AN OVERVIEW OF INDUSTRIAL MODEL PREDICTIVE CONTROL TECHNOLOGY

S. Joe Qin
Department of Chemical Engineering
The University of Texas at Austin
Austin, TX 78712

Thomas A. Badgwell
Department of Chemical Engineering
Rice University
Houston, TX 77251

Abstract

This paper provides an overview of commercially available Model Predictive Control (MPC) technology, 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 then 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.

Keywords

Industrial Survey, Model Predictive Control

Introduction

Model Predictive Control (MPC) refers to a class of algorithms that compute a sequence of manipulated variable adjustments in order to optimize the future behavior of a plant. 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, aerospace, metallurgy, and pulp and paper.

Several authors have published excellent reviews of MPC theoretical issues, including the papers of García et al. (García, Prett and Morari, 1989), Ricker (Ricker, 1991), Morari and Lee (Morari and Lee, 1991), Muske and Rawlings (Muske and Rawlings, 1993) and Rawlings et al. (Rawlings, Meadows and Muske, 1994). The other papers in the present session by Mayne (Mayne, 1996) and Lee (Lee, 1996) summarize the very latest technical developments in MPC control theory. Froisy provides a vendor's perspective on industrial MPC technology and summarizes likely future developments (Froisy, 1994). A recent survey of MPC technology in Japan provides a wealth of information on application issues from the point of view of MPC users (Ohshima, Ohno and Hashimoto, 1995).

The primary purpose of this paper is to present an overview of commercially available MPC technology. A brief history of MPC technology development is presented first. A general MPC controller formulation is then described to provide a basis for discussion of the commercial products. Results of our industrial survey are then presented. The survey is not exhaustive in that several well-known companies either were not asked to participate, *chose* not to participate, or responded too late to be included in this paper. Nevertheless we believe that the products discussed here are sufficiently representative to allow us to draw conclusions regarding the current state of MPC technology. Significantly unique features of each offering are outlined and discussed. MPC applications to date by each vendor are summarized by application area. The final section presents a view of the next generation of MPC technology, emphasizing potential business and research opportunities.

A Brief History of Industrial MPC

This section presents an abbreviated history of industrial MPC technology. Control algorithms are emphasized here because relatively little published information is available on the identification technology.

The development of modern control concepts can be traced to the work of Kalman in the early 1960's, who sought to determine when a linear control system can be said to be optimal (Kalman, 1960a; Kalman, 1960b). Kalman studied a Linear Quadratic Regulator (LQR) designed to minimize an quadratic objective function. The process to be controlled can be described by a discrete-time, deterministic linear state-space model:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\,\mathbf{x}_k + \mathbf{B}\,\mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{C}\,\mathbf{x}_k \end{aligned} \tag{1}$$

The vector $\mathbf{u}$ represents process inputs, or manipulated variables; vector $\mathbf{y}$ describes process output measurements. The vector $\mathbf{x}$ represents process states. Figure 1 provides a schematic representation of a state space model. The state vector is defined such that knowing its value at time k and future inputs allows one to predict how the plant will evolve for all future time. Much of the power of Kalman's work relies on the fact that this general process model was used.

The objective function to be minimized penalizes 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:

$$J = \sum_{j=1}^{\infty} \left(||\mathbf{x}_{k+j}||^2_{\mathbf{Q}} + ||\mathbf{u}_{k+j}||^2_{\mathbf{R}} \right) \tag{2}$$

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

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

Implicit in this formulation is the assumption that all variables are written in terms of deviations from a desired steady-state. The solution to the LQR problem was shown to be a proportional controller, with a gain matrix $\mathbf{K}$ computed from the solution of a matrix Ricatti equation:

$$\mathbf{u}_k = -\mathbf{K}\,\mathbf{x}_k \tag{4}$$

The infinite prediction horizon of the LQR algorithm endowed the algorithm with powerful stabilizing properties; it was shown to be stabilizing for any reasonable linear plant (stablizable and detectable) as long as the objective function weight matrices $\mathbf{Q}$ and $\mathbf{R}$ are positive definite. A dual theory was developed to estimate plant states from noisy input and output measurements, using what is now known as a *Kalman Filter*. The combined LQR controller and Kalman filter is called a Linear Quadratic Gaussian (LQG) controller. Constraints on the process inputs, states and outputs were not considered in the development of LQG theory.

Although LQG theory provides an elegant and powerful solution to the problem of controlling an unconstrained linear plant, it 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 and Papon, 1978; García et al., 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 and Gillette, 1980). A successful industrial controller 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 on-line 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 Model Predictive Control (MPC) technology.

In modern processing plants the MPC controller is part of a multi-level hierarchy of control functions. This is illustrated in Figure 2, which shows a conventional control structure on the left for Unit 1 and a model predictive control structure on the right for Unit 2. Similar hierarchical structures have been described by Richalet et al. (Richalet et al., 1978) and Prett and García (Prett and Garcìa, 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.

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, for example, described a moving horizon controller in 1963 (Propoi, 1963). Lee and Markus (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.

IDCOM

The first description of MPC control applications was presented by Richalet et al. at a 1976 conference (Richalet, Rault, Testud and Papon, 1976) and later summarized in a 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 a black-box process representation of the process, shown in Figure 3. From this point of view the process inputs influence the process outputs directly. Process inputs are divided into manipulated variables (MV's) which the controller adjusts, and disturbance variables (DV'S) which are not available for control. Process outputs are referred to as controlled variables (CV's). 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:

$$y_{k+j} = \sum_{i=1}^{N} h_i \, u_{k+j-i} \tag{5}$$

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 structural distance between the plant and model impulse responses in parameter space. The resulting iterative algorithm makes small adjustments to the coefficients at each step in such a way that the structural distance continuously decreases. The algorithm was shown to converge to unbiased parameter estimates in the face of noisy output measurements. The control problem was solved using the same algorithm by noting that control is the mathematical dual of identification. In the identification problem one knows the outputs and the inputs and wishes to estimate the coefficients; in the control problem one knows the desired output trajectory and the model coefficients and the goal is to estimate the required inputs. Because the output prediction appears as a dot product of input and coefficient vectors, the same algorithm can be used to find either one. 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 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 nonlinear 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 Figure 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., for example, (Prett and Garcìa, 1988) describe a very similar hierarchy in their 1988 text *Fundamental Process Control*.

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.

DMC

Engineers at Shell Oil developed their own independent MPC technology in the early 1970's, 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 and Ramaker, 1979) and at the 1980 Joint Automatic Control Conference (Cutler and Ramaker, 1980). In a companion paper at the 1980 meeting Prett and Gillette (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:

$$y_{k+j} = \sum_{i=1}^{N-1} s_i \, \Delta u_{k+j-i} \; + \; s_N \, u_{k+j-N} \qquad (6)$$

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 (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. The overall FCCU control system was implemented in a multi-level hierarchy, with a nonlinear steady-state FCCU model at the top. At the start of each optimization cycle, parameters in the nonlinear model were estimated so as to match model predictions with measured steady-state operating data. The calibrated nonlinear model was then perturbed numerically to generate partial derivatives of each process output with respect to each process input (the matrix of partial derivatives is known as the Jacobian matrix in numerical analysis). The partial derivatives were then used in a Linear Program (LP) to compute a new economic optimal operating point for the FCCU, subject to steady-state process constraints. The optimal process input and output targets were then passed to a DMC algorithm for implementation. As soon as the DMC controller moved the unit to the new steady state the optimization cycle was repeated. This separation of the control system into constrained steady-state optimization and dynamic control is quite similar to the structure described by Richalet et al. and has since become standard in industrial control system design.

The DMC algorithm had the job of moving from the system from one optimal steady-state to another. Although the LP solution provided optimal targets for process inputs and outputs, dynamic disturbances could potentially cause the DMC algorithm to move inputs away from their optimal steady-state targets in order to keep outputs at their steady-state targets. Since moving one input away from its optimal target may be much more expensive than moving another, the control system should determine this trade-off in a rational way. The DMC algorithm was modified to account for such trade-offs by including an additional equation for each input in the process model. The new equation required that the sum of all moves for a particular input should equal the total adjustment required to bring that input to its optimal steady-state target. This allowed the inputs some freedom to move dynamically but required that the steady-state input solution be satisfied in a least-squares sense, with trade-offs determined by the appropriate objective function weights.

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.

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 and Haydel, 1983). García and Morshedi published a more comprehensive description several years later (García and Morshedi, 1986).

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 began with a clear and concise presentation of the unconstrained DMC algorithm, including an interesting discussion of tuning. Their experience showed that the DMC algorithm was closed loop stable when the prediction horizon was set long enough to include the steady-state effect of all computed input moves. This is supported by a rigorous proof presented by García and Morari (García and Morari, 1982) which shows that the DMC algorithm is nominally stabilizing for a sufficiently large prediction horizon.

García and Morshedi then 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 any reasonable problem and so the resulting optimization problem is convex. This means that a solution can be found readily using standard commercial optimization codes.

The default QDMC algorithm requires strict enforcement of input and output constraints at each point of the prediction horizon. Constraints that are strictly enforced are referred to as *hard constraints*. This is illustrated in Figure 10. In practice García and Morshedi report that hard output constraints are typically required to be satisfied over only a portion of the horizon which they refer to as the *constraint window*. The constraint window generally starts at some point in the future and continues on to steady state. They report that if non-minimum phase dynamics are present, performance is improved by pushing the constraint window farther into the future. This amounts to ignoring hard output constraints during the initial portion of the closed loop response. It is interesting that Rawlings and Muske recently arrived at the same solution to ensure feasibility of their infinite-horizon algorithm (Rawlings and Muske, 1993) after a careful theoretical analysis. They show that output constraints can be made feasible by relaxing them for a finite time j_1, and they derive an upper bound for j_1.

García and Morshedi report another option for handling output constraints that may be useful when non-minimum phase dynamics are present. When output constraint violations are predicted to occur, one can attempt to minimize the violation in a least-squared sense. This is the *soft constraint* concept illustrated in the middle plot of Figure 10. García and Morshedi described an approximate implementation of the soft constraint concept using a setpoint approximation, illustrated at the bottom of Figure 10. The setpoint approximation idea is to guess a priori where the optimal solution will require a constraint violation, and penalize this deviation by adding a setpoint that forces the output to stick to the constraint boundary. One way to guess where output violations will occur is to examine output predictions based on the optimal input solution from the previous time step. Because it is difficult to guess where the true solution at the current time step will require a constraint violation, the setpoint approximation method is generally sub-optimal.

