

This document contains a post-print version of the paper

Dynamic optimization of a slab reheating furnace with consistent approximation of control variables

authored by **A. Steinboeck, K. Graichen, and A. Kugi**
and published in *IEEE Transactions on Control Systems Technology*.

The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing. Please, scroll down for the article.

Cite this article as:

A. Steinboeck, K. Graichen, and A. Kugi, "Dynamic optimization of a slab reheating furnace with consistent approximation of control variables", *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1444–1456, 2011.
DOI: [10.1109/TCST.2010.2087379](https://doi.org/10.1109/TCST.2010.2087379)

BibTex entry:

```
@ARTICLE{steinboeck11c,  
  AUTHOR = {Steinboeck, A. and Graichen, K. and Kugi, A.},  
  TITLE = {Dynamic optimization of a slab reheating furnace with consistent approximation of control variables},  
  JOURNAL = {IEEE Transactions on Control Systems Technology},  
  YEAR = {2011},  
  volume = {16},  
  number = {6},  
  pages = {1444-1456},  
  doi = {10.1109/TCST.2010.2087379}  
}
```

Link to original paper:

<http://dx.doi.org/10.1109/TCST.2010.2087379>

Read more ACIN papers or get this document:

<http://www.acin.tuwien.ac.at/literature>

Contact:

Automation and Control Institute (ACIN)
Vienna University of Technology
Gusshausstrasse 27-29/E376
1040 Vienna, Austria

Internet: www.acin.tuwien.ac.at
E-mail: office@acin.tuwien.ac.at
Phone: +43 1 58801 37601
Fax: +43 1 58801 37699

Copyright notice:

© 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Dynamic Optimization of a Slab Reheating Furnace with Consistent Approximation of Control Variables

Andreas Steinboeck, Knut Graichen, *Member, IEEE*, and Andreas Kugi, *Member, IEEE*

Abstract—A dynamic optimization method is developed for temperature control of steel slabs in a continuous reheating furnace. The work was stimulated by the need for furnace control concepts that are computationally undemanding, robust, accurate, and capable of non-steady-state operating scenarios, where the properties and the temperature goals of slabs may vary significantly. The proposed hierarchical control structure is based on a continuous-time switched nonlinear model and uses the furnace zone temperatures as intermediate control variables. Consistent approximation is applied to obtain a parametric optimization problem that can be efficiently solved with the quasi-Newton method. Constraints on system states and control variables are considered by penalty terms in the cost function and saturation functions, respectively. The optimization method plans temperature trajectories for both the furnace and the slabs, which may be useful for open-loop control and feedforward branches of two-degrees-of-freedom control structures. The capabilities of the method are demonstrated in an example problem.

Index Terms—Consistent approximation, dynamic optimization, non-steady-state operation, open-loop control, optimal control, receding horizon optimization, reheating furnace for steel slabs, switched nonlinear system, trajectory planning.

I. INTRODUCTION

CONTINUOUS reheating furnaces heat up steel slabs to temperatures required for hot working processes. The control of these slab reheating furnaces is a challenging task because the system dynamics is often nonlinear, switched, and unknown in advance. Moreover, the number of controllable inputs is usually small compared to the multitude of control objectives, and there are several restrictions on both the inputs and the system states. Since, the control algorithms have to be executed in real-time, the available computational power may be a limiting factor. These challenges motivated the development of a tailored, model-based *optimal control method* for the temperature of slabs that are processed in a reheating furnace. The method is expound in this paper.

The treatise first describes the process of reheating steel slabs and the associated difficulties in terms of control (Subsection I-A). Subsection I-B refers to existing optimal control strategies for slab reheating furnaces, and Subsection I-C

outlines the motivation for developing a new control scheme based on dynamic optimization. A mathematical model of a representative slab reheating furnace is briefly introduced in Section II, followed by a cascaded furnace control system in Section III. Section IV, which can be considered as the major contribution of this paper, describes an optimal temperature control method that is used as a part of the cascaded control scheme. The paper is concluded with an example problem in Section V, which demonstrates the feasibility of the proposed method. Generally, an attempt is made to present at least the basic formulae and algorithms necessary to review and utilize the control approach.

A. Temperature Control of a Slab Reheating Furnace

Continuous reheating furnaces are used in the steel industry for heat treatment or reheating of slabs, billets, or similar products before they can undergo mechanical working, e. g., in a hot rolling mill. The term *continuous* means that the semifinished products are gradually conveyed through the longitudinal interior of the furnace, which is equipped with gas- or oil-fired burners. The word *continuous* is somewhat misleading because the slab movement itself is usually discontinuous. In a *pusher-type* slab reheating furnace, for instance, the slabs are arranged in one or several parallel rows and pushed through the furnace, as indicated in Fig. 1. The slabs slide on skids such that they can absorb heat at both the bottom and the top side. In so-called *walking-beam* furnaces, the skids are replaced by beams that alternately carry the slabs while slowly moving back and forth.

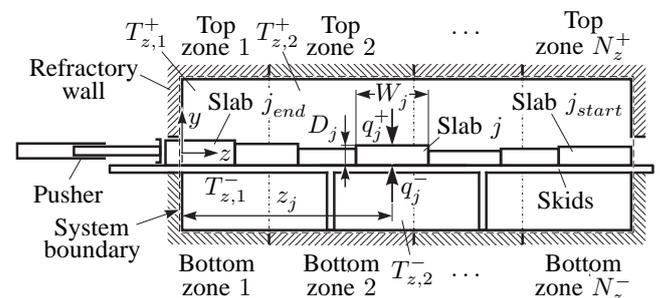


Fig. 1. Pusher-type slab reheating furnace (not to scale, symbols explained in Section II).

The control of continuous reheating furnaces is an interesting yet challenging field. Important *performance characteristics* of furnace control systems are processing costs, energy consumption, throughput of reheated material, and accuracy

Manuscript received June 25, 2010; revised October 01, 2010; accepted October 04, 2010. Manuscript received in final form October 09, 2010. Date of publication February 17, 2011; date of current version September 16, 2011. Recommended by Associate Editor F. A. Cuzzola. This work was supported by AG der Dillinger Hüttenwerke.

A. Steinboeck and A. Kugi are with the Automation and Control Institute, Vienna University of Technology, 1040 Vienna, Austria (e-mail: andreas.steinboeck@tuwien.ac.at, kugi@acin.tuwien.ac.at).

K. Graichen is with the Institute of Measurement, Control, and Microtechnology, University of Ulm, 89081 Ulm, Germany (e-mail: knut.graichen@uni-ulm.de).

of the controlled temperatures. Some points that may render temperature control of a slab reheating furnace an intricate task are:

- The temperature distribution inside the slabs cannot be measured.
- Slab reheating furnaces are nonlinear multiple-input-multiple-output systems and exhibit a switched dynamic behavior, i. e., the governing differential equations depend discontinuously on the time.
- There may be interdependencies of physical quantities which are too complex or too uncertain to be adequately reflected in a mathematical model.
- Both the slab and the furnace temperatures are subject to constraints.
- The control inputs are constrained, because the burners are located at certain points in the furnace and the fuel supply rates to the burners are limited.
- Depending on the process operation strategy, the slabs may considerably vary in size, material properties, available reheating time, initial temperature, desired final temperature, monetary value, etc. In fact, there are operating scenarios where each slab inside the furnace is a one-off product.
- When the slabs are withdrawn from the furnace, both their mean temperature and their temperature profile may have to satisfy certain requirements. Frequently, a homogeneous final temperature profile is desired.
- Metal forming processes are usually batch processes. Moreover, there may occur unscheduled standstills or delays of the workflow caused by upstream or downstream process steps. Therefore, the slab movement is generally discontinuous and sometimes unforeseeable.

Most temperature control systems for slab reheating furnaces, e. g., [1]–[12] are realized as hierarchical or modular control structures, which may significantly reduce the complexity of the control task. In cascaded control structures, the inner loops are usually controlling the furnace temperatures [3], [9], [10], [12]. Thus set-point values of furnace temperatures have to be selected by some high-level controllers.

B. Optimal Control Strategies for Slab Reheating Furnaces

Optimal control may be suitable for targeting the above-mentioned challenges in slab reheating processes. This expectation is supported by the following reasons:

- The nature and the slow dynamics of the slab reheating process call for a control strategy that utilizes *future information* about the configuration of the plant and the performance objectives—a strategy that is nicely implemented by dynamic optimization.
- Optimization is the first choice if a proper formal consideration of ‘higher’ or integral control objectives like minimum total energy consumption or maximum throughput is desired.
- In slab reheating furnaces the number of controlled variables (slab temperatures) usually significantly exceeds the number of controllable inputs (temperatures of the furnace interior being separated into zones).

A discussion of the rich literature on optimal control and optimization of slab reheating furnaces would exceed the scope of this paper. In general, however, two practical optimization strategies can be distinguished:

- (a) A given (dynamic) optimization problem may be discretized, e. g., by direct transcription [13] or multiple shooting [14]. Then, the obtained parametric optimization problem can be solved by standard methods of numerical optimization. This strategy is used for slab furnace control in [2], [11] (linear programming, simplex algorithm) and [1], [15] (quadratic programming).
- (b) Utilizing the calculus of variations or Pontryagin’s maximum principle [16], a dynamic optimization problem may be rewritten as a two-point boundary value problem representing the necessary optimality conditions. An application in furnace control is described in [17].

In either case, an efficient implementation of the utilized optimization algorithm is especially important for closed-loop control, e. g. in the context of *model predictive control* (cf. [1], [15], [18] for applications in slab furnace control).

The dynamic optimization scheme presented in [15] permits an efficient implementation because the algorithm is split into two steps: First, a low-dimensional optimization problem is solved to design piecewise polynomial trajectories as set-points for slab temperatures. Second, a linearized model is used by a model predictive controller to compute fuel flow rates to the burners such that the slab temperatures follow their set-point trajectories. The (1-dimensional) heat conduction problem inside the slabs is efficiently solved by the method of weighted residuals (collocation method).

A cascaded control structure that utilizes model predictive control in either loop is suggested in [1]. The outer loop computes optimal furnace temperatures based on a steady-state furnace model. The optimal set-point values are passed on to the inner loop, which utilizes a linear model predictive controller for both the slab movement and the fuel flow rates. The control system presented in [18] is similar but assumes that the slab movement is unalterable.

Another control algorithm described in [6] derives heat inputs for each slab to reach predefined temperatures at certain control points in the furnace. In a second step, a quadratic optimization problem selects the local furnace temperatures such that the deviations between the derived and expected heat inputs into the slabs are minimized. Effectively, there is no optimization horizon like in model predictive control.

An alternative strategy to reduce the computational requirements is to assume steady-state operating conditions. In the case of continuous slab reheating furnaces, steady-state does not mean that all time derivatives vanish, but that all slabs have equal properties including equidistant entry and exit times. With these assumptions, even accurate nonlinear models can be used to find optimum slab reheating trajectories [2] or (constant) optimum set-points of furnace temperatures [11], [19].

C. Motivation

Most published optimal control algorithms for large-scale slab furnaces, including the aforementioned, have either been

developed for steady-state operation or are based on relatively simple models—sometimes even linearized models—which may limit the performance of the control system in terms of accuracy. This is particularly relevant for model predictive control. Clearly, the longer an optimization horizon, the more important is the accuracy of the model used.

