A Framework for Nonlinear Model-Predictive Control Using Object-Oriented Modeling with a Case Study in Power Plant Start-Up

Larsson, Per-Ola; Casella, Francesco; Magnusson, Fredrik; Andersson, Joel; Diehl, Moritz; Åkesson, Johan

Published in:
IEEE Conference on Computer Aided Control System Design (CACSD), 2013

DOI:
10.1109/CACSD.2013.6663487

2013

Document Version:
Peer reviewed version (aka post-print)

Link to publication

Citation for published version (APA):
Larsson, P.-O., Casella, F., Magnusson, F., Andersson, J., Diehl, M., & Åkesson, J. (2013). A Framework for Nonlinear Model-Predictive Control Using Object-Oriented Modeling with a Case Study in Power Plant Start-Up. In *IEEE Conference on Computer Aided Control System Design (CACSD), 2013* (pp. 346-351). IEEE - Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/CACSD.2013.6663487

Total number of authors:
6

A Framework for Nonlinear Model-Predictive Control Using Object-Oriented Modeling with a Case Study in Power Plant Start-Up

Per-Ola Larsson, Francesco Casella, Fredrik Magnusson, Joel Andersson, Moritz Diehl, Johan Åkesson

Abstract— **In this paper, nonlinear model predictive control (NMPC) is applied to the start-up of a combined-cycle power plant. An object-oriented first-principle model library expressed in the high-level language Modelica has been written for the plant and used to set up the simulation and optimization models. The NMPC optimization problems are both encoded, using a high-level notation, and solved in the open-source framework JModelica.org. The results demonstrate the effectiveness of the framework and its high-level description. It bridges the gap between an intuitive physical modeling format and state of the art numerical optimization algorithms. Promising closed-loop control results are shown for plant start-up when the NMPC model contains parametric errors and the simulation model, corresponding to the real plant, is subject to disturbances.**

I. INTRODUCTION

Increasing economic competition, as well as growing environmental concerns, have pushed the optimization of industrial processes during the last decades, also thanks to the ever-increasing availability of low-cost computational resources and high-performance optimization algorithms. Applications of optimization techniques range from plant-wide static set-point optimization, to the improvement of economic performance, to on-line model predictive control (MPC) to track set-points and to reject disturbances. Increasingly rigorous and complex models are employed when formulating and solving optimization problems. This trend has increased the effort required to encode models suitable for optimization, calling for more effective and user-friendly environments supporting this activity. Also, in typical situations the optimization problem is often encoded in tool-specific, proprietary description formats, which limit the portability and re-use of models for different purposes.

The framework demonstrated in this paper is based on the non-proprietary object-oriented modeling language Modelica and on the open-source platform JModelica.org [1]. This approach overcomes some of the difficulties often associated with development of models to be used for optimization purposes by *i)* relying on an high-level open language supported by several tools for physical system modeling, *ii)* using a high-level language, the Optimica extension, for the formulation of the dynamic optimization problem, and *iii)* using a computational platform allowing the same model to be simulated, optimized and analyzed in a fully integrated way, possibly using different optimization strategies and numerical solvers, with a unified, high-level front-end.

P. Larsson, F. Magnusson and J. Åkesson are with the Department of Automatic Control, Lund University, Sweden, {perola|fredrik.magnusson|jakesson}@control.lth.se.

F. Casella is with Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy, francesco.casella@polimi.it.

J. Andersson and M. Diehl are with the Electrical Engineering Department, K.U. Leuven, Belgium, {joel.andersson|Moritz.Diehl}@esat.kuleuven.be.

Modelica models have been used in the past to solve optimal control problems in the field of power plant control, see, e.g., [2], requiring extensive manual coding to interface the simulation code generated by the Modelica compiler with nonlinear optimization codes. More recently, the Modelica language and the JModelica.org platform were used in a first attempt to demonstrate the applicability of fully integrated object-oriented modeling and dynamic optimization techniques and tools to solve such problems [3]. In that case, the plant model, albeit simple enough to be handled for optimization purposes, was derived from first-principle laws and built in a modular way, according to the object-oriented modeling methodology. An open-loop dynamic optimization problem was solved, i.e., the minimum-time start-up of a combined-cycle plant under a turbine rotor stress constraint.

