

# Testing Power Plant Control Systems in Modelica

Kilian Link<sup>a</sup>, Leo Gall<sup>b</sup>

Julien Bonifay<sup>a</sup>, Matthias Buggert<sup>a</sup>

<sup>a</sup>Siemens AG, Energy Sector, Erlangen, Germany

<sup>b</sup>Bausch-Gall GmbH, Munich, Germany

## Abstract

This paper describes the status and findings of the development of a Modelica based test system for power plant controls. The development is part of the ITEA2 project MODRIO which has the main goal to improve the operation of plants by utilizing modeling, simulation and optimization techniques.

The control test application shown in this paper demonstrates the usability of Modelica to run comprehensive tests for plant controllers involving large parts of the physical plant and the control system. However it highlights the need for further development, in areas such as: the test system, the discrete part of the models, the Modelica language, Modelica Association standards such as FMI and ModelicaXML and the tool support for these.

*Keywords: SCADA; DCS; ModelicaXML; MODRIO; Controller Test*

## 1 Introduction

Flexibility in operation of power plants becomes more and more important. This trend started in the nineties of the last century due to the opening of the energy markets world wide. The growing part of renewable energy production and their positive discrimination to be dispatched accelerated the evolution of the remaining fossil power plants. Therefore demand for control quality of these plants increased.

Power plants are controlled by distributed control systems (DCS) [6]. This article focuses on the Siemens DCS SPPA-T3000, see [3] for more details. However, most of the results are transferable to all up to date DCSs.

The goal of the development presented here is to allow the easy testing of controllers for power plants developed in a DCS such as SPPA-T3000 at every

control engineer's desk. Therefore the effort to run these tests needs to be acceptable and iterative improvement of the controls needs to be supported. Having such a tool available offers some great benefits:

- Tests on the real plant shall be avoided, since these are:
  - Expensive, due to cost of fuel and loss of profit.
  - Dangerous, since they might cause damages.
  - Limited with respect to the number of tests and the allowed operating range.
- Speeds up the development of improved controllers.
- Will improve the quality and enable systematic quality assurance (QA).

During the requirements engineering phase it became apparent that users, i.e. the control engineers, consider using a test system for very different tasks and in all different phases of a project. This means that the system under development needs to cover different aspects such as early test of concepts on the one hand and regression testing for maintenance of existing control standards on the other. Apparently, this needs high flexibility on the modeling and simulation for the plant model in order to cover all different kinds of physical effects of the plant in an appropriate level of detail.

An agile software development approach was chosen for this project, because the control engineers, modeling experts and software developers had the chance to team up for this development. Having a working prototype early on, turned out to be very useful.

Everything afore mentioned, lead in the direction of a fully Modelica based approach, at least for the fast development of a prototype. Due to the urgent need for control testing in a running project this fully Modelica based prototype has than be applied to a

real life example which highlighted some important advantages of the Modelica solution:

1. No special support of the DCS (besides already existing export features, see section 3).
2. Comprehensive insight in the control and plant model.
3. No real-time requirement.
4. No need to develop interfaces, e.g. to a virtualized control system such as SPPA-S3000 [2].
5. Only a Modelica tool needed for deployment.

The major disadvantage is that this approach introduces a new source for bugs: The Modelica model of the control system (see section 3). Compared to the advantages mentioned above this seems to be acceptable.

## 2 Background

### 2.1 SPPA-T3000

SPPA-T3000 is a Siemens DCS system of the latest generation. Like most of its competitors the communication back bone is the IEC Fieldbus standard. It offers integrated functionality for plant operation (including HMI), Alarming, Data Archives, Engineering of the controls, Diagnostics for optimized plant maintenance and interfaces for field device communication [3]. It is notable that the online runtime system of SPPA-T3000 still consists of programmable logic controllers (PLCs), i.e. the SIMATIC S7. But even so the power plant control needs to be run on a few of these PLCs for performance reasons, the DCS system still provides a global namespace for all signals.

The global namespace for all signals introduces a motivation for systematic naming. In power plants the KKS (Power Plant Classification System) system is still commonly used, although newer standards like RDP-PP based on EN 81346 are available. Typically the plant model is more general and therefore less strictly named. Disregarding which of the mentioned naming standards are used for the control system, the interface to the plant model needs to be taken into consideration for the design of a control test system for power plants.

