



Stability properties of the adaptive horizon multi-stage MPC

Zawadi Mdoe^a, Dinesh Krishnamoorthy^b, Johannes Jäschke^{a,*}

^a Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Sem Sælands vei 4, 7491, Trondheim, Norway

^b Department of Mechanical Engineering, Eindhoven University of Technology (TU/e), Groene Loper 15, 5600 MB, Eindhoven, Netherlands



ARTICLE INFO

Article history:

Received 10 October 2022

Received in revised form 3 April 2023

Accepted 20 May 2023

Available online 20 June 2023

Keywords:

Fast MPC

Multi-stage MPC

Computational delay

Parametric sensitivity

Stability

ABSTRACT

This paper presents an adaptive horizon multi-stage model-predictive control (MPC) algorithm. It establishes appropriate criteria for recursive feasibility and robust stability using the theory of input-to-state practical stability (ISpS). The proposed algorithm employs parametric nonlinear programming (NLP) sensitivity and terminal ingredients to determine the minimum stabilizing prediction horizon for all the scenarios considered in the subsequent iterations of the multi-stage MPC. This technique notably decreases the computational cost in nonlinear model-predictive control systems with uncertainty, as they involve solving large and complex optimization problems. The efficacy of the controller is illustrated using three numerical examples that illustrate a reduction in computational delay in multi-stage MPC.

© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

This paper investigates a way to reduce the computational cost of model predictive control (MPC). MPC has become increasingly popular in process systems engineering due to its inherent capacity to handle complex constrained multivariate systems [1]. There has been a broader interest in nonlinear MPC (NMPC) because of the significant nonlinear dynamics in process systems. However, this requires solving a nonlinear program (NLP) which due to its numerical complexity may result in a large computation cost [2]. The computation cost is further increased when uncertainty is explicitly considered, in robust MPC approaches, such as in multi-stage MPC formulations.

Multi-stage MPC samples from an uncertain parameter set to obtain a finite set of discrete realizations and then propagate the uncertainty into the future as a growing scenario tree [3,4]. The size of the optimization problem increases exponentially as the prediction horizon expands, eventually becoming computationally intractable. As a result, a significant computational delay occurs, affecting control performance. This poses a major challenge in implementing multi-stage MPC. de la Peña et al. [5] and Lucia et al. [6] suggest using a robust horizon to decrease the number of variables, but an additional reduction may be required when there is a substantial number of parameter realizations. Reducing the problem size minimizes computational delay and therefore improves the multi-stage MPC controller performance.

Several methods have been proposed in MPC literature to enhance computational efficiency. One approach is the adaptive horizon MPC which aims at providing an online horizon update for subsequent MPC iterations. This framework shortens the prediction horizon as the controlled system approaches a stabilizing region. The problem size is reduced, leading to shorter solution times while preserving the MPC stability properties. Krener [7,8] proposed using the properties of the Lyapunov function to detect stabilization before shortening the horizon. This method requires the determination of a good Lyapunov function, which can be difficult for nonlinear systems. Griffith et al. [9] employs linearization around the operating point to establish a stabilizing region where the horizon can be truncated. Despite their successful implementation in standard (nominal) MPC, the adaptive horizon techniques have not yet been employed in multi-stage MPC. Multi-stage MPC has significantly larger solution times than standard MPC because of the explicit consideration of multiple scenarios in its optimization problem. Hence, the adaptive horizon strategy is even more beneficial for multi-stage MPC, particularly for nonlinear systems.

In this paper, we extend the adaptive horizon scheme proposed in [9] for multi-stage MPC. This approach requires performing one-step-ahead predictions, terminal region calculations, and selection of the sufficient (stabilizing) prediction horizon length. Although this was studied for the standard MPC in [9], the extension of these ideas to multi-stage MPC where we have multiple scenarios is not trivial, because (i) the determination of terminal ingredients depends on the nature of the scenario tree, and (ii) the horizon update using parametric sensitivity needs to ensure that the one-step-ahead approximate solution for each scenario reaches their stabilizing terminal regions. A proof of

* Corresponding author.

E-mail addresses: zawadi.n.mdoe@ntnu.no (Z. Mdoe),

d.krishnamoorthy@tue.nl (D. Krishnamoorthy), johannes.jaschke@ntnu.no (J. Jäschke).

concept was very briefly presented in our earlier work [10]. This paper extends our previous work by describing the approach in full detail, providing a robust stability analysis of the resulting controller based on Input-to-State practical Stability (ISpS), and demonstrating the performance on a more complex case study.

The remainder of the paper is organized as follows. Sections 2 and 3 define the standard MPC and the fixed horizon multi-stage MPC, respectively. Section 4 presents the adaptive horizon multi-stage MPC showing how terminal ingredients are computed, and the horizon update algorithm for multi-stage MPC. The main results are in Section 5 where the recursive feasibility and stability properties of adaptive horizon multi-stage MPC are presented for both cases when the scenario tree is fully branched or has a robust horizon. Section 6 presents three numerical examples to show the performance of the adaptive horizon multi-stage MPC controller. The first example demonstrates its control performance with recursive feasibility and robust stability guarantees, and the remaining examples compare the controller performance against the fixed horizon multi-stage MPC for different robust horizons. Finally, Section 7 outlines the main findings and future work avenues.

The uncertain process is represented in discrete-time form by the following difference equation:

$$x_{k+1} = f(x_k, u_k, d_k) \quad (1)$$

where $x_k \in \mathbb{X} \subset \mathbb{R}^{n_x}$ are the states, $u_k \in \mathbb{U} \subset \mathbb{R}^{n_u}$ are the control inputs, and $d_k \in \mathbb{D} \subset \mathbb{R}^{n_d}$ are the uncertain parameters (disturbances) at time step t_k where $k \geq 0$ is the time index. The set of real numbers is denoted by \mathbb{R} , and the set of integers by \mathbb{Z} , with the subscript $+$ referring to their non-negative subspaces. The Euclidean vector norm is denoted as $\|\cdot\|$ and the corresponding induced matrix norm as $\|\cdot\|$. The function $f: \mathbb{R}^{n_x+n_u+n_d} \mapsto \mathbb{R}^{n_x}$ is assumed to be twice differentiable in all its arguments, has Lipschitz continuous second derivatives, and is assumed to be transformed to obtain the nominal steady state model as $f(0, 0, d^0) = 0$ where d^0 is the nominal disturbance.

2. Standard MPC

We define the standard MPC as follows. For system (1) the controller receives the current state x_k and assumes a nominal parameter d^0 . The optimal control problem (OCP) for standard MPC is a parametric NLP in x_k and d^0 written as follows:

$$V_N^{\text{nom}}(x_k, d^0) = \min_{z_i, v_i} \psi(z_N, d^0) + \sum_{i=0}^{N-1} \ell(z_i, v_i, d^0) \quad (2a)$$

$$\text{s.t. } z_{i+1} = f(z_i, v_i, d^0), \quad i = 0, \dots, N-1 \quad (2b)$$

$$z_0 = x_k, \quad (2c)$$

$$z_i \in \mathbb{X}, \quad v_i \in \mathbb{U}, \quad (2d)$$

$$z_N \in \mathbb{X}_f \quad (2e)$$

where N is the prediction horizon, V_N^{nom} is the optimal cost, z_i and v_i are the state and input variable vectors, respectively at time t_{k+i} . The objective (2a) consists of a stage cost $\ell: \mathbb{R}^{n_x+n_u+n_d} \mapsto \mathbb{R}$, and a terminal cost $\psi: \mathbb{R}^{n_x+n_d} \mapsto \mathbb{R}$.¹ The initial condition (2c) is the current state x_k . The Eqs. (2b) are constraints predicting system dynamics, and (2d) are the bounds on state and input variables. The terminal constraints are in (2e), where $\mathbb{X}_f \subset \mathbb{X}$ is the terminal region set.

The standard MPC receives x_k and solves (2) for an optimal input sequence $u_{[0, N-1]}^*$ and their corresponding state predictions

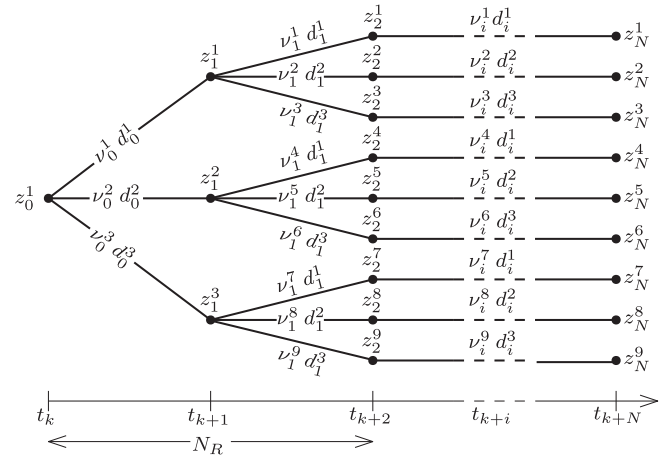


Fig. 1. A scenario tree with nine scenarios, prediction horizon N , and robust horizon $N_R = 2$. The dashed lines represent all the nodes in between t_{k+2} and t_{k+N} .

$z_{[0, N]}^*$ in the time interval $(t_k, t_{k+N}]$ at every iteration k . The first stage input from the sequence is applied to the plant as u_k . Then the prediction horizon is shifted one step, and the problem is re-optimized with the new state x_{k+1} .

Since the current optimal input is obtained with respect to x_k , the standard MPC policy yields a feedback control law $u_k = \kappa(x_k)$, where $\kappa: \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_u}$. This gives standard MPC with limited robustness to model uncertainty [11–14]. However, standard MPC exhibits a substantial decline in performance for significant plant-model mismatch due to uncertainty.

3. Multi-stage MPC

Unlike standard MPC, multi-stage MPC explicitly takes into account the model uncertainty. Multi-stage MPC models uncertainty as a tree of discrete scenarios (called a scenario tree) used to depict the evolution of uncertainty along the prediction horizon [3,6]. To reduce the problem size in the scenario tree that grows exponentially with the prediction horizon, it is often proposed to limit the branching of the tree only up to a certain stage, called the robust horizon [5,6]. Fig. 1 shows a scenario tree with a robust horizon $N_R = 2$ and nine scenarios, where branching stops at t_{k+N_R} and the parameter realizations are kept constant until t_{k+N} . The number of scenarios is given by $\text{card}(\mathbb{C}) = (N_D)^{N_R}$, where \mathbb{C} is the scenario index set and N_D is the number of discrete parameter realizations. The problem size is significantly reduced when $N_R \ll N$, and the fully branched scenario tree is recovered by setting $N_R = N$.

Given x_k at t_k multi-stage MPC requires solving the following problem:

$$V_N^{\text{ms}}(x_k, d^c) = \min_{z_i^c, v_i^c} \sum_{c \in \mathbb{C}} \omega_c \left(\psi(z_N^c, d_{N-1}^c) + \sum_{i=0}^{N-1} \ell(z_i^c, v_i^c, d_i^c) \right) \quad (3a)$$

$$\text{s.t. } z_{i+1}^c = f(z_i^c, v_i^c, d_i^c), \quad i = 0, \dots, N-1 \quad (3b)$$

$$z_0^c = x_k, \quad (3c)$$

$$v_i^c = v_i^{c'}, \quad \{(c, c') \mid z_i^c = z_i^{c'}\} \quad (3d)$$

$$d_{i-1}^c = d_i^c, \quad i \geq N_R \quad (3e)$$

$$z_i^c \in \mathbb{X}, \quad v_i^c \in \mathbb{U}, \quad d_i^c \in \mathbb{D}, \quad (3f)$$

$$z_N^c \in \mathbb{X}_f, \quad (3g)$$

$$\forall c, c' \in \mathbb{C}$$

¹ We have chosen to include the parameter d in the stage cost. This could, for example, model changes in the setpoint, or other cost function parameters that are set by the user.

where z_i^c , v_i^c , and d_i^c represent the state, input, and disturbance vectors at stage i and scenario c . The set \mathbb{C} is the scenario index set, and the nominal scenario has a zero index. The objective function (3a) is the weighted sum of the stage and terminal costs across all the scenarios with ω_c being the probability for each scenario. Non-anticipativity constraints (NACs) ensuring that inputs from a common parent node in the scenario tree (Fig. 1) must be equal are given by (3d). This is because it is only possible to make a single control action $u_k = v_0^*$ on the plant. (3e) is included to show that parameter realizations are constant after the robust horizon. The state and control inputs inequality constraints across all scenarios are given in (3f) and \mathbb{D} is the set of the discrete parameter realizations.

Problem (3) is a parametric optimization problem in x_k and \mathbf{d}^c , where \mathbf{d}^c denotes $\mathbf{d}_{i=0, \dots, N-1}^{c=1, \dots, \text{card}(\mathbb{C})}$ which is the concatenated vector of all parameter realizations in the scenario tree (across all scenarios and across all time steps in the prediction horizon).

