Korean J. Chem. Eng., 38(7), 1317-1332 (2021)
DOI: 10.1007/s11814-021-0791-7

pISSN: 0256-1115
eISSN: 1975-7220

INVITED REVIEW PAPER

A perspective on nonlinear model predictive control

Lorenz Theodor Biegler[†]

Chemical Engineering Department, Carnegie Mellon University Pittsburgh, PA 15213, USA

(*Received 13 March 2021 • Accepted 21 March 2021*)

Abstract−Model predictive control (MPC) is widely accepted as a generic multivariable controller with constraint handling. More recently, MPC has been extended to nonlinear model predictive control (NMPC) in order to realize high-performance control of highly nonlinear processes. In particular, NMPC allows incorporation of detailed process models (validated by off-line analysis) and also integrates with on-line optimization strategies consistent with higher-level tasks, such as scheduling and planning. NMPC for tracking and so-called "economic" stage costs has been developed, and fundamental stability and robustness properties of NMPC have been analyzed. This perspective provides an overview of NMPC concepts and approaches, as well as the underlying optimization strategies that support the solution strategies. In addition, three challenging process case studies are presented to demonstrate the effectiveness of NMPC.

Keywords: Nonlinear Programming, NLP Sensitivity, Nonlinear Model Preditive Control, Economic NMPC. Multi-stage NMPC

INTRODUCTION

Today's chemical processes are challenged by highly competitive market forces and tight operating constraints that demand on-line optimization of dynamic process systems. This can be observed in batch processes that require optimal operating recipes, and maintenance of high performance in the face of process disturbances and model mismatch. In addition, continuous processes undergo a number of dynamic changes, including slow degradation of performance, transitions due to product grade changes, operation of cyclic processes and management of upsets and transients. Recent advances in large-scale optimization algorithms and modeling strategies are key enablers to address these challenges for dynamic optimization.

In addition to the availability of dynamic system models and powerful solution strategies for off-line optimization, nonlinear model predictive control (NMPC) has become a powerful mechanism for on-line optimization. This is due to its ability to incorporate nonlinear dynamic models, often driven by first principles, perform multi-variable optimization, satisfy input and state constraints and perform in a robustly stable manner. Moreover, dynamic optimization models can be extended to link operating decisions to off-line decisions and also to integrate and coordinate the operations of multiple process systems.

For the operation of complex, integrated process systems, the impact of nonlinear models extends conventional MPC beyond mere regulation at a setpoint. Instead, NMPC can follow dynamic trajectories accurately and enforce complex recipes for batch and continuous processes. In addition, to estimate model states and parameters from process measurements, moving horizon estimation (MHE) with first principle models leads to a straightforward optimization strategy to validate plant behavior, assess observability of states and parameters from process data, and confirm model suitability through statistical inference.

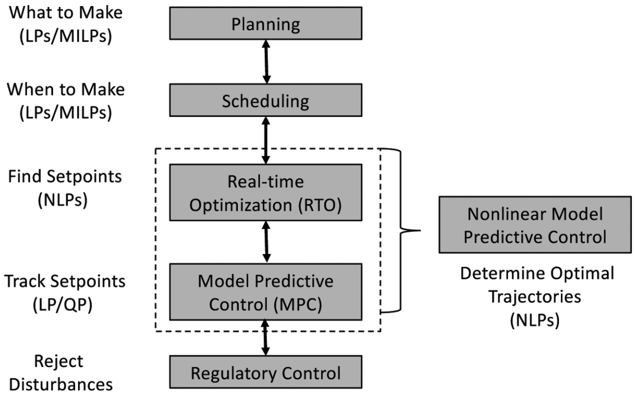


Fig. 1. Corporate-wide decision hierarchy.

NMPC also serves as a vehicle that allows the integration with on-line optimization strategies, consistent with higher level tasks such as scheduling and planning operations, as shown in Fig. 1. Key properties for the realization of this decision hierarchy are 1) feasible realizations must be transferred from the top down, and equally important, 2) accurate performance potential must be reflected from the bottom up. Through the integration of RTO and MPC models, NMPC can be viewed as the enabling element that incorporates nonlinear dynamic models into a comprehensive real-time optimization strategy [1].

Advanced optimization concepts and algorithms are the key enablers to recognize the benefits of integrated process operations [2]. First, nonlinear optimization formulations are needed to provide reliable and robust solutions, and ensure asymptotic and robust

[†]To whom correspondence should be addressed.
E-mail: biegler@cmu.edu

stability properties. Second, time critical solutions of NMPC and MHE subproblems are essential for on-line implementation and avoidance of computational feedback delays. Finally, for practical deployment, these solution strategies must be implemented within a comprehensive optimization modeling framework that leads to transparent incorporation of nonlinear process models, and allows seamless interactions with optimization, sensitivity, plant simulation and data management.

This study summarizes recent efforts along these directions. The next section introduces the formulation of NMPC problems that satisfy requirements for nominal and robust stability. The third section develops well-posed NMPC formulations and properties of the optimization problem as well as stability properties. In the fourth section we extend these formulations and properties to develop fast NMPC strategies with sensitivity-based NLP solutions that couple off-line optimization with fast on-line updates. The fifth section presents an optimization modeling framework for NMPC and MHE problems, while the sixth section presents three NMPC comprehensive case studies that highlight the concepts in the previous sections. Finally, the last section summarizes the paper and provides future research directions.

ON-LINE DYNAMIC OPTIMIZATION

For the plant model we consider the differential-algebraic equation (DAE) system in the following semi-explicit form:

$$\frac{dx}{dt} = F(x(t), y(t), u(t), w(t)),\ x(0) = x_0 \qquad (1)$$
$$0 = G(x(t), y(t), u(t), w(t))$$

where $x(t) \in \Re^{n_x}$, $y(t) \in \Re^{n_y}$, $u(t) \in \Re^{n_u}$ and $w(t) \in \Re^{n_w}$ are the differential states, algebraic states, controls and disturbance signals, respectively. We assume that the DAE system (1) is index-1 so that the algebraic equations can be solved for y in terms of (x, u, w). For the digital implementation, we consider the inputs (u, w) as zero-order holds at t_k, (u(k), w(k)), and implicitly solve (1) between t_k and t_{k+1} as:

$$x(k+1) = x(k) + \int_{t_k}^{t_{k+1}} F(x, y(x, u(k), w(k)), u(k), w(k))dt \qquad (2)$$

where we define time steps t_k with integers k>0. This leads to the discrete-time nonlinear dynamic plant with uncertainty:

$$\begin{aligned} x(k+1) &= \hat{f}(x(k), u(k), w(k)) \\ &= f(x(k), u(k)) + f_d(x(k), w(k)). \end{aligned} \qquad (3)$$

We define the mapping f: $\Re^{n_x+n_u} \rightarrow \Re^{n_x}$ with f(0, 0)=0, which represents the nominal model. The term f_d: $\Re^{n_x+n_u+n_w} \rightarrow \Re^{n_x}$ is used to describe modeling errors, estimation errors and disturbances. We assume that f(·, ·) and f_d(·, ·) are Lipschitz continuous with respect to their arguments, and that the noise w(k) is drawn from a bounded set $\mathcal{W}$

In addition to the dynamic model (3), key elements for online dynamic optimization include the choice of objective function, which is defined in terms of stage costs $\psi(x, u)$: $\Re^{n_x+n_u} \rightarrow \Re$ at t_k. These stage costs are either tracking functions, i.e., deviation from a setpoint, or economic functions, i.e., performance, cost, profit or yield.

We assume that the states and controls are restricted to the domains $\mathbb{X}$ and $\mathbb{U}$, respectively. The set $\mathbb{U}$ is compact and contains the origin, while $\mathbb{X}$ is closed and contains the origin in its interior.

Given the (estimated) state x(k) we consider the standard NMPC controller (see [3]), defined by the following parametric nonlinear programming problem (NLP):

$$J_N(x(k)) := \min_{z_l,\, v_l}\ \Psi(z_N) + \sum_{l=0}^{N-1} \psi(z_l, v_l) \qquad (4a)$$
$$\text{s.t. } z_{l+1} = f(z_l, v_l) \quad l = 0, \ldots N-1 \qquad (4b)$$
$$z_0 = x(k) \qquad (4c)$$
$$z_l \in \mathbb{X},\ v_l \in \mathbb{U},\ z_N \in \mathbb{X}_f. \qquad (4d)$$

where z_l, v_l are the predicted state and control variables at time *l* in the horizon.

For NMPC applications, the dynamic optimization problem is formulated over a time horizon of length N. For batch processes, this problem is solved at each time step over a finite, shrinking horizon with terminal properties at the end of the horizon. Continuous processes are often posed as infinite horizon problems. However, to make them tractable for nonlinear systems, a moving horizon window of length N is formulated with terminal constraints, i.e., $x(k+N) \in \mathbb{X}_f$ with the terminal set $\mathbb{X}_f \subset \mathbb{X}$. In addition, we add a terminal cost denoted by $\Psi(\cdot) : \Re^{n_x} \rightarrow \Re$. Both stage and terminal costs are assumed to be Lipschitz continuous in their arguments.

In problem (4) N is assumed to be sufficiently long and $\Psi(z_N)$ is sufficiently large so that $z_N \in \mathbb{X}_f$ holds for the solution of (4). As shown in [4,5], the $\mathbb{X}_f$ constraint can even be omitted in (4) under these conditions. Alternately, terminal regions can also be calculated and imposed, based on concepts in [6]. Moreover, Griffith et al. [7] calculate $\mathbb{X}_f$ and determine N adaptively; this approach leads to an inactive terminal constraint, with $z_N \in \mathbb{X}_f$ in the interior of the terminal region. Moreover, as developed in [6-8], constructive methods are available to determine $\Psi(\cdot)$ and $\mathbb{X}_f$.

After solving Problem (4) the control action is extracted from the optimal trajectory $\{z_0^*, \ldots, z_N^*, v_0^*, \ldots, v_{n-1}^*\}$ as $u(k)=v_0^*$. At the next time step, the plant evolves as in (3) and we shift the time sequence one step forward, k=k+1, obtain the next state estimate x(k+1) and solve the next NMPC problem (4). This recursive strategy gives rise to the feedback law, $u(k)=\kappa(x(k))$ with $\kappa(\cdot) : \Re^{n_x} \rightarrow \Re^{n_u}$ and system (3) becomes:

$$\begin{aligned} x(k+1) &= \hat{f}(x(k), \kappa(x(k)), w(k)) \\ &= f(x(k), \kappa(x(k))) + f_d(x(k), w(k)) \end{aligned} \qquad (5)$$

We refer to the above strategy as *ideal* NMPC (iNMPC), where the on-line calculation time and its influence on delayed control injection are neglected. Correspondingly we denote the feedback law of iNMPC as $u^{id}(k)=\kappa^{id}(x(k))$. iNMPC has well-known stability properties (see [3,9]), which we review in the third section.

State Estimation for NMPC

Because NMPC strategies require knowledge of all of the plant state variables x(k), the estimation of these states from the plant measurements y(k) is essential. A number of formulations can be used for this task, including embedded observers and nonlinear extensions of Kalman filters [10]. In particular, moving horizon estimation (MHE) provides a consistent coupling to NMPC, because it relies on an optimization formulation that can use the same dynamic models to estimate sequences of states, parameters and disturbances from plant measurements, up to the current sampling time.

While a comprehensive review of state estimation is beyond the scope of this article, a brief summary of updates is provided here. Recent results in robustness and stability of MHE have been developed in [3,11], and a number of efficient solution strategies have been developed in [12-14].

For state estimation the MHE problem uses $\mathcal{N}$ measurement vectors in the past y(k–$\mathcal{N}$); …, y(k) to find the most likely sequence of state estimates $x_{-\mathcal{N}|k}$, …, $x_{0|k}$. Based on maximum likelihood principles, the resulting MHE optimization problem is defined as follows:

$$\begin{aligned}\min_{x_{-\mathcal{N}},\, w_k}\ & \Phi_{-\mathcal{N}}(x_{-\mathcal{N}|k}, x_{-\mathcal{N}|k-1}, \Pi_{-\mathcal{N}|k-1})\\ & + \sum_{i=-\mathcal{N}}^{0} (y(k+i)-h(x_{i|k}))^T R_i^{-1}(y(k+i)-h(x_{i|k}))\\ & + \sum_{i=-\mathcal{N}}^{-1} w_{i|k}^T Q_i^{-1} w_{i|k} \qquad \text{(6a)}\\ \text{s.t. } & x_{l+1|k} = f(x_{l|k}, u_{l|k}) + f_d(x_{l|k}, w_{l|k}), \qquad \text{(6b)}\\ & w_{l|k} \in \mathbb{W},\ \ l \in \{-\mathcal{N}, -\mathcal{N}+1, \ldots, -1\} \qquad \text{(6c)}\end{aligned}$$

where $\Phi_{-\mathcal{N}}$ (·) is the arrival cost, $x_{-\mathcal{N}|k-1}$ and $\Pi_{-\mathcal{N}|k-1}$ are the prior state estimate and its covariance, R_i and Q_i are the measurement noise and disturbance covariance matrices and $w_{i|k}$ is the process disturbance. By solving (6) the most likely sequence of estimated disturbances and states is determined for the data set and prior estimates. Also, the current state of the plant is determined from MHE (i.e., x(k) : =$x_{0|k}$). Note that only a finite data window of length $\mathcal{N}$ is used for this problem. Therefore, an arrival cost $\Phi_{-\mathcal{N}}$ (·) summarizes all previous data before the current horizon window k–$\mathcal{N}$.

