M. Diehl[1], I. Uslu[2], R. Findeisen[3], S. Schwarzkopf[2], F. Allgöwer[3], H. G. Bock[1], T. Bürner[1], E. D. Gilles[2,4], A. Kienle[4], J. P. Schlöder[1], and E. Stein[4]

Real-Time Optimization for Large Scale Processes: Nonlinear Model Predictive Control of a High Purity Distillation Column

Preprint ##-##

Real-Time Optimization for Large Scale Processes: Nonlinear Model Predictive Control of a High Purity Distillation Column

M. Diehl[1], I. Uslu[2], R. Findeisen[3], S. Schwarzkopf[2], F. Allgöwer[3], H. G. Bock[1], T. Bürner[1], E. D. Gilles,[2,4] A. Kienle[4], J. P. Schlöder[1], and E. Stein[4]

4th September 2001

Abstract

The purpose of this paper is an experimental proof-of-concept of the application of NMPC for large scale systems using specialized dynamic optimization strategies. For this aim we investigate the application of modern, computationally efficient NMPC schemes and real-time optimization techniques to a nontrivial process control example, namely the control of a high purity binary distillation column. All necessary steps are discussed, from formulation of a DAE model with 164 states up to the final application to the experimental apparatus. Especially an efficient real-time optimization scheme based on the direct multiple shooting method is introduced. It is characterized by an *initial value embedding* strategy, that allows to immediately respond to disturbances, and *real-time iterations*, that dovetail the optimization iterations with the real process development. Using this scheme, sampling times of 10 seconds are feasible on a standard PC. This shows that an efficient NMPC scheme based on large scale DAE models is feasible for the real-time control of a pilot scale distillation column.

Introduction

Over the last two decades linear model predictive control has emerged as a powerful and widely used control technique, especially in the process industry. Recently there is growing interest in model predictive control for *nonlinear* systems in academia and in the industrial process control community, and the properties of a variety of NMPC schemes have been investigated theoretically

[1]Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg
[2]Institut für Systemdynamik und Regelungstechnik (ISR), University of Stuttgart
[3]Institute for Systems Theory in Engineering (IST), University of Stuttgart
[4]Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

(see e.g. [DMS00, ABQ$^+$99] for a review). In addition, there has been significant progress in the area of dynamic process optimization that made on-line optimization for NMPC feasible [Wri96, Bie00, BDLS00] (compare also to the overview article on optimization on moving horizon in this book [BBB$^+$01]), and simulation studies have shown the real-time feasibility even for large scale process models [BDS$^+$00, DBS$^+$01], as considered in this paper.

The main purpose of the paper is an experimental proof-of-concept of the application of NMPC for large scale systems using specialized dynamic optimization strategies. In particular, we consider the experimental application of NMPC to a high purity binary distillation column. We want to show that NMPC can be applied to large scale chemical processes, if well suited optimization strategies are used, and that it leads to a reasonable control performance without much tuning. This is experimentally validated, after addressing the challenges of the practical realization as parameter estimation, on-line state estimation and reliable data transfer with an existing process control system. Considering the optimization based control of distillation columns, we also refer to the research article [HLM$^+$01] which deals with the problem of probabilistic constraints.

The paper is structured as follows: In Section 1 we describe the considered distillation column as well as the used DAE model. Section 2 contains the utilized formulation of the NMPC optimization problem, and in Section 3 we present the employed real-time optimization algorithm. In Section 4 we describe the experimental setup for the closed-loop experiments, and comment on state estimation. Section 5 contains experimental closed-loop results and the observed computation times, and gives a short comparison with a conventional PI controller.

1 Distillation Column and Equations

The experimental implementation of NMPC was carried out on a pilot plant distillation column for the separation of a binary mixture of Methanol and n-Propanol. The desired product compositions are minimum 0.99 mol/mol (low boiling component) for the top product and maximum 0.01 mol/mol for the bottom product.

The column has a diameter of 0.10 m and a height of 7 m and consists of $N = 40$ bubble cap trays. The overhead vapour is totally condensed in a water cooled condenser which is open to atmosphere. The reboiler is heated electrically. Several variables are measured and monitored on-line during each experiment, such as temperatures of feed and reflux streams, at the reboiler and condenser and on each tray of the column, volumetric flow rates of feed, reflux, distillate and bottom product streams and the column pressure. Fluid dynamic stable operation of the column is checked by the pressure drop along the column for all operating conditions presented in this study. The nominal operating conditions of the plant are listed in Table 1.

The flowsheet of the distillation system is shown in Figure 1. The preheated feed stream F_{vol} enters the column at the feed tray as saturated liquid. It can be switched automatically between two feed tanks in order to introduce well defined disturbances in feed concentrations.

Process inputs available for control purposes are the heat input to the reboiler, Q, and the reflux flow rate L_{vol}. Although the main control purpose is to maintain the product purity specifications for a distillation column, product composition measurements are often expensive, unreliable and with delays. Therefore in this study temperatures T_{14} and T_{28} on trays 14 and 28 are selected as two controlled variables (cf. Sec. 2).

A distributed control system (DCS) is used for data acquisition and the basic control loops of the flow rates, the heat input, the liquid levels in the reboiler and the condenser. To implement more advanced control schemes the DCS is connected to a PC from and to which direct access from UNIX workstations is possible. The NMPC scheme and the state estimator are implemented on UNIX workstations, i.e. the DCS is only used for data acquisition and the basic control loops.

Feed rate, F_{vol} [l/h]	14.0
Feed composition, x_F	0.32
Feed temperature, T_F [^oC]	70.0
Top composition, x_{41}	0.99
Bottom composition, x_0	0.0006
Temperature tray 28 T_{28} [^oC]	70.0
Temperature tray 14 T_{14} [^oC]	88.0
Reflux flow, L_{vol} [l/h]	4.3
Heat input, Q [kW]	2.5
Top pressure [bar]	0.97
Reboiler holdup [l]	8.5
Condenser holdup [l]	0.17

Table 1: Nominal operating conditions

1.1 Differential Algebraic Model

Depending on the model simplifications different kinds of models can be obtained for the dynamics of the distillation column. For the predictions in the NMPC controller we use a (simple) equilibrium stage model, which is considered to capture the main features of the column dynamics. The presented nonlinear DAE model is based on the following assumptions:

- total condenser
- negligible vapor holdup
- constant molar liquid holdup

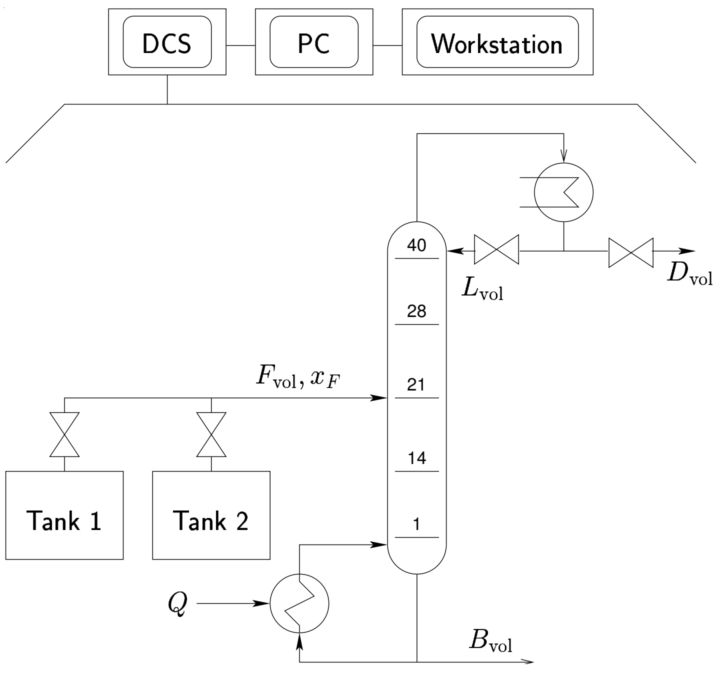


Figure 1: Flowsheet of the distillation column

- perfect mixing
- the mixture is at equilibrium temperature
- Murphree efficiency is applied for each tray