4. Adaptive horizon multi-stage MPC

Adaptive horizon algorithms provide an update for the prediction horizon length of the MPC problem. The prediction horizon is not an optimization variable in these approaches, resulting in faster horizon updates compared to solving a mixed-integer NLP (MINLP) in variable horizon MPC approach [15]. The adaptive horizon MPC has been implemented for reference tracking in [9], and then extended for economic MPC in [16]. In this paper, we aim to extend the algorithm in [9] to multi-stage MPC for reference tracking objectives.

4.1. Variable transformation

Setpoint tracking under uncertainty implies tracking different implicit references for each scenario. Therefore, the solution to (3) has states and control inputs tracking different implicit references for each parameter realization. It is possible to enforce a common zero by reformulating (3) with transformed variables. But first, let us define the following system properties:

Definition 1 (*Mapping scenarios to parameter realizations*). There exists a function $r: \mathbb{C} \mapsto \mathbb{Q}$ mapping any scenario c to the index of the parameter realization at its leaf node, where \mathbb{Q} is the set of parameter realization indices.

Definition 2 (*Stable and optimal equilibrium pairs for each parameter realization*). For any given parameter realization $d^r \in \mathbb{D}$, a state-input pair $(x^r, u^r) \in \mathbb{X} \times \mathbb{U}$ is a *stable equilibrium pair* for system (1) if $x^r = f(x^r, u^r, d^r)$ holds. Further, if it yields the lowest stage cost among all equilibrium points, then it is the *optimal equilibrium pair* (x_f^r, u_f^r) .

Then let us make the following assumption.

Assumption 1. There exists an optimal equilibrium pair (x_f^r, u_f^r) for each parameter realization $d^r \in \mathbb{D}$. The optimal equilibrium pairs for each parameter realization are the implicit references for the corresponding scenarios.

Assumption 1 implies that there exists a solution to the steady state optimization problem of (1) for each parameter realization $d^r \in \mathbb{D}$.

To enforce a common zero to all the scenarios, reformulate (3) using the following:

$$z_i^c = z_i^c - x_f^r, \quad v_i^c = v_i^c - v_f^r \quad (4)$$

for all $c \in \mathbb{C}$, and for all $r \in \mathbb{Q}$. The transformed system model becomes:

$$\bar{z}_{i+1}^c = \bar{f}(\bar{z}_i^c, \bar{v}_i^c, d_i^c) = f(\bar{z}_i^c + x_f^r, \bar{v}_i^c + u_f^r, d_i^c) - x_f^r \quad (5)$$

such that $\bar{z}_i^c \in \bar{\mathbb{X}}$, $\bar{v}_i^c \in \bar{\mathbb{U}}$, where $\bar{\mathbb{X}}$ and $\bar{\mathbb{U}}$ are the corresponding new feasible sets of the transformed system. Eqs. (4) and (5) imply when $(\bar{z}_i^c, \bar{v}_i^c) = (0, 0)$ we have $x_f^r = f(x_f^r, u_f^r, d_i^c)$ for all $c \in \mathbb{C}$, and for all $r \in \mathbb{Q}$. We define transformed stage costs as:

$$\bar{\ell}(\bar{z}_i^c, \bar{v}_i^c, d_i^c) = \ell(\bar{z}_i^c + x_f^r, \bar{v}_i^c + u_f^r, d_i^c) - \ell(x_f^r, u_f^r, d_i^c) \quad (6)$$

and transformed terminal costs as:

$$\bar{\psi}(\bar{z}_N^c, d_{N-1}^c) = \psi(\bar{z}_N^c + x_f^r, d_{N-1}^c) - \psi(x_f^r, d_{N-1}^c) \quad (7)$$

such that $\bar{\psi}(0, d_{N-1}^c) = \bar{\ell}(0, 0, d_i^c) = 0$ for all $c \in \mathbb{C}$.

The problem (3) is transformed using Eq. (4)–(7) to obtain a common terminal region containing zero. This will be useful to establish closed-loop stability later in Section 5.

To keep the notation simple, we will continue using the original notation. However, from here on, all variables and functions in Eq. (3) are assumed to be transformed.

Now that the desired system properties have been defined, the next subsection discusses the methodology to compute the terminal ingredients for each parameter realization in the multi-stage MPC.

4.2. Terminal ingredients for the adaptive horizon multi-stage MPC

The adaptive horizon multi-stage MPC requires (i) a terminal cost ψ , and (ii) a positive invariant terminal region \mathbb{X}_f . The terminal cost is the weighted average of the terminal costs across all parameter realizations. For each parameter realization, a terminal cost function is computed beforehand, based on a linearized system about its optimal equilibrium pair [17]. The terminal region is assumed to be the region where a terminal control law on the nonlinear system has a negligible error. But the determination of the terminal control law depends on whether the scenario tree is fully branched or has a robust horizon.

4.2.1. Terminal ingredients for a scenario tree with a robust horizon

First, consider when the scenario tree has a robust horizon i.e. $N_R < N$. The scenario tree has no NACs at the leaf nodes leading to decoupled terminal control input variables (v_{N-1}^c) across all the scenarios. Different stabilizing control laws may exist inside the terminal region for each parameter realization. Then the terminal regions for each parameter realization are approximated independently. We use an infinite horizon LQR for the control law in the terminal region and the terminal cost.

Assumption 2. Given a parameter realization $d^r \in \mathbb{D}$ for all $r \in \mathbb{Q}$, there exists a stabilizing LQR with a local control law $h_f^r(x) := u_f^r - K_r x$ such that $f(x, h_f^r(x), d^r) \in \mathbb{X}_f^r$ for all $x \in \mathbb{X}_f^r$.

Assumption 2 considers that different LQR controllers may exist in the terminal region across the different scenarios.

For a given parameter realization d^r , linearize (1) about (x_f^r, u_f^r) and write the resulting system as a sum of the linear and nonlinear terms as follows:

$$x_{k+1} = A_{K_r} x_k + \phi(x_k, h_f^r(x_k), d^r) \quad (8)$$

where $A_{K_r} = A_r - B_r K_r$, $A_r = \nabla_x^T f(0, 0, d^r)$ and $B_r = \nabla_u^T f(0, 0, d^r)$ and (A_r, B_r) is stabilizable, $\phi: \mathbb{X} \times \mathbb{U} \times \mathbb{D} \mapsto \mathbb{X}$ is the nonlinear part of the dynamics. The infinite horizon LQR for the linearized system with the parameter realization d^r is:

$$\begin{aligned} \psi(x_k, d^r) &= x_k^T P_r x_k = \min_{v_i} \sum_{i=0}^{\infty} (z_i^T Q z_i + v_i^T R v_i) \\ \text{s.t. } z_0 &= x_k, \quad z_{i+1} = A_r z_i + B_r v_i, \quad \forall i = 0, \dots, \infty \end{aligned} \quad (9)$$

where $Q \succ 0$, $R \succ 0$ are tuning matrices. $P_r \succ 0$ is the Riccati matrix for d^r .

To determine the terminal region for the LQR on the nonlinear system, there must exist an upper bound on the nonlinear effects for any parameter realization given by [Lemma 1](#).

Lemma 1. For any parameter realization $d^r \in \mathbb{D}$, there exists $M_r, q_r \in \mathbb{R}_+$ such that $|\phi(x, d^r)| \leq M_r |x|^{q_r}, \forall x \in \mathbb{X}$ where $\phi(x, d^r) := \phi(x, h_f^r(x), d^r)$.

Proof. Consider a single d^r and then follow the proof of Theorem 14 in [\[9\]](#). \square

Analytical determination of the linearization error and its bound is tedious or impossible. The bounds are determined offline by one-step simulations instead. One-step simulations from random initial states are used to obtain the linearization error numerically by evaluating for each d^r , the difference $\phi(x, d^r) = f(x, h_f^r(x), d^r) - A_{K_r}x, \forall x \in \mathbb{X}$.

Assumption 3. For a given d^r , there exists a terminal region of attraction \mathbb{X}_f^r around (x_f^r, u_f^r) such that $\mathbb{X}_f^r := \{z \mid |z - x_f^r| \leq c_f^r\}$ where c_f^r is the terminal radius corresponding to d^r .

[Assumption 3](#) means that the terminal region set for each parameter realization is approximated by the interior of an n -sphere with radius c_f^r . After fitting M_r and q_r for all parameter realizations according to [Lemma 1](#), and [Assumption 3](#) holds, the terminal region radii are computed using [Lemma 2](#).

Lemma 2. Suppose [Assumption 3](#) holds, the terminal radii c_f^r for all $d^r \in \mathbb{D}$ depend on their corresponding linearization error bounds which are given by:

$$c_f^r := \left(\frac{-\bar{\sigma}_r A_r + \sqrt{(\bar{\sigma}_r A_r)^2 + (\underline{\lambda}_{W_r} - \epsilon_{lQ}) A_r}}{\Lambda_r M_r} \right)^{\frac{1}{q_r-1}} \quad (10)$$

where $\bar{\sigma}_r$ is the maximum singular value of A_{K_r} , $\bar{\lambda}_{W_r}$ and $\underline{\lambda}_{W_r}$ are the maximum and minimum eigenvalues of $W_r := Q + K_r^\top R K_r$, $\Lambda_r := \frac{\bar{\lambda}_{W_r}}{(1-\bar{\sigma}_r)^2}$, and $\epsilon_{lQ} > 0$ is a small constant: an allowable tolerance for the terminal cost $\psi(x, d^r)$.

Proof. As in [Lemma 1](#), consider a single d^r and use the proof from [\[9\]](#). \square

4.2.2. Terminal ingredients for a fully branched scenario tree

Let us consider the case when the scenario tree is fully branched i.e. $N_R = N$. The stabilizing control law for each parameter realization inside the terminal region cannot be determined independently due to the NACs at the leaf node. Therefore, a common terminal control law must exist (i.e. $h_f^r(x) = \kappa_f(x)$, for all r).

After linearizing [\(1\)](#) as outlined in Section 4.2.1, the following steps are used to determine the terminal ingredients for the adaptive horizon multi-stage MPC with a fully branched scenario tree:

1. Select matrices Q , and R , for the LQR and choose a K such that the common terminal control law $\kappa_f(x) = -Kx$ is stabilizing for all parameter realizations r , and $-Kx \in \mathbb{U}, \forall x \in \mathbb{X}$,
2. Solve for P_r in the discrete Lyapunov equation $A_r^\top P_r A_r + Q = P_r^\top$, for each parameter realization r ,
3. Use Q, R, A_{K_r} to obtain the terminal radii for each parameter realization r by evaluating [\(10\)](#) in [Lemma 2](#) where $A_{K_r} = A_r - B_r K$. Then pick the smallest as the common terminal region's radius,

4. Obtain the terminal cost function as the weighted sum $\psi(x) = \sum_r x^\top P_r x$.

Steps 3 and 4 above are similar procedures as in Section 4.2.1 only that K is common for all realizations r .

4.3. Problem formulation

Now that the terminal ingredients for each parameter realization have been defined, we present the formulation for the adaptive horizon multi-stage MPC problem. The formulation is obtained by replacing the fixed horizon N in [\(3\)](#) with a variable horizon N_k as follows:

$$V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^c) = \min_{z_i^c, v_i^c} \sum_{c \in \mathbb{C}} \omega_c \left(\psi(z_{N_k}^c, d_{N_k-1}^c) + \sum_{i=0}^{N_k-1} \ell(z_i^c, v_i^c, d_i^c) \right) \quad (11a)$$

$$\text{s.t. } z_{i+1}^c = f(z_i^c, v_i^c, d_i^c), \quad i = 0, \dots, N_k - 1 \quad (11b)$$

$$z_0^c = x_k, \quad (11c)$$

$$v_i^c = v_i^{c'}, \quad \{(c, c') \mid z_i^c = z_i^{c'}\} \quad (11d)$$

$$d_{i-1}^c = d_i^c, \quad i \geq N_R \quad (11e)$$

$$z_i^c \in \mathbb{X}, \quad v_i^c \in \mathbb{U}, \quad d_i^c \in \mathbb{D}, \quad (11f)$$

$$z_N^c \in \mathbb{X}_f^c, \quad (11g)$$

$$\forall c, c' \in \mathbb{C}$$

where $V_{N_k}^{\text{ahm}}$ is the optimal cost, and N_k is the prediction horizon of the current MPC iteration given by a horizon update algorithm. The horizon update algorithm requires one-step-ahead predictions obtained from parametric NLP sensitivities that are discussed in the next subsection.

