An Overview of Nonlinear Model Predictive Control Applications

S. Joe Qin and Thomas A. Badgwell

Abstract. This paper provides an overview of nonlinear model predictive control (NMPC) applications in industry, focusing primarily on recent applications reported by NMPC vendors. A brief summary of NMPC theory is presented to highlight issues pertinent to NMPC applications. Five industrial NMPC implementations are then discussed with reference to modeling, control, optimization, and implementation issues. Results from several industrial applications are presented to illustrate the benefits possible with NMPC technology. A discussion of future needs in NMPC theory and practice is provided to conclude the paper.

1. Introduction

The term Model Predictive Control (MPC) describes a class of computer control algorithms that control the future behavior of a plant through the use of an explicit process model. At each control interval the MPC algorithm computes an open-loop sequence of manipulated variable adjustments in order to optimize future plant behavior. The first input in the optimal sequence is injected into the plant, and the entire optimization is repeated at subsequent control intervals. MPC technology was originally developed for power plant and petroleum refinery applications, but can now be found in a wide variety of manufacturing environments including chemicals, food processing, automotive, aerospace, metallurgy, and pulp and paper. Theoretical and practical issues associated with MPC technology are summarized in several recent review articles. Qin and Badgwell (1997) present a brief history of MPC technology and a survey of industrial applications in [25]. Meadows and Rawlings summarize theoretical properties of MPC algorithms in [16]. Morari and Lee discuss the past, present, and future of MPC technology in [18].

The success of MPC technology as a process control paradigm can be attributed to three important factors. First and foremost is the incorporation of an explicit process model into the control calculation. This allows the controller, in principle, to deal directly with all significant features of the process dynamics. Secondly the MPC algorithm considers plant behavior over a future horizon in time. This means that the effects of feedforward and feedback disturbances can be anticipated and removed, allowing the controller to drive the plant more closely along a desired future trajectory. Finally the MPC controller considers process input, state and output constraints directly in the control calculation. This means that constraint violations are far less likely, resulting in tighter control at the optimal constrained steady-state for the process. It is the inclusion of constraints that most clearly distinguishes MPC from other process control paradigms.

Though manufacturing processes are inherently nonlinear, the vast majority of MPC applications to date are based on linear dynamic models, the most common being step and impulse response models derived from the convolution integral. There are several potential reasons for this. Linear empirical models can be identified in a straightforward manner from process test data. In addition, most applications to date have been in refinery processing [25], where the goal is largely to maintain the process at a desired steady-state (regulator problem), rather than moving rapidly from one operating point to another (servo problem). A carefully identified linear model is sufficiently accurate in the neighborhood of a single operating point for such applications, especially if high quality feedback measurements are available. Finally, by using a linear model and a quadratic objective, the nominal MPC algorithm takes the form of a highly structured convex Quadratic Program (QP), for which reliable solution algorithms and software can easily be found [31]. This is important because the solution algorithm must converge reliably to the optimum in no more than a few tens of seconds to be useful in manufacturing applications. For these reasons, in many cases a linear model will provide the majority of the benefits possible with MPC technology.

Nevertheless, there are cases where nonlinear effects are significant enough to justify the use of NMPC technology. These include at least two broad categories of applications:

- Regulator control problems where the process is highly nonlinear and subject to large frequent disturbances (pH control, etc.)
- Servo control problems where the operating points change frequently and span a sufficiently wide range of nonlinear process dynamics (polymer manufacturing, ammonia synthesis, etc.).

It is interesting to note that some of the very first MPC papers describe ways to address nonlinear process behavior while still retaining a linear dynamic model in the control algorithm. Richalet et al. [28], for example, describe how nonlinear behavior due to load changes in a steam power plant application was handled by executing their Identification and Command (IDCOM) algorithm at a variable frequency. Prett and Gillette [24] describe applying a Dynamic Matrix Control (DMC) algorithm to control a fluid catalytic cracking unit. Model gains were obtained at each control iteration by perturbing a detailed nonlinear steady-state model.

While theoretical aspects of NMPC algorithms have been discussed quite effectively in several recent publications (see, for example, [14] and [16]), descriptions of industrial NMPC applications are much more difficult to find. A rare exception

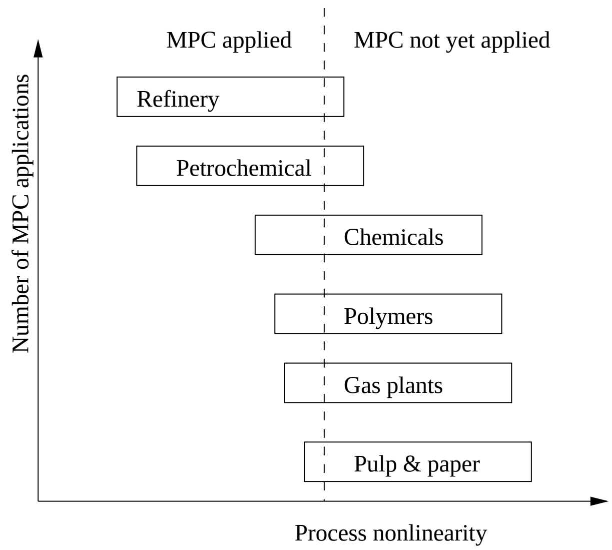

FIGURE 1. Distribution of MPC applications versus the degree of process nonlinearity.

can be found in the paper by Ogunnaike and Wright presented at the CPC-V conference [20]. This is probably due to the fact that industrial activity in NMPC applications has only begun to take off in the last few years.

In a previous survey of MPC technology [25], over 2200 commercial applications were discovered. However, almost all of these were implemented with linear models and were clustered in refinery and petrochemical processes. In preparing this paper the authors found a sizable number of NMPC applications in areas where MPC has not traditionally been applied. Figure 1 shows a rough distribution of the number of MPC applications versus the degree of process nonlinearity. MPC technology has not yet penetrated deeply into areas where process nonlinearities are strong and market demands require frequent changes in operating conditions. It is these areas that provide the greatest opportunity for NMPC applications.

The primary purpose of this paper is to provide a snapshot of the current state-of-the-art in NMPC applications. A brief summary of NMPC theory is presented to highlight what is known about closed-loop properties and to emphasize issues pertinent to NMPC applications. Then several industrial NMPC implementations are discussed in terms of modeling, control, optimization, and implementation issues. A few illustrative industrial applications are then discussed in detail. The paper concludes with a discussion of future needs and trends in NMPC theory and applications.

2. Theoretical Foundations of NMPC

For the sake of discussion we first define a simplified NMPC algorithm. The calculations necessary for an implementable MPC algorithm are described in detail in [25]. For a more thorough discussion of theoretical issues pertaining to NMPC the reader is referred to [14].

