

Design and Simulation of an Efficient Power Plant Boiler Using OpenModelica

Ancy S Alphonson
M.Tech C.I.S.
S.I.S.T., Trivandrum
ancya20@gmail.com

Abhirraj N Metkar
Senior Engineer
C.I.G., CDAC
ametkar@cdac.in

Lizmol Stephen
Assistant Professor in CSE
S.I.S.T., Trivandrum
lizmol007@gmail.com

Abstract

Simulation is the representation of the behaviour of a real process through the resolution of a mathematical model. Particular attention has been devoted to model the boiler furnace dynamics well, because power plant modeling experts have been successively neglecting the boiler furnaces in their attempts due to the complexity in modeling and non-availability of proper measurements for furnace variables. In order to explore the details about modeling and simulation of a boiler, Simulink and OpenModelica are used. Modelica is a new generation Modeling language. Openmodelica is an open source Modelica environment developed and supported by Linkoping University and the Open Source Modelica Consortium (OSMC). Modelica is also an object-oriented equation based programming language.

Keywords: Modelica, OpenModelica, Simulink,

1. Introduction

The goal is to develop a model that captures the key dynamical properties of the boiler over a wide operating range. The model is based on physical principles and has a number of parameters, most of which are determined from construction data and few of them from field test data. From modeling point of view, the boiler is divided into five subsystems and for each subsystem, mathematical models are developed based on the modeling principles and the knowledge gained during the boiler system study. The subsystem models are :Boiler furnace model, Boiler drum model, Primary superheater model, Attemperator model and Secondary superheater model. The subsystem models are then integrated to obtain a full-fledged integrated

model for the boiler. Simulation experiments are carried out based on the nominal values of system variables and parameters collected from a BHEL, 210MW Boiler.

Openmodelica [1] is an open source Modelica environment developed and supported by Linkoping University and the Open Source Modelica Consortium (OSMC). Modelica is a language for modeling of complex physical systems such as Robotics, Aircrafts, Satellites etc. The OpenModelica environment consists of several interconnected subsystems. The goal is to create a complete modeling and compilation and simulation environment based on free software distributed in source code and executable form which is intended for use in research, teaching, and industry. The OpenModelica environment is a collection of tools, called OpenModelica Tools, to create a complete Modelica model. After instantiating the models can be simulated and the results plotted as a chart.

2. Related Work

Mathematical Modeling and Simulation of a boiler using Simulink is an existing work. Simulink, developed by MathWorks, is a commercial tool for modeling, simulating and analyzing multi-domain dynamic systems. Simulink is a block diagram environment for multi domain simulation and Model-Based Design.

Simulink is most commonly used in control theory and digital signal processing for multi domain simulation and Model-Based Design. Compared to OpenModelica there are several drawbacks for Simulink, they are 1) Simulink does not have a standardized graphical notation. 2) Simulink is based on the signal-oriented paradigm which always forces a causal dependence between inputs and outputs of a block when solving equation systems. 3) Simulink

does not support inheritance-concepts for classification of components in order to enable their reuse.

3. OpenModelica

OpenModelica is an open-source Modelica-based modeling and simulation environment intended for industrial and academic usage. Its long-term development is supported by a non-profit organization – the Open Source Modelica Consortium (OSMC). The goal with the OpenModelica effort is to create a comprehensive Open Source Modelica modeling, compilation and simulation environment based on free software distributed in binary and source code form for research, teaching, and industrial usage.

Modelica is primarily a modeling language, sometimes called hardware description language, that allows you to specify mathematical models of complex physical systems, e.g. for the purpose of computer simulation of dynamic systems where behavior evolves as a function of time. Modelica is also an object-oriented equation based programming language, oriented towards computational applications with high complexity requiring high performance.

