Preprints of the
9th International Symposium on Advanced Control of Chemical Processes
The International Federation of Automatic Control
June 7-10, 2015, Whistler, British Columbia, Canada

Scenario-based Model Predictive Control: Recursive Feasibility and Stability

Michael Maiworm * **Tobias Bäthge** *,**,1 **Rolf Findeisen** *

* *Institute for Automation Engineering, Otto-von-Guericke University, Magdeburg, Germany.*
{michael.maiworm, tobias.baethge, rolf.findeisen}@ovgu.de
** *Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany.*

Abstract: Many processes are influenced by uncertain parameters or external disturbances, such as temperature changes. The control of such systems is in general challenging. In this work, we consider robust multi-scenario Model Predictive Control (MPC). Its central idea is to assume a finite number of possible values for the uncertainties and to model their combinations in a scenario tree. We adapt the classical dual mode approach of nominal MPC to establish recursive feasibility and stability for the multi-scenario case, using a common terminal region and common terminal cost function for all uncertainty realizations. For linear systems, the computation of these ingredients can be formulated as a semidefinite program. In a simulation, we apply the suggested approach to building climate control and show that it robustly stabilizes the system while a standard MPC controller violates state constraints and becomes infeasible.

Keywords: Predictive control; robust control; stability; recursive feasibility; uncertainty; scenario tree; building climate control.

1. INTRODUCTION

Model Predictive Control (MPC) is an optimization-based control scheme naturally capable of dealing with multi-input multi-output systems. In addition, MPC allows to include input and state constraints in the controller design. For these reasons, MPC is increasingly used in industrial applications (Mayne (2014); Di Cairano (2012); Qin and Badgwell (2003)).

However, MPC needs an accurate model to predict the system behavior and to achieve good and reliable performance. Unfortunately, uncertainties (e.g. noise, unknown disturbances, model-plant-mismatch due to inaccurate modeling) are always present and deteriorate the performance of the controller. They may also destroy important properties like recursive feasibility and stability. For this reason, extensions to MPC have been developed over the last few decades that allow to explicitly take uncertainties into account, guaranteeing constraint satisfaction, recursive feasibility, and stability. This includes e.g. min-max MPC (Rawlings and Mayne (2009)), tube-based MPC (Mayne et al. (2006, 2009)), feedback MPC (Mayne et al. (2000); Bertsekas (2005)), and relaxation-based robust MPC (Streif et al. (2014)).

In the frame of this work, we consider scenario-based robust MPC, also denoted as Multi-Scenario Model Predictive Control (MS-MPC), see e.g. Lucia et al. (2012). The central idea of this approach is to assume or approximate discrete values for an uncertainty and to capture their combinations over time by means of a scenario tree. Each scenario starts from the *root node* (the initial value $x(k)$) and ends in one of the leaves (cf. Fig. 1).

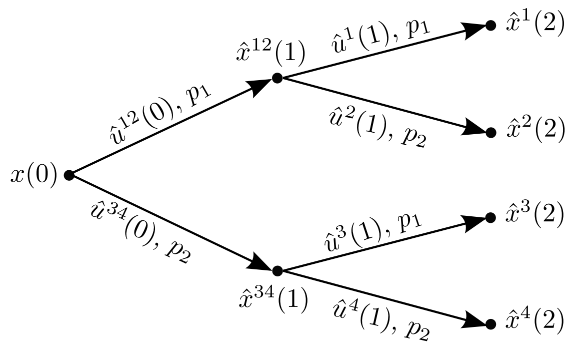


Fig. 1. Scenario tree with a prediction horizon of $N = 2$ and the discrete uncertain parameter set $\mathcal{P} = \{p_1, p_2\}$. The time instant is set to $k := 0$.

While this is an approximation for continuous disturbances, the approach shows good performance compared to standard MPC if the uncertainty values are selected suitably. In addition, it could be shown that the multi-scenario optimization problem remains feasible where standard MPC does not (Lucia et al. (2012)).

A major drawback of MS-MPC is that the problem size grows exponentially with the prediction horizon and the numbers of uncertain parameters and different values that these parameters can take. One way to circumvent this problem is to assume that the scenario tree stops branching further after a defined stage within the prediction horizon, called *robust horizon* N_r (Lucia and Engell (2012)).

[1] Support by the International Max Plank Research School (IMPRS) for Advanced Methods in Process and Systems Engineering, Magdeburg, Germany, is acknowledged.

In this work, we investigate how to adapt MPC to guarantee recursive feasibility and stability for the described multi-scenario case, using a terminal region, a terminal penalty, and a virtual local control law. The contributions of this paper can be summarized as follows:

- Adaptation of the stability approaches of nominal MPC to guarantee recursive feasibility and stability for the multi-scenario case.
- Computation of suitable terminal regions and terminal cost functions for multiple scenarios.
- Application of MS-MPC to building climate control, namely the cooling of a cold storage house.

The outline of the paper is as follows. Section 2 presents the MS-MPC problem formulation and the necessary adaptations to establish recursive feasibility and stability. This leads to the definition of a suitable common terminal region and common terminal cost function, which are investigated in detail and computed in Section 3. Afterwards, the MS-MPC scheme is employed for an example of building climate control in Section 4. The work concludes with Section 5, which summarizes the results and highlights possible future extensions.