Based on the formulation of (6), the on-line implementation of coupled MHE-NMPC is shown in Fig. 2. At time t_k problem (6) is solved to determine x(k) and problem (4) is solved, using x(k) to determine u(k), which is injected into the plant. At the next time step t_{k+1} the MHE and NMPC subproblems are updated and the optimization cycle repeats. Further information on the formulation and solution strategies for MHE problems can be found in [3,15,16].

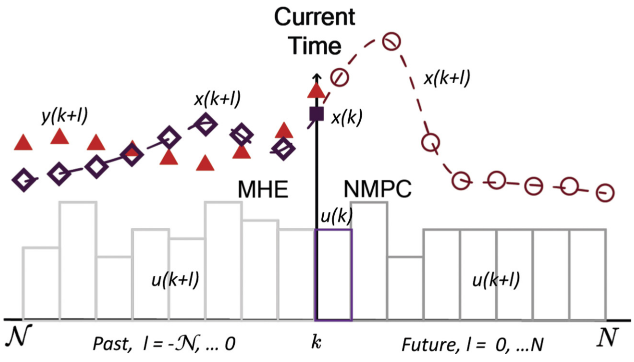


Fig. 2. Nonlinear model-based estimation and control: Coupled MHE and NMPC.

NONLINEAR PROGRAMMING ELEMENTS AND PROPERTIES FOR NMPC

The solution of problem (4) is time critical for on-line implementation. As a result, careful attention must be paid to formulating NLPs that can be solved efficiently and reliably at each time step. Moreover, since problem (4) is a parametric problem that depends on changes in data from the plant, it is essential that this problem be well-posed with well-characterized solutions.

Elements of NLP Formulations

The NMPC (4) and MHE (6) formulations can incorporate the dynamic model (3) in a number of ways. To form this system, the DAE (1) needs to be solved from t_k to t_{k+1} through an appropriate, discretization strategy of high accuracy (high order) and numerical stability. General DAE solvers can be classified into two types [17].

- *Single-step discretization (e.g., Runge-Kutta (R-K) methods).* High order implicit R-K methods are widely used and have A-stable properties, which are useful for stiff systems. Moreover, as self-starting methods they do not require backward interpolation and easily handle discontinuous input profiles (u(k), w(k)) accurately.

- *Linear Multi-Step discretizations (e.g., Adams methods and Backward Difference Formulae (BDF)).* Implemented as predictor-corrector methods of high order, LMS must be initialized by single step methods, and only BDF methods are reliable for stiff systems. To solve initial value problems (IVPs), LMS methods are generally faster than R-K methods, but because they require backward interpolation, discontinuous input profiles (u(k), w(k)) will require frequent restarting.

The choice of discretization methods strongly impacts the solution strategy for (4). The sequential (or nested) approach with an embedded initial value (IV) DAE solver is easy to implement and leads to a smaller optimization problem. However, with many decision variables, calculation of gradients through direct or adjoint sensitivity becomes expensive, and convergence noise from the IV solver also propagates to the gradients; second derivatives are rarely calculated due to the increased computational cost. Moreover, dynamic systems with open-loop instabilities may lead to failure of the IV solver.

In contrast, a full, R-K type discretization of (1) (e.g., orthogonal collocation on finite elements) leads to a high order representation for (3), and a large, sparse optimization problem (4). With this approach, first, and even second, derivatives for (4) can be evaluated through automatic differentiation and are much cheaper to calculate. The fully discretized formulation of (4) allows a simultaneous strategy for solution of the dynamic system along with the optimization problem. Moreover, such an approach has the same dichotomy property as boundary value solvers, which can handle open-loop instability reliably and efficiently. On the other hand, the simultaneous (or full discretization) approach requires a large-scale NLP solver for efficient performance. An illustration of the structure of the sequential and simultaneous approaches is presented in Fig. 3.

To solve the NMPC and MHE problems, NLP solvers can be classified as follows.

- *Derivative free optimization algorithms (e.g., genetic algorithms,*

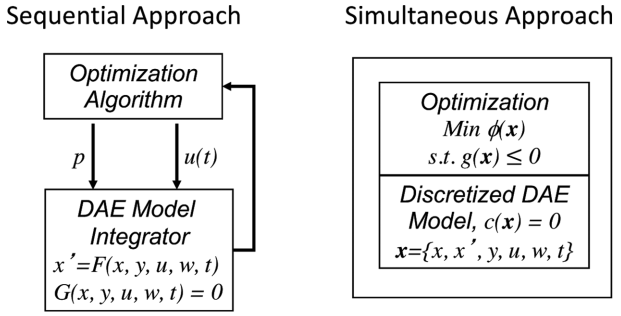


Fig. 3. Structure of sequential and simultaneous approaches for dynamic optimization.

simulated annealing, particle swarm, pattern searches [18]) require no derivatives and are primarily designed for unconstrained problems. While they are easy to apply, especially with sequential approaches, they become very expensive for optimization problems with more than about 100 variables, and are not suited for large NMPC problems.

- *Active set strategies such as successive quadratic programming (SQP) and reduced gradient methods (e.g., GRG, CONOPT,* [2]) allow simultaneous minimization and constraint satisfaction. They have superlinear convergence properties and work well for moderately sized NLPs (>100 decision variables).

- *Barrier methods with exact Hessians and sparse Newton methods that solve the optimality conditions directly (e.g., IPOPT, KNITRO, LOQO,* [19]) allow full exploitation of the problem structure, have quadratic convergence and can handle very large NLPs (>1,000,000 variables).

- A byproduct of the barrier method leads to the sensitivity of nearby optimal solutions, at virtually no cost, even for very large NLPs. As described in sections 4 and 5, sIPOPT and k_aug are recent implementations of this approach.

More information on NLP solvers for dynamic optimization problems can be found in [2,20].

Existence, Uniqueness and Robustness of NLP Solutions

In a classic paper [21] three examples are presented and analyzed to show that NMPC, as formulated in (4), is not robust to data perturbations, because the NLP does not have feasible solutions for some realizations of the input data x(k), w(k). More recently, it was shown in [22] that these failures are due to lack of regularity conditions in the NLP formulation, and suitable modifications to problem (4) can overcome them. Motivated by these studies, this section presents sufficient conditions for existence, uniqueness and robustness of NLP solutions, which are essential for the realization of stable and robust NMPC.

To analyze the stability properties of model predictive control, the Lyapunov function is directly related to the objective function in (4). Asymptotic and robust stability within this framework relies on satisfaction of boundedness and descent properties of the Lyapunov function, as well as recursive feasibility for the sequence of NLP subproblems (4). These properties require that (4) be well-posed, with solutions that are uniformly continuous with respect to the problem input data, (e.g., wrt x(k)). This property is essential to ensure that any bounded input leads to a bounded solution (z_l, v_l) of (4).

Moreover, differentiability of the solution of (4), particularly for u(k) $(:=v_0)$ wrt x(k), leads to fast sensitivity-based strategies for online NMPC computations. As a result, careful formulation of (4) is essential to satisfy boundedness, continuity and differentiability properties, which lead to stable and robust NMPC strategies.

By rewriting Problem (4), with $\mathbf{x}=(z_0;\ \ldots,\ z_N,\ v_0,\ \ldots,\ v_{N-1})$ and p=x(k) as:

$$\min_{x} \phi(\mathbf{x};\ p),\ \text{s.t.}\ c(\mathbf{x};\ p)=0,\ g(\mathbf{x};\ p)\le 0, \tag{7}$$

we characterize its solution as a Karush-Kuhn-Tucker (KKT) point, which satisfies the following conditions for (7).

Definition 1 (KKT, [23]) The KKT conditions for Problem (7) are given by:

$$\begin{aligned} &\nabla\phi(\mathbf{x}^*)+\nabla c(\mathbf{x}^*)\lambda+\nabla g(\mathbf{x}^*)\nu=0 \\ &c(\mathbf{x}^*)=0,\ 0\le \nu\perp g(\mathbf{x}^*)\le 0 \end{aligned} \tag{8}$$

for some multipliers $(\lambda,\ \nu)$, where $\mathbf{x}^*$ is a KKT point[1]. We also define $\mathscr{L}=\phi(\mathbf{x})+c(\mathbf{x})^T\lambda+g(\mathbf{x})^T\nu$ as the Lagrange function of (7).

A constraint qualification (CQ) is required so that a KKT point is necessary for a local minimizer of (7). For Problem (7) the following CQ is widely invoked.

Definition 2 (LICQ, [23]) The linear independence constraint qualification (LICQ) holds at $\mathbf{x}^*$ when the gradient vectors

$$\nabla c(\mathbf{x}^*;\ p)\ \text{and}\ \nabla g_j(\mathbf{x}^*;\ p);\ \ j\in J\ \text{where}\ J=\{j|g_j(\mathbf{x}^*;\ p)=0\} \tag{9}$$

are linearly independent. LICQ also implies that the multipliers $(\lambda,\ \nu)$ are unique.

To develop a sufficient condition for local optimality, we define the strong second-order condition (SSOSC), which requires positive definiteness of the reduced Hessian of $\mathscr{L}$ as follows.

Definition 3 (SSOSC, [24]) The KKT point at $\mathbf{x}^*$ with multipliers λ and ν is a strict local optimum if the following strong second-order sufficient conditions (SSOSC) hold at $\mathbf{x}^*$:

$$q^T\nabla_{xx}\mathscr{L}(\mathbf{x}^*,\ \lambda,\ \nu;\ p)q>0\ \ \text{for all}\ q\ne 0 \tag{10}$$

such that

$$\begin{aligned} &\nabla c_i(\mathbf{x}^*;\ p)^T q=0,\ i=1,\ \ldots,\ n_c \\ &\nabla g_j(\mathbf{x}^*;\ p)^T q=0,\ \text{for}\ \nu_j>0,\ j\in J. \end{aligned} \tag{11}$$

Note that by adding $\|\mathbf{x}-\mathbf{x}^*\|_W^2$ to the objective in (7), where W is a positive definite weighting matrix, the KKT conditions of (7) are unchanged and $(\mathbf{x}^*,\ \lambda,\ \nu)$ remains a KKT point. Moreover, by defining the matrix Z as a basis of the nullspace of strongly active constraint gradients (11) and choosing W, with sufficiently large eigenvalues for Z^TWZ, then SSOSC can always be satisfied at $\mathbf{x}^*$ (In fact, adding such a term is an automatic regularization feature in the IPOPT solver.).

For solutions of (7) that satisfy LICQ and SSOSC, Lipschitz continuity of $\mathbf{x}^*(p)$ can be guaranteed by the *strong regularity theorem* due to [24], who treated the KKT conditions as generalized equations. To ensure *differentiability* of $\mathbf{x}^*(p)$ with respect to p, we consider the following definition and theorem.

Definition 4 (Strict Complementarity (SC), [25]) At a KKT

[1] Note the convention $\{\nabla c(x)\}_{ij}=\partial c_j/\partial x_i$ for row i and column j.

point of (7) ($\mathbf{x}^*$, λ, ν), the strict complementarity condition (SC) is defined by $\nu_j - g_j(\mathbf{x}^*; p) > 0$ for each $j \in J$.

At the solution of (7), the following theorem provides the foundation for sensitivity-based NMPC.

Theorem 1 *[25] Let $\mathbf{x}^*(p)$ be a KKT point that satisfies (8), and assume that SC, LICQ and SSOSC hold at $\mathbf{x}^*$. Further let the functions ϕ, c, g be at least ℓ+1 times differentiable in $\mathbf{x}$ and ℓ times differentiable in p.* (We assume $\ell \geq 2$.) Then

- $\mathbf{x}^*$ is an isolated minimizer, and the associated multipliers λ and ν are unique.

- for p in a neighborhood of p_0 the set of active constraints remains unchanged,

- for p in a neighborhood of p_0 there exists an ℓ times differentiable function $s(p)=[\mathbf{x}^*(p)^T, \lambda(p)^T, \nu(p)^T]$, that corresponds to a locally unique minimum for (7).

More general results on Lipschitz continuity of the solution of (7) can be derived under the following assumptions.

Definition 5 (MFCQ, [23]) For Problem (7), the Mangasarian-Fromovitz constraint qualification (MFCQ) holds at the optimal point $\mathbf{x}^*(p)$ if and only if a) $\nabla c(\mathbf{x}^*; p)$ is linearly independent, and b) there exists a vector q such that

$$\nabla c(\mathbf{x}^*; p)^T q = 0, \ \nabla g_j(\mathbf{x}^*; p)^T q < 0 \quad j \in J. \tag{12}$$

MFCQ implies that the set of KKT multipliers is a bounded convex polytope [26]. Another useful constraint qualification is given as:

Definition 6 (CRCQ, [27]) For Problem (7), the constant rank constraint qualification (CRCQ) holds at ($\mathbf{x}^*$; p_0), when all subsets of the active constraint gradients $\nabla g_j(\mathbf{x}; p)$ $j \in J$ and $\nabla c(\mathbf{x}; p)$ retain constant rank for all points in a neighborhood of ($\mathbf{x}^*$; p_0).

Finally, if MFCQ holds at a KKT point but not LICQ, the multipliers (λ, ν) are bounded but no longer unique, and a more general second order condition is needed.

