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

# Quadrature-based scenario tree generation for Nonlinear Model Predictive Control

**Conrad Leidereiter** * **Andreas Potschka** * **Hans Georg Bock** *

* *Interdisciplinary Center for Scientific Computing, Heidelberg University*

**Abstract:** A relatively recent approach for robust Nonlinear Model Predictive Control (NMPC) is based on scenario trees with a so-called recourse formulation. This approach is of interest, because it is less conservative than worst-case robustification approaches. A major challenge when using scenario trees for robust NMPC is the large number of scenarios, which grows exponentially. This exponential growth quickly becomes a bottleneck for the computational costs, which need to stay within bounds that permit real-time applicability. We present how to generate scenarios based on a quadrature rule for the expectation value of an arbitrary economic objective function. The use of sparse grids for the quadrature of the high-dimensional stochastic integrals yields a drastically smaller number of scenarios than the tensor grid approaches used so far. We compare the performance of several robust NMPC approaches for a distillation column with three normally distributed uncertain parameters within a simulated Monte-Carlo controller testbed.

Keywords: Nonlinear Model Predictive Control; Robust Optimization; Scenario Trees; Sparse grids; Monte-Carlo controller evaluation.

## 1. INTRODUCTION

In Nonlinear Model Predictive Control (NMPC) we iteratively solve nonlinear optimization problems to obtain optimal feedback control inputs for an underlying dynamic system. Plant–model mismatch and unforeseen disturbances in real-world applications are often large, such that NMPC can only be used when the feedback controls are computed in a robust sense (see, e.g., Bertsimas et al. [2011]). Otherwise, violations of system constraints can cause severe economical detriments or even safety-critical situations. Robust NMPC techniques treat systematic model mismatch and disturbances as probabilistic perturbations to the nominal model, e.g., via game-theoretic worst-case approaches with linearization of the inner adverse player optimization problem (Diehl et al. [2006a]). This approach, however, suffers from a high conservatism, because it safeguards against all possible perturbation realizations at the same time. Lucia et al. [2013] have shown that by using a scenario tree approach this conservatism can be efficiently reduced while at the same time maintaining feasibility with high probability.

The main challenge in Scenario Tree NMPC is the high numerical effort for optimizing all scenarios at the same time. As we shall see later, there is even an exponential growth on the number of scenarios in the number of decision stages. Thus, the question of the choice of scenarios is of paramount importance for real-time feasibility of Scenario Tree NMPC. In this article, we propose a method of scenario generation based on sparse-grid quadrature rules [Gerstner and Griebel, 1998], which are a commonly used tool in the field of uncertainty quantification.

## 2. NOMINAL NMPC

In NMPC, we compute a feedback control for a dynamic system at a time $t_0$ by solving an optimal control problem on a sufficiently long prediction horizon $[t_0, t_{\text{end}}]$ in the future. The solution of the prediction problem depends on an estimate of the current system state at $t_0$, which is usually obtained approximately by the means of measurements and estimation techniques like the (extended) Kalman Filter. Alternatively, the current system state can be approximated by a dynamic parameter estimation problem on a time-horizon $[t_{-k}, t_0]$ in the past [Haseltine and Rawlings, 2005, Diehl et al., 2006b]. Then, we shift the prediction and estimation horizons into the future by a small timestep (the so-called sampling time) and repeat the prediction and estimation task. If the controlled process has a finite end time, we might have to shrink the prediction horizon at the end.

In this contribution, we focus on the prediction problem on the current prediction horizon $I = [t_0, t_{\text{end}}]$. In each NMPC iteration, we need to solve one optimal control problem for states $x : I \to \mathbb{R}^{n_x}$ and controls $u : I \to \mathbb{R}^{n_u}$. Possibly uncertain parameters $p \in \mathbb{R}^d$ enter the dynamics of the system and serve as the main lever for robustification in this article. We assume that the model functions

$$f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^d \to \mathbb{R}^{n_x}, \qquad g_f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_f},$$
$$g_c : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^d \to \mathbb{R}^{n_c}, \qquad \Phi : \mathbb{R}^{n_x} \times \mathbb{R}^d \to \mathbb{R}$$