The model consists principally of overall and component material balances and energy balance for each tray ℓ where $\ell = 1, 2, \ldots, N$ and N is the total number of trays (compare also Figure 2). For notational convenience the index $\ell = 0$ is used for the reboiler and $\ell = N + 1$ for the condenser. The following balance equations are each written for the trays, the reboiler and the condenser, in this order. Since the molar liquid holdup, n_ℓ, is constant, overall material balances become:

$$0 = L_{\ell+1} - L_\ell + V_{\ell-1} - V_\ell + F_\ell \tag{1}$$
$$0 = L_1 - L_0 - V_0 \tag{2}$$
$$0 = V_N - L_{N+1} - D \tag{3}$$

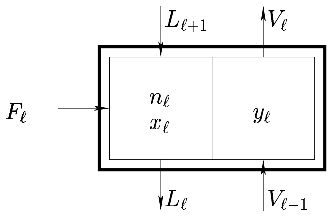


Figure 2: Distillation tray, ℓ

Component material balance for components $k = 1, N_c - 1$:

$$\begin{aligned} n_\ell \dot{x}_{\ell,k} &= L_{\ell+1}x_{\ell+1,k} - L_\ell x_{\ell,k} + V_{\ell-1}y_{\ell-1,k} - V_\ell y_{\ell,k} + F_\ell x_{F,\ell,k} & (4)\\ n_0\dot{x}_{0,k} &= L_1 x_{1,k} - L_0 x_{0,k} - V_0 y_{0,k} & (5)\\ n_{N+1}\dot{x}_{N+1,k} &= V_N y_{N,k} - (L_{N+1} + D)x_{N+1,k} & (6)\end{aligned}$$

Energy balances:

$$\begin{aligned} n_\ell \dot{h}^L_\ell &= n_\ell \sum_{i=1}^{N_c-1} (h^L_{\ell,k} - h^L_{\ell,N_c})\dot{x}_{\ell,k} + n_\ell c^L_{p,\ell}\dot{T}_\ell \\ &= L_{\ell+1}h^L_{\ell+1} - L_\ell h^L_\ell + V_{\ell-1}h^V_{\ell-1} - V_\ell h^V_\ell + F_\ell h^L_{F,\ell} & (7)\\ n_0\dot{h}^L_0 &= n_0 \sum_{i=1}^{N_c-1} (h^L_{0,k} - h^L_{0,N_c})\dot{x}_{0,k} + n_0 c^L_{p,0}\dot{T}_0 \\ &= L_1 h^L_1 - L_0 h^L_0 - V_0 h^V_0 + Q - Q_{\text{loss}} & (8)\end{aligned}$$

In the equations above, the molar concentrations of the liquid and vapour phases are represented by $x_{\ell,k}$ and $y_{\ell,k}$; k is the index for components and N_c is the total number of components. For the process of interest in this study $N = 40$ and $N_c = 2$. The molar vapour and liquid fluxes are denoted by V_ℓ and L_ℓ, the molar feed and distillate flows by F_ℓ and D. F_ℓ becomes 0 if the tray ℓ has no feed stream. A single feed stream is introduced to the column on tray 21. The energy balance for the reboiler includes terms for the heat input, Q, and the possible heat loss, Q_{loss}. There are only $N_c - 1$ independent component balances since by definition

$$\begin{aligned} \Sigma_k x_{\ell,k} &= 1 & (9)\\ \Sigma_k y_{\ell,k} &= 1 & (10)\end{aligned}$$

Assuming an ideal mixture, the vapour phase composition in equilibrium with the liquid phase, y^*_ℓ, is described by Rault's law:

$$y^*_{\ell,k} = \frac{P^s_k(T_\ell)x_{\ell,k}}{P_\ell} \qquad (11)$$

Here $P^s_\ell(T_\ell)$ is the equilibrium vapour pressure of the pure components and determined by the Antoine equation in terms of the temperature, T_ℓ, and constant parameters A, B and C :

$$P^s_k(T_\ell) = \exp\left(A_k - \frac{B_k}{T_\ell + C_k}\right) \tag{12}$$

It should be noted that $\dot{T}$ in the energy balances (7) and (8) is obtained by implicit differentiation of the combination of (10) - (12).

To account for the deviation from thermodynamic equilibrium due to finite mass transfer resistance the definition of Murphree efficiency, α_ℓ, is applied for each tray:

$$\alpha_\ell = \frac{y_{\ell,k} - y_{\ell-1,k}}{y^*_{\ell,k} - y_{\ell-1,k}} \quad \ell = 1, \dots, N \tag{13}$$

To determine the (constant) pressures we assume that the condenser pressure is fixed to the outside pressure, i.e., $P_{N+1} = P_{\text{top}}$, whereas the pressures P_ℓ on the trays and the reboiler are calculated under the assumption of a constant pressure drop, ΔP_ℓ, from tray to tray, i.e.,

$$P_\ell = P_{\ell+1} + \Delta P_\ell \quad \ell = N, N-1, \dots, 2, 1, 0. \tag{14}$$

Additionally to the molar flow rates, which cannot be measured, also the *volumetric* flow rates of the feed and reflux streams, F_{vol} and L_{vol}, are required. They can be measured and controlled by the DCS. For example L_{vol} is determined from $L_{\text{vol}} = L_{N+1}v^m$, where v^m is the molar volume of the reflux stream.

The enthalpies of the liquid and vapour phases, h^L_ℓ and h^V_ℓ, partial molar volumes, v^m, and heat capacities, $c^L_{p,\ell}$, are defined as functions of temperature and composition; the effect of pressure on h_ℓ and v^m are neglected in the model. For details of the correlations the reader is refered to [Die01].

1.1.1 Summarizing the DAE:

We can subsume all system states in two vectors $\mathbf{x}$ and $\mathbf{z}$ which denote the differential and the algebraic state vectors, respectively.

The (molar) Methanol concentrations in reboiler, on the 40 trays, and in the condenser x_ℓ for $\ell = 0, 1, \dots, N+1$ are the 42 components of the differential state vector $\mathbf{x}$. The liquid and vapor (molar) fluxes L_ℓ and V_ℓ ($\ell = 1, 2, \dots, N$) out of the 40 trays as well as the 42 temperatures T_ℓ ($\ell = 0, 1, 2, \dots, N+1$) of reboiler, trays and condenser form the 122 components of the algebraic state vector $\mathbf{z} = (L_1, \dots, L_N, V_1, \dots, V_N, T_0, \dots, T_{N+1})^T$ [1]. Note that those algebraic variables that can easily be eliminated (as e.g. h^L_ℓ, y_ℓ, $P^s_k(T_\ell)$, etc.) do not count as algebraic variables in this formulation.

[1]The equilibrium temperature of the condenser mixture helps to define the temperature of the reflux, when not specified. Otherwise, this last algebraic variable could be eliminated without changing the dynamics.

The two components of the control vector $\mathbf{u} = (L_{\mathrm{vol}}, Q)^T$ are the volumetric reflux flow, L_{vol}, out of the condenser, and the heat input, Q, determining the molar vapour flux out of the reboiler. All system parameters can be subsumed in a vector $\mathbf{p}$. The resulting model has 42 differential equations $\mathbf{f}$, and 122 algebraic equations $\mathbf{g}$.

We can write the DAE system, which has index one, in the following summarized form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) \quad (15)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}). \quad (16)$$

1.2 Estimation of the Model Parameters

In the actual application, the performance of NMPC depends on the quality of the model. Considering this fact, steady state and and open-loop dynamic experiments have been performed. To obtain measurements of the dynamic behaviour of the column step changes in the feed rate F_{vol} and composition x_F, the reflux rate L_{vol}, and heat input Q were performed.

Measurements of *all* temperatures $T_0, \dots, T_{N+1}$ were taken for least squares fitting of the simulated to the observed behaviour. The assumptions for this fit are that the tray efficiencies are constant on each of the two column sections, i.e. $\alpha_1 = \dots = \alpha_{N_F}$ and $\alpha_{N_F+1} = \dots = \alpha_N$, that the pressure losses are equal on all trays: $\Delta P_0 = \dots = \Delta P_N$, and that the molar tray and condenser holdups coincide: $n_1 = \dots = n_N$. Therefore, the parameters to be adjusted to the dynamic experimental data were: Q_{loss}, α_1, α_{N_F+1}, ΔP_0, and n_1. The parameter estimation was performed with the off-line version of the multiple shooting method that is described below in the context of real-time optimization. For details we refer to [Die01].

