Model Predictive Controllers: A Critical Synthesis of Theory and Industrial Needs

Michael Nikolaou[1]
Chemical Engineering Dept.
University of Houston
Houston, TX 77204-4792

Abstract – After several years of efforts, constrained model predictive control (MPC), the de facto standard algorithm for advanced control in process industries, has finally succumbed to rigorous analysis. Yet successful practical implementations of MPC were already in place almost two decades before a rigorous stability proof for constrained MPC was published. What is then the importance of recent theoretical results for practical MPC applications? In this publication we present a pedagogical overview of some of the most important recent developments in MPC theory, and discuss their implications for the future of MPC theory and practice.

[1] (713) 743 4309, fax: (713) 743 4323, email: nikolaou@uh.edu.

TABLE OF CONTENTS

"*There is nothing more practical than a good theory.*"
Boltzmann

1 Introduction

The last couple of decades have witnessed a steady growth in the use of computers for advanced control of process plants. Rapid improvements in computer hardware, combined with stiff foreign and domestic competition and government regulations have been largely responsible for this development. With over 2000 industrial installations, model predictive control (MPC) is currently the most widely implemented advanced process control technology for process plants (Qin and Badgwell, 1996). As is frequently the case, the idea of MPC appears to have been proposed long before MPC came to the forefront (Propoi, 1963; Rafal and Stevens, 1968; Nour-Eldin, 1971). Not unlike many technical inventions, MPC was first implemented in industry – under various guises and names – long before a thorough understanding of its theoretical properties was available. Academic interest in MPC started growing in the mid eighties, particularly after two workshops organized by Shell (Prett and Morari, 1987; Prett et al., 1990). The understanding of MPC properties generated by pivotal academic investigations (Morari and Garcia, 1982; Rawlings and Muske, 1993) has now built a strong conceptual and practical framework for both practitioners and theoreticians. While several issues in that framework are still open, there is now a strong foundation. The purpose of this paper is to examine some of the recent developments in the theory of MPC, discuss their theoretical and practical implications, and propose directions for future development and research on MPC. We hope that both practitioners and theoreticians will find the discussion useful. We would like to stress that this work does not purport to be an exhaustive discussion on MPC to any extent beyond what the title of the work implies. In particular, important practical issues such as the efficiency and effectiveness of various numerical algorithms used to solve the on-line optimization problems, human factors, fault tolerance, detection and diagnosis, and programming environments for MPC implementation are hardly touched upon in any way other than what pertains to their implications for theoretically expected MPC properties.

2 What is MPC?

While the MPC paradigm encompasses several different variants, each one with its own special features, all MPC systems rely on the idea of generating values for process inputs as solutions of an *on-line (real-time) optimization* problem. That problem is constructed on the basis of a process model and process measurements. Process measurements provide the feedback (and, optionally, feedforward) element in the MPC structure. Figure 1 shows the structure of a typical MPC system. It makes it clear that a number of possibilities exist for

- Input-output model,
- disturbance prediction,
- objective,
- measurement,
- constraints, and
- sampling period (how frequently the on-line optimization problem is solved).

Regardless of the particular choice made for the above elements, on-line optimization is the common thread tying them together. Indeed, the possibilities for on-line optimization (Marlin and Hrymak, 1997) are numerous, as discussed in section 6.2.2.1.

Figure 1 also makes it clear that the behavior of an MPC system can be quite complicated, because the control action is determined as the result of the on-line optimization problem. While engineering intuition may frequently be used in the analysis of the behavior or in the design of MPC systems, theory can provide valuable help. Theory can augment human judgement and intuition in the development and implementation of better MPC systems that can realize their full potential as "advanced" control systems. Some of the benefits of improved MPC systems are better control performance, less down time, reduced maintenance requirements, and improved flexibility and agility.

The origins of MPC applications in industry are quite interesting. While the author is more familiar with US industry, developments overseas seem to have followed a similar path. The first use of computers to calculate an *on-line* economic optimal operating point for a process unit appears to have taken place in the late nineteen fifties. Åström and Wittenmark (1984, p. 3) cite March 12, 1959 as the first day when a computer control system went on-line at a Texaco refinery in Port Arthur, Texas. The computer control system, designed by Ramo-Wooldridge (later TRW), relied on an RW-300 computer. Baxley and Bradshaw (1998) mention that around the same time (1959)

Union Carbide, in collaboration with Ramo-Wooldridge, implemented an on-line computer control and optimization system, based also on the RW300, at the Seadrift, Texas plant's ethylene oxide unit. The implementation was not a classical mathematical programming type optimization. It was an implied "maximize production" optimization with a feed allocation algorithm for multiple parallel reactors followed by a serial train of reactors to convert all the remaining ethylene before exhausting to the air. However, there was no open publication reporting this venture. Baxley and Bradshaw (1998) believe that the first open report related to a similar computer control venture was by Monsanto. It appears that computer control and on-line optimization were ideas whose time had come. It also appears that on-line optimization was performed every few hours at the supervisory level, using steady-state models. Certainly, the speed and storage capacity of computers available at the time must have played a role. As the capability of computers increased, so did the size and sophistication of on-line optimization. Early projects usually included ethylene units and major oil refinery processes such as crude distillation units and fluid catalytic cracking (FCC) units (Darby and White, 1988). "The objective function was generally an economic one but we had the flexibility to select alternative ones if the operating and/or business environment suggested another, e.g., maximize ethylene production, minimize ethylene costs, etc. We were getting the tools to be more sophisticated and we took advantage of them where it made economic sense." (Baxley and Bradshaw, 1998).

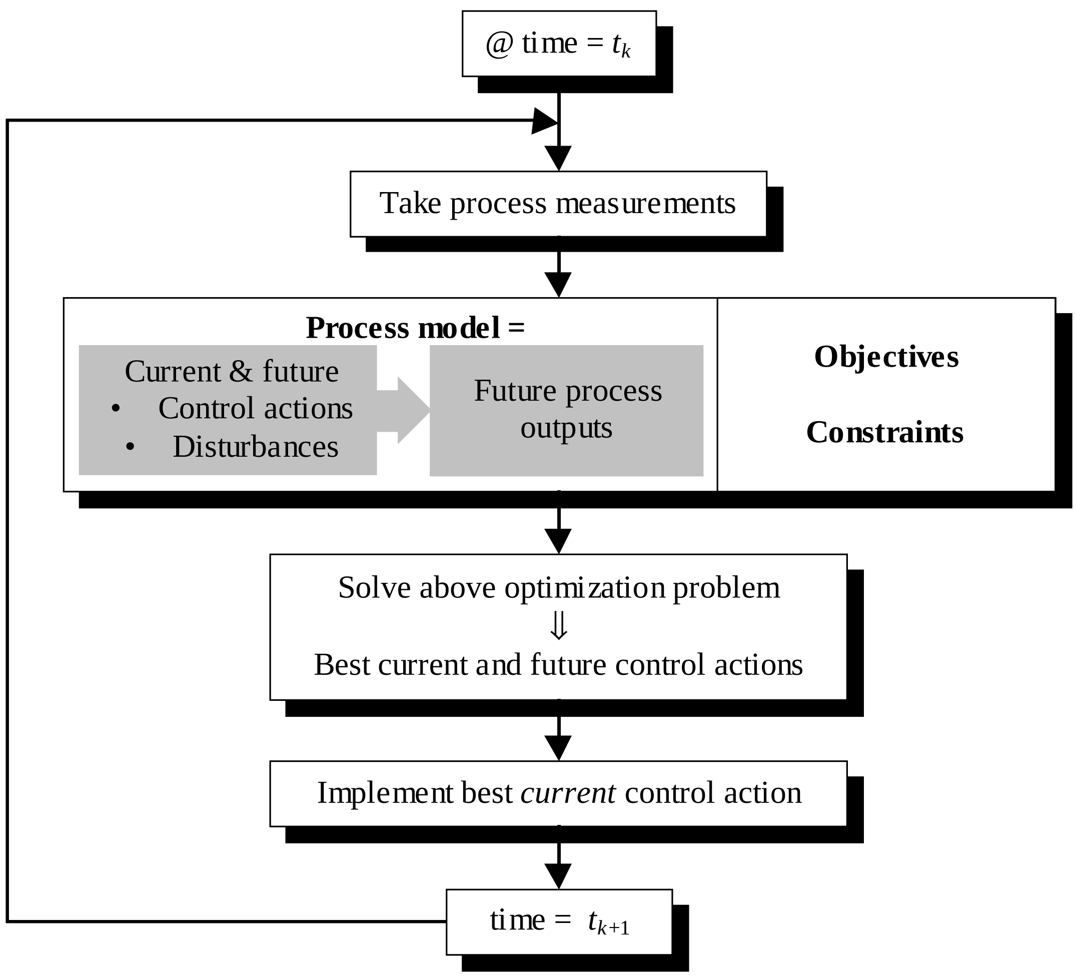


Figure 1. Model Predictive Control Scheme

In the early seventies, practitioners of process control in the chemical industry capitalized on the increasing speed and storage capacity of computers, by expanding on-line optimization to process regulation through more frequent optimization. This necessitated the use of dynamic models in the formulation of on-line optimization problems that would be solved every few minutes. What we today call MPC was conceived as a control algorithm that met a key requirement, not explicitly handled by other control algorithms: The handling of inequality constraints. Efficient use of energy, of paramount importance after the 1973 energy crisis, must have been a major factor that forced oil refineries and petrochemical plants to operate close to constraints, thus making constrained control a necessity.

At the time, the connection of MPC to classical control theory was, at best, fuzzy, as manifested by the title of perhaps the first journal publication reporting the successful application of the MPC algorithm in the chemical process industry: "Model Predictive ***Heuristic*** Control: Applications to Industrial Processes" (Richalet et al., 1978). As is often the case, ingenuity and engineering insight arrived at the same result that others had reached after taking a different route, whether theoretical or heuristic. Where the MPC idea first appeared as a concept is difficult to trace. Prett and García (1988) cite Propoi (1963) as the first who published essentially an MPC algorithm. Rafal and Stevens (1968) presented essentially an MPC algorithm with quadratic cost, linear constraints, and moving horizon of length one. They controlled a distillation column for which they used a first-principles nonlinear model that they linearized at each time step. In many ways, that publication contained several of the elements that today's MPC systems include. It is evident that the authors were fully aware of the limitations of the horizon of length one, but were limited by the computational power available at the time:

> *…the step-by-step optimal control need not be overall optimal. … In the present work, the one-step approach is taken because it is amenable to practical solution of the problem and is well suited to nonlinear situations where updating linearization is useful.* (Rafal and Stevens, 1968, p. 85).

Mayne et al. (1998) provide a quote from Lee and Markus (1967, p. 423) which essentially describes the MPC algorithm. Nour-Eldin (1971, p. 41), among others, explicitly describes the on-line constrained optimization idea, starting from the principle of optimality in dynamic programming:

> [2]*Zusammenfassend: Beim Zeitpunkt* t_{k-1} *wird das Optimum von* Z_k *gesucht. Der resultierende Steuerungsvektor* $\underline{U}^*(k)$ *hängt von* $\underline{x}(k-1)$ *ab und enthält sämtliche Steuervektoren* $\underline{u}_k^*$, $\underline{u}_{k+1}^*, \ldots,$ $\underline{u}_N^*$ *welche den Prozess während dem Intervall* $[t_{k-1}, T]$ *optimal steuern. Von diesem Steuervektoren verwendet man den Vektor* $\underline{u}_k^*$ *(welcher von* $\underline{x}(k-1)$ *abhängt) als Steuervektor für das nächste Intervall* $[t_{k-1}, t_k]$. *Beim nächsten Zeitpunkt* t_k *wird ein neuer Steuervektor* $\underline{u}_{k+1}^*$ *bestimmt. Dieser wird aus der Zielfunktion* Z_{k+1} *berechnet und ist von* $\underline{x}(k)$ *abhängig. Damit wird der Vektor* $\underline{u}_k$, *welcher im Intervall* τ_k *verwendet wird, vom Zustandsvektor* $\underline{x}(k-1)$ *abhängig. Das gesuchte Rückführungsgesetz besteht somit aus der Lösung einer convexen Optimierungsaufgabe bei jedem Zeitpunkt* t_{k-1} $(k = 1, 2, \cdots, N)$ (Underlining in the original text.)

While the value of on-line constrained optimization is explicitly identified in the above passage, the concept of moving horizon is missing. That is, perhaps, due to the fact that the author was concerned with the design of autopilots for airplane landing, a task that has a finite duration *T*. The algorithm described above is, essentially, the mathematical equivalent of MPC for a batch chemical process.

In the sixties and seventies, in contrast to literature references to the constrained on-line optimization performed by MPC, which were only sporadic, there was an already vast and growing literature on a related problem, the linear-quadratic regulator (LQR) either in deterministic or stochastic settings. Simply stated, the LQR problem is

---

[2] Summarizing: At the time point t_{k-1} the optimum of the [quadratic objective function] Z_k is sought. The resulting control [input] vector $\underline{U}^*(k)$ depends on $\underline{x}(k-1)$ and contains all control [input] vectors $\underline{u}_k^*$, $\underline{u}_{k+1}^*, \ldots,$ $\underline{u}_N^*$ which control the process optimally over the interval $[t_{k-1}, T]$. Of these control [input] vectors, one implements the vector $\underline{u}_k^*$ (which depends on $\underline{x}(k-1)$) as input vector for the next interval $[t_{k-1}, t_k]$. At the next time point t_k a new input vector $\underline{u}_{k+1}^*$ is determined. This is calculated from the objective function Z_{k+1} and is dependent on $\underline{x}(k)$. Therefore, the vector $\underline{u}_k$, which is implemented in the interval τ_k, is dependent on the state vector $\underline{x}(k-1)$. Hence, the sought feedback law consists of the solution of a convex optimization problem at each time point t_{k-1} $(k = 1, 2, \cdots, N)$. (Translation by the author.)

(1) $$\min_{\mathbf{x},\mathbf{u}} \sum_{i=0}^{p} \mathbf{x}[i]^T \mathbf{Q}\mathbf{x}[i] + \mathbf{u}[i]^T \mathbf{R}\mathbf{u}[i]$$

where

(2) $$\mathbf{x}[i+1] = \mathbf{A}\mathbf{x}[i] + \mathbf{B}\mathbf{u}[i], \quad \mathbf{x}[0] = \mathbf{x}_0$$

and the optimization horizon length p could be finite or infinite. A celebrated result of LQR theory was

(3) $$\mathbf{u}_{opt}[i] = \mathbf{K}[i]\mathbf{x}_{opt}[i], \quad i = 1, \cdots, p$$

known as the "feedback form" of the optimal solution to the optimization problem posed by eqns. (1) and (2). The state feedback gain $\mathbf{K}[i]$ is not fixed, and is computed from the corresponding Riccati equation. Yet, for finite p, this viewpoint of feedback referred to a set of shrinking horizons ending at the same time point, thus corresponding to a control task that would end at time p. Of course, p could be equal to infinity (in which case $\mathbf{K}$ would be fixed) but then that formulation would not lend itself to optimization subject to inequality constraints (for which no explicit solution similar to eqn. (3) could be obtained) because it would involve an infinite number of decision variables. The ideas of a finite moving horizon and on-line optimization somehow did not occupy much of the research community, although it appears that they were known. In fact, a lot of effort was expended to *avoid* on-line optimization, given the limited capabilities of computers of that era and the fast sampling of systems for which LQR was developed (e.g. aerospace).

Over the years, the heuristics of the early works on MPC were complemented by rigorous analysis that elucidated the essential features, properties, advantages, and limitations of MPC. In the next section, we will start with a smooth introduction to a simple (if not limiting) MPC formulation, that was popular in the early days of MPC. We will then identify some of its many variants.

2.1 A traditional MPC formulation

Consider a stable single-input-single-output (SISO) process with input u and output y. A formulation of the MPC on-line optimization problem can be as follows: At time k find

(4) $$\min_{u[k|k],\cdots,u[k+p-1|k]} \sum_{i=1}^{p} w_i \left(y[k+i|k] - y^{sp}\right)^2 + \sum_{i=1}^{m} r_i \Delta u[k+i-1|k]^2$$

subject to

(5) $$u_{\max} \geq u[k+i-1|k] \geq u_{\min},\ i = 1, \cdots, m$$

(6) $$\Delta u_{\max} \geq \Delta u[k+i-1|k] \geq -\Delta u_{\max},\ i = 1, \cdots, m$$

(7) $$y_{\max} \geq y[k+i|k] \geq y_{\min},\ i = 1, \cdots, p$$

where p and $m < p$ are the lengths of the process output prediction and manipulated process input horizons, respectively; $u[k+i-1|k]$, $i = 1, \cdots, p$, is the set of future process input values with respect to which the optimization will be performed, where

(8) $$u[k+i|k] = u[k+m-1|k], \quad i = m, \cdots, p-1;$$

y^{SP} is the set-point; and Δ is the backward difference operator, i.e.

(9) $$\Delta u[k+i-1|k] \mathrel{\hat{=}} u[k+i-1|k] - u[k+i-2|k]$$

In typical MPC fashion (Prett and Garcia, 1988), the above optimization problem is solved at time k, and the optimal input $u[k] = u_{opt}[k|k]$ is applied to the process. This procedure is repeated at subsequent times k+1, k+2, etc.

It is clear that the above problem formulation necessitates the prediction of future outputs $y[k+i|k]$. This, in turn, makes necessary the use of a *model* for the *process* and external *disturbances*. To start the discussion on process models, assume that the following finite-impulse-response (FIR) model describes the dynamics of the controlled process:

(10) $$y[k] = \sum_{j=1}^{n} h_i u[k-j] + d[k]$$

where h_i are the model coefficients (convolution kernel) and d is a disturbance. Then

(11) $$y[k+i|k] = \sum_{j=1}^{n} h_j u[k+i-j|k] + d[k+i|k]$$

where

(12) $$u[k+i-j|k] = u[k+i-j], \quad i-j<0$$

The prediction of the future disturbance $d[k+i|k]$ clearly can be neither certain nor exact. An approximation or simplification has to be employed, such as

(13) $$d[k+i|k] = d[k|k] = y[k] - \sum_{j=1}^{n} h_j u[k-j]$$

where $y[k]$ is the measured value of the process output y at sampling point k and $u[k-j]$ are past values of the process input u. Substitution of eqns. (11) to (13) into eqns. (4) to (7) yields

(14) $$\min_{u[k|k],\cdots,u[k+p-1|k]} \sum_{i=1}^{p} w_i \left(\sum_{j=1}^{n} h_j u[k+i-j|k] - \sum_{j=1}^{n} h_j u[k-j] + y[k] - y^{sp} \right)^2 + \sum_{i=1}^{m} r_i \Delta u[k+i-1|k]^2$$

subject to

(15) $$u_{\max} \geq u[k+i-1|k] \geq u_{\min}\,, \qquad i = 1,2,\cdots,m$$

(16) $$\Delta u_{\max} \geq \Delta u[k+i-1|k] \geq -\Delta u_{\max}\,, \qquad i = 1,2,\cdots,m$$

(17) $$y_{\max} \geq \sum_{j=1}^{n} h_j u[k+i-j|k] - \sum_{j=1}^{n} h_j u[k-j] + y[k] \geq y_{\min}, \quad i = 1,\cdots,p$$

The above optimization problem is a quadratic programming problem, which can be easily solved at each time k.

2.2 Expanding the traditional MPC formulation

The above formulation of MPC was typical in the first industrial implementations that dealt with stable processes modeled by finite-impulse-response (FIR) models. FIR models, although not essential for characterizing a model-based algorithm as MPC, have certain advantages from a practical implementation viewpoint: Time delays and complex dynamics can be represented with equal ease. Mistakes in the characterization of colored additive noise as white in open-loop experiments introduce no bias in parameter estimates. No advanced knowledge of modeling and identification techniques is necessary if simple step-response experiments are used for process identification. Instead of the observer or state estimator of classic optimal control theory, a model of the process is employed directly in the algorithm to predict future process outputs (Morari, 1988). Their main disadvantage is the use of too-many parameters (overparametrization), which becomes even more pronounced in the multivariable case.

While a large class of processes can be treated by that formulation, more general classes can be handled by more general MPC formulations concentrating on the following characteristics:

- *Unstable process model*. In that case, an FIR process model cannot be used. A state-space model such as

(18) $$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{B}u[k] + \mathbf{E}d[k] \\ y[k] &= \mathbf{C}\mathbf{x}[k] + \mathbf{D}u[k] + \mathbf{F}e[k] \end{aligned}$$

or a deterministic auto-regressive-moving-average with exogenous input (DARMAX) model

(19) $$y[k] = \sum_{i=1}^{n_u} h_i u[k-i] + \sum_{i=1}^{n_y} g_i y[k-i] + \sum_{i=1}^{n_d} f_i d[k-i]$$

can be used. As MPC systems grow in size, the probability of including an unstable or marginally stable (integrating) unit increases. For such units, models as in eqns. (18) or (19) are necessary.

- *Nonlinear process model*. The nonlinearity of chemical processes is well documented (Shinskey, pp. 55-56, 1967; Foss, 1973; Buckley, 1981; García and Prett, 1986; Morari, 1986; IEEE Report, 1987; NRC Committee Report, p. 148, 1988; Fleming, 1988; Prett and García, p. 18, 1988; Edgar, 1989; Longwell, 1991; Bequette, 1991; Kane, 1993; Allgöwer and Doyle, 1997; Ogunnaike and Wright, 1997). Typical examples are distillation columns and reactors. Because nonlinear models are defined by what they are not (namely linear) there exist a number of possibilities for representing nonlinear systems. First-principles, empirical, or hybrid models in state-space or input-output frameworks are all possible. In addition, the development and adaptation of such models is a central issue in MPC.

- *Stochastic disturbance model*. There are various possibilities for using stochastic disturbance models other than the zero-order model shown in eqn. (13) (Ljung, 1987).