Thus, there is a need for accurate dynamic optimization methods based on furnace models that account for nonlinear effects like radiative heat transfer and dynamic interaction between the various sections of a furnace. The methods should be capable of optimizing *non-steady-state* (transient) furnace operation for frequently *varying* product types and they should account for relevant constraints on both system states and inputs. Moreover, the optimization methods should ensure that the desired final temperature distributions inside the slabs are realized.

These goals stimulated the development of a tailored, computationally inexpensive optimization approach for continuous slab reheating furnaces, that is presented in this paper. The method builds on optimal control theory [16], [20] and *consistent approximation* of input variables [21]–[24] in order to discretize the optimization problem. The underlying furnace model assumes that the local furnace temperatures serve as control inputs. Therefore, the furnace temperatures are used as intermediate control variables of a cascaded control structure. The proposed method is readily suitable for open-loop control and for trajectory planning in the feedforward branch of a two-degrees-of-freedom control structure. It may also be used for (suboptimal) model predictive control [25], [26]. However, for simplicity and in order to demonstrate the accuracy of the approach, this paper is focussed on open-loop control. In [2], [3], it is shown that open-loop control of slab temperatures is safe and feasible as long as the furnace temperatures are feedback controlled or at least monitored.

Compared to the furnace control methods referenced in Subsection I-B, the proposed optimization approach has the following advantages:

- It utilizes a furnace model that uses local furnace temperatures as inputs. Therefore, the model adequately accounts for those nonlinear phenomena (e. g., radiative heat transfer) that are not compensated by some low-level feedback control loops.
- It is suitable for switched dynamical systems like the slab reheating furnace. The number of system states may change at the switching points.
- The optimization algorithm permits an implementation that is efficient in terms of both memory requirements and computational load (see also Remark 14).
- In particular, it is capable of optimizing high-dimensional dynamical systems. The furnace system can have significantly more than 100 state variables. The number depends on the current stock of slabs inside the furnace.
- The furnace control problem is restricted by several constraints on both the system states and the inputs. If all restrictions were implemented as hard constraints, the control problem might be not feasible—a worst case that is not acceptable for real control applications. In this respect, the proposed optimization method is favorable,

because some constraints are realized by penalty terms only. This ensures that a feasible solution is always found.

- The algorithm accounts for the transient dynamic behavior of the system. Hence, the method is capable of optimizing the furnace operation even if the temperature goals and other properties of the slabs vary considerably.
- The algorithm is systematically developed, exhibits linear time complexity, and can be easily implemented in any standard high-level language.

II. MATHEMATICAL MODEL

Developing a model of a slab reheating furnace involves a trade-off between accuracy and mathematical complexity. A first important question concerns the system inputs. For a continuous slab reheating furnace, the fuel flow rates to the burners can be considered as primary physical inputs. Comprehensive furnace models, e. g., [1], [2], [12], [15], [18], [27], [28], indeed use the fuel flow rates as system inputs. However, these models are usually too complex as to be used in controllers or optimization algorithms.

Therefore, simplified mathematical furnace models are proposed for instance in [3], [8]–[10], [19], [29], [30]. In these analyzes, it is assumed that the furnace temperatures serve as system inputs, which can be independently chosen. Temperatures of the furnace interior are usually measured by thermocouples and controlled by low-level feedback control loops (cf. Section III). Therefore, some of the nonlinear dynamic phenomena that influence the local furnace temperature (e. g., the combustion process or the mass flow of the flue gases) are compensated by low-level control loops.

The mathematical model used in this paper was proposed in [30]. Its accuracy has been verified by temperature measurements with instrumented test slabs that were processed in a real furnace [28]. Comparisons with the measured slab temperatures showed that the accuracy achieved by the model is absolutely sufficient for the considered optimization and control task [30].

The model is outlined in the following. It accounts for radiative heat transfer in the furnace and heat conduction inside the slabs. After introducing some nomenclature in Subsection II-A, the dynamics of a single slab is analyzed. The equations thus obtained are then assembled for the whole furnace and interlinked by radiative boundary conditions in Subsection II-C. Finally, the model is summarized taking into account also slabs that are outside the furnace. Since the model is intended for optimization and control applications, great store was set by keeping the mathematical complexity at a modest level.

A. Slab Management, Geometry, and Position

Consider a reheating furnace as shown in Fig. 1, and let the indices $j \in \mathbb{N}$ uniquely identify all slabs. Moreover, let j_{start} designate the next slab to leave the furnace and j_{end} the last slab that was pushed in. That means, the slabs $j \in J = \{j_{start}, j_{start} + 1, \dots, j_{end}\}$ are currently reheated in the furnace. The slab j is pushed into the furnace at the time $t_{j,0}$, and it is removed at $t_{j,exit}$. At the times $t_{j_{start}+1,0}$ and

$t_{j_{end},exit}$, j_{start} and j_{end} are updated according to $j_{start} = j_{start} + 1$ and $j_{end} = j_{end} + 1$, respectively. Likewise, the number of slabs $N_s = |J|$ inside the furnace is updated at these times. Typical furnaces may contain more than 30 slabs, i. e., many more slabs than shown in the schematic diagram of Fig. 1.

In the yz -coordinate system defined in Fig. 1, slabs can only move in positive z -direction. Thus, z_j , i. e., the z -position of the center of the slab j , is a monotonously non-decreasing function of time t . Moreover, there is a *local* vertical coordinate y , which is 0 at the mid-plane of the respective slab j .

In this paper, a furnace with only one row of slabs is considered. If a furnace accommodates two or more parallel rows, averaging techniques are recommended to obtain a single representative slab at each position z_j . Generally, it is advisable to plan the reheating schedule such that only slabs with similar or equal properties (thickness D_j , width W_j , material, desired final temperature, etc.) are placed in parallel.

B. Heat Conduction Problem

The temperature field inside the slabs can be described by the heat conduction equation, which constitutes an initial-boundary value problem. The Galerkin method was employed in [30] to solve the 1-dimensional heat conduction problem

$$\rho_j c_j \frac{\partial T_j}{\partial t} = \frac{\partial}{\partial y} \left(\lambda_j \frac{\partial T_j}{\partial y} \right) \quad y \in (-D_j/2, D_j/2), t > t_{j,0}$$

for a *single* slab j with the mass density ρ_j , the specific heat capacity c_j , the thermal conductivity λ_j , and initial conditions $T_j(y, t_{j,0}) = T_{j,0}(y)$. The slab is subject to Neumann boundary conditions $q_j^\mp(t) = \mp \lambda_j \partial T_j / \partial y|_{y=\mp D_j/2}$ with the heat flux densities $q_j^\mp(t)$ into the bottom and the top slab surface. Throughout this paper, quantities belonging to the bottom and the top half of the furnace are designated by the superscripts $-$ and $+$, respectively.

Applying the Galerkin method, the temperature field inside the slab is approximated as $T_j(y, t) = \mathbf{x}_j^\top(t) \mathbf{h}_j(y)$, where the state vector $\mathbf{x}_j(t) = [x_{j,1}(t), x_{j,2}(t), x_{j,3}(t)]^\top$ contains the so-called Galerkin coefficients. As in [30], the orthogonal polynomials $h_{j,1}(y) = 1$, $h_{j,2}(y) = 2y/D_j$, and $h_{j,3}(y) = (2y/D_j)^2 - 1/3$ are used as Galerkin ansatz functions, which are summarized in the vector $\mathbf{h}_j(y) = [h_{j,1}(y), h_{j,2}(y), h_{j,3}(y)]^\top$. Then, the system dynamics follows as

$$\dot{\mathbf{x}}_j(t) = \mathbf{a}_j \mathbf{x}_j(t) + \mathbf{b}_j^- q_j^-(t) + \mathbf{b}_j^+ q_j^+(t) \quad t > t_{j,0} \quad (1a)$$

(cf. [30], [31]) with the initial value $\mathbf{x}_j(t_{j,0}) = \mathbf{x}_{j,0}$ corresponding to the initial temperature profile $T_j(y, t_{j,0}) = T_{j,0}(y)$ and the vectorial coefficients

$$\mathbf{a}_j = -\frac{12\lambda_j}{\rho_j c_j D_j^2} \text{diag} \{0 \quad 1 \quad 5\} \quad (1b)$$

$$\mathbf{b}_j^\mp = \frac{1}{\rho_j c_j D_j} [1 \quad \mp 3 \quad 15/2]^\top. \quad (1c)$$

Remark 1. Throughout this paper, it is assumed that λ_j and c_j are constant. Hence, the model (1) is, so far, *linear*. However,

an extension to temperature-dependent material parameters is possible (cf. [30], [31]) and would not alter or limit the presented optimization method. Subsection II-C will reveal that the favorable property of linearity is lost as soon as the inputs $q_j^\mp(t)$ are further specified.

In light of the intended control application, the Galerkin approximation proved useful, because it holds a direct physical interpretation of $\mathbf{x}_j(t)$: $x_{j,1}(t)$ is the mean temperature, $x_{j,2}(t)$ defines the asymmetry of the temperature profile, and $x_{j,3}(t)$ corresponds to the symmetric temperature inhomogeneity. Therefore, the chosen formulation facilitates an active homogenization of the slab temperature profile towards the end of the respective reheating period $[t_{j,0}, t_{j,exit}]$. Moreover, the formulation allows a reasonable approximation of the slab *surface* temperatures as $T_j(\mp D_j/2, t) = [1, \mp 1, 2/3] \mathbf{x}_j(t)$, which is important for radiation boundary conditions (cf. Subsection II-C). In [31] and [30], it is demonstrated that the accuracy achieved by the Galerkin approximation is absolutely sufficient for the intended application.

If the states and the heat inputs of all slabs $j \in J$ are assembled as $\mathbf{X}(t) = [\mathbf{x}_{j_{start}}^\top(t), \mathbf{x}_{j_{start}+1}^\top(t), \dots, \mathbf{x}_{j_{end}}^\top(t)]^\top$ and $\mathbf{q}^\mp(t) = [q_{j_{start}}^\mp(t), q_{j_{start}+1}^\mp(t), \dots, q_{j_{end}}^\mp(t)]^\top$, respectively, the dynamics of the whole furnace system can be described by the *linear* system

$$\dot{\mathbf{X}}(t) = \mathbf{A} \mathbf{X}(t) + \mathbf{B}^- \mathbf{q}^-(t) + \mathbf{B}^+ \mathbf{q}^+(t) \quad (2a)$$

(cf. [30]) with the sparse matrices

$$\mathbf{A} = [\delta_{i,j} \mathbf{a}_j]_{i=j_{start} \dots j_{end}, j=j_{start} \dots j_{end}} \quad (2b)$$

$$\mathbf{B}^\mp = [\delta_{i,j} \mathbf{b}_j^\mp]_{i=j_{start} \dots j_{end}, j=j_{start} \dots j_{end}}, \quad (2c)$$

and the Kronecker delta $\delta_{i,j}$. Note that the components as well as the dimensions $3N_s$ and N_s of $\mathbf{X}(t)$ and $\mathbf{q}^\mp(t)$, respectively, may vary at the times $t_{j,0}$ and $t_{j,exit}$. At first glance, (2) is a decoupled model. However, this is actually not the case, because $\mathbf{q}^\mp(t)$ do not constitute truly independent system inputs, as figured out in the following.