The main contribution of this paper is twofold:

- extend the framework presented in [3], which addressed an open-loop dynamic optimization problem, to on-line nonlinear model-predictive control (NMPC), an emerging control technique typically based on first-principle models [4], [5], [6], [7];
- demonstrate the capability of the proposed integrated modelling and optimization platform to support significant applications in the energy sector, by means of an exemplary case study.

For the sake of the present work, which is based on simulation, the state of the plant is assumed to be accessible: this is a first step towards the implementation of on-line NMPC with state estimation, that is the subject of on-going work. The results of a case study considering the same problem previously addressed in an open-loop context in [3] are presented in this paper; modeling errors and unmeasured disturbances were additionally taken into account, in order to assess the closed-loop performance in realistic use scenarios.

On-line optimal control is now widely used in the chemical and petro-chemical industry, where optimality can be crucial for competitiveness, and the very long time constants of the processes ease the design and operation of such control systems; conversely, it is seldom used in the energy conversion field, where control strategies are still mainly based on PID control. Increased competition on electrical power markets, increased penetration of renewable energy, with its inherent variability, and the adoption of innovative conversion processes is rapidly changing the scenario, making NMPC very attractive for its inherent ability of handling multivariable nonlinear processes with operational constraints and with optimality requirements in terms of efficiency, availability, low environmental impact, etc. This paper thus also aims at promoting a framework for easier and smoother adoption of on-line optimal control techniques in that context.

The paper is organized as follows: Section II gives an overview of the Modelica modeling language and the JModelica.org optimization framework used when solving the NMPC problem and Section III presents the model of the combined-cycle power plant, which is the subject of this case study. Further, Section IV gives the NMPC problem formulation of the start-up problem and Section V gives the optimization results when NMPC is used on the combined-cycle power plant model. Finally, Section VI provides a summary and future work directions.

II. MODELING AND OPTIMIZATION FRAMEWORK

A. Modelica

In this paper, Modelica, see [8], is used as the description language for the dynamic model of a combined-cycle power plant. Modelica is an equation-based, object-oriented language targeting modeling of heterogeneous physical systems. The ability to encode declarative equations, as opposed to assignment statements, provides a convenient environment for the modeler. Accordingly, there is no need to manually solve the equations for the derivatives, which is common in block-based modeling formalisms. Rather, the underlying mathematical formalism in Modelica is that of differential algebraic equations (DAEs). Modular modeling is extensively supported by the possibility of defining a-causal physical ports for elementary components, of building systems by the hierarchical aggregation of sub-systems, and of managing model variants by replacing some parts of the model by others sharing the same physical ports.

Modelica is a non-proprietary language supported by a growing number of tools, featuring a companion, freely available standard library of physical components in a wide range of domains, including thermodynamics, mechanics, electronics, thermal systems, and control. The concept of developing model libraries is advantageous in terms of knowledge re-use. For the sake of this work, a small library for optimization-oriented modeling of power plants has been built, starting from the work described in [3] and further updated for this paper.

B. JModelica.org

JModelica.org [1] is an open-source platform for simulation and optimization of Modelica models and has been used previously for energy, chemical, and automotive systems, see [3], [9], [10]. Whereas standard Modelica tools, such as Dymola [11] and OpenModelica [12], mainly focus on the simulation of physical systems, JModelica.org also targets large-scale dynamic optimization. JModelica.org is developed in collaboration between industry and academia, with the purpose of creating an industrially viable platform using state-of-the-art numerical algorithms to optimize the design and operation of complex physical systems.

The Modelica language is largely designed for simulation-based analysis. To accommodate the need for conveniently formulating dynamic optimization problems based on models described by Modelica code, the Modelica extension Optimica [13] has been developed and integrated into JModelica.org. Optimica enables the extension of a Modelica model to include the formulation of a dynamic optimization problem based on the model, such as an optimal control or parameter estimation problem. This is done by introducing objectives, variable bounds, path constraints and by specifying the optimization variables.

The main components of JModelica.org are the Modelica and Optimica compilers, which are implemented in Java using JastAdd [14], and the three modeling interfaces Functional Mock-up Interface (FMI), JModelica.org Model Interface (JMI) and a new symbolic XML-based format based on FMI, which represents the model in terms of differential-algebraic equations.