are sufficiently smooth. They describe the system dynamics $f$, mixed control-state path constraints $g_c$, terminal state constraints $g_f$, and the objective function $\Phi$. We denote the current (estimated) system state by $x_0 \in \mathbb{R}^{n_x}$. The prediction problem then reads

$$\min_{x(t),u(t)} \Phi(x(t_{\text{end}}), p) \tag{1a}$$
$$\text{s.t. } \dot{x}(t) = f(x(t), u(t), p), \quad t \in [t_0, t_{\text{end}}], \tag{1b}$$
$$x(0) = x_0, \tag{1c}$$
$$0 \leq g_c(x(t), u(t), p), \quad t \in [t_0, t_{\text{end}}], \tag{1d}$$
$$0 \leq g_f(x(t_{\text{end}})). \tag{1e}$$

We can regard (1) as a parametric optimal control problem in $q = (x_0, p)$. From now on we assume that there always exists a locally unique optimum of (1), which we denote by $x^*(.;q), u^*(.;q)$. In this sense, ideal NMPC can be regarded as the computation of a nonlinear feedback law $u^*$ for the closed loop system

$$\dot{x}(t) = f(x(t), u^*(t; x(t), p), p).$$

Of course, discretization errors and feedback delay must be taken into account in practice.

Direct methods like Direct Multiple Shooting [Bock and Plitt, 1984] or Direct Collocation [Tsang et al., 1975, Bär, 1983, Biegler, 1984] have proven to be versatile and efficient for the solution of (1). However, in the presence of hard real-time requirements, the resulting discretized problems must be treated with advanced numerical methods. The Real-Time Iteration scheme [Diehl et al., 2003, 2005] for instance solves the resulting discrete multiple shooting problems only approximately and reduces the feedback time further through a splitting of the iteration into a preparation and feedback phase. The resulting initial value embedding idea is also possible in the case of Direct Collocation (see, e.g., Zavala and Biegler [2009]). Further enhancements can be obtained by the exploitation of the different time-scales of trajectories and their sensitivities in the framework of the Multi-Level Iteration [Bock et al., 2007, Kirches et al., 2010, Frasch et al., 2012].

## 3. ROBUST NMPC AND SCENARIO TREES

The major obstacle in the application of advanced control methods like NMPC in real-world systems is the presence of the often large mismatch between model and plant or random process disturbances. We try to attack this problem via robustification of the NMPC scheme.

From now on, we assume that the parameter vector $p$ originates from a fixed probability space $(\Omega, \mathcal{F}, \mu)$ with $\Omega \subset \mathbb{R}^d$, $\mathcal{F}$ the corresponding Borel sigma algebra and $\mu$ the measure corresponding to a known distribution. In nominal NMPC, we solve a discrete counterpart of (1) using only the mean value of $p$. Worst-case robustification approaches like Diehl et al. [2006b] take the rather pessimistic viewpoint of an adverse player always choosing the worst parameter realization. A relatively recent alternative approach is the use of scenario trees for robust NMPC [Lucia et al., 2013]: Instead of only using the mean of $p$, we consider $m \in \mathbb{N}$ different realizations of $p$ simultaneously. Additionally, we allow for changes of $p$ in time on $n_d \in \mathbb{N}$ so-called decision points. In the case of only $d = 1$ uncertain parameters, $n_d = 2$ decision points, and $m = 3$ realizations, we obtain $(d \cdot m)^{n_d} = 9$ scenarios. The scenarions can be linked together to form a scenario tree if their parameter realizations coincide in the beginning, compare Fig. 1.

In contrast to the worst-case approach, we can have a full set of controls for each scenario separately. Thus, scenario tree NMPC exhibits less controller conservatism than worst-case approaches [Lucia et al., 2013], because one set of controls does not need to be feasible for all possible realizations of $p$. In order to have a well-defined feedback control and not to violate a causality principle, we must require so-called non-anticipativity constraints to couple the controls of different scenarios if their realization histories coincide. In Fig. 1 for instance, we require the non-anticipativity constraints

$$u_0^1 = u_0^2 = u_0^3, \qquad u_1^1 = u_1^2 = u_1^3,$$
$$u_1^4 = u_1^5 = u_1^6, \qquad u_1^7 = u_1^8 = u_1^9.$$

As we have seen, the number $(d \cdot m)^{n_d}$ of scenarios grows exponentially in $n_d$. Thus, the choice of scenarios is of major importance.

