PERGAMON

Available online at www.sciencedirect.com

SCIENCE DIRECT

Control Engineering Practice 11 (2003) 733–764

CONTROL ENGINEERING PRACTICE

www.elsevier.com/locate/conengprac

A survey of industrial model predictive control technology

S. Joe Qin[a,*], Thomas A. Badgwell[b,1]

[a] Department of Chemical Engineering, The University of Texas at Austin, 1 Texas Lonhorns, C0400, Austin, TX 78712, USA
[b] Aspen Technology, Inc., 1293 Eldridge Parkway, Houston, TX 77077, USA

Received 8 November 2001; accepted 31 August 2002

Abstract

This paper provides an overview of commercially available model predictive control (MPC) technology, both linear and nonlinear, based primarily on data provided by MPC vendors. A brief history of industrial MPC technology is presented first, followed by results of our vendor survey of MPC control and identification technology. A general MPC control algorithm is presented, and approaches taken by each vendor for the different aspects of the calculation are described. Identification technology is reviewed to determine similarities and differences between the various approaches. MPC applications performed by each vendor are summarized by application area. The final section presents a vision of the next generation of MPC technology, with an emphasis on potential business and research opportunities.

1. Introduction

Model predictive control (MPC) refers to a class of computer control algorithms that utilize an explicit process model to predict the future response of a plant. At each control interval an MPC algorithm attempts to optimize future plant behavior by computing a sequence of future manipulated variable adjustments. The first input in the optimal sequence is then sent into the plant, and the entire calculation is repeated at subsequent control intervals. Originally developed to meet the specialized control needs of power plants and petroleum refineries, MPC technology can now be found in a wide variety of application areas including chemicals, food processing, automotive, and aerospace applications.

Several recent publications provide a good introduction to theoretical and practical issues associated with MPC technology. Rawlings (2000) provides an excellent introductory tutorial aimed at control practitioners. Allgower, Badgwell, Qin, Rawlings, and Wright (1999) present a more comprehensive overview of nonlinear MPC and moving horizon estimation, including a summary of recent theoretical developments and numerical solution techniques. Mayne, Rawlings, Rao, and Scokaert (2000) provide a comprehensive review of theoretical results on the closed-loop behavior of MPC algorithms. Notable past reviews of MPC theory include those of García, Prett, and Morari (1989); Ricker (1991); Morari and Lee (1991); Muske and Rawlings (1993), Rawlings, Meadows, and Muske (1994); Mayne (1997), and Lee and Cooley (1997). Several books on MPC have recently been published (Allgower & Zheng, 2000; Kouvaritakis & Cannon, 2001; Maciejowski, 2002).

The authors presented a survey of industrial MPC technology based on linear models at the 1996 Chemical Process Control V Conference (Qin & Badgwell, 1997), summarizing applications through 1995. We presented a review of industrial MPC applications using nonlinear models at the 1998 Nonlinear Model Predictive Control workshop held in Ascona, Switzerland (Qin and Badgwell, 2000). Froisy (1994) and Kulhavy, Lu, and Samad (2001) describe industrial MPC practice and future developments from the vendor's viewpoint. Young, Bartusiak, and Fontaine (2001), Downs (2001), and Hillestad and Andersen (1994) report development of MPC technology within operating companies. A survey of MPC technology in Japan provides a wealth of information on application issues from the point of view of MPC users (Ohshima, Ohno, & Hashimoto, 1995).

*Corresponding author. Tel.: +1-512-471-4417; fax: +1-512-471-7060.
E-mail address: qin@che.utexas.edu (S.J. Qin).
[1] At the time of the survey TAB was with Rice University.

0967-0661/02/$ - see front matter
PII: S0967-0661(02)00186-7

In recent years the MPC landscape has changed drastically, with a large increase in the number of reported applications, significant improvements in technical capability, and mergers between several of the vendor companies. The primary purpose of this paper is to present an updated, representative snapshot of commercially available MPC technology. The information reported here was collected from vendors starting in mid-1999, reflecting the status of MPC practice just prior to the new millennium, roughly 25 years after the first applications.

A brief history of MPC technology development is presented first, followed by the results of our industrial survey. Significant features of each offering are outlined and discussed. MPC applications to date by each vendor are then summarized by application area. The final section presents a view of next-generation MPC technology, emphasizing potential business and research opportunities.

2. A brief history of industrial MPC

This section presents an abbreviated history of industrial MPC technology. Fig. 1 shows an evolutionary tree for the most significant industrial MPC algorithms, illustrating their connections in a concise way. Control algorithms are emphasized here because relatively little information is available on the development of industrial identification technology. The following sub-sections describe key algorithms on the MPC evolutionary tree.

2.1. LQG

The development of modern control concepts can be traced to the work of Kalman et al. in the early 1960s (Kalman, 1960a, b). A greatly simplified description of their results will be presented here as a reference point for the discussion to come. In the discrete-time context, the process considered by Kalman and co-workers can be described by a discrete-time, linear state-space model:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{G}\mathbf{w}_k, \tag{1a}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \xi_k. \tag{1b}$$

The vector $\mathbf{u}$ represents process inputs, or manipulated variables, and vector $\mathbf{y}$ describes measured process outputs. The vector $\mathbf{x}$ represents process states to be controlled. The state disturbance $\mathbf{w}_k$ and measurement noise ξ_k are independent Gaussian noise with zero mean. The initial state $\mathbf{x}_0$ is assumed to be Gaussian with non-zero mean.

The objective function Φ to be minimized penalizes expected values of squared input and state deviations from the origin and includes separate state and input weight matrices $\mathbf{Q}$ and $\mathbf{R}$ to allow for tuning trade-offs:

$$\Phi = \mathscr{E}(J); \qquad J = \sum_{j=1}^{\infty} (\|\mathbf{x}_{k+j}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_{k+j}\|_{\mathbf{R}}^2). \tag{2}$$

The norm terms in the objective function are defined as follows:

$$\|\mathbf{x}\|_{\mathbf{Q}}^2 = \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x}. \tag{3}$$

Implicit in this formulation is the assumption that all variables are written in terms of deviations from a desired steady state. It was found that the solution to this problem, known as the *linear quadratic Gaussian* (LQG) controller, involves two separate steps. At time interval k, the output measurement $\mathbf{y}_k$ is first used to obtain an optimal state estimate $\hat{\mathbf{x}}_{k|k}$:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}\mathbf{u}_{k-1}, \tag{4a}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_f(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_{k|k-1}). \tag{4b}$$

Then the optimal input $\mathbf{u}_k$ is computed using an optimal proportional state controller:

$$\mathbf{u}_k = -\mathbf{K}_c\hat{\mathbf{x}}_{k|k}. \tag{5}$$

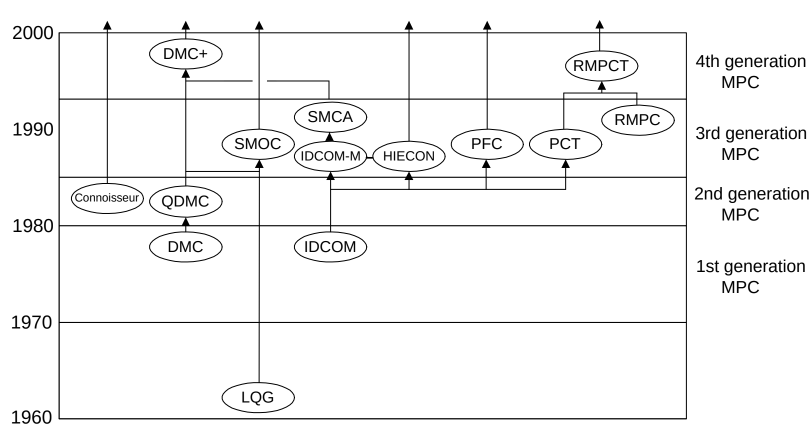

Fig. 1. Approximate genealogy of linear MPC algorithms.

Here, the notation $\hat{\mathbf{x}}_{i|j}$ refers to the state estimate at time i given information up to and including time j. The Kalman filter gain $\mathbf{K}_f$ is computed from the solution of a matrix Ricatti equation. The controller gain $\mathbf{K}_c$ can be found by constructing a dual Ricatti equation, so that the same numerical techniques and software can be used for both calculations.

The infinite prediction horizon of the LQG algorithm endows the algorithm with powerful stabilizing properties. For the case of a perfect model, it was shown to be stabilizing for any reasonable linear plant (stabilizable and the states are detectable through the quadratic criterion) as long as $\mathbf{Q}$ is positive semidefinite and $\mathbf{R}$ is positive definite.

Extensions to handle practical issues such as controlling outputs, achieving offset-free control, and computing the steady-state targets followed rapidly (Kwakernaak & Sivan, 1972). However, constraints on the process inputs, states and outputs were generally not addressed in the development of LQG theory.

LQG theory soon became a standard approach to solve control problems in a wide range of application areas. Goodwin, Graebe, and Salgado (2001) estimate that there may be thousands of real-world applications of LQG with roughly 400 patents per year based on the Kalman filter. However, it has had little impact on control technology development in the process industries. The most significant of the reasons cited for this failure include (Richalet, Rault, Testud, & Papon, 1976; García, Prett, & Morari, 1989):

- constraints;
- process nonlinearities;
- model uncertainty (robustness);
- unique performance criteria;
- cultural reasons (people, education, etc.).

It is well known that the economic operating point of a typical process unit often lies at the intersection of constraints (Prett & Gillette, 1980). A successful industrial controller for the process industries must therefore maintain the system as close as possible to constraints without violating them. In addition, process units are typically complex, nonlinear, constrained multivariable systems whose dynamic behavior changes with time due to such effects as changes in operating conditions and catalyst aging. Process units are also quite individual so that development of process models from fundamental physics and chemistry is difficult to justify economically. Indeed, the application areas where LQG theory had a more immediate impact, such as the aerospace industry, are characterized by physical systems for which it is technically and economically feasible to develop accurate fundamental models. Process units may also have unique performance criteria that are difficult to express in the LQG framework, requiring time-dependent output weights or additional logic to delineate different operating modes. However, the most significant reasons that LQG theory failed to have a strong impact may have been related to the culture of the industrial process control community at the time, in which instrument technicians and control engineers either had no exposure to LQG concepts or regarded them as impractical.

This environment led to the development, *in industry*, of a more general model based control methodology in which the dynamic optimization problem is solved online at each control execution. Process inputs are computed so as to optimize future plant behavior over a time interval known as the *prediction horizon*. In the general case any desired objective function can be used. Plant dynamics are described by an explicit process *model* which can take, in principle, any required mathematical form. Process input and output constraints are included directly in the problem formulation so that future constraint violations are anticipated and prevented. The first input of the optimal input sequence is injected into the plant and the problem is solved again at the next time interval using updated process measurements. In addition to developing more flexible control technology, new process identification technology was developed to allow quick estimation of empirical dynamic models from test data, substantially reducing the cost of model development. This new methodology for industrial process modeling and control is what we now refer to as MPC technology.

In modern processing plants the MPC controller is part of a multi-level hierarchy of control functions. This is illustrated in Fig. 2, which shows a conventional control structure on the left for Unit 1 and a MPC structure on the right for Unit 2. Similar hierarchical structures have been described by Richalet, Rault, Testud, and Papon (1978) and Prett and García (1988). At the top of the structure a plant-wide optimizer determines optimal steady-state settings for each unit in the plant. These may be sent to local optimizers at each unit which run more frequently or consider a more detailed unit model than is possible at the plant-wide level. The unit optimizer computes an optimal economic steady state and passes this to the dynamic constraint control system for implementation. The dynamic constraint control must move the plant from one constrained steady state to another while minimizing constraint violations along the way. In the conventional structure this is accomplished by using a combination of PID algorithms, lead-lag (L/L) blocks and high/low select logic. It is often difficult to translate the control requirements at this level into an appropriate conventional control structure. In the MPC methodology this combination of blocks is replaced by a single MPC controller.

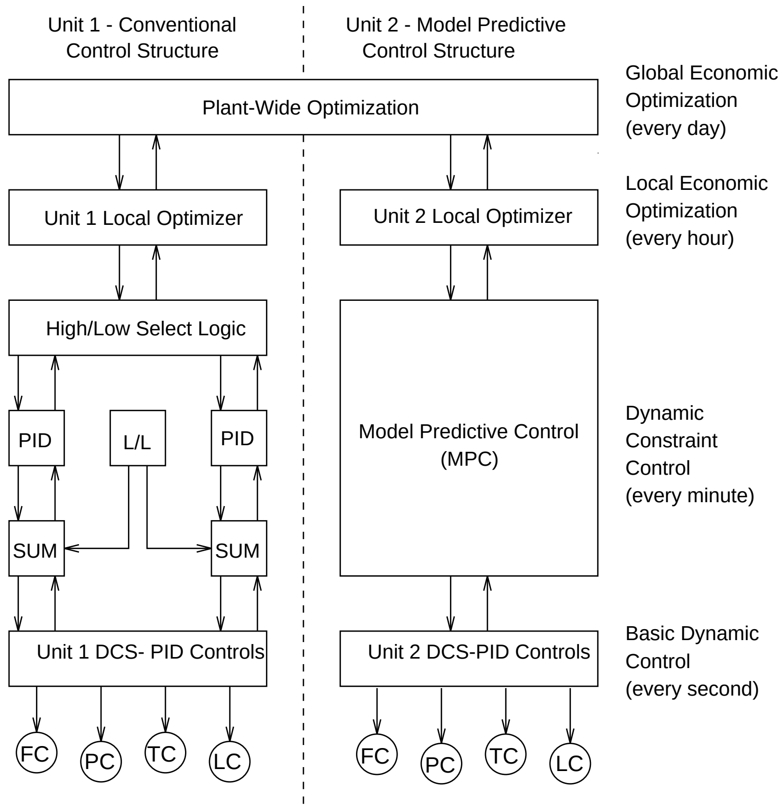

Fig. 2. Hierarchy of control system functions in a typical processing plant. Conventional structure is shown at the left; MPC structure is shown at the right.

Although the development and application of MPC technology was driven by industry, it should be noted that the idea of controlling a system by solving a sequence of open-loop dynamic optimization problems was not new. Propoi (1963), for example, described a moving horizon controller. Lee and Markus (1967) anticipated current MPC practice in their 1967 optimal control text:

> One technique for obtaining a feedback controller synthesis from knowledge of open-loop controllers is to measure the current control process state and then compute very rapidly for the open-loop control function. The first portion of this function is then used during a short time interval, after which a new measurement of the function is computed for this new measurement. The procedure is then repeated.

There is, however, a wide gap between theory and practice. The essential contribution of industry was to put these ideas into practice on operating units. Out of this experience came a fresh set of problems that has kept theoreticians busy ever since.

2.2. IDCOM

The first description of MPC control applications was presented by Richalet et al. in 1976 Conference (Richalet et al., 1976) and later summarized in 1978 *Automatica* paper (Richalet et al., 1978). They described their approach as model predictive heuristic control (MPHC). The solution software was referred to as IDCOM, an acronym for Identification and Command. The distinguishing features of the IDCOM approach are:

- impulse response model for the plant, linear in inputs or internal variables;
- quadratic performance objective over a finite prediction horizon;
- future plant output behavior specified by a reference trajectory;
- input and output constraints included in the formulation;
- optimal inputs computed using a heuristic iterative algorithm, interpreted as the dual of identification.

Richalet et al. chose an input–output representation of the process in which the process inputs influence the process outputs directly. Process inputs are divided into manipulated variables (MVs) which the controller adjusts, and disturbance variables (DVS) which are not available for control. Process outputs are referred to as controlled variables (CVs). They chose to describe the relationship between process inputs and outputs using a discrete-time finite impulse response (FIR) model. For the single input, single output (SISO) case the FIR model looks like:

$$y_{k+j} = \sum_{i=1}^{N} h_i u_{k+j-i}. \tag{6}$$

This model predicts that the output at a given time depends on a linear combination of past input values; the summation weights h_i are the impulse response coefficients. The sum is truncated at the point where past inputs no longer influence the output; this representation is therefore only possible for stable plants.

The finite impulse response was identified from plant test data using an algorithm designed to minimize the distance between the plant and model impulse responses in parameter space. The control problem was solved using the same algorithm by noting that control is the mathematical dual of identification. The iterative nature of the control algorithm allows input and output constraints to be checked as the algorithm proceeds to a solution. Because the control law is not linear and could not be expressed as a transfer function, Richalet et al. refer to it as *heuristic*. In today's context the algorithm would be referred to as a linear MPC controller.

The MPHC algorithm drives the predicted future output trajectory as closely as possible to a reference trajectory, defined as a first order path from the current output value to the desired setpoint. The speed of the desired closed-loop response is set by the time constant of the reference trajectory. This is important in practice because it provides a natural way to control the aggressiveness of the algorithm; increasing the time constant leads to a slower but more robust controller.

Richalet et al. make the important point that dynamic control must be embedded in a hierarchy of plant control functions in order to be effective. They describe four levels of control, very similar to the structure shown in Fig. 2:

- Level 3—Time and space scheduling of production.
- Level 2—Optimization of setpoints to minimize costs and ensure quality and quantity of production.
- Level 1—Dynamic multivariable control of the plant.
- Level 0—Control of ancillary systems; PID control of valves.