Assume that the plant to be controlled can be described by the following discrete-time, nonlinear, state-space model:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k, \mathbf{w}_k\right) && (1) \\ \mathbf{y}_k &= \mathbf{g}\left(\mathbf{x}_k\right) + \xi_k && (2) \end{aligned}$$

where $\mathbf{u}_k \in \Re^{m_u}$ is a vector of m_u process inputs or manipulated variables (MV's), $\mathbf{y}_k \in \Re^{m_y}$ is a vector of m_y process outputs or controlled variables (CV's), $\mathbf{x}_k \in \Re^n$ is a vector of n state variables, $\mathbf{v}_k \in \Re^{m_v}$ is a vector of m_v measured disturbance variables (DV's), $\mathbf{w}_k \in \Re^{m_w}$ is a vector of m_w unmeasured DV's or noise, and $\xi_k \in \Re^{m_y}$ is a vector of measurement noise.

The control problem to be solved is to compute a sequence of inputs $\{\mathbf{u}_k\}$ that will take the process from its current state $\mathbf{x}_k$ to a desired steady-state $\mathbf{x}_s$. The desired steady-state $(\mathbf{y}_s, \mathbf{x}_s, \mathbf{u}_s)$ is determined by a local steady-state optimization, which may be based on an economic objective. The optimal steady-state must be recalculated at each time step because disturbances entering the plant may change the location of the optimal operating point. Feedforward disturbances are removed by incorporating their effects into the model $\mathbf{f}$. Feedback disturbances are typically handled by assuming that a step disturbance has entered at the output and that it will remain constant for all future time. To accomplish this, a bias term that compares the current predicted output $\mathbf{y}_k$ to the current measured output $\mathbf{y}_k^m$ is computed:

$$\mathbf{b}_k = \mathbf{y}_k^m - \mathbf{y}_k \tag{3}$$

The bias $\mathbf{b}_k$ term is added to the model for use in subsequent predictions:

$$\mathbf{y}_{k+j} = \mathbf{g}\left(\mathbf{x}_{k+j}\right) + \mathbf{b}_k \tag{4}$$

The NMPC control algorithms described in this paper minimize the following dynamic objective:

$$\begin{aligned} J &= \sum\nolimits_{j=1}^{P} \|\mathbf{e}_{k+j}^y\|_{\mathbf{Q}_j}^q + \sum\nolimits_{j=0}^{M-1} \|\Delta \mathbf{u}_{k+j}\|_{\mathbf{S}_j}^q + \\ & \sum\nolimits_{j=0}^{M-1} \|\mathbf{e}_{k+j}^u\|_{\mathbf{R}_j}^q + \|\mathbf{s}\|_{\mathbf{T}}^q && (5) \end{aligned}$$

subject to a model constraint:

$$\begin{aligned} \mathbf{x}_{k+j} &= \mathbf{f}\left(\mathbf{x}_{k+j-1}, \mathbf{u}_{k+j-1}\right) && \forall\, j = 1, P \\ \mathbf{y}_{k+j} &= \mathbf{g}\left(\mathbf{x}_{k+j}\right) + \mathbf{b}_k && \forall\, j = 1, P \end{aligned}$$

and subject to inequality constraints:

$$
\begin{array}{ccccccl}
\underline{\mathbf{y}}_j - \mathbf{s} & \leq & \mathbf{y}_{k+j} & \leq & \bar{\mathbf{y}}_j + \mathbf{s} & & \forall\, j = 1, P \\
\underline{\mathbf{u}} & \leq & \mathbf{u}_{k+j} & \leq & \bar{\mathbf{u}} & & \forall\, j = 0, M-1 \\
\Delta\underline{\mathbf{u}} & \leq & \Delta\mathbf{u}_{k+j} & \leq & \Delta\bar{\mathbf{u}} & & \forall\, j = 0, M-1 \\
\mathbf{s} & \geq & 0 & & & &
\end{array}
$$

The objective function in Eq. 5 involves three conflicting contributions. Future output behavior is controlled by penalizing deviations from the desired steady-state $\mathbf{y}_s$, defined as $\mathbf{e}^y_{k+j} \equiv \mathbf{y}_{k+j} - \mathbf{y}_s$, over a *prediction horizon* of length P. Future input deviations from the desired steady-state input $\mathbf{u}_s$ are controlled using input penalties defined as $\mathbf{e}^u_{k+j} \equiv \mathbf{u}_{k+j} - \mathbf{u}_s$, over a *control horizon* of length M. Rapid input changes are penalized with a separate term involving the moves $\Delta\mathbf{u}_{k+j}$. The size of the deviations is measured by a vector norm, usually either an L_1 or L_2 norm ($q = 1, 2$). The relative importance of the objective function contributions is controlled by setting the time dependent weight matrices $\mathbf{Q}_j$, $\mathbf{S}_j$, and $\mathbf{R}_j$; these are chosen to be positive definite. The last term in the objective is used to minimize the size of output constraint violations; the weight matrix $\mathbf{T}$ is also chosen to be positive definite. The solution is a set of M input adjustments:

$$\mathbf{u}^M = (\mathbf{u}_k,\ \mathbf{u}_{k+1}, \ldots, \mathbf{u}_{k+M-1}) \tag{6}$$

The first input $\mathbf{u}_k$ is injected into the plant and the calculation is repeated at the next sample time. The fact that one can solve the closed-loop control problem through a sequence of open-loop optimizations was recognized very early in the development of optimal control theory [11]. One can view the NMPC solution as a way of turning an intractable closed-loop computation into a sequence of tractable open-loop calculations [14].

In principle the NMPC method is limited to those problems for which a global optimal solution to the dynamic optimization can be found between one control execution and the next. With a linear model and a quadratic objective, the resulting optimization problem takes the form of a highly structured convex Quadratic Program (QP) for which there exists a unique optimal solution. Several reliable standard solution codes are available for this problem. Introduction of a nonlinear model leads, in the general case, to a loss of convexity. This means that it is much more difficult to find a solution, and once found, it cannot be guaranteed to be globally optimal. For both cases, recent research efforts are aimed at exploiting the structure to improve the efficiency and reliability of solution codes [31].

It was also recognized early in the development of optimal control theory that no matter how the control problem is solved, optimality does not necessarily imply closed-loop stability, even when the model represents the true plant perfectly [6]. Under certain conditions, however, this problem can be overcome through proper construction of the NMPC algorithm. The decoupling of optimality and closed-loop stability is an issue that is still not widely appreciated by industrial practitioners.

Recent research efforts on the problem of stability of NMPC with a perfect model (the so-called *nominal stability problem*) has produced three basic solutions.

The first solution, proposed by Keerthi and Gilbert [9], involves adding a terminal state constraint to the NMPC algorithm of the form:

$$\mathbf{x}_{k+P} = \mathbf{x}_s \tag{7}$$

With such a constraint enforced, the objective function for the controller (Eq. 5) becomes a Lyapunov function for the closed loop system, leading to nominal stability. Unfortunately such a constraint may be quite difficult to satisfy in real time; exact satisfaction requires an infinite number of iterations for the numerical solution code. This motivated Michalska and Mayne [17] to seek a less stringent stability requirement. Their main idea is to define a neighborhood W around the desired steady-state $\mathbf{x}_s$ within which the system can be steered to $\mathbf{x}_s$ by a constant linear feedback controller. They add to the NMPC algorithm a constraint of the form:

$$(\mathbf{x}_{k+P} - \mathbf{x}_s) \in W \tag{8}$$

If the current state $\mathbf{x}_k$ lies outside this region then the NMPC algorithm described above is solved with constraint 8. Once inside the region W the control switches to the previously determined constant linear feedback controller. Michalska and Mayne describe this as a *dual-mode* controller.