- *Stochastic objective function*: The above MPC formulation assumes that future process outputs are deterministic over the finite optimization horizon. For a more realistic representation of future process outputs, one may consider a probabilistic (stochastic) prediction for $\mathbf{y}[k+i|k]$ and formulate an objective function that contains the expectation of appropriate functionals. For example, if $\mathbf{y}[k+i|k]$ is probabilistic, then the expectation of the functional in eqn. (4) could be used. This formulation, known as open-loop optimal feedback, does not take into account the fact that additional information would be available at future time points $k+i$, and assumes that the system will essentially run in open-loop fashion over the optimization horizon. An alternative, producing a closed-loop optimal feedback law, relies on the dynamic programming formulation of an objective function such as the following:

(20) $$\min_{u[k|k]}\left(w_1\left(y[k+1|k]-y^{sp}\right)^2+\Delta u[k|k]^2+\min_{u[k+1|k+1]}\left(w_2\left(y[k+2|k+1]-y^{sp}\right)^2+\Delta u[k+1|k+1]^2+\cdots\right)\right)$$

While the open-loop optimal feedback law does not result in unwieldy computational requirements, the closed-loop optimal feedback law is considerably more complicated. For several practical problems the open-loop optimal feedback law produces results that are close to those produced by the closed-loop optimal feedback law. However, there are cases for which the open-loop optimal feedback law may be far inferior to the closed-loop optimal feedback law. Rawlings et al. (1994) present a related example on a generic staged system. Lee and Yu (1997) show that open-loop optimal feedback is, in general, inferior to closed-loop optimal feedback for nonlinear processes and linear processes with uncertain coefficients. They also develop a number of explicit closed-loop optimal feedback laws for a number of unconstrained MPC cases.

- *Available measurements*: For controlled variables that are not directly measurable, measurements have to be inferred by measurements of secondary variables and/or laboratory analysis of samples. Good inference relies on reliable models. In addition, the results of laboratory analysis, usually produced much less frequently than inferential estimates, have to be fused with the inferential estimates produced by secondary measurements.

For MPC systems that use state-space models, usually not all states are measurable, thus making state estimators necessary (Lee et al., 1994).

- *Constraints*: While constraints placing bounds on process inputs are trivial to formulate, constraints on process outputs are more elusive, because future process outputs $\mathbf{y}[k+i|k]$ are predicted in terms of a model. If the probability density function of $\mathbf{y}[k+i|k]$ is known, then deterministic constraints on $\mathbf{y}[k+i|k]$ can be replaced by probabilistic constraints of the form

(21) $$\Pr\{\mathbf{y}[k+i|k]\le y_{\max}\}\ge\alpha$$

(Schwarm and Nikolaou, 1997).

- *Sampling period*: The selection of the time points t_k at which on-line optimization is performed (Figure 1) is an important task, albeit not as widely studied as other MPC design tasks. Things become more interesting when measurements or decisions take place at different time intervals for different variables (Lee et al., 1992). The multitude of different time-scales is usually handled through decomposition of the overall on-line optimization problem into independently solved subproblems, each at a different time-scale.

2.3 MPC without inequality constraints

When there are no inequality constraints (eqns. (15) to (17)), the minimization of the quadratic objective function in eqn. (4) has a simple closed-form solution, which can be expressed as follows. Eqns. (11) and (13) yield

(22) $$\mathbf{y}_{k+1|k}^{k+p|k}=\mathbf{H}\mathbf{u}_{k|k}^{k+p-1|k}+\mathbf{G}\mathbf{u}_{k-n}^{k-1}+\mathbf{y}[k]$$

where, assuming that $p>n$,

(23) $$\mathbf{y}_{k+1|k}^{k+p|k}\triangleq\left[y[k+1|k]\quad\cdots\quad y[k+p|k]\right]^T$$

(24) $$\mathbf{u}_{k|k}^{k+p-1|k}\triangleq\left[u[k\mid k]\quad\cdots\quad u[k+p-1\mid k]\right]^T$$

(25) $$\mathbf{u}_{k-n}^{k-1}\triangleq\left[u[k-n]\quad\cdots\quad u[k-1]\right]^T$$

(26) $$\mathbf{y}[k]\triangleq\underbrace{\left[y[k]\quad\cdots\quad y[k]\right]^T}_{p}$$

$$\mathbf{H} \hat{=} \underbrace{\begin{bmatrix} h_1 & 0 & \cdots & \cdots & 0 \\ \vdots & h_1 & 0 & & \vdots \\ h_n & & \ddots & \ddots & \vdots \\ 0 & \ddots & & \ddots & 0 \\ \vdots & 0 & h_n & \cdots & h_1 \end{bmatrix}}_{p} \Bigg\} p \tag{27}$$

$$\mathbf{G} \hat{=} \underbrace{\begin{bmatrix} 0 & h_n & \cdots & h_2 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & h_n \\ \vdots & & & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}}_{n} \Bigg\} p - \underbrace{\begin{bmatrix} h_n & \cdots & h_1 \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ h_n & \cdots & h_1 \end{bmatrix}}_{n} \Bigg\} p \tag{28}$$

and eqn. (4) becomes

$$\begin{aligned} &\min_{\mathbf{u}_{k|k}^{k+p-1|k}} \left(\mathbf{y}_{k+1|k}^{k+p|k} - \mathbf{y}^{SP}\right)^T \mathbf{W}\left(\mathbf{y}_{k+1|k}^{k+p|k} - \mathbf{y}^{SP}\right) + \Delta\mathbf{u}_{k|k}^{k+m-1|k\,T} \mathbf{R}\Delta\mathbf{u}_{k|k}^{k+m-1|k} = \\ &= \min_{\mathbf{u}_{k|k}^{k+m-1|k}} \left(\mathbf{HJu}_{k|k}^{k+m-1|k} + \mathbf{Gu}_{k-n}^{k-1} + \mathbf{y}[k] - \mathbf{y}^{SP}\right)^T \mathbf{W}\left(\mathbf{HJu}_{k|k}^{k+m-1|k} + \mathbf{Gu}_{k-n}^{k-1} + \mathbf{y}[k] - \mathbf{y}^{SP}\right) + \\ &\quad + \left((\mathbf{I}-\mathbf{P})\mathbf{u}_{k|k}^{k+m-1|k} - \mathbf{Qu}_{k-n}^{k-1}\right)^T \mathbf{R}\left((\mathbf{I}-\mathbf{P})\mathbf{u}_{k|k}^{k+m-1|k} - \mathbf{Qu}_{k-n}^{k-1}\right) \end{aligned} \tag{29}$$

where

$$\mathbf{y}^{SP} = \underbrace{\left[y^{SP} \quad \cdots \quad y^{SP}\right]}_{p}{}^T \tag{30}$$

$$\mathbf{u}_{k|k}^{k+m-1|k} \hat{=} \left[u[k\,|\,k] \quad \cdots \quad u[k+m-1\,|\,k]\right]^T \tag{31}$$

$$\Delta\mathbf{u}_{k|k}^{k+m-1|k} \hat{=} \left[\Delta u[k\,|\,k] \quad \cdots \quad \Delta u[k+m-1\,|\,k]\right]^T \tag{32}$$

$$\mathbf{W} \hat{=} Diag\left(w_1 \quad \cdots \quad w_p\right) \tag{33}$$

$$\mathbf{R} \hat{=} Diag\left(r_1 \quad \cdots \quad r_m\right) \tag{34}$$

$$\mathbf{P} \hat{=} \overbrace{\begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ 1 & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}}^{m} \Bigg\} m \tag{35}$$

$$\mathbf{Q} \hat{=} \overbrace{\begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & 0 & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}}^{n} \Bigg\} m \tag{36}$$

and

$$\textbf{(37)} \qquad \mathbf{J} \triangleq \underbrace{\begin{bmatrix} 1 & 0 & \cdots & & 0 \\ 0 & 1 & & & \\ \vdots & & \ddots & & \\ \vdots & & & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ \hdashline 0 & \cdots & \cdots & 0 & 1 \\ \vdots & & & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}}_{m} \begin{matrix} \left.\vphantom{\begin{matrix}1\\1\\1\\1\\1\end{matrix}}\right\} m \\ \left.\vphantom{\begin{matrix}1\\1\\1\end{matrix}}\right\} p-m \end{matrix}$$

The straightforward solution of the optimization problem in the above equation is

$$\textbf{(38)} \qquad \begin{aligned} \mathbf{u}_{opt}{}_{k|k}^{k+p-1|k} &= \mathbf{J}\mathbf{u}_{opt}{}_{k|k}^{k+m-1|k} \\ &= \mathbf{J}\left[(\mathbf{I}-\mathbf{P})^T\mathbf{R}(\mathbf{I}-\mathbf{P})+\mathbf{J}^T\mathbf{H}^T\mathbf{W}\mathbf{H}\mathbf{J}\right]^{-1}\left[\mathbf{J}^T\mathbf{H}^T\mathbf{W}\mathbf{e}[k]-\left((\mathbf{I}-\mathbf{P})^T\mathbf{R}\mathbf{Q}+\mathbf{J}^T\mathbf{H}^T\mathbf{W}\mathbf{G}\right)\mathbf{u}_{k-n}^{k-1}\right] \end{aligned}$$

where $\mathbf{e}[k] \triangleq \mathbf{y}^{SP}[k]-\mathbf{y}[k]$, and the input $u[k]$ that will eventually be implemented will be

$$u[k] = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}}_{p}\mathbf{u}_{opt}{}_{k|k}^{k+p-1|k}$$

Therefore, the controller is a linear time-invariant controller, and no on-line optimization is needed. Linear control theory, for which there is a vast literature, can equivalently be used in the analysis or design of unconstrained MPC (Morari and Garcia, 1982). A similar result can be obtained for several MPC variants, as long as the objective function in eqn. (4) remains a quadratic function of $\mathbf{u}_{opt}{}_{k|k}^{k+p-1|k}$ and the process model in eqn. (22) remains linear in $\mathbf{u}_{opt}{}_{k|k}^{k+p-1|k}$. Incidentally, notice that the appearance of the measured process output $\mathbf{y}[k]$ in eqn. (22) introduces the measurement information needed for MPC to be a feedback controller. This is in the spirit of classical linear optimal control theory, in which the controlled process state $\mathbf{x}[k]$ contains the feedback information needed by the controller.

Whether one performs the analysis and design using directly a closed form of MPC such as in eqn. (4) or its equivalent on-line optimization form, eqn. (38), is a matter of convenience in translation of engineering requirements into equations. For example, eqn. (38) can be used to determine the poles of the controller, and consequently, the closed-loop behavior (e.g, stability, zero offset, etc.). On the other hand, eqn. (4) can be directly used to help tune the MPC system. For example, it intuitively makes it clear that the "smaller" the matrix **R**, the faster the closed-loop will be, at the risk of approaching instability. Similarly, the process output y will track step changes in the setpoint y^{SP} if the prediction horizon length p is long enough. An overview of MPC within a linear control framework can be found in Mosca (1995). Clarke and coworkers have used the term generalized predictive control (GPC) to describe an essentially unconstrained MPC algorithm (Clarke et al., 1987; Bitmead et al., 1990).

The situation is quite different when inequality constraints are included in the MPC on-line optimization problem. In the sequel, we will refer to "inequality constrained MPC" simply as "constrained MPC". For constrained MPC, no closed-form (explicit) solution can be written. Because different inequality constraints may be active at each time, a constrained MPC controller is not linear, making the entire closed loop nonlinear. To analyze and design constrained MPC systems requires an approach that is not based on linear control theory. We will present the basic ideas in Section 3. We will then present some examples that show the interesting behavior that MPC may demonstrate, and we will subsequently explain how MPC theory can conceptually simplify and practically improve MPC.

3 Stability

3.1 What is stability?

The concept of stability is central in the study of dynamical systems. Loosely speaking, stability is a dynamical system's property related to "good" long-run behavior of that system. While stability by itself may not necessarily guarantee satisfactory performance of a dynamical system, it is not conceivable that a dynamical system may perform well without being stable.

Stability can be quantified in several different ways, each providing insight into particular aspects of a dynamical system's behavior. Mathematical descriptions of the system and its surroundings are necessary for quantitative results. Two broad classes of stability definitions are associated with

(a) stability with respect to initial conditions and
(b) input-output stability,

respectively. The two classes are complementary to each other and can also be combined. For linear systems the two classes are, in general, equivalent. However, they are different (although interrelated) for nonlinear dynamical systems. Next, we make these ideas precise and illustrate their implications through a number of examples. The discussion encompasses both discrete- and continuous-time systems. Full details of pertinent mathematical underpinnings can be found in standard textbooks on nonlinear systems, such as Vidyasagar (1993)

3.1.1 Stability with respect to initial conditions

Consider a system described by the vector difference equation

(39) $$\mathbf{x}[k+1] = \mathbf{f}(k, \mathbf{x}[k]), \quad k \geq 0$$

where $\mathbf{x} : \mathbf{Z}_+ \to \Re^n$ and $\mathbf{f} : \mathbf{Z}_+ \times \Re^n \to \Re^n$. Without loss of generality, the vector **0** is assumed to be an equilibrium point of the system (39). The system is said to be stable around the equilibrium point **0** if the state **x** eventually returns to **0** when its initial value is anywhere within a small neighborhood around **0**. The preceding statement is accurately captured in the following definitions.

Definition 1 – Stability

The equilibrium point **0** at time k_0 of eqn. (39) is said to be *stable* at time k_0 if, for any $\varepsilon > 0$, there exists a $\delta(k_0, \varepsilon) > 0$ such that

(40) $$\|\mathbf{x}[k_0]\| < \delta(k_0, \varepsilon) \Rightarrow \|\mathbf{x}[k]\| < \varepsilon, \forall k \geq k_0$$

Definition 2 – Uniform stability

The equilibrium point **0** at time k_0 of eqn. (39) is said to be *uniformly stable* over $[k_0, \infty)$ if, for any $\varepsilon > 0$, there exists a $\delta(\varepsilon) > 0$ such that

(41) $$\left.\begin{array}{r} \|\mathbf{x}[\ell]\| < \delta(\varepsilon) \\ \ell \geq k_0 \end{array}\right\} \Rightarrow \|\mathbf{x}[k]\| < \varepsilon, \forall k \geq \ell \, .$$

Definition 3 – Asymptotic stability

The equilibrium point **0** at time k_0 of eqn. (39) is said to be *asymptotically stable* at time k_0 if

(a) it is stable at time k_0 and
(b) (b) there exists a $\delta(k_0) > 0$ such that

(42) $$\|\mathbf{x}[k_0]\| < \delta(k_0) \Rightarrow \lim_{k \to \infty} \|\mathbf{x}[k]\| = 0 \, .$$

Definition 4 – Uniform asymptotic stability

The equilibrium point **0** at time k_0 of eqn. (39) is said to be *uniformly asymptotically stable* over $[k_0, \infty)$ if

(a) it is uniformly stable over $[k_0, \infty)$ and
(b) (b) there exists a $\delta > 0$ such that

(43) $$\left.\begin{array}{r} \|\mathbf{x}[\ell]\| < \delta \\ \ell > k_0 \end{array}\right\} \Rightarrow \lim_{k \to \infty} \|\mathbf{x}[k]\| = 0 \, .$$

Definition 5 – Global asymptotic stability

The equilibrium point **0** at time k_0 of eqn. (39) is said to be *globally asymptotically stable* if

(44) $$\lim_{k \to \infty} \|\mathbf{x}[k]\| = 0$$

for any $\mathbf{x}[k_0]$.

Remarks

- Although there are no requirements on the magnitude of δ that appears in the above definitions, in practice δ is desired to be as large as possible, to ensure the largest possible range of initial states that eventually go to 0.

- It is not important what particular norm ||*|| in $\Re^n$ is used in equations (40) and (41), because any two norms $||*||_a$ and $||*||_b$ in $\Re^n$ are equivalent, i.e. there exist positive constants k_1 and k_2 such that $k_1\|\mathbf{x}\|_a \le \|\mathbf{x}\|_b \le k_2\|\mathbf{x}\|_a$ for any $\mathbf{x} \in \Re^n$. We will see that the choice of particular norm is important in input-output stability.
- As the above definitions imply, for a system not to be stable around 0, it is not necessary for the system to produce signals that grow without bounds. The following example illustrates the case.

Example 1 – Unstable system with bounded output

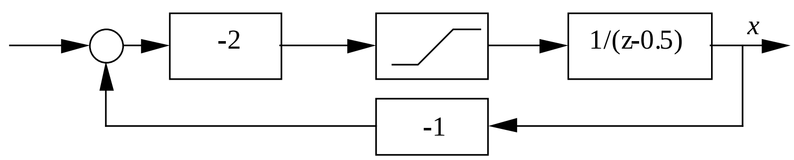
For the feedback system described by the equation

(45) $$x[k+1] = 0.5x[k] + Sat(2x[k]), \quad x[0] = 0.01,$$

where the saturation function is defined as

(46) $$Sat(y) = \begin{cases} 1 & \text{if } y > 1 \\ y & \text{if } -1 \le y \le 1, \\ -1 & \text{if } y < -1 \end{cases}$$

the point 0 is an unstable equilibrium point, because x moves away from 0 for any $x[0] \neq 0$, but it does not ever grow without bound, as Figure 2 shows.



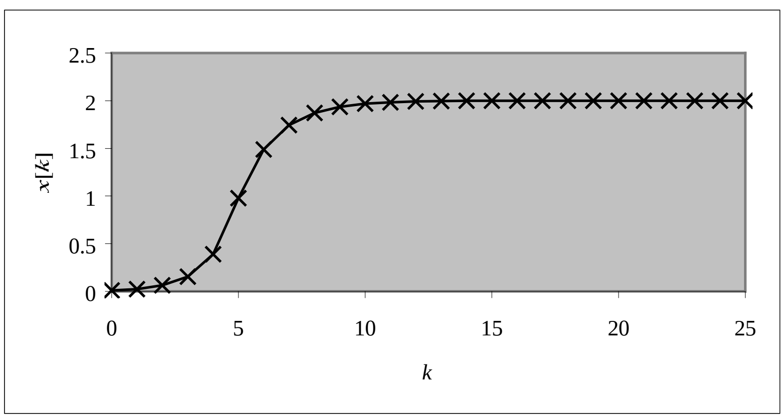


Figure 2. A bounded-output system that is unstable with respect to initial conditions $(x[0] = 0.01)$

Example 2 – Unstable CSTR with bounded output

Consider the reaction $R \to P$ occurring in a non-isothermal jacket-cooled CSTR with three steady states, A, B, C, corresponding to the intersection points of the two lines shown in Figure 3 (Stephanopoulos, 1984, p. 8). The steady state B, corresponding to the temperature T_2 is unstable with respect to initial conditions. Indeed, if the CSTR is initially at temperature $T_2 + \varepsilon$, then it will eventually reach either of the finite temperatures T_1 or T_3, according to whether ε is negative or positive.

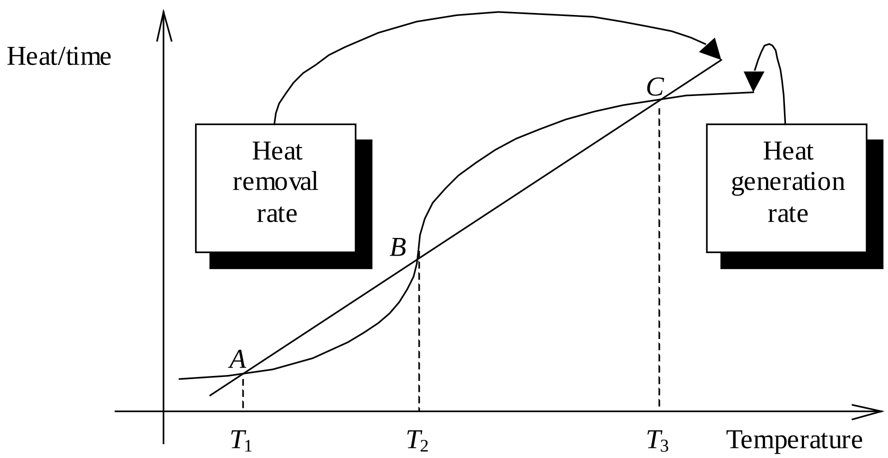


Figure 3. The three steady states of the non-isothermal CSTR in Example 2

3.1.2 Input-output stability

Input-output stability refers to the effect of system inputs to system outputs. Therefore, the emphasis is how the magnitudes of inputs and outputs are related. We make these ideas of magnitude, system, and stability more precise below.

Definition 6 – *p*-norms

The magnitude of a signal $\mathbf{x} : \mathbf{Z}_+ \to \Re^n$ is quantified by its *p*-norm, defined as

(47)
$$\|\mathbf{x}\|_p \triangleq \left(\sum_{k=1}^{\infty} \|\mathbf{x}[k]\|^p \right)^{1/p}$$

where $1 \le p \le \infty$ and $\|\mathbf{x}[k]\|$ can be any Euclidean norm in $\Re^n$.

Based on the above definition, we can provide a first definition of stability.

Definition 7 – Bounded-input-bounded-output (BIBO) stability

A system *S*, mapping an input signal **u** to an output signal **x** with $S(\mathbf{0}) = \mathbf{0}$, is stable if bounded inputs produce bounded outputs, i.e.,

(48)
$$\|\mathbf{u}\|_p < \infty \Rightarrow \|\mathbf{x}\|_q < \infty$$

An alternative statement of eqn. (48) is

(49)
$$\mathbf{u} \in l_p^m \Rightarrow \mathbf{x} \in l_q^m$$

where

(50)
$$l_p^m \triangleq \{\mathbf{z} : \mathrm{Z} \to \Re^m \,\big|\, \|\mathbf{z}\|_p < \infty\}$$

A usual convention is to choose $p = q$ in the above Definition 7, although different values for *p* and *q* may be selected. For example, the option $p = \infty, q = 2$ may be selected so that step inputs (for which $\|\mathbf{u}\|_2 = \infty$ but $\|\mathbf{u}\|_\infty < \infty$) can be included.

While Definition 7 is useful in characterizing *instability* in a meaningful way, it is not always as useful in characterizing *stability* in a meaningful way as well, as the following example shows.

Example 3 – BIBO stability

Consider the system of the following Figure 4.

