Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Efficient Robust Economic Nonlinear Model Predictive Control of an Industrial Batch Reactor

Sergio Lucia * **Joel A.E. Andersson** ** **Heiko Brandt** * **Ala Bouaswaig** ***
Moritz Diehl ** **Sebastian Engell** *

* *Process Dynamics and Operations Group, TU Dortmund, 44227 Dortmund, Germany (e-mail: {sergio.lucia, sebastian.engell, heiko.brandt} @bci.tu-dortmund.de).*
** *OPTEC Optimization in Engineering Center, Electrical Engineering Department ESAT-STADIUS, KU Leuven, 3001 Leuven, Belgium (e-mail: {joel.andersson, moritz.diehl}@esat.kuleuven.be).*
*** *BASF SE, 67056 Ludwigshafen, Germany (e-mail: {ala.bouaswaig@basf.com.}*

Abstract: In this paper we present a systematic and efficient approach to deal with uncertainty in Nonlinear Model Predictive Control (NMPC). The main idea of the approach is to represent the NMPC setting as a real-time decision problem under uncertainty that is formulated as a multi-stage stochastic problem with recourse, based on a description of the uncertainty by a scenario tree. This formulation explicitly takes into account the fact that new information will be available in the future and thus reduces the conservativeness compared to open-loop worst-case approaches. We show that the proposed multi-stage NMPC formulation can deal with significant plant-model mismatch as it is usually encountered in the process industry and still satisfies tight constraints for the different values of the uncertain parameters, in contrast to standard NMPC. The use of an economic cost function leads to a superior performance compared to the standard tracking formulation. The potential of the approach is demonstrated for an industrial case study provided by BASF SE in the context of the European Project EMBOCON. The numerical solution of the resulting large optimization problems is implemented using the optimization framework CasADi.

1. INTRODUCTION

Model Predictive Control (MPC) is a popular control strategy especially in the process industry because it can deal with multivariate systems and constraints on states and inputs can be formulated easily. The control task can be directly formulated in terms of the online optimization of a relevant performance criterion instead of set-point tracking of certain variables leading to superior economic performance (Engell [2007]). The stability and the performance of MPC and of its nonlinear counterpart (NMPC) are strongly influenced by the accuracy of the model that is used for the predictions. Most industrial processes cannot be described exactly and are subject to disturbances and therefore robust MPC approaches have been studied during last years. One of the first robust MPC approaches was min-max open-loop MPC (Campo and Morari [1987]) which optimizes a sequence of control inputs with respect to the cost for the worst-case realization of the uncertainty. This is an open-loop formulation that ignores the presence of feedback that enables to react to the realization of the uncertainty. Therefore this approach is known to provide conservative results. Closed-loop (or feedback) min-max MPC was proposed in Witsenhausen [1968] and in Scokaert and Mayne [1998], where an optimization over a sequence of control policies is performed. This results however in an infinite-dimensional optimization problem that is very difficult to solve unless the structure of the policy is fixed a priori as in Hadjiyiannis et al. [2011], leading to suboptimality. In this paper we propose the use of multi-stage NMPC under the assumption that the evolution of the uncertainty can be represented as a tree of discrete scenarios, as in Scokaert and Mayne [1998]. The representation by scenarios enables the computation of the next inputs taking into account the adaptation of the future inputs to the new information that will be gained at later points in time (i.e recourse). This approach has been shown to lead to excellent performance Lucia et al. [2013], Lucia and Engell [2013]. The price for this gain in performance (compared to a min-max formulation) and robustness (compared to NMPC based only upon a nominal model) is that the size of the resulting optimization problem grows exponentially with the prediction horizon and with the number of uncertainties. For the linear case, Muñoz de la Peña et al. [2005] also proposed to use multi-stage optimization in MPC. The contribution of this paper is to show that multi-stage NMPC can solve the online performance optimization problem for an industrial case-study provided by BASF with important uncertainties in a systematic and efficient manner. We also compare the standard set-point tracking strategy to the use of an economic cost function for this example. The second contribution is the efficient implementation of the approach using CasADi (Andersson et al.

⋆ The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number FP7-ICT-2009-4 248940 (EMBOCON) and from the Deutsche Forschungsgemeinschaft (DFG, German Research Council) under grant agreement number EN 152/39-1. Also acknowledged are the Research Council KUL (PFV/10/002, GOA/10/09 MaNet, GOA/10/11), the Flemish Government (IOF/KP/SCORES4CHEM), FWO (G.0320.08, G.0377.09); IWT (SBO LeCoPro), Belgian Federal Science Policy Office (IUAP P7) and EU (FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), Eurostars SMART, ACCM.