FMI [15] is a standard defining a tool-independent format for representation of hybrid dynamic models on ordinary differential equation (ODE) form. The standard is supported by most Modelica tools, among many others. In this paper, JModelica.org is used to compile the Modelica model into a Functional Mock-up Unit (FMU), thus transforming it from a DAE into an ODE. JModelica.org's interface to SUNDIALS [16] is then used to simulate the model.

JMI is a runtime library designed solely for JModelica.org, and has long been the main interface for dynamic optimization in JModelica.org. The main optimization algorithm in JMI is collocation-based and implemented in C. It relies on CppAD [17] to compute and evaluate derivatives. However, in this paper the new XML-based format is instead used for optimization purposes. This format is an extension of the XML format used in FMI and is described in [18]. The format uses a DAE representation of the model instead of an ODE representation, as in the previous two cases. It is designed to use a model representation that is as general as possible, allowing for the formulation of a wide variety of problems based on Modelica code, in particular dynamic optimization problems described by Optimica code.

The user interacts with the various components and provided functionalities of JModelica.org via the scripting language Python: this gives extreme flexibility in terms of the workflow of the final application. In the case discussed in this paper, a few lines of Python code were sufficient to implement the receding horizon MPC algorithm, starting from basic functionalities such as ”set initial state of the model”, ”solve dynamic optimization problem over a certain time interval”, and ”set initial guess for nonlinear solver”, based on the results obtained at the previous time step.

C. Solving the Optimal Control problem by Direct Collocation in CasADi

The algorithm used in this paper to solve the optimal control problems arising in the NMPC was developed in [19] and utilizes the new XML-based format. The algorithm implements a direct and local collocation method, see [20], on finite elements using Radau points and Lagrange polynomials. It is implemented in Python within the JModelica.org framwork, using the open-source optimization package CasADi [21]. Using CasADi's symbolic syntax, it is possible to transcribe the infinite dimensional optimal control problem into a finite dimensional nonlinear programming problem (NLP); the user has full control on the details of the transcription process, such as the number and length of intervals, order of the interpolation polynomials, etc. The NLP is then solved using the primal-dual interior point method IPOPT v.3.10.1 [22] using MA57 as a linear solver. The first and second derivatives required by IPOPT are automatically and efficiently generated by CasADi, using automatic differentiation (AD) techniques [23]. It has previously been shown, see [24], that this approach is not only more convenient, being implemented in a high-level language such as Python, but also considerably faster than the previous C-based implementation.

III. COMBINED-CYCLE PLANT MODEL

A. Plant Overview

The system chosen for the present case study is a combined-cycle power plant, in which the exhaust gases of a gas turbine drive a heat recovery steam generator, which feeds a steam turbine to produce additional power. The shaft of the steam turbine is often the most critical component when starting up the plant: the difference between the surface temperature, which is in contact with live steam, and the average temperature, which has slow dynamics due to the huge thermal inertia of the component, gives rise to mechanical stresses that limit the operating life-time of the turbine. The life-time consumption is a function of the peak stress level achieved during the start-up cycle. On the one hand, this means that the peak level should be controlled, in order to avoid excessive wear and tear of the component; on the other hand, once a certain stress level is deemed acceptable, it might be convenient to push the load increase rate in order keep the stress level at that level as long as possible, because this will not cause any additional fatigue on the turbine, and will allow for a faster start-up time.

Large size combined-cycle power plants have two or three levels of pressure and a dozen or more heat exchangers along the flue gas path. For the sake of the present study, a simplified and somewhat idealized plant structure has been assumed, with only one level of pressure and just three heat exchangers: one economizer, one evaporator, and one superheater. The goal is to include all the main physical phenomena taking place in a commercial power plant, while keeping the complexity of the model low and avoiding the need for much proprietary data to parameterize the model. Once solving the NMPC problem on such a proof-of-concept simplified model is proven feasible, it might be possible to move to more accurate and detailed plant models, referring to specific commercial units.

B. Modeling

The plant model used is an updated version of the model presented in [3]. Sizing and operation data were adapted from a previous study where the combined-cycle plant start-up problem was also considered [25].