### 2.2 SiemensPower

The Modelica library SiemensPower is used for dynamic simulations throughout the entire lifetime of a power plant: from conception to decommissioning.

The primary target is to provide information on dynamic behavior of a plant to verify the feasibility of process concepts and control strategies.

Validation of operating and safety procedures as well as tests of control concepts before implementation on real plants is performed, using SiemensPower models.

Key features of SiemensPower are:

- Wide range of component models especially for combined cycle power plants.
- Control package to implement real plant control in simulation models that allows parsing of controls from SPPA-T3000.
- High performance TTSE (Tabular Taylor Series Expansion) water/steam function.

In the context of this paper, all equations describing the dynamics of the plant and of controls are implemented in SiemensPower.

## 3 Bringing the Control System into Modelica

This section describes how functional block diagrams are automatically transferred from SPPA-T3000 to a controller model in Modelica which can be simulated together with the plant model. Figure 1 shows the two steps of exporting and converting functional block diagrams.

First, the control engineer has to define a set of functional block diagrams which are to be tested. This subset of the full-plant control system is then exported from SPPA-T3000 into an XML file describing the diagram layout, connections and parameter settings.

Second, the XML output is translated into a Modelica package containing one model for each functional block diagram and a top-level class containing all open controller inputs. Therefore a customized T3000-Modelica parser has been developed. During the conversion, the graphical layout of block diagrams is preserved as much as possible in order to allow analyzing and changing the control layout in the same style as in SPPA-T3000. But there are limitations for blocks with conditional inputs. For example, T3000 icons are automatically adapted in height, according to the number of activated input connectors. Unfortunately in current Modelica, the means of adapting icons and connectors according to parameter settings are very limited.

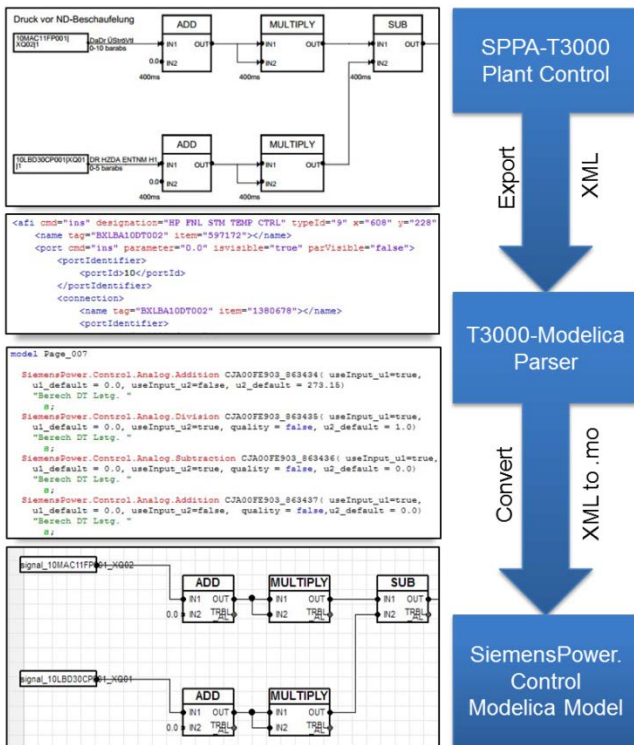


Figure 1: Process from SPPA-T3000 XML to Modelica

The parsed controller package contains the layout, hierarchy and parameterization of control blocks as Modelica instances, but lacks the actual equations. All relevant blocks have been re-implemented in Modelica, similar to the approach of [1]. There is an option to use purely discrete equations or continuous equations as much as possible. Re-implementing all controller equations and integrating them into the component library SiemensPower was a considerable effort. This work has already been initiated for earlier projects, as there was a need for real-world control blocks in dynamic simulation studies. The validation work of all control blocks is still ongoing.

The global name space of SPPA-T3000 (see section 2.1) is replicated by the connection to an outer expandable connector (bus). Therefore, the parsed functional block diagrams do not have input/output connectors on the top level, instead they are connected via bus signals in KKS naming scheme.