A third solution to the nominal stability problem, described by Meadows et al. [15], involves setting the prediction horizon and control horizons to infinity $P, M \to \infty$. For this case the objective function in Eq. 5 also serves as a suitable Lyapunov function, leading to nominal stability. They demonstrate that if the initial NMPC calculation has a feasible solution, then a feasible solution exists at each subsequent time step.

These theoretical results provide a foundation upon which to build an implementable NMPC controller. The challenge of the industrial practitioner is to take these ideas to the marketplace, which means that a number of additional practical issues must be confronted. Among other things, one must choose an appropriate model form, decide how best to identify or derive the model, and develop a reliable numerical solution method. The following section describes how five NMPC vendors have addressed these issues.

3. Industrial Implementations of NMPC

In this section we describe the control algorithms used in several commercial NMPC products. Table 1 lists the products that we examined and the companies supplying them. This list is by no means exhaustive; products not described here include Treiber Controls' OPB (VanDoren, 1997), for example. However we believe that the technology sold by these companies is representative of the current state-of-the-art.

Table 2 provides information on the details of each algorithm, including the model types used, options at each step in the control calculation, and the optimization algorithm used to compute the solution. The control algorithm entries

TABLE 1. NMPC Companies and Product Names

Company	Product Name (Acronym)
Adersa	Predictive Functional Control (PFC)
Aspen Technology	Aspen Target
Continental Controls	Multivariable Control (MVC)
DOT Products	NOVA Nonlinear Controller (NOVA-NLC)
Pavilion Technologies	Process Perfecter

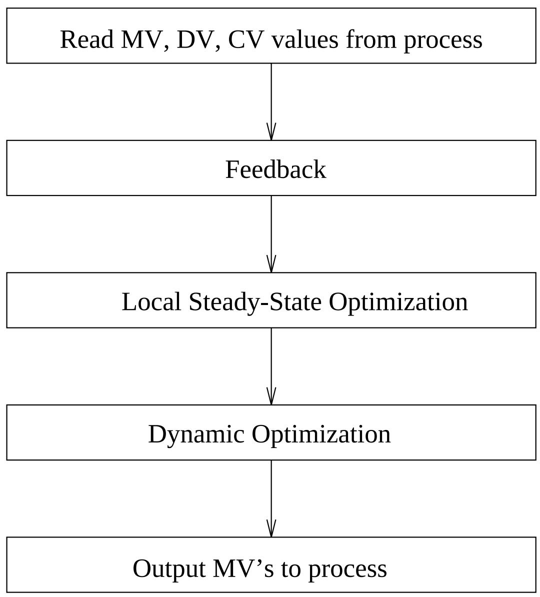

FIGURE 2. A general NMPC control calculation

correspond to the steps of the simplified NMPC control calculation illustrated in Figure 2. The following sub-sections describe these aspects in greater detail.

3.1. Models

The first issue encountered in NMPC implementation is the derivation of a dynamic nonlinear model suitable for model predictive control. In the general practice of linear MPC, the majority of dynamic models are derived from plant testing and system identification. For NMPC, however, the issue of plant testing and system identification becomes much more complicated. In this subsection we present process modeling methods used in the industrial practice of NMPC, which include system identification methods and first principles approaches.

Table 2. Comparison of Industrial NMPC Control Technology

Company	Adersa	Aspen Technology	Continental Controls	DOT Products	Pavilion Technologies
Algorithm	PFC	Aspen Target	MVC	NOVA NLC	Process Perfecter
Model Forms[1]	NSS-FP S,I,U	NSS S,I,U	SNP-IO S	NSS-FP S,I	NNN-IO S,I,U
Feedback[2]	CD,ID	CD,ID,EKF	CD	CD,EKF	CD,ID
SS Opt Obj[3]	Q[I,O]	Q[I,O]	Q[I,O]	-	Q[I,O]
SS Opt Const[4]	IH,OH	IH,OH	IH,OS	-	IH,OH
Dyn Opt Obj[5]	Q[I,O]	Q[I,O,M]	Q[I,O,M]	(Q,A)[I,O,M]	Q[I,O]
Dyn Opt Const[6]	IC,OH	IH,OH,OS	IH,OS	IH,OH,OS	IH,OH,OS
Output Traj[7]	S,Z,RT	S,Z,RT	S,Z,RT	S,Z,RT	S,Z,TW
Output Horiz[8]	FH,CP	FH,CP	FH	FH	FH
Input Param[9]	BF,SM	MM	SM	MM	MM
Sol. Method[10]	NLS	QP QP-Quick	GRG GRG2	MCNLP Nova	GRG2

[1] Model Form: (IO) Input-Output, (FP) First-Principles, (NSS) Nonlinear State-Space, (NNN) Nonlinear Neural Net, (SNP) Static Nonlinear Polynomial, (S) Stable, (I) Integrating, (U) Unstable
[2] Feedback: (CD) Constant Output Disturbance, (ID) Integrating Output Disturbance, (EKF) Extended Kalman Filter
[3] Steady-State Optimization Objective: (Q) Quadratic, (I) Inputs, (O) Outputs
[4] Steady-State Optimization Constraints: (IH) Input Hard maximum, minimum, and rate of change constraints, (OH) Output Hard maximum and minimum constraints
[5] Dynamic Optimization Objective: (Q) Quadratic, (A) One norm, (I) Inputs, (O) Outputs, (M) Input Moves
[6] Dynamic Optimization Constraints: (IH) Input Hard maximum, minimum, and rate of change constraints, (IC) Input Clipped maximum, minimum, and rate of change constraints, (OH) Output Hard maximum and minimum constraints, (OS) Output Soft maximum and minimum constraints
[7] Output Trajectory: (S) Setpoint, (Z) Zone, (RT) Reference Trajectory, (TW) Trajectory Weighting
[8] Output Horizon: (FH) Finite Horizon, (CP) Coincidence Points
[9] Input Parameterization: (SM) Single Move, (MM) Multiple Move, (BF) Basis Functions
[10] Solution Method: (NLS) Nonlinear Least Squares, (QP) Quadratic Program, (GRG) Generalized Reduced Gradient, (GD) Gradient Descent, (MCNLP) Mixed Complementarity Nonlinear Program

3.1.1. State-Space Models Because step response and impulse response models are non-parsimonic, a class of state-space model is adopted in the Aspen Target[1] product by Aspen Technology, which has a linear dynamic state equation and a

[1]This product was formerly known as NeuCOP II.

nonlinear output relation:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}_u\mathbf{u}_k + \mathbf{B}_v\mathbf{v}_k \tag{9}$$
$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) \tag{10}$$

where the MV's and DV's can have predetermined time-delays. More specifically, the output nonlinearity is modeled with a linear relation superimposed with a nonlinear neural network, that is,

$$\mathbf{g}(\mathbf{x}_k) = \mathbf{C}\mathbf{x}_k + \mathcal{N}(\mathbf{x}_k) \tag{11}$$

Since the state vector $\mathbf{x}$ is not necessarily limited to physical variables, this nonlinear model appears to be more general than measurement nonlinearity. For example, a Wiener model with a dynamic linear model followed by a static nonlinear mapping can be represented in this form. It is claimed that this type of nonlinear model can approximate any discrete time nonlinear processes with fading memory (Sentoni, et al. 1998) .