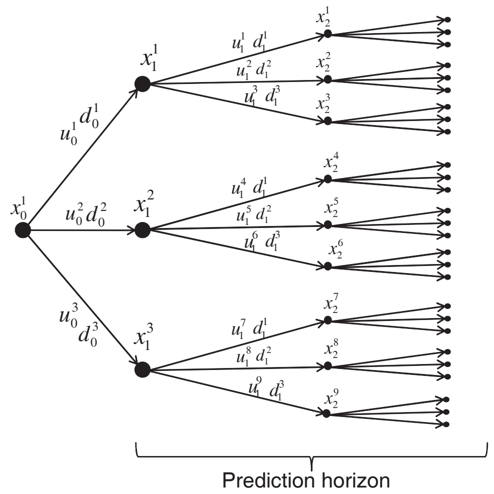

Fig. 1. Scenario tree representation of the uncertainty evolution for multi-stage NMPC.

[2012a]) leading to significantly faster solution times compared to the ones reported in Lucia et al. [2013]. Using CasADi, the proposed controller can be implemented satisfying the real-time requirements of the process. The remainder of the paper is organized as follows. The proposed multi-stage NMPC approach is introduced in Section 2. In Section 3, the implementation of the approach using CasADi is presented together with the main features of this framework. The model of the industrial batch reactor is presented in Section 4, while the corresponding control task and the optimization problem solved at each sampling time are described in Section 5. In Section 6 we present the main results of the paper, showing that multi-stage NMPC outperforms standard NMPC. The conclusions and future work are presented in Section 7.

2. MULTI-STAGE NONLINEAR MODEL PREDICTIVE CONTROL

Multi-stage NMPC is based on the description of the evolution of the uncertainties as a scenario tree (see Fig. 1). The future control inputs can depend on the previously observed values of the uncertainties, and act as recourse variables that counteract the effects of the uncertainties. This reduces the conservativeness of the approach significantly in comparison with other typical robust NMPC settings such as open-loop min-max NMPC or multi-scenario approaches (e.g. Huang and Biegler [2009]) where no feedback is introduced in the predictions. The constraints are guaranteed to be satisfied for all the values of the uncertainty that are considered in the tree. The scenario tree setting assumes a discrete-time formulation of an uncertain nonlinear system that can be written as:

$$x_{k+1}^{j} = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), \tag{1}$$

where each state x_{k+1}^j is a function of the previous state $x_k^{p(j)}$, the control input u_k^j and the realization r of the uncertainty at stage k, $d_k^{r(j)}$ (for example in Fig.1, $x_2^7 = f(x_1^3, u_1^7, d_1^1)$). For simplicity in the presentation we consider that the tree has the same number of branches at all nodes, given by $d_k^{r(j)} \in \{d_k^1, d_k^2 ... d_k^s\}$ at stage k for s different possible values of the uncertainty. Also, in order to clarify the notation, the index set of all occurring indices (j,k) is denoted by I. The control inputs cannot anticipate the value of the uncertainty at a certain point in time, and this is enforced by the non-anticipativity constraints that require all the control inputs that branch at the same node to be equal (for example in Fig. 1, $u_0^1 = u_0^2 = u_0^3$; $u_1^1 = u_1^2 = u_1^3$; ...). The optimization problem resulting from the multi-stage formulation in a scenario-based setting can be written as:

$$\min_{x_k^j, u_k^j \forall (j,k) \in I} \quad ||\tilde{J}(x_{k+1}^j, u_k^j)||_\alpha \tag{2}$$

subject to:

$$\begin{aligned}
x_{k+1}^j &= f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), && \forall\, (j, k+1) \in I, && \text{(3a)}\\
x_k^j &\in \mathbb{X}\,, && \forall\, (j,k) \in I, && \text{(3b)}\\
u_k^j &\in \mathbb{U}\,, && \forall\, (j,k) \in I, && \text{(3c)}\\
u_k^j &= u_k^l \text{ if } x_k^{p(j)} = x_k^{p(l)} && \forall\, (j,k),(l,k) \in I, && \text{(3d)}
\end{aligned}$$

where $\mathbb{X}$ and $\mathbb{U}$ represent simple bounds on state and control and $||\tilde{J}(x_{k+1}^j, u_k^j)||_\alpha$ is the α-norm of a vector $\tilde{J}(x_{k+1}^j, u_k^j)$ that contains in each component the cost of each scenario S_i multiplied by its probability ω_i, that is, $\tilde{J} = [\omega_1 \cdot J_1, ..., \omega_N \cdot J_N]$, for N different scenarios where $J_i(x_{k+1}^j, u_k^j)$ is the cost of each scenario defined as:

$$J_i(x_{k+1}^j, u_k^j) = \sum_{k=0}^{K-1} L(x_{k+1}^j, u_k^j), \ \ \forall\, x_{k+1}^j, u_k^j \in S_i, \tag{4}$$

where $L(x_{k+1}^j, u_k^j)$ is the stage cost. The non-anticipativity constraints in (3d) enforce that the decisions u_k^j with the same parent node $x_k^{p(j)}$ should be the same. With this general formulation, if the α-norm is chosen with $\alpha = 1$ the resulting optimization problem represents the multi-stage NMPC approach, whereas if $\alpha = \infty$ is chosen, a *closed-loop* min-max approach is obtained, in which feedback is taken explicitly into account. If the number of scenarios is $N = 1$, the problem is reduced to standard NMPC. The probabilities ω_i can be set a priori or be adapted according to information obtained from estimating the parameters if such information is available (see Lucia et al. [2013]). Note that for a general nonlinear system, robust constraint satisfaction cannot be guaranteed for the values that are not explicitly in the tree. However, the value of the parameters that produce the worst-case scenario are often on the boundaries of the considered parameter interval (Srinivasan et al. [2002]) and therefore the scenario tree should include the combinations of the extreme values of the parameters, which produces good results in practice (see Lucia et al. [2013] for a more detailed discussion on the choice of a suitable scenario tree).