The plant model is built in a modular way by connecting elemental components, as shown in the object diagram of

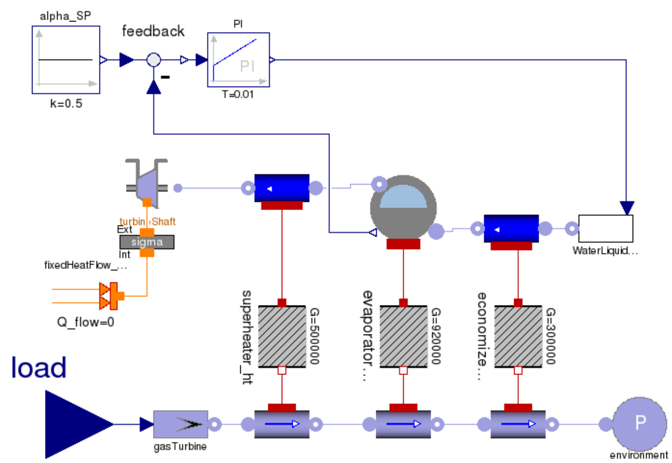

Fig. 1. Object diagram of Modelica plant model.

Figure 1. Port connections correspond to domain-specific equations; for example, the connections of thermal (red) ports state that the temperature at the connected ports are equal and that the sum of the heat flows entering the ports is zero.

At the lower left of the diagram, the gas turbine model is found. The dynamics of the gas turbine controller is much faster than the dynamics of the steam generator, therefore ideal and infinitely fast control was assumed for the model. This means that the exhaust gas mass flow and temperature are given by piecewise-linear functions of the load set-point, which is the control input for the NMPC algorithm. Until a 50% load level is achieved, the exhaust flow is constant and the temperature increases, while above 50% load, where temperature control kicks in, constant temperature and increasing exhaust flow are assumed.

The heat-recovery steam generator is described by the connection of lumped-parameter models of the steam side and flue-gas side of the heat exchangers. Each segment of the flue-gas side is modeled by a dynamic energy balance equation, while changes in mass storage and pressure losses are neglected, since they are negligible for the purposes of this system model. The same modeling assumptions hold for the water-steam side of the evaporator and superheater. In the case of the evaporator, a simplified drum boiler model is used, accounting for lumped-parameters dynamic mass and energy balances. The drum boiler is complete with a PI level controller, acting on the feed water flow rate; this controller is straightforward to design and implement and is not part of the optimal control problem, so it is included in the plant model along with the physical component descriptions.

The flue gas, liquid water and superheated steam are all modeled as constant c_p fluids for simplicity, while the properties of saturated liquid and vapour in the drum are described by suitably tuned polynomials. Since the water-steam side heat transfer coefficients are usually much larger than the flue-gas side ones, the heat exchanger wall temperature is assumed to be the same as the water temperature in all three components of the steam generator – in other words, the water-steam side components also include the thermal inertia of the heat exchanger walls, using the fluid temperature as state variable.

The heat transfer between the steam-side components (i.e., their outer tube wall surfaces) and the flue-gas side components is modeled by three convective heat transfer components, in which the heat flow is proportional to the difference between the average temperatures on the two connectors, by means of total heat conductivity coefficients which were tuned in order to get the correct outlet temperatures at full load conditions.

The steam turbine is modeled by the connection of a thermo-hydraulic part, which describes the expansion of steam to a fixed-pressure condenser with a constant isentropic efficiency, and of the thermal model of the section of the rotor shaft corresponding to the first row of blades. The latter model is described by Fourier's heat equation for a hollow cylinder, having as boundary conditions zero heat flow on the inner surface (corresponding to the hole inside the shaft), and convective heat transfer with the live steam entering the turbine on the other side. The partial differential equations have been discretized by the finite difference method, using 8 temperature nodes, 6 of which are state variables. As mentioned above, the external rotor surface stress is eventually proportional to the difference between the surface temperature and the rotor temperature averaged through its thickness.

The reader interested in further details is referred to the Modelica package containing all the models definitions, which will be included in the upcoming releases of JModelica.org.

The above-described object-oriented model is processed by the JModelica.org tool and eventually cast into a differential-algebraic equations (DAE) system:

$$0 = F(\dot{x}, x, w, u, \mathbf{p}), \tag{1}$$

where x, w and u are the state variable, algebraic variable, and control input variable vectors, respectively, and $\mathbf{p}$ is the model parameter vector. In total, the model has 14 states (three temperatures for the flue gas side, two temperatures for the economizer and superheater steam side, pressure and void fraction of the evaporator, six temperatures of the turbine shaft, and the state of the PI level controller), 152 algebraic variables, and 1 input.

