Modelling and Simulation of Thermal Power Plants

Jonas Eborn

Department of Automatic Control
Lund Institute of Technology

DISCLAIMER

Department of Automatic Control
Lund Institute of Technology
Box 118
S-221 00 Lund Sweden

Document name
LICENTIATE THESIS

Date of issue
February 1998

Document Number
ISRN LUTFD2/TFRT--3219--SE

Author(s)
Jonas Eborn

Supervisor
Sven Erik Mattsson and Karl Johan Åström

Sponsoring organisation
Swedish National Board for Industrial and Technical Development (NUTEK) and Sydkraft Research Foundation.

Title and subtitle
Modelling and Simulation of Thermal Power Plants

Abstract

Mathematical modelling and simulation are important tools when dealing with engineering systems that today are becoming increasingly more complex. Integrated production and recycling of materials are trends that give rise to heterogenous systems, which are difficult to handle within one area of expertise.

Model libraries are an excellent way to package engineering knowledge of systems and units to be reused by those who are not experts in modelling. Many commercial packages provide good model libraries, but they are usually domain-specific and closed. Heterogenous, multi-domain systems requires open model libraries written in general purpose modelling languages.

This thesis describes a model database for thermal power plants written in the object-oriented modelling language OMOLA. The models are based on first principles. Subunits describe volumes with pressure and enthalpy dynamics and flows of heat or different media. The subunits are used to build basic units such as pumps, valves and heat exchangers which can be used to build system models. Several applications are described; a heat recovery steam generator, equipment for juice blending, steam generation in a sulphuric acid plant and a condensing steam plate heat exchanger.

Model libraries for industrial use must be validated against measured data. The thesis describes how parameter estimation methods can be used for model validation. Results from a case-study on parameter optimization of a non-linear drum boiler model show how the technique can be used.

Key words
Computer aided engineering, modelling, object-oriented modelling, Omola, power plant, process models, control systems, model validation, parameter estimation.

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
133

Security classification

Recipient's notes

The report may be ordered from the Department of Automatic Control or borrowed through:
University Library 2, Box 3, S-221 00 Lund, Sweden
Fax +46 46 222 44 22 E-mail ub2@uub2.lu.se

Modelling and Simulation of Thermal Power Plants

Modelling and Simulation of Thermal Power Plants

Jonas Eborn

Department of Automatic Control
Lund Institute of Technology
Lund, February 1998

Tillägnas min morfar
Torsten Svensson,
som aldrig fick chansen att se
frukten av mina studier.

⋆ 5/3 1922 † 1/1 1998

Dedicated to my grandfather
Torsten Svensson,
who never got the chance to see
the results of my work.

Department of Automatic Control
Lund Institute of Technology
Box 118
SE-221 00 LUND
Sweden

ISSN 0280–5316
ISRN LUTFD2/TFRT–3219–SE

Printed in Sweden by Reprocentralen, Lunds Universitet.
Lund 1998

Contents

Acknowledgements

First of all I would like to thank professor Åström and my supervisor Sven Erik Mattsson for always having faith in me. I am very grateful for the work they and other senior staff at the department do to acquire funding for many graduate students.

I am also forever indebted to Bernt Nilsson with whom I shared room and work for a long time, but sadly not any longer. Those were the days, my friend.

The architect of Omola/OmSim, Mats Andersson, and the maintainer of OmSim today, Tomas Schöntal, share my gratitude for providing such a good working environment and for correcting numerous bugs during the development of **K2**.

Many thanks goes to the Maple/Unix/Linux-wizard James Sørlie who worked with me during his post-doc work in Lund. Good luck in Denmark!

I also must thank all Master's students that have contributed to this thesis. Both those that contributed directly; thanks Jonas, Peter, Ola and Patrik, and others that did it indirectly by asking a lot of questions on OmSim and on modelling in general.

To everyone at the department, thanks for making this such a wonderful place to work.

Finally, I would like to express my sincere gratitude and joy for having such a great family. My wife, Pernilla, and children, Elise and Per, have always been there and provided support (in a non-technical sense). Their love is what keeps me smiling and going on.

This project has been supported by Sydkraft AB under the project name ”Modelling and Control of Energy Systems”. Some funding have also been provided by the Swedish National Board for Industrial and Technical Development (NUTEK) programme ”Complex Systems”, Dnr 96-10653. Their financial support is gratefully acknowledged.

Chapter 1

Introduction

Abstract

In this introductory chapter we give some motivation to why this work is important and where this piece fits into the jigsaw puzzle of automatic control.

Automatic control is today used in almost every technical system in all possible engineering domains. To be able to design and implement a control system, some kind of model of the real world system is needed. Modelling provides a bridge between the real world and the automatic control world, see Figure 1.1. Usually, a mathematical model is needed in all different aspects of the work done in automatic control; analysis, control design and simulation.

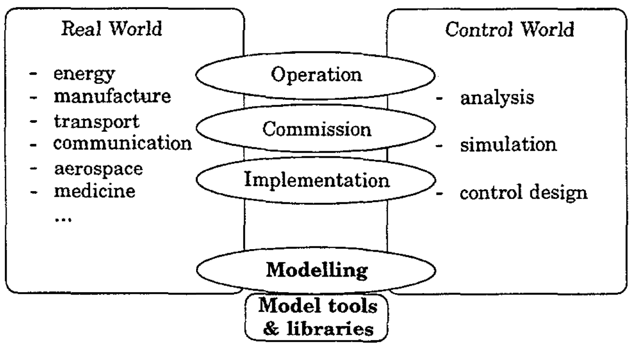


Figure 1.1. Modelling as a bridge between the real world and control.

1.1 Why Modelling?

A good model provides knowledge of a system. With the increased understanding that a model of the process can give also comes better possibilities to increase quality, safety and economy. This has been very well expressed by an executive at one of the largest companies in the process industry:

> Modeling and simulation technologies are keys to achieve *manufacturing excellence* and to *assess risk* in unit operations. As we make our plant more flexible to respond to *business opportunities*, efficient *modeling and simulation* techniques will become *commonly used tools*.
>
> Ralph P. Schenkler, Exxon

There is a need for tools that can provide insight into complex systems. Technical systems are becoming increasingly more complex mainly for two reasons,

Integration of systems introduces tight couplings where previously parts could be designed and operated independently. Integration comes from recirculation of materials to reduce energy consumption and pollution. It also comes from reduction or removal of buffers to reduce production times and increase the flexibility of the plant, as stated in the quote above.

Heterogeneity in systems forces a mixture of several engineering disciplines to be considered simultaneously. Heterogeneity is of course a consequence of integration, for example in mechatronic systems where mechanics, electronics and control algorithms interact closely, but it is also an additional difficulty, adding complexity. Different traditions of how to treat systems give rise to, for example, mixtures of continuous time and discrete time systems.

Mathematical modelling is a fundamental tool to tackle complex systems.

The main use of models of complex systems is today in simulation. Simulation is important since it provides the possibility to study the behaviour of a model and draw conclusions concerning the real world system. Simulation has been the main tool to verify the demands that should be achieved by a control system, e.g.,

- Environmental demands
- Safety demands
- Economical demands
- Quality/performance demands

The focus that for a long time has been on simulation is now changing towards analysis. Tools that from models of complex systems can extract simpler models useful for analysis is still lacking though. This is, however, outside the scope of this thesis, but some comments on such tools are given in Chapter 6.

Different Fields - Different Traditions

In different areas of engineering there have been very different traditions in what kind of models and how modelling was used. Some examples are:

Automatic Control – Block diagram modelling, either in frequency domain or with state-space models.

Circuit Simulation – Signal flow modelling, large nets with many similar components.

Chemical Processes – Static design calculations, using flow-sheeting to build process diagrams.

In this thesis we are concerned with physical modelling, which can form a basis in any of these modelling methods. For example in automatic control, where physical modelling coexists with system identification methods which also is a way to find models. In the chemical process industry physical modelling is the most natural approach, but there the focus has traditionally been on static mass and energy balances and not dynamic properties which are most important for control purposes.

Why Model Libraries?

The scope of this thesis is mainly how to build and use model libraries for thermal power systems. Providing model libraries is an excellent way to package modelling knowledge that can help others with similar problems. Good model libraries are often the primary reason to use special purpose simulation software, like Spice and Saber for electrical circuits, Adams for mechanical systems and EMTP for power systems. All of these programs provide extensive model libraries that can be used to simulate systems within their particular domain. The drawback of these special purpose libraries is that the models are closed, they can not be altered or even inspected by the user. Since they only contain models from one domain it is also difficult or impossible to use them for complex, heterogenous systems.

Ideally, there should be good model libraries available that are both open and extensible. By building model libraries for special domains in a general modelling language you provide both the domain knowledge that special purpose software has and the extensibility and possibilities for multi-domain modelling that general modelling software has. This is the goal of this thesis.

1.2 Contributions

The main contributions of this thesis are

- the development of the **K2** model database.
- the application of this database to the modelling of a thermal power plant.
- the usage of the database in a number of Master's theses, demonstrating the applicability of **K2** to a wide range of process applications.

Finally, the thesis also contains a paper concerning parameter estimation. This part is considered more a direction of future research, although it shows the usefulness of integrating software packages for modelling and optimization.

1.3 Published Papers

The work in this thesis is primarily based on three conference papers and one journal article:

EBORN, J. and H. OLSSON (1995): "Modelling and simulation of an industrial control loop with friction." In *Proceedings of the 4th IEEE Conference on Control Applications*, pp. 316–322. Albany, New York.

NILSSON, B. and J. EBORN (1998): "Object-oriented modelling of thermal power plants." *Mathematical Modelling of Systems*, **4**. To appear.

EBORN, J. and B. NILSSON (1996): "Simulation of a thermal power plant using an object-oriented model database." In *IFAC'96, Preprints 13th World Congress of IFAC*, vol. O, pp. 121–126. San Francisco, California.

EBORN, J. and J. SØRLIE (1997): "Parameter optimization of a non-linear boiler model." In SYDOW, Ed., *15th IMACS World Congress*, vol. 5, pp. 725–730. W&T Verlag, Berlin, Germany.

These articles are reprinted as Paper A-D in the second part of the thesis. Only typographical changes have been made to the articles, concerning figures and references.

All of the three conference papers were presented by the author. A shorter version of the journal article was presented by Bernt Nilsson at the EuroSim conference in Vienna, Nilsson and Eborn (1995). The rewriting of this paper into an article was primarily done by Bernt, but it is still based on joint work. The work on the last paper was done in close cooperation with James Sørlie. It uses a model definition interface developed during the same time, Sørlie (1997).

In the following chapters further comments and clarifications on the papers are given. Each chapter corresponds to one paper in order: 2–A, 3–B, 4–C. Chapter 5 is later material which is based on experiences from using the **K2** database in several master's theses. In Chapter 6 some concluding comments are given and also ideas on future work are discussed, including the ideas presented in Paper D.

Chapter 2

General Aspects on Object-Oriented Modelling

Abstract

In this chapter we introduce the fundamental concepts of physical modelling and give some examples to demonstrate the differences between formulations. Comparisons of object-oriented modelling with other methods like block-diagram approaches are also made.

2.1 Introduction

This thesis deals with object-oriented modelling of physical systems. I was introduced to this field of research through my Master's thesis which was concerned with modelling of an industrial pneumatic control valve, see Eborn (1994). The goal of the Master's thesis was to explain problems with fluctuations in the controlled pulp fibre concentration (also called consistency) using simulations of the control loop with different friction models. Paper A is a reprinted version of Eborn and Olsson (1995), based on the Master's thesis. The scope of that paper is mainly the control loop application and the examined friction models. This chapter will try to give some of the general aspects on using object-oriented modelling, which is not covered in any of the published papers.

2.2 Modelling Paradigms

Traditionally in control and in computer simulation, modelling has been made in a procedural, block-oriented manner. This tradition comes more from concern with computational aspects than from user concerns. The modeller has to perform the tedious work of transforming a physical description in terms of balance- and constitutive equations into an explicit ordinary differential equation, ODE, system

$$\dot{x} = f(t, x) \tag{2.1}$$

This form of mathematical model is almost a computational program. It is a *procedure* for calculating state derivatives. There is a lot of well-proven numerical software that can be used to solve these differential equations. Many powerful commercial simulation packages exist which use this type of models, with libraries of predefined computational blocks. What these programs offer is mostly an interface to the numerical algorithms and also a graphical way of programming a computational model, but little help in constructing the model.

Object-oriented modelling on the other hand tries to describe each part of a system as an object with a certain behaviour. This modelling paradigm is equation- or constraint-based[1]. Each object is described by fundamental physical relations, natural laws, and not by a procedural function relating inputs to outputs. With this way of modelling the user is more concerned with the interface to the model objects and the model equations than with the computational order. This paradigm relies on the symbolic methods that modern computing offers since the model equations before simulation need to be manipulated into a differential and algebraic equation, DAE, system

$$g(t, x, \dot{x}, v) = 0 \tag{2.2}$$

The difference is that this manipulation is done by the computer and not by the user. For a description of some of the symbolic methods used, see Mattsson (1995). The difference between the procedural and

[1]This is in agreement with the terminology in computer science, where *procedural* and *constraint* are classes of programming languages. In Abelson and Sussman (1985) there is a nice example of a system for constraint propagation.

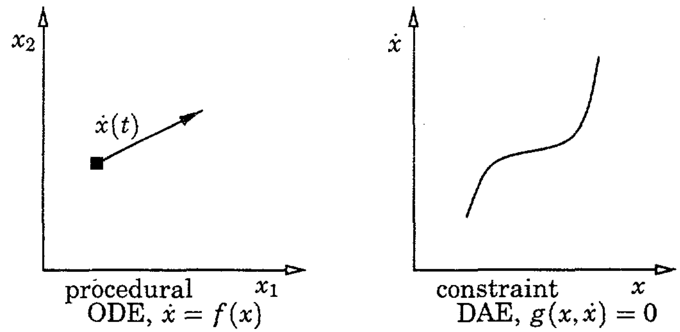


Figure 2.1. Different mathematical formulations of a model. An equation-based model defines constraints for possible behaviours in the state-space and not an explicit time-trajectory like a procedural model.

constraint formulations can be illustrated as in Figure 2.1. With the procedural formulation the user gives a function for the direction to move in the state-space. Constraint modelling on the other hand just gives a relation that defines *possible* behaviours for the system.

EXAMPLE 2.1

As an example consider the modelling of a pneumatic spring-return actuator, common in the process industry. It consists of a pneumatic chamber with a diaphragm connected to a spring. A drawing and schematic of the system together with the constitutive relations are given below. The ideal-gas law as it is used here assumes isothermal operation.

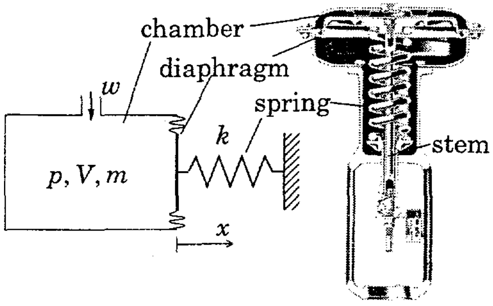
Constitutive relations:

Chamber:

$$
\begin{aligned}
\dot{m} &= w \\
V &= Ax + V_0 \\
pV &= m \cdot const \\
F &= pA
\end{aligned}
$$

Spring:

$$F = kx$$



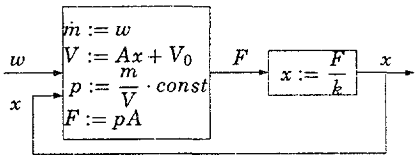


Figure 2.2. Block diagram of the pneumatic diaphragm.

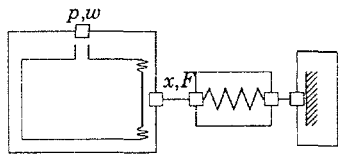


Figure 2.3. Object diagram of the pneumatic diaphragm.

If this system should be simulated in some block-diagram software, a possible representation would look like the diagram in Figure 2.2. Compare this with the object diagram in Figure 2.3. In the object diagram the connections imply relations between terminal variables, not computational causality as in the block diagram. In the block diagram some of the physical structure of the system is lost. Also note that since the force, F, from the diaphragm depends directly on the position, x, there is an algebraic loop, i. e., a non-linear algebraic equation system in the variables $\{V, p, F, x\}$. This loop is inherent in the system description, but often not handled very well by block-diagram software.

A possible extension of this model would be to account for the mass of the diaphragm and the attached stem. Then the block diagram would change into the one showed in Figure 2.4. Note that the description of the spring changes from $x := \frac{F}{k}$ to $F := kx$. The causality of the spring equation is reversed. This means that a model library with components applicable to this example would need to contain two *different* spring

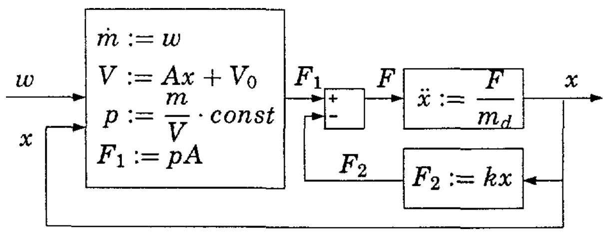


Figure 2.4. Changed block diagram of the pneumatic diaphragm, mass included.

models, depending on the computational causality. Including mass in an object-oriented model would simply mean adding a mass object between the chamber and the spring in Figure 2.3. This would not in any way affect the descriptions of the spring or the chamber. □

A key point in the previous example concerns *causality*. This is a tricky subject which can be treated philosophically, as done in *Critique of Pure Reason* by Immanuel Kant, or more pragmatically. A thorough description of causality in the context of physical modelling is found in Strömberg (1994).

The concept of causality used here is *computational causality*, which gives the order of calculations to compute unknown variables from known ones. The point of Example 2.1 is that causality is a property of the *system*, depending on the choice of inputs for a particular experiment. The spring relation $F = kx$ has no causality in itself. The computational causality is imposed on the spring by the choice of input (w) and how we choose to model the rest of the system (including mass or not). The fundamental drawback of procedural modelling is that it forces a causal description of every component of a system. Thus the component model can not be reused in the description of another system or for another experiment. *Causal* descriptions prevent *modularity*.

Block Diagram Modelling

Procedural models is the formulation used in almost all commercial block-diagram software. Building models of systems using input-output blocks has been the most common method in many engineering domains, especially in automatic control where the in-out formulation comes naturally from seeing a system as having manipulated inputs that affect the measured outputs.

Also for thermal power plants block-diagram models have been used a lot. The book Ordys *et al.* (1994) describes modelling and simulation of a general structure thermal power plant. The approach taken there is state-space modelling; breaking down the system in modules and for each of these modules determining inputs, outputs and states. This is possible through detailed analysis of each subsystem together with a global analysis of information flows. They also make some simplifying assumptions that decouples the modules, e. g., assuming con-

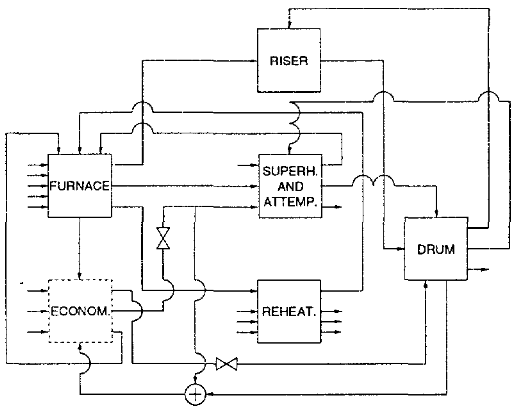


Figure 2.5. Block diagram of a boiler configuration, adopted from Ordys *et al.* (1994).

stant pressure-drop across a valve which decouples it from the pressure downstream. This kind of assumption might be appropriate in a specific application, but it makes the model of the valve *application specific* and thus not reusable in a model of another system.

Block diagrams for large systems tend to become very complex. As an example of this the block diagram of the boiler module in the Skegton unit described in Ordys *et al.* (1994) is reprinted in Figure 2.5. There are a lot of interconnections and dependencies between the blocks in the module. Each of these blocks are in turn described by block-diagrams or Fortran code.

EXAMPLE 2.2
As an illustration of the disadvantages of block-diagram modelling we look at a model for a cooling system, basically a network of tubes in which a hydraulic liquid flows. The block-diagram in Figure 2.6 is considerably messier than the system it represents, the marked part of the flow network in Figure 2.7. The problem with a procedural model of a flow network is that you need to pass information both in the forward and backward direction, giving rise to the complicated feedback connections between the line models. Another drawback is that the procedural formulation requires four variants of basically the

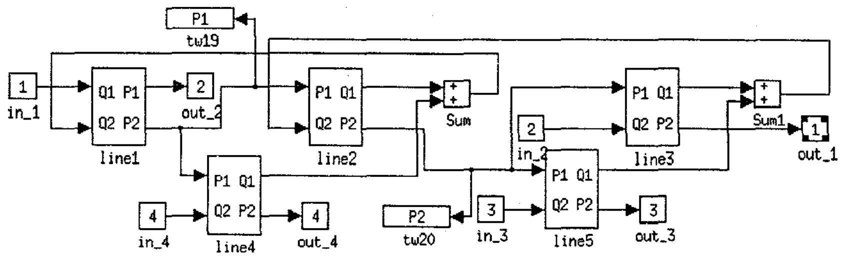


Figure 2.6. Block-diagram of upper part of the flow network in Example 2.2.

same line model, depending on which of the four pressures and flows should be computed from the others. Two of the four variants are used in Figure 2.6, other variants in the rest of the network model. The need for several model variants is a major drawback, since changes to the line model means individually updating each of the variants. Manually changing several variants like that is a very error-prone procedure.

□

One approach that can help the user working with causal modelling is to provide sorting of the equations. One of the first implementations of this was SIMNON, developed by Elmqvist (1972) at the department of Automatic Control. This approach still requires that causal equations

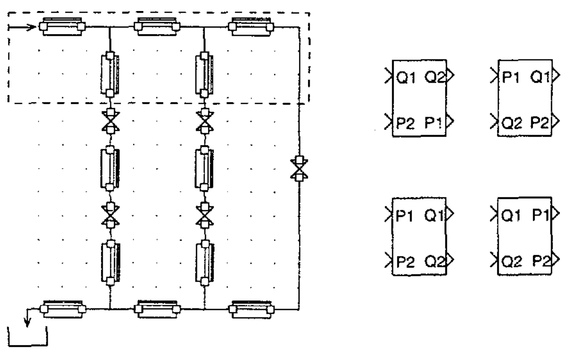


Figure 2.7. Flow network from Example 2.2 and four variants of a line model.

are specified, so you do need different models of a spring like in Example 2.1, but it can provide help with the causality between the blocks. A modern example of a software that uses this technique is EASY5®. They claim, with some right, that this is a major advantage over their competitors like Simulink® and SystemBuild®. It is true that sorting of equations makes it easier to build reusable model blocks, but still you cannot have a true model library unless you allow constraint models. Some efforts at supporting this has been announced by The Boeing Company.

Object-oriented Modelling

The modelling paradigm considered in this thesis is object-oriented modelling. It may also be called constraint modelling or truly equation-based as opposed to the procedural, block-oriented manner described above. Object-oriented is a word that is very often misused, sometimes used for graphical tools referring to model blocks as objects, other times used because the software is written in some object-oriented programming language like C++.

The "true" meaning of object-oriented modelling should be that the modelling language has some of the properties that object-oriented programming has. Properties like the *class* concept, *inheritance, abstraction* and *specialization*. In object-oriented modelling each model is treated as an object, described by a class, which can be seen as a blue-print of the model. The class has attributes which can be locally defined or inherited from a *super-class*. Attributes can be either simple variables and equations or other objects. Through inheritance a common structure of a group of objects can be defined in a super-class, while different internal descriptions are kept in the sub-classes. The sub-classes are said to be specializations of the super-class.

Abstraction is a powerful tool to support complex system modelling, it implies the possibility to use a model without detailed knowledge of its internal structure or description. Necessary information to use a model object should be kept in its interface, which includes parameters and terminal variables. The interface contains all parts of the model that can be accessed from the outside. The internal behaviour

EASY5 is a trademark of The Boeing Company, Simulink is a trademark of The MathWorks, Inc. and SystemBuild is a trademark of Integrated Systems, Inc.

description can be hidden, or *encapsulated*, and should not be accessed by other objects.

A more detailed explanation of the concepts of object-oriented modelling and of the modelling language OMOLA can be found in Andersson (1994).

Bond-graph Modelling

An older but interesting paradigm that has its basis in physical analogies between different energy domains, like electrical and mechanical, was introduced by Paynter (1961). It is called *bond-graph* modelling and the name comes from that it is a graphical description of systems using bonds between elements. Each bond describes the power flow, which is the product of two conjugate variables, e. g., current and voltage in the electrical domain and velocity and force in the mechanical domain. Physical analogies show that capacitors and compliances (springs) can be described by a common C-element storing flow (current/velocity), and that inductors and inertias (masses) can be described by an I-element storing effort (voltage/force). An excellent book on modelling which describes bond graphs without being devoted to them is Cellier (1991).

Bond graphs are very useful for simpler systems in the domains where there are natural power variables since graphical analysis methods exist that provide a lot of information of the system. For example, there are methods to automatically derive the computational causality of a bond graph and transform it to ODE form (2.1). However, bond graphs do not work that well in all domains, for example in thermodynamics you need to describe the energy interaction in up to three layers, creating a multi-layer bond graph, since you must simultaneously consider mass, energy and momentum flows.

Chapter 3

K2 – Thermal Model Database

Abstract

This chapter is a discussion of the contents in the article Nilsson and Eborn (1998), reprinted as Paper B. In the article a set of basic model libraries necessary for building models of thermal power plants is presented. Here we discuss the modelling principle used; the separation of processes into dynamics in control volumes and static relations in flow modules. These two types of subunits can be used to build models of physical equipment, units. The structuring of these models in the model database **K2** is also discussed.

3.1 Introduction

The main contribution of this thesis is the building of the **K2** model database and showing that a model database is useful in modelling of thermal power and other process applications. The basic modelling principle in **K2** is that physical phenomena in thermal systems can be divided in compartments or nodes with dynamic behaviour and flow modules or branches with static behaviour. This separation idea was proposed by Jan Tuszyński at Sydkraft Konsult AB and it is further discussed below. Some work in the line of this was done by Holmberg (1992) and Bernt Nilsson before I started working on what later became **K2**.