2. MULTI-SCENARIO MODEL PREDICTIVE CONTROL

We consider a general nonlinear discrete time system

$$\begin{aligned} x^+ &= f(x,u,p) \\ \text{s.t. } x &\in \mathcal{X} \\ u &\in \mathcal{U} \\ p &\in \mathcal{P} \ , \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ denotes the states, x^+ the states at the next time instant, $u \in \mathbb{R}^{n_u}$ the inputs, and $p \in \mathbb{R}^{n_p}$ the uncertain parameters. The inputs and states are constrained by the closed and convex sets $\mathcal{U}$ and $\mathcal{X}$. The fundamental assumption of the MS-MPC approach is that, at each time step, the uncertainties p are drawn from a discrete set of the form $\mathcal{P} = \{p_i \in \mathbb{R}^{n_p} \mid i = 1, \ldots, s\}$, with $s \in \mathbb{N}$ denoting the number of possible values p can take. A discrete set $\mathcal{P}$ may be the true nature of a specific uncertainty, e.g. the number of lost packages in a communication network or the possible faults in a system (fault-tolerant control). Likewise, a discrete set $\mathcal{P}$ may arise from the discretization of a continuous set (Goodwin et al. (2009)).

We consider all unique combinatorial permutations of the elements of $\mathcal{P}$ over the prediction horizon. For a given prediction horizon $N \in \mathbb{N}$ and the total number s of possible values of the parameter p, the number of possible combinations is $N_s = s^N$. These unique combinations are represented by the parameter sequences

$$\mathbf{p}^j(k) = \left\{p^j(k), p^j(k+1), \ldots, p^j(k+N-1)\right\}$$

with $p^j(k+i) \in \mathcal{P}$ for all $i \in \mathcal{I}_{0:N-1} = \{0, \ldots, N-1\}$ and all $j \in \mathcal{I}_{1:N_s} = \{1, \ldots, N_s\}$. The superscript j is the scenario index and k denotes the current time step. We call each of these sequences $\mathbf{p}^j(k)$ a *parameter scenario* as it resembles a unique realization of the uncertain parameter evolution over the prediction horizon.

In total, we obtain N_s different parameter scenarios and to each of them, we can associate a predicted input and state sequence $\hat{\mathbf{u}}^j(k)$ and $\hat{\mathbf{x}}^j(k)$. Predicted variables are denoted with a $\hat{\cdot}$. To account for the different scenarios, we write the prediction model as

$$\hat{x}^j(k+1+i) = f\left(\hat{x}^j(k+i), \hat{u}^j(k+i), p^j(k+i)\right) \tag{2}$$

with $i \in \mathcal{I}_{0:N-1}$ and $j \in \mathcal{I}_{1:N_s}$.

Scenario tree complexity In the scenario tree, the controls leaving one node have to be the same for all scenarios that share that particular node because they are based on the same past information and no exact uncertainty realization can be anticipated. These are the so-called *non-anticipatory constraints* (Goodwin et al. (2009)). They can be reformulated as linear equality constraints of the form

$$\begin{aligned} \Gamma\tilde{\mathbf{x}}(k) &= 0 \\ \Gamma\tilde{\mathbf{u}}(k) &= 0 \ , \end{aligned} \tag{3}$$

where Γ is a sparse matrix of appropriate dimensions and $\tilde{\mathbf{x}}(k)$ and $\tilde{\mathbf{u}}(k)$ are the extended state and input sequences containing the states and inputs of all scenarios. The non-anticipatory constraints matrices are the same for states and inputs because equal inputs are computed for equal states (or nodes) in the scenario tree.

In the general case, $\tilde{\mathbf{u}}(k)$ consists of Ns^N elements, the prediction horizon multiplied by the number of scenarios. Yet, the non-anticipatory constraints reduce the number of independent optimization variables in $\tilde{\mathbf{u}}(k)$. This number is equal to the sum of all nodes from the root node to the nodes at the penultimate stage, $\sum_{i=0}^{N-1} s^i = \frac{s^N-1}{s-1}$.

The inherent computational reduction can be expressed by

$$\frac{\sum_{i=0}^{N-1} s^i}{Ns^N} = \frac{\frac{s^N-1}{s-1}}{Ns^N} = \frac{s^N-1}{Ns^N(s-1)} \tag{4}$$

and can then be approximated for large s or N by

$$\approx \frac{s^N}{Ns^N(s-1)} = \frac{1}{N(s-1)} \ . \tag{5}$$

Thus, for large s or N, (5) computes the percentage to which the number of optimization variables is reduced, compared to all elements in $\tilde{\mathbf{u}}(k)$. For instance, for $s = 6$ and $N = 8$, the number of optimization variables is reduced to 2.5 % of the total number of optimization variables contained in $\tilde{\mathbf{u}}(k)$ for that case. However, despite the reduction by the non-anticipatory constraints, the number of independent optimization variables still grows exponentially with the prediction horizon.

Cost function As it is unknown which of the scenarios will be the actual realization of the uncertainty evolution, all of them have to be taken into consideration, i.e., we have to optimize over the full scenario tree. In this optimization, we use the cost function

$$V_N(x(k), \tilde{\mathbf{u}}(k)) = \sum_{j=1}^{N_s} \omega_j J_j \tag{6}$$

with

$$J_j = \sum_{i=0}^{N-1} \ell\left(\hat{x}^j(k+i), \hat{u}^j(k+i)\right) + V_\mathrm{f}\left(\hat{x}^j(k+N)\right) \ ,$$

where J_j is the cost associated with scenario j, i.e., the cost of one branch from the root node $x(k)$ to its final leaf node $\hat{x}^j(k+N)$. The stage cost is $\ell(\cdot)$ and $V_\mathrm{f}(x)$ denotes the terminal cost function which is assumed to be the same for all scenarios. For a specific scenario, its probability to occur is given by ω_j. The sum over all probabilities has to be one, i.e., $\sum_{j=1}^{N_s} \omega_j = 1$.