3. IMPLEMENTATION DETAILS

Multi-stage NMPC has been implemented in this work using the optimization tool CasADi (Andersson et al. [2012a] and Andersson et al. [2012b]). CasADi is an open-source framework for C++ and Python for numerical optimization in general and optimal control in particular. The main idea of the tool is to provide users with the ability to easily and efficiently implement optimal control algorithms using a wide range of numerical formulations and solution techniques, including the use of multiple shooting and collocation, rather than offering just one Optimal Control Problem (OCP) solver. In particular, using CasADi the user can construct a very general high-level symbolic representation of the nonlinear programming problem. This representation, which internally is represented as an expression graph, may contain millions of operations and contain calls to such operations as ODE/DAE integrators. The NLP can be solved by any existing NLP solver or, alternatively, be solved by a custom written NLP solver based on one of the quadratic programming solvers that are coupled to CasADi. The solver interfaces rely on CasADi to efficiently calculate derivative information. From the symbolic representation of the NLP cost function and constraint function, CasADi automatically generates symbolic expressions for the Jacobian of the constraints and the Hessian of the Lagrangian function. This generation is done using a state-of-the-art implementation of algorithmic-differentiation (Griewank and Walther [2008]) on a general expression graph made up of matrix-valued atomic operations. The Jacobian and Hessian generation is based upon efficient automatic generation of the sparsity patterns before graph coloring techniques are applied to find an efficient way to calculate the Jacobian or Hessian. When ODE/DAE integrators are included in the symbolic expressions, forward and adjoint sensitivity analysis (Cao et al. [2003]) will be automatically invoked. By relieving the user from the tedious and error-prone work of calculating and passing derivative information and performing ODE/DAE sensitivity analysis, and by offering a convenient working environment in a high-level language Python, the use of CasADi reduces the effort needed to implement an optimal control solution significantly without compromising the efficiency of the solution. The results reported in Andersson et al. [2012a] show that CasADi performs faster than other state-of-the-art software such as the AMPL Solver Library (ASL).

4. MODEL OF THE SEMI-BATCH REACTOR

The case-study under consideration concerns an industrial batch polymerization reactor that is equipped with a jacket and an external heat exchanger that are used to cool or to heat the reactor. The dynamics of the process are described by the following set of ordinary differential equations:

$$
\begin{aligned}
\dot{m}_{\mathrm{W}} &= \dot{m}_{\mathrm{W,F}} \\
\dot{m}_{\mathrm{A}} &= \dot{m}_{\mathrm{A,F}} - k_{\mathrm{R1}}\, m_{\mathrm{A,R}} - p_1\, k_{\mathrm{R2}}\, m_{\mathrm{AWT}}\, m_{\mathrm{A}}/m_{\mathrm{ges}}, \\
\dot{m}_{\mathrm{P}} &= k_{\mathrm{R1}}\, m_{\mathrm{A,R}} + p_1\, k_{\mathrm{R2}}\, m_{\mathrm{AWT}}\, m_{\mathrm{A}}/m_{\mathrm{ges}}, \\
\dot{T}_{\mathrm{R}} &= 1/(c_{\mathrm{p,R}} m_{\mathrm{ges}})\, [\dot{m}_{\mathrm{F}}\, c_{\mathrm{p,F}}\, (T_{\mathrm{F}} - T_{\mathrm{R}}) + \Delta H_{\mathrm{R}} k_{\mathrm{R1}} m_{\mathrm{A,R}} \\
&\quad - k_{\mathrm{K}} A\, (T_{\mathrm{R}} - T_{\mathrm{S}}) - \dot{m}_{\mathrm{AWT}}\, c_{\mathrm{p,R}}\, (T_{\mathrm{R}} - T_{\mathrm{EK}})], \\
\dot{T}_S &= 1/(c_{\mathrm{p,S}} m_{\mathrm{S}})\, [k_{\mathrm{K}} A\, (T_{\mathrm{R}} - T_{\mathrm{S}}) - k_{\mathrm{K}} A\, (T_{\mathrm{S}} - T_{\mathrm{M}})], \\
\dot{T}_{\mathrm{M}} &= 1/(c_{\mathrm{p,W}} m_{\mathrm{M,KW}})\, [\dot{m}_{\mathrm{M,KW}}\, c_{\mathrm{p,W}}\, \left(T_{\mathrm{M}}^{\mathrm{IN}} - T_{\mathrm{M}}\right) \\
&\quad + k_{\mathrm{K}} A\, (T_{\mathrm{S}} - T_{\mathrm{M}})], \\
\dot{T}_{\mathrm{EK}} &= 1/(c_{\mathrm{p,R}} m_{\mathrm{AWT}})\, [\dot{m}_{\mathrm{AWT}} c_{\mathrm{p,W}}\, (T_{\mathrm{R}} - T_{\mathrm{EK}}) \\
&\quad - \alpha\, (T_{\mathrm{EK}} - T_{\mathrm{AWT}}) + p_1\, k_{\mathrm{R2}}\, m_{\mathrm{A}}\, m_{\mathrm{AWT}} \Delta H_{\mathrm{R}}/m_{\mathrm{ges}}], \qquad (5) \\
\dot{T}_{\mathrm{AWT}} &= [\dot{m}_{\mathrm{AWT,KW}}\, c_{\mathrm{p,W}}\, (T_{\mathrm{AWT}}^{\mathrm{IN}} - T_{\mathrm{AWT}}) \\
&\quad - \alpha\, (T_{\mathrm{AWT}} - T_{\mathrm{EK}})]/(c_{\mathrm{p,W}} m_{\mathrm{AWT,KW}}),
\end{aligned}
$$