4.4. NLP sensitivity

Parametric NLP sensitivities are of our interest because problem [\(11\)](#) is parametric in x_k and \mathbf{d}^c , and we will use them in the horizon update algorithm in Section 4.5, and for stability analysis in Section 5. Problem [\(11\)](#) can be written as a general parametric NLP in the form:

$$\min_w J(w, p) \quad \text{s.t.} \quad h(w, p) = 0, \quad g(w, p) \leq 0 \quad (12)$$

where, $w \in \mathbb{R}^{n_w}$ is the vector of all optimization variables (z_i^c, v_i^c), and $p \in \mathbb{R}^{n_p}$ is the vector of NLP parameters (x_k, \mathbf{d}^c). The problem [\(12\)](#) has a scalar objective function $J: \mathbb{R}^{n_w+n_p} \mapsto \mathbb{R}$, equality constraints $h: \mathbb{R}^{n_w+n_p} \mapsto \mathbb{R}^{n_e}$, and inequality constraints $g: \mathbb{R}^{n_w+n_p} \mapsto \mathbb{R}^{n_i}$. Then the function $\mathcal{L}(w, \lambda, \mu, p) = J(w, p) + \lambda^\top h(w, p) + \mu^\top g(w, p)$ is the Lagrange function of [\(12\)](#) where, λ and μ are the Lagrange multipliers of appropriate dimension.

A point w^* that [\(12\)](#) satisfies the Karush-Kuhn-Tucker conditions is known as a KKT point [\[18\]](#). Given a parameter p , and a point w^* that satisfy:

$$\begin{aligned} \nabla_w \mathcal{L}(w^*, \lambda^*, \mu^*, p) &= 0 \\ h(w^*, p) &= 0 \\ g(w^*, p) &\leq 0 \\ 0 &\leq \mu^* \perp g(w^*, p) \leq 0 \end{aligned} \quad (13)$$

for some multipliers (λ, μ) , where w^* is called a KKT point. The set of all multipliers λ and μ that satisfy the KKT conditions for a certain parameter p is denoted as $\mathcal{M}(p)$. The active constraint set is given by $\mathcal{A}(w^*) = \{j \mid g_j(w^*, p) = 0\}$. \perp is the complementarity operator, i.e. either μ or g (or both) must be zero. Then we define strict complementarity as follows.

Definition 3 (SC, [18]). At the KKT point w^* of (12) with multipliers (λ^*, μ^*) , strict complementarity (SC) condition holds if $\mu_j + g_j(w^*, p) > 0$ for all $j \in \mathcal{A}(w^*)$.

For the KKT conditions to be first-order necessary optimality conditions, the KKT point w^* requires a constraint qualification to be satisfied. Further, it requires a constraint qualification to be a local minimizer of (12). The following is the definition of one well-known constraint qualification.

Definition 4 (LICQ, [18]). The linear independence constraint qualification (LICQ) holds at w^* when the gradient vectors $\nabla h_i(w^*, p)$ and $\nabla g_j(w^*, p)$ for all i , and for all $j \in \mathcal{A}(w^*)$ are linearly independent. LICQ implies that the multipliers λ^*, μ^* are unique i.e. $\mathcal{M}(p)$ is a singleton.

Given LICQ, the KKT conditions are necessary but not sufficient conditions for optimality. A second-order condition is needed to guarantee a minimizer:

Definition 5 (SSOSC, [19]). The strong second-order sufficient condition (SSOSC) holds at w^* with multipliers λ^* and μ^* if $q^\top \nabla_{ww}^2 \mathcal{L}(w^*, \lambda^*, \mu^*, p) q > 0$ for all vectors $q \neq 0$, such that $\nabla h_i(w^*, p)^\top q = 0$, for all i and $\nabla g_j(w^*, p)^\top q = 0$ for all $j \in \mathcal{A}(w^*) \cap \{j \mid \mu_j > 0\}$.

Assuming SC, LICQ, and SSOSC and applying the Implicit Function Theorem (IFT) to the KKT conditions in (13) leads to [Theorem 3](#).

Theorem 3 (IFT applied to KKT conditions). Assume a KKT point $w^*(p_0)$ that satisfies (13), and that SC, LICQ and SSOSC hold at $w^*(p_0)$. Furthermore, the functions F , h , and g are at least $k+1$ times differentiable in w and k times differentiable in p . Then the primal-dual solution $s^*(p)^\top = [w^*(p)^\top, \lambda^*(p)^\top, \mu^*(p)^\top]^\top$ has the following are the properties:

- $s^*(p_0)$ is an isolated local minimizer of (12) at p_0 and contains unique multipliers $(\lambda^*(p_0), \mu^*(p_0))$.
- For p in a neighborhood of p_0 the active constraint set $\mathcal{A}(w^*)$ remains unchanged.
- For p in a neighborhood of p_0 there exists a unique, continuous and differentiable function $s^*(p)$ which is a local minimizer satisfying SSOSC and LICQ for (12) at p .
- There exists positive Lipschitz constants L_s, L_v such that $|s^*(p) - s^*(p_0)| \leq L_s |p - p_0|$ and $|J(p) - J(p_0)| \leq L_v |p - p_0|$, where $|\cdot|$ is the Euclidean norm.

Proof. see Fiacco [19]. \square

The theorem above establishes Lipschitz continuity in the optimal solution and optimal objective function of (11) with respect to x_k and d^c . This result is important for the stability analysis. [Theorem 3](#) ensures we can always obtain from (13) the following linear system for NLP sensitivity if SC, LICQ, and SSOSC hold at a KKT point $w^*(p)$,

$$\begin{bmatrix} \nabla_{ww}^2 \mathcal{L} & \nabla h & \nabla g_{\mathcal{A}} \\ \nabla h^\top & 0 & 0 \\ \nabla g_{\mathcal{A}}^\top & 0 & 0 \end{bmatrix} \begin{bmatrix} \nabla_p w^* \\ \nabla_p \lambda^* \\ \nabla_p \mu_{\mathcal{A}}^* \end{bmatrix} = \begin{bmatrix} \nabla_{wp}^2 \mathcal{L} \\ \nabla h^\top \\ \nabla g_{\mathcal{A}}^\top \end{bmatrix} \quad (14)$$

where all the derivatives of are evaluated at w^*, p_0 , $(\lambda^*, \mu^*) \in \mathcal{M}(p_0)$, and $g_{\mathcal{A}}$ and $\mu_{\mathcal{A}}^*$ are vectors with elements g_j and μ_j^* for all $j \in \mathcal{A}(w^*)$, respectively. The Taylor expansion of the primal solution at p_0 is $w^*(p) = w^*(p_0) + \nabla_p w^*^\top (p - p_0) + \mathcal{O}(|p - p_0|^2)$. As a result, a first-order approximation is obtained in the neighborhood of p_0 using:

$$w^*(p) \approx w^*(p_0) + \nabla_p w^*^\top (p - p_0) \quad (15)$$

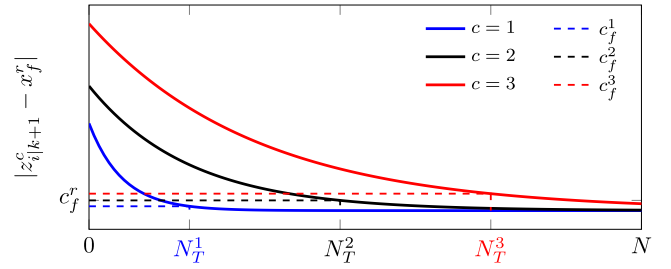


Fig. 2. Sensitivity prediction (solid lines) of (11) for three scenarios at t_{k+1} , and the terminal regions (below the dashed lines). N_{k+1} will be reduced from N to at least N_T^3 .

The KKT matrix in (14) is evaluated at p_0 and is obtained for free from the previous solution. The benefit of the sensitivity step (15) is to perform one-step-ahead predictions of the solution at the subsequent MPC iteration with a minimal computational effort compared to solving the NLP from scratch. The following subsection presents the horizon update algorithm for adaptive horizon multi-stage MPC using the sensitivity update.

4.5. Horizon update algorithm

The algorithm begins by selecting a long horizon N_{\max} , a safety factor $N_{\min} > N_R$, and computing the terminal ingredients offline. In each MPC iteration, the algorithm determines a horizon update such that all scenarios reach their terminal regions irrespective of the initial state in the subsequent MPC iteration. [Fig. 2](#) is a simple illustration of how a new horizon length is determined for three scenarios. The one-step-ahead prediction of the solution to (11) from a sensitivity update is in solid lines and the terminal regions for each scenario are below the dashed lines. Each scenario reaches its respective terminal region at stage N_T^c . To ensure that each scenario reaches its terminal region at t_{k+1} , the new prediction horizon must be at least equal to the largest N_T^c . Therefore in [Fig. 2](#) N_{k+1} must be greater than N_T^3 .

Algorithm 1 presents the horizon update algorithm for multi-stage MPC.

Algorithm 1 Horizon update for multi-stage MPC

- 1: Define $N_{\max}, N_{\min} > N_R$
- 2: Determine \mathbb{X}_f^r, P_r for all $d^r \in \mathbb{D}$
- 3: Initialize: $k \leftarrow 0, N_0 \leftarrow N_{\max}$
- 4: $w^*(x_k) \leftarrow$ Solve (11) with N_k and initial state x_k
- 5: $\mathcal{X}_{k+1}^r \leftarrow \{x_{k+1}^r \mid x_{k+1}^r = f(x_k, u_k, d^r) \forall d^r \in \mathbb{D}\}$
- 6: $\nabla_{x_k} w^* \leftarrow$ Evaluate (14) to obtain NLP sensitivities at x_k
- 7: $\mathcal{W}_{k+1}^* \leftarrow \{\text{get } w^*(x_{k+1}^r) \text{ using (15)} \forall x_{k+1}^r \in \mathcal{X}_{k+1}^r\}$
- 8: **for all** $w^*(x_{k+1}^r) \in \mathcal{W}_{k+1}^*$ **do**
- 9: **if** $z_{N_k|k+1}^c \in \mathbb{X}_f^r \forall c \in \mathbb{C}$ **then**
- 10: $N_T^c \leftarrow$ step at which \mathbb{X}_f^r is reached
- 11: $N_T(x_{k+1}^r) \leftarrow \max_{c \in \mathbb{C}} N_T^c$
- 12: **else**
- 13: $N_T(x_{k+1}^r) \leftarrow N_{\max}$
- 14: **end if**
- 15: **end for**
- 16: $N_T \leftarrow \max_{x_{k+1}^r \in \mathcal{X}_{k+1}^r} N_T(x_{k+1}^r)$
- 17: $N_{k+1} \leftarrow \min(N_{\max}, N_T + N_{\min})$
- 18: $k \leftarrow k + 1$
- 19: Go to Step 4

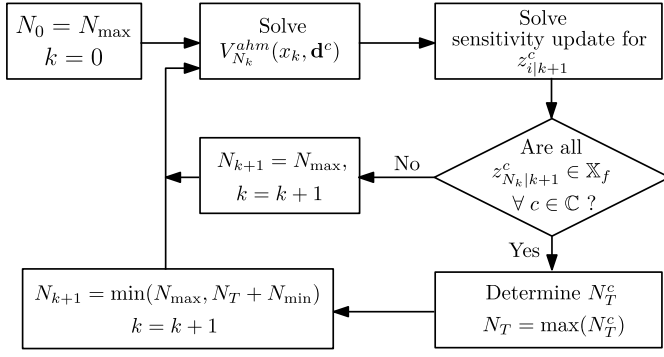


Fig. 3. Horizon update algorithm for multi-stage MPC that gives admissible prediction horizons for subsequent MPC iterations.

The flowchart in Fig. 3 summarizes Algorithm 1 which is a mapping of the current state and prediction horizon to the subsequent prediction horizon. The mapping is defined as a function in Definition 6 as follows:

Definition 6 (Horizon update function). Let \mathcal{N} be the universal set of admissible horizon lengths such that $\mathcal{N} = \{N \mid N_{\min} \leq N \leq N_{\max}, N \in \mathbb{Z}_+\}$. Further, there exists a subset of horizon lengths $\mathcal{N}_k \subset \mathcal{N}$ that define the feasible problems (11) at time t_k . We define a function $H: \mathbb{R}^{n_x} \times \mathcal{N} \times \mathbb{R}^{n_x} \mapsto \mathcal{N}$ that determines the horizon lengths for the adaptive horizon multi-stage algorithm such that $N_{k+1} = H(x_k, N_k, z_{k+1|k}^c) \in \mathcal{N}_{k+1}$.

5. Stability properties of the adaptive horizon multi-stage MPC

Applying the optimal control input computed by the MPC controller to system (1) results in a closed-loop uncertain system whose dynamics are denoted by $x_{k+1} = F(x_k, d_k)$. We will perform a stability analysis for this closed-loop uncertain system. Input-to-state practical stability (ISpS) is the suitable framework to analyze the stability properties of such a system. We start with recollecting some definitions in ISpS stability analysis of discrete-time nonlinear systems:

Definition 7 (Comparison functions [20]). A function $\alpha: \mathbb{R}_+ \mapsto \mathbb{R}_+$ is said to be of class \mathcal{K} if $\alpha(0) = 0$, $\alpha(n) > 0$ for all $n > 0$ and is strictly increasing. It becomes of class \mathcal{K}_∞ if in addition, α is unbounded. A function $\beta: \mathbb{R}_+ \times \mathbb{Z}_+ \mapsto \mathbb{R}_+$ is of class \mathcal{KL} if $\beta(\cdot, k)$ is of class \mathcal{K} for each fixed $k \geq 0$, and $\beta(n, \cdot)$ is decreasing for each fixed $n \geq 0$ and $\beta(n, k) \rightarrow 0$ as $k \rightarrow \infty$.