C. Radiative Heat Transfer

As the analysis is extended from the level of slabs to the whole furnace system, the local furnace zone temperatures $\mathbf{T}_z^\mp(t) = [T_{z,1}^\mp(t), T_{z,2}^\mp(t), \dots, T_{z,N_z^\mp}^\mp(t)]^\top$ (cf. Fig. 1) are introduced as new system inputs. Therefore, the furnace is divided into N_z^\mp volume sections, each consisting of a bottom and a top zone separated by the slabs. Usually, the number of slabs significantly exceeds the number of furnace zones, i. e., $N_s \gg N_z^\mp$. The temperatures $\mathbf{T}_z^\mp(t)$, which are assumed to be homogeneously distributed within each zone, represent a combination of local flue gas temperatures and wall surface temperatures. In this paper, the inputs $\mathbf{T}_z^\mp(t)$ are designed, meaning that they serve as optimization variables. This is more realistic than controlling $\mathbf{q}^\mp(t)$, because in the real furnace system $\mathbf{T}_z^\mp(t)$ can be measured and controlled by some low-level feedback control loops, which is not possible for $\mathbf{q}^\mp(t)$.

Assuming gray-body radiation and applying the Stefan-Boltzmann law with the net radiation method [32]–[35] to the

multisurface enclosure formed by the furnace yield the static radiative heat transfer model [30]

$$\mathbf{q}^\mp(t) = \mathbf{P}_z^\mp(t) (\mathbf{T}_z^\mp(t))^4 + \mathbf{P}_s^\mp(t) (\mathbf{M}^\mp \mathbf{X}(t))^4. \quad (3)$$

This expression is separately evaluated for the bottom and the top half of the furnace. The 4th power terms are characteristic for radiative heat exchange; the 4th powers are applied to each component of the respective vector. The $N_s \times 3N_s$ sparse matrix $\mathbf{M}^\mp = [\delta_{i,j} [1, \mp 1, 2/3]]_{i=1 \dots N_s, j=1 \dots 3N_s}$ maps $\mathbf{X}(t)$ to the bottom and top slab surface temperatures. The matrices $\mathbf{P}_z^\mp(t)$ and $\mathbf{P}_s^\mp(t)$ straightforwardly follow from the net radiation method and depend on the geometry of the furnace as well as the radiative properties of the participating surfaces, e.g., their emittances. A derivation of $\mathbf{P}_z^\mp(t)$ and $\mathbf{P}_s^\mp(t)$ is presented in [30]. The discontinuous slab movement causes these matrices to be piecewise constant with changes occurring only if the slabs are pushed forward.

D. Continuous-time Switched Dynamic Model

Consider that the furnace operation is to be analyzed within the time interval $[\tau_0, \tau_1]$. Then, joining (2) and (3) furnishes the continuous-time model

$$\dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{u}(t), t) \quad (4)$$

with the initial state $\mathbf{X}(\tau_0) = \mathbf{X}_0$ and the input $\mathbf{u}(t) = [(\mathbf{T}_z^-(t))^\top, (\mathbf{T}_z^+(t))^\top]^\top$.

Recall that $\mathbf{X}(t)$ may change its components and dimension. Therefore, to bring (4) into line with the standard notation, the state vector may be conceptually extended by the temperature states of slabs outside the furnace ($j \notin J$), namely, $\mathbf{x}(t) = [\dots, \mathbf{x}_{j_{start}-1}^\top(t), \mathbf{X}^\top(t), \mathbf{x}_{j_{end}+1}^\top(t), \dots]^\top$. For the additional slabs, the trivial differential equation $\dot{\mathbf{x}}_j(t) = \mathbf{0} \forall j \notin J$ is used. Hence,

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) = \begin{bmatrix} \mathbf{0} \\ \mathbf{F}(\mathbf{X}(t), \mathbf{u}(t), t) \\ \mathbf{0} \end{bmatrix} \quad (5)$$

with the initial state $\mathbf{x}(\tau_0) = \mathbf{x}_0$. The discontinuous time dependence of the non-autonomous system (5) reflects that the furnace exhibits a switched dynamic structure. It is emphasized that the number of inputs $\mathbf{u}(t)$ falls significantly below the number of current (active) states $\mathbf{X}(t)$, which may limit control possibilities. For computer implementation, the continuous-time model can be integrated using any standard solver algorithm. A simple and numerically robust discrete-time representation is suggested, for instance, in [30], [31].

Remark 2. For a brief discussion on open-loop stability of the system, let the trajectories $\tilde{\mathbf{X}}(t)$ and $\tilde{\mathbf{u}}(t)$ be a solution of (4). Consider, moreover, that the furnace system exhibits some initial state $\mathbf{X}_0 \neq \tilde{\mathbf{X}}(\tau_0)$ and that its inputs take the form $\mathbf{u}(t) = \tilde{\mathbf{u}}(t)$. Then, the control error $\mathbf{X}(t) - \tilde{\mathbf{X}}(t)$ decreases exponentially in the sense of some norm. The corresponding proof is based on Lyapunov's direct method [36], [37] and can be found in [30]. This is the essential justification for operating the furnace system with open-loop control, as outlined in the following.

III. OPEN-LOOP TEMPERATURE CONTROL SYSTEM

This section briefly describes the temperature control tasks associated with a continuous slab reheating furnace. It, therefore, builds a framework that accommodates the dynamic optimization approach developed in Section IV. Considering the multiple inputs and control parameters of a slab reheating furnace, it is reasonable to split up the task into hierarchical levels as indicated in Fig. 2.

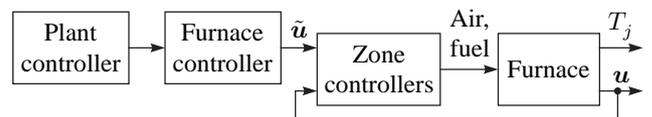


Fig. 2. Hierarchical open-loop control of a slab reheating furnace.

Roughly speaking, the (supervisory) *plant controller* provides the (high-level) *furnace controller* with any information required to govern the furnace operation. The furnace controller designs set-point values $\tilde{\mathbf{u}}(t)$ for the furnace zone temperatures $\mathbf{u}(t) = [(\mathbf{T}_z^-(t))^\top, (\mathbf{T}_z^+(t))^\top]^\top$. Finally, the (low-level) *zone controllers* aim to realize $\mathbf{u}(t) = \tilde{\mathbf{u}}(t)$ as good as possible. The latter constitutes a standard requirement for cascaded control loops. The main contribution of this paper concerns the design of the high-level furnace controller.

A. Supervisory Plant Controller

Supervisory plant control is not only concerned with the furnace but coordinates the whole production line. Thus, it links the furnace operation with upstream and downstream process steps like slab handling devices, scale breakers, rolling mills, or cooling units. In this paper, it is assumed that the plant controller sets at least the parameters shown in Table I.

TABLE I
SOME PARAMETERS OF SLAB j SPECIFIED BY THE PLANT CONTROLLER

Variables	Description
$t_{j,0}$	Time when the slab enters the furnace
$t_{j,exit}$	Time when the slab is withdrawn from the furnace
$\mathbf{x}_{j,0}$	Temperature state at $t_{j,0}$
$\tilde{\mathbf{x}}_{j,end}$	Desired temperature state at $t_{j,exit}$
$T_{j,abs,max}$...	Upper bound of slab temperature
$T_{j,end,min}$...	Lower bound of slab temperature at $t_{j,exit}$
$T_{j,end,max}$...	Upper bound of slab temperature at $t_{j,exit}$
c_j, λ_j	Specific heat capacity and thermal conductivity
D_j, W_j	Thickness and width of the slab
w_j	Weighting factor reflecting the monetary value

The values given in Table I implicitly define the sequence of slabs and their path-time diagrams. Note that the desired final slab temperature state $\tilde{\mathbf{x}}_{j,end}$ may call for some inhomogeneous temperature profile, which can be meaningful for subsequent production steps. However, for the majority of slabs, the desired final temperature profile is homogeneous.

B. High-level Furnace Controller

The task of the high-level furnace controller is to provide set-point values $\tilde{\mathbf{u}}(t)$ for the furnace zone temperatures $\mathbf{u}(t)$

such that the slab temperatures best reach their desired values (cf. $\tilde{\mathbf{x}}_{j,end}$ in Table I). For this purpose, a tailored optimal control method is suggested in Section IV. It may be considered as the *core* of the furnace temperature control system.

Several constraints limit the slab reheating process and, therefore, have to be allowed for when designing $\tilde{\mathbf{u}}(t)$. First, there are individual constraints

$$T_j(y, t) \leq T_{j,abs,max} \quad \forall j \in \mathbb{N}, y \in [-D_j/2, D_j/2], t \in [t_{j,0}, t_{j,exit}] \quad (6a)$$

$$T_{j,end,min} \leq T_j(y, t_{j,exit}) \leq T_{j,end,max} \quad \forall j \in \mathbb{N}, y \in [-D_j/2, D_j/2], \quad (6b)$$

on the slab temperatures, as defined in Table I. While (6a) should avoid overheating of the slab material, (6b) specifies a quality limit for the final slab temperature profile. There may be many more restrictions on the slab temperature trajectory. However, since they can be incorporated in a way analogous to (6), they are omitted in this paper.

Second, the zone temperatures as well as their slopes are limited by

$$\mathbf{T}_{z,min}^\mp(t) < \mathbf{T}_z^\mp(t) < \mathbf{T}_{z,max}^\mp(t) \quad \forall t \in [\tau_0, \tau_1] \quad (7a)$$

$$\dot{\mathbf{T}}_{z,min}^\mp(t) \leq \dot{\mathbf{T}}_z^\mp(t) \leq \dot{\mathbf{T}}_{z,max}^\mp(t) \quad \forall t \in (\tau_0, \tau_1). \quad (7b)$$

Here, the inequality signs are to be applied to corresponding components of the respective vectors. These constraints materialize safety measures as well as physical limitations of the burner equipment and the inner control loop. In many cases, $\mathbf{T}_{z,min}^\mp(t)$, $\mathbf{T}_{z,max}^\mp(t)$, $\dot{\mathbf{T}}_{z,min}^\mp(t)$, and $\dot{\mathbf{T}}_{z,max}^\mp(t)$ are constant. However, a tailored time-variant design of these constraints allows pursuing special control strategies, like manual start up or shut down of the furnace, temporary production interruption, and direct setting of desired zone temperatures. Although adjusting the constraints is a ‘safe’ way of manual intervention, it forces the control algorithm to deviate from its optimal solution and may thus diminish the reheating quality of slabs.

If (6) and (7) are implemented as *hard* constraints, the control problem defined by the supervisory plant controller may be unsolvable. Therefore, the following section adopts a problem formulation with mainly *soft* constraints. The advantage of this approach is that a solution of the problem is *always* feasible.

IV. DYNAMIC OPTIMIZATION

Now that the furnace model and a possible temperature control system have been outlined, the focus is shifted to an optimization method that serves as a high-level furnace controller. Subsection IV-A states a constrained optimal control problem, which is converted into an unconstrained optimization problem in Subsections IV-B and IV-C. The method of consistent approximation of inputs is employed in Subsection IV-D to obtain a parametric optimization problem that is solved by means of the quasi-Newton method in Subsection IV-E. Subsection IV-F touches upon the selection of appropriate optimization intervals. Throughout the section, an attempt is made to provide a reasonably general derivation, except for the final Subsection IV-G, where the method is specialized for the

considered furnace control problem. In view of the intended application in furnace control, it is particularly important to keep the computational requirements small because real-time execution of the control algorithm is desired.