The four most important features of Modelica are: 1) Modelica is based on equations instead of assignment statements. This permits acausal modeling that gives better reuse of classes since equations do not specify a certain data flow direction. Thus a Modelica class can adapt to more than one data flow context. 2) Modelica has multi-domain modeling capability [2], meaning that model components corresponding to physical objects from several different domains such as e.g. electrical, mechanical, thermodynamic, hydraulic, biological and control applications can be described and connected. 3) Modelica is an object-oriented language with a general class concept that unifies classes, generics known as templates in C++, and general subtyping into a single language construct. This facilitates reuse of components and evolution of models [2]. 4) Modelica has a strong software component model, with constructs for creating and connecting components. Thus the language is ideally suited as an architectural description language for complex physical systems, and to some extent for software systems.

The OpenModelica environment consists of several interconnected subsystems [3], as depicted in Figure 1. The following subsystems are currently integrated in the OpenModelica environment:

An Interactive Session Handler that parses and interprets commands and Modelica expressions

for evaluation, simulation, plotting etc. The session handler also contains simple history facilities, and completion of file names and certain identifiers in commands.

A Modelica Compiler Subsystem, translating Modelica to C code, with a symbol table containing definitions of classes, functions and variables. Such definitions can be predefined, user-defined, or obtained from libraries. The compiler also includes a Modelica interpreter for interactive usage and constant expression evaluation. The subsystem also includes facilities for building simulation executables linked with selected numerical ODE or DAE solvers.

An Execution and Runtime Module, this module currently executes compiled binary code from translated expressions and functions, as well as simulation code from equation based models, linked with numerical solvers. In the near future event handling facilities will be included for the discrete and hybrid parts of the Modelica language.

Eclipse Plug-in Editor/Browser, the Eclipse plug-in called MDT (Modelica Development Tooling) provides file and class hierarchy browsing and text editing capabilities. Some syntax highlighting facilities are also included. The Eclipse framework has the advantage of making it easier to add future extensions such as refactoring and cross referencing support.

OM Notebook DrModelica Model Editor. This subsystem provides a lightweight notebook editor, compared to the more advanced Mathematica note books available in MathModelica. This basic functionality still allows essentially the whole DrModelica tutorial to be handled. Hierarchical text documents with chapters and sections can be represented and edited, including basic formatting. Cells can contain ordinary text or Modelica models and expressions, which can be evaluated and simulated. However no mathematical typesetting facilities are yet available in the cells of this notebook editor.

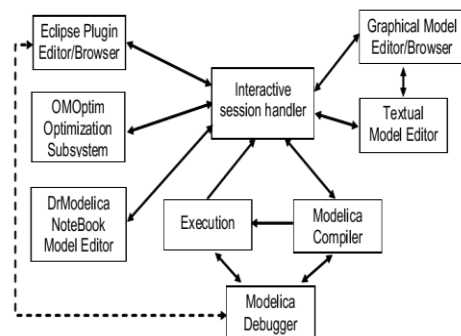


Fig. 1 Modelica Environment

Graphical Model Editor/Browser OMEdit: This is a graphical connection editor, for component based model design by connecting instances of Modelica classes, and browsing Modelica model libraries for reading and picking component models. The graphical model editor also includes a textual editor for editing model class definitions, and a window for interactive Modelica command evaluation.

Optimization Subsystem OMEdit. This is an optimization subsystem for OpenModelica, currently for design optimization choosing an optimal set of design parameters for a model. The current version has a graphical user interface, provides genetic optimization algorithms and Pareto front optimization, works integrated with the simulators and automatically accesses variables and design parameters from the Modelica model.

Modelica Debugger. The current implementation of debugger provides debugging for an extended algorithmic subset of Modelica, excluding equation-based models and some other features, but including some meta-programming and model transformation extensions to Modelica. This is conventional full-feature debugger, using Eclipse for displaying the source code during stepping, setting breakpoints, etc. Various back-trace and inspection commands are available. The debugger also includes a data-view browser for browsing hierarchical data such as tree- or list structures in extended Modelica.

