FAST NONLINEAR MODEL PREDICTIVE CONTROL: OPTIMIZATION STRATEGY AND INDUSTRIAL PROCESS APPLICATIONS

F. D'Amato and A. Kumar
GE Global Research
Niskayuna, NY 12309, USA

R. Lopez-Negrete and L. T. Biegler
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

With the widespread availability of model predictive control (MPC), nonlinear MPC provides a natural extension to include nonlinear models for trajectory tracking and dynamic optimization. NMPC can include first principle models developed for off-line dynamic studies as well as nonlinear data-driven models, but requires the application of efficient large-scale optimization strategies to avoid computational delays and to ensure stability, robustness and superior performance. This study presents the application of the recently developed advanced step NMPC (asNMPC) strategy. This approach solves the detailed optimization problem in background and applies a sensitivity-based update on-line. Two large-scale process case studies: detailed distillation control and multi-step operation for steam generation in a power plant. In both cases, efficient and robust controller performance is achieved with nonlinear dynamic optimization.

Keywords

Nonlinear model predictive control (NMPC), dynamic optimization, nonlinear programming (NLP), NLP sensitivity

Introduction

Linear Model Predictive Control (MPC) has been widely applied for advanced control for over 30 years. Its key advantages are that it serves as a general-purpose MIMO controller, facilitates the addition of constraints on inputs and outputs, and is well adapted to the slower time scales in process control. Most MPC implementations rely on linear dynamic models in either state-space or step response form, and standard formulations enable the treatment of feedforward disturbances, offset correction for model mismatch, and robustness to noise.

Moreover, MPC extensions have been developed to handle empirical nonlinear models including neural networks, Volterra series, and Wiener and Hammerstein models. Moreover, hybrid logic models have been developed that include linear models with binary variables and multi-models. Finally, nonlinear MPC has also been developed with first principle differential-algebraic models, which allow a direct link to off-line dynamic simulation and planning. The evolution of MPC to NMPC requires the application of nonlinear Differential Algebraic Equation (DAE) models, particularly first principle models that describe the process over a wide range. This extension is essential for nonlinear processes that exhibit wide swings in operation, require frequent product transitions, and consist of multi-stage and non-standard operations. Surveys of NMPC can be found in Qin and Badgwell (2000), Allgoewer and Zheng (2000), Camacho and Bordons (2007) and Findeisen et al. (2007). In addition, NMPC is a natural vehicle for Dynamic Real-time Optimization (D-RTO) as it provides compatibility with the controller and process dynamics and is readily adapted to handle on-line uncertainties due to noise and process variations. Numerous applications of D-RTO have been described in Groetschel et al. (2001).

On the other hand, the realization of NMPC requires the application of a fast nonlinear programming (NLP) solver for time-critical, on-line optimization. Clearly, if the NLP solver cannot compute the desired input within a single sampling time, it cannot exploit the current state of the process and performance will deteriorate. Moreover, stability of the controller is based on an instantaneous injection of the input, once the process state is obtained. Online computation of this input by an NLP solver leads to a delay that degrades performance and may even

destabilize the control system (Findeisen and Allgoewer, 2004; Santos et al., 2002). The past five years have seen the development of modifications to NMPC that address computational delay. Findeisen and Allgower (2004) extended the NMPC approach as well as its stability theory to account for computational delay. In addition, a number of fast NMPC strategies have been developed by Diehl et al. (2002), Ohtsuka (2004), Nagy et al. (2007), Franke and Doppelhammer, 2007; Bartusiak (2007) and others. Among these methods is an advanced step NMPC (asNMPC) approach that is based on NLP sensitivity, which requires very little on-line computation (Zavala et al., 2008a).

The next section briefly develops the asNMPC controller and summarizes the solution strategy, development of the sensitivity-based control scheme and stability properties for asNMPC. Following this, we discuss the importance of the modeling environment and implementation for this optimization-based controller. The third section presents a distillation case study within the AMPL optimization-based modeling environment, while the fourth section presents a power generation case study within a simulation-based modeling environment. These case studies contrast different modeling and formulation characteristics on performance, and also demonstrate the potential for further improvement. The fifth section discusses related, on-going work for sensitivity-based NMPC along with future challenges. The final section summarizes the paper and provides conclusions.

Development of Fast NMPC

Consider the optimization problem for nonlinear model predictive control written over the moving time horizon shown in Figure 1.

$$J(x(k)) = \min \quad \Psi(z_N) + \sum_{l=0}^{N} \psi(z_l, v_l)$$
$$s.t. \;\; z_{l+1} = f(z_l, v_l),\; z_0 = x(k),\; l = 0,\ldots N-1, \qquad (1)$$
$$z_l \in X,\; z_N \in X_f,\; v_l \in U$$

where $x(k)$ is the state of the plant at time k and $u(k)$ is the corresponding manipulated input variable. For the horizon starting at time t_k, z_l and v_l are the predicted states and inputs at step l. From the solution of this problem, we obtain $u(k) = v_0$ and inject it into the plant. In the nominal case, this drives the state of the plant towards $x(k+1) = f(x(k), u(k))$. Once $x(k+1)$ is known, the prediction horizon is shifted forward by one sampling instant, to time t_{k+1} and problem (1) is solved with $x(k+1)$ as the initial condition to find $u(k+1)$. This recursive strategy gives rise to the ideal NMPC controller (neglecting computational delay).

For this approach we desire that the optimized objective $J(x(k))$ be a Lyapunov function in order to guarantee stability of the closed-loop system. To satisfy this, we assume the terminal penalty satisfies $\Psi(z) > 0$, for all z in $X_f \setminus \{0\}$ and that a local control law $u = h_f(z)$ can be defined on X_f, so that for $f(z, h_f(z))$ in X_f, for all z in X_f and

$$\Psi(f(z, h_f(z)) - \Psi(z) \leq -\psi(z, h_f(z)) \text{ for all } z \text{ in } X_f.$$

Also, the optimal stage cost $\psi(x, u)$ satisfies $\alpha_p(|x|) \leq \psi(x, u) \leq \alpha_q(|x|)$ where $\alpha_p(|x|)$ and $\alpha_q(|x|)$ are $\mathcal{K}$ functions. These assumptions apply to radial basis (e.g., tracking-type) cost functions and also allow certain economically based cost terms. In particular, the use of economic objectives within MPC formulations is also addressed in a number of studies (Bartusiak, 2007; Odloak et al., 2002; Sbarbaro and Johansen, 2005; Huang et al., 2011a, 2011b; Kadam and Marquardt, 2007; Engell, 2007; Rawlings and Amrit, 2008; Diehl et al., 2011; Wuerth et al., 2009).

Now consider the state of the plant, $x(k-1)$, at t_{k-1} and that we already have the control action $u(k-1)$. In the nominal case the system evolves according to the dynamic model in (1) starting at t_{k-1}, and we can *predict* the future state exactly (i.e.; $x(k) = f(x(k-1), u(k-1))$) and compute the future control action by solving problem (1) in advance. If this problem can be solved before t_k, then $u(k)$ will already be available *without on-line computational delay.* For this case, it is also easy to prove (Zavala and Biegler, 2009a) that this strategy has identical nominal stability properties as the standard or ideal NMPC controller.

Of course, a realistic controller must also be robust to model mismatch, unmeasured disturbances and measurement noise. Here, NMPC provides a mechanism to react to these features with some inherent robustness. In particular, this tolerance to mismatch and disturbances can be characterized by input-to-state stability (Magni and Scattolini, 2007; Zavala and Biegler, 2009). Moreover, in the presence of these uncertainties, a realistic extension of our NMPC strategy is to consider a perturbed solution from problem (1) that considers the *actual state* $x(k)$ and not just its prediction.

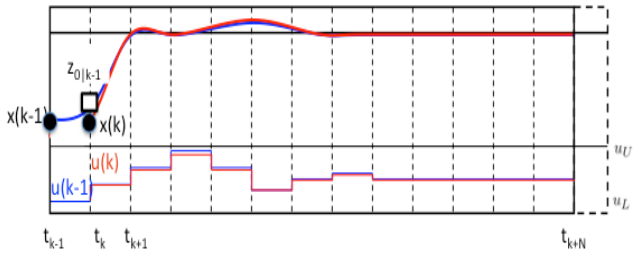


Figure 1: Moving Horizon for asNMPC

To develop the perturbed solution within the framework of the interior point NLP solver, we can introduce additional slack variables and equations in order to represent problem (1) equivalently as:

$$\min F(\mathbf{x}, p_0), \; s.t. \; c(\mathbf{x}, p_0) = 0, \; \mathbf{x} \geq 0. \qquad (2)$$

where $\boldsymbol{x}$ contains all the variables of problem (1) and p_0 is a nominal parameter value (i.e., the initial condition for z_0). Note that the NLP (2) can be formulated by discretizing the dynamic model using collocation on finite elements, known as a direct transcription formulation (Betts, 2010; Biegler, 2010). If we obtain the solution with p_0 set to the predicted state, then we require a perturbed solution of (2) for the parameter p set to the actual state, $x(k)$. In particular, this solution also contains the new input $u(k)$.

An approximate perturbed solution can be obtained through NLP sensitivity analysis, and this task is particularly efficient for interior-point NLP solvers. In particular, the IPOPT solver addresses problem (2) by applying Newton's method to the following equations:

$$\begin{aligned} \nabla F(\boldsymbol{x}, p_0) + \nabla c(\boldsymbol{x}, p_0)\,\lambda - \nu = 0 \\ c(\boldsymbol{x}, p_0) = 0 \\ XVe = \mu e \end{aligned} \qquad (3)$$

where $X = diag(\boldsymbol{x})$, $V = diag(\nu)$, and the sequence of barrier parameters μ is reduced to zero so that the solution of a sequence of problems (3) converges to the solution of problem (2). From the optimality conditions of (2) evaluated at the solution $\boldsymbol{x}^*$ one can obtain, under mild regularity conditions of the NLP (Fiacco, 1983) a second order estimate of the perturbed solution to:

$$\min F(\boldsymbol{x}, p),\ s.t.\ c(\boldsymbol{x}, p) = 0,\ \boldsymbol{x} \geq 0. \qquad (3)$$

i.e., $\Delta\boldsymbol{x} \sim \boldsymbol{x}^*(p) - \boldsymbol{x}^*(p_0)$ from the linear system:

$$\begin{bmatrix} W^* & A^* & -I \\ A^{*T} & 0 & 0 \\ V^* & 0 & X^* \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\nu \end{bmatrix} = - \begin{bmatrix} \nabla L(x^*, \lambda^*, \nu^*, p) \\ c(x^*, p) \\ 0 \end{bmatrix} \qquad (4)$$

where $A^* = \nabla c(\boldsymbol{x}^*, p_0)$ and $W^* = \nabla_{xx} L(\boldsymbol{x}^*, \lambda^*, \nu^*, p_0) = \nabla_{xx} F(\boldsymbol{x}^*, p_0) + \nabla_{xx} c(\boldsymbol{x}^*, p_0)^T \lambda^*$. Because the KKT matrix in (4) is identical to the Newton iteration matrix used in IPOPT, it is already available in factorized form. Hence, once the new state $x(k)$ is known, the change from $p_0 \rightarrow p$ is noted and the desired approximate solution can be obtained with a single on-line backsolve (Zavala et al., 2008a). As described in Zavala et al. (2008a) and Zavala and Biegler (2009a), this on-line step usually requires less than 1% of the dynamic optimization calculation.

The resulting *advanced step NMPC* (as-NMPC) controller therefore consists of the following steps:

1. Obtain $x(k-1)$ and $u(k-1)$ from the previous cycle, evaluate $z_{0|k-1} = p_0 = f(x(k-1), u(k-1))$ and solve (1) in background with p_0 (open square in Figure 1) as initial condition. This yields the blue profile in Figure 1.
2. Once the measured (or estimated) state $x(k)$ (black dot in Figure 1) is obtained from the plant, set this value to p and obtain the perturbed solution on-line using the linear system (4) derived from (2). This yields the red profile in Figure 1. Extract $u(k)$ (shown in red) from the perturbed solution and inject into the plant.
3. Set $k = k+1$ and return to Step 1.

Note from the above steps that the advanced step NMPC controller is able to handle the nonlinearity of the system since it updates the KKT matrix at each time step, while avoiding the difficulty of computational delay.

Summary of Advanced Step Properties

As mentioned above, *asNMPC* enjoys the same nominal stability properties as with the related *ideal NMPC* formulation (which cannot be implemented without computational delay). Moreover, in the case of model mismatch, and measurement and process noise, *asNMPC* also has input to state (ISS) stability. Zavala and Biegler (2009a) showed with a small nonlinear example that the amount of uncertainty that can be tolerated by *asNMPC* is comparable (up to second order) to ideal NMPC. Moreover, this robust stability property has been extended to input to state practical stability (ISpS) by Huang et al. (2011) to include a wide range of observers including Extended Kalman Filters.

The advanced step concept has also been extended to optimization formulations for moving horizon estimators (MHE), which provide on-line parameter and state estimation (Zavala et al, 2008b). Again, an NLP sensitivity-based strategy is used that predicts the next measurement, solves the MHE optimization problem in background and uses sensitivity of the KKT system to update the state and parameter estimates, once the new measurement is obtained. When combined with asNMPC, the advanced step MHE performs very well (Zavala and Biegler, 2010) and leads to offset-free formulations to deal with state and output disturbances as well as state and parameter estimation (Huang et al., 2010; Huang et al., 2011c).

More recently, asNMPC formulations have been extended to deal with economic objective functions so that D-RTO can be performed directly, without the need of a tracking function (Huang et al, 2011a,b). Because this approach leads to the optimization of the actual plant, and not to a setpoint determined by off-line optimization, significant performance improvements can be realized (Zavala and Biegler, 2009a). Finally, recent work has also seen the extension of asNMPC to incorporation of uncertainty within the moving horizon state estimation and optimization problems. This approach allows flexible

extensions to uncertainty models and confidence intervals over the prediction horizon.

Finally, both asNMPC and asMHE strategies have been applied to a number of process applications including distillation columns and chemical reactors. Two particularly large-scale applications include an air separation column with 1520 DAEs, 117,000 variables and 240 degrees of freedom as well as an LDPE polymerization reactor with about 10000 DAEs and 350, 000 variables. Despite their large size, both of these asNMPC/asMHE applications require only a few CPU seconds of on-line computation (Zavala and Biegler, 2009a; Huang et al, 2009).

Implementation Issues

The large-scale demonstrations of the advanced-step strategy for NMPC and MHE mentioned above were obtained using an efficient NLP solver (IPOPT) coupled to a state-of-the art optimization environment (AMPL). This environment allows communication to MATLAB and data handling tools. In developing these applications, some important requirements are needed to realize these implementations efficiently.

The asNMPC and asMHE strategies require the fast solution of the moving horizon NLP problem in order to provide background solutions within sampling times. The NLP formulation for (1) can be realized from first principles differential-algebraic (DAE) models through the application of orthogonal collocation on finite elements. This leads to a large-scale algebraic model suitable for an efficient NLP solver. Nevertheless, efficient solution of the resulting NLP problem (1) requires careful attention to develop smooth, well-defined functions as well as well-conditioned derivative matrices. Guidelines for developing these models are noted in Biegler (2010). In particular, models with internal calculation loops lead to a deterioration of these properties and introduce convergence noise and inaccurate functions and derivatives. In particular, for poorly conditioned problems even small levels of convergence noise can cause difficulties for NLP solvers.

In the solution of problem (1), the keys to efficient performance rest on two factors. First, to achieve fast (nearly quadratic) convergence requires exact first and second derivatives as well as effective methods for inertia detection and correction of the KKT matrix in (4). As described in Wächter and Biegler (2006), regularization terms are added by IPOPT if A is rank deficient, or the nullspace projection of W is not positive definite. This ensures a well-defined Newton step to solve (3). Second, the large linear system that represents the Newton step for (3) requires the dominant CPU time for the NLP solver, and is determined by the sparse matrix package and the exploitation of sparsity of the model equations. As noted in Zavala and Biegler (2009b), the formulation of the collocation model and the choice of sparse linear solver have a huge influence on the performance of the NMPC implementation.

To perform the sensitivity calculation, the requirements for NLP implementations apply to the sensitivity analysis as well. In particular, at the solution of (1) it is important that the regularization terms for the KKT matrix be set to zero. This signals that the inertia of the KKT matrix is correct, SSOC and LICQ are satisfied and the NLP sensitivity analysis can be applied correctly. The concepts of NLP Sensitivity have been tailored to the algorithm in IPOPT, and are compatible with its suite of linear decompositions and solvers. The resulting sensitivity approach is embodied within open source software, called sIPOPT, and is described in Pirnay et al. (2011).

Distillation Case Study

High purity distillation applications represent interesting challenges for modeling and model reduction, control and optimization. These issues have been addressed in a number of nonlinear control studies (see, e.g., Balasubramhanya and Doyle, 1997; Kumar and Daoutidis, 2003; Levine and Rouchon, 1991). In this section we discuss a case study where we implement the advance step controller to a large-scale binary distillation column. The model consists of the so-called MESH equations for each tray (Diehl, 2001). In other words, we have equations for mass and energy balances, thermodynamic equilibrium, and composition summation, for each tray, the condenser, and the reboiler. Furthermore, we assume that the dynamics of the vapor phase are negligible. This leads to an index 2 system of DAEs (Diehl 2001), which can be reduced to index 1 using transformations described in Cervantes and Biegler (1998) and Lopez-Negrete (2007). After applying the index reduction scheme and numbering the trays from bottom to top with the reboiler being $i=0$ and the condenser $i=N_T+1$, the model is given by the following equations. Note that the coefficients for the physical property equations (enthalpy, Antoine equation, etc.) can be found in Diehl (2001, 2002).

Mass balance: For each stage, the overall mass balance must hold, and the rate of change of the molar hold-up (M_i) is given by the difference in the flows to and from the adjacent stages

$$\dot{M}_i = V_{i-1} - V_i + L_{i+1} - L_i + F_i \qquad (1)$$

with V_i and L_i being the vapor and liquid flow rates. Here the molar feed flow F_i is entering the column at stage $i=21$ (all other feeds are set to zero). For the condenser, the distillate is given by D, thus we have

$$\dot{M}_{N_T+1} = V_{N_T} - D - L_{N_T+1} \tag{2}$$

On the other hand, for the reboiler the bottoms flow rate is *B*.

$$\dot{M}_0 = L_1 - V_0 - B \tag{3}$$

Moreover, the reflux ratio is modeled with

$$R = \frac{L_{N_T+1}}{D} \tag{4}$$

Additionally, the tray component-wise mass balance, assuming that only liquid molar holdup is of importance, is given by

$$\dot{M}_i x_i + M_i \dot{x}_i = V_{i-1} y_{i-1} - V_i y_i + L_{i+1} x_{i+1} - L_i x_i + F_i z_{f,i} \tag{5}$$

where x_i is the molar fraction of the volatile component in the liquid phase. Using (5) the above equation yields

$$\begin{aligned} M_i \dot{x}_i = {} & V_{i-1}(y_{i-1} - x_i) - V_i(y_i - x_i) + \\ & L_{i+1}(x_{i+1} - x_i) + F_i(z_{f,i} - x_i) \end{aligned} \tag{6}$$

For the reboiler and condenser we have,

$$\begin{aligned} \dot{M}_0 \dot{x}_0 &= L_1(x_1 - x_0) - V_0(y_0 - x_0) \\ \dot{M}_{N_T+1} \dot{x}_{N_T+1} &= V_{N_T}(y_{N_T} - x_{N_T+1}), \end{aligned} \tag{7}$$

respectively.

Equilibrium and Summation Equations: The total pressure P_i on each tray is assumed to be constant, with a pressure drop ΔP_i from top to bottom.

$$P_{i-1} = P_i + \Delta P_i, \quad i = 1, \ldots, N_T + 1 \tag{8}$$

The pressure of the condenser is set to P_{NT+1}= 93.9 kPa, while the pressure drop was set to 250 Pa per tray for the stripping section and 190 Pa per tray for the rectifying section. Thermodynamic equilibrium was modeled using Raoult's Law, so that the pressure of each tray is

$$\mathrm{P}_i = \mathrm{P}^s_{i,1}(T_i) x_i + (1 - x_i) P^s_{i,2}(T_i), \tag{9}$$

where the vapor pressures are computed using Antoine's Equation

$$\mathrm{P}^s_{i,j} = \exp\left(A_j - \frac{B_j}{T_i + C_j}\right). \tag{10}$$

Note that the tray temperatures are implicitly defined by Equation (14). However, to reduce the index of the model, we need the time derivative of the temperature, which is determined by applying the implicit function theorem to Equation (14) to yield

$$\dot{T}_i = -\frac{(P^s_{i,1} - P^s_{i,2})\dot{x}_i}{\frac{\partial P^s_{i,1}}{\partial T_i} x_i + \frac{\partial P^s_{i,1}}{\partial T_i}(1 - x_i)}. \tag{11}$$

The partial derivatives of the vapor pressures can be obtained from the Antoine Equations above.

Moreover, to account for non-equilibrium behavior in the mixtures tray efficiencies α_i are considered in the summation equations. Thus for each tray we have

$$y_i = \alpha_i x_i \frac{P^s_{i,1}}{P_i} + (1 - \alpha_i) y_{i-1}, \tag{12}$$

and for $i=0$ (reboiler) the term simplifies to

$$y_0 = x_0 \frac{P^s_{0,1}}{P_0}. \tag{13}$$

Energy balance: The liquid enthalpy is determined with

$$h^L_i(x_i, T_i) = x_i \bar{h}^L_{i,1}(T_i) + (1 - x_i) \bar{h}^L_{i,2}(T_i), \tag{14}$$

and the vapor enthalpy is given by

$$h^v_i(y_i, T_i, P_i) = y_i \bar{h}^V_{i,1}(T_i, P_i) + (1 - y_i) \bar{h}^V_{i,2}(T_i, P_i). \tag{15}$$

In addition, expressions for the pure liquid ($\bar{h}^L_{i,j}(T_i)$) and vapor enthalpies ($\bar{h}^V_{i,j}(T_i, P_i)$) required in the above equations can be found in Diehl (2001). On the other hand, the energy balance for the trays are given by

$$\begin{aligned} \dot{M}_i h^L_i + M_i \left(\dot{x}_i \frac{\partial h^L_i}{\partial x_i} + \dot{T}_i \frac{\partial h^L_i}{\partial T_i} \right) = {} & V_{i-1} h^V_{i-1} - V_i h^V_i + \\ & L_{i+1} h^L_{i+1} - L_i h^L_i + F_i h^L(z_{f,i}, T_{f,i}, P_{f,i}). \end{aligned} \tag{16}$$

For the reboiler the added heat duty is Q_R, and a loss term (Q_{loss}) is also considered, thus

$$\begin{aligned} & \dot{M}_0 h^L_0 + M_0 \left(\dot{x}_0 \frac{\partial h^L_0}{\partial x_0} + \dot{T}_0 \frac{\partial h^L_0}{\partial T_0} \right) = \\ & Q_R - Q_{loss} - V_0 h^V_0 + L_1 h^L_1 - B h^L_0. \end{aligned} \tag{17}$$

The energy balance for the condenser is

$$\dot{M}_{N_T+1} h^L_{N_T+1} + M_{N_T+1}\left(\dot{x}_{N_T+1}\frac{\partial h^L_{N_T+1}}{\partial x_{N_T+1}} + \dot{T}_{N_T+1}\frac{\partial h^L_{N_T+1}}{\partial T_{N_T+1}}\right) = V_{N_T+1} h^V_{N_T+1} - Q_C, \quad (18)$$

where the condenser heat duty is Q_C. The partial derivatives of the previous equations can be derived from Equations (16) and (19). Note that substituting equations (5)-(7), (10)-(12), and (16) into the energy balance equations yields a set of purely algebraic equations for the vapor flow rates and the condenser heat duty.

Hydrodynamics: To determine the liquid flow rates from each tray we use the Francis weir formula. For that we first need to determine the liquid volume holdup of each tray. These are related to the molar volumes through the molar volume $V_i^m(x_i, T_i)$ by the following equation

$$n_i^v = M_i V_i^m(x_i, T_i), \quad (17)$$

and the molar volumes are computed as the sum of the molar volumes of the pure components weighted by their molar fraction

$$V_i^m(x_i, T_i) = x_i \overline{V}_{i,1}^m(T_i) + (1 - x_i)\overline{V}_{i,2}^m(T_i), \quad (18)$$

where $\overline{V}_{i,1}^m(T_i)$ and $\overline{V}_{i,2}^m(T_i)$ are the temperature dependent molar volumes of pure Methanol and n-Propanol, and expressions for these equation can also be found in Diehl (2002). Finally, the liquid flow rates are calculated by the Francis weir equation given by

$$L_i V_i^m(x_i, T_i) = W_i (n_i^v - n_i^{v,ref})^{\frac{3}{2}}, \quad (19)$$

where W_i is the weir constant.

For this example we considered 60 s sampling times, and 10 sampling times in the predictive horizon. As determined by trial and error, this long prediction horizon allows us to determine a reasonably accurate, approximate NMPC solution to an infinite horizon formulation. The continuous time DAE model is transformed into a discrete time model using collocation. Thus, using 3 point Radau collocation, the NLP consists of 19814 variables and 19794 equality constraints. To account for plant model mismatch we added noise to the differential variables (total molar holdup and liquid compositions at each tray). The noise was assumed uncorrelated, zero mean and Gaussian with variance 10^{-4} for the holdups and 10^{-6} for the compositions.

The control variables of this problem are the reboiler heat Q_R and the reflux ratio R. In the objective function for the NLP we consider only temperatures from trays 14 and 28, since they are much more sensitive to changes than the overhead and bottom product streams. We also include the distillate flow rate and condenser heat duty in the objective. Note that we have numbered the trays from bottom to top. Also, we consider the penalization to changes in the inputs R and Q_R. Finally, set point information and objective function weights are summarized in Table 1.

	Objective function weights	Set point 1	Set point 2
T_{14} [K]	10^4	351.0321	356.6504
T_{28} [K]	10^4	337.4470	346.9535
D [mol/s]	10^{-1}	1.1152	18.3859
Q_C [J]	10^{-1}	8.955×10^5	1.6221×10^6
R [-]	10^2	-	-
Q_R [J]	10^2	-	-

Table 1: Set point information and objective function weights for the distillation column example.

Figure 1 shows the simulation results comparing the advanced step NMPC with the ideal case. The average solution time of the NLP was 9.4 CPU s, while the advanced step takes an average of 0.063 CPU s. Note that this implies that the computational delay has been reduced to less than 1%, and thus the performance of the NMPC should improve. Both the ideal and advanced step NMPC strategies were able to perform the set point change, however the latter adds practically no computational delays, which is important for real time optimization schemes.

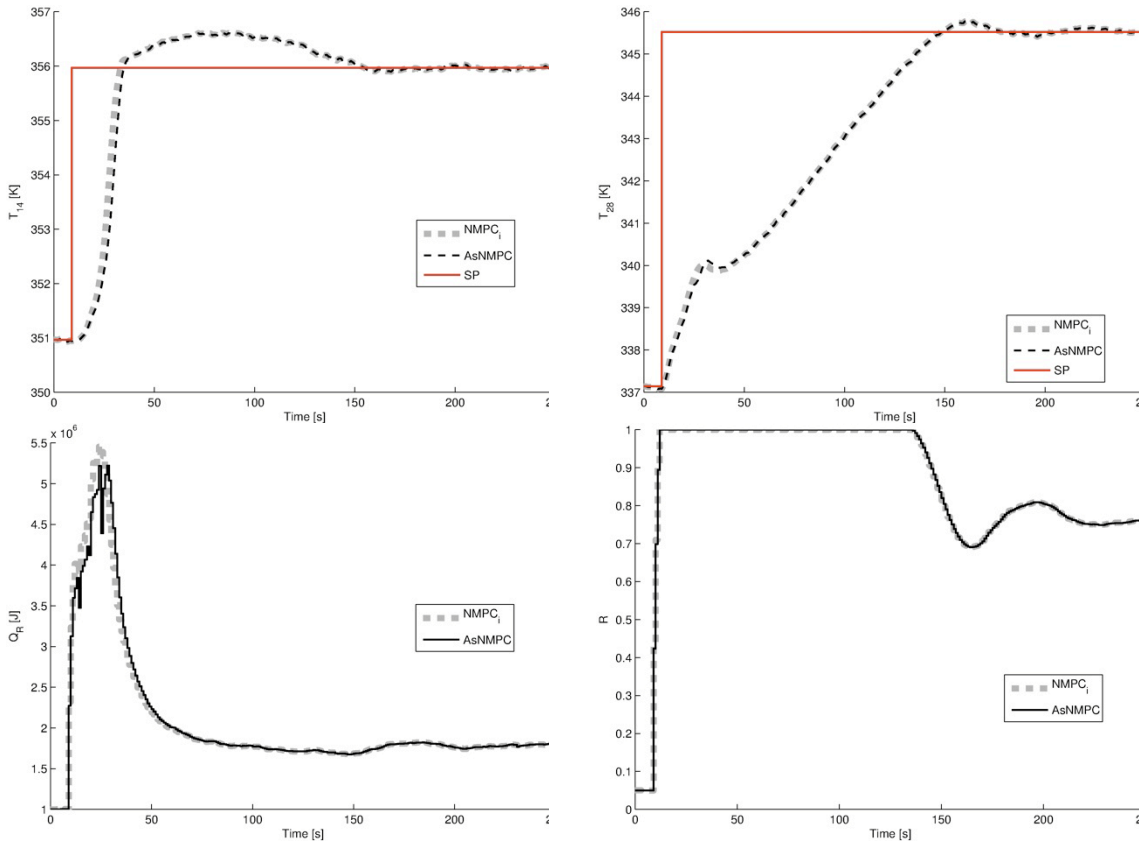


Figure 1: asNMPC control of a binary distillation column (a) Temperature of tray 14, (b) Temperature of tray 28, (c) Reboiler heat, and (d) Reflux rate.

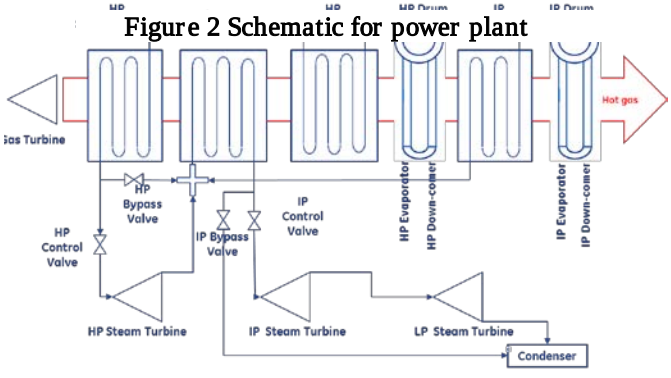
GE Case Study

A numerical study for the use of NMPC in the startup and loading of a power plant is presented in this section. Reduction of startup time and fuel consumption in fossil fuel power plants is receiving increased attention in part to reduce operating costs and become more competitive in electricity markets, and also to have a more agile response to compensate the volatility of renewable sources in the electrical grid.

The use of NMPC on combined cycle plants has been reported in (D'Amato, 2006), an industrial application addressing a limited section of the startup. This example analyzes a potential use of NMPC in an extended scope including additional phases of the startup, more actuation devices and increased range of validity of component models.

The plant under consideration consists in a combination of a gas turbine, a three-pressure steam generation system and the corresponding steam turbines (see Figure 2). The process of generating electrical energy from the fuel's chemical energy can be described as follows. Ambient air is pressurized in a compressor and directed into a combustion chamber that burns natural gas. The gasses resulting from combustion are expanded in the gas turbine to produce mechanical power, and then enter the steam generator. In turn, the steam generator includes a multiplicity of heat exchangers carefully arranged to transfer the hot gas thermal energy into three water streams, converting sub-cooled liquid into super-heated steam at three different pressures (low, intermediate and high). Finally, the resulting high, intermediate, and low pressure steam streams are expanded in corresponding steam turbines to produce mechanical power. The electrical power is generated with two electrical generators: one attached to the gas turbine and another to the common shaft of the three steam turbines.

Figure 2 Schematic for power plant



The startup process brings the plant from a state in which the turbines have zero rotational speed and there is no flow of gas, air, steam or liquid water throughout the plant, to a fully operational state in which the plant delivers electric

power to the grid or to a downstream process. As a result of the wide range of operation, the models are essentially nonlinear to capture, for example, the full extent of pressure-flow characteristics, heat transfer coefficients, mechanical properties of metallic components, and the thermodynamic properties of steam and gas.

The model used in the NMPC controller includes dynamic equations for energy, mass and momentum balance in the main components, described in Table 3. For example, detailed descriptions of dynamical models of power plant components can be found in Thomas (1999) and Casella and Leva (2006). This example focuses on the loading stage that begins when the steam turbines reach full speed and with the generators synchronized to the grid, and ends when both gas and steam turbines at full load. During this process, the steam bypass valves are gradually closed, while the gas turbine fuel valve and steam turbine control valves are gradually open. The main operational constraints are valve amplitude and rate limits, drum level upper and lower bounds, maximum temperatures and maximum stresses in steam turbine rotor. The rotor stresses include a mechanical component due to centrifugal forces and the torque transmitted to the shaft, and a thermal component due to the temperature gradients originated from the mismatch between the steam and the rotor metal temperatures through the steam turbine (see, for example, Boley (1985) and Timoshenko and Goodier (1970)). The model used here is further complicated with steam property calculations, performed with numerical subroutines, some of which are iterative in nature. The output constraints are implemented as soft constraints, and they are enforced using an l_1 penalty function added to the objective function of the NLP.

Component	Functionality
Steam turbine	Expands inlet steam to a lower pressure producing mechanical energy
Gas turbine	Burns fuel and expands hot gasses to a lower pressure producing mechanical energy
Drum	Receives feed-water and mixture of water and steam from evaporator and produces saturated steam
Evaporator	Receives heat from GT exhaust hot gasses, to boil down-comer water and send the water and steam mixture to the drum.
Down-Comer	Vertical pipe taking liquid water from the drum and feeding the evaporator
Super-Heater	Uses heat from GT exhaust to increase temperature of steam
Reheater	Uses heat from GT exhaust to increase temperature of steam expanded in the HP turbine

Table 2: Power plant components.

The MPC control problem consists in minimizing the distance between the generated power and the target dispatch power, subject to all the operational constraints. The manipulated variables are the opening of the gas turbine load valve, the cascaded bypass valve, and the HP and IP bypass valves. Also, it is assumed that the initial state is available at every control step.

After discretizing the DAE model using Radau collocation and 3 collocation points, the resulting optimization problem has 13870 variables and 12510 equality constraints. A sampling time of 60 seconds and a prediction horizon of 10 steps have been used in this example. The numerical performance of MPC in this application is analyzed through closed loop simulations where the plant model and the controller model have similar structure. The actual state *x(k)* is obtained by simulating the plant model during one control interval, starting at initial conditions *x(k-1)* and using the inputs *u(k-1).* To account for modeling, measurement and estimation errors, the actual state has been perturbed with white noise before being used by the controller. Figures 3 and 4 compare the closed loop operation results of the ideal NMPC (i.e., negligible computational delay is assumed), $\text{NMPC}_{\text{delay}}$ (the NLP solution time is added as delay), and asNMPC during a 50 minute operation. The NLPs for the controllers are implemented using IPOPT through the C++ interface. In this case, first derivatives are approximated using finite differences, and the Hessian is approximated using the Gauss-Newton approach. The efficiency of IPOPT is reduced because derivatives are not exact, and simplifications are done in the Hessian. This has an impact in the number of iterations, and thus the time needed to solve the NLP increases. For example, the distillation case study shown above the NLP has almost 6000 more variables than here. However, since for that case we used exact first and second derivatives, the average solution time is 9.4 CPU s and here it is 50.23 CPU s. Moreover, when using exact derivatives we can set stricter tolerances; for the distillation column IPOPT converges with a tolerance of 10^{-8}, while here we had to set it to 10^{-3}.

The advanced step implementation shows no apparent degradation of performance with respect to the ideal implementation. In fact, in Figure 3 compares the values of the inputs computed with the asNMPC, ideal NMPC, and the naïve implementation that considers the computational delay of solving the NLP. Note that the ideal and asNMPC overlap. On the other hand, when we consider the delay of solving the NLP online we see that the gas load oscillates, and thus the performance of the controller is degraded. Moreover, since the power output of the gas turbine is related directly to this input, we expect that this output will also oscillate, and if the delay was much longer this could potentially destabilize the system. Moreover, the average solution time for the NLP in this simulation is 50.23 CPU seconds while the advanced step computation takes an average of 0.98 CPU seconds. This drastic increase in computational time, when compared to the previous example, it is due to two reasons: IPOPT requires more iterations to converge due to numerical inaccuracies in derivative computations, and function evaluations are

slower due to internal computation loops in the calculation of physical properties (e.g., steam tables). Nevertheless note that in this case that solving the approximate NLP in between sample times in the asNMPC implementation is still possible, and the online computational delay is reduced from almost one minute to less than one second.

In Figure 4, we show some outputs from the plant. In particular, here we show the temperatures of the steam coming out of the High Pressure (HP) and the Intermediate Pressure (IP) steam turbines. These variables have been scaled with respect to their upper bounds to protect intellectual property. However, here it is important to note that the response of the plant satisfies process constraints, and the behavior of both the ideal and asNMPC are very close. In other words, the asNMPC behaves as an ideal controller, but has a much smaller on-line computational delay, thus improving the performance of the controller.

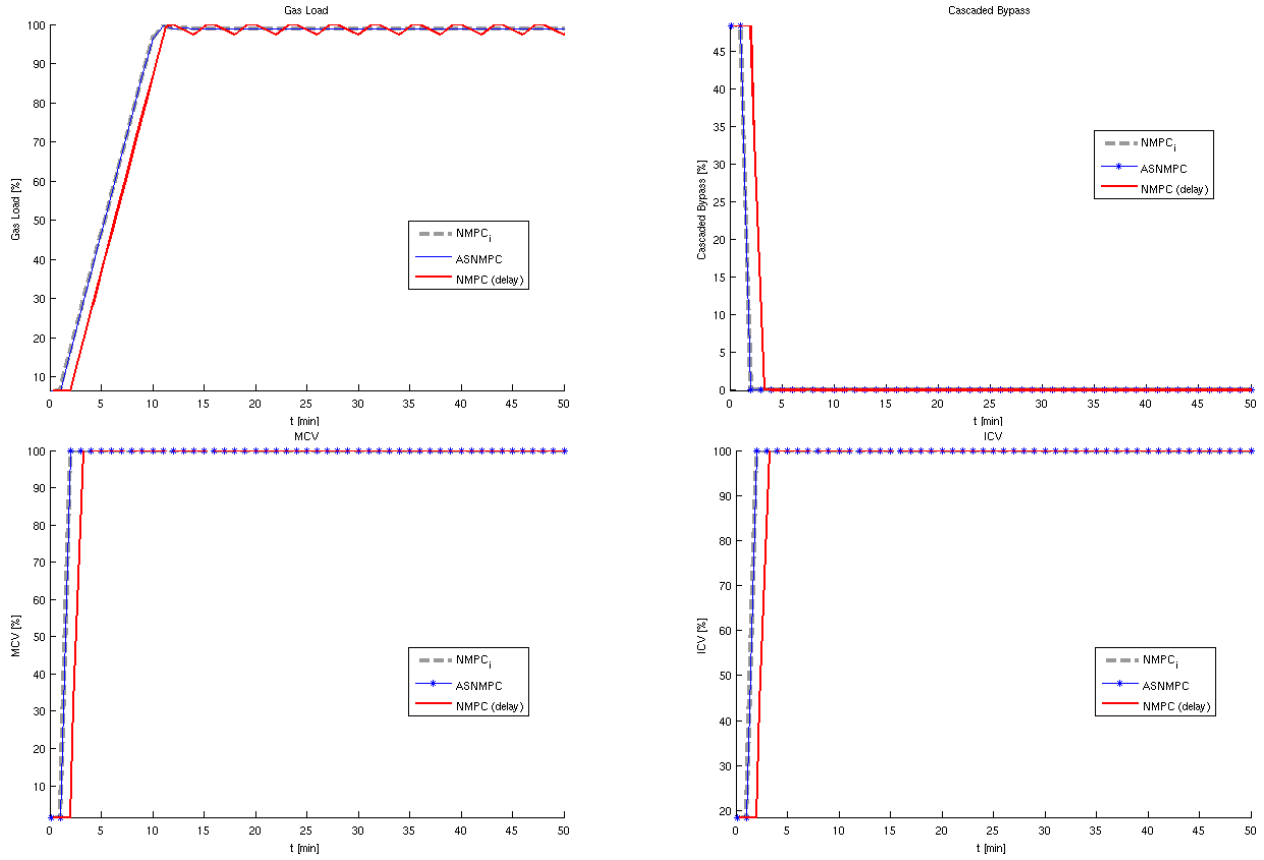


Figure 3: Input values from the closed loop simulation of the power plant.

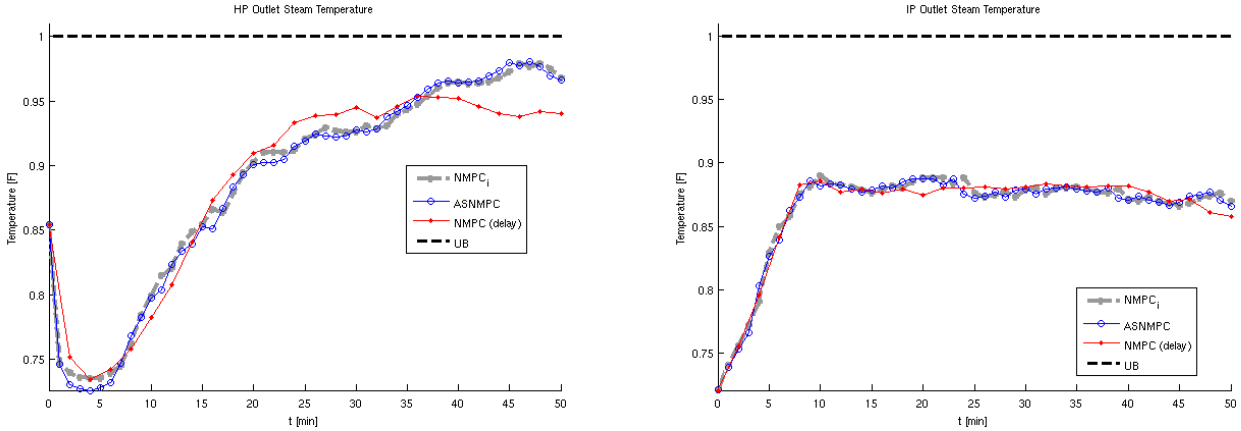


Figure 4: Performance outputs from the closed loop simulation of the power plant.

Related and Future Work

Here we consider the extension of asNMPC in a number of directions: coupling with state estimation, advanced NLP sensitivity, economic objective functions and the incorporation of uncertainty.

An advanced step approach has been developed in relation to state estimation problems. In particular, Moving Horizon Estimation is proposed as a superior constrained state estimator, and on-line implementations of it can be achieved with the aid of NLP sensitivity. For example, Zavala et al. (2008a) introduce an asMHE strategy, where the associated NLP is solved in between sample times using approximate measurement information, and then the solution is updated on-line once the true measurement is obtained. In this case, covariance information is computed via the Extended Kalman Filter (EKF) equations. Lopez-Negrete and Biegler (2011) propose a related method using a different MHE formulation. Here, the initial condition information is extracted from the smoothed solution of a previous MHE, and the covariance information is extracted directly from the optimality conditions of the NLP using the covariance-reduced Hessian relationship described in the same work.

Next, the application of NLP sensitivity can be extended to further reduce on-line computation. A promising strategy is to reconfigure the advanced step if the background computation requires more than one sampling time. Under these conditions, a more sophisticated sensitivity calculation is required based on the Schur complement of the KKT matrix. In a similar manner, the asNMPC method can be extended to deal with changes in active sets in the sensitivity calculation. Preliminary results show that there is little additional work required to obtain better performance in these directions.

In addition, while economic objectives can be directly incorporated with problem (1), it is unlikely that Lyapunov stability can generally be proved for such functions. Instead, it is likely that such problems need to be convexified in order to preserve stability properties (Diehl et al., 2011; Huang et al., 2011a). In particular, Huang et al., 2011b demonstrate a tuning strategy that allows economic terms to be regularized so that they can be bounded below by $\mathcal{K}$-functions that can be used to ensure Lyapunov stability.

Finally, the presence of model mismatch and unmeasured disturbances is typically handled through state estimation. However, such an approach has clear limitations particularly if measurements are not sufficiently informative. An alternate approach is to incorporate uncertainty within a robust for the receding horizon NLP problem. A straightforward approach to extend the NLP formulation is to approximate the uncertainty distribution through a discretization, thus leading to the solution of multi-scenario NLP problems. Although more expensive to solve, this approach leads to conservative solutions to the optimization problem but with guaranteed feasibility. Such an approach has been applied recently to NMPC strategies that include uncertainty from weather forecasting information (Zavala et al., 2009c).

Conclusions

We present an advanced approach for nonlinear MPC that allows the use of complex nonlinear models for on-line optimization, but with significant reduction in on-line computation costs. This *advanced-step* concept leads to process performance similar to ideal NMPC, but usually with 2-3 orders of magnitude less on-line computation. The key concept to this result is the use of NLP sensitivity, to compute fast approximations of NLP solutions on-line, with more detailed computations performed and updated in background. In particular, we describe the application to optimal control problems to derive NMPC formulations that are suitable for on-line implementation. These implementations are demonstrated with two large scale problems; a binary distillation column and an industrial power plant. Furthermore, we compare two different implementations of the NLP using IPOPT. In the first we compute derivative information using automatic differentiation via AMPL (i.e., exact derivatives), while in the latter these data are obtained with finite differences for first derivatives and second derivatives replaced by a Gauss-Newton approximation.

Comparing the performance the advanced step NMPC (asNMPC) implementation used for the distillation case study with that of the power plant we can point out the following. Both implementations behave very similarly to their ideal NMPC counterparts. Thus, the error in the approximated solution from the sensitivity update tends to be negligible. However, an important difference between the implementations arises. When using exact derivative information (e.g., the distillation column), we note a significant improvement in the behavior of the NLP solver. In the distillation case study IPOPT takes about 10 times less time to converge than in the power plant example. In addition, when using numerical approximations of the derivative, the solver is unable to converge to the same small tolerance used in the other implementation. Therefore, using exact derivative information and avoiding internal computation loops in the models can dramatically reduce the computational time required to solve the NLPs associated with the controller.

Current and future work deals with the extension of asNMPC in a number of directions, including performing state and parameter estimation in an efficient way, spanning longer background solution times with more sophisticated NLP sensitivity strategies, and incorporating

uncertainty using robust NLP formulations directly within NMPC.

Acknowledgments

Funding from the National Science Foundation (under Grant CBET-0756264) and from General Electric Corporation is gratefully acknowledged.

References

F. Allgoewer and A. Zheng (eds.), *Nonlinear Model Predictive Control*, Birkhaeuser, Basel (2000)

Balasubramhanya, L.; Doyle, F. J. Nonlinear Control of High Purity Distillation Column Using a Travelling-Wave Model. *AIChE J.* **1997**, *43*, 703.

R. D. Bartusiak, "NLMPC: A platform for optimal control of feed- or product-flexible manufacturing," In Assessment and Future Directions of NMPC, page 338. Springer, Berlin, 2007.

J. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM Series on Advances in Design and Control, Philadelphia, PA, USA, 2001.

L. T. Biegler, *Nonlinear Programming: Concepts, Algorithms and Applications to Chemical Engineering*, SIAM, Philadelphia, PA (2010)

B. Boley, J. Wiener. (1985). *Theory of Thermal Stresses.* Dover.

F. Casella, A. Leva. (2006). Modelling of Thermo-Hydraulic Power Generation Processes using Modelica. *Mathematical and Computer Modeling of Dynamical Systems*, Vol 12 No 1, 19-33.

E.F. Camacho and C. Bordons, "Nonlinear model predictive conrol: an introductory survey," In Assessment and Future Directions of NMPC, pages 1–16. Springer, Berlin, 2007.

D'Amato, F. (2006). Industrial Application of Model Predictive Control Solution for Power Plant Startups. *Conference on Control Applications* . Munich: IEEE.

A.M. Cervantes and L.T. Biegler, Large-Scale DAE Optimization Using a Simultaneous NLP Formulation. AIChE Journal 44(5), 1998, pp. 1038–1050.

Diehl, M., Real-Time Optimization for Large Scale Nonlinear Processes. Ph.D. thesis, Universität Heidelberg, 2001.

Diehl, M.; Bock, H.G.; Schloder, J.P.; Findeisen, R.; Nagy, Z.; and Allgower, F., Real-Time Optimization and Nonlinear Model Predictive Control of Process Governed by Differential-Algebraic Equations. Journal of Process Control 12, 2002, pp. 577–585.

M. Diehl, R. Amrit, and J. B. Rawlings. A Lyapunov function for economic optimizing model predictive control. *IEEE Trans. Auto. Cont.,* 56 (3):703-707, 2011.

S. Engell. Feedback control for optimal process operation. *J. of Process Control. 2007, 17:203.*

A. V. Fiacco. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press, New York, 1983.

R. Findeisen and F. Allgöwer, "Computational delay in nonlinear model predictive control," Proc. Int. Symp. Adv. Control of Chemical Processes, ADCHEM'03, 2004.

R. Findeisen, F. Allgoewer and L. T. Biegler, *Assessment and Future Directions of NMPC*, Springer, Berlin, 2007

R. Franke and J. Doppelhamer, "Integration of advanced model based control with industrial IT," In Assessment and Future Directions of NMPC, page 368. Springer, Berlin, 2007.

M. Grötschel, S. Krumke, and J. Rambau (eds.). *Online Optimization of Large Systems*. Springer, Berlin, 2001.

R. Huang, V. M. Zavala and L. T. Biegler, "Advanced Step Nonlinear Model Predictive Control for Air Separation Units," Journal of Process Control, 19 (2009) pp. 678-685

R. Huang, L. T. Biegler and S. C. Patwardhan, "Offset-free Advanced Step Nonlinear Model Predictive Control Based on Moving Horizon Estimation," I & EC Research, 49, 17, pp. 7882-7890 (2010)

R. Huang, E. Harinath, L. T. Biegler, "Lyapunov Stability of Economically-Oriented NMPC for Cyclic Processes," Journal of Process Control, 21, pp. 501 - 509 (2011a)

R. Huang, E. Harinath, L. T. Biegler, "Robust stability of an economically-oriented in finite horizon NMPC for cyclic processes," submitted for publication (2011b)

R. Huang, S. C. Patwardhan and L. T. Biegler, "Robust stability of nonlinear model predictive control based on extended Kalman filter," submitted for publication (2011c)

J. Kadam and W. Marquardt. "Integration of economical optimization and control for intentionally transient process operation," In Assessment and Future Directions of NMPC, page 419. Springer, Berlin, 2007.

Kumar, A. and P. Daoutidis, "Nonlinear Model Reduction and Control for High-Purity Distillation. Columns," I & EC Res., 42, 4495 (2003).

Levine, J.; Rouchon, P. "Quality Control of Binary Distillation Columns Based on Nonlinear Aggregated Models," Automatica, 27, 463 (1991)

R. Lopez-Negrete and L. T. Biegler, "A Moving Horizon Estimator for Multi-Rated Measurements Based on Nonlinear Programming Sensitivity," submitted for publication (2011)

R. Lopez-Negrete. and A. Flores-Tlacuahuac, Optimal Start-Up and Product Transition Policies of a Reactive Distillation Column. Ind. Eng. Chem. Res. 46, 2007, pp. 2092–2111.

L. Magni and R. Scattolini, "Robustness and robust design of MPC for nonlinear systems," In Assessment and Future Directions of NMPC, page 124. Springer, Berlin, 2007.

Z. K. Nagy, R. Franke, B. Mahn, and F. Allgöwer, "Real-time implementation of nonlinear model predictive control of batch processes in an industrial framework," In Assessment and Future Directions of NMPC, page 429. Springer, Berlin, 2007.

D. Odloak A.C. Zanin, M. Tvrzska de Gouvea, "Integrating realtime optimization into the model predictive controller of the FCC system," Control Engineering Practice, 10:819–831, 2002.

Ohtsuka, T., "A continuation/GMRES method for fast computation of receding horizon control" Automatica, 40, pp. 563-574 (2004)

H. Pirnay, R. Lopez-Negrete, L. T. Biegler, "Optimal Sensitivity Based on IPOPT," submitted for publication (2011)

S. Qin and T. Badgwell, "An overview of nonlinear model predictive control applications," Allgoewer, F. and A. Zheng (eds.), Nonlinear Model Predictive Control, Birkhaeuser, Basel,
2000.

C.V. Rao, J.B. Rawlings, and D.Q. Mayne, "Constrained state estimation for nonlinear discrete time systems: Stability and moving horizon approximations," IEEE TAC, 48, 2, 2003.

J.B. Rawlings and R. Amrit. Optimizing process economic performance. In *Assessment and Future Directions of Nonlinear Model Predictive Control.* Pavia, Italy, 2008.

Santos, L. O., P. Afonso, J. Castro, N. Oliveira and L. T. Biegler, "On-line Implementation of Nonlinear MPC: An experimental case study," Control Engineering Practice, 9, p. 847-852 (2001)

D. Sbarbaro T.A. Johansen, "Lyapunov-based optimizing control of nonlinear blending processes," IEEE Trans. Control Syst. Technol., 13(4):631–638, 2005.

Thomas, P. (1999). *Simulation of Industrial Processes.* Butterworth-Heinemann.

S. Timoshenko, J. Goodier (1970). *Theory of Elasticity.* McGraw-Hill.

A. Wächter and L.T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," Math. Program., 106 (1):25–57, 2006.

L. Wurth, J.B. Rawlings and W. Marquardt. Economic dynamic real-time optimization and nonlinear model predictive control. *ADCHEM, Istanbul, Turkey, 2009.*

Victor M. Zavala, Carl D. Laird, and L. T. Biegler, "Fast implementations and rigorous models: can both be accommodated in NMPC?," J. Robust and Nonlinear Control, 18:800–815, 2008a.

Zavala, V., M., C. D. Laird and L. T. Biegler, "A Fast Computational Framework for Large-Scale Moving Horizon Estimation," Journal of Process Control, 18(9), pp. 876-884 2008b.

Zavala, V., and L. T. Biegler, "The advanced-step NMPC controller: optimality, stability and robustness," Automatica 45, pp. 86-93 (2009a)

Zavala, V. M. and L. T. Biegler, "Optimization-Based Strategies for the Operation of Low-Density Polyethylene Tubular Reactors: Nonlinear Model Predictive Control," Comp. Chem. Eng., 33, pp. 1735-1746 (2009b)

Zavala, V. M., E. M. Constantinescu, M. Anitescu, and T. Krause. On-line economic optimization of energy systems using weather forecast information. *Journal of Process Control*, 19:1725-1736 (2009c).