C. Simulation and Optimization Models

Compared to [3], the main goal of this paper is to demonstrate the use of object-oriented modeling tools and methodologies in a closed-loop optimal control framework. In order to make the closed-loop tests more representative of real-life operating conditions, modeling errors were introduced in the model used in the NMPC and unmeasured disturbances were introduced in the model used for the simulation of the plant.

One of the main uncertainties when modeling these plants lies in the description of the whole heat transfer process. This has been accounted for in the model used in the NMPC by changing the thermal conductances of the heat exchangers between 5% and 10% of their nominal values, which are used in the process simulator closing the loop.

The main unmeasured disturbance acting on the plant is given by the fluctuations of the gas turbine exhaust flow and temperature. This has been modeled by adding a random disturbance to both variables in the simulation model. The random disturbance is obtained by generating a pseudo-random binary sequence, which is then low-pass filtered, in order to obtain band-limited white noise. The resulting RMS deviation of the mass flow is 2% of the nominal value, while the RMS deviation of the temperature is 2% of the difference between the minimum and maximum flue gas temperatures.

IV. NONLINEAR MODEL PREDICTIVE CONTROL FORMULATION OF PLANT START-UP

The start-up problem for the combined-cycle power plant is to transfer the plant from the initial state to full load. The initial state of the plant considered in this paper corresponds to the time instant directly after the electrical generator has been connected to the power grid. The initial steam turbine rotor temperature is assumed to be uniform and equal to the initial steam temperature, corresponding to a warm start-up condition. Since the plant operates in sliding pressure mode, the full load state is reached when the evaporator pressure p reaches at the reference target value p_{ref}.

Several constraints on the power plant operation are set during start-up. The evaporator pressure p is increased by increasing the load u on the plant, which is the control variable of the plant used in the NMPC. As the temperature is increased with the load, the thermal stress σ on the steam turbine rotor transiently increases, slowly settling down when the temperature increase stops. Assuming that the stress over time follows a monotonically non-decreasing curve towards its peak value and monotonically non-increasing curve after the peak value been reached, then the peak value is an indicator of rotor life-time shortage due to the overall power plant start-up cycle. Thus, upper limits on the stress σ will be set in the NMPC optimization problem, both as a soft constraint and as a hard constraint.

Apart from the stress limiting the start-up operation, constraints are also set on the rate of change of the gas turbine load, i.e., the rate of change of the control variable u. First of all, the load may not decrease. This will avoid having multiple cycling of the stress level σ during the start-up. Secondly, the load increase rate may not exceed the maximum value prescribed by the manufacturer. Additionally, a maximum load on the plant is also set.

The NMPC optimization problem to be solved in each iteration is formulated using a quadratic cost function as,

$$\begin{aligned}
&\min_{\dot{u},\epsilon} \int_{t_0}^{t_0+T_p} \left(q_p \left(p - p_{\text{ref}}\right)^2 + q_{\dot{u}} \dot{u}^2 \right) dt + q_\epsilon \epsilon^2 \\
&\text{s.t. } 0 = F(\dot{x}, x, w, u, \mathbf{p}), \quad u = \int_{t_0}^{t} \dot{u}\, d\tau \\
&\qquad \sigma \leq \sigma_{\max} \\
&\qquad \sigma - \epsilon \leq \alpha \sigma_{\max}, \ \epsilon \geq 0 \\
&\qquad 0 \leq \dot{u} \leq du_{\max} \\
&\qquad 0 \leq u \leq u_{\max} \\
&\qquad x(t_0) = x_0, \ u(t_0) = u_0,
\end{aligned} \tag{2}$$

where q_p, $q_{\dot{u}}$ and q_ϵ are design weights, t_0 the initial time for each iteration, and T_p is the prediction horizon length. The decision variables in the optimization problem are the control signal derivative $\dot{u}$ and the slack variable ϵ for the soft constraint on the stress σ.