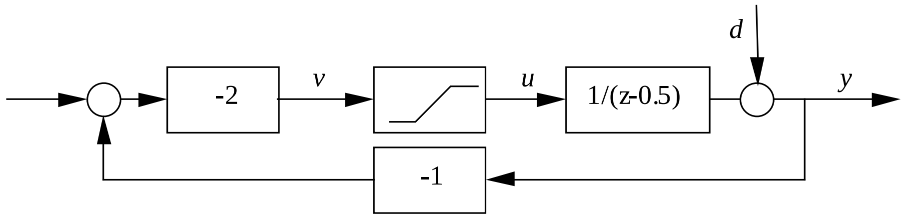


Figure 4. The feedback system of Example 3.

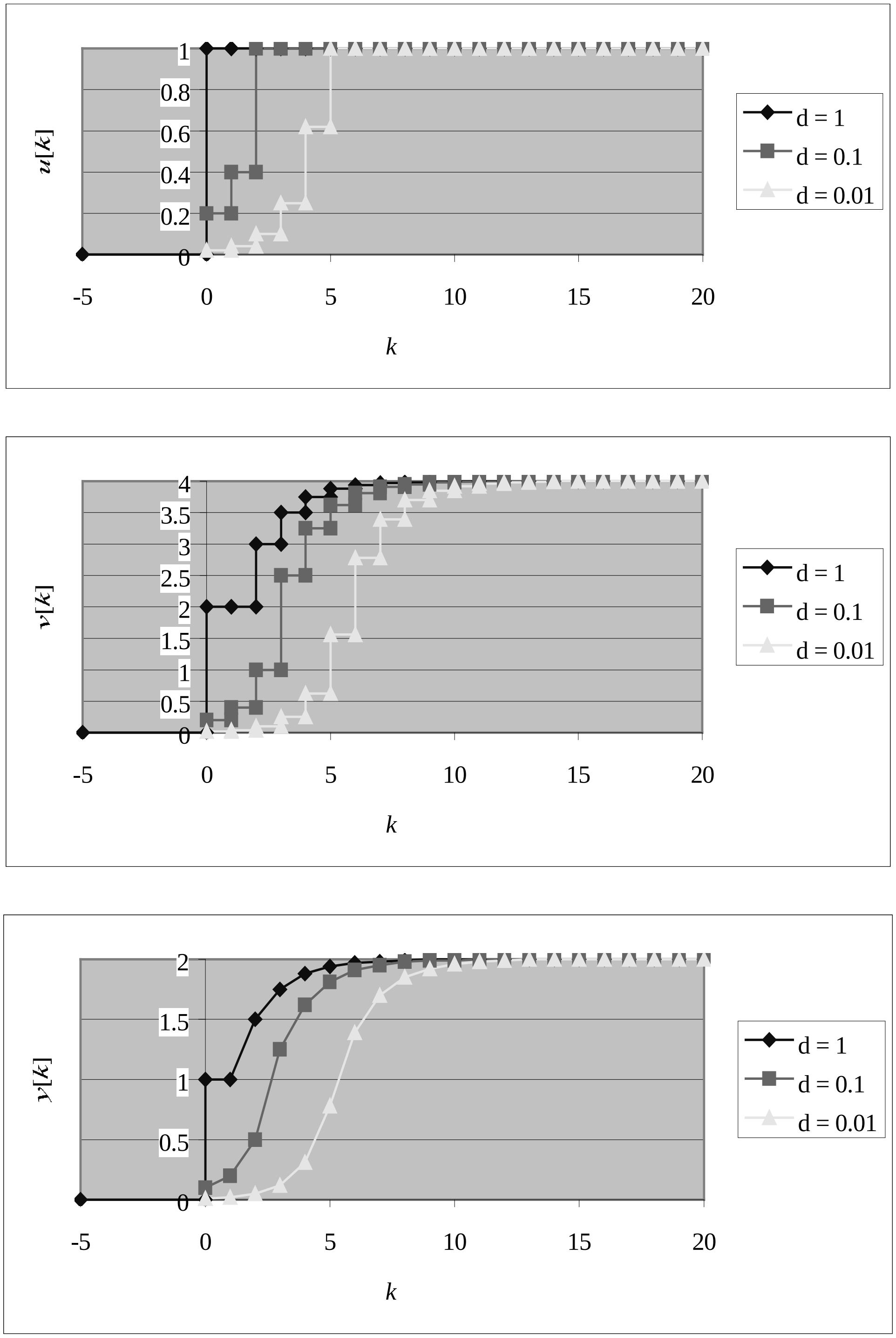


Figure 5. A system that is input-output BIBO stable.

A pulse disturbance *d* of any amplitude drives the output *y* of the system to the final steady-state value 2, while the input *u* reaches a steady state value of 1, and the variable *v* reaches a steady state value of 4. Figure 5 shows the response of the system for pulses *d* of amplitude 1 (diamond), 0.1 (square), and 0.01 (triangle). It is clear that the system output *y* does not return to the desired value of 0, so the system should not be characterized as stable. Yet the BIBO stability criterion with the ∞-norm would characterize this system as stable, because $\|y\|_\infty < \infty$ for any

input d. Of course, a different norm $\|y\|_p$ with $1 \leq p < \infty$ might be used, in which case the system would be characterized as BIBO unstable. The compromise would be that systems generating signals such as $y[k] = \frac{1}{(k+1)^{1/p}}$ would also be BIBO unstable (because $\|y\|_p^p = \sum_{k=0}^{\infty} \left(\frac{1}{(k+1)^{1/p}} \right)^p = \infty$) although $\lim_{k \to \infty} y[k] = 0$.
A better definition of stability would require not only that bounded inputs produce bounded outputs, but also that the amplification of bounded inputs by the system is finite. More precisely, we have the following definition of *finite-gain* stability.

Definition 8 – Finite-gain (FG) stability

A system $S : l_p^m \to l_q^n : \mathbf{u} \mapsto \mathbf{x} \triangleq S\mathbf{u}$, is *finite-gain* stable if the gain (induced norm) of S is finite, i.e.,

(51)
$$\|S\|_{i,pq} \triangleq \sup_{\mathbf{u} \in l_p^m - \{\mathbf{0}\}} \frac{\|\mathbf{x}\|_q}{\|\mathbf{u}\|_p} < \infty$$

The advantage of FG stability over BIBO stability is that systems such as in Example 3 no longer have to be characterized as stable, a conclusion that agrees with intuition. Indeed, for Example 3 we have that pulses of infinitesimally small amplitude drive the output y to 2, consequently

(52)
$$\|S\|_{i,\infty\infty} \triangleq \sup_{d \in l_\infty^1 - \{0\}} \frac{\|y\|_\infty}{\|d\|_\infty} \geq \lim_{\|d\|_\infty \to 0} \frac{2}{\|d\|_\infty} = \infty$$

The shortcoming of Definition 8 is that the input signal $\mathbf{u}$ can vary over the entire space l_p^m. This creates two problems:

(a) The entire space l_p^m may contain physically meaningless signals; and

(b) The stability characteristics of S may be different over different subsets of l_p^m.

The following two examples clarify the above statements.

Example 4 – Selecting physically meaningful inputs to characterize stability

Consider a continuous stirred-tank heater modeled by the following equations, in continuous time:

(53)
$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{1}{V}\left[\left(F_s + u(t)\right)\left(T_i - T(t)\right) + \frac{UA\left(T_c - T(t)\right)}{\rho c_p} \right]$$
$$y(t) = T(t) - T_i - \frac{UA/\rho c_p}{F_s + UA/\rho c_p}(T_c - T_i)$$

where

V = heater volume
F_s = volumetric feed flowrate at steady state
T_i = feed temperature
T = heater temperature
T_c = heating coil temperature
U = heat transfer coefficient
A = heat exchange area
ρ = density of liquid in heater
c_p = specific heat of liquid in heater

The above equation defines an operator $S : u \mapsto y$, where u refers to the feed flowrate and y to temperature, both in deviation form. It can be shown (Nikolaou and Manousiouthakis, 1989) that

(54)
$$\|S\|_{i,\infty\infty} \triangleq \sup_{u \in l_\infty^1 - \{0\}} \frac{\|y\|_\infty}{\|u\|_\infty} = \infty$$

where the supremum is attained for

(55)
$$u(t) = -F_s - \frac{UA}{\rho c_p} \Rightarrow F(t) = -\frac{UA}{\rho c_p}, \quad t \geq 0$$

This suggests that a *negative flowrate* $F(t)$ would result in instability. However, since the flowrate is always *nonnegative*, this instability warning is of little value. In fact, computation of the gain of *S over the set* $W \hat{=} \{u | u_{\min} \le u(t) < \infty\}$, where $-F_s \le u_{\min}$, yields

$$\|S\|_{i,\infty\infty,W} \hat{=} \sup_{u \in W-\{0\}} \frac{\|y\|_\infty}{\|u\|_\infty} = \frac{(T_c - T_i)\frac{UA}{\rho c_p}}{\left(\frac{UA}{\rho c_p} + F_s + u_{\min}\right)^2} < \infty \tag{56}$$

implying that the system is indeed FG stable for all physically meaningful inputs, as one would intuitively expect.

Example 5 – Stability dependence on the set of inputs

Consider a continuous stirred-tank reactor (CSTR) modeled by the following equations, in continuous time:

$$\begin{aligned} \frac{\mathrm{d}c_A}{\mathrm{d}t} &= \frac{F(t)}{V}\left(C_{Ai} - C_A(t)\right) - C_A(t)\alpha e^{-\frac{E}{RT(t)}} \\ \frac{\mathrm{d}T}{\mathrm{d}t} &= \frac{F(t)}{V}\left(T_i - T(t)\right) - \frac{\Delta H_R}{\rho c_p} C_A(t)\alpha e^{-\frac{E}{RT(t)}} - \frac{Q}{V\rho c_p} \\ u(t) &= F(t) - F_s \\ y(t) &= T(t) - T_s \end{aligned} \tag{57}$$

where

F = volumetric feed flowrate
V = CSTR volume = 1.36 m^3
C_A = concentration of A in CSTR
C_{Ai} = concentration of A in inlet stream = 8,008 mol/m^3
α = kinetic constant = 7.08×10^7 $1/hr$
E/R = activation energy/gas constant = 8,375 K
T = temperature in CSTR
T_i = inlet temperature = 373.3 K
ΔH_R = heat of reaction = -69,775 j/mol
ρ = density of liquid in CSTR = 800.8 kg/m^3
c_p = specific heat of liquid in CSTR = 3,140 $j/kg\text{-}K$
Q = heat removal rate = 1.055×10^8 j/hr

The reactor has three steady states. Eigenvalue analysis of the linearized system around the steady state corresponding to

$$\begin{aligned} F_s &= 1.133\ m^3/hr \\ T_s &= 547.6\ K \\ C_{As} &= 393.2\ mol/m^3 \end{aligned} \tag{58}$$

reveals that the above steady-state is locally stable with respect to initial conditions. Consequently, it is input-output stable for inputs of small enough magnitude. To determine what "small enough" means in the previous sentence requires careful analysis. For example, single pulse changes in u of magnitude $-0.5F_s$ or $+0.5F_s$ do not reveal any instabilities (the CSTR returns to the original steady state, Figure 6 and Figure 7), but successive pulse changes of magnitudes $-0.5F_s$ and then $+0.5F_s$ drive the CSTR to instability ($\|S\|_{i,22} = \infty$), as Figure 8 demonstrates. Therefore, this CSTR is not input-output stable for u bounded in the interval $[-0.5F_s, 0.5F_s]$.

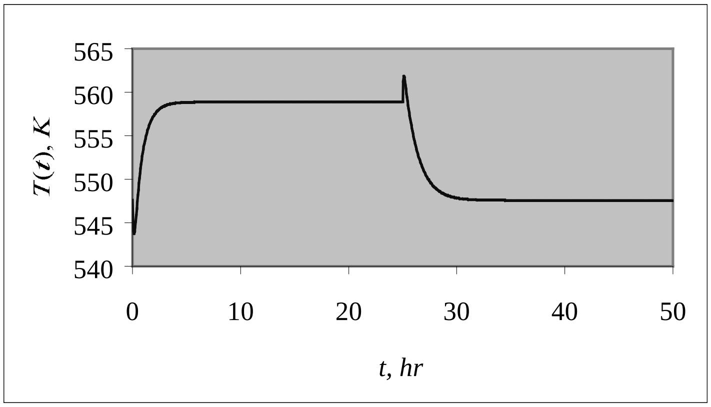


Figure 6. Response of CSTR to the flowrate pulse $F(t) = \begin{cases} F_s, & t < 0 \\ 1.5F_s, & 0 \le t < 25 \\ F_s, & t \ge 25 \end{cases}$

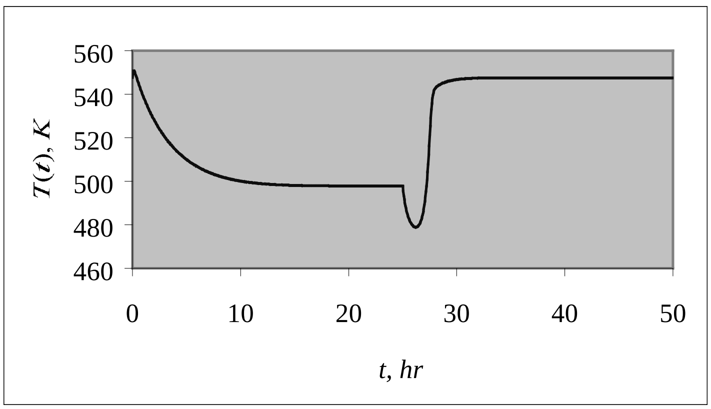


Figure 7. Response of CSTR to the flowrate pulse $F(t) = \begin{cases} F_s, & t < 0 \\ 0.5F_s, & 0 \le t < 25 \\ F_s, & t \ge 25 \end{cases}$

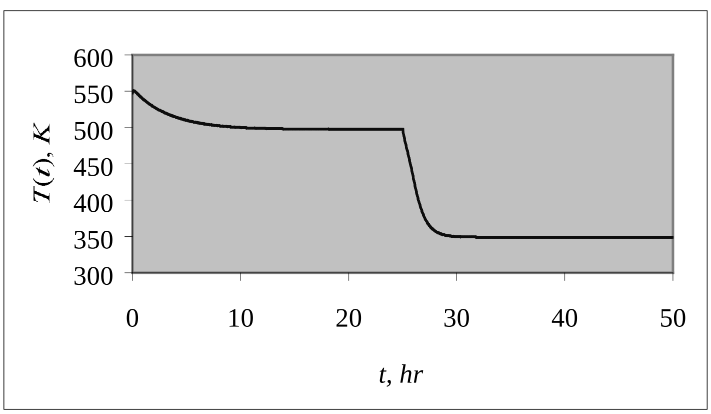


Figure 8. Response of CSTR to the flowrate pulses $F(t) = \begin{cases} F_s, & t < 0 \\ 0.5F_s, & 0 \le t < 25 \\ 1.5F_s, & t \ge 25 \end{cases}$

Choi and Manousiouthakis (1997) recently introduced the concept of finite-gain/initial conditions stability, to combine the insights provided by each of the above two kinds of stability:

Definition 9 – Finite-gain/initial conditions stability

A system $S: l_p^m \to l_q^n : \mathbf{u} \mapsto \mathbf{x} \triangleq S\mathbf{u}$, is *finite-gain* stable over the set **U** for initial conditions **s**[0] in the set **S** if the following inequality holds:

(59)
$$\sup_{\substack{\mathbf{u}\in\mathbf{U} \\ \mathbf{s}[0]\in\mathbf{S}}} \frac{\|\mathbf{x}\|_q}{\|\mathbf{u}\|_p} < \infty$$

The advantage of the above definition is that it gives a complete characterization of the stability behavior of a system. Its disadvantage is that the computation of the left-hand side in eqn. (59) is not trivial.

3.2 Is stability important?

As the above section 3.1 emphasized, stability is a fundamental property of a dynamic system that summarizes the long-term behavior of that system. There are two important implications of this statement:

(a) MPC controllers should result in closed-loops that are stable. Therefore, if an optimal MPC system is to be designed, only candidates from a set of stabilizing MPC controllers should be considered in the design. Ideally, the set of all stabilizing MPC controllers should be known. That set is difficult to determine for constrained MPC, given the richness allowed in the structure of constrained MPC. However, one can find subsets of the set of all stabilizing MPC controllers with constraints. Selecting MPC controllers from such a subset can have significant implications on closed-loop performance, as section 5.3 demonstrates. For unconstrained MPC controllers with linear models, which are equivalent to linear, time-invariant controllers as shown in section 2.3, the set of all controllers that can stabilize a given linear plant can be explicitly parametrized in terms of a single stable transfer function through the celebrated Youla-parametrization (Vidyasagar, 1985). For stable plants, the Youla-parametrization is the same as the internal model control (IMC) structure (Morari and Zafiriou, 1989).

(b) The above discussion in section 3.1 is most relevant for continuous processes, for which operating time can theoretically extend to infinity. For batch processes, operating time is finite, consequently stability should be examined in a different framework. For example, instability that would generate signals that grow without bounds might not be detrimental, provided that the rate of growth is very small with respect to the batch cycle time.

4 The behavior of MPC systems

The examples that follow attempt to expose some MPC pitfalls. All examples are intentionally kept as simple as possible, to easily expose the theoretical issues associated with each one of them. Because of that simplicity, the issues highlighted in each example could be easily addressed by intuitive design (tuning) improvements of the corresponding MPC systems. Our purpose, however, is not to use these examples to test the ultimate capabilities of MPC. Large multivariable systems, for example, would be better suited for that purpose. Our intention is to help the reader understand the theoretical issues associated with these examples by easily following some of the associated calculations.

4.1 Feasibility of on-line optimization

Because MPC requires the solution of an optimization problem at each time step, the feasibility of that problem must be ensured. For example, the optimization problem posed in equations (14) to (17) may be infeasible. If the on-line optimization problem is not feasible, then some constraints would have to be relaxed. Finding what constraints to relax in order to get a feasible problem with optimal deterioration of the objective function is extremely difficult, since it is an *np*-hard problem. A possible (and partial) remedy to the problem is to consider constraint softening variables ε on process output constraints, e.g.

(60)
$$y_{\max} + \varepsilon \geq y[k+i|k] \geq y_{\min} - \varepsilon,\ i = 1, \cdots, p$$

and include a penalty term such as ε^2 in the objective function. As we will discuss in section, feasibility, in addition to being a practical consideration, is also important for closed-loop stability of MPC. In fact, algorithms have been developed by Mayne and co-workers, which merely require the existence of a feasible instead of optimal solution of the on-line optimization problem, to guarantee closed-loop stability of an MPC system.

4.2 Nonminimum phase and short horizons

Example 6 – Closed-loop stability for a nonminimum-phase process

Consider a nonminimum-phase process described by the equation

(61)
$$y[k] = h_1 u[k-1] + h_2 u[k-2] + h_3 u[k-3] + h_4 u[k-4] + d[k]$$

with $h_1 = 0$ (dead-time), $h_2 = -1$ (inverse response), $h_3 = 2$, $h_4 = 0$ (Genceli and Nikolaou, 1993; Genceli, 1993). Assume no modeling uncertainty. For that process consider the constrained MPC on-line optimization

(62)
$$\min_{u[k|k],\cdots,u[k+p-1|k]} \sum_{i=1}^{p} \left| y[k+i|k] - y^{sp} \right| + \sum_{i=0}^{m} r_i \left| \Delta u[k+i|k] \right|$$

with , $p = 3$ and $m = 1$, subject to the constraints

$$u_{\min} = -0.2 \le u[k+i|k] \le 0.2 = u_{\max}$$

The above optimization can be trivially transformed to linear programming. A step disturbance equal to –0.05 and a step setpoint change equal to 0.05 enter the closed loop at time *k*=0. For move suppression coefficient values $r_0 = r_1 < 0.5$ the resulting closed-loop response is shown in Figure 9.

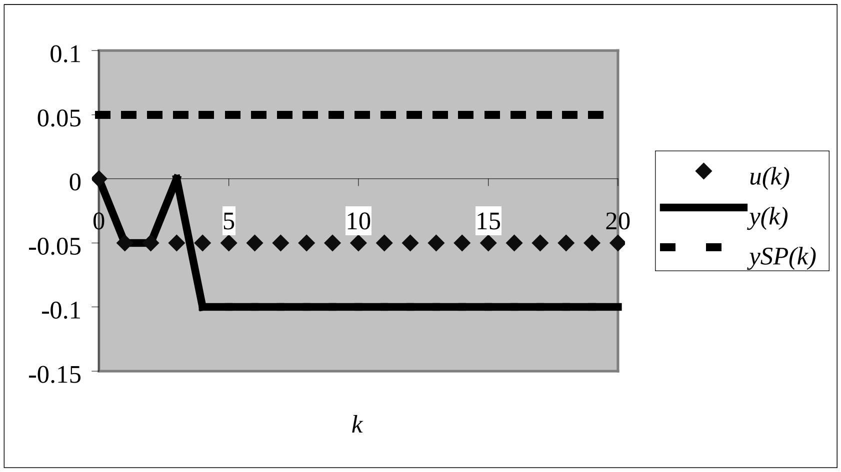


Figure 9. Closed-loop response for the process of Example 6, with $r_0 = r_1 < 0.5$

The closed-loop is clearly unstable. If the move suppression coefficients take values $r_0 = r_1 \ge 0.5$ (to penalize the move suppression coefficients even more) the closed loop remains unstable, as shown in the following Figure.