A true soft constraint can be implemented by adding a slack variable to an inequality constraint, and then adding the slack variable to the objective function to be minimized. This approach has been studied by several researchers, including Ricker et al. (Ricker, Subrahmanian and Sim, 1988), Zafiriou and Chiou (Zafiriou and Chiou, 1993), and Genceli and Nikolaou (Genceli and Nikolaou, 1993). Zheng and Morari (Zheng and Morari, 1995) recently analyzed an infinite horizon MPC algorithm with soft output constraints implemented in this way. They show that global asymptotic stability can be guaranteed provided that the plant is not unstable.

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 12x12 (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.

IDCOM-M, HIECON, and PCT

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. Wouldn't 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 Figure 4, 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 manipulated variables (MV's) as controlled variables (CV'S), 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 MV's available than there are CV's 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 CV's than MV's; 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 tradeoffs 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 commented on this problem in their text on *Fundamental Process Control* (Prett and Garcìa, 1988):

> *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 and Setpoint, Inc. to develop a new version of the IDCOM algorithm. 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 Adersa version was referred to as HIECON (Hierarchical Constraint Control). The IDCOM-M controller was first described in a paper by Grosdidier et al. (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 and 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 single future point in time for each output, called the coincidence point, 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

The IDCOM-M controller retains the linear impulse response plant model used by the original IDCOM algorithm. The IDCOM-M controller allows the capability to include purely integrating responses, however. These are assumed to describe the response of the first order derivative of the output with respect to time.

The IDCOM-M algorithm includes a controllability supervisor which decides, based on the current set of available inputs and outputs, which outputs can be independently controlled. The selection is based on the effective condition number of the plant gain matrix; a list of controllability priorities is used to determine which outputs to drop from the problem if an ill-conditioned set is encountered.

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.

In many cases the solution to the output optimization is not unique. When additional degrees of freedom are present a second input optimization is performed. A quadratic input objective function is minimized subject to equality constraints that preserve the outputs found in the output optimization. The inputs are driven as closely as possible to their *Ideal Resting Values* (IRV's) which may come, for example, from an overlying steady-state optimizer. By default the IRV for a given input is set to its current measured value. The input optimization makes the most effective use of available degrees of freedom without altering the optimal output solution.

The IDCOM-M calculation is greatly simplified by computing a single move for each input. This *input blocking* assumption results in a loss of performance but provides additional robustness to modeling errors. In practice this has been an acceptable trade-off. Badgwell has analyzed the robustness properties of input-blocking for the SISO case (Badgwell, 1995).

In the IDCOM-M context, constraints are divided into hard and soft categories, with the understanding that hard constraints must be ranked in order of priority. When the calculation becomes infeasible, the lowest priority hard constraint is dropped and the calculation is repeated. One can specify several constraints to have the same priority, and it is possible to require that the control turn itself off and notify the operator if constraints above a given priority level cannot be enforced.

Grosdidier et al. (Grosdidier et al., 1988) describe the flow of a typical calculation:

- Determine available process inputs and outputs
- Determine the list of controllable outputs (controllability supervisor)
- Output Optimization
- Input Optimization

Grosdidier et al. 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.

Engineers at Profimatics addressed similar issues in the development of their PCT algorithm (Predictive Control Technology). This alogrithm also uses constraint prioritization to recover from infeasibility. Another interesting feature is known as *predict-back*, in which unmeasured disturbances that enter lower level PID loops are estimated and used for feedforward control. This feature is very useful when a lower level PID loop has slow dynamics, where the predict-back estimate helps the MPC controller respond to an unmeasured disturbance much faster.

The IDCOM-M algorithm is one of several that represent a *third generation* of MPC technology; others include the PCT algorithm sold by Profimatics, the RMPC controller developed by Honeywell, and the PFC algorithm developed by Adersa. 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, and allows for a wider range of process dynamics and controller specifications.

Survey of MPC Technology Products

Commercial MPC technology has developed considerably since the introduction of third generation technology nearly a decade ago. We recently surveyed five MPC vendors in order to assess the current status of commercial MPC technology. We believe that this list is representative in that the technology sold by these companies represents the industrial state of the art;

we fully recognize that we have omitted some major MPC vendors from our survey. 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 (Shell, Exxon, etc.). These MPC packages are either well-known to academic researchers (e.g., QDMC from Shell Oil) or not known at all for proprietary reasons.

The companies surveyed and their product names and acronyms are listed in Table 1. Initial data were collected from industrial vendors of MPC technology using a written survey. Blank copies of the survey 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 survey are summarized in Tables 2, 3 and 4. 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 choice of an appropriate MPC package for a particular application is a complex question that must be answered on a case by case basis; such issues are beyond the scope of this paper.

We focus here on the main aspects of the control and identification technology. We fully understand that a sound industrial offering must address many needs not necessarily related to the mathematics of the algorithms; these include software and hardware compatibility, user interface requirements, personnel training, and configuration and maintenance issues. It should also be clear that the descriptions given here are necessarily incomplete, since every MPC product has proprietary features. With this understanding in mind, we first discuss the process models at the core of MPC technology and then describe the details of a typical MPC calculation. Subsequent sections describe how different MPC vendors approach the different aspects of implementing MPC technology.

Process Models

The mathematical form of the process model defines the scope of an MPC algorithm. Tables 3 and 4 show that a wide variety of model forms are used in industrial MPC algorithms. All of the control and identification algorithms described here use time-invariant models. A general nonlinear discrete-time state space model may be described as

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k, \mathbf{w}_k) \tag{7}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) + \xi_k \tag{8}$$

where $\mathbf{u}_k \in \Re^{m_u}$ is a vector of m_u MV's, $\mathbf{y}_k \in \Re^{m_y}$ is a vector of m_y CV's, $\mathbf{x}_k \in \Re^n$ is a vector of n state variables, $\mathbf{v}_k \in \Re^{m_v}$ is a vector of m_v measured DV's, $\mathbf{w}_k \in \Re^{m_w}$ is a vector of m_w unmeasured DV's or noise.

The PFC algorithm is the only one considered in this survey that allows for nonlinear and unstable linear internal models. Nonlinear dynamics can be entered in the form of the nonlinear state space model shown above. The PFC algorithm uses transfer functions or ARX models to describe linear unstable dynamics. The remaining MPC products are designed based on Linear Time-Invariant (LTI) process models with stable or integrating dynamics. Nonlinearities may be accounted for in an approximate way by using a local linear model or nonlinear transformation of a specific CV. The SMC-Idcom algorithm, for example, allows the model gains to be adjusted on-line.

A very general discrete-time LTI model form is the linear state space model:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}_u\mathbf{u}_k + \mathbf{B}_v\mathbf{v}_k + \mathbf{B}_w\mathbf{w}_k \tag{9}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \xi_k \tag{10}$$

An equivalent transfer function model in the form of matrix fraction description(Kailath, 1980) can be written as:

$$\begin{aligned}\mathbf{y}_k = {} & [\mathbf{I} - \mathbf{\Phi_y}(\mathbf{q^{-1}})]^{-1}[\mathbf{\Phi_u}(\mathbf{q^{-1}})\mathbf{u_k} + \mathbf{\Phi_v}(\mathbf{q^{-1}})\mathbf{v_k} + \\ & \mathbf{\Phi}_w(q^{-1})\mathbf{w}_k] + \xi_k\end{aligned} \tag{11}$$

where q^{-1} is a backward shift operator. The output error identification approach (Ljung, 1987) minimizes the measurement error ξ_k, which results in nonlinear parameter estimation. Multiplying $[\mathbf{I} - \mathbf{\Phi_y}(\mathbf{q^{-1}})]$ on both sides of the above equation results in an *autoregressive model with exogenous inputs* (ARX),

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

where

$$\zeta_k = [\mathbf{I} - \mathbf{\Phi_y}(\mathbf{q^{-1}})]\xi_\mathbf{k} \tag{13}$$

The equation error identification approach minimizes ζ_k, which is colored noise even though the measurement noise ξ_k is white. For a stable system, a Finite Impulse Response (FIR) model can be derived as an approximation to the transfer function model:

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

This model form is used by the SMC-Idcom, HIECON, and OPC 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:

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

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

$$\mathbf{y}_k = \sum_{i=1}^{k} \mathbf{S}_i^u \Delta \mathbf{u}_{k-i} + \sum_{i=1}^{k} \mathbf{S}_i^v \Delta \mathbf{v}_{k-i} + \sum_{i=1}^{k} \mathbf{S}_i^w \Delta \mathbf{w}_{k-i} + \xi_k \quad (16)$$

where $\mathbf{S}_0 = \mathbf{0}$ and $\mathbf{S}_j = \mathbf{S}_N$ for $j > N$. Note that the summation goes from the initial time to the current time k. The FSR model is used by the DMC and RMPCT algorithms. The FIR model is related to the FSR model through:

$$\mathbf{H}_i = \mathbf{S}_i - \mathbf{S}_{i-1} \quad (17)$$

The SMC-Idcom and RMPCT algorithms also provide the option to enter Laplace transfer function models. All of the algorithms allow control of processes with integrating dynamics, either by modeling the time derivative of the output response or by using a modified feedback procedure.

A General MPC Control Calculation

MPC controllers are designed to drive the process from one constrained steady state to another. They may receive an optimal steady-state operating point from an overlying optimizer, as shown in Figure 2, or they may compute an 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 CV's to their steady-state optimal values (dynamic output optimization)
3. drive the MV's to their steady-state optimal values using remaining degrees of freedom (dynamic input optimization)
4. prevent excessive movement of MV's
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. Table 3 summarizes how each of the MPC vendors has accomplished this translation.

Figure 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 (DV's and MV's) and process outputs (CV's). 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 following sections describe other aspects of the calculation in greater detail.

Output Feedback

The model update step is where feedback enters the loop. All of the industrial MPC algorithms surveyed here use the same form of feedback for stable processes, based on comparing the current measured process output $\mathbf{y}_k^m$ to the current predicted output $\mathbf{y}_k$:

$$\mathbf{b}_k = \mathbf{y}_k^m - \mathbf{y}_k \quad (18)$$

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

$$\mathbf{y}_{k+j} = \mathbf{g}\left(\mathbf{x}_{k+j}\right) + \mathbf{b}_k \quad (19)$$

This form of feedback is equivalent to assuming an output disturbance that remains constant for all future time (Morari and Lee, 1991; Lee, Morari and García, 1994). Rawlings et al. (Rawlings et al., 1994) show that this method of feedback removes steady-state offset, which provides theoretical support for its use.

Variations of this basic feedback approach are used for the case of integrating dynamics. In RMPCT, for example, both a model bias and the rate of change of the bias are used for CV's that have integrating elements. DMC uses a rotation factor to combine a contribution from the bias term with a contribution from the rate of change of the bias term.

Additional practical details of the bias term calculation should be noted. The bias calculation may be filtered to remove high frequency noise; the RMPCT and SMC-Idcom algorithms provide this option. 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 one can 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. The DMC and SMC-Idcom algorithms provide this feature.