Definition 7 (GSSOSC, [28]) The generalized strong second-order sufficient condition (GSSOSC) holds at $\mathbf{x}^*$ when SSOSC holds for all KKT multipliers (λ, ν) that satisfy the KKT conditions of (7).

MFCQ and GSSOSC at KKT points are the weakest conditions under which a perturbed solution of (7) is locally unique and $\mathbf{x}^*(p)$ is Lipschitz continuous [29]. On the other hand, since the active sets may be nonunique, we cannot expect $\mathbf{x}^*(p)$ to be differentiable. Instead with MFCQ, GSSOSC and CRCQ, *directional derivatives* for $\mathbf{x}^*(p)$ can be calculated with a particular QP formulation [28]. These properties facilitate a path following algorithm [30,31] that allows $\mathbf{x}^*(p)$ to be computed through a continuation strategy in p. This algorithm, and related approaches, are sufficient to obtain sensitivity updates in an NMPC context.

NMPC Problem Reformulation

To exploit the properties of NLP solutions and sensitivity, we strengthen our assumptions on the dynamic model and objective function in (4), and assume that f (x, u), ψ(x, u) and Ψ(x, u) have Lipschitz continuous second derivatives with respect to their arguments. As detailed below, we also assume N sufficiently long and $\Psi(z_N)$ sufficiently large, so that $z_N \in \mathbb{X}_f$ holds in the nominal case (see [4,5]).

Moreover, we relax $\mathbb{X}$ and $\mathbb{X}_f$ with ℓ_1 penalty terms to develop a more robust modification of (4). Without loss of generality $\mathbb{X}$, $\mathbb{X}_f$ and $\mathbb{U}$ can be represented by simple upper and lower bounds, and we write $\mathbb{X}$, $\mathbb{X}_f$ as inequalities $g(z_l) \leq 0$ and $g(z_N) \leq 0$, respectively. This leads to the following reformulation of (4):

$$\begin{aligned} J_N(x(k)) := \min_{z_l,\, v_l,\, \xi_l} \ & \Psi(z_N) + \sum_{l=0}^{N-1} \psi(z_l, v_l) + \sum_{l=0}^{N} \rho \xi_l^T \mathbf{1} \\ \text{s.t. } & z_{l+1} = f(z_l, v_l), \ l = 0, \ldots, N-1 \\ & z_0 = x(k) \\ & g(z_l) \leq \xi_l, \ \xi_l \geq 0; \ l = 0, \ldots, N \\ & v_l \in \mathbb{U}, \ l = 0, \ldots, N-1 \end{aligned} \tag{13}$$

where ξ_l is a penalty variable vector and $\mathbf{1} = [1, 1, \ldots, 1]^T$. This leads to a soft constrained problem for which the NLP always has a feasible solution. Moreover, for the equality constraints, we note that the Jacobian with respect to z_l is block lower triangular, with identity matrices on the diagonal. Hence, the equality constraint gradients contain a nonsingular basis matrix, and are linearly independent. Therefore, it is straightforward to show that MFCQ always holds at the solution of (13) (see [31]), and because the inequalities are simple bounds, CRCQ is also satisfied. Under these conditions the multipliers of (13) are bounded. Moreover, GSSOSC is also straightforward to satisfy through addition of a sufficiently large quadratic regularization term. These terms are compatible with tracking stage costs as well as economic stage costs [31].

With MFCQ, CRCQ and GSSOSC satisfied, selecting ρ larger than a finite threshold, $\rho > \bar{\rho}$, will drive ξ_l to zero, where $\bar{\rho}$ is the dual norm of the multipliers at the solution of Problem (4). If ξ_l=0, then the solution of (13) is identical to the solution of (4), and nominal stability properties with (13) are identical to those of (4). Finally, since a solution with ξ_l>0 for arbitrarily large ρ implies that Problem (4) is locally infeasible, we assume that a finite $\bar{\rho}$ can be found as long as Problem (4) is well-posed.

The above properties ensure that the solution of (13) is Lipschitz continuous (and therefore uniformly continuous) for all perturbations of p=x(k). This is required for nominal and robust stability properties described below. Moreover, differentiability of the solution can be obtained through inexact solutions of (13), via barrier formulations and Theorem 1. Hence, the barrier (or interior point) approach provides continuous and differential perturbed solutions of (13) with respect to the initial state and disturbances.

Interior-Point NLPs

To explore continuity and sensitivity properties of Problem (13), we rewrite (7) slightly as:

$$\min_{x, r} \phi(\mathbf{x}; p), \ \text{s.t. } c(\mathbf{x}; p) = 0, \ g(\mathbf{x}; p) + r = 0, \ r \geq 0 \tag{14}$$

With interior-point solvers, the inequality constraints of problem (14) are handled implicitly by adding barrier terms to the objective function,

$$\min \ \phi(\mathbf{x}; p) - \mu \sum_{j=1}^{n_r} \ln(r^{(j)}), \tag{15a}$$

$$\text{s.t. } c(\mathbf{x}; p) = 0, \ g(\mathbf{x}; p) + r = 0 \tag{15b}$$

where $r^{(j)}$ denotes the j^{th} component of slack variable vector r. Solving (15) for the sequence of $\mu^l \rightarrow 0$, with l=0, 1, 2, …, ∞, leads to solution of (14). As shown in [32], convergence of solutions of (15) to (14) has been proved under GSSOSC and MFCQ.

For a given barrier parameter value μ, IPOPT [33] solves the primal-dual optimality conditions of barrier problems (15) directly,

$$\nabla\phi(\mathbf{x};\,p)+\nabla c(\mathbf{x};\,p)\lambda+\nabla g(\mathbf{x};\,p)\nu=0 \tag{16a}$$

$$c(\mathbf{x};\,p)=0 \tag{16b}$$

$$g(\mathbf{x};\,p)+r=0 \tag{16c}$$

$$\mathbf{R}\nu=\mu\mathbf{1}, \tag{16d}$$

where $\mathbf{R}$=diag(r) and $\mathbf{1}=[1, 1, \ldots, 1]^T$. To solve this system of nonlinear equations, IPOPT uses an exact Newton method; at the *i*th Newton iteration, the search direction is computed by linearization of the KKT conditions (16),

$$\left[\begin{array}{cc|cc} \mathbf{W}_i & \mathbf{A}^{\mathbf{c}}{}_i & \mathbf{A}^{\mathbf{g}}{}_i & 0 \\ \mathbf{A}^{\mathbf{c}T}_i & 0 & 0 & 0 \\ \hline \mathbf{A}^{\mathbf{g}T}_i & 0 & 0 & I \\ 0 & 0 & \mathbf{R}_i & \mathbf{V}_i \end{array}\right]\left[\begin{array}{c} \Delta\mathbf{x}_i \\ \Delta\lambda_i \\ \hline \Delta\nu_i \\ \Delta r_i \end{array}\right]=-\left[\begin{array}{c} \nabla L(\mathbf{x}_i,\,\lambda_i,\,\nu_i;\,p) \\ c(\mathbf{x}_i;\,p) \\ \hline g(\mathbf{x}_i;\,p)+r \\ \mathbf{R}_i\nu-\mu\mathbf{1} \end{array}\right] \tag{17}$$

where $\mathbf{A}^{\mathbf{c}}{}_i : =\nabla c(\mathbf{x}_i;\,p)$, $\mathbf{A}^{\mathbf{g}}{}_i : =\nabla g(\mathbf{x}_i;\,p)$ and $\mathbf{W}_i\in\Re^{n\times n}$ is the Hessian of the Lagrange function, $\mathscr{L}(x,\,\lambda,\,\nu)=\phi(x)+c(x)^T\lambda+(g(x)+r)^T\nu$. After solving a sequence of barrier problems for $\mu\rightarrow0$, the solver returns the solution $s^{*,T}(p)=[\mathbf{x}^{*,T}\ \lambda^{*,T}\ \nu^{*,T},\ r^{*,T}]$ for Problem (14).

Stability Properties of Ideal NMPC

Based on discrete Lyapunov concepts, the nominal and robust stability results for the iNMPC controller, $u^{id}=\kappa^{id}(x)$ [9,34] obtained from the solution of (13), can be summarized as follows.

Definition 8 (Comparison Functions) A function $\alpha:\mathbb{R}_+\rightarrow\mathbb{R}_+$ is of class $\mathscr{K}$ if it is continuous, strictly increasing, and $\alpha(0)=0$. A function $\alpha:\mathbb{R}_+\rightarrow\mathbb{R}_+$ is of class $\mathscr{K}_\infty$ if it is a $\mathscr{K}$ function and $\lim_{s\rightarrow\infty}\alpha(s)=\infty$. A function $\beta(s,\,k):\mathbb{R}_+\times\mathbb{Z}_+\rightarrow\mathbb{R}_+$ is of class $\mathscr{KL}$ if, for each $k\geq0$, $\beta(\cdot,\,k)$ is a $\mathscr{K}$ function, and, for each $s\geq0$, $\beta(s,\,\cdot)$ is nonincreasing and $\lim_{k\rightarrow\infty}\beta(s,\,k)=0$.

Assumption 2 *(Nominal Stability Assumptions of iNMPC)*

1. The terminal penalty $\Psi(\cdot)$, satisfies $\Psi(z)>0$, $\forall z\in\mathbb{X}_f\backslash\{0\}$,

2. There exists a local control law $u=\kappa_f(z)$ defined on $\mathbb{X}_f$, such that $f(z,\,\kappa_f(z))\in\mathbb{X}_f$, $\forall z\in\mathbb{X}_f$, and $\Psi(f(z,\,\kappa_f(z)))-\Psi(z)\leq-\psi(z,\,\kappa_f(z))$, $\forall z\in\mathbb{X}_f$.

3. The optimal stage cost $\psi(x,\,u)=\psi(x,\,\kappa(x))$ satisfies $\alpha_p(|x|)\leq\psi(x,\,u)\leq\alpha_q(|x|)$ where $\alpha_p(\cdot)$ and $\alpha_q(\cdot)$ are $\mathscr{K}$ functions.

The nominal stability property for iNMPC is given by the following theorem.

Theorem 3 *(Nominal Stability of iNMPC, [3]) Consider the moving horizon problem (13) and associated control law $u=u^{id}$, that satisfies Assumption 2. Then, $J_N(x)$ from (13) is a Lyapunov function and the closed-loop system is asymptotically stable.*

From the proof of this theorem, asymptotic stability is obtained through a sufficient reduction of the Lyapunov function so that the sequence of states x(k) attracts to the zero setpoint. On the other hand, the robust input-to-state stability (ISS) property allows an additional term related to |w| in the difference of Lyapunov functions. In this case, the sequence of states x(k) attracts only to a neighborhood related to |w|, around the setpoint. For the analysis of robust stability properties of iNMPC ISS conditions [9,35] are defined by:

Definition 9 (Input-to-State Stability)

1. The system (3) is ISS in $\mathbb{X}$ if there exists a $\mathscr{KL}$ function β, and a $\mathscr{K}$ function γ such that for all w in the bounded set $\mathscr{W}$

$$|x(k)|\leq\beta(|x(0)|,\,k)+\gamma(|w|),\ \forall k\geq0,\ \forall x(0)\in\mathbb{X} \tag{18}$$

2. A function $V(\cdot)$ is called an ISS-Lyapunov function for system (3) if there exist a set $\mathbb{X}$, $\mathscr{K}_\infty$ functions α_1, α_2, α_3, σ, $\forall x\in\mathbb{X}$, $\forall w\in\mathscr{W}$, and $\alpha_1(|x|)\leq V(x)\leq\alpha_2(|x|)$ and $V(\hat{f}(x,\,u,\,w))-V(x)\leq-\alpha_3(|x|)+\sigma(|w|)$

If $\mathbb{X}$ is a robustly invariant set for system (3) and $V(\cdot)$ is an ISS-Lyapunov function for this system, then the resulting system is ISS in $\mathbb{X}$ [6,9].

For the iNMPC controller, we recognize that with $\bar{x}(k+1)=f(x(k),\,u(k))$, there exists a future mismatch $x(k+1)-\bar{x}(k+1)=f_d(x(k),\,w(k))$ at the next time step, giving rise to two different problems with optimal costs $J_N(\bar{x}(k+1))$ and $J_N(x(k+1))$, respectively. We define the mismatch term, $\varepsilon(x(k+1)) : =J_N(x(k+1))-J_N(\bar{x}(k+1))$ and from Theorem 1, there exists a local positive Lipschitz constant L_J such that $\forall x\in\mathbb{X}$, $\varepsilon(x(k+1))\leq L_J|f_d(x(k),\,w(k))|$.

We make the following assumptions and establish robust stability of the iNMPC controller from the following theorem.

Assumption 4 *(Robust Stability Assumptions)*

1. *The solution $s^*(p)$ in Problem (14) satisfies LICQ, SOSC and SC. From Theorem 1, the objective function and $s^*(p)$ are therefore continuous and differentiable with respect to p and the resulting feedback law, derived from $s^*(p)$, can be represented as $u=\kappa(x)$.*

2. *$f_d(x,\,w)$ is Lipschitz with respect to its arguments so that $|f_d(x,\,0)|\leq\alpha_0(|x|)$, where $\alpha_0(|x|)$ is a $\mathscr{K}$ function.*

Theorem 5 *(Robust ISS Stability of iNMPC [9], [35]) Under Assumptions 2 and 4 with $\alpha_0(|x|)\leq\frac{\eta}{L_J}\alpha_p(|x|)$ and $\eta\in(0,\,1)$, the cost function $J_N(x)$ obtained from the solution of (13) is an ISS-Lyapunov function and the resulting closed-loop system is ISS stable.*

ON-LINE NMPC STRATEGIES AND STABILITY PROPERTIES

To enable fast on-line solution of (14) for NMPC, we consider the sensitivity of the NLP solution of (14) with respect to the data p. Here, the optimal primal and dual variables can be treated as implicit functions of p. For a *sufficiently small* $\mu^l>0$, the KKT conditions (16) can be expressed as the equations $\varphi(s^*(p);\,p)=0$ and we denote $\mathbf{K}^*(p_0)$ as the Jacobian of $\varphi(s^*(p);\,p)$ with respect to s^*.