Due to extensive bus usage, the T3000-Modelica parser can avoid generating a lot of connect equations through the model hierarchy. Furthermore changes in the control layout are facilitated as all signals can be retrieved from bus at any level, just as in the authoring tool SPPA-T3000. Due to the high number of bus instances in the model, compiling was not possible. The reasons were memory and performance limitations during translation. This has been solved by using the inner/outer concept of Modelica.

## 4 Plant Model

The overall combined cycle power plant and its district heating system has been modeled based on the in-house Modelica library SiemensPower.

### 4.1 Description of Plant Model

For this prototype the control of a district heating system of a combined cycle plant (CCPP) was investigated. The district heating system uses steam from steam turbine extractions in order to heat the water that will be supplied for district heating. The main function of the controller is to control the supply temperature of the water by means of several valves in the plant. Figure 2 shows a simplified schema of the model.

The model of the CCPP serves as dynamic and consistent boundary conditions for the district heating system. It was therefore reduced to a minimal level of detail. The gas turbine is a steady state model based on tables, the steam turbine sections are modelled using Stodola’s equation and a constant isentropic efficiency and the heat recovery steam generator is a lumped time delay model as proposed by VDI 3508.

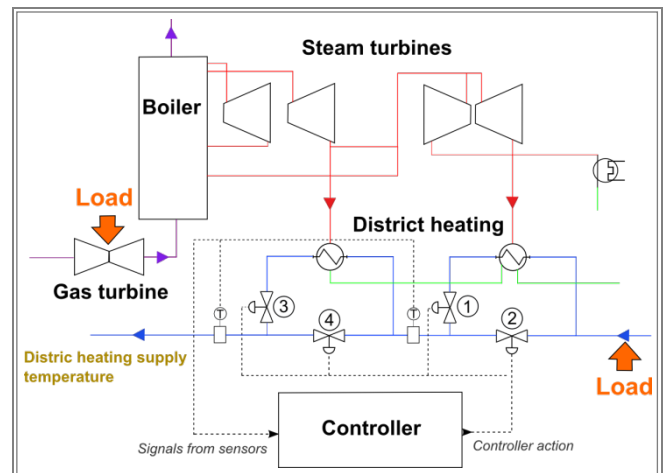


Figure 2: Simplified schema of the overall plant model with its controller

### 4.2 Components of District Heating System

A district heating system has long piping connections which can be longer than 100 meters, thus leading to large delays in the transport of enthalpy. To model this accurately while avoiding a huge number of elements in the discretized pipes, transmission line pipe models with spatial distribution for the enthalpy are used.

In the transmission line the enthalpy is delayed with a time  $\tau$ , solution of the following equation:

$$Length\ of\ pipe = \int_{t-\tau}^t velocity(t) dt$$

The controlled butterfly valves are modelled according to valve pressure loss correlation using a zeta (pressure loss coefficient) characteristic curve as function of opening angle:

$$mass\ flow = \frac{cross\ sectional\ area \times \sqrt{2 \times pressure\ difference \times density}}{zeta(angle)} massflow = crosssectionalarea \times \frac{\sqrt{2 \times pressuredifference \times density}}{zeta(angle)} massflow = crosssectionalarea \times \sqrt{\frac{2 \times pressuredifference \times density}{zeta(angle)}}$$

To have realistic temperature transients, the heat exchange surfaces and especially the condensing heater need also to be correctly modeled. Hence, specific heat correlations for one and two phase flows are used [7].

Other important parts of the plant for a reliable controller test are the sensors. Due to the delay of enthalpy transport in the connection pipe, the exact position of the sensors is crucial. Further time delay within the sensors themselves must be also considered.

### 4.3 Size of Model

To improve the performance of the plant model an effort has been made to reduce the size of the non linear systems of equations at its maximum. This has been principally achieved by increasing the amount of continuous states, where appropriate. No system with higher amount of unknowns than 1 are left after manipulation of the non linear systems by the Modelica simulation tool (Dymola).

The plant model has around 464 continuous states and 4739 time-varying variables.

After adding the controller, the model keeps the same amount of continuous states as all controller variables are discrete and has in total 17095 time-varying variables.

## 5 Connecting Plant Model and Control System

A Test Unit is a Modelica model which instantiates one version of the plant model as well as the controller model (System under Test).