2 Nonlinear Model Predictive Controller Setup

As usual in distillation control, the product purities x_B and x_D at reboiler and condenser are not controlled directly – instead, an inferential control scheme which controls the deviation of the temperatures on tray 14 and 28 from a given setpoint is used. Earlier investigations have shown that the temperatures (respectively concentrations) on these trays are much more sensitive to changes in the inputs of the system than the product concentrations [AR92]. It can be expected, that if these concentrations are kept constant, the product purities are safely maintained for a large range of process conditions. Since the tray temperatures correspond directly to the concentrations via the Antoine equation and the temperatures on tray 14 and 28 are measured directly, we refine the controll objective to keep these temperatures as close to their setpoint values as possible. In the following we will use $\tilde{\mathbf{T}}(\mathbf{z}) := (T_{14}, T_{28})^T$ for the controlled temperatures and $\tilde{\mathbf{T}}_{\mathrm{ref}} := \left(\; T_{14}^{\mathrm{ref}}, T_{28}^{\mathrm{ref}} \; \right)^T$ for the desired setpoints.

A desired steady state $\mathbf{x}_s$, $\mathbf{z}_s$, and the corresponding control $\mathbf{u}_s$ as well as the steady state temperatures can be determined as the solution of the steady state equation

$$\begin{aligned} \mathbf{f}(\mathbf{x}_s, \mathbf{z}_s, \mathbf{u}_s, \mathbf{p}) &= \mathbf{0}, \\ \mathbf{g}(\mathbf{x}_s, \mathbf{z}_s, \mathbf{u}_s, \mathbf{p}) &= \mathbf{0}, \\ \tilde{\mathbf{T}}(\mathbf{z}_s) - \tilde{\mathbf{T}}_{\text{ref}} &= \mathbf{0}, . \end{aligned} \tag{17}$$

In the following we will refer to this set of equations as $\mathbf{r}(\mathbf{x}_s, \mathbf{z}_s, \mathbf{u}_s, \mathbf{p}) = \mathbf{0}$. Notice that last equation restricts the steady state to satisfy the inferential control aim of keeping the temperatures at the fixed reference values. The necessary degrees of freedom are the two components of the steady state controls $\mathbf{u}_s$.

The open-loop objective is formulated as the integral of a least squares term $||\mathbf{l}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{u}_s, \mathbf{p})||_2^2$ with

$$\mathbf{l}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{u}_s, \mathbf{p}) := \begin{pmatrix} \tilde{\mathbf{T}}(\mathbf{z}) - \tilde{\mathbf{T}}_{\text{ref}} \\ \mathbf{R}(\mathbf{u} - \mathbf{u}_s) \end{pmatrix}. \tag{18}$$

The second component is introduced for regularization, with a small diagonal weighting matrix $\mathbf{R} = \text{diag}(0.05\,^\circ\text{C}\,\text{h}\,\text{l}^{-1}, 0.05\,^\circ\text{C}\,\text{kW}^{-1})$.

To ensure nominal stability of the closed loop-system, we follow here a somewhat practical approach based on results given in [DMS96, CA98]. We append an additional prediction interval $[t_0 + T_c, t_0 + T_p]$ to the control horizon $[t_0, t_0 + T_c]$, with the controls fixed to the setpoint values $\mathbf{u}_s$ determined by (17). If the control horizon is sufficiently large, the closed-loop system will be stable. A horizon length of $T_p - T_c = 3600$ seconds proved to be sufficient in all performed experiments.

2.0.1 The optimal control problem:

The resulting optimal control problem is formulated as follows:

$$\min_{\mathbf{u}(\cdot), \mathbf{x}(\cdot), \mathbf{p}} \quad \int_{t_0}^{t_0+T_p} ||\mathbf{l}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{u}_s, \mathbf{p})||_2^2 \, dt \tag{19}$$

subject to the model DAE

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) \\ \mathbf{0} &= \mathbf{g}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) \end{aligned} \quad \text{for } t \in [t_0, t_0 + T_p]$$

with

$$\mathbf{u}(t) = \mathbf{u}_s \quad \text{for } t \in [t_0 + T_c, t_0 + T_p].$$

Furthermore the initial values for the differential states and values for the system parameters are given by:

$$\begin{aligned} \mathbf{x}(t_0) &= \mathbf{x}_0, \\ \mathbf{p} &= \mathbf{p}_0 = \text{constant}. \end{aligned}$$

All state and control inequality constraints are combined to:

$$\tilde{\mathbf{c}}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) \geq \mathbf{0} \quad t \in [t_0, t_0 + T_p].$$

In particular, we require that the bottoms product and distillate streams L_0 and V_{N+1} cannot become negative, which implicitly leads to "natural" upper limits on the inputs Q and L_{vol}. Additionally, explicit lower and upper bounds for Q and L_{vol} are given.

The steady state control $\mathbf{u_s}$ is determined by the steady state equation

$$\mathbf{0} = \mathbf{r}(\mathbf{x_s}, \mathbf{z_s}, \mathbf{u_s}, \mathbf{p}).$$

3 Real-time Solution of the NMPC Optimization Problems

In this section we will describe the newly developed *real-time iteration* scheme that we employed for the on-line computations in this study. For a more detailed description of the algorithm we refer to [Die01]; cf. also [BDS+00, FAD+00, DBS+01]

The scheme is based on the direct multiple shooting method [Pli81, BP84], which is introduced in the overview article [BBB+01] in this book (Section 5, "Introduction into Direct Solution Algorithms"), to which we explictly refer here. We stay close to the notation used in this article, with one important difference: in contrast to direct multiple shooting for ODEs, we need to account for the algebraic states in the DAE model equations.

Direct Multiple Shooting for DAE: In addition to the *differential* node values $\mathbf{s}_i^x$ (which are denoted $\mathbf{s}_i$ in [BBB+01]), we also introduce *algebraic* node values $\mathbf{s}_i^z$. For simplicity we use a piecewise constant control representation with control parameters $\mathbf{q}_0, \dots, \mathbf{q}_{N-1}$ on the N multiple shooting intervals. The prediction horizon $[t_0 + t_c, t_0 + T_p]$ is chosen to be the last interval, so that $t_{N-1} = t_0 + T_c$ and $t_N = t_0 + T_p$, with constant steady state control $\mathbf{q}_{N-1} := \mathbf{u}_s$.

On each subinterval $[t_i, t_{i+1}]$ we compute the independent trajectories $\mathbf{x}_i(t)$ and $\mathbf{z}_i(t)$ as the solution of a *relaxed* initial value problem:

$$\begin{aligned} \dot{\mathbf{x}}_i(t) &= \mathbf{f}(\mathbf{x}_i(t), \mathbf{z}_i(t), \mathbf{q}_i, \mathbf{p}) && (20) \\ \mathbf{0} &= \mathbf{g}(\mathbf{x}_i(t), \mathbf{z}_i(t), \mathbf{q}_i, \mathbf{p}) - e^{-\beta \frac{t - t_i}{t_{i+1} - t_i}} \mathbf{g}(\mathbf{s}_i^x, \mathbf{s}_i^z, \mathbf{q}_i, \mathbf{p}) && (21) \\ \mathbf{x}_i(t_i) &= \mathbf{s}_i^x, \quad \mathbf{z}_i(t_i) = \mathbf{s}_i^z && (22) \end{aligned}$$

The decaying subtrahend in (21) with $\beta > 0$ is deliberately introduced to allow an efficient DAE solution for initial values $\mathbf{s}_i^z$ that may violate temporarily the consistency conditions (16) (cf. Bock et al. [BES88], Schulz et al. [SBS98]). Note that the i-th multiple shooting trajectories $\mathbf{x}_i(t), \mathbf{z}_i(t)$ on $[t_i, t_{i+1}]$ are functions of $\mathbf{s}_i^x, \mathbf{s}_i^z, \mathbf{q}_i$, and $\mathbf{p}$, so that we will write: $\mathbf{x}_i(t; \mathbf{s}_i^x, \mathbf{s}_i^z, \mathbf{q}_i, \mathbf{p})$, and $\mathbf{z}_i(t; \mathbf{s}_i^x, \mathbf{s}_i^z, \mathbf{q}_i, \mathbf{p})$.