To compute approximate solutions around a nominal solution $s^*(p_0)$, we note that when LICQ and SSOSC are satisfied at the solution of (15), Theorem 1 holds. Therefore, application of the implicit function theorem to (16) at $s^*(p_0)$ leads to:

$$\mathbf{K}^*(p_0)\frac{\partial s^*}{\partial p}=-\left.\frac{\partial\varphi(s(p);\,p)}{\partial p}\right|_{s^*(p_0),\,p_0} \tag{19}$$

and first-order estimates of neighboring solutions are obtained from:

$$s^*(p)=s^*(p_0)+\frac{\partial s^{*T}}{\partial p}(p-p_0)+O(|p-p_0|^2)+O(\mu) \tag{20}$$

$$\tilde{s}(p)=s^*(p_0)+\frac{\partial s^{*T}}{\partial p}(p-p_0) \tag{21}$$

where $\tilde{s}(p)$ is an approximate solution of $s^*(p)$ and we assume that μ is small enough that the $O(\mu)$ term is negligible. From continuity and differentiability of the optimal solution vector, there exists a positive Lipschitz constant L_q such that,

$$|\tilde{s}(p) - s^*(p)| \le L_q |p - p_0|^2. \tag{22}$$

and a first order difference approximation $\Delta s(p) = \tilde{s}(p) - s^*(p_0)$ can be obtained from:

$$\mathbf{K}^*(p_0)\Delta s(p) = -\varphi(s^*(p_0); p). \tag{23}$$

Also, for the step length $\tau \in [0, 1]$, the quantity

$$\hat{s} = s^*(p_0) + \tau \Delta s(p) \tag{24}$$

has the property:

$$|\hat{s} - s^*(p)| \le (1 - \tau)\left|\frac{\partial s^{*T}}{\partial p}(p - p_0)\right| + L_q |p - p_0|^2. \tag{25}$$

This result will be used in a heuristic strategy, called clipping, described below.

Active Set Changes

When $(p-p_0)$ induces a change in the active constraint set at the solution of (14), $\hat{r}^{(j)}<0$ or $\hat{\nu}^{(j)}<0$ with $\tau=1$, and the KKT conditions are violated. Moreover, tracking $s^*(p)$ becomes nonsmooth, the linearization (23) is invalid and Theorem 1 no longer holds. Nevertheless, GSSOSC, MFCQ and CRCQ in (14) still lead to Lipschitz continuity of $s^*(p)$ as well as *directional differentiability* of $\mathbf{x}^*(p)$ along $(p-p_0)$. To deal with this case, path-following algorithms have been developed in [30,31,36] to track $\mathbf{x}^*(p)$ with respect to p. These methods provide a rigorous treatment of NLP sensitivity, but are more expensive than the simple update (23).

A cheap alternative to path-following, called "clipping in the first interval," perturbs the solution only up to the active set change, using (24) with $\tau<1$. The clipping strategy chooses the largest value of $\tau \in (0, 1]$ so that the correction for $u(k+1) (= v_0^* + \tau \Delta v_0) \in \mathbb{U}$, since feasibility of the soft constrained formulation (13) only requires $v_l \in \mathbb{U}$. Therefore, clipping ensures that the perturbed control variable value, u(k) remains within its bounds but at the expense of a higher disturbance error. Because no additional computational cost is required beyond (23), clipping is incorporated within advanced step NMPC, which is developed and analyzed in the following section. Additional discussion of these strategies can be found in [31].

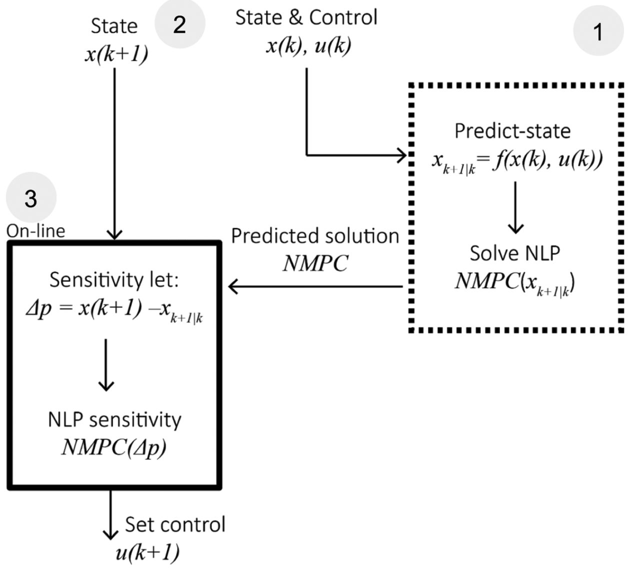


Fig. 4. Sketch of advanced step NMPC for $N_s=1$ [1]. Having (x(k), u(k) predict $x_{k+1|k}$ and solve Problem (4) in background [2]. Wait until x(k+1) arrives [3]. Apply a fast correction to v_0 from (4) using the sensitivity update (23) to determine u(k+1). Set k : =k+1 and repeat.

The Advanced-step Approach

The above NLP and sensitivity formulations comprise the off-line and on-line components, respectively, of advanced step NMPC strategies shown in Fig. 4. The advanced step approach is rooted in [37], which divides the NMPC computation into two phases, an off-line phase that solves the NMPC problem for the desired control, based on predicted states, and an on-line phase that corrects the desired control based on updated states (or estimates), and injects the control into the plant. With this approach, the only computational delay in determining the control is due to the sensitivity correction, which is usually a tiny fraction of the time to solve Problem (4).

At time t_k we use the current estimate x(k) and control u(k) to predict the future state at t_{k+N_s}, where N_s time steps are needed to solve (13) in background. Using the predicted state $x(k+N_s)$ we begin execution of Problem (13) at t_k. From solution of (13) we obtain the KKT matrix $\mathbf{K}^*$. Once the actual state (or estimate) $x(k+N_s)$ becomes available, we compute a sequence of fast backsolves with $\mathbf{K}^*$ to obtain the control action u(k+j), j=1, …N_s. Consequently, the proposed framework for the advanced-multistep NMPC strategy, with clipping to ensure $u(k+j) \in \mathbb{U}$, can be summarized as follows.

In background, between t_k and t_{k+N_s}:

1. Use (z_l, v_l) from the previous NMPC solution to predict the future state through $\bar{x}(k+j) = f(x(k+j-1), u(k+j-1))$, j=1, …, N_s. Set $(p_0=)z_0=\bar{x}(k+N_s)$ and solve the *predicted* form of Problem (13).

2. At the solution $s^*(p_0)$, retain factors of $\mathbf{K}^*(p_0)$.

On-line, at t_{k+j} j=1, …, N_s:

1. Obtain the state estimate x(k+j) and set p=x(k+j). For $N_s=1$, compute the sensitivity step $\Delta s(p)$ from (23), or from a Schur-complement extension to (23) for $N_s>1$ (see [38]). Extract the appropriate element of $\Delta s(p)$, called Δv_0 which corresponds to the perturbed control variable at $l=0$.

(a) Calculate $\tilde{v}_{j-1} = v_{j-1}^* + \Delta v_{j-1}$: Note that elements of v_{j-1}^* at strongly active bounds have zero perturbations.

(b) If $\tilde{v}_{j-1} \in \mathbb{U}$, set $\tau=1$.

(c) Else, find the largest value of $\tau \in [0, 1]$ such that u(k+j) $(:= v_{j-1}^* + \tau \Delta v_{j-1}) \in \mathbb{U}$.

2. Set $u(k+j) = v_{j-1}^* + \tau \Delta v_{j-1} \in \mathbb{U}$, and return to the background phase with $k := k+N_s$.

Note that in the nominal case, where $f_d(x, w)=0$ and $\Delta s(p)=0$, both asNMPC and iNMPC produce identical control actions, e.g., $u^{id}(k+1) = u^{as}(k+1)$. More details on advanced step NMPC can be found in [39]. Moreover, stability and robustness properties of the advanced step strategies have been analyzed in [22] for $N_s=1$ and in [38] for $N_s>1$; these are summarized next.

Stability Properties of Advanced Step NMPC

Building on the robust stability properties of Ideal NMPC, the robustness properties of the asNMPC controller also need to consider the effect of the approximation due to NLP sensitivity. The forward simulation $\bar{x}(k+1)=f(x(k), u(k))$ predicts the future state at t_{k+1} but the plant will evolve with uncertain dynamics generating $x(k+1)$. Moreover, we need to distinguish between iNMPC using $u^{id}(k)$ and asNMPC, which generates $u^{as}(k)=\kappa^{as}(x(k))$. To interpret this difference we consider the related problem:

$$\begin{aligned}\hat{J}(x(k), \hat{u}(k)) :&= \min_{z_l, v_l} \Psi(z_N)+\psi(x(k), \hat{u}(k))+\sum_{l=0}^{N-1}\psi(z_l, v_l)\\ \text{s.t. } z_{l+1}&=f(z_l, v_l) \quad l=0, \ldots N-1 \\ z_0&=f(x(k), \hat{u}(k)),\ v_l\in\mathbb{U}\end{aligned} \tag{26}$$

This problem has an equivalent solution to Problem (4) and we consider $\hat{J}(x, \hat{u})$ as our candidate ISS Lyapunov function. We define $J^{as}(x(k)) :=\hat{J}(x(k), u^{as}(k))$, $J^{id}(x(k)) :=\hat{J}(x(k), u^{id}(k))$, and also $J^{id}(\bar{x}(k)) :=\hat{J}(\bar{x}(k), \bar{u}^{id}(k))$, where $u^{id}(k)$ and $\bar{u}^{id}(k)$ are determined as $\hat{u}(k)\in\mathbb{U}$ in (26). For the next time step, we define the following residuals as:

$$\varepsilon_s(x(k+1)) :=J^{id}(x(k+1))-J^{id}(\bar{x}(k+1)) \tag{27a}$$

$$\varepsilon_{as}(x(k+1)) :=J^{as}(x(k+1))-J^{id}(x(k+1)) \tag{27b}$$

where ε_s accounts for the model mismatch while ε_{as} accounts for approximation errors introduced by NLP sensitivity. From (25) and Theorem 1 we have positive Lipschitz constants L_J, L_u, L_{sv} and L_{sq} such that $\forall x\in\mathbb{X}$,

$$\begin{aligned}&\varepsilon_s(x(k+1))\le L_J|x(k+1)-\bar{x}(k+1)|\le L_J|f_d(x(k), w(k))|\\ &\varepsilon_{as}(x(k+1))\le L_u(|u^{as}(k+1)-u^{id}(k+1)|)\\ &\qquad\le L_J((1-\tau)L_{sv}+L_{sq}|f_d(x(k), w(k))|)\cdot|f_d(x(k), w(k))|\end{aligned}$$

Note that for $\tau=1$ (i.e., without clipping) ε_{as} is only $O(|f_d|^2)$.

By comparing the successive costs $J^{as}(x(k))$ and $J^{as}(x(k+1))$, we arrive at the ISS property shown in [22] for $N_s=1$ and in [38] for $N_s>1$.

Theorem 6 *(Robust Stability of asNMPC) Given Assumptions 2 and 4 with $\mathcal{K}$ functions $\alpha_0(|x|)$ and $\alpha_p(|x|)$ that satisfy $\alpha_0(|x|)\le\eta\alpha_p(|x|)/(L_J L_{as})$ where $L_{as}=(1+L_{sv}+L_{sq}(\alpha_0(|x|)+2L_d w_{max})$, $|w(k)|\le w_{max}$ and $\eta\in(0, 1)$. Then the cost function $J^{as}(x)$ obtained from the solution of (26) with $\hat{u}=u^{as}$ is an ISS-Lyapunov function and the resulting closed-loop system is ISS stable.*

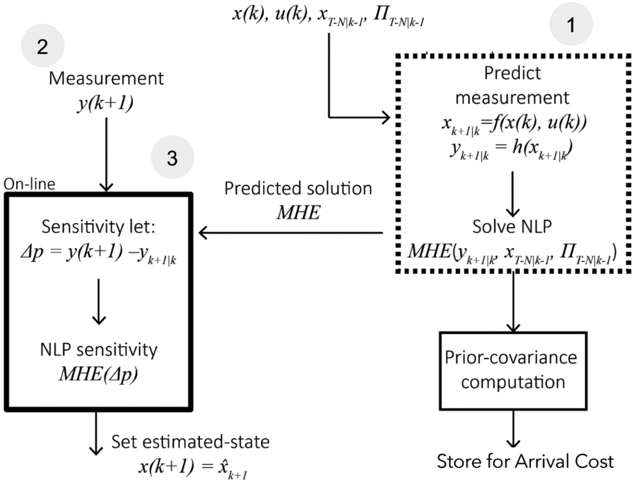


Fig. 5. Sketch of advanced step MHE for $N_s=1$ [1]. Having (x(k), u(k) predict $y_{k+1|k}$ and solve Problem (6) in background [2]. Wait until y(k+1) arrives [3]. Apply a fast correction to $x_{k+1|k}$ from (6) using the sensitivity update (23) to determine the estimated state x(k+1). Set k : =k+1 and repeat.

