Optimized Control of Hot-Gas Cycle for Solar Thermal Power Plants

Jan Gall Dirk Abel
IRT – Institut für Regelungstechnik, RWTH Aachen, Steinbachstr. 54, D-52074 Aachen, Germany
j.gall@irt.rwth-aachen.de
Nils Ahlbrink Robert Pitz-Paal
DLR – Deutsches Zentrum für Luft- und Raumfahrt, Linder Höhe, D-51147 Köln, Germany
Joel Andersson Moritz Diehl
Electrical Engineering Department (ESAT-SCD), KU Leuven, B-3000 Leuven, Belgium
Cristiano Teixeira Boura Mark Schmitz Bernhard Hoffschmidt
SIJ – Solar-Institut Jülich, FH Aachen, Heinrich-Mussmann-Str.5, 52428 Jülich, Germany

Abstract

In this paper, the overall modeling approach for an optimized control of a hot-gas cycle with its different components for solar thermal power plants is pointed out.
For control purposes a linear model-based controller (MPC) was implemented in Modelica based on an external state-of-the-art QP solver linked to the Modelica model.

Keywords: solar energy, control, optimization

1 Introduction

1.1 Background

One possible answer to address climate change is using solar instead of fossil energy. Among other technologies central receiver systems (CRS) using air as heat transfer medium are being investigated. A demonstration plant (STJ) has just been completed.
The STJ uses 18000 m² of sun-tracking mirrors (heliostats) to heat up air to 700 °C which in turn generates superheated steam, driving turbine and generator. A storage system can take up the thermal energy for one full-load hour. By adjusting the rate of the volume flow of two blowers, it is possible to charge or discharge the storage during operation. The Virtual Institute of Central Receiver Power Plants (vICERP) has been founded to solve the demanding requirements for the optimal plant control under the strongly fluctuating energy input.

1.2 Scope of Paper

In this paper, the overall modeling approach for an optimized control of a hot-gas cycle for solar thermal power plants is pointed out. A detailed description of the modeling of the receiver and the heliostat field can be found in an affiliated conference paper by Ahlbrink et al. [1]. The emphasis of the modeling work lays on the development of dynamic component models to be used in control systems. Depending on the control task, the discretization has to be adapted. Main components of the hot-gas cycle are the solar thermal receiver and the storage system. The steam cycle is preliminarily only included as heat sink.

2 Modeling

The modeling efforts are shared among the vICERP partner institutions. Therefore, it is crucial to use a common model setup to ensure a proper use of the models. A common test platform provides the necessary interfaces, so that new, improved modules can easily be integrated and tested. The models are based on the open source library Modelica_Fluid [2]. The vICERP library uses a finite volume approach with staggered grid method implemented with flow and volumes elements [3]. The mass and energy balances are considered in the volume element. A formulation of the balance equations from Hirsch [4] is implemented using pressure and specific enthalpy as state variables. The momentum equation is reduced to a pressure drop equation and formulated in a flow element. Models like the receiver, storage system, steam generator are setup in a way that the models end with a flow element. Thus, to the outside they

behave like a pressure drop element. Volume models are needed to interconnect the components.
Figure 1 shows the top-level of the model developed in Dymola/Modelica. Several different components can be identified in the figure: the heliostat field and receiver on the top left, the storage in the middle, a simple model of the water steam cycle on the right and the two blowers on the bottom right. The following sections give a brief introduction to the models.