where:

$$
\begin{aligned}
U &= m_{\mathrm{P}}/(m_{\mathrm{A}} + m_{\mathrm{P}}), \\
m_{\mathrm{ges}} &= m_{\mathrm{W}} + m_{\mathrm{A}} + m_{\mathrm{P}}, \\
k_{\mathrm{R1}} &= k_0 e^{\frac{-E_a}{R T_{\mathrm{R}}}}\, (k_{\mathrm{U1}}\, (1 - U) + k_{\mathrm{U2}} U)\,, \\
k_{\mathrm{R2}} &= k_0 e^{\frac{-E_a}{R T_{\mathrm{EK}}}}\, (k_{\mathrm{U1}}\, (1 - U) + k_{\mathrm{U2}} U)\,, \\
k_{\mathrm{K}} &= (m_{\mathrm{W}}\, k_{\mathrm{WS}} + m_{\mathrm{A}}\, k_{\mathrm{AS}} + m_{\mathrm{P}}\, k_{\mathrm{PS}})/m_{\mathrm{ges}}, \\
m_{\mathrm{A,R}} &= m_{\mathrm{A}} - m_{\mathrm{A}} m_{\mathrm{AWT}}/m_{\mathrm{ges}}.
\end{aligned}
$$

The model includes mass balances for the hold-ups of water, monomer feed and product (m_{W}, m_{A}, m_{P}) and energy balances for the reactor (T_{R}), the vessel (T_{S}), the jacket (T_{M}), the product leaving the external heat exchanger (T_{EK}) and the coolant leaving the external heat exchanger (T_{AWT}). The available control inputs are the feed flow $\dot{m}_{\mathrm{F}}$, the coolant temperature at the inlet of the jacket $T_{\mathrm{M}}^{\mathrm{IN}}$ and the coolant temperature at the inlet of the external heat exchanger $T_{\mathrm{AWT}}^{\mathrm{IN}}$. Realistic model parameters have been provided by BASF SE.

5. CONTROL PROBLEM

The control task under consideration is the production of one batch of polymer in the minimum possible time satisfying tight temperature constraints that are essential for meeting the quality specifications of the product. We consider that the reactor temperature T_{R} should be in a range of $\pm 1.5°$C around the desired reaction temperature $T_{\mathrm{set}} = 90°$C. The constraints that have to be satisfied for the different states together with the initial conditions are summarized in Table 1 and the constraints for the control inputs are stated in Table. 2. The maximum amount of material that can be fed into the reactor is $\int \dot{m}_{\mathrm{F}}\, \mathrm{d}t = 30000$ kg and the batch is considered to be finished when the total amount of polymer produced is $m_{\mathrm{P}}^{\mathrm{end}} = 20680$ kg. The control task must be achieved even under the presence of uncertainties of some critical model parameters. For the study presented here, we consider the parameters ΔH_{R}, that determines the heat generated by the polymerization reaction and k_0, that determines the speed of the reaction to be uncertain. Even small changes of these parameters have a large influence on both the batch time and the reactor temperature that has to be maintained between tight bounds. In order to model the end of the monomer feeding phase in the prediction of the NMPC controller, we extend the model presented in the previous section by a new differential state that accounts for the total amount of material that has been fed until a certain point in time, that is, $\dot{m}_{\mathrm{A}}^{\mathrm{acc}} = \dot{m}_{\mathrm{F}}$ and a constraint is included such that $0 < m_{\mathrm{A}}^{\mathrm{acc}} < m_{\mathrm{F}}^{\mathrm{max}} = 30000$ kg. By doing this, it is not necessary to divide the control task into different phases (feeding phase, holding phase...), simplifying the implementation and avoiding the need for switching between different cost functions. The optimization problem that has to be solved at each sampling time can be written as:

Table 1. Initial conditions and state constraints

States	Init. cond.	Min.	Max.	Unit
m_{W}	10000	0	inf.	[kg]
m_{A}	853	0	inf.	[kg]
m_{P}	26.5	0	inf.	[kg]
T_{R}	90.0	$T_{\mathrm{set}} - 1.5$	$T_{\mathrm{set}} + 1.5$	[°C]
T_{S}	90.0	0	100	[°C]
T_{M}	90.0	0	100	[°C]
T_{EK}	35.0	0	100	[°C]
T_{AWT}	35.0	0	100	[°C]

Table 2. Bounds on manipulated variables

Control	Min.	Max.	Unit
$\dot{m}_{\mathrm{F}}$	0	3×10^4	$\left[\frac{\mathrm{kg}}{\mathrm{h}}\right]$
$T_{\mathrm{M}}^{\mathrm{IN}}$	60	100	[°C]
$T_{\mathrm{AWT}}^{\mathrm{IN}}$	60	100	[°C]

$$\min_{x_k^j, u_k^j \forall (j,k) \in I} \qquad J_{\mathrm{batch}}(x_{k+1}^j, u_k^j) \qquad (6)$$

subject to:

$$x_{k+1}^j = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), \qquad (7a)$$

$$T_{\text{set}} - 1.5 \leq T^j_{\text{R},k} \leq T_{\text{set}} + 1.5, \tag{7b}$$
$$0 \leq T^j_{\text{S},k} \leq 100, \tag{7c}$$
$$0 \leq T^j_{\text{M},k} \leq 100, \tag{7d}$$
$$0 \leq T^j_{\text{EK},k} \leq 100, \tag{7e}$$
$$0 \leq T^j_{\text{AWT},k} \leq 100, \tag{7f}$$
$$0 \leq m^{\text{acc},j}_{A,k} \leq m^{\text{max}}_{A}, \tag{7g}$$
$$0 \leq \dot{m}^j_{\text{F},k} \leq 3 \times 10^4, \tag{7h}$$
$$60 \leq T^{\text{IN},j}_{M,k} \leq 100, \tag{7i}$$
$$60 \leq T^{\text{IN},j}_{AWT,k} \leq 100, \tag{7j}$$
$$u^j_k = u^l_k \text{ if } x^{p(j)}_k = x^{p(l)}_k \quad \forall\,(j,k),(l,k) \in I, \tag{7k}$$

where all the constraints are applied to all the states ($x^j_k = [m^j_{\text{W},k}, m^j_{\text{A},k}, m^j_{\text{P},k}, T^j_{\text{R},k}, T^j_{\text{S},k}, T^j_{\text{M},k}, T^j_{\text{EK},k}, T^j_{\text{AWT},k}]^{\text{T}}$) and control inputs ($u^j_k = [\dot{m}^j_{\text{F},k}, T^{\text{IN},j}_{\text{M},k}, T^{\text{IN},j}_{\text{AWT},k}]^{\text{T}}$) in the scenario tree. (7a) represents the discretized dynamics of the system, (7b-7g) denote the constraints on the states, (7h-7j) represent the constraints on the input variables, and the non-ancitipativity constraints are included in (7k).

We compare two possible cost functions $J_{\text{batch}}(x^j_{k+1}, u^j_k)$. The first cost function, given by (8), represents the maximization of the mass of polymer produced during the prediction interval together with a set-point tracking term for the reactor temperature. Small regularization terms are added in order to penalize the control movements so that a smooth control input is achieved. The tracking cost function can be written as:

$$\begin{aligned} J_{\text{track}}(x^j_{k+1}, u^j_k) = \sum_{i=1}^{N} \omega_i \sum_{k=0}^{K-1} & -m^j_{\text{P},k+1} + q(T^j_{\text{R},k+1} - T_{\text{set}})^2 \\ & + r_1(\Delta \dot{m}^j_{\text{F},k})^2 + r_2(\Delta T^{\text{IN},j}_{\text{M},k})^2 + r_3(\Delta T^{\text{IN},j}_{\text{AWT},k})^2 \\ & \forall\, m^j_{\text{P},k+1}, T^j_{\text{R},k+1}, \dot{m}^j_{\text{F},k}, T^{\text{IN},j}_{\text{M},k}, T^{\text{IN},j}_{\text{AWT},k} \in S_i, \end{aligned} \tag{8}$$

where q, r_1, r_2 and r_3 are tuning parameters. An alternative is to avoid the use of a tracking term and to directly formulate an economic cost function, that is defined as follows:

$$\begin{aligned} J_{\text{eco}}(x^j_{k+1}, u^j_k) = \sum_{i=1}^{N} \omega_i \sum_{k=0}^{K-1} & -m^j_{\text{P},k+1} \\ & + r_1(\Delta \dot{m}^j_{\text{F},k})^2 + r_2(\Delta T^{\text{IN},j}_{\text{M},k})^2 + r_3(\Delta T^{\text{IN},j}_{\text{AWT},k})^2 \\ & \forall\, m^j_{\text{P},k+1}, \dot{m}^j_{\text{F},k}, T^{\text{IN},j}_{\text{M},k}, T^{\text{IN},j}_{\text{AWT},k} \in S_i. \end{aligned} \tag{9}$$