## 4. SCENARIO TREE GENERATION FROM QUADRATURE FORMULAS

The design of a suitable scenario tree is always a trade-off between the coverage of the uncertainty space and the computation cost of large trees.

To motivate our approach we move back to (1). Instead of minimizing $\Phi$ for only one realization of $p$, we take the expectation of

$$\Phi^*(x_0, p) := \Phi(x^*(t_{\text{end}}; x_0, p), p)$$

in the space of uncertain parameters as an objective function. It can be expressed as an integral over the $d$-dimensional probability space. When computing the integral value numerically, we require a reliable quadrature in high dimensions. To this end, we approximate the expectation value with a sum over a finite set $\Gamma \subset \Omega$ according to

$$\mathbb{E}(\Phi^*(x_0, p)) = \int_\Omega \Phi^*(x_0, p) d\mu(p) = \int_\Omega \Phi^*(x_0, p) f_\mu(p) dp$$
$$\approx \sum_{p \in \Gamma} w(p) \Phi^*(x_0, p) f_\mu(p).$$

We then interpret every $p \in \Gamma$ as one parameter realization. Thus, we have $m = |\Gamma|$. It is often argued that a

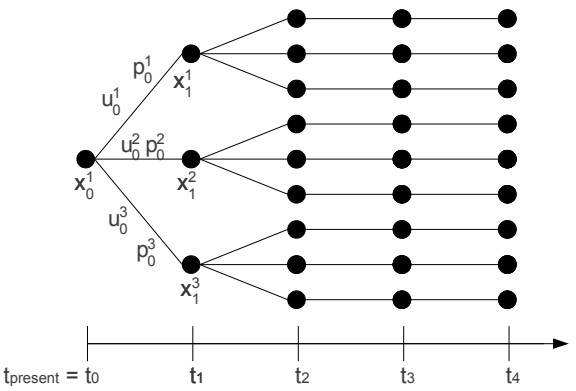


Fig. 1. For $d = 1$ uncertain parameters, $n_d = 2$ decision points $t_0$ and $t_1$, and $m = 3$ realizations, we obtain 9 scenarios, which can be linked together to form a scenario tree.

robust scenario tree must contain the combined extreme values for all uncertain parameters. However, our probabilistic setting usually does not allow for identification of extreme values. In the following we consider $\Phi^* f_\mu$ with bounded mixed derivative, cf. (2). If we choose for instance $f_\mu$ such that the uncertain parameters are normally distributed and the cost function such that the bounded mixed derivative condition holds, an extreme realization of all uncertain parameters at the same time is highly unlikely on the basis of their joint distribution.

Along this line of arguments, we employ quadrature formulas with grid points that resolve the underlying probability space accurately, but have benign computation costs in higher dimensions.

## 5. SPARSE GRIDS

In the area of uncertainty quantification sparse grids are the method of choice for the evaluation of high-dimensional integrals. Function evaluations of $\Phi^*$ at the nodes are expensive in our case, because we need to simulate a full scenario for each node. Therefore, we aim at reducing the number of grid points compared to a full tensor grid with $m^d$ points without sacrificing accuracy on the basis of sparse grids.

Following Gerstner and Griebel [1998], we now explain how to approximate the integral of a function $F : \Omega \to \mathbb{R}$ with sparse grid quadrature. We denote the exact value by

$$I^d(F) = \int_\Omega F(x)\mathrm{d}x.$$

We consider a sequence of quadrature formulas on level $l \in \mathbb{N}$ with $n_l^d$ underlying points, $n_l^d < n_{l+1}^d$. Then the exact integral can be approximated by

$$Q_l^d(F) := \sum_{i=1}^{n_l^d} w_{li} F(x_{li})$$

with quadrature weights $w_{li} \in \mathbb{R}$ and node points $x_{li} \in \Omega, i = 1, \dots, n_l^d$. Moreover, the underlying quadrature grid on level $l$ is denoted by

$$\Gamma_l^d := \{x_{li} : 1 \le i \le n_l^d\} \subset \Omega.$$

The construction of sparse grids was proposed by Smolyak [1963] for functions with bounded mixed derivatives of order $r$, denoted by $\mathcal{W}_d^r$,

$$\mathcal{W}_d^r := \left\{ F : \Omega \to \mathbb{R}, \left\| \frac{\partial^{|\mathbf{s}|} F}{\partial x_1^{s_1} \cdots x_d^{s_d}} \right\|_\infty < \infty, s_i \le r \right\}, \quad (2)$$

with multi-index $\mathbf{s} \in \mathbb{N}^d$ and $|\mathbf{s}| = \sum_{i=1}^d s_i$.