Figure 3 shows the current approach for signal handling. Besides connecting controller and plant model, the Test Unit is used for adding boundary conditions. Test Cases are then created by extending the Test Unit and adding disturbances and changing parameters. A benefit of storing all Test Cases in the Modelica package hierarchy is having one single source of information during simulation and testing.

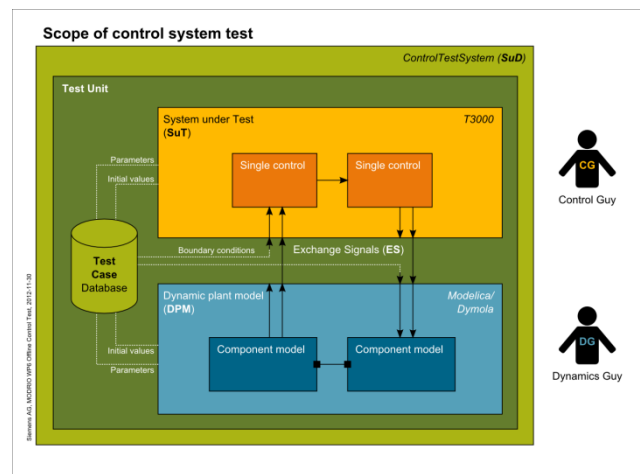


Figure 3: Test Unit with plant model and controller

The tested functional block diagrams usually depend on hundreds of Boolean and Real signals from the power plant control system. Therefore, setting appropriate boundary conditions is only possible in cooperation between control developers and dynamic simulation engineers. When connecting the models, additional unit conversions are necessary. The controllers work with non-SI units, which are adapted to different customer requests around the world. In order to avoid errors from manually converting units, an automated conversion of units is planned. The support through tools and the Modelica language for handling of non-SI units and factors could be improved (e.g. %).

Due to the chosen package structure, the controller can be updated (re-parsed) without requiring changes to the Test Unit, as long as no new boundary conditions are required. This enables efficient testing of various controller revisions.

## 6 Simulation

### 6.1 Performance Issues

The performance of plant model in open loop is very acceptable with a computation time considerably shorter than real time. Also, the computation of the discrete control signals at each one of its cycle time needs only an insignificant amount of computing resources. However, for the complete Test Unit, the time events generated at each cycle time of the control as well as the way how these time events are handled by the DAE solver dramatically deteriorate the performance of the plant model.

If it is not possible to avoid the generation of events at each control cycle time, one solution to improve the performance is by simply increasing the cycle time in order to reduce the amount of events. The cycle time in the DCS can typically go down to 100 ms. A cycle time of 400 ms can be chosen for the models, which greatly improves the performance. However, small instabilities in control actions can be noticed with this higher cycle time.

Another way to improve simulation performance is to change the DAE solver settings and the way it handles events. The Dassl solver starts for example its iteration after an event with the minimal step size, thus worsening the performance. While this can be modified for the Dassl solver in Dymola by manipulating the dsin.txt file, there is unfortunately a general lack of possibilities to influence solver settings in a user-friendly way.

An alternative attempt was to use FMI for Co-Simulation by packaging the controller in a FMU. This would offer the possibility to use two different solvers, a fixed step size solver (e.g. Euler) for the controller and a variable step size solver (e.g. Dassl) for the plant model. Due to the large size of the control model in terms of block instances (23000 in total) and amount of parameters, the simulation of the FMU in Dymola was unfortunately unsuccessful.

### 6.2 Results and Validation

A first validation test has been performed comparing the model behavior with measurement data from a real plant.

Boundaries of the model are set according to the measurements. As the model is covering the overall plant only few boundaries are needed. The necessary boundaries are the gas turbine load, the district heating load, and the set point for the district heating supply temperature.

The validation shows that the behavior of the model fits very well to the plant behavior. For instance, this can be observed for the valve openings on Figure 4.