In both cases the cost is calculated as the sum over all the N scenarios S_i along the prediction horizon K. We thus approximate the time-optimal control problem by a maximization of the amount of polymer produced in the reactor, which has been proven by simulations to have a very similar performance compared to the original minimum time problem. For the application of multi-stage NMPC to the industrial batch polymerization reactor presented in the previous section, we consider a scenario tree that branches only in the first stage and then the uncertainty is considered to remain constant. This simplification has been demonstrated to provide good results even for time varying uncertainties (Lucia and Engell [2013]) and results in smaller computation times. Thus the scenario tree used to generate all the results below with multi-stage NMPC has 9 scenarios, that include the combinations of the maximum, minimum and nominal values of the 2 uncertain parameters, ΔH_{R} and k_0 with a long prediction horizon $K = 30$. The implementation in CasADi makes it possible to compare the solution of the multi-stage NMPC problem using direct multiple-shooting or direct collocation for the discretization of the ODEs. For the results presented with a collocation approach we use Radau collocation points, with interpolating polynomials of order 2 and using 2 collocation intervals for each control discretization. For the multiple-shooting approach we use CasADi's interface to SUNDIALS (Hindmarsh et al. [2005]).

The sampling time of the controller is $t_{\text{s}} = 60$ s. with a prediction horizon of $K = 30$ steps. We assume that full state feedback is available at each sampling time. The tuning parameters in the cost functions (8-9) have the same values for all the simulations and they are $r_1 = 0.1$, $r_2 = 0.02$, $r_3 = 0.01$ and $q = 10000$. The resulting nonlinear programs (NLPs) are solved by IPOPT (Wächter and Biegler [2006]) using exact Hessians. All the optimization problems are solved on a standard laptop with an Intel i-5 processor at 2.30GHz running Ubuntu on a virtual machine with one core and 2 GB of RAM.

6. RESULTS

If the model is assumed to be perfect, i.e. there is no plant-model mismatch, it is possible to solve the problem described above with standard NMPC, using either the economic cost function ($J_{\text{eco}}(x^j_{k+1}, u^j_k)$) or the cost function that includes a classical set-point tracking term ($J_{\text{track}}(x^j_{k+1}, u^j_k)$). As expected, the use of an economic cost function reduces the batch time since the reactor can work at a higher temperature. In this case the batch time reduction is around a 6.5%, as can be seen in Fig. 2, where we also show a comparison between the solution using collocation (solid lines) and multiple shooting (dashed lines). Both provide very similar results and for the remainder of the paper only the results with the collocation approach are shown.

When using an economic cost function, the process is typically operated at one of its constraints. Since models for industrial applications are imperfect, constraint violations will therefore be unavoidable, if no additional measures are taken. The use of a tracking term is often used as a way to deal with the inaccuracy of the model. This is however not enough to handle significant uncertainties. Fig. 3 shows different simulations of the standard NMPC controller when the parameters ΔH_{R} and k_0 are uncertain and vary by $\pm 15\%$ with respect to their nominal values, but are considered to remain constant during a batch. Each trajectory shows the result of the NMPC controller for a different scenario. As it can be seen, the NMPC controller with a tracking term in the cost function cannot satisfy the constraints for all the possible scenarios. A simple modification to increase the amount of feedback that is used in the standard NMPC scheme with a tracking term is to introduce a so called bias-term. By doing this, at each sampling time the set point used in the optimizer ($T^{\text{opt}}_{\text{set}}$) is updated using a proportional rule, i.e., $T^{\text{opt}}_{\text{set}} \leftarrow T^{\text{opt}}_{\text{set}} + K(T_{\text{set}} - T_{\text{R}})$, with $K = 0.015$ where T_{set} is the real setpoint and T_{R} is the state of the real plant. The performance of the controller is slightly better than without the use of a bias term, but constraint violations still occur. After analyzing the results, it can be concluded that constraint violations occur for the scenarios with the maximum value of the parameter ΔH_R. Therefore, in order to have a

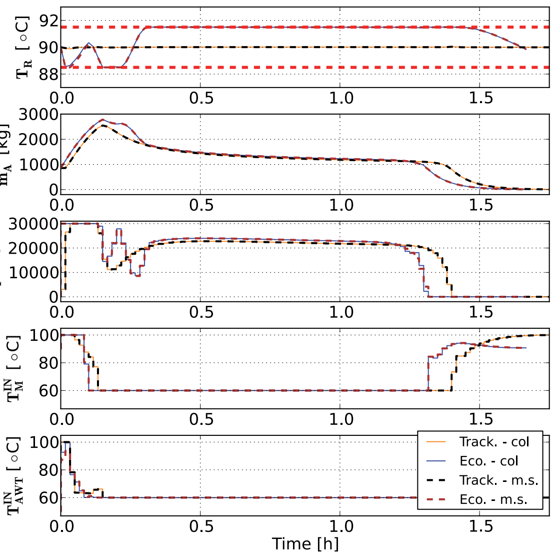

Fig. 2. Reactor temperature (with constraints indicated), monomer mass, and control inputs for standard NMPC with a tracking term in the cost function and with an economic cost function using collocation (solid lines) and multiple shooting (dashed lines) in the case of no plant-model mismatch.