Determining the Controlled Sub-process

Once the model has been updated the controller must determine which MV's can be manipulated and which CV's 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 Figure 4.

The RMPCT and DMC algorithms provide an additional mechanism to prevent low level control saturation by including the low level control outputs (e.g., valve position) in the control formulation as additional CVs. These CV's are then forced to stay within high and low saturation limits by treating them as *range* or *zone* control variables. In this configuration, the number of MV's is typically less than the number of CV's, which include both range CV's and setpoint CV's. However, the number of setpoint CV's is typically less than or equal to the number of MV's. If a range CV is well within the saturation constraints, it has no effect on the objective function. In this case, the control configuration reduces to a typical fat or square plant. It should be noted that other CV's which need not to be controlled tightly at a setpoint value (e.g., surge tank level) are also treated as range CV's. When these CV's are within constraint limits, no MV action is required for these CV's. This releases additional degrees of freedom to drive remaining CV's or MV's to their targets.

The DMC, RMPCT, and SMC-Idcom algorithms distinguish between a *critical* CV failure and a *non-critical* CV failure. If a non-critical CV fails, the DMC controller completely removes it from the control calculation. The RMPCT and SMC-Idcom 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 non-critical CV fails for a specified period of time, RMPCT drops this CV from the control objective function.

If a critical CV fails, the DMC and RMPCT controllers turn off immediately. The SMC-Idcom algorithm, however, tries to maintain control for the part of the process that is not affected by the critical CV failure. The SMC-Idcom algorithm also allows a sensor to be temporarily turned off for calibration without interrupting control.

The OPC algorithm does not distinguish between critical and non-critical CV's; if a CV fails, its measurement is replaced with a model estimate.

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.

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 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. For a process with gain matrix G, the condition number of $G^T G$ provides a measure of process ill-conditioning; a high condition number means that small changes in the future error vector 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 subprocesses 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 internal model if necessary. Three strategies are currently used by MPC controllers to accomplish this; singular value thresholding, controlled variable ranking and input move suppression.

The Singular Value Thresholding (SVT) method used by the RMPCT controller involves decomposing the process model using a singular value decomposition. Singular values below a threshold magnitude are discarded, and a process model with a much lower condition number is then reassembled and used for control. The neglected singular values represent the direction along which the process hardly moves even if a large MV change is applied; the SVT method gives up this direction 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 these singular values is small comparing to model uncertainty, it may be better to neglect them anyway. After thresholding, the collinear CV's are approximated with the principal singular direction. In the case of two collinear CV's, for example, this principal direction is a weighted average of the two CV's. 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 and the results will be approximately equivalent to the priority approach used in SMC-Idcom.

The SMC-Idcom algorithm addresses this issue using a user-defined set of CV controllability ranks. When a high condition number is detected, the controller drops low priority CV's until a well-conditioned sub-process remains. The sub-process will be controlled without erratic input movement but the low priority CV's will be uncontrolled. Note, however, that if a low priority CV is dropped because it's open loop response is close to that of a high priority output, it will follow the high priority CV and will therefore still be controlled in a loose sense. In the case of two collinear CV's having no differentiable priority, it may be desirable to use an weighted average of the two.

Controllers that use input move suppression, such as the DMC and OPC algorithms, 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 limit of infinite move suppression the condition number becomes one for all sub-processes. There probably exists a set of finite move suppression factors which guarantee that all sub-processes have a condition number greater than a desired threshold value. In the case of two collinear CV's, 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.

Local Steady-State Optimization

The DMC, SMC-Idcom, and RMPCT controllers split the control calculation into a local steady-state optimization followed by a dynamic optimization. Optimal steady-state targets are computed for each input and output; these are then passed to a dynamic optimization to compute the optimal input vector. This should not be confused with the more comprehensive nonlinear optimization that takes place above the MPC algorithm in the plant control hierarchy (see Figure 2). The local steady-state optimization uses a linear 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 linear dynamic model used in the dynamic optimization.

The DMC controller uses an LP to do the local steady-state optimization. The optimization is carried out subject to hard input and soft output constraints. The LP is used primarily to enforce steady-state input and output constraints and to determine optimal steady-state input and output targets for the thin and fat plant cases of Figure 4. Input optimization in the fat plant case is accomplished using economic factors which describe the cost of using each input. In the thin plant case the output error trade-offs are evaluated using slack variable weights.

The RMPCT and PFC algorithms provide an additional level of flexibility by allowing for both linear and quadratic terms in the steady-state objective function. They also includes hard input and output constraints.

The SMC-Idcom algorithm solves the local steady-state optimization problem using a sequence of quadratic programs. CV's 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 CV's by grouping them together at the same priority level. The calculation proceeds by optimizing the highest priority CV's first, subject to hard and soft output constraints on the same CV's and all input hard constraints. Subsequent optimizations preserve the future trajectory of high priority CV's 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.

Dynamic Optimization Objectives

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

$$\begin{aligned} J \;=\; & \sum_{j=1}^{P} \|\mathbf{e}^y_{k+j}\|^2_{\mathbf{Q}_j} + \sum_{j=0}^{M-1} \|\Delta \mathbf{u}_{k+j}\|^2_{\mathbf{S}_j} + \\ & \sum_{j=0}^{M-1} \|\mathbf{e}^u_{k+j}\|^2_{\mathbf{R}_j} \end{aligned} \tag{20}$$

subject to a model constraint:

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

and subject to inequality constraints:

$$\begin{aligned} \underline{\mathbf{y}}_j \;\le\; & \mathbf{y}_{k+j} \;\le\; \bar{\mathbf{y}}_j && \forall\, j = 1, P \\ \underline{\mathbf{u}} \;\le\; & \mathbf{u}_{k+j} \;\le\; \bar{\mathbf{u}} && \forall\, j = 0, M-1 \\ \Delta\underline{\mathbf{u}} \;\le\; & \Delta\mathbf{u}_{k+j} \;\le\; \Delta\bar{\mathbf{u}} && \forall\, j = 0, M-1 \end{aligned}$$

The objective function 20 involves three conflicting contributions. Future output behavior is controlled by penalizing deviations from a desired response, defined by $\mathbf{e}^y_{k+j}$, over a prediction horizon of length P. Future input behavior is controlled using input penalties defined by $\mathbf{e}^u_{k+j}$ 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 relative importance of the objective function contributions are controlled by setting the time dependent weight matrices $\mathbf{Q}_j$, $\mathbf{S}_j$, and $\mathbf{R}_j$; these are assumed to be positive semi-definite.

The solution is a set of M input adjustments:

$$\mathbf{u}^M = (\mathbf{u}_k,\ \mathbf{u}_{k+1}, \ldots, \mathbf{u}_{k+M-1}) \tag{21}$$

The first input $\mathbf{u}_k$ is injected into the plant and the calculation is repeated.

The DMC, SMC-Idcom, RMPCT, PFC and OPC algorithms use a single quadratic objective function similar to 20. The DMC and OPC algorithms penalize the last input in order to drive the system towards the optimal steady state:

$$\begin{aligned} \mathbf{R}_j &= \mathbf{0}; \forall j < M-1 \\ \mathbf{R}_{M-1} &\neq \mathbf{0} \end{aligned} \tag{22}$$

If the final input weight is large enough and the process is stable, it is approximately equivalent to having a terminal state constraint. If the dynamic solution is significantly different from the LP targets, which means the terminal states are not effectively constrained, the DMC controller will be turned off. This setting may provide nominal stability for DMC controller; further analysis is necessary to firmly establish this result. The final input weights are also applicable to integrating processes where the derivative of the integrator is driven to zero.

The SMC-Idcom, RMPCT, HIECON and PFC controllers do not penalize input moves directly. The SMC-Idcom, HIECON and PFC algorithms use a predefined 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:

$$\begin{aligned} (\mathbf{u}^M, \mathbf{y}^r) &= \arg\min \sum_{j=1}^{P} \|\mathbf{y}_{k+j} - \mathbf{y}^r_{k+j}\|^2_{\mathbf{Q}} + \\ &\quad \|\mathbf{u}_{k+M-1} - \mathbf{u}_{ss}\|^2_{\mathbf{R}} \end{aligned} \tag{23}$$

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 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 CV's and MV's. In the case of an infinite number of solutions, which is also typical due to "relaxing" the trajectory, a minimum norm solution to the MV's is obtained due to the use of singular value thresholding. This provides a similar effect to the move suppression used in DMC.

Using a single objective function means that trade-offs between the three contributions must be resolved using the the relative weight matrices $\mathbf{Q}_j$, $\mathbf{S}_j$, and $\mathbf{R}_j$. The HIECON algorithm resolves conflicting dynamic control objectives by solving a sequence of multiple 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 20 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 20 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.

Constraint Formulations

There are basically two types of constraints used in industrial MPC technology; hard and soft. These are illustrated in Figure 10. Hard constraints, shown in the top of Figure 10 are those which should never be violated. Soft constraints, shown in the middle of Figure 10 are those for which some violation may be allowed; the violation is generally subjected to a quadratic penalty in the objective function.

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 MV's at any single control execution. The PFC algorithm also accommodates maximum and minimum input acceleration constraints which are useful in mechanical servo control applications. The SMC-Idcom, RMPCT, HIECON and OPC algorithms perform rigorous optimizations subject to the hard input constraints. The DMC and PFC algorithms, however, enforce input hard constraints in a sub-optimal manner in the sense that the solution does not generally satisfy the Karush-Kuhn-Tucker (KKT) conditions. In the DMC 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.

The SMC-Idcom, RMPCT, HIECON and PFC algorithms enforce output soft constraints in an optimal manner as part of the dynamic optimization. The DMC and OPC algorithms, however, use a setpoint approximation to enforce soft constraints. This is illustrated at the bottom of Figure 10. 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 LP target is used as the setpoint instead.

The SMC-Idcom and RMPCT algorithms consider hard output constraints only in the local steady-state optimization. This prevents problems which may occur due to enforcing output constraints early in the prediction horizon of the dynamic optimization (see the discussion in Zafiriou's paper, for example (Zafiriou, 1990)). Only the HIECON and PFC algorithms consider hard output constraints in the dynamic optimization. Hard output constraints are ranked in order of priority so that low priority constraints can be dropped when the problem becomes infeasible. The PFC algorithm also considers hard output constraints in the steady-state optimization.