3.2 Modelling Principles

Thermodynamic systems are described by three conservation laws; the conservation of mass, energy and momentum. In their general form they are given by partial differential equations in three dimensions. This is not a useful approach for our purposes, instead we consider lumped differential equations for a discrete *control volume* where properties only vary along the flow direction, see Figure 3.2. This gives us three differential balance equations in m, e and mv:

$$\begin{aligned}\frac{dm}{dt} &= w_{in} - w_{ut} \\ \frac{de}{dt} &= w_{in}h_{in} - w_{ut}h_{ut} - W_s + Q \\ \frac{d(mv)}{dt} &= w_{in}v_{in} - w_{ut}v_{ut} + \Sigma F\end{aligned}$$

Explanations of the symbols used are given in Appendix F.

The momentum dynamics, which determine the flows in the system, are typically an order of magnitude faster than the other dynamics in medium to large power plants. This is illustrated in Figure 3.1. The control system also acts on the slower time-scale since main controlled variables are water levels and pressures in the power plant. Thus the faster dynamics are of less importance in a model built for the purpose of control analysis and design. When the frequencies of the dynamics are of different orders of magnitude the faster dynamics can be neglected and replaced by static relationships. This has been done for the momentum dynamics in Appendix E.

When we approximate the momentum dynamics with static flow relations we can split up the system equations into two types of *sub-*

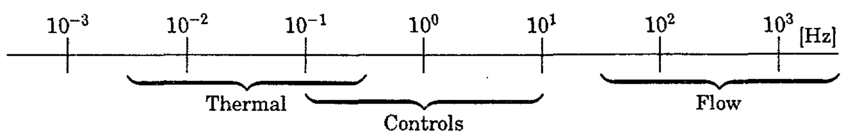


Figure 3.1. Frequency ranges of different dynamics, adopted from Hyllseth (1991).

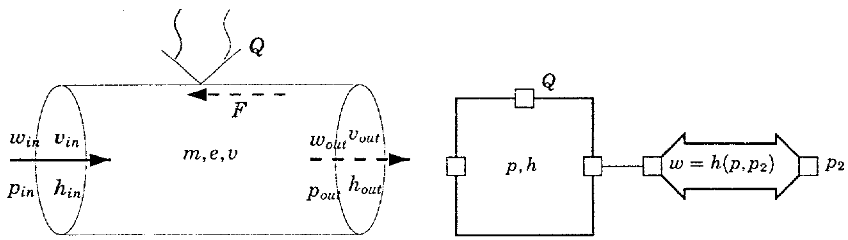


Figure 3.2. Control volume considered for the lumped balance equations and the same volume separated into one compartment and one flow module.

units, dynamic compartments and static flow modules. These are the basic model building blocks in the **K2** database. The dynamics in the compartments describe the mass and energy balances. To simplify calculations and calls to steam table functions, the balance equations have been rewritten into state equations in pressure and enthalpy. These derivations are also given in Appendix E.

3.3 Model Structuring

In hierarchical modelling of complex systems, models and behaviours are built up from simpler component models. Different users are working on different abstraction levels in a model hierarchy. The levels are illustrated in Figure 3.3. The separation into different abstraction levels is a support for a multi-user environment; the system designer works on the topology level, using unit models to build his plant, a unit model developer works on the behaviour level with subunit models, while descriptions of phenomena are developed by modelling experts.

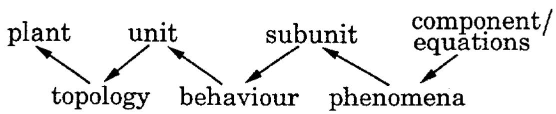


Figure 3.3. Abstraction levels in hierarchical modelling.

The abstraction levels are also reflected in the inheritance structure of the model class tree. Structuring of the **K2** database is based on the structuring guidelines given in Nilsson (1993). The abstraction levels are reflected in the *granularity* classification, {system, unit, subunit, component}, seen in Figure 3.4. The class tree shown here only displays the classes used in the simple example in the article. The higher structural levels, *application* and *granularity*, are used mainly for building up the class tree. A third *conceptual* level is used as the basis for the division of the database into a set of libraries. This is not shown in the figure since the division in this case coincides with the *interface class* level which describes the terminal interaction between different classes. The classes in the last *model class* level contain the actual model equations. This level is sometimes subdivided to be able to collect equations common to a group of classes into one superclass.

The structural levels have been used as suffixes in a naming convention for classes in the libraries. All class names end in `AC`, `GC`, `CC`, `IC` for *application class* etc. The suffix `FM` means *full model* and is used for model classes that can be used as sub-models in a system. The suffix `MM` stands for *medium model*.

Terminal Structure

A part of the model libraries that is very important for the usability of the libraries is the terminal structure. Terminal classes define the possible types of interaction between components in models. With a well-defined terminal structure increased reusability is obtained since unit models then can be easily connected.

The two major types of interaction occuring in thermodynamics are medium flows and heat flows. These are defined by the `FlowInTC`

```
FlowInTC ISA RecordTerm WITH
  w ISA MassFlowInTC;
  p ISA PressureTC;
  h ISA EnthalpyTC;
  M ISA WaterMediumTC;
END;
```

```
WaterMediumTC ISA MediumTC WITH
  q ISA PhaseTC WITH
    value := 'Water;
  END;
  z ISA HeightTC;
END;
```

Listing 3.1 Flow terminal definition (left) and medium terminal for water (right). The name suffix TC means Terminal Class.

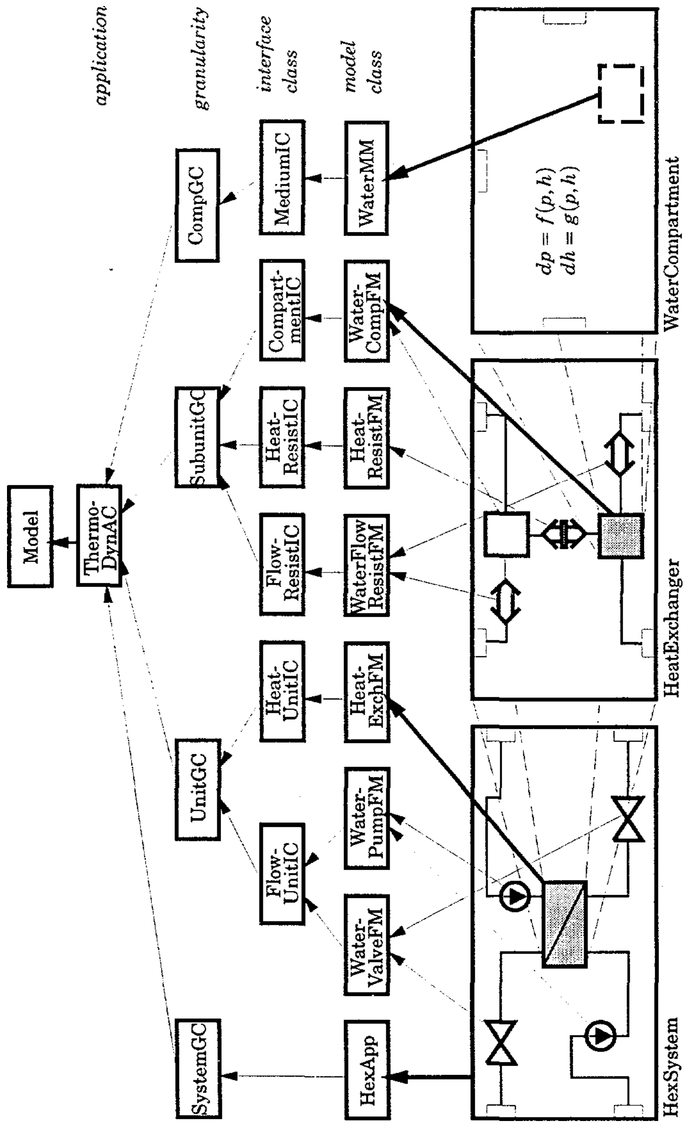


Figure 3.4. Class tree showing the inheritance structure of the **K2** database.

```
HeatRateInTC ISA RecordTerm      HeatMediumTC ISA RecordTerm
WITH                             WITH
  q ISA HeatRateTC;                p ISA PressureTC;
  Tin,                             w ISA MassFlowTC;
  Tout ISA TemperatureTC;          Gmix ISA GasCompositionTC;
END;                             END;
```

Listing 3.2 Heat flow rate terminal (left) and heat medium terminal (right).

and `HeatTransferInTC` terminal classes. These classes are structured terminals which contain sub-terminals for necessary interaction variables. The `FlowInTC` class is shown in Listing 3.1 as an example. Besides the physical variables pressure, mass flow and enthalpy, the flow terminal also contains a medium specific terminal, `M`. This in turn is also structured and contains the name of the medium, [Water, Steam, Flash, Gas], and other medium specific properties. The medium terminal is very useful since it provides consistency checking when units in a plant model are connected. The medium must be the same on both sides of a connection, conversion from one medium to another can only occur within a unit, such as a boiler or a condenser. Other properties that can be introduced in the medium terminal are for example height for water medium and temperature for steam and flue gas. Height gives pressure differences for a dense medium like water but can be neglected for the others, temperature is propagated from compartments to flow modules to avoid unnecessary calls to steam tables. The medium specific terminal also adds adaptability. For example in applications where the gas composition is changing due to reactions a composition description could be included in the medium terminal.

Heat interaction is described with the `HeatTransferInTC` class that consists of two sub-terminals for rate and medium information, these terminals are shown in Listing 3.2. The rate terminal contains heat flow, q, and two temperatures, T_{in} and T_{out}. Both temperatures are necessary for calculation of the logarithmic mean temperature, which is normally used for heat flow calculations. The medium terminal holds quantities that are important for calculation of heat resistances.

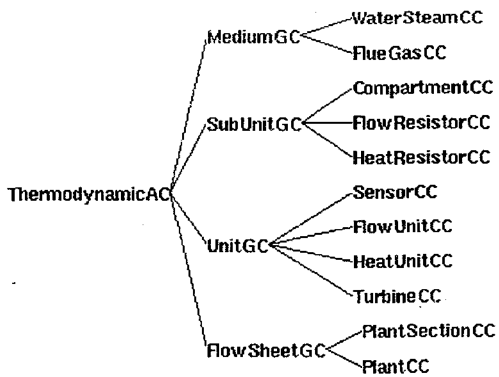


Figure 3.5. Classes in the K2ClassTree library. Leaves in the tree correspond to libraries in the database.

3.4 K2 Libraries

The model libraries are grouped according to the granularity concept in three main groups: unit, subunit and component libraries. In each of these there are a few libraries. The structuring of the libraries is reflected in the inheritance tree shown in Figure 3.5. This tree is built up from classes in a special library, `K2ClassTreeLib`. Below follows short descriptions of models in the libraries and the regions of validity for the models. More descriptions and code listings can be found in Nilsson and Eborn (1994).

Unit Libraries

K2FlowUnitLib contains models of flow equipment, which are all static. Some examples are:

`WaterPumpFM` is a simple pump model relating pumping power to flow and pressure.

`WaterValveFM` is a valve model with linear characteristic, valve opening $y \in [0, 1]$. There is also `CritValveFM` for steam with large pressure drops where flow can become critical.

`FlowJunctionFM` and `FlowSplitFM` are for merging and splitting flows with perfect mixing.

`SprayTempFM` is a model of an attemperator where water is sprayed into a steam flow. Perfect mixing and instant evaporation is assumed.

K2HeatUnitLib is a library of various heat exchanging equipment:

`HeatExchangerFM` is a basic model of a heat exchanger as shown in Figure B.2. There is also a more complex model with three sections called `HeatExchangerFM3`.

`SuperHeaterFM` is a heat exchanger with flue gas on the primary side and steam on the secondary side. Another specialized model is `EconomizerFM`, which is a water-gas heat exchanger.

`BoilerFM` is a model of a drum boiler with evaporator, drum and a PID level controller. Another complex unit consisting of several units is the `SuperHeaterSystemFM`, a two stage super-heater with spray attemperator and temperature control.

K2SensorLib should contain models of measurement equipment. Only a few models are available:

`TempSensorFM` describes a simple temperature sensor with first-order dynamics.

`FlowTransmitterFM` is a static flow sensor model.

K2TurbineLib contains a steam turbine model which uses isentropic thermal efficiency, η_t, to calculate the produced power. Pressure dynamics are included through a small steam volume.

Subunit Libraries

K2CompartmentLib holds different control volume models. Most of them use the same state equations, derived in Appendix E, but different medium models, e.g., `Steam`, `Gas`, `Flash` and `DrumCompartmentFM`. The last one is different in that it also has liquid level calculations. Other models are:

`WaterCompartmentFM` which uses an incompressible water medium model and thus has no pressure dynamics.

`OpenCompartmentFM` which is a model of an open container at atmospheric pressure. It has volume dynamics instead of pressure.

K2FlowLib holds the static flow resistor models used for valve and tube models. Some examples are:

`WaterFlowResistorFM` is a model for incompressible flow. This approximation is also used for compressible media, e. g., `Steam-` and `GasFlowResistorFM`. The flow error is 1-2% when the pressure drop is less than 5% of the total pressure.

`CriticalExpansionM` can be used for large pressure drops since it handles both sub- and super-critical flows of compressible media.

`TubeLossFactorFunction` is a model for frictional losses in cylindrical tubes. It can use either the class `LaminarFrictionFactor` or `TurbulentFrictionFactor` to calculate the losses.

K2HeatFlowLib contains different heat transfer models, all but one use the logarithmic mean temperature difference, ΔT_{lm}, for heat flow calculations.

`HeatTransferFM` is a simple model using a constant heat transfer coefficient and plain temperature difference. `HeatTransfer2FM` and `HeatTransfer2FM` instead use ΔT_{lm} for parallel and countercurrent flows respectively.

`HeatResistorFM` is a complex model using separate sub-models for convective and conductive heat resistances in boundary layers and metal wall.

`MetalResistorFM` is similar to `HeatResistorFM` but also has temperature dynamics to account for thermal inertia of the metal wall.

Component Libraries

K2MediumLib contains medium descriptions of water, steam, flash and flue gas.

`WaterVarMM`, `SaturatedWaterSteamMM` and `SuperHeatedSteamMM` use calls to steam table functions to calculate temperature, density, ρ, and partial derivatives of ρ. Their validity depends of course on the steam tables, which are described below.

`WaterConstMM` assumes incompressible water and constant density.

`FlueGasMM` uses the ideal gas law to calculate density for mixtures of N_2, CO_2, O_2, Ar, SO_2, NO and H_2O. Gas temperature is given by the polynomial function `Th` described below.

`SteamFlowMM` and other flow medium models are used in heat and flow calculations. They encapsulate properties like viscosity, μ, conductivity, λ, and heat capacity, C_p, which are needed for flow calculations. The properties are given by polynomial expressions which are valid over the interesting temperature ranges, 50-300°C for water and 100-500°C for steam.

K2ClassTreeLib holds the basic classes that build up the class tree, see Figure 3.5.

K2TerminalLib contains all terminal classes used for heat and media flows. Important terminals were described in Section 3.3.

K2EndTerminalLib has special terminal classes that specify pressure or mass flow into a system. These classes are used to specify the *boundary conditions* of an experiment. There are many possible combinations that are not of practical use, but only a number of interesting ones are in the library. Examples are: `WaterCompWoutTC`, `SteamCompWinTC`, `GasFlowPinTC` and `SteamFlowPoutTC`. The name of each class reflects its use. It consists of Medium + Connected subunit + Quantity + Direction.

K2BasicLib contains some basic variable classes used as functions.

`LogMean` calculates the logarithmic mean temperature difference. For small differences it uses an approximation that has better numeric properties.

`Th` is a polynomial approximation of flue gas temperature as a function of enthalpy. It is used to avoid implicit calculations since the steam tables only have the inverse function. The accuracy of the approximation is ±0.1°C over the temperature range 100-700°C.

`WaterResistance` gives the convective heat resistance, $1/\alpha$, for water flowing in a cylindrical tube. Similar classes are `SteamResistance` and `GasResistance`.

`WallResistance` gives conductive heat resistance through a metal wall.

Region of Validity for Steam Tables

Almost all of the models in the libraries listed above make use of calls to steam table functions to obtain medium properties, although these function calls are encapsulated in the medium models. This makes the validity of the steam tables important. The implementation used is a C-code version of the IFC 1967 standard. This is a somewhat old standard but the implementation has the advantage that almost all properties are calculated directly from the states, pressure and enthalpy. No iterative searching in the steam tables is necessary, which would slow down simulations. Drawbacks are that the standard is defined in different regions which are discontinuous and that the implementation lacks functions for derivatives of properties. The discontinuity of the regions means that a model can not be valid over a phase boundary, the library must contain separate models for super-heated steam, two-phase mixture of steam and water and sub-cooled water. The property functions are also somewhat "shaky" close to the phase boundaries and this can give rise to numeric difficulties. The lack of derivatives has been handled by computing numeric differentials, these are numerically not good close to the phase boundaries.

Besides these drawbacks the steam tables are valid for a wide range of pressures and temperatures. They are not valid below 0°C and 1000 Pa or in critical or supercritical states, over 220 bar.

3.5 Pros and Cons of an Object-Oriented Model Database

The obvious benefit from using object-oriented ideas in a model database is that it allows for true reuse of model classes. Only one version of the model class is kept in the model database. This class can then be reused in many instances in different system models.

A benefit from using model databases is that it enables inexperienced users to model larger and more complex systems in a shorter time-span. This has been demonstrated by the many Master's thesis projects that have been using the **K2** libraries. Some of them are presented in Chapter 5. Master students often have no prior training in physical modelling when they start their projects, nevertheless they succeed in modelling fairly complex systems in 4-5 months.

A drawback is that users that don't have the experience gained from learning modelling in a bottom-up fashion can connect models in a way that was not intended when the model library was built. This generates low-level errors (missing or redundant equations, inconsistencies etc.) that the inexperienced user has little possibility of understanding. In Master's projects this experience has to be taught by the supervisor, which makes it necessary to give a lot of help in the beginning of a project. For commercial model libraries this kind of teaching is not feasible, so the problem calls for some other solution. Some connection and consistency checks can be built into the model library, making it possible to detect errors early in the modelling process. Possibly, more advanced symbolic algorithms could also be used to generate high-level error messages and pin-point problems in the model more exactly.

Industrial Experiences

An early version of the **K2** model database was tested and evaluated by industrial users, Tuszyński (1995). The report finds significant advantages with object-oriented methods but also some drawbacks with the organization of the models in the database.

One specific drawback with object-oriented modelling that was reported is the depth of the models in a library. The use of inheritance in several generations sometimes obscures what the actual equations of an object are. This has been avoided in the **K2** libraries by only having model equations in the last two (sometimes three) generations of a model. On the top levels in the inheritance tree only structuring and interface information is kept. The problem could also be avoided by having a model browser that collects and displays all local and inherited equations simultaneously.

Chapter 4

Modelling and Simulation of a Power Plant

Abstract

In the first application of the **K2** model database, the water-steam cycle of an experimental power plant in Värnamo was modelled. The work was first presented in Eborn and Nilsson (1996), found in Paper C. Since the article focuses mainly on the model and the simulation results, this chapter discusses another important issue; parameterization of unit and plant models.

4.1 Introduction

This chapter concerns the first of a number of applications where the **K2** database has been used to build an application model. Other applications are described in the following chapter. The purposes of these applications have been several. Firstly, to see that **K2** could be used to build a model of a power plant and to verify the behaviour of the models in the database by simulation. Secondly, to allow the database to grow by "evolution". Each new application contains new features that need to be modelled and these new models are then added to the database. Another motivation has been to study how the "ordinary user", i. e., master students, work with **K2**.

As the first major application of **K2** a model of part of Värnamoverket power plant was built. Värnamoverket is a small experimental plant which uses bio-gas produced from wood chips as fuel in a gas turbine. It is a combined cycle plant, the hot flue gases from the gas turbine is used to generate steam which runs a steam turbine and also gives waste heat that can be used for district heating. The part of Värnamoverket modelled in the project is the complete water-steam cycle, i. e., the steam generator together with turbine and condenser. The model has the same structure as the actual plant, although some details concerning the preheater and the condenser cooling has been neglected. However, the plant parameters and variable values have not been taken from Värnamoverket, since these values were not available at the time of writing the article.

Please note that the word *pan* is used instead of steam generator throughout Paper C. This is a mix-up with the Swedish word *panna* which means boiler. Although this is a bad (or good) example of Swenglish, it has not been changed. The article is reprinted in its original state.

4.2 Parameterization

It is very important to have a natural parameterization of a model. Users of a library should be able to enter parameters in a familiar way and not be forced to do manual recalculations to enter parameters on some specific form. At the same time it is important not to have too many different versions of a model since it is confusing to have a large number of similar models to choose from. The flow equation of a valve model is always basically the same, while you can imagine a lot of different valve models, see the example below.

Valve Models

The basic flow equation for incompressible flow through a valve can be written as, (E.8),

$$w = A(y)\sqrt{\frac{2\rho\Delta p}{z_{loss}}}$$

where the flow area depends on the valve opening, y. Two different expressions for the flow area and possible parameterizations are shown in Table 4.1.

Valve type	Area expression	Parameters	
Linear	$A = A_{max} y$	A_{max}	maximum flow area
	$A_{max} = \pi d^2/4$	d	flow diameter
Logarithmic	$A = A_{v0} e^{y \ln(R)}$	A_{v0}	minimum flow area
		R	rangeability
	$A_{v0} = K_{v0}/3600$	K_{v0}	valve coefficient
	$K_{v0} = K_v/R$	K_v	valve coefficient

Table 4.1 Examples of parameterizations of flow area calculations in valves.

In both of the given examples the model is over-parametrized; since there are relations among the parameters the user is not free to set all their values. Instead, the user may set the parameters he is familiar with. In this way several different parameterizations may be used in parallel. In the case of a logarithmic valve the usual parameters to set are K_v and R but you may also set A_{v0}, R or K_{v0}, K_v and you can imagine other parameterizations with diameter too. To have this freedom requires that the modelling tool has support for solving parameter equations. In OMOLA parameter equations are allowed, but the current implementation of OMSIM can only solve parameter equations that are linear in the unknowns. In the linear valve example it is thus not possible to specify A_{max} and solve for d.

The new modelling language Modelica™, Mattsson *et al.* (1997), has introduced an interesting new concept, *class parameterization*. This concept allows parameters to determine the class of an object in a model. For the valve example above this could be utilized to have several function classes with different parameterizations to specify the flow area of a valve. Which function class to use is then specified when an instance of the valve model is created.

Heat Exchangers

There are very many different types of heat exchangers and heat exchanger configurations, e. g., tubular, shell-tube and plate heat exchangers in either parallel, counter- or cross-flow arrangements. Con-

Modelica is a trademark of the *Modelica Design Group*.

Type	Conductance expr.	Variables and parameters	
Tubular with convective and conductive heat resistances.	$k = \frac{1}{U_1+U_{wall}+U_2}$	U	heat resistance
	$U_i = 1/\alpha_i$	α	heat coefficient
	$\alpha_i = Nu_i\lambda_i/d_i$	d	diameter
	$Nu_i = C_i Re_i^{m_i} Pr_i^{n_i}$	Re, Pr, Nu	dim.less groups
		C, m, n	constants
	$U_{wall} = \ln(\frac{d_1}{d_2})/\lambda_w$	λ	heat conductiv.
Part load model for gas-water heat exchangers.	$k = k^0 \cdot (\frac{\omega_g}{\omega_g^0})^m (1 - \frac{(T_g^0 - T_g)}{2000})$	ω_g	gas mass flow
		m	geometric const.
		T_g	gas mean temp.

Table 4.2 Examples of calculation models for heat conductance, k, in two different heat exchanger types. The part load model is taken from Kehlhofer (1978). Superscript 0 denotes values taken at the design operating point (full load).

sequently there is also a need for many different parameterizations of heat exchanger models. However, the basic heat flow equation can, at least statically, be written as

$$Q = kA\Delta T_{lm}$$

where heat flow depends on heat conductance, area and a suitable temperature difference (logarithmic mean in this case).

As examples, two different expressions for the heat conductance are shown in Table 4.2. In this case the different models are not over-parametrized, but they are very different in structure. The simpler part load model is very useful for practical models of heat exchangers in combined cycle power plants, where measured data and design calculations are available, but not detailed information of the heat exchanger geometry. The more complex heat resistance model is useful for more theoretical calculations from geometric information, but also for heat exchangers where measured Nusselt number correlations are available, for example for plate heat exchangers.

For structuring of models with different calculation models for heat conductance, the *class parametrization* concept mentioned in the previous section would be very useful. In the current OMOLA implementation of **K2** different heat transfer models are encapsulated in heat flow modules. Since they are polymorphic they can replace each other in a heat exchanger model, but not quite as easily as with Modelica class parameters.

Chapter 5

Using K2 for General Process Applications

Abstract

This chapter illustrates how the **K2** libraries have been used to build models of many different processes. The applications cover several different areas: In-line juice blending, sulphur dioxide production and plate heat exchangers in steam applications. The work has been carried out in a number of Master's thesis projects.

5.1 Introduction

As mentioned in Chapter 3 one of the great benefits of using model libraries is that it enables relatively inexperienced users to model rather complex systems and achieve results in a short time-span. This benefit is demonstrated by a number of Master's thesis projects that have been completed at the department, supervised by the author. The Master's projects that have used the **K2** model database are

- Accuracy Verification of a Continuous Juice Blending Process using Simulation, Klevhag (1996).
- Modeling of the Steam Generation in a Sulfuric Acid Plant, Stojnic (1997).
- Modelling and Control of a Plate Heat Exchanger in Steam Applications, Löfgren and Svensson (1997).

At the time of writing this, there is also an ongoing Master's project concerning modelling of heating, ventilation and air conditioning systems, HVAC.

These projects span a relatively large application area. What they have in common is that they are all *process* related, i. e., treating some process liquid or gas. The **K2** database was developed for thermal power applications, but the unit models are general enough to cover also other process applications. Each of these projects has its special features though and modifications to the library models have therefore been necessary. In the following sections the projects and their special features are described. These descriptions serve to illustrate the use of a general model library and how it can be adapted to specific needs.

5.2 Alblend In-line Juice Blending

The purpose of the Alblend project was to evaluate a new type of continuous juice blending equipment called Tetra Alblend® which was being developed at Tetra Pak Processing Systems AB during the course of the project. The main focus of the evaluation was the performance and accuracy of the concentration control equipment. The model of the Alblend system was also to be used to investigate production limitations and possibilities for redesign, e. g., if it could be possible to use a smaller buffer tank with tight level control.