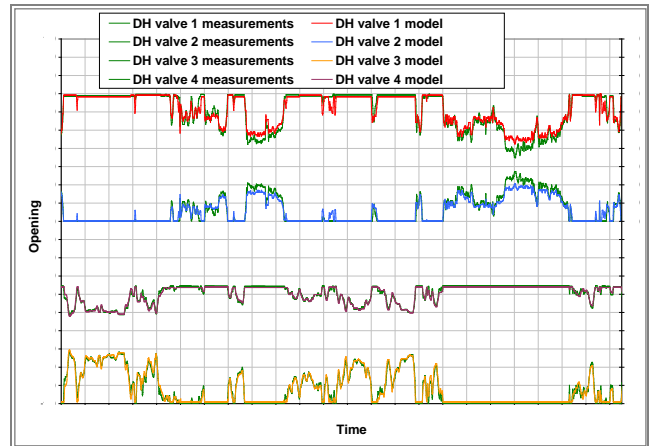


Figure 4: Comparison of real plant and model for the valve openings

## 7 Conclusion

This paper shows that the simulation based test of control systems in Modelica is possible for industrial use cases.

The self-contained simulation setup revealed interesting benefits. First of all, absolutely no changes to the authoring tool (SPPA-T3000) were necessary. Secondly, the tested output file of SPPA-T3000 could be the same as used on the real plant.

Even if the Functional Mock-Up Interface (FMI) is “en vogue” in industrial applications, the tool dependencies are much lower with our self-contained simulation model in Modelica. We can avoid a complex co-simulation setup for testing and are open to fast changes on the plant model and controller as well as Test Cases. All parts of the model retain their hierarchical structure and can be investigated deeply during testing. This helps control engineers not only to work with but also to trust in the simulation based testing procedure.

On the downside, we are reaching the limits of what current Modelica tools can handle in terms of number of signals and simulation performance. And of course there is substantial maintenance effort in regards to SiemensPower library and the T3000-Modelica parser in order to provide a stable control test system for in-house users.



## 8 Outlook

Encouraged by the success of the existing prototype the development of the control test system will be continued. Naturally the prototype needs to be improved with respect to many aspects and the most severe shall be mentioned here.

First of all, the controller part is a fully time discrete model which is modeled using existing Modelica elements such as “when” clauses and “pre()”. Besides the sampling rate or cycle time being defined for the global scope, i.e. contrary to the real DCS all controllers run on the same cycle time. This simplification did not limit the capabilities of the prototype for the particular case. However, it is a strong limitation in general. Note, the limitation is not on the export of SPPA-T3000 or the parsing to Modelica, but on the modeling of the functional controller blocks. It is evident that Modelica’s new synchronous features [4] will help to improve the situation. Both, on the modeling part utilizing clocks and sampled systems as well as on the simulation part guaranteeing deterministic behavior.

Another missing part to allow model based control development is the backwards parsing to SPPA-T3000. This step is needed to close the cycle in case modifications to the controller have been implemented during testing in Modelica. Again, the development of this feature might be backed up by a recent Modelica Association development, namely ModelicaXML. If this is standardized and supported by Modelica tools it might be a big step in the right direction. Then, the remaining task to transform one standardized XML format into another standardized XML format will be of very limited effort.

## Acknowledgements

The German Ministry BMBF has partially funded this work (BMBF funding code: 01IS12022A) within the ITEA2 project MODRIO [5].

## References

- [1] Marco Bonvini, Alberto Leva (2012). [A Modelica Library for Industrial Control Systems](#), Proceedings of the 9th International MODELICA Conference, September 3-5, 2012, Munich, Germany
- [2] Siemens AG, SPPA-S3000 Simulator, <http://www.energy.siemens.com/hq/en/automation/power-generation/simulation/sppa-s3000.htm>
- [3] Siemens AG, SPPA-T3000 Control System, <http://www.energy.siemens.com/hq/en/automation/power-generation/sppa-t3000.htm>
- [4] Hilding Elmqvist, Martin Otter, Sven Erik Mattsson (2012). [Fundamentals of Synchronous Control in Modelica](#), Proceedings of the 9th International Modelica Conference, September 3-5, 2012, Munich, Germany
- [5] MODRIO - Model Driven Physical Systems Operation, <http://www.modrio.org>
- [6] Wikipedia, The Free Encyclopedia., Distributed control system, 19 January 2014, [http://en.wikipedia.org/wiki/Distributed\\_control\\_system](http://en.wikipedia.org/wiki/Distributed_control_system)
- [7] Verein Deutscher Ingenieure / VDI-Gesellschaft Verfahrenstechnik und Chemieingenieurwesen (GVC), VDI-Wärmeatlas, Edition 8, 1997