### 5.1 Smolyak's algorithm

Let $l \in \mathbb{N}$. For $F \in \mathcal{W}_1^r$, $w_{li} \in \mathbb{R}$, and $x_{li} \in \Gamma_l^d$, we consider the one-dimensional quadrature formula

$$\mathcal{Q}_l^1(F) = \sum_{i=1}^{n_l^1} w_{li} F(x_{li}).$$

We then define the difference formulas

$$\Delta_l^1(F) := (\mathcal{Q}_l^1 - \mathcal{Q}_{l-1}^1)F \quad \text{with} \quad \Delta_0^1(F) := 0.$$

The difference formulas are quadrature formulas on the grid $\Gamma_l^1 \cup \Gamma_{l-1}^1$. If the quadrature formulas are nested, that is $\Gamma_{l-1}^1 \subset \Gamma_l^1$, then the underlying grid of the difference formula $\Delta_l^1$ is $\Gamma_l^1$.

To lift up the one-dimensional formulas to $d$-dimensional formulas for $F \in \mathcal{W}_d^r$, we define the tensor product of quadrature formulas $(\mathcal{Q}_{l_1}^1 \otimes \ldots \otimes \mathcal{Q}_{l_d}^1)$ as the sum over all possible combinations.

$$(\mathcal{Q}_{l_1}^1 \otimes \cdots \otimes \mathcal{Q}_{l_d}^1)(F) := \sum_{i_1=1}^{n_{l_1}^1} \cdots \sum_{i_d=1}^{n_{l_d}^1} w_{l_1 i_1} \cdot w_{l_d i_d} \cdot F(x_{l_1 i_1}, \cdots, x_{l_d i_d}).$$

Smolyak's formula for $F \in \mathcal{W}_d^r$, $l \in \mathbb{N}$, and multi-index $\mathbf{k} \in \mathbb{N}^d$ can then be expressed as

$$\mathcal{Q}_l^d(F) := \sum_{|\mathbf{k}| \le l+d-1} (\Delta_{k_1}^1 \otimes \ldots \otimes \Delta_{k_d}^1)(F). \quad (3)$$

The underlying grid for formula (3) is called sparse grid.

Compared to the sparse grid formula (3), the full tensor product formula

$$\sum_{j=1}^{d} \sum_{1 \le k_j \le l} (\Delta_{k_1}^1 \otimes \ldots \otimes \Delta_{k_d}^1)(F)$$

corresponds to summation over the whole cube of indices $\{\mathbf{k} : k_j \le l, \quad j = 1, \dots, d\}$. Instead, the sparse grid formula (3) sums over a much smaller simplex of indices $\{\mathbf{k} : |\mathbf{k}| \le l + d - 1\}$.

Alternatively to the formulation with difference formulas $\Delta_{k_j}^1$, we can denote Smolyak's formula in terms of $\mathcal{Q}_{k_j}^1$ by

$$\mathcal{Q}_l^d(F) = \sum_{|\mathbf{k}|=l}^{l+d-1} (-1)^{(l+d-|\mathbf{k}|-1)} \binom{d-1}{|\mathbf{k}|-l} (\mathcal{Q}_{k_1}^1 \otimes \cdots \otimes \mathcal{Q}_{k_d}^1)(F).$$

Sparse grids of levels 0, 1, 2, and 3 as well as a tensor grid in dimension $d = 3$ are depicted in Fig. 2, 3, 4 and 5.

