Nonlinear Programming Strategies for State Estimation and Model Predictive Control

Victor M. Zavala and Lorenz T. Biegler

Abstract. Sensitivity-based strategies for on-line moving horizon estimation (MHE) and nonlinear model predictive control (NMPC) are presented both from a stability and computational perspective. These strategies make use of full-space interior-point nonlinear programming (NLP) algorithms and NLP sensitivity concepts. In particular, NLP sensitivity allows us to partition the solution of the optimization problems into background and negligible on-line computations, thus avoiding the problem of computational delay even with large dynamic models. We demonstrate these developments through a distributed polymerization reactor model containing around 10,000 differential and algebraic equations (DAEs).

Keywords: large-scale, MHE, NMPC , nonlinear programming, sensitivity, interior-point methods, sparse linear algebra.

1 Introduction

General model-based control frameworks based on MHE and NMPC represent an attractive alternative for the operation of complex processes. These frameworks allow the incorporation of highly sophisticated dynamic process models and the direct handling of multivariable interactions and operational constraints. In addition, the potential of incorporating detailed first-principles models allows a closer interaction of the controller with traditional economic optimization layers such as real-time optimization (RTO). Crucial enabling developments for this include: a) increased process understanding leading to highly-detailed first-principles dynamic process models, b) enhanced formulations with stability and robustness guarantees, c) advances in numerical

Victor M. Zavala and Lorenz T. Biegler
Department of Chemical Engineering, Carnegie Mellon University, USA
e-mail: vzavala@mcs.anl.gov, lb01@andrew.cmu.edu

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 419–432.
springerlink.com

strategies for DAE-constrained optimization and NLP algorithms, and d) advances in computational resources including the availability of parallel and multi-core technology.

In this work, special emphasis is made on the numerical solution aspects and performance of combined MHE and NMPC strategies. In particular, a general solution framework based on interior-point NLP solvers and sensitivity concepts is considered. In the following section, we introduce some basic concepts and notation and describe specific formulations of the MHE and NMPC nonlinear programming problems. In Section 3 we discuss advantages of interior-point NLP solvers and present some basic NLP sensitivity results. In Section 4 we derive advanced-step approximation strategies for MHE and NMPC, based on NLP sensitivity to reduce on-line computational time. We also discuss their general stability and performance properties, especially when both are applied together. In Section 5, the potential of the combined MHE and NMPC solution framework is demonstrated on a large-scale case study involving the simultaneous monitoring and control of a distributed low-density polyethylene tubular reactor. The paper then closes with general conclusions and recommendations.

2 MHE and NMPC Formulations

We begin with a discrete-time dynamic model of an *uncertain* plant of the form,

$$x_{k+1} = f(x_k, u_k) + \xi_k, \qquad y_{k+1} = \chi(x_{k+1}) + v_{k+1} \tag{1a}$$

where $x_k \in \Re^{n_x}$ is the *true* plant state at time instant t_k and $u_k \in \Re^{n_u}$ is the implemented control action. The nonlinear dynamic model $f(\cdot,\cdot) : \Re^{n_x+n_u} \rightarrow \Re^{n_x}$ is the nominal model and satisfies $f(0,0) = 0$. The observed output $y_k \in \Re^{n_y}$ with $n_y \leq n_x$ is related to the state-space x_k through the nonlinear mapping $\chi(\cdot) : \Re^{n_x} \rightarrow \Re^{n_y}$. The true plant deviates from the nominal prediction due to the process disturbance $\xi_k \in \Re^{n_x}$ and measurement noise $v_k \in \Re^{n_y}$.

Assume that the plant is currently located at sampling time t_k with the output and input measurements $\eta_k^{mhe} := \{y_{k-N}, ..., y_k, u_{k-N}, ..., u_{k-1}\}$ distributed over a horizon containing N steps. The output measurement covariance is given by $\mathbf{R} \in \Re^{n_y \times n_y}$. The *a priori* estimate of the past state of the plant is denoted as $\bar{x}_{k-N}$ and has an associated covariance $\mathbf{\Pi_{0,k}} \in \Re^{n_x \times n_x}$. Using this information, we would like to compute an estimate $\tilde{x}_k$ of the current state x_k. In order to do this, we solve the MHE problem,

$$\mathcal{M}(\eta_k^{mhe}) \quad \min_{z_0} \; \|z_0 - \bar{x}_{k-N}\|^2_{\mathbf{\Pi^{-1}_{0,k}}} + \sum_{l=0}^{N} \|y_{k+l-N} - \chi(z_l)\|^2_{\mathbf{R}^{-1}} \tag{2a}$$

$$\text{s.t.} \quad z_{l+1} = f(z_l, u_{k+l-N}), \;\; l = 0, ..., N-1 \tag{2b}$$

$$z_l \in \mathbb{X} \tag{2c}$$

All the MHE problem data can be summarized in the vector η_k^{mhe}. Symbols $z_l \in \Re^{n_x}$ are *internal* decision variables of the optimization problem. This problem has n_x degrees of freedom corresponding to z_0. From the solution trajectory, $\{z_0^*, ..., z_N^*\}$, we obtain the optimal estimate $\tilde{x}_k = z_N^*$ with associated estimation error $e_k := \tilde{x}_k - x_k$. Using this estimate, we define the problem data $\eta_k^{mpc} := \tilde{x}_k$ for the NMPC problem,