4. Boiler Operation

The principle of boiler operation is simple: the aim is to evaporate water and superheat the steam obtained using the energy which is released in a combustion reaction

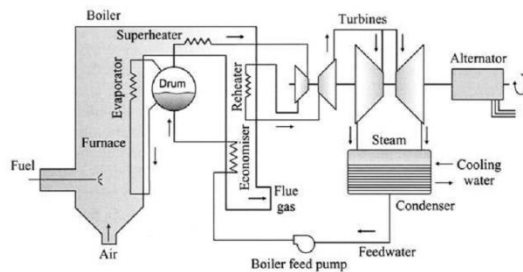


Fig. 2 A typical power plant

Before the feedwater enters into the actual steam generator system, it passes through a heat exchanger where it is preheated before it enters the economizer. Saturated steam taken from the boiler is

used as the heating medium. Using the flue gases generated in the actual boiler as the heating medium, the water continues to be heated in the economizer but it does not turn into vapour. From the economizer, the water goes to the steam drum and from there it passes through the downcomers into the water drum. It is then distributed through the different steam generation circuits associated with the combustion chamber and the convection bank[5].

As the water rises through the tubes (by natural circulation) part of it turns into vapour, so that it forms a mix of vapour and water when it returns to the steam drum. The steam generated in the drum must be separated from the water, and for this purpose a cyclonic separator system has been installed. The dry steam leaving the steam drum goes to the superheating section which is formed by two superheaters separated by an attemperator. When it leaves the secondary superheater, the steam goes to process. This boiler, which has been chosen as a basis for the model, produces superheated steam at 40 bar and 380°C from water at 120°C [5].

5. Modeling and Simulation of Boiler

In my work, mathematical modeling and simulation [7] of the boiler is done using OpenModelica. The boiler is divided into five subsystems such as Furnace, Drum, Primary Superheater, Attemperator and Secondary Superheater. Each subsystem models are modeled using mathematical equations [6] based on law of conservation of mass and energy. Two equations for the Secondary Superheater is given below:

Rate of change in internal energy of SSH steam is calculated by the equation

$$V_s Y_s C_v (dT_s/dt) = a_i \alpha_{ms} (T_m - T_s) + F_s (C_{si} T_{si} - C_s T_s).$$

Heat energy stored in the SSH tube metal is calculated by the equation,

$$M_m C_m (dT_m/dt) = a_o \alpha_{gm} (T_g - T_m) - a_i \alpha_{ms} (T_m - T_s).$$

Table 1 Variable Description

Variables	Description
V_s	Control volume of SSH
Y_s	Specific weight of steam in SSH
C_v	Specific heat of steam at constant volume in SSH
T_s	Main steam temperature
a_i	Inside heat transfer area of SSH
α_{ms}	Metal to steam heat transfer

	coefficient
T_m	SSH inlet temperature
F_s	Mass flow rate of steam in SSH
C_{si}	Specific heat of steam at SSH inlet
T_{si}	SSH inlet steam temperature
C_s	Specific heat of main steam
M_m	Mass of SSH metal
C_m	Specific heat of SSH metal
a_o	Outside heat transfer area
α_{gm}	Gas to metal heat transfer coefficient
T_g	Temperature of furnace gas

Similarly each subsystem models had a number of mathematical equations and they are modeled. The advantages of using mathematical models are 1) A process can be analysed in depth, determining which variables or parameters are critical and have a decisive effect on overall system behaviour. 2) They are of great help in the search for optimum operating conditions. 3) Limit or extreme conditions, far away from the normal operating conditions, can be studied safely to analyse their consequences.

The subsystem models are then integrated to obtain a full-fledged integrated model for the boiler. Simulation experiments are carried out based on the nominal values of system variables and parameters collected from a BHEL, 210MW Boiler. Simulation is the representation of the behaviour of a real process through the resolution of a mathematical model. More similar simulation results are obtained using OpenModelica. Compared to simulink, Modelica has a standardized general purpose graphical notation for modeling different views of the system definition. Modelica follows the equation-based modeling paradigm and enables acausal modeling, Modelica support inheritance-concepts [1].