The first quadratic term in the objective function ensures that the pressure p reaches the target p_{ref} as quickly as possible. The second term penalizes fast changes and oscillations of the gas turbine load, which could unnecessarily stress the gas turbine unit; its weight is chosen big enough to ensure that no such oscillations arise during the transient. The third term introduces a soft constraint on the stress, which kicks in when σ is closer to the limit $\sigma_{\max}$ by less than α percent. This term was added to provide some margin between the average value of σ and its upper bound $\sigma_{\max}$; this helped avoiding unfeasible NLPs, caused by the effects of the temperature disturbances injected by the gas turbine when the stress is already at its maximum level.

Extending the model F with an integrator at the input, and thus having u as a state and instead using $\dot{u}$ as a decision variable, yields a straight-forward way of introducing constraints on $\dot{u}$ compared to, e.g.,high-pass filtering of u. It also better reflects the actual requirements of the problem: there is nothing wrong with a high value of u (on the contrary, the goal is to get at full load as quickly as possible), while fast changes and repeated oscillations of the gas turbine load should be avoided, preferring smooth load changes. The initial values x_0 and u_0 are set directly from measurements and knowledge of the control signal, respectively.

V. NUMERICAL EXAMPLE OF CLOSED-LOOP PLANT START-UP

The optimization problem in Eq. (2) can directly be formulated using the Optimica extension for optimization and the Modelica model of the power plant. The prediction horizon is set to $T_p = 500$ s and the collocation scheme used has elements of length 100 s, where the states and algebraic variables are approximated using Lagrange polynomials of order three and two, respectively. The control signal derivative $\dot{u}$ is set element-wise constant and in the three last elements in the optimization interval, a constraint of $\dot{u} = 0$ is enforced. Thus, any effect on p and σ due to the last change of $\dot{u}$ will be accounted for in each NMPC optimization time interval. For each NMPC iteration, the optimized control signal derivative is applied to the simulation model, corresponding to the real plant, for 100 s, i.e., one element.

As all trajectories over the optimization horizon are solved for simultaneously when using a collocation method, good initial guesses of all variables at the collocation points are crucial in this nonlinear optimization problem. For the first NMPC iteration, the result of a simulation of the initial stationary point is used, while the next-coming NMPC iterations use the previous NMPC iteration as initial guess.

With the extension of u as a state, the optimization model has 15 state variables and 152 algebraic variables, which gives a total of 2976 variables in the NLP after discretization using the collocation scheme in JModelica.org. Each NLP optimization is solved by IPOPT in 0.4-0.8 s when using an Intel® Core™ i7-2600 CPU@3.40GHz.

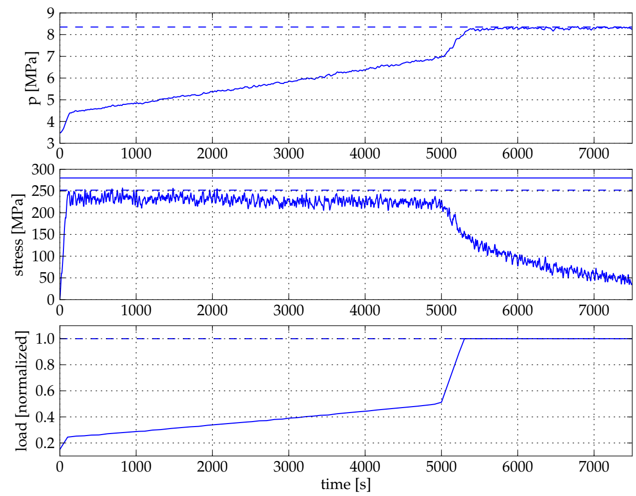

Fig. 2. Closed loop control with NMPC. *Top:* Evaporator pressure p and reference value p_{ref} (dashed). *Middle*: Rotor stress σ, soft constraint (dashed) and hard constraint (solid). *Bottom*: Turbine load u and upper limit (dashed).

The resulting closed-loop control of the start-up can be found in Figure 2, showing the evaporator pressure p, stress σ and load u. Figure 3 shows the corresponding variables in each NMPC iteration result. During the first 200 s, the load on the gas turbine is increased as fast as possible such that the stress on the steam turbine rotor tends towards the soft constraint. After that, and up to 5000 s, the load is increased such that the stress is as high as possible according to the NMPC model. However, due to modeling errors and disturbances, the stress on the rotor in the simulation model is varying considerably. As the load reaches 50%, additional increase will not increase the exhaust gas temperature, only the flow rate, which yields only small increases in the steam temperature. Thus, after this point in time it is feasible to increase the load at the maximum allowed rate while at the same time the stress level decreases. The plant is considered to reach full load at 5500 s.