Definition 8 (Regional Input-to-State practical Stability (ISpS), [21]). An autonomous system $x^+ = F(x, d)$ is ISpS in $\Gamma \subseteq \mathbb{R}^n$ and $0 \in \Gamma$ if there exists a \mathcal{KL} function β , a \mathcal{K} function γ , and a constant $c \geq 0$ such that for all $k \geq 0$

$$\|x_k\| \leq \beta(\|x_0\|, k) + \gamma(\|d_{[0,k-1]} - d^0\|) + c, \quad \forall x_0 \in \Gamma \quad (16)$$

where $d_{[0,k-1]} := [d_0, d_1, \dots, d_{k-1}]$ is the sequence of true parameter realizations and d^0 is a sequence of nominal parameters of the same length.

Definition 9 (ISpS Lyapunov function in Γ , [22]). Suppose Γ is an RPI set and there exists a compact set $\Theta \subseteq \Gamma$ with the origin as an interior point. A function $V: \mathbb{R}^n \mapsto \mathbb{R}_+$ is an ISpS Lyapunov function in Γ for the system $x^+ = F(x, d)$, if there exist \mathcal{K}_∞

functions $\alpha_1, \alpha_2, \alpha_3, \mathcal{K}$ function σ , and constants $c_1, c_2 \geq 0$ such that:

$$V(x) \geq \alpha_1(\|x\|), \quad \forall x \in \Gamma \quad (17a)$$

$$V(x) \leq \alpha_2(\|x\|) + c_1, \quad \forall x \in \Theta \quad (17b)$$

$$V(x^+) - V(x) \leq -\alpha_3(\|x\|) + \sigma(\|d - d^0\|) + c_2, \quad \forall x \in \Gamma, \forall d, d^0 \in \Omega. \quad (17c)$$

where Ω is the uncertain parameter (disturbance) set.

A standard MPC equivalent with an adaptive horizon can be obtained from (11) by setting $d_i^c = d_i^0$ for all $c \in \mathbb{C}$ as follows:

$$V_{N_k}^{ahm}(x_k, d^0) = \min_{z_i^c, v_i^c} \sum_{c \in \mathbb{C}} \omega_c \left(\psi(z_{N_k}^c, d_{N_k-1}^0) + \sum_{i=0}^{N_k-1} \ell(z_i^c, v_i^c, d_i^0) \right) \quad (18a)$$

$$\text{s.t. } z_{i+1}^c = f(z_i^c, v_i^c, d_i^0), \quad i = 0, \dots, N_k - 1 \quad (18b)$$

$$z_0^c = x_k, \quad (18c)$$

$$v_i^c = v_i^{c'}, \quad \{(c, c') \mid z_i^c = z_i^{c'}\} \quad (18d)$$

$$d_{i-1}^0 = d_i^0, \quad i = N_R, \dots, N_k - 1 \quad (18e)$$

$$z_i^c \in \mathbb{X}, \quad v_i^c \in \mathbb{U}, \quad d_i^0 \in \mathbb{D}, \quad (18f)$$

$$z_{N_k}^c \in \mathbb{X}_f, \quad \forall c, c' \in \mathbb{C}, \quad (18g)$$

where (18) effectively consists of c identical copies of the standard MPC with an adaptive horizon.

It follows from Theorem 3 that the difference between the primal solutions (z_i, v_i) of (11) and (18) is $|w^*(d^c) - w^*(d^0)| \leq L_s \|\Delta d\|$, where $w^* = [z^{*\top}, v^{*\top}]^\top$, $L_s > 0$ is the Lipschitz constant, and $\|\Delta d\|$ is defined as:

$$\|\Delta d\| := \max_{d^r, d^{r'} \in \mathbb{D}} \|d^r - d^{r'}\| \quad (19)$$

that is the maximum difference in the disturbance vector between any two parameter realizations. Since the evolution of (1) in closed-loop depends on the true disturbance d_k , then $x_{k+1} = f(x_k, \kappa(x_k), d_k)$, and $x_{k+1} \leq f(x_k, \kappa(x_k), d_k^0) + \mathcal{O}(\|\Delta d\|)$. This result facilitates the recursive feasibility and ISpS stability analysis for the adaptive horizon multi-stage MPC that follows. But first, let us define the following basic assumptions:

Assumption 4 (Basic assumptions for adaptive horizon multi-stage MPC). The adaptive horizon multi-stage MPC has the following properties:

- Lipschitz continuity:** The functions f, ℓ , and ψ are Lipschitz continuous with respect to x, u and d in the compact set $\mathbb{X} \times \mathbb{U} \times \text{convhull}(\mathbb{D})$.
- Constraint set:** The constraint sets \mathbb{X} and \mathbb{U} are compact, and contain the origin in their interiors.
- The solution to (11) satisfies LICQ and SSOSC** such that Theorem 3 applies.

With these properties in Assumption 4, the Lipschitz continuity holds w.r.t. x_k and d^c for the optimal solution and the optimal cost $V_{N_k}^{ahm}$.

Assumption 5 (Property of the stage cost). Given a common control law $\kappa: \mathbb{X} \mapsto \mathbb{U}$ then for any parameter realization $d^r \in \mathbb{D}$,

² The LICQ condition may be relaxed to the Mangasarian–Fromovitz constraint qualification (MFCQ), the constant rank constraint qualification (CRCQ), and the general strong second-order sufficient condition (GSSOSC). In that case, a similar Lipschitz argument can be made and applied in the following analysis [23,24].

the stage cost ℓ is bounded as follows: $\alpha_p(|x|) \leq \ell(x, \kappa(x), d^r) \leq \alpha_q(|x|) + \sigma_q(|d^r - d^0|)$ where $\alpha_p(\cdot), \alpha_q(\cdot) \in \mathcal{K}_\infty$, and $\sigma_q(\cdot) \in \mathcal{K}$.

Assumption 5 requires that the stage cost is positive, with lower and upper bounds proportional to the state at the nominal disturbance. When the disturbance realization is not nominal, the upper bound increases by a term proportional to the distance of the disturbance realization from the nominal.

Assumption 6 (Assumptions on terminal conditions of the adaptive horizon multi-stage MPC). Let **Assumption 4** hold for all parameter realizations $d^r \in \mathbb{D}$ then:

- There exists a common terminal region $\mathbb{X}_f = \bigcap_{r \in \mathcal{Q}} \mathbb{X}_f^r$, that is robust control invariant.
- $\mathbb{X}_f \subseteq \mathbb{X}$ is compact and contains the origin in its interior.
- The terminal cost is bounded as follows: $\alpha_{p,\psi}(|x|) \leq \psi(x, d^r) \leq \alpha_{q,\psi}(|x|) + \sigma_{q,\psi}(|d^r - d^0|)$ where $\alpha_{p,\psi}, \alpha_{q,\psi} \in \mathcal{K}_\infty$, and $\sigma_{q,\psi} \in \mathcal{K}$.
- The terminal cost is a local control Lyapunov function $\psi(f(x, \kappa_f(x), d^r), d^r) - \psi(x, d^r) \leq -\ell(x, \kappa_f(x), d^r)$ for all $x \in \mathbb{X}_f$.

Assumption 6A is needed because a common terminal region must exist for a possible reduction of the prediction horizon. The selection of a prediction horizon where all scenarios reach their respective terminal regions in Step 11 of Algorithm 1 implies that the common terminal region is reached. This region may be a null set when $|\Delta \mathbf{d}|$ is sufficiently large. Further, this can be determined beforehand when computing the terminal ingredients.

5.1. Recursive feasibility of the adaptive horizon multi-stage MPC with a fully branched scenario tree

Before stability analysis, recursive feasibility must be guaranteed by ensuring robust constraint satisfaction. Given the assumptions above, we first present the following result for a fully branched scenario tree i.e. $N_R = N_k$:

Theorem 4 (Recursive feasibility of the adaptive horizon multi-stage MPC). Suppose **Assumptions 4, 5** and **6** hold, then problem (11) from a fully branched scenario tree $N_R = N_k$ is recursively feasible.

Proof. Let **Assumptions 4** and **6** hold and consider the fully branched multi-stage MPC with $N_R = N_k$. Proposition 4 in [25] and Theorem 6 in [26] show that if a robust control invariant terminal region exists then the fully branched multi-stage MPC with a fixed horizon $N_R = N_k$ is recursively feasible. Since the adaptive horizon, multi-stage MPC with $N_R = N_k$ ensures that the horizon is updated such that the common terminal region is always reached, it is also recursively feasible for any horizon update N_{k+1} given by Algorithm 1. \square

5.2. Recursive feasibility of the adaptive horizon multi-stage MPC with a robust horizon

Recursive feasibility cannot be guaranteed by using the same arguments in **Theorem 4** when the scenario tree is not fully branched. But the infeasibility of problem (11) when $N_R < N_k$ may be avoided by relaxing the state inequalities using soft constraints [27]. However, robust constraint satisfaction is not guaranteed because the state trajectories may cross the feasible set \mathbb{X} . Therefore, before proceeding to the stability analysis the following assumption must be made:

Assumption 7. Problem (11) is feasible at time t_k when $N_k = N_{\max}$. Furthermore, if problem (11) at time t_k with x_k and N_k is feasible, then so is problem (11) solved at time t_{k+1} with x_{k+1} and $N_{k+1} = H(x_k, N_k, x_{k+1}) \in \mathcal{N}_{k+1}$ for all $x_k, x_{k+1} \in \mathbb{X}$, $N_k \in \mathcal{N}_{k+1}$.

Note that **Assumption 7** can always be satisfied by choosing $N_R = N_k$. (cf. **Theorem 4**). The increase in the problem size with a long robust horizon required to satisfy **Assumption 7** also motivates the need for shortening the horizon length using our proposed Algorithm 1. **Assumption 7** ensures that H (see **Definition 6**) produces feasible horizon lengths for any robust horizon $N_R \leq N_k$, essentially implying recursive feasibility. This allows us to present results on the ISpS property of the adaptive horizon multi-stage MPC.

5.3. ISpS for adaptive horizon multi-stage MPC

First, the relationships between the linear and nonlinear systems are needed because the nonlinear system (1) is controlled by a stabilizing linear controller inside the common terminal region \mathbb{X}_f . Consider the nominal problem (18) inside the terminal region evaluated using a stabilizing terminal control law κ_f^0 :

$$V_N^{\text{lqr}}(x_k, \mathbf{d}^0) := \sum_{c \in \mathcal{C}} \omega_c \left(\psi(z_N^0, d_{N-1}^0) + \sum_{i=0}^{N-1} \ell(z_i^0, \kappa_f^0(z_i^0), d_i^0) \right) \quad (20)$$

where $z_{i+1}^0 = A_0 z_i^0 + B_0 v_i^0 + \psi(z_i^0, d^0)$, $\forall i = 0, \dots, N-1$, and $V_N^{\text{lqr}}(x_k, \mathbf{d}^0)$ is the nominal LQR cost. The relationships between the optimal costs of the nominal equivalents of the LQR and adaptive horizon multi-stage MPC in the terminal region are given in **Lemma 5**.

Lemma 5 (Effect of linearization on costs inside the terminal region). There exists $\alpha_n \in \mathcal{K}_\infty$ such that $|V_N^{\text{lqr}}(x, \mathbf{d}^0) - \psi(x, d^0)| \leq \alpha_n(|x|)$ and $\alpha_v \in \mathcal{K}_\infty$ such that $|\psi(x, d^0) - V_N^{\text{ahm}}(x, \mathbf{d}^0)| \leq \alpha_v(|x|)$ for all $x \in \mathbb{X}_f$ and for all $N \in \mathcal{N}$.

Proof. Because this is a nominal problem, the result holds because of Lemma 18 and 19 in [9]. \square

Assumption 8. The solution to (18) with a prediction horizon $N_k \geq N_{\min}$ satisfies:

$$\alpha_n(|z_{N_k|k}^0|) - \alpha_p(|x_k|) \leq -\alpha_3(|x_k|) \text{ if } N_{k+1} \geq N_k \quad (21a)$$

$$\alpha_v(|z_{N_{k+1}+1|k}^0|) - \alpha_p(|x_k|) \leq -\alpha_3(|x_k|) \text{ if } N_{k+1} < N_k \quad (21b)$$

for some $\alpha_3 \in \mathcal{K}_\infty$, where $N_{k+1} = H(x_k, N_k, z_{N_{k+1}+1|k}^0) \in \mathcal{N}_{k+1}$, α_p satisfy **Assumption 5**, and α_n, α_v satisfy **Lemma 5**.

This assumption implies that the approximation error from using the stabilizing linear control law inside the terminal region must be negligible. **Assumption 8** can be satisfied by finding a suitably large N_{\min} that is determined through simulations as explained in [9].

Theorem 6 (ISpS stability of adaptive horizon multi-stage MPC). Suppose **Assumptions 5** and **6** hold for (11), the optimal value function V^{ahm} is an ISpS Lyapunov function and the resulting closed-loop system is ISpS stable.