They point out that significant benefits do not come from simply reducing the variations of a controlled variable through better dynamic control at level 1. The real economic benefits come at level 2 where better dynamic control allows the controlled variable setpoint to be moved closer to a constraint without violating it. This argument provides the basic economic motivation for using MPC technology. This concept of a hierarchy of control functions is fundamental to advanced control applications and seems to have been followed by many practitioners. Prett and García (1988), for example, describe a very similar hierarchy.

Richalet et al. describe applications of the MPHC algorithm to a fluid catalytic cracking unit (FCCU) main fractionator column, a power plant steam generator and a poly-vinyl chloride (PVC) plant. All of these examples are constrained multivariable processes. The main fractionator example involved controlling key tray temperatures to stabilize the composition of heavy and light product streams. The controller adjusted product flowrates to compensate for inlet temperature disturbances and to maintain the level of a key internal tray. The power plant steam generator problem involved controlling the temperature and pressure of steam delivered to the turbine. This application is interesting because the process response time varied inversely with load on the system. This nonlinearity was overcome by executing the controller with a variable sample time. Benefits for the main fractionator application were reported as $150,000/yr, due to increasing the flowrate of the light product stream. Combined energy savings from two columns in the PVC plant were reported as $220,000/yr.

2.3. DMC

Engineers at Shell Oil developed their own independent MPC technology in the early 1970s, with an initial application in 1973. Cutler and Ramaker presented details of an unconstrained multivariable control algorithm which they named dynamic matrix control (DMC) at the 1979 National AIChE meeting (Cutler & Ramaker, 1979) and at the 1980 Joint Automatic Control Conference (Cutler & Ramaker, 1980). In a companion paper at the 1980 meeting Prett and Gillette (1980) described an application of DMC technology to an FCCU reactor/regenerator in which the algorithm was modified to handle nonlinearities and constraints. Neither paper discussed their process identification technology. Key features of the DMC control algorithm include:

- linear step response model for the plant;
- quadratic performance objective over a finite prediction horizon;
- future plant output behavior specified by trying to follow the setpoint as closely as possible;

- optimal inputs computed as the solution to a least-squares problem.

The linear step response model used by the DMC algorithm relates changes in a process output to a weighted sum of past input changes, referred to as input moves. For the SISO case the step response model looks like:

$$y_{k+j} = \sum_{i=1}^{N-1} s_i \, \Delta u_{k+j-i} + s_N u_{k+j-N}. \tag{7}$$

The move weights s_i are the step response coefficients. Mathematically the step response can be defined as the integral of the impulse response; given one model form the other can be easily obtained. Multiple outputs were handled by superposition. By using the step response model one can write predicted future output changes as a linear combination of future input moves. The matrix that ties the two together is the so-called *Dynamic Matrix*. Using this representation allows the optimal move vector to be computed analytically as the solution to a least-squares problem. Feedforward control is readily included in this formulation by modifying the predicted future outputs. In practice the required matrix inverse can be computed off-line to save computation. Only the first row of the final controller gain matrix needs to be stored because only the first move needs to be computed.

The objective of a DMC controller is to drive the output as close to the setpoint as possible in a least-squares sense with a penalty term on the MV moves. This results in smaller computed input moves and a less aggressive output response. As with the IDCOM reference trajectory, this technique provides a degree of robustness to model error. Move suppression factors also provide an important numerical benefit in that they can be used to directly improve the conditioning of the numerical solution.

Cutler and Ramaker showed results from a furnace temperature control application to demonstrate improved control quality using the DMC algorithm. Feedforward response of the DMC algorithm to inlet temperature changes was superior to that of a conventional PID lead/lag compensator.

In their paper Prett and Gillette (1980) described an application of DMC technology to FCCU reactor/regenerator control. Four such applications were already completed and two additional applications were underway at the time the paper was written. Prett and Gillette described additional modifications to the DMC algorithm to prevent violation of absolute input constraints. When a predicted future input came sufficiently close to an absolute constraint, an extra equation was added to the process model that would drive the input back into the feasible region. These were referred to as time variant constraints. Because the decision to add the equation had to be made on-line, the matrix inverse solution had to be recomputed at each control execution. Prett and Gillette developed a matrix tearing solution in which the original matrix inverse could be computed off-line, requiring only the matrix inverse corresponding to active time variant constraints to be computed on-line.

The initial IDCOM and DMC algorithms represent the *first generation* of MPC technology; they had an enormous impact on industrial process control and served to define the industrial MPC paradigm.

2.4. QDMC

The original IDCOM and DMC algorithms provided excellent control of unconstrained multivariable processes. Constraint handling, however, was still somewhat ad hoc. Engineers at Shell Oil addressed this weakness by posing the DMC algorithm as a quadratic program (QP) in which input and output constraints appear explicitly. Cutler et al. first described the QDMC algorithm in a 1983 AIChE conference paper (Cutler, Morshedi, & Haydel, 1983). García and Morshedi (1986) published a more comprehensive description several years later.

Key features of the QDMC algorithm include:

- linear step response model for the plant;
- quadratic performance objective over a finite prediction horizon;
- future plant output behavior specified by trying to follow the setpoint as closely as possible subject to a move suppression term;
- optimal inputs computed as the solution to a quadratic program.

García and Morshedi show how the DMC objective function can be re-written in the form of a standard QP. Future projected outputs can be related directly back to the input move vector through the dynamic matrix; this allows all input and output constraints to be collected into a matrix inequality involving the input move vector. Although the QDMC algorithm is a somewhat advanced control algorithm, the QP itself is one of the simplest possible optimization problems that one could pose. The Hessian of the QP is positive definite for linear plants and so the resulting optimization problem is convex. This means that a solution can be found readily using standard commercial optimization codes.

García and Morshedi wrapped up their paper by presenting results from a pyrolysis furnace application. The QDMC controller adjusted fuel gas pressure in three burners in order to control stream temperature at three locations in the furnace. Their test results demonstrated dynamic enforcement of input constraints and decoupling of the temperature dynamics. They reported good results on many applications within Shell on problems as large as 12×12 (12 process outputs and 12 process inputs). They stated that above all, the QDMC algorithm had proven particularly profitable in an on-line optimization environment, providing a smooth transition from one constrained operating point to another.

The QDMC algorithm can be regarded as representing a *second generation* of MPC technology, comprised of algorithms which provide a systematic way to implement input and output constraints. This was accomplished by posing the MPC problem as a QP, with the solution provided by standard QP codes.

2.5. IDCOM-M, HIECON, SMCA, and SMOC

As MPC technology gained wider acceptance, and problems tackled by MPC technology grew larger and more complex, control engineers implementing second generation MPC technology ran into other practical problems. The QDMC algorithm provided a systematic approach to incorporate hard input and output constraints, but there was no clear way to handle an infeasible solution. For example it is possible for a feedforward disturbance to lead to an infeasible QP; what should the control do to recover from infeasibility? The soft constraint formulation is not completely satisfactory because it means that all constraints will be violated to some extent, as determined by the relative weights. Clearly some output constraints are more important than others, however, and should never be violated. Would not it make sense then to shed low priority constraints in order to satisfy higher priority ones?

In practice, process inputs and outputs can be lost in real time due to signal hardware failure, valve saturation or direct operator intervention. They can just as easily come back into the control problem at any sample interval. This means that the structure of the problem and the degrees of freedom available to the control can change dynamically. This is illustrated in Fig. 3, which illustrates the shape of the process transfer function matrix for three general cases. The *square plant* case, which occurs when the plant has just as many MVs as CVs, leads to a control problem with a unique solution. In the real world, square is rare. More common is the *fat plant* case, in which there are more MVs available than there are CVs to control. The extra degrees of freedom available in this case can be put to use for additional objectives, such as moving the plant closer to an optimal operating point. When valves become saturated or lower level control action is lost, the plant may reach a condition in which there are more CVs than MVs; this is the *thin plant* case. In this situation it will not be possible to meet all of the control objectives; the control specifications must be relaxed somehow, for example by minimizing CV violations in a least-squared sense.

Fault tolerance is also an important practical issue. Rather than simply turning itself off as signals are lost, a practical MPC controller should remain online and try to make the best of the sub-plant under its control. A major barrier to achieving this goal is that a well conditioned multivariable plant may contain a number of poorly conditioned sub-plants. In practice an MPC controller must recognize and screen out poorly conditioned sub-plants before they result in erratic control action.

It also became increasingly difficult to translate control requirements into relative weights for a single objective function. Including all the required trade-offs in a single objective function means that relative weights have to be assigned to the value of output setpoint violations, output soft constraint violations, inputs moves, and optimal input target violations. For large problems it is not easy to translate control specifications into a consistent set of relative weights. In some cases it does not make sense to include these variables in the same objective function; driving the inputs to their optimal targets may lead to larger violation of output soft constraints, for example. Even when a consistent set of relative weights can be found, care must be taken to avoid scaling problems that lead to an ill-conditioned solution. Prett and García (1988) commented on this problem:

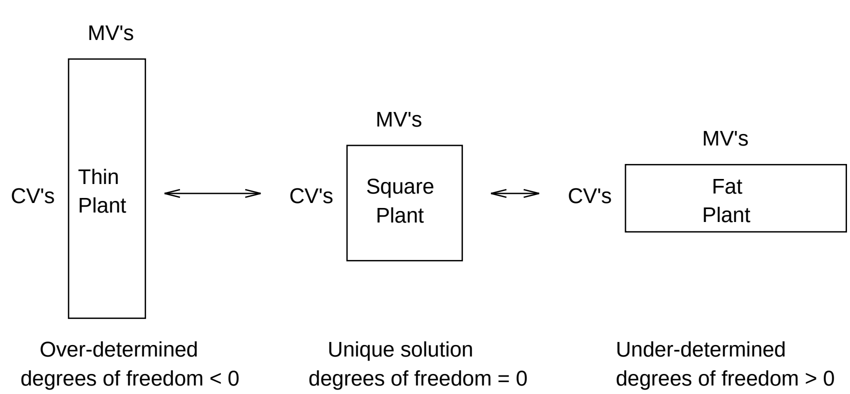

Fig. 3. Process structure determines the degrees of freedom available to the controller. Adapted from Froisy (1994).

The combination of multiple objectives into one objective (function) does not allow the designer to reflect the true performance requirements.

These issues motivated engineers at Adersa, Setpoint, Inc., and Shell (France) to develop new versions of MPC algorithms. The version marketed by Setpoint was called IDCOM-M (the M was to distinguish this from a single input/single output version called IDCOM-S), while the nearly identical Adersa version was referred to as hierarchical constraint control (HIECON). The IDCOM-M controller was first described in a paper by Grosdidier, Froisy, and Hammann (1988). A second paper presented at the 1990 AIChE conference describes an application of IDCOM-M to the Shell Fundamental Control Problem (Froisy & Matsko, 1990) and provides additional details concerning the constraint methodology. Distinguishing features of the IDCOM-M algorithm include:

- linear impulse response model of plant;
- controllability supervisor to screen out ill-conditioned plant subsets;
- multi-objective function formulation; quadratic output objective followed by a quadratic input objective;
- controls a subset of future points in time for each output, called the coincidence points, chosen from a reference trajectory;
- a single move is computed for each input;
- constraints can be hard or soft, with hard constraints ranked in order of priority.

An important distinction of the IDCOM-M algorithm is that it uses *two separate objective functions*, one for the outputs and then, if there are extra degrees of freedom, one for the inputs. A quadratic output objective function is minimized first subject to hard input constraints. Each output is driven as closely as possible to a desired value at a single point in time known as the *coincidence point*. The name comes from the fact that this is where the desired and predicted values should coincide. The desired output value comes from a first order reference trajectory that starts at the current measured value and leads smoothly to the setpoint. Each output has two basic tuning parameters; a coincidence point and a closed-loop response time, used to define the reference trajectory.

Grosdidier et al. (1988) provide simulation results for a representative FCCU regenerator control problem. The problem involves controlling flue gas composition, flue gas temperature, and regenerator bed temperature by manipulating feed oil flow, recycle oil flow and air to the regenerator. The first simulation example demonstrates how using multiple inputs can improve dynamic performance while reaching a pre-determined optimal steady-state condition. A second example demonstrates how the controller switches from controlling one output to controlling another when a measured disturbance causes a constraint violation. A third example demonstrates the need for the controllability supervisor. When an oxygen analyzer fails, the controllability supervisor is left with only flue gas temperature and regenerator bed temperature to consider. It correctly detects that controlling both would lead to an ill-conditioned problem; this is because these outputs respond in a very similar way to the inputs. Based on a pre-set priority it elects to control only the flue gas temperature. When the controllability supervisor is turned off the same simulation scenario leads to erratic and unacceptable input adjustments.

Setpoint engineers continued to improve the IDCOM-M technology, and eventually combined their identification, simulation, configuration, and control products into a single integrated offering called SMCA, for Setpoint Multivariable Control Architecture. An improved numerical solution engine allowed them to solve a sequence of separate steady-state target optimizations, providing a natural way to incorporate multiple ranked control objectives and constraints.

In the late 1980's engineers at Shell Research in France developed the Shell Multivariable Optimizing Controller (SMOC) (Marquis & Broustail, 1998; Yousfi & Tournier, 1991) which they described as a bridge between state-space and MPC algorithms. They sought to combine the constraint handling features of MPC with the richer framework for feedback offered by state-space methods. To motivate this effort they discussed the control of a hydrotreater unit with four reactor beds in series. The control system must maintain average bed temperature at a desired setpoint and hold temperature differences between the beds close to a desired profile, while preventing violation of maximum temperature limits within each reactor. Manipulated variables include the first reactor inlet temperature and quench flows between the beds. A typical MPC input/output model would view the manipulated variables as inputs and the bed temperatures as independent outputs, and a constant output disturbance would be assigned to each bed temperature. But it is clear that the bed temperatures are not independent, in that a disturbance in the first bed will ultimately affect all three downstream reactors. In addition, the controlled variables are not the process outputs but rather linear combinations thereof. State-space control design methods offer a natural solution to these problems but they do not provide an optimal way to enforce maximum bed temperature constraints.

The SMOC algorithm includes several features that are now considered essential to a "modern" MPC formulation:

- State-space models are used so that the full range of linear dynamics can be represented (stable, unstable, and integrating).

- An explicit disturbance model describes the effect of unmeasured disturbances; the constant output disturbance is simply a special case.
- A Kalman filter is used to estimate the plant states and unmeasured disturbances from output measurements.
- A distinction is introduced between *controlled variables* appearing in the control objective and *feedback variables* that are used for state estimation.
- Input and output constraints are enforced via a QP formulation.

The SMOC algorithm is nearly equivalent to solving the LQR problem with input and output constraints, except that it is still formulated on a finite horizon. As such, it does not inherit the strong stabilizing properties of the LQR algorithm. A stabilizing, infinite-horizon formulation of the constrained LQR algorithm would come only after academics began to embrace the MPC paradigm in the 1990s (Rawlings & Muske, 1993; Scokaert & Rawlings, 1998).

The IDCOM-M, HIECON, SMCA, and SMOC algorithms represent a *third generation* of MPC technology; others include the PCT algorithm sold by Profimatics, and the RMPC algorithm sold by Honeywell. This generation distinguishes between several levels of constraints (hard, soft, ranked), provides some mechanism to recover from an infeasible solution, addresses the issues resulting from a control structure that changes in real time, provides a richer set of options for feedback, and allows for a wider range of process dynamics (stable, integrating and unstable) and controller specifications.

2.6. DMC-plus and RMPCT

In the last 5 years, increased competition and the mergers of several MPC vendors have led to significant changes in the industrial MPC landscape. In late 1995 Honeywell purchased Profimatics, Inc. and formed Honeywell Hi-Spec Solutions. The RMPC algorithm offered by Honeywell was merged with the Profimatics PCT controller to create their current offering called RMPCT. In early 1996, Aspen Technology Inc. purchased both Setpoint, Inc. and DMC Corporation. This was followed by acquisition of Treiber Controls in 1998. The SMCA and DMC technologies were subsequently merged to create Aspen Technology's current DMC-plus product. DMC-plus and RMPCT are representative of the *fourth generation* MPC technology sold today, with features such as:

- Windows-based graphical user interfaces.
- Multiple optimization levels to address prioritized control objectives.
- Additional flexibility in the steady-state target optimization, including QP and economic objectives.
- Direct consideration of model uncertainty (robust control design).
- Improved identification technology based on prediction error method and sub-space ID methods.

These and other MPC algorithms currently available in the marketplace are described in greater detail in the next section.

3. Survey of MPC technology products

The industrial MPC technology has changed considerably since the publication of our first survey 5 years ago (Qin & Badgwell, 1996), which included data from five vendors: Adersa, DMC, Honeywell, Setpoint, and Treiber Controls. In late 1995 Honeywell purchased Profimatics and formed Honeywell Hi-Spec. In early 1996, Setpoint and DMC were both acquired by Aspen Technology. Two years later Aspen purchased Treiber Controls so that three of the companies in our original survey had merged into one. These mergers, continued product development, and the emergence of viable nonlinear MPC products (Qin & Badgwell, 1998) changed the MPC market enough for us to believe that an updated survey would be worthwhile.