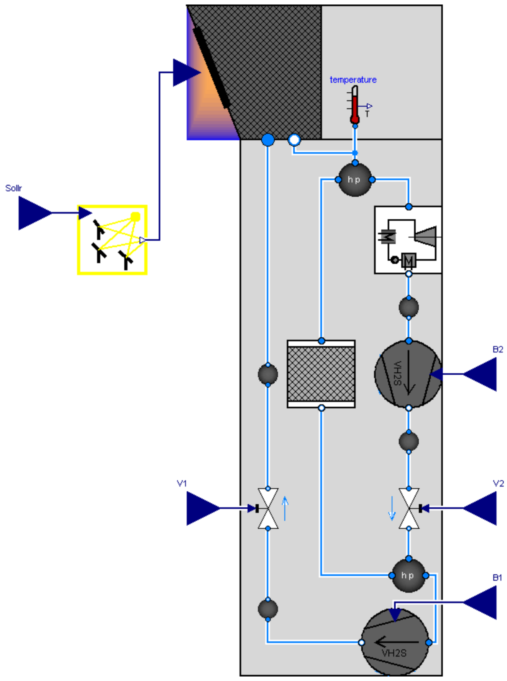

Figure 1. Screenshot of the model in Dymola

2.1 Heliostat field and Receiver

The 2200 heliostats that focus the sunlight onto the receiver are calculated by a special Monte-Carlo ray-tracing code, called STRAL [1], which generates a flux map on a surface which coincides with the receiver. The receiver is modeled in Modelica. The output is an averaged temperature for the air mass flow entering the hot-gas system.

2.2 Hot-Air Pipes and Blowers

The models for hot-air pipes are simplified using one volume and one flow element for each pipe. The blower models include the characteristic curve of the blower provided by the manufacturer. Implemented in a lookup table, this map allows the calculation of a resulting mass flow given the power input and pressure difference between inlet and outlet port of the blower.

2.3 Storage

A thermal storage system is used as a buffer that stores energy at times of high irradiances and enables operation of the plant after sunset or during periods of reduced solar input. The developed storage model enables the analysis of different operation conditions of the power plant. The storage behavior is similar to that of a regenerator. The hot air flows through the storage material and heats it up. During discharge, the air flows in reverse direction and cools down the storage material, while being heated up.
The storage model is divided into storage cells. Each cell element describes the characteristic material and flow phenomena, which are included in differential equations. Thus, each cell element computes two temperatures which represent the local temperature of the storage material and the local temperature of the fluid.
The model enables the description of charging, discharging and stand-by operation. Additionally heat losses during stand-by periods are calculated. Thus, temperature profiles inside the storage can be computed for any time in the simulation process.
Figure 2 shows the temperature profile for the 100%- and 0%-storage capacity load situation.

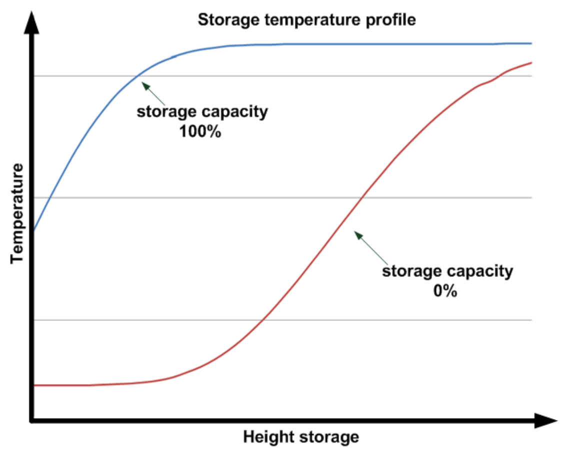

Figure 2. Temperature profile of the storage system for 100%- and 0%- storage capacity

2.4 Heat Sink

Whereas in the final system the steam cycle will be modeled in detail, it is – at this stage – merely integrated as a heat sink, featuring qualitatively the steam cycle's anticipated behavior.

3 Control

3.1 Basic automation scheme

The simulation of the operational behavior of the complete plant requires an integrated control scheme within the model to ensure compliance with given limits of absolute and gradient values. As a first concept, a basic automation scheme has been developed based on a wiring of SISO control loops with PID controllers. This scheme should on one hand assure a safe operation of the plant under normal operation conditions and on the other hand be a measure of performance for more sophisticated control schemes. The tuning of the different controllers has been done in MATLAB using a response optimization technique. An extract of the scheme is depicted in Figure 3.