The Alblend system is shown in Figure 5.1. The subsystems to the left supply mixing water, juice concentrate and recirculated waste juice to the mixing system. Concentration control is done with the mixing valve V60 based on measurements from the concentration sensor BxT. There is also level control on the buffer tank in the lower right corner. The mixed juice leaving the system to the right goes on to equipment for pasteurization and packaging.

The special features of this project included:

- A different process liquid, juice, instead of water calls for a concentration description in the models. At the same time there is no need to describe temperature in this system, so concentration can replace enthalpy which is accounted for in **K2**.

Alblend is a registered trade mark of Tetra Pak Processing Systems AB.

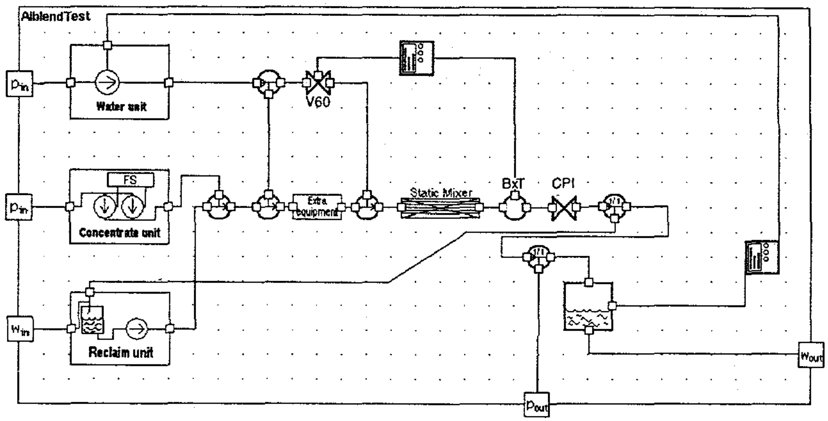


Figure 5.1. Omola model of the Tetra Alblend® system.

- The juice concentration also changes the density of the liquid which called for slight changes in the medium description.
- The pumps in the system were all speed controlled; centrifugal pumps for water and reclaim and so called positive displacement pumps for juice concentrate. The speed control required inclusion of speed dynamics in the pump descriptions. Pump-characteristics were also taken from pump-specific data for each of these.
- A flow supervisor, FS, was included in the concentrate unit model. It is a discrete control equipment used to control one small and one large displacement pump to give a concentrate flow which is a fraction of the measured water flow. Since this has no connections to the **K2** models it will not be described here.

Juice concentration description

The inclusion of juice concentration in the models for the Alblend project shows how easy it is to include new features into a model library in an object-oriented setting. The concentration description only needs to be added in some of the model classes in the model library. Primarily, it should be added in the terminals describing product flow,

```
FlowInTC ISA RecordTerm WITH
  w ISA MassFlowInTC;
  p ISA PressureTC;
  M ISA WaterMediumTC;
  C ISA SimpleTerminal;
END;
```

```
FlowResistorIC ISA Model WITH
terminals:
  Fin ISA FlowInTC;
  Fout ISA FlowOutTC;
  Fin.C AT Fout.C;
END;
```

Listing 5.1 Concentration description in flow terminals and propagation of concentration in flow modules.

see Listing 5.1. The definition of `FlowInTC` is the same as in **K2** but enthalpy, `h`, has been taken out and been replaced by concentration, `C`. Concentration is described as a simple terminal without any unit or quantity since it is dimensionless.

When the terminal definition has been altered, equations describing propagation and concentration dynamics should be added to unit model classes, basically of two types; flow modules and control volumes. In the flow modules no concentration dynamics is needed, the inflowing concentration is just propagated to the outflow. This equation can then be included in an interface class, `FlowResistorIC`, which is the common super class of all flow modules, see the right side of Listing 5.1. Including the concentration description in just one class will then take care of all different valves, pumps, etc. In control volumes like the static mixer and the buffer tank model, differential equations describing the concentration dynamics need to be added.

With this relatively small number of changes in the model library, juice concentration is included in the model library. In the Alblend project, changes to the model library were made prior to building the system model. However, the same technique could be used to include concentration in an existing system model, since a system model is just an instance of the classes in the model library.

Pump characteristics

The volumetric flow rate through a centrifugal pump is a non-linear function of both speed and pressure rise (also called head) over the pump. In a model of a pump you need to give a two-dimensional function for the pump characteristics, relating the three variables q, n and Δp. Pump characteristics are usually given as curves in a flow versus pressure chart for a couple of fixed speeds. These curves can be used to find the two-dimensional function needed.

In the Alblend project the curve for maximum speed was used to find a polynomial expression for pressure rise

$$\Delta p(q) = c_1 q^2 + c_2$$

By appropriate transformation of Δp and q with the current speed, this expression could be used to determine the flow rate at any given pressure and speed. The transformations were taken from three curves given in the data sheet for the pump.

Another common approach to describe pump characteristics is to fit parameters to the function

$$\Delta p(q) = k_1 n_{rel} + 2k_2 n_{rel} q - k_3 q^2$$

where $n_{rel} = n - n_{min}$ is relative to some minimum speed. The parameters can be fitted with a least-square method. Using this method on the Alblend water pump gives results similar to those obtained with the method described above. The squared error when compared to points from the data sheet is a little smaller. This comes at the expense of not getting $\max(\Delta p)$ when the flow is zero as in (5.2), which would be natural. However, the only significant differences are at low flow and high speed, where the pump seldom is operated.

5.3 Sulphur Dioxide Production

A second Master's project was a cooperation with Kemira Kemi AB in Helsingborg. Kemira manufacture both sulphuric acid and sulphur dioxide, which is used for bleaching in the paper industry.

Sulphur dioxide is obtained when atomic sulphur is burnt at very high temperatures, 1800°C. There is also a very large amount of heat released in the process. The heat is used to generate steam and electric power, much in the same way as in a conventional thermal power plant. The power production also has large economical implications. In fact, the power produced gives up to 50% of the net profit, which makes this a very important aspect of the process.

The goal of the Kemira project was to develop a model that increased the understanding of the dynamics in the process. In a previous Master's thesis at another department a static model of the combustion process had been developed. This was included in a Visual Basic program for the operators to use in planning of the production. A static model is of little use during transients like a start-up of the whole process. Maintaining operator experience in transient behaviour is also difficult since start-ups only occur once every two or three years. The process is in continuous operation even during large holidays. A long-term goal at Kemira Kemi is to obtain a simulator for operator training that can capture the kind of transient behaviour that occurs during start-ups.

The model of the Kemira plant is shown in Figure 5.2. It is a process very similar to a thermal power plant, with burner, steam generator and boiler drum, and thus very suitable for the **K2** models. There are two distinct differences:

- The flue gas composition is different since the plant burns sulphur instead of ordinary oil or gas. The flue gas contains both sulphur gas and sulphur dioxide.
- The burning of sulphur also gives very high temperatures in the steam generator compared to gas-turbine plants. At high temperatures radiative heat transfer must be taken into account.

Gas composition

In the **K2** libraries flue gas composition is described by a composition vector,

$$x = [n\mathrm{N}_2, n\mathrm{CO}_2, n\mathrm{O}_2, n\mathrm{Ar}, n\mathrm{SO}_2, n\mathrm{NO}, n\mathrm{H}_2\mathrm{O}]$$

where each component is the molar fraction of that substance. The composition vector includes the most important components found in air and flue gas coming from fossil fuels. The vector can also be used in calls to table functions calculating temperature, specific heat etc.

The flue gas from combustion of sulphur also contain most of these components, but since the reaction is kept under oxygen deficit there

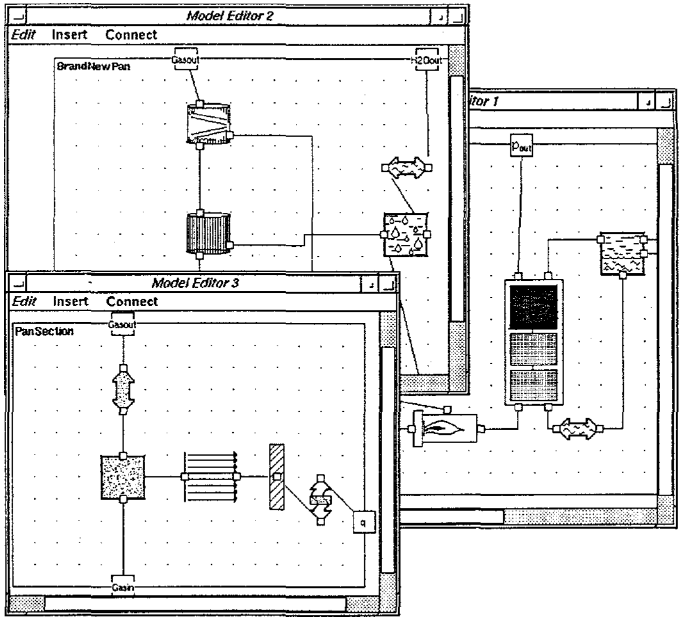


Figure 5.2. Model view of the Kemira plant with details of the waste heat steam generator and one of the sections.

is also some unburnt sulphur remaining in the flue gas. In the composition vector we let the concentration of nS_2 replace nAr since argon is unimportant,

$$x = [nN_2, nCO_2, nO_2, nS_2, nSO_2, nNO, nH_2O]$$

The composition vector is used in a gas medium description which is part of the flow terminals, see Listing 5.2. It is important to include composition information in the terminals. Gas composition from the burner changes with time, and this affects very much the thermal characteristics in the steam generator. The content of SO_2 is a main factor determining radiation. Also, later stages of the process contain reactions which alter the composition. Remaining S_2 is burnt in an after-burner and in later catalytic stages SO_2 is converted to SO_3 for

```
GasMediumTC ISA RecordTerminal WITH
  q ISA PhaseTC WITH value := 'Gas; END;
  T ISA K2TerminalLib::TemperatureTC;
  x ISA K2TerminalLib::GasCompositionTC;
  % Gas cont. N2,CO2,O2,S2,SO2,NO and H2O
END;
```

Listing 5.2 Gas medium description with composition used in flow terminals.

the production of sulphuric acid. However, these stages were not modelled in this project, so there is no SO_3 component in the composition vector.

Radiative heat transfer

In the original application of the **K2** database, described in Paper C, flue gas temperature stays below 500°C. This means that heat transfer primarily is due to convection, radiation effects can be neglected since they give less than 1% of the total heat transfer.

In the Kemira project it was quite the opposite. Flue gas temperatures well over 1000°C give a lot of thermal radiation and since the steam generator has been designed accordingly the contribution from convective heat transfer is negligible. Holman (1992) gives an expression for the total heat transferred due to radiation derived from the Stefan-Boltzmann law of radiation

$$\frac{Q}{A} = \varepsilon_g \sigma T_g^4 - \alpha_g \sigma T_w^4 \tag{5.1}$$

where the emittance, ε_g, and absorbance, α_g, of the gas depends on temperatures, total pressure and partial pressure of SO_2. This expression was used in the model of radiative heat transfer seen in the lower left plot in Figure 5.2. The thermal capacitance of the metal wall is represented by first-order dynamics in the wall model and the heat transfer between the wall and the steam-water mixture in the risers is modelled by a convective heat transfer module taken from the **K2** libraries.

5.4 Condensing Steam in Plate Heat Exchangers

In a third project the **K2** libraries have been used to describe an experimental setup with a plate heat exchanger, PHE. The setup is used to test temperature control in condensing steam applications and was kindly made available by Alfa Laval Thermal. The modelled parts of the setup are shown in Figure 5.3. The temperature on the secondary side of the PHE is controlled by changing the flow of steam into the primary side of the PHE. At the outlet of the primary side a steam trap is fitted, which allows only condensate and no steam to flow out of the system. Some parts of the experimental setup have not been modelled since they were considered unimportant or not used in the experiments. Unmodelled parts include a second heat exchanger for sub-cooling of condensate and a water pump that can be used for circulation of the secondary side water flow.

The goal of this Master's project was to obtain reasonable parameters for the PID controller used in the temperature control loop in a simplified way. The Master's thesis showed that experiments on the model could be used instead of experiments on the real plant to obtain controller parameters, although the model did not fit the measurements perfectly.

The application in this project was very well suited for the **K2** libraries. No alterations to the basic terminals or subunits were neces-

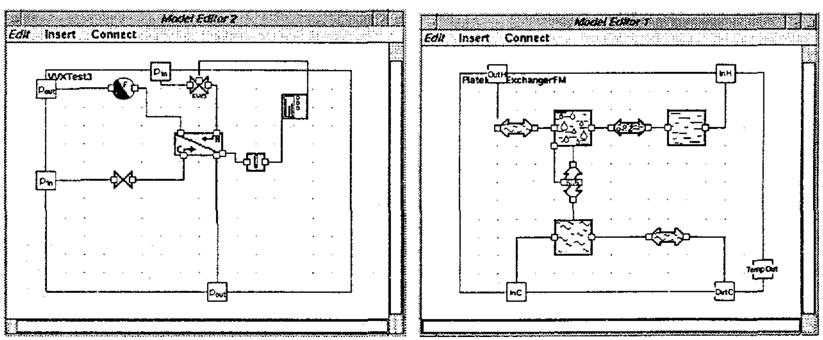


Figure 5.3. Models of the condensing steam setup and the plate heat exchanger.

sary. However, a few new units were created since condensing steam had not been modelled in this much detail before. New unit models were

Steam valve with logarithmic characteristic. Instead of the usual linear characteristic between valve opening and opening area the expression

$$A = A_{min}e^{y \ln R}$$

was used, where y is the valve opening and R is the valve rangeability. This is a straightforward addition to the general steam valve model and was used to create models of differently sized valves.

Plate heat exchanger with detailed description of steam dynamics and condensate level, further described below.

Steam trap, which was modelled as a two-phase compartment, only letting condensate water out when more than half-full.

Plate Heat Exchanger

Since the main task of this project was to model a plate heat exchanger where steam is condensing on the primary side a more detailed description of the primary side compared to the one used in the more general heat exchanger models found in **K2** was needed. The model of the PHE seen to the right in Figure 5.3 has been split up in two volumes on the primary side. The steam volume closest to the InH terminal represents super-heated steam at the PHE entrance. This volume is not active in the heat transfer since the energy contribution is negligible compared to the condensation energy. Instead of the super-heated steam temperature the condensation temperature at the current pressure is used as inflowing temperature in the thermal calculations. A flash volume represents the mixture of condensate and steam. The volume was complemented with calculations of condensate level since the heat transfer characteristics are drastically different above and below this level.

Heat transfer characteristics are given by the equations in the heat flow module between the flash and the water volume. The heat resistance through the metal wall and on the water side is calculated in the same way as in the **K2** libraries although parameter values specific for this PHE were obtained from Alfa Laval Thermal. For heat resistance on the condensating side an expression also given by Alfa Laval Thermal was used

$$\alpha = 1.06 \left[\frac{\lambda^3 \rho^2 g \gamma'}{\mu H \Delta T}\right]^{1/4} \left(\frac{H}{D_H}\right)^{0.134}$$

where the heat transfer coefficient, α, depends on thermal properties of the condensate, $\{\lambda, \rho, \mu\}$, latent heat, γ', temperature difference between steam and plate, ΔT, and plate dimensions, H and D_H.

With this description of the heat transfer the thermal charactersistics of the heat exchanger are captured very well and give results close to measured values.

Pressure characteristics are very complex for the primary side of the heat exchanger. For one-phase flow, like the water flow on the secondary side of the PHE, it is rather easy to measure and get a correlation between pressure drop and flow rate. This is standard procedure for PHEs and Alfa Laval Thermal provided us with such correlations. In the case of condensating steam it is much more complicated since the flow shows three very different behaviours depending on the heat load. Depending on how much of the steam that condensates you get either one-phase steam or condensate flow or a two-phase partially mixed flow which is very difficult to describe. It is not possible to get any simple correlation from measurements and theoretical quantitative descriptions are hard to find. In most literature there are qualitative descriptions of different flow regimes for two-phase flow but these are of no use for actual calculations. In **K2** only simple one-phase approximations are used for two-phase flow. This is a drawback since it resulted in simulated pressures on the primary side that were much higher than what was obtained from measurements. It was not of great importance in this project since the thermal characteristics were the main concern, but needs to be dealt with for further use of the plate heat exchanger model.

5.5 Conclusions

Applications are always interesting. They are the final test whether results are useful or not. The broad range of applications for which the **K2** libraries has been used shows that it is a good tool that supports model development and helps inexperienced users (Master's students) with dynamic modelling of complex systems.

It is practically impossible to have a model database that covers all possible variations of process systems, but this has not been the intention. In every application above the "toy example" level you encounter unique or special equipment that can not be found in general process model libraries. Thus it is much more important to have an open model library, that has models that can be *adapted* to the specific needs of an application. The examples show that the **K2** libraries can be adapted to suit other process applications than thermal power plants. This is different from commercial simulation packages which often have extensive model libraries, but you can not get inside the models and often have great difficulties creating your own.

Another important feature of model libraries is numeric robustness. The models should be built in a way that helps the user avoid numerical difficulties. This is a weak point in **K2** that has not been properly taken care of. It is necessary to build models that can handle extreme cases like empty tanks, reversing flow etc., even though these cases never will occur in real operation. When you first start experimenting with a model the best thing to hope for is poor guesses for all initial values. This causes large transients in the simulation that provokes extreme behaviours. If you have a model that is robust with respect to these extremes it is easier to find a feasible initial point to start further experiments from. For example in the Kemira project the riser tubes in the steam generator (the flash volume to the right in `BrandNewPan` in Figure 5.2) was first modelled as three separate volumes. This had to be simpified to just one volume since the fast pressure dynamics in the volume models caused reversing flows that broke down the simulation.

Chapter 6

Conclusions

Modelling and simulation of dynamical systems is a difficult area for several reasons. It is not taught very well to engineering students, since it is an area of expertise which typically is scattered in the engineering faculty. Modelling and simulation also require skills in many different subjects:

1. good knowledge of the particular domain
2. programming skills, depending on the type of tool used
3. numerical analysis, to correctly interpret results

Model libraries can provide support that helps inexperienced users attack complex problems. The support lies mainly within the two first items above, but also to some extent in the third since a correct model library can help in avoiding numerical difficulties.

Modelling languages and simulation software tools on the other hand provide support for the two last items. The structure and syntax of the modelling language are important for ease-of-use and the quality of models in a model library.

The main conclusion of this thesis is that modelling of complex systems is supported by well-structured model libraries. In Chapter 1 the contributions of the thesis were listed. Here we give a summary of features of model libraries that are important.

Terminals and interfaces should be well-defined. A good set of terminal classes increases the *reusability* of unit models in a library. It can also increase the *adaptability* of a model library to similar applications, as is demonstrated in Chapter 5.

Structure is very important. A well-defined structure gives good overview of the models in the library and supports organization.

Composition and decomposition are important properties. The model library should be such that more complex systems can be composed from simpler elements. It should also be easy to decompose large sections into smaller ones.

Parameterization of models in a library is very important for the *usability* of the library. Parameter equations should be supported by the modelling tool to simplify parameterization of models. Examples of this are given in Chapter 4.

These important features impose specific requirements on the modelling tools used. For example composition and decomposition are greatly simplified by using constraint based modelling. It is a distinct advantage to build model libraries using balance and constraint equations directly instead of computational assignments. Then there is no need for several versions of a model differing only in computational causality.

Another requirement mentioned above is the ability to use parameter equations to specify relations among parameters in a model interface. One step further from this is to allow for *class parameterization*, i.e., to have the possibility to change the super class of a component via a class parameter. Then different media in for example a valve model could be specified on the instance level instead of having several valve models with different medium models. This possibility is one of the important language constructs that have been incorporated in the new modelling language Modelica™, see Mattsson *et al.* (1997).

6.1 Future Work

Model Validation

Model libraries must be validated against experimental data to be of industrial value. Unfortunately there are no systematic approaches to

Modelica is a trademark of the *Modelica Design Group*.

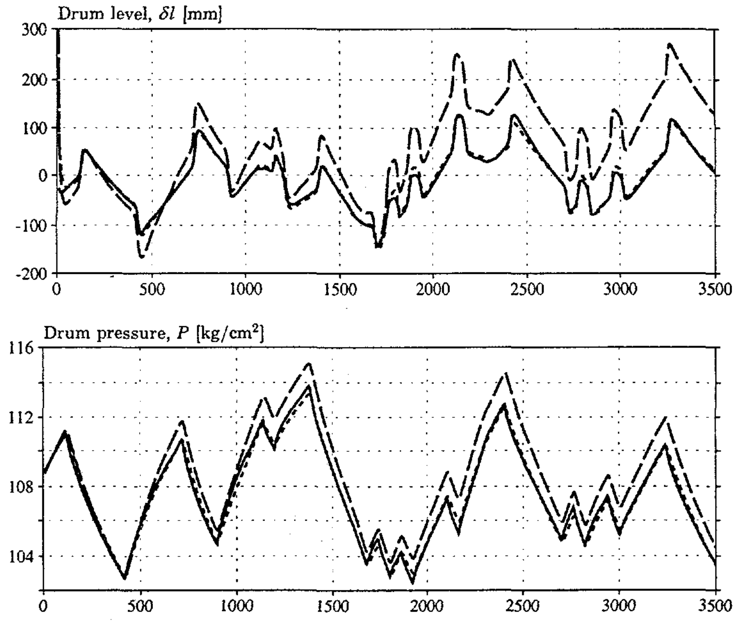


Figure 6.1. Comparison of experimental data(—) and simulations of model with nominal(– –) and estimated parameters(· · ·).

validation of physical models. The idea of using nonlinear parameter optimization for validation is described in the paper in Appendix D. Additional work was done after submission of this paper, which resulted in the report Sørlie and Eborn (1997). With only minor changes in the model structure the almost perfect fit shown in Figure 6.1 was obtained. It is interesting to note that there are only small differences between the nominal and the optimized parameter values in this case. The main improvement in Figure 6.1 comes from optimization of the integrator gains which reduce the drift in the simulated output. This shows that the model structure used almost perfectly describes the system and also that the model is useful for control design since it captures all the fast dynamics even with nominal parameter values.

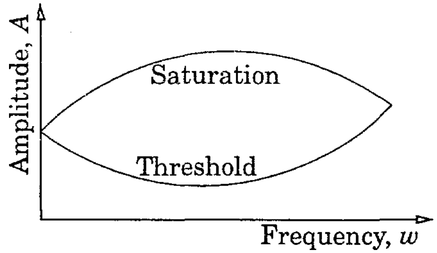


Figure 6.2. The linearity lemon, adopted from p.199 in Gille *et al.* (1959), describes the region of validity for linear transfer functions. The region is restricted from above by saturation phenomena, from below by threshold nonlinearities and at high frequencies by neglected dynamics.

The work described in Appendix D needs to be taken further to arrive at a systematic model validation approach. New experimental data from Värnamoverket could be used to validate the **K2** libraries and the application model built.

It is also important to have the possibility to check the validity of a model during simulation, to see that the results are reliable and that no part of the model has been used in a region where it is not valid. In the modelling language Modelica special constructs for model checking have been added. Validity checking can be added to a model both by using validity ranges for variables (*min, max* attributes) and also by stating *assert* equations, i. e., equations that are not used for the simulation but checked to produce warnings if they become invalid during the simulation.

Uncertainty descriptions

Descriptions of parametric uncertainty is an area in linear systems theory that has grown rapidly during the last fifteen years, although the concept of uncertainty has existed much longer than that. For example Gille *et al.* (1959) proposed to capture the region of validity of a linear model with the notion of the *linearity lemon*, see Figure 6.2. This lemon shape could just as well be used to illustrate uncertainty.

The key idea of robust control theory is to capture uncertainty as additional blocks, see Figure 6.3. Uncertainty descriptions are often

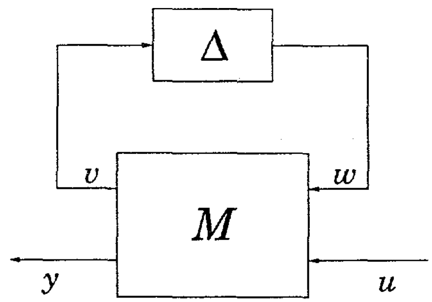


Figure 6.3. Uncertain system on diagonal perturbation form, taken from Lantz (1997)

derived from comparing a more detailed, possibly nonlinear, system model to a simpler one that you want to work with in control design. To have tools that could extract a simple model for design from a complex physical model and also supply bounds for parametric uncertainty would be very powerful. In OMSIM a linearized model can be automatically derived from a nonlinear OMOLA model. In a recent Master's thesis, Lantz (1997) used this feature to export a linear model of a power network with parametric uncertainty on a form that can be used in the Matlab® μ-toolbox.

Another interesting possibility to get a bound on the amount of uncertainty in a model could be to obtain it from the model validation procedure. In the parameter optimization described in Appendix D output error models are included. Through optimization the error models give a measure of the output error variance and thus shows how closely the model describes reality. More structured uncertainty descriptions could possibly be obtained by using other error models in the optimization.

Model Analysis and Simplification

The focus in modelling has for a long time been on simulation. Today this is changing towards analysis and design. It would be useful to have tools that could extract simplified models from complex models, e. g., static relations, linearized models or low-order models with descriptions of parametric uncertainty. While doing manual model simplification you use knowledge that certain dynamics are on a different time-scale and thus can be neglected, or that variables and terms in equations are of a smaller order of magnitude and can be removed from the model. Current work at the department explores how this kind of information can be extracted automatically from a model. For this kind of model analysis it is useful to have both symbolic and numeric tools. Symbolic manipulation can give information on the structure of a problem. Analysis of the incidence matrix of an equation system shows where there are strong or weak couplings between different parts of the system, sometimes this information is not enough and then numeric values can be used.

A General Thermo-hydraulic Library

The **K2** model database was developed with a specific application in mind, the heat recovery steam generator at Värnamoverket. It is thus limited in scope to thermal waste-heat plants. Nevertheless, it has been used successfully in several other applications.

The definition of the object-oriented modelling language Modelica™ offers new possibilities for building general model libraries. One possibility, *class parameterization*, was mentioned in the previous section. Another concept, *generic structures*, makes it easy to construct repeated structures, see Mattsson (1997). Another advantage with Modelica is that it is supported by many research groups and companies and aims at becoming a de facto standard for model exchange. Thus libraries written in Modelica will be portable to different simulation platforms.

We are now in the process of defining and building a new general thermo-hydraulic Modelica base library, which will cover all the basic libraries in **K2** and also include parts that where neglected or not included in **K2**, e. g., flow momentum dynamics, bidirectional and two-phase flows. The work is planned to be finished this summer at the same time as the first commercial simulation tools for Modelica are released.