$$\mathcal{P}(\eta_k^{mpc}) \quad \min_{v_l} \quad \Psi(z_N) + \sum_{l=0}^{N-1} \psi(z_l, v_l) \tag{3a}$$

$$\text{s.t. } z_{l+1} = f(z_l, v_l) \quad l = 0, \ldots N-1 \tag{3b}$$

$$z_0 = \tilde{x}_k \tag{3c}$$

$$z_l \in \mathbb{X}, v_l \in \mathbb{U} \tag{3d}$$

where $v_l \in \Re^{n_u}$ are internal decision variables. This problem has $(N-1) \times n_u$ degrees of freedom corresponding to $v_l, l = 0, ..., N-1$. Here, we assume that the states and controls are restricted to the domains $\mathbb{X}$ and $\mathbb{U}$, respectively. The stage cost is defined by $\psi(\cdot, \cdot) : \Re^{n_x+n_u} \rightarrow \Re$, while the terminal cost is denoted by $\Psi(\cdot) : \Re^{n_x+n_u} \rightarrow \Re$. The control action is extracted from the trajectory optimal trajectory $\{z_0^* ... z_N^* \, v_0^*, ..., v_{N-1}^*\}$ as $u_k = v_0^* := h(\tilde{x}_k)$, and $h(\cdot)$ denotes the feedback law. Note that this control action is *inaccurate* because the true state of the plant is x_k and not the estimate $\tilde{x}_k$. That is, the estimation error acts as an additional disturbance. At the next time, the plant will evolve as,

$$x_{k+1} = f(x_k, h(\tilde{x}_k)) + \xi_k, \qquad y_{k+1} = h(x_{k+1}) + v_{k+1} \tag{4}$$

With this, we shift the measurement sequence one step forward to obtain $\eta_{k+1}^{mhe} := \{y_{k-N+1}, ..., y_{k+1}, u_{k-N+1}, ..., u_k\}$, and we solve the new MHE problem. Having the new state estimate $\tilde{x}_{k+1}$ we solve the next NMPC problem.

Note that the above formulations are rather simplified. This makes them convenient for the conceptual analysis in subsequent sections. In practical applications, both NMPC and MHE problems are solved as *general* continuous-time DAE-constrained optimization problems. In this work, we assume that a full discretization approach is used to derive the discrete-time NMPC and MHE formulations. In this case, these NLP problems will be *sparse.* This is a crucial property to be exploited in the following sections.

A problem that is normally encountered in model-based control frameworks is that there exists a computational feedback delay equal to the solution time of the MHE and NMPC problems. In large-scale applications (say $n_x \approx 100 - 10,000$), this computational delay might dominate the time constant of the plant and destabilize the process. Therefore, we seek to derive strategies to reduce the on-line computational time. The first crucial component of these strategies is a fast NLP algorithm. In the next section, we discuss some of the advantages that interior-point NLP solvers offer for the solution of very large problems.

3 Full-Space Interior-Point NLP Solvers

The NLP problems (2) and (3) can be posed in the general form,

$$\mathcal{N}(\eta)\ \min_{\mathbf{x}} F(\mathbf{x},\eta) \tag{5a}$$

$$s.t.\ \mathbf{c}(\mathbf{x},\eta)=0 \tag{5b}$$

$$\mathbf{x}\geq 0 \tag{5c}$$

where $\mathbf{x}\in\Re^{n_{\mathbf{x}}}$ is variable vector containing all the states and controls and η is the data vector.

Full-space interior-point solvers have become a popular choice for the solution of large-scale and sparse NLPs. In particular, the solvers LOQO, KNITRO and IPOPT are widely used. In this work, we use IPOPT, an open-source NLP solver originally developed in our research group [1]. In interior-point solvers, the inequality constraints of problem (5) are handled *implicitly* by adding barrier terms to the objective function,

$$\min_{\mathbf{x}}\ F(\mathbf{x},\eta)-\mu_\ell\sum_{j=1}^{n_{\mathbf{x}}}\ln(\mathbf{x}^{(j)}),\quad \text{s.t.}\quad \mathbf{c}(\mathbf{x},\eta)=0 \tag{6}$$

where $\mathbf{x}^{(j)}$ denotes the jth component of vector $\mathbf{x}$. Solving (6) for a decaying sequence of $\mu_\ell\rightarrow 0,\ell\rightarrow\infty$ results in an efficient strategy to solve the original NLP (5). IPOPT solves the Karush-Kuhn-Tucker (KKT) conditions of this sequence of barrier problems (6),

$$\nabla_{\mathbf{x}}F(\mathbf{x},\eta)+\nabla_{\mathbf{x}}\mathbf{c}(\mathbf{x},\eta)\lambda-\nu=0 \tag{7a}$$