Advanced Step Features for MHE

As with NMPC, MHE can also be reformulated using the sensitivity computations in (23). As shown in Fig. 5, future output measurements are predicted from the DAE model, and the MHE problem (6) is solved in background within N_s steps. Once the actual measurements are obtained, sensitivity updates to the background MHE solution provide new state estimates that are used for NMPC. Detailed formulations of advanced step MHE are presented in [13] for $N_s=1$ and in [40] for $N_s>1$. A demonstration of advanced step MHE is provided in Section 6.1.

Moreover, the MHE formulation (6) can be modified to deal with gross errors and outliers in the output measurements. In this case the quadratic stage costs in (6) are modified by using the method of M-estimators (e.g., Huber or Hampl estimators) that put less weight on measurements with high residuals. Such an approach was developed, analyzed and demonstrated in [41] and leads to less biased estimates.

Terminal Conditions

As seen in the Section 3, terminal costs $\Psi(z_N)$ and terminal regions $\mathbb{X}_f$ are essential components to guarantee the asymptotic and ISS stability for NMPC. In the terminal region we assume there exists a stabilizing controller that takes z_N to zero as $l\rightarrow\infty$, with a terminal cost bounded by $\Psi(z_N)$. A descent property of the terminal controller is stated in Assumption 2 and directly applied in Theorems 3, 5 and 6. As shown in [4,5], a sufficiently long horizon N and sufficiently large terminal cost in problem (4) ensure that z_N will lie in the terminal region. On the other hand, finding these quantities is case dependent, may require considerable off-line tuning and can lead to higher computational costs for the controller.

Over the past decade, a number of strategies [6,42-44] have been developed to determine suitable $\mathbb{X}_f$ and $\Psi(x)$. Typical calculation approaches derive a controller, $u=-Kx$, such as an LQG regulator, for the linearized process model at the setpoint $z=0$ and a terminal cost $\Psi(x)=x^T(P)x$ can be calculated from the Ricatti equation. The error due to this linearization is given by:

$$\zeta(x)=|f(x, -Kx)-Ax+BKx|$$

and a terminal region $\mathbb{X}_f$ is determined where all $x\in\mathbb{X}_f$ satisfy the descent property:

$$f(x, -Kx)^T(P)\,f(x, -Kx)-x^T(P)x\le-\varepsilon|x|^2.$$

Using a suitable error bound $\zeta(x)$, $\mathbb{X}_f$ can be found through global solution of a nonconvex optimization problem. In [7,8], $\zeta(x)$ is given by $M|x|^q$, with fitted parameters M, q>0, which leads to a constructive terminal region, $x\in\{|x|\in c_f\}\subseteq\mathbb{X}_f$ with c_f calculated in closed form.

This terminal region allows the introduction of adaptive horizons that can expand or contract based on behavior of Lyapunov functions calculated from the NMPC controller. Here the adaptive horizon is estimated at the next time step by using advanced step NMPC predictions along with a safety margin. As shown in [7], this adaptive horizon approach is able to achieve significant reduction in average computation time, with the same performance as fixed horizon NMPC.

Economic NMPC

In most NMPC applications the stage costs are tracking functions that measure deviations of the state variables from a desired setpoint or trajectory. In Assumption 2, these costs are bounded by $\mathcal{K}$ functions to ensure asymptotic and ISS stability.

While the extension of the NMPC controller to economic stage costs (based on profit, operating or some other performance measure) is straightforward, these economic stage costs are generally not $\mathcal{K}$ functions and therefore do not satisfy the stability assumptions. As a result, advanced formulations of the NLP subproblem are required to realize stability and robustness properties. As shown in [45], a dissipativity property can be used to establish the stability for eNMPC. However, it is generally difficult to characterize dissipative systems in the context of Problem (4). Instead, sufficient conditions for dissipativity include NLP reformulations with so-called rotated stage costs that need to be strongly dual [46] or strongly convex [31]. Most economic stage costs do not satisfy these conditions, although they can be satisfied with regularized stage costs, e.g.,

$$\psi^{mod}(z_l, v_l) = \psi^{ec}(z_l, v_l) + \rho\psi^{tr}(z_l, v_l) \tag{28}$$

where $\psi^{tr}(z_l, v_l) = \|z_l - z^*\|_Q^2 + \|v_l - v^*\|_R^2$ is the tracking stage cost and (z^*, v^*) is the steady state optimum for the economic stage cost. Regularization strategies have been proposed in [31,47,48]. These studies develop approaches that ensure that reformulated stage costs ψ^{mod} are $\mathcal{K}$ functions, and that both nominal and robust stability can be proved. However, these approaches (called eNMPC-reg) may be conservative, especially for large-scale problems. An alternate eNMPC strategy (called eNMPC-sc) was recently developed based on direct incorporation of *stability constraints* based on tracking functions [49], i.e.,

$$J^{tr}(x(k)) - J^{tr}(x(k-1)) \leq -\delta\psi^{tr}(x(k-1), u(k-1)) \tag{29}$$

where the constant $\delta \in (0, 1]$ and

$$J^{tr}(x(t)) = \Psi^{tr}(z_N) + \sum_{l=0}^{N-1} \psi^{tr}(z_l, v_l).$$

A case study that compares these approaches is presented in Section 6.2.

Robust, Multistage NMPC

ISS holds for the soft constrained NLP (13), but to ensure satisfaction of hard constraints, the optimization formulation should be extended to incorporate uncertainty directly within the NLP framework. For this, we consider a slight modification to the dynamic system (3):

$$x(k+1) = \hat{f}(x(k), u(k), d(k)) \tag{30}$$

where $d(k) \in \mathbb{R}^{n_d}$ represents the plant uncertainty within the continuous uncertainty set $\mathbb{D}$, and can include w(k), model uncertainty and other input disturbances. Discretization of $d \in \mathbb{D}$ is typically done of the form $(d_{(i),l}^{max}, d_{(i),l}^{nom}, d_{(i),l}^{min})$ for the maximum, nominal and minimum values of each element (i) in vector d at stage *l*.

This discretization facilitates a multi-model extension of (4) that provides worst case performance measures [50-52]. In this approach, a single optimal control profile is determined while the constraints are imposed over N_M model scenarios, each corresponding to a different realization of the uncertain parameters. This leads to the following multimodel open-loop optimal control problem (*mmOCP*) formulation:

$$\begin{aligned}
\min_{\substack{x_l^j,\\ j=1,\ldots,N_M,\, l=0,\ldots,N\\ u_l,\, l=0,\ldots,N-1}} \quad \max_{j=1,\ldots,N_M} \; & \phi(x_N^j) \qquad\qquad (31)\\
\text{s.t. } & x_{l+1}^j = f(x_l^j, u_l, d_l^j),\\
& \forall j \in \{1, \ldots, N_M\}, \forall l \in \{0, \ldots, N-1\}\\
& x_0^j = x(0), \forall j \in \{1, \ldots, N_M\}\\
& g_i(x_l^j) \leq 0,\\
& \forall j \in \{1, \ldots, N_M\}, \forall l \in \{0, \ldots, N\}, \forall i \in I_l\\
& u_l \in U, \forall l \in \{0, \ldots, N-1\}.
\end{aligned}$$

For this worst-case realization each of the N_M scenarios is selected at an extreme vertex d_l^j of the uncertain parameter (along with any other critical values that correspond to the worst case). Also, the nominal open-loop optimal control problem can be obtained as a special case of (31) with $N_M=1$ with $d^1=d^{nom}$. Problem (31) can also be solved in a moving horizon framework as an extension to problem (4). However, by capturing the uncertainties and maintaining feasibility with a single, common control profile, this multimodel formulation may lead to very conservative performance.

Less conservative multi-stage NMPC formulations have been proposed, based on concepts of stochastic programming [53-55]. While these formulations are more expensive to solve, they ensure constraint satisfaction and lead to significantly better performance. Discretizing each element $d_{(i),l}$ with maximum, nominal and minimum values leads to the scenario tree in Fig. 6, with separate controls for each branch. The resulting optimization problem at t_k with the current state x_k is formulated as follows.

$$\min_{z_l^c, v_l^c} \sum_{c \in \mathbb{C}} \omega^c \left(\phi(z_N^c, d_{N-1}^c) + \sum_{l=0}^{N-1} \varphi(z_l^c, v_l^c, d_l^c) \right) \tag{32a}$$

$$\text{s.t.} \quad z_{l+1}^c = f(z_l^c, v_l^c, d_l^c) \quad l = 0, \ldots, N-1 \tag{32b}$$

$$z_0^c = x_k \tag{32c}$$

$$v_l^c = v_l^{c'} \quad \{(c, c')|z_l^c = z_l^{c'}\} \tag{32d}$$

$$z_l^c \in \mathbb{X}, v_l^c \in \mathbb{U}, z_N^c \in \mathbb{X}_f, d_l^c \in \mathbb{D} \tag{32e}$$

$$\forall c, c' \in \mathbb{C} \tag{32f}$$

where $\mathbb{C}$ is the set of all scenarios, ω_c is the weight for each scenario, and z_l^c, v_l^c, d_l^c represent the vectors of state variables, control variables and uncertain parameters at stage *l* and scenario c. The objective function in (32a) is the weighted sum of stage costs across all the scenarios. Eq. (32d) represents non-anticipativity constraints (NACs), where the controls corresponding to the same state parent node must be equal. This follows because a control based on d_l cannot be determined before d_l is realized. The last equation (32f) defines a robust horizon N_r<N which requires the uncertain parameters to remain constant after the robust horizon. This limits excessive branching of the scenario tree and makes the solution of problem (32) tractable.

While multi-stage NMPC based on (32) leads to high performance control under uncertainty, efficient optimization strategies are needed to solve these problems in real time. In recent studies advanced step [54] and scenario generation and pruning concepts [56-58] have been developed for multi-stage NMPC that lead to faster solution strategies and consideration of larger dynamic sys-

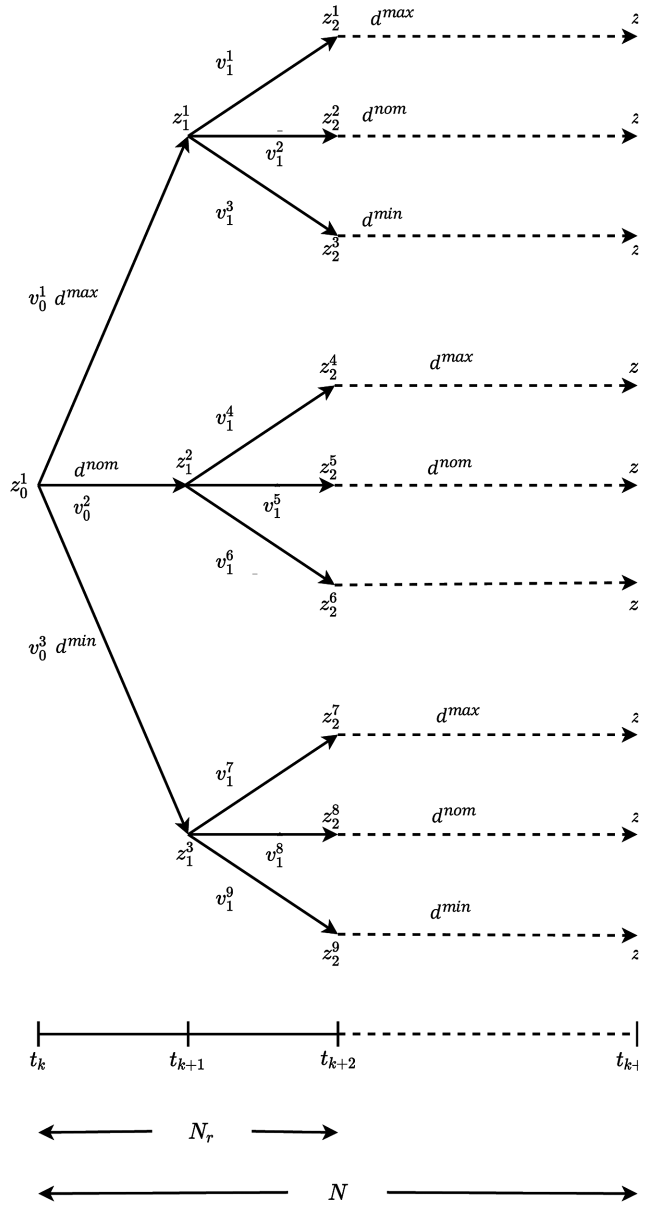


Fig. 6. Reduced scenario tree with n_d=1, N=3 and N_r=2.

tems. A case study that illustrates this approach is presented in Section 6.3.

CAPRESE: OPTIMIZATION MODELING PLATFORM FOR SENSITIVITY-BASED NMPC

New modeling and optimization tools have created an opportunity for the creation of advanced frameworks for problems in design, control and operations. These optimization platforms incorporate the capability of high-level programming languages instead of application-specific or proprietary languages like MATLAB, GAMS, AMPL or AIMMS. Interfacing these platforms with state-of-art solvers also makes it easier to deal with more complex problems. Of particular interest are optimal control problems (OCPs) that incorporate first-principles process models as part of a mathematical optimization problem. Within the optimization algorithm, these DAE models can be handled directly by means of multiple-shooting or orthogonal collocation.