A. Optimal Control Problem

Consider a non-autonomous system $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ (e. g., (5)) with states $\mathbf{x}(t) \in \mathbb{R}^n$, initial states $\mathbf{x}(\tau_0) = \mathbf{x}_0$, and inputs $\mathbf{u}(t) \in \mathbb{R}^m$. The function $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \times [\tau_0, \tau_1] \rightarrow \mathbb{R}^n$ may discontinuously depend on the time t (switched system), but local Lipschitz continuity with respect to $\mathbf{x}(t)$ and continuity with respect to $\mathbf{u}(t)$ are required. The system may be controlled by solving the optimal control problem

$$\underset{\mathbf{u} \in \mathcal{U}}{\text{minimize}} \quad \bar{S}(\mathbf{u}) = \bar{L}(\mathbf{x}(\tau_1)) + \int_{\tau_0}^{\tau_1} \bar{l}(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (8a)$$

$$\text{subject to} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \quad \forall t \in (\tau_0, \tau_1) \quad (8b)$$

$$\mathbf{x}(\tau_0) = \mathbf{x}_0 \quad (8c)$$

$$\underline{\mathbf{u}}(t) \leq \mathbf{u}(t) \leq \bar{\mathbf{u}}(t) \quad \forall t \in [\tau_0, \tau_1] \quad (8d)$$

$$\underline{\dot{\mathbf{u}}}(t) \leq \dot{\mathbf{u}}(t) \leq \bar{\dot{\mathbf{u}}}(t) \quad \forall t \in (\tau_0, \tau_1) \quad (8e)$$

$$\underline{\mathbf{c}}(t) \leq \mathbf{c}(\mathbf{x}(t)) \leq \bar{\mathbf{c}}(t) \quad \forall t \in [\tau_0, \tau_1] \quad (8f)$$

$$\underline{\mathbf{C}} \leq \mathbf{C}(\mathbf{x}(\tau_1)) \leq \bar{\mathbf{C}}. \quad (8g)$$

Here, \mathcal{U} denotes the set of bounded continuous functions defined in the interval $[\tau_0, \tau_1]$. At points t where $\dot{\mathbf{u}}(t)$ is discontinuous, (8e) is applied to both limits $\lim_{\tau \rightarrow t^-} \dot{\mathbf{u}}(\tau)$ and $\lim_{\tau \rightarrow t^+} \dot{\mathbf{u}}(\tau)$. Note that the constraints $\underline{\mathbf{u}}(t)$ and $\bar{\mathbf{u}}(t)$ are independent of $\dot{\mathbf{u}}(t)$ and $\bar{\dot{\mathbf{u}}}(t)$, respectively.

For the time being, the length of the user-defined optimization interval $[\tau_0, \tau_1]$ is arbitrary yet finite. A discussion about optimization intervals is postponed until Subsection IV-F. The cost functional $\bar{S} : \mathcal{U} \rightarrow \mathbb{R}$ consists of some terminal cost \bar{L} and some integral cost \bar{l} . They will be further specified in the following.

Due to the complexity and switching structure of (8b) and the (so far) arbitrary constraints (8d)–(8g), an application of standard results about existence and uniqueness of optimal control problems [38], [39] is difficult. In fact, specific constraints and initial conditions that render a solution of (8) infeasible can be easily found. To overcome this issue, some constraints are relaxed in the next section.

Remark 3. The above formulation is applicable to the furnace control system, since (8b) refers to (5), (8d) to (7a), (8e) to (7b), (8f) to (6a), and (8g) to (6b).

B. Introduction of Soft Constraints

First, the restrictions (8e)–(8g) are approximately reproduced by additional exterior penalty terms in the cost functional \bar{S} , i. e., the restrictions are implemented as soft constraints. Although the conversion of hard constraints into soft ones is only an approximation, there are good reasons for following this approach:

- Enforcing the restrictions (8e)–(8g) as hard constraints might entail an unfeasible optimization problem (8). However, the difficulty can be easily resolved if hard constraints are replaced by soft ones.

- Any terminal constraint (cf. (8g)) is converted into additional cost terms in \bar{S} . The absence of terminal constraints—as will be seen later—facilitates an uncomplicated and efficient solution of the optimization problem.

Consider the norm $\|\xi\|_W = \xi^T W \xi$, where ξ and W are some vector and positive semi-definite weighting matrix, respectively. Then, the cost functional \bar{S} from (8a) is replaced by

$$S(\mathbf{u}) = L(\mathbf{x}(\tau_1)) + \int_{\tau_0}^{\tau_1} l(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (9a)$$

with

$$\begin{aligned} L(\mathbf{x}(\tau_1)) &= \bar{L}(\mathbf{x}(\tau_1)) \\ &+ \underbrace{\left\| \min(\mathbf{0}, \mathbf{C}(\mathbf{x}(\tau_1)) - \underline{\mathbf{C}}) + \max(\mathbf{0}, \mathbf{C}(\mathbf{x}(\tau_1)) - \bar{\mathbf{C}}) \right\|_{\mathbf{W}_c}}_{L_1} \end{aligned} \quad (9b)$$

$$\begin{aligned} l(\mathbf{x}(t), \mathbf{u}(t), t) &= \bar{l}(\mathbf{x}(t), \mathbf{u}(t), t) \\ &+ \underbrace{\left\| \min(\mathbf{0}, \mathbf{c}(\mathbf{x}(t)) - \underline{\mathbf{c}}(t)) + \max(\mathbf{0}, \mathbf{c}(\mathbf{x}(t)) - \bar{\mathbf{c}}(t)) \right\|_{\mathbf{W}_c}}_{l_1} \\ &+ \underbrace{\left\| \min(\mathbf{0}, \dot{\mathbf{u}}(t) - \underline{\dot{\mathbf{u}}}(t)) + \max(\mathbf{0}, \dot{\mathbf{u}}(t) - \bar{\dot{\mathbf{u}}}(t)) \right\|_{\mathbf{W}_u}}_{l_2}, \end{aligned} \quad (9c)$$

and positive semi-definite matrices \mathbf{W}_c , \mathbf{W}_c , and \mathbf{W}_u containing penalty coefficients. \mathbf{W}_c and \mathbf{W}_u may be time-dependent. The fact that the chosen *exterior* penalty terms cannot ensure precise compliance with the restrictions (8e)–(8g) may be practically accounted for by a more conservative design of $\underline{\mathbf{C}}$, $\bar{\mathbf{C}}$, $\underline{\mathbf{c}}(t)$, $\bar{\mathbf{c}}(t)$, $\underline{\dot{\mathbf{u}}}(t)$, and $\bar{\dot{\mathbf{u}}}(t)$.

C. Incorporation of Input Constraints

Because of the foregoing considerations, the input constraints (8d) are the only inequality constraints that are left in (8). This type of restrictions can be readily incorporated if the inputs are transformed according to a monotonous saturation function $u = \varphi(U, \underline{u}, \bar{u}) \in [\underline{u}, \bar{u}]$ with the original *constrained* input $u \in [\underline{u}, \bar{u}]$ and a new *unconstrained* input $U \in \mathbb{R}$. An example for $\varphi(U, \underline{u}, \bar{u})$ is shown in Fig. 3.

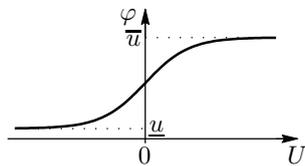


Fig. 3. Monotonous saturation function to account for input constraints.

Remark 4. The use of saturation functions for handling input constraints is presented in more detail in [40], [41]. In particular, it is shown in [40] that a constrained interval $[\underline{u}, \bar{u}]$ of the original input u corresponds to a singular arc of the unconstrained input U in the transformed optimal control problem. To account for this problem, a so-called regularization term εU^2 with some small $\varepsilon \in \mathbb{R}^+$ is added to the integrand l

in (9c). The term can also be conceptualized as an interior barrier function in the original optimal control problem. The convergence properties for $\varepsilon \rightarrow 0$ are investigated in [41].

Consider that φ is applied to vectors by individual evaluation of each vector component. Hence, (8d) is tantamount to $\mathbf{u}(t) = \varphi(\mathbf{U}(t), \underline{\mathbf{u}}(t), \bar{\mathbf{u}}(t))$ with the *unconstrained* optimization variable $\mathbf{U} \in \mathcal{U}$. For consistency, the bold character φ is used for the vector function. Thus, (8) can be reformulated as an *unconstrained* optimal control problem

$$\underset{\mathbf{U} \in \mathcal{U}}{\text{minimize}} \quad S(\varphi(\mathbf{U}, \underline{\mathbf{u}}, \bar{\mathbf{u}})) \quad (10a)$$

$$\text{subject to} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \varphi(\mathbf{U}(t), \underline{\mathbf{u}}(t), \bar{\mathbf{u}}(t)), t) \quad (10b)$$

$$\mathbf{x}(\tau_0) = \mathbf{x}_0 \quad (10c)$$

with S from (9a). If an iterative solution method is adopted, an initial guess for \mathbf{U} is easily found, because *any* input $\mathbf{U} \in \mathcal{U}$ yields a feasible solution. The method of consistent approximation is utilized in the following. It allows condensing (10) to a *parametric* optimization problem.

D. Consistent Approximation of Input Variables

Consider that the optimization variable \mathbf{U} is restricted to some finite dimensional function space, allowing \mathbf{U} to be defined by the expansion

$$\mathbf{U}(t) = \sum_{k=k_0}^{k_1} \eta_k \psi_k(t) \quad (11)$$

with bounded $\eta_k \in \mathbb{R}^m$ and bounded continuous functions $\psi_k : [\tau_0, \tau_1] \rightarrow \mathbb{R}$. The method is known as *consistent approximation* of input variables [21]–[24]. The values η_k serve as new optimization parameters and can be summarized in the parameter vector $\boldsymbol{\eta} = [\eta_{k_0}^T, \eta_{k_0+1}^T, \dots, \eta_{k_1}^T]^T \in \mathbb{R}^M$ with $M = m(k_1 - k_0 + 1)$. Consequently, the original cost *functional* S from (9a) becomes a cost *function* in terms of $\boldsymbol{\eta}$. For a concise notation, the function

$$\phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}(t), \bar{\mathbf{u}}(t)) = \varphi \left(\sum_{k=k_0}^{k_1} \eta_k \psi_k(t), \underline{\mathbf{u}}(t), \bar{\mathbf{u}}(t) \right) \quad (12)$$

is introduced. Thus, the dynamic optimal control problem (10) can be transcribed as a simple unconstrained parametric optimization problem

$$\underset{\boldsymbol{\eta} \in \mathbb{R}^M}{\text{minimize}} \quad S(\phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}, \bar{\mathbf{u}})) \quad (13a)$$

$$\text{subject to} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}(t), \bar{\mathbf{u}}(t)), t) \quad (13b)$$

$$\mathbf{x}(\tau_0) = \mathbf{x}_0. \quad (13c)$$

Consider a discrete-time domain with sampling points t_k ($k \in \{k_0, k_0 + 1, \dots, k_1\}$) ranging from $t_{k_0} = \tau_0$ to $t_{k_1} = \tau_1$. The sampling period $T_k = t_k - t_{k-1}$ does not need to be constant over time. The triangular shape function

$$\psi_k(t) = \begin{cases} \frac{t-t_{k-1}}{T_k} & \text{if } t \in [t_{k-1}, t_k] \\ \frac{t_{k+1}-t}{T_{k+1}} & \text{if } t \in (t_k, t_{k+1}] \\ 0 & \text{else} \end{cases} \quad (14)$$

proved suitable and is used throughout this paper. With this choice, \mathbf{U} from (11) is a linear interpolation of the nodal values $\boldsymbol{\eta}_k$. Thus, $\mathbf{U} \in \mathcal{U}$.