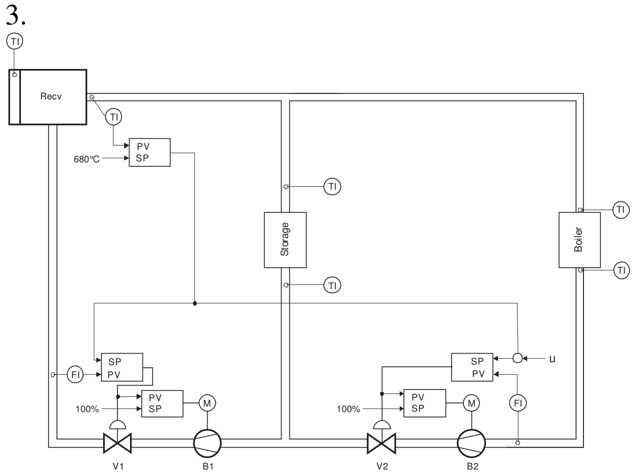

Figure 3. Basic automation scheme

The measurement signals for the control scheme are different volume flows and temperature information. Actuating variables are the speed of the two blowers and different valves located in the air cycle. The main goal of the control scheme is to maintain the outlet air temperature of the receiver constant at 680 °C. This is achieved by controlling the air volume flow through the receiver. As a consequence of an increasing volume flow through the receiver, the temperature of the outgoing air decreases. The temperature difference from the design point is used in an outer control loop of a cascaded structure, which feeds the required volume flow as setpoint to the inner control loop. The inner loop accesses two actuators for adjusting the volume flow, a blower and a valve mounted directly after the blower. The blower is obviously necessary to generate the air flow. The use of the valve is justified for two reasons. First, the blower itself has a low pressure drop during standstill periods such that an airstream just flows through it if the stream is generated by the other blower installed in series. Second, the blower is limited to a minimal rotational speed. Therefore, the valve is closed appropriately to set volume flows below the threshold given by the blower itself.

3.2 Model predictive control

The vICERP project includes the application of a *model predictive controller (MPC)*. This makes use of the dynamic model of the plant that has been developed for the simulation to predict future behavior of the plant with regard to changes in actuating variables.

With an MPC approach, it is also possible to include a natural objective function (maximize produced energy, minimize risk of boiler shutdown during transients, minimize time to start-up etc.) as well as imposing first-principle constraints such as bounds on variables or periodicity constraints.

The scope of this paper is limited to a *linear* model predictive controller, which aims at demonstrating the concept as well as the basic software coupling.

It is the goal of the project to make extensive use of the full *nonlinear* model of the plant to find an optimal controller that works in nominal operation as well as during start-up, shut-down and during sudden changes in the weather conditions (if these changes can be predicted, this prediction should be taken into account). This will be described in more detail in the outlook of this paper.

3.3 Linearized model

The MPC controller is based on a linear state space model of the form:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{1}$$

This model was obtained from the non-linear Modelica model by using the *linearizeModel* command.

3.4 MPC implementation

Based on the above representation the controller is able to predict the future behavior of the plant regarding to a future trajectory for the input (and possible disturbances). This can be expressed in an equation of the form

$$\begin{aligned} &Y(k) = \\ &\Psi \cdot x(k) + \Upsilon \cdot u(k-1) + \Theta \cdot \Delta U(k) + \Xi \cdot D_m(k) \end{aligned} \tag{2}$$

for suitable matrices Ψ, Υ, Θ and Ξ [9]. The different terms represent the free and forced response of the plant, together with the response to future trajectories of the inputs and disturbances. Combined with a given reference trajectory for the outputs and additional linear constraints on the states and inputs, this can be reformulated as an optimization problem of the form

$$\min_{\Delta U(k)} \Delta U(k)^T \cdot H \cdot \Delta U(k) - G^T \cdot \Delta U(k) \tag{3}$$

with Hessian matrix H and gradient vector G. This is a standard optimization problem known as the *Quadratic Programming (QP)* problem.

3.5 QP Solver

The MPC controller requires the above quadratic program to be solved at each sampling time. This is carried out with the QP solver qpOASES [5], which uses an online active-set method particularly suited for MPC problems [6].
To make qpOASES, which is written in C++, fully compatible with Modelica, a C interface has been written. By using Modelica's *external objects*, the QP solver is able to retain memory between calls. This fact can be expected to grow importance once the MPC controller is extended to nonlinear models.
The Modelica interface to qpOASES is available upon request from the authors of the paper (LGPL license).