Proof. To show the ISpS property of the adaptive horizon multi-stage MPC, a Lyapunov function must exist that satisfies the three conditions in (17). *Lower bound:* Suppose **Assumption 5** holds then there exists a lower bound (17a) on the optimal value function.

$$V_N^{\text{ahm}}(x_k, \mathbf{d}^c) \geq \ell(x_k, \kappa_N(x_k), d_0^0) \geq \alpha_p(|x_k|) \quad \forall x_k \in \mathbb{X} \quad (22)$$

Upper bound: The existence of an upper bound (17b) is shown by using Assumption 4A, the boundedness property of the optimal cost of standard MPC recursively forward [1], and Assumption 6C as follows:

$$V_N^{\text{ahm}}(x_k, \mathbf{d}^c) \leq V_N^{\text{ahm}}(x_k, \mathbf{d}^0) + L_v |\Delta \mathbf{d}| \quad (23a)$$

$$\leq \psi(x_k, \mathbf{d}^0) + L_v |\Delta \mathbf{d}| \quad (23b)$$

$$\leq \alpha_{q,\psi}(|x_k|) + c_1 \quad (23c)$$

for all $x_k \in \mathbb{X}_f$, where $c_1 = L_v |\Delta \mathbf{d}|$ is a constant. It also applies to all $x_k \in \mathbb{X}$ by induction because $\mathbb{X}_f \subseteq \mathbb{X}$, and $V_N^{\text{ahm}}(x_k, \mathbf{d}^0)$ is continuous at the origin.

Note that the adaptive horizon multi-stage MPC control law may be discontinuous because it optimizes with different prediction horizons provided by Algorithm 1 at each time step. However, our analysis considers the properties of the optimal cost function V_N^{ahm} which may still be continuous even with a discontinuous control law, given that the input constraint set is continuous. [1]

Function descent: The descent property (17), which is the main part of stability analysis, is shown for two possible cases of horizon update:

1. A decreasing prediction horizon $N_R \leq N_{k+1} < N_k$: An approximate solution for (18) at time t_{k+1} is found by shifting the optimal solution at time t_k as follows:

$$\hat{v}_{i|k+1}^0 = v_{i+1|k}^0, \quad i = 0, \dots, N_{k+1} - 1, \quad (24)$$

and the corresponding initialization for state variables:

$$\hat{z}_{0|k+1}^0 = z_{1|k}^0, \quad (25a)$$

$$\hat{z}_{i+1|k+1}^0 = f(\hat{z}_{i|k}^0, \hat{v}_{i|k}^0, \mathbf{d}_i^0) \quad (25b)$$

leads to a suboptimal cost such that the descent inequality becomes:

$$\begin{aligned} & V_{N_{k+1}}^{\text{ahm}}(f(x_k, \kappa(x_k), \mathbf{d}_0^0), \mathbf{d}^0) - V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^0) \\ & \leq \sum_{c \in \mathbb{C}} \omega_c \left(\psi(\hat{z}_{N_{k+1}|k+1}^0, \mathbf{d}_{N_{k+1}-1}^0) + \sum_{i=0}^{N_{k+1}-1} \ell(\hat{z}_{i|k+1}^0, \hat{v}_{i|k+1}^0, \mathbf{d}_i^0) \right) \\ & - \sum_{c \in \mathbb{C}} \omega_c \left(\psi(v_{N_k|k}^0, \mathbf{d}_{N_k-1}^0) - \sum_{i=0}^{N_k-1} \ell(z_{i|k}^0, v_{i|k}^0, \mathbf{d}_i^0) \right) \end{aligned} \quad (26a)$$

$$\begin{aligned} & = -\ell(x_k, \kappa(x_k), \mathbf{d}_0^0) + \psi(\hat{z}_{N_{k+1}|k+1}^0, \mathbf{d}_{N_{k+1}-1}^0) \\ & + \sum_{i=0}^{N_{k+1}-1} (\ell(\hat{z}_{i|k+1}^0, \hat{v}_{i|k+1}^0, \mathbf{d}_i^0) - \ell(z_{i+1|k}^0, v_{i+1|k}^0, \mathbf{d}_i^0)) \end{aligned} \quad (26b)$$

$$\begin{aligned} & - \psi(z_{N_k|k}^0, \mathbf{d}_{N_k-1}^0) - \sum_{i=N_{k+1}+1}^{N_k-1} \ell(\hat{z}_{i|k}^0, \hat{v}_{i|k}^0, \mathbf{d}_i^0) \\ & = -\ell(x_k, \kappa(x_k), \mathbf{d}_0^0) + \psi(\hat{z}_{N_{k+1}|k+1}^0, \mathbf{d}_{N_{k+1}-1}^0) \\ & - \psi(z_{N_k|k}^0, \mathbf{d}_{N_k-1}^0) - \sum_{i=N_{k+1}+1}^{N_k-1} \ell(\hat{z}_{i|k}^0, \hat{v}_{i|k}^0, \mathbf{d}_i^0) \end{aligned} \quad (26c)$$

$$\begin{aligned} & = -\ell(x_k, \kappa(x_k), \mathbf{d}_0^0) \\ & + \psi(\hat{z}_{N_{k+1}|k+1}^0, \mathbf{d}_{N_{k+1}-1}^0) - V_{N_k-N_{k+1}+1}^{\text{ahm}}(z_{N_{k+1}+1|k}^0, \mathbf{d}^0) \end{aligned} \quad (26d)$$

In the presence of true uncertainty, the successor state is $x_{k+1} = f(x_k, \kappa(x_k), d_k)$ where d_k is the true parameter realization, such that:

$$\begin{aligned} & V_{N_{k+1}}^{\text{ahm}}(x_{k+1}, \mathbf{d}^0) - V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^0) \leq -\ell(x_k, \kappa(x_k), \mathbf{d}_0^0) \\ & + \psi(\hat{z}_{N_{k+1}|k+1}^0, \mathbf{d}_{N_{k+1}-1}^0) - V_{N_k-N_{k+1}+1}^{\text{ahm}}(z_{N_{k+1}+1|k}^0, \mathbf{d}^0) \\ & + L_k |d_k - d_i^0| \end{aligned} \quad (26e)$$

Then from Eqs. (11) and (18) we have:

$$V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^c) = V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^0) + \mathcal{O}(|\Delta \mathbf{d}|), \text{ and} \quad (27a)$$

$$V_{N_{k+1}}^{\text{ahm}}(x_{k+1}, \mathbf{d}^c) = V_{N_{k+1}}^{\text{ahm}}(x_{k+1}, \mathbf{d}^0) + \mathcal{O}(|\Delta \mathbf{d}|) \quad (27b)$$

that can be combined with (26e) to form:

$$\begin{aligned} & V_{N_{k+1}}^{\text{ahm}}(x_{k+1}, \mathbf{d}^c) - V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^c) \leq -\ell(x_k, \kappa(x_k), \mathbf{d}_0^0) \\ & + \psi(\hat{z}_{N_{k+1}|k+1}^0, \mathbf{d}_{N_{k+1}-1}^0) - V_{N_k-N_{k+1}+1}^{\text{ahm}}(z_{N_{k+1}+1|k}^0, \mathbf{d}^0) \\ & + L_k |d_k - d_i^0| + (V_{N_{k+1}}^{\text{ahm}}(x_{k+1}, \mathbf{d}^c) - V_{N_{k+1}}^{\text{ahm}}(x_{k+1}, \mathbf{d}^0)) \\ & + (V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^c) - V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^0)) \end{aligned} \quad (28a)$$

$$\leq -\ell(x_k, \kappa(x_k), \mathbf{d}_0^0) + \psi(\hat{z}_{N_{k+1}|k+1}^0, \mathbf{d}_{N_{k+1}-1}^0) - V_{N_k-N_{k+1}+1}^{\text{ahm}}(z_{N_{k+1}+1|k}^0, \mathbf{d}^0) + L_k |d_k - d_i^0| + 2L_v |\Delta \mathbf{d}| \quad (28b)$$

$$\leq -\alpha_p(|x_k|) + \alpha_v(|z_{N_{k+1}+1|k}^0|) + L_k |d_k - d_i^0| + 2L_v |\Delta \mathbf{d}| \quad (28c)$$

$$\leq -\alpha_3(|x_k|) + \sigma(|d_k - d_i^0|) + c_2 \quad (28d)$$

where $c_2 = 2L_v |\Delta \mathbf{d}| \geq 0$ and (28c) follows from Lemma 5, Eq. (28d) follows from (21b).

2. An increasing prediction horizon $N_{k+1} \geq N_k$: Similar to the first case approximate the solution of (18) at t_{k+1} by shifting:

$$\hat{v}_{i|k+1}^0 = \begin{cases} v_{i+1|k}^0 & \forall i = 0, \dots, N_k - 2 \\ -K_0 z_i^0 & \forall i = N_k - 1, \dots, N_{k+1} - 1 \end{cases} \quad (29)$$

The initialization of the states is the same as in (25). Then the descent inequality becomes:

$$V_{N_{k+1}}^{\text{ahm}}(f(x_k, \kappa(x_k), \mathbf{d}_0^0), \mathbf{d}^0) - V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^0) \quad (30a)$$

$$\begin{aligned} & \leq \sum_{c \in \mathbb{C}} \omega_c \left(\psi(\hat{z}_{N_{k+1}|k+1}^0, \mathbf{d}_{N_{k+1}-1}^0) + \sum_{i=0}^{N_{k+1}-1} \ell(\hat{z}_{i|k+1}^0, \hat{v}_{i|k+1}^0, \mathbf{d}_i^0) \right) \\ & - \sum_{c \in \mathbb{C}} \omega_c \left(\psi(z_{N_k|k}^0, \mathbf{d}_{N_k-1}^0) - \sum_{i=0}^{N_k-1} \ell(z_{i|k}^0, v_{i|k}^0, \mathbf{d}_i^0) \right) \end{aligned} \quad (30b)$$

$$\begin{aligned} & = -\ell(x_k, \kappa(x_k), \mathbf{d}_0^0) \\ & + \sum_{i=0}^{N_k-2} (\ell(\hat{z}_{i|k+1}^0, \hat{v}_{i|k+1}^0, \mathbf{d}_i^0) - \ell(z_{i+1|k}^0, v_{i+1|k}^0, \mathbf{d}_i^0)) \\ & + \psi(\hat{z}_{N_{k+1}|k+1}^0, \mathbf{d}_{N_{k+1}-1}^0) + \sum_{i=N_k-1}^{N_{k+1}-1} \ell(\hat{z}_{i|k+1}^0, \hat{v}_{i|k+1}^0, \mathbf{d}_i^0) \\ & - \psi(z_{N_k|k}^0, \mathbf{d}_{N_k-1}^0) \end{aligned} \quad (30c)$$

$$\begin{aligned} & = -\ell(x_k, \kappa(x_k), \mathbf{d}_0^0) + \psi(\hat{z}_{N_{k+1}|k+1}^0, \mathbf{d}_{N_{k+1}-1}^0) \\ & + \sum_{i=N_k-1}^{N_{k+1}-1} (\ell(\hat{z}_{i|k+1}^0, \hat{v}_{i|k+1}^0, \mathbf{d}_i^0) - \psi(z_{N_k|k}^0, \mathbf{d}_{N_k-1}^0)) \end{aligned} \quad (30d)$$

$$\begin{aligned} & = -\ell(x_k, \kappa(x_k), \mathbf{d}_0^0) + V_{N_k-N_{k+1}+1}^{\text{ahm}}(z_{N_k|k}^0, \mathbf{d}^0) \\ & - \psi(z_{N_k|k}^0, \mathbf{d}_{N_k-1}^0) \end{aligned} \quad (30e)$$

In the presence of true parameter realization d_k then:

$$\begin{aligned} & V_{N_{k+1}}^{\text{ahm}}(x_{k+1}, \mathbf{d}^0) - V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^0) \leq -\ell(x_k, \kappa(x_k), \mathbf{d}_0^0) \\ & + V_{N_{k+1}-N_k+1}^{\text{ahm}}(z_{N_k|k}^0, \mathbf{d}^0) - \psi(z_{N_k|k}^0, \mathbf{d}_{N_k-1}^0) \\ & + L_k |d_k - d_i^0| \end{aligned} \quad (30f)$$

Substituting (27a) and (27b) on (30f) we have:

$$\begin{aligned} & V_{N_{k+1}}^{\text{ahm}}(x_{k+1}, \mathbf{d}^c) - V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^c) \leq -\ell(x_k, \kappa(x_k), \mathbf{d}_0^0) \\ & + V_{N_{k+1}-N_k+1}^{\text{ahm}}(z_{N_k|k}^0, \mathbf{d}^0) - \psi(z_{N_k|k}^0, \mathbf{d}_{N_k-1}^0) \end{aligned} \quad (31a)$$

$$\begin{aligned} & + L_k |d_k - d_i^0| + 2L_v |\Delta \mathbf{d}| \\ & \leq -\ell(x_k, \kappa(x_k), \mathbf{d}_0^0) + V_{N_{k+1}-N_k+1}^{\text{ahm}}(z_{N_k|k}^0, \mathbf{d}^0) \\ & - \psi(z_{N_k|k}^0, \mathbf{d}_{N_k-1}^0) + L_k |d_k - d_i^0| + 2L_v |\Delta \mathbf{d}| \end{aligned} \quad (31b)$$

$$\leq -\alpha_p(|x_k|) + \alpha_n(|z_{N_k,k}^0|) + L_k|d_k - d_i^0| + 2L_v|\Delta \mathbf{d}| \quad (31c)$$

$$\leq -\alpha_3(|x_k|) + \sigma(|d_k - d_i^0|) + c_2 \quad (31d)$$

where $c_2 = 2L_v|\Delta \mathbf{d}| \geq 0$, and (31c) follows from Lemma 5, Eq. (31d) follows from (21a).