We began collecting data for the present survey in mid-1999, when we solicited information from eight vendors in order to assess the current status of commercial MPC technology. The companies surveyed and their product names and descriptions are listed in Tables 1 and 2. In this survey we added two new companies offering linear MPC products: Shell Global Solutions (SGS) and Invensys Systems, Inc. (ISI) (Lewis, Evans, & Sandoz, 1991) in the UK. Three nonlinear MPC vendors were also included: Continental Controls,

Table 1
Companies and products included in Linear MPC technology survey

Company	Product name	Description
Adersa	HIECON	Hierarchical constraint control
	PFC	Predictive functional control
	GLIDE	Identification package
Aspen Tech	DMC-plus	Dynamic matrix control package
	DMC-plus model	Identification package
Honeywell Hi-Spec	RMPCT	Robust model predictive control technology
Shell Global Solutions	SMOC-II[a]	Shell multivariable optimizing control
Invensys	Connoisseur	Control and identification package

[a] SMOC-I was licensed to MDC Technology and Yokogawa in the past. Shell global solutions is the organization that markets the current SMOC technology.

DOT Products, and Pavilion Technologies. We believe that the technology sold by these companies is representative of the industrial state of the art; we fully recognize that we have omitted some MPC vendors from our survey, especially those who just entered the market (e.g., Fisher-Rosemount, ABB). Some companies were not asked to participate, some *chose* not to participate, and some responded too late to be included in the paper. Only companies which have documented successful MPC applications were asked to participate.

It should be noted that several companies make use of MPC technology developed in-house but were not included in the survey because they do not offer their technology externally. These MPC packages are either well known to academic researchers or not known at all for proprietary reasons. The SMOC algorithm originally developed at Shell France is included in this survey because it is now commercially available through SGS. MDC Technology, Inc. and Yokogawa had license agreements with Shell.

Initial data in this survey were collected from industrial MPC vendors using a written survey. Blank copies of the survey form are available upon request from the authors. Survey information was supplemented by published papers, product literature (DMC Corp., 1994; Setpoint Inc., 1993; Honeywell Inc., 1995), and personal communication between the authors and vendor representatives. Results of the linear MPC survey are summarized in Tables 3, 4 and 6. Nonlinear MPC survey results are summarized separately in Tables 5 and 7. While the data are provided by the vendors, the analysis is that of the authors. In presenting the survey results our intention is to highlight the important features of each algorithm; it is not our intent to determine the superiority of one product versus another. The purpose of showing the application numbers is to give a relative magnitude on how MPC is applied to different areas. The absolute numbers are not very important as they are changing fast. The numbers are not exactly comparable as the size of each MPC application can be very different. With this understanding in mind, we first discuss the overall procedure for control design and tuning. Then we describe the various model forms used for both the linear and nonlinear technology. The last two sections summarize the main features of the identification and control products sold by each vendor.

Table 2
Companies and products included in Nonlinear MPC technology survey

Company	Product name	Description
Adersa	PFC	Predictive functional control
Aspen Tech	Aspen Target	Nonlinear MPC package
Continental Controls, Inc.	MVC	Multivariable control
DOT Products	NOVA-NLC	NOVA nonlinear controller
Pavilion Technologies	Process Perfecter	Nonlinear control

3.1. Control design and tuning

The MPC control design and tuning procedure is generally described as follows (DMC Corp., 1994; Setpoint Inc., 1993; Honeywell Inc., 1995):

- From the stated control objectives, define the size of the problem, and determine the relevant CVs, MVs, and DVs.
- Test the plant systematically by varying MVs and DVs; capture and store the real-time data showing how the CVs respond.
- Derive a dynamic model either from first-principles or from the plant test data using an identification package.
- Configure the MPC controller and enter initial tuning parameters.
- Test the controller off-line using closed-loop simulation to verify the controller performance.

Table 3
Comparison of linear MPC identification technology

Product	Test protocol	Model form[a]	Est. method[b]	Uncert. bound
DMC-plus	step, PRBS	VFIR, LSS	MLS	Yes
RMPCT[c]	PRBS, step	FIR, ARX, BJ	LS, GN, PEM	Yes
AIDA[d]	PRBS, step	LSS, FIR, TF, MM	PEM-LS, GN	Yes
Glide	non-PRBS	TF	GD, GN, GM	Yes
Connoisseur	PRBS, step	FIR, ARX, MM	RLS, PEM	Yes

[a] Model Form: finite impulse response (FIR), velocity FIR (VFIR), Laplace transfer function (TF), linear state-space (LSS), auto-regressive with exogenous input (ARX), Box–Jenkins (BJ), multi-model (MM).
[b] Est. method: least-squares (LS), modified LS (MLS), recursive LS (RLS), subspace ID (SMI), Gauss–Newton (GN), prediction error method (PEM), gradient descent (GD), global method (GM).
[c] The commercial name for RMPCT is profit-controller.
[d] AIDA: advanced identification data analysis.

Table 4
Comparison of linear MPC control technology

Company	Aspen Tech	Honeywell Hi-Spec	Adersa	Adersa	Invensys	SGS
Product	DMC-plus	RMPCT	HIECON	PFC	Connois.	SMOC
Linear Model Forms[a]	FSR L,S,I,U	ARX, TF L,S,I,U	FIR L,S,I	LSS,TF,ARX L,N,S,I,U	ARX,FIR L,S,I,U	LSS L,S,I,U
Feedback[b]	CD, ID	CD, ID	CD, ID	CD, ID	CD, ID	KF
Rem Ill-cond[c]	IMS	SVT	—	—	IMS	IMS
SS Opt Obj[d]	L/Q[I, O], …, R	Q[I,O]	—	Q[I,O]	L[I,O]	Q[I,O],R
SS Opt Const[e]	IH,OS,R	IH,OH	—	IH,OH	IH,OH	IH,OS
Dyn Opt Obj[f]	Q[I,O,M],S	Q[I,O]	Q[O],Q[I]	Q[I,O],S	Q[I,O,M]	Q[I,O]
Dyn Opt Const[g]	IH	IH,OS	IH,OH,OS,R	IA,OH,OS,R	IH,OS,R	IH,OS
Output Traj[h]	S,Z	S,Z,F	S,Z,RT	S,Z,RT	S,Z	S,Z,RTB,F
Output Horiz[i]	FH	FH	FH	CP	FH	FH
Input Param[j]	MMB	MM	SM	BF	MMB	MMB
Sol. Method[k]	SLS	ASQP	ASQP	LS	ASQP	ASQP
References	Cutler and Ramaker (1979) and DMC Corp., (1994)	Honeywell Inc., (1995)	Richalet (1993)	Richalet (1993)		Marquis and Broustail (1998)

[a] Model form: Finite impulse response (FIR), finite step response (FSR), Laplace transfer function (TF), linear state-space (LSS), auto-regressive with exogenous input (ARX), linear (L), nonlinear (N), stable (S), integrating (I), unstable (U).
[b] Feedback: Constant output disturbance (CD), integrating output disturbance (ID), Kalman filter (KF).
[c] Removal of Ill-conditioning: Singular value thresholding (SVT), input move suppression (IMS).
[d] Steady-state optimization objective: linear (L), quadratic (Q), inputs (I), outputs (O), multiple sequential objectives (…), outputs ranked in order of priority (R).
[e] Steady-state optimization constraints: Input hard maximum, minimum, and rate of change constraints (IH), output hard maximum and minimum constraints (OH), constraints ranked in order of priority (R).
[f] Dynamic optimization objective: Quadratic (Q), inputs (I), Outputs (O), input moves (M), sub-optimal solution (S).
[g] Dynamic optimization constraints: Input hard maximum, minimum and rate of change constraints, IH with input acceleration constraints (IA), output hard maximum and minimum constraints (OH), output soft maximum and minimum constraints (OS), constraints ranked in order of priority (R).
[h] Output trajectory: Setpoint (S), zone (Z), reference trajectory (RT), RT bounds (RTB), funnel (F).
[i] Output horizon: Finite horizon (FH), coincidence points (CP).
[j] Input parameterization: Single move (SM), multiple move (MM), MM with blocking (MMB), basis functions (BF).
[k] Solution method: Least squares (LS), sequential LS (SLS), active set quadratic program (ASQP).

- Download the configured controller to the destination machine and test the model predictions in *open-loop* mode.
- Close the loop and refine the tuning as needed.

All of the MPC packages surveyed here provide software tools to help with the control design, model development, and closed-loop simulation steps. A significant amount of time is currently spent at the closed-loop simulation step to verify acceptable performance and robustness of the control system. Typically, tests are performed to check the regulatory and servo response of each CV, and system response to violations of major constraints is verified. The final tuning is then tested for sensitivity to model mismatch by varying the gain and dynamics of key process models. However, even the most thorough simulation testing usually cannot exhaust all possible scenarios.

Of the products surveyed here, only the RMPCT package provides robust tuning in an automatic way. This is accomplished using a min–max design procedure in which the user enters estimates of model uncertainty directly. Tuning parameters are computed to optimize performance for the worst case model mismatch. Robustness checks for the other MPC controllers are performed by closed-loop simulation.

3.2. Process models

The technical scope of an MPC product is largely defined by the form of process model that it uses. Tables 3 and 5 show that a wide variety of linear and nonlinear model forms are used in industrial MPC algorithms. It is helpful to visualize these models in a two-dimensional space, as illustrated in Fig. 4. The horizontal axis refers to the source of information used for model development, and the vertical axis designates whether the model is linear or nonlinear. The far left side of the diagram represents *empirical* models that are derived exclusively from process test data. Because empirical models mainly perform fitting between the points of a data set, they generally cannot be expected to accurately predict process behavior beyond the range of the test data used to develop them. At the far right side of the diagram lie

Table 5
Comparison of nonlinear MPC control technology

Company	Adersa	Aspen Technology	Continental Controls	DOT Products	Pavilion Technologies
Product	PFC	Aspen Target	MVC	NOVA NLC	Process Perfecter
Nonlinear model	NSS-FP	NSS-NNN	SNP-ARX	NSS-FP	NNN-ARX
Forms[a]	S,I,U	S,I,U	S	S,I	S,I,U
Feedback[b]	CD,ID	CD,ID,EKF	CD	CD	CD,ID
Rem Ill-cond[c]	—	IMS	IMS	IMS	—
SS Opt Obj[d]	Q[I,O]	Q[I,O]	Q[I,O]	—	Q[I,O]
SS Opt Const[e]	IH,OH	IH,OH	IH,OS	—	IH,OH,OS
Dyn Opt Obj[f]	Q[I,O],S	Q[I,O,M]	Q[I,O,M]	(Q,A)[I,O,M]	Q[I,O]
Dyn Opt Const[g]	IA,OH,OS,R	IH,OS-l1	IH,OS	IH,OH,OS	IH,OS
Output Traj[h]	S,Z,RT	S,Z,RT	S,Z,RT	S,Z,RTUL	S,Z,TW
Output Horiz[i]	CP	CP	FH	FH	FH
Input Param[j]	BF	MM	SM	MM	MM
Sol. Method[k]	NLS	QPKWIK	GRG2	NOVA	GRG2
References	Richalet (1993)	De Oliveira and Biegler (1994, 1995), Sentoni et al. (1998), Zhao et al. (1998), Zhao, Guiver, Neelakantan, and Biegler (1999) and Turner and Guiver (2000)	Berkowitz and Papadopoulos (1995), MVC 3.0 User Manual (1995), Berkowitz, Papadopoulos, Colwell, and Moran (1996), Poe and Munsif (1998)	Bartusiak and Fontaine (1997) and Young et al. (2001)	Demoro, Axelrud, Johnston, and Martin, 1997, Keeler, Martin, Boe, Piche, Mathur, and Johnston, (1996), Martin et al. (1998); Martin and Johnston (1998) and Piche et al. (2000)

[a] Model form: Input–output (IO), first-principles (FP), nonlinear state-space (NSS), nonlinear neural net (NNN), static nonlinear polynomial (SNP), stable (S), integrating (I), unstable (U).
[b] Feedback: Constant output disturbance (CD), integrating output disturbance (ID), extended Kalman filter (EKF).
[c] Removal of Ill-conditioning: Input move suppression (IMS).
[d] Steady-state optimization objective: Quadratic (Q), inputs (I), outputs (O).
[e] Steady-state optimization constraints: Input hard maximum, minimum, and rate of change constraints (IH), output hard maximum and minimum constraints (OH).
[f] Dynamic optimization objective: Quadratic (Q), one norm (A), inputs (I), outputs (O), input moves (M).
[g] Dynamic optimization constraints: input hard maximum, minimum and rate of change constraints (IH), IH with input acceleration constraints (IA), output hard maximum and minimum constraints (OH), output soft maximum and minimum constraints (OS), output soft constraints with l_1 exact penalty treatment (OS-l1) (De Oliveira and Biegler, 1994).
[h] Output trajectory: Setpoint (S), Zone (Z), reference trajectory (RT), upper and lower reference trajectories (RTUL), trajectory weighting (TW).
[i] Output horizon: finite horizon (FH), coincidence points (CP).
[j] Input parameterization: Single move (SM), multiple move (MM), basis functions (BF).
[k] Solution method: Nonlinear least squares (NLS), multi-step Newton method (QPKWIK) generalized reduced gradient (GRG), mixed complementarity nonlinear program (NOVA).

models derived purely from theoretical considerations such as mass and energy balances. These *first-principles* models are typically more expensive to develop, but are able to predict process behavior over a much wider range of operating conditions. In reality process models used in MPC technology are based on an effective combination of process data and theory. First principles models, for example, are typically calibrated by using process test data to estimate key parameters. Likewise, empirical models are often adjusted to account for known process physics; for example in some cases a model gain may be known to have a certain sign or value.

The MPC products surveyed here use time-invariant models that fill three quadrants of Fig. 4; nonlinear first-principles models, nonlinear empirical models, and linear empirical models. The various model forms can be derived as special cases of a general continuous-time nonlinear state-space model:

$$\dot{\mathbf{x}} = \bar{\mathbf{f}}(\mathbf{x}, \mathbf{u}, \mathbf{v}, \mathbf{w}), \tag{8a}$$

$$\mathbf{y} = \bar{\mathbf{g}}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\xi}, \tag{8b}$$

where $\mathbf{u} \in \Re^{m_u}$ is a vector of MVs, $\mathbf{y} \in \Re^{m_y}$ is a vector of CVs, $\mathbf{x} \in \Re^{n}$ is a vector of state variables, $\mathbf{v} \in \Re^{m_v}$ is a vector of measured DVs, $\mathbf{w} \in \Re^{m_w}$ is a vector of unmeasured DVs or process noise, and $\boldsymbol{\xi} \in \Re^{m_\xi}$ is a vector of measurement noise. The following sections describe each model type in more detail.

Table 6
Summary of linear MPC applications by areas (estimates based on vendor survey; estimates do not include applications by companies who have licensed vendor technology)[a]

Area	Aspen Technology	Honeywell Hi-Spec	Adersa[b]	Invensys	SGS[c]	Total
Refining	1200	480	280	25		1985
Petrochemicals	450	80	—	20		550
Chemicals	100	20	3	21		144
Pulp and paper	18	50	—	—		68
Air & Gas	—	10	—	—		10
Utility	—	10	—	4		14
Mining/Metallurgy	8	6	7	16		37
Food Processing	—	—	41	10		51
Polymer	17	—	—	—		17
Furnaces	—	—	42	3		45
Aerospace/Defense	—	—	13	—		13
Automotive	—	—	7	—		7
Unclassified	40	40	1045	26	450	1601
Total	1833	696	1438	125	450	4542
First App.	DMC:1985 IDCOM-M:1987 OPC:1987	PCT:1984 RMPCT:1991	IDCOM:1973 HIECON:1986	1984	1985	
Largest App.	603 × 283	225 × 85	—	31 × 12	—	

[a] The numbers reflect a snapshot survey conducted in mid-1999 and should not be read as static. A recent update by one vendor showed 80% increase in the number of applications.
[b] Adersa applications through January 1, 1996 are reported here. Since there are many embedded Adersa applications, it is difficult to accurately report their number or distribution. Adersa's product literature indicates over 1000 applications of PFC alone by January 1, 1996.
[c] The number of applications of SMOC includes in-house applications by Shell, which are unclassified. Therefore, only a total number is estimated here.

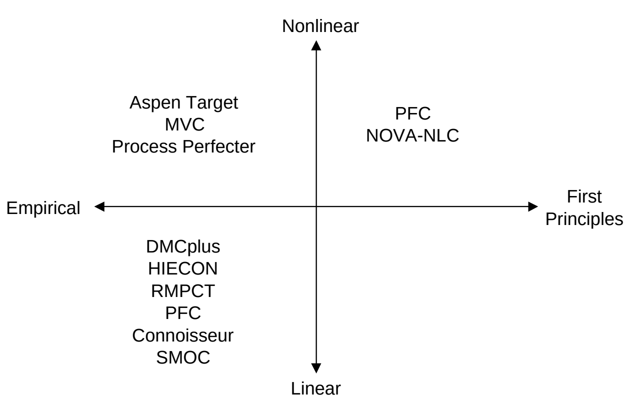

Fig. 4. Classification of model types used in industrial MPC algorithms.