Chapter 7

References

ABELSON, H. and G. J. SUSSMAN (1985): *Structure and Interpretation of Computer Programs.* MIT Press, Cambridge, MA.

ANDERSSON, M. (1994): *Object-Oriented Modeling and Simulation of Hybrid Systems.* PhD thesis ISRN LUTFD2/TFRT--1043--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

BIRD, R., W. STEWART, and E. LIGHTFOOT (1960): *Transport Phenomena.* Wiley & Sons, New York.

CELLIER, F. (1991): *Continuous System Modeling.* Springer-Verlag, New York, NY.

COULSON, J. and J. RICHARDSON (1977): *Chemical Engineering, vol. 1.* Pergamon Press, Oxford.

EBORN, J. (1994): "Modelling and simulation of an industrial control loop with friction." Master thesis ISRN LUTFD2/TFRT--5501--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

EBORN, J. and B. NILSSON (1996): "Simulation of a thermal power plant using an object-oriented model database." In *IFAC'96, Preprints 13th World Congress of IFAC,* vol. O, pp. 121–126. San Francisco, California.

EBORN, J. and H. OLSSON (1995): "Modelling and simulation of an industrial control loop with friction." In *Proceedings of the 4th IEEE Conference on Control Applications,* pp. 316–322. Albany, New York.

EBORN, J. and J. SØRLIE (1997): "Parameter optimization of a non-linear boiler model." In SYDOW, Ed., *15th IMACS World Congress*, vol. 5, pp. 725–730. W & T Verlag, Berlin, Germany.

ELMQVIST, H. (1972): "SIMNON – Ett interaktivt simuleringsprogram för olinjära system," (An interactive simulation program for non-linear systems). Technical Report TFRT-5113. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

GILLE, J.-C., P. DECAULNE, and M. PELEGRIN (1959): *Feedback control systems*. McGraw-Hill, New York, NY.

HOLMAN, J. (1992): *Heat Transfer*. McGraw-Hill, New York, NY.

HOLMBERG, U. (1992): "Omola models for heat recovery steam generators." Report ISRN LUTFD2/TFRT--7485--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

HYLLSETH, M. (1991): *Dynamic Simulation of Heat Recovery Boilers by Using the Sequential Modular Technique*. PhD thesis NTH-ITE 1991:29, Norwegian Institute of Technology, Trondheim, Norway.

KEHLHOFER, R. (1978): "Calculation for part-load operation of combined gas/steam turbine plants." *Brown Boveri Review*, **65:10**, pp. 672–679.

KLEVHAG, J. (1996): "Accuracy verification of continuous juice blending process using simulation." Master thesis ISRN LUTFD2/TFRT--5554--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

LANTZ, M. (1997): "Robustness analysis using Omola with application to power networks." Master thesis ISRN LUTFD2/TFRT--5585--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

LÖFGREN, O. and P. SVENSSON (1997): "Modelling and control of a plate heat exchanger in steam applications." Master thesis ISRN LUTFD2/TFRT--5584--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

MATTSSON, S. E. (1995): "Simulation of object-oriented continuous time models." *Mathematics and Computers in Simulation*, **39**, pp. 513–518.

MATTSSON, S. E. (1997): "On modeling of heat exchangers in Modelica." In HAHN AND LEHMANN, Eds., *Proceedings of the 1997 European Simulation Symposium (ESS'97)*, pp. 127–133. SCS, The Society for Computer Simulation International, Passau, Germany.

MATTSSON, S. E., H. ELMQVIST, and J. F. BROENINK (1997): "Modelica: An international effort to design the next generation modelling language." *Journal A, Benelux Quarterly Journal on Automatic Control*, **38:3**, pp. 16–19. Special issue on Computer Aided Control System Design, CACSD.

NILSSON, B. (1993): *Object-Oriented Modeling of Chemical Processes.* PhD thesis ISRN LUTFD2/TFRT--1041--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

NILSSON, B. and J. EBORN (1994): "K2 model database—Tutorial and reference manual." Report ISRN LUTFD2/TFRT--7528--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

NILSSON, B. and J. EBORN (1995): "An object-oriented model database for thermal power plants." In BREITENECKER AND HUSINSKY, Eds., *Eurosim '95 Simulation Congress*, pp. 747–752. Elsevier.

NILSSON, B. and J. EBORN (1998): "Object-oriented modelling of thermal power plants." *Mathematical Modelling of Systems*, **4**. To appear.

ORDYS, A., R. KATEBI, M. JOHNSON, and M. GRIMBLE (1994): *Modelling and Simulation of Power Generation Plants.* Springer-Verlag, London, Great-Britain.

PAYNTER, H. (1961): *Analysis and Design of Engineering Systems.* MIT Press, Cambridge, MA.

SØRLIE, J. (1997): "On an interface between OmSim and IdKit." Report ISRN LUTFD2/TFRT--7562--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

SØRLIE, J. and J. EBORN (1997): "A grey-box identification case study: The åström–Bell drum-boiler model." Technical Report ISRN LUTFD2/TFRT--7563--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

STOJNIC, P. (1997): "Modeling of steam generation in a sulphuric acid plant." Technical Report ISRN LUTFD2/TFRT--5577--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

STRÖMBERG, J.-E. (1994): *A mode switching modelling philosophy.* PhD thesis 353, Dept. of Electrical Engineering, Linköping University, Linköping, Sweden.

TUSZYŃSKI, J. (1995): "Modellutveckling för energibibliotek till LTHs simuleringsprogram (OmSim)." Technical Report EV-9411m0471. Sydkraft Konsult AB, Malmö, Sweden.

Paper A

Modelling and Simulation of an Industrial Control Loop with Friction

Jonas Eborn and Henrik Olsson

Abstract

The paper describes the modelling and simulation of a loop for controlling the concentration of fibres in pulp. Real data have been recorded for such a loop. The data show fluctuations in the fibre concentration measurements. The objective of the paper is to see if these fluctuations can be captured and reproduced by a nonlinear model of the loop. The model will include a description of friction in the control valve. Different friction models are investigated in order to determine what friction characteristics if any that are important for the recorded behaviour. For the modelling we have used OMOLA, an object-oriented modelling language, and for the simulations OMSIM, a simulating tool developed for use with OMOLA.

Keywords: Modeling, simulation, friction, process control

A.1 Introduction

To be able to produce high quality paper there are a number of key points in the process that needs to be controlled well. One of these is

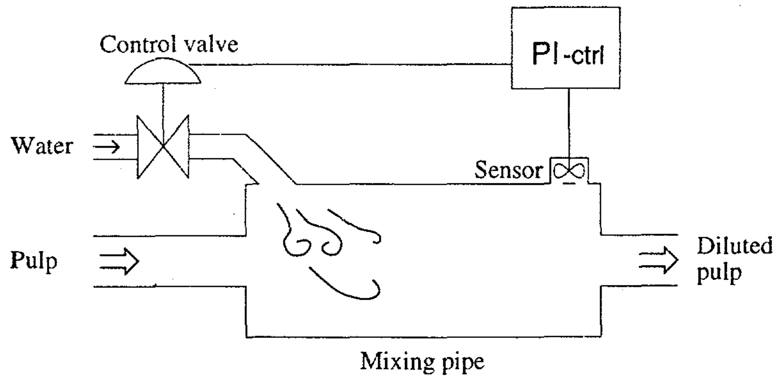


Figure A.1. A schematic diagram of the setup for pulp fibre concentration control.

the first step in the process where the pulp received from the pulp mill is fed into the paper process.

The concentration of fibres in pulp is controlled by diluting the pulp with water. A fairly simple setup used in many paper mills is shown in Figure A.1. Pulp flowing through a pipe is diluted with water at the beginning of an enlargement of the pipe. The fibre concentration is measured at the downstream end of the enlargement. The controller in the loop is a pneumatic PI-controller, which supplies the reference value for the control valve. The control valve consists of a pneumatic positioner, a pneumatic actuator and a valve. Figure A.2 shows recorded data from such a control loop. The output signal, y, is normalized to take values between zero and one and so is the control signal, v.

In this particular control loop there is a problem with concentration fluctuations. The fluctuations appear after some period of operation and increase slowly in amplitude until the problem has to be dealt with. Maintenance has to be done while the process is stopped and can therefore be costly. In a survey of Canadian pulp and paper industry it was found that one third of the control valves suffer from problems with friction and backlash, see Bialkowski (1993). In this paper we examine if the observed fluctuations can be reproduced by a nonlinear model of the control loop including friction in the control valve. We will also seek a remedy to the problem.

A.2 Object-oriented modelling in Omola

This section will give you a brief explanation of the modelling language OMOLA, see Andersson (1994). OMOLA contains object-oriented features such as modularization, abstraction, inheritance and specialization to support model development as well as reuse of models. What is especially appealing with OMOLA and object-oriented modelling in general is the intuitive approach. Components are defined, consisting of an interior describing the behaviour and terminals describing the interface between components. The components are then assembled to complex models. By inheritance, properties common to a group of related classes can be collected in a superclass. Doing so, three things are achieved; the reusability is increased when more abstract classes are used; maintenance is simplified since code does not have to be repeated and by arranging the model library in a tree structure it becomes more legible and easier to use.

A model can be a complex model, which is composed of sub-models,

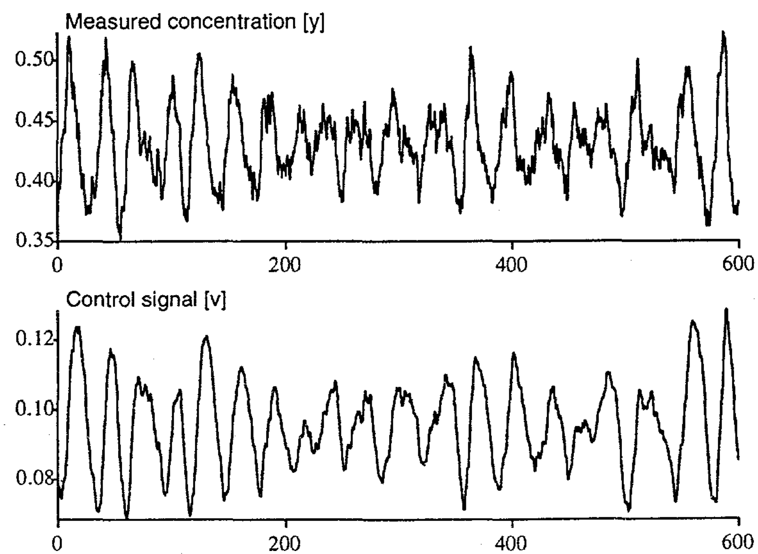


Figure A.2. Concentration fluctuations during operation.

or a basic component, which is so simple that no further decomposition is possible or necessary. The behaviour of a component is described by algebraic and differential equations. OMOLA also supports discrete-event handling and difference equations. The use of discrete events is in fact crucial for efficient simulations using some of the friction models in this paper.

A.3 The Control Loop

This section gives a brief description of the parts of the control loop and some examples as to how they have been modelled.

The mixing pipe is described in a simplified way as instantaneous mixing at the diluting point together with a transport delay. For perfect mixing, the steady-state equation of the fibre concentration is simple, $C_{out} = C_{in}Q_{in}/(Q_{in} + Q_{water})$, where C is the fibre concentration and Q is the volume flow rate of pulp and water respectively. In the model developed, C_{in} and Q_{in} are considered constant. The dynamics of the mixing pipe is attributed to the concentration sensor, placed in a small chamber in the side wall of the pipe. The sensor has been modelled as a first-order lag with a time delay, which includes the transport delay of the pulp flow through the mixing pipe.

The PI-controller is of an older pneumatic type and the only information available is the controller parameters. The controller is modelled as a standard parallel PI-controller where the control signal is limited to values between 0 and 1. This signal is then transformed into a pressure signal in the range 20-100kPa. Between the controller and the positioner is a pneumatic transmission line over 20 meters long. It introduces a lag between the signal pressures at the controller and at the positioner. To describe this lag the flow equation of the transmission line can be solved, see Andersen (1967). The assumptions are; laminar, one-dimensional flow; the polytropic gas equation is used for heat transfer; and the pressure function along the line is assumed to have an exponential time dependence. The solution can then be expressed in hyperbolic functions of the complex variable s. At low frequencies a truncated power series expansions can be used. This renders a second-order oscillating transfer function relating input and output pressure with the natural frequency and damping

$$\omega_n = \frac{12\beta}{\sqrt{72 + 6E_2^2}} \qquad \zeta_n = \frac{E_2}{4\beta}\omega_n$$

The terminal volume has here been neglected compared to the line volume. The parameters are

$$\beta = \frac{1}{L}\sqrt{\frac{nRT}{M}} \qquad E_2 = 8\pi\frac{\eta\beta L^2}{p_0 n A}$$

where L is line length, η is the viscosity of air at temperature T, n is the polytropic exponent, M is the molar mass and R is the universal gas constant, p_0 is the mean pressure and A is cross-section area. This transfer function has also been implemented in the model of the controller.

The control valve is the most complex unit and the key part in explaining the limit cycle behaviour of the control loop. It consists of the pneumatic positioner and actuator which are mounted on the valve and used as a power servo to control the position of the valve. The important dynamics of the system are those of the positioner and actuator and therefore these parts are described individually.

The positioner is a nonlinear control device controlling the flow of compressed air into and out of the cylinder chambers of the actuator. The flow is a nonlinear function of the position error and since the chamber pressures are roughly the integral of the flow the positioner–actuator will act as an integrating controller. The heart of the construction is the pilot valve with spool in Figure A.3. The spool is moved by a lever, which at equilibrium is kept in the center position by a balance between the control pressure and a feedback force from the position of the actuator and valve. The pilot valve has variable structure since only two of the four possible passages through it will be active at the same time. Each air flow passage is seen as an orifice, a restriction of air flow, with variable area. The opening area of the orifice is a function of spool position. This function is essential to calculate the flow through the orifices. An expression of the true flow area is difficult to derive because of the complex geometry. Instead an approximation has been made interpolating between small and large spool displacements.

The flow equation used at the orifices is the ideal gas equation for mass flow, see Andersen (1967),

$$W = A_e p_1 \left\{ \frac{2\gamma M}{(\gamma-1)RT_1} \left[\left(\frac{p_2}{p_1}\right)^{\frac{2}{\gamma}} - \left(\frac{p_2}{p_1}\right)^{\frac{\gamma+1}{\gamma}} \right] \right\}^{\frac{1}{2}}$$

where A_e is the effective cross-section area and γ is the ratio of specific heats. Under certain assumptions p_1 and p_2 are interpreted as the upstream and downstream pressure respectively and T_1 as upstream absolute temperature. This makes the relation practically useful for flow calculations at restrictions. As it stands it is valid for calculations of sub-sonic flow. When the ratio between p_2 and p_1 falls below a critical pressure ratio the flow becomes sonic. The pressure ratio in the flow equation is then replaced by this constant ratio and the flow will only depend on p_1.

The actuator is double-acting, with pressure on both sides of the piston. A common assumption made to calculate the expansion of air in the cylinder chambers is that it follows the polytropic equation $p\left(\frac{V}{m}\right)^n = constant$. The polytropic exponent n can be chosen between $n = 1$ for isothermal expansion and $n = \gamma = 1.4$ for adiabatic expansion. Since the thermal energy content of air is low compared to that

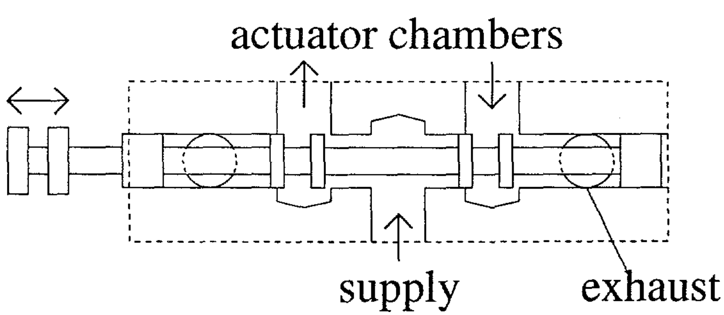


Figure A.3. Positioner pilot valve in slightly open position.

of the cylinder and piston, it is reasonable to assume that the cylinder remains at constant temperature. Only isothermal expansion is used throughout the simulations since charging and discharging is slow.

There is friction present in both the actuator and the valve, but since the actuator and the valve are rigidly coupled, the friction of the valve can be included in the actuator friction model. Only one friction model is used to account for the total friction.

The valve is a ball segment valve. In this application the valve operates very near the closed position. The control signal is 10%±3%. For such small changes the valve characteristic will be close to linear. The control valve has been modelled as a linear valve with the gain parameter set according to steady-state data.

The process parameters are somewhat uncertain. The models that have been developed in this work are based on product information and fundamental mechanics and thermodynamics. The only measured data available is the measurements during normal operation shown in Figure A.2 and an identification experiment made to determine a process model valid from control signal to measured concentration. This experiment gave estimates of the parameters of a first-order process model with dead-time. The gain estimate is approximately 3 while the estimates for time constant and dead-time both are approximately 10 seconds. When all the sources of dynamics in the loop model excluding the concentration sensor are added together, half of the dead-time and a fifth of the time constant of the estimated first-order model can be accounted for. The remaining dynamics are attributed to the concentration sensor.

A.4 Friction Modelling

The friction force is always opposed to the direction of motion. The classical descriptions of friction include Coulomb, static and viscous friction. The friction force is given as a static function of the velocity although the discontinuous static friction needs to be handled in a special way.

In the more detailed behaviour of friction there are a few other phenomena that need to be considered; the *Stribeck effect* is that the drop in friction force when motion begins is smooth and not discontinuous; the *Dahl effect* is the spring-like behaviour that occur during sticking and the term *frictional lag* is sometimes used for the hysteresis in friction force seen when moving in one direction at varying velocities. Furthermore friction shows a random behaviour due to the irregularities of the surfaces in contact.

This work will concentrate on four models of friction with slightly different features; a classical friction model, an extended Dahl model as described in Canudas de Wit *et al.* (1995), the Bristle model from Haessig and Friedland (1990), and the Seven parameter model from Armstrong-Hélouvry (1991). The classical model is a static model. The other three models are all dynamic and contain, to various extents, the phenomena described previously.

The classical model has been implemented to allow for wellbehaved simulations. During motion the friction force is described as $F = F_c \operatorname{sign}(v) + F_v v$, where F_c is the Coulomb friction and F_v is the viscous friction constant. When stuck F equals the driving force, i.e.,, the sum of all other forces. Switching between the two modes Stuck/Slip is accomplished by discrete events. Sticking begins when the velocity falls below a certain zero-velocity tolerance. Slipping begins when the driving force becomes greater than F_s, the maximum static friction.

The extended Dahl model describes friction force by means of a differential equation

$$\begin{cases} \dot{z} = v[1 - \operatorname{sign}(v) z / g(v)] \\ F = K_c z + d\dot{z} + F_v v \end{cases}$$

where K_c is a stiffness parameter, d is a damping coefficient and $g(v)$ determines the steady-state relationship between velocity and friction force. If $g(v) = F_c$ and $d = F_v = 0$ then this model reduces to the original Dahl model, see Dahl (1977). The model is interesting because it is simple and dynamic, there is no need for discrete events or mode switching. The model also describes all the important frictional phenomena: spring-like behaviour, arbitrary steady-state characteristics, frictional lag and varying break-away force.

The Bristle model is designed to emulate the randomness of friction. The frictional force is summed from spring action of minute bristles, or bonds, which break at a certain deflection and then become reestablished. Static friction is attained through a number of extra bristles which are active near zero velocity. The model is laborious to use in simulations due to the discontinuous behaviour and as a result of this, the large number of discrete events needed.

The Seven parameter model consists of two models. During sticking it models friction as a very stiff spring $F = K_c x$, where x is the displacement. During sliding, the force is given by Coulomb, viscous and Stribeck friction with frictional lag

$$F = \text{sign}(v)\left(F_c + \frac{F_{s,act}(t_2) - F_c}{1 + (\frac{v(t-\tau_L)}{v_S})^2}\right) + F_v v$$

where τ_L is the time delay of the frictional lag and v_S is the Stribeck velocity. The actual static friction, $F_{s,act}$, depends on the duration of the last sticking period, t_2. A major drawback of this model is that it is not specified when switching between the two modes should be done and how x should be chosen when sticking begins.

A.5 Simulations

In order to study the effects of friction on the performance of the control loop simulations have been made in the OMSIM simulation environment, see Andersson (1994).

The integration method used is specified by the mathematical formulation of the model. The model is written in the form of differential algebraic equations, DAEs, with differential equations describing the dynamics of the system and algebraic equations for constraints. Of the implicit solvers supplied by OMSIM one, DASRT, has been chosen. It is equipped with a root-finding algorithm necessary for simulations of models with state-dependent events.

The initial values of the state variables of the model must fulfill the constraints given by the algebraic equations when the simulation is started. The model of the control loop has up to 22 state variables depending on the friction model used, e. g.,, the states of the actuator chambers, control pressure, positions, and velocities of the positioner–actuator and the integral value of the controller. To make it easy to calculate the state variables the simulations are started close to a stationary point. The operating point of the system is defined by the mean values of the signals in Figure A.2, $\overline{y} = 0.43$ and $\overline{v} = 0.1$. From these many of the initial values can be calculated. To perturb the system slightly the initial value of the measured output is set to $y = 0.4$.

The parameter values of the friction models cannot be determined without detailed experimental studies. This has not been possible. Reasonable values have been chosen to be used in the simulations. Friction in a pneumatic actuator amounts to about 20% of the maximum load of the actuator, which is 5-6 kN. Therefore the Coulomb friction parameter is set to 1000 N. Static friction is chosen to be 50% higher than Coulomb friction. The other nominal parameters are chosen as F_v=0, $K_c = 10^8\,\mathrm{N/m}$ and $d = 9000\,\mathrm{Ns/m}$ according to recommendations in the references, mainly from Armstrong-Hélouvry (1991). The Stribeck velocity of the extended Dahl and Seven parameter models is chosen as $v_S = 2 \cdot 10^{-4}\,\mathrm{m/s}$ and τ_L as 0.02 s.

Influence of Different Friction Models

The simulations of the control loop are made using the different friction models and the parameter values above if no other value is specified. The simulations will make it possible to make comparisons of the friction models and study the effects of different frictional phenomena on the performance of the control loop. The behaviour can then be compared with the recorded data in Figure A.2.

The classical friction model is used as a benchmark to compare with the other simulations. The simulation recordings are taken after the initial transient has disappeared. The variables plotted are: the output of the concentration measurement, the valve opening, the control signal from the PI-controller and the friction force. The con-

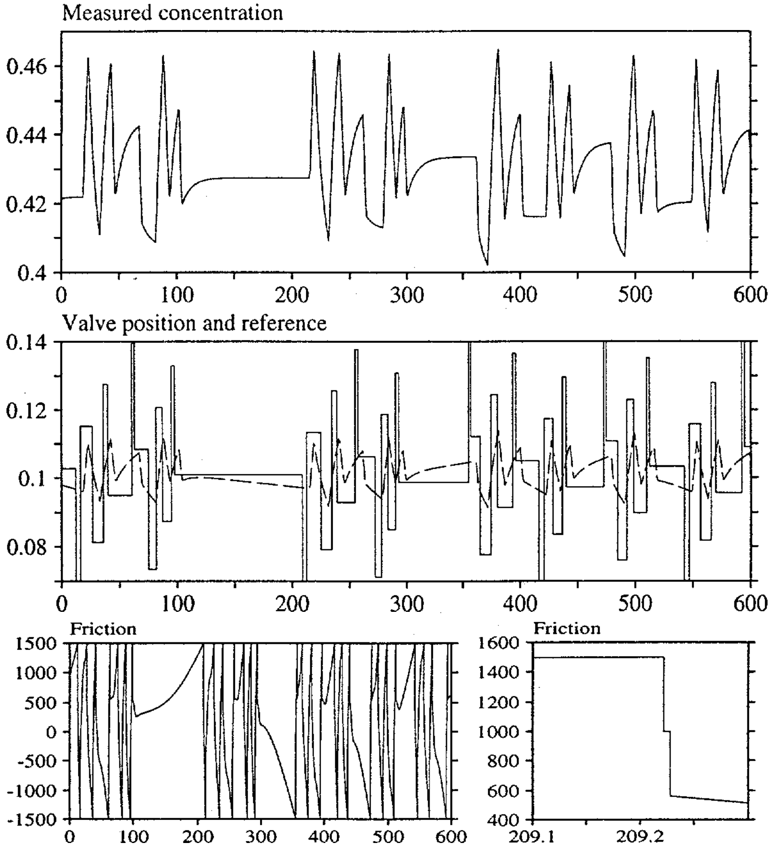


Figure A.4. Bench mark simulation run with the classical friction model. The detail on the right shows the friction force during a slipping period.

trol signal can be seen as a reference to the positioner and is plotted together with the valve position.

The result of the bench mark simulation is shown in Figure A.4. The output looks like a series of step responses and this is indeed the case. On this time-scale the valve moves almost instantly between different positions. The detail of the friction force shows that the slipping period is approximately a hundredth of a second. Since friction force changes

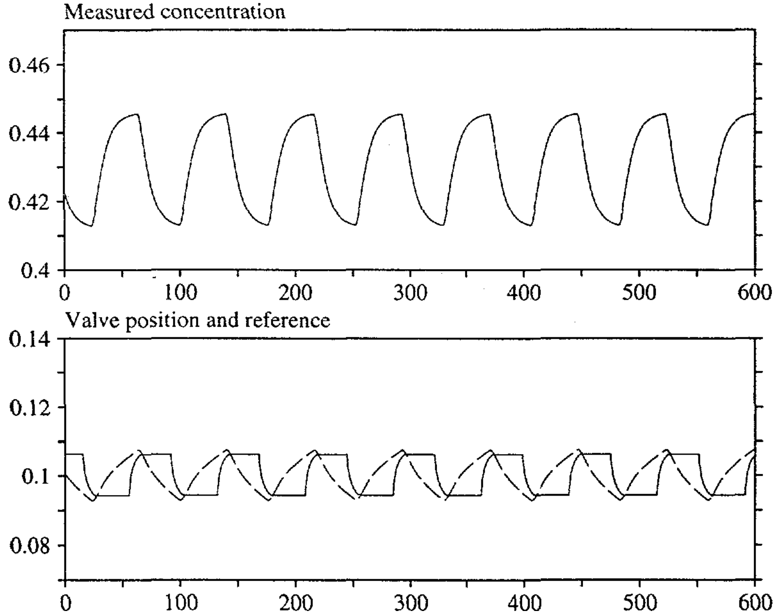


Figure A.5. The classical friction model with $F_c = F_s = 1000\,\mathrm{N}$.

instantly from F_s to F_c the accelerating force on the actuator piston will be very large compared to the inertia. But as soon as the piston begins to move the driving force decreases because of an equalization of the pressures in the actuator chambers. When the driving force becomes smaller than F_c the piston slows down until it sticks. The friction force then becomes equal to the driving force. The behaviour is similar to the one seen in Figure A.2 although the amplitude of the fluctuations is smaller. Another thing noted are the long, calm stretches at $t \approx 100\,\mathrm{s}$ and $t \approx 300\,\mathrm{s}$. When the valve sticks close to the desired position it will remain there for a longer period of time while actuator force builds up. This is a very slow process since the position error, which determines the pilot valve opening in the positioner, is small.