6. Simulation Results

Simulation result of Secondary Superheater is shown as below. Fig.3 shows the simulation result with simulink and Fig.4 shows the simulation result with OpenModelica, Both the graphs gives the same results. T_s (Main steam temperature) and T_m (SSH inlet temperature) are plotted in the graph. Same graphs are obtained with both Simulink and Openmodelica. Similarly all the subsystem models such as furnace, drum, primary superheater, attemperator were simulated and same graphs and results are obtained with OpenModelica when compared to Simulink. With OpenModelica we can

reuse all the models since it is using the Object Oriented concepts. Thus the language is ideally suited as an architectural description language for complex physical systems, and to some extent for software systems.

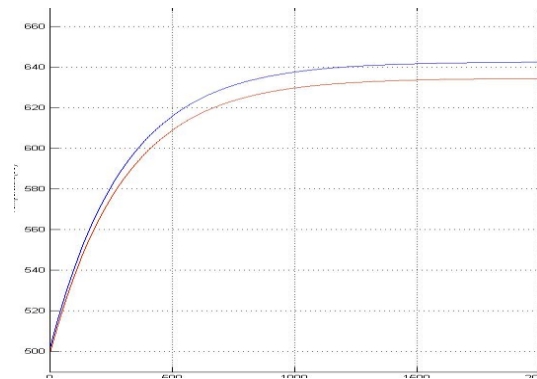


Fig.3 Simulink Result

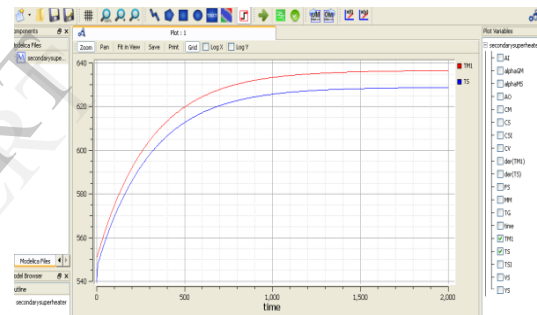


Fig.4 OpenModelica Result

7. Conclusion

The work mainly deals with Modeling and Simulation of a 210MW power plant boiler using OpenModelica. From modelling point of view, the boiler is divided into five subsystems and for each subsystem, mathematical models are developed based on the modeling principles and the knowledge gained during the boiler system study. Individually each subsystem models are simulated and corresponding graphs are obtained. More than 35 equations and more than 75 variables are used for modeling. An integrated model is also developed using the inheritance concepts of OpenModelica and simulated. The graphs obtained are compared with graphs obtained by simulink. The results were very similar.

8. References

- [1] Peter Fritzson. "Principles of Object-Oriented Modeling and Simulation with Modelica 2.1" Wiley-IEEE Press, 2003.
- [2] Modelica Association. "Modelica – A Unified Object-Oriented Language for Physical Systems Modeling - Language Specification" Version 2.1 Jan. 2004.
- [3] Peter Fritzson, Peter Aronsson, Adrian Pop, "OpenModelica- A free open source environment for system modeling, simulation, and teaching" proceedings of the 2006 IEEE conference on computer aided control systems design
- [4] Modelica Association. "The Modelica Language Specification" May 2009. www.modelica.org
- [5] Susana Pelayo Díaz. "Modeling and Simulation of an Industrial Steam Boiler with ECOSIMPRO." 1st Meeting of EcosimPro Users, UNED, Madrid, 3-4 May 2001
- [6] Jaideep Panjeta, H.M. Rai, Naresh Kumar, Nancy Arora, "Process control of Power Plant Superheater- A Matlab/Simulink Approach", 2012 Second International Conference on Advanced Computing & Communication Technologies.
- [7] Sven Erik Mattsson, Karl J. Astrom, Rodney D. Bell, "A Natural Approach to Modeling Physical Systems", Proceedings of the 35th Conference on Decision and Control Kobe, Japan December 1996.
- [8] Dr. S. Rominus Valsalam, Anish S, B. Raja Singh, "Boiler Modelling and Optimal Control of Steam Temperature in Power Plants" 6th IFAC Symposium on Power Plants and Power Systems Control (2009).
- [9] Damien Flynn, (2003) Thermal Power Plant Simulation and Control, The Institution of Electrical Engineers, London, UK.