3.2.1. Nonlinear first-principles models

Nonlinear first-principles models used by the NOVA-NLC algorithm are derived from mass and energy balances, and take exactly the form shown above in 8. Unknown model parameters such as heat transfer coefficients and reaction kinetic constants are either estimated off-line from test data or on-line using an extended Kalman filter (EKF). In a typical application the process model has between 10 and 100 differential algebraic equations.

The PFC algorithm can be used with several different model types. The most general of these is a discrete-time first-principles model that can be derived from 8 by integrating across the sample time:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k, \mathbf{w}_k), \tag{9a}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) + \boldsymbol{\xi}_k, \tag{9b}$$

although special care should be taken for stiff systems.

3.2.2. Linear empirical models

Linear empirical models have been used in the majority of MPC applications to date, so it is no surprise that most of the current MPC products are based on this model type. A wide variety of model forms are used, but they can all be derived from 9 by linearizing about an operating point to get:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}_u\mathbf{u}_k + \mathbf{B}_v\mathbf{v}_k + \mathbf{B}_w\mathbf{w}_k, \tag{10a}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \boldsymbol{\xi}_k. \tag{10b}$$

The SMOC and PFC algorithms can use this model form. An equivalent discrete-time transfer function model can be written in the form of a matrix fraction description (Kailath, 1980):

$$\mathbf{y}_k = [\mathbf{I} - \boldsymbol{\Phi}_y(q^{-1})]^{-1}[\boldsymbol{\Phi}_u(q^{-1})\mathbf{u}_k + \boldsymbol{\Phi}_v(q^{-1})\mathbf{v}_k + \boldsymbol{\Phi}_w(q^{-1})\mathbf{w}_k] + \boldsymbol{\xi}_k, \tag{11}$$

where q^{-1} is a backward shift operator. The output error identification approach (Ljung, 1999) minimizes the measurement error ξ_k, which results in nonlinear parameter estimation. Multiplying $[\mathbf{I} - \mathbf{\Phi}_y(q^{-1})]$ on both sides of the above equation results in an *autoregressive model with exogenous inputs* (ARX),

$$\mathbf{y}_k = \mathbf{\Phi}_y(q^{-1})\mathbf{y}_k + \mathbf{\Phi}_u(q^{-1})\mathbf{u}_k + \mathbf{\Phi}_v(q^{-1})\mathbf{v}_k + \mathbf{\Phi}_w(q^{-1})\mathbf{w}_k + \zeta_k, \tag{12a}$$

where

$$\zeta_k = [\mathbf{I} - \mathbf{\Phi}_y(q^{-1})]\xi_k. \tag{12b}$$

This model form is used by the RMPCT, PFC, and Connoisseur algorithms. The equation error identification approach minimizes ζ_k, which is colored noise even though the measurement noise ξ_k is white. The RMPCT identification algorithm also provides an option for the Box–Jenkins model, that lumps the error terms in to one term ε_k:

$$\mathbf{y}_k = [\mathbf{I} - \mathbf{\Phi}_y(q^{-1})]^{-1}[\mathbf{\Phi}_u(q^{-1})\mathbf{u}_k + \mathbf{\Phi}_v(q^{-1})\mathbf{v}_k] + [\mathbf{\Theta}_\varepsilon(q^{-1})]^{-1}\mathbf{\Phi}_\varepsilon(q^{-1})\boldsymbol{\varepsilon}_k. \tag{13}$$

For a stable system, a FIR model can be derived as an approximation to the discrete-time transfer function model 11:

$$\mathbf{y}_k = \sum_{i=1}^{N_u} \mathbf{H}_i^u \mathbf{u}_{k-i} + \sum_{i=1}^{N_v} \mathbf{H}_i^v \mathbf{v}_{k-i} + \sum_{i=1}^{N_w} \mathbf{H}_i^w \mathbf{w}_{k-i} + \xi_k. \tag{14}$$

This model form is used by the DMC-plus and HIECON algorithms. Typically the sample time is chosen so that from 30 to 120 coefficients are required to describe the full open-loop response. An equivalent velocity form is useful in identification:

$$\Delta\mathbf{y}_k = \sum_{i=1}^{N_u} \mathbf{H}_i^u \Delta\mathbf{u}_{k-i} + \sum_{i=1}^{N_v} \mathbf{H}_i^v \Delta\mathbf{v}_{k-i} + \sum_{i=1}^{N_w} \mathbf{H}_i^w \Delta\mathbf{w}_{k-i} + \Delta\xi_k. \tag{15}$$

An alternative model form is the finite step response model (FSR) (Cutler, 1983); given by:

$$\mathbf{y}_k = \sum_{i=1}^{N_u-1} \mathbf{S}_i^u \Delta\mathbf{u}_{k-i} + \mathbf{S}_{N_u}^u \mathbf{u}_{k-N_u} + \sum_{i=1}^{N_v-1} \mathbf{S}_i^v \Delta\mathbf{v}_{k-i} + \mathbf{S}_{N_v}^v \mathbf{v}_{k-N_v} + \sum_{i=1}^{N_w-1} \mathbf{S}_i^w \Delta\mathbf{w}_{k-i} + \mathbf{S}_{N_w}^v \mathbf{w}_{k-N_w} + \xi_k, \tag{16}$$

where $\mathbf{S}_j = \sum_{i=1}^{j} \mathbf{H}_i$ and $\mathbf{H}_i = \mathbf{S}_i - \mathbf{S}_{i-1}$. The FSR model is used by the DMC-plus and RMPCT algorithms. The RMPCT, Connoisseur, and PFC algorithms also provide the option to enter a Laplace transfer function model. This model form is then automatically converted to a discrete-time model form for use in the control calculations.

3.2.3. Nonlinear empirical models

Two basic types of nonlinear empirical models are used in the products that we surveyed. The Aspen Target product uses a discrete-time linear model for the state dynamics, with an output equation that includes a linear term summed with a nonlinear term:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}_u\mathbf{u}_k + \mathbf{B}_v\mathbf{v}_k + \mathbf{B}_w\mathbf{w}_k, \tag{17a}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}_u\mathbf{u}_k + N(\mathbf{x}_k, \mathbf{u}_k) + \xi_k. \tag{17b}$$

Only stable processes can be controlled by the Aspen Target product, so the eigenvalues of A must lie strictly within the unit circle. The nonlinear function N is obtained from a neural network. 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, Biegler, Guiver, & Zhao, 1998).

It is well known that neural networks can be unreliable when used to extrapolate beyond the range of the training data. The main problem is that for a sigmoidal neural network, the model derivatives fall to zero as the network extrapolates beyond the range of its training data set. The Aspen Target product deals with this problem by calculating a *model confidence index* (MCI) 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 EKF to correct for model-plant mismatch and unmeasured disturbances (Zhao, Guiver, & Sentoni, 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.

The MVC algorithm and the Process Perfecter use nonlinear input–output models. To simplify the system identification task, both products use a static nonlinear model superimposed upon a linear dynamic model.

Martin, Boe, Keeler, Timmer, and Havener (1998) and later Piche, Sayyar-Rodsari, Johnson, and Gerules (2000) 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:

$$\delta u_k = u_k - u_s, \tag{18a}$$

$$\delta y_k = y_k - y_s, \tag{18b}$$

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

$$y_s = h_s(u_s). \tag{19}$$

The deviation variables follow a second-order linear dynamic relation:

$$\delta y_k = \sum_{i=1}^{2} a_i \delta y_{k-i} + b_i \delta u_{k-i}. \tag{20}$$

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 testing is needed only for the dynamic sub-model. Bounds are enforced on the model gains in order to improve the quality of the neural network for control applications.

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,

$$K_s(u_k) = K_s^i + \frac{K_s^f - K_s^i}{u_s^f - u_s^i}\, \delta u_k, \tag{21}$$

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

$$K_s^i = \left.\frac{\mathrm{d}y_s}{\mathrm{d}u_s}\right|_{u_s^i}, \tag{22a}$$

$$K_s^f = \left.\frac{\mathrm{d}y_s}{\mathrm{d}u_s}\right|_{u_s^f}, \tag{22b}$$

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. (21) into the linear sub-model yields,

$$\delta y_k = \sum_{i=1}^{2} a_i \delta y_{k-i} + \bar{b}_i \delta u_{k-i} + g_i \delta u_{k-i}^2, \tag{23a}$$

where

$$\bar{b}_i = \frac{b_i K_s^i (1 - \sum_{j=1}^{n} a_j)}{\sum_{j=1}^{n} b_j}, \tag{23b}$$

$$g_i = \frac{b_i (1 - \sum_{j=1}^{n} a_j) K_s^f - K_s^i}{\sum_{j=1}^{n} b_j \qquad u_s^f - u_s^i}. \tag{23c}$$

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. (23a). 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.

3.3. MPC modeling and identification technology

Table 3 summarizes essential details of the modeling and identification technology sold by each vendor. Models are usually developed using process response data, obtained by stimulating the process inputs with a carefully designed test sequence. A few vendors such as Adersa and DOT Products advocate the use of first principles models.

3.3.1. Test protocols

Test signals are required to excite both steady-state (low frequency) and dynamic (medium to high frequency) dynamics of a process. A process model is then identified from the process input–output data. Many vendors believe that the plant test is the single most important phase in the implementation of DMC-plus controllers. To prepare for a formal plant test, a pre-test is usually necessary for three reasons: (i) to step each MV and adjust existing instruments and PID controllers; (ii) to obtain the time to steady state for each CV; and (iii) to obtain data for initial identification.

Most identification packages test one (or at most several) manipulated variables at a time and fix other variables at their steady state. This approach is valid as long as the process is assumed linear and superposition works. A few packages allow several MVs to change simultaneously with uncorrelated signals for different MVs. The plant test is run 24 hours a day with engineers monitoring the plant. Each MV is stepped 8 to 15 times, with the output (CV) signal to noise ratio at least six. The plant test may take up to 5–15 days, depending on the time to steady state and number of variables of the unit. Two requirements are imposed during the test: (i) no PID configuration or tuning changes are allowed;

and (ii) operators may intervene during the test to avoid critical situations, but no synchronizing or correlated moves are allowed. One may merge data from multiple test periods, which allows the user to cut out a period of data which may be corrupted with disturbances.

If the lower level PID control tuning changes significantly then it may be necessary to construct a new process model. A model is identified between the input and output, and this is combined by discrete convolution with the new input setpoint to input model.

It appears that PRBS or PRBS-like stepping signals are the primary test signals used by the identification packages. The GLIDE package uses a binary signal in which the step lengths are optimized in a dedicated way. Others use a step test with random magnitude or more random signals like the PRBS (e.g., DMC-plus-Model, Connoisseur, and RMPCT).

3.3.2. Linear model identification

The model parameter estimation approaches in the MPC products are mainly based on minimizing the following least-squares criterion,

$$J = \sum_{k=1}^{L} \|\mathbf{y}_k - \mathbf{y}_k^m\|^2, \tag{24}$$

using either an equation error approach or an output error approach (Ljung, 1987). The major difference between the equation error approach and the output error approach appears in identifying ARX or transfer function models. In the equation error approach, past output *measurements* are fed back to the model in Eqn. (12a),

$$\mathbf{y}_k = \mathbf{\Phi}_y(q^{-1})\mathbf{y}_k^m + \mathbf{\Phi}_u(q^{-1})\mathbf{u}_k + \mathbf{\Phi}_v(q^{-1})\mathbf{v}_k, \tag{25}$$

while in the output error approach, the past model output *estimates* are fed back to the model,

$$\mathbf{y}_k = \mathbf{\Phi}_y(q^{-1})\mathbf{y}_k + \mathbf{\Phi}_u(q^{-1})\mathbf{u}_k + \mathbf{\Phi}_v(q^{-1})\mathbf{v}_k. \tag{26}$$

The equation error approach produces a linear least-squares problem, but the estimates are biased even though the measurement noise $\boldsymbol{\xi}$ in Eqn. (11) is white. The output error approach is unbiased given white measurement noise. However, the ARX model parameters appear nonlinearly in the model, which requires nonlinear parameter estimation. One may also see that the equation error approach is a one-step ahead prediction approach with reference to $\mathbf{y}_k^m$, while the output error approach is a long range prediction approach since it does not use $\mathbf{y}_k^m$.

Using FIR models results in a linear-in-parameter model and an output error approach, but the estimation variance may be inflated due to possible overparametrization. In DMC-plus-Model, a least-squares method is used to estimate FIR model parameters in velocity form (Eqn. (15)). The advantage of using the velocity form is to reduce the effect of a step-like unmeasured disturbance (Cutler & Yocum, 1991). However, the velocity form is sensitive to high frequency noise. Therefore, DMC-plus-Model allows the data to be smoothed prior to fitting a model. The FIR coefficients are then converted into FSR coefficients for control. Connoisseur uses recursive least squares in a prediction error formulation to implement adaptive features.

It is worth noting that subspace model identification (SMI) (Larimore, 1990; Ljung, 1999) algorithms are now implemented in several MPC modeling algorithms. SMOC uses canonical variate analysis (CVA) to identify a state-space model which is also the model form used in the SMOC controller. Several other vendors are developing and testing their own versions of SMI algorithms.

RMPCT adopts a three-step approach: (i) identify either a Box–Jenkins model using PEM or an FIR model using Cholesky decomposition; (ii) fit the identified model to a low-order ARX model to smooth out large variance due to possible overparametrization in the FIR model. The output error approach is used to fit the ARX model via a Gauss–Newton method; and (iii) convert the ARX models into Laplace transfer functions. As an alternative to (ii) and (iii), RMPCT has the option to fit the identified model directly to a fixed structure Laplace transfer function. When the model is used in control, the transfer function models are discretized into FSR models based on a given sampling interval. The advantage of this approach is that one has the flexibility to choose different sampling intervals than that used in data collection.

Model uncertainty bounds are provided in several products such as RMPCT. In GLIDE, continuous transfer function models are identified directly by using gradient descent or Gauss–Newton approaches. Then model uncertainty is identified by a global method, which finds a region in the parameter space where the fitting criterion is less than a given value. This given value must be larger than the minimum of the criterion in order to find a feasible region.

Most linear MPC products allow the user to apply nonlinear transformations to variables that exhibit significant nonlinearity. For example, a logarithm transformation is often performed on composition variables for distillation column control.

3.3.3. Nonlinear model identification

The most difficult issue in nonlinear empirical modeling is not the selection of a nonlinear form, be it polynomial or neural network, but rather the selection of a robust and reliable identification algorithm. For example, the Aspen Target 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 submodel 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 $\{y_j,\ j = 1, 2, \ldots, m_y\}$ 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 $\{\mathbf{x}_k, k = 1, 2, \ldots, K\}$. If the model converges, i.e., no further reduction in model order, 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.

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 $\mathbf{A}$ 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 a PLS algorithm makes the estimation of the $\mathbf{C}$ matrix well-conditioned. The iteration between the estimation of $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ 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 in Aspen Target is interesting; the benefit of this treatment is yet to be demonstrated.

3.4. MPC control technology

MPC controllers are designed to drive the process from one constrained steady state to another. They may receive optimal steady-state targets from an overlying optimizer, as shown in Fig. 2, or they may compute an economically optimal operating point using an internal steady-state optimizer. The general objectives of an MPC controller, in order of importance, are:

1. prevent violation of input and output constraints;
2. drive the CVs to their steady-state optimal values (dynamic output optimization);
3. drive the MVs to their steady-state optimal values using remaining degrees of freedom (dynamic input optimization);
4. prevent excessive movement of MVs;
5. when signals and actuators fail, control as much of the plant as possible.

The translation of these objectives into a mathematical problem statement involves a number of approximations and trade-offs that define the basic character of the controller. Like any design problem there are many possible solutions; it is no surprise that there are a number of different MPC control formulations. Tables 4 and 5 summarize how each of the MPC vendors has accomplished this translation.

Fig. 5 illustrates the flow of a representative MPC calculation at each control execution. The first step is to read the current values of process inputs (DVs and MVs) and process outputs (CVs). In addition to their numerical values, each measurement carries with it a sensor status to indicate whether the sensor is functioning properly or not. Each MV will also carry information on the status of the associated lower level control function or valve; if saturated then the MV will be permitted to move in one direction only. If the MV controller is disabled then the MV cannot be used for control but can be considered a measured disturbance (DV).

The remaining steps of the calculation essentially answer three questions:

- where is the process now, and where is it heading? (output feedback);
- where should the process go to at steady state? (local steady-state optimization);
- what is the best way to drive the process to where it needs to go? (dynamic optimization).

The following sections describe these and other aspects of the MPC calculation in greater detail.