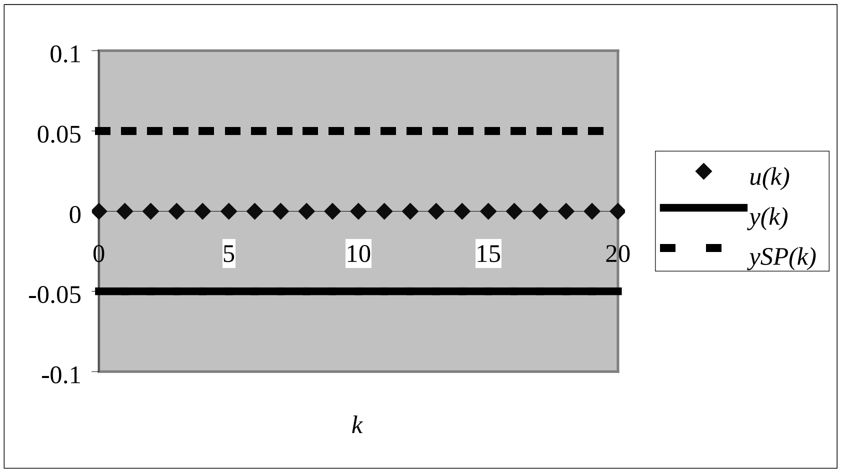


Figure 10. Closed-loop response for the process of Example 6, with $r_0 = r_1 \ge 0.5$

The instability is due to the non-minimum phase characteristics of the process. While a longer optimization horizon length, p, might easily solve the problem, a simple remedy can also be obtained by considering the following end-constraint on u in the on-line optimization:

(63)
$$u[k+i|k] = \frac{y^{SP} - y[k] + \sum_{j=1}^{N} g_j u[k-j]}{\sum_{j=1}^{N} g_j} \quad \text{for all} \quad i \geq m$$

where g_j are model estimates of the coefficients h_j. The meaning of the above eqn. (63) is that the value of the process input u at the end of its horizon should correspond to a steady state value that would produce zero steady state offset

$$y^{SP} - \left(u[k+i|k] \sum_{j=1}^{N} g_j + \overbrace{y[k] - \sum_{j=1}^{N} g_j u[k-j]}^{d[k+\infty|k]} \right)$$

for the process model output $y[k+\infty|k]$. The closed-loop response for move suppression coefficient values $r_0 = r_1 = 2.7$ is shown in Figure 11. It turns out that these values for the move suppression coefficients are sufficient for robust stability of the closed loop, when the modeling errors e_1, e_2, e_3, e_4 for the coefficients h_1, h_2, h_3, h_4 are bounded as $|e_1| = |h_1 - g_1| \leq 0.12$, $|e_2| = |h_2 - g_2| \leq 0.10$, $|e_3| = |h_3 - g_3| \leq 0.08$, $|e_4| = |h_4 - g_4| \leq 0.05$ (Genceli and Nikolaou, 1993). The key to achieving stability was the end-constraint, eqn. (63). It turns out that inclusion of an end-constraint of the type

(64)
$$\mathbf{x}[p] = \mathbf{0}$$

in the on-line optimization performed by MPC is a convenient way to generate a controller structure for which stability can be easily shown (section 5.1).

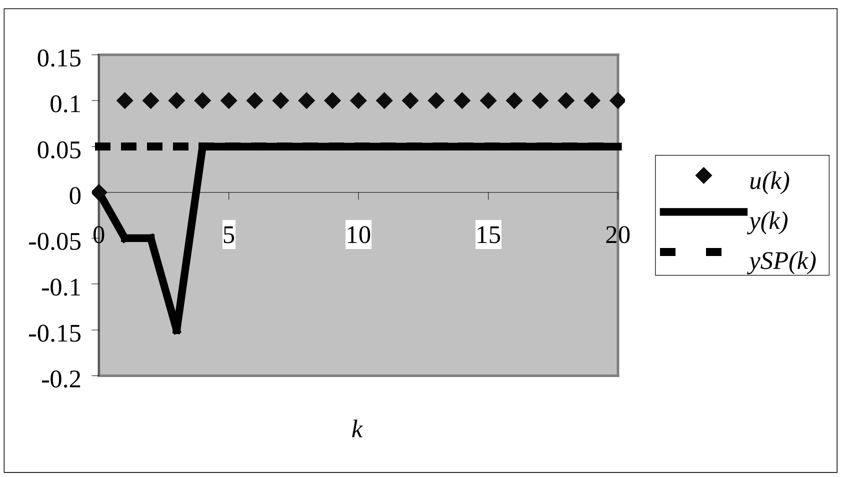


Figure 11. Closed-loop response for the process of Example 6, with $r_0 = r_1 = 2.7$, end-condition enforced.

4.3 Integrators and Unstable Units

With the dimension of multivariable MPC systems ever increasing, the probability of dealing with a MIMO process that contains an integrator or an unstable unit also increases. For such units the use of FIR models, as used by certain traditional commercial algorithms such as dynamic matrix control (DMC), is not feasible. Integrators or unstable units raise no problems if state-space or DARMAX model MPC formulations are used. As we will discuss below, theory developed for MPC with state-space or DARMAX models encompasses all linear, time-invariant, lumped-parameter systems and consequently has broader applicability.

In contrast to constrained MPC of *stable* plants, constrained MPC of *unstable* plants has the complication that the tightness of constraints, the magnitude and pattern of external signals, and the initial conditions *all* affect the stability of the closed loop. The following simple example illustrates what may happen with a simple unstable system.

Example 7 – Stability regions

Consider the unstable plant $P(s) = \frac{-s+1}{s-3}$ controlled by the P controller $C(s) = 2$. The controller output is constrained between –1 and 1. A disturbance, d, is added to the controller output, to create the final input to the plant. The following figures show responses to three different disturbances.

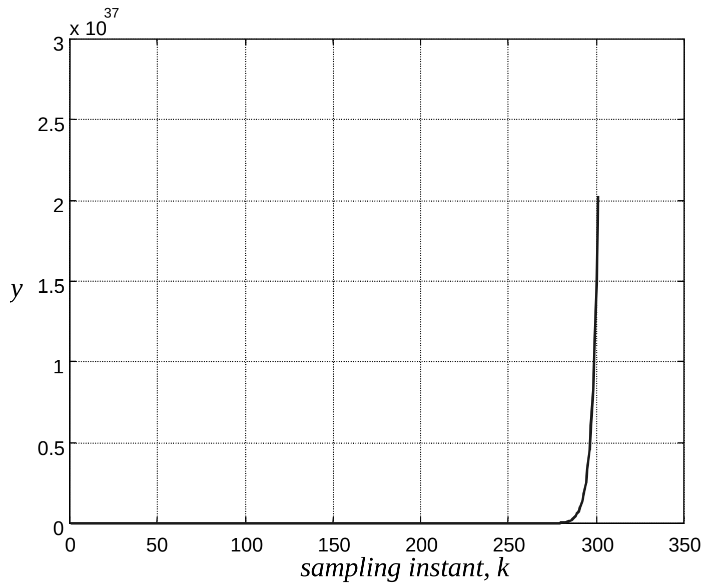


Figure 12. Plant response to step disturbance $d = 0.5$ at $t = 1$ ($k = 10$) for Example 7.

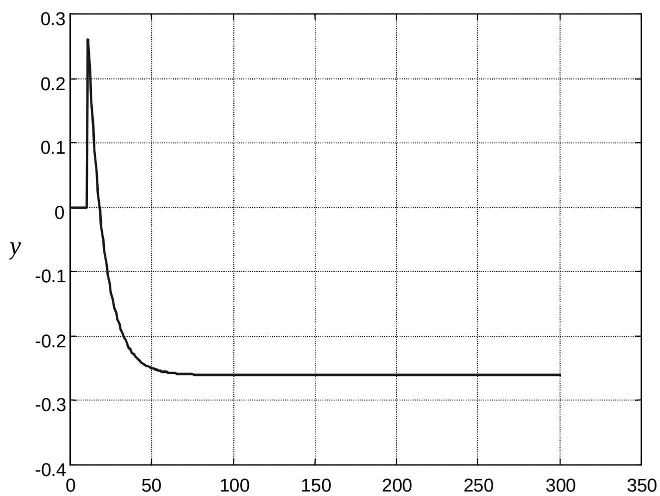


Figure 13. Plant response to step disturbance $d = 0.26$ at $t = 1$ ($k = 10$) for Example 7.

The response in Figure 12, resulting from the step disturbance $d = 0.5$, clearly corresponds to unstable closed-loop behavior. The response in Figure 13, resulting from a smaller step disturbance $d = 0.26$, shows a bounded plant output. However, one cannot say that disturbances of amplitude 0.26 or smaller result in stable closed-loop behavior. Indeed, as Figure 15 shows, the plant response to the pulse disturbance of amplitude 0.26, shown in Figure 14, is clearly unstable.

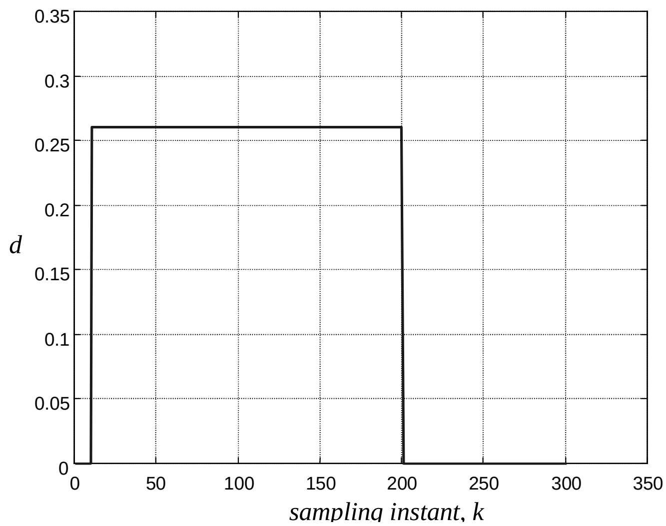


Figure 14. Pulse disturbance for Example 7.

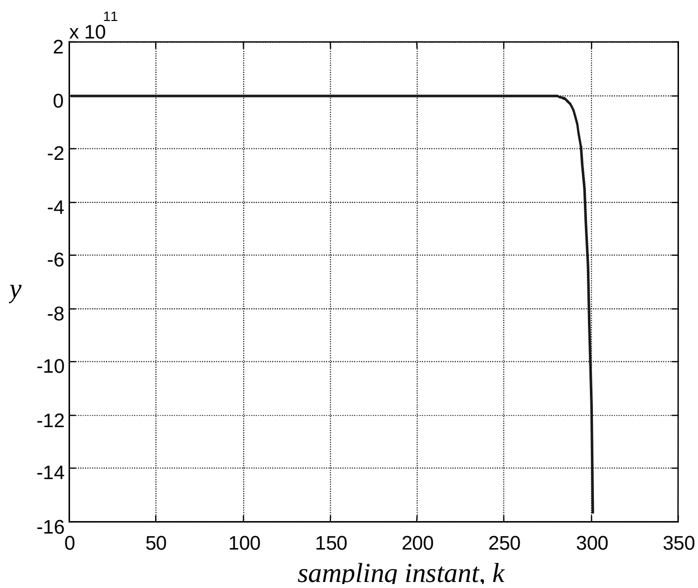


Figure 15. Output for pulse disturbance in Example 7.

The above simple example makes it clear that characterization of external inputs that do not destabilize a closed loop is by no means trivial, when the plant is unstable. Similar comments can be made about dependence of stability on the initial state of the system. For the case of a linear, unstable system with bounded inputs and without external disturbances, Zheng and Morari (1995) have developed an algorithm that can determine the domain of attraction for the initial state of the system.

4.4 Nonlinearity

MPC systems that employ nonlinear models may exhibit increased complexity due to two main factors:

(a) Nonlinear programming, required for the solution of the MPC on-line optimization problem, does not produce exact solutions but rather solutions that are optimal within a certain prespecified precision tolerance, or even locally optimal, if the optimization problem is nonconvex.

(b) Even if the global optimum of the on-line optimization problem is assumed to be exactly reached, MPC behavior may show patterns that would not be intuitively expected. For instance, Rawlings et al. (1994) discuss two simple examples of MPC applied to nonlinear systems, where the state feedback law turns out to be a discontinuous function of the state, either because of stability requirements, or due to the structure of MPC. As a result, standard stability results that rely on continuity of the feedback law cannot be employed.

Example 8 – A nonlinear process that cannot be stabilized by a continuous feedback law

In the first example, the following two-state, one-input system is considered:

$$x_1[k+1] = x_1[k] + u[k]$$

(65) $$x_2[k+1] = x_2[k] + u[k]^3$$

$$x_1[0],\ x_2[0] \ \ \text{given}$$

Meadows et al. (1995) showed that the following MPC nonlinear program, corresponding to a moving horizon of length 3, results in a closed loop that is globally asymptotically stable around the equilibrium point $\mathbf{x}_e = (0,0)$. (Recall that an equilibrium point x_e is globally asymptotically stable if $\mathbf{x}[k] \to \mathbf{x}_e$ as $k \to \infty$ for any initial point $\mathbf{x}[0]$.)

(66) $$\min_{u[k|k],u[k+1|k],u[k+2|k]} \sum_{i=0}^{2} \left(x_1[k+i|k]^2 + x_2[k+i|k]^2 + u[k+i|k]^2 \right)$$

subject to the terminal constraint

(67) $$x_1[k+3|k] = x_2[k+3|k] = 0$$

Meadows et al. (1995) also showed that horizons of length less than 3 cannot globally asymptotically stabilize this system, while horizons of length larger than 3 will result in less aggressive control action. The control law u(x) for the above horizon of length 3 turns out to be continuous at the origin, but has discontinuity points away from the origin. In fact, no continuous state feedback law can stabilize the system of eqns. (65). To show that, following Meadows et al. (1995), first note that any stabilizing control law must allow both positive and negative input values for **x**. If the control is strictly positive, trajectories originating in the first quadrant move away from the origin under positive control action. If the control is strictly negative, trajectories originating in the third quadrant also move away from the origin. Yet $u(\mathbf{x})$ cannot be identically zero for any nonzero **x**. If it were, then this **x** would be a fixed point of the dynamic system and trajectories containing this **x** would not converge to the origin. We have the situation in which the feedback control law must assume both negative and positive values away from the origin, yet must be zero nowhere away from the origin. Therefore, the control law must be discontinuous.

Example 9 – A finite prediction horizon may not be a good approximation of an infinite one for nonlinear processes

In the second example, consider the following single-state, single-input system:

(68) $$x[k+1] = x[k]^2 + u[k]^2 - (x[k]^2 + u[k]^2)^2$$

with the MPC controller

(69) $$\min_{u[k|k],u[k+1|k]} \sum_{i=0}^{1} \left(x[k+i|k]^2 + u[k+i|k]^2 \right)$$

subject to the terminal constraint

(70) $$x[k+2|k] = 0$$

which is feasible if the initial state is restricted such as $|x[k]| \le 1$. The control law resulting from the above optimization, eqns. (69) and (70), is

(71) $$u(x) = \begin{cases} 0 & x = 0 \\ \pm\sqrt{1-x^2} & 0 < |x| \le 1 \end{cases}$$

resulting in an optimal cost

(72) $$J_{opt}(x) = \begin{cases} 0 & x = 0 \\ 1 & 0 < |x| \le 1 \end{cases}.$$

Both $u(x)$ and $J_{opt}(x)$ are discontinuous at the origin. Therefore, stability theorems that rely on continuity cannot be used. Yet, it is simple to check by inspection that the feedback law of eqn. (71) (with either sign chosen) is asymptotically stabilizing. However, the continuous feedback control law $u(x) = 0$, resulting in the closed-loop system

(73) $$x[k+1] = x[k]^2 - x[k]^4 \text{,}$$

is asymptotically stabilizing for initial conditions in [-1, 1]. The actual closed-loop cost incurred using this feedback control law is $\sum_{k=0}^{\infty} x[k]^2$.

It turns out, (Rawlings, 1994) that for initial conditions in [-1, 1], the closed-loop cost of the zero control action is always less than that for the optimal MPC controller with fixed horizon. Since the actual incurred cost is calculated over an infinite horizon, it is reasonable to ask whether the minimum cost of the finite horizon MPC problem would approach the cost incurred using the zero controller as the horizon length tends to infinity. The answer is negative. In the finite horizon MPC on-line optimization problem we require that the terminal constraint $x[p] = 0$ be satisfied, where p is the horizon length. Because of the structure of the problem, the closed-loop MPC cost is 1 for all horizon lengths. This examples demonstrates that the intuitive idea of using the terminal constraint $x[p] = 0$ and a large value of p in order to approximate the desired infinite horizon behavior, an idea that works for linear systems, does not work in general for nonlinear systems.

4.5 Model uncertainty

Example 10 – Ensuring robust stability of a heavy oil fractionator

Vuthandam et al. (1995) considered the top 2×2 subsystem of the heavy oil fractionator modeled in the Shell Standard Process Control Problem (Prett and Garcia, 1988) as

(74) $$P(s) = \begin{bmatrix} \frac{4.05e^{-27s}}{50s+1} & \frac{1.77e^{-28s}}{60s+1} \\ \frac{5.39e^{-18s}}{50s+1} & \frac{4.05e^{-27s}}{60s+1} \end{bmatrix}$$

Discretization for a sampling period of 4 minutes yields a corresponding discrete-time model that is used in the following MPC on-line objective function

(75) $$J[k] \hat{=} \sum_{j=1}^{2}\sum_{i=1}^{7}\left(y_j[k+i|k] - y_j^{SP}\right)^2 + \sum_{j=1}^{2}\sum_{i=0}^{3} r_{j,i}\Delta u_j[k+i|k]^2 + \sum_{i=1}^{7}\varepsilon_1[k+i|k]^2$$

with the constraints

(76) $$\begin{gathered} -3 \le \Delta u_2[k] \le 3 \\ -5 \le u_2[k] \le 5 \\ -0.5 - \varepsilon_1[k+i|k] \le y_1[k+i|k] \le 0.5 + \varepsilon_1[k+i|k] \end{gathered}$$

with setpoints

(77) $$y_1^{SP} = y_2^{SP} = 0$$

and step disturbances

(78) $$\begin{aligned} d_1 &= 1.2 \\ d_2 &= -0.5 \end{aligned}$$

In addition to the above constraints, the end-constraint

(79) $$\mathbf{u}[k+m+i|k] = \mathbf{G}^{-1}(\mathbf{y}^{sp} - \mathbf{d}[k|k]) \text{,} \quad i \ge 0$$

is considered, where **G** is the steady-state gain of the process. Simulation of the above system verified the robust stability analysis of the above authors, as shown in the following table and corresponding figures.

End-condition, eqn. (79)	Input move suppression coefficients	Closed-loop behavior
Not used	$r_{10} = r_{20} = 0.10$ $r_{11} = r_{21} = 0.12$ $r_{12} = r_{22} = 0.12$ $r_{13} = r_{23} = 0.12$	Unstable

Used	$r_{10} = r_{20} = 0.06$ $r_{11} = r_{21} = 0.07$ $r_{12} = r_{22} = 0.07$ $r_{13} = r_{23} = 0.07$	Unstable
Used	$r_{10} = r_{20} = 10.82$ $r_{11} = r_{21} = 11.15$ $r_{12} = r_{22} = 11.46$ $r_{13} = r_{23} = 11.86$	Robustly stable (rigorous proof)

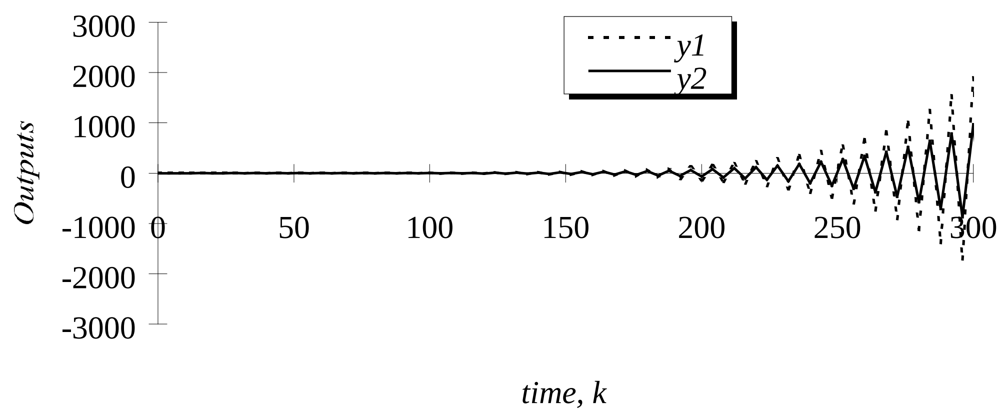


Figure 16. Closed-loop response for Example 10, Case 1.

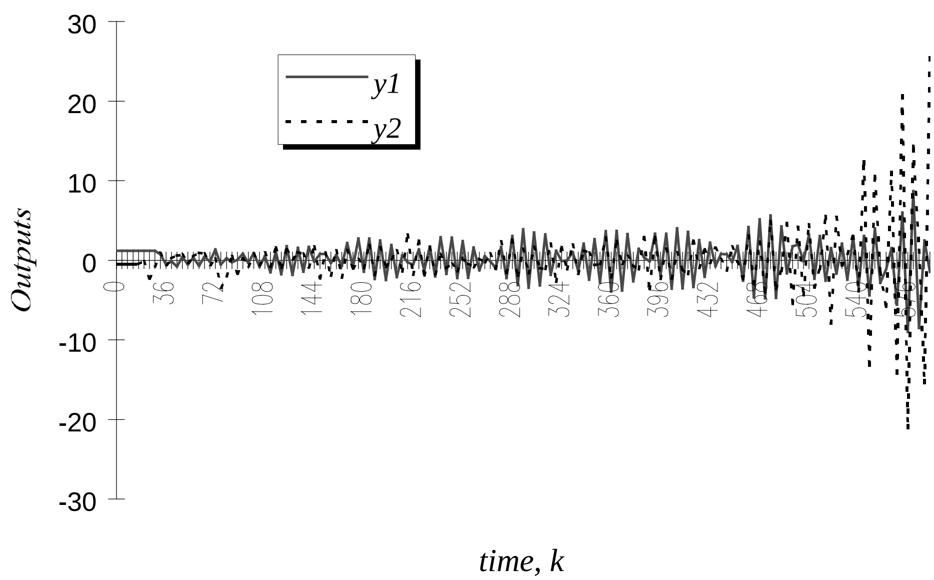


Figure 17. Closed-loop response for Example 10, Case 2.