To remove the freedom introduced by the DAE relaxation, additional equalities $\mathbf{g}(\mathbf{s}_i^x, \mathbf{s}_i^z, \mathbf{q}_i, \mathbf{p}) = \mathbf{0}$ have to be added to the NLP formulation given in [BBB$^+$01], and taking account of the additional variables $\mathbf{x}_s, \mathbf{z}_s, \mathbf{u}_s, \mathbf{p}$ as well of the steady state constraint (17), we formulate the following structured **Nonlinear Program (NLP)** in the unknowns $\boldsymbol{\xi} := (\mathbf{s}_0^x, \ldots, \mathbf{s}_N^x, \mathbf{s}_0^z, \ldots, \mathbf{s}_{N-1}^z, \mathbf{q}_0, \ldots, \mathbf{q}_{N-2}, \mathbf{x}_s, \mathbf{z}_s, \mathbf{u}_s, \mathbf{p})$:

$$\min_{\boldsymbol{\xi}} \sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} \|\mathbf{l}(\mathbf{x}_i(t; \mathbf{s}_i^x, \mathbf{s}_i^z, \mathbf{q}_i, \mathbf{p}), \mathbf{z}_i(t; \mathbf{s}_i^x, \mathbf{s}_i^z, \mathbf{q}_i, \mathbf{p}), \mathbf{q}_i, \mathbf{u}_s, \mathbf{p})\|_2^2 \, dt \tag{23}$$

subject to

$$\begin{aligned}
\mathbf{s}_0^x &= \mathbf{x_0}, \quad \mathbf{p} = \mathbf{p_0}, && && (24)\\
\mathbf{s}_{i+1}^x &= \mathbf{x}_i(t_{i+1}; \mathbf{s}_i^x, \mathbf{s}_i^z, \mathbf{q}_i, \mathbf{p}), && i = 0, \ldots, N-1, && (25)\\
\mathbf{0} &= \mathbf{g}(\mathbf{s}_i^x, \mathbf{s}_i^z, \mathbf{q}_i, \mathbf{p}), && i = 0, \ldots, N-1, && (26)\\
\mathbf{0} &\leq \tilde{\mathbf{c}}(\mathbf{s}_i^x, \mathbf{s}_i^z, \mathbf{q}_i, \mathbf{p}), && i = 0, \ldots, N-1, && (27)\\
\mathbf{0} &= \mathbf{r}(\mathbf{x_s}, \mathbf{z_s}, \mathbf{u_s}, \mathbf{p}). && && (28)
\end{aligned}$$

3.1 Real-Time Iterations and Initial Value Embedding

In the real-time context, the above NLP (23)-(28) has to be solved several times; a crucial observation is that the optimization problems differ *only in the values* $\mathbf{x_0}$ *and* $\mathbf{p_0}$, which enter the problem through the *linear* constraints (24). Instead of considering a sequence of unrelated optimization problems $P(\mathbf{x_0}, \mathbf{p_0})$, each of which is solved independently by an iterative SQP type method, we shift the focus towards the solution iterations themselves: the real-time iteration scheme can be considered as a sequence of Newton type iterates towards the solution of the above problem, with the particularity that the values for $\mathbf{x_0}$ and $\mathbf{p_0}$ are changed *during* the iterations.

The variant of the algorithm that we used in this study is based on the constrained Gauss-Newton method. For the current iterate $\boldsymbol{\xi}_k$ and the current values $(\mathbf{x}_0)_{k+1}$ and $(\mathbf{p}_0)_{k+1}$, all problem functions are linearized to yield the following, specially structured **Quadratic Program (QP)** in the variables $\Delta\boldsymbol{\xi} = (\Delta\mathbf{s}_0^x, \ldots, \Delta\mathbf{s}_N^x, \Delta\mathbf{s}_0^z, \ldots, \Delta\mathbf{s}_{N-1}^z, \Delta\mathbf{q}_0, \ldots, \Delta\mathbf{q}_{N-2}, \Delta\mathbf{x}_s, \Delta\mathbf{z}_s, \Delta\mathbf{u}_s, \Delta\mathbf{p})$:

$$\min_{\Delta\boldsymbol{\xi}} \sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} \|\mathbf{l}_i(t) + \mathbf{L}_i(t) \, (\Delta\mathbf{s}_i^{x\,T}, \Delta\mathbf{s}_i^{z\,T}, \Delta\mathbf{q}_i{}^T, \Delta\mathbf{u}_s{}^T, \Delta\mathbf{p}^T)^T\|_2^2 \, dt \tag{29}$$

subject to

$$\begin{aligned}
\Delta\mathbf{s}_0^x &= (\mathbf{x}_0)_{k+1} - \mathbf{s}_0^x, \quad \Delta\mathbf{p} = (\mathbf{p}_0)_{k+1} - \mathbf{p}, && && (30)\\
\Delta\mathbf{s}_{i+1}^x &= \mathbf{x}_i + \mathbf{X}_i \, (\Delta\mathbf{s}_i^{x\,T}, \Delta\mathbf{s}_i^{z\,T}, \Delta\mathbf{q}_i{}^T, \Delta\mathbf{p}^T)^T, && i = 0, \ldots, N-1, && (31)\\
\mathbf{0} &= \mathbf{g}_i + \mathbf{G}_i^z \Delta\mathbf{s}_i^z + \mathbf{G}_i \, (\Delta\mathbf{s}_i^{x\,T}, \Delta\mathbf{q}_i{}^T, \Delta\mathbf{p}^T)^T, && i = 0, \ldots, N-1, && (32)\\
\mathbf{0} &\leq \tilde{\mathbf{c}}_i + \tilde{\mathbf{C}}_i \, (\Delta\mathbf{s}_i^{x\,T}, \Delta\mathbf{s}_i^{z\,T}, \Delta\mathbf{q}_i{}^T, \Delta\mathbf{p}^T)^T, && i = 0, \ldots, N-1, && (33)\\
\mathbf{0} &= \mathbf{r} + \mathbf{R} \, (\Delta\mathbf{x}_s{}^T, \Delta\mathbf{z}_s{}^T, \Delta\mathbf{u}_s{}^T)^T + \mathbf{R}^p \Delta\mathbf{p}. && && (34)
\end{aligned}$$

The solution $\Delta\boldsymbol{\xi}_k$ of this quadratic program is first used to determine the control $(\mathbf{q}_0)_{k+1} := (\mathbf{q}_0)_k + (\Delta\mathbf{q}_0)_k$ which is immediately given to the plant, and secondly to compute the next real-time iterate:

$$\boldsymbol{\xi}_{k+1} := \boldsymbol{\xi}_k + \Delta\boldsymbol{\xi}_k. \tag{35}$$

The iterations *never terminate*. Instead, the values $(\mathbf{x}_0)_{k+1}$ and $(\mathbf{p}_0)_{k+1}$ are changed from one iteration to the next, according to the current state and parameter estimates.

Remark 1: Note that the objective function in (29) can, neglecting a constant, equivalently be written as

$$\begin{aligned}&\textstyle\sum_{i=0}^{N-1}\{(\Delta\mathbf{s}_i^{x\,T}, \Delta\mathbf{s}_i^{z\,T}, \Delta\mathbf{q}_i{}^T, \Delta\mathbf{u}_s{}^T, \Delta\mathbf{p}^T)\,\mathbf{h}_i \\ &+\tfrac{1}{2}(\Delta\mathbf{s}_i^{x\,T}, \Delta\mathbf{s}_i^{z\,T}, \Delta\mathbf{q}_i{}^T, \Delta\mathbf{u}_s{}^T, \Delta\mathbf{p}^T)\,\mathbf{H}_i\,(\Delta\mathbf{s}_i^{x\,T}, \Delta\mathbf{s}_i^{z\,T}, \Delta\mathbf{q}_i{}^T, \Delta\mathbf{u}_s{}^T, \Delta\mathbf{p}^T)^T\}.\end{aligned}$$

with Hessian blocks and gradient vectors

$$\mathbf{H}_i := 2\int_{t_i}^{t_{i+1}} \mathbf{L}_i(t)^T\mathbf{L}_i(t)\,dt \quad \text{and} \quad \mathbf{h}_i := 2\int_{t_i}^{t_{i+1}} \mathbf{L}_i(t)^T\mathbf{l}_i(t)\,dt.$$

This formulation explicitly shows that the linear least squares problem (29)-(34) is nothing else than a finite dimensional quadratic programming problem. In fact, the Gauss-Newton matrices $\mathbf{H}_i$ can be regarded a cheap approximation of the *exact Hessian* blocks $\mathbf{H}_i^{\text{exact}}$, as they arise in the exact Hessian SQP method.