Thus $V_{N_k}^{\text{ahm}}$ satisfies the descent property (17) for all the possible horizon length updates. Hence the adaptive horizon multi-stage MPC is ISpS stable. \square

6. Numerical examples

6.1. Example 1 - Spring damped mass system

This is a simple example adapted from [28] used to demonstrate the performance of an adaptive horizon multi-stage MPC with a fully branched scenario tree, and therefore suitable terminal conditions that guarantee recursive feasibility and ISpS stability (see Section 5.1). The system model is written as:

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -\frac{k_0}{m}e^{-x_1}x_1 - \frac{h_d}{m}x_2 + \frac{u}{m}$$

where the state variable vector $x = [x_1, x_2]^T$ includes the displacement and the velocity of the mass $m = 1$ kg, respectively. $k = k_0e^{-x_1}$ is the elastic constant of the spring and $k_0 = 0.33$ N/m. The damping factor h_d is uncertain and can have three different possible values $h_d^r = \{1.0, 2.0, 4.0\}$ Ns/m with equal probabilities. The damping factor is assumed to vary unpredictably between sampling intervals of $\Delta t = 0.4$ s.

The objective is to control the mass to its equilibrium position $x_1 = 0$ using an external force u . The stage cost is given by $\ell = x^T Q x + u^T R u$ and h_d varies randomly at each time step with the previously defined probabilities.

6.1.1. Offline approximation of terminal ingredients

The LQR method explained in Section 4.2 is applied to find suitable ψ and \mathbb{X}_f . The tuning matrices used are $Q = \text{diag}([30, 20])$ and $R = 1$. The terminal control law must be common for all scenarios due to full branching (see Section 4.2.2). Therefore, $K = [1.7409 \ 2.0959]$ is chosen with $\kappa_f(x) = -Kx$ for all $x \in \mathbb{X}_f$. Different values of P_r are computed using the Lyapunov equation to obtain the respective terminal costs for each realization. The terminal radii are determined by performing 10^5 one-step simulations of the closed-loop system and estimating the linearization error bound. The linearization error of this system is independent of h_d , so the simulations were performed for only one of the disturbance realizations. Fig. 4 is a plot of the linearization error against $|\Delta x|$, and the upper bound on the linearization error is fitted by the solid red line. After obtaining bound parameters $M_r = 0.3235$ and $q_r = 2.2176$, the terminal radii were found to be $c_f^r \approx \{0.8032, 0.8922, 0.6690\}$.

6.1.2. Simulation results

The simulations are done using JuMP v.0.21.10 [29] as the NLP modeler in a Julia [30] environment. The NLP solver used is IPOPT 3.13.4 [31], and the linear solver is HSL-MA57 [32] on a 2.6 GHz Intel Core-i7 with 16 GB memory.

An initial prediction horizon of $N_0 = 8$ is chosen with $N_{\min} = 2$. The scenario tree is fully branched at each MPC iteration making a total of 6561 scenarios at the first iteration. Fig. 5 shows two sets of closed-loop simulations of the adaptive horizon multi-stage MPC that are done with initial conditions $x_0 = [-4, 4]^T$ m (blue line) and $[5.3, 2]^T$ m (black line). The state bounds are $x_1 \in [-10, 10]$ m and $x_2 \in [-4, 10]$ m/s and the control input bounds are $u \leq |6.0 \text{ N}|$. The system is controlled to its setpoint and the horizon length N_k is reduced as the setpoint is approached. Eventually, the N_k is reduced to 3 from 8 cutting the

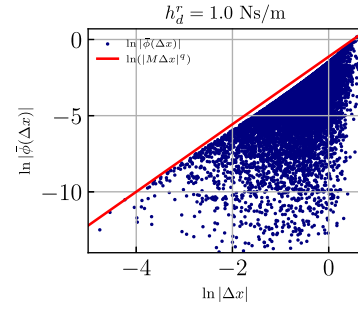


Fig. 4. Spring-damper-mass – plots of linearization error against $|\Delta x|$ for 10000 simulations showing the bound (red line).

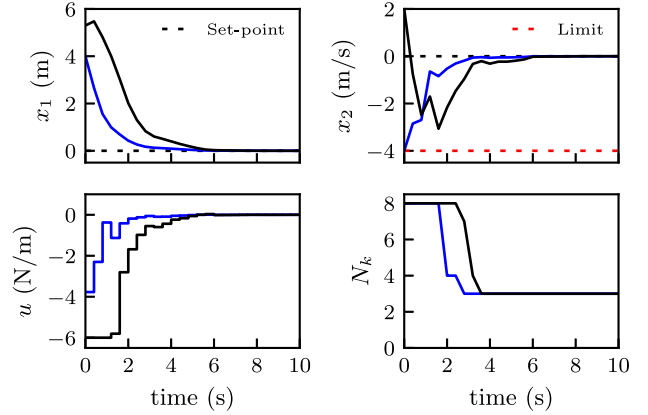


Fig. 5. Spring-damper-mass – simulation results from two initial conditions $x_0 = [-4, 4]$ (blue lines) and $x_0 = [5.3, 2]^T$ (black lines) showing the control performance of a fully branched adaptive horizon multi-stage MPC.

number of scenarios from 6561 to 27. Therefore, this example also demonstrates the advantage of the adaptive horizon method for computational cost savings in multi-stage MPC with a fully branched scenario tree.

6.2. Example 2 – Cooled CSTR

The rest of the numerical examples consider a robust horizon (i.e. no full branching), therefore recursive feasibility is not necessarily guaranteed (see Section 5.2). However, the aim is to compare the performances of the adaptive horizon multi-stage MPC (varying horizon lengths) with fixed horizon multi-stage MPC and to demonstrate the computational cost savings as a result of the adaptive horizon algorithm.

Consider the control of a CSTR with a cooling jacket example from [33]. The dynamics of the cooled CSTR are given by,

$$\begin{aligned} \dot{c}_A &= F(c_{A,0} - c_A) - k_1 c_A - k_3 c_A^2 \\ \dot{c}_B &= -F c_B + k_1 c_A - k_2 c_B \\ \dot{T}_R &= F(T_{in} - T_R) + \frac{k_W A_R}{\rho c_p V_R} (T_j - T_R) \\ &\quad - \frac{k_1 c_A \Delta H_{AB} + k_2 c_B \Delta H_{BC} + k_3 c_A^2 \Delta H_{AD}}{\rho c_p} \\ \dot{T}_j &= \frac{1}{m_j c_{p,j}} (\dot{Q}_j + k_W A_R (T_R - T_j)) \end{aligned}$$

where reaction rates k_i follow the Arrhenius law, $k_i = A_i \exp\left(\frac{-E_i}{RT_R}\right)$.

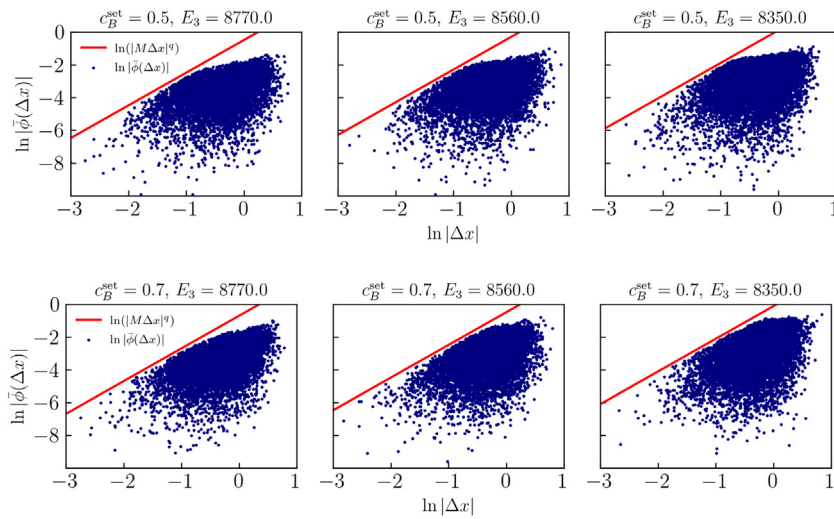


Fig. 6. CSTR – plots of linearization error against $|\Delta x|$ for 10000 simulations for each parametric realization of E_3 , at each setpoint $c_B^{\text{set}} = 0.5 \text{ mol}/\ell$ and $0.7 \text{ mol}/\ell$ showing their bounds (red lines).

Table 1
CSTR – Bounds on states and inputs.

Variable	Initial condition	Minimum	Maximum	Unit
c_A	0.8	0.1	5.0	mol/l
c_B	0.5	0.1	5.9	mol/l
T_R	134.14	50	140	°C
T_J	134.0	50	180	°C
F	18.83	0.0	35	/h
\dot{Q}_J	-4495.7	-8500	0	kJ/h

The state variable vector $x = [c_A, c_B, T_R, T_J]^T$ consists of the concentrations of A and B, reactor and coolant temperatures, respectively. The control inputs u , are inlet flow per reactor volume $F = V_{\text{in}}/V_R$, and cooling rate \dot{Q}_J .

The control objective is to regulate c_B at a desired setpoint. The system operates at two setpoints: $c_B^{\text{set}} = 0.5 \text{ mol}/\ell$, and $c_B^{\text{set}} = 0.7 \text{ mol}/\ell$. The activation energy $E_3 = 8560 \pm 2.5\% \text{ K}$ is the only uncertain parameter in the system. Table 1 summarizes the system bounds and its initial conditions. The stage cost is a setpoint tracking squared error of c_B plus control movement penalization terms $\Delta F_i = F_i - F_{i-1}$ and $\Delta \dot{Q}_k = \dot{Q}_k - \dot{Q}_{k-1}$ given by $\ell_k = (c_{B_i} - c_B^{\text{set}})^2 + r_{\Delta 1} \Delta F_i^2 + r_{\Delta 2} \Delta \dot{Q}_k^2$, where the control penalties are $r_{\Delta 1} = 10^{-5}$ and $r_{\Delta 2} = 10^{-7}$. Regularization terms are added to impose implicit references on the remaining states and strong convexity [24].

The initial prediction horizon is $N_0 = 40$. Terminal constraints are included for all the scenarios. The simulations are performed for $N_R = 1, 2$. We sample three parameter realizations, such that when $N_R = 2$ there are 9 scenarios as in Fig. 1.

6.2.1. Offline approximation of terminal ingredients

To design a stabilizing LQR for each parametric realization, we use $Q = I_4$, and $R = \text{diag}([10^{-3} \ 10^{-4}])$. As outlined before, 10^5 one-step simulations are performed offline from random initial states to compute linearization error for each parametric realization and each operating point. Fig. 6 shows the plots of the linearization error against $|\Delta x|$ for each parametric realization and each setpoint. All the plots show an upper bound on the linearization error (solid red lines in Fig. 6) from which M_r and q_r values are fitted. The terminal region radii for each parametric realization are obtained by evaluating (10) as shown in Table 2.

Table 2
CSTR – linearization error bounds and terminal radii.

E_3 (K)	$c_B^{\text{set}} = 0.5 \text{ mol}/\ell$			$c_B^{\text{set}} = 0.7 \text{ mol}/\ell$		
	M_r	q_r	c_f^r	M_r	q_r	c_f^r
8774	0.62	2.0	0.1429	0.50	2.0	0.1718
8560	0.75	2.0	0.1159	0.62	2.0	0.1337
8346	1.12	2.0	0.0747	0.90	2.0	0.0846

6.2.2. Simulation results

A sample time of 18 s is used, and the process is simulated for 0.6 h. At the beginning $c_B^{\text{set}} = 0.5 \text{ mol}/\ell$ and it changes to $c_B^{\text{set}} = 0.7 \text{ mol}/\ell$ at $t_k = 0.3 \text{ h}$. The uncertain parameter E_3 is a random sequence of the sampled realizations. The control performance of the adaptive horizon multi-stage MPC is compared with the fixed horizon multi-stage MPC in this system. The value of $N_{\min} = 5$ in the adaptive horizon multi-stage algorithm. Figs. 7 and 8 show closed-loop simulation results when $N_R = 1, 2$, respectively. The optimal input sequences and state trajectories for both controllers are nearly identical. The two controllers show similar performance in tracking changes in setpoint c_B^{set} . This implies that the adaptive horizon algorithm does not affect the tracking performance of the multi-stage MPC.