### 5.2 Error bounds

Especially in higher dimensions, the number of underlying quadrature nodes for sparse grid quadrature is much smaller compared to tensor grid quadrature. In the nested case, the number of quadrature points of a sparse grid is

$$n_l^d = \sum_{|\mathbf{k}| \le l+d-1} n_{k_1}^1 \cdot \ldots \cdot n_{k_d}^1.$$

If we assume $n_l^1 = \mathcal{O}(2^l)$, which is a justified assumption for one-dimensional quadrature rules like the trapezoidal rule or the Clenshaw-Curtis rule, we arrive at $n_l^d = \mathcal{O}(2^l \cdot l^{(d-1)})$. In contrast, the number of grid points for full tensor product rules is $\mathcal{O}(2^{ld})$.

To formulate error bounds for sparse grid quadrature, we start with an error bound $E_l^1(F)$ of the one-dimensional quadrature formulas with positive weights, as for example in the case of the Clenshaw-Curtis rule. If we assume that $f \in C^r$, the approximation

$$|E_l^1(F)| = \mathcal{O}((n_l^1)^{-r})$$

holds. We take such a quadrature formula as a basis for Smolyak's algorithm and additionally assume $F \in \mathcal{W}_d^r$ and

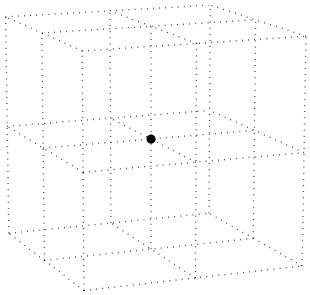

Fig. 2. The sparse grid of level $l = 0$ in dimension $d = 3$ is a single point.

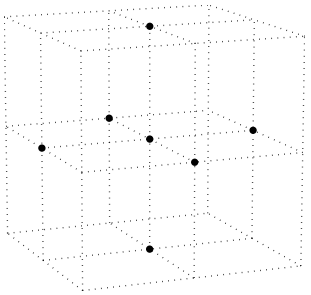

Fig. 3. The sparse grid of level $l = 1$ in dimension $d = 3$ consists of 7 points on the coordinate axes.

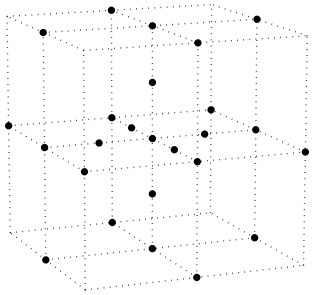

Fig. 4. The sparse grid of level $l = 2$ in dimension $d = 3$ has 25 points which cluster at the coordinate axes.

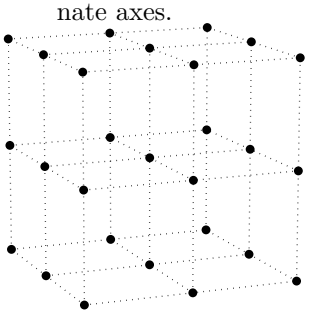

Fig. 5. The full tensor grid with $m = 3$ in dimension $d = 3$ already consists of 27 points.

$n_l^1 = \mathcal{O}(2^l)$. Then the error of sparse grid quadrature is [Gerstner and Griebel, 1998]

$$|E_l^d(F)| = \mathcal{O}(2^{-lr} \cdot l^{(d-1)(r+1)}).$$

As stated in Schillings [2011], the approximation quality of sparse grids even outperforms tensor grid approximation quality. The composition of sparse grid points, which cluster along the axes (Fig. 4), is better than the composition of tensor grid points (Fig. 5).

## 6. MONTE-CARLO TESTBED

For the evaluation of feasibility and performance of different NMPC controllers we construct a simplified reality. For each sampling interval, we draw realizations of the uncertain parameters from their joint distribution, allowing jumps in the parameters on the sampling grid. We further assume that the controller in question has perfect knowledge of the current system state and of the parameter realizations in the previous sampling intervals. We want to stress, however, that the NMPC controller does not know the current realization of the uncertain parameters.

We then produce a large number of random sequences of parameters. For each sequence of parameter realizations we obtain a closed-loop objective and constraint value. These values can then be investigated with statistical methods. Thus, we can reduce the expected controller performance and feasibility to a few values for meaningful comparison.

| | $V$ | $m$ | $m_C$ |
|---|---|---|---|
| Mean | 100 | 0.1 | 0.1 |
| Standard deviation $\sigma_i$ | 5 | 0.005 | 0.005 |

Table 1. The standard deviation of all uncertain parameters in the distillation column is 5% of its mean.