$$\mathbf{c}(\mathbf{x},\eta)=0 \tag{7b}$$

$$\mathbf{X}\cdot\mathbf{V}\,e=\mu_\ell e \tag{7c}$$

where $\mathbf{X}=\text{diag}(\mathbf{x}),\mathbf{V}=\text{diag}(\nu)$ and $e\in\Re^{n_{\mathbf{x}}}$ is a vector of ones. Symbols $\lambda\in\Re^{n_\lambda}$ and $\nu\in\Re^{n_{\mathbf{x}}}$ are Lagrange multipliers for the equality constraints and bounds, respectively. To solve this system of nonlinear equations we apply an exact Newton method with the iteration sequence initialized at $s_o^T:=[\mathbf{x}_o^T\ \lambda_o^T\ \nu_o^T]$. At the ith iteration, the search direction $\varDelta s_i=s_{i+1}-s_i$ is computed by linearization of the KKT conditions (7),

$$\begin{bmatrix}\mathbf{H}_i & \mathbf{A}_i & -\mathbb{I}_{n_{\mathbf{x}}}\\ {\mathbf{A}_i}^T & 0 & 0\\ \mathbf{V}_i & 0 & \mathbf{X}_i\end{bmatrix}\begin{bmatrix}\varDelta\mathbf{x}_i\\ \varDelta\lambda_i\\ \varDelta\nu_i\end{bmatrix}=-\begin{bmatrix}\nabla_{\mathbf{x}}F(\mathbf{x}_i)+\mathbf{A}_i\lambda_i-\nu_i\\ \mathbf{c}(\mathbf{x}_i)\\ \mathbf{X}_i\mathbf{V}_ie-\mu_\ell e\end{bmatrix} \tag{8}$$

where $\mathbf{A}_i:=\nabla_{\mathbf{x}}\mathbf{c}(\mathbf{x}_i,\eta)$, $\mathbf{H}_i\in\Re^{n_{\mathbf{x}}\times n_{\mathbf{x}}}$ is the Hessian of the Lagrange function $\mathcal{L}F(\mathbf{x}_i,\eta)+\lambda_i^T\mathbf{c}(\mathbf{x}_i,\eta)-{\nu_i}^T\mathbf{x}_i$ and $\mathbb{I}_{n_{\mathbf{x}}}$ denotes the identity matrix.

We provide exact Hessian and Jacobian information through the modeling platform AMPL. With this, Newton's method guarantees fast local convergence and is able to handle problems with many degrees of freedom without altering these convergence properties. After solving a sequence of barrier problems for $\mu_\ell \rightarrow 0$, the solver returns the optimal solution triplet $s_*^T = [\mathbf{x}_*^T \; \lambda_*^T \; \nu_*^T]$ which implicitly defines the *active-set* (set of variables satisfying $\mathbf{x}^{(j)} = 0$).

3.1 Computational Issues

Solving the KKT system (8) is the most computationally intensive step in the solution of the NLP. A crucial advantage that interior-point solvers offer over active-set solvers is that the structure of the KKT matrix in (8) *does not change* between iterations. This facilitates the design of tailored linear algebra strategies to exploit special structures. For instance, the KKT matrix arising from DAE-constrained optimization problems has a natural forward structure (almost-block-diagonal) in time and classical Riccati-like recursions and condensing techniques are often applied, where the complexity of these solution strategies scales linearly with the horizon length N, but cubically with the number of states n_x and controls n_u. On the other hand, specialized strategies have been developed that reduce the cubic computational complexity and also preserve numerical stability in the face of unstable dynamics [3, 4].

In IPOPT, we use a symmetric indefinite factorization of the KKT matrix (with $\Delta\nu_i$ eliminated). With this, we exploit only the *sparsity pattern* of the KKT matrix. The computational complexity of this strategy is in general very favorable, scaling nearly linearly and at most quadratically with the overall dimensions of the NLP (e.g. length of prediction horizon, number of states and number of degrees of freedom). This general approach also remains stable in the face of unstable dynamics. However, significant fill-in and computer memory bottlenecks might arise during the factorization step if the sparsity pattern is not properly exploited. In order to factorize the KKT matrix, we use the linear solver MA57 from the Harwell library [5]. Since the structure of the KKT matrix does not change between iterations, the linear solver needs to analyze the sparsity pattern *only once*. During this analysis phase, the linear solver permutes the matrix to reduce fill-in and computer memory requirements in the factorization phase. Two reordering strategies are normally used in MA57. The first is an approximate minimum degree (AMD) ordering algorithm while the second is a nested dissection algorithm based on the multi-level graph partitioning strategy, implemented in Metis [6]. For very large-scale problems, these nested dissection techniques excel at identifying high-level (coarse-grained) structures and thus play a crucial role in the factorization time and reliability of the linear solver. These notable advances in numerical linear algebra can dramatically expand the application scope of NMPC and MHE.

IPOPT also applies a regularization scheme to the KKT matrix in order to account for directions of negative curvature and rank-deficient Jacobians which are commonly encountered in highly nonlinear NLPs and/or ill-posed formulations. Directions of negative curvature are detected implicitly through the linear solver, which returns the so-called inertia of the KKT matrix (number of positive, negative and zero eigenvalues). If the inertia is *correct* at the solution, no regularization is necessary and we can guarantee that the optimal point is a well-defined minimum satisfying strong second order conditions (SSOC) and the linear independence qualification of the constraints (LICQ) [7]. In the context of NMPC and MHE, checking for SSOC is important since this is directly related to properties of the dynamic system such as controllability and observability. Consequently, checking for SSOC through the inertial properties of the KKT matrix is another important advantage of using a general factorization strategy, as opposed to other tailored linear algebra strategies.