2.1 Recursive Feasibility

The scenario formulation transforms the initially uncertain problem into a nominal MPC problem by spanning a scenario tree. Therefore, we can make use of nominal MPC theory to establish recursive feasibility and stability for MS-MPC. In particular, we adapt the classical approach of nominal MPC, see e.g. Findeisen et al. (2007) or Rawlings and Mayne (2009), where the terminal state ends in a control invariant terminal region and is penalized by a suitable terminal cost function. For the sake of clarity, we neglect the estimate nature, indicated by $\hat{\cdot}$, of the state and input prediction.

We assume that the following two assumptions hold:

Assumption 1. (Continuity). The functions $f(x,u,p)$, $\ell(x,u)$, and $V_\mathrm{f}(x)$ are continuous, with $f(0,0,p) = 0 \ \forall p \in \mathcal{P}$, $\ell(0,0) = 0$, and $V_\mathrm{f}(0) = 0$.

Assumption 2. (Constraints). The sets $\mathcal{X}$ and $\mathcal{X}_\mathrm{f} \subseteq \mathcal{X}$ are closed, and $\mathcal{U}$ is compact. Each set contains the origin.

Establishing recursive feasibility is equal to requiring that the terminal state of each scenario ends in a control invariant terminal region. We assume that this region is the same for all scenarios and denote it as *common terminal region* Ω_f (cf. Fig. 2).

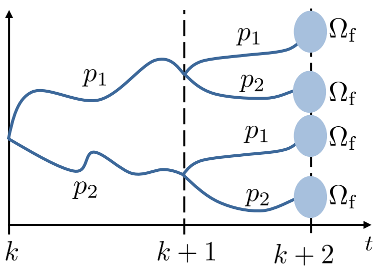


Fig. 2. Scenario tree with $s = 2$ and $N = 2$ and appended terminal region.

The question which arises is, how to define and obtain a common control invariant terminal region for all scenarios. To this end, we look at the problem in a different way. Each realization of the parameter $p \in \mathcal{P} = \{p_1, \ldots, p_s\}$ can be interpreted as a realization or instance of the given system, i.e.,

$$x^+ = f(x,u,p_j), \quad \forall j \in \{1,\ldots,s\}$$
$$\downarrow$$
$$x^+ = f_j(x,u), \quad \forall j \in \{1,\ldots,s\} \ .$$

This way, we can also reinterpret the scenario tree in such a way that the branching is caused by switching to another system realization (cf. Fig. 3). At each node of the tree, the state evolves according to the current system instance. This is also true at the terminal state. Assuming that the state is in the terminal region Ω_f at $i = N$, recursive feasibility translates to ensuring that the state stays in Ω_f for all possible system instances.

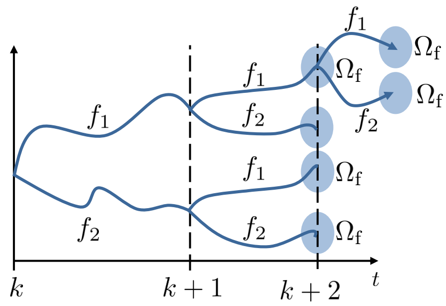


Fig. 3. Scenario tree using system instances.

Thus, a suitable terminal region must be invariant for the s different system instances.

Assumption 3. (Common terminal region). The common terminal region Ω_f is control invariant for $x^+ = f_j(x,u)$, $\forall j \in \{1,\ldots,s\}$ with $u \in \mathcal{U}$.

Proposition 4. (Recursive feasibility). Suppose that Assumptions 1, 2, and 3 hold, then the multi-scenario MPC is recursively feasible.

Proof. Following the proof for the nominal case and using the notation from Rawlings and Mayne (2009), let Ω_N be the set of all initial conditions for which a solution to the multi-scenario optimal control problem exists. The underlying idea is to set $\Omega_0 := \Omega_\mathrm{f}$ and to define

$$\Omega_1 = \left\{x \in \mathcal{X} \mid \exists u \in \mathcal{U} : \ f_j(x,u) \in \Omega_0 \quad \forall j \in \{1,\ldots,s\}\right\} .$$

So, Ω_1 is the set of all states for which a control can be found such that the state is driven to Ω_f, independent from the exact system instance. This implies that $\Omega_\mathrm{f} \subseteq \Omega_1$. Since Ω_f is control invariant by Assumption 3, it follows that Ω_1 is also control invariant. By backward recursion and induction, we arrive at the conclusion that Ω_N is positive invariant and hence, multi-scenario MPC is recursively feasible.

2.2 Stability

To establish stability, we need more assumptions on the stage and terminal costs, $\ell(\cdot)$ and $V_\mathrm{f}(x)$, respectively. Since the proposed formulation assigns an individual cost function to each scenario, it also provides the framework to include an individual *terminal* cost function $V_\mathrm{f}^j(x)$ to each scenario.

Assumption 5. (Basic stability assumption). For all $x \in \Omega_\mathrm{f}$ and for all $j \in \{1,\ldots,N_s\}$,

$$\min_{\tilde{\mathbf{u}}(k)\in\mathcal{U}} \left\{V_\mathrm{f}^j(f(x,u,p)) + \ell(x,u) \mid f(x,u,p) \in \Omega_\mathrm{f}\right\} \leq V_\mathrm{f}^j(x)$$

holds for all $p \in \mathcal{P}$.