4 Simulation Results

For evaluation of the model and different control schemes the simulation results according to the following scenario are presented in Figure 4.
The plant is operating in its stationary design point (i. e. outlet temperature at the receiver is at 680 °C) with a constant solar irradiation. At time *t = 100 s*, a sinusoidal disturbance with a period of 600 s and an amplitude of 50% of the previous irradiation acts on the input. After 1200 seconds the input remains constant again.
The upper part of the figure shows this disturbance of the solar irradiation. In the lower part the responses of the air outlet temperature at the receiver with different controllers are depicted. The main goal of this feedback control is disturbance rejection, i. e. it should assure a constant outlet temperature by adjusting the air volume flow through the receiver appropriately.

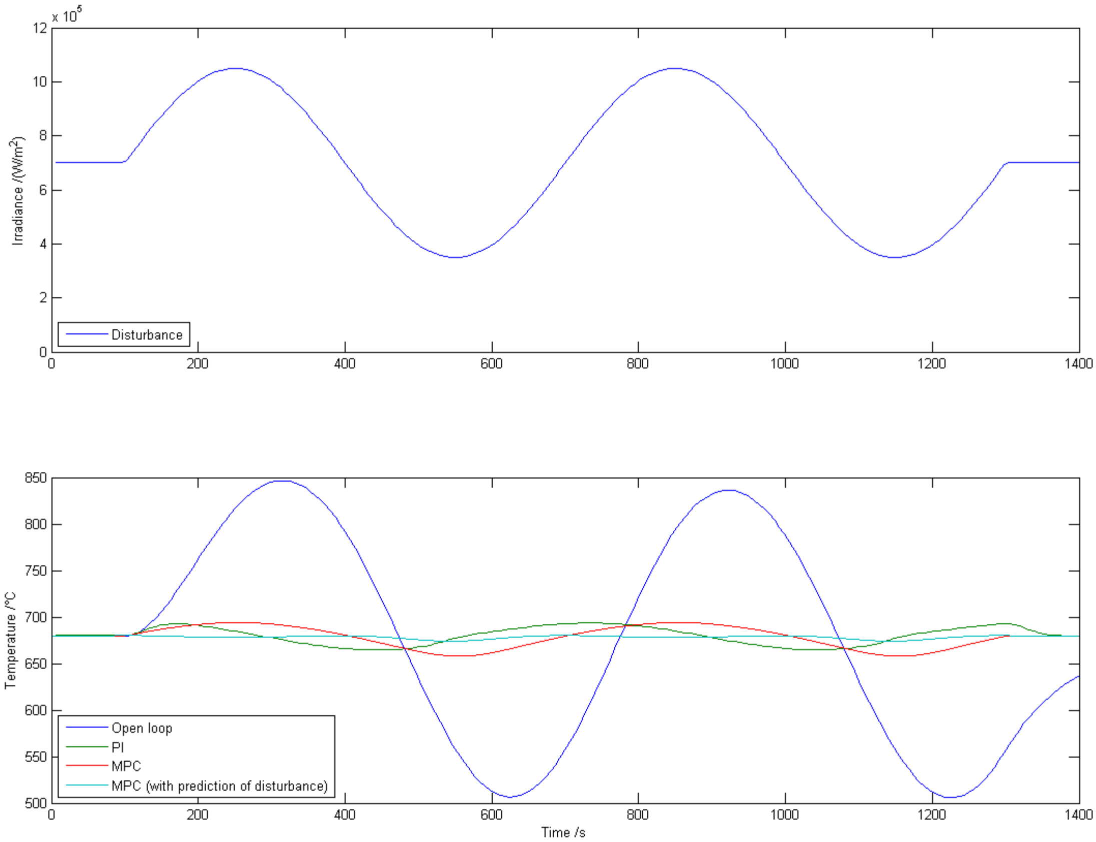

Figure 4. Simulation results

The first case is just an open loop simulation of the system, i. e. no control actions are performed at all and all manipulated variables, especially the setpoints of the blowers, remain constant.
The second curve shows the resulting characteristics if the control scheme as depicted in Figure 3 based on PI controllers is used. In this case the maximal deviation of the outlet temperature is about 15 °C.
In the following two cases the implemented MPC-block was used to control the air temperature. Therefore the PI controller in Figure 3 which uses the air temperature as measurement variable in the outer control loop was simply replaced by the MPC-block. The inner controllers directly manipulating the actors remain the same. Although the model-based controller has inherently the ability to cope with systems with multiple in- and outputs, it this case it is just used with a single in- and output. The controller has a sampling interval of 0.5 s and uses an internal model with 25 states at a discretization interval of 2 seconds. It has a prediction horizon of $n_p = 150$ and a control horizon of $n_u = 30$ (i. e. it predicts the response of the plant 300 seconds in the future). As one can see in Figure 4, the controller shows comparable results to the PI controller.
In the fourth case, the MPC-block was extended to also incorporate the influence of the disturbance on the system by feedforward control. If the supplied solar energy can not only be measured, but also predicted (e. g. by weather forecast or vision-based [8]), it is possible that the MPC also uses this information for prediction. In this case the controller achieves the best performance with only minimal deviation from the setpoint at 680 °C.

5 Conclusion

In this paper we have presented a first-principle model for a central receiver solar power plant with open volumetric receiver. The model includes the different components of the plant, e. g. receiver, storage, and is used for simulation and optimization purposes of both the separate components and also the plant behavior as a whole.