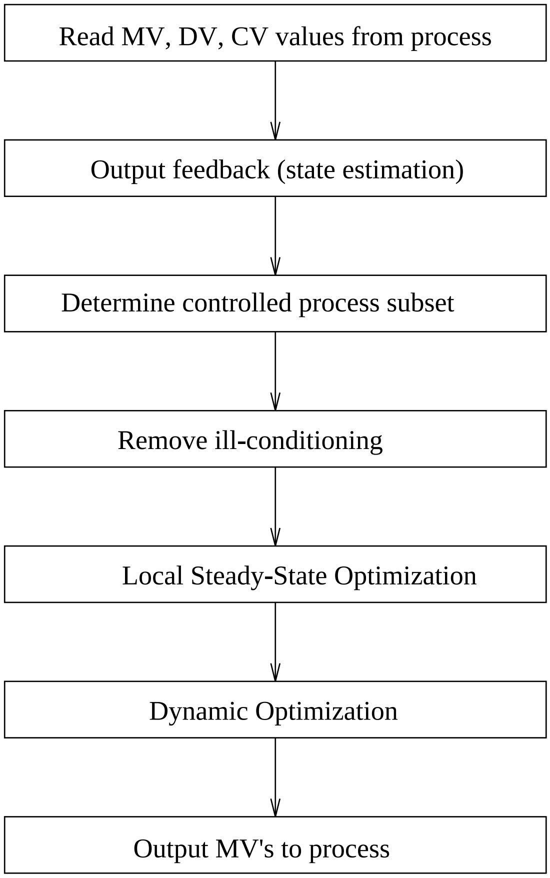

Fig. 5. Flow of MPC calculation at each control execution.

3.4.1. Output feedback (state estimation)

In the output feedback step, the controller makes use of available measurements in order to estimate the dynamic state of the system. It is at this point in the calculation where the failure to embrace LQG concepts has had the most detrimental impact on industrial MPC technology. Instead of separating the overall problem into its two natural components, state estimation and state control, most industrial MPC products do not incorporate the concept of a process state at all, and rely instead upon ad-hoc biasing schemes to incorporate feedback. This has several implications:

- Additional process output measurements (non-CVs) that could improve the accuracy of state estimation are not easily incorporated into the control structure;
- Additional effort is required to control linear combinations of process states (average bed temperature, for example), or to control unmeasured outputs;
- Unmeasured disturbance model options are severely limited;
- Ad hoc fixes must be introduced to remove steady-state offset for integrating and unstable systems;
- Measurement noise must be addressed in a sub-optimal manner.

Currently, there are only two MPC products that exploit Kalman filter technology for output feedback. As discussed earlier, the limitations listed above motivated the original development of the SMOC algorithm (Marquis & Broustail, 1998). The currently sold SMOC product still retains a Kalman filter. The Aspen Target product includes an EKF for feedback, based on a straightforward extension for nonlinear systems (Kailath, Sayed, & Hassibi, 2000).

For stable processes, the remaining MPC algorithms included in our survey use the same form of feedback, based on comparing the current measured process output $\mathbf{y}_k^m$ to the current predicted output $\mathbf{y}_k$:

$$\mathbf{b}_k = \mathbf{y}_k^m - \mathbf{y}_k. \tag{27}$$

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

$$\mathbf{y}_{k+j} = \mathbf{g}(\mathbf{x}_{k+j}, \mathbf{u}_{k+j}) + \mathbf{b}_k. \tag{28}$$

This form of feedback is equivalent to assuming that a step disturbance enters at the output and remains constant for all future time (Morari & Lee, 1991; Lee, Morari, & García, 1994). Muske and Rawlings (1993) analyzed this assumption in the context of the Kalman filter 4; the corresponding filter gain is

$$\mathbf{K}_f = [\mathbf{0}\ \mathbf{I}]^{\mathrm{T}}$$

which means no feedback for the process state estimates and identity feedback for the output disturbance.

Muske and Rawlings (1993) show that a wide variety of other disturbance models is possible. In particular they show that the constant output disturbance model leads to sluggish rejection of disturbances entering at the process input, a point also made by Shinskey in his criticism of MPC (Shinskey, 1994). This problem can be addressed directly by building an input disturbance model, similar to the internal model principle.

In the context of the discrete-time linear model 10, a general disturbance model can be written as:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}_u\mathbf{u}_k + \mathbf{B}_v\mathbf{v}_k + \mathbf{B}_w\mathbf{w}_k + \mathbf{B}_s\mathbf{s}_k, \tag{29a}$$

$$\mathbf{s}_{k+1} = \mathbf{s}_k, \tag{29b}$$

$$\mathbf{p}_{k+1} = \mathbf{p}_k, \tag{29c}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \boldsymbol{\xi}_k + \mathbf{C}_p\mathbf{p}_k, \tag{29d}$$

Here $\mathbf{s}_k \in \Re^{m_s}$ represents a constant state disturbance, and $\mathbf{p}_k \in \Re^{m_p}$ represents a constant output disturbance. The control designer does not have complete freedom in specifying the number and location of disturbances, however, because the disturbance states must be observable. In practice this means that the total number of disturbances cannot exceed the number of measured outputs ($m_s + m_p \leqslant m_y$), and that for a given process model the disturbance is not allowed to enter in certain state, or output directions (Muske, 1995). Design of sensible disturbance models for MPC applications is a ripe area for future academic research, especially for nonlinear systems.

For stable processes, the constant output disturbance model provides integral action to the controller, effectively removing steady-state offset due to disturbances and model mismatch (Rawlings et al., 1994). For integrating and unstable processes, however, the constant output disturbance model fails because the estimator contains the unstable poles of the process (Muske & Rawlings, 1993). This can be addressed easily through the use of input or state disturbance models, but most industrial MPC are forced to address the issue using ad hoc fixes, since they do not incorporate a Kalman filter. The approach used in the DMC-plus product is typical, where it is assumed that a constant integrating disturbance has entered the process, and a *rotation factor* is used to describe the fraction of prediction error due to the constant integrating disturbance, relative to the constant output disturbance. For a specific output variable that is an integrating mode, the corresponding disturbance model can be represented as:

$$x_{k+1} = x_k + b_u u_k + b_v v_k + b_w w_k + \alpha p_k, \tag{30a}$$

$$p_{k+1} = p_k, \tag{30b}$$

$$y_k = c x_k + d u_k + \xi_k + p_k, \tag{30c}$$

where α is the rotation factor. In this case it can be shown that the disturbance p_k is observable only if the rotation factor α is nonzero. While this scheme provides offset-free control for integrating systems, choice of an appropriate rotation factor is difficult in practice, and control of noisy integrating systems is often poor.

The MPC products include features to address a variety of feedback scenarios encountered in applications. In some cases the CV measurement may not be available at each control execution; this may happen, for example, when the CV measurement is provided by an analyzer. In this case a typical solution is to skip the bias update for the affected CV for a number of control intervals. A counter is provided to disable control of the CV if too many executions go by without feedback.

3.4.2. Determining the controlled sub-process

After the process state has been estimated, the controller must determine which MVs can be manipulated and which CVs should be controlled. In general, if the measurement status for a CV is good, and the operator has enabled control of the CV, then it should be controlled. An MV must meet the same criteria to be used for control; in addition, however, the lower level control functions must also be available for manipulation. If the lower level controller is saturated high or low, one can add a temporary hard MV constraint to the problem to prevent moving the MV in the wrong direction. If the lower level control function is disabled, the MV cannot be used for control. In this case it should be treated as a DV. From these decisions a controlled subprocess is defined at each control execution. In general the shape of the subprocess changes in real-time as illustrated in Fig. 3.

In applications, it is typical to include the low level control outputs (e.g., valve positions) in the MPC control formulation as additional CVs. These CVs are then forced to stay within high and low saturation limits by treating them as *range* or *zone* control variables. This serves to keep the lower level controllers away from valve constraints that would otherwise compromise their performance.

In most MPC products, sensor faults are limited to complete failure that goes beyond pre-specified control limits. Sensor faults such as significant bias and drifting that are within normal limits are generally not detected or identified in these products.

The DMC-plus, Connoisseur, and RMPCT algorithms distinguish between a *critical* CV failure and a *non-critical* CV failure. If a noncritical CV fails, the RMPCT and DMC-plus algorithms continue control action by setting the failed CV measurement to the model predicted value, which means there is no feedback for the failed CV. If the noncritical CV fails for a specified period of time, RMPCT drops this CV from the control objective function. If a critical CV fails, the DMC-plus and RMPCT controllers turn-off immediately.

3.4.3. Removal of ill-conditioning

At any particular control execution, the process encountered by the controller may require excessive input movement in order to control the outputs independently. This problem may arise, for example, if two controlled outputs respond in an almost identical way to the available inputs. Consider how difficult it would be to independently control adjacent tray temperatures in a distillation column, or to control both regenerator and cyclone temperature in an FCCU. It is important to note that this is a feature of the *process* to be controlled; any algorithm which attempts to control an ill-conditioned process must address this problem. A high condition number in the process gain matrix means that small changes in controller error will lead to large MV moves.

Although the conditioning of the full control problem will almost certainly be checked at the design phase, it is nearly impossible to check all possible sub-processes which may be encountered during future operation. It is therefore important to examine the condition number of the controlled sub-process at each control execution and to remove ill-conditioning in the model if necessary. Two strategies are currently used by MPC controllers to accomplish this task; singular value thresholding, and input move suppression.

The singular value thresholding (SVT) method used by the RMPCT controller involves decomposing the process model based on singular values using URV decomposition. Singular values below a threshold magnitude are discarded, and a process model with a lower condition number is then reassembled and used for control. The neglected singular values represent directions along which the process hardly moves even if a large MV change is applied; the SVT method gives up these directions to avoid erratic MV changes. This method solves the ill-conditioning problem at the expense of neglecting the smallest singular values. If the magnitude of the neglected singular values is small compared to the model uncertainty, it may be better to neglect them anyway. After SVT, the collinear CVs are approximated with the principal singular direction. In the case of two collinear CVs, for example, this principal direction is a weighted average of the two CVs. Note that the SVT approach is sensitive to output weighting. If one CV is weighted much more heavily than another, this CV will represent the principal singular direction.

Controllers that use input move suppression (IMS), such as the DMC-plus algorithm, provide an alternative strategy for dealing with ill-conditioning. Input move suppression factors increase the magnitude of the diagonal elements of the matrix to be inverted in the least-squares solution, directly lowering the condition number. The move suppression values can be adjusted to the point that erratic input movement is avoided for the commonly encountered sub-processes. In the case of two collinear CVs, the move suppression approach gives up a little bit on moving each CV towards its target. The move suppression solution is similar to that of the SVT solution in the sense that it tends to minimize the norm of the MV moves. In the limit of infinite move suppression the condition number becomes one for all sub-processes. It would be useful to find a set of finite move suppression factors which guarantee that all sub-processes have a condition number greater than a desired threshold value; the authors are not aware of any academic work that addresses this question.

3.4.4. *Local steady-state optimization*

Almost all of the MPC products we surveyed perform a separate local steady-state optimization at each control cycle to compute steady-state input, state, or output targets. This is necessary because the optimal targets may change at any time step due to disturbances entering the loop or operator inputs that redefine the control problem. The optimization problem is typically formulated so as to drive steady-state inputs and outputs as closely as possible to targets determined by the local economic optimization (just above the MPC algorithm in Fig. 2), without violating input and output constraints. Constant disturbances estimated in the output feedback step appear explicitly in the steady-state optimization so that they can be removed. Rao and Rawlings describe a representative formulation of the problem (Rao & Rawlings, 1999). The local steady-state optimization uses a steady-state model which may come from linearizing a comprehensive nonlinear model at each control execution or may simply be the steady-state version of the dynamic model used in the dynamic optimization.

The Connoisseur controller uses a Linear Program (LP) to do the local steady-state optimization. The distinguishing feature of an LP solution is that the optimal steady-state targets will lie at the vertex of the constraint boundary. If the constraint boundary changes frequently due to model mismatch or noise, the optimal steady-state solution may bounce around unnecessarily, leading to poor overall control performance. Typical solutions to this problem include heavily filtering the output signals and detuning the optimizer by adding a term that minimizes movement of the solution. These solutions slow down the movement of the steady-state target but also hamper rejection of process disturbances. An alternative solution based on direct incorporation of model uncertainty has been proposed, but has not yet been implemented commercially (Kassmann, Badgwell, & Hawkins, 2000).

The RMPCT, PFC, Aspen Target, MVC, and process perfecter algorithms use Quadratic Programs (QP) for the steady-state target calculation. The QP solution does not necessarily lie at the constraint boundary, so the optimal steady-state targets tend not to bounce around as much as for an LP solution.

The DMC-plus algorithm solves the local steady-state optimization problem using a sequence of LPs and/or QPs. CVs are ranked by priority such that control performance of a given CV will never be sacrificed in order to improve performance of a lower priority CV. The prediction error can be spread across a set of CVs by grouping them together at the same priority level. The calculation proceeds by optimizing the highest priority CVs first, subject to hard and soft output constraints on the same CVs and all input hard constraints. Subsequent optimizations preserve the future trajectory of high priority CVs through the use of equality constraints. Likewise inputs can be ranked in priority order so that inputs are moved sequentially towards their optimal values when extra degrees of freedom permit.

All of the products we surveyed enforce hard input constraints in the steady-state optimization. One can always find steady-state targets that are feasible with respect to the input constraints, provided they are defined in a sensible way. The same is not true, however, for output constraints. This is because if a large disturbance enters the process, it may not be possible, given the available input space, to completely remove the disturbance at steady-state. For this reason, the DMC-plus, SMOC, MVC, and Process Perfecter products provide for soft output constraints in the steady-state optimization. These allow for some violation of an output constraint, but the magnitude of the violation is minimized in the objective function. The remaining products may be subject to steady-state infeasibility problems unless other precautions are taken.

Similar infeasibility problems may arise due to failure to zero out integrating modes of the process at steady state. The DMC-plus algorithm, for example, includes an explicit constraint that forces integrating (ramp) CV variables to line out at steady state. If this constraint is violated, due perhaps to a large disturbance entering the loop, the DMC-plus product will shut off with a warning message. Other products most likely address this issue in a similar way. An alternative solution is to penalize the violation using a soft constraint, so that the controller will try its best to line out the integrating CVs at steady state without shutting off. If the disturbance diminishes, or the operator provides more room with the input constraints, it is possible that the next steady-state calculation will be feasible.

The NOVA-NLC controller does not perform a separate steady-state optimization; the steady-state and dynamic optimizations are performed simultaneously. This may improve controller performance as long as the steady-state and dynamic objectives do not conflict. For example, with separate objectives it is possible that a steady-state target may be computed that cannot be achieved, due to constraints that appear only in the dynamic optimization. Combining the objectives leads to a more complex numerical problem, however.

3.4.5. Dynamic optimization

At the dynamic optimization level, an MPC controller must compute a set of MV adjustments that will drive the process to the desired steady-state operating point without violating constraints. All of the MPC products we surveyed can be described (approximately) as minimizing the following dynamic objective function:

$$J(\mathbf{u}^M) = \sum_{j=1}^{P} \{\|\mathbf{e}^y_{k+j}\|^q_{\mathbf{Q}_j} + \|\mathbf{s}_j\|^q_{\mathbf{T}}\} + \sum_{j=0}^{M-1} \{\|\mathbf{e}^u_{k+j}\|^q_{\mathbf{R}_j} + \|\Delta\mathbf{u}_{k+j}\|^q_{\mathbf{S}_j}\} \quad (31a)$$

subject to a model constraint:

$$\begin{aligned} &\mathbf{x}_{k+j} = \mathbf{f}(\mathbf{x}_{k+j-1}, \mathbf{u}_{k+j-1}) \quad \forall j = 1, P, \\ &\mathbf{y}_{k+j} = \mathbf{g}(\mathbf{x}_{k+j}, \mathbf{u}_{k+j}) \quad \forall j = 1, P \end{aligned} \quad (31b)$$

and subject to inequality constraints:

$$\begin{aligned} &\underline{\mathbf{y}} - \mathbf{s}_j \leqslant \mathbf{y}_{k+j} \leqslant \bar{\mathbf{y}} + \mathbf{s}_j \quad \forall j = 1, P, \\ &\mathbf{s}_j \geqslant 0 \quad \forall j = 1, P, \\ &\underline{\mathbf{u}} \leqslant \mathbf{u}_{k+j} \leqslant \bar{\mathbf{u}} \quad \forall j = 0, M-1, \\ &\Delta\underline{\mathbf{u}} \leqslant \Delta\mathbf{u}_{k+j} \leqslant \Delta\bar{\mathbf{u}} \quad \forall j = 0, M-1. \end{aligned} \quad (31c)$$

The objective function in Eq. (31a) involves four conflicting contributions. Future output behavior is controlled by penalizing deviations from the desired output trajectory $\mathbf{y}^r_{k+j}$, defined as $\mathbf{e}^y_{k+j} \equiv \mathbf{y}_{k+j} - \mathbf{y}^r_{k+j}$, over a *prediction horizon* of length P. Output constraint violations are penalized by minimizing the size of output constraint slack variables $\mathbf{s}_j$. 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 $\mathbf{Q}_j$, $\mathbf{T}_j$, $\mathbf{S}_j$, and $\mathbf{R}_j$; these are chosen to be positive semi-definite. The solution is a set of M input adjustments:

$$\mathbf{u}^M = (\mathbf{u}_k, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_{k+M-1}). \quad (32)$$

Most of the MPC controllers use a quadratic objective function similar to 31a ($q = 2$) for the dynamic optimization. For this case the dynamic optimization takes the form of a QP, and can be solved reliably using standard software. However, for very large problems, or very fast processes, there may not be sufficient time available to solve a QP. For these reasons the DMC-plus and PFC algorithms use fast sub-optimal algorithms to generate approximate solutions to the dynamic optimization. In the DMC-plus algorithm, when an input is predicted to violate a maximum or minimum limit it is set equal to the limit and the calculation is repeated with the MV removed. The PFC algorithm performs the calculation without constraints and then clips the input values if they exceed hard constraints. Both of these techniques will prevent violation of hard input constraints but will, in general, involve a loss of performance that is difficult to predict. Typically the resulting performance is acceptable, but the solutions do not satisfy the Karush–Kuhn–Tucker necessary conditions for optimality.