3.2 NLP Sensitivity and Warm-Starts

Problem (5) is parametric in the data η and the optimal primal and dual variables can be treated as implicit functions of η. For a *sufficiently small* μ_ℓ, the KKT conditions (7) of the barrier problem (6) can be expressed as $\varphi(s(\eta), \eta) = 0$ and we define $\mathbf{K}_*(\eta_0)$ as the KKT matrix in (8).

We are interested in computing fast approximate solutions for neighboring problems around an already available nominal solution $s_*(\eta_0)$. In order to do this, we make use of the following classical results,

Theorem 1. *(NLP Sensitivity) [7, 8]. If $F(\cdot)$ and $c(\cdot)$ of the parametric problem $\mathcal{N}(\eta)$ are twice continuously differentiable in a neighborhood of the nominal solution $s_*(\eta_0)$ and this solution satisfies LICQ and SSOC, then $s_*(\eta_0)$ is an isolated local minimizer of $\mathcal{N}(\eta_0)$ and the associated Lagrange multipliers are unique. Moreover, for η in a neighborhood of η_0 there exists a unique, continuous and differentiable vector function $s_*(\eta, N)$ which is a local minimizer satisfying SSOC and LICQ for $\mathcal{N}(\eta)$. Finally, there exists a positive Lipschitz constant L such that $\|s_*(\eta, N) - s_*(\eta_0, N)\| \leq L\|\eta - \eta_0\|$ along with a positive Lipschitz constant L_F such that the optimal values $F(\eta)$ and $F(\eta_0)$ satisfy $\|F(\eta) - F(\eta_0)\| \leq L_F\|\eta - \eta_0\|$.*

Under these results, a step $\Delta s(\eta)$ computed from,

$$\begin{aligned}\mathbf{K}_*(\eta_0)\Delta s(\eta) &= -\left(\varphi(s_*(\eta_0), \eta) - \varphi(s_*(\eta_0), \eta_0)\right)\\ &= -\varphi(s_*(\eta_0), \eta).\end{aligned} \tag{9}$$

with $\Delta s(\eta) = \tilde{s}(\eta) - s_*(\eta_0)$, is a Newton step taken from $s_*(\eta_0)$ towards the solution of a neighboring problem $\mathcal{N}(\eta)$. Consequently, $\tilde{s}(\eta)$ satisfies,

$$\|\tilde{s}(\eta) - s_*(\eta)\| \leq L_s\|\eta - \eta_0\|^2 \tag{10}$$

with $L_s > 0$. Furthermore, since the KKT matrix $\mathbf{K}_*(\eta_0)$ is already available from the solution of the nominal problem $\mathcal{N}(\eta_0)$, computing this step requires only a *single backsolve* which can be performed orders of magnitude faster than the factorization of the KKT matrix.

Since the approximate solution $\tilde{s}(\eta)$ is accurate to first order, we can use it as the initial guess $s_o(\eta)$ to warm-start the NLP $\mathcal{N}(\eta)$. For instance, if the perturbation $(\eta - \eta_0)$ does not induce an active-set change, we can fix μ to a small value (e.g. say 1×10^{-6}) and reuse the KKT matrix $\mathbf{K}_*(\eta_0)$ to perform fast fixed-point iterations on the system,

$$\mathbf{K}_*(\eta_0)\Delta s_i(\eta) = -\varphi(s_i(\eta), \eta) \tag{11}$$

with $s_o = s_*(\eta_0)$. With this, we can reduce the primal and dual infeasibility of the perturbed problem $\mathcal{N}(\eta)$ until no further progress can be made with the fixed KKT matrix. For sufficiently small perturbations, these fast fixed-point iterations can converge to the solution of the perturbed problem $s_*(\eta)$. However, for large perturbations, the KKT matrix needs to be reevaluated and refactorized.

When the perturbation $\eta - \eta_0$ induces an active-set change, the linearization of the complementarity relaxation (7c) contained in the nominal KKT matrix $\mathbf{K}_*(\eta_0)$ will drive the first Newton iteration *outside* of the feasible region and the sensitivity approximation is inconsistent. To compute a fast sensitivity approximation, one could reuse the factorization of the KKT matrix through a Schur complement scheme to correct the active-set (e.g. add slack variables and constraints to drop and fix variables and bound multipliers) [9]. This is equivalent to an active-set sequential quadratic programming (SQP) iteration. Fixed-point iterations can also be performed in this way.

In the context of the proposed MHE and NMPC formulations, we define the optimal solutions,

$$s^*_{MHE} := \{z^*_0, ..., z^*_{N-1}, z^*_N, \lambda^*_1, ..., \lambda^*_{N-1}, \lambda^*_N\} \tag{12a}$$

$$s^*_{MPC} := \{z^*_0, ..., z^*_{N-1}, z^*_N, v^*_0, ..., v^*_{N-2}, v^*_{N-1}, \lambda^*_0, ..., \lambda^*_{N-1}, \lambda^*_N\}. \tag{12b}$$

The associated sensitivity approximations are denoted as $\tilde{s}_{MHE}$ and $\tilde{s}_{MPC}$, respectively, and the corresponding *warm-start* vectors as s^o_{MHE} and s^o_{MPC}. Notice that we have not included the bound multipliers in order to simplify the presentation.