The most difficult issue in nonlinear modeling is not the selection of a nonlinear relation, but rather the selection of a robust and reliable identification algorithm. The identification algorithm discussed in Zhao, et al. (1998) builds one model for each output separately. For a process having m_y output variables, overall m_y MISO sub-models are built. The following procedure is employed to identify each sub-model from process data.

1. Specify a rough time constant for each input-output pair, then a series of first order filters or a Laguerre model is constructed for each input (Zhao, et al. 1998; Sentoni, et al. 1998). The filter states for all inputs comprise the state vector $\mathbf{x}$.
2. A static linear model is built for each output $\{y_j, j = 1, 2, \cdots, m_y\}$ using the state vector $\mathbf{x}$ as inputs using partial least squares (PLS).
3. Model reduction is then performed on the input-state-output model identified in Steps 1 and 2 using principal component analysis and internal balancing to eliminate highly collinear state variables.
4. The reduced model is rearranged in a state-space model $(\mathbf{A}, \mathbf{B})$, which is used to generate the state sequence $\{\mathbf{x}_k, k = 1, 2, \cdots, K\}$. If the model converges, go to the next step; otherwise, return to step 2.
5. A PLS model is built between the state vector $\mathbf{x}$ and the output y_j. The PLS model coefficients form the $\mathbf{C}$ matrix.
6. A neural network model is built between the PLS latent factors in the previous step and the PLS residual of the output y_j. This step generates the nonlinear static map $g_j(\mathbf{x})$. The use of the PLS latent factors instead of the state vectors is to improve the robustness of the neural network training and reduce the size of the neural network.

Besides the identification of the state-space model, a *model confidence index* (MCI) is also calculated on-line. If the MCI indicates that the neural network prediction is unreliable, the neural net nonlinear map is gradually turned off and the model calculation relies on the linear portion $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ only. Another feature of this modeling algorithm is the use of extended Kalman filters (EKF) to correct for model-plant mismatch and unmeasured disturbances (Zhao, et al. 1998). The EKF provides a bias and gain correction to the model on-line. This function replaces the constant output error feedback scheme typically employed in MPC practice.

A novel feature of the identification algorithm is that the dynamic model is built with filters and the filter states are used to predict the output variables. Due to the simplistic filter structure, each input variable has its own set of state variables, making the **A** matrix block-diagonal. This treatment assumes that each state variable is only affected by one input variable, i.e., the inputs are decoupled. For the typical case where input variables are coupled, the algorithm could generate state variables that are linearly dependent or collinear. In other words, the resulting state vector would not be a minimal realization. Nevertheless, the use of PLS algorithm makes the estimation of the **C** matrix well-conditioned. The iteration between the estimation of $\mathbf{A}, \mathbf{B}$ and **C** matrices will likely eliminate the initial error in estimating the process time constants.

Process nonlinearity is added to the model with concern for model validity using the model confidence index. When the model is used for extrapolation, only the linear portion of the model is used. The use of EKF for output error feedback is interesting; the benefit of this treatment is yet to be demonstrated.

3.1.2. Input-Output Models The MVC algorithm from Continental Controls and the Process Perfecter from Pavilion Technologies use input-output models. To simplify the system identification task, both products use a static nonlinear model superimposed upon a linear dynamic model.

Martin, et al. (1998) describe the details of the Process Perfecter modeling approach. Their presentation is in single-input-single-output form, but the concept is applicable to multi-input-multi-output models. It is assumed that the process input and output can be decomposed into a steady-state portion which obeys a nonlinear static model and a deviation portion that follows a dynamic model. For any input u_k and output y_k, the deviation variables are calculated as follows,

$$\delta u_k = u_k - u_s \tag{12}$$

$$\delta y_k = y_k - y_s \tag{13}$$

where u_s and y_s are the steady-state values for the input and output, respectively, and follow a rather general nonlinear relation:

$$y_s = h_s(u_s) \tag{14}$$

The deviation variables follow a general linear dynamic relation:

$$\delta y_k = \sum_{i=1}^{n} a_i \delta y_{k-i} + b_i \delta u_{k-i} \tag{15}$$

The identification of the linear dynamic model is based on plant test data from pulse tests, while the nonlinear static model is a neural network built from historical data. It is believed that the historical data contain rich steady-state information and plant test is needed only for the dynamic sub-model.

The use of the composite model in the control step can be described as follows. Based on the desired output target y_s^d, a nonlinear optimization program calculates the best input and output values u_s^f and y_s^f using the nonlinear static model. During the dynamic controller calculation, the nonlinear static gain is approximated by a linear interpolation of the initial and final steady-state gains,

$$K_s(u_k) = K_s^i + \frac{K_s^f - K_s^i}{u_s^f - u_s^i}\delta u_k \tag{16}$$

where u_s^i and u_s^f are the current and the next steady-state values for the input, respectively, and

$$K_s^i = \frac{dy_s}{du_s}|_{u_s^i} \tag{17}$$

$$K_s^f = \frac{dy_s}{du_s}|_{u_s^f} \tag{18}$$

which are evaluated using the static nonlinear model. Bounds on K_s^i and K_s^f can be applied. Substituting the approximate gain Eq. 16 into the linear sub-model yields,

$$\delta y_k = \sum_{i=1}^{n} a_i \delta y_{k-i} + \bar{b}_i \delta u_{k-i} + g_i \delta u_{k-i}^2 \tag{19}$$

where

$$\bar{b}_i = \frac{b_i K_s^i (1 - \sum_{j=1}^{n} a_j)}{\sum_{j=1}^{n} b_j} \tag{20}$$

$$g_i = \frac{b_i (1 - \sum_{j=1}^{n} a_j)}{\sum_{j=1}^{n} b_j} \frac{K_s^f - K_s^i}{u_s^f - u_s^i} \tag{21}$$

The purpose of this approximation is to reduce computational complexity during the control calculation.

It can be seen that the steady-state target values are calculated from a nonlinear static model, whereas the dynamic control moves are calculated based on the quadratic model in Eq. 19. However, the quadratic model coefficients (i.e., the local gain) change from one control execution to the next, simply because they are rescaled to match the local gain of the static nonlinear model. This approximation strategy can be interpreted as a successive linearization at the initial and final states followed by a linear interpolation of the linearized gains. The interpolation strategy resembles gain-scheduling, but the overall model is different from gain scheduling because of the gain re-scaling. This model makes the assumption that the process dynamics remain linear over the entire range of operation. Asymmetric dynamics (e.g., different local time constants), as a result, cannot be represented by this model.