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 Figure 6. The shaded areas correspond to the $\mathbf{e}^y_{k+j}$ and $\mathbf{e}^u_{k+j}$ terms in 20. All of the controllers provide the option to drive the CV's to a fixed setpoint, with deviations on both sides penalized in the objective function. In practice this type of specification is very aggressive and may lead to very large input adjustments, unless the controller is detuned in some fashion. This is particularly important when the internal model differs significantly from the process. The DMC and OPC 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 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 algorithm, for example, uses a *dynamic weighting* strategy to implement the zone using objective function penalties. When the CV is predicted to lie within its zone, its weight is set to zero so the controller will ignore it. If the CV is near one edge of the zone, its weight increases gradually depending on how close it is to the constraint. If the CV violates the constraint, the weight is set to a large value. This causes CV's which violate or almost violate constraints to be driven towards the zone boundaries.

The SMC-Idcom, HIECON, and PFC algorithms provide a CV reference trajectory option. The reference trajectory idea is to bring the CV up to the setpoint more slowly, in order to avoid overshoot. A first order curve is drawn from the current CV value to the setpoint, with the speed of the response determined by a trajectory time constant. Future CV deviations from the reference trajectory are penalized. In the limit of a zero time constant the reference trajectory reverts back to a pure setpoint; for this case, however, the controller would be sensitive to model mismatch unless some other strategy such as move suppression is also being used. 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 and Zafiriou, 1989). In general, as the reference trajectory time constant increases, the controller is able to tolerate larger model mismatch.

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 Figure 6, 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 SMC-Idcom algorithm uses a variation of the funnel when a zone CV falls out of it's range. In this case a funnel is defined using a one-sided reference trajectory that terminates just inside the zone boundary.

A potential advantage of the funnel is illustrated in Figure 7. Consider the case of a reference trajectory, shown on the left side of Figure 7. If a disturbance causes the predicted future CV to reach the setpoint more quickly than the reference trajectory allows, the controller will take action to bring the CV back down to the defined trajectory. In the same situation the funnel shown on the right side of Figure 7 would take no action at all.

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. The SMC-Idcom algorithm also provides an option to ramp the MV along a straight line to it's optimal value.

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 Figure 8. The finite output horizon formulation is used by all of the algorithms discussed in this paper. The length of the horizon P is a basic tuning parameter for these controllers, and is generally set long enough to capture the steady-state effects of all computed future MV moves.

The SMC-Idcom and PFC controllers allow the option to simplify the calculation by considering only a subset of points in the prediction horizon. This concept is illustrated at the bottom of Figure 8. These points are called *coincidence points* because the desired and predicted future outputs are required to coincide at these points. A separate set of coincidence points can be defined for each output, which is useful when one output responds quickly relative to another. 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 Figure 9. The DMC, RMPCT, and OPC algorithms compute a set of future moves to be spread over a finite *control horizon*, as shown at the top of Figure 9. The length of the control horizon M is another basic tuning parameter for these controllers. Increased performance is obtained as M increases, at the expense of additional computation.

The SMC-Idcom and HIECON algorithms compute a single future input move, as shown in the middle of Figure 9. This greatly simplifies the calculation for these algorithms, which is helpful because they solve a series of optimization problems at each control execution. The use of a single move involves a sacrifice of closed loop performance that is difficult to quantify, however.

The PFC controller parameterizes the input function using a set of polynomial basis functions. A possible solution is illustrated at the bottom of Figure 9. This allows a relatively complex input profile to be specified over a large (potentially infinite) control horizon, using a small number of unknown parameters. This may provide an advantage when controlling nonlinear systems. Choosing the family of basis functions establishes many of the features of the computed input profile; this is one way to ensure a smooth input signal, for example. If a polynomial basis is chosen then the order can be selected so as to follow a polynomial setpoint signal with no lag. This feature is important for mechanical servo control applications.

Control Design and Tuning

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

1. From the stated control objectives, define the size of the problem, and determine the relevant CV's, MV's, and DV's
2. Test the plant systematically by varying MV's and DV's; capture and store the real-time data showing how the CV's respond
3. Derive a dynamic model from the plant test data using an identification package
4. Configure the MPC controller and enter initial tuning parameters
5. Test the controller off-line using closed loop simulation to verify the controller performance.
6. Download the configured controller to the destination machine and test the model predictions in *open-loop* mode
7. Commission the controller and refine the tuning as needed.

All of the MPC packages surveyed here provide software tools to help with the control design, process identification and closed loop simulation steps. A significant amount of time is currently spent at the closed loop simulation step to verify acceptable performance 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.

Controller tuning is always required not only for stability, but also for trade-offs between performance and robustness. It is true that a high performance controller will not be robust with respect to model mismatch, but a low performance controller is not necessarily robust. Most MPC products provide tuning "knobs" to de-tune the controller. In DMC and OPC, two types of tuning parameters are used: (1) move suppression factors, which are weights on $\Delta\mathbf{u}$ and (2) equal concern error factors, which are the inverse of output weights. The move suppression factors change the aggressiveness of the controller, while the equal concern error factors normalize the importance of each CV.

The SMC-Idcom, HIECON, and PFC controllers use the time constant of the reference trajectory as the main tuning parameter. Smaller time constants demand more aggressive control, while larger time constants result in less aggressive action. One may start with the trajectory time constant equal to the open loop time constant of the CV, then refine the tuning based on performance/robustness trade-offs.

A performance ratio is provided in RMPCT which is defined as the ratio of the required closed loop settling time to the weighted-average open loop settling time. The performance ratio is used to determine the length of the funnel, which is somewhat similar to the settling time of a setpoint trajectory. A performance ratio equal to one means that the closed loop settling time is equal to the open loop settling time. A performance ratio less than one results in a more aggressive controller. Only one tuning parameter per CV needs to be specified. Independent tuning is available in RMPCT for feedforward control, which allows the user to achieve faster response in feedforward control than in setpoint tracking.

The RMPCT package provides 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 remaining MPC controllers are performed by closed loop simulation.

All of the MPC packages provide a way to test final controller performance by closed loop simulation. It is particularly important to check the response of the final controller with respect to key constraint violations. It is almost impossible, however, to test all possible situations of active constraints for a realistic problem. The problem becomes even more intractable if one wishes to test performance in the presence of model mismatch. This is one place where academic research can help industrial MPC practice significantly. Rawlings and Muske (Rawlings and Muske, 1993), for example, have shown that their infinite horizon MPC algorithm is nominally stabilizing in the presence of constraints if the initial optimization problem is feasible. Additional work is needed to extend this result to the case of an imperfect plant model. Vuthandam et al. (Vuthandam, Genceli and Nikolaou, 1995) have done this already for a modified QDMC algorithm; their results should be useful to many industrial practitioners.

Identification Test Protocol

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. DMCC believes that the plant test is the single most important phase in the implementation of DMC 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 such as DMI and SMC-Test allow several MV's to change simultaneously with uncorrelated signals for different MV's. For DMI, 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 10 to 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. The SMC-Model package provides a convenient interface for such calculations.

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 or pulse test in combination with PRBS (e.g., OPC, SMC-Test, and RMPCT). As a special case, OPC allows one to use operating data as the ultimate source to build steady-state models for quality variables (Lines, Hartlen, Paquin, Treiber, de Tremblay and Bell, 1993). These variables may be measured by an analytical sensor or laboratory analysis. The following steady-state model is used for analyzer variables:

$$\mathbf{y}_k = \mathbf{y}_{k-1} + \mathbf{K}(\mathbf{u}_k - \mathbf{u}_{k-1}) + \mathbf{K}_f(\mathbf{y}_k^m - \mathbf{y}_{k-d}) \quad (24)$$

where $\mathbf{K}$ is the steady-state gain matrix identified from the operating data. The vector $\mathbf{y}_k^m$ represents current analyzer measurements which have a time delay represented by d. $\mathbf{K}_f$ is an adjustable feedback gain matrix to correct the model estimation. The entire relation may be thought of as a steady-state Kalman filter. The same mechanism can be applied to dynamic models as well.

Model Forms and Parameter Estimation

The model forms used in identification are generally more diversified than those used for control. Table 4 lists various model forms used in different MPC products. The identification approaches in the MPC products are mainly based on minimizing the following least squares criterion,

$$J = \sum_{k=1}^{L} \|\mathbf{y}_k - \mathbf{y}_k^m\|^2 \tag{25}$$

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. 12,

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

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

$$\mathbf{y}_k = \mathbf{\Phi}_y(q^{-1})\mathbf{y}_k + \mathbf{\Phi}_u(q^{-1})\mathbf{u}_k + \mathbf{\Phi}_v(q^{-1})\mathbf{v}_k \tag{27}$$

The equation error approach formulates a linear least squares problem, but the estimates are biased even though the measurement noise ξ 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 DMI, 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 and Yocum, 1991). However, the velocity form is sensitive to high frequency noise. Therefore, pre-smoothing for the process data is done in DMI before fitting models to the data. The FIR coefficients are then converted into FSR coefficients for control.

RMPCT adopts a three-step approach: (i) identify an FIR model using least squares; (ii) fit the FIR coefficients to a low-order ARX model to smooth out induced variance due to possible overparametrization in the FIR model. The output error approach is used to fit the ARX model and a Gauss-Newton method is used to estimate the parameters; and (iii) convert the ARX models into Laplace transfer functions. 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. OPC provides options to identify FIR and ARMAX models directly from data. The ARMAX model is used on a single-input-single-output basis. Note that OPC estimates the noise dynamics by using moving average terms in the model. These models are finally converted into FIR models for control.

It appears that different philosophies exist in choosing the model forms for identification. DMC believes that complex dynamics can only be identified with high-order FIR models and worries little about overparametrization. A few other products are more concerned with overparametrization which may induce unrealistic variance, and use low order transfer function models as an alternative.

Model uncertainty bounds are provided in several products such as OPC, RMPCT, and SMC-Model. 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 products apply nonlinear transformations to variables that exhibit significant nonlinearity. For example, a logarithm transformation is often performed on composition variables.

Applications Summary

Table 2 summarizes the reported applications experience of each MPC vendor surveyed. Note that this is a count of completed MPC *applications* reported by each vendor; for a particular problem one vendor may use a single large controller while another may use several smaller controllers. In some cases a single controller is designed and then subsequently used in thousands of copies; this may happen with an automobile application, for example. Note also that this is a count of MPC applications performed by the vendors themselves; this does not include in-house applications performed by licensees of vendor technology. Vendors were given wide latitude to report the numbers in any distribution that they wished to use.

The total number of reported MPC applications is currently over 2200. All of the vendors report a considerable number of applications in progress so it is likely that this number will continue to increase rapidly. Table 2 shows that MPC technology can now be found in a wide variety of application areas. The majority of applications (67 %) are in refining, 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.

The DMC Corporation reports the largest total number of applications, at 600, with the other four vendors reporting between 350 and 450 applications each. Table 2 shows that DMC, Setpoint, Honeywell and Treiber Controls are highly focused in refining and petrochemicals, with a handful of applications in other areas. Adersa appears to have the broadest range of experience with applications in the food processing, aerospace and automotive areas, among others.