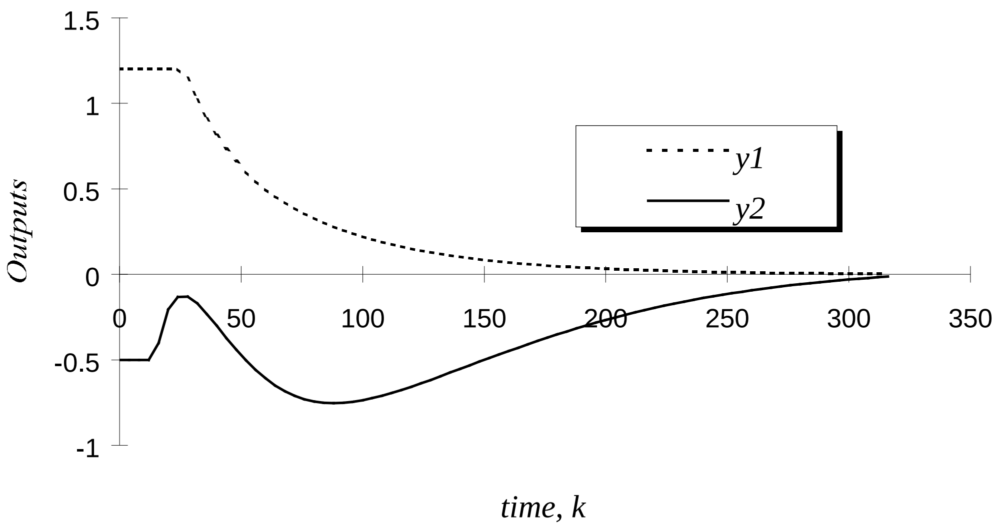


Figure 18. Closed-loop response for Example 10, Case 3.

4.6 Fragility

Because MPC relies on the numerical solution of an on-line optimization problem, it may find a solution to that problem which is not *exactly* equal to the expected theoretical solution. Is closed-loop stability going to be adversely affected by that discrepancy between the theoretically expected MPC behavior and the actual (numerical) MPC behavior? An affirmative answer was given by Keel and Bhattacharya (1997), who demonstrated, by example, that there are linear time-invariant stabilizing controllers for which extremely small variations of their coefficients may render the controllers destabilizing, even though the controllers may nominally satisfy optimality criteria such as H_2, H_∞, l_1, or μ, as well as robustness criteria. Borrowing from the above authors, consider the following example:

Example 11 – Sensitivity of closed-loop stability to small variations in controller parameters

For the stable transfer function

(80)
$$P(s) = \frac{-s+1}{s^2+s+2}$$

the optimal controller

(81)
$$C(s) = \frac{q_6 s^6 + q_5 s^5 + q_4 s^4 + q_3 s^3 + q_2 s^2 + q_1 s + q_0}{p_6 s^6 + p_5 s^5 + p_4 s^4 + p_3 s^3 + p_2 s^2 + p_1 s}$$

is designed by minimizing a weighted H_2 norm of the closed-loop transfer function. The values of the controller parameters are given in Table 1.

Table 1

Parameter	*Value*	*Parameter*	*Value*
q_6	1.0002	p_6	0.0001
q_5	3.0406	p_5	1.0205
q_4	8.1210	p_4	2.1007
q_3	13.2010	p_3	5.1403
q_2	15.2004	p_2	6.06
q_1	12.08	p_1	2.0
q_0	4.0		

The poles of the resulting closed loop are well in the left-half plane, as supported by the Nyquist plot of $P(s)C(s)$ shown in Figure 19. Therefore the nominal closed-loop is stable. Yet a small change $\Delta\mathbf{p}$ in the nominal controller parameters $\mathbf{p}$, such that

(82)
$$\frac{\|\Delta\mathbf{p}\|_2}{\|\mathbf{p}\|_2} = 3.74\times 10^{-6}$$

can destabilize the closed loop. For example, if

(83)
$$\Delta\mathbf{p} = 10^{-4}\times[-0.321 \quad -0.009 \quad 0.002 \quad 0.000 \quad -0.000 \quad -0.000 \quad 0.000 \quad -1.000 \quad 0.332 \quad 0.005 \quad -0.002 \quad -0.000 \quad 0.000]^T$$

then the closed loop has a pole at ~10^8.

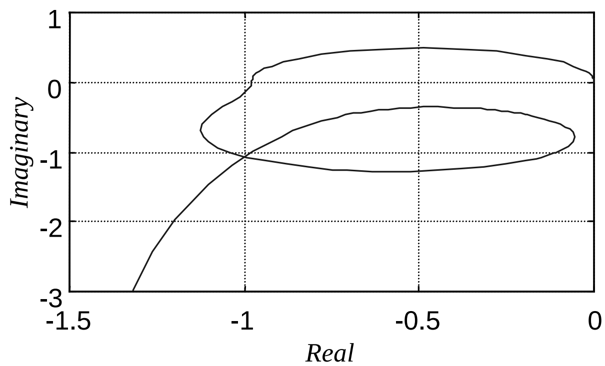


Figure 19. Nyquist plot of $P(s)C(s)$ for Example 11.

The above problem of extreme sensitivity of closed-loop stability to small variations in controller parameters has been termed fragility. Given that unconstrained MPC with quadratic cost and linear model is equivalent to a linear time-invariant controller, as demonstrated in section 2.3, it is clear that similar fragility problems may appear with MPC as well. Fragility might have a more realistic probability of being an instability threat in constrained MPC, where, as discussed in more detail in Section 4.6, the results of on-line optimization may not be exact, such as in the case of nonlinear programming with multiple optima or with equality constraints. Fragility problems may even emerge in computer implementation of control algorithms where floating-point arithmetic introduces a truncation (round-off) error (Williamson, 1991).

Of course, MPC controllers would have to be robust with respect to plant uncertainty, which is usually orders of magnitude larger than controller uncertainty. From that viewpoint, controller fragility would be an issue of practical significance if small controller uncertainty could cause instability for plants close to or at the boundary of the set of uncertain plants considered in controller design.

4.7 Constraints

Zafiriou (1991) used a number of examples to demonstrate that the presence of constraints can have a dramatic and often counter-intuitive effect on MPC stability properties and can render tuning rules developed for stability or robustness of unconstrained MPC incorrect. The following examples show how the addition of constraints to a robustly stable unconstrained MPC system can lead to instabilities.

Example 12

Consider the process

(84)
$$p(s) = \frac{e^{-0.15s}}{s+1}$$

modeled by

(85)
$$\tilde{p}(s) = \frac{1}{s+1}$$

A sampling period of 0.1 is used. The following MPC system is used to control the process:

(86)
$$\min_{\Delta u[k]} y[k+1|k]^2 + 0.16\Delta u[k|k]^2$$

(87) $$-1 \leq y[k+1|k] \leq 1$$

Two points are to be made:

(a) While for step disturbances $d \leq 1.70$ the output y returns to the setpoint $y^{sp} = 0$, for step disturbances $d \geq 1.75$ the output oscillates with amplitude that grows without bound. Therefore, unlike in the case of linear systems, the stability characteristics of the above constrained MPC system depend on the magnitude of external disturbances.

(b) Perhaps counterintuitively, relaxing the controller by removing the output constraint, eqn. (87), can be shown to result in a linear time-invariant controller that robustly stabilizes the closed loop for disturbances of any amplitude.

Example 13

Consider the process

(88) $$p(s) = \tilde{p}(s) = \frac{-s+1}{(s+1)(2s+1)}$$

A sampling period of 0.3 and an FIR model with n=50 coefficients are used. The following MPC system is used to control the process:

(89) $$\min_{\Delta u[k|k],\cdots,\Delta u[k+m-1|k]} \sum_{i=1}^{p} y[k+i|k]^2$$

(90) $$-0.3 \leq y[k+2|k] \leq 0.3$$

If the output constraint, eqn. (4) were not present, then the choice $m = 1$ and a sufficiently large $p \geq n + m$ would stabilize the closed loop, in the absence of process/model mismatch. However, the presence of the output constraint destabilizes the closed loop. As $p \to \infty$, then the closed loop largest root approaches 1.45, for m=1, and 2.63, for m=2. Again, the presence of output constraints destabilizes the closed loop instead of tightening control.

5 A theory for MPC with predictable properties

5.1 Stability

Because no equivalent linear time-invariant controller exists for a constrained MPC controller, determining the stability of a constrained MPC closed-loop system cannot rely on methods suitable only for linear systems, such as determination of the poles of the closed-loop transfer matrix. Recent developments have shed light on the stability of constrained MPC. The seeds of proving stability for constrained MPC were already planted by Economou (1985) who postulated, before his life reached a tragic end, that closed-loop stability for constrained MPC could be established using the contraction mapping theorem. Mayne and Michalska (1991) and Rawlings and Muske (1993), through seminal papers, developed working frameworks for establishing closed-loop stability of continuous-time and discrete-time constrained MPC systems, respectively. These works provided the right tools for the study of a long considered difficult control problem. Following below is a discussion of pertinent ideas.

5.1.1 MPC with linear model – A prototypical stability proof

Consider the following state-space description of a linear, time-invariant (not necessarily stable) process

(91) $$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k]$$

where $\mathbf{x} \in \Re^{n_x}$, $\mathbf{u} \in \Re^{n_u}$. We assume perfect knowledge of the matrices **A** and **B**, full state information **x**[k], and absence of disturbances. The MPC objective at time k is

(92) $$\min_{\mathbf{u}[k|k],\cdots,\mathbf{u}[k+p-1|k]} J[k]$$

where

(93) $$J[k] \mathrel{\hat{=}} \sum_{i=1}^{p} \mathbf{x}[k+i|k]^T \mathbf{W}\mathbf{x}[k+i|k] + \sum_{i=1}^{p} \mathbf{u}[k+i-1|k]^T \mathbf{R}\mathbf{u}[k+i-1|k]$$

W is a positive definite matrix and **R** is a positive semi-definite matrix. State and input constraints are

(94) $$\mathbf{G}[k+i-1]\mathbf{u}[k+i-1|k] \leq \mathbf{g}[k+i-1], \quad i = 1,\cdots,p$$

(95) $$\mathbf{H}[k+i]\mathbf{x}[k+i|k] \le \mathbf{h}[k+i], \quad i = 1, \cdots, p$$

The above constraints are assumed to define non-empty (convex) regions containing the point (**0**, **0**).

Closed-loop MPC stability can be established using the following Lyapunov argument (Rawlings et al., 1994). Assume that $\mathbf{G}[k+i-1]$, $\mathbf{g}[k+i-1]$, $\mathbf{H}[k+i]$, and $\mathbf{h}[k+i]$ are independent of *k* and *i*. Consider a solution

(96) $$\mathbf{U}_{opt\, k|k}^{\;\;k+p-1|k} = \{\mathbf{u}_{opt}[k|k], \quad \cdots, \quad \mathbf{u}_{opt}[k+p-1|k]\}$$

to eqn. (92) at time *k*, and assume that *p* is large enough so that $\mathbf{x}[p] = \mathbf{0}$. Consider the following candidate for control input at time *k*+1:

(97) $$\mathbf{U}_{k+1|k+1}^{k+p|k+1} = \{\mathbf{u}_{opt}[k+1|k], \cdots, \mathbf{u}_{opt}[k+p-1|k], \mathbf{0}\}.$$

$\mathbf{U}_{k+1|k+1}^{k+p|k+1}$ is feasible at time *k*+1, because it contains inputs **u** that satisfied the same constraints at time *k*. The above feasible input results in a value of the objective function *J*[*k*+1] that satisfies

(98) $$J[k+1] = J_{opt}[k] - \mathbf{x}[k]^T \mathbf{W}\mathbf{x}[k] - \mathbf{u}[k]^T \mathbf{R}\mathbf{u}[k]$$

Because of optimality, the above equation yields

(99) $$\begin{aligned} J_{opt}[k+1] &\le J[k+1] \\ &= J_{opt}[k] - \mathbf{x}[k]^T \mathbf{W}\mathbf{x}[k] - \mathbf{u}[k]^T \mathbf{R}\mathbf{u}[k] \\ &\le J_{opt}[k] \end{aligned}$$

where the last inequality results from the positive semi-definiteness of **W** and positive definiteness of **R**. Therefore, the sequence $\{J[k]\}_{k=k_0}^{\infty}$ is non-increasing. It is also bounded from below by 0. Consequently, the sequence $\{J[k]\}_{k=k_0}^{\infty}$ converges, i.e. $\lim_{k\to\infty} J[k] = a$. To show that $a = 0$, rearrange eqn. (99), to get

$$\mathbf{x}[k]^T \mathbf{W}\mathbf{x}[k] + \mathbf{u}[k]^T \mathbf{R}\mathbf{u}[k] \le J_{opt}[k] - J_{opt}[k+1] \Rightarrow$$

$$\lim_{k\to\infty} \left(\mathbf{x}[k]^T \mathbf{W}\mathbf{x}[k] + \mathbf{u}[k]^T \mathbf{R}\mathbf{u}[k]\right) \le \lim_{k\to\infty} J_{opt}[k] - \lim_{k\to\infty} J_{opt}[k+1] = 0 \Rightarrow$$

$$\lim_{k\to\infty} \mathbf{x}[k] = \mathbf{0}, \quad \lim_{k\to\infty} \mathbf{u}[k] = \mathbf{0}$$

and closed-loop stability is proven.

5.1.2 MPC with nonlinear model

The Lyapunov-like stability proof presented in section 5.1.1 can be extended to nonlinear systems. Because nonlinear systems are defined by what they are not (namely linear), the framework of the ensuing proof must be precisely defined. Borrowing from Rawlings (1994) and Meadows et al. (1995), we consider dynamical systems of the form

(100) $$\mathbf{x}[k+1] = \mathbf{f}(\mathbf{x}[k], \mathbf{u}[k]).$$

where $\mathbf{f} : \Re^n \times \Re^m \to \Re^n$ is continuous and $\mathbf{f}(\mathbf{0},\mathbf{0}) = \mathbf{0}$. The state **x** is assumed to be measurable. The MPC on-line optimization problem becomes

(101) $$\min_{\mathbf{u}[k|k], \cdots, \mathbf{u}[k+p-1|k]} I[k]$$

where

(102) $$I[k] \hat{=} \sum_{i=0}^{p-1} L(\mathbf{x}[k+i|k], \mathbf{u}[k+i|k])$$

subject to eqn. (101) and the state, input, and terminal constraints

(103) $$\mathbf{x}[k+i|k] \in X[k+i], \quad i = 0, \cdots, p-1$$

(104) $$\mathbf{u}[k+i|k] \in U[k+i], \quad i = 0, \cdots, p-1.$$

The function $L : \Re^n \times \Re^m \to \Re$ that appears in eqn. (102) is assumed to satisfy the following properties:

(105) $$L(\mathbf{0},\mathbf{0}) = 0$$

(106) There exists a nondecreasing function $\gamma : [0,\infty) \to [0,\infty)$ such that $\gamma(0) = 0$ and $0 < \gamma(\|\mathbf{x},\mathbf{u}\|) \le L(\mathbf{x},\mathbf{u})$ for all $(\mathbf{x},\mathbf{u}) \ne (\mathbf{0},\mathbf{0})$.

These lead to the following additional properties of *L*:

(107) $$L(\mathbf{x},\mathbf{u}) > 0 \ \forall(\mathbf{x},\mathbf{u}) \ne (\mathbf{0},\mathbf{0})$$

(108) $$L(\mathbf{x},\mathbf{u}) = 0 \Leftrightarrow (\mathbf{x},\mathbf{u}) = (\mathbf{0},\mathbf{0})$$

(109) $$L(\mathbf{x},\mathbf{u}) \to 0 \Rightarrow (\mathbf{x},\mathbf{u}) \to (\mathbf{0},\mathbf{0})$$

Notice that the function *J* in eqn. (93) satisfies all of the above conditions.

As in the linear case, a proof for closed-loop stability can be constructed, if it can be guaranteed that

(110) $$\mathbf{x}[k+p|k] = \mathbf{0}\,.$$

Eqn. **(110**) can be satisfied if the moving horizon length, *p*, is chosen to be "large enough", or if the constraint in eqn. **(110**) is directly incorporated in the on-line optimization problem. In either case, a closed-loop stability proof can be constructed as follows.

5.1.2.1 A prototypical stability proof for MPC with nonlinear model

As in the linear case, we assume perfect knowledge of *f*, full state information x[*k*], and absence of disturbances. The constraints are assumed to define non-empty (convex) regions containing the point (0, 0). Assume also that *X*[*k*+*i*] and *U*[*k*+*i*] are independent of *k* and *i*. Assume that there exists a solution

(111) $$\mathbf{U}_{opt\,k|k}^{\;k+p-1|k}] = \{\mathbf{u}_{opt}[k|k], \quad \cdots, \quad \mathbf{u}_{opt}[k+p-1|k]\}$$

to eqn. (101) at time .*k*, corresponding to $\mathbf{x}[p] = \mathbf{0}$. Consider the following candidate for control input at time *k*+1:

(112) $$\mathbf{U}_{k+1|k+1}^{k+p|k+1} = \{\mathbf{u}_{opt}[k+1|k], \cdots, \mathbf{u}_{opt}[k+p-1|k], \mathbf{0}\}.$$

$\mathbf{U}_{k+1|k+1}^{k+p|k+1}$ is feasible at time *k*+1, because it contains inputs u that satisfied the same constraints at time *k*, the point (0, 0) has been assumed to be feasible, and *f* is assumed to be known perfectly. The above feasible input results in a value of the objective function *I*[*k*+1] that satisfies

(113) $$\begin{aligned} I_{opt}[k+1] &\le I[k+1] \\ &= I_{opt}[k] - L(\mathbf{x}[k],\mathbf{u}[k]) \\ &\le I_{opt}[k] \end{aligned}$$

where the last inequality results from the positive semi-definiteness of *L*. Therefore, the sequence $\{I[k]\}_{k=k_0}^{\infty}$ is non-increasing. It is also bounded from below by 0. Consequently, the sequence $\{I[k]\}_{k=k_0}^{\infty}$ converges to a limit *b*. To show that *b* = 0, rearrange eqn. (113), to get

$$0 \le L(\mathbf{x}[k],\mathbf{u}[k]) \le I_{opt}[k] - I_{opt}[k+1] \Rightarrow$$
$$0 \le \lim_{k\to\infty} L(\mathbf{x}[k],\mathbf{u}[k]) \le \lim_{k\to\infty}(I_{opt}[k] - I_{opt}[k+1]) = 0 \Rightarrow$$
$$\lim_{k\to\infty} L(\mathbf{x}[k],\mathbf{u}[k]) = 0 \Rightarrow$$
$$\lim_{k\to\infty}(\mathbf{x}[k],\mathbf{u}[k]) = (\mathbf{0},\mathbf{0})$$

where the last equality follows from property **(109**) of the function *L*. This completes the proof of closed-loop stability.

5.1.3 The stability proof and MPC practice

The above prototypical stability proofs are relatively simple. They rely on the inequality

(114) $$0 \le \left(L(\mathbf{x}[k],\mathbf{u}[k]) \le\right) I_{opt}[k] - I_{opt}[k+1]$$

Proving the above inequality hinges on a number of assumptions, of which the following are important:

i. The on-line optimization problem, eqns. (92) or (101), is feasible.
ii. The state can be steered to the setpoint in at most *p* steps, i.e. $\mathbf{x}[p] = \mathbf{0}$.
iii. The controlled process model **A** and **B** in eqn. (91) or **f** in eqn. (100) is perfect.
iv. There are no external disturbances.

v. The state **x** is measurable.
vi. The input and state constraints, eqns. (94) and (95) or (103) and (104) are time-independent.
vii. The global optimum of the on-line optimization problem, including the terminal constraint $\mathbf{x}[k+p|k]=\mathbf{0}$ can be computed exactly.

For stable processes, MPC practitioners have traditionally ensured that the above assumptions i and ii and are satisfied by

(a) selecting large enough p and
(b) performing the optimization with respect to $\mathbf{u}[k|k],\ldots, \mathbf{u}[k+m|k]$, where $m<<p$.

Rawlings and Muske (1993) have shown that the above idea can be extended to unstable processes. In addition to guaranteeing stability, their approach provides a computationally efficient way for on-line implementation. Their idea is to start with a finite control (decision) horizon but an infinite prediction (objective function) horizon, i.e., $m<\infty$ and $p=\infty$, and then use the principle of optimality and results from optimal control theory to substitute the infinite prediction horizon objective by a finite prediction horizon objective plus a terminal penalty term of the form

(115) $$\mathbf{x}[p]^T \mathbf{P}\mathbf{x}[p]$$

corresponding to the optimal value of the truncated part of the original objective function. Chen and Allgöwer (1996) have presented an extension of the above idea to MPC with nonlinear model and input constraints. They compute the terminal penalty term off-line as the solution of an appropriate Lyapunov equation. Genceli and Nikolaou (1995) have shown how to ensure feasibility and subsequently ensure robust stability for nonlinear MPC with Volterra models.

Selecting large enough p is not the only way to guarantee the above two assumptions i and ii. One could directly include an end-constraint $\mathbf{x}[p]=\mathbf{0}$ in the on-line optimization problem, an idea proposed by several investigators (Kleinman, 1970; Thomas, 1975; Keerthi and Gilbert, 1988; Mayne and Michalska, 1990). This constraint does not pose any serious computational challenges in on-line implementation. Other options are also possible, based, for example, on constraining $\mathbf{x}(p)$ to belong to a small neighborhood of the set point (Mayne, 1996) or on state contraction arguments (Morari and de Oliveira, 1997; Mayne, 1997).

Unstable processes pose the additional challenge that stabilization is possible only if the state $\mathbf{x}(k)$ lies in a certain domain, so that, even though the input may be constrained (eqn.(5)) enough control action can be available. If the state is not in the stabilizability domain, then nothing can be done to steer the state to the setpoint.

The feasibility of state constraints is a common issue (see, e.g., Theorem 1 in Rawlings et al., 1994). For example, when simple output constraints have to be satisfied, such as in eqn.(7), then it might occur that not enough control action is available, because of constraints such as in eqn. (5). If such infeasibility is detected, one can use additional relaxation variables to modify output constraints as

(116) $$y_{\max}+\varepsilon \geq y[k+i|k] \geq y_{\min}-\varepsilon,\ i=1,\cdots,p$$

and add a term $q\varepsilon^2$ to the objective function in eqn. (4).