For control purposes a generic linear model-based controller (MPC) was implemented and achieves reasonable results. The implementation is based on an external state-of-the-art QP solver linked to the Modelica model for the calculation of optimal control actions.
Future work aims at not only using optimal control for the air cycle as presented in this paper, but also to extend this approach to other areas of the plant, e. g. storage regulation.

6 Outlook

6.1 Non-linear MPC

A non-linear MPC controller is obtained if the linear state space model (1) is a replaced with a continuous state space model of the form:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), p, t) \\ y(t) &= g(x(t), u(t), p, t) \end{aligned} \tag{4}$$

The dynamics are now described by a non-linear ordinary differential equation (ODE) and the discrete time k has been replaced by the continuous time t.
Included is also the dependence of a set of parameters p.
A model of the form (2) is always available since simulating a translated Modelica model is always equivalent to integrating a (possibly hybrid) *differential-algebraic equation*, due to construction of the Modelica language [10].
By adding a quadratic objective function and introducing the prediction horizon T_p and control horizon T_c, we obtain an *optimal control problem in differential-algebraic-equations* of the form:

$$\begin{aligned} \underset{x(\cdot),u(\cdot),y(\cdot),p}{\text{minimize}} \quad & \int_0^{T_p} \left\| y - y_{\mathrm{ref}} \right\|_Q^2 dt + \int_0^{T_c} \left\| u - u_{\mathrm{ref}} \right\|_R^2 dt \\ \text{subject to} \quad & \dot{x}(t) = f(x(t), u(t), p, t) \\ & y(t) = g(x(t), u(t), p, t) \\ & y_{\mathrm{lb}} \le y \le y_{\mathrm{ub}} \quad \text{(output bounds)} \\ & u_{\mathrm{lb}} \le u \le u_{\mathrm{ub}} \quad \text{(control bounds)} \\ & p_{\mathrm{lb}} \le p \le p_{\mathrm{ub}} \quad \text{(parameter bounds)} \\ & x(0) = x_0 \quad \text{(initial value)} \end{aligned} \tag{5}$$

where P is a positive definite and Q is a positive semi-definite matrix. Bounds are denoted by the subscript "ub" and "lb" for upper and lower bounds respectively.
Problem (5) is an infinite dimensional optimization problem that can be efficiently solved by parameterization of both the control u and the state x using a *simultaneous* method such as *direct multiple shooting* and *collocation*.
State-of-the-art numerical methods for solving such and similar dynamic optimization problem have been implemented in the software package ACADOtoolkit [12], developed by OPTEC. The software is available open-source under the LGPL license, allowing it to be linked also with commercial code, and work is underway to fully integrate it with Modelica.
The integration consists of two parts. Firstly, it should be possible to call ACADOtoolkit from Mod-