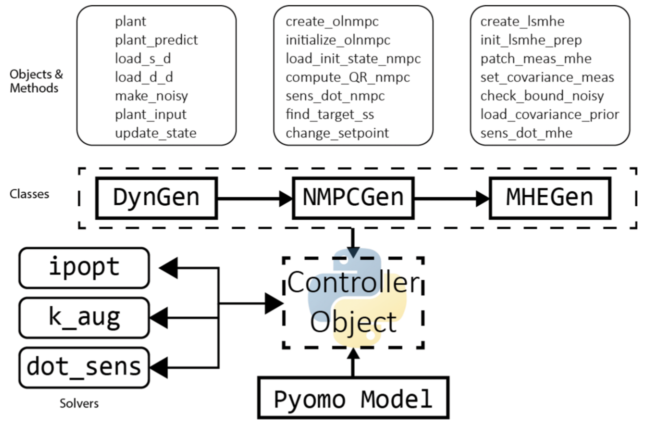


Fig. 7. The CAPRESE Pyomo Framework for NMPC and MHE.

Several frameworks have been developed for model-based optimal control, including ACADO [59], CasADi [60], APMonitor [61], JuMP [62] and Pyomo. In Pyomo [63] the dynamic optimization environment requires a combination of accessible modelling interfaces to solution algorithms. Rather than creating a new programming language, Pyomo provides the necessary optimization objects in Python, which promote the flexibility of applications for optimization-based algorithms for dynamic systems. Here, we describe a new optimization modeling framework that includes NMPC for the control computation and MHE for state estimation. Both of these are enhanced for real time nonlinear control and state estimation through NLP sensitivity.

Fig. 7 presents the framework entitled Control and Adaptation with PREdictive SEnsitivity (CAPRESE), which facilitates the automatic construction of NMPC and MHE problems with given dynamic models. CAPRESE also incorporates algorithmic management issues such as initialization, diagnostics, post-optimality analysis and sensitivity computations. This open framework is straightforward to extend, though some applications might require tailored algorithmic considerations.

The CAPRESE framework for NMPC/MHE is displayed in Fig. 3 and comprises three main classes. The base class DynGen incorporates the dynamic model discretized in time using Pyomo.DAE [64] and provides simple plant simulations, predictions and data management. This class is inherited by NMPCGen, which is subsequently inherited by the MHEGen class. These classes are responsible for the formulation and solution of problems (4) and (6), respectively, and they are equipped with relevant methods for the underlying optimization problem (e.g. compute arrival cost for MHE or find target states for NMPC). Communication between models inside the framework leads to a closed loop between plant, estimator and controller. In addition, the user declares the model, states, controls, setpoints, and corresponding bounds.

In CAPRESE, sensitivity computations for NMPC and MHE based on (23) have been modified to further economize off-line and on-line computation and provide greater functionality. In previous work, sIPOPT [65] was used to compute the sensitivity information, but it offers only limited capability for structured KKT systems. To overcome these limitations, k_aug was created to compute sensitivity matrices from the solution of (23). k_aug incorporates various sparse linear algebra libraries and computes sensitivities from the KKT conditions (16) through an "adjoint" formulation of (23), where the only on-line sensitivity computations are simple matrix-vector products. The on-line component is shown in Fig. 3 as dot_sens, where sensitivity-based updates of the control variables are performed, using off-line information from the sensitivity matrix.

More information on the capabilities and structure of CAPRESE can be found in [66].

NMPC CASE STUDIES

The NMPC and MHE advances discussed above are illustrated in this section with challenging nonlinear applications of control and dynamic optimization. The past two decades have seen widespread applications of NMPC on important and challenging, real-world processes. Here we concentrate on three case studies: a demonstration of asNMPC and asMHE with CAPRESE for a distillation unit, economic NMPC for a distillation system, and multi-stage NMPC for a semi-batch polymer reaction system.

1. Advanced Step NMPC with CAPRESE

For the CAPRESE case study, the distillation column shown in Fig. 8 is modeled with dynamic MESH (mass, equilibrium, summation, heat) equations as presented in [67]. The DAE model considers the separation of a binary mixture of methanol and n-propanol, with tray-by-tray mass and energy balances, Raoult's Law for thermodynamic equilibrium, and tray liquid flows determined by the Francis weir formula. For a column with 40 trays, the DAE model has 84 differential equations. Temperatures are measured on each tray, with temperature setpoints on trays 14 and 28. The control variables are the reboiler heat duty and the reflux ratio. Thus, n_u=2, n_x=84. n_y=40.

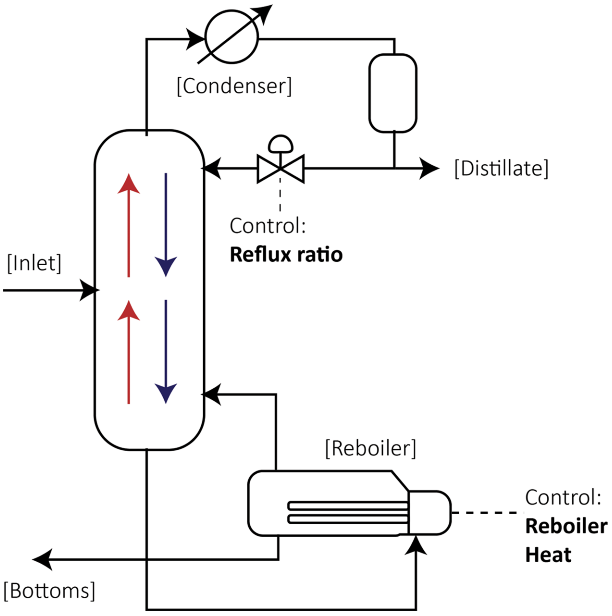


Fig. 8. Case study 1: Distillation column schematic.

Table 1. Distillation case study in CAPRESE using ideal and advanced step NMPC/MHE. average timings for the optimization and sensitivity in CPU seconds, using IPOPT 3.12 and intel i7-6700 CPUs. asNMPC online computation is with *dot sens*

CAPRESE Steps	iNMPC/iMHE	asNMPC/asMHE
NLP (IPOPT)	11.0/8.50	11.0/8.50
Red. Hess. (k_aug)	-/3.54	-/3.54
Sens.Matrix	-/-	1.87/2.58
Online Comp.	**11.0/12.04**	**0.10/0.49**

The continuous time DAE model (1) is transformed into a discrete time model (3) using 3 point Radau collocation. MHE and NMPC were run with the advanced step and ideal strategies (assuming no computational delay), with ten-step horizons (N=$\mathcal{N}$=10) and 60s sampling times. The resulting NLP problems have 20672 variables and 19792 equations in the case of MHE (6), and 19036 variables and 19016 equations for NMPC (4). Also, random noise was introduced on all plant states. The results for this case study are shown in Fig. 9.

For state estimation the maximum relative error considers all the states of the problem. It can be seen, that these errors remain small for most of the time. However, at the setpoint change at t_k=350, the relative error grows very quickly. Nevertheless, advanced-step MHE follows similar behavior to ideal MHE with only a small difference in performance. Ideal NMPC provides reasonably good performance, even under the presence of noise, and advanced-step NMPC follows the ideal case very closely. The average timings for this problem are displayed in Table 1. Because the on-line computations require much less time than offline, the total computational delay for both asMHE and asNMPC amounts to only 0.59 CPU seconds, 39 times less than the 23.04 CPU seconds required for the ideal strategies. More information on this case study can be found in [66].

2. Economic NMPC for Distillation Systems

For the second case study we consider the dynamic process shown in Fig. 10 with two distillation columns in series [68] with three chemical components, A, B, C. The bottom product of the first column is fed to the second column, and the distillation flowsheet is shown in Fig. 10. The distillate of the first column has 95 mol% A, the distillate of the second has 95 mol% B, and the bottom of the second column has 95 mol% C. The vapor-liquid equilibria are modeled using constant relative volatility, and Francis weir formula is used to model the tray hydraulics. In addition, dynamic component and total mass balances are written for each tray as well as the condenser and reboiler. More details of the model can be found in [49,68]. Each column has 41 equilibrium stages including the reboiler, leading to 246 states and 8 controls.

The objective is to minimize the stage costs for feed, and energy to the reboilers, minus the cost of the products, i.e.,

$$\psi^{ec}=p_F\cdot F_1+p_V(V_{\bar{B}1}+V_{\bar{B}2})-p_{D1}\cdot D_1-p_2\cdot D_2-p_{B2}\cdot B_2$$

where p_F=p_{D1}=p_{B2}=1, p_{D2}=2, p_V=0.008\$/mol are the respective prices of feed, D_1, D_2, B_2 and reboiled vapor in the two columns. The composition of the feed is 40 mol% A, 20 mol% B, and 40 mol%

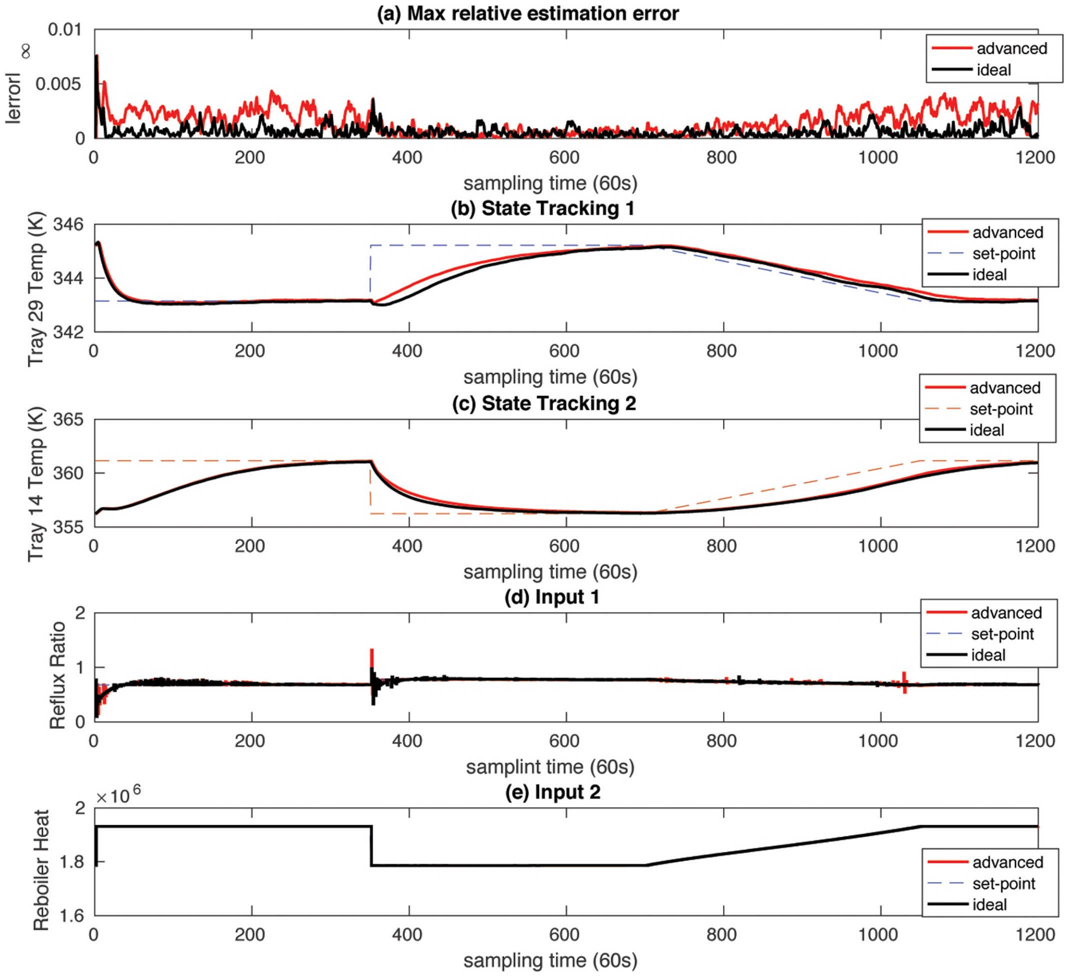


Fig. 9. Case study 1: Performance profiles for MHE (a) followed by two outputs (b), (c) and two inputs (d), (e) for NMPC.

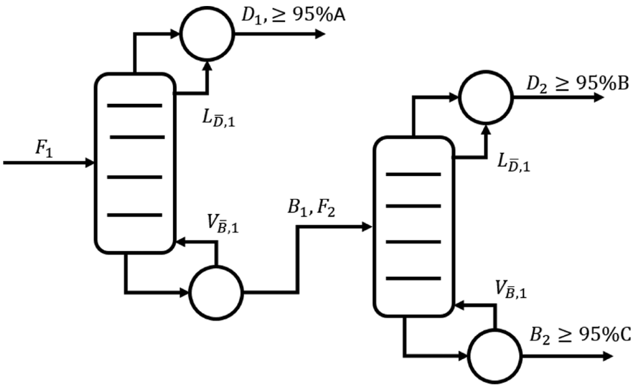


Fig. 10. Case study 2: Distillation flowsheet.

C. The product purity is implemented as an inequality constraint. The DAE system is discretized using three point Radau collocation with a finite element length of 1 min and N=25. The NLP (13) for eNMPC has 120,000 variables, 108,000 equality constraints, and 14,000 inequality constraints. The models are implemented in AMPL and solved with IPOPT.