The stability proof can be extended to handle bounded external disturbances (to address assumption iv) by additional "book-keeping", although the results may be conservative. Alternatively, one may introduce an integrator in the process output and show stability for an integrating system without disturbance[3].

The above issues and their implications for improving MPC are discussed in Section 6.

5.2 Robust stability and fragility of constrained MPC

To show inequality (114), the preceding assumptions i through vii were made. When assumptions i, iv, and v are not satisfied, *robustness* issues arise, because the process behaves differently than assumed by the controller. When assumption vii is not satisfied, then *fragility* issues arise, because the controller behaves differently than designed. It should be mentioned that the issue of fragility is not confined to constrained MPC systems. Keel and Bhattacharyya (1997) recently showed that even in linear time-invariant control systems, there are controllers for which extremely small deviations of the controller parameter values from their designed values can result in instability (see section 4.6).

[3] For example, the FIR model of eqn. (10) can be substituted by $y[k]=y[k-1]+\sum_{j=1}^{n} h_i u[k-j]$, thereby eliminating the step disturbance d.

5.2.1 Robust stability

To ensure robustness of stability in the presence of model uncertainty, one has to make sure that the inequality (114) is satisfied, even though the matrices **A** and **B** in eqn. (91) or the function **f** in eqn. (100) are not perfectly known. The inequality (114) also has to be satisfied when external disturbances enter the system. Two approaches have appeared in literature.

(a) Choose the parameters of the standard MPC on-line objective function $I[k]$ (e.g. the matrices **Q** and **R** in $J[k]$ for the linear case) in such a way, that $0 \le I_{opt}[k] - I_{opt}[k+1]$. In other words, robustness is attempted through better tuning of MPC, with minor modifications of the MPC structure.

(b) Modify the on-line MPC optimization problem, by adding constraints that help stabilize the closed-loop system, without critical dependence on tuning. For example, add to the on-line MPC optimization problem an additional constraint of the type

(117)
$$0 \le I_{opt}[k-1] - I[k]\,.$$

The advantage of the first approach is that the form of the standard MPC optimization problem is retained. Its disadvantage is that the on-line objective may become unrealistically conservative, depending on the magnitude of process model uncertainty. The advantage of the second approach is that the on-line objective may be formulated in a way that reflects the true control objective, without regard to stability, because the latter is enforced by the constraint of eqn. (117). Its disadvantage is that the MPC on-line optimization problem may become complicated.

5.2.1.1 MPC tuning for robust stability

Consider the following MPC on-line optimization problem for an open-loop stable, multi-input-multi-output (MIMO) system, modeled by a finite-impulse-response (FIR) model:

(118)
$$\min_{\varepsilon[k+1|k],\cdots,\varepsilon[k+n_w|k],\Delta\mathbf{u}[k|k],\cdots,\Delta\mathbf{u}[k+m|k]} J[k]$$

where

(119)
$$J[k] \triangleq \sum_{j=1}^{n_o} v_j \sum_{i=1}^{p} \left(y_j[k+i|k] - y_j^{SP} \right)^2 + \sum_{j=1}^{n_o} w_j \sum_{i=1}^{n_w} \varepsilon_j[k+i|k]^2 + \sum_{j=1}^{n_i}\sum_{i=0}^{m} r_{i,j} \Delta u_j[k+i|k]^2$$

subject to

Process output prediction

(120)
$$\mathbf{y}[k+i|k] = \sum_{j=1}^{N} \mathbf{G}^{[j]}\mathbf{u}[k+i-j|k] + \mathbf{d}[k+i|k], \quad i = 1,\cdots,p$$

Disturbance prediction

(121)
$$\mathbf{d}[k+i|k] = \mathbf{y}[k] - \sum_{j=1}^{N} \mathbf{G}^{[j]}\mathbf{u}[k-j], \quad i = 1,\cdots,p$$

Input move constraints

(122)
$$-\Delta\mathbf{u}_{\max} \le \Delta\mathbf{u}[k+i|k] \le \Delta\mathbf{u}_{\max}, \quad i = 0,\cdots,m$$

Input constraints

(123)
$$\mathbf{u}_{\min} \le \mathbf{u}[k+i|k] \le \mathbf{u}_{\max}, \quad i = 0,\cdots,m$$

Softened output constraints

(124)
$$\mathbf{y}_{\min} - \varepsilon[k+i|k] \le \mathbf{y}[k+i|k] \le \mathbf{y}_{\max} + \varepsilon[k+i|k], \quad i = 1,\cdots,n_w$$

End constraints

(125)
$$\mathbf{u}[k+m+i|k] = \left(\sum_{j=1}^{N} \mathbf{G}^{[j]} \right)^{-1} \left(\mathbf{y}^{SP} - \mathbf{d}[k+m+i|k] \right), \quad i \ge 0$$

where n_I is the number of process inputs; n_o is the number of process outputs; n_w is the number of inputs, is the number of time steps over which output constraints are enforced; $\mathbf{G}^{[j]}$ are matrices of the FIR coefficients of the process model. The real process output is assumed to be

(126)
$$\mathbf{y}[k] = \sum_{j=1}^{N} \mathbf{H}^{[j]}\mathbf{u}[k-j] + \mathbf{d}[k]\,.$$

Notice that the model kernel $\{\mathbf{G}^{[j]}\}_{j=1}^{N}$ is different from the true kernel $\{\mathbf{H}^{[j]}\}_{j=1}^{N}$, with the modeling error bounded as

(127) $$\left|\mathbf{H}^{[j]} - \mathbf{G}^{[j]}\right| \le \mathbf{E}_{\max}^{[j]} .$$

External disturbances are assumed to be bounded as

(128) $$\mathbf{d}_{\min} \le \mathbf{d}[k] \le \mathbf{d}_{\max}$$

and

(129) $$-\Delta\mathbf{d}_{\max} \le \Delta\mathbf{d}[k] \le \Delta\mathbf{d}_{\max}$$

where

(130) $$\Delta\mathbf{d}_{\max} \begin{cases} \ge 0, & k \le M \\ = 0 & k > M \end{cases}$$

For the above MPC system, eqns. (118) through (125), Vuthandam et al. (1995) developed sufficient conditions for robust stability with zero offset. These conditions can be used directly for calculation of minimum values for the prediction and control horizon lengths, p and m, respectively, as well as for the move suppression coefficients r_{ji}, which are not equal over the finite control horizon. Since the robust stability conditions are sufficient, they are conservative, particularly for very large modeling uncertainty bounds.

The proof relies on selecting p, m, and r_{ji} to satisfy the inequality

(131) $$0 \le J'_{opt}[k] - J'_{opt}[k+1]$$

with

(132) $$\begin{aligned} J'[k] \mathrel{\hat{=}} & \sum_{j=1}^{n_o} v_j \left(y_j[k] - y_j^{SP}\right)^2 + \\ & + \sum_{j=1}^{n_o} v_j \sum_{i=1}^{p} \left(y_j[k+i|k] - y_j^{SP}\right)^2 + \\ & + \sum_{j=1}^{n_o} w_j \sum_{i=1}^{n_w} \varepsilon_j[k+i|k]^2 + \\ & + \sum_{j=1}^{n_i} \sum_{i=-N+1}^{m} r_{i,j} \Delta u_j[k+i|k]^2 + \\ & + f[k] \end{aligned}$$

where the function $f[k]$ is an auxiliary function that helps prove stability, as required by eqn. (136). Satisfaction of inequality **(131**) implies that the sequence $\{J'_{opt}[k]\}_{k=k_0}^{\infty}$ is convergent. Then, the end-condition, eqn. (125) is used to show that the sequence $\{J'_{opt}[k]\}_{k=k_0}^{\infty}$ converges to 0. The proof starts with the inequality

(133) $$0 \le J'[k+1] - J'_{opt}[k+1] \Leftrightarrow J'_{opt}[k] - J'[k+1] \le J'_{opt}[k] - J'_{opt}[k+1]$$

where $J'[k+1]$ corresponds to a set of input values $\{\Delta\mathbf{u}[k+1|k+1], \cdots, \Delta\mathbf{u}[k+m|k+1], \Delta\mathbf{u}[k+m+1|k+1]\}$ to be selected such that

(134) $$0 \le J'_{opt}[k] - J'[k+1]$$

For the feasible input set

(135) $$\begin{aligned} \Delta\mathbf{u}[k+1|k+1] &= \Delta\mathbf{u}_{opt}[k+1|k] \\ & \cdots \\ \Delta\mathbf{u}[k+m|k+1] &= \Delta\mathbf{u}_{opt}[k+m|k] \\ \Delta\mathbf{u}[k+m+1|k+1] &= \text{to be determined for feasibility} \end{aligned}$$

inequality (134) becomes

(136)
$$
\begin{aligned}
0 \le & \sum_{j=1}^{n_o} v_j \left(y_j[k] - y_j^{SP} \right)^2 - \\
& - \sum_{j=1}^{n_o} v_j \left(y_j[k+1+p|k+1] - y_j^{SP} \right)^2 + \\
& + \sum_{j=1}^{n_o} v_j \left[\left(y_{j,opt}[k+1|k] - y_j^{SP} \right)^2 - \left(y_j[k+1] - y_j^{SP} \right)^2 \right] + \\
& + \sum_{j=1}^{n_o} v_j \sum_{i=2}^{p} \left[\left(y_{j,opt}[k+i|k] - y_j^{SP} \right)^2 - \left(y_j[k+i|k+1] - y_j^{SP} \right)^2 \right] + \\
& + \sum_{j=1}^{n_o} w_j \sum_{i=1}^{n_w} \left[\varepsilon_{j,opt}[k+i|k]^2 - \varepsilon_j[k+i|k+1]^2 \right] + \\
& + \sum_{j=1}^{n_i} \sum_{i=-N+1}^{m} (r_{i,j} - r_{i-1,j}) \Delta u_{j,opt}[k+i|k]^2 \\
& + f[k] - f[k+1]
\end{aligned}
$$

All terms in the above inequality, after lengthy manipulations and strengthening of inequalities, can be expressed in terms of inputs Δu squared. The resulting expression is of the form

(137)
$$0 \le \sum_{j=1}^{n_i} \sum_{i=-N+1}^{m} (r_{i,j} - r_{i-1,j} + a_{i,j}) \Delta u_{j,opt}[k+i|k]^2$$

where the positive constants $a_{i,j}$ depend on the model, uncertainty bounds and input bounds. For that inequality to be true it is sufficient to have

(138)
$$r_{i-1,j} \le r_{i,j} + a_{i,j}$$

with

(139)
$$r_{-N,j} = 0$$

Note that the above inequality in eqn. (138) implies that weights of the input move suppression term containing Δu gradually increase. Details can be found in Vuthandam et al. (1995) and Genceli (1993). A similar result can be found in Genceli and Nikolaou (1993) for MPC with l_1-norm based on-line objective. Variations for various MPC formulations have also been presented. Zheng and Morari (1993) and Lee and Yu (1997) have presented results on MPC formulations employing on-line optimization of the form

(140)
$$\min_{\mathbf{u}} \max_{\mathbf{p}} J[k, \mathbf{u}, \mathbf{p}]$$

where the vector **p** refers to process model parameters that are uncertain. The idea of eqn. (140) is that the optimal input for the worst possible process model is computed at each time step *k*.

5.2.1.2 Modifying the MPC algorithm for robust stability

Stability constrained MPC

Instead of tuning the MPC algorithm for robust stability, an inequality of the type (117) can be added to the on-line optimization, to ensure that the MPC controller automatically belongs to a set of stabilizing controllers. Through this modification, the designer can concentrate on formulating an on-line optimization objective that reflects the intended control objective.

Cheng and Krogh (1996) developed that idea in what they call stability constrained receding horizon control. While their results are for unconstrained MPC, extension of their algorithm to constrained MPC is trivial. Following their work, consider a plant in controllable form

(141)
$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}u[k]$$

where

(142)
$$\mathbf{A} \triangleq \begin{bmatrix} \mathbf{0}_{(N-M)\times M} & \mathbf{I}_{(N-M)} \\ \mathbf{A}_1 & \mathbf{A}_2 \end{bmatrix}$$

(143) $$\mathbf{B} \mathrel{\hat{=}} \begin{bmatrix} \mathbf{0}_{(N-M)\times M} \\ \mathbf{B}_2 \end{bmatrix}$$

(144) $$\mathbf{x}[k] \mathrel{\hat{=}} \begin{bmatrix} \mathbf{x}^{(1)}[k] \\ \mathbf{x}^{(2)}[k] \end{bmatrix}$$

with $\mathbf{B}_2 \in \Re^{M\times M}$ invertible, $\mathbf{x}^{(1)}[k] \in \Re^M$, $\mathbf{x}^{(2)}[k] \in \Re^{N-M}$. The stability constrained receding horizon control algorithm is given by the following steps.

At time step k, minimize an objective function

(145) $$H[k]$$

over a finite horizon of length p, subject to

(146) $$\begin{aligned} &\mathbf{x}[k+i+1|k] = \mathbf{A}\mathbf{x}[k+i|k] + \mathbf{B}u[k+i|k],\ i = 0,\cdots,p-1 \\ &\mathbf{x}[k|k] = \mathbf{x}[k] \end{aligned}$$

(147) $$\boxed{\begin{aligned} &\|\mathbf{x}[k+i+1|k]\|^2 \le (1-\sigma_k)\hat{l}_k,\ i = 0,\cdots,p-1 \\ &\sigma_k \ge \varepsilon_k \end{aligned}}$$ **(*stability constraint*)**

(148) $$\begin{aligned} &\hat{l}_k = \max\{l_k, \|\mathbf{x}[k]\|^2\} \\ &l_k = \begin{cases} arbitrary,\ k = 0 \\ \max\{\|x[k-1+i|k-1]\|^2,\ i = 1,\cdots,p\},\ k \ge 1 \end{cases} \\ &\varepsilon_k = \frac{\|\mathbf{x}^{(1)}[k]\|^2}{c\hat{l}_k} \\ &c \ge 1 \end{aligned}$$

Cheng and Krogh (1996) give a stability proof for the above algorithm. They extend their algorithm to include state estimation in Cheng and Krogh (1997).

Robust-stability-constrained MPC

Badgwell (1997) has taken the idea of stability constrained MPC a step further, by developing an MPC formulation in which a constraint used in the on-line optimization problem guarantees robust stability of closed-loop MPC for stable linear processes. The trick, again, is to make sure that an inequality of the type (117) is satisfied for all possible models that describe the controlled process behavior. The set of these models is assumed to be known during controller design.

Following Badgwell (1997), consider that the real process behavior is described by the stable state-space equations

(149) $$\mathbf{x}[k+1] = \overline{\mathbf{A}}\mathbf{x}[k] + \overline{\mathbf{B}}\mathbf{u}[k]$$

where $\mathbf{x}[k] \in \Re^n$, $\mathbf{u}[k] \in \Re^M$, and the process parameters $\overline{\theta} \mathrel{\hat{=}} (\overline{\mathbf{A}}, \overline{\mathbf{B}})$ are not known exactly, but are known to belong to a set

(150) $$\Omega \mathrel{\hat{=}} \{\theta_1,\cdots,\theta_{p_m}\} = \{(\mathbf{A}_1,\mathbf{B}_1),\cdots,(\mathbf{A}_{p_m},\mathbf{B}_{p_m})\}$$

of p_m distinct models. A nominal model

(151) $$\tilde{\theta} = (\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$$

is used. For robust asymptotic stability, the state should be driven to the origin, while satisfying input, input move, and state constraints. No external disturbances are considered. Under the above assumptions, the robust-stability-constrained MPC algorithm is as follows:

(152) $$\min_{\varepsilon[k+1|k],\cdots,\varepsilon[k+n_w|k],\mathbf{u}[k|k],\cdots,\mathbf{u}[k+m|k]} \Gamma(\tilde{\mathbf{x}}[k], \{\mathbf{u}[k+i-1|k]\}_{i=1}^{i=\infty}, \tilde{\theta}, \{\varepsilon[k+i-1|k]\}_{i=1}^{i=n_w}, \mathbf{T}_k)$$

where

(153)

$$\Gamma(\tilde{\mathbf{x}}[k],\{\mathbf{u}[k+i-1|k]\}_{i=1}^{i=\infty},\tilde{\theta},\{\varepsilon[k+i-1|k]\}_{i=1}^{i=n_w},\mathbf{T}_k)\ \hat{=}$$
$$\sum_{i=1}^{\infty}\tilde{\mathbf{x}}[k+i|k]^T\,\mathbf{W}\tilde{\mathbf{x}}[k+i|k]+\sum_{i=1}^{\infty}\mathbf{u}[k+i-1|k]^T\,\mathbf{R}\mathbf{u}[k+i-1|k]+\left(\mathbf{e}_{k+1|k}^{k+n_w|k}\right)^T\mathbf{T}[k]\left(\mathbf{e}_{k+1|k}^{k+n_w|k}\right)$$

subject to

Process state prediction

(154)

$$\tilde{\mathbf{x}}[k+i|k]=\tilde{\mathbf{A}}\tilde{\mathbf{x}}[k+i-1|k]+\tilde{\mathbf{B}}\mathbf{u}[k+i-1|k],$$
$$\tilde{\mathbf{x}}[k|k]=\mathbf{x}[k]$$

Input move constraints

(155)

$$\Delta\mathbf{u}_{\min}\le\Delta\mathbf{u}[k+i|k]\le\Delta\mathbf{u}_{\max},\quad i=0,\cdots,m$$

Input constraints

(156)

$$\mathbf{u}_{\min}\le\mathbf{u}[k+i|k]\le\mathbf{u}_{\max},\quad i=0,\cdots,m-1$$
$$\mathbf{u}[k+i|k]=\mathbf{0},\quad i\ge m$$

Softened state constraints

(157)

$$\left.\begin{array}{l}\mathbf{x}_{\min}-\varepsilon[k+i|k]\le\mathbf{x}_{\ell}[k+i|k]\le\mathbf{x}_{\max}+\varepsilon[k+i|k],\quad \ell=0,\cdots,p_m\\ \varepsilon[k+i|k]\ge\varepsilon_{\min}>\mathbf{0}\end{array}\right\},\quad i=1,\cdots,n_w$$

Robust stability constraints

(158)

$$\boxed{\begin{array}{l}\Gamma(\mathbf{x}[k],\{\mathbf{u}[k+i-1|k]\}_{i=1}^{i=\infty},\theta_{\ell},\{\varepsilon[k+i-1|k]\}_{i=1}^{i=n_w},\mathbf{T}_k)\le\\ \Gamma(\mathbf{x}[k],\{\mathbf{u}_{opt}[k+i-1|k-1]\}_{i=1}^{i=\infty},\theta_{\ell},\{\varepsilon_{opt}[k+i-1|k-1]\}_{i=1}^{i=n_w},\mathbf{T}_k)\end{array}},\quad \ell=0,\cdots,p_m$$

where

(159)

$$\mathbf{x}_{\ell}[k+i|k]=\mathbf{A}_{\ell}\mathbf{x}_{\ell}[k+i-1|k]+\mathbf{B}_{\ell}\mathbf{u}[k+i-1|k],$$
$$\mathbf{x}_{\ell}[k|k]=\mathbf{x}[k]$$

(160)

$$\mathbf{T}[k]=\frac{\left(\mathbf{e}_{k|k-1}^{k+n_w-1|k-1}\right)_{opt}^T\mathbf{T}[k-1]\left(\mathbf{e}_{k|k-1}^{k+n_w-1|k-1}\right)_{opt}}{\hat{\mathbf{e}}^T\mathbf{T}[k-1]\hat{\mathbf{e}}}\mathbf{T}[k-1]$$
$$\hat{\mathbf{e}}=\arg\min\left\|\mathbf{e}-\left(\mathbf{e}_{k|k-1}^{k+n_w-1|k-1}\right)_{opt}\right\|$$

subject to

$$\mathbf{e}\ge\mathbf{e}_{\min}>\mathbf{0}$$
$$\mathbf{x}_{\min}-\mathbf{e}\le\mathbf{x}_{opt}[k+i|k-1]\le\mathbf{x}_{\max}+\mathbf{e},\quad i=1,\cdots,n_w$$

and $\Delta\mathbf{u}_{\min}<0<\Delta\mathbf{u}_{\max}$, $\mathbf{u}_{\min}<0<\Delta\mathbf{u}_{\max}$, $\mathbf{x}_{\min}<0<\mathbf{x}_{\max}$.

The overall optimization problem is convex and has a feasible solution, therefore it is guaranteed to have a unique optimal solution.

The model linearity assumption, eqn (154) is not critical. Badgwell (1997) has shown how the above ideas can be readily extended to the case of stable nonlinear plants.

A critical assumption in the above formulation is eqn. (150), which assumes that a set of distinct models captures modeling uncertainty. Ideally, one would like to have a continuum of models such that the real plant is one point in that continuum. The continuum could be approximated by considering a very large number of distinct models, with the obvious trade-off of increase in the dimensionality of the on-line optimization problem.

5.2.2 Fragility

The stability proofs developed in the previous sections implicitly assume that an exact solution of the MPC on-line optimization can be obtained. However, an exact solution may not always be obtained in cases such as the following:

- The on-line optimization problem is non-convex, therefore guarantees for reaching the global optimum may be hard to obtain.
- The on-line optimization problem is nonlinear and involves equality constraints. Satisfaction of those constraints is not exact, but approximate (within ε).

In the first of the above two cases, a local optimum may be obtained that is far from the global optimum. In such case, stability analysis based on attainment of global optimum would entirely break down. In the second case, if the on-line optimization problem is convex, then the solution found numerically would be close to the exact solution. It might then be concluded that stability analysis would be valid, at least for small error in the approximation of the exact MPC system by the one approximately (numerically) computed on-line, provided that continuity arguments would be valid. It turns out, however, that this is not necessarily true. Keel and Bhattacharyya (1997) have shown that there exist linear time-invariant fragile controllers, i.e. such that closed-loop stability is highly sensitive to variations in controller parameters. In that context, the fragility properties of MPC should be rigorously examined.