This implies that, for each uncertainty realization, there exists an input such that the terminal region is control invariant, i.e., Assumption 5 implies Assumption 3. Note that Assumption 5 is satisfied, if each $V_\mathrm{f}^j(x)$ is a control Lyapunov function for all $p \in \mathcal{P}$. Furthermore, Assumption 5 only states that each terminal cost function $V_\mathrm{f}^j(x)$ has to satisfy the descent property, but not whether all $V_\mathrm{f}^j(x)$ have to be different or equal. Thus, if one Lyapunov function $V_\mathrm{f}(x)$ that satisfies this property can be found, it can also be used for all remaining scenarios. Hence, as in the case of the terminal region, we can use one *common terminal cost function* $V_\mathrm{f}(x)$. However, this is just a special case of the foregoing assumption. All $V_\mathrm{f}^j(x)$ can be different, as long as Assumption 5 is satisfied. Furthermore, we assume:

Assumption 6. (Bounds on stage and terminal costs). The stage cost $\ell(x,u)$ and the terminal costs $V_\mathrm{f}^j(x)$ satisfy

$$\ell(x,u) \geq \alpha_1(|x|) \quad \forall x \in \Omega_N,\ \forall u \in \mathcal{U}$$
$$V_\mathrm{f}^j(x) \leq \alpha_2^j(|x|) \quad \forall x \in \Omega_\mathrm{f} \text{ and } \forall j \in \{1,\ldots,N_s\},$$

in which $\alpha_1(x)$ and $\alpha_2^j(x)$ are $\mathcal{K}_\infty$ functions.

We can now formulate the following theorem:

Theorem 7. (Multi-scenario MPC stability). Suppose that Assumption 1, 2, 3, 5, and 6 are satisfied and that Ω_f contains the origin in its interior. Then, the origin is asymptotically stable with a region of attraction Ω_N for the system $x^+ = f_j(x, \kappa_N(x))$ for all $j \in \{1,\ldots,s\}$.

Proof. We again follow the structure of the proof for the nominal case, see e.g. Rawlings and Mayne (2009). The basic idea is that Assumptions 1, 2, and 3 need to be satisfied for recursive feasibility. Assumptions 5 and 6 ensure that the value function is a Lyapunov function for $x^+ = f_j(x, \kappa_N(x))$ for all $j \in \{1,\ldots,s\}$ on the domain Ω_N. Hence, the origin is asymptotically stable for all system instances.

2.3 Multi-scenario MPC formulation

The multi-scenario optimal control problem formulation used in MPC becomes

$$\begin{aligned}
\min_{\tilde{\mathbf{u}}(k)} \quad & V_N(x(k), \tilde{\mathbf{u}}(k)) \\
\text{s.t.} \quad & \forall i \in \mathcal{I}_{0:N-1} \text{ and } \forall j \in \mathcal{I}_{1:N_s} : \\
& \hat{x}^j(k+i+1) = f\big(\hat{x}^j(k+i), \hat{u}^j(k+i), p^j(k+i)\big) \\
& \hat{x}^j(k) = x(k) \\
& \hat{u}^j(k+i) \in \mathcal{U} \\
& \hat{x}^j(k+i) \in \mathcal{X} \\
& \hat{x}^j(k+N) \in \Omega_\mathrm{f} \\
& p^j(k+i) \in \mathcal{P} \\
& \Gamma \tilde{\mathbf{u}}(k) = 0\ ,
\end{aligned} \tag{7}$$

where $x(k)$ is the initial condition. The solution to (7) is denoted by $\tilde{\mathbf{u}}^*(k)$ and its first element $\hat{u}^*(k)$ is applied to the plant. Hence, MPC establishes the control law $u(k) = \kappa_N(x(k)) = \hat{u}^*(k)$. If the solution is inserted into the cost function, we obtain the so-called *value function* $V_N\big(x(k), \tilde{\mathbf{u}}^*(k)\big)$.

3. TERMINAL REGION COMPUTATION

In the previous section, we presented an approach that guarantees recursive feasibility and stability for multi-scenario MPC, using suitable terminal regions and terminal penalties. The determination of these ingredients is very challenging in the general nonlinear case. Therefore, we focus on the linear case in this section and show how to obtain a common terminal region and a common terminal cost function, such that Assumptions 3, 5, and 6 are satisfied. For that purpose, we use a quadratic function as the common terminal cost and a linear terminal control law. A level set of the cost function serves as the common terminal region. Constraint satisfaction is achieved by using the support function of the involved sets. The determination of the common terminal region is formulated as a semidefinite program.

3.1 Invariance of the Terminal Region

We consider a linear discrete time system of the form

$$\begin{aligned}
x^+ &= A_j x + B_j u, \quad \forall j \in \{1,\ldots,s\} \\
\text{s.t. } & x \in \mathcal{X} \\
& u \in \mathcal{U}\ ,
\end{aligned} \tag{8}$$

where the matrices A_j and B_j are realizations of the uncertain parameter $p \in \mathcal{P}$.

