39.July, 1996, pp.26-28

[3] Liu Zuojun. The Control of AC System in Intelligent Building. Master Paper, Hebei University of Technology, Tianjin: PR China, 2000,pp.12-35

[4] Computational fluid dynamics An advanced active tool in environmental management and education R.A. Pitarma, Polytechnic Institute of Guarda, Guarda, Portugal, J.E. Ramos, Polytechnic Institute of Leiria,, Leiria, Portugal, M.E. Ferreira,, Polytechnic Institute of Guarda, Guarda, Portugal, and M.G. Carvalho, Technical University of Lisbon-IST, Lisbon, Portugal

[5] Formaldehyde as a Basis for Residential Ventilation Rates, M.H. Sherman and A.T. Hodgson, Indoor Environment Department , Environmental Energy Technologies Division , Lawrence Berkeley National Laboratory, Berkeley, CA 94720 , April 28, 2002

[6] Residential Passive Ventilation Systems: Evaluation and Design,James W. Axley

[7] Report No 17, Indoor Air Quality and the Use of Energy in Building prepared by WORKING GROUP 12

www.inive.org/medias/ECA/ECA_Report17.pdf

[8] ASHRAE'S RESIDENTIAL VENTILATION STANDARD: EXEGESIS OF PROPOSED STANDARD 62.2, M.H. Sherman, Indoor Environment Department, Environmental Energy Technologies Division Lawrence Berkeley National Laboratory, Berkeley, CA 94720, April 1999

[9] The Energy Saving Control of Air Condition System in Intelligent Building, Liu Zuojun'.2 Xie Wenlong2 Huang Yalou'

[10] Simulation and reasoning, intelligent building thermal problem detection, ali M.malkawi