Remark 5. Considering the piecewise linear shape of ψ_k , it may be advantageous to replace (12) by

$$\phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}(t), \overline{\mathbf{u}}(t)) = \sum_{k=k_0}^{k_1} \varphi(\boldsymbol{\eta}_k, \underline{\mathbf{u}}(t_k), \overline{\mathbf{u}}(t_k)) \psi_k(t). \quad (15)$$

This alternative formulation of \mathbf{u} facilitates the definition of the constraints $\underline{\mathbf{u}}$ and $\overline{\mathbf{u}}$ in the discrete-time domain, i. e., the introduction of saturation functions (cf. Fig. 3) merely after applying the method of consistent approximation. Although (12) and (15) are not equivalent, both formulations adequately realize the input constraints (8d) in practical terms. For the remaining theoretical discussion, it does not matter whether (12) or (15) is used, and the example problem in Section V will be solved based on (15).

E. Finding an Optimum Solution

The parametric optimization problem (13) can be straightforwardly solved by various numeric methods, e. g., the *gradient method* (method of the steepest descent), the *conjugate gradient method*, and the *quasi-Newton method* with the Davidon-Fletcher-Powell (DFP) formula or the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula [16], [20], [42], [43]. In terms of convergence, the conjugate gradient method and the quasi-Newton method are usually super-linear, whereas the gradient method exhibits a linear convergence rate [43], [44].

All these methods have in common that they require the gradient $\mathbf{g}(\boldsymbol{\eta}) = (\text{d}S(\phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}, \overline{\mathbf{u}}))/\text{d}\boldsymbol{\eta})^\top$ of the cost function in (13a) (cf. Algorithm I further down). A convenient way of calculating $\mathbf{g}(\boldsymbol{\eta})$ is the adjoint-based approach

$$\mathbf{g}(\boldsymbol{\eta}) = \int_{\tau_0}^{\tau_1} \left(\frac{\partial H(\mathbf{x}(t), \mathbf{p}(t), \mathbf{u}, t)}{\partial \mathbf{u}} \Big|_{\mathbf{u}=\phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}(t), \overline{\mathbf{u}}(t))} \right. \\ \left. \frac{\text{d}\phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}(t), \overline{\mathbf{u}}(t))}{\text{d}\boldsymbol{\eta}} \right)^\top \text{d}t \quad (16a)$$

with

$$H(\mathbf{x}(t), \mathbf{p}(t), \mathbf{u}(t), t) = l(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{p}^\top(t) \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (16b)$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}(t), \overline{\mathbf{u}}(t)), t) \quad (16c)$$

$$\mathbf{x}(\tau_0) = \mathbf{x}_0 \quad (16d)$$

$$\dot{\mathbf{p}}^\top(t) = - \frac{\partial H(\mathbf{x}, \mathbf{p}(t), \phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}(t), \overline{\mathbf{u}}(t)), t)}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}(t)} \quad (16e)$$

$$\mathbf{p}^\top(\tau_1) = \frac{\partial L(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}(\tau_1)}. \quad (16f)$$

H is known as the *Hamiltonian* and $\mathbf{p}(t)$ are *adjoint states*. The proof of (16) is straightforward and can be found, for instance, in [24].

Remark 6. An advantage of the choice (14) is that the integration interval $[\tau_0, \tau_1]$ in (16a) reduces for many components of $\mathbf{g}(\boldsymbol{\eta})$. For instance, the subvector $\text{d}S(\phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}, \overline{\mathbf{u}}))/\text{d}\boldsymbol{\eta}_k$

only requires integration from t_{k-1} to t_{k+1} , because $\text{d}\phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}(t), \overline{\mathbf{u}}(t))/\text{d}\boldsymbol{\eta}_k = \mathbf{0} \forall t \notin [t_{k-1}, t_{k+1}]$ (cf. [24]).

Remark 7. In a similar way as (16e), (16c) is readily obtained from $\dot{\mathbf{x}}^\top(t) = \partial H(\mathbf{x}(t), \mathbf{p}, \phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}(t), \overline{\mathbf{u}}(t)), t)/\partial \mathbf{p}$.

Remark 8. For the optimization problem (13),

$$\mathbf{g}(\boldsymbol{\eta}) = \left(\frac{\text{d}S(\phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}, \overline{\mathbf{u}}))}{\text{d}\boldsymbol{\eta}} \right)^\top = \mathbf{0} \quad (17a)$$

$$\frac{\text{d}^2 S(\phi(\boldsymbol{\eta}, t, \underline{\mathbf{u}}, \overline{\mathbf{u}}))}{\text{d}\boldsymbol{\eta} \text{d}\boldsymbol{\eta}} \geq \mathbf{0} \quad (17b)$$

are necessary optimality conditions of first and second order, respectively. The condition (17b) means that the *Hessian* is positive semi-definite. For simplicity, however, the condition (17b) is frequently not explicitly verified. Therefore, the (exact) computation of the Hessian is not further discussed in this paper.

Remark 9. Using $\mathbf{g}(\boldsymbol{\eta}) = \mathbf{0}$ in (16) furnishes a two-point boundary value problem that allows—in principle—the direct computation of the optimal value of $\boldsymbol{\eta}$ (and the corresponding trajectories $\mathbf{x}(t)$ and $\mathbf{p}(t)$). Since an analytical solution is hardly feasible, approximate techniques like the *sweep method* or the *shooting method* [16], [20] are commonly applied. However, a different numerical approach that does not strictly require $\mathbf{g}(\boldsymbol{\eta}) = \mathbf{0}$ is used in this paper as outlined in the following.

Based on (16a), the quasi-Newton method may serve as an iterative solution technique for the optimization problem (13). The corresponding algorithm is the main result of this section and proceeds as follows:

ALGORITHM I ADJOINT-BASED QUASI-NEWTON METHOD

- (a) Provide an initial guess for $\boldsymbol{\eta}$ and set $\mathbf{H}^{-1} = \mathbf{I}$.
- (b) Integrate (16c) in forward direction, evaluate (16f), integrate (16e) in backward direction, and compute $\mathbf{g}(\boldsymbol{\eta})$ from (16a).
- (c) Compute a new search direction $\mathbf{s} = -\mathbf{H}^{-1} \mathbf{g}(\boldsymbol{\eta})$, determine the optimal step length

$$\bar{\alpha} = \arg \min_{\alpha \in \mathbb{R}^+} \{S(\phi(\boldsymbol{\eta} + \alpha \mathbf{s}, t, \underline{\mathbf{u}}, \overline{\mathbf{u}}))\}, \quad (18)$$

and apply the update $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} + \bar{\alpha} \mathbf{s}$.

- (d) Integrate (16c) in forward direction.
- (e) Stop if a suitable solution $\boldsymbol{\eta}$ has been found (termination criterion of the iterative optimization).
- (f) Compute (16f) and integrate (16e) in backward direction.
- (g) Set $\bar{\mathbf{g}} = \mathbf{g}(\boldsymbol{\eta})$ and compute the new value $\mathbf{g}(\boldsymbol{\eta})$ from (16a).
- (h) Evaluate $\mathbf{y} = \mathbf{g}(\boldsymbol{\eta}) - \bar{\mathbf{g}}$ as well as

$$\mathbf{H}^{-1} \leftarrow \left(\mathbf{I} - \frac{\mathbf{s} \mathbf{y}^\top}{\mathbf{y}^\top \mathbf{s}} \right) \mathbf{H}^{-1} \left(\mathbf{I} - \frac{\mathbf{y} \mathbf{s}^\top}{\mathbf{y}^\top \mathbf{s}} \right) + \bar{\alpha} \frac{\mathbf{s} \mathbf{s}^\top}{\mathbf{y}^\top \mathbf{s}} \quad (19)$$

(cf. [42], [43]) and restart at (c).

\mathbf{H}^{-1} is an approximation of the inverse of the Hessian (cf. (17b)), which is recurrently updated according to the BFGS

formula (19). The user-defined termination criterion in step (e) may, for instance, utilize some norm $\|\mathbf{g}(\boldsymbol{\eta})\|$ or the decrement of S . If the algorithm attains the optimal solution, (17a) is satisfied, and the decrement of S vanishes.

Remark 10. The differential equation (16e) for the adjoint states $\mathbf{p}(t)$ is often unstable, which makes its integration in backward direction numerically advantageous. The effect may be of particular interest if the optimization problem is to be repeatedly solved within a receding horizon scheme as suggested in Section IV-F.

Remark 11. It follows from (16e) that the adjoint states $\mathbf{p}(t)$ are continuous, even at switching times of the system. The reason for this favorable fact is that the switching times are predefined and, hence, independent of the system states $\mathbf{x}(t)$.

Note that (16c) and (16e) can be integrated with any standard ODE-solver. The advantage that (16c) and (16e) can be integrated *successively* (steps (b), (d) and (f)) is essentially based on the structure of the optimization problem (13); it does not contain any terminal constraints but only some terminal cost function $L(\mathbf{x}(\tau_1))$.

Remark 12. The selection of the initial guess for $\boldsymbol{\eta}$ in step (a) of Algorithm I is not critical in terms of the quality of the optimized solution or the convergence behavior. Generally, $\boldsymbol{\eta} = \mathbf{0}$ works well. Alternatively, previous optimization results may be utilized as new initial guesses, as will be described in Subsection IV-F.

The subsidiary optimization problem (18) of Algorithm I is particularly simple, because it is only 1-dimensional. Therefore, it is often called *line search* problem. In this paper, it is realized as a point-by-point search proceeding in the following way:

ALGORITHM II LINE SEARCH

- (a) Use the initial guess $\alpha = 1$ and compute $S(\phi(\boldsymbol{\eta} + \alpha \mathbf{s}, t, \underline{\mathbf{u}}, \bar{\mathbf{u}}))$.
- (b) Reduce α until the corresponding S is increasing.
- (c) Use the last three values of α and S , fit a quadratic polynomial, and return the position of its minimum as the optimal value $\bar{\alpha}$.

Remark 13. In Algorithm I, the quasi-Newton method is outlined with the BFGS formula (19). Essentially, \mathbf{s} is computed as a function of $\mathbf{g}(\boldsymbol{\eta})$ as well as previous values of $\mathbf{g}(\boldsymbol{\eta})$, \mathbf{s} , and $\bar{\alpha}$. This straightforward method proved convenient for the considered problem and, hence, is used throughout the paper. Many other numeric solution techniques like the gradient method, the conjugate gradient method, or the quasi-Newton method with the DFP formula operate in almost the same way as Algorithm I (cf. [43]). Only, the computation of the search direction \mathbf{s} may differ. For instance, the gradient method is simply realized by $\mathbf{s} = -\mathbf{g}(\boldsymbol{\eta})$ and omitting step (h), meaning that the direction of the steepest descent serves as a new search direction \mathbf{s} .