The DMC-plus and SMOC algorithms penalize only the last input deviation in order to drive the system towards the optimal steady state:

$$\begin{aligned} &\mathbf{R}_j = \mathbf{0} \quad \forall j < M-1, \\ &\mathbf{R}_{M-1} \geqslant \mathbf{0}. \end{aligned} \quad (33)$$

If the final input weight is large enough and the process is stable, this is approximately equivalent to having a terminal state constraint. If the dynamic solution at the end of the horizon is significantly different from the steady-state targets, which means the terminal states are not effectively constrained, the DMC-plus controller will automatically turn off. This setting effectively provides nominal stability for DMC-plus controller. The final input weights are also applicable to integrating processes where the derivative of the integrator is driven to zero.

The RMPCT, HIECON, PFC, and SMOC algorithms do not penalize input moves directly ($\mathbf{S}_j = 0$). The HIECON and PFC algorithms use a predefined output reference trajectory to avoid aggressive MV moves. The RMPCT controller defines a funnel, which will be described later in the paper, and finds the optimal trajectory and optimal MV moves by minimizing:

$$(\mathbf{u}^M, \mathbf{y}^r) = \arg\min \sum_{j=1}^{P} \|\mathbf{y}_{k+j} - \mathbf{y}^r_{k+j}\|^2_{\mathbf{Q}} + \|\mathbf{u}_{k+M-1} - \mathbf{u}_s\|^2_{\mathbf{R}} \quad (34)$$

subject to the funnel constraints. The relative priority of the two terms is set by the two weighting matrices. In the case that the first term is completely satisfied, which is typical due to the funnel formulation, the CV error will vanish and the minimization is in fact performed on the second term only. In this case the results will be similar to having two separate objectives on CVs and MVs. In the case of an infinite number of solutions, which is also typical due to "relaxing" the trajectory, a minimum norm solution to the MVs is obtained due to the use of singular value thresholding.

Using a single objective function in the dynamic optimization means that trade-offs between the four contributions must be resolved using the relative weight matrices $\mathbf{Q}_j$, $\mathbf{T}_j$, $\mathbf{S}_j$, and $\mathbf{R}_j$. The HIECON algorithm resolves conflicting dynamic control objectives by solving separate CV and MV optimization problems. The decision is made, *a priori*, that CV errors are more important than MV errors. A quadratic output optimization problem is solved first, similar to 31 but including only the $\mathbf{e}^y_{k+j}$ terms. For the thin and square plant cases this will provide a unique solution and the calculation terminates. For the fat plant case there are remaining degrees of freedom that can be used to optimize the input settings. For this case the controller solves a separate quadratic input optimization problem, similar to 31 but including only the $\mathbf{e}^u_{k+j}$ terms. The input optimization includes a set of equality constraints that preserve the future output trajectories found in the output optimization. This eliminates the need to set weights to determine the trade-off between output and input errors, at the cost of additional computation.

The PFC controller includes only the process input and output terms in the dynamic objective, and uses constant weight matrices ($\mathbf{Q}_j = \mathbf{Q}$, $\mathbf{T}_j = \mathbf{0}$, $\mathbf{R}_j = \mathbf{R}$, $\mathbf{S}_j = \mathbf{0}$, $q = 2$). The Aspen Target and MVC products include all four terms with constant weights ($\mathbf{Q}_j = \mathbf{Q}$, $\mathbf{T}_j = \mathbf{T}$, $\mathbf{R}_j = \mathbf{R}$, $\mathbf{S}_j = \mathbf{S}$, $q = 2$). The NOVA-NLC product adds to this the option of one norms ($\mathbf{Q}_j = \mathbf{Q}$, $\mathbf{T}_j = \mathbf{T}$, $\mathbf{R}_j = \mathbf{R}$, $\mathbf{S}_j = \mathbf{S}$, $q = 1, 2$).

Instead of using a reference trajectory, the Process Perfecter product ($\mathbf{T}_j = \mathbf{T}$, $\mathbf{R}_j = \mathbf{0}$, $\mathbf{S}_j = \mathbf{0}$, $q = 2$) uses a dynamic objective with *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.

3.4.6. Constraint formulations

There are three types of constraints commonly used in industrial MPC technology; hard, soft, and setpoint approximation. These are illustrated in Fig. 6. Hard constraints, shown in the top of Fig. 6, are those which should never be violated. Soft constraints, shown in the middle of Fig. 6 are those for which some violation may be allowed; the violation is typically minimized using a quadratic penalty in the objective function.

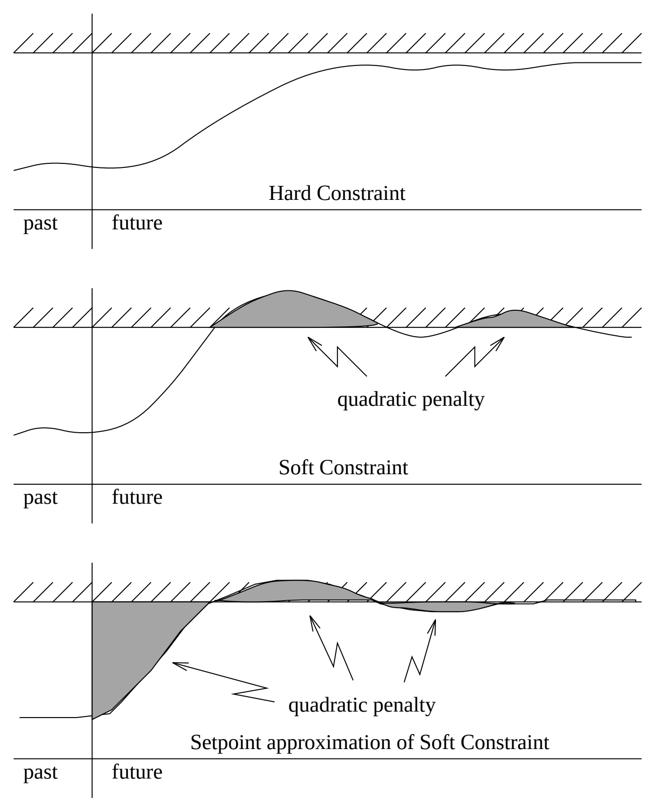

Fig. 6. The three basic types of constraint; hard, soft and setpoint approximation. Hard constraints (top) should not be violated in the future. Soft constraints (middle) may be violated in the future, but the violation is penalized in the objective function. Setpoint approximation of constraint (bottom) penalizes deviations above and below the constraint. Shades areas show violations penalized in the dynamic optimization. Adapted from Froisy (1994).

Another way to handle soft constraints is to use a setpoint approximation, as illustrated at the bottom of Fig. 6. Setpoints are defined for each soft constraint, resulting in objective function penalties on both sides of the constraint. The output weight is adjusted dynamically, however, so that the weight becomes significant only when the CV comes close to the constraint. When a violation is predicted the weight is increased to a large value so that the control can bring the CV back to its constraint limit. As soon as the CV is within the constraint limit, the steady-state target is used as the setpoint instead.

All of the MPC algorithms allow hard MV 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 MVs at any single control execution. The PFC algorithm also accommodates maximum and minimum MV acceleration constraints which are useful in mechanical servo control applications.

With the exception of the DMC-plus algorithm, all of the MPC products enforce soft output constraints in the dynamic optimization. Hard output constraints are provided as options in the PFC, HIECON, and NOVA-NLC algorithms, but these must be used carefully, because enforcement of hard output constraints can lead to closed-loop instability for nonminimum phase processes (Zafiriou, 1990; Muske & Rawlings, 1993). Hard output constraints can also cause feasibility problems, especially if a large disturbance enters the process. In the PFC and HIECON algorithms, hard output constraints are ranked in order of priority so that low priority constraints can be dropped when the problem becomes infeasible. The Connoisseur algorithm provides constraint optimization to match the number of active constraints with the number of degrees of freedom available to the controller, i.e., the number of unconstrained MVs.

3.4.7. Output and input trajectories

Industrial MPC controllers use four basic options to specify future CV behavior; a setpoint, zone, reference trajectory or funnel. These are illustrated in Fig. 7. The shaded areas correspond to the $\mathbf{e}^y_{k+j}$ and $\mathbf{e}^u_{k+j}$ terms in 31a. All of the controllers provide the option to drive the CVs 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 internal model differs significantly from the process. Several of the MPC algorithms use move suppression factors for this purpose.

All of the controllers also provide a CV zone control option, designed to keep the CV within a zone defined

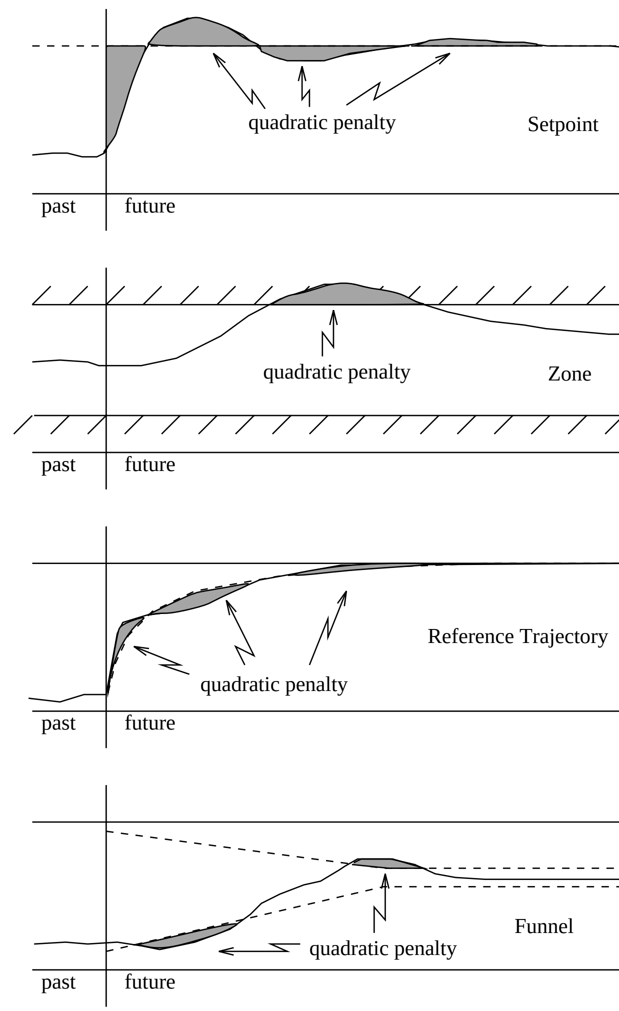

Fig. 7. Four options for specifying future CV behavior; setpoint, zone, reference trajectory and funnel. Shaded areas show violations penalized in the dynamic optimization.

by upper and lower boundaries. One way to implement zone control is to define upper and lower soft constraints. Other implementations are possible, however. The DMC-plus algorithm, for example, uses the setpoint approximation of soft constraints to implement the upper and lower zone boundaries.

The HIECON, PFC, SMOC, Aspen Target, MVC, and NOVA-NLC algorithms provide an option to specify a desired future path for each CV called a *reference trajectory*. A first or second order curve is drawn from the current CV value to the setpoint, with the speed of the response determined by one or more trajectory time constants. Future CV deviations from the reference trajectory are penalized. In the limit of zero time constants 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. A drawback of the reference trajectory formulation is that it penalizes the output when it happens to drift too quickly towards the setpoint, as might happen in response to an unmeasured disturbance. If the CV moves too quickly due to model mismatch, however, the reference trajectory is beneficial in that it will slow down the CV and minimize overshoot. The reference trajectory can be interpreted mathematically as a filter in the feedback path, similar to the robustness filter recommended by IMC theory (Morari & Zafiriou, 1989). In general, as the reference trajectory time constants increase, the controller is able to tolerate larger model mismatch.

The NOVA-NLC reference trajectory is slightly more general in that it allows separate specification of upper and lower trajectories with different dynamics and setpoints. In the controller objective, only deviations above the upper trajectory and deviations below the lower trajectory are penalized. This provides additional freedom during the transient that the controller can utilize for other tasks.

The RMPCT algorithm attempts to keep each CV within a user defined zone, with setpoints defined by setting the maximum and minimum zone limits equal to each other. When the CV goes outside the zone, the RMPCT algorithm defines a CV funnel, shown at the bottom of Fig. 7, to bring the CV back within its range. The slope of the funnel is determined by a user defined performance ratio, defined as the desired time to return to the zone divided by the open-loop response time. A weighted average open-loop response time is used for multivariable systems.

The SMOC algorithm uses a variation of funnel when a zone CV falls out of its range. In this case a zone with reference trajectory bounds is defined to drive the CV back into the control zone.

The NOVA-NLC upper and lower trajectories, RMPCT funnel, and SMOC reference trajectory bounds are advantageous in that if a disturbance causes the predicted future CV to reach the setpoint more quickly than a reference trajectory allows, the controller will take no action. This effectively frees up degrees of freedom for the controller to achieve other objectives. In the same situation, other controllers would move the inputs to bring the CV back onto the defined trajectory. This is illustrated in Fig. 8 for the case of the RMPCT funnel.

All of the MPC algorithms surveyed here provide MV setpoints to drive the inputs towards their optimal values when there are sufficient degrees of freedom.

3.4.8. *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*. This is illustrated at the top of Fig. 9. The finite output horizon formulation is used

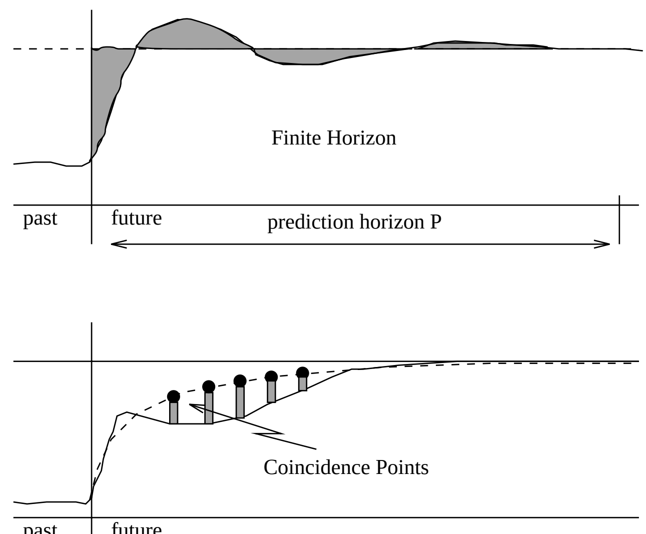

Fig. 9. Output horizon options. Finite horizon (top) includes P future points. A subset of the prediction horizon, called the coincidence points (bottom) may also be used. Shaded areas show violations penalized in the dynamic optimization.

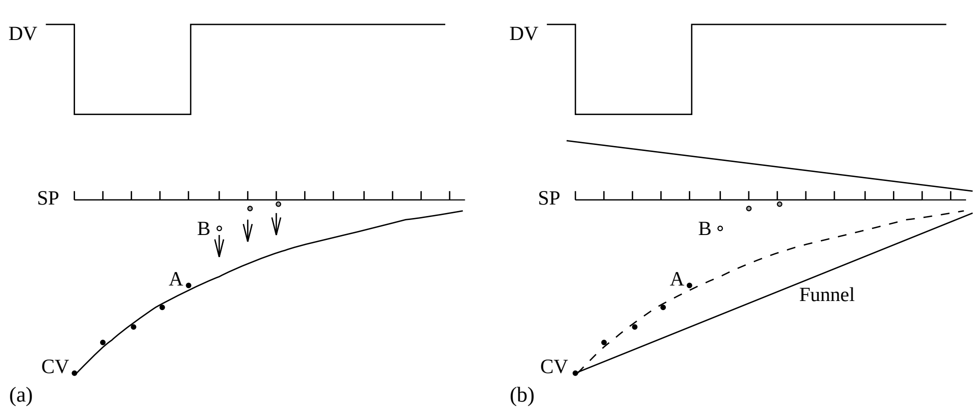

Fig. 8. MPC based on a funnel allows a CV to move back to the setpoint faster than a trajectory would require if a pulse disturbance releases. A trajectory based MPC would try to move away from the setpoint to follow the trajectory.