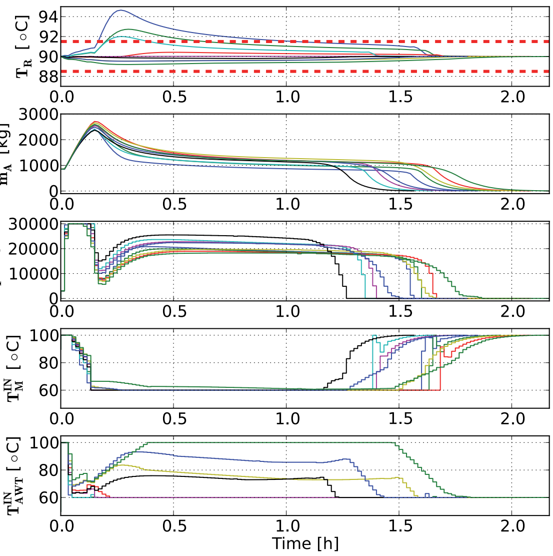

Fig. 3. Reactor temperature, monomer mass, and control inputs for standard NMPC with tracking term. The different trajectories show the different scenarios of the uncertainty with mismatches of $\pm 15\%$ with respect to their nominal values.

fair comparison with the proposed multi-stage NMPC, we show also simulations (see Fig. 4) of standard NMPC with a tracking and a bias term in which the model used in the optimizer has the worst-case values for the uncertain parameters, i.e. $\Delta H_{\mathrm{R}} = 1.15 \cdot \Delta H_{\mathrm{R}}^{\mathrm{nom}}$ and $k_0 = 1.15 \cdot k_0^{\mathrm{nom}}$. This makes the controller act more conservatively, leading to the satisfaction of the constraints for all the scenarios. However, the resulting batch times are much higher than the ones obtained with the proposed multi-stage NMPC with economic cost function, that are shown in Fig. 5.

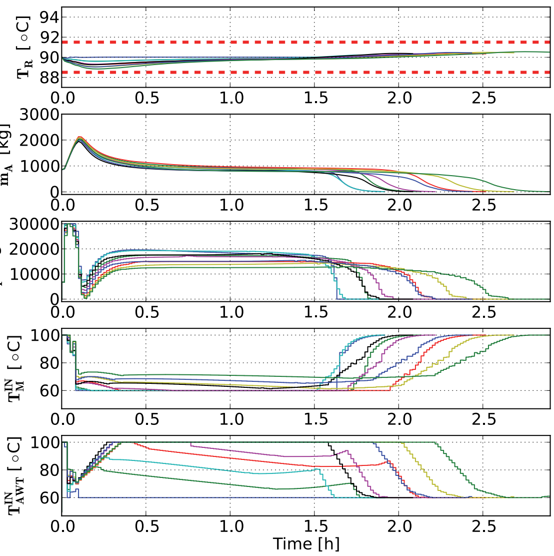

Fig. 4. Reactor temperature, monomer mass, and control inputs for standard NMPC with tracking and bias term using the worst-case values of the uncertain parameters in the model of the optimizer. The different trajectories show the different scenarios of the uncertainty with mismatches of $\pm 15\%$ with respect to their nominal values.

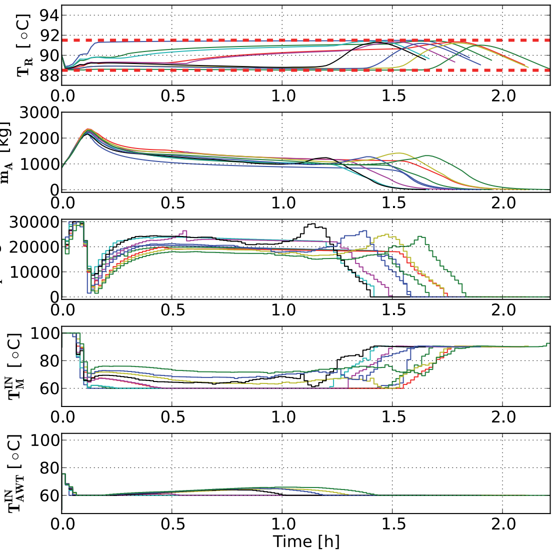

Fig. 5. Reactor temperature, monomer mass, and control inputs for multi-stage NMPC with an economic cost function. The different trajectories show the different scenarios of the uncertainty with mismatches of $\pm 15\%$ with respect to their nominal values.