F. Finite Optimization Horizon

Algorithm I computes an (approximately) optimal value for $\boldsymbol{\eta} \in \mathbb{R}^M$, i.e., it solves a finite-dimensional optimization problem. Since the computational effort of the algorithm rises whereas its convergence behavior deteriorates as the dimension of the optimization problem increases, it is important to keep the number of optimization variables, i.e., $M = m(k_1 - k_0 + 1)$, at a moderate level. The number of controllable inputs m is more or less unalterable, whereas $k_1 - k_0 + 1$ depends on the length of the optimization horizon $[\tau_0, \tau_1]$ and the chosen sampling periods T_k .

If for the considered furnace system the optimization horizon $[\tau_0, \tau_1]$ is long enough, it happens that the optimal trajectory $\mathbf{u}(t)$ at the beginning of the horizon is almost independent of the optimal trajectory $\mathbf{u}(t)$ towards the end of the horizon. Here, the term *long enough* means approximately $2/3$ of the residence time of an average slab. This special trait of the furnace process is the justification for slicing the time domain into several *overlapping intervals* and for applying Algorithm I individually to each interval. The approach is often denoted as receding horizon control.

Hence, several low-dimensional optimization problems are solved rather than a single high-dimensional one. In general, the receding horizon approach furnishes satisfactory results if the optimization intervals are scheduled to have sufficient overlap.

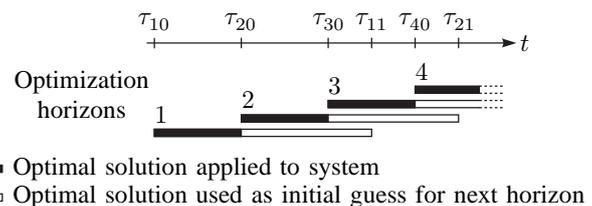


Fig. 4. Receding horizon approach with overlapping time intervals.

Fig. 4 outlines the strategy of finite optimization horizons. The horizon 1 ranging from τ_{10} to τ_{11} is computed first, and the solution valid in the interval $[\tau_{10}, \tau_{20}]$ is accepted as final optimal value. The solution in the overlapping interval $[\tau_{20}, \tau_{11}]$ is generally discarded, but may be utilized as a good initial guess for the optimization horizon 2. Subsequent optimization horizons are treated in the same manner.

Remark 14. The suggested optimization algorithm can also be used in a suboptimal way by executing a fixed number of iterations in each sampling step. Stability properties of such a suboptimal strategy in the context of model predictive control are studied for instance in [25], [26]. Moreover, the gradient method can be implemented with modest requirements in terms of memory and computing power, as is demonstrated for fast mechatronic systems in [45], [46].

G. Parameterization for Furnace Control System

Thus far, the section described a solution technique for the general optimization problem (8). The following specializes the technique to suit to the requirements of a slab reheating

furnace and in particular to the high-level furnace controller outlined in Subsection III-B. Note that Remark 3 at the end of Subsection IV-A has already established a link between the optimal control problem and the considered furnace control task.

The design of the cost functional $S(\mathbf{u}) = L(\mathbf{x}(\tau_1)) + \int_{\tau_0}^{\tau_1} l(\mathbf{x}(t), \mathbf{u}(t), t) dt$ (cf. (9a)) belonging to some optimization horizon $[\tau_0, \tau_1]$ is a crucial point for the performance of the high-level furnace controller. The design should allow for both the pertinent control objectives and a number of (soft) constraints. In the sequel, $L(\mathbf{x}(\tau_1))$ and $l(\mathbf{x}(t), \mathbf{u}(t), t)$ will be composed according to (9b) and (9c) in the form of $L = \bar{L} + L_1$ and $l = \bar{l} + l_1 + l_2$, respectively. However, \bar{l} is not required and therefore set to $\bar{l} = 0$.

The basic function

$$\bar{L}(\mathbf{x}(\tau_1)) = \sum_{j \in \mathbb{N} | t_{j,exit} \in [\tau_0, \tau_1]} \|\mathbf{x}_j(\tau_1) - \tilde{\mathbf{x}}_{j,end}\|_{\mathbf{W}_{end}} w_j \quad (20a)$$

implements the fundamental control objective that each slab, which leaves the furnace within the optimization interval $[\tau_0, \tau_1]$, reaches its desired final state. Recall that $\dot{\mathbf{x}}_j(t) = \mathbf{0} \forall t \geq t_{j,exit}$ was assumed in Subsection II-D (cf. (5)). $\mathbf{W}_{end} \in \mathbb{R}^{3 \times 3}$ is a constant positive semi-definite matrix which penalizes the final control deviation in terms of slab temperature states. The weighting factor w_j reflects the monetary value of the slab (cf. Table I).

Next, consider the upper bound $T_{j,abs,max}$ on the slab temperature trajectories according to (6a). It is of the type (8f) and can be implemented by adding

$$l_1(\mathbf{x}(t), t) = W_{lim} \sum_{j \in J} \max(0, \max_y(T_j(y, t)) - T_{j,abs,max})^2 w_j \quad (20b)$$

to the integral cost l . $W_{lim} \in \mathbb{R}^+$ is some constant weighting factor. In a similar way, the terminal constraints $T_j(y, t_{j,exit}) \in [T_{j,end,min}, T_{j,end,max}]$ from (6b) can be considered: The penalty term

$$L_1(\mathbf{x}(\tau_1)) = W_{end,lim} \sum_{j \in \mathbb{N} | t_{j,exit} \in [\tau_0, \tau_1]} \left(\max(0, \max_y(T_j(y, \tau_1)) - T_{j,end,max})^2 + \min(0, \min_y(T_j(y, \tau_1)) - T_{j,end,min})^2 \right) w_j \quad (20c)$$

with the constant weighting factor $W_{end,lim} \in \mathbb{R}^+$ is added to the cost function L (cf. (8g) and (9b)).

It remains to incorporate the restrictions (7), which limit the zone temperatures $T_z^\mp(t)$ as well as their slopes. Since, the constraint (7a) is of the sort (8d), it can be implemented following the lines of Subsection IV-C. Thus, the transformation

$$\mathbf{u}(t) = \phi\left(\boldsymbol{\eta}, t, \underbrace{\begin{bmatrix} T_{z,min}^-(t) \\ T_{z,min}^+(t) \end{bmatrix}}_{\underline{\mathbf{u}}(t)}, \underbrace{\begin{bmatrix} T_{z,max}^-(t) \\ T_{z,max}^+(t) \end{bmatrix}}_{\bar{\mathbf{u}}(t)}\right) \quad (21)$$

(cf. (12) or (15)) ensures that the optimized trajectories $T_z^\mp(t)$ strictly adhere to (7a). In contrast, the restriction (7b) being

of the type (8e) is merely implemented as a soft constraint. The pattern (9c) is adopted by adding the penalty function

$$l_2(\mathbf{u}(t), t) = \|\mathbf{e}(\mathbf{u}(t), t)\|_{\mathbf{W}_z} \quad (22)$$

with the vector

$$\mathbf{e}(\mathbf{u}(t), t) = \min\left(\mathbf{0}, \dot{\mathbf{u}}(t) - \underbrace{\begin{bmatrix} \dot{T}_{z,min}^-(t) \\ \dot{T}_{z,min}^+(t) \end{bmatrix}}_{\underline{\dot{\mathbf{u}}}(t)}\right) + \max\left(\mathbf{0}, \dot{\mathbf{u}}(t) - \underbrace{\begin{bmatrix} \dot{T}_{z,max}^-(t) \\ \dot{T}_{z,max}^+(t) \end{bmatrix}}_{\bar{\dot{\mathbf{u}}}(t)}\right)$$

and some positive semi-definite weighting matrix \mathbf{W}_z to the integral cost l .

Remark 15. Finally, it is worth considering the role of slabs that are currently outside the furnace, i. e., slabs having an index $j \notin J$. These slabs are governed by the trivial differential equation $\dot{\mathbf{x}}_j(t) = \mathbf{0}$ (cf. (5)) and, hence, are independent of the current control $\mathbf{u}(t)$. Therefore, it seems very reasonable that the integrand l does not contain $\mathbf{x}_j(t) \forall j \notin J$ (cf. (20b) and (22)). Based on this assumption, it follows from the differential equation (16e) that adjoint states (components of $\mathbf{p}(t)$) are constant whenever the corresponding slab is outside the furnace. This may cut short the integration of (16e).

Remark 16. The cost terms (20) and (22) satisfy the differentiability requirements imposed by (16a), (16e), and (16f).

Remark 17. The cost terms (20) and (22) do not explicitly take the energy or fuel consumption into account, because neither of these quantities is available in the underlying model. Therefore, the proposed method is not capable of systematically minimizing the energy consumption. Utilizing a more sophisticated model that uses the fuel flow rates as inputs (cf. [1], [2], [12], [15], [18], [28]) would significantly complicate the optimization routine. However, if a term like $\|\mathbf{u}(t)\|_{\mathbf{W}_T}$ with some positive semi-definite matrix \mathbf{W}_T were added to the integral cost l , the lowest possible zone temperature level would be maintained along the z -direction of the furnace. As demonstrated in [2], [47]–[50], this approach minimizes the energy consumption, because the bulk heat input is shifted towards the end of the furnace meaning that the flue gas is given more time to transfer thermal energy to the slabs.

As suggested in Subsection IV-F, the optimal control problem is consecutively solved for overlapping finite horizons. For the considered furnace system, it is recommendable to choose intervals of 3 h to 4 h. If longer intervals are used, the computational effort may increase and the convergence rate may suffer. The time offset between neighboring intervals should not exceed 1 h. These empirically found values can be motivated by the dynamics of the system. The input values $\mathbf{u}(t)$ applied at the instant t have a great influence on the system in the near future. Clearly, their influence decays as time goes on, and in practical terms the influence of $\mathbf{u}(t)$ is negligible after $t + 3$ h.

V. EXAMPLE PROBLEM

The optimization method worked out in the previous sections is now applied to an example problem of a continuous reheating furnace with $N_z^\mp = 5$ zones. The sequence of slabs and some of the parameters of Table I were randomly generated. This can be interpreted as a worst case scenario for testing the performance of the proposed dynamic optimization scheme. A good supervisory plant controller may schedule the slabs in a better order, which then permits more accurate reheating of slabs. In practice, however, the production plan or other process steps often force the supervisory plant controller to arrange slabs with little consideration of the slab reheating process. Thus, the following scenario is realistic.

A. Problem Formulation

The considered furnace is 35 m long and is assumed to have the same properties and input capabilities in the bottom and the top half. Therefore, the equivalence $T_z^-(t) = T_z^+(t)$ is introduced to halve the number of optimization variables. All slabs have the width $W_j = 3$ m such that $N_s = 12$. Since the slabs are withdrawn from the furnace at 0.5 h intervals, they stay inside the furnace for 6 h, i. e., $t_{j,exit} = t_{j,0} + 6$ h. The resulting path-time diagram is given in Fig. 5a).