The prediction horizon and total computation times per iteration for the two controllers when $N_R = 1, 2$ are plotted in Figs. 9 and 10, respectively. The adaptive horizon multi-stage MPC reduces the prediction horizon significantly, thus saving the computation time needed in each multi-stage MPC iteration. There is an 86% savings in CPU time on average from fixed horizon multi-stage MPC with $N_R = 1$ by using the adaptive horizon multi-stage MPC. When $N_R = 2$, there is also a 92% savings in CPU time on average. The regulatory performances of the two controllers are nearly identical but the prediction horizon and computation time are reduced significantly for the adaptive horizon multi-stage MPC.

6.3. Example 3 – Quad-tank system

The second example is from [34], on the control of a quad-tank system with four tanks. The liquid levels in the four tanks

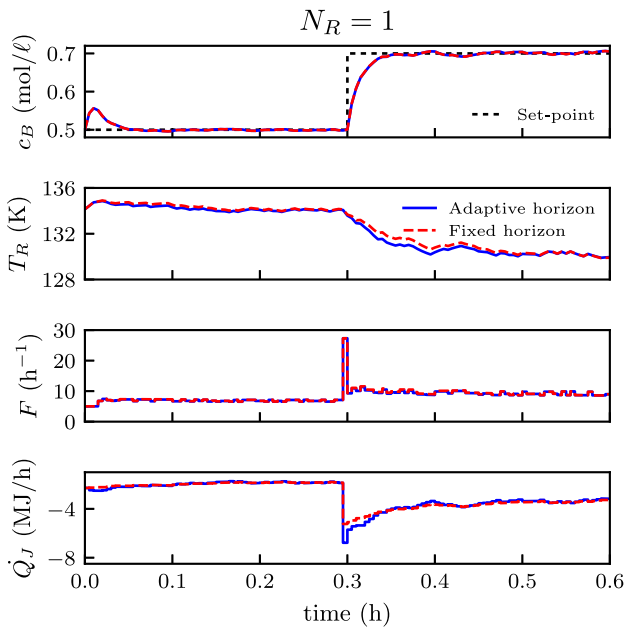


Fig. 7. CSTR – Simulation results comparing the adaptive horizon multi-stage MPC with the fixed horizon multi-stage MPC when robust horizon $N_R = 1$.

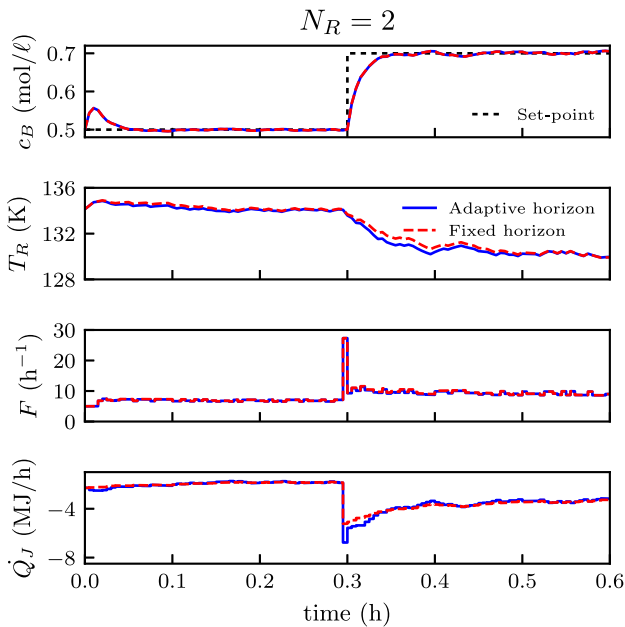


Fig. 8. CSTR – Simulation results comparing the adaptive horizon multi-stage MPC with the fixed horizon multi-stage MPC when robust horizon $N_R = 2$.

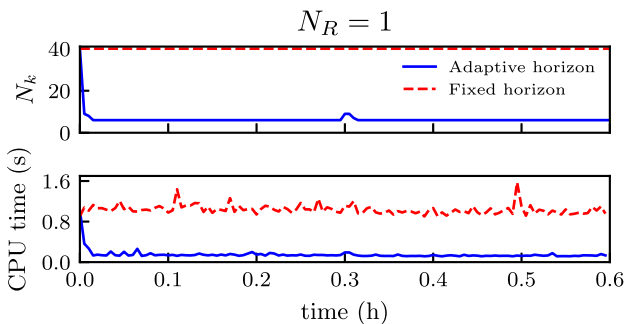


Fig. 9. CSTR – Simulation results comparing prediction horizons and computation times per iteration when robust horizon $N_R = 1$.

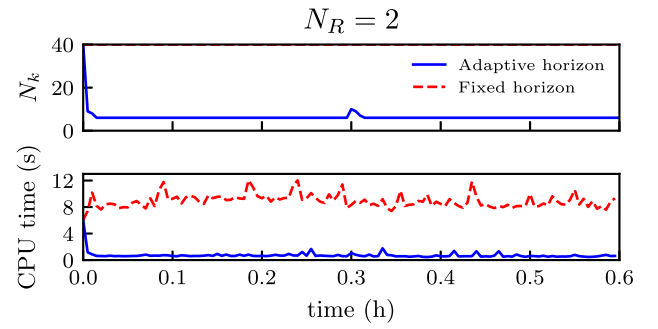


Fig. 10. CSTR – Simulation results comparing prediction horizons and computation times per iteration when robust horizon $N_R = 2$.

Table 3

Quad-tank – Bounds on states and inputs.

Variable	Minimum	Maximum	Unit
x_1	7.5	28.0	cm
x_2	7.5	28.0	cm
x_3	14.2	28.0	cm
x_4	4.5	21.3	cm
u_1	0.0	60.0	ml/s
u_2	0.0	60.0	ml/s

Table 4

Pulse changes to state variables applied to the quad-tank system.

k	x_1	x_2	x_3	x_4
0	28 cm	28 cm	14.2 cm	21.3 cm
50	28 cm	14 cm	28 cm	21.3 cm
100	28 cm	14 cm	14.2 cm	21.3 cm

are described by the following set of differential equations:

$$\begin{aligned}\dot{x}_1 &= -\frac{a_1}{A_1}\sqrt{2gx_1} + \frac{a_3}{A_1}\sqrt{2gx_3} + \frac{\gamma_1}{A_1}u_1 \\ \dot{x}_2 &= -\frac{a_2}{A_2}\sqrt{2gx_2} + \frac{a_4}{A_2}\sqrt{2gx_4} + \frac{\gamma_2}{A_2}u_2 \\ \dot{x}_3 &= -\frac{a_3}{A_3}\sqrt{2gx_3} + \frac{1-\gamma_2}{A_3}u_2 \\ \dot{x}_4 &= -\frac{a_4}{A_4}\sqrt{2gx_4} + \frac{1-\gamma_1}{A_4}u_1\end{aligned}$$

where x_i represents the liquid level in tank i , u_i is the flow rate of pump i . The system has four states: $x = [x_1, x_2, x_3, x_4]^T$ and two control inputs: $u = [u_1, u_2]^T$. A_i and a_i are the cross sectional areas of the tank i and its outlet, respectively. The valve coefficients γ_1 and γ_2 are the uncertain parameters of this system. The values of $\gamma_1, \gamma_2 = 0.4 \pm 0.05$ and the system bounds are shown in Table 3.

The controller has a reference tracking objective, regulating the two levels in the lower tanks (tanks 1 and 2). The setpoints are $x_1^{\text{set}} = x_2^{\text{set}} = 14$ cm. We introduce predefined pulse changes in the state values at specific iterations to reset reference tracking as shown in Table 4.

The closed-loop multi-stage MPC simulations are run for 150 iterations with a sample time of 10 s. The stage cost function is given by: $\ell_k = (x_{1i} - x_1^{\text{set}})^2 + (x_{2i} - x_2^{\text{set}})^2 + r_\Delta(\Delta u_{1i}^2 + \Delta u_{2i}^2)$, where $\Delta u_{1i} = u_{1i} - u_{1i-1}$ and $\Delta u_{2i} = u_{2i} - u_{2i-1}$ are the control movement terms for the pump flow-rates that are penalized in the objective function with the penalty parameter $r_\Delta = 0.01$.

6.3.1. Offline approximation of terminal ingredients

LQR controller tuning $Q = 1.5I_4$ and $R = I_2$ are selected for the stabilizing controller for each parameter realization of γ_1 and γ_2 . As done previously, 10^5 one-step offline simulations

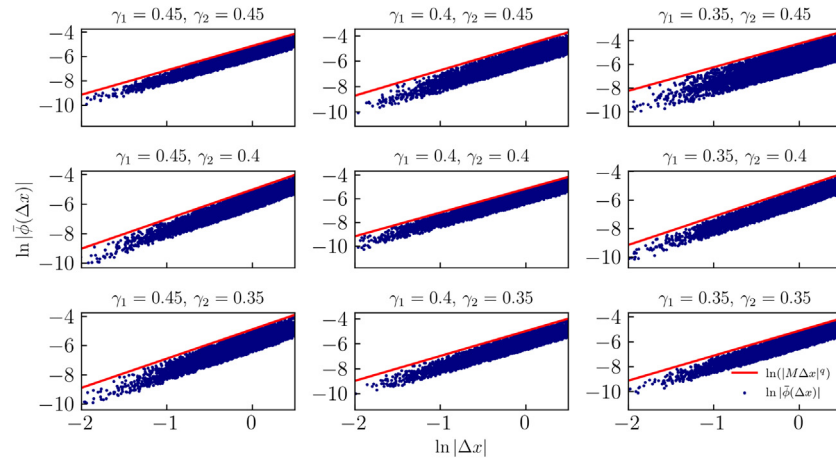


Fig. 11. Quad-tank – matrix of plots of linearization error against $|\Delta x|$ of 10,000 simulations for every uncertain parameter realization.

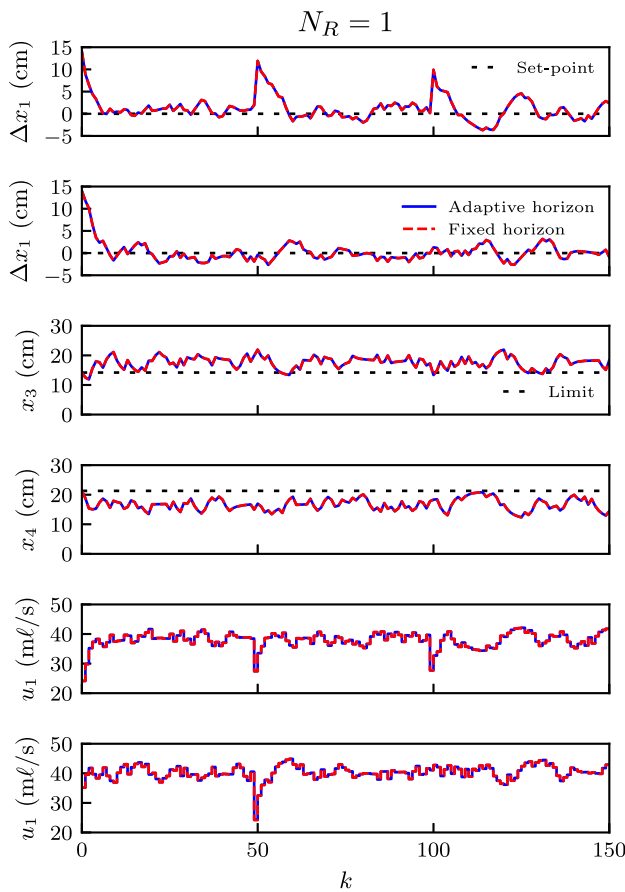


Fig. 12. Quad-tank – Simulation results comparing fixed horizon multi-stage and adaptive horizon multi-stage MPC with $N_R = 1$.

from randomly sampled initial state values are performed for each parameter realization to compute the linearization error. The set \mathbb{D} is obtained via grid-based sampling including the nominal and extreme parameter values only. A total of 9 parameter realizations are considered for the two uncertain parameters.

Fig. 11 is a matrix of linearization error plots from the one-step simulations with estimated upper bounds (solid red lines) for each uncertain parameter realization. The terminal radii for each

Table 5

Quad-tank – linearization error bounds and terminal radii.

$[\gamma_1; \gamma_2]$	M_r	q_r	c_f^r
[0.45; 0.45]	0.0058	2.0	30.43
[0.40; 0.45]	0.0088	2.0	19.99
[0.35; 0.45]	0.0142	2.0	12.12
[0.45; 0.40]	0.0065	2.0	27.04
[0.40; 0.40]	0.0056	2.0	31.77
[0.35; 0.40]	0.0057	2.0	31.57
[0.45; 0.35]	0.0074	2.0	23.69
[0.40; 0.35]	0.0068	2.0	26.63
[0.35; 0.35]	0.0059	2.0	30.97

parameter realization are computed using (10). The estimated linearization error bounds and terminal radii for each parameter realization are shown in Table 5.