elica. This is made possible by implementing a plain C interface which can be called from Modelica code using external objects just like the qpOASES interface. ACADOtoolkit is written exclusively in C/C++ and avoids linking with external software, so it is very suitable to use together with Modelica tools as well as on embedded systems. Real-time optimal control is one of the aims of the project.

The second, much larger part of the integration consists of extending the software so that it can use models formulated in Modelica. Small dynamical models can easily be coded directly in C++, but for complex models, a better solution is to import the model equations into ACADOtoolkit. This will undoubtedly require the extension of the software to deal with e.g. *hybrid systems* and *integer valued controls*.

There are efforts to standardize the interaction between equation-based, object-oriented modeling languages such as gProms and Modelica on one hand, and computer algebra tools and other mathematical software on the other [7]. The two open-source Modelica projects OpenModelica and JModelica [11] both offer the possibility to export the flattened simulation problem (i. e. variables, initial values, model equations, etc.) in the ModelicaXML format which in turn uses MathML to describe the model equations.

To make also ACADOtoolkit conformant with this standard, so that models defined in ModelicaXML code can used by the software, an XML module is being developed. This module parses the ModelicaXML code and translates the model equations into ACADOtoolkit's internal (symbolic) representation.

A further standardization is to use the Optimica, a language extension for the formulation of optimal control problems, to formulate the optimal control problems. This provides an abstraction which is useful to help engineers and scientists formulate optimal control problems in a structured way [11].

Acknowledgements

The authors would like to thank for financial support granted by the Initiative and Networking Fund of the Helmholtz Association, the state of North Rhine-Westfalia, and the European Union/European regional development fund.

References

[1] Ahlbrink N., Belhomme B., Pitz-Paal R.: *Modeling and Simulation of a Solar Thermal Power Plant with Open Volumetric Air Receiver.* Conference Proceedings, Modelica Conference 2009, Como

[2] Casella F., Otter M., Proelss K., Richter C., Tummescheid H.: *The Modelica Fluid and Media library for modeling of incompressible and compressible thermo-fluid pipe networks.* Conference Proceedings, Modelica Conference 2006, Vienna, September 4-5 2006

[3] Tummescheid H.: *Design and Implementation of Object-Oriented Model Libraries using Modelica.* Thesis, Department of Automatic Control, Lund Institute of Technology, Lund, August 2002

[4] Hirsch T.: *Dynamische Systemsimulation und Auslegung des Abscheidesystems für die solare Direktverdampfung in Parabolrinnenkollektoren.* Fortschrittsberichte VDI, Reihe 6, Nr. 535, VDI Verlag, Düsseldorf, 2005

[5] www.qpoases.org

[6] Ferreau H. J., Bock H. G., Diehl M.: *An online active set strategy to overcome the limitations of explicit MPC.* International Journal of Robust and Nonlinear Control 18:816-830 (2008)

[7] Casella F., Donida F., Lovera M.: *Beyond Simulation: Computer Aided Control System Design Using Equation-Based Object Oriented Modelling for the Next Decade.* EOOLT 2008, Paphos

[8] López-Martínez M., Vargas M., Rubio F. R.: *Vision-Based System for the Safe Operation of a Solar Power Tower Plant.* Proceedings of the 8th Ibero-American Conference on AI: Advances in Artificial Intelligence, Vol. 2527, pp. 943 – 952, 2002

[9] Maciejowski J. M.: *Predictive Control with Constraints.* Prentice Hall, 2002

[10] Fritzson P.: *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*, Wiley-IEEE Press, 2003

[11] Åkesson, J.: *Optimica – An Extension of Modelica Supporting Dynamic Optimization*, Conference Proceedings, Modelica 2008

[12] www.acadotoolkit.org