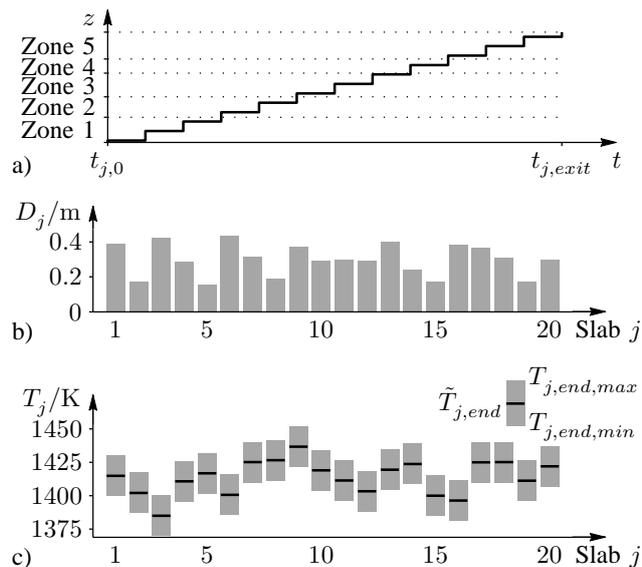


Fig. 5. Overview of slab properties, a) path-time diagram, b) slab thickness, c) desired final slab temperature.

The slab thicknesses (cf. Fig. 5b)) vary in the range 0.15 m to 0.44 m, whereas the weighting factor w_j was set to a constant value, i. e., all slabs are considered equally valuable. In this example it is assumed that each slab should attain a homogeneous final temperature profile, specified by $\tilde{\mathbf{x}}_{j,end} = [\tilde{T}_{j,end}, 0, 0]^T$. The randomly chosen values $\tilde{T}_{j,end}$ are shown in Fig. 5c) together with the constraints $T_{j,end,min}$ and $T_{j,end,max}$. To demonstrate the performance of the method, the restrictive constraints $T_{j,end,min} = \tilde{T}_{j,end} - 15$ K and $T_{j,abs,max} = T_{j,end,max} = \tilde{T}_{j,end} + 15$ K are used in this

example. In reality, most temperature tolerance bands are wider.

The considered planning horizon ranges from $\tau_0 = 0$ h to $\tau_1 = 10$ h and is separated into 20 shorter optimization intervals (length 3.5 h, overlap 3 h, cf. Fig. 4). The consistent approximation of the input variables $\mathbf{u}(t)$ is realized by means of (21) with (14), (15), and sampling periods $T_k \leq 2$ min. If T_k were increased, the computational load would significantly decrease, although the reheating quality of slabs would still be acceptable. Notable accuracy problems occur only if the order of magnitude of T_k approaches 0.5 h, i. e., the length of the pushing intervals.

B. Simulation Results

The limits specifying the constraints (7) are chosen to be constant with respect to t depending on the capacity of burners installed in the respective zone. Fig. 6a) shows that the optimized temperature $T_{z,4}^\mp$ of zone 4 strictly obeys the constraints $T_{z,4,min}^\mp$ and $T_{z,4,max}^\mp$. However, since the restriction (7b) is implemented as a soft constraint only (cf. (22)), the time derivative $\dot{T}_{z,4}^\mp$ may sometimes fall slightly outside the permissible range $[\dot{T}_{z,4,min}^\mp, \dot{T}_{z,4,max}^\mp]$, which is indicated in Fig. 6a) by the thin inclined lines. In practical terms, this is acceptable, particularly because $\dot{T}_{z,4,min}^\mp$ and $\dot{T}_{z,4,max}^\mp$ may be nominally shifted. Similar conclusions can be drawn from Fig. 6b) for the remaining zones.

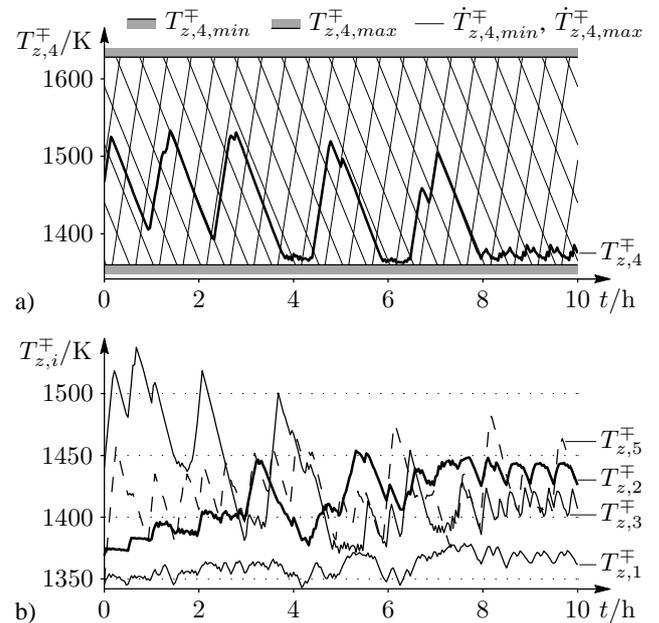


Fig. 6. Planned furnace zone temperatures, a) zone 4, b) zones 1, 2, 3, and 5.

In addition to the planned zone temperatures, the algorithm calculates the expected slab temperature trajectories. For two representative slabs $j = 15$ and $j = 16$, the temperature curves are shown in Figs. 7a) and b), respectively. The thick lines correspond to the mean temperatures, whereas the thin lines refer to the minimum and the maximum slab temperature.

The temperature of slab $j = 15$ is bumping up against the limit $T_{j,abs,max}$, mainly because the subsequent slabs are significantly thicker (see Fig. 5b)), implying that they require more heat input. Briefly before the slab $j = 15$ leaves the furnace, the temperature in the last zone is reduced such that the slab temperature reaches its desired final value $\tilde{T}_{j,end}$. Moreover, this reduction of the furnace temperature causes the mean temperature of slab $j = 16$ to temporarily halt at approximately 1388K (cf. the zoomed region in Fig. 7b)). Once more, it is emphasized that only the restriction (7a) is implemented as a hard constraint. All other restrictions are realized by penalty terms in the cost function and, therefore, may be violated if otherwise a solution of the optimization problem is unfeasible.

It can be concluded from Figs. 7a) and b) that a reheating time of 6 h is needlessly long for thin slabs like $j = 15$. However, it is appropriate for thicker slabs like $j = 16$. Fig. 7c) shows that the final temperature profile is well homogenized for the majority of the slabs. Noticeable inhomogeneities occur just with slabs that significantly differ from their neighbors in terms of thickness or desired final temperature. All slabs reach their desired final temperature range $[T_{j,end,min}, T_{j,end,max}]$.

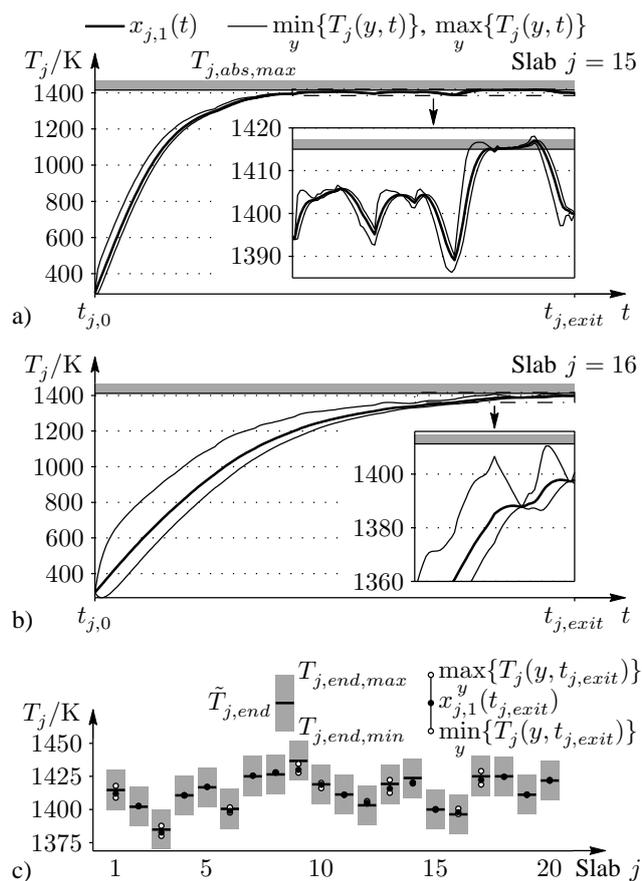


Fig. 7. Slab temperatures, a) temperature trajectory of slab $j = 15$, b) temperature trajectory of slab $j = 16$, c) final slab temperatures.

VI. CONCLUSION

The presented straightforward optimization method plans temperature trajectories of a continuous slab reheating furnace

under non-steady-state operating conditions. The approach uses consistent approximation of inputs to obtain a parametric optimization problem. The simple algorithm is geared to the specific needs of controlling slab reheating furnaces. However, the principle can be readily transferred to other nonlinear systems being similar in the sense of restrictions on inputs and system states, switched dynamics, and multiple control objectives paired with only few inputs.

The proposed approach essentially builds on the stipulation that the low-level controllers succeed in realizing the optimized furnace zone temperatures. If, for instance due to limited burner capacity, the inner control loop cannot satisfy this assumption, the reheating quality of the slabs may deteriorate. Another potential drawback is that some computations in Algorithm I are formulated in the continuous-time domain (cf. 16), which requires a numeric integration scheme for computer implementation. The choice of the integration algorithm certainly has an influence on the numerical accuracy and the required CPU time. For the considered furnace system (with slow dynamics), the accuracy achieved by a quasi-continuous implementation and standard ODE-solvers proved sufficient.

The proposed method is suitable for trajectory planning in open-loop or two-degrees-of-freedom control structures. It furnishes two main results: set-point trajectories of the furnace zone temperatures and expected slab temperatures. Having both types of trajectories at hand gives more flexibility for further control methods, particularly for two-degrees-of-freedom control. The principle may also be adopted in model predictive control schemes, but there the optimization routine has to be carried out repeatedly, meaning that a computationally efficient implementation is more important.

The structure of the employed nonlinear dynamic model turned out to be beneficial for both the convergence rate and the computational load of the iterative algorithm. Moreover, the chosen Galerkin approximation of the slab temperature proved useful in so far as it facilitates the active homogenization of the final slab temperature profile by the controller. The fact that the considered system is switched does not hamper the optimization approach, because the adjoint variables are continuous throughout. Moreover, the computational effort of the algorithm is rather moderate such that trajectories can be planned in real-time. By virtue of separating the optimization horizon into several overlapping intervals, the algorithm exhibits linear time complexity. Another advantage of the developed optimization method is that a feasible solution always exists and can be easily found.

The presented example problem demonstrates the feasibility of the approach. Constraints on the temperatures of the furnace and the slabs are adequately reflected, and the achieved reheating quality of the slabs is acceptable. The method scales well to larger problems where the furnace may contain more than 30 slabs. Encouraged by the obtained numeric results, it is planned to develop a similar optimization scheme for the discrete-time domain, which may further reduce the computational load. After additional verification of the method by means of the sophisticated furnace model developed in [28], it is intended to utilize the approach for trajectory planning in the high-level controller of a real furnace system.

ACKNOWLEDGMENT

The authors kindly express their gratitude to AG der Dillinger Hüttenwerke for both technical support and financing.