3.1.3. First Principles Models Since empirical modeling approaches can be unreliable and require tremendous amount of experimental data, two of the vendors provide the option to use first principles models. These products usually ask the user to provide the first principles models with some kind of open equation editor, then the control algorithms can use the user-supplied models to calculate future control moves. NOVA-NLC from DOT Products and Treiber Controls' OPB (VanDoren, 1997) fall in this category. In both cases, model parameters must be estimated from plant data.

Hybrid modeling approaches that combine first principles knowledge with empirical modeling are also found in the commercial packages. The Process Perfecter uses a combination of first principles models in conjunction with empirical models (Demoro, et al. 1997). The first principles models can be steady-state balance equations, a nonlinear function of physical variables that generates another physically meaning variable, such as production rate, or simply gain directions to validate empirical models.

3.2. Output Feedback

In the face of unmeasured disturbances and model errors, some form of feedback is required to remove steady-state offset. As discussed earlier, the most common approach method for incorporating feedback into MPC algorithms involves comparing the measured and predicted process outputs [25]. The difference between the two is added to future output predictions to bias them in the direction of the measured output. This can be interpreted as assuming that an unmeasured step disturbance enters at the process output and remains constant for all future time. For the case of a linear model and no active constraints, Rawlings, et al. [27] have shown that this form of feedback leads to offset-free control. As can be seen in Table 2, all five NMPC algorithms described here provide the constant output feedback option.

When the process has a pure integrator, the constant output disturbance assumption will no longer lead to offset-free control. For this case it is common to assume that an integrating disturbance with a constraint ramp rate has entered at the output [25]. The PFC, Aspen Target, and Process Pefecter algorithms provide this feedback option.

It is well-known from linear control theory that additional knowledge about unmeasured disturbances can be exploited to provide better feedback by designing a Kalman Filter [7]. Muske and Rawlings demonstrate how this can be accomplished in the context of MPC [19]. It is interesting to note that two of the NMPC algorithms in Table 2 provide options for output feedback based on a nonlinear generalization of the Kalman Filter known as the Extended Kalman Filter (EKF) [26]. Aspen Target provides an EKF to estimate both a bias and a feedback gain. NOVA-NLC uses an EKF to develop complete state and noise estimates.

3.3. Steady-State Optimization

The PFC, Aspen Target, MVC, and Process Perfecter controllers split the control calculation into a local steady-state optimization followed by a dynamic optimization. Optimal steady-state targets are computed for each input and output; these are then passed to a dynamic optimization to compute the optimal input sequence required to move toward these targets. From Table 2 it can be seen that these calculations involve optimizing a quadratic objective that includes input and output contributions. The exception is the NOVA-NLC controller that performs the dynamic and steady-state optimizations simultaneously.

3.4. Dynamic Optimization

At the dynamic optimization level, an MPC controller must compute a set of MV adjustments that will drive the process to the steady-state operating point without violating constraints. All of the algorithms described here use a form of the objective given in Eq. 5.

The PFC controller includes only the process input and output terms in the dynamic objective, and uses constant weight matrices ($Q_j = Q$, $R_j = R$, $S_j = 0$, $q = 2$). The Aspen target and MVC products include all three terms with constant weights ($Q_j = Q$, $R_j = R$, $S_j = S$, $q = 2$). The NOVA-NLC product adds to this the option of one norms ($Q_j = Q$, $R_j = R$, $S_j = S$, $q = 1, 2$).

Instead of using a reference trajectory ($R_j = 0$, $S_j = 0$,$q = 2$), the Process Perfecter product uses a dynamic objective of the form, *trajectory weighting* that makes $\mathbf{Q}_j$ gradually increase over the horizon P. With this type of weighting, control errors at the beginning of the horizon are less important than those towards the end of the horizon, thus allowing a smoother control action.

3.5. Constraint Formulations

There are basically two types of constraints used in industrial MPC technology; hard and soft [25]. Hard constraints are those which should never be violated. Soft constraints allow the possibility of a violation; the magnitude of the violation is generally subjected to a quadratic penalty in the objective function.

All of the NMPC algorithms described here allow hard input maximum, minimum, and rate of change constraints to be defined. These are generally defined so as to keep the lower level MV controllers in a controllable range, and to prevent violent movement of the MV's at any single control execution. The PFC algorithm also accommodates maximum and minimum input acceleration constraints which are useful in mechanical servo control applications.

The Aspen Target, MVC, NOVA-NLC, and Process Perfecter algorithms perform rigorous optimizations subject to the hard input constraints. The PFC algorithm, however, enforces input hard constraints only after performing an unconstrained optimization. This is accomplished by clipping input values that exceed the input constraints. It should be noted that this method does not, in general, result in an optimal solution in the sense of satisfying the Karush-Kuhn-Tucker (KKT) conditions for optimality.

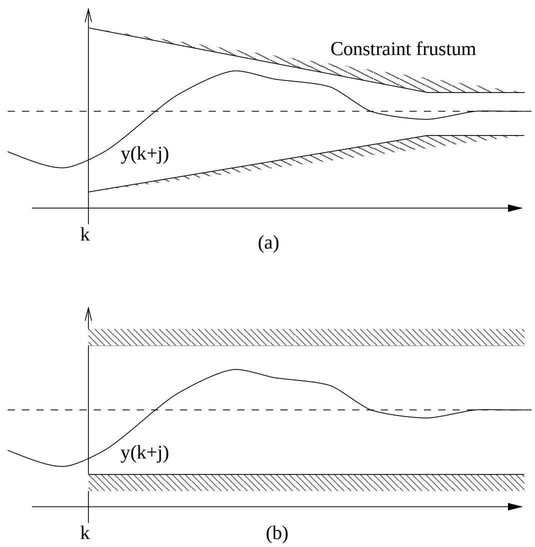

FIGURE 3. Soft constraint formulations: (a) a frustum; (b) a rectangular zone.

All control products enforce output constraints as part of the dynamic optimization, as shown in Eq. 5. The Aspen Target, NOVA-NLC, and Process Perfecter products allow options for both hard and soft output constraints. The PFC product allows only hard output constraints, while the MVC product allows only soft output constraints. The exclusive use of hard output constraints is generally avoided in MPC technology because a disturbance can cause such a controller to lose feasibility.

The Process Perfecter product applies soft constraint by using a *frustum method*, as depicted in Figure 3(a). Compared to the typical zone formulation as shown in Figure 3(b), the frustum permits a larger control error in the beginning of the horizon than in the end, but no error is allowed outside the frustum. At the end of the horizon the frustum can have a non-zero zone, instead of merging to a single line, which is determined based on the accuracy of the process model to allow for model errors.

3.6. Output Trajectories

Industrial MPC controllers use four basic options to specify future CV behavior; a setpoint, zone, reference trajectory or funnel [25]. All of the NMPC controllers described here provide the option to drive the CV's to a fixed setpoint, with deviations on both sides penalized in the objective function. In practice this type of specification is very aggressive and may lead to very large input adjustments, unless the controller is detuned in some fashion. This is particularly important when the model differs significantly from the true process. For this reason all of the controllers provide some way to detune the controller using either move suppression, a reference trajectory, or time-dependent weights.