4 Advanced-Step MHE and NMPC Strategies

It is possible to minimize the on-line time required to solve the MHE problem and then the NMPC problem to two fast backsolves using an advanced-step framework [2, 10]. Imagine that at time t_k we know the control action u_k and we would like to obtain an estimate of the future state x_{k+1} but we don't know the future measurement y_{k+1}. Nevertheless, we can use the current estimate $\tilde{x}_k$ and control u_k to predict the future state and associated measurement,

$$\bar{x}_{k+1} = f(\tilde{x}_k, u_k), \qquad \bar{y}_{k+1} = \chi(\bar{x}_{k+1}) \tag{13}$$

to complete the problem data $\bar{\eta}^{mhe}_{k+1} := \{y_{k+1-N}, ..., \bar{y}_{k+1}, u_{k-N}, ..., u_k\}$ and start the solution of the predicted problem $\mathcal{M}(\bar{\eta}^{mhe}_{k+1})$. Simultaneously, we can use the predicted state to define $\bar{\eta}^{mpc}_{k+1} := \bar{x}_{k+1}$ and start the solution of the predicted problem $\mathcal{P}(\bar{\eta}^{mpc}_{k+1})$. Note that both problems are decoupled so this can be done simultaneously and thus reduce the sampling time. At the solution of these problems, we hold the corresponding KKT matrices $\mathbf{K}^{mhe}_*$ and $\mathbf{K}^{mpc}_*$.

Once the true measurement y_{k+1} becomes available, we compute a fast backsolve with $\mathbf{K}^{mhe}_*$ to obtain an *approximate* state estimate $\tilde{x}^{as}_{k+1}$ which differs from the optimal state estimate $\tilde{x}_{k+1}$ and the true state x_{k+1}. Using the approximate state estimate we perform a fast backsolve with $\mathbf{K}^{mpc}_*$ to obtain the approximate control action $u_{k+1} = h^{as}(\tilde{x}^{as}_{k+1})$. Of course, this also differs from the ideal NMPC control $h(\tilde{x}_{k+1})$.

To warm-start the background problems at the next sampling time, we use the approximate solutions $\tilde{s}_{MHE}$ and $\tilde{s}_{MPC}$ to generate the shifted warm-start sequences for the next problems $\mathcal{M}(\bar{\eta}^{mhe}_{k+2})$ and $\mathcal{P}(\bar{\eta}^{mpc}_{k+2})$ [11],

$$s^o_{MHE} := \{\tilde{z}_1, ..., \tilde{z}_N, f(\tilde{x}^{as}_{k+1}, u_{k+1}), \tilde{\lambda}_2, ..., \tilde{\lambda}_N, 0\} \tag{14a}$$

$$s^o_{MPC} := \{\tilde{z}_1, ..., \tilde{z}_N, \tilde{z}_N, \tilde{v}_1, ..., \tilde{v}_{N-1}, \tilde{v}_{N-1}, \tilde{\lambda}_1, ..., \tilde{\lambda}_N, \tilde{\lambda}_N\}. \tag{14b}$$

from which we update the KKT matrices in between sampling times. Note that the approximate solutions $\tilde{s}_{MHE}$ and $\tilde{s}_{MPC}$ can also be refined in background using fixed-point iterations with $\mathbf{K}^{mhe}_*$ and $\mathbf{K}^{mpc}_*$ *before* using them to generate the warm-start sequences. We summarize the proposed framework for the advanced-step MHE and NMPC strategies, *asMHE* and *asNMPC*, respectively, as follows:

In background, between t_k and t_{k+1}:

1. Use *current* estimate $\tilde{x}^{as}_k$ and control u_k to predict the future state $\bar{x}_{k+1} = f(\tilde{x}^{as}_k, u_k)$ and corresponding output measurement $\bar{y}_{k+1} = \chi(\bar{x}_{k+1})$.
2. Define the data $\bar{\eta}^{mhe}_{k+1} = \{y_{k+1-N}...y_k, \bar{y}_{k+1}, u_{k+1-N}, ..., u_k\}$ and $\bar{\eta}^{mpc}_{k+1} = \bar{x}_{k+1}$. Use the available warm-start points s^o_{MHE} and s^o_{MPC} to solve the predicted problems $\mathcal{M}_N(\bar{\eta}^{mhe}_{k+1})$ and $\mathcal{P}_N(\bar{\eta}^{mhe}_{k+1})$.
3. Hold the KKT matrices $\mathbf{K}^{mhe}_*$ and $\mathbf{K}^{mpc}_*$.

On-line, at t_{k+1}:

1. Obtain the true measurement y_{k+1} and define the *true* MHE data η^{mhe}_{k+1}. Reuse factorization of $\mathbf{K}^{mhe}_*$ to quickly compute $\tilde{s}_{MHE}$ from (9) and extract $\tilde{x}^{as}_{k+1}$.
2. Use $\tilde{x}^{as}_{k+1}$ to define the *true* NMPC problem data η^{mpc}_{k+1}. Reuse factorization of $\mathbf{K}^{mpc}_*$ to quickly compute $\tilde{s}_{MPC}$ from (9) and extract $u_{k+1} = h^{as}(\tilde{x}^{as}_{k+1})$.
3. If necessary, refine $\tilde{s}_{MHE}$ and $\tilde{s}_{MPC}$. Generate the warm-starts s^o_{MHE} and s^o_{MPC}, set $k := k + 1$, and return to background.