REFERENCES

- [1] L. Balbis, J. Balderud, and M. J. Grimble, "Nonlinear predictive control of steel slab reheating furnace," *Proceedings of the American Control Conference, Seattle, Washington, USA*, pp. 1679–1684, June 2008.
- [2] H. Ezure, Y. Seki, N. Yamaguchi, and H. Shinonaga, "Development of a simulator to calculate an optimal slab heating pattern for reheat furnaces," *Electrical Engineering in Japan*, vol. 120, no. 3, pp. 42–53, 1997.
- [3] F. Hollander and S. P. A. Zuurbier, "Design, development and performance of online computer control in a 3-zone reheating furnace," *Iron and Steel Engineer*, vol. 59, no. 1, pp. 44–52, January 1982.
- [4] M. Honner, Z. Vesely, M. Svantner, P. Litos, K. Bauer, M. Hercik, and A. Kuba, "Optimization of continuous industrial furnace operation," *Proceedings of the 6th European Conference on Industrial Furnaces and Boilers, Lisbon, Portugal*, pp. 85–94, April 2002.
- [5] B. Leden, "A control system for fuel optimization of reheating furnaces," *Scandinavian Journal of Metallurgy*, vol. 15, pp. 16–24, 1986.
- [6] P. Marino, A. Pignotti, and D. Solis, "Control of pusher furnaces for steel slab reheating using a numerical model," *Latin American Applied Research*, vol. 34, no. 4, pp. 249–255, 2004.
- [7] J. L. Roth, H. Sierpinski, J. Chabanier, and J. M. Germe, "Computer control of slab furnaces based on physical models," *Iron and Steel Engineer*, vol. 63, no. 8, pp. 41–47, August 1986.
- [8] H. Sibarani and Y. Samyudia, "Robust nonlinear slab temperature control design for an industrial reheating furnace," *Computer Aided Chemical Engineering*, vol. 18, pp. 811–816, 2004.
- [9] D. Staalman, "The funnel model for accurate slab temperature in reheating furnaces," *La Revue de Métallurgie*, vol. 101, no. 7, pp. 453–459, June 2004.
- [10] F. Vode, A. Jaklič, T. Kokalj, and D. Matko, "A furnace control system for tracing reference reheating curves," *Steel Research International, Metal Forming*, vol. 79, no. 5, pp. 364–370, 2008.
- [11] Z. Wang, Q. Wu, and T. Chai, "Optimal-setting control for complicated industrial processes and its application study," *Control Engineering Practice*, vol. 12, pp. 65–74, January 2004.
- [12] B. Zhang, Z. Chen, L. Xu, J. Wang, J. Zhang, and H. Shao, "The modeling and control of a reheating furnace," *Proceedings of the American Control Conference, Anchorage, Alaska, USA*, pp. 3823–3828, May 2002.
- [13] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, March–April 1998.
- [14] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, pp. 577–585, 2002.
- [15] N. Yoshitani, T. Ueyama, and M. Usui, "Optimal slab heating control with temperature trajectory optimization," *Proceedings of the 20th International Conference on Industrial Electronics, Control and Instrumentation, IECON'94*, vol. 3, pp. 1567–1572, September 1994.
- [16] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control*. New York: John Wiley & Sons, 1975.
- [17] S. Banerjee, D. Sanyal, S. Sen, and I. K. Puri, "A methodology to control direct-fired furnaces," *International Journal of Heat and Mass Transfer*, vol. 47, pp. 5247–5256, 2004.
- [18] H. Ko, J. Kim, T. Yoon, M. Lim, D. Yang, and I. Jun, "Modeling and predictive control of a reheating furnace," *Proceedings of the American Control Conference, Chicago, USA*, vol. 4, pp. 2725–2729, 2000.
- [19] L. Rixin and N. Baolin, "Mathematical model for dynamic operation and optimum control of pusher type slab reheating furnace," *Industrial Heating*, vol. 59, no. 3, pp. 60–62, March 1992.
- [20] M. Papageorgiou, *Optimierung*. München: Oldenbourg Verlag, 1991.
- [21] V. Azhmyakov, *Consistent Approximations of Constrained Optimal Control Problems*. Berlin: Logos, 2007.
- [22] E. Polak, *Optimization: Algorithms and consistent approximations*, ser. Applied Mathematical Sciences. New York: Springer, 1997, no. 124.
- [23] A. Schwartz, "Theory and implementation of methods based on Runge-Kutta integration for solving optimal control problems," Ph.D. dissertation, EECS Department, University of California, Berkeley, 1996.
- [24] K. L. Teo, C. J. Goh, and K. H. Wong, *A unified computational approach to optimal control problems*, ser. Pitman Monographs and Surveys in Pure and Applied Mathematics. New York: John Wiley & Sons, 1991.
- [25] D. DeHaan and M. Guay, "A real-time framework for model-predictive control of continuous-time nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 11, pp. 2047–2057, 2007.
- [26] K. Graichen and A. Kugi, "Stability and incremental improvement of suboptimal MPC without terminal constraints," *Accepted for publication in IEEE Transactions on Automatic Control*, 2010.
- [27] P. V. Barr, "The development, verification, and application of a steady-state thermal model for the pusher-type reheat furnace," *Metallurgical and Materials Transactions B*, vol. 26B, pp. 851–869, August 1995.
- [28] D. Wild, T. Meurer, and A. Kugi, "Modelling and experimental model validation for a pusher-type reheating furnace," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 15, no. 3, pp. 209–232, June 2009.
- [29] L. M. Pedersen and B. Wittenmark, "On the reheat furnace control problem," *Proceedings of the American Control Conference*, pp. 3811–3815, June 1998.
- [30] A. Steinboeck, D. Wild, T. Kiefer, and A. Kugi, "A mathematical model of a slab reheating furnace with radiative heat transfer and non-participating gaseous media," *International Journal of Heat and Mass Transfer*, vol. 53, pp. 5933–5946, 2010.
- [31] —, "A flexible time integration method for the 1D heat conduction problem," *Proceedings of the 6th Vienna Conference on Mathematical Modelling, Vienna, Austria, ARGESIM Report no. 35*, pp. 1204–1214, February 2009.
- [32] H. D. Baehr and K. Stephan, *Heat and Mass Transfer*, 2nd ed. Berlin Heidelberg: Springer-Verlag, 2006.
- [33] F. P. Incropera, D. P. DeWitt, T. L. Bergman, and A. S. Lavine, *Fundamentals of Heat and Mass Transfer*, 6th ed. Hoboken, NJ: John Wiley & Sons, 2007.
- [34] J. H. Lienhard IV and J. H. Lienhard V, *A Heat Transfer Textbook*, 3rd ed. Cambridge, Massachusetts: Phlogiston Press, 2002.
- [35] M. F. Modest, *Radiative Heat Transfer*, 2nd ed. New York: Academic Press, 2003.
- [36] H. K. Khalil, *Nonlinear Systems*, 3rd ed. New Jersey: Prentice Hall, 2002.
- [37] M. Vidyasagar, *Nonlinear systems analysis*, 2nd ed., ser. Classics in Applied Mathematics. Philadelphia, USA: SIAM, 1992, no. 42.
- [38] L. D. Berkovitz, *Optimal Control Theory*. New York: Springer, 1974.
- [39] E. B. Lee and L. Markus., *Foundations of optimal control theory*, ser. The SIAM Series in Applied Mathematics. New York: John Wiley & Sons, 1967.
- [40] K. Graichen and N. Petit, "Constructive methods for initialization and handling mixed state-input constraints in optimal control," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1334–1343, 2008.
- [41] —, "Incorporating a class of constraints into the dynamics of optimal control problems," *Optimal Control Applications and Methods*, vol. 33, no. 6, pp. 537–561, 2009.
- [42] C. T. Kelley, *Iterative Methods for Optimization*, ser. Frontiers in Applied Mathematics. Philadelphia: Society for Industrial and Applied Mathematics, 1999, no. 18.
- [43] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., ser. Springer Series in Operations Research. New York: Springer, 2006.
- [44] J. C. Dunn, "On l^2 conditions and the gradient projection method for optimal control problems," *SIAM Journal on Control and Optimization*, vol. 34, no. 4, pp. 1270–1290, 1996.
- [45] K. Graichen, T. Kiefer, and A. Kugi, "Real-time trajectory optimization under input constraints for a flatness-controlled laboratory helicopter," *Proceedings of the European Control Conference (ECC) 2009, Budapest, Hungary*, pp. 2061–2066, 2009.
- [46] K. Graichen, M. Egretzberger, and A. Kugi, "Suboptimal model predictive control of a laboratory crane," *8th IFAC Symposium on Nonlinear Control Systems (NOLCOS 2010), Bologna, Italy*, September 2010.
- [47] P. Fontana, A. Boggiano, A. Furinghetti, G. Cabras, and C. Simoncini, "An advanced computer control system for reheat furnaces," *Iron and Steel Engineer*, vol. 60, no. 8, pp. 55–62, August 1983.
- [48] R. Schurko, C. Weinstein, M. Hanne, and D. Pellicchia, "Computer control of reheat furnaces: A comparison of strategies and applications," *Iron and Steel Engineer*, vol. 64, no. 5, pp. 37–42, May 1987.
- [49] F. Shenvar, "Walking beam furnace supervisory control at Inland's 80-in. hot strip mill," *Iron and Steel Engineer*, vol. 71, no. 7, pp. 25–34, July 1994.
- [50] R. Westdorp, "Development and simulation of a control strategy for a reheating furnace," *Journal A*, vol. 29, no. 2, pp. 11–16, 1988.



Andreas Steinboeck received the M.Sc. degree in mechatronics from Loughborough University, Loughborough, UK, in 2005 and the Dipl.-Ing. degree in mechatronics from Johannes Kepler University, Linz, Austria, in 2007.

Since 2007, he works as a research assistant at the Automation and Control Institute at Vienna University of Technology. His main research interests cover numerical methods in engineering, micromechanics of periodical structures, stability analysis, imperfection sensitive structures as well as modeling, optimization, and control of nonlinear dynamical systems.



Knut Graichen received the Dipl.-Ing. degree in engineering cybernetics as well as his Ph.D. (Dr.-Ing.) degree from the University of Stuttgart, Stuttgart, Germany in 2002 and 2006, respectively. In 2007, he spent a year as a post-doctoral researcher with the Centre Automatique et Systèmes, Ecole Nationale Supérieure des Mines de Paris, France, before he joined the Automation and Control Institute at Vienna University of Technology, Austria as a Senior Researcher in 2008. Since August 2010, he is Professor at the Institute of Measurement, Control, and

Microtechnology at the University of Ulm, Germany. His main research interests cover nonlinear feedforward and feedback control design as well as optimal and model predictive control of nonlinear systems.



Andreas Kugi (M'94) received the Dipl.-Ing. degree in electrical engineering from Graz University of Technology, Austria, and the Ph.D. (Dr. techn.) degree in control engineering from Johannes Kepler University (JKU), Linz, Austria, in 1992 and 1995, respectively. From 1995 to 2000 he worked as an assistant professor and from 2000 to 2002 as an associate professor at JKU. He received his "Habilitation" degree in the field of automatic control and control theory from JKU in 2000. In 2002, he was appointed full professor at Saarland University,

Saarbrücken, Germany, where he held the Chair of System Theory and Automatic Control until May 2007. Since June 2007 he is a full professor for complex dynamical systems and head of the Automation and Control Institute at Vienna University of Technology, Austria. Since 2010, he is corresponding member of the Austrian Academy of Sciences.

His research interests include the physics-based modeling and control of (nonlinear) mechatronic systems, differential geometric and algebraic methods for nonlinear control, and the controller design for infinite-dimensional systems. He is involved in several industrial research projects in the field of automotive applications, hydraulic servo-drives, smart structures and rolling mill applications. He is Editor-in Chief of the *Control Engineering Practice*.