A different behaviour is seen when there is no static friction, $F_s = F_c$, see Figure A.5. The concentration fluctuations are smoother and slower because the accelerating force is smaller. When slipping starts the friction force does not decrease. It remains the same until slipping stops and the valve sticks. It is interesting to note that although $F_s = F_c$ the fluctuations continue. Merely Coulomb friction is enough for limit cycles to appear.

As implemented here the classical friction model is very well-suited for simulation. This requires that the transition from sticking to slipping is handled by events, which makes the simulation efficient.

The extended Dahl model shows a behaviour resembling very much that of the classical friction model, see Figure A.6. The fluctuations are perhaps a little slower due to the damping included through the parameter d and the addition of Stribeck friction. When the difference between valve position and reference is large the valve still jumps quickly to a new position. The difference from the case of no or little Stribeck friction is that the valve does not need to come as close to the desired position for the movement to be smooth. When the Stribeck velocity is very high, $v_S \geq 0.05\,\mathrm{m/s}$, the smoothing effect will cause a behaviour similar to that in Figure A.5. In that case the velocity of the actuator piston never exceeds the Stribeck velocity.

The extended Dahl model with a large spring constant is a stiff problem and can be difficult to handle for the simulator. There are however integration methods for stiff problems available which can be used since the extended Dahl model does not require any event handling.

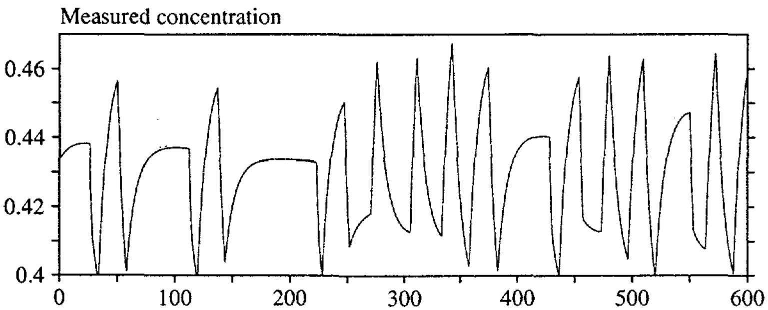


Figure A.6. Simulation of the extended Dahl model.

The Bristle model yields results similar to the ones observed before. It can give simulation results slightly more irregular than the other models but there are no fundamental differences. The simulation was made with a total spring constant $K_c = 10^7\,\text{N/m}$. (The spring constant of each bristle is $10^6\,\text{N/m}$.) Using $K_c = 10^8\,\text{N/m}$ as in the simulations of the other models would be desirable but the simulation time would be more than tenfold.

From the plot of the friction force in Figure A.7 it is not hard to see why this model is laborious to use in simulations. During a slipping period the model will produce several hundred events a few microseconds apart. This will slow down the simulation enormously. There is also a problem when motion stops. The spring-like bristles will generate undamped oscillations in the friction force unless the bristle positions are chosen to make the sum of bristle forces equal the driving force. This makes it necessary to have the driving force, u, as an input to the friction model, as was the case with the classical friction model.

The Seven parameter model gives a surprisingly regular result in the simulation shown in Figure A.8. It looks like the system approaches a stable limit cycle. Besides this the differences from the previous simulations are small.

The model is not at all suited for simulation. The switching between the two modes, sticking and sliding, is not specified in Armstrong-Hélouvry (1991). The model in sticking is an undamped spring. The

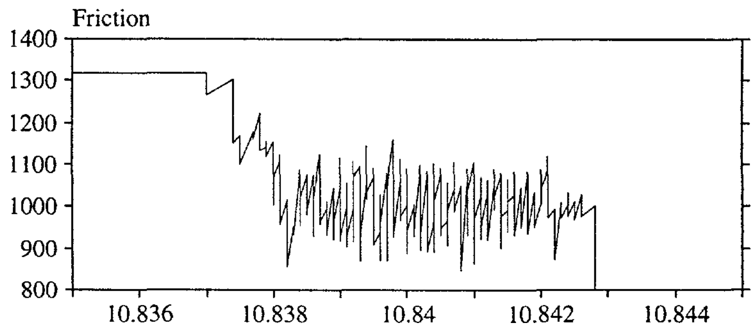


Figure A.7. Friction force during a simulation of the Bristle model.

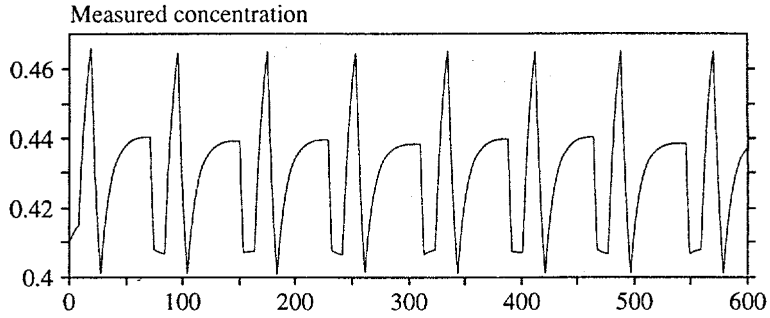


Figure A.8. The Seven parameter model simulation.

displacement must be chosen as $x = u/K_c$ otherwise there will be undamped oscillations in the friction force. Thus this model too requires knowledge of the driving force, u. Also, the use of a time delay, τ_L, makes the system difficult to simulate. In OmSim, the delay function interpolates between values stored for plotting. To be able to catch the velocity peaks when the position changes quickly the plotting interval has to be smaller than $\tau_L/2 = 0.01\,\text{s}$. Since the DAE-solver stops at every plotting interval the simulation will be very inefficient. The simulation in Figure A.8 took four hours to perform on a SUN SPARC-station 2. This is more than fifteen times slower than the classical model.

Influence of Process Gain

As stated in Section A.3 the process gain is about three times. This is a rather uncertain figure since it is taken from an identification experiment on an industrial application. To see how the gain affects the observed concentration fluctuations, simulations were made with the extended Dahl model and 60% higher process gain. The Stribeck velocity was $v_S = 0.005\,\text{m/s}$. In Figure A.9 it can be seen that the amplitude increases more than 100% and the fluctuations are also more irregular and of higher frequency. Except for noise the behaviour is very similar to the observed behaviour of Figure A.2, both in amplitude, frequency as well as in the irregularity of the fluctuations.

The process gain is high mainly because the control valve is much larger than necessary. If a smaller valve which could operate at 50% valve opening was used the gain would decrease with 60-80%. The amplitude of the concentration fluctuations would then be smaller and since the gain in the controller could be increased the positioner–actuator would be less sensitive to friction and backlash. Figure A.10 shows a simulation where the gain has been reduced with 80% and the controller parameters have been altered accordingly. The problem with the valve moving back and forth remains but the lower gain makes the concentration fluctuations a lot smaller. The simulation indicate that if a smaller valve could be used in practice, the problem would be reduced to a minimum. To stop the valve chattering the positioner need to be redesigned with a deadzone in the pilot valve. This would stop the actuating force from increasing when the position error is small. The PI-controller in the loop also has to be changed to allow for small concentration errors without increasing the control signal.

A.6 Conclusions

This paper has described the modelling of a concentration control loop in a paper mill. The major effort was made to model the nonlinear behaviour of the pneumatic positioner–actuator. A library of several friction models has also been developed. The models have been devel-

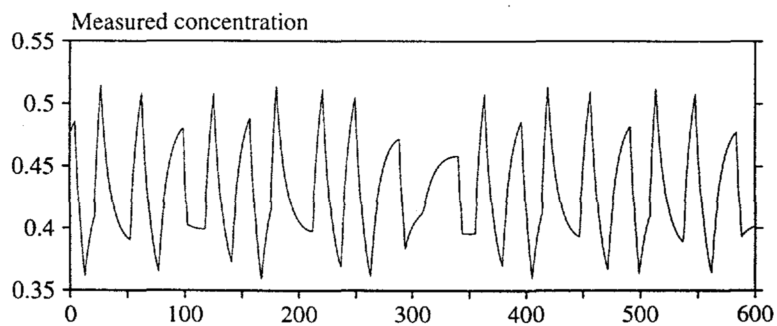


Figure A.9. Slightly higher process gain produces a result very similar to the observed in Figure A.2.

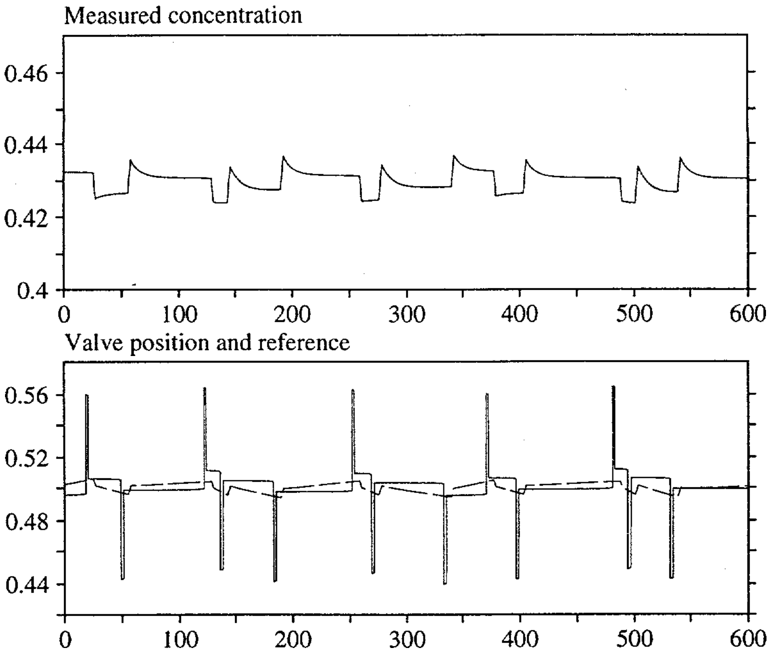


Figure A.10. When a smaller valve is used the fluctuations are significantly reduced but not eliminated.

oped using OMOLA, an object-oriented modelling language. The friction models have been used in combination with the control loop model in simulations to study the effects of friction on the control loop behaviour. The simulations have been made in the OMSIM working environment and show that the observed behaviour of the control loop with large variations in concentration can be reproduced by the model. It is noteworthy that friction alone can cause this behaviour. A common opinion in the industry is that backlash has to be present to create the observed oscillations. Even the use of a simple friction model with only Coulomb friction causes an oscillating behaviour.

The effect of the process gain on the behaviour of the control loop has also been studied. The large gain caused by using an oversized valve at small valve openings is one part of the explanation to the large concentration variations. The simulations indicate that using a smaller valve operating at larger openings would reduce the amplitude of the variations. Combining this with deadzones in the positioner and PI-controller would probably solve the control problem at the cost of a small constant concentration error.

The friction models are to varying extents suited for simulation. The classical friction model in its original formulation is difficult to use. However, an implementation with events is efficient and fast to use in simulations. The requirement to have the driving force as an input to the model besides velocity is an additional drawback, which is shared with the Bristle and the Seven parameter model. The long simulation times required also make the last two unsuited for simulation. The extended Dahl model has the appealing quality of being simple due to the fact that it is dynamic. It also contains all the important frictional phenomena.

References

ANDERSEN, B. W. (1967): *The Analysis and Design of Pneumatic Systems*. John Wiley & Sons.

ANDERSSON, M. (1994): *Object-Oriented Modeling and Simulation of Hybrid Systems*. PhD thesis ISRN LUTFD2/TFRT--1043--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

ARMSTRONG-HÉLOUVRY, B. (1991): *Control of Machines with Friction*. Kluwer Academic Publ., Boston, Ma.

BIALKOWSKI, W. L. (1993): "Dreams versus reality: A view from both sides of the gap." *Pulp and Paper Canada*, **94:11**.

CANUDAS DE WIT, C., H. OLSSON, K. J. ÅSTRÖM, and P. LISCHINSKY (1995): "A new model for control of systems with friction." *IEEE Transactions on Automatic Control*, **40:3**.

DAHL, P. (1977): "Measurement of solid friction parameters of ball bearings." In *Proc. 6th Ann. Symp. on Increm. Motion, Control Syst. and Dev.* Univ. Illinois.

HAESSIG, D. and B. FRIEDLAND (1990): "On the Modeling and Simulation of Friction." In *Proc. of the 1990 American Control Conf., San Diego*, pp. 1256–61.

Paper B

Object-Oriented Modelling of Thermal Power Plants

Bernt Nilsson and Jonas Eborn

Abstract

This paper presents a set of model libraries, called K2, for modelling of thermal power plants. The models are based on first principles and describe mainly the dynamic mass and energy properties of the modelled system. The K2 models are described in the object-oriented modelling language OMOLA and the libraries are organized in an OMOLA model database. The libraries are grouped into three different sets, namely unit libraries, subunit libraries and model component libraries. The unit models are used to build up plant system models, which are application dependent. The units are composed of subunits. The subunits describe different physical phenomena and a set of subunits build up the behaviour of the unit model. Model components are used to facilitate the development of new units and subunits. OMOLA models can be simulated in the OMSIM simulation environment and the K2 model database has been used in a case study of the dynamics in an HRSG plant.

Keywords: Computer aided engineering, modelling, object-oriented modelling, power plant, process models.

B.1 Introduction

An object-oriented approach to modelling and simulation of thermal power plants is presented in this paper. The emphasis of the presentation is upon the model decomposition and the library organization. The set of model libraries are called K2. A Heat Recovery Steam Generation plant, HRSG, is modelled by the use of sub-models from the K2 object-oriented model database, see Nilsson and Eborn (1994; 1995), and it is simulated in a case study, see Eborn and Nilsson (1994). The models in K2 are written in OMOLA, an object-oriented modelling language, see Andersson (1994), Mattsson and Andersson (1992) and Nilsson (1993). OMOLA supports hierarchical structure decomposition of models into sub-models. Behaviour is described by equations and/or events. OMOLA can model combined discrete event and continuous time models, i.e., hybrid models. It also supports inheritance of properties between the model classes. A model subclass inherits the attributes from its super class. OMSIM the OMOLA simulation environment, is implemented in C++ under Unix and X Windows. The K2 model libraries support model development in three different levels, i.e., systems, units and subunits, in order to facilitate multi-user and multi-disciplinary modelling.

Thermal power plants are well suited for object-oriented modelling because of the extensive use of standard components like pumps, valves, heat exchangers, boilers etc. This makes it possible to capture a large number of power plant configurations with a limited number of model classes. The physical units can be described by a set of subunit objects representing different physical phenomena in the unit. Examples are control volumes of medium, here called compartments, and flowing medium, here called flow resistors. The number of subunits is quite small.

One example of a simulation tool commonly used today is MMS (Modular Modeling System) which uses the simulation language ACSL (Advanced Continuous Simulation Language). MMS is marked by Babcock & Wilcox and ACSL is a product from Mitchell & Gauthier. ACSL is a CSSL based simulation language that require system models on state space form. MMS contain modules of commonly used power plant equipment that can be put together in order to model a plant configuration. The Skegton Unit Component Models, described in Ordys *et al.* (1994), is another example of a model library for thermal power plants. These models are described on state-space form for the use in a CSSL type simulator, e.g., MatrixX. A sequential modular approach to simulation of HRSG plants is presented by Hyllseth (1991). The K2 unit models capture the same kind of models as the Skegton Unit Component Models, but they are expressed on equation-oriented form without specifying causality. The OMSIM environment makes a symbolic manipulation of the equation set before the generation of the simulation code. This increases the reusability of model objects and the number of different simulation scenarios that are possible to study.

A number of different approaches to object-oriented modelling are reported in the literature, see the survey Marquardt (1994). One can divide the approaches in two major categories, general modelling languages and process oriented modelling systems. Examples of general modelling languages are OMOLA [Mattsson and Andersson (1992)], DYMOLA [Cellier and Elmqvist (1993)] and ASCEND [Piela *et al.* (1991)]. Examples of process oriented approaches are MODEL.LA [Stephanopoulos *et al.* (1990)], VEDA [Marquardt (1993)] and HPT [Woods (1993)]. Similar ideas are also found in Foss *et al.* (1995) and Wasbø and Foss (1995). The K2 model database is implemented in a general modelling language and adds the spirit of process oriented modelling language. The subunit decomposition, into compartment and flow resistors in K2, have much in common with the the decomposition into devices and connections in VEDA, see Marquardt (1993).

The paper is organized as follows. In Section 2 the organization of the K2 model database is presented together with the major model classes. The mathematical interpretation of the model structures is presented in Section 3 and the HRSG plant application in Section 4. The limitations of K2 is discussed in Section 5 and conclusions are drawn in the last Section 6.

B.2 The K2 Model Database

The models in the K2 model database are based on first principles and describe mainly the dynamic and static behaviour in the water/steam cycle of thermal power plant configurations. A K2 based application is decomposed into four major hierarchical layers; namely application system models, unit models, subunit models and model components. The system descriptions are application oriented and are developed by a user of the K2 model database. In the actual database there are unit libraries, subunit libraries and model component libraries. A detailed presentation of the K2 model database is found in Nilsson and Eborn (1994) and it is also briefly presented in Nilsson and Eborn (1995). The development of process applications in an object-oriented modelling language is also discussed in Nilsson (1994) and Nilsson (1996). The basic idea with this guideline of decomposition is to facilitate abstraction of details for different users. Potential users are categorized into system developers, unit model developers and developers of phenomena descriptions.

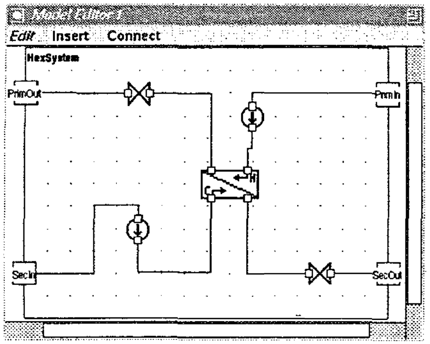
System Modelling using Unit Models

Unit models have corresponding physical representations, like pumps, valves, heat exchangers etc. The unit models are used to create systems and subsystems. An example of a subsystem is a boiler which is composed of a number of units, like a drum, circulation pump, riser, valves. A boiler system is used as a subsystem in a larger configuration of a pan or an entire thermal power plant. The unit models are grouped in unit libraries and are reused by the system developer for graphical development of a plant section or a plant configuration.

A small heat exchanger system is illustrated in Fig. B.1. Two different water flows are pumped through a cocurrent heat exchanger. There are pumps on the inlets of the exchanger and two corresponding valves on the outlets.

Unit Modelling using Subunit Models

Subunit models are descriptions of physical phenomena in control volumes or in transport interfaces. There are two main types of subunits: compartments and flow resistors. The *compartment subunit* models describe mainly the mass and energy balances over a control volume. The compartment subunit interact with other compartments via flow resistors. The *flow resistor subunits* describe the transport of medium or heat. Compartments as well as resistors have an internal medium description, possible media are water, water/steam mixture, steam or flue gas.



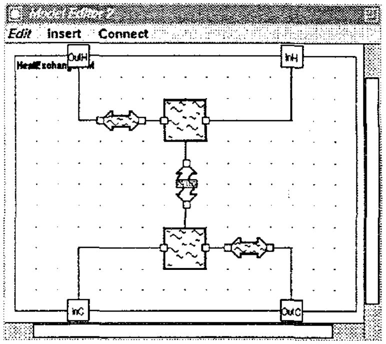


Figure B.1. A system model composed of a set of unit models.

Figure B.2. The internal structure of subunits inside an heat exchanger unit.

One example of a unit model composed of a set of subunits is illustrated in Fig. B.2. The heat exchanger dynamics is captured by the two compartments. Both sides of the exchanger has water as medium. The medium flow is described by flow resistors and the heat interaction between the two compartments is described by the heat flow resistor. The subunit decomposition of compartments and flow resistors model objects is inspired by the decomposition into devices and connections in VeDa, see Marquardt (1993).

Compartment Subunits A compartment subunit model describes unsteady-state mass and energy balances and steady-state momentum balances over a control volume. The mass and energy balances are rewritten into pressure and enthalpy differential equations. The major difference between the different compartments, seen in Fig. B.3, is the

Figure B.3. A set of compartment subunits.

Figure B.4. A set of medium flow resistor subunits.

internal medium model which contains all steam table calculations describing media specific functions and parameters.

Flow Resistor Subunits The flow resistors describe the static relation between flow rate and a driving force. A medium flow resistor describe the relation between medium flow and pressures and a heat flow resistor describe the relation between the heat flow and temperatures. Once again the internal medium model is used to calculate proper physical parameters for different media. Some of the medium flow resistor subunit classes are shown in Fig. B.4. The medium flow resistors contain a model compartment describing the friction factor, a nonlinear function. There are friction factor model components for valve losses, tube losses etc. The heat flow resistor contains model components that describe the heat resistance between the components. In a tubular heat exchanger there is heat resistance outside the tube, in the tube wall and inside the tube.

Model Components

All model classes discussed above are composed of model component attributes which are classes that can be reused. Composite models, like systems and units, are composed of sub-model classes and connections. They also have terminals for the interaction with surrounding models. This means that there is a need for a common terminal library.

Model Interaction The flow terminals, i.e., the interfaces describing system, unit and subunit interaction, are composed of the following sub-terminals; mass flow rate, pressure, enthalpy and one sub-terminal specifying medium. The medium specific sub-terminal is used to make application consistency checks, i.e., a steam flow terminal can not be connected to a water flow terminal. It is important to notice that the terminals always have the same structure facilitating reuse of all model classes in different situations. In some cases this means loss of efficiency. One example is the calculations of medium properties in compartments which could be reused in nearby connected flow resistors. This is not done to avoid an increased number of sub-terminals and loss of generality. The choice of interface variables made here is not unique. Two sub-terminals are chosen to be equal to the state of the medium, pressure and enthalpy. One example of an alternative

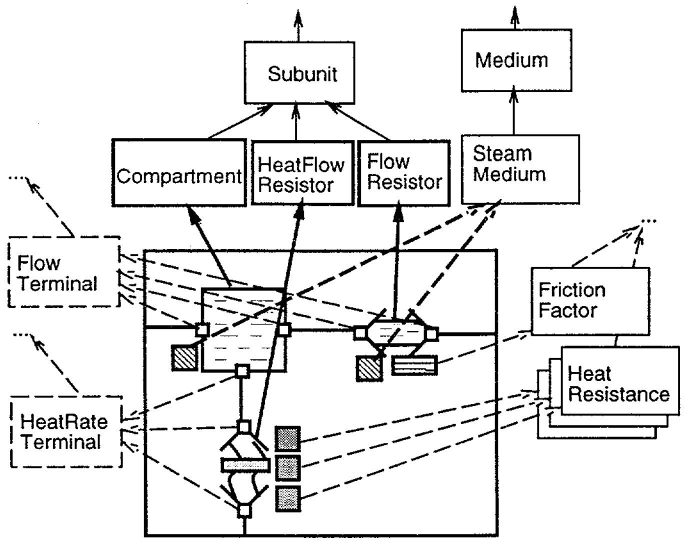


Figure B.5. Reuse of model components on the subunit level.

choice of interface variables is volumetric flow rate, temperature and pressure.

The heat interaction is also described by a common heat rate terminal and it is composed of sub-terminals describing temperatures on the inlet and outlet and of the heat flow rate. The use of different terminals is shown in Fig. B.5.

Model Reuse Primitive models, like subunits, are composed of variables and equations. But some of the equations can be encapsulated in objects. One example is the reuse of the calculation of thermodynamic properties in different media. These expressions use steam tables and media dependent parameters. Other examples are the use of friction factors and heat resistance objects in the flow resistors. The reuse of model components in subunits is seen in Fig. B.5. This means that the development of subunits is supported by a set of libraries for variables, terminals, medium models and functions.

B.3 Interpretation of Subunit Structures

The phenomena oriented subunit models are described by equations. These equations are physical relations derived from fundamental laws. In principle the compartments contain the dynamic states and the change of the states can be calculated if the flow rates in and out are known. The flow resistors contain equations that can calculate the flow rate if the states on each side of the resistor are known.

In OmSim all equations are sorted to generate an efficient simulation code which is used by the numerical routine. A model results in a set of differential-algebraic equations, DAEs, that can be solved by the numerical routines. By the use of symbolic manipulation OmSim tries to generate more efficient simulation code, preferable ordinary differential equations, ODEs, on explicit state-space form. There are numerical solvers for both ODE and DAE problems.

Three different cases of subunit structures can be identified, namely alternating net structure, resistors in series and compartments in series. These structures are illustrated in Fig. B.6 and B.7 and are discussed below.

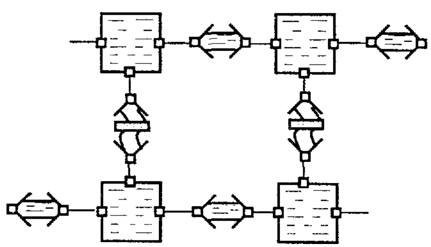


Figure B.6. Case I, alternating net structure.

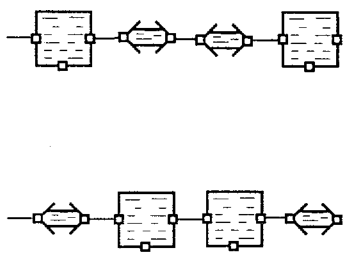


Figure B.7. *top*: Case II, flow resistors in series. *bottom*: Case III, compartments in series.

Case I: alternating net structure Net structures of compartments and flow resistors connected one after the other as in Fig. B.6 can be manipulated and transformed to ODEs on state-space form. In principle the values of the states are used to calculated all flow rates in the net which then are used to calculate the derivatives. It is desirable that the user describe the problem in this way.