The bottom of Table 2 shows that largest applications to date by each vendor, in the form of (outputs)x(inputs). The numbers show a difference in philosophy that is a matter of some controversy. The DMC Corporation recommends solving a large control problem with a single controller 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. This is an issue that needs further attention from the academic community.

Limitations of Existing Technology

Morari and Lee pointed out several problems with existing technology in their CPC-IV paper (Morari and Lee, 1991). Muske and Rawlings have also pointed out limitations of existing industrial MPC technology (Muske and Rawlings, 1993). These include:

- impulse and step response models are over-parameterized and limit application of the algorithm to strictly stable processes
- sub-optimal solution of the dynamic optimization
- sub-optimal feedback (constant output disturbance assumption)
- tuning is required to achieve nominal stability
- model uncertainty is not addressed adequately

Impulse and step response models are known to be over-parameterized. The dynamics of a first order process, for example, can be described by three numbers using a parametric model (gain, time constant and deadtime). An impulse response will typically require from 30 to 120 coefficients to describe the same dynamics. These difficulties can be overcome at the identification step by first identifying a low order model and then calculating the impulse response coefficients. However the impulse response model still requires more storage space than is necessary. 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 an ARX or state-space model.

Sub-optimal solutions to the dynamic optimization 20 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, unless it can be shown that the sub-optimal solution is always very nearly optimal. 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.

The bias update feedback technique used by industrial MPC controllers can be interpreted as assuming an output disturbance that remains constant for all future time (Morari and Lee, 1991; Muske and Rawlings, 1993). This is probably the best assumption that can be used in the absence of detailed disturbance information, but better feedback is possible if the distribution of disturbances can be characterized more carefully. Morari and Lee have shown how to extend MPC technology to achieve better feedback while still retaining the step response model (Morari and Lee, 1991). Muske and Rawlings have demonstrated how better performance can be achieved by using a state space model and an optimal state observer (Muske and Rawlings, 1993).

Tuning MPC controllers for stable operation in the presence of constraints may be difficult, even when the process model is perfect. This is why so much effort is spent on closed loop simulation prior to commissioning a controller. Rawlings and Muske addressed this issue directly in the development of their infinite horizon MPC algorithm (Rawlings and Muske, 1993). They used the LQR controller of Kalman (Eqn. 2) as a starting point. The Kalman LQR is stable for any reasonable choice of tuning parameters, due to the use of an infinite prediction horizon. They developed an infinite horizon MPC controller that is guaranteed to be closed loop stable, for the nominal case, *in the presence of constraints*. Feasibility of the initial QP is enough to guarantee constrained stabilizability. When the QP is infeasible, Muske and Rawlings propose giving up on all of the constraints for a short period of time, much like the constraint window idea used by the QDMC algorithm (García and Morshedi, 1986). Other options for recovering feasibility are also possible, such as dropping low priority constraints. Zheng and Morari (Zheng and Morari, 1995) recently analyzed an infinite horizon MPC algorithm which uses soft output constraints to avoid infeasibility. They show that the closed loop system using output feedback is stable for a strictly stable plant.

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 convenient 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. This is one area where academic research can help. Vuthandam et al. for example, have recently presented robust stability conditions for a modified QDMC algorithm (Vuthandam et al., 1995). More research is needed in this area.

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 (Froisy, 1994) to come up with a composite view of future MPC technology.

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 probably 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 horizons will include options for multiple moves or parameterization using basis functions.

Nonlinear MPC

MPC using nonlinear models is likely to become more common as users demand higher performance and new software tools make nonlinear models more readily available. Developing adequate nonlinear empirical models may be very challenging, however. Test signals such as PRBS that are adequate for linear models are not likely to provide adequate excitation of nonlinear systems. Also there is no model form that is clearly suitable to represent general nonlinear processes. Froisy (Froisy, 1994) points out that second order Volterra models may bridge the gap between linear empirical models and nonlinear fundamental models in the near future. Genceli and Nikolaou, for example, have studied the use of second order Volterra series with a modified QDMC controller (Genceli and Nikolaou, 1995). However, nonlinear empirical models such as Volterra series or neural networks do not seem to extrapolate well.

An alternative approach would be to use first-principles models developed from well known mass, momentum, and energy conservation laws. However, the cost of developing a reasonably accurate first-principles model is likely to be prohibitive until new software tools and validation procedures become available. Hybrid models that integrate steady state nonlinear first-principles models with dynamic empirical models (linear or nonlinear) may prove most promising for the near future. Gain-scheduling with linear dynamic models is an example of this approach.

From a theoretical point of view using a nonlinear model changes the control problem from a convex QP to a non-convex Non-Linear Program (NLP), the solution of which is much more difficult. There is no guarantee, for example, that the global optimum can be found. Bequette describes several approaches to solving the general nonlinear MPC problem in his review of nonlinear control for chemical processes (Bequette, 1991). Although solving the nonlinear MPC problem at each time step is much more difficult, Rawlings et al. (Rawlings et al., 1994) have shown that the nominal Lyapunov stability argument presented for linear models carries over to the general nonlinear case with minor modifications. The most important change is that a constraint must be included to zero the states at the end of the input horizon. They point out, however, that nonlinear MPC may require unexpected input adjustments. They present an interesting example in which a simple nonlinearity in the process model leads to a discontinuous feedback control law. This implies that tuning nonlinear MPC controllers may be very difficult, particularly for the case of model mismatch.

Adaptive MPC

A few adaptive MPC algorithms such as the GPC algorithm introduced by Clarke et al. have been proposed (Clarke, Mohtadi and Tuffs, 1987) but only a single adaptive MPC algorithm has reached the marketplace (STAR from Dot Products (Dollar, Melton, Morshedi, Glasgow and 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.

Robust Stability of 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 the powerful theoretical results recently presented for MPC with a perfect model (Muske and Rawlings, 1993; Zheng and Morari, 1995) will be extended to include model mismatch in the near future. This has already been accomplished for a modified QDMC algorithm using an impulse response model (Vuthandam et al., 1995). New robust stability guarantees will then be combined with uncertainty estimates from identification software to greatly simplify design and tuning of MPC controllers.

Robust Identification Schemes

The use of FIR models results in overparametrization. In practice this leads to some kind of engineering modification, such as a smoothness factor to "regularize" the model parameters. Although these modifications are proprietary, regularization (smoothness) and biased regression algorithms (ridge regression or partial least squares) are common approaches to dealing with overparametrization. An alternative approach would be to identify the process model based on parametric models, such as transfer function models or state space representation, then convert the model into desired form for the controller to use.

Conclusions

MPC technology has progressed steadily in the twenty two years since the first IDCOM and DMC applications. Survey data reveal approximately 2200 applications to date, with a solid foundation in refining and petrochemicals, and significant penetration into a wide range of application areas from chemicals to food processing.

Current generation MPC technology offers significant new capabilities but the controllers still retain, for the most part, an IDCOM-like or a DMC-like personality. The SMC-Idcom and HIECON algorithms are IDCOM-like controllers which have evolved to use multiple objective functions and ranked constraints. The DMC, RMPCT and OPC algorithms are DMC-like controllers that use a single dynamic objective function to evaluate control and economic trade-offs using weighting factors. The PFC controller inherits some of the IDCOM personality but is significantly different in that it can accommodate nonlinear and unstable processes and uses basis functions to parameterize the input function.

An important observation is that industrial MPC controllers almost always use empirical dynamic models identified from test data. The impact of identification theory on process modeling is perhaps comparable to the impact of optimal control theory on model predictive control. It is probably safe to say that MPC practice is one of the largest application areas of system identification. The current success of MPC technology may be due to carefully designed plant tests.

Another observation is that process identification and control design are clearly separated in current MPC technology. Efforts towards integrating identification and control design may bring significant benefits to industrial practice. For example, uncertainty estimates from process identification could be used more directly in robust control design. Ill-conditioned process structures could be reflected in the identified models and also used in control design.

Choosing an MPC technology for a given application is a complex question involving issues not addressed in this paper. It is the opinion of the authors that for most applications, a knowledgeable control engineer could probably achieve acceptable control performance using any of the packages discussed here, although the time and effort required may differ. If the process is nonlinear or unstable, or needs to track a complex setpoint trajectory with no offset, the PFC algorithm may offer significant advantages. If a vendor is to be selected to design and implement the 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 problems 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 an 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 could provide control practitioners and theoreticians with plenty of work in the foreseeable future.

Several technical advances have not yet been incorporated into industrial MPC technology. These include using an infinite prediction horizon to guarantee nominal closed loop stability, and using linear estimation theory to improve output feedback. In addition, robust stability conditions have been developed for a modified QDMC algorithm. It would seem that the company which first implements these advances will have a significant marketing and technical advantage.

The future of MPC technology is bright, with all of the vendors surveyed here reporting significant applications in progress. Next-generation MPC technology is likely to include multiple objective functions, an infinite prediction horizon, nonlinear process models, better use of model uncertainty estimates, and better handling of ill-conditioning.

Acknowledgments

The authors wish to thank the vendor representatives who responded to our MPC survey for their time and effort. We also wish to thank one anonymous reviewer and the following people for their constructive criticism of the initial draft: Jim Downs, Bjarne Foss, Brian Froisy, Pierre Grosdidier, Rob Hawkins, Evelio Hernandez, Joseph Lu, Jim Rawlings, Jaques Richalet and Steve Treiber.

References

Badgwell, T. A. (1995). Robust stability conditions for a SISO model predictive control algorithm. Manuscript in preparation.

Bequette, B. W. (1991). Nonlinear control of chemical processes: A review, *Ind. Eng. Chem. Res.* **30**: 1391–1413.

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

Cutler, C., Morshedi, A. and Haydel, J. (1983). An industrial perspective on advanced control, *AIChE Annual Meeting*, Washington, D.C.

Cutler, C. R. and Ramaker, B. L. (1979). Dynamic matrix control—a computer control algorithm, AIChE National Meeting, Houston, TX.

Cutler, C. R. and Ramaker, B. L. (1980). Dynamic matrix control—a computer control algorithm, *Proceedings of the Joint Automatic Control Conference*.

Cutler, C. R. and Yocum, F. H. (1991). Experience with the DMC inverse for identification, *in* Y. Arkun and W. H. Ray (eds), *Chemical Process Control—CPC IV*, Fourth International Conference on Chemical Process Control, Elsevier, Amsterdam, pp. 297–317.

DMC Corp. (1994). [DMC]TM: Technology Overview. Product literature from DMC Corp.

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

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

Froisy, J. B. and Matsko, T. (1990). Idcom-m application to the shell fundamental control problem, AIChE Annual Meeting.

García, C. E. and Morari, M. (1982). Internal model control. 1. A unifying review and some new results, *Ind. Eng. Chem. Proc. Des. Dev.* **21**: 308–323.