A number of authors (see, for example, Scokaert et al., 1998) have developed MPC variants and corresponding stability proofs, which overcome the above two problems by

(a) requiring that the on-line optimization reaches a *feasible* (sub-optimal) solution of a corresponding problem, and/or

(b) substituting equality constraints of the type

(161) $$\mathbf{f}(\mathbf{x}) = \mathbf{0}$$

by inequality constraints of the type

(162) $$-\delta \leq \mathbf{f}(\mathbf{x}) \leq \delta$$

where δ is a vector with "small" entries. This ensures that the end-constraint can be satisfied *exactly* and, consequently, stability analysis can be rigorously valid.

5.3 Performance and robust performance

Rigorous results for the performance of constrained MPC are lacking. However, there is a number of propositions on how the performance of MPC could be improved. Such propositions rely on

(a) modifying the structure of MPC for robust performance,

(b) tuning MPC for robust performance, and

(c) developing efficient algorithms for the numerical solution of the MPC on-line optimization problem, thus enabling the formulation of more complex and realistic on-line optimization problems that would in turn improve performance.

The expected results of these propositions are difficult to quantify. Nevertheless, the proposed ideas have intuitive appeal and appear to be promising.

One proposition is to formulate MPC in the *closed-loop* optimal feedback form (see section 2.2). The main challenge of this proposition is the difficulty of solving the on-line optimization problem. Kothare et al. (1996) propose a formulation that reduces the on-line optimization problem to semi-definite programming, which can be solved efficiently using interior point methods.

A second proposition relies on the idea that the on-line optimization problem is unconstrained after a certain time-step in the finite moving horizon. Where in the finite horizon that happens is determined by examining whether the state has entered a certain invariant set (Mayne, 1997). Once that happens, then closed-form expressions can be used for the objective function from that time point to the end of the optimization horizon, *p*. The idea is particularly useful for MPC with nonlinear models, for which the computational load of the on-line optimization is substantial. A related idea was presented by Rawlings and Muske (1993), where the on-line optimization problem has a finite control horizon length, *m*, and infinite prediction horizon length, *p*, but the objective function is truncated, because the result of the optimization is known after a certain time point.

Of course, as mentioned above, the mere development of more efficient optimization algorithms could indirectly improve performance. This could happen, for example, through the use of nonlinear instead of linear models in on-line optimization. As stated in the Introduction, the discussion of numerical efficiency issues is beyond the scope of this discussion.

A third proposition has been discussed in section 5.2.1.2. The idea is that by using a robust stability constraint (eqns. (147) or (158)), MPC will be stabilizing, therefore true performance objectives may translated into values for the tuning parameters of MPC, without worrying about potential instabilities resulting from poor tuning. However, that translation of performance objectives to values for MPC tuning parameters is not always straightforward.

A fourth proposition was discussed by Vuthandam et al. (1995). Their idea is that the values of the MPC tuning parameters must satisfy robust stability requirements. It turns out that for the robust stability requirements developed by the above authors, performance improves as the prediction horizon length, p, increases from its minimum value to larger values, but after a certain point performance deteriorates as p increases further. This happens because for very large p the input move terms in the on-line objective function must be penalized so much, that the controller becomes very sluggish and performance suffers. Results such as the above depend on the form of the robust stability conditions. If such conditions are only sufficient, as is the case with Vuthandam et al. (1995), then performance related results may be conservative.

6 How can theory help develop better MPC systems?

6.1 Conceptual unification and clarification

The theoretical developments of the last few years have provided a solid foundation for the analysis and synthesis of constrained MPC systems. We now have a tool, summarized perhaps by eqn. (100), by which we can study the stability (and other important properties) of constrained MPC and its variants in a framework that removes guessing or ambiguity and supports rigorous statements about the expected behavior of MPC. This framework is valuable for both researchers and practitioners of MPC.

Of course, as with every theory, practitioners cannot rely solely on theory when they design real MPC systems. Real-world problems have their idiosyncratic peculiarities that may vary from case to case. It is imperative that the MPC designer understand the entire engineering framework within which an MPC system is going to function. But theory could help predict what could or would happen as a result of a specific design, thereby steering the design towards promising directions and reducing reliance to unnecessary improvisation. That help is valuable for the novice and reassuring for the expert. Theory can help augment the designer’s intuition in a very productive way. For example, theory could augment the value or shorten the duration of extensive simulations or plant testing frequently performed before the final commissioning of an MPC system. As another example, the concept of best achievable performance by a constrained MPC system is mentioned. Similarly to the minimum-work concept in thermodynamics, control theory should provide achievable targets of performance and should do so under practical conditions, e.g., in the presence of inequality constraints and model inaccuracies (Morari, 1988). Chmielewski and Manousiouthakis (1996) have provided such a result by proving that the best performance achievable by an MPC systems with input and state inequality constraints and objective function over an infinite horizon can be determined by solving a single, finite dimensional, convex program of known size.

For researchers, the existence of a working theoretical framework provides impetus to attack MPC problems with confidence that was lacking before the seminal paper by Rawlings and Muske (1993). Despite the avalanche of recent results capitalizing on the use of eqn. (100), there appears to be significant room for extensions, improvements, and new ideas on MPC. Paralleling, in a way, the introduction and wide spread of other important frameworks in control theory (such as the universally recognized equations $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$), the MPC stability framework could have significant implications for developments in control theory for systems different from the chemical process systems for which MPC was first extensively implemented.

A factor whose importance for future developments cannot be overstated is the interplay between theory and practice. Abstraction of practical problems spurs theoretical research with industrial relevance, while consistent and systematic exploration of theoretical ideas (a painstaking process) can yield otherwise unimaginable or unexpected breakthroughs.

6.2 Improving MPC

The benefits of a framework for the rigorous study of MPC properties are not confined to the mere proof of MPC properties. More importantly, MPC theory can lead to discoveries by which MPC can be improved. In fact, the proofs of theoretical results frequently contain the seeds for substantial MPC improvements through new formulations. The existence of a theory that can be used to analyze fairly complex MPC systems allows researchers to propose high-performance new formulations whose properties can be rigorously analyzed, at least for working prototypes. The algorithmic complexity of such formulations might be high, but their functionality would also be high. It is reassuring to know that even when the analysis is not trivial, it is definitely feasible. In designing such systems the designer would have theory as an invaluable aid that could augment intuition. Moreover, theory could also provide guidelines for the efficient use of such systems by end-users, by helping them predict what the effects of tweaking would be. The algorithmic complexity of on-line optimization would be hidden from the end-user.

The ensuing discussion shows some open issues and recent new ideas in constrained MPC. The list is neither complete, nor time-invariant. Some of the following ideas were directly inspired by recent MPC theory. Others were developed rather independently, but knowledge of the fact that theory exists that can be used to study them makes those ideas more appealing from both a theoretical and practical viewpoint.

6.2.1 Process models

Before discussing issues related to the different classes of models that may be used with MPC, we should stress that the issue of model structure selection and model identification is at the heart of MPC. "The greatest contribution of Ziegler and Nichols (1942) is not the famous tuning rule but the 'cycling' identification technique: they understood precisely what minimum model information was necessary for tuning and developed a simple, reliable method for obtaining this information." (Morari, 1988).

6.2.1.1 Linear vs. nonlinear models

While it is well recognized that *all* physical systems are, in principle, nonlinear (section 2.2), the challenge is when a nonlinear approach is necessary for the solution of a problem. It is then important to answer questions such as

- When to use nonlinear models?
- For what processes?
- What is the structure of such models?
- What are the properties of MPC when particular models are used?
- How can such models be developed (from first principles and/or experimental data)?
- What is the cost/benefit of using nonlinear models in MPC?

6.2.1.2 Input-output vs. state-space models

Traditional MPC algorithms, such as the quadratic dynamic matrix control (QDMC) algorithm (García and Morshedi, 1986) relied on FIR models. While FIR models have several advantageous features (section 2.2), they are not essential for the characterization of a control algorithm as MPC. DARMAX or state-space models have their own relative advantages and can be used equally well in MPC. With state-space models the issue of state estimation naturally arises. A new class of state estimators, introduced by Robertson et al. (1996) is discussed in the following section.

6.2.1.3 Moving horizon-based state estimation for state-space models

If a state-space model is used by an MPC system, then the problem of state estimation arises, since, usually, not all states can be measured. State estimation within a stochastic system framework has a rich history, the turning point being the celebrated work of Kalman (1960). The Kalman filter has limitations when applied to nonlinear systems (as the extended Kalman filter). In addition, the Kalman filter cannot explicitly handle constraints on estimated states. Yet, state variables such as mass fraction or temperature are constrained, and corresponding constraints should be satisfied by estimates of these variables. Robertson et al. (1996) presented a new class of state estimators. Their approach to the constrained state estimation problem relies on the moving horizon concept and on-line optimization. States are estimated by considering data over a finite *past* moving horizon and by minimizing a square-error criterion. In fact, the approach introduced by the above authors can estimate both states and model parameters.

The system whose states and parameters are estimated is represented by the difference equations

(163)
$$\begin{bmatrix}\mathbf{x}[k+1]\\ \mathbf{p}[k+1]\end{bmatrix}=\begin{bmatrix}\mathbf{f}(\mathbf{x}[k],\mathbf{p}[k],\mathbf{u}[k])\\ \mathbf{p}[k]\end{bmatrix}+\begin{bmatrix}\mathbf{w}_x[k]\\ \mathbf{w}_p[k]\end{bmatrix}$$
$$\mathbf{y}[k]=\mathbf{g}(\mathbf{x}[k],\mathbf{p}[k])+\mathbf{v}[k]$$

where $\mathbf{w}_{x,}$, $\mathbf{w}_p$, and $\mathbf{v}$ are white noise vectors with zero mean; and the vector-valued vector function $\mathbf{f}$ represents the solution

(164)
$$\mathbf{x}[k+1]=\mathbf{x}[k]+\int_{t_k}^{t_{k+1}}\phi\big(\mathbf{x}(t),\mathbf{p}[k],\mathbf{u}[k]\big)\mathrm{d}t+\int_{t_k}^{t_{k+1}}\omega(t)\mathrm{d}t$$

of the differential equation $\mathrm{d}\mathbf{x}/\mathrm{d}t=\phi\big(\mathbf{x}(t),\mathbf{p}[k],\mathbf{u}[k]\big)+\omega(t)$. The least-squares estimate of the state

(165) $$\mathbf{z}[k] \triangleq \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{p}[k] \end{bmatrix}$$

and noise vectors

(166) $$\mathbf{w}[k] \triangleq \begin{bmatrix} \mathbf{w}_x[k] \\ \mathbf{w}_p[k] \end{bmatrix}$$

and $\mathbf{v}[k]$ at time point k is obtained by the following constrained minimization

(167) $$\min_{\mathbf{z}[k-m+1],\mathbf{v},\mathbf{w}} \left(\mathbf{z}[k-m+1]-\mathbf{z}[k-m+1|k-m]\right)^T \mathbf{P}[k-m+1|k-m]^{-1}\left(\mathbf{z}[k-m+1]-\mathbf{z}[k-m+1|k-m]\right)+ \sum_{i=k-m+1}^{k}\mathbf{v}[i]^T\mathbf{R}^{-1}\mathbf{v}[i]+\sum_{i=k-m+1}^{k}\mathbf{w}[i]^T\mathbf{Q}^{-1}\mathbf{w}[i]$$

subject to the equality constraints of eqn. (163) and the inequality constraints

(168) $$\begin{aligned} &\mathbf{v}_{\min} \le \mathbf{v}[i] \le \mathbf{v}_{\max}, \quad k-m+1 \le i \le k \\ &\mathbf{x}_{\min} \le \mathbf{x}[i] \le \mathbf{x}_{\max}, \quad k-m+1 \le i \le k \\ &\mathbf{p}_{\min} \le \mathbf{p}[i] \le \mathbf{p}_{\max}, \quad k-m+1 \le i \le k \\ &\mathbf{w}_{\min} \le \mathbf{w}[i] \le \mathbf{w}_{\max}, \quad k-m+1 \le i \le k-1 \end{aligned}$$

The matrices **P**, **Q**, and **R** can be interpreted as covariance matrices of corresponding random variables, for which probability density functions are normal subject to truncation of their tail ends, dictated by the constraints of eqn. (168).

The on-line optimization problem posed by eqn. (167) can be solved by nonlinear programming algorithms, if the model in eqn. (163) is nonlinear, or by standard QP algorithms, for a linear model. Note that the moving horizon keeps the size of the optimization problem fixed, by discarding one old measurement for each new one received. The effect of the initial estimate $\mathbf{z}[k-m+1|k-m]$ becomes negligible as the horizon length m increases. This is the duality counterpart of the MPC moving horizon requirement that the state should reach a desired value at the end of the moving prediction horizon. In fact the duality between the above state estimation approach and MPC parallels the duality between LQR and Kalman filtering.

As in the case of MPC, the performance of the proposed approach depends on the accuracy of the model used.

6.2.1.4 MPCI: Expanding the MPC/on-line optimization paradigm to adaptive control

To maintain the closed-loop performance of an MPC system, it may become necessary to update the process model originally developed off-line. Control objectives and constraints most often dictate that this update has to take effect while the process remains under MPC (Cutler, 1995; Qin and Badgwell, 1997). This task is known as *closed-loop identification*. Frequently occurring scenarios where closed-loop identification is desired include the following:

- Because of equipment wear, a modified process model is needed (without shutting down the process), for tight future control.
- The process has to operate in a new regime where an accurate process model is not available, yet the cost of off-line identification experiments for the development of such a model is prohibitively high, thus making closed-loop identification necessary.
- Process identification is conducted off-line, but environmental, safety, and quality constraints still have to be satisfied.

Closed-loop identification has been addressed extensively in a linear stochastic control setting (Åström and Wittenmark, 1989). Good discussions of early results from a stochastic control viewpoint are presented by Box (1976) and Gustavsson et al. (1977). Landau and Karimi (1997) provide an evaluation of recursive algorithms for closed-loop identification. Van den Hof and Schrama (1994), Gevers (1993), and Bayard and Mettler (1993) review recent research on new criteria for closed-loop identification of state space or input-output models for control purposes.

The main challenge of closed-loop identification is that feedback control leads to quiescent process behavior and poor conditions for process identification, because the process is not excited (see, for example, Radenkovic and Ydstie (1995) and references therein). Traditional methods for excitation of a process, (Söderström et al., 1975; Fu and Sastry, 1991; Klauw et al., 1994; Ljung, 1987; 1993; Schrama, 1992) under closed-loop control through the addition of *external* dithering signals to the process input or setpoint have the weaknesses that controller performance is adversely affected in a way that may be difficult to predict, because it depends on the very process being identified under closed-loop control. To remedy these problems, Genceli and Nikolaou (1996) introduced the *simultaneous Model Predictive Control and Identification* (MPCI) paradigm. MPCI relies on *on-line optimization over a finite future horizon* (Figure 20). Its main difference from standard MPC is that MPCI employs the well known *persistent excitation* (PE) condition (Goodwin and Sin, 1984) to create additional constraints on the process inputs in the following kind of on-line optimization problem, solved at each sampling instant *k*:

(169) $$\underset{\substack{\text{process input values}\\ \text{over control horizon}}}{\text{minimize}} \quad [\textit{control objective over optimization horizon}]$$

subject to

(170) *Standard MPC constraints*

(171) *Persistent excitation constraints on inputs over finite horizon*.

The form of the PE constraint depends only on the model structure considered by the identifier and is independent of the behavior of the identified plant. Of course, the model structure should be "close" (but not necessarily contain) the real plant structure.

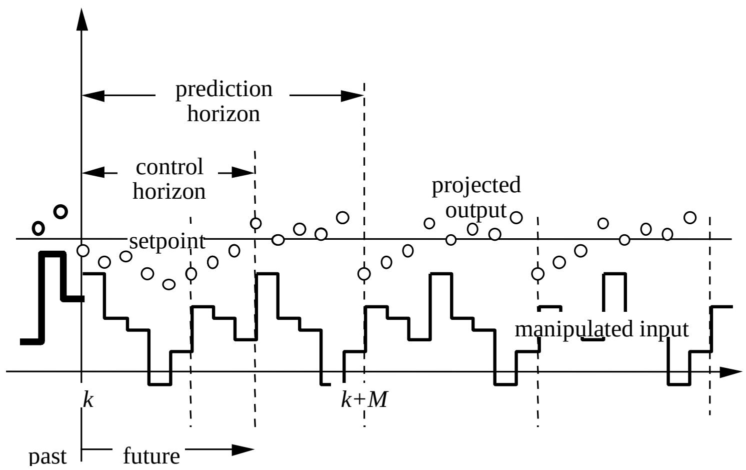


Figure 20. The MPCI moving horizon. Notice the unsettling projected plant output and the periodicity of the manipulated input.

The above formulation defines a new class of adaptive controllers. By placing the computational load on the computer-based controller that will perform on-line optimization, MPCI greatly simplifies the issue of closed-loop model parameter convergence. In addition, constraints are explicitly incorporated in the MPCI on-line optimization. By contrast, most of the existing adaptive control theory requires the controller designer to make demanding assumptions that are frequently difficult to assert.

To explain MPCI quantitatively, consider, for simplicity, a single-input-single-output (SISO) process modeled as

(172) $$y[k] = \sum_{i=1}^{m} a_i u[k-i] + \sum_{i=1}^{n} b_i y[k-i] + d[k] =$$

(173) $$= \phi[k-1]^T \theta + w[k]$$

where $y[k]$ is the process output; $u[k]$ is the process input; $d[k]$ is a constant disturbance, d, plus white noise with zero mean, $w[k]$;

(174) $$\theta \hat{=} [a_1 \cdots a_m b_1 \cdots b_n d]^T$$

is the parameter vector to be identified; and

(175) $$\phi[k-1]^T = [u[k-1] \cdots u[k-m]\ y[k-1] \cdots y\,[k-n]\ 1]$$

Using the strong PE condition and eqns.(169) to (171), one can formulate an MPCI on-line optimization problem at time *k* as follows.

(176) $$\min_{u[k|k],\cdots,u[k+M-1|k],\mu,\varepsilon,\rho} \sum_{i=1}^{M}\left[\omega_i \left(y[k+i|k] - y^{sp}\right)^2 + r_i \Delta u[k+i-1|k]^2\right] + q_1 \mu^2 + q_2 \varepsilon^2 - q_3 \rho \quad \textbf{(Objective)}$$

subject to

(177)l $$u_{\max} \geq u[k+i-1|k] \geq u_{\min}\ , \quad i=1,2,\cdots,M$$ **(Input constraint)**

(178) $$\Delta u_{\max} \geq \Delta u[k+i-1|k] \geq -\Delta u_{\max}\ , \quad i=1,2,\cdots,M$$ **(Input move constraint)**

(179) $$y_{\max} + \varepsilon \geq y[y+i|k] \geq y_{\min} - \underbrace{\varepsilon}_{\substack{\text{output}\\\text{constraint}\\\text{softening}\\\text{variable}}}\ , \quad i=1,2,\cdots,M$$ **(Output constraint)**

(180) $$y(k+i) = \phi(k+i-1)^T\,\overline{\theta}(k)\ , \quad i=1,2,\cdots,M$$ **(Future output prediction)**

(181) $$\overline{\theta}[k] = \left(\sum_{j=1}^{s}\phi[k-j]\phi[k-j]^T\right)^{-1}\left[\phi[k-1] \quad \cdots \quad \phi[k-s]\right]\mathbf{y}[k]$$ **(Parameter estimate)**

(182) $$\begin{aligned}&\sum_{j=1}^{s}\phi[k+i-j|k]\phi[k+i-j|k]^T \succeq (\rho-\mu)\mathbf{I} \succ \mathbf{0}\\ &\underbrace{\rho}_{\substack{\text{PE}\\\text{level}}} > \underbrace{\mu}_{\substack{\text{PE}\\\text{softening}\\\text{variable}}} \geq 0\end{aligned} \qquad i=1,2,\cdots,M$$ **(PE constraints)**

where

(183) $$\mathbf{y}[k] \mathrel{\hat{=}} [y[k]\cdots y[k-s+1]]^T$$

(184) $$\phi[k-j-1]^T \mathrel{\hat{=}} \left[u[k-j-1]\cdots u[k-j-m]y[k-j-1]\cdots y[k-j-n] \quad 1\right]$$

with all past values of inputs u and outputs y assumed to be known. Note that eqn. (182) for i=1 ensures that the *closed-loop* input u is persistently exciting. In typical MPC fashion, the above optimization problem is solved at time k, and the optimal $u(k)$ is applied to the process. This procedure is repeated at subsequent times k+1, k+2, etc. For the numerical solution of the MPCI on-line optimization problem, Genceli and Nikolaou (1996) have developed a successive semi-definite programming algorithm, with proven convergence to a local optimum.

6.2.2 Objective

6.2.2.1 Multi-scale MPC

Perhaps the most compelling impetus behind computer integration of process operations is the opportunity to closely coordinate (*integrate*) a range of individual activities, in order to achieve overall corporate objectives. In optimization jargon, the ultimate target of computer integration is to relate dispersed individual activities to an overall corporate objective function, that could, in turn, be used in optimal decision making over time. So far, the development of a manageable all-inclusive objective function has been practically beyond reach, due to the enormous complexity of the problem. As a remedy, hierarchical decomposition of the problem and optimal decision making at each level are employed. This decomposition is usually realized according to the hierarchical structure of Figure 21 (Bassett et al., 1994). Note that early applications of computer-based on-line optimization worked at the top levels of the process operations hierarchy, where decisions are made less frequently that at lower levels and, consequently, the limited speed, input/output and storage capacity of early computers was not an impediment.