6.3.2. Simulation results

The parameters γ_1 and γ_2 have an uncertainty range of ± 0.05 about their nominal value. Similar to Example 2, the simulations are performed with a parameter sequence randomly sampled from {max, nom, min} values. Again, there are two simulation sets comparing the control performances of the adaptive horizon and fixed horizon multi-stage MPC when $N_R = 1, 2$. Simulation results of the quad-tank system for the two controllers when $N_R = 1, 2$ are plotted in Figs. 12 and 13, respectively.

It is seen from the levels x_1, x_2 that the setpoint tracking performance of the two controllers for both cases of $N_R = 1, 2$ are again nearly identical. There are also no significant constraint violations in the bounds of x_3, x_4 for both controllers. Therefore, there are no performance losses in both reference tracking and robustness of the multi-stage MPC by including the horizon update algorithm.

The prediction horizon and computation time per iteration for both controllers when $N_R = 1, 2$ are plotted in Figs. 14 and 15, respectively.

There is a reduction in the prediction horizon as the process approaches its setpoint. The horizon update algorithm is unaffected by the pulse changes in the tank levels. This is probably because the terminal radii values (see Table 5) are of the same order as the tank levels. Therefore the pulse changes do not displace the system far enough from the terminal region to cause a significant horizon increase. The average CPU time of the fixed horizon multi-stage MPC was significantly reduced by 74% with an adaptive horizon when $N_R = 1$. Similarly, the CPU time reduction is 33% when $N_R = 2$.

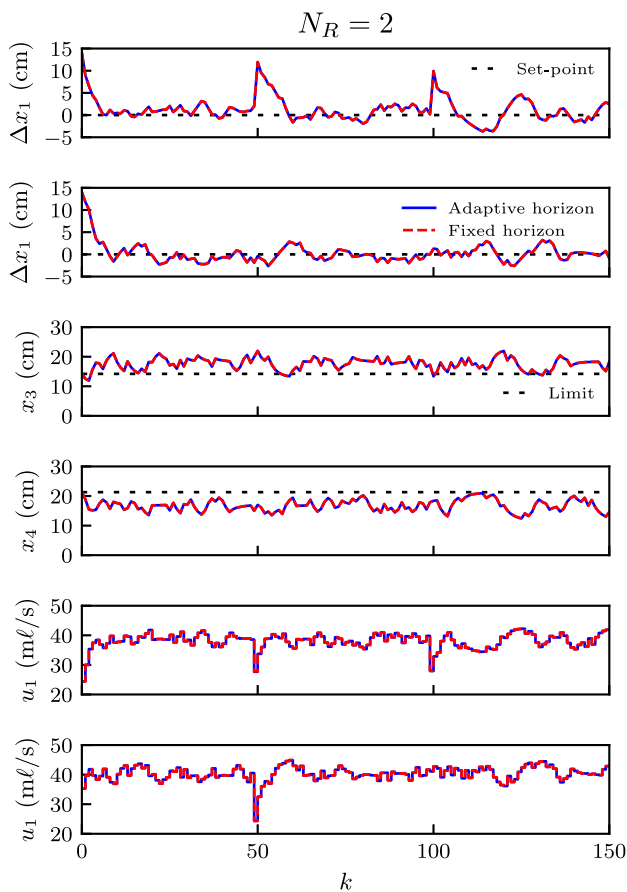


Fig. 13. Quad-tank – Simulation results comparing fixed horizon multi-stage and adaptive horizon multi-stage MPC with $N_R = 2$.

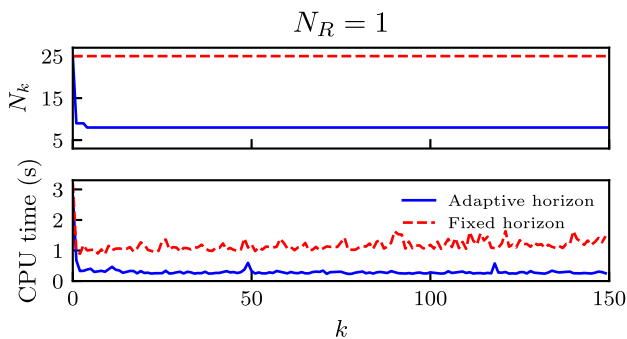


Fig. 14. Quad-tank – plot comparing prediction horizons and computation times at each iteration for ideal multi-stage and adaptive horizon multi-stage MPC with $N_R = 1$.

7. Conclusion

This paper adopts the existing adaptive horizon algorithm and extends it for multi-stage MPC to improve much-needed computational efficiency. First, the terminal costs and regions are computed independently for each uncertain parameter realization about their optimal steady states. We assume that a common terminal region exists for all uncertain parameter realizations, and the prediction horizon length is updated such that it is always reached.

The adaptive horizon framework is shown to be recursively feasible when the scenario tree is fully branched. Implementation of soft constraints may be used to avoid problem infeasibility

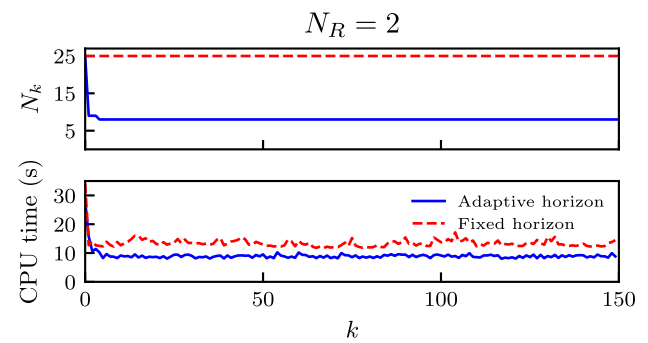


Fig. 15. Quad-tank – plot comparing prediction horizons and computation times at each iteration for ideal multi-stage and adaptive horizon multi-stage MPC with $N_R = 2$.

in practice when the scenario tree is not fully branched, but then robust constraint satisfaction is not guaranteed. The ISpS stability of the fixed horizon multi-stage MPC is retained in this framework by assuming negligible linear control error inside the common terminal region.

Simulation results on three examples demonstrate an effective reduction in the prediction horizon and computational delay, provided the system progressively approaches its setpoint, and that it is not significantly disturbed away from it. Future avenues for this work include handling economic costs, and the use of parametric sensitivities for both horizon and critical scenario updates to further reduce computation time.

CRedit authorship contribution statement

Zawadi Mdoe: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Dinesh Krishnamoorthy:** Conceptualization, Methodology, Writing – review & editing. **Johannes Jäschke:** Conceptualization, Methodology, Writing – original draft, Writing review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

- [1] J.B. Rawlings, D.Q. Mayne, M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, Vol. 2, Nob Hill Publishing Madison, WI, 2017.
- [2] K.G. Murty, S.N. Kabad, Some NP-Complete Problems in Quadratic and Nonlinear Programming, Technical Report, 1985.
- [3] P.O. Scokaert, D.Q. Mayne, Min-max feedback model predictive control for constrained linear systems, *IEEE Trans. Automat. Control* 43 (8) (1998) 1136–1142.
- [4] S. Lucia, S. Engell, Robust nonlinear model predictive control of a batch bioreactor using multi-stage stochastic programming, in: 2013 European Control Conference, ECC, IEEE, 2013, pp. 4124–4129.
- [5] D.M. de la Peña, T. Alamo, A. Bemporad, E.F. Camacho, Feedback min-max model predictive control based on a quadratic cost function, in: 2006 American Control Conference, IEEE, 2006, pp. 6–pp.
- [6] S. Lucia, T. Finkler, S. Engell, Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty, *J. Process Control* 23 (9) (2013) 1306–1319.
- [7] A.J. Krener, Adaptive horizon model predictive control, *IFAC-PapersOnLine* 51 (13) (2018) 31–36.

- [8] A.J. Krener, Adaptive horizon model predictive control and Al'brekht's method, in: *Encyclopedia of Systems and Control*, Springer, London, 2019, pp. 1–14, http://dx.doi.org/10.1007/978-1-4471-5102-9_100071.
- [9] D.W. Griffith, L.T. Biegler, S.C. Patwardhan, Robustly stable adaptive horizon nonlinear model predictive control, *J. Process Control* 70 (2018) 109–122.
- [10] Z. Mdoe, D. Krishnamoorthy, J. Jäschke, Adaptive horizon multistage nonlinear model predictive control, in: *2021 American Control Conference, ACC, IEEE*, 2021, pp. 2088–2093.
- [11] G. De Nicolao, L. Magni, R. Scattolini, On the robustness of receding-horizon control with terminal constraints, *IEEE Trans. Automat. Control* 41 (3) (1996) 451–453.
- [12] G. Pannocchia, J.B. Rawlings, S.J. Wright, Conditions under which suboptimal nonlinear MPC is inherently robust, *Systems Control Lett.* 60 (9) (2011) 747–755.
- [13] S. Yu, M. Reble, H. Chen, F. Allgöwer, Inherent robustness properties of quasi-infinite horizon nonlinear model predictive control, *Automatica* 50 (9) (2014) 2269–2280.
- [14] D.A. Allan, C.N. Bates, M.J. Risbeck, J.B. Rawlings, On the inherent robustness of optimal and suboptimal nonlinear MPC, *Systems Control Lett.* 106 (2017) 68–78.
- [15] R.C. Shekhar, Variable Horizon Model Predictive Control: Robustness and Optimality (Ph.D. thesis), University of Cambridge, 2012.
- [16] D. Krishnamoorthy, L.T. Biegler, J. Jäschke, Adaptive horizon economic nonlinear model predictive control, *J. Process Control* 92 (2020) 108–118.
- [17] H. Chen, F. Allgöwer, A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability, *Automatica* 34 (10) (1998) 1205–1217.
- [18] J. Nocedal, S. Wright, *Numerical Optimization*, Springer Science & Business Media, 2006.
- [19] A.V. Fiacco, Sensitivity analysis for nonlinear programming using penalty methods, *Math. Program.* 10 (1) (1976) 287–311.
- [20] E. Sontag, Smooth stabilization implies coprime factorization, *IEEE Trans. Automat. Control* 34 (4) (1989) 435–443, <http://dx.doi.org/10.1109/9.28018>.
- [21] E.D. Sontag, Y. Wang, New characterizations of input-to-state stability, *IEEE Trans. Automat. Control* 41 (9) (1996) 1283–1294.
- [22] D. Limón, T. Alamo, F. Salas, E.F. Camacho, Input to state stability of min-max MPC controllers for nonlinear systems with bounded uncertainties, *Automatica* 42 (5) (2006) 797–803.
- [23] D. Ralph, S. Dempe, Directional derivatives of the solution of a parametric nonlinear program, *Math. Program.* 70 (1–3) (1995) 159–172.
- [24] J. Jäschke, X. Yang, L.T. Biegler, Fast economic model predictive control based on NLP-sensitivities, *J. Process Control* 24 (8) (2014) 1260–1272.
- [25] M. Maiworm, T. Bähge, R. Findeisen, Scenario-based model predictive control: Recursive feasibility and stability, *IFAC-PapersOnLine* 48 (8) (2015) 50–56.
- [26] S. Lucia, S. Subramanian, D. Limon, S. Engell, Stability properties of multi-stage nonlinear model predictive control, *Systems Control Lett.* 143 (2020) 104743.
- [27] M. Thombre, Z.J. Yu, J. Jäschke, L.T. Biegler, Sensitivity-Assisted multistage nonlinear model predictive control: Robustness, stability and computational efficiency, *Comput. Chem. Eng.* 148 (2021) 107269.
- [28] D.M. Raimondo, D. Limon, M. Lazar, L. Magni, E.F. Camacho, Min-max model predictive control of nonlinear systems: A unifying overview on stability, *Eur. J. Control* 15 (1) (2009) 5–21.
- [29] I. Dunning, J. Huchette, M. Lubin, JuMP: A modeling language for mathematical optimization, *SIAM Rev.* 59 (2) (2017) 295–320, <http://dx.doi.org/10.1137/15M1020575>.
- [30] J. Bezanson, A. Edelman, S. Karpinski, V.B. Shah, Julia: A fresh approach to numerical computing, *SIAM Rev.* 59 (1) (2017) 65–98, <http://dx.doi.org/10.1137/141000671>.
- [31] A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Math. Program.* 106 (1) (2006) 25–57.
- [32] STFC Rutherford Appleton Laboratory, HSL. A collection of Fortran codes for large-scale scientific computation, 2019, <http://www.hsl.rl.ac.uk>.
- [33] K.-U. Klatt, S. Engell, Gain-scheduling trajectory control of a continuous stirred tank reactor, *Comput. Chem. Eng.* 22 (4–5) (1998) 491–502.
- [34] T. Raff, S. Huber, Z.K. Nagy, F. Allgöwer, Nonlinear model predictive control of a four tank system: An experimental stability study, in: *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, IEEE, 2006, pp. 237–242.