Finding regularization weights ρ for (28) that ensure positive definite Hessians of ψ^{mod} over a high-dimensional state space is cumbersome, and for this we apply Gershgorin's theorem to estimate ρ (See [69] for more detail on this calculation and evaluation.).

We compare solution times for the specific cases of eNMPC-sc where (29) is implemented with δ=0.01, and for eNMPC-reg (28) with 100% of the Gershgorin weight ρ, along with smaller percentages. Solution of the eNMPC-sc problems requires an average of 271 CPU seconds and 188 IPOPT iterations, while eNMPC-reg averages only 83 seconds and 70 iterations. Although we see that regularization allows faster computation, Table 2 shows that it also leads to more conservative economic performance.

Table 2 presents a comparison of accumulated stage costs ($\Sigma_{k=0}^{9}$ $\psi^{ec}(x_k, u_k)- \psi^{ec}_{ss}$) from the same initial condition. This metric shows the improvement obtained over tracking the optimal setpoint.

From the nominal results, it is apparent that eNMPC-sc provides better performance over the regularized formulation eNMPC-reg, and it also performs better than economic stage costs alone, for which stability cannot be guaranteed. Also, note that reducing the regularization percentage does little to improve performance.

In Table 2 we also consider cases with additive noise w_k in the feed rate and composition. Here w_k is sampled from a normal distribution with standard deviations of 0.1 for feed rate and 0.01 for

Table 2. Distillation example with objective improvements given by $\sum_{k=0}^{9}(\psi^{ec}(\mathbf{x}_k, \mathbf{u}_k) - \psi^{ec}_{ss})$

Case	nominal	w/ noise
Tracking	−20.903	−20.736
eNMPC-reg 100%	−22.665	−21.662
50%	−22.676	−21.511
25%	−22.658	−21.361
eNMPC-sc, δ=0.99	−25.876	−24.283
0.9	−28.933	−24.162
0.5	−29.701	−24.234
0.1	−29.453	−25.656
0.01	−29.693	−24.039
Economic	−27.081	−24.479

A and B mole fractions. As shown in Table 2, eNMPC-sc using (29) shows significantly improved economic performance over eNMPC-reg (28), with and without noise. Moreover, performance with economic stage costs alone is decreased in the presence of noise. More information on this case study can be found in [49].

On-line Optimization of Semi-batch Process

The third case study deals with NMPC formulations for batch-type processes that operate over a finite time period. In this case the NMPC formulation is implemented over a shrinking horizon. For these processes, the presence of uncertainty, from input disturbances and model inaccuracies, often requires careful attention. This is particularly true when meeting process constraints that deal with equipment limitations, product specifications and safe operation. For these issues, we explore the application of robust NMPC formulations mmOCP (31) and msNMPC (32).

This case study considers an anionic homopolymerization process operated in semi-batch mode. Polymerization of propylene oxide (PO) to polypropylene glycol (PPG) is challenging to control. The reaction is highly exothermic and needs to satisfy strict safety constraints, along with stringent product specifications. Moreover, the model has several uncertain parameters that can impact safe and high performance operation. Fig. 11 presents the process unit as well as its underlying reaction mechanism. The employed process model is obtained from a first-principles model derived in [70]. A detailed model of this process is presented in [57]; additional modeling and optimization studies include [52,55].

The NMPC controller has an economic objective function to minimize the final batch time. Operational constraints include time dependent temperature constraints (T∈[373 K, 423 K]) and adiabatic temperature rise (T_{ad}≤443 K), along with endpoint constraints on the product quality indicators: number average molecular weight (NAMW), unreacted monomer conversion and fraction of unsaturated byproduct.

As seen in Fig. 11, the process is controlled by monomer feedrate and the cooling jacket temperature. Since these enter linearly in (4), optimal control of this system can give rise to singular control problems, which may lead to undesirable oscillatory control profiles [2]. We avoid this behavior by regularizing the control displacement and writing the stage cost as follows:

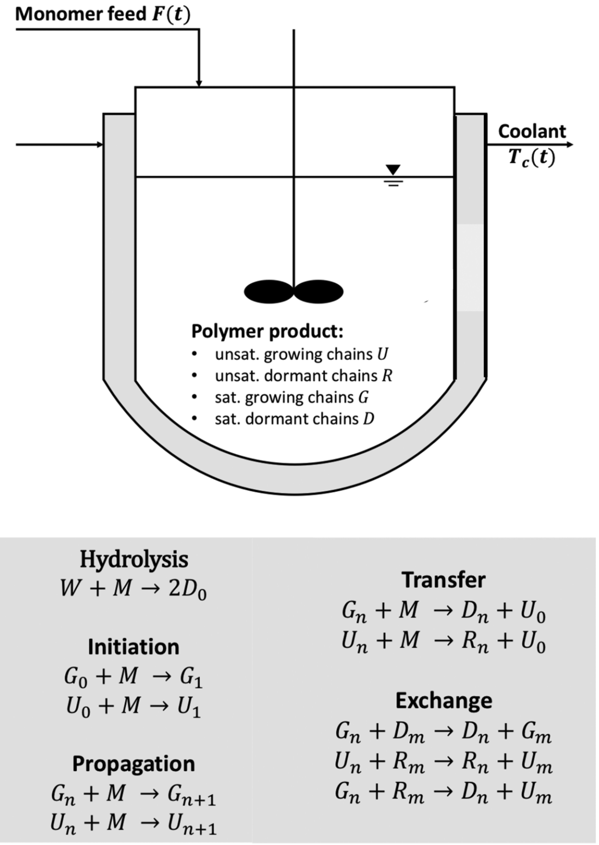

Monomer feed $F(t)$
Coolant
$T_c(t)$
Polymer product:
• unsat. growing chains U
• unsat. dormant chains R
• sat. growing chains G
• sat. dormant chains D
Hydrolysis
$W + M \rightarrow 2D_0$
Initiation
$G_0 + M \rightarrow G_1$
$U_0 + M \rightarrow U_1$
Propagation
$G_n + M \rightarrow G_{n+1}$
$U_n + M \rightarrow U_{n+1}$
Transfer
$G_n + M \rightarrow D_n + U_0$
$U_n + M \rightarrow R_n + U_0$
Exchange
$G_n + D_m \rightarrow D_n + G_m$
$U_n + R_m \rightarrow R_n + U_m$
$G_n + R_m \rightarrow D_n + U_m$

Fig. 11. Case study 3: Schematic of the PO polymerization process and summary of the underlying reaction mechanisms.

$$\psi_l^{mod,j} := \Delta t_l^j + (u_l^j - u_{l-1})^T (u_l^j - u_{l-1}). \tag{33}$$

This stabilization has negligible influence on performance, and by summing Δt_i^j in each scenario the remaining batch time is minimized at every time step in the NMPC scheme.

The uncertainty vector d consists of three elements: A_p, the pre-exponential factor for the propagation reaction rate, A_i, the pre-exponential factor for the initiation reaction rate, and U, the jacket heat transfer coefficient, which can vary due to fouling. All three parameters have strong influence on model predictions and the process constraints. Also, the process state is inferred from incomplete and noisy state measurements.

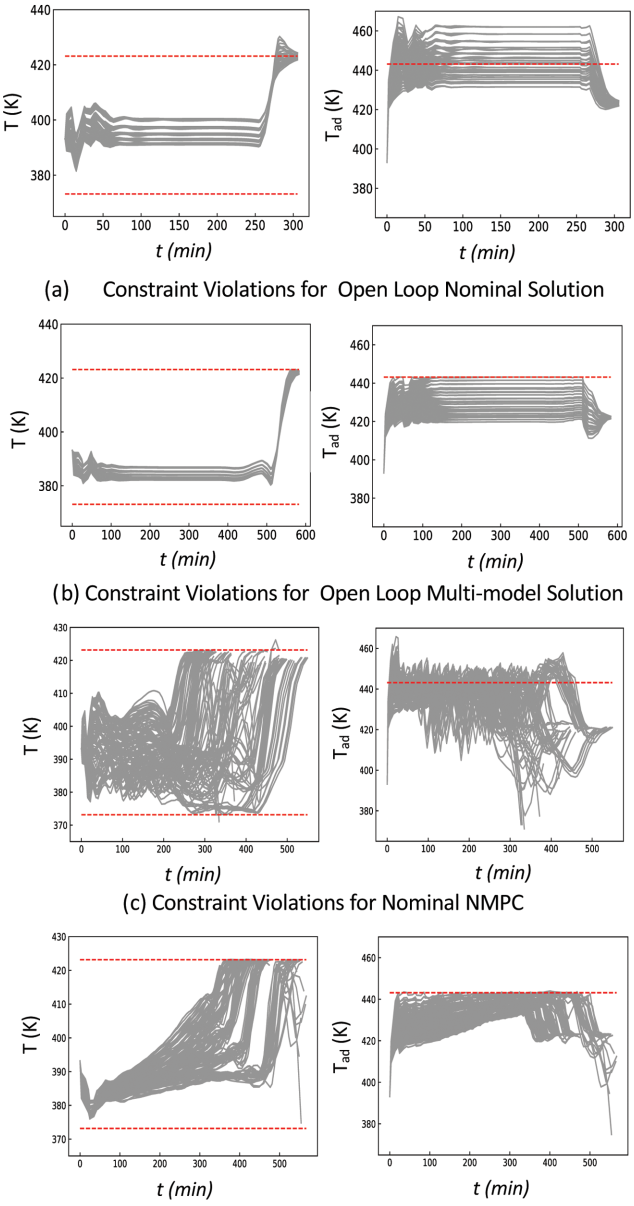
Moreover, the time intervals Δt_l^j in the stage costs are degrees of freedom in the optimization, with non-anticipativity constraints imposed in the multistage problem (32) along with a requirement that Δt_l^j decrease monotonically as l increases. This avoids overly optimistic choices in early phases of the process. The number of control time intervals is chosen as N=24, which allows a reasonable sampling time and sufficiently accurate discretization.

The dynamic optimization problems, mmOCP (31) and msNMPC (32), are solved using a simultaneous collocation approach, where the underlying differential-algebraic process model is discretized with three-point Radau collocation on finite elements, and solved using IPOPT 3.12 [33]. Parametric sensitivity for adaptation of the uncertainty set is extracted from the KKT-system at the optimal solution of the corresponding NLPs using k_aug [66]. The overall framework for the shrinking horizon NMPC is embedded in Python using Pyomo as the modeling language.

Dynamic Optimization Results

Both open-loop and NMPC control approaches are implemented and compared using systematic sampling from the uncertainty set $d \in \mathbb{D}$. The evaluation of the closed-loop system simulations is performed for all parameter realizations on the grid of $\{-20\%, -10\%, 0\%, 10\%, 20\%\}^3$ deviating from the nominal parameter values.

We apply two open-loop control strategies for trajectory optimization. The first determines economically optimal control profiles for the nominal process model. The second employs the multi-model approach determined by the *mmOCP* problem.



(a) Constraint Violations for Open Loop Nominal Solution

(b) Constraint Violations for Open Loop Multi-model Solution

(c) Constraint Violations for Nominal NMPC

(d) Constraint Violations for Multi-stage NMPC

Fig. 12. Temperature T and adiabatic temperature rise T_{ad} profiles for all parameter realizations from the sampled grid compared to the path constraint levels (dashed red lines).

Figs. 12(a), (b) show the open-loop profiles for temperature and adiabatic temperature rise, obtained by exhaustive sampling of the uncertainty set. For the nominal solution (a) the results show large violation of path and endpoint constraints for several parameter realizations. In contrast, the multi-model approach (b) yields robust satisfaction of the constraints for all sampled parameter realizations.

Next, we compare the control performance of nominal NMPC and multi-stage NMPC, where we set $N_r=1$ in problem (32). The reactor temperature and adiabatic temperature rise profiles are shown in Figs. 12(c), (d). Nominal NMPC shows severe constraint violation for the adiabatic temperature rise, while multi-stage NMPC essentially removes these violations and substantially improves robustness.

While Fig. 12 exhibits constraint violations with the four approaches, these results can be further improved [57] if on-line state estimation (6) is coupled to the NMPC problem (4). Table 3 compares the objective functions for the four approaches. Without state estimation, the comparison shows that multi-stage NMPC and *mmOCP* essentially eliminate the constraint violations. However, *mmOCP* requires almost double the operating time. In contrast, multi-stage NMPC leads to improved operation by relying on feedback during the process. Compared to *mmOCP*, the average operating time is reduced by 23%.

Moreover, with state estimation, performance of **multistage NMPC** further improves, as shown in Table 3. Compared to non-robust **nominal NMPC**, **multi-stage NMPC** with state estimation requires an increase of only 7.6% in average operating time. As a result, multi-stage NMPC formulations provide robust NMPC solutions for challenging applications and significantly better performance than worst case approaches.

Finally, for nominal NMPC, 99% of all cases were solved in less than 1 CPU s on a laptop computer. For multi-stage NMPC, 99% of the NLP problems were solved in less than 10 CPU s, with the maximum timing at 13 CPU s. This is well below the sampling time (10 minutes) required for this polymerization process. Additional comparisons and computational details can be found in [57].

CONCLUSIONS AND FUTURE DIRECTIONS

Nonlinear model predictive control (NMPC) is an effective and efficient vehicle for on-line dynamic optimization, with significant benefits for challenging real-world problems in process engineering. This article reviews several important directions for the realization of large-scale process applications with NMPC.