For simplicity, we use Direct Multiple Shooting without Real-Time and Multi-Level Iterations.

## 7. CASE STUDY

We compare three NMPC controllers and consider their advantages and disadvantages on the basis of the statistical Monte-Carlo testbed results. More precisely, we compare nominal NMPC with Scenario Tree NMPC using $m = 3$ on a sparse grid of level $l = 1$ (Fig. 3) and on a full tensor grid (Fig. 5). In other words, we compare Scenario Tree NMPC controllers with 1, 7, and 27 scenarios.

We solve a problem of distillation control [Diehl et al., 2006a]. The model consists of five distillation trays. At the bottom of the distillation column, the reboiler content $M_0$ with concentration $\xi_0$ is heated and produces vapour. We denote by $\xi_i$ the concentration in tray $i = 1 \dots 5$. The vapour equilibrium in the reboiler and in each tray is given by an algebraic relation $y(\xi_i) = \xi_i(1+\alpha)/(\xi_i+\alpha)$ depending on a constant parameter $\alpha \in \mathbb{R}$.

The distillation column can be controlled by the reflux ratio $R(t)$: At the top of the column, the vapour condenses and a liquid molar flux $L = VR/(1+R)$ is fed back into the column, where $V \in \mathbb{R}$ is the vapour molar flux. We assume that the parameter $L$ is constant in each tray. At the top of the column, distillate content $M_D$ with concentration $x_D$ is produced. As the state vector we obtain

$$x = (M_0, \xi_0, \xi_1, \dots, \xi_6, M_D, x_D, \alpha)^T.$$

The dynamics of the distillation column are modelled as a system of ordinary differential equations

$$\begin{aligned}
\dot{M}_0 &= -V + L, \\
\dot{\xi}_0 &= {M_0}^{-1}(L\xi_1 - Vy(\xi_0) + (V-L)\xi_0), \\
\dot{\xi}_i &= m^{-1}(L\xi_{i+1} - Vy(\xi_i) + Vy(\xi_{i-1}) - L\xi_i), \\
&\qquad i = 1, \dots, 5, \\
\dot{\xi}_6 &= {m_C}^{-1}V(y(\xi_5) - \xi_6), \\
\dot{M}_D &= V - L, \\
\dot{x}_D &= {M_D}^{-1}(V-L)(\xi_6 - x_D).
\end{aligned}$$

For this study, we regard the vapour molar flux $V$, the molar holdup of each tray $m$, and the molar holdup of the condenser $m_C$ as uncertain. We assume that all uncertain quantities are stochastically independent and normally distributed with fixed mean and variance according to Tab. 1.

The probability space of uncertainties is $(\mathbb{R}^3, \mathcal{B}(\mathbb{R}^3), \mu)$ with $\mu$ the measure corresponding to the joint distribution of $p = (V, m, m_C)^T$.

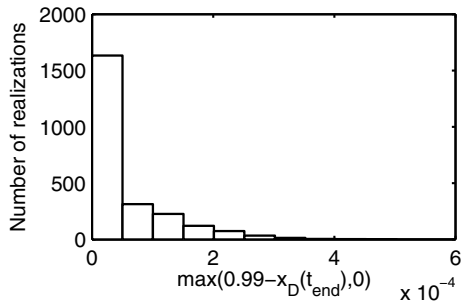


Fig. 6. A terminal constraint, the purity of the destillate, is sometimes violated in the case of nominal NMPC.

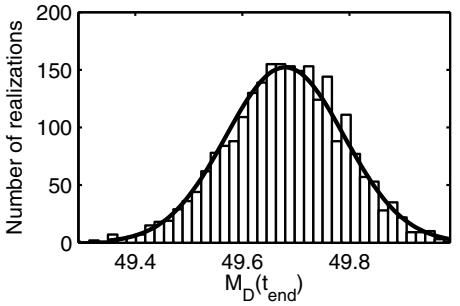


Fig. 7. In case of single point quadrature, nominal NMPC, the histogram of the produced amount of distillate for 2420 realizations can be fitted by a normal distribution around the mean of 49.680.