García, C. E. and Morshedi, A. M. (1986). Quadratic programming solution of dynamic matrix control (QDMC), *Chem. Eng. Commun.* **46**: 73–87.

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

Genceli, H. and Nikolaou, M. (1993). Robust stability analysis of constrained l_1–norm model predictive control, *AIChE J.* **39**(12): 1954–1965.

Genceli, H. and Nikolaou, M. (1995). Design of robust constrained model-predictive controllers with volterra series, *AIChE J.* **41**(9): 2098–2107.

Grosdidier, P., Froisy, B. and Hammann, M. (1988). The IDCOM-M controller, *in* T. J. McAvoy, Y. Arkun and E. Zafiriou (eds), *Proceedings of the 1988 IFAC Workshop on Model Based Process Control*, Pergamon Press, Oxford, pp. 31–36.

Honeywell, Inc. (1995). RMPCT Concepts Reference. Product literature from Honeywell, Inc.

Kailath, T. (1980). *Linear Systems*, Prentice-Hall, Englewood Cliffs, New Jersey.

Kalman, R. E. (1960a). Contributions to the theory of optimal control, *Bull. Soc. Math. Mex.* **5**: 102–119.

Kalman, R. E. (1960b). A new approach to linear filtering and prediction problems, *Trans. ASME, J. Basic Engineeering* pp. 35–45.

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

Lee, J. H. (1996). Recent advances in model predictive control and other related areas, *Chemical Process Control—CPC V*, CACHE, Tahoe City, California.

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

Lines, B., Hartlen, D., Paquin, F., Treiber, S., de Tremblay, M. and Bell, M. (1993). Polyethylene reactor modeling and control design, *Hydrocarbon Process.* pp. 119–124.

Ljung, L. (1987). *System Identification: Theory for the User*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Mayne, D. Q. (1996). Nonlinear model predictive control: An assessment, *Chemical Process Control—CPC V*, CACHE, Tahoe City, California.

Morari, M. and Lee, J. H. (1991). Model predictive control: The good, the bad, and the ugly, *in* Y. Arkun and W. H. Ray (eds), *Chemical Process Control—CPC IV*, Fourth International Conference on Chemical Process Control, Elsevier, Amsterdam, pp. 419–444.

Morari, M. and Zafiriou, E. (1989). *Robust Process Control*, Prentice-Hall, Englewood Cliffs, New Jersey.

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

Ohshima, M., Ohno, H. and 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.

Prett, D. M. and Garcìa, C. E. (1988). *Fundamental Process Control*, Butterworths, Boston.

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

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

Rawlings, J. B., Meadows, E. S. and Muske, K. R. (1994). Nonlinear model predictive control: A tutorial and survey, *ADCHEM '94 Proceedings, Kyoto, Japan*.

Rawlings, J. B. and Muske, K. R. (1993). Stability of constrained receding horizon control, *IEEE Trans. Auto. Cont.* **38**(10): 1512–1516.

Richalet, J., Rault, A., Testud, J. L. and Papon, J. (1976). Algorithmic control of industrial processes, *Proceedings of the 4th IFAC Symposium on Identification and System Parameter Estimation*, pp. 1119–1167.

Richalet, J., Rault, A., Testud, J. L. and 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 and W. H. Ray (eds), *Chemical Process Control—CPC IV*, Fourth International Conference on Chemical Process Control, Elsevier, Amsterdam, pp. 271–296.

Ricker, N. L., Subrahmanian, T. and Sim, T. (1988). Case studies of model-predictive control in pulp and paper production, *in* T. J. McAvoy, Y. Arkun and E. Zafiriou (eds), *Proceedings of the 1988 IFAC Workshop on Model Based Process Control*, Pergamon Press, Oxford, pp. 13–22.

Setpoint, Inc. (1993). SMC-Idcom: A State-of-the-Art Multivariable Predictive Controller. Product literature from Setpoint, Inc.

Vuthandam, P., Genceli, H. and Nikolaou, M. (1995). Performance bounds for robust quadratic dynamic matrix control with end condition, *AIChE J.* **41**(9): 2083–2097.

Zafiriou, E. (1990). Robust model predictive control of processes with hard constraints, *Comput. Chem. Eng.* **14**(4/5): 359–371.

Zafiriou, E. and Chiou, H.-W. (1993). Output constraint softening for SISO model predictive control, *Proceedings of the 1993 American Control Conference*, pp. 372–376.

Zheng, A. and Morari, M. (1995). Stability of model predictive control with mixed constraints, *IEEE Trans. Auto. Cont.* **40**: 1818–1823.

Tables and Figures

Table 1. Companies surveyed, their Product Names and Acronyms

Company	Acronym	Product Name (Function)
Adersa	HIECON	Hierarchical Constraint Control
	PFC	Predictive Functional Control
	GLIDE	(Identification package)
DMC Corp.	DMC	Dynamic Matrix Control
	DMI	Dynamic Matrix Identification
Honeywell	RMPCT	Robust Model Predictive Control Technology
Profimatics	PCT	Predictive Control Technology
Setpoint Inc.	SMCA	Setpoint Multivariable Control Architecture
	SMC-Idcom	(Multivariable control package)
	SMC-Test	(Plant test package)
	SMC-Model	(Identification package)
Treiber Controls	OPC	Optimum Predictive Control

Table 2. Summary of Reported MPC Vendor Applications by Areas (estimates based on vendor survey; estimates do not include applications by companies who have licensed vendor technology)

Area	DMC Corp.	Setpoint Inc.	Honeywell Profimatics	Adersa	Treiber Controls	Total
Refining	360	320	290	280	250	1500
Petrochemicals	210	40	40	-	-	290
Chemicals	10	20	10	3	150	193
Pulp and Paper	10	-	30	-	5	45
Gas	-	-	5	-	-	5
Utility	-	-	2	-	-	2
Air Separation	-	-	-	-	5	5
Mining/Metallurgy	-	2	-	7	6	15
Food Processing	-	-	-	41	-	41
Furnaces	-	-	-	42	-	42
Aerospace/Defense	-	-	-	13	-	13
Automotive	-	-	-	7	-	7
Other	10	20	-	45	-	75
Total	600	402	377	438	416	2233
First App	DMC:1985	IDCOM-M:1987 SMCA:1993	PCT:1984 RMPCT:1991	IDCOM:1973 HIECON:1986	OPC:1987	
Largest App	603x283	35x28	28x20	-	24x19	

Table 3. Comparison of Industrial MPC Control Technology

Company	DMC Corp	Setpoint Inc	Honeywell IAC	Adersa	Adersa	Treiber Controls
Algorithm	DMC	SMC-Idcom	RMPCT	HIECON	PFC	OPC
Model Forms[1]	SR L,S,I	IR,TF L,S,I	SR,TF,ARX L,S,I	IR L,S,I	SS,TF,ARX L,N,S,I,U	IR L,S,I
Feedback[2]	CD, ID	CD, ID	CD, ID	CD, ID	CD, ID	CD, ID
Rem Ill-cond[3]	IMS	RCV	SVT	-	-	IMS
SS Opt Obj[4]	L[I,O]	Q[O]...Q[I],R	Q[I,O]	-	Q[I,O]	-
SS Opt Const[5]	IH,OS	IH,OH,OS,R	IH,OH	-	IH,OH	-
Dyn Opt Obj[6]	Q[I,O,M],S	Q[I,O]	Q[I,O]	Q[O],Q[I]	Q[I,O],S	Q[I,O,M],S
Dyn Opt Const[7]	IH,OS	IH,OS	IH,OS	IH,OH,OS,R	IC,OH,OS,R	IH,OS
Output Traj[8]	S,Z	S,Z,RT,F	S,Z,F	S,Z,RT	S,Z,RT	S,Z
Output Horiz[9]	FH	FH,MP	FH	FH	FH,MP	FH
Input Param[10]	MM	SM	MM	SM	BF,SM	MM

[1] Model Form: (IR) finite Impulse Response, (SR) finite Step Response, (TF) Laplace Transfer Function, (SS) State-Space, (ARX) Auto-Regressive Exogenous input, (L) Linear, (N) Nonlinear, (S) Stable, (I) Integrating, (U) Unstable
[2] Feedback: (CD) Constant output Disturbance, (ID) Integrating output Disturbance
[3] Removal of Ill-conditioning: (RCV) Ranked Controlled Variables, (SVT) Singular Value Thresholding, (IMS) Input Move Suppression
[4] Steady-State Optimization Objective: (L) Linear, (Q) Quadratic, (I) Inputs, (O) Outputs, (...) multiple sequential objectives, (R) outputs Ranked in order of priority
[5] Steady-State Optimization Constraints: (IH) Input Hard maximum, minimum and rate of change constraints, (OH) Output Hard maximum and minimum constraints
[6] Dynamic Optimization Objective: (Q) Quadratic, (I) Inputs, (O) Outputs, (M) Input Moves, (S) Sub-optimal solution
[7] Dynamic Optimization Constraints: (IH) Input Hard maximum, minimum and rate of change constraints, (OH) Output Hard maximum and minimum constraints, (OS) Output Soft maximum and minimum constraints, (IC) Input Clipped maximum, minimum and rate of change constraints (sub-optimal), (R) constraints Ranked in order of priority
[8] Output Trajectory: (S) Setpoint, (Z) Zone, (RT) Reference Trajectory, Funnel (F),
[9] Output Horizon: (FH) Finite Horizon, (MP) Multiple Point
[10] Input Parameterization: (SM) Single Move, (MM) Multiple Move, (BF) Basis Functions

Table 4. Comparison of Industrial MPC Identification Technology

Product	Test Protocol	Model Form	Estimation Method	Uncertainty Bound
DMC	multi-steps	(velocity) FIR	modified LS	No
SMC-Model	PRBS, step	FIR, ARX	output error LS[1]	Yes
RMPCT	PRBS, step	FIR, ARX	LS, G-N[2]	Yes
Glide	non-PRBS	G(s)	GD[3],G-N, GM[4]	Yes
OPC	PRBS, step, pulse, operating data	FIR, ARMAX	LS	Yes

[1] LS: Least Squares
[2] G-N: Gauss-Newton
[3] GD: Gradient Descent
[4] GM: Global Methods

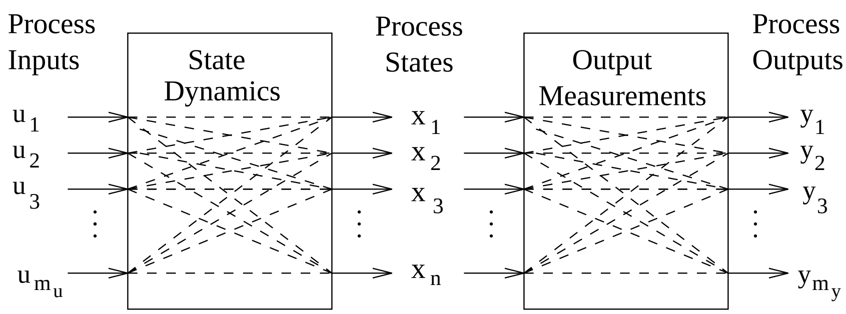

Figure 1. State-space process description.