Table 3. Maximum, average and minimum operating time (minutes) attained by different control approaches without (italic) and with (bold) on-line estimation

Control approach	t_f^{max}	t_f^{avg}	t_f^{min}
open-loop nominal		——— 305.5 ———	
open-loop multi-model		——— 582.8 ———	
nominal NMPC	549.2	379.9	277.2
multi-stage NMPC	566.5	449.9	387.0
nominal NMPC	531.6	360.7	246.4
multi-stage NMPC	549.3	388.0	272.6

First, we note that time critical solution of NMPC and its companion moving horizon estimation (MHE) problem requires efficient and reliable formulations and solution strategies. In particular, the formulation of NMPC problems must be robust to input data from the plant. The NLP formulation therefore needs to obtain solutions that satisfy suitable constraint qualifications and second-order conditions (at least MFCQ and GSSOSC). Here we show that these requirements can be ensured through regularized stage costs and relaxed state constraints. These are essential for nominal and ISS stability for NMPC.

Second, NMPC problems can be solved quickly by powerful solution strategies that use full discretization (based on orthogonal collocation on finite elements) and fast, large-scale NLP solvers. In particular, barrier NLP solvers provide fast solutions and allow sensitivity with respect to NLP solutions, at virtually no additional computational cost. Moreover, these properties allow the development of *advanced step* approaches for NMPC and MHE problems, where most of the NLP solver effort is performed in the background and very fast control updates are performed, once the plant state is available. As a result, computational delay is virtually eliminated.

The implementation of these advances and their realization in real-world applications requires optimization frameworks and platforms that support modeling of the NLP, seamless interaction with the NLP solver and management of the NMPC and MHE closed-loop tasks. An example framework is the CAPRESE system build on the Pyomo platform.

Finally, the advances described in this article are illustrated by three case studies. The first provides a demonstration of advanced step NMPC and MHE approaches on an 84-state distillation control problem, using the CAPRESE framework. On-line solution of both the NMPC and MHE problems require only 0.59 CPU s of on-line computation, 39 times less than with ideal NMPC/MHE. The second case study examines improved formulation of economic NMPC for a large distillation system. Here, both regularization and stability-constrained strategies are compared. The results show stable eNMPC problems can be formulated with significantly improved performance over tracking an optimal setpoint. The third case study deals with on-line dynamic optimization under uncertainty for a semi-batch polymerization process. Here, the improved performance of multi-stage NMPC is demonstrated over both nominal NMPC, which can suffer significant constraint violations, and worst-case NMPC, which can lead to very conservative performance.

While these case studies demonstrate the effectiveness of recent advances for NMPC, they also motivate a number of future research areas. In particular, there is a clear need for more powerful NLP solvers and platforms to tackle larger NMPC problems, including multistage NMPC. These developments will be aided by further application of decomposition strategies and high-performance computing. A number of open questions relating to stability and robustness also need further attention, especially for eNMPC and multistage NMPC. Finally, the impact of these NMPC advances also needs to be realized in the areas of decentralized and distributed control.

In summary, further advances in NMPC and exploration of exciting research questions will continue to engage the control community and lead to even more useful and effective strategies. Ultimately, these will make on-line dynamic optimization an indispensible tool for process applications.

REFERENCES

1. L. T. Biegler and V. M. Zavala, *Comput. Chem. Eng.*, **33**, 575 (2009).
2. L. T. Biegler, *Nonlinear programming: Concepts, algorithms, and applications to chemical processes*, SIAM, Philadelphia, PA (2010).
3. J. B. Rawlings, D. Q. Mayne and M. M. Diehl, *Model predictive control: Theory, computation and design*, Nob Hill Publishing, LLC. (2020).
4. G. Pannocchia, J. Rawlings and S. Wright, *Systems & Control Letters*, **60**, 747 (2011).
5. L. Grüne, *Automatica*, **49**, 725 (2013).
6. H. Chen and F. Allgöwer, *Automatica*, **34**, 1205 (1998).
7. D. W. Griffith, L. T. Biegler and S. C. Patwardhan, *J. Process Control*, **70**, 109 (2018).
8. C. Rajhans, D. W. Griffith, S. C. Patwardhan, L. T. Biegler and H. K. Pillai, *J. Process Control*, **83**, 30 (2019).
9. L. Magni and R. Scattolini, in *Assessment and future directions of nonlinear model predictive control*, R. Findeisen, F. Allgöwer, L. Biegler Eds., Springer, Berlin (2007).
10. A. Jazwinski, *Stochastic processess and filtering theory*, Dover Publications, Mineola, New York (2007).
11. L. Ji, J. B. Rawlings, W. Hu, A. Wynn and M. Diehl, *IEEE Transactions on Automatic Control*, **61**(11), 3509 (2016).
12. C. V. Rao, J. B. Rawlings and D. Q. Mayne, *IEEE transactions on Automatic Control*, **48**(2), 246 (2003).
13. V. Zavala, C. Laird and L. Biegler, *J. Process Control*, **18**, 876 (2008)
14. A. Wynn, M. Vukov and M. Diehl, *IEEE Transactions on Automatic Control*, **59**(8), 2215 (2014).
15. R. López-Negrete, S. C. Patwardhan and L. T. Biegler, in *Computer Aided Chem. Eng.: 10th Int. Symp. on Process Systems Eng.*, **27**, 1299 (2009).
16. R. López-Negrete, S. C. Patwardhan and L. T. Biegler, *J. Process Control*, **21**(6), 909 (2011).
17. U. M. Ascher and L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*, SIAM, Philadelphia (1998).
18. A. R. Conn, K. Scheinberg and L. N. Vicente, *Introduction to derivative-free optimization*, SIAM, Philadelphia, PA, USA (2009).
19. J. Nocedal and S. Wright, *Numerical optimization*, 2nd Ed., SpringerScience+BusinessMedia, LLC, New York (2006).
20. J. Betts, *Practical methods for optimal control using nonlinear programming*, SIAM, Philadelphia, PA (2001).
21. G. Grimm, M. Messina, S. Tuna and A. Teel, *Automatica*, **40**, 1729 (2004).
22. X. Yang, D. W. Griffith and L. T. Biegler, *Proc. 5th IFAC Conference on Nonlinear Model Predictive Control, IFAC-PapersOnLine*, **48**(23), 388 (2015).
23. R. Fletcher, *Practical methods of optimization*, Wiley, New York (1987)
24. S. M. Robinson, *Math. Oper. Res.*, **5**, 43 (1980).
25. A. Fiacco, *Introduction to sensitivity and stability analysis in nonlinear programming*, Academic Press, New York (1983).
26. J. Gauvin, *Mathematical Programming*, **12**(1), 136 (1977).

27. R. Janin, in *Sensitivity, stability and parametric analysis, mathematical programming studies*, vol. 21, A. Fiacco Ed., Springer Berlin Heidelberg (1984).
28. D. Ralph and S. Dempe, *Mathematical Programming*, **70**(1-3), 159 (1995).
29. M. Kojima, in *Analysis and computation of fixed points*, S. M. Robinson Ed., Academic Press, New York (1980).
30. V. Kungurtsev and J. Jäschke, *SIAM J. Optimization*, **27**(1), 538 (2017).
31. J. Jäschke, X. Yang and L. T. Biegler, *J. Process Control*, **24**, 1260 (2014).
32. A. Forsgren, P. Gill and M. Wright, *SIAM Rev.*, **44**(4), 525 (2002).
33. A. Wächter and L. T. Biegler, *Mathematical Programming*, **106**(1), 25 (2006).
34. S. Keerthi and E. Gilbert, *IEEE Trans. Auto. Cont.*, **57**, 265 (1988).
35. Z. Jiang and Y. Wang, *Automatica*, **37**, 857 (2001).
36. V. Zavala and M. Anitescu, *SIAM J. Control Optim.*, **48**, 5444 (2010).
37. M. Diehl, H. Bock and J. Schlöder, *SIAM J. Control Optimization*, **43**, 1714 (2005).
38. Y. Kim, D. M. Thierry and L. T. Biegler, *J. Process Control*, **96**, 82 (2020).
39. V. M. Zavala and L. T. Biegler, *Automatica*, **45**, 86 (2009).
40. Y. Kim, K. H. Lin, D. M. Thierry and L. T. Biegler, ADCHEM IFAC Conference to appear (2021).
41. B. L. Nicholson, R. Lopez-Negrete and L. T. Biegler, *Comp. Chem. Eng.*, **70**, 149 (2014).
42. S. Lucia, P. Rumschinski, A. J. Krener and R. Findeisen, *IFAC Papers Online*, **48**(23), 254 (2015).
43. M. Lazar and M. Tetteroo, *IFAC Papers Online*, **51**(20), 141 (2018).
44. C. Rajhans, S. Patwardhan and H. Pillai, *Proc. 12th IEEE Intl. Conf. Control and Automation*, 98 (2016).
45. D. Angeli, R. Amrit and J. Rawlings, *IEEE Trans. Auto. Cont.*, **57**(7), 1615 (2012).
46. M. Diehl, R. Amrit and J. B. Rawlings, *IEEE Trans. Auto. Cont.*, **56**(3), 703 (2011).
47. M. Z. Yu and L. T. Biegler, 10th IFAC International Symposium on Advanced Control of Chemical Processes (ADCHEM 2018), 103 (2018).
48. D. Krishnamoorthy, L. T. Biegler and J. Jaeschke, *J. Process Control*, **92**, 108 (2020).
49. D. W. Griffith, V. M. Zavala and L. T. Biegler, *J. Process Control*, **57**, 116 (2017).
50. B. Srinivasan, D. Bonvin, E. Visser and S. Palanki, *Comput. Chem. Eng.*, **27**(1), 27 (2003).
51. M. Diehl and J. Bjornberg, *IEEE Transactions on Automatic Control*, **49**(12), 2253 (2004).
52. T. Y. Jung, Y. Nie, J. H. Lee and L. T. Biegler, Proceedings 9th International Symposium on Advanced Control of Chemical Processes, IFAC ADCHEM, IFAC-PapersOnLine, **48**(8), 164 (2015).
53. S. Lucia, T. Finkler and S. Engell, *J. Process Control*, 23(9), 1306 (2013).
54. Z. Yu and L. T. Biegler, *J. Process Control*, **84**, 192 (2019).
55. H. Jang, J. H. Lee and L. T. Biegler, Proceedings of DYCOPS-CAB 2016, IFAC Papers Online, 37 (2016).
56. J. Puschke and A. Mitsos, *J. Process Control*, **69**, 6 (2018).
57. F. Holtorf, A. Mitsos and L. T. Biegler, *J. Process Control*, **80**, 167 (2019).
58. M. Thombre, Z. Yu, J. Jäschke and L. T. Biegler, *Comput. Chem. Eng.*, **148**, 107269 (2021).
59. B. Houska, H. J. Ferreau and M. Diehl, *Optimal Control Appl. Methods*, **32**, 298 (2011).
60. J. Andersson, J. Gillis, G. Horn, J. B. Rawlings and M. Diehl, *Mathematical Programming Computation*, **11**(1), 1 (2019).
61. J. D. Hedengren, R. A. Shishavan, K. M. Powell and T. F. Edgar, *Comput. Chem. Eng.*, **70**, 133 (2014).
62. I. Dunning, J. Huchette and M. Lubin, *SIAM Rev.*, **59**(2), 295 (2017).
63. W. Hart, C. Laird, J. P. Watson, D. Woodruff, G. Hackebeil, B. Nicholson and J. Siirola, *Pyomo â optimization modeling in python*, Springer, New York (2017).
64. B. L. Nicholson, J. D. Siirola, J. P. Watson, V. M. Zavala and L. T. Biegler, *Mathematical Programming Computation*, **10**, 187 (2018).
65. H. Pirnay, R. López-Negrete and L. T. Biegler, *Math. Programming Computation*, **4**, 307 (2012).
66. D. M. Thierry and L. T. Biegler, *AIChE J.*, **65**(7), 1 (2019).
67. R. Lopez-Negrete, F. J. DâAmato, L. T. Biegler and A. Kumar, *Comput. Chem. Eng.*, **51**, 55 (2013).
68. R. Leer, *Self-optimizing control structures for active constraint regions of a sequence of distillation columns*, Master's thesis, Norwegian University of Science and Technology (2012).
69. X. Yang, *Advanced-multi-step and economically oriented nonlinear model predictive control*, Ph.D. thesis, Carnegie Mellon University (2015).
70. Y. Nie, L. T. Biegler, C. M. Villa and J. Wassick, *AIChE J.*, **59**(7), 2515 (2013).

Lorenz T. (Larry) Biegler is the Covestro University Professor of Chemical Engineering at Carnegie Mellon University. His research interests lie in computer aided process engineering (CAPE) and include flowsheet optimization, optimization of systems of differential and algebraic equations, reactor network synthesis, nonlinear process control and real-time optimization. Contributions in these areas include analysis and development of nonlinear programming algorithms, optimization software design and application to real-world chemical processes and energy systems.

He is an author on over 500 archival publications and two textbooks, has edited 11 technical books and given numerous invited presentations at national and international conferences. His awards include the Lewis Award, Walker Award and Computers in Chemical Engineering Award, given by AIChE; the Lectureship Award, Curtis McGraw Research Award and CACHE Computing Award, given by ASEE; the INFORMS Computing Prize, and an honorary doctorate in engineering sciences from the Technical University of Berlin. He is a Fellow of AIChE, IFAC and SIAM, and a member of the National Academy of Engineering.