All of the controllers also provide a CV zone control option, designed to keep the CV within a zone defined by upper and lower boundaries. A simple way to implement zone control is to define soft output constraints at the upper and lower boundaries.

The PFC, Aspen Target, MVC, and NOVA-NLC algorithms provide a CV reference trajectory option, in which the CV is required to follow a smooth path from its current value to the setpoint. Typically a first order path is defined using an operator entered closed loop time constant. In the limit of a zero time constant the reference trajectory reverts back to a pure setpoint; for this case, however, the controller would be sensitive to model mismatch unless some other strategy such as move suppression is also being used. In general, as the reference trajectory time constant increases, the controller is able to tolerate larger model mismatch.

3.7. Output Horizon and Input Parameterization

Industrial MPC controllers generally evaluate future CV behavior over a finite set of future time intervals called the *prediction horizon* [25]. This finite output horizon formulation is used by all of the algorithms discussed in this paper. The length of the horizon P is a basic tuning parameter for these controllers, and is generally set long enough to capture the steady-state effects of all computed future MV moves. This is an approximation of the infinite horizon solution for closed loop stability discussed earlier, and may explain why none of the industrial NMPC algorithms considered here include a terminal state constraint.

The PFC and Aspen Target controllers allow the option to simplify the calculation by considering only a subset of future points called *coincidence points*, so named because the desired and predicted future outputs are required to coincide at these points. A separate set of coincidence points can be defined for each output, which is useful when one output responds quickly relative to another.

Industrial MPC controllers use three different methods to parameterize the MV profile; a single move, multiple moves, and basis functions [25]. The MVC product computes a single future input value; the PFC controller also provides this option. The Aspen Target, NOVA-NLC, and Process Perfecter controllers can compute a sequence of future moves spread over a finite *control horizon*. The length of the control horizon M is another basic tuning parameter for these controllers. Better control performance is obtained as M increases, at the expense of additional computation.

The PFC controller parameterizes the input function using a set of polynomial basis functions. This allows a relatively complex input profile to be specified over a large (potentially infinite) control horizon, using a small number of unknown parameters. This may provide an advantage when controlling nonlinear systems. Choosing the family of basis functions establishes many of the features of the computed input profile; this is one way to ensure a smooth input signal, for example.

If a polynomial basis is chosen then the order can be selected so as to follow a polynomial setpoint signal with no lag. This feature is important for mechanical servo control applications.

3.8. Solution Methods

The PFC controller performs an unconstrained optimization using a nonlinear least-squares algorithm. The solution can be computed very rapidly, allowing the controller to be used for short sample time applications such as missile tracking. Some performance loss may be expected, however, since input constraints are enforced by clipping.

The Aspen Target product uses the QP-quick feasible path algorithm developed by Oliveira and Biegler [21, 22]. This algorithm employs a Newton-type solution and makes use of analytical model derivatives. Due to the sparseness of the state space model in Aspen Target, the derivative computation is straightforward. One advantage of this QP-quick algorithm is that intermediate solutions, although not optimal, are guaranteed feasible. This permits early termination of the optimization algorithm if the optimum is not found within the sampling time. Aspen Target uses the same QP-quick engine for local steady-state optimization and the dynamic MV calculation.

The MVC and Process Perfecter products use a generalized reduced gradient (GRG) code called GRG2 developed by Lasdon and Warren [10]. The NOVA-NLC product uses the NOVA optimization package, a proprietary mixed complementarity nonlinear programming code developed by DOT Products.

3.9. Implementation Issues

The following implementation issues must be considered in a commercial NMPC application.

1. *Modeling issues.* The first difficult issue is how to perform plant tests. To capture any nonlinearity in the process, extensive testing using a multi-level design is desired. This will make the testing period much longer than that of a linear plant test. To reduce the plant test period, the Process Perfecter uses historical data to build a nonlinear steady-state model and a rather short pulse test to build a linear local model. Alternatively, first principles models are used when they are available.
2. *Control and optimization issues.* Because of the use of a nonlinear model, the NMPC calculation usually involves a non-convex nonlinear program, for which the numerical solution is very challenging.
3. *Justification of the NMPC effort.* Because of the difficulties involved in NMPC implementations, the added benefit of applying NMPC has to be justified. To deal with this problem most products provide linear MPC as a back-up. In the case that NMPC is not needed or difficult to implement, linear MPC is implemented instead.
4. *Graphical user interface.* As the Windows NT operating system becomes widely acceptable in the process industry, most NMPC products provide

an easy-to-use graphical user interface which takes advantage of Windows based features.

4. Summary of NMPC Applications

The past three years have seen rapid progress in the development and application of industrial NMPC technology. Table 3 summarizes reported applications for the products considered in this paper. A total of 88 applications are reported in a wide range of application areas. The number of actual NMPC applications is likely to be significantly larger since only a handful of vendors were included in this survey, and only those applications known to the vendors were reported. It is possible that end-users have applied these products to other problems or have developed their own NMPC algorithms. Most of the applications reported here have taken place in the last three years, so the number of applications can be expected to grow rapidly in the near future.

TABLE 3. Summary of Reported NMPC Vendor Applications

Area	Adersa	Aspen Technology	Continental Controls	DOT Products	Pavilion Technologies	Total
Air and Gas			18			18
Chemicals	2		15		5	22
Food Processing					9	9
Polymers				5	15	20
Pulp & Paper					1	1
Refining					13	13
Utilities		1	2			3
Others	1		1			2
Total	3	1	36	5	43	88

While MPC applications are concentrated in refining [25], NMPC applications reported here cover a much broader range of application areas. Areas with the largest number of reported NMPC applications include chemicals, polymers, and air and gas processing.

Although not shown in the table, it has been observed that the size and scope of NMPC applications are typically much smaller than that of linear MPC applications [13]. This is likely due to the computational complexity of NMPC algorithms.

5. NMPC Application Examples

In this section we describe several successful applications of NMPC technology.

5.1. MVC: Application to an Ammonia Plant

Poe and Munsif [23] describe an application of the MVC product to control a plant producing 1450 tons/day of ammonia using natural gas and air as feedstocks. Key factors used to justify this application include:

- dynamic market supply and demand effects on natural gas price and product prices
- capacity and throughput limitations
- variations in gas feedstock rates, quality, and composition
- environmental limitations

The basic objective of the MVC controller is to maximize a profit function computed by subtracting natural gas feed and fuel gas costs from ammonia product, carbon dioxide, and steam export revenues. To achieve this the controller uses an overall economic optimization module to compute optimal steady-state targets for the plant, which are sent to seven separate dynamic controllers:

- hydraulic (steam and pressure balance) module
- primary reformer furnace temperature control module
- primary reformer riser temperature balance control module
- secondary reformer module
- shift/methanator module
- carbon dioxide removal module
- ammonia converter module

Poe and Munsif [23] describe the control modules in some detail; here we focus only on the ammonia converter module.