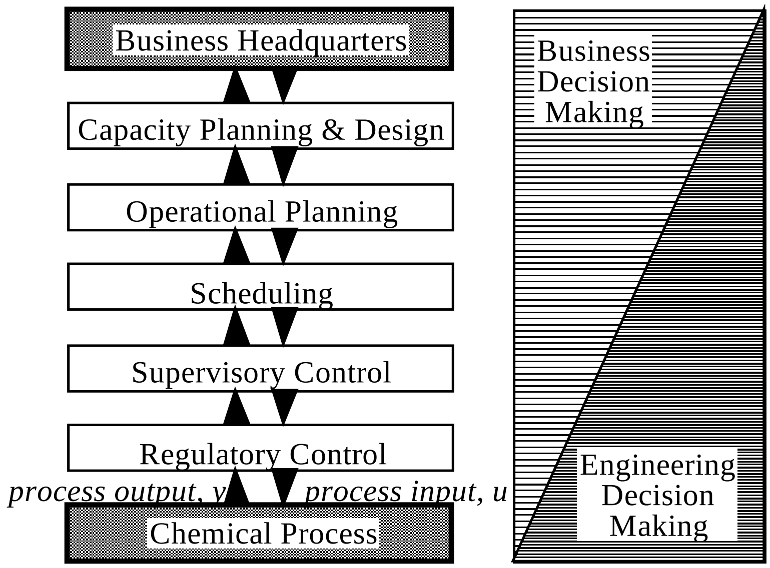


Figure 21. Process Operations Hierarchy in the chemical process industries.

The implicit assumption in the above decomposition is that the aggregate of the individually optimal decisions will be close to the overall optimal decision at each point in time. Frequently, this is not the case. Therefore, there exists a strong incentive to establish a framework for the formulation and solution of optimization problems that integrate as many levels as possible above the chemical process level of the Process Operations Hierarchy (Prett and García, 1988; Kantor et al., 1997).

Why is it not trivial to perform an integrated optimization, by merely combining the individual optimization problems at each level of the Process Operations Hierarchy to form a single optimization problem? There is a number of reasons, summarized below:

- *Dimensionality* – Each level in Fig. 1 is associated with a different time-scale (over which decisions are made) that can range from split-seconds, at the Regulatory Control level, to years, at the Capacity Planning & Design level. The mere combination of individual level optimization problems into one big problem that would span all time scales would render the dimensionality of the latter unmanageable.
- *Engineering/Business concepts* – While engineering considerations dominate the lower levels of the Process Operations Hierarchy, business concepts emerge at the higher levels. Therefore, a variety of individual objectives of different nature emerge that are not trivial to combine, either at the conceptual or the implementation level.
- *Optimization paradigms* – Various optimization paradigms have found application at each level (e.g., stochastic programming, mixed integer-nonlinear programming, quadratic programming, linear programming, etc.). However, it is not obvious what would be a promising paradigm for the overall optimization problem.
- *Software implementation* – The complexity of the integrated optimization problem is exacerbated when implementation issues are considered. A unifying framework is needed that will allow both software and humans involved with various levels of the Process Operations Hierarchy to seamlessly communicate with one another in a decision making process over time.

The above reasons that make the overall problem difficult suggest that a concerted attack is needed, from both the engineering and business ends of the problem. It should be stressed that, while there may be some common mathematical tools used in both engineering and business, the bottleneck in computer integration of process operations is not the lack of *solution to a given mathematical problem,* but rather the need for the *formulation of a mathematical problem that both corresponds to physical reality and is amenable to solution*..

Stephanopoulos et al. (1997) have recently used a wavelet-transform based formalism to develop process models at multiple scales and use them in MPC. That formalism hinges on using transfer functions that localize both time and scale, unlike standard (Laplace or z-domain) transfer functions, which localize scale (frequency), or standard difference or differential equation models, which localize time. Based on this formalism, the above authors address MPC related issues such as simulation of linear systems, optimal control, state estimation, optimal fusion of measurements, closed-loop stability, constraint satisfaction, and horizon length determination. In relation to the last task, Michalska and Mayne (1993) have proposed a variable horizon algorithm for MPC with nonlinear models. Their approach addresses the difficulty of global optimum requirements in stability proofs. The moving horizon length, p, is a decision variable of the on-line optimization. Closed-loop stability is established by arguments such as eqn. (113).

6.2.2.2 Dynamic programming (*closed-loop* optimal feedback)

As discussed in sections 2.2 and 5.3, the main reason for not implementing the closed-loop optimal feedback MPC form is the difficulty of the associated optimization problem. If inequality constraints are not present, then an explicit closed-form controller can be determined, as Lee and Cooley (1995) have shown. This is an area where significant developments can be expected.

6.2.3 Constraints

6.2.3.1 MPC with end-constraint

Perhaps the most important practical outcome of our recent understanding of MPC stability properties is the importance of the end-constraint in eqns. (63) or (64). Such a constraint has already been incorporated in certain commercial packages with minimal effort, either heuristically or following theoretical research publications. MPC theory has made it clear that including an end-constraint in MPC on-line optimization is not merely a matter of company preference or software legacy, but rather an important step towards endowing the MPC algorithm with improved properties.

6.2.3.2 Chance constrained MPC: Robustness with respect to output constraint satisfaction

While MPC constraints that bound process inputs can be easily ensured to be satisfied by the actual system, constraints on process outputs are more elusive. That is because future process outputs within an MPC moving horizon have to be predicted on the basis of a process model (involving the process and disturbances). Because the model involves uncertainty, process output predictions are also uncertain. This uncertainty in process output predictions may result in adverse violation of output constraints by the actual closed-loop system, even though predicted outputs over the moving horizon might have been properly constrained. Consequently, a method of incorporating model uncertainty into the output constraints of the on-line optimization is needed. This would improve the robustness of constrained MPC. In this paper we introduce an approach towards achieving that goal.

The proposed approach relies on formulating output constraints of the type $y_{\min} \le y \le y_{\max}$ as chance constraints of the type

(185) $$\Pr\{y_{\min} \le y \le y_{\max}\} \ge \alpha$$

where Pr{A} is the probability of event A occurring, y is the process output bounded by $y_{\min}$ and $y_{\max}$, and α is the specified probability, or confidence level, that the output constraint would be satisfied. Under the assumption that the process output y is predicted by a linear model with normally distributed coefficients, the above chance constraint can be reformulated as a convex, deterministic constraint on process inputs. This new constraint can then be readily incorporated into the standard MPC formulation. The resulting on-line optimization problem can be solved using reliable convex optimization algorithms such as FSQP (Lawrence et al., 1997).

7 Future needs

7.1 Is better MPC needed?

A seasoned practitioner would probably be in better position to answer the above question. But then, a more relevant question might be "Is better MPC *possible*?" We claim that the answer is affirmative (to both questions!).

In our discussions with MPC practitioners (certainly not with a statistically representative sample) the most frequently expressed improvement need has been to increase the time that MPC is not in the manual mode. The reasons, however, why MPC is switched to "manual" vary widely. It appears that improvements are needed in various areas (e.g. model development, computation, programming, communications, user interface), not just MPC theory. But theory, is important, as the preceding sections of this work tried to explain. Of course, as MPC matures to a commodity status, the particular algorithm included in a commercial MPC software package, albeit very important, becomes only one of the elements that can make an MPC product successful in the marketplace. As with many products, the cost associated with MPC development, implementation, and maintenance has to be compared against its technical and economical benefits. The term "better MPC" need not imply a new variant of the traditional MPC algorithm, but rather a better way of using computers in computer-aided process operations. For example, section 6.2.2.1 made the case about integrating various levels of the process operations hierarchy. While expressing the need for such integration is relatively easy, the complexity of the problem is high enough not to allow a simple solution as a matter of implementation. Indeed, understanding the practical limitations as well as the theoretical properties of a complex computer-integrated system is a formidable challenge. Because of that, it appears that collaboration between academic and industrial forces would be beneficial for the advancement of computer-aided process operations technology.

7.2 Is more MPC theory needed?

Yes! While there has been significant progress, there are still several open issues related to MPC robustness, adaptation, nonlinearity handling, performance monitoring, model building, computation, and implementation. In general terms, there are two theoretical challenges associated with advancing MPC technology:

(a) Development of new classes of control strategies, and
(b) Rigorous analysis of the properties of control strategies.

Practice has shown that both challenges are important to address (Morari, 1991). MPC is only one tool in the broader area of computer-aided process engineering. With computer power almost doubling every year and widespread availability of highly interconnected computers in process plants, the long-term potential for dramatic developments in computer-assisted process operations is enormous (Boston et al., 1993; Ramaker et al., 1997; Rosenzweig, 1993). While improved MPC systems may be internally complex, the complexity of the design (e.g. translation of qualitative engineering requirements to design parameter specifications), operation, and maintenance of such a systems by process engineers and operators should be low, to ensure successful implementation, (Birchfield, 1997). "[In the past] complexity of design and operation were traded for the simplicity of the calculation [performed by the controller]. If control engineers had the computing devices of today when they began to develop control theory, the evolution of control theory would probably have followed a path that **simplified the design and operation** and **increased the complexity of control calculations**" (Cutler, 1995).

Our opinion is that effective use of computers will rely on integration of several different entities, performing different functions and effectively communicating with one another as well as with humans (Minsky, 1986; Stephanopoulos and Han, 1995). A broadening spectrum of process engineering activities will be delegated to computers (Nikolaou and Joseph, 1995). While MPC will remain at the core of such activity, peripheral activities and communication around regulatory control loops (e.g. process and controller monitoring, controller adaptation, communication among different control layers) will grow. Although no single dominant paradigm for such activities exists at present, it appears that MPC has a very important role to play.

Acknowledgment – Three anonymous reviewers suggested numerous improvements on the original manuscript. The author acknowledges their contribution with gratitude.

8 References

Allgöwer, F., and F. J. Doyle, III, "Nonlinear Process Control: Which Way to the Promised Land?," *Fifth International Conference on Chemical Process Control*, Kantor, J. C., C. E. García, and B. Carnahan, (Editors), AIChE Symposium Series, **93**, 24-45 (1997).

Åström, K. J., and B. Wittenmark, *Adaptive Control*, Addison-Wesley (1989).

Åström, K. J., and B. Wittenmark, *Computer Control Systems: Theory and Design*, Prentice Hall (1984).

Badgwell, T. A., "A Robust Model Predictive Control Algorithm for Stable Nonlinear Plants", *Preprints of ADCHEM 97*, Banff, Canada (1997).

Bassett, M. H., F. J. Doyle III, G. K. Gudva, J. F. Pekny, G. V. Reklaitis, S. Subrahmanyam, M. G. Zentner, "Perspectives on Model Based Integration of Process Operations", *Proceedings of ADCHEM 94*, Kyoto, Japan, (1994).

Baxley, R. A., and J. Bradshaw, *Personal communication*, Texas A&M University (1998).

Bayard, D. S., Y. Yam, and E. Mettler, "A Criterion for Joint Optimization of Identification and Robust Control," *IEEE Trans. on Autom. Control*, **37**, 986 (1992).

Bequette, B. W., "Nonlinear Control of Chemical Processes: A Review", *Ind. Eng. Chem. Res.*, **30**, 1391-1413 (1991).

Birchfield, G. S., "Trends in Optimization and Advanced Process Control in the Refinery Industry", *Chemical Process Control V*, Tahoe City, CA (1996).

Bitmead, R. R., M. Gevers, and V. Wertz, *Adaptive Optimal Control – The Thinking Man's GPC*, Prentice-Hall (1990).

Boston, J. F., H. I. Britt, M. T. Tayyabkhan, "Computing in 2001. Software: Tackling Tougher Tasks", *Chemical Engineering Progress*, **89**, 11, 7 (1993).

Box, E. P. G., "Parameter Estimation with Closed-loop Operating Data", *Technometrics*, **18**, 4 (1976).

Buckley, P. S., *Second Eng. Found. Conf. on Chem. Proc. Contr.*, Sea Island, GA, Jan. (1981).

Chen H., and F. Allgöwer, "A quasi-infinite horizon nonlinear predictive control scheme with guaranteed stability", Report AUT96-28, ETH, http://www.aut.ee.ethz.ch/cgi-bin/reports.cgi (1996).

Cheng, X. and B. H. Krogh, "Stability-Constrained Model Predictive Control with State Estimation", *ACC Proceedings* (1997).

Cheng, X., and B. H. Krogh, "A New Approach to Guaranteed Stability for Receding Horizon Control", *13th IFAC World Congress*, San Francisco, 433-438 (1996).

Chmielewski, D., and V. Manousiouthakis, "On Constrained Infinite Time Linear Quadratic Optimal Control", *Sys. Cont. Let.*, **29**, 121-129 (1996).

Choi, J., and V. Manousiouthakis, "Bounded Input/Initial State Bounded Output Stability over Ball", *AIChE Annual Meeting*, paper 191a (1997).

Clarke, D. W., Mohtadi, C., and Tuffs, P. S., "Generalized Predictive Control. Part 1: The Basic Algorithms", *Automatica*, **23**, 2, 137-148 (1987).

Clarke, D. W., Mohtadi, C., and Tuffs, P. S., "Generalized Predictive Control. Part 1: Extensions and Interpretation", *Automatica*, **23**, 2, 149-160 (1987).

Cutler, C. R., "An Industrial Perspective on the Evolution of Control Technology", *Methods of Model Based Process Control*, R. Berber (Ed.), 643-658, Kluwer (1995).

Darby, M. L., and D. C. White, "On-Line Optimization of Complex Process Units", *Chemical Engineering Progress*, 51-59 (October 1998).

Economou, C. G., *An Operator Theory Approach to Nonlinear Controller Design*, Ph.D. Thesis, Chemical Engineering, California Institute of Technology (1985).

Edgar, T. F., "Current Problems in Process Control", *IEEE Control Systems magazine*, 13-15 (1989).

Fleming, W. H.. (Chair), *SIAM Report of the Panel on Future Directions in Control Theory: A Mathematical Perspective*, SIAM Publication (1988).

Fu, L. C., and S. Sastry, "Frequency Domain Synthesis of Optimal Inputs for On-line Identification and Adaptive Control," *IEEE Trans. on Autom. Control*, 3**6**, 353 (1991).

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

García, C. E., and D. M. Prett, "Design Methodology based on the Fundamental Control Problem Formulation", *Shell Process Control Workshop*, Houston, TX (1986).

Genceli, H. and M. Nikolaou, "Design of Robust Constrained Nonlinear Model Predictive Controllers with Volterra Series", *AIChE J.*, **41**, 9, 2098-2107 (1995).

Genceli, H., and M. Nikolaou, "New Approach to Constrained Predictive Control with Simultaneous Model Identification", *AIChE J.*, **42**, 10, 2857-2869 (1996).

Gevers, M., "Towards a Joint Design of Identification and Control?", *Essays on Control: Perspectives in the Theory and its Applications*, H. L. Trentelman and J. C. Willems, Eds., 111-151 (1993).

Goodwin, G. C., and K. S. Sin, *Adaptive Filtering: Prediction and Control*, Prentice Hall (1984).

Gustavsson, I., L Ljung, and T. Söderström, "Identifiability of processes in closed loop–identifiability and accuracy aspects", *Automatica*, **13**, 59-75 (1977).

Kalman, R. E., “A New Approach to Linear Filtering and Prediction Problems”, *Trans. ASME, Ser. D: J. Basic Eng.*, **82**, 35 (1960).

Kane, L., “How combined technologies aid model-based control”, *IN CONTROL*, vol VI, No. 3, 6-7 (1993).

Kantor, J. C., C. E. García, and B. Carnahan, (Editors) *Fifth International Conference on Chemical Process Control*, AIChE Symposium Series, **93** (1997).

Keel, L. H., and S. P. Bhattacharyya, “Robust, Fragile, or Optimal?”, *IEEE Trans. AC*, **42**, 8, 1098-1105 (1997).

Klauw Van Der, A. C., G. E. Van Ingen, A. Van Rhijn, S. Olivier, P. P. J. Van Den Bosch, R. A. de Callafon, “Closed-loop Identification of a Distillation Column,” *3rd IEEE Conference*, (1994).

Kleinman, B. L., “An easy way to stabilize a linear constant system”, *IEEE Trans. AC*, **15**, 12, 693 (1970).

Kothare, M. V., V. Balakrishnan, and M. Morari, “Robust constrained model predictive control using linear matrix inequalities”, *Automatica*, **32**, 10, 1361-1379 (1996).

Landau, I. D., and A. Karimi, “Recursive Algorithms for Identification in Closed Loop: A Unified Approach and Evaluation”, *Automatica*, **33**, 8, 1499-1523 (1997).

Lee, J. H., and B. L. Cooley, “Optimal Feedback Control Strategies for State-Space Systems with Stochastic Parameters”, *IEEE Trans. AC*, in press (1998).

Lee, J. H., and Z. Yu, “Worst-Case Formulation of Model Predictive Control for Systems with Bounded Parameters”, *Automatica*, **33**, 763-781 (1997).

Lee, J. H., M. Morari, and C. E. García, “State Space Interpretation of Model Predictive Control”, *Automatica*, **30**, 707-717 (1994).

Lee, J. H., M. S. Gelormino, and M. Morari, “Model Predictive Control of Multi-Rate Sampled Data Systems”, *Int. J. Control*, **55**, 153-191 (1992).

Ljung, L., “Information Contents in Identification Data from Closed-loop Operation,” *Proc. 32nd Conference on Decision and Control*, San Antonio, TX (1993)

Ljung, L., *System Identification: Theory for the User*, Prentice-Hall (1987).

Longwell, E. J., “Chemical Processes and Nonlinear Control Technology”, *Proceedings of CPC IV*, 445-476 (1991)

Marlin, T. E., and A. N. Hrymak, “Real-time Operations Optimization of Continuous Processes”, *Fifth International Conference on Chemical Process Control*, Kantor, J. C., C. E. García, and B. Carnahan (Editors), AIChE Symposium Series, **93**, 156-164 (1997).

Mayne, D. Q., “Nonlinear Model Predictive Control: An Assessment”, *Fifth International Conference on Chemical Process Control*, Kantor, J. C., C. E. García, and B. Carnahan, (Editors), AIChE Symposium Series, **93**, 217-231 (1997).

Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Model Predictive Control: A Review”, *Automatica*, submitted (1998).

Meadows, E. S., M.A. Henson, J.W. Eaton, and J. B. Rawlings, "Receding Horizon Control and Discontinuous State Feedback Stabilization," *Int. J. Control*, 62 (5), 1217—1229 (1995).

Minsky, M., *The Society of Mind*, Simon and Schuster (1986).

Morari, M., “Advances in Process Control Theory”, *Chemical Engineering Progress*, 60-67 (October, 1988).

Morari, M., “Model Predictive Control - The Good, The Bad and The Ugly”, *Proc. of The Conf. On Chem. Process Control IV*, South Padre Island, TX, (1991).

Morari, M., “Three Critiques of Process Control revisited a Decade Later”, *Shell Process Control Workshop*, Houston, TX (1986).

Morari, M., and E. Zafiriou, *Robust Process Control*, Prentice Hall (1989).

Morari, M., and S. L. de Oliveira, “Contractive Model Predictive Control for Constrained Nonlinear Systems”, *IEEE Trans. AC*, accepted (1997).

Mosca, E., *Optimal Predictive and Adaptive Control*, Prentice Hall (1995).

National Research Council Committee, *Frontiers in Chemical Engineering: Research Needs and Opportunities*, National Academy Press (1988).

Nikolaou, M. and V. Manousiouthakis, “A Hybrid Approach to Nonlinear System Stability and Performance”, *AIChE Journal*, **35,** 4, 559-572 (1989).

Nikolaou, M., and B. Joseph, “Intelligent Control”, *ISPE 95 Proceedings*, 68-69 (1995).

Nour Eldin, H. A., *Optimierung linearer Regelsysteme mit quadratischer Zielfunktion*, Springer-Verlag (1970).

Ogunnaike, B., and R. Wright, “Industrial Applications of Nonlinear Control”, *Fifth International Conference on Chemical Process Control*, Kantor, J. C., C. E. García, and B. Carnahan (Editors), AIChE Symposium Series, **93**, 46-59 (1997).

Prett, D. M., and C. E. García, *Fundamental Process Control*, Butterworths, Stoneham, MA (1988).

Prett, D. M., and M. Morari (Editors), *Shell Process Control Workshop*, Butterworths (1987).

Prett, D. M., C. E. García, and B. L. Ramaker, *The Second Shell Process Control Workshop*, Butterworths (1990).

Qin, J., and T. Badgwell, “An Overview of Industrial Model Predictive Control Technology”, *Fifth International Conference on Chemical Process Control*, Kantor, J. C., C. E. García, and B. Carnahan (Editors), AIChE Symposium Series, **93**, 232-256 (1997).

Radenkovic, M. S., and B. Erik Ydstie, “Using Persistent Excitation with Fixed Energy to Stabilize Adaptive Controllers and Obtain Hard Bounds for the Parameter Estimation Error”, *SIAM J. Contr. and Optimization*, **33**, 4, 1224-1246 (1995).

Rafal, M. D., and W. F. Stevens, “Discrete Dynamic Optimization Applied to On-Line Optimal Control”, *AIChE J.*, **14**, 1, 85-91 (1968).

Ramaker, B., H. Lau, and E. Hernandez, “Control Technology Challenges for the Future”, *Chemical Process Control V preprints*, Tahoe City, CA (1996).

Rawlings, J. B., and K. R. Muske, “The stability of constrained receding horizon control”, *IEEE Trans. AC*, **AC-38**, 1512-1516 (1993).

Rawlings, J. B., E. S. Meadows, and K. R. Muske, “Nonlinear Model Predictive Control: A Tutorial and Survey”, *Proceedings of ADCHEM ’94*, 203-214, Kyoto, Japan (1994).

Richalet, J. A., A Rault, J. D. Testud, and J. Papon, “Model Predictive Heuristic Control: Application to Industrial Processes”, *Automatica*, 13, 413 (1978).

Robertson, D. G., J. H. Lee, and J. B. Rawlings, “A moving horizon-based approach for least-squares estimation”, *AIChE J.*, **42**, 8, 2209-2224 (1996).

Rosenzweig, M., “Chemical Engineering Computing in 2001”, *Chemical Engineering Progress*, **89**, 11, 7 (1993).

Schrama, R. J. P., “Accurate Identification for Control: The Necessity of an Iterative Scheme,” *IEEE Trans. on Autom. Control*