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.

Most of the MPC controllers use a multiple point output horizon; this means that predicted objective function contributions are evaluated at each point in the future. The PFC, Process Perfecter, and Aspen Target controllers allow the option to simplify the calculation by considering only a subset of points in the prediction horizon called *coincidence points*. This concept is illustrated at the bottom of Fig. 9. A separate set of coincidence points can be defined for each output, which is useful when one output responds quickly relative to another. The full finite output horizon can be selected as a special case.

Industrial MPC controllers use three different methods to parameterize the MV profile; these are illustrated in Fig. 10. Most of the MPC algorithms compute a sequence of future moves to be spread over a finite *control horizon*, as shown at the top of Fig. 10. The length of the control horizon M is another basic tuning parameter for these controllers. The most common

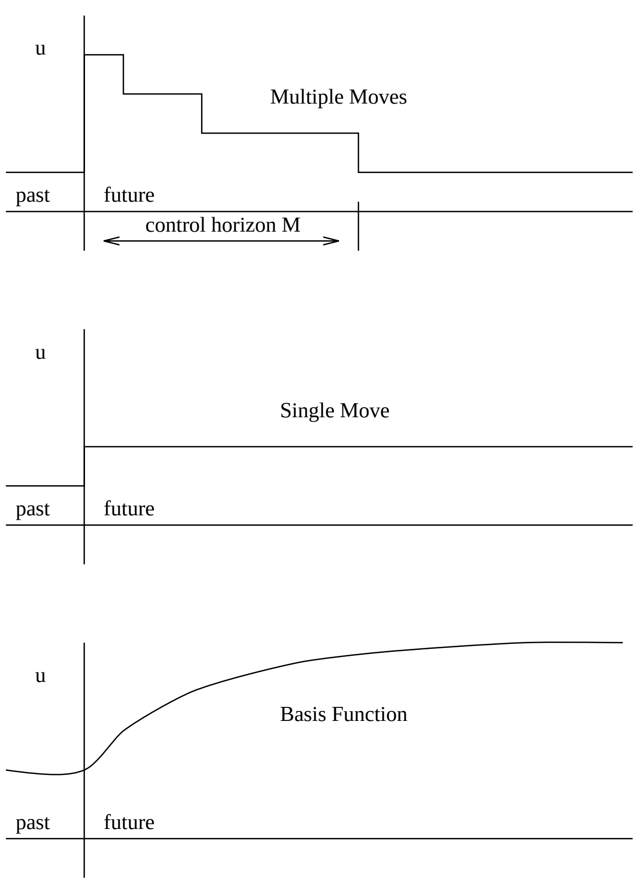

Fig. 10. Input parameterization options. Multiple move option (top), single move option (middle), basis function parameterization (bottom).

parameterization, referred to as multiple moves, means that a separate input adjustment is computed for each time point on the control horizon. Performance improves as M increases, at the expense of additional computation. To simplify the calculation, the Aspen Target, Connoisseur, and RMPCT algorithms provide a move blocking option, allowing the user to specify points on the control horizon where moves will not be computed. This reduces the dimension of the resulting optimization problem at the possible cost of control performance.

The HIECON and MVC algorithms compute a single future input move, as shown in the middle of Fig. 10. This greatly simplifies the calculation for these algorithms, which is helpful for the HIECON algorithm where two separate dynamic optimizations are solved at each control execution. However, using a single move involves a sacrifice of closed-loop performance that may be difficult to quantify.

The PFC controller parameterizes the future input profile using a set of polynomial basis functions. A possible solution is illustrated at the bottom of Fig. 10. 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 often important for mechanical servo control applications.

3.4.9. Numerical solution methods

It is interesting to consider the various numerical solution methods used in the MPC control calculations. Here we focus on the dynamic optimization step, which takes the form of the nonlinear program 31. For the linear MPC technology, this reduces to a QP. The PFC algorithm with linear models solves this QP in the simplest possible way, using a least-squares solution followed by clipping to enforce input constraints. While this is clearly sub-optimal, only a single linear system must be solved, so the PFC controller can be applied to very fast processes. The DMC-plus algorithm is slightly more complex in that it solves a sequence of least-squares problems in order to more accurately approximate the QP solution.

The remaining linear products use various active set QP methods to solve for the input profile. Unfortunately, these algorithms do not exploit the problem structure efficiently, and the solution time generally grows in proportion to M^3, where M is the control horizon. More efficient solutions are possible if the model is retained as an equality constraint, yielding a solution time that grows linearly in proportion to M. Rao, Wright, and Rawlings describe one such method based on interior point techniques (Rao, Wright, & Rawlings, 1998).

The nonlinear MPC products must of course use more powerful solution techniques; these are usually based upon a sequence of iterations in which a linearized version of the problem is solved. The Aspen Target product uses a multi-step Newton-type algorithm developed by De Oliveira and Biegler (1995, 1994), named QPKWIK, which has the advantage that intermediate solutions, although not optimal, are guaranteed feasible. This permits early termination of the optimization algorithm if necessary which guarantees a feasible solution. Aspen Target uses the same QPKWIK engine for local steady-state optimization and the dynamic MV calculation.

The MVC and process perfecter products use a generalized reduced gradient code called GRG2 developed by Lasdon and Warren (1986). The NOVA-NLC product uses the NOVA optimization package, a proprietary mixed complementarity nonlinear programming code developed by DOT Products.

The PFC controller, when used with a nonlinear model, 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.

4. Applications summary

Tables 6 and 7 summarize the reported applications of linear and nonlinear MPC technology through 1999. It is important to note that the vendors were free to define what constitutes an *application*; for this reason one must be careful when drawing conclusions from these data. For example, in some cases a single application may be defined as a large MPC controller that encompasses an entire chemical plant. In other cases, such as an automobile engine control, the vendor may report a single application even when several thousand copies of the completed controller are sold. Note also that this is a count of MPC applications performed by the vendors themselves; this does not include a significant number of in-house applications performed by licensees of vendor technology, such as ExxonMobil and DuPont. Nor does it include any consideration of MPC technology developed completely in-house by operating companies such as Eastman Chemical.

More than 4600 total MPC applications are reported in Tables 6 and 7, over twice the number in our previous survey from 5 years earlier (Qin & Badgwell, 1997). It is therefore safe to conclude that overall usage of MPC technology is still growing rapidly. All of the vendors report a significant number of applications in progress so it is likely that this number will continue to increase in the coming years. It should be noted that the size of these applications ranges from a single variable to several hundreds of variables. Therefore, one should not use the number of applications as an indication of market share.

Tables 6 and 7 show that MPC technology can now be found in a wide variety of application areas. The largest single block of applications is in refining, which amounts to 67% of all classified applications. This is also one of the original application areas where MPC technology has a solid track record of success. A significant number of applications can also be found in petrochemicals and chemicals, although it has taken longer for MPC technology to break into these areas. Significant growth areas include the chemicals, pulp and paper, food processing, aerospace and automotive industries.

Table 6 shows that AspenTech and Honeywell Hi-Spec are highly focused in refining and petrochemicals, with a handful of applications in other areas. Adersa

Table 7
Summary of nonlinear MPC applications by areas (estimates based on vendor survey; estimates do not include applications by companies who have licensed vendor technology)

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		1		5	15	21
Pulp & Paper					1	1
Refining					13	13
Utilities		5	2			7
Unclassified	1		1			2
Total	3	6	36	5	43	93

and Invensys apparently have a broader range of experience with applications in the food processing, mining/metallurgy, aerospace and automotive areas, among others. The applications reported by Adersa include a number of embedded PFC applications, so it is difficult to report their number or distribution. While only a total number was reported by SGS, this includes a number of in-house SMOC applications by Shell, so the distribution is likely to be shifted towards refining and petrochemical applications.

The bottom of Table 6 lists the largest linear MPC applications to date by each vendor, in the form of (outputs)×(inputs). The numbers show a difference in philosophy that is a matter of some controversy. AspenTech prefers to solve a large control problem with a single controller application whenever possible; they report an olefins application with 603 outputs and 283 inputs. Other vendors prefer to break the problem up into meaningful sub-processes.

The nonlinear MPC applications reported in Table 7 are spread more evenly among a number of application areas. Areas with the largest number of reported NMPC applications include chemicals, polymers, and air and gas processing. It has been observed that the size and scope of NMPC applications are typically much smaller than that of linear MPC applications (Martin & Johnston, 1998). This is likely due to the computational complexity of NMPC algorithms.

5. Limitations of existing technology

Many of the currently available industrial MPC algorithms suffer from limitations inherited from the original DMC and IDCOM technology. Problems with the control technology, which have been discussed by others (Morari & Lee, 1991; Muske & Rawlings, 1993), include:

- limited model choices;
- sub-optimal feedback;
- lack of nominal stability;
- sub-optimal or inefficient solution of the dynamic optimization.

Two of the linear MPC algorithms, DMC-plus and HIECON, rely on convolution models (impulse and step response). This can be problematic when controlling a process with widely varying time constants; for this case it is typical to sacrifice dynamic control of the fast process modes in order to keep the model length reasonable. A potentially more significant problem with the impulse and step response models is that they are limited to strictly stable processes. While it is certainly possible to modify the algorithms to accommodate a pure integrator, these modifications may lead to other problems, such as adding the derivative of a noisy output signal into the feedback path. It is not possible, in general, to represent an unstable process using an impulse response model. All of these problems can be overcome by using an auto-regressive parametric model form such as a state-space or ARX model.

The bias update feedback technique used by industrial MPC controllers is probably the best assumption that can be used for stable plants in the total absence of disturbance and measurement information, but better feedback is possible if the distribution of disturbances can be characterized more carefully. The bias update method implicitly assumes that there are no stochastic disturbances affecting the system state, and the measurement is perfect. Measurement noise is typically dealt with separately in applications using ad hoc filtering techniques. The bias update fails for purely integrating systems, so a number of ad hoc methods have been developed by practitioners to handle this case. These methods tend to work poorly in the presence of significant measurement noise. Industrial practitioners need to recognize that the Kalman filter provides a rich framework to address these problems. Muske and Rawlings have demonstrated how better performance can be achieved for MPC applications by using a state-space model and an optimal state observer (Muske & Rawlings, 1993). Of course, these problems motivated the original development of the SMOC algorithm (Marquis & Broustail, 1998), which uses a Kalman filter for feedback.

Tuning MPC controllers for stable operation in the presence of constraints may be difficult, even when the process model is perfect. Currently much effort is spent on closed-loop simulation prior to commissioning a controller. It must be recognized, however, that simulating all possible combinations of active constraints is impossible in practice. It seems far better to build an algorithm that is inherently stable in all such cases; in control theory this is known as a *nominally stabilizing* algorithm. Scokaert and Rawlings (1998) show how this can be done for the linear case by using infinite prediction and control horizons. Chen and Allgöwer present a quasi-infinite horizon method for nonlinear systems (Chen & Allgower, 1995). Research results for the nonlinear case are covered thoroughly by Mayne et al. (2000). None of the currently available industrial MPC algorithms makes full use of these ideas.

Sub-optimal solutions to the dynamic optimization 31 are used in several of the packages, presumably in order to speed up the solution time. This seems difficult to justify for the case of refining and petrochemical applications, where the controllers run on the order of once each minute. Goodwin et al. (2001) give an example where unconstrained LQR solution with anti-windup has significant performance deterioration compared to an MPC solution under severe constraints. However, for high speed applications where the controller must execute in a few milliseconds, such as tracking the position of a missile, it may not be feasible to solve a QP at every control execution. For this case a good sub-optimal solution may be the only option.

None of the linear MPC products exploit the structure of QP in the dynamic optimization. The singular value technique in RMPCT is one of the few efforts in using robust numerical techniques. By properly exploiting the structure of the problem, it should be possible to handle significantly larger and/or faster processes (Rao et al., 1998).

Model uncertainty is not addressed adequately by current MPC technology. While most of the identification packages provide estimates of model uncertainty, only one vendor (Honeywell) provides a way to use this information in the control design. All of the MPC algorithms provide a way to detune the control to improve robustness, but the trade-off between performance and robustness is generally not very clear. Until this connection is made, it will not be possible to determine when a model is accurate enough for a particular control application. There are many robust MPC results available in the academic literature, but most focus only the dynamic optimization and will not work for something as simple as a setpoint change. Promising results that may eventually impact MPC practice include the robust stability conditions presented by Vuthandam et al. for a modified QDMC algorithm (Vuthandam, Genceli, & Nikolaou, 1995); the robust MPC algorithm presented by Kothare, Balakrishnan, and Morari (1996) and the robust steady-state target calculation described by Kassmann, Badgwell, and Hawkins (2000). More research is needed in this area.

Current model identification technology suffers from a number of limitations:

- plant test signals are typically step-like and require close attention of experienced engineers during the test;
- there is no tool to determine whether the collected data are adequate to represent the process dynamics for MPC design; in practice the plant is generally over tested, increasing the implementation cost during the test period;
- identification algorithms are mainly of the least-squares type; there are only a few that use modern subspace identification and prediction error methods;
- statistical efficiency and consistency of identification algorithms are not addressed;
- there is a lack of model validation methods to verify whether the identified model is adequate for control or whether significant model deterioration has occurred after the MPC controller has been commissioned;
- there is little practice except in Connoisseur in multivariable closed-loop identification which would be very useful for on-line model updating or adaptive MPC implementation;
- there is no systematic approach for building nonlinear dynamic models for NMPC. Guidelines for plant tests are needed to build a reliable nonlinear model. This is important because even more test data will be required to develop an empirical nonlinear model than an empirical linear model.

6. Next-generation MPC technology

MPC vendors were asked to describe their vision of next-generation MPC technology. Their responses were combined with our own views and the earlier analysis of Froisy (1994) to come up with a composite view of future MPC technology.

6.1. Basic controller formulation

Because it is so difficult to express all of the relevant control objectives in a single objective function, next-generation MPC technology will utilize multiple objective functions. The infinite prediction horizon has beneficial theoretical properties and will probably become a standard feature. Output and input trajectory options will include setpoints, zones, trajectories and funnels. Input parameterization using basis functions may become more widespread, and infinite control horizons with moves computed at each control interval may also become possible.

6.2. Adaptive MPC

A few adaptive MPC algorithms such as the GPC algorithm introduced by Clarke et al. have been proposed (Clarke, Mohtadi, & Tuffs, 1987) but only two adaptive MPC algorithms have reached the marketplace (Connoisseur from Invensys and STAR from Dot Products (Dollar, Melton, Morshedi, Glasgow, & Repsher, 1993)). This is despite the strong market incentive for a self-tuning MPC controller. This reflects the difficulty of doing adaptive control in the real world. Barring a theoretical breakthrough, this situation is not likely to change in the near future.

On the other hand, adaptive and on-demand tuning PID controllers have been very successful in the marketplace. This suggests that adaptive MPC controllers may emerge for SISO loops as adaptive PID technology is generalized to handle more difficult dynamics.

6.3. Robust MPC

With one exception (Honeywell), industrial MPC controllers rely solely on brute-force simulation to evaluate the effects of model mismatch. Robust stability guarantees would significantly reduce the time required to tune and test industrial MPC algorithms. It is likely that recent robust research results (Kassmann et al., 2000; Kothare et al., 1996; Scokaert & Mayne, 1998) will eventually make their way into MPC products. Robust stability guarantees will then be combined with uncertainty estimates from identification software to greatly simplify design and tuning of MPC controllers.

6.4. Nonlinear MPC

Next-generation MPC technology will allow nonlinear models to be developed by seamlessly combining process knowledge with operating data. A continuous-time fundamental model will be defined from a graphical representation of the plant. Process test signals will be designed automatically so as to explore important regions of the operating space where the model is inadequate for control. Closed-loop process tests will be conducted using PRBS signals, with minimal on-site time requirements. Owing to the difficulty in data based modeling of nonlinear processes, first principles models and other alternative modeling methods (Foss, Lohmann, & Marquardt, 1998) will become necessary in practice.

7. Discussion and conclusions

MPC technology has progressed steadily since the first IDCOM and DMC applications over 25 years ago. Our survey data show that the number of MPC applications has approximately doubled in 4 years from 1995 to 1999. MPC applications show a solid foundation in refining and petrochemicals, and significant penetration into a wide range of application areas from chemicals to food processing. The MPC technology landscape has changed dramatically in recent years, making it difficult for us to keep track of the swift progress in academic research and industrial applications. Major recent developments include:

- Technological changes due to the development of new algorithms by academia and industry, computer hardware and software advances, and the breadth of industrial applications.
- Organizational changes due to acquisitions and mergers, such as the acquisition by Aspen Technology of DMCC, Setpoint, Treiber Controls, and Neuralware.
- Integration of MPC technology into Distributed Control System (DCS) platforms, illustrated by the acquisition of Predictive Control Ltd. by Invensys, the parent company of Foxboro.
- Increased applications by end-users who license MPC software from vendors.