Case II: resistor series In practice flow resistors are often connected in series describing flow nets with neglected dynamics, see top Fig. B.7. One example is a series connections of pipes, valves and pumps, all described by flow resistors with different friction factors. The mathematical problem is to find the unknown states between the flow resistors. The flow rates through the resistors are the same and the states must fulfill the two different nonlinear equations. In the general case the algebraic states can not be eliminated symbolically. Therefore structures like this will generate DAE systems.

Case III: compartment series A not so common structure is compartments in series, see bottom Fig. B.7. This can occur if tanks or vessels are connected in series without flow equipment in between. This means that there is no relation between pressure and flow rate. On the other hand the connection generates an equation indicating that the pressures in the compartments are equal. The mathematical problem is to find the unknown pressure and the unknown flow rate using the two pressure state equations. By the use of symbolic manipulation this is done and it is possible because of the choice of interface variables. The states and the interface variables are related by simple linear equations, in K2 they are equal, which makes symbolic manipulation possible. After the manipulation the problem generates a set of ODEs. Note that other choices of state variables can generate a nonlinear relation between states and interface variables and thus in the general case making it impossible to do symbolic elimination of the algebraic constraints. The mathematical problem then becomes a set of DAEs after manipulation..

This discussion points out the mathematical properties of using subunit models for unit modelling. The OMSIM simulator takes care of ODE and DAE systems and the user does not have to make model approximations to get rid of these properties, but the model user should be aware of the mathematical problem that must be solved by the simulator.

B.4 HRSG plant application

The first major application of the K2 model database was the modelling and simulation of the HRSG side of a combined cycle power plant. HRSG stands for heat recovery steam generation. Exhaust gas from a gas turbine flows through a series of heat exchangers to produce heat for the evaporation of water to steam. The steam is used to produce electrical energy in a steam turbine and to heat water used for district heating. The condensed steam is recycled to the feedwater system before it enters the pan section again.

HRSG Modelling

Using the object-oriented methodology supported by OMOLA and the K2 model database the HRSG plant is decomposed in sections; deaerator, pan, steam turbine and condenser. These sections are in its turn built by assembling unit models taken from the unit libraries. The low-level control of variables like pressure and level is also described in the model.

The application models are grouped into a plant section library and one plant configuration library. It is easy to develop new plant configurations by reusing plant sections and units. Some commonly used plant sections like super heater system and boiler configuration should perhaps be added to the K2 database.

The HRSG application model is seen in Fig. B.8. The application is described as a structure of connected plant sections and it is built graphically. The plant sections are also application specific and are composed of units, e.g., pumps, boilers, etc. The decomposition of the boiler is partly seen in Fig. B.8. The boiler is composed of a set of units like circulation pump, evaporator, drum etc. The evaporator is a heat exchanger with saturated water on one side and flue gas on the other. The heat transfer between the two sides are modelled by a heat flow resistor subunit. In this particular case the heat resistance is decomposed into three parts, namely resistance from the water to the tube, the tube wall and the resistance on the gas side. Graphically this decomposition is seen in the four windows in the bottom left of Fig. B.8.

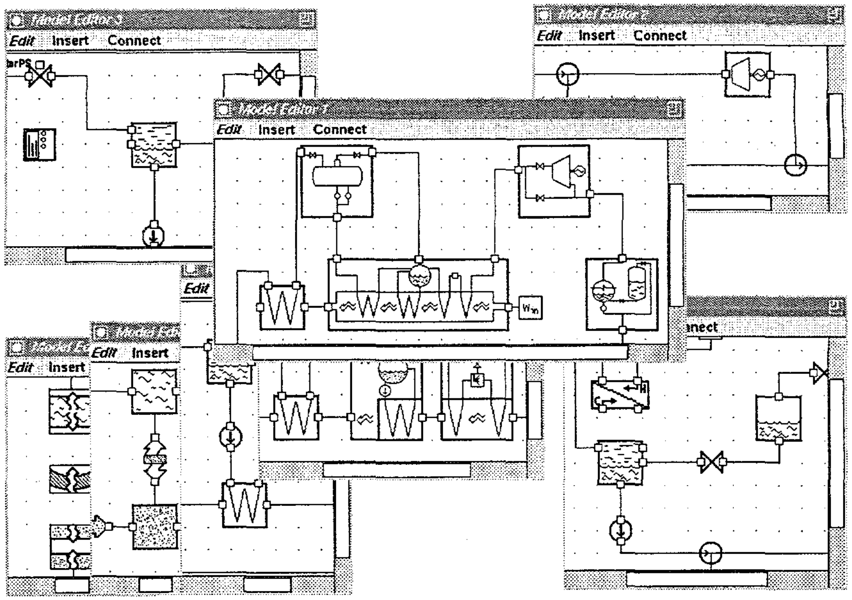


Figure B.8. Structure decomposition of the HRSG plant model in OmSim.

HRSG Application Study

The HRSG model has been used in a simulation study to evaluate different control strategies. This study has been made in the simulation environment OmSim. The simulator transforms the model into a differential-algebraic equation system with 61 state variables. The user has access to all the variables and parameters of the model and thus different variations to the control setpoints and to the plant load can be simulated interactively. The dynamic response of the plant and its control system can be evaluated from plots of the major variables. The model is found to perform well in simulations, see Eborn and Nilsson (1994).

One simulation of the HRSG plant model is shown in Fig. B.9 to B.11. The simulation experiment is a setpoint change of the pressure demand on the steam turbine. The setpoint is decreased 10% and this causes changes of all flow rates and pressures in the system.

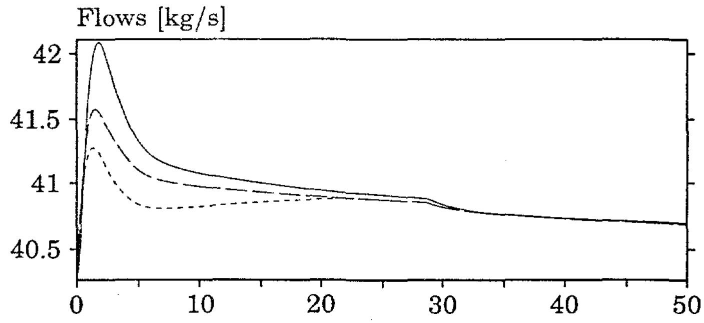


Figure B.9. Flow changes when the turbine pressure demand is decreased. (Solid line - feedwater, dashed - steam, dotted - turbine flow).

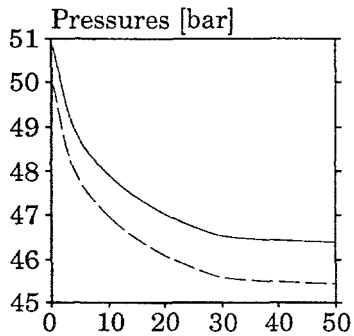


Figure B.10. Pressures decrease in response to the changed demand. (Solid line - drum pressure, dashed - turbine pressure.)

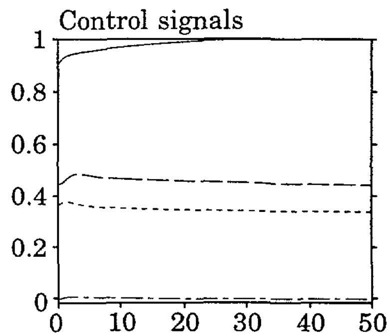


Figure B.11. Main control variables with decreasing turbine pressure. (Solid line - throttle valve, dashed - deaerator, dotted - feedwater, dash-dotted - bypass.)

B.5 Discussion

The K2 organization makes it possible to use object-oriented descriptions for the development of unit model behaviour. The advantage is decomposition of unit behaviour in phenomena objects and the model description is focused on the physical phenomena instead of the equations. The disadvantage is loss of efficiency due to encapsulated subunits that require local calculations of variables.

The types of phenomena captured by the set of subunits are limited in the current version of K2. The compartments only describe dynamic mass and energy balances. Momentum balances are assumed to be constant and are described mainly in the flow resistors. Dynamic momentum balances can be included without major changes. There is also a need for more flow resistor classes to capture a wider range of applications, like radiation, phase equilibria, two-phase flows etc.

All descriptions are based on lumped behaviour in the control volumes. A natural extension is to allow spatial descriptions both in the compartments and in the flow resistors. This results in distributed parameter systems and partial differential equations, PDEs. To handle PDEs efficiently there is a need for additional language constructions in OMOLA and manipulation procedures in OMSIM.

B.6 Conclusions

This paper illustrates the use of an object-oriented modelling language for modelling of thermal power plants. The major results are the organization of the model database for multiple users, called K2, and the model decomposition levels; system, unit and subunit. The open general modelling language, OMOLA, makes it possible to add new models. The organization of the model libraries also makes it easy to add new models in a structured way with a given level of detail. At this level the modelling is application oriented. The K2 model database gives the user the opportunities to work at different levels of detail and supports the user with a process application oriented environment together with the power of a general modelling language.

Acknowledgements

We would like to thank Sven Erik Mattsson, Mats Andersson and Tomas Schönthal for all help during our work with OMOLA and OMSIM. We are also grateful for all help and encouragement from Jan

Tuszynski and Ola Bernersson, Sydkraft Konsult AB. This work has been supported by Sydkraft AB.

References

ANDERSSON, M. (1994): *Object-Oriented Modeling and Simulation of Hybrid Systems.* PhD thesis ISRN LUTFD2/TFRT--1043--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

CELLIER, F. and H. ELMQVIST (1993): "Automated Formula Manipulation Supports Object-Oriented Continuous-System Modeling." *IEEE Control Systems Magazine*, **13:2**, pp. 28–38.

EBORN, J. and B. NILSSON (1994): "Object-oriented modelling and simulation of a power plant. Application study in the K2 project." Report ISRN LUTFD2/TFRT--7527--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

FOSS, B., S. WASBØ, and O. ØGÅRD (1995): "Object-oriented Process Modelling Applied to a Reactor." In BREITENECKER AND HUSINSKY, Eds., *Eurosim '95 Simulation Congress*, pp. 463–468. Elsevier, Amsterdam, Netherlands.

HYLLSETH, M. (1991): *Dynamic Simulation of Heat Recovery Boilers by Using the Sequential Modular Technique.* PhD thesis NTH-ITE 1991:29, Norwegian Institute of Technology, Trondheim, Norway.

MARQUARDT, W. (1993): "An Object-oriented Representation of Structured Process Models." In *European Symposium on Computer Aided Process Engineering – 1.*

MARQUARDT, W. (1994): "Trends in Computer-Aided Process Modeling." In *Process Systems Engineering, PSE'94,* vol. 1, pp. 1–24. Kyongju, Korea.

MATTSSON, S. E. and M. ANDERSSON (1992): "The ideas behind Omola." In *Proceedings of the 1992 IEEE Symposium on Computer-Aided Control System Design, CACSD '92.* Napa, California.

NILSSON, B. (1993): *Object-Oriented Modeling of Chemical Processes.* PhD thesis ISRN LUTFD2/TFRT--1041--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

NILSSON, B. (1994): "Guidelines for process model libraries using an object-oriented approach." In *Proceedings of the European Simulation Multiconference, ESM'94,* pp. 349–353. SCS, The Society for Computer Simulation, Barcelona, Spain.

NILSSON, B. (1996): "Experiences of Developing Process Model Libraries in Omola." In DAVIS *et al.*, Eds., *International Conference on Intelligent Systems in Process Engineering, ISPE'95*, vol. 92 of *AIChE Symposium Series no. 312*, pp. 388–392. CACHE and AIChE.

NILSSON, B. and J. EBORN (1994): "K2 model database—Tutorial and reference manual." Report ISRN LUTFD2/TFRT--7528--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

NILSSON, B. and J. EBORN (1995): "An object-oriented model database for thermal power plants." In BREITENECKER AND HUSINSKY, Eds., *Eurosim '95 Simulation Congress*, pp. 747–752. Elsevier.

ORDYS, A., R. KATEBI, M. JOHNSON, and M. GRIMBLE (1994): *Modelling and Simulation of Power Generation Plants.* Springer-Verlag, London, Great-Britain.

PIELA, P., T. EPPERLY, K. WESTERBERG, and A. WESTERBERG (1991): "ASCEND: An Object-Oriented Computer Environment for Modeling and Analysis: the Modeling Language." *Computers & Chemical Engineering*, **15:1**, pp. 53–72.

STEPHANOPOULOS, G., G. HENNING, and H. LEONE (1990): "MODEL.LA. A Modeling Language for Process Engineering—I. The Formal Framework." *Computers & Chemical Engineering*, **14:8**, pp. 813–846.

WASBØ, S. and B. FOSS (1995): "An Object-Oriented Method for Process Modelling." In RAWLINGS, Ed., *DYCORD+'95 - 4th IFAC Symposium on Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes*, pp. 117–122.

WOODS, E. (1993): *The Hybrid Phenomena Theory.* PhD thesis NTH-ITK 1993:60, Norwegian Institute of Technology, Trondheim, Norway.

Paper C

Simulation of a Power Plant using an Object-Oriented Model Database

Jonas Eborn and Bernt Nilsson

Abstract

An object-oriented approach to modelling and simulation of a Heat Recovery Steam Generation plant, HRSG, using a model database called K2 is presented. This database is a set of model libraries for modelling of thermal power plants described in the object-oriented modelling language OMOLA. The models in the database are based on first principles and describe mainly the dynamic behaviour in the water/steam cycle. The plant studied in this paper is the HRSG side of a combined cycle power plant. Using the object-oriented methodology supported by OMOLA and K2 the HRSG plant is decomposed in its sections and these sections are in turn built by assembling unit models taken from the unit libraries. The HRSG model has subsequently been used in a simulation study to evaluate different control strategies.

Keywords: Object modelling technique, process model, simulation, power generation.

C.1 Introduction

Thermal power plants are well suited for object-oriented modelling because of the extensive use of standard components like pumps, valves, heat exchangers, boilers etc. This makes it possible to capture a large number of power plant configurations with a limited number of model objects. The aim of this application study is to be able to simulate a small power plant, a so called *heat recovery steam generation plant*, HRSG. This and other types of power generation plants are discussed in Ordys *et al.* (1994). A sequential modular approach to modeling and simulation of a heat recovery boiler is presented in Hyllseth (1991).

The HRSG plant configuration, studied in this report, is seen in Figure C.1. Condensate water is pumped from the condenser to the deaerator via a preheater into the pan[1]. Feedwater is then pumped from the deaerator to the drum via an economizer which heats up the water to boiling temperature. The water in the drum is circulated through the evaporator and the two phases water and steam then flashes, separates, into the drum. Saturated steam is extracted from the drum and passes two super-heaters to produce super-heated steam before it enters the steam turbine. In the turbine the steam flow expands to low pressure and it is then condensed to water again in the condenser and can be recirculated to the deaerator.

The classes used in the modelling of the application has been structured according to the guidelines in Nilsson (1993, 1994). The plant can be viewed in a top-down fashion on different levels of complexity, called granularity levels; flowsheet, unit and subunit level, see Figure C.2. On the flowsheet level the plant itself and its sections are described. The unit level is where the building blocks performing unit operations, like pumps, tanks and heat exchangers are described and the subunit level is the lowest level where balance equations, functions and basic components are used for the behavioral description. These different levels will be dealt with in the following sections.

The K2 model database is a set of model libraries for modelling of thermal power plants described in the object-oriented modelling language OMOLA, see Andersson (1994) and Mattsson and Andersson (1992). The models in the database are based on first principles and describe mainly the dynamic behaviour in the water/steam cycle, see Nilsson and Eborn (1994). In the database there are unit libraries, subunit libraries and model component libraries. The unit libraries

[1] *Correction:* Using the word "pan" here is bad swenglish, it should be replaced by **steam generator**.

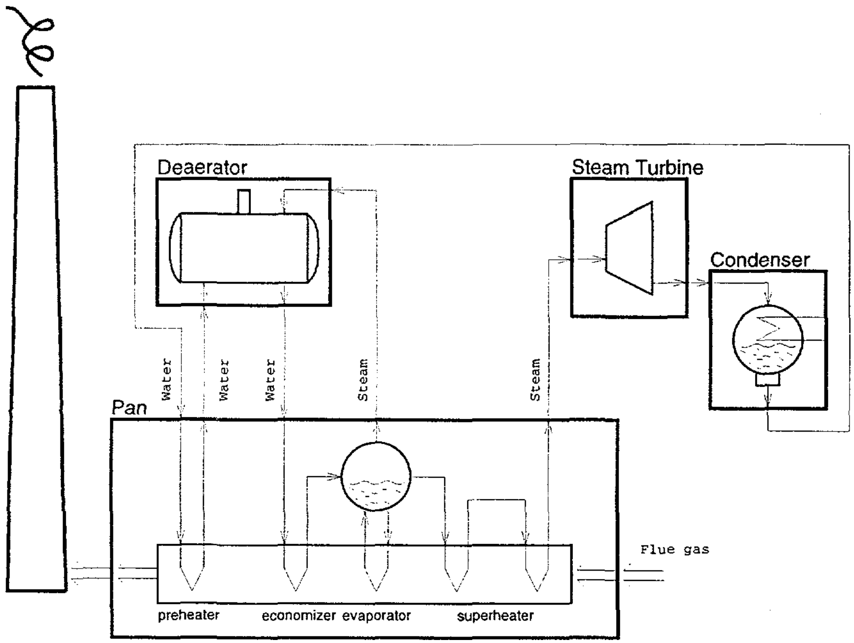


Figure C.1. The configuration of a heat recovery steam generation plant.

contain models of basic physical objects of varying complexity; pumps, valves, heat exchangers, boilers and turbines. The subunits are models of different media, compartments and flow resistors. Examples of media are sub-cooled water, water/steam mixtures, super-heated steam and flue gas. The compartment models are control volumes containing different media and the flow resistors describe different relationships between flow and pressure. The model component libraries contain general functions, variables, terminals and super classes used for structuring purposes. The database is further described in Nilsson and Eborn (1995).

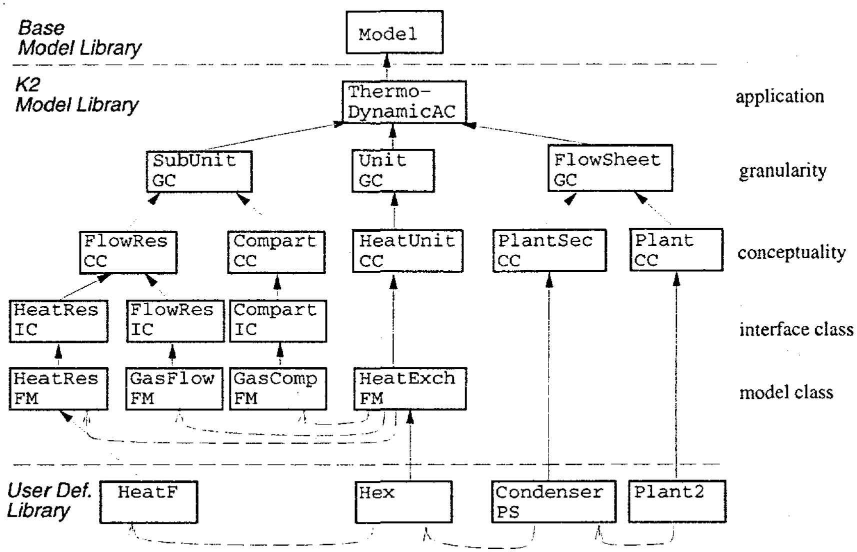


Figure C.2. Structuring guidelines of the model library. The suffices AC (Application Class), GC, CC and IC mark that these are classes used as super classes for structuring purposes only while the classes with suffix FM (Full Model) are complete and can be used in simulations.

C.2 Flowsheet models

On the flowsheet level the different sections of the plant are described. These sections are constructed from unit models, like pumps, tanks and heat exchangers. The four sections of the studied HRSG plant are described in the following; deaerator, pan, steam turbine and condenser.

Deaerator Section

The deaerator is modelled as a two-phase pressurized tank with water kept at 105 ℃. Preheated condensate water enters the tank through the recycle valve, which is used to control the water level. Another valve is used to control the tank pressure by drawing a small amount of steam from the boiler drum. The pumping power of the feedwater pump is governed by a feedforward from the feed valve controller of the boiler to ensure that the pressure drop over the feed valve does not increase when the valve is closed.

Pan Section

The pan section consists of preheater, economizer, boiler and super-heater. Condensate water is preheated by the flue gas in the preheater, which is the last heat exchanger of five in the flue gas pathway. Before the feedwater enters the boiler it passes a heat exchanger called the economizer, where it is heated up to boiling temperature. In the boiler water is evaporated and then the saturated steam is passed through the super-heater where it is heated up further. The layout of the pan is shown in Figure C.4.

Feedwater enters the boiler drum through the feed valve which is used to control the water level in the drum. Water is circulated with a pump through the evaporator in order to generate steam. When

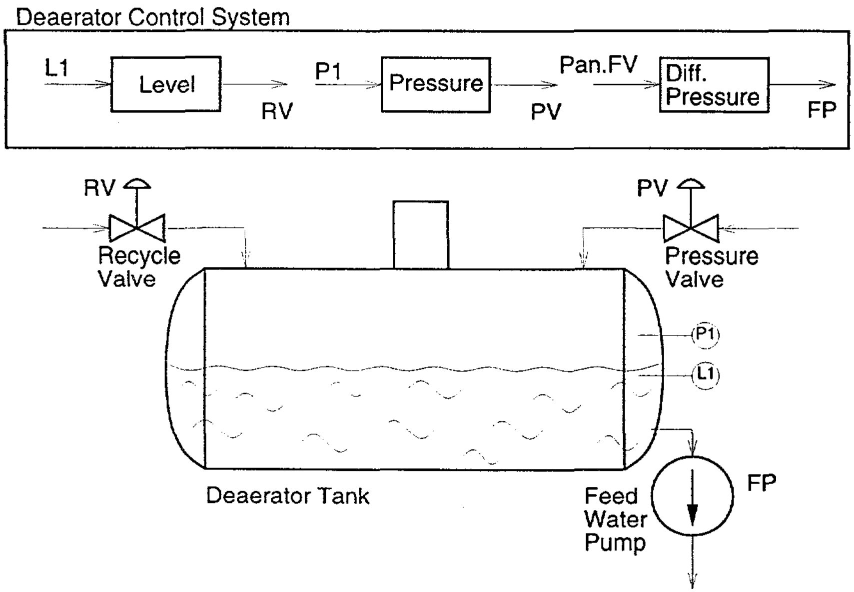


Figure C.3. The deaerator configuration.

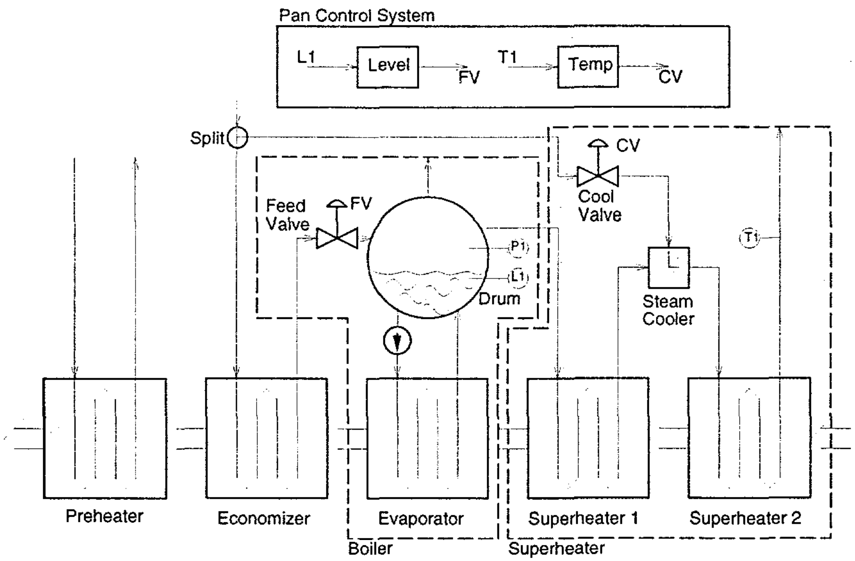


Figure C.4. The pan section configuration with the internal structure in boiler and super-heater.

the mixture of water and steam from the evaporator enters the boiler drum it flashes, separates in two phases. The saturated steam produced leaves the drum to the following super-heater.

The super-heater is composed of two heat exchangers with a spray cooler between them. In the super-heater the saturated steam from the boiler is heated up close to the temperature of the flue gas entering the pan, approximately 500 ℃. With the spray cooler the temperature of the steam leaving the super-heater can be controlled.

Steam Turbine Section

The super-heated steam is expanded in the steam turbine producing electrical energy. The produced output power is proportional to the steam flow through the turbine and the enthalpy drop over the turbine. The throttle valve and the bypass valve are used to control the turbine inflow pressure and also the pressure upstream to the boiler drum. The bypass valve makes it possible to bypass steam directly to the

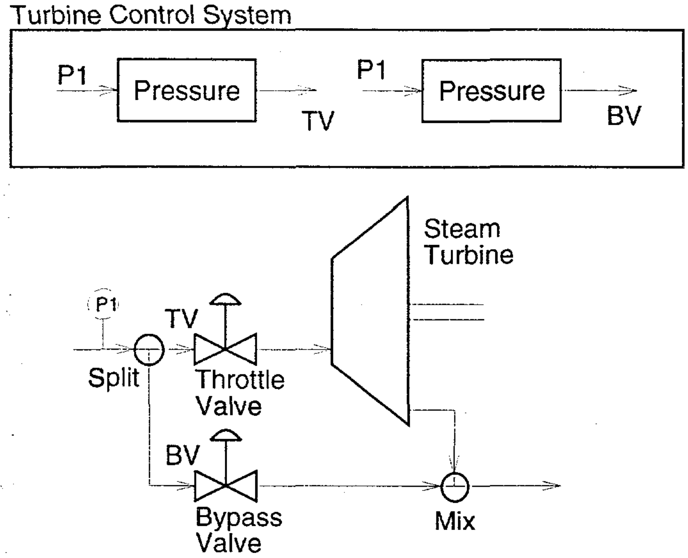


Figure C.5. The steam turbine configuration.

condenser. It is only open during startup and large pressure transients.

Condenser Section

Waste heat in the steam from the turbine is removed by cooling with district-heating water in the condenser and the steam condenses to water. It is then immediately pumped out by the condenser pump to the deaerator. Only a small amount of water is kept in the condenser drum. Make-up water is instead supplied by the buffer tank, which is kept at atmospheric pressure. With the two water valves the logic controller can be used to keep the condenser level and the condensate water pressure within limits.