Remark 2: Instead of using $\boldsymbol{\xi}_k + \Delta\boldsymbol{\xi}_k$ directly as the new iterate (we call this the *warm start* strategy) it is alternatively possible to account for the movement of the horizon in time, if the multiple shooting intervals are equally spaced with interval lengths that correspond to a sampling time. To achieve this, a *shift* in the problem variables is performed before the new iterate is defined, i.e., Eq. (35) is replaced by $\boldsymbol{\xi}_{k+1} := S(\boldsymbol{\xi}_k + \Delta\boldsymbol{\xi}_k)$, where S is a shift operator which removes the variables of the first interval, shifts all variables by one interval, and appends new guesses on the last interval. For details, see [Die01].

Remark 3: For the related class of *shrinking horizon* problems (as defined in [BBB+01]), we can prove contractivity of the real-time algorithm under mild conditions, if plant and model coincide after an initial disturbance [Die01]. The contractivity result can conceptually be generalized to the shift strategy on long horizons, but in practice warm start and shift strategy show very similar performance [DBLS99].

Remark 4: The formulation of the constraints (24) in the NLP (23)-(28) can be considered an *initial value embedding* of each optimization problem into the manifold of perturbed problems. It allows a very efficient transition from one optimization problem to the next: let us for a moment assume that $\boldsymbol{\xi}_k$

is equal to the *exact* solution $\boldsymbol{\xi}_k^*$ of the optimization problem $P((\mathbf{x}_0)_k, (\mathbf{p}_0)_k)$, and that the above QP (29)-(34) is formulated with the *exact* Hessian blocks $\mathbf{H}_i^{\text{exact}}$. Then it can be shown under mild conditions that the next real-time iterate $\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k^* + \Delta\boldsymbol{\xi}_k$ is a first order prediction for the solution $\boldsymbol{\xi}_{k+1}^*$ of the optimization problem $P((\mathbf{x}_0)_{k+1}, (\mathbf{p}_0)_{k+1})$, i.e.

$$\|\boldsymbol{\xi}_{k+1} - \boldsymbol{\xi}_{k+1}^*\| = O\left(\left\| \begin{array}{c} (\mathbf{x}_0)_{k+1} - (\mathbf{x}_0)_k \\ (\mathbf{p}_0)_{k+1} - (\mathbf{p}_0)_k \end{array} \right\|^2 \right),$$

even at points where the active set changes [Die01]. In practice, the initial value embedding strategy ensures that the real-time iterates $\boldsymbol{\xi}_k$ stay close to the exact solutions $\boldsymbol{\xi}_k^*$, even if a Gauss-Newton Hessian is employed instead of the exact one.

3.1.1 Real-Time QP Generation and Solution:

The generation and solution of the structured quadratic program (29)-(34) are dovetailed in the real-time iteration scheme. The initial value embedding turns out to be crucial for the real-time performance, as it allows to prepare large parts of the QP solution without knowledge of $(\mathbf{x}_0)_{k+1}, (\mathbf{p}_0)_{k+1}$. The following steps are performed during each real-time iteration:

1. Reduction: Generate the equality constraints (32) by linearizing the consistency conditions (26), and resolve (32) to eliminate $\Delta\mathbf{s}_i^z$ from the problem ($\mathbf{G}_i^z$ is invertible due to the index one assumption). Similarly, generate (34) and resolve it to eliminate $\Delta\mathbf{x}_s, \Delta\mathbf{z}_s, \Delta\mathbf{u}_s$ from the problem (assuming that the square matrix $\mathbf{R}$ is invertible).

2. DAE solution and derivative generation: Solve the relaxed initial value problems (20)-(22) and compute simultaneously the directional derivatives of the trajectories $\mathbf{x}_i(t)$ $\mathbf{z}_i(t)$ in the reduced directions, using the principle of *internal numerical differentiation* (IND) as described by Bock [Boc81]. This yields the reduced version of (31).[2] Linearize also the constraints (27) and generate a reduced version of (33).

3. Gradient and Hessian generation: Compute a reduced version of the gradient integrals $\mathbf{h}_i$ and of the Gauss-Newton Hessian blocks $\mathbf{H}_i$. The sparse DAE solver DAESOL [Bau00] has been adapted to compute numerical approximations of reduced versions of $\mathbf{h}_i$ and $\mathbf{H}_i$ *simultaneously* with the sensitivity calculations from step 2, with negligible additional costs [Die01].

4. Condensing: Using the (reduced) linearized continuity conditions (31), eliminate the variables $\Delta\mathbf{s}_1^x, \ldots, \Delta\mathbf{s}_N^x$. Project the objective gradient and Hessian, as well as the linearized path constraints (33) onto the space of the remaining variables $\Delta\mathbf{s}_0^x$, $\Delta\mathbf{q}_0, \ldots, \Delta\mathbf{q}_{N-2}, \Delta\mathbf{p}$.

[2]Note that the full matrix $\mathbf{X}_i$ is *never* computed in this *partial reduction approach* that was developed by Leineweber [Lei99].

5. Step generation: take the current values $(\mathbf{x}_0)_{k+1}$, $(\mathbf{p}_0)_{k+1}$ from the state estimator, and eliminate $\Delta \mathbf{s}_0^x := (\mathbf{x}_0)_k - \mathbf{s}_0^x$ and $\Delta \mathbf{p} := (\mathbf{p}_0)_k - \mathbf{p}$, and generate a fully condensed QP in the variables $(\Delta \mathbf{q}_0, \dots, \Delta \mathbf{q}_{N-2})$. Solve this QP with an efficient dense QP solver using an active set strategy. Give the value $\mathbf{q}_0 + \Delta \mathbf{q}_0$ immediately as a control to the plant.

6. Expansion: Expand the solution to yield values for all variables $\Delta \boldsymbol{\xi}_k$ and perform the iteration $\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \Delta \boldsymbol{\xi}_k$. Go to 1.

Note that the computations of step 5 are in typical applications orders of magnitude shorter than the overall computations of one cycle. This means that the response delay of one sampling time, that is present in all previous NMPC optimization schemes, is practically avoided. It is interesting to observe that step 5 corresponds to the solution of a QP in *linear* MPC, which is based on a system linearization along the currently best predicted trajectory. Note, however, that this trajectory is updated after each iteration, and that the real-time iteration scheme maintains all advantages of a fully nonlinear treatment of the optimization problems.

The major computational costs for each real-time iteration, those of step 2, scale roughly linear with the number N of predicted control intervals.

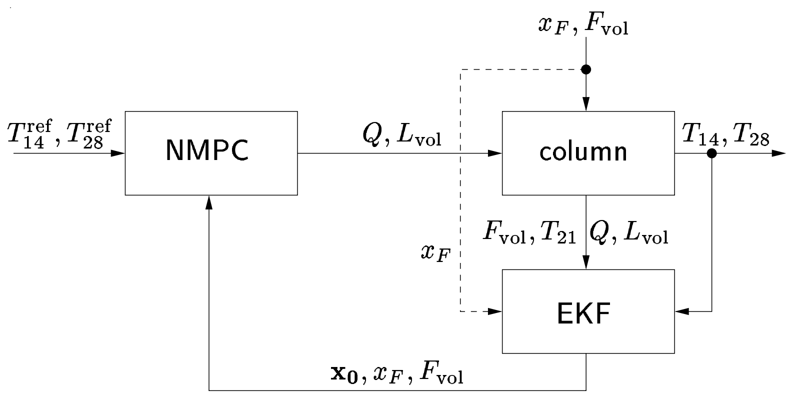
A comparison of the closed-loop behaviour of the real-time iteration strategy with a full iteration scheme, where each optimization problem is iterated to convergence, can be seen in Figure 7. In the shown example scenario even the full iteration scheme (that was started using the initial value embedding) was nearly always able to meet the limit of 10 seconds sampling time; the comparison shows no significant differences in the closed loop behaviour. Note, however, that bounds on the number of iterations for the full iteration scheme are difficult to establish.