A summary of all the results is provided in Table 3. It is important to note that multi-stage NMPC satisfies all the constraints for all the scenarios and has a clearly better performance (20.24 % batch time reduction in average) than standard NMPC with the worst-case parameters in the model. It is thus clear that multi-stage NMPC with an economic cost function outperforms all the other algorithms. The efficient implementation using CasADi makes it possible to solve the resulting optimization problem which has around 17000 variables and constraints in real-time with a standard computer. The average computation time per optimization problem is 2.86 s in the case of multi-

Table 3. Performance comparison between standard NMPC, standard NMPC with bias term, standard NMPC with bias term using the worst-case value of the parameters in the model used in the optimizer and multi-stage NMPC.

Scenario		**Batch time in hours**			
Unc. in ΔH_R	**Unc. in** k_0	**Standard NMPC**	**Standard NMPC + bias**	**Standard (w.c.) NMPC + bias**	**Multi-stage NMPC**
+15%	+15%	infeasible	infeasible	1.91	1.85
+15%	+0%	infeasible	infeasible	2.13	1.95
+15%	-15%	infeasible	infeasible	2.52	2.1
+0%	+15%	1.65	1.68	1.92	1.67
+0%	+0%	1.75	1.75	2.22	1.78
+0%	-15%	2.03	2.02	2.68	2.12
-15%	+15%	1.57	1.55	2.08	1.65
-15%	+0%	1.80	1.78	2.43	1.9
-15%	-15%	2.17	2.15	2.90	2.22

stage NMPC and 0.25 s in the case of standard NMPC, so the computation time grows only slightly faster than linearly in the number of scenarios.

7. CONCLUSIONS

In this paper we have presented the application of a new robust nonlinear model predictive control approach to an industrial batch polymerization reactor benchmark problem provided by BASF SE. The approach is based on the representation of the uncertainty as a scenario tree introducing explicitly feedback information in the prediction of the controller, which drastically improves the performance of the controller. The simulation results show that multi-stage NMPC is able to satisfy the tight temperature constraints on the temperature of the reactor for all scenarios under consideration that represent variations of $\pm 15\%$ in critical parameters while standard NMPC and standard NMPC with bias term fail because of the plant model-mistmatch. Also, multi-stage NMPC results in shorter batch times than NMPC with the worst-case parameters. In addition, the use of an economic cost function makes the use of a classical set-point tracking term unnecessary, enhancing the performance of the controller. The approach has been efficiently implemented using the optimization framework CasADi, which leads to an efficient solution of the nonlinear programming problem that results from the multi-stage NMPC formulation, using automatically generated exact first and second order derivative information. This helps to cope with the main challenge of the approach: the size of the resulting optimization problem. Future work will be focused on strategies for the automatic generation of suitable scenario trees, on the explicit exploitation of the structure of the NLP and on the implementation of the approach to a real polymerization reactor.

REFERENCES

J. Andersson, J. Åkesson, and M. Diehl. CasADi – A symbolic package for automatic differentiation and optimal control. In *Recent Advances in Algorithmic Differentiation*, Lecture Notes in Computational Science and Engineering, pages 297–307, 2012a.

J. Andersson, J. Åkesson, and M. Diehl. Dynamic optimization with CasADi. In *Proc. of the 51st IEEE Conference on Decision and Control*, pages 681–686, 2012b.

P.J. Campo and M. Morari. Robust model predictive control. In *Proc. of the American Control Conference*, pages 1021–1026, 1987.

Y. Cao, S. Li, L.R. Petzold, and R. Serban. Adjoint Sensitivity Analysis for Differential-Algebraic Equations: The Adjoint DAE System and its Numerical Solution. *SIAM Journal on Scientific Computing*, 24(3):1076–1089, 2003.

S. Engell. Feedback control for optimal process operation. *Journal of Process Control*, 17:203–219, 2007.

A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, 2008.

M.J. Hadjiyiannis, P.J. Goulart, and D. Kuhn. An efficient method to estimate the suboptimality of affine controllers. *IEEE Transactions on Automatic Control*, 56:2841– 2853, 2011.

A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Tran. on Mathematical Software,*, 31:363–396, 2005.

R. Huang and L. Biegler. Robust nonlinear model predictive controller design based on multi-scenario formulation. In *Proc. of the American Control Conference*, pages 2341–2342, 2009.

S. Lucia and S. Engell. Robust nonlinear model predictive control of a batch bioreactor using multi-stage stochastic programming. In *Proc. of the 12th European Control Conference, Zurich*, pages 4124–4129, 2013.

S. Lucia, T. Finkler, and S. Engell. Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *Journal of Process Control*, 23: 1306–1319, 2013.

D. Muñoz de la Peña, A. Bemporad, and T. Alamo. Stochastic programming applied to model predictive control. In *Proc. of the 44th IEEE Conference on Decision and Control*, pages 1361–1366, 2005.

P.O.M. Scokaert and D.Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998.

B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki. Dynamic optimization of batch processes II. role of measurements in handling uncertainty. *Computers & Chemical Engineering*, 27:27–44, 2002.

A. Wächter and L. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106: 25–57, 2006.

H.S. Witsenhausen. A minimax control problem for sampled linear systems. *IEEE Transactions on Automatic Control*, 13:5–21, 1968.