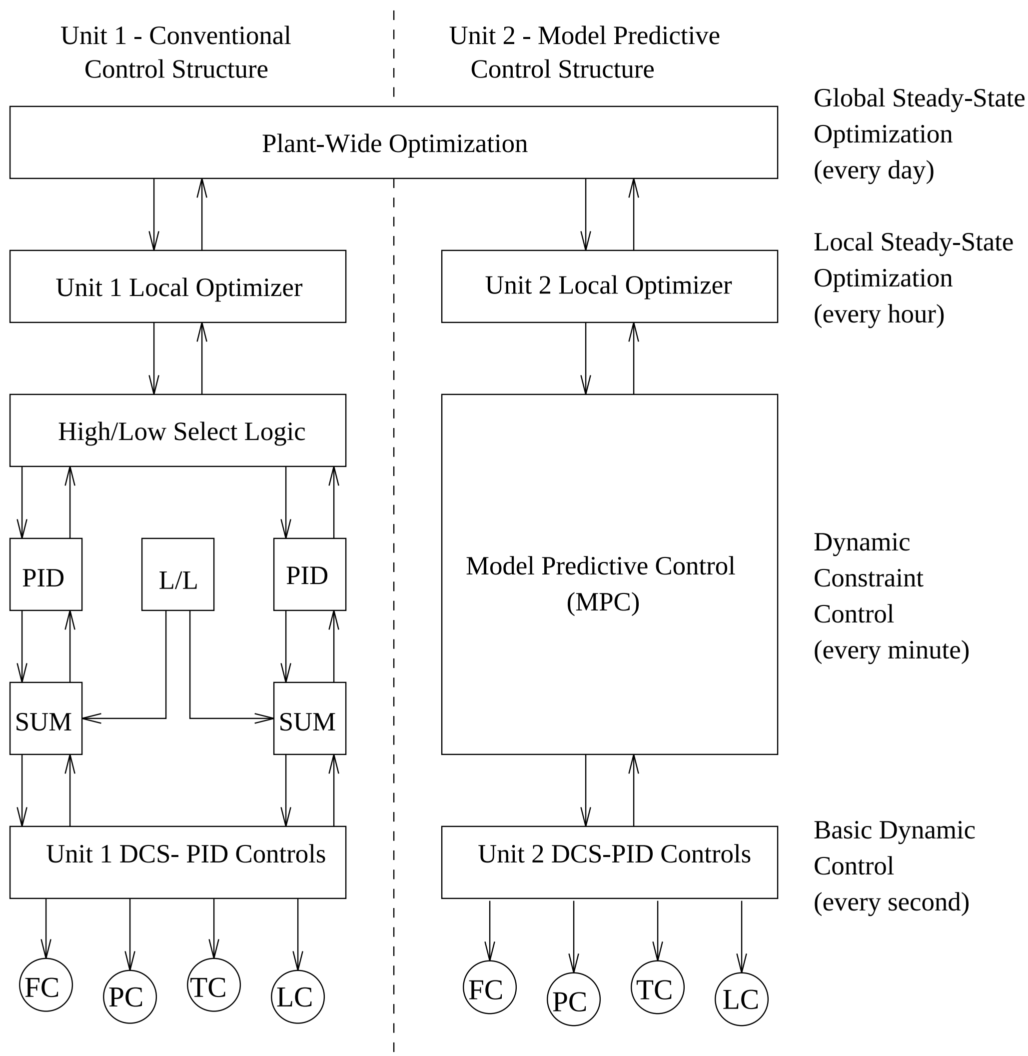

Figure 2. Hierarchy of control system functions in a typical processing plant. Conventional structure is shown at the left; MPC structure is shown at the right.

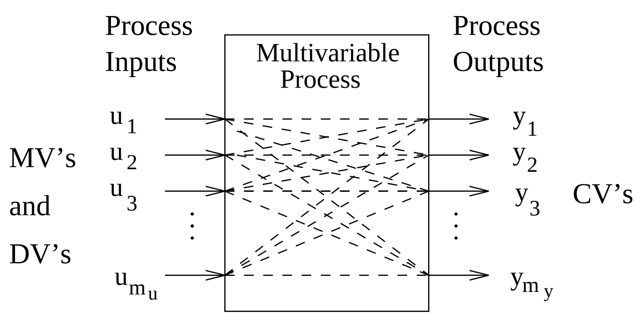

Figure 3. *Input-output process description used in industrial MPC technology. Process inputs u_i consist of two types; manipulated variables (MV's) and disturbance variables (DV's). Process outputs are the controlled variables (CV's).*

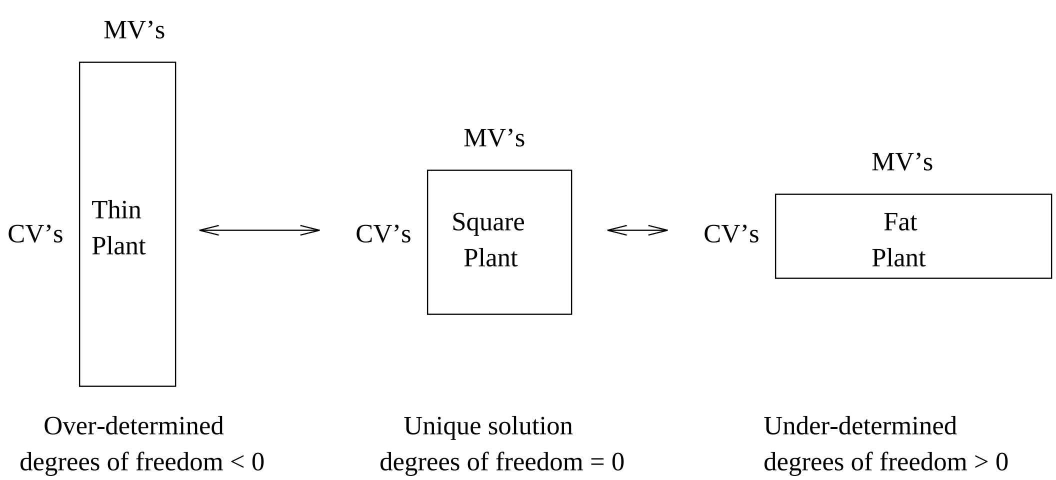

Figure 4. *Process structure determines the degrees of freedom available to the controller. Adapted from Froisy (Froisy, 1994).*

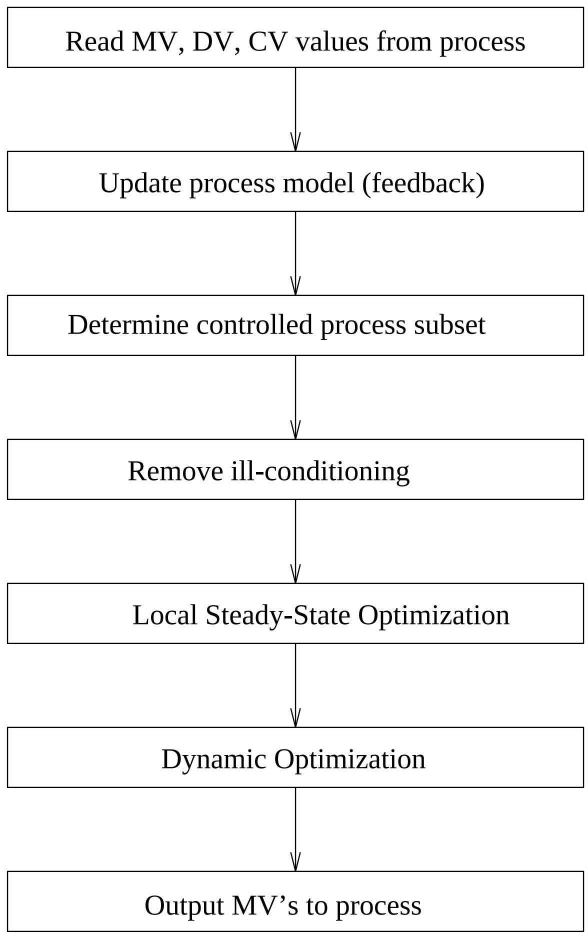

Figure 5. Flow of MPC calculation at each control execution.

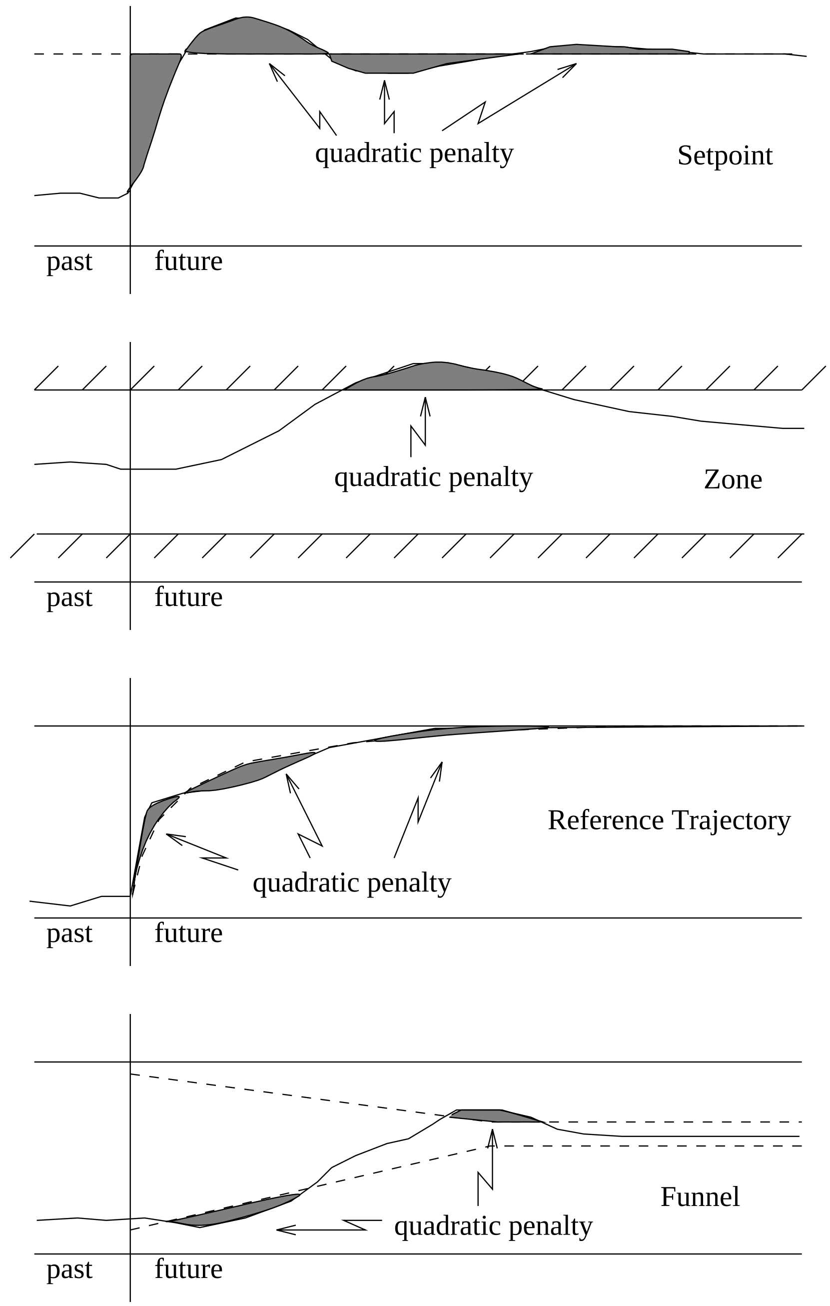

Figure 6. *Four options for specifying future CV behavior; setpoint, zone, reference trajectory and funnel. Shaded areas show violations penalized in the dynamic optimization.*