For the terminal controller, we choose the linear control law $u = \kappa_\mathrm{f}(x) = Kx$ with $K = RP$ and $P = P^\mathsf{T} > 0$ of appropriate dimensions. For the common terminal cost, we use the quadratic function $V_f(x) = x^\mathsf{T} P x$. The variables to be computed are the matrices R and P. The matrix P is determined such that $V_\mathrm{f}(x)$ is a Lyapunov function for all closed-loop system instances. Therefore, P must be positive definite and we need to ensure that $V_\mathrm{f}(x^+) - V_\mathrm{f}(x)$ is negative semidefinite for all system instances. Inserting the system (8) and algebraic reformulation yields

$$Q - (A_j Q + B_j R)^\mathsf{T} Q^{-1} (A_j Q + B_j R) \geq 0 \tag{9}$$

for all $j \in \{1,\ldots,s\}$ with $Q = P^{-1}$. The sought variables are Q and R and they appear in a nonlinear manner in (9), which makes their determination difficult. Therefore, the expression is reformulated into a set of linear matrix inequalities (LMIs) by means of the Schur complement (Boyd and Vandenberghe (2004)). We obtain

$$\begin{bmatrix} Q & (A_j Q + B_j R) \\ (A_j Q + B_j R)^\mathsf{T} & Q \end{bmatrix} \geq 0, \quad \forall j \in \{1,\ldots,s\}\ , \tag{10}$$

which is equivalent to (9) and hence, can be used instead of it.

This is a set of LMIs, which are affine in the unknowns Q and R. Hence, (10) can be solved efficiently and yields the matrix P that on the one hand makes $V_f(x) = x^\mathsf{T} P x$ a Lyapunov function for the closed-loop system and, on the other hand, defines the terminal control law. We choose a level set of the Lyapunov function as the common terminal region:

$$\Omega_\mathrm{f} := \left\{ x \in \mathbb{R}^{n_x} \mid x^\mathsf{T} P x \leq 1 \right\}\ . \tag{11}$$

Hence, $u = \kappa_\mathrm{f}(x) = Kx$ with $K = RP$ is a stabilizing terminal controller for all system instances on Ω_f. Note that (11) describes an ellipsoid in the state space.

3.2 Constraint Satisfaction

In the following, we assume that the constraints can be represented by polyhedral sets of the form

$$\mathcal{X} = \left\{ x \in \mathbb{R}^{n_x} \mid c_i^\mathsf{T} x \leq r_i,\ i = 1,\ldots,N_\mathcal{X} \right\} \tag{12a}$$
$$\mathcal{U} = \left\{ u \in \mathbb{R}^{n_u} \mid a_l^\mathsf{T} u \leq s_l,\ l = 1,\ldots,N_\mathcal{U} \right\}\ , \tag{12b}$$

where $N_{\mathcal{X}}$, $N_{\mathcal{U}} \in \mathbb{N}$ denote the respective numbers of inequalities by which the constraint sets are defined.

Satisfaction of state constraints translates to

$$\Omega_{\mathrm{f}} \subseteq \mathcal{X} \tag{13a}$$

and since $u = Kx \in \mathcal{U}$ for all $x \in \Omega_{\mathrm{f}}$, satisfaction of input constraints translates to

$$K\Omega_{\mathrm{f}} \subseteq \mathcal{U} \ . \tag{13b}$$

These are general set conditions which, in general, are hard to verify. In the case of closed convex sets, the *support function* of the involved sets allows us to reformulate conditions (13) as LMIs (Blanchini and Miani (2008); Kolmanovsky and Gilbert (1998)). The support function of an ellipsoidal set as defined by (11) is given by $h_{\Omega_{\mathrm{f}}}(\eta) = \sqrt{\eta^{\mathsf{T}} P^{-1} \eta}$ with $\eta \in \mathbb{R}^{n_x}$.

Given the two closed convex sets Ω_{f} and $\mathcal{X}$, checking the state constraint condition (13a), $\Omega_{\mathrm{f}} \subseteq \mathcal{X}$, is equivalent to checking if $h_{\Omega_{\mathrm{f}}}(\eta) \leq h_{\mathcal{X}}(\eta)$ holds for all η. In case that $\mathcal{X}$ is a polyhedral set as defined in (12), checking $\Omega_{\mathrm{f}} \subseteq \mathcal{X}$ simplifies even further to verifying that $h_{\Omega_{\mathrm{f}}}(c_i) \leq r_i$ holds for all $i \in \{1, \ldots, N_{\mathcal{X}}\}$. Putting all this together, we arrive at the equivalence

$$\Omega_{\mathrm{f}} \subseteq \mathcal{X} \quad \Leftrightarrow \quad \sqrt{c_i^{\mathsf{T}} P^{-1} c_i} \leq r_i, \quad \forall i \in \{1, \ldots, N_{\mathcal{X}}\} \ .$$

The right hand expression can be reformulated by means of the Schur complement into the set of LMIs given by

$$\begin{bmatrix} Q & (Qc_i) \\ (Qc_i)^{\mathsf{T}} & r_i^2 \end{bmatrix} \geq 0, \quad \forall i \in \{1, \ldots, N_{\mathcal{X}}\} \ , \tag{14}$$

with $Q = P^{-1}$. Hence, the state constraints are satisfied, if a Q can be found that solves (14). Note that the Q in (14) is the same as in (10).

The same procedure can be applied to transform the input constraint condition (13b) into a set of LMIs. We get

$$\begin{bmatrix} Q & (R^{\mathsf{T}} a_l) \\ (R^{\mathsf{T}} a_l)^{\mathsf{T}} & s_l^2 \end{bmatrix} \geq 0, \quad \forall l \in \{1, \ldots, N_{\mathcal{U}}\} \ , \tag{15}$$

with $Q = P^{-1}$. Hence, the input constraints are satisfied if a Q can be found that satisfies (15).