The control $R(t)$ and the states are bounded according to

$$0 \leq R(t) \leq 15, \quad t \in I,$$
$$x_{\min} \leq x(t) \leq x_{\max}, \quad t \in I.$$

We further require that the purity of the distillate is at least 99% at $t_{\text{end}}$, i.e.,

$$0 \leq g_f(x(t_{\text{end}})) = x_D(t_{\text{end}}) - 0.99.$$

This terminal constraint ensures the quality of the distillate. The objective function is the amount of product

$$\Phi(x(t_{\text{end}}), p) = -M_D(t_{\text{end}}).$$

When applying the quadrature-based scenario tree generation to the distillation column, we use the current time $t_0$ as the only decision point, therefore $n_d = 1$. Additionally, we consider only the closed compact interval $[-3\sigma_i, 3\sigma_i]$ for each uncertain parameter component and neglect the tails of the distribution.

The most important property of the distillate is the purity, because often the full batch needs to be thrown away if it does not meet the product specifications.

In Fig. 6 we depict the purity constraint violation for nominal NMPC. For a certain value of the constraint violation at the $x$-axis we plot the corresponding number of realizations with the specific violation in histogram bins. In the cases of sparse and tensor grid Scenario Tree NMPC, however, we have not observed any purity constraint violations. This observation underlines the robustness of the Scenario Tree approach.

Fig. 7, 8, and 9 show histogram plots of the negative objective $M_D(t_{\text{end}})$ for the cases of nominal NMPC, sparse grid, and full tensor grid Scenario Tree NMPC, respectively. The

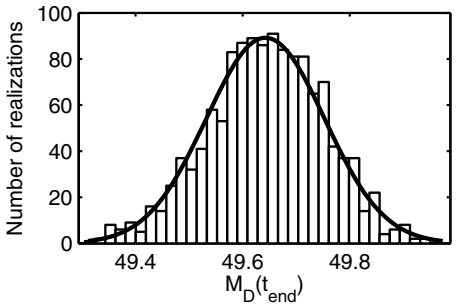


Fig. 8. With an underlying sparse grid for quadrature, the histogram of the produced amount of distillate for 1302 realizations can be fitted by a normal distribution around the mean of 49.643.

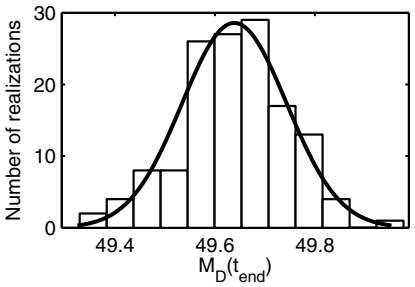


Fig. 9. For the tensor grid quadrature the histogram of the produced amount of distillate for 139 realizations can be fitted by a normal distribution around the mean of 49.639.

| | nominal | sparse grid | tensor grid |
|---|---|---|---|
| Mean | 49.680 | 49.643 | 49.639 |
| Standard deviation $\sigma_i$ | 0.111 | 0.110 | 0.105 |
| p-value | 0.081 | 0.118 | 0.264 |

Table 2. The the mean and standard deviation of the amount of distillate in the terminal time vary for different quadrature methods. The p-values indicate how extreme the realization is. With higher p-value the null-hypothesis, normal distribution, cannot be rejected anymore.

realizations of $\Phi^*$ are sorted into bins on the $x$-axis. The number of realizations is on the $y$-axis. We see, that the bin heights can be approximated nicely by a gaussian bell curve. We want to stress that it is not clear a-priori, how a normally distributed input influences the output variables of a nonlinear system. In the test case of the distillation column, however, the null-hypothesis that $M_D(t_{\text{end}})$ is normally distributed cannot be rejected based on the p-value of a $\chi^2$ goodness-of-fit test (compare Tab. 2).

When we look at the mean values in Tab. 2, nominal NMPC leads to a higher amount of distillate than in the other two cases. However, this is mainly due to the infeasibilities mentioned before. Thus, this behavior is rather an economical drawback than an advantage. When comparing the means and variances of the sparse grid case and full tensor grid case in Tab. 2, there is only a marginal difference. Thus, we can conclude that the sparse grid Scenario Tree NMPC with 74% less scenarios

yields virtually the same controller performance as the full tensor grid Scenario Tree NMPC. Therefore the sparse grid should be prefered to the full tensor grid in order to save computation time without sacrificing controller performance.