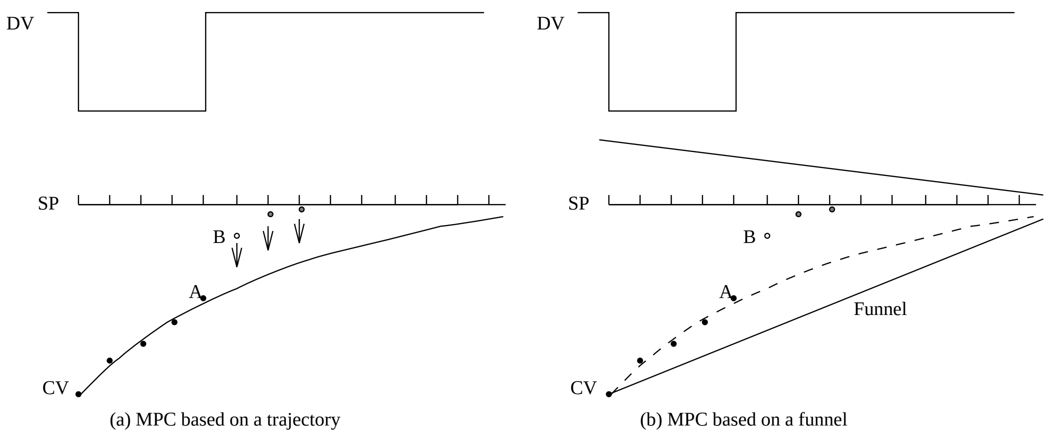

Figure 7. *MPC based on a funnel allows a CV to move back to the setpoint faster than a trajectory would require if a pulse disturbance releases. A trajectory based MPC would try to move away from the setpoint to follow the trajectory.*

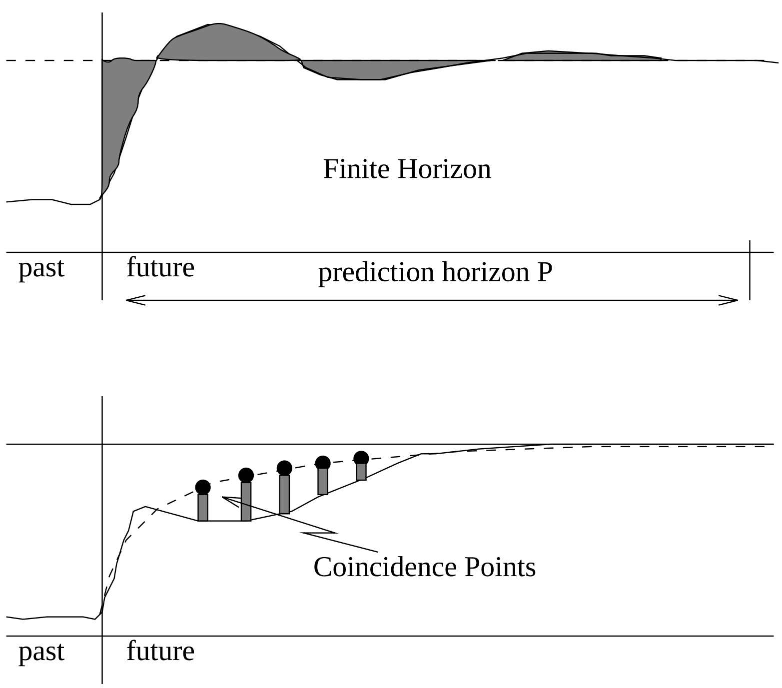

Figure 8. *Output horizon options. Finite horizon (top) includes P future points. A subset of the prediction horizon, called the coincidence points (bottom) may also be used. Shaded areas show violations penalized in the dynamic optimization.*

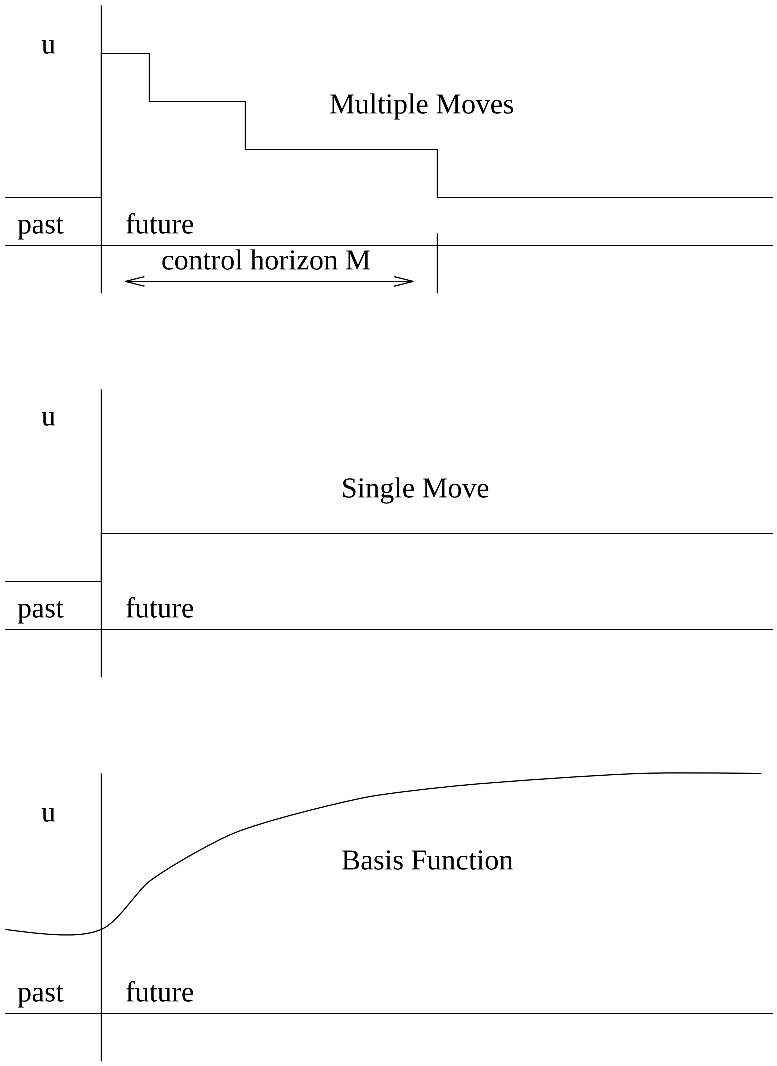

Figure 9. *Input parameterization options. Multiple move option (top), single move option (middle), basis function parameterization (bottom).*

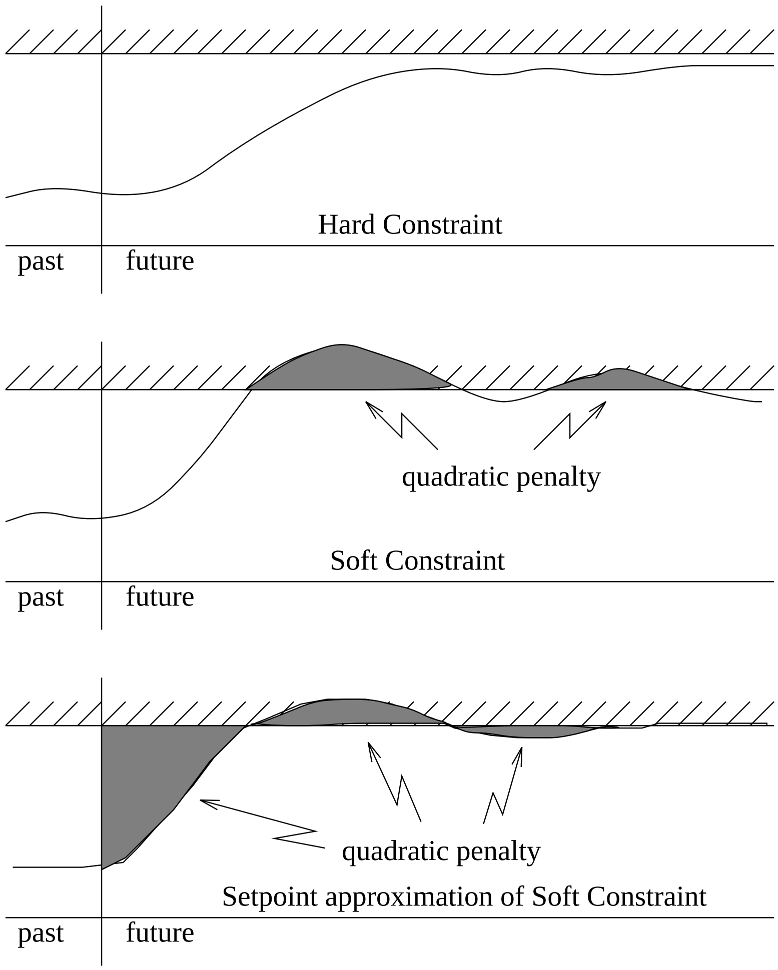

Figure 10. *The three basic types of constraint; hard, soft and setpoint approximation. Hard constraints (top) should not be violated in the future. Soft constraints (middle) may be violated in the future, but the violation is penalized in the objective function. Setpoint approximation of constraint (bottom) penalizes deviations above and below the constraint. Shades areas show violations penalized in the dynamic optimization. Adapted from Froisy (Froisy, 1994).*