The result is comparable to the optimal open loop start-up trajectories found in [3]. However, due to the imperfect model in the NMPC, disturbances in the simulation model and a soft constraint on σ that is lower than the constraint used in [3], the plant start-up is approximately 1300 s slower.

VI. SUMMARY AND FUTURE WORK

Modelica models for a combined-cycle power plant have been developed, to build simulation and optimization models for an NMPC problem. Within the open-source framework JModelica.org, the NMPC optimization problem has been formulated using the Optimica extension of Modelica, then transcribed by means of direct collocation methods into NLPs that have been solved by IPOPT. The obtained performance would also be adequate for real-time NMPC control, though a more detailed model would probably be required for an actual industrial application. One might also consider this framework as a rapid prototyping environment, to quickly test different model variants and control strategies, until a

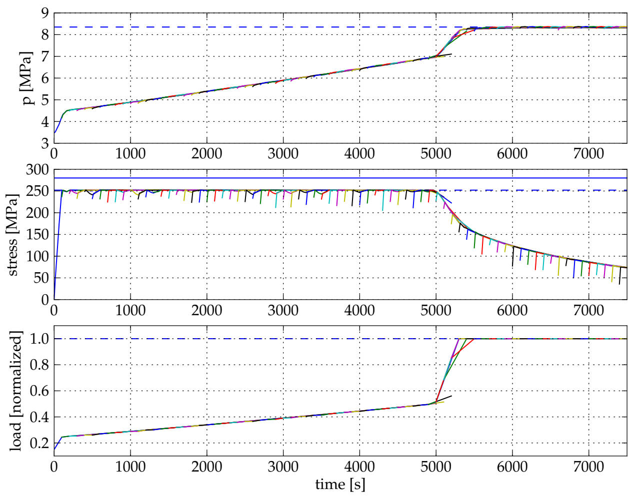

Fig. 3. NMPC iteration results at closed loop control. *Top:* Evaporator pressure p and reference value p_{ref} (dashed). *Middle*: Rotor stress σ, soft constraint (dashed) and hard constraint (solid). *Bottom*: Turbine load u and upper limit (dashed).

good one is found; then, the actual real-time implementation could also be carried out by other means.

The results obtained with this case study suggest that the presented framework could be successfully employed for applications of NMPC in the field of power plants and energy conversion systems. The full source code will be made available in future releases of JModelica.org, so it could be used a source of inspiration for similar projects.

Within this framework, the user may concentrate on the high-level description of the model and optimization problem, while the tool takes care of interfacing to state-of-the art numerical optimization routines. The use of high-level declarative modelling formalisms considerably shortens the design cycle; this is advantageous, as the controller design is an iterative process, in which the level of detail of the model and the optimal control law are repeatedly adapted until a satisfactory design is found.

An important future development of this work is the design of a state estimator. Using the tools available in JModelica.org, an extended Kalman filter (EKF) may be implemented. It is important to stress that the same high-level description and end-user convenience of the EKF, as for the modeling and optimization framework, would be used.

Several drastic simplifications were made in the plant model presented in this paper. Another interesting extension would be to give the model a higher level of detail, e.g., by using distributed parameter heat exchanger models, by including more accurate fluid property models, and possibly by considering two or three levels of pressure in the plant. More elaborate control strategies, e.g. taking advantage of desuperheating flows, could also be studied.

References

[1] J. Åkesson, K.-E. Årzén, M. Gäfvert, T. Bergdahl, and H. Tummescheit, "Modeling and optimization with Optimica and JModelica.org—languages and tools for solving large-scale dynamic optimization problem," *Computers and Chemical Engineering*, vol. 34, no. 11, pp. 1737–1749, Nov. 2010.