4.1 Stability Issues

It is clear that both the state estimate and the associated control action are *suboptimal* due to the presence of NLP approximation errors. Here, we are interested in assessing the impact of these errors in the stability of the closed-loop system. From the controller point of view, we are interested in finding sufficient conditions under which the closed-loop remains stable in the face of disturbances and NLP sensitivity errors. Due to space limitations we outline the main results here and refer the interested reader to [2] for more details.

To start the discussion, we first note that solving the predicted problem $\mathcal{P}(\bar{x}_{k+1})$ in the *asNMPC* controller is equivalent to solving the *extended* problem,

$$\mathcal{P}_{N+1}(\eta_k^{mpc}) \quad \min_{v_l} \quad \Psi(z_N) + \psi(x_k, u_k) + \sum_{l=0}^{N-1} \psi(z_l, v_l) \tag{15a}$$

$$\text{s.t. } z_{l+1} = f(z_l, v_l) \quad l = 0, \ldots N-1 \tag{15b}$$

$$z_0 = f(x_k, u_k) \tag{15c}$$

$$z_l \in \mathbb{X}, v_l \in \mathbb{U} \tag{15d}$$

with fixed x_k, $u_k = h(x_k)$ and $\eta_k^{mpc} = \{x_k, h(x_k)\}$. For the optimal or ideal NMPC controller (instantaneous optimal solutions), we consider the neighboring costs of the extended problems with *perfect* state information $J_{x_k}^{h(x_k)} := J_{N+1}(x_k, h(x_k))$ and $J_{x_{k+1}}^{h(x_{k+1})} := J_{N+1}(x_{k+1}, h(x_{k+1}))$ as reference points. As observed by Muske and Rawlings [12], since the *implemented* control action is based on the state *estimate* $\tilde{x}_k$ coming from MHE and not on the true state x_k, we consider this as an additional disturbance to the closed-loop system through the cost $J_{\hat{x}_{k+1}}^{h(\hat{x}_{k+1})}$ where $\hat{x}_{k+1} = f(x_k, h(\tilde{x}_k)) + \xi_k$. From Lipschitz continuity of the cost function we have,

$$|J_{\hat{x}_{k+1}}^{h(\hat{x}_{k+1})} - J_{x_{k+1}}^{h(x_{k+1})}| \leq L_J L_f L_h \|x_k - \tilde{x}_k\|.$$

Explicit bounds and convergence properties on the estimator error $\|x_k - \tilde{x}_k\|$ can be established for the MHE formulation (2) [15]. Moreover, we can also treat this error as another disturbance ξ_k and define $\tilde{x}_k := x_k + \xi_k$. This allows us to restate the following robustness result for the combined *asMHE* and *asNMPC* strategies.

Theorem 2 (*Theorem 6 in [2]*)**.** *Assume that the NLPs for (2) and (3) can be solved within one sampling time. Assume also that nominal and robust stability assumptions for ideal NMPC hold (see [2]), then there exist bounds on the noise ξ and v for which the cost function $J_{N+1}(x)$, obtained from the combined asMHE-asNMPC strategy, is an input to state stable (ISS) Lyapunov function, and the resulting closed-loop system is ISS stable.*

5 Case Study

We demonstrate the performance of the proposed advanced-step framework on a low-density polyethylene (LDPE) tubular reactor process. A schematic representation of a typical multi-zone LDPE reactor is presented in Figure 1. In these reactors, high-pressure (2000-3000 atm) ethylene polymerizes through a free-radical mechanism in the presence of peroxide initiators, which are fed at multiple zones in order to start and stop the polymerization. The large amounts of heat produced by polymerization are removed at each zone using cooling water, along with multiple feeds of ethylene that cool the ethylene-polymer reacting mixture flowing inside the reactor core. Initiator flow rates, ethylene side-streams flow rates and temperatures, and the cooling water inlet temperatures and flow rates can be manipulated to achieve an axial reactor temperature profile that produces a desired polymer grade. A common problem in these reactors is that polymer accumulates (i.e., fouls) on the reactor walls. The resulting fouling layer blocks heat flow to the jacket cooling water and can be seen as a persistent dynamic disturbance. In the absence of a suitable control system, this fouling layer will eventually lead to thermal runaway. A centralized model-based control strategy based on a first-principles reactor model can deal effectively with fouling monitoring, zone control decoupling and direct optimization of the overall process economics (e.g. maximize production, minimize energy consumption). Nevertheless, LDPE reactor models consist of very large sets of PDAEs that describe the evolution of the reactor mixture and of the cooling water temperature along the axial and time dimension. After axial discretization, a typical LDPE reactor model can easily contain more than 10,000 DAEs.