4 Experimental Setup

In Section 5 we will demonstrate the real-time feasibility of the presented NMPC scheme. Furthermore, we give a short comparison of the achieved results with a conventional PI-controller. The purpose of the performance comparison is to show that NMPC does lead to reasonable performance without much tuning required. In the following, we shortly outline the two used controller setups.

4.1 NMPC Controller Setup

For the real-time application we implemented the presented NMPC scheme using the real-time optimization strategy given. For the NMPC setup we assume that the column is given in LV configuration, i.e., we use L_{vol} and the heat input Q into the boiler (which corresponds to the vapor flow out of the boiler) as manipulated variable. State estimates are obtained using a variant of an extended Kalman filter. Figure 3 shows the overall controller/plant/estimator setup. As described in Section 2, we use a inferential control scheme, i.e., the



x_F, F_{vol}
$T_{14}^{\mathrm{ref}}, T_{28}^{\mathrm{ref}}$
NMPC
Q, L_{vol}
column
T_{14}, T_{28}
F_{vol}, T_{21}
Q, L_{vol}
x_F
EKF
$\mathbf{x_0}, x_F, F_{\mathrm{vol}}$

Figure 3: Closed loop NMPC setup

product concentrations are not directly controlled, but instead the temperatures T_{14} and T_{28} which are directly measured at the column. The deviation of these temperatures from the desired reference temperatures T_{14}^{ref} and T_{28}^{ref} are weighted in the stage cost function, cf. Eq. (18).

4.1.1 State Estimation:

To obtain an estimate of the 42 differential system states and of the model parameter x_F we have implemented a variant of an Extended Kalman Filter (EKF). To improve the performance of the estimator, the temperature T_{21} is fed into the state estimator additionally to the controlled temperatures T_{14} and T_{28}. Together with the implemented controls (Q and L_{vol}), the measured feed flow rate F_{vol} is also given to the estimator. The usage of the *measured* heat input and reflux flow (in contrast to the optimization output) is necessary to overcome input disturbances, since both values can only be controlled indirectly giving the setpoints to the low-level control loops in the DCS.

The EKF is based on subsequent linearizations of the system model at each current estimate; each measurement is compared with the prediction of the nonlinear model, and the estimated state is corrected according to the deviation. The weight of past measurement information is kept in a weighting matrix, which is updated according to the current system linearization.

In contrast to an ordinary EKF the implemented estimator can incorporate additional knowledge about states and parameters in form of bounds. This is especially useful as the tray concentrations have to be in the interval $[0, 1]$ to make a reasonable DAE solution possible. For details, we refer to [Die01].

In Section 5 we will consider two scenarios for the "estimates" of the feed concentration disturbance. In the first scenario it is assumed that the time and value of the disturbance is known exactly and directly fed into the EKF. In the second scenario we consider the disturbance in the feed concentration x_F as unknown, i.e., the value of x_F is also estimated by the EKF.

4.1.2 Tuning of NMPC controller and EKF:

The NMPC controller and the EKF were tuned independently based on simulations and measured data, respectively. The EKF receives new measurements and provides new state estimates every 10 seconds, and the NMPC optimizer solves one optimization problem in this period. The control inputs on the control horizon were parameterized as piecewise constant, with 10 control intervals each of 120 seconds length, followed by a prediction interval of $T_p - T_c = 3600$ seconds with the inputs fixed to the steady state values $\mathbf{u}_s$ (cf. Sec. 3). Note that it does not cause any difficulty that the interval length of 120 seconds in the control horizon is *not* equal to the sampling time of 10 seconds.

4.1.3 Implementation of the NMPC setup:

The NMPC controller was implemented on a Unix workstation running under Linux. The open-loop optimization problem was solved on-line using our specially tailored version of MUSCOD2 as outlined in Section 3. The EKF was implemented using MATLAB and the integration routine DAESOL [Bau00] to obtain the necessary derivatives. The file transfer between the DCS and the workstation was done using a PC connected to the DCS from which the measured and manipulated variables could be read and written to via ftp, which was done every 10 seconds.

4.2 Configuration of the PI control loops

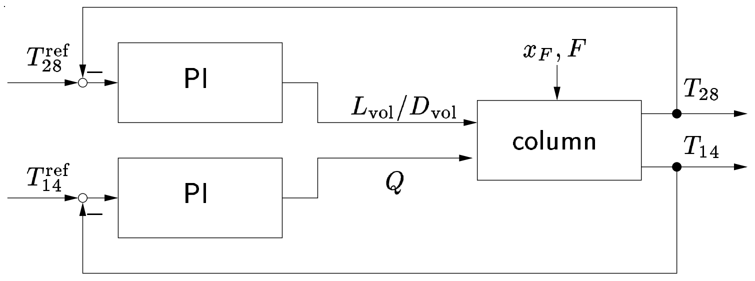


Figure 4: Closed loop PI setup

The conventional control scheme used consists of two decoupled single-input/single-output PI loops which where already implemented on the column. In contrast to the NMPC setup a L/D,V configuration is used. The controlled variables are, as in the NMPC case the temperatures on trays 14 and 28. The manipulated variables are the heat input Q to the boiler (corresponding to the liquid flow V out of the boiler) and the reflux ratio $L_{\text{vol}}/D_{\text{vol}}$. However, for comparisons we plot in section 5 the reflux flow rate for the decoupled PI as well as the NMPC controller.

4.2.1 PI Controller Tuning:

To achieve good control performance, the PI controllers are tuned as follows: In a first step, a Ziegeler-Nichols tuning based on the process reaction curve method is performed for each loop. This method was chosen since it is easy to apply and does only require single step tests. In a second step, the resulting PI controllers were detuned to compensate for the interactions between the control loops.

4.2.2 Implementation of the PI Setup:

The PI controllers were already implemented using the basic control function in the used DCS. The data collection was done using a PC connected to the DCS and the Unix workstations, compare Figure 1.

5 Experimental Results

In this section we present results on the computational demand and performance of the PI control setup and the NMPC controller using the dynamic optimization strategy outlined in Section 3. The main result here is an experimental proof-of-concept that NMPC can be used in real-time even for large scale models. The performance results given are only supposed to show that NMPC, without much further tuning, does lead to satisfying control performance.

5.1 Considered Disturbance Scenarios

The NMPC scheme and the decoupled PI controllers were tested on various scenarios. As scenarios we used a series of step changes in the feed flow rate (F_{vol}); a step change in the feed composition (x_F); and a short reflux breakdown ($L_{\text{vol}} = 0.5$). In the following figures the reflux flow rate, L_{vol}, is plotted for both NMPC and PI controllers although the corresponding manipulated variable is reflux ratio, ($L_{\text{vol}}/D_{\text{vol}}$), for the PI controller setup.