[2] K. Krüger, R. Franke, and M. Rode, "Optimization of boiler start-up using a nonlinear boiler model and hard constraints," *Energy*, vol. 29, no. 12–15, pp. 2239 – 2251, 2004, efficiency, Costs, Optimization, Simulation and Environmental Impact of Energy Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360544204001008

[3] F. Casella, F. Donida, and J. Åkesson, "Object-oriented modeling and optimal control: A case study in power plant start-up," in *18th IFAC World Congress*, Milano, Italy, Aug. 2011.

[4] S. Keerthi and E. Gilbert, "Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations," *Journal of Optimization Theory and Applications*, vol. 57, no. 2, pp. 265–293, 1988.

[5] L. Biegler and J. Rawlings, "Optimization approaches to nonlinear model predictive control," in *Proc. 4th International Conference on Chemical Process Control - CPC IV*, W. Ray and Y. Arkun, Eds. AIChE, CACHE, 1991, pp. 543–571.

[6] F. Allgöwer and A. Zheng, *Nonlinear Predictive Control*, ser. Progress in Systems Theory. Basel Boston Berlin: Birkhäuser, 2000, vol. 26.

[7] R. Findeisen, F. Allgöwer, and L. Biegler, Eds., *Assessment and Future Directions of Nonlinear Model Predictive Control*, ser. Lecture Notes in Control and Information Sciences. Springer, 2006.

[8] The Modelica Association, "The Modelica Association Home Page," http://www.modelica.org, 2011.

[9] P.-O. Larsson, "Optimization of Low-Level Controllers and High-Level Polymer Grade Changes," Ph.D. dissertation, Department of Automatic Control, Lund University, Sweden, Nov. 2011.

[10] A. Widd, "Physical Modeling and Control of Low Temperature Combustion in Engines," Ph.D. dissertation, Department of Automatic Control, Lund University, Sweden, Apr. 2012.

[11] Dassault Systemès, "Dymola home page," http://www.3ds.com/products/catia/portfolio/dymola.

[12] OpenModelica, "OpenModelica home page," http://www.openmodelica.org/.

[13] J. Åkesson, "Optimica—an extension of modelica supporting dynamic optimization," in *6th International Modelica Conference 2008*. Modelica Association, Mar. 2008.

[14] J. Åkesson, T. Ekman, and G. Hedin, "Implementation of a Modelica compiler using JastAdd attribute grammars," *Science of Computer Programming*, vol. 75, no. 1-2, pp. 21–38, Jan. 2010.

[15] MODELISAR, "Functional Mock-up Interface home page," http://www.modelisar.com/.

[16] A. Hindmarsh, P. Brown, K. Grant, S. Lee, R. Serban, D. Shumaker, and C. Woodward, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software*, vol. 31, pp. 363–396, 2005.

[17] B. M. Bell, "CppAD home page," http://www.coin-or.org/CppAD/.

[18] R. Parrotto, J. Åkesson, and F. Casella, "An XML representation of DAE systems obtained from continuous-time Modelica models," in *3rd International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, Oslo, Norway, Oct. 3 2010, pp. 91–98. [Online]. Available: http://www.ep.liu.se/ecp/047/010/ecp4710010.pdf

[19] F. Magnusson, "Collocation methods in JModelica.org," Master's thesis, Department of Automatic Control, Lund University, Sweden, Feb 2012.

[20] L. T. Biegler, *Nonlinear Programming*, ser. MOS-SIAM Series on Optimization. SIAM, 2010.

[21] J. Andersson, J. Åkesson, and M. Diehl, "CasADi – A symbolic package for automatic differentiation and optimal control," in *Proc. 6th International Conference on Automatic Differentiation*, ser. Lecture Notes in Computational Science and Engineering, S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther, Eds. Berlin: Springer, 2012.

[22] A. Wächter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[23] A. Griewank and A. Walther, *Evaluating Derivatives*, 2nd ed. SIAM, 2008.

[24] J. Andersson, J. Åkesson, F. Casella, and M. Diehl, "Integration of CasADi and JModelica.org," in *8th International Modelica Conference*, Mar. 2011.

[25] F. Casella and F. Pretolani, "Fast start-up of a combined-cycle power plant: a simulation study with Modelica," in *Proceedings 5th International Modelica Conference*, C. Kral, Ed. Vienna, Austria: Modelica Association, Sep. 6–8 2006, pp. 3–10. [Online]. Available: http://www.modelica.org/events/modelica2006/Proceedings/sessions/Session1a1.pdf