C.3 Unit Models

The hierarchical structure and function of the HRSG sections was discussed in the previous section. The sections are built from processing units. In the HRSG plant the following units are used; heat exchangers, pressurized and atmospheric tanks, a steam turbine, pumps, control valves, mixers and splitters and a spray cooler. There are also additional units for the control of the plant. The units are discussed in more detail in the following subsections.

Subunits and Interface

All processing units are decomposed into subunits. These subunits can be categorized in two major groups namely *compartments* and *flow resistors*. Dynamic mass and energy balances, expressed as pressure and enthalpy, are described in compartment models. The media and heat flow between compartments are described by flow resistors. There are compartments and flow resistors for one-phase media, like water, steam or flue gas, and there are also two-phase compartments for the mixture of water and steam. The modelling of the subunits is further described in Nilsson and Eborn (1994). The units are interconnected mainly by media flows. These flow interfaces are described by record

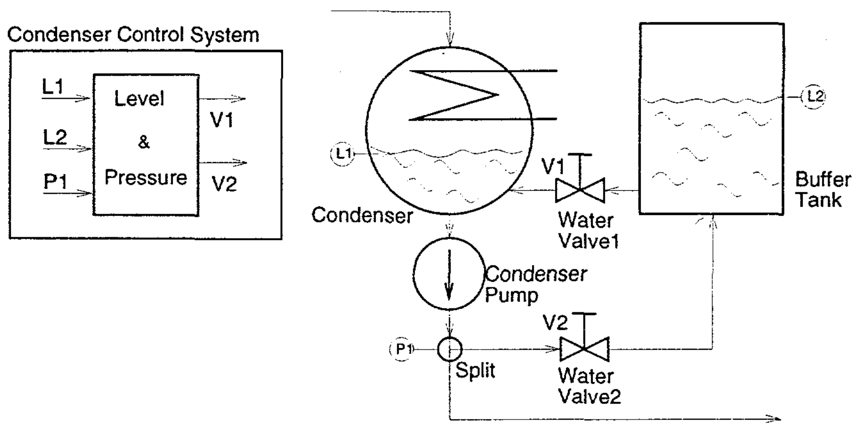


Figure C.6. *The condenser configuration.*

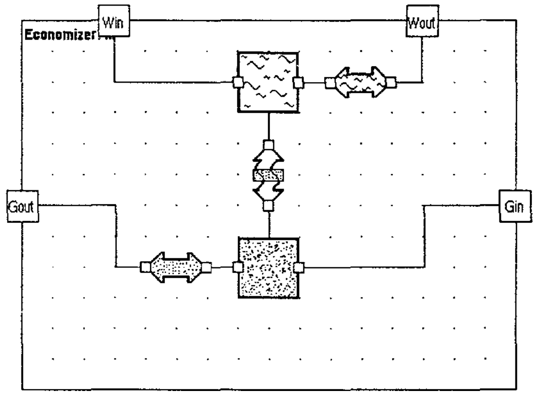


Figure C.7. The configuration of a heat exchanger, i. e.,, an economizer.

terminals of five variables. Three of them are common to all media; mass flow, pressure and enthalpy. The fourth and fifth variables are media specific, one determines the medium phase and the other is depending on this, e. g.,, height, temperature or quality. By using media specific terminals you gain error control since it is more difficult for a user of the model database to make the wrong connections, but you loose polymorphism. That means that you cannot simply change the media description in a heat exchanger to use it in another place, you also need to change the terminals.

Heat units

The heat units are mainly different types of heat exchanger models, each one composed of two compartments describing the two sides of the unit. The two compartments describe the dynamics of the media on the primary and secondary side. The five heat exchangers in the pan all have flue gas on one side and water or steam on the other. In the condenser there is a heat exchanger with water on one side and condensing steam on the other. The heat interaction between the two compartments is described by a heat flow resistor and on each compartment outflow there are flow resistors. The configuration of the economizer in the pan section can be seen in Figure C.7.

Tank models

The boiler drum, condenser drum and the deaerator tank are pressurized tanks modelled as two-phase compartment models. In the condenser section there is also a buffer tank. This tank is assumed to be at atmospheric pressure and is modelled as an open water compartment with mass dynamics instead of pressure dynamics.

Flow units

The flow units used in the HRSG model are; pumps, valves, mixers, splitters and spray coolers. Pumps and valves are modelled as flow resistors. Both kinds are static descriptions of the relation between pressures and medium flow. The valves are pressure drop descriptions for water or steam. The pump model uses a relation between the pressure 'drop' (negative in this case) and the applied pumping power. The remaining flow units contain no flow description but simply describe a junction of several flows of the same medium, possibly of different phases. In a mixer two flows of the same media and phase is mixed. In a splitter one flow is split into two flows with the same enthalpy. A steam cooler sprays a small water flow into a large steam flow to decrease the steam temperature. The cooler is modelled as a two-phase mixer.

Turbine

The turbine in the HRSG plant is a steam turbine, modelled as a steam compartment and a critical expansion flow resistor. The compartment accounts for the volume dynamics in the turbine and also creates a small lag between the inflow and the outflow. It also has the function of breaking up the large algebraic system which would be created if the dynamics of the turbine volume were neglected. The enthalpy drop in the turbine is calculated with an isentropical thermal efficiency approximation.

Controllers

The SISO-controllers used in the plant are analog PID-controllers implemented with switching between manual/automatic operation and controller saturation. The controller models are taken from a control equipment database described in Nilsson (1993). In the condenser sys-

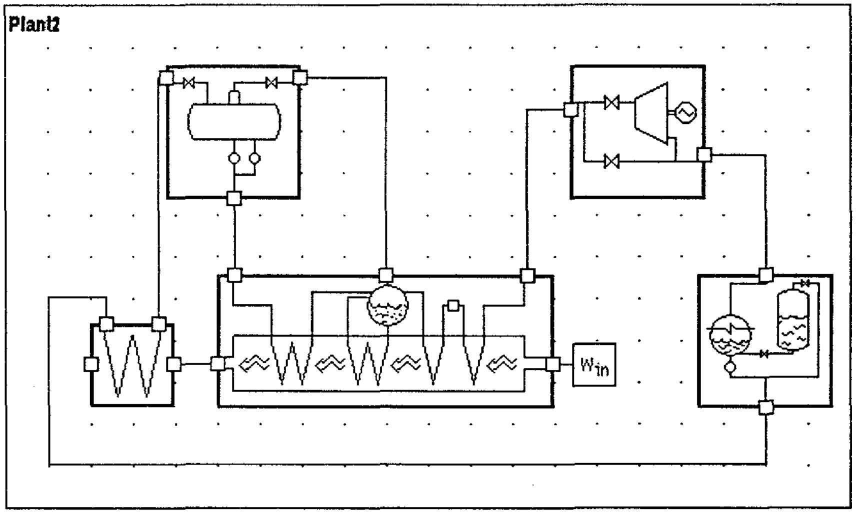


Figure C.8. The HRSG model.

tem there is also a special logic controller built from analog and digital function blocks, these are also taken from the control database.

C.4 Simulation Studies

The HRSG model has been used in a simulation study to evaluate different control strategies. This study has been made in the simulation environment OMSIM, see Mattsson and Andersson (1992). The user has access to all the variables and parameters of the model and thus different variations to the control setpoints and to the plant load can be simulated interactively. The dynamic response of the plant and its control system can be evaluated from plots of the major variables. The model is found to perform very well in the simulations and some sample plots are shown in this paper. The model as seen in the graphical editor is shown in Figure C.8.

Simulation setup

The plant is assumed to receive a flue-gas flow of 350 kg/s at h = 627 kJ/kg, T = 500 °C from the gas turbine. The initial state of the plant is assumed to be steady which gives a circulation flow that is a little larger than 40 kg/s. To start the simulation you also need initial values of most of the 61 dynamic and algebraic state variables. In the following simulations some deviations from the steady state have been studied.

Turbine control setpoint changes

The steam turbine system consists of a steam turbine and a bypass leading hot steam directly to the condenser. The pressure before the turbine is controlled by two PI-controllers and control valves, the throttle valve and the bypass valve. In the initial state the throttle valve opening is about 89% and the bypass valve is closed. The bypass is used during the startup procedure and to reduce sudden peaks in steam pressure. During normal operation the bypass valve is closed.

Case 1: The behaviour of the plant when the turbine pressure setpoints are changed is studied in a simulation where the pressure setpoint is decreased with 10% to 45 bars. The flow through the turbine increases rapidly first and then settles at a slightly higher flow rate than before. The bypass valve also opens for a short period of time but closes again when the pressure goes under 46 bars, which is the setpoint of the bypass valve controller. The controlled pressure does not reach the lower setpoint since the throttle valve saturates after 38 seconds. The total efficiency goes up ever so little due to the higher flow rate although this also causes a slightly higher back pressure in the turbine. The results are shown in the Figure C.9.

Flue-gas flow variations

Supervisory load control of the gas turbine will cause the flue-gas flow and thus the heat flow into the pan to increase or decrease. The effects of this have been studied in two simulations.

Case 2: First a simulation where the flue-gas flow increased slowly during 10 seconds to 400 kg/s. The results are shown in Figures C.10-C.11. The increased heat flow makes the circulation flow and drum

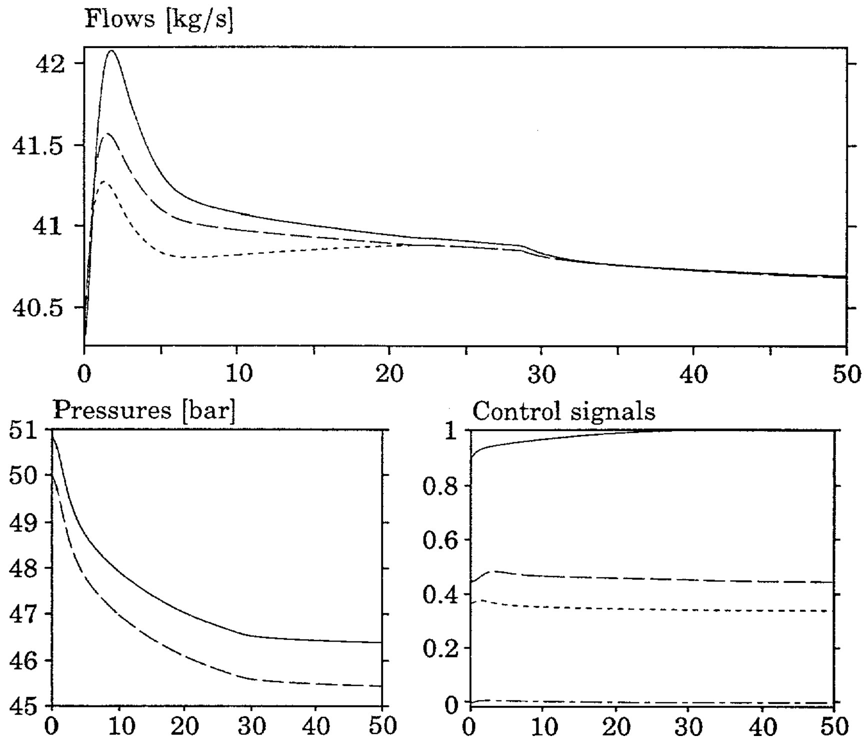


Figure C.9. Simulation results from case 1: *top:* Process flows (From the top: feedwater, steam and turbine flow.) *left:* Boiler drum and turbine pressures, *right:* the main control variables (From the top: throttle, deaerator, boiler, bypass valve.).

pressure increase but the turbine controllers returns the pressure to the reference value after 2-300 seconds. The reason why the pressure at first is brought down and then starts increasing again after 64 s is that the throttle valve saturates at this time. Then the bypass controller starts opening the bypass valve at 132 s and the pressure is brought to the bypass controllers setpoint, which is 51 bars. The control signal of the bypass controller is very small in the plot, only 0.1%, but this is

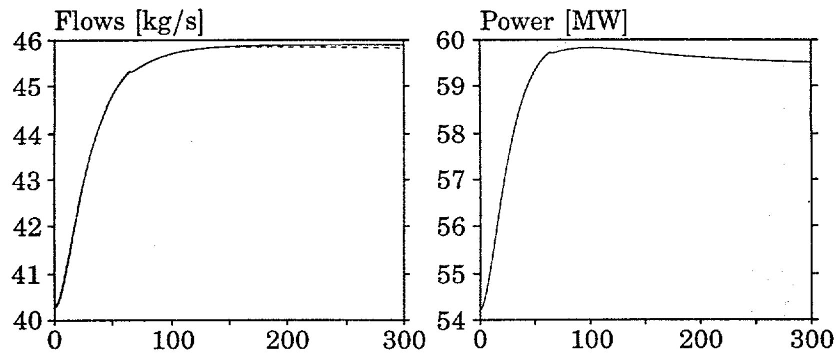


Figure C.10. Case 2, *left:* flow changes, *right:* produced power.

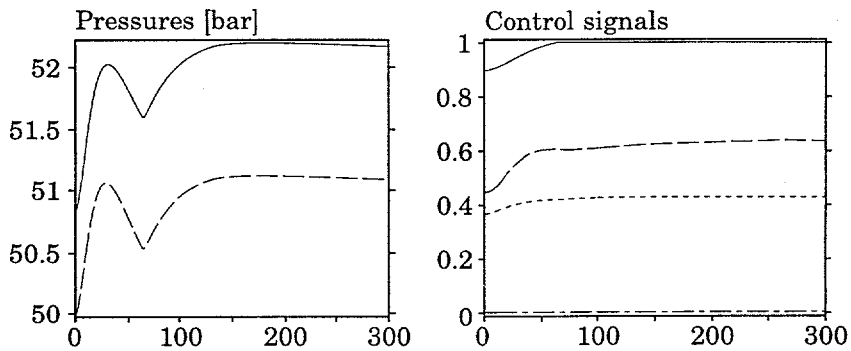


Figure C.11. Case 2, *left:* main pressures *right:* control variables.

sufficient to keep the pressure at the desired level. The increased heat flow makes the output power go up from 54 to 60 MW.

Case 3: When the flue-gas flow is decreased to 300 kg/s the results look like in Figure C.12. The pressures and flows start to decrease immediately but the turbine controller catches up quickly and stabilizes the pressure at the setpoint after 200 seconds. The flow decrease makes the output power drop from 54 to 48 MW.

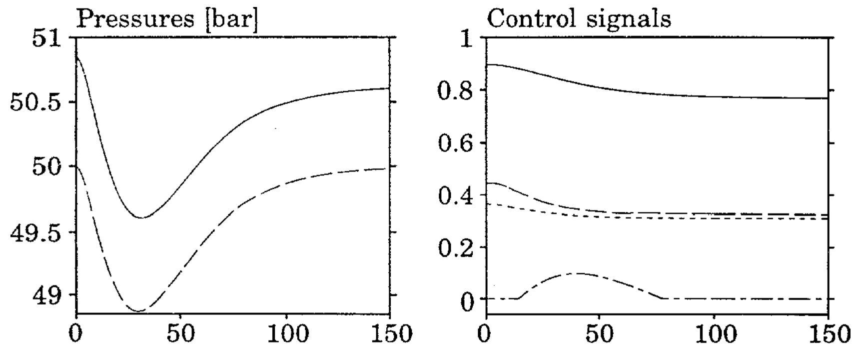


Figure C.12. Case 3, *left:* main pressures, *right:* control variables with decreasing flue-gas flow. (The dotted line is the steam injection valve.)

C.5 Conclusions

This paper presents an application of the modelling and simulation environment OMSIM. The application is a heat recovery steam generator plant and illustrates well the power of using object-oriented model databases for assembling a large scale model with complex structure. The database models are based on first principles, i. e.,, mass and energy balances. The sample simulations show that the models perform well.

C.6 Acknowledgements

The authors would like to thank the people behind OMSIM, Sven Erik Mattsson, Mats Andersson and Tomas Schönthal. We also must thank Jan Tuszynski and Ola Bernersson at Sydkraft Konsult AB for many valuable discussions. The work has been supported by Sydkraft and by the Swedish National Board for Industrial and Technical Development.

References

ANDERSSON, M. (1994): *Object-Oriented Modeling and Simulation of Hybrid Systems.* PhD thesis ISRN LUTFD2/TFRT--1043--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

HYLLSETH, M. (1991): *Dynamic Simulation of Heat Recovery Boilers by Using the Sequential Modular Technique.* PhD thesis NTH-ITE 1991:29, Norwegian Institute of Technology, Trondheim, Norway.

MATTSSON, S. E. and M. ANDERSSON (1992): "The ideas behind Omola." In *Proceedings of the 1992 IEEE Symposium on Computer-Aided Control System Design, CACSD '92.* Napa, California.

NILSSON, B. (1993): *Object-Oriented Modeling of Chemical Processes.* PhD thesis ISRN LUTFD2/TFRT--1041--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

NILSSON, B. (1994): "Guidelines for process model libraries using an object-oriented approach." In *Proceedings of the European Simulation Multiconference, ESM'94,* pp. 349–353. SCS, The Society for Computer Simulation, Barcelona, Spain.

NILSSON, B. and J. EBORN (1994): "K2 model database—Tutorial and reference manual." Report ISRN LUTFD2/TFRT--7528--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

NILSSON, B. and J. EBORN (1995): "An object-oriented model database for thermal power plants." In BREITENECKER AND HUSINSKY, Eds., *Eurosim '95 Simulation Congress,* pp. 747–752. Elsevier.

ORDYS, A., R. KATEBI, M. JOHNSON, and M. GRIMBLE (1994): *Modelling and Simulation of Power Generation Plants.* Springer-Verlag, London, Great-Britain.

Paper D

Parameter Optimization of a Non-linear Boiler Model

Jonas Eborn and James Sørlie

Abstract

The object of this study is a steam generation process model developed by Åström and Bell. The paper reports improvements obtained by tuning uncertain physical parameters as well as a verification of the model structure. Employing measurement data and methods from system identification, the paper demonstrates the complementary nature of first-principle modeling and identification. Results show that statistical methods, combined with a systematic search strategy, allow improvement of a larger number of parameters than is possible through manual tuning.

Keywords: control systems, nonlinear modeling, optimization, parameter estimation, system identification.

D.1 Introduction

Models built upon first principles are useful for both analysis and design. The physical relevance and insight they provide often reveal structural properties of the actual process, something which black-box models cannot do. With first-principle models there are frequently parameters which are uncertain or even impossible to determine without detailed measurements. Often, to achieve a good visual fit to measured data, trial-and-error techniques are used to adjust parameters manually. In this paper, we study the use of system identification applied to first-principle models; cf. Bohlin (1991). Our aim is to demonstrate a systematic approach to the task of parameter tuning.

As a case study, we use the third and fourth-order non-linear implicit differential equation models developed by Åström and Bell (1988; 1993; 1996). The mathematical model has a relatively small number of physical parameters and input conversion factors. Most of the physical parameters are well determined from construction data. However, some parameters are obtained by rough estimates, like metal masses and a friction factor in the flow through the down-comers/risers circuit. The friction factor has been manually adjusted to achieve good simulation results. It is of interest to see what improvements can be obtained thru optimization of these uncertain parameters.

Software tools that have been used for this study are the modeling environment OMSIM, see Andersson (1994), and the grey-box identification tool-kit IDKIT, see Graebe (1990); Sørlie (1996). Motivating this paper is our desire to demonstrate the complementary nature of modeling and identification, as well as the need for software tools integrating modeling and simulation with parameter optimization of general non-linear first-principle models.

D.2 Model Definition

The details of the model's derivation are given in Åström and Bell(1988; 1993; 1996); here, we briefly survey their results. Printing provisions prohibit including all the modeling equations. They have been programmed in OMOLA, see Andersson (1994); Sørlie and Eborn (1997), and are available upon request from the authors.

An idealized physical model for the system is shown in Figure D.1. Steam vapor is vented from the drum with flow-rate q_s. Feed-water enters the drum in a sub-cooled liquid state with flow-rate q_{fw} and temperature T_{fw}. Steam vapor is generated by channeling the liquid phase from the drum through a down-comers/risers circuit. The heat flow-rate Q into the risers comes from the combustion of fuel. The flow-rate into the circuit q_{dc} is driven by the density gradient caused by the

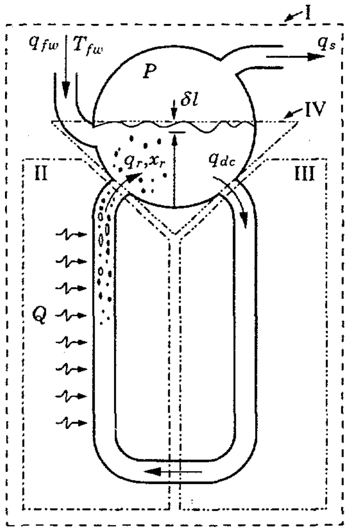


Figure D.1. Ideal physical model of a steam generation process.

phase change in the risers. At the risers outlet, the two-phase mixture is characterized by the mass flow-rate q_r and vapor mass-fraction x_r.

The fundamental modeling simplification is that the two phases of water inside the system are everywhere in a saturated thermodynamic state. With this assumption, all thermodynamic properties can be characterized by one independent variable. The drum pressure P is chosen to be this key state variable since it is the most globally uniform variable in the system. Another key assumption is an instantaneous and uniform thermal equilibrium between water and metal everywhere. This simplifies including thermal capacitance effects.

Indicated in Figure D.1 are the boundaries of four thermodynamic control volumes. Mass and energy balances for the global control volume (c.v. I) yield two state equations. The state variables are pressure P and the total volume of liquid water in the system V_{wt}. By combining the mass and energy balances for c.v. II to eliminate the flow-rate q_r, a third state equation is derived with the vapor mass-fraction x_r as state variable. By considering fluid friction in c.v. III, a fluid momentum balance establishes the flow-rate q_{dc}. A combination of the mass and energy balances for c.v. IV yields a fourth state equation with state variable V_{sd}, the volume of steam vapor below the liquid surface. Assembled in matrix notation, the fourth-order model structure (i.e., a set of implicit differential state equations) is:

$$\mathcal{M}_4: \quad \begin{bmatrix} e_{11} & e_{12} & 0 & 0 \\ e_{21} & e_{22} & 0 & 0 \\ 0 & e_{32} & e_{33} & 0 \\ e_{41} & e_{42} & e_{43} & e_{44} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial t} V_{wt} \\ \frac{\partial}{\partial t} P \\ \frac{\partial}{\partial t} x_r \\ \frac{\partial}{\partial t} V_{sd} \end{bmatrix} = \begin{bmatrix} q_{fw} - q_s \\ Q + h_{fw} q_{fw} - h_s q_s + \Delta_{\mathrm{I}} \\ Q - h_c x_r q_{dc} + \Delta_{\mathrm{II}} \\ \dfrac{V_{sd} - V_{sd}^0}{\tau_{sd}} + \dfrac{\left(h_{fw} - h_w\right) q_{fw} + \Delta_{\mathrm{IV}}}{\rho_s h_c} \end{bmatrix} \quad \text{(D.1)}$$

The elements of the coefficient matrix e_{11}, e_{12}, e_{21}, etc., are state dependent. The complexity of these expressions prohibits including them here; see Sørlie and Eborn (1997). On the right, Δ_{I}, Δ_{II} and Δ_{IV} represent under-modeling, i.e., unmodeled energy interactions (nominally taken to be zero). The initial state conditions are parameterized $[V_{wt}^0, P^0, x_r^0, V_{sd}^0]^T$. In addition to these, the model involves seven physical parameters: metal masses m_d, m_r, m_{dc}, volumes V_d, V_r, V_{dc}, and a fluid friction coefficient in the down-comers k. Known constants are the specific heats C_{fw} and C_p for the feed-water and metal respectively.

For the purpose of level control, Bell and Åström (1996) proposed a variational measurement model for the liquid level in the drum:

$$\delta l = \left(\left(V_{wd} - V_{wd}^0\right) + \left(V_{sd} - V_{sd}^0\right)\right) / A_d. \quad \text{(D.2)}$$

The level variation δl is caused by variations in the volumes of liquid in the drum V_{wd}[1] and the steam below the surface V_{sd}. This model

[1] $V_{wd}(t) = V_{wt}(t) - V_{dc} - (1 - \alpha_r(t))\, V_r$ where α_r is the total volume fraction of steam in the risers, i.e., $\alpha_r = V_{sr}/V_r$; an approximation with form $\alpha_r \approx \mathrm{fcn}(P, x_r)$ is given in Åström and Bell (1988); Åström and Bell (1993).

introduces one additional physical parameter: A_d, the drum's cross-sectional area at the nominal level. The aim of including variation in V_{sd} is to capture the level dynamics known as the "shrink-and-swell" effect; cf. Bell and Åström (1996).

To assess the necessity of including the fourth state equation in $\mathcal{M}_4$, we shall investigate parameter optimization of both third and fourth-order model structures. In the third-order structure $\mathcal{M}_3$, the state variable V_{sd} in (D.2) is replaced with an instantaneous value. Engineering judgment suggests several approximations for its value. The fourth-order structure $\mathcal{M}_4$ involves a similar set of hypotheses for the bubble-residence time-constant τ_{sd}. The heuristics for the values which have been tested are:

$$V_{sd} = \begin{cases} b_1 & \text{hyp. } 0, \\ b_1 \alpha_r V_r & \text{hyp. } 1, \\ b_1 x_r q_r & \text{hyp. } 2, \end{cases} \qquad \tau_{sd} = \begin{cases} \dfrac{b_1 \rho_s}{q_s} (V_d - V_{wd}) & \text{hyp. } 0, \\ \dfrac{b_1 \rho_s}{x_r q_r} (2V_{sd}^0 - V_{sd}) & \text{hyp. } 1. \end{cases}$$

In each model structure, b_1 is a "grey-box" parameter to be optimized.

For parameter optimization, we use three datasets obtained from experiments reported in Åström and Eklund (1972). Figure D.2 shows the simulation interface to the five measured inputs: two steam flows, feed-water flow, feed-water temperature and fuel flow. An interface with real data necessitates conversion factors; the simulation schematic shows several. Most uncertain is the calibration of the heat input Q. Because the chemical energy content of the fuel is known to vary, this gain has been probabilistically modeled with a nominal value Q_{cf} and a known, bounded range Q_{rng}. More certain are the calibrations of the steam mass flow-rates $q_{s_{cf}}$ and feed-water mass flow-rate $q_{fw_{cf}}$. Assuming liquid flow measurements are more precise than vapor flow measurements, we shall consider the later[2] a known constant. This leaves as additional parameters for optimization Q_{cf} and $q_{s_{cf}}$.