## 8. CONCLUSION

We have proposed a new method to generate scenario trees for robust NMPC of uncertain systems with randomly distributed parameters. Our approach is based on high-dimensional quadrature rules, which can be efficiently generated in an a-priori fashion on the basis of sparse grids. The resulting scenarios differ from usually used ones in that the extreme corner points of parameter realizations are not included. In the case of distributed parameters, these corner cases are simply highly unlikely to realize.

We have demonstrated the efficiency of our approach for the case of robust control of a destillation column within a virtual Monte-Carlo testbed, for which we could obtain the same controller performance and feasibility with considering 74% less scenarios compared to a conventional full tensor grid scenario tree approach. This means a drastic reduction of computational time and opens up the possibility to use robust NMPC for a much wider range of problems than could be tackled up to now.

## ACKNOWLEDGEMENTS

We gratefully acknowledge support of the German Research Foundation within the DFG cluster program "Optimization-based Control of Uncertain Systems" under grant BO864/15-1.

## REFERENCES

V. Bär. Ein Kollokationsverfahren zur numerischen Lösung allgemeiner Mehrpunktrandwertaufgaben mit Schalt– und Sprungbedingungen mit Anwendungen in der optimalen Steuerung und der Parameteridentifizierung. Diploma thesis, Rheinische Friedrich–Wilhelms–Universität zu Bonn, 1983.

D. Bertsimas, D. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53: 464–501, 2011.

L.T. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers & Chemical Engineering*, 8:243–248, 1984.

H.G. Bock and K.J. Plitt. A Multiple Shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress*, pages 243–247, Budapest, 1984. Pergamon Press.

H.G. Bock, M. Diehl, E.A. Kostina, and J.P. Schlöder. Constrained optimal feedback control of systems governd by large differential algebraic equations. In L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, editors, *Real-Time PDE-Constrained Optimization*, chapter 1, pages 3–24. SIAM, 2007.

M. Diehl, R. Findeisen, F. Allgöwer, J.P. Schlöder, and H.G. Bock. Stability of nonlinear model predictive control in the presence of errors due to numerical online optimization. In *Proc. 43th IEEE Conf. Decision Contr.*, pages 1419–1424, Maui, Hawaii, 2003.

M. Diehl, H.G. Bock, and J.P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5):1714–1736, 2005.

M. Diehl, H.G. Bock, and E. Kostina. An approximation technique for robust nonlinear optimization. *Mathematical Programming*, Ser. B 107:213–230, 2006a.

M. Diehl, P. Kuehl, H.G Bock, and J.P. Schlöder. Schnelle Algorithmen für die Zustands- und Parameterschätzung auf bewegten Horizonten. *Automatisierungstechnik*, 54 (12):602–613, 2006b.

J.V. Frasch, L. Wirsching, S. Sager, and H.G. Bock. Mixed-level iteration schemes for nonlinear model predictive control. In *Proceedings of the IFAC Conference on Nonlinear Model Predictive Control*, 2012.

T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18:209–232, 1998.

E.L. Haseltine and J.B. Rawlings. Critical evaluation of extended Kalman filtering and Moving-Horizon Estimation. *Industrial and Engineering Chemistry Research*, 44:2451–2460, 2005.

C. Kirches, L. Wirsching, S. Sager, and H.G. Bock. Efficient numerics for nonlinear model predictive control. In M. Diehl, F. Glineur, E. Jarlebring, and W. Michiels, editors, *Recent Advances in Optimization and its Applications in Engineering*, pages 339–359. Springer, 2010.

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.

C. Schillings. *Optimal aerodynamic design under uncertainties*. PhD Thesis, Univ. Trier, 2011.

S.A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR*, 4:240–243, 1963.

T.H. Tsang, D.M. Himmelblau, and T.F. Edgar. Optimal control via collocation and non-linear programming. *International Journal on Control*, 21:763–768, 1975.

V.M. Zavala and L.T. Biegler. The advanced–step NMPC controller: optimality, stability and robustness. *Automatica*, 45(1):86–93, 2009.