5.1.1 Feed Flow Change:

Figure 5 shows the controlled outputs (T_{28} and T_{14}) and input responses (L_{vol} and Q) when the feed flow rate, F_{vol}, is changed stepwise first -10 % at t= 0.57 h, then $+20$ % at t=1.16 h, finally -10 % at t=1.52 h. The plots on the left hand side show the results of NMPC and those on the right hand side belong to the PI controller. As can be seen, while there is no obvious difference between the two schemes in the first phase, the performance of NMPC is better than that of PI in the second and third phases. In the final phase, NMPC can handle with the corresponding load change in about 20 min. whereas the PI controller requires 40 min. In the case of PI controllers, the slightly higher

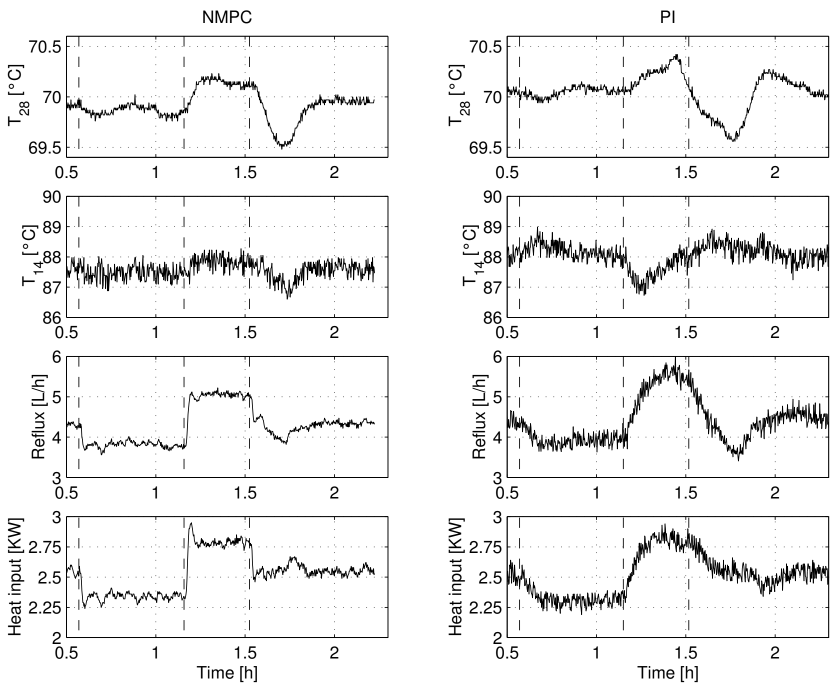


Figure 5: Comparison of closed loop NMPC and conventional PI controller performances for step disturbances in F_{vol}

deviation of the controlled variables from their set points can be explained by the fact that no feedforward disturbance information is used, whereas NMPC uses the disturbance information. This advantage of NMPC results in finding the optimum values for L_{vol} and Q at once while the manipulated variables move very slowly for PI controllers following the change in feedback measurements.

5.1.2 Feed Concentration Change:

For the next test, a step change in the feed composition is considered ; x_F is decreased from 0.320 to 0.272 at t=1.0 h. In Figure 6, the first column on the left hand side illustrates the results of NMPC which incorporates EKF estimating x_F. As can be seen, the controlled outputs (T_{14} and T_{28}) oscillate with small deviations from the set points but NMPC performance is still acceptable.

From the last two columns of Figure 6 it is evident that the performance of NMPC with known disturbance in x_F outperforms the PI controller in maintaining the controlled variable T_{28}.

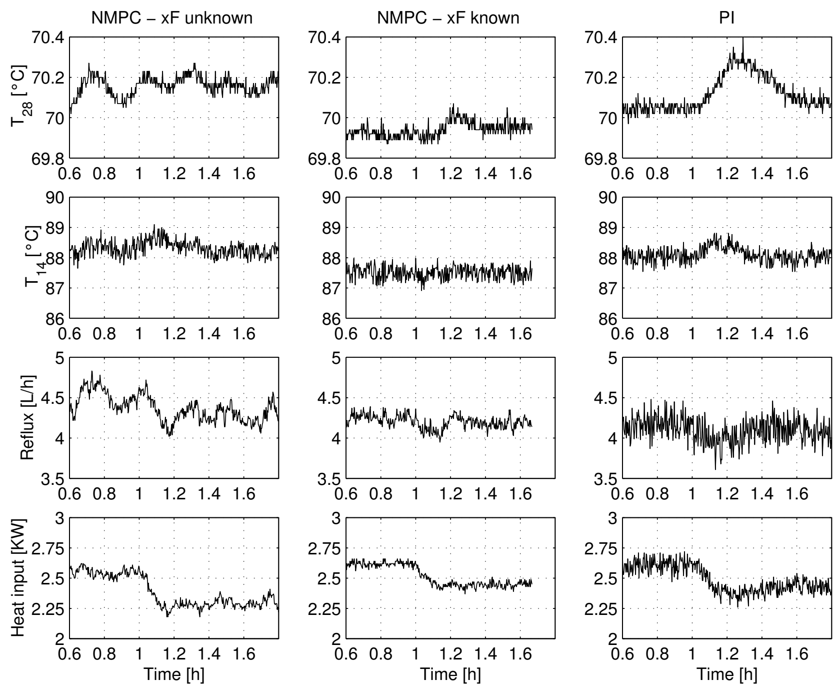


Figure 6: Step change in x_F: Comparison of closed loop NMPC – for unknown and known x_F – with the conventional PI controller performances

Comparison of real-time iterations with a full-iteration scheme: Figure 7 compares the computational and control performance of the real-time iteration scheme, that we used in all presented experiments, with a full iteration scheme, for the same scenario as in Fig. 6 (x_F known). The full iteration scheme was initialized by an initial value embedding, but then iterated until a prespecified convergence criterion was satisfied (KKT tolerance of 10^{-3}). If the optimization time exceeded the sampling time of 10 seconds, the old control was used for another sampling period.

It can be seen that the control performance is nearly identical in our example, but that the CPU times differ significantly. Note that the real-time iteration scheme had unused CPU capacity as the sampling time was fixed to 10 seconds in both schemes.

The CPU time variations in the real-time iterations are due to integrator adaptivity. The largest computation times occur for both schemes at the moment when the step change in x_F happens, which makes large changes in the predicted trajectory necessary.

Due to the unused capacity of the real-time iteration scheme, more detailed process models are computationally feasible for the same sampling time. First experimental tests with a larger and stiffer DAE model (82 differential and 122

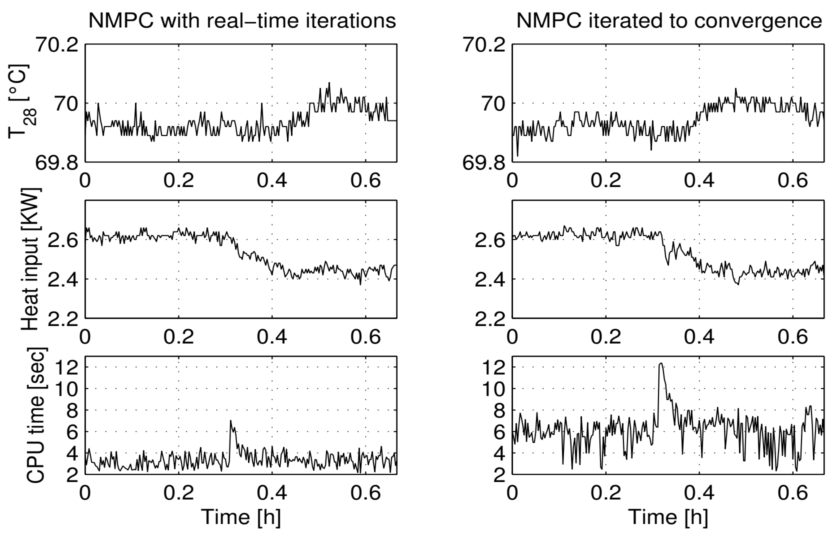


Figure 7: Performance and CPU time comparison of real-time iterations (cf. Fig. 6, x_F known) with full iterations to convergence, on a standard PC.

algebraic states) have shown that the corresponding NMPC controller is still real-time implementable [Die01].

5.1.3 Short Reflux Breakdown:

In the previous two cases the disturbing effects of load changes (in the feed flow and composition) on controlled variables are reasonably small. In order to have large disturbance effects on T_{14} and T_{28} we applied a reflux flow breakdown on the system starting at 0.22 h. for a short period of time (approximately 4 minutes). After the reflux flow is switched on again, NMPC and PI controller are able to bring the T_{14} and T_{28} back to their set points within 1 hour. However, both controllers lead to slight oscillations in the closed loop.

It is interesting to note that the PI controller of T_{28} shows an aggressive response during the initial transition period. We have experienced that if the reflux breakdown is applied for a longer period of time, the PI controller of T_{28} becomes completely unstable.

5.2 Discussion of Computational Demand and Closed-loop Performance

The presented experiments show that NMPC can handle the control problem satisfactorily while being real-time implementable. The results are comparable to the performance of (existing) PI controllers for a moderate range of disturbances.