An MHE estimator and an NMPC controller based on first-principles LDPE reactor models have been reported in [13, 14]. While these reports stress the benefits of these strategies for the LDPE process, little emphasis has been placed on the computational limitations associated to their on-line solution. Here, we consider these issues through the proposed advanced-step control framework where we effectively minimize the on-line computation with negligible approximation errors. We simulate the scenario in which the reactor is fouled and cleaned over time, by ramping the reactor heat-transfer coefficients (HTCs) down and up. Because this effect is directly reflected through HTCs in the LDPE reactor model, we do not estimate the process disturbance ξ_k, and instead use the MHE estimator to estimate the HTCs

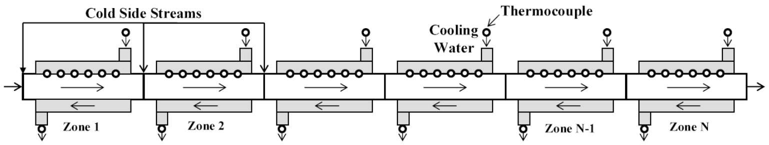

Fig. 1 Schematic representation of multi-zone LDPE tubular reactor

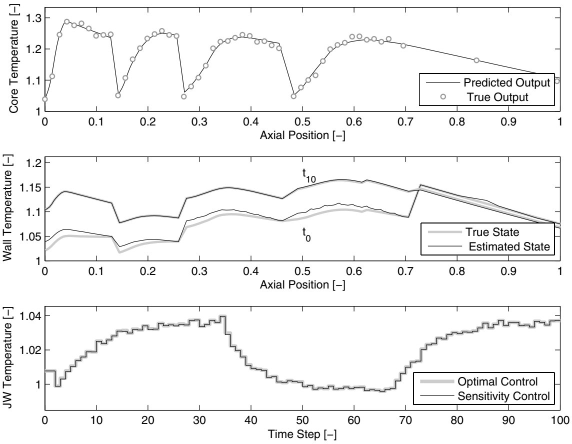

Fig. 2 Performance of advanced-step MHE and NMPC in LDPE case study

and the unmeasured model states (e.g. wall temperature profile) at each time step. For the MHE estimator, y_k consists of multiple measurements of the reactor core temperature and the output jacket temperatures in each zone. The objective of the NMPC controller is to use the estimated reactor state $\tilde{x}_k^{as}$ to drive the axial reactor temperature profile to the specified target profile. In order to do this, the NMPC controller uses the multiple inputs distributed along the reactor to obtain $u_k = h^{as}(\tilde{x}_k^{as})$. In this simulated scenario, we generate the plant response x_k from the model with the *true* HTCs. In addition, the plant is initialized at a different state from that of the NMPC controller. Finally, we corrupt the output measurements with Gaussian noise.

Since the plant response differs from that of the NMPC controller prediction and we introduce noise, the *asMHE* estimator will see a difference between the measured and the predicted outputs (see top graph of Figure 2) and will correct on-line using NLP sensitivity. We have found that the approximation errors are negligible and the *asMHE* estimator has almost identical convergence properties to that of the ideal MHE estimator. In the middle graph of Figure 2, we see that while the estimate of the reactor wall profile is inaccurate at t_0, the dashed and solid lines coincide by t_{10}, and the *asMHE* estimator converges to the *true* reactor wall profile (and the one obtained from ideal MHE) using reactor core measurements in about 10 time steps. Using the estimated states and HTCs, the *asNMPC* controller then updates the predicted state on-line. In the bottom graph of Figure 2 we present the closed-loop response of one of the jacket water inlet temperatures for the *asNMPC* controller and its ideal NMPC counterpart. As can be seen, both control actions are identical. In this graph we can also appreciate how the HTC cycles influence the controller response.

In the top graph of Figure 3 we present the total wall-clock time required to refine the perturbed solution, generate the warm-start point and solve the

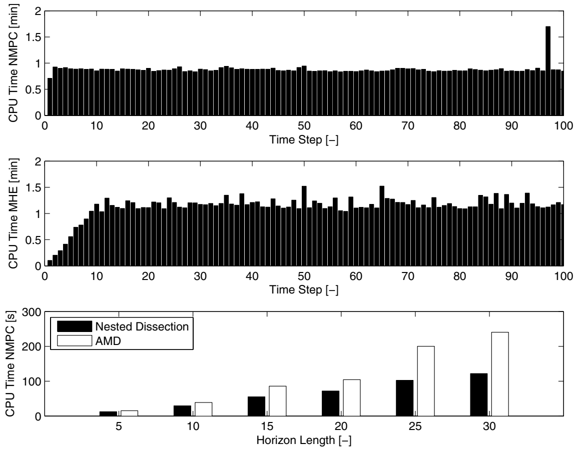

Fig. 3 Computational results. Background tasks NMPC (top). Background tasks MHE (middle). Scale-up of NMPC problem (bottom)