3.3 Optimization Problem

In order to find a common terminal region Ω_{f} with associated terminal controller $\kappa_{\mathrm{f}}(x)$, we use (10), (14), and (15) as constraints to an optimization problem whose objective it is to maximize the common terminal region. This can be achieved by maximizing $\log(\det(Q))$ (see Boyd (1994); Boyd and Vandenberghe (2004)):

$$\begin{aligned} \max_{Q,R} \ & \log\big(\det(Q)\big) \\ \text{s.t. } & Q = Q^{\mathsf{T}} > 0 \\ & \begin{bmatrix} Q & (A_j Q + B_j R)^{\mathsf{T}} \\ (A_j Q + B_j R) & Q \end{bmatrix} \geq 0, \quad \forall j \in \{1, \ldots, s\} \\ & \begin{bmatrix} Q & (Qc_i) \\ (Qc_i)^{\mathsf{T}} & r_i^2 \end{bmatrix} \geq 0, \quad \forall i \in \{1, \ldots, N_{\mathcal{X}}\} \\ & \begin{bmatrix} Q & (R^{\mathsf{T}} a_l) \\ (R^{\mathsf{T}} a_l)^{\mathsf{T}} & s_l^2 \end{bmatrix} \geq 0, \quad \forall l \in \{1, \ldots, N_{\mathcal{U}}\} \ . \end{aligned} \tag{16}$$

This optimization problem is a semidefinite program. It can be efficiently solved, e.g. in MATLAB together with the YALMIP toolbox.

We can now reformulate Theorem 7 for the linear case.

Theorem 8. (Linear multi-scenario MPC stability). Suppose that Assumptions 1 and 2 are satisfied and that a solution to (16) exists. Then, the origin is asymptotically stable with a region of attraction Ω_N for the system $x^+ = A_j x + B_j \kappa_N(x(k))$ for all $j \in \{1, \ldots, s\}$.

Proof. By construction, the solution to (16) yields the matrix P that makes $x^{\mathsf{T}} P x$ a common control Lyapunov function and Ω_{f} defined by (11) a common terminal region for the uncertain linear discrete time system (8). Hence, Assumptions 3, 5, and 6 are satisfied and Theorem 7 guarantees asymptotic stability of the origin of the closed loop system $x^+ = A_j x + B_j \kappa_N(x(k))$ for all $j \in \{1, \ldots, s\}$.

4. BUILDING CLIMATE CONTROL

We apply the MS-MPC approach (7) with the terminal ingredients computed by (16) to building climate control, which is a well-suited application for Model Predictive Control, see e.g. Oldewurtel et al. (2012). Since MPC is a control method that uses predictions to determine a suitable input, weather predictions could be included easily.

In our simulations, we use a model that is based on Bacher and Madsen (2011). It can account for external influences, such as ambient temperature and solar radiation. We focus on the four-states version of the model, adapted to resemble a cold storage house.

4.1 Model

The model equations, on the basis of Bacher and Madsen (2011), are

$$\dot{T}_{\mathrm{s}} = \frac{1}{T_{\mathrm{s}} C_{\mathrm{s}}} (T_{\mathrm{i}} - T_{\mathrm{s}}) \tag{17a}$$

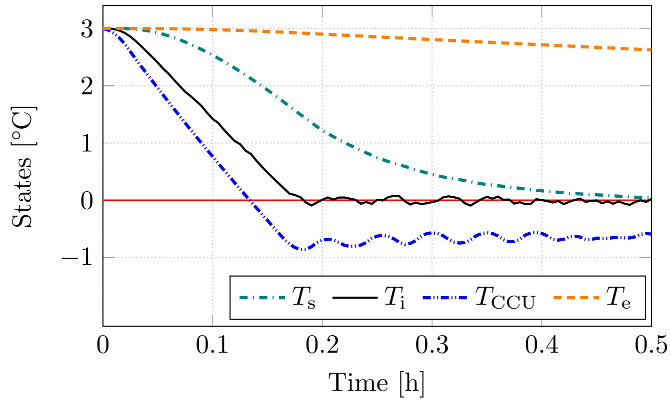
$$\begin{aligned} \dot{T}_{\mathrm{i}} = & \frac{1}{T_{\mathrm{s}} C_{\mathrm{i}}} (T_{\mathrm{s}} - T_{\mathrm{i}}) + \frac{1}{R_{\mathrm{ih}} C_{\mathrm{i}}} (T_{\mathrm{CCU}} - T_{\mathrm{i}}) \\ & + \frac{1}{R_{\mathrm{ie}} C_{\mathrm{i}}} (T_{\mathrm{e}} - T_{\mathrm{i}}) + \frac{1}{R_{\mathrm{ia}} C_{\mathrm{i}}} (T_{\mathrm{a}} - T_{\mathrm{i}}) \end{aligned} \tag{17b}$$

$$\dot{T}_{\mathrm{CCU}} = \frac{1}{R_{\mathrm{ih}} C_{\mathrm{h}}} (T_{\mathrm{i}} - T_{\mathrm{CCU}}) + \frac{1}{C_{\mathrm{h}}} \Phi_{\mathrm{CCU}} \tag{17c}$$