The ammonia converter is a standard Kellogg "quench converter" design, consisting of three nearly adiabatic catalyst beds, between which fresh feed is introduced to cool the reaction products. The converter control module manipulates the feed flow to the first bed as well as the quench flows to all three beds. This is done in order to maintain the three bed inlet temperatures at their optimal steady-state targets. Output constraints considered by the control include bed outlet temperatures and quench flow valve positions. Feedforward control is provided for changes in feed flowrate, temperature, and pressure, hydrogen/nitrogen ratio, and inert composition.

Figure 4 shows results for the second converter bed; results for the other beds were similar. The MVC controller was able to significantly reduce temperature variations at the bed inlet and outlet, allowing the average reaction temperature to be increased without violating the bed outlet constraint. Overall the plant's net fuel consumption was lowered by 1.8%, while the net production of ammonia increased by 0.7%.

5.2. Process Perfecter: Application to a Polypropylene Process

Demoro, et al. (1997) reported a successful application of NMPC to a polypropylene process using the Process Perfecter product. The polymer process is extremely nonlinear with complex reactions and involves frequent grade changes. Demoro,

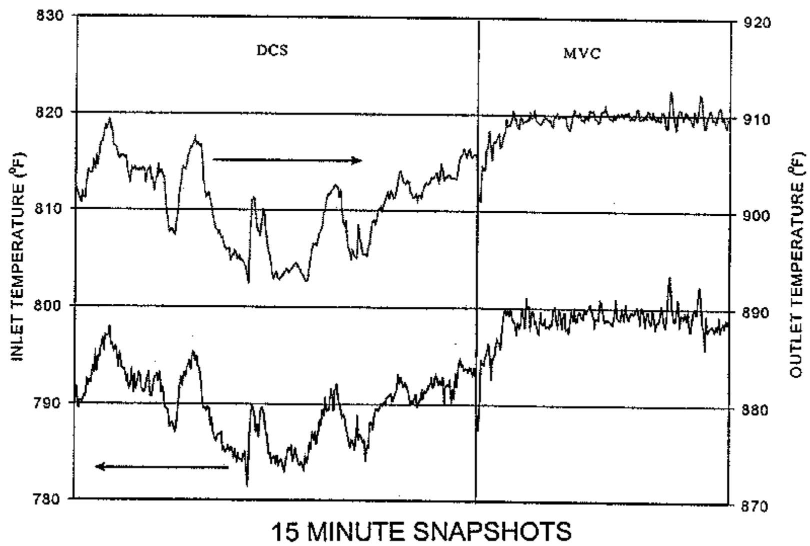

FIGURE 4. MVC Results: Ammonia converter bed 2 temperatures

et al. (1997) also compare the results of a linear MPC with those of a NMPC on the same process, demonstrating the additional benefits of NMPC for this type of processes.

The polypropylene process is a bulk liquid phase (slurry) process that carries out Ziegler-Natta polymerization in a loop reactor. The control and optimization objectives are:

- minimize product variability;
- minimize grade transition time; and
- maximize production rate.

The following three controlled variables were selected:

- production rate, which is a calculated variable from a heat balance;
- solids in slurry, which is calculated from the density of the slurry; and
- melt flow, which is measured via lab analysis using melt flow rheometry about every 4 hours.

The three selected manipulated variables are:

- catalyst flow;
- feed flow of monomers; and
- modifier concentration, which is cascaded with the modifier flow controller.

The following two feedforward disturbance variables are used in the NMPC:

- inerts concentration in the monomer feed; and
- reactor temperature.

Because none of the CV's are directly measured on-line, soft sensors were built using either first principles or neural networks from process data. The reactor production rate was calculated with unsteady state mass and energy balance models. The amount of solids in slurry was calculated from the density. The melt flow rate was estimated using a neural network, with input variables being either original sensor measurements or transformed based on physical understanding of them process, for instance, production rate and component hold-up. Since several sensors are analytical measurements which are prone to gross errors, a sensor validation module screens the data before it is used in optimization and control.

The static nonlinear process model was then built from historical data using neural networks, which also estimated the melt flow rate from lab test data. This model was integrated with a dynamic linear submodel using the Process Perfecter software. To determine the benefits of using NMPC for this process, a linear model was also built for the process to configure a linear MPC.

Demoro et al. (1997) reported three control experiments on the polypropylene process. The first one used the linear model, the second used the nonlinear model, and third used the nonlinear model in conjunction with an upper level nonlinear optimizer. In each experiment the polymer melt flow rate was changed from 30 to 35. It was observed that the linear MPC was unable to accomplish the transition within 30 minutes, since the process is highly nonlinear and the controller was detuned to achieve stability. The nonlinear MPC algorithm was able accomplish the transition very quickly. In the third experiment where an optimizer was used, both the rate maximization and grade transition were accomplished within the process constraints.

5.3. Aspen Target: Application to a Pulverized Coal Fired Boiler

Aspen Technology [4] reported an application of the Aspen Target to a pulverized coal fired boiler control. The objectives are to (i) improve boiler efficiency, (ii) reduce NOx emission, and (iii) reduce lose of ignition. The process consists of pulverizers for crushing the coal to improving firing, boilers, and a turbine. The coal burners are swirled type low-NOx burners with a boiler steam capacity of 650 tons/hour.

During coal combustion, moisture and oxygen are believed to dominate the formation of NOx and the model relation is nonlinear. Without detailed knowledge about the implementation, it is reported that the Aspen Target controller was able to reduce NOx emission by 15-25%, increase boiler efficiency by 0.1-0.3%, and decrease loss of ignition by 2%.

6. Future Needs for NMPC Technology Development

Although many academic research results are available for NMPC, many issues that affect industrial practice are as yet unresolved:

1. *Modeling approaches.* There is no systematic approach for building nonlinear dynamic models for NMPC. In the case of empirical approaches, guidelines for plant tests are needed to build a reliable model.
2. *Optimization.* Speed and the assurance of a reliable solution in real-time are major limiting factors in existing applications. Interior point methods show great promise here.
3. *Output feedback.* Most current NMPC implementations use the traditional bias correction to the model prediction based on current measurements. While this approach is meaningful for linear MPC because of the principle of superposition, it is questionable how general this approach is to nonlinear processes. Nonlinear state observers may provide an optimal approach to this issue.
4. *Justification of NMPC.* Criteria on where NMPC is needed are desirable but difficult to obtain. Benchmarks on the justification of NMPC are needed on a array of industrial processes. Unfortunately, only one such activity has been reported so far[3].
5. *Other issues.* Other issues that are applicable to linear MPC should also be of the same level of concern for NMPC, if not more [25]. These issues include multiple prioritized objective functions, determining controllable sub-processes, tuning, ill-conditioning, and fault tolerance.

7. Conclusions

We can draw the following conclusions.