In addition to the stochastic modeling of the gain Q_{cf}, the simulation interface includes simple stochastic input and output-error models. The focus of this paper is parameter optimization in a deterministic setting; accordingly, on the instantaneous output-error models will be

[2] *Correction:* The word "later" here refers to $q_{fw_{cf}}$ and not vapor flow measurements.

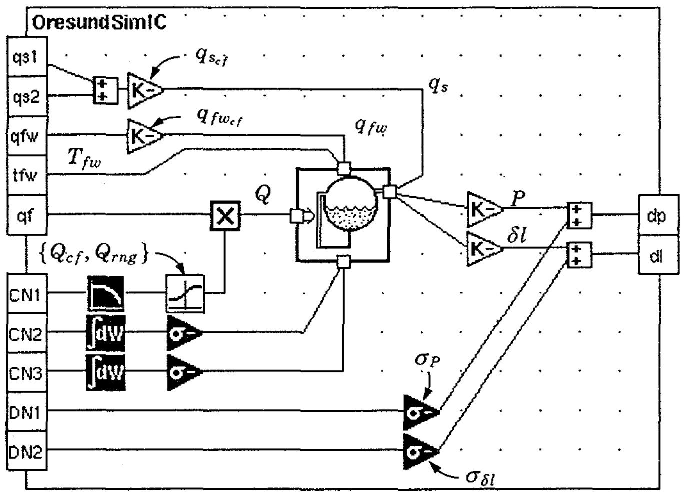


Figure D.2. Omola definition of the simulation interface to signals of the experimental data, and stochastic input and output-error models.

investigated here. In Sørlie and Eborn (1997), the input-error models are used to investigate the effects of under-modeling, i. e., Δ_I etc. Summarizing the model definition, we have third and fourth-order model structures $\mathcal{M}_3$ and $\mathcal{M}_4$ with parameterization $[m_d, m_r, m_{dc}, V_d, V_r, V_{dc}, k, A_d, b_1, Q_{cf}, q_{s_{cf}}]$.

D.3 Parameter Optimization

Parameter estimation has been investigated with the third and fourth-order model structures $\mathcal{M}_3$ and $\mathcal{M}_4$. The three datasets involve perturbation on different inputs; steam flow, fuel flow and feed-water flow. The IDKIT software, see Graebe (1990), utilizes a gradient search method to minimize the likelihood function for the given observations. Along with the parameter estimates, the software also calculates values of the Akaike Information Criterion (AIC) and the loss function; these values are useful for comparisons and hypothesis testing. The search method requires good initial guesses for the parameters. Reasonable values were obtained from Åström and Bell (1993; 1996). Notationally, we denote a nominally parameterized model $M_3(\Theta_0)$. Estimated models are named in a similar fashion; e.g.,, $M_3(\hat{\Theta}_1)$ is the model obtained from dataset 1 with the third-order model structure.

Choosing free parameters: Augmentation and over-parameterization

The modeling goal of this study was to obtain a good deterministic model. Nevertheless, an error description is very important for the optimization. Purely deterministic models can not explain everything in the data. In optimization, this leads to convergence problems. By first estimating measurement error variances with nominal physical parameters and subsequently augmenting the free parameter space, a good deterministic model is obtained. The estimated error variances give a measure of the model uncertainty. The principle used when choosing free parameters (i.e., free for search) is to start with the parameters that are least known or have a large impact on outputs. In our case this means error variances, input conversion factors and some initial values. Then, augmentation to include hypothesis testing and optimization of physical parameters can be done.

When choosing what parameters to optimize, over-parameterization is an important issue. This is very common in physical models based on first principles. Energy storage in metal e.g., depends only on the product mC_p; thus these parameters can not be estimated independently. More subtle over-parameterizations may be overlooked. In this study we set out to estimate the friction factor, k. It was deemed impossible since k mainly affects offset in the drum level, which makes it coupled to the initial value x_r^0.[3] The coupling can be seen by examining the parameter sensitivities, i.e., the Hessian of the likelihood function. This reveals cross-couplings between such parameters. Evaluation of estimated models with different, fixed values for k show that the model behavior is very insensitive to changes in k. In fact, the AIC value is least for the nominal value k=0.005.

[3] Assuming the system was in near equilibrium during the experiments, the equilibrium solution for the third state equation can be used to parameterize x_r^0.

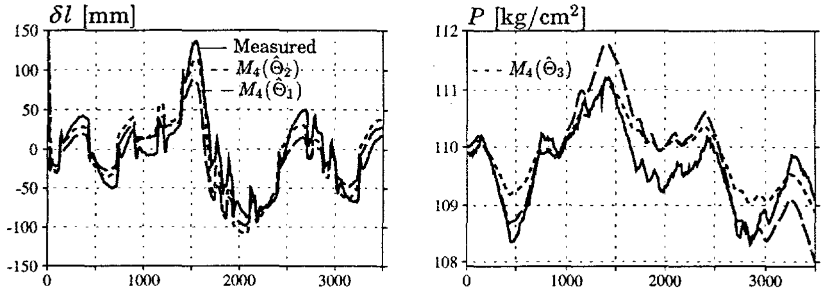


Figure D.3. Model obtained from dataset 1 validated against drum-level from dataset 2 and pressure from dataset 3. Compared also with the best model for each dataset.

Estimates for third-order model structure

The main result from running optimization on the third-order model is that the total mass, $m = m_r + m_d + m_{dc}$, is estimated to be 500 tons, considerably more than the nominal value 300 tons. The difference is not surprising though since the nominal value was obtained by rather rough calculations. The estimate is consistent with the first two datasets, while the third dataset gives estimates considerably higher, $m > 1000$ tons. This result is less reliable though, since the excitation in this dataset is very small and time variations in Q_{cf} has a considerable impact on the pressure, which is the output affected by m; see Figure D.3.

Testing the different hypotheses concerning V_{sd} reveals that hypothesis 2 is considerably better in conjunction with the first dataset, yielding AIC=2920 compared to 2980 and 3080 for hypotheses 1 and 0 respectively. This is not the case when applied to the other datasets. Optimization suppresses the influence of V_{sd} by reducing the factor b_1; values of the Akaike criterion are almost exactly the same with different hypotheses. This suggests that the simple hypotheses are insufficient and there are additional dynamics concerning V_{sd}. These are introduced as a state in the fourth-order model.

Dataset:	1. Steam flow				2. Fuel flow				3. Feed-water flow			
	m	σ_P	$\sigma_{\delta l}$	AIC	m	σ_P	$\sigma_{\delta l}$	AIC	m	σ_P	$\sigma_{\delta l}$	AIC
Model	[ton]	[bar]	[mm]		[ton]	[bar]	[mm]		[ton]	[bar]	[mm]	
$M_3(\Theta_0)$	300	0.9	50	3494	300	1.3	43	3634	300	0.63	74	3484
$M_3(\hat{\Theta})$	542	0.8	27	2920	501	1.2	30	3304	1080	0.3	20	2046
$M_4(\hat{\Theta})$	460	0.8	10	2217	392	1.2	20	3030	1054	0.3	10	1622
Cross-validation of $M_4(\hat{\Theta}_1)$					460	1.3	23	3174	460	0.5	10	1885

Table D.1 Estimation results for different model structures and datasets.

Estimates for fourth-order model structure

Results from the fourth-order model are consistent with the previous results in that the estimates of the total mass give similar results and the additional dynamics give an appreciable contribution; drum level error and AIC values decrease according to Table D.1.

With the additional dynamics in the fourth-order structure there are also possibilities to optimize other parameters. Besides total mass, also drum and riser mass influence the behavior. These have been estimated on the first dataset and were found to be: m_d=61 tons, m_r=272 tons. These figures are reasonable relative to each other and the total mass. Also the grey-box factor scaling the bubble time constant τ_{sd} was estimated to be b_1=1.9. To check the validity of the results this model was used on the other datasets after estimation of only conversion factors and x_r^0. It was also compared to the best possible model for those datasets. Comparisons can be seen in Table D.1 and in Figure D.3.

The validated model performs almost as good as the best ones for each dataset. In the fuel-flow data there seems to be an overshoot phenomenon in drum-level not caught by the model, this explains why the model optimized on the first dataset gives a higher AIC value. In the third dataset there is the problem with low excitation and time-varying Q_{cf} mentioned earlier.

D.4 Structure determination

The task of structure determination is supported by having modeling and simulation tools integrated with optimization/identification tools. In this study we have used OmSim to create different model structures and hypotheses and test them in simulation to see their qualitative behavior. The model equations are then exported to IdKit for parameter optimization and hypothesis testing. Statistical measures like the AIC give an objective evaluation of the model structures. This together with the subjective measure obtained in simulation provide the necessary information whether to accept or reject a model structure.

In this case study two different model structures were compared; the third and fourth-order models described in previous sections. The statistical measures given in Table D.1 all show that the fourth-order model describes drum-level much better and this is confirmed in the simulations shown in Figure D.4, see the differences in drum-level behavior at t =1100 and 1700 seconds. The attempt at cross-validation of the fourth-order model could be confusing just looking at the AIC values since in dataset 3 the AIC=1622 for the best model is much lower than 1885. But in simulations it can be seen that this mainly depends on deficiencies in the experiment, e.g., time-varying conversion factors. Qualitatively, the model from the first dataset $M_4(\hat{\Theta}_1)$ behaves better than the statistically 'best' model; in Figure D.3, the increased mass estimates in $M_4(\hat{\Theta}_3)$ effectively flatten the pressure variations

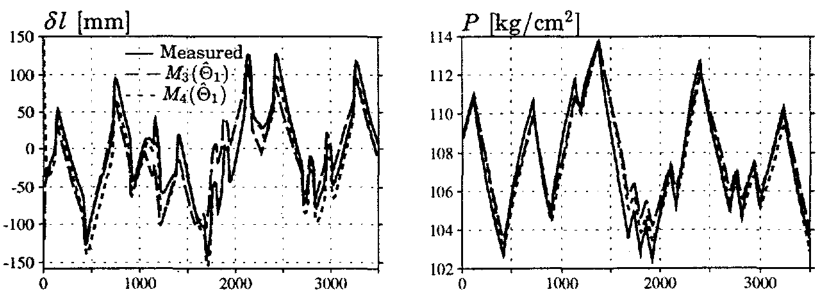


Figure D.4. Simulations of third and fourth-order models compared to data from dataset 1.

Dataset:	1. Steam flow				2. Fuel flow				3. Feed-water flow			
Hypoth.	b_1	σ_P	$\sigma_{\delta l}$	AIC	b_1	σ_P	$\sigma_{\delta l}$	AIC	b_1	σ_P	$\sigma_{\delta l}$	AIC
0	0.5	0.8	14	2476	0.22	1.2	27	3232	0.43	0.34	14	1859
1	1.9	0.8	10	2220	2.4	1.2	20	3030	1.9	0.34	10	1622

Table D.2 Evaluation of hypotheses of the fourth-order model structure.

and inadvertently suppress dynamics present in $M_4(\hat{\Theta}_1)$.

A close integration of modeling and identification tools also makes it easier to test different model hypotheses. For the third-order model this was reported in the previous section. In the fourth-order model there are two hypotheses concerning bubble residence time, τ_{sd}. These have been tested in favor of hypothesis 1; see Table D.2. AIC values for all three datasets are lower and the scaling factors b_1 are all close to the same value, which favors hypothesis 1.

D.5 Conclusions

This paper reports on parameter estimation on two different non-linear model structures for a drum-boiler process. The results verify that the fourth-order structure better describes the complicated drum-level dynamics. A large number of uncertain physical parameters have been estimated, a task which would have been impossible by commonly used trial-and-error testing. This is especially true since uncertain input conversion factors introduce drift in simulations. Using a search strategy to first estimate conversion factors and error variances, and then iteratively augment free parameter space, up to 10 parameters can be simultaneously optimized without difficulty. Statistical measures of the model fidelity, together with qualitative information obtained in simulation, give the information needed to assess different model hypotheses.

We believe that the dual nature of modeling and identification demonstrated here is very powerful and should be supported by well integrated software tools. For linear model structures such tools exist. In the non-linear case much remains to be done.

Acknowledgments

This work was supported by a research grant from Sydkraft AB.

References

ÅSTRÖM, K. J. and R. D. BELL (1988): "Simple drum-boiler models." In *IFAC Int.Symposium on Power Systems, Modelling and Control Applications.* Brussels, Belgium.

ÅSTRÖM, K. J. and R. D. BELL (1993): "A nonlinear model for steam generation processes." In *Preprints IFAC 12th World Congress.* Sydney, Australia.

ANDERSSON, M. (1994): *Object-Oriented Modeling and Simulation of Hybrid Systems.* Ph.D. thesis TFRT–1043–SE, Dept. of Automatic Control, Lund Inst. of Technology, Lund, Sweden.

ÅSTRÖM, K. and K. EKLUND (1972): "A simplified non-linear model of a drum boiler-turbine unit." *Int. J. Control,* **16:1**, pp. 145–169.

BELL, R. D. and K. J. ÅSTRÖM (1996): "A fourth order non-linear model for drum-boiler dynamics." In *IFAC'96, Preprints 13th World Congress,* vol. O, pp. 31–36. San Francisco, CA.

BOHLIN, T. (1991): *Interactive System Identification: Prospects and Pitfalls.* Springer-Verlag, Berlin, Germany.

GRAEBE, S. F. (1990): *Theory and Implementation of Gray Box Identification.* Ph.D. thesis TRITA–REG–9006, Royal Institute of Technology, Dept. of Automatic Control, Stockholm, Sweden.

SØRLIE, J. and J. EBORN (1997): "A grey-box identification case study: The Åström–Bell drum-boiler model." Technical Report ISRN LUTFD2/TFRT--7563--SE. Dept. of Automatic Control, Lund Inst. of Technology, Lund, Sweden.

SØRLIE, J. A. (1996): *On Grey-Box Model Definition and Symbolic Derivation of Extended Kalman Filters.* Ph.D. thesis TRITA–REG–9601, S3–Automatic Control, Royal Institute of Technology, Stockholm, Sweden.

Appendix E

Derivations of Equations

E.1 State Transformation

A fundamental property of a volume of some medium is the conservation of mass and energy. Thus, to describe the state of the medium it would be possible to use mass and energy. However, it is often inconvenient to use mass and energy as state variables. They are very difficult to measure directly and they are not intuitive. Instead the state of the medium must be expressed in two other state variables. For one-phase media, like gases, pressure and temperature is a very appealing choice. This cannot be used for phase transitions from water to steam since the state of a mixture of water and steam would be undetermined, temperature is constant under constant pressure during the evaporation of water to steam.

A common choice of state variables, which we have used here, is then absolute pressure, p, and specific enthalpy, h. There are other choices, e. g., density and temperature, but practical considerations influenced our choice. Steam table function calls in OmSim use pressure and enthalpy as states. It is also preferable to have pressure as a state since this is used in flow equations.

In the following the transformation of balance equations in mass and energy into differential equations in pressure and enthalpy is shown.

Balance equations: Elementary balance equations of mass, m, and energy, e are

$$\frac{dm}{dt} = \Sigma w_i - \Sigma w_o \tag{E.1}$$

$$\frac{de}{dt} = \Sigma w_i h_i - \Sigma w_o h_o + \Sigma Q \tag{E.2}$$

The entering and leaving mass flows are w_i and w_o and h_i and h_o are the entering and leaving specific enthalpies. Heat transferred to the volume is represented by Q.

Rewrite the energy balance: We can use the specific energy, $e = mu$, and the definition of enthalpy, $mh = mu + pV$, to rewrite the energy balance in (E.2)

$$\frac{d(mh)}{dt} = \frac{dm}{dt}h + m\frac{dh}{dt} = \frac{de}{dt} + \frac{d(pV)}{dt}$$

In this expression we can solve for the enthalpy derivative and get

$$\frac{dh}{dt} = \frac{1}{m}(\frac{de}{dt} - h\frac{dm}{dt} + V\frac{dp}{dt} + p\frac{dV}{dt}) \tag{E.3}$$

Rewrite the mass balance: To rewrite the mass balance we express the density derivative using the chain rule (Note that $\rho = \rho(p,h)$.)

$$\frac{d\rho}{dt} = \left.\frac{\partial\rho}{\partial p}\right|_h \frac{dp}{dt} + \left.\frac{\partial\rho}{\partial h}\right|_p \frac{dh}{dt} = \alpha_p\frac{dp}{dt} + \alpha_h\frac{dh}{dt} \tag{E.4}$$

Mass and its derivative can then be expressed in terms of the density as

$$m = \rho V \quad \Rightarrow \quad \frac{dm}{dt} = V\frac{d\rho}{dt} + \rho\frac{dV}{dt} \quad \Rightarrow \quad \frac{d\rho}{dt} = \frac{1}{V}(\frac{dm}{dt} - \rho\frac{dV}{dt})$$

By solving (E.4) for pressure derivative and then substituting the above expression for the density derivative, we get the following pressure expression

$$\frac{dp}{dt} = \frac{1}{\alpha_p}(\frac{d\rho}{dt} - \alpha_h\frac{dh}{dt}) = \frac{1}{\alpha_p}(\frac{1}{V}(\frac{dm}{dt} - \rho\frac{dV}{dt}) - \alpha_h\frac{dh}{dt}) \tag{E.5}$$

This gives us two differential equations describing pressure, (E.5), and enthalpy, (E.3) depending on each other. These equations can now be rewritten in order to obtain explicit differential equations for pressure and enthalpy. This is strictly not necessary since OmSim can solve implicit differential equations, but it is desirable for efficiency reasons.

By substituting the enthalpy derivative (E.3) into (E.5) we obtain

$$\frac{dp}{dt} = \frac{1}{\alpha_p}(\frac{1}{V}(\frac{dm}{dt} - \frac{m}{V}\frac{dV}{dt}) - \alpha_h(\frac{1}{m}(\frac{de}{dt} - h\frac{dm}{dt} + V\frac{dp}{dt} + p\frac{dV}{dt})))$$

and collecting terms for the pressure derivative gives us

$$(\alpha_p + \alpha_h\frac{V}{m})\frac{dp}{dt} = (\frac{1}{V} + \alpha_h\frac{h}{m})\frac{dm}{dt} - \alpha_h\frac{1}{m}\frac{de}{dt} - (\frac{m}{V^2} + \alpha_h\frac{p}{m})\frac{dV}{dt}$$

By solving the expression above for dp/dt and substituting it into the enthalpy equation, (E.3), we obtain

$$\begin{aligned}\frac{dh}{dt} &= \frac{1}{m}(\frac{de}{dt} - h\frac{dm}{dt} + p\frac{dV}{dt} \\ &\quad + \frac{V}{\alpha_p + \alpha_h\frac{V}{m}}((\frac{1}{V} + \alpha_h\frac{h}{m})\frac{dm}{dt} - \alpha_h\frac{1}{m}\frac{de}{dt} - (\frac{m}{V^2} + \alpha_h\frac{p}{m})\frac{dV}{dt}))\end{aligned}$$

Pressure and enthalpy state equations: The transformation results in the following two differential equations describing the dynamics in pressure and enthalpy

$$\begin{aligned}\frac{dp}{dt} &= \frac{\rho}{\alpha_p\rho + \alpha_h}((\frac{\rho + \alpha_h h}{\rho V})\frac{dm}{dt} - \frac{\alpha_h}{\rho V}\frac{de}{dt} - (\frac{\rho^2 + \alpha_h p}{\rho V})\frac{dV}{dt}) \\ \frac{dh}{dt} &= \frac{\rho}{\alpha_p\rho + \alpha_h}((\frac{1 - \alpha_p h}{\rho V})\frac{dm}{dt} + \frac{\alpha_p}{\rho V}\frac{de}{dt} + (\frac{\alpha_p p - \rho}{\rho V})\frac{dV}{dt})\end{aligned}$$

In the compartment models the coefficients of the differential equations are written as auxiliary variables, T_{ij}. The original balance equations, Equations E.1 and E.2, are used to give the mass derivative, dm, and the energy derivative, de. This gives us the equations

$$\begin{aligned}p' &= T_{11}dm + T_{12}de + T_{13}dV \\ h' &= T_{21}dm + T_{22}de + T_{23}dV\end{aligned}$$

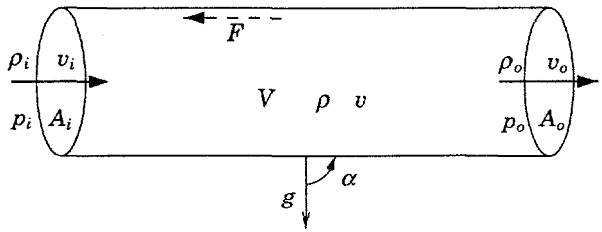


Figure E.1. Control volume for the momentum balance.

All the coefficients, T_{ij}, are expressed using different medium properties that are calculated in a medium sub-model. Note that in most of the compartment models the volume is constant which means that $\frac{dV}{dt} = 0$.

E.2 Momentum Balance

In its most general form the momentum balance (also called Cauchy's first law of motion) involves complex relations of the velocity field and stress tensor in three dimensions. This is very well covered in text books like Bird *et al.* (1960). Here we put up the momentum balance for a fixed control volume, e. g., a length of piping. Then the momentum balance takes the form

$$V\frac{d}{dt}\rho v = \overbrace{A_i\rho_i v_i^2 - A_o\rho_o v_o^2}^{\text{convection terms}} + \overbrace{A_i p_i - A_o p_o}^{\text{pressure force}} - F + \overbrace{V\rho g\cos\alpha}^{\text{gravitation}} \qquad \text{(E.6)}$$

The different variables are introduced in Figure E.1. Subscript values refer to inflow and outflow properties while v and ρ are suitable mean values for the total volume, V, of medium, F is the total friction force from the walls.

In the flow descriptions in **K2** we use an approximation of the general momentum balance. By assuming that momentum dynamics are fast we can neglect the derivative term. We also assume that there is no accumulation of mass in a flow module, i. e., the mass inflow and outflow are the same

$$w_i = A_i \rho_i v_i = w_o = A_o \rho_o v_o = w$$

In general we can also assume that the flow area is constant at the inflow and outflow, $A_i = A_o = A$. This gives us the following *static* flow description

$$0 = \frac{w^2}{A}\left(\frac{1}{\rho_i} - \frac{1}{\rho_o}\right) + A(p_i - p_o) - F + V\rho g \cos\alpha \tag{E.7}$$

This expression will be used to solve for the static mass flow, w. To do this an expression for the friction force is needed since F depends on the flow. In Coulson and Richardson (1977) a dimensionless friction factor, ϕ, is used. This gives the friction force as

$$F = \phi \rho v^2 \pi d l$$

where d and l are the diameter and length of the pipe. Expressions for ϕ are also given for the cases of laminar and turbulent flow in a pipe. We have used another common way of describing frictional losses, using a loss factor including both the friction factor and dimensions of the pipe. This gives us the expressions

$$z_{loss} = 8\phi\frac{l}{d} \qquad \Rightarrow \qquad F = \frac{z_{loss}}{2}\rho v^2 A = \frac{z_{loss}}{2}\frac{w^2}{\rho A}$$

which eventually will give simple expressions for the mass flow.

Incompressible flow equation

For water and other liquids where density varies very little it is natural to make the assumption that they are incompressible, $\rho = constant$. This makes the first term in (E.7) disappear. If we then substitute the expression for F and solve the flow equation we obtain

$$w = A\sqrt{\frac{2\rho}{z_{loss}}(\Delta p + \rho g \Delta z)} \tag{E.8}$$

where $\Delta z = \frac{V}{A}\cos\alpha$ is the height difference between the two sides of the pipe.

Compressible flow equation

For gases and steam the density varies substantially with pressure and thus you can not make the assumption that they are incompressible. However, the density is often very low, which makes the gravitation term in (E.7) very small. Neglecting this term and again solving the flow equation for mass flow we obtain

$$w = A\sqrt{\frac{\Delta p}{\frac{1}{\rho_o} - \frac{1}{\rho_i} + \frac{z_{loss}}{2\rho}}}$$

which is a slightly more complicated expression than (E.8). In this expression we need to calculate the mean density, ρ, either from the two other densities or from some mean pressure. To avoid this we instead use the approximation that $\rho = \rho_i$ and get the expression for compressible flow as

$$w = A\sqrt{\frac{\rho\,\Delta p}{\frac{\rho}{\rho_o} - 1 + \frac{z_{loss}}{2}}} \tag{E.9}$$

This approximation gives an error of less than 1% compared to using the mean density, $\rho = (\rho_i + \rho_o)/2$, when the density variation is about 5%.

Appendix F

Notation and Symbols

F.1 Table of Notation, Units and Symbols

Much of the naming in this thesis corresponds to some kind of standard notation used in thermodynamic literature. However, some ambiguities exist. To clear out these and help the reader not familiar with thermodynamics follows below a table of most symbols used in the thesis.

Symbol	Unit	Description/Quantity
α	W/m^2K	Heat transfer coefficient
α_g	1	Gas radiation absorbance
α_p	kg/m^3Pa	Partial derivative of density with respect to pressure at constant enthalpy, $\left.\frac{\partial\rho}{\partial p}\right\|_h$
α_h	kg^2/m^3J	Partial derivative of density with respect to enthalpy at constant pressure, $\left.\frac{\partial\rho}{\partial h}\right\|_p$
γ'	J/kg	Latent heat, or evaporation energy
ΔT_{lm}	K	Logarithmic mean temperature difference
ε_g	1	Gas radiation emittance
η_t	1	Isentropic thermal efficiency factor
λ	W/Km	Thermal conductivity
μ	Pa s	Dynamic viscosity

Symbol	Unit	Description/Quantity
ρ	kg/m^3	Density
Φ	1	Friction factor, dimensionless
A	m^2	Area
C_p	J/kgK	Heat capacity
d	m	Diameter
D_H	m	Hydraulic diameter
e	J	Total energy
F	N	Force
g	m/s^2	Gravitational constant, 9.81
h	J/kg	Specific enthalpy
k	W/m^{2}K	Overall heat transfer coefficient
K_v	(m^2)	Valve coefficient, water flow rate at Δp=1 bar
m	kg	Mass
n	1/s	Rotational speed
Nu	1	Nusselt number, dimensionless
p	Pa	Pressure
Pr	1	Prandtl number, dimensionless
Re	1	Reynolds number, dimensionless
q	m^3/s	Volumetric flow
Q	J/s	Heat flow
u	J/kg	Specific inner energy
U	m^{2}K/W	Heat resistance, $1/\alpha$
V	m^3	Volume
v	m/s	Flow velocity
w	kg/s	Mass flow
W_s	W	Shaft work
z_{loss}	1	Loss factor, dimensionless