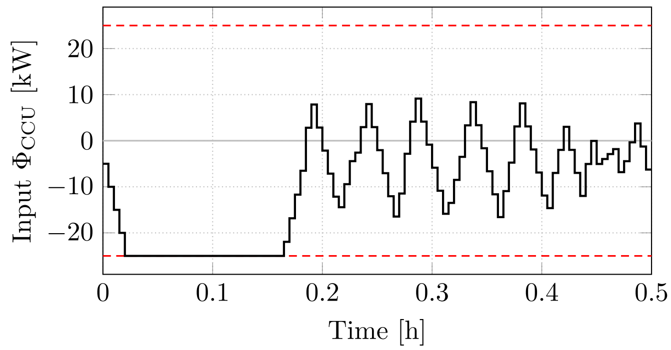
$$\dot{T}_{\mathrm{e}} = \frac{1}{R_{\mathrm{ie}} C_{\mathrm{e}}} (T_{\mathrm{i}} - T_{\mathrm{e}}) + \frac{1}{R_{\mathrm{ea}} C_{\mathrm{e}}} (T_{\mathrm{a}} - T_{\mathrm{e}}) \ , \tag{17d}$$

where T_{s}, T_{i}, T_{CCU}, and T_{e} are the temperatures of the sensor, the interior, the climate control unit (CCU), and the building envelope (walls). The input to the system is the CCU power Φ_{CCU}. Moreover, the model underlies external influences in the form of the ambient temperature T_{a}. Values for the model parameters can be found in Bacher and Madsen (2011).

We use this model for a cold storage house in which e.g. food is stored, and we consider the uncertain case of an open or closed door. This can be modeled as an uncertainty in the parameter $R_{\mathrm{ia}} \in \{1.77, 5.31\}$. The control task is to bring and keep the room temperature down to $0\,^{\circ}\mathrm{C}$ as close as possible, without going below

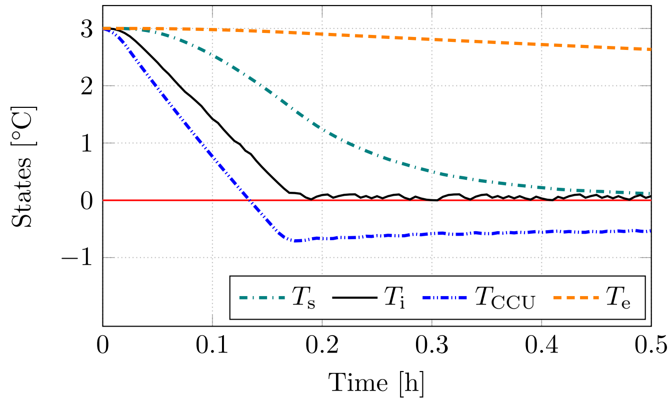


(a) Evolution of the states.

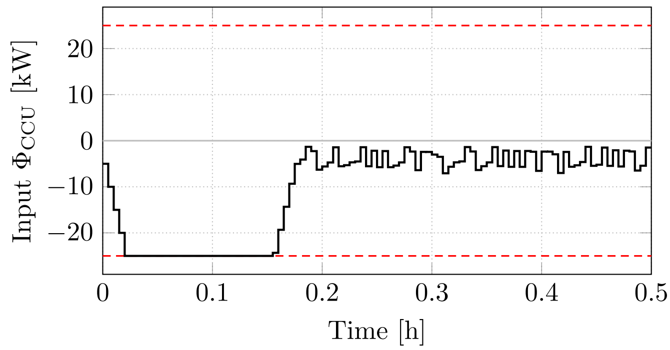


(b) Input signal Φ_{CCU}.

Fig. 4. Simulation results for the nominal case. The room temperature T_i violates the state constraints. The control input Φ_h underlies high fluctuations and switches between heating and cooling.



(a) Evolution of the states.



(b) Input signal Φ_{CCU}.

Fig. 5. Simulation results for the robust case. The room temperature T_i stays above the freezing point of 0 °C. The control input Φ_h varies only in a small range and does not switch between heating and cooling.

that freezing point, and without letting the stored goods get warmer than 10 °C, i.e., the room temperature T_i is constrained to $[0, 10]$ °C. For the cooling or heating, the climate control unit can provide a power Φ_{CCU} in the range of $[-25, 25]$ kW. Positive and negative values correspond to heating or cooling, respectively. Without loss of generality, we assume a constant ambient temperature of 3 °C. The various temperatures in the room (initial state) shall also be 3 °C, to show that the controllers are able to perform a set point change.

We compare a nominal MPC controller, which assumes the nominal value $R_{ia} = 1.77$ for the whole simulation time, with a scenario-based MPC controller, which does not make assumptions about the actual uncertainties. The actual uncertainty realization is described by a randomly generated sequence of the two parameter values of R_{ia}, i.e., the door is opened or closed randomly during the simulation.

4.2 Implementation

The continuous model is discretized using Euler's method with a sampling time of $t_s = 18$ s. The resulting discrete time system is used in a Quadratically-Constrained Quadratic Program (QCQP) which results from the regular linear MPC to QP reformulation, and the quadratic terminal constraint.

Both the nominal and the multi-scenario MPC controller use a prediction horizon of $N = 20$. The MS-MPC controller furthermore uses a robust horizon of $N_r = 3$, after which the scenario tree stops branching. In the considered example, this value is a good compromise between controller performance and computational burden. In general, the determination of N_r is still an open question. For details regarding the influence of N_r, see Lucia et al. (2013).