Current generation linear MPC technology offers significant new capabilities, but several unnecessary limitations still remain. Some products do not allow the use of parametric models. Feedback options are severely limited, often leading to poor control of integrating systems and slow rejection of unmeasured disturbances. The algorithms are not nominally stabilizing, so that tuning choices must be tested extensively through closed-loop simulation. None of the linear MPC algorithms makes use of modern numerical solution methods, limiting the size and speed of processes that they can be applied to.

Applications of nonlinear MPC products will continue to grow in those areas where linear MPC technology has proven difficult to apply, such as control of polymerization processes. The nonlinear MPC algorithms reported here differ in the simplifications used to generate a tractable control calculation. The models are usually specified as linear dynamic subsystems with added nonlinearity. The NOVA-NLC controller is the only current offering that allows the use of a rigorous nonlinear model derived from first principles.

An important observation is that the majority of industrial MPC controllers use empirical dynamic models identified from test data. Improvements in identification technology that result in shorter overall process testing times will have a large positive impact on the bottom line for MPC technology users.

Choosing an MPC technology for a given application is a complex question involving issues not addressed in this paper. Here, we have emphasized the technical capabilities of MPC technology. However, if a vendor is to be selected to design and implement an advanced control system, it would be wise to weigh heavily their experience with the particular process in question.

Research needs as perceived by industry are mostly control engineering issues, not algorithm issues. Industrial practitioners do not perceive closed-loop stability, for example, to be a serious problem. Their questions are more like: Which variables should be used for control? When is a model good enough to stop the identification plant test? How do you determine the source of a problem when a controller is performing poorly? When can the added expense of a nonlinear MPC controller be justified? How do you design a control system for an entire plant? How do you estimate the benefits of a control system? Answering these questions will provide control practitioners and theoreticians with plenty of work in the foreseeable future.

Just as academicians do not agree on which control algorithms are best, industrial practitioners also have very different views on the future trends of MPC. Just as PID controllers have many different implementations (series form, parallel form, etc.), MPC controllers are not anticipated to converge to a single form in the foreseeable future. While academic research will continue to develop new algorithms and prove new facts in the MPC domain, industrial development will consider its own priorities with a healthy interaction with academia. Although there is still plenty of room to refine or develop new MPC algorithms, pushing the technology to new application areas could be at least equally important and challenging. Given the uneven distribution in the MPC applications today, it is safe to anticipate that much more development in both theory and practice is still ahead of us.

Acknowledgements

The authors wish to thank the vendor representatives who responded to our MPC survey for their time and effort. We also wish to thank the following people for their constructive criticism of this paper: Pierre Carrette, Jim Downs, Harry Forbes, Brian Froisy, Pierre Grosdidier, John Guiver, Rob Hawkins, Dean Kassmann, Geoff Lewis, Joseph Lu, Greg Martin, Steve Piche, Jim Rawlings, Jaques Richalet, Andy Speziale, Douglas White, Robert Young, and Hong Zhao.

References

Allgower, F., Badgwell, T. A., Qin, S. J., Rawlings, J. B., & Wright, S. J. (1999). Nonlinear predictive control and moving horizon estimation—an introductory overview. In P. M. Frank (Ed.), *Advances in control: highlights of ECC '99*. Berlin: Springer.

Allgower, F., & Zheng, A., (Eds.). (2000). *Nonlinear model predictive control, progress in systems and control theory*. Vol. 26. Basel, Boston, Berlin: Birkhauser Verlag.

Bartusiak, R. D., & Fontaine, R. W. (1997). *Feedback method for controlling non-linear processes*. US Patent 5682309.

Berkowitz, P., & Papadopoulos, M. (1995). *Multivariable process control method and apparatus*. US Patent 5396416.

Berkowitz, P., Papadopoulos, M., Colwell, L., & Moran, M. (1996). *Multivariable process control method and apparatus*. US Patent 5488561.

Chen, H., & Allgower, F. (1995). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, *34*(10), 1205–1218.

Clarke, D. W., Mohtadi, C., & Tuffs, P. S. (1987). Generalized predictive control—Part I. The basic algorithm. *Automatica*, *23*(2), 137–148.

MVC 3.0 User Manual (1995). Continental Controls, Inc. Product Literature.

Cutler, C., Morshedi, A., & Haydel, J. (1983). An industrial perspective on advanced control. In *AICHE annual meeting*, Washington, DC, October 1983.

Cutler, C. R., & Ramaker, B. L. (1979). Dynamic matrix control—a computer control algorithm. *AICHE national meeting*, Houston, TX, April 1979.

Cutler, C. R., & Ramaker, B. L. (1980). Dynamic matrix control—a computer control algorithm. In *Proceedings of the joint automatic control conference*.

Cutler, C. R., & Yocum, F. H. (1991). Experience with the DMC inverse for identification. In Y. Arkun, W. H. Ray (Eds.), *Chemical process control—CPC IV. Fourth international conference on chemical process control* (pp. 297–317). Amsterdam: Elsevier.

Demoro, E., Axelrud, C., Johnston, D., & Martin, G. (1997). Neural network modeling and control of polypropylene process. In *Society of plastics engineers international conference*, Houston, TX.

DMC Corp. [*DMC*]™ (1994). *Technology overview*. Product literature from DMC Corp., July 1994.

Dollar, R., Melton, L. L., Morshedi, A. M., Glasgow, D. T., & Repsher, K. W. (1993). Consider adaptive multivariable predictive controllers. *Hydrocarbon Processing*, *10*, 109–112.

Downs, J. J. (2001). Linking control strategy design and model predictive control. In *Preprints*: *Chemical process control-6*, *assessment and new directions for research* (*CPC VI*), Tuscon, AZ, January 2001 (pp. 411–422).

Foss, B. A., Lohmann, B., & Marquardt, W. (1998). A field study of the industrial modeling process. *Journal of Process Control*, *8*(5,6), 325–338.

Froisy, J. B. (1994). Model predictive control: Past, present and future. *ISA Transactions*, *33*, 235–243.

Froisy, J. B., & Matsko, T. (1990). IDCOM-M application to the Shell fundamental control problem. *AICHE annual meeting*, November 1990.

García, C. E., & Morshedi, A. M. (1986). Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, *46*, 73–87.

García, C. E., Prett, D. M., & Morari, M. (1989). Model predictive control: Theory and practice—a survey. *Automatica*, *25*(3), 335–348.

Goodwin, G. C., Graebe, S. F., & Salgado, M. E. (2001). *Control system design*. Englewood Cliffs, NJ: Prentice Hall.

Grosdidier, P., Froisy, B., & Hammann, M. (1988). The IDCOM-M controller. In T. J. McAvoy, Y. Arkun, & E. Zafiriou (Eds.), *Proceedings of the 1988 IFAC workshop on model based process control* (pp. 31–36). Oxford: Pergamon Press.

Hillestad, M., & Andersen, K. S. (1994). Model predictive control for grade transitions of a polypropylene reactor. In *Proceedings of the 4th European symposium on computer aided process engineering* (*ESCAPE* 4), Dublin, March 1994.

Honeywell Inc. (1995). *RMPCT concepts reference*. Product literature from Honeywell, Inc., October 1995.

Kailath, T. (1980). *Linear systems*. Englewood Cliffs, NJ: Prentice-Hall.

Kailath, T., Sayed, A. H., & Hassibi, B. (2000). *Linear estimation*. Upper Saddle River, NJ: Prentice-Hall.

Kalman, R. E. (1960a). Contributions to the theory of optimal control. *Bulletin de la Societe Mathematique de Mexicana*, *5*, 102–119.

Kalman, R. E. (1960b). A new approach to linear filtering and prediction problems. *Transactions of ASME*, *Journal of Basic Engineering*, *87*, 35–45.

Kassmann, D. E., Badgwell, T. A., & Hawkins, R. B. (2000). Robust steady-state target calculation for model predictive control. *A.I.CH.E. Journal*, *46*(5), 1007–1024.

Keeler, J., Martin, G., Boe, G., Piche, S., Mathur, U., & Johnston, D. (1996). *The process perfecter*: *the next step in multivariable control and optimization*. Technical report, Pavilion Technologies, Inc., Austin, TX.

Kothare, M. V., Balakrishnan, V., & Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, *32*, 1361–1379.

Kouvaritakis, B., & Cannon, M. (Eds.) (2001). *Nonlinear predictive control, theory and practice*. London: The IEE.

Kulhavy, R., Lu, J., & Samad, T. (2001). Emerging technologies for enterprise optimization in the process industries. In *Preprints*: *chemical process control*-6, *assessment and new directions for research* (*CPC VI*), Tuscon, Arizona, January 2001 (pp. 411–422).

Kwakernaak, H., & Sivan, R. (1972). *Linear optimal control systems*. New York: Wiley.

Larimore, W. E. (1990). Canonical variate analysis in identification, filtering and adaptive control. In *Proceedings of the 29th conference on decision and control* (pp. 596–604).

Lasdon, L. S., & Warren, A. D. (1986). *GRG2 user's guide*. Cleveland, OH: Cleveland State University.

Lee, J. H., & Cooley, B. (1997). Recent advances in model predictive control and other related areas. In J. C. Kantor, C. E. Garcia, B. Carnahan (Eds.), *Fifth international conference on chemical process control*, AICHE and CACHE, (pp. 201–216b).

Lee, E. B., & Markus, L. (1967). *Foundations of optimal control theory*. New York: Wiley.

Lee, J. H., Morari, M., & García, C. E. (1994). State-space interpretation of model predictive control. *Automatica*, *30*(4), 707–717.

Lewis, D. G., Evans, C. L., & Sandoz, D. J. (1991). Application of predictive control techniques to a distillation column. *Journal of Process Control*, *1*, 17–26.

Ljung, L. (1987). *System identification: theory for the user*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Ljung, L. (1999). *System identification: theory for the user*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Maciejowski, J. M. (2002). *Predictive control with constraints*. Englewood Cliffs, NJ: Prentice Hall.

Marquis, P., & Broustail, J. P. (1998). SMOC, a bridge between state space and model predictive controllers: Application to the automation of a hydrotreating unit. In T. J. McAvoy, Y. Arkun, & E. Zafiriou (Eds.), *Proceedings of the 1988 IFAC workshop on model based process control* (pp. 37–43). Oxford: Pergamon Press.

Martin, G., & Boe, G., Keeler, J., Timmer, D., Havener, J. (1998). *Method and apparatus for modeling dynamic and steady-state processes for prediction, control, and optimization*. US Patent.

Martin, G., & Johnston, D. (1998). Continuous model-based optimization. In *Hydrocarbon processing's process optimization conference*, Houston, TX.

Mayne, D. Q. (1997). Nonlinear model predictive control: An assessment. In J. C. Kantor, C. E. Garcia, & B. Carnahan (Eds.), *Fifth international conference on chemical process control* AICHE and CACHE, (pp. 217–231).

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*, 789–814.

Morari, M., & Lee, J. H. (1991). Model predictive control: The good, the bad, and the ugly. In Y. Arkun, & W. H. Ray (Eds.), *Chemical process control—CPC IV, Fourth international conference on chemical process control* (pp. 419–444). Amsterdam: Elsevier.

Morari, M., & Zafiriou, E. (1989). *Robust process control*. Englewood Cliffs, NJ: Prentice-Hall.

Muske, K. R. (1995). *Linear model predictive control of chemical processes*. PhD thesis, The University of Texas at Austin.

Muske, K. R., & Rawlings, J. B. (1993). Model predictive control with linear models. *A.I.CH.E Journal*, *39*(2), 262–287.

Ohshima, M., Ohno, H., & Hashimoto, I. (1995). Model predictive control: Experiences in the university-industry joint projects and statistics on MPC applications in Japan. *International workshop on predictive and receding horizon control*, Korea, October 1995.

De Oliveira, N. M. C., & Biegler, L. T. (1995). An extension of newtontype algorithms for nonlinear process control. *Automatica*, *31*, 281–286.

De Oliveira, N. M. C., & Biegler, L. T. (1994). Constraint handling and stability properties of model-predictive control. *A.I.CH.E Journal*, *40*(7), 1138–1155.

Piche, S., Sayyar-Rodsari, B., Johnson, D., & Gerules, M. (2000). Nonlinear model predictive control using neural networks. *IEEE Control Systems Magazine* 53–62.

Poe, W., & Munsif, H. (1998). Benefits of advanced process control and economic optimization to petrochemical processes. In *Hydrocarbon processing's process optimization conference*, Houston, TX.

Prett, D. M., & García, C. E. (1988). *Fundamental process control*. Boston: Butterworths.

Prett, D. M., & Gillette, R. D. (1980). Optimization and constrained multivariable control of a catalytic cracking unit. In *Proceedings of the joint automatic control conference*.

Propoi, A. I. (1963). Use of linear programming methods for synthesizing sampled-data automatic systems. *Automatic Remote Control*, *24*(7), 837–844.

Qin, S. J., & Badgwell, T. A. (1996). An overview of industrial model predictive control technology. In *Chemical process control—CPC V*, CACHE. Tahoe City, CA, January 1996.

Qin, S. J., & Badgwell, T. A. (1997). An overview of industrial model predictive control technology. In J. C. Kantor, C. E. Garcia, & B. Carnahan (Eds.), *Chemical process control—V, Fifth international conference on chemical process control* CACHE and AICHE, (pp. 232–256).

Qin, S. J., & Badgwell, T. A. (1998). An overview of nonlinear model predictive control applications. In F. Algower, & A. Zheng (Eds.), *Nonlinear model predictive control workshop—assessment and future directions*, Ascona, Switzerland.

Qin, S. J., & Badgwell, T. A. (2000). An overview of nonlinear model predictive control applications. In J. C. Kantor, C. E. Garcia, & B. Carnahan (Eds.), *Nonlinear model predictive control*. Basel: Birkhauser.

Rao, C. V., & Rawlings, J. B. (1999). Steady states and constraints in model predictive control. *A.I.CH.E. Journal*, *45*(6), 1266–1278.

Rao, C. V., Wright, S. J., & Rawlings, J. B. (1998). Application of interior-point methods to model predictive control. *Journal of Optimization Theory Application*, *99*, 723–757.

Rawlings, J. B. (2000). Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, *20*, 38–52.

Rawlings, J. B., Meadows, E. S., & Muske, K. (1994). Nonlinear model predictive control: a tutorial and survey. In *Proceedings of IFAC ADCHEM*, Japan.

Rawlings, J. B., & Muske, K. R. (1993). Stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, *38*(10), 1512–1516.

Richalet, J. (1993). Industrial applications of model-based control. *Automatica*, *29*, 1251–1274.

Richalet, J., Rault, A., Testud, J. L., & Papon, J. (1976). Algorithmic control of industrial processes. In *Proceedings of the 4th IFAC symposium on identification and system parameter estimation*. (pp. 1119–1167).

Richalet, J., Rault, A., Testud, J. L., & Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes. *Automatica*, *14*, 413–428.

Ricker, N. L. (1991). Model predictive control: State of the art. In Y. Arkun, W. H. Ray (Eds.), *Chemical process control—CPC IV, Fourth international conference on chemical process control* (pp. 271–296). Amsterdam: Elsevier.

Scokaert, P. O. M., & Mayne, D. Q. (1998). Min–max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, *43*(8), 1136–1142.

Scokaert, P. O. M., & Rawlings, J. B. (1998). Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, *43*(8), 1163–1169.

Sentoni, G. B., Biegler, L. T., Guiver, J. B., & Zhao, H. (1998). State-space nonlinear process modeling. *A.I.CH.E Journal*, *44*(10), 2229–2239.

Setpoint Inc. (1993). *SMC-IDCOM*: *A state-of-the-art multivariable predictive controller*. Product literature from Setpoint, Inc., October 1993.

Shinskey, F. G. (1994). *Feedback controllers for the process industries*. New York: McGraw-Hill.

Vuthandam, P., Genceli, H., & Nikolaou, M. (1995). Performance bounds for robust quadratic dynamic matrix control with end condition. *A.I.CH.E Journal*, *41*(9), 2083–2097.

Young, R. E., Bartusiak, R. B., & Fontaine, R. B. (2001). Evolution of an industrial nonlinear model predictive controller. In *Preprints: Chemical process control—CPC VI*, Tucson, Arizona (pp. 399–410). CACHE.

Yousfi, C., & Tournier, R. (1991). Steady-state optimization inside model predictive control In *Proceedings of ACC'91*, Boston, MA (pp. 1866–1870).

Zafiriou, E. (1990). Robust model predictive control of processes with hard constraints. *Computers in Chemical Engineering*, *14*(4/5), 359–371.

Zhao, H., Guiver, J., Neelakantan, R., & Biegler, L. T. (1999). A nonlinear industrial model predictive controller using integrated PLS and neural state space model. In *IFAC 14th triennial world congress*, Beijing, Peoples Republic of China.

Zhao, H., Guiver, J. P., & Sentoni, G. B. (1998). An identification approach to nonlinear state space model for industrial multivariable model predictive control. In *Proceedings of the 1998 American control conference*, Philadelphia, Pennsylvannia (pp. 796–800).