background NMPC problem. This time also includes some overhead coming from I/O communication tasks and from AMPL, which requires some time to generate the derivative information before calling the NLP solver. A prediction horizon of $N = 10$ time steps (20 minutes) and sampling times of 2 minutes have been used. The NMPC problem consists of an NLP with 80,950 constraints and 370 degrees of freedom. As can be seen, the overall background time is around 60 seconds and is well below the specified sampling time. A single factorization of the KKT matrix takes 15.34 seconds, a single fixed-point iteration requires 0.1 seconds, and an average of 5 fixed point iterations are required to solve the NLP. In the middle graph of Figure 3, we present total background times for the MHE estimator. The estimator is initialized in batch mode (accumulate measurements until an estimator horizon of N time steps is filled). Once the estimation horizon is complete, the background tasks take around 70 seconds to be completed. The MHE problem consists of an NLP with 80,300 constraints and 648 degrees of freedom. One fixed-point iteration requires 0.12 seconds and an average of 10 fixed point iterations solve the NLP. In the bottom graph of Figure 3, we present scale-up results of the solution time for the NMPC problem with increasing horizon length. We compare the impact of AMD and nested dissection sparse matrix reordering on the solution time of the background NLP problem (without refinement or overhead). The multi-level nested dissection strategy is more efficient here and achieves a linear scale-up. Using this strategy, a $N = 30$ NMPC problem with 242,850 constraints and 1,110 degrees of freedom is solved in around 2 minutes, the factorization of the KKT matrix takes 32.31 seconds and a fixed-point iteration requires 0.33 seconds. The AMD strategy shows quadratic scale-up and the largest problem requires 4 minutes. This difference can be attributed to the fact that the Metis nested dissection algorithm is much more efficient in identifying coarse-grained structures in the NMPC problem (LDPE multi-zone model, DAE forward structure, etc.), while AMD tends to focus on fine-grained structures. All calculations were obtained using a quad-core Intel processor running Linux at 2.4 GHz.

6 Conclusions

In this work, we present computational strategies for MHE and NMPC problems. In particular, a general solution framework based on interior-point NLP solvers and sensitivity concepts is considered. We emphasize that exploiting the overall sparsity pattern of the KKT matrix arising in NMPC and MHE problems leads to a computationally efficient and stable strategy to compute the Newton step. We analyze the impact of different reordering techniques of the KKT matrix on the factorization time and computer memory limitations. In particular, we present NLP sensitivity-based strategies for MHE and NMPC that reduce the on-line computation time to only two fast backsolves. This negligible computation effectively removes the problem of computational delay even for very large NLP models. Finally, we discuss stability issues of the NMPC controller in the face of sensitivity errors and demonstrate the developments in a distributed polymerization reactor process, where highly accurate solutions can be obtained in a negligible amount of time.

References

1. Wächter, A., Biegler, L.T.: On The Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. Math. Programm. 106, 25–57 (2006)
2. Zavala, V.M., Biegler, L.T.: The Advanced Step NMPC Controller: Stability, Optimality and Robustness. Automatica 45, 86–93 (2009)
3. Zavala, V.M., Laird, C.D., Biegler, L.T.: Fast Implementations and Rigorous Models: Can Both be Accommodated in NMPC? Int. Journal of Robust and Nonlinear Control 18, 800–815 (2008)
4. Schäfer, A., Kühl, P., Diehl, M., Schlöder, J., Bock, H.G.: Fast reduced multiple shooting methods for nonlinear model predictive control. Chemical Engineering and Processing 46, 1200–1214 (2007)
5. Duff, I.S.: MA57 - A Code for the Solution of Sparse Symmetric Definite and Indefinite Systems. ACM Transactions on Mathematical Software 30, 118–144 (2004)
6. Karypis, G., Kumar, V.: A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. SIAM J. Sci. Comput. 20, 359–392 (1999)
7. Fiacco, A.V.: Introduction to Sensitivity and Stability Analysis in Nonlinear Programming. Academic Press, New York (1983)
8. Büskens, C., Maurer, H.: Sensitivity Analysis and Real-Time Control of Parametric Control Problems Using Nonlinear Programming Methods. In: Grötschel, M., et al. (eds.) On-line Optimization of Large-scale Systems. Springer, Berlin (2001)

9. Bartlett, R.A., Biegler, L.T., Backstrom, J., Gopal, V.: Quadratic Programming Algorithms for Large-Scale Model Predictive Control. Journal of Process Control 12, 775–795 (2002)
10. Zavala, V.M., Laird, C.D., Biegler, L.T.: A Fast Moving Horizon Estimation Algorithm Based on Nonlinear Programming Sensitivity. Journal of Process Control 18, 876–884 (2008)
11. Findeisen, R., Diehl, M., Burner, T., Allgöwer, F., Bock, H.G., Schlöder, J.P.: Efficient Output Feedback Nonlinear Model Predictive Control. In: Proceedings of American Control Conference, vol. 6, pp. 4752–4757 (2002)
12. Muske, K.R., Meadows, E.S., Rawlings, J.B.: The Stability of Constrained Receding Horizon Control with State Estimation. In: Proceedings of American Control Conference, vol. 3, pp. 2837–2841 (1994)
13. Zavala, V.M., Biegler, L.T.: Large-Scale Nonlinear Programming Strategies for the Operation of Low-Density Polyethylene Tubular Reactors. In: Proceedings of ESCAPE 18, Lyon (2008)
14. Zavala, V.M., Biegler, L.T.: Optimization-Based Strategies for the Operation of Low-Density Polyethylene Tubular Reactors: Moving Horizon Estimation. Comp. and Chem. Eng. 33, 379–390 (2009)
15. Alessandri, A., Baglietto, M., Battistelli, G.: Moving-Horizon State Estimation for Nonlinear Discrete-Time Systems: New Stability Results and Approximation Schemes. Automatica 44, 1753–1765 (2008)