Using (16), the matrix P defining the terminal region for the considered case was computed to

$$P = 10^{-3} \cdot \begin{pmatrix} 7.40 & -0.96 & 0.61 & -0.03 & -7.03 \\ -0.96 & 17.37 & -7.73 & -1.88 & -6.80 \\ 0.61 & -7.73 & 22.60 & 4.15 & -19.63 \\ -0.03 & -1.88 & 4.15 & 9.87 & -12.11 \\ -7.03 & -6.80 & -19.63 & -12.11 & 103.33 \end{pmatrix}.$$

4.3 Simulation Results

Fig. 4a shows that the nominal MPC controller is not robust, i.e., it violates the state constraints, whereas the scenario-based MPC controller is feasible at all time steps and does not violate the state constraints (see Fig. 5a) and, hence, is robust. The "price" for the achieved robustness is a slight reduction of the performance, as the average room temperature will be above 0 °C.

The control inputs, shown in Fig. 4b and Fig. 5b, show equal behavior during the transition phase. Once the constraint is reached, the nominal MPC controller starts oscillating in magnitudes of up to 25 kW, whereas the MS-MPC controller only varies in a range of about 5 kW, resulting in less wear on the climate control unit.

Using MATLAB's `fmincon` on a standard desktop PC, the calculation times for solving the optimization problems in the MPC schemes lie in the range of seconds. Thus, real-time implementation of the MS-MPC controller is possible, given the used sampling time of 18 s.

5. SUMMARY AND CONCLUSION

This work focuses on recursive feasibility and stability of Multi-Scenario Model Predictive Control. We consider the uncertain parameter case, assuming discrete uncertainty realizations modeled as a scenario tree. In order to establish recursive feasibility and stability, we extend the classical dual mode approach of nominal MPC. This leads to the formulation of a common control invariant terminal region and a common terminal cost function for the optimal control problem.

For the computation of these ingredients for the linear case, we set up a semidefinite program accounting for the invariance property and the rigorous satisfaction of input and state constraints, given by closed and convex sets. We apply this approach in a building climate control scenario. The simulations show that the MS-MPC controller with the computed terminal region fulfills the task without violating the constraints, whereas a standard MPC controller violates the state constraints at several points.

Future works could investigate the computation of a terminal region and terminal cost function for certain classes of nonlinear systems, e.g. systems with sector-bounded nonlinearities. To extend the work on time-independent parameter uncertainties, further research could also explore the case of uncertain additive disturbances.

REFERENCES

Bacher, P. and Madsen, H. (2011). Identifying suitable models for the heat dynamics of buildings. *Energy and Buildings*, 43(7), 1511–1522.

Bertsekas, D.P. (2005). *Dynamic Programming and Optimal Control*, volume 1 of *Athena Scientific Optimization and Computation Series*. Athena Scientific, Belmont, Massachusetts, 3rd edition.

Blanchini, F. and Miani, S. (2008). *Set-Theoretic Methods in Control*. Systems & Control: Foundations & Applications. Birkhäuser, Boston.

Boyd, S.P. (1994). *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *SIAM Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia.

Boyd, S.P. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge.

Di Cairano, S. (2012). An industry perspective on MPC in large volumes applications: Potential benefits and open challenges. In *4th IFAC Nonlinear Model Predictive Control Conference*, 52–59. Noordwijkerhout.

Findeisen, R., Raff, T., and Allgöwer, F. (2007). Sampled-Data Nonlinear Model Predictive Control for Constrained Continuous Time Systems. In S. Tarbouriech, G. Garcia, and A.H. Glattfelder (eds.), *Advanced Strategies in Control Systems with Input and Output Constraints*, volume 346 of *Lecture Notes in Control and Information Sciences*, 207–235. Springer, Berlin.

Goodwin, G.C., Østergaard, J., Quevedo, D.E., and Feuer, A. (2009). A vector quantization approach to scenario generation for stochastic NMPC. In L. Magni, D.M. Raimondo, and F. Allgöwer (eds.), *Nonlinear Model Predictive Control*, volume 384 of *Lecture Notes in Control and Information Sciences*, 235–248. Springer, Berlin.

Kolmanovsky, I. and Gilbert, E.G. (1998). Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, 4(4), 317–367.

Lucia, S. and Engell, S. (2012). Multi-stage and two-stage robust nonlinear model predictive control. In *4th IFAC Nonlinear Model Predictive Control Conference*, 181–186. Noordwijkerhout.

Lucia, S., Finkler, T., and Engell, S. (2013). Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *Journal of Process Control*, 23(9), 1306–1319.

Lucia, S., Finkler, T.F., Basak, D., and Engell, S. (2012). A new robust NMPC scheme and its application to a semi-batch reactor example. In *8th IFAC Symposium on Advanced Control of Chemical Processes*, 69–74. Singapore.

Mayne, D.Q., Raković, S., Findeisen, R., and Allgöwer, F. (2006). Robust output feedback model predictive control of constrained linear systems. *Automatica*, 42(7), 1217–1222.

Mayne, D.Q., Raković, S., Findeisen, R., and Allgöwer, F. (2009). Robust output feedback model predictive control of constrained linear systems: Time varying case. *Automatica*, 45(9), 2082–2087.

Mayne, D.Q., Rawlings, J., Rao, C., and Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967–2986.

Oldewurtel, F., Parisio, A., Jones, C.N., Gyalistras, D., Gwerder, M., Stauch, V., Lehmann, B., and Morari, M. (2012). Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, 45, 15–27.

Qin, S. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764.

Rawlings, J.B. and Mayne, D.Q. (2009). *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison.

Streif, S., Kögel, M., Bäthge, T., and Findeisen, R. (2014). Robust nonlinear model predictive control with constraint satisfaction: A relaxation-based approach. In *19th IFAC World Congress*, 11073–11079. Cape Town.