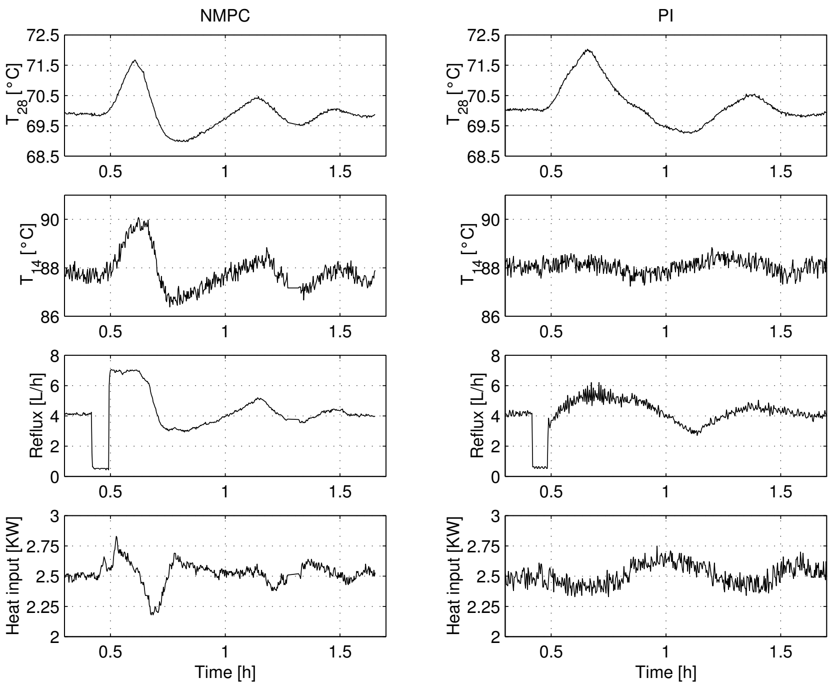


Figure 8: Comparison of closed loop NMPC and conventional PI controller performances for a short time reflux (L_{vol}) breakdown

Due to the efficiency of our real-time optimization scheme a more complex model would still be feasible for on-line implementation.

One of the challenges to be solved is to improve the state estimation and disturbance detection, as load changes which are unlikely to be measured in an industrial application, like x_F, need to be estimated for a realistic NMPC scheme. As can be seen in Figure 6, the closed loop performance when x_F is estimated by the EKF could still be improved.

It is the aim of future work to show the benefits of real-time NMPC applications for more complex processes or for a wide range of operating conditions like start-up periods of chemical processes.

6 Conclusions

We have presented an experimental proof-of-concept of the application of NMPC to the control of a high purity binary distillation column.

An efficient real-time optimization scheme based on the direct multiple shooting method is described. Among its features are an initial value embedding strategy, that allows to immediately respond to disturbances, and real-time iterations, that dovetail the optimization iterations with the real process development. This approach makes sampling times of 10 seconds for a system model of 164th order possible on a standard PC.

Our study shows that real-time implementation of NMPC using large scale DAE models is feasible for the control of a pilot scale distillation column, if efficient numerical optimization techniques are used.

References

[ABQ$^+$99] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright. Nonlinear predictive control and moving horizon estimation – An introductory overview. In P. M. Frank, editor, *Advances in Control, Highlights of ECC'99*, pages 391–449. Springer, 1999.

[AR92] F. Allgöwer and J. Raisch. Multivariable controller design for an industrial distillation column. In N.K Nichols and D.H. Owens, editors, *The Mathematics of Control Theory.* Clarendon Press, Oxford, 1992.

[Bau00] I. Bauer. *Numerische Verfahren zur Lösung von Anfangswertaufgaben und zur Generierung von ersten und zweiten Ableitungen mit Anwendungen bei Optimierungsaufgaben in Chemie und Verfahrenstechnik.* PhD thesis, University of Heidelberg, 2000.

[BBB$^+$01] T. Binder, L. Blank, H.G. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kronseder, W.Marquardt, J.P. Schlöder, and O. v.Stryk. Introduction to model based optimization of chemical processes on moving horizons. In *to be announced.* Springer, 2001.

[BDLS00] H.G. Bock, M. Diehl, D. Leineweber, and J. Schlöder. A direct multiple shooting method for real-time optimization of nonlinear DAE processes. In F. Allgöwer and A. Zheng, editors, *Nonlinear Predictive Control*, pages 245–268. Birkhäuser, 2000.

[BDS$^+$00] H.G. Bock, M. Diehl, J.P. Schlöder, F. Allgöwer, R. Findeisen, and Z. Nagy. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. In *ADCHEM2000 - International Symposium on Advanced Control of Chemical Processes*, volume 2, pages 695–703, Pisa, 2000.

[BES88] H.G. Bock, E. Eich, and J.P. Schlöder. Numerical solution of constrained least squares boundary value problems in differential-algebraic equations. In K. Strehmel, editor, *Numerical Treatment of Differential Equations.* Teubner, Leipzig, 1988.

[Bie00] L. Biegler. Efficient solution of dynamic optimization and NMPC problems. In F. Allgöwer and A. Zheng, editors, *Nonlinear Predictive Control.* Birkhäuser, 2000.

[Boc81] H.G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In K. H. Ebert, P. Deuflhard, and W. Jäger, editors, *Modelling of Chemical Reaction Systems*, volume 18 of *Springer Series in Chemical Physics*. Springer, Heidelberg, 1981.

[BP84] H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proc. 9th IFAC World Congress*, Budapest, 1984.

[CA98] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1218, 1998.

[DBLS99] M. Diehl, H.G. Bock, D.B. Leineweber, and J.P. Schlöder. Efficient direct multiple shooting in nonlinear model predictive control. In F. Keil, W. Mackens, H. Voß, and J. Werther, editors, *Scientific Computing in Chemical Engineering II*, volume 2, pages 218–227, Berlin, 1999. Springer.

[DBS$^+$01] M. Diehl, H.G. Bock, J.P. Schloeder, R. Findeisen, Z. Nagy, and F. Allgoewer. Real-time optimization and nonlinear model predictive control of processes governed by large-scale DAE. *J. Proc. Contr.*, 2001.

[Die01] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, University of Heidelberg, 2001.

[DMS96] G. De Nicolao, L. Magni, and R. Scattolini. Stabilizing nonlinear receding horizon control via a nonquadratic terminal state penalty. In *Symposium on Control, Optimization and Supervision, CESA'96 IMACS Multiconference*, pages 185–187, Lille, 1996.

[DMS00] G. De Nicolao, L. Magni, and R. Scattolini. Stability and robustness of nonlinear receding horizon control. In F. Allgöwer and A. Zheng, editors, *Nonlinear Predictive Control*, pages 3–23. Birkhäuser, 2000.

[FAD$^+$00] R. Findeisen, F. Allgöwer, M. Diehl, H.G. Bock, J.P. Schlöder, and Z. Nagy. Efficient nonlinear model predictive control. Submitted to 6th International Conference on Chemical Process Control – CPC VI, 2000.

[HLM$^+$01] R. Henrion, P. Li, A. Moeller, M. Wendt, and G. Wozny. Optimal control of a continuous distillation process under probabilistic constraints. In M. Groetschel, S.O. Krumke, and J. Rambau, editors, *Online Optimization of Large Scale Systems: State of the Art*. Springer, 2001.

[Lei99] D.B. Leineweber. *Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models*, volume 613 of *Fortschr.-Ber. VDI Reihe 3, Verfahrenstechnik*. VDI Verlag, Düsseldorf, 1999.

[Pli81] K.J. Plitt. Ein superlinear konvergentes Mehrzielverfahren zur direkten Berechnung beschränkter optimaler Steuerungen. Master's thesis, University of Bonn, 1981.

[SBS98] V.H. Schulz, H.G. Bock, and M.C. Steinbach. Exploiting invariants in the numerical solution of multipoint boundary value problems for daes. *SIAM J. Sci. Comp.*, 19:440–467, 1998.

[Wri96] S. J. Wright. Applying new optimization algorithms to model predictive control. In J.C. Kantor, C.E. Garcia, and B. Carnahan, editors, *Fifth International Conference on Chemical Process Control – CPC V*, pages 147–155. American Institute of Chemical Engineers, 1996.