- The past three years have seen rapid progress in the development and application of NMPC algorithms, with a total of 88 applications reported by the vendors included in this survey.
- The algorithms reported here differ in the simplifications used to generate a tractable control calculation; all of them, however, are based on adding a nonlinear model to a proven MPC formulation.
- None of the currently available NMPC algorithms includes the terminal state constraints or infinite prediction horizon required by control theory for nominal stability; instead they rely implicitly upon setting the prediction horizon long enough to effectively approximate an infinite horizon.
- The three most significant obstacles to NMPC applications are: nonlinear model development, state estimation, and rapid, reliable solution of the control algorithm in real time.
- Future needs for NMPC technology include development of a systematic approach for nonlinear model identification, nonlinear estimation methods, reliable numerical solution techniques, and better methods for justifying NMPC applications.

Acknowledgments

The authors would like to thank Jacques Richalet of Adersa, Hong Zhao of Aspen Technology, Humil Munsif, William Poe, and Ujjal Basu of Continental Controls, Mike Morshedi and Jeff Renfro of DOT Products, Jim Keeler, Steve Piche, and Doug Johnston of Pavilion Technologies, and Steve Treiber of Treiber Controls for their cooperation in providing information for this paper. The financial support of the National Science Foundation (grant CTS-9714950) and the Texas Advanced Technology Program (grant 003604-033) is gratefully acknowledged.

References

[1] P. Berkowitz and M. Papadopoulos. Multivariable process control method and apparatus. U.S. Patent, March 7 1995. No. 5396416.

[2] P. Berkowitz, M. Papadopoulos, L. Colwell, and M. Moran. Multivariable process control method and apparatus. U.S. Patent, Jan. 30 1996. No. 5488561.

[3] E. Demoro, C. Axelrud, D. Johnston, and G. Martin. Neural network modeling and control of polypropylene process. In *Society of Plastic Engineers Int'l Conference*, Houston, Texas, Feb. 23-26 1997.

[4] Aspen Technology, Inc. Pulverized coal fired boiler optimization and control. Aspen Technology Product Literature, 1998.

[5] Continental Controls, Inc. Production description: MVC 3.0. Houston, Texas, August 1995.

[6] R. E. Kalman. Contributions to the theory of optimal control. *Bull. Soc. Math. Mex.*, 5:102–119, 1960.

[7] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME, J. Basic Engineeering*, pages 95–108, March 1961.

[8] J. Keeler, G. Martin, G. Boe, S. Piche, U. Mathur, and D. Johnston. The process perfecter: The next step in multivariable control and optimization. Technical report, Pavilion Technologies, Inc., 1996.

[9] S. S. Keerthi and E. G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *J. Optim. Theory Appl.*, 57(2):265–293, May 1988.

[10] L. S. Lasdon and A. D. Waren. *GRG2 User's Guide*. Department of Computer and Information Science, Cleveland State University, Cleveland, Ohio, 1986.

[11] E. B. Lee and L. Markus. *Foundations of Optimal Control Theory*. John Wiley and Sons, New York, 1967.

[12] G. Martin, G. Boe, J. Keeler, D. Timmer, and J. Havener. Method and apparatus for modeling dynamic and steady-state processes for prediction, control and optimization. US Patent, 1998.

[13] G. Martin and D. Johnston. Continuous model-based optimization. In *Process Optimization Conference*, Houston, Texas, March 24-26 1998. Gulf Publishing Co. and Hydrocarbon Processing.

[14] D. Q. Mayne. Nonlinear model predictive control: An assessment. In Jeffrey C. Kantor, Carlos E. Garcia, and Brice Carnahan, editors, *Fifth International Conference on Chemical Process Control*, pages 217–231. AIChE and CACHE, 1997.

[15] Edward S. Meadows, Michael A. Henson, John W. Eaton, and James B. Rawlings. Receding horizon control and discontinuous state feedback stabilization. *Int. J. Control*, pages 1217–1229, 1995.

[16] Edward S. Meadows and James B. Rawlings. Model Predictive Control. In Michael A. Henson and Dale E. Seborg, editors, *Nonlinear Process Control*, chapter 5, pages 233–310. Prentice Hall, 1997.

[17] Hanna Michalska and David Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. Auto. Cont.*, 38(11):1623–1633, 1993.

[18] Manfred Morari and Jay H. Lee. Model Predictive Control: Past, Present and Future. In *Proceedings of* PSE/Escape '97*, Trondheim, Norway*, 1997.

[19] Kenneth R. Muske and James B. Rawlings. Model predictive control with linear models. *AIChE J.*, 39(2):262–287, 1993.

[20] B. A. Ogunnaike and R. A. Wright. Industrial applications of nonlinear control. In Jeffrey C. Kantor, Carlos E. Garcia, and Brice Carnahan, editors, *Fifth International Conference on Chemical Process Control*, pages 47–59. AIChE and CACHE, 1997.

[21] N.M.C. Oliveira and L.T. Biegler. Constraint handling and stability properties of model-predictive control. *AIChE J.*, 40:1138–1155, 1994.

[22] N.M.C. Oliveira and L.T. Biegler. An extension of Newton-type algorithms for nonlinear process control. *Automatica*, 31:281–286, 1995.

[23] William Poe and Himal Munsif. Benefits of advanced process control and economic optimization to petrochemical processes. Hydrocarbon Processing's Process Optimization Conference and Exhibition, March 1998.

[24] D. M. Prett and R. D. Gillette. Optimization and constrained multivariable control of a catalytic cracking unit. In *Proceedings of the Joint Automatic Control Conference*, 1980.

[25] S. Joe Qin and Thomas A. Badgwell. An Overview of Industrial Model Predictive Control Technology. In Jeffrey C. Kantor, Carlos E. Garcia, and Brice Carnahan, editors, *Fifth International Conference on Chemical Process Control*, pages 232–256. AIChE and CACHE, 1997.

[26] W. Fred Ramirez. *Process Control and Identification*. Academic Press, New York, New York, 1994.

[27] James B. Rawlings, Edward S. Meadows, and Ken Muske. Nonlinear model predictive control: a tutorial and survey. In *Proceedings of IFAC ADCHEM*, Japan, 1994.

[28] J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14:413–428, 1978.

[29] G.B. Sentoni, J.P. Guiver, H. Zhao, and L.T. Biegler. State space nonlinear process modeling: identification and universality. Revised for AIChE Journal, March 1998.

[30] Vance J. VanDoren. Multivariable controllers enter the mainstream. *Control Engineering*, pages 107–112, March 1997.

[31] Steven J. Wright. Applying New Optimization Algorithms to Model Predictive Control. In Jeffrey C. Kantor, Carlos E. Garcia, and Brice Carnahan, editors, *Chemical*

Process Control: Assessment and New Directions for Research, AIChE Symposium Series 316, pages 147–155. AIChE and CACHE, 1997.

[32] H. Zhao, J.P. Guiver, and G.B. Sentoni. An identification approach to nonlinear state space model for industrial multivariable model predictive control. In *American Control Conference*, Philadelphia, PA, June 21-24 1998.

S. Joe Qin, Department of Chemical Engineering, The University of Texas at Austin, Austin, TX 78712 and Thomas A. Badgwell, Chemical Engineering Department, MS-362, 6100 Main Street, Rice University, Houston, TX 77005-1892
E-mail address: `qin@che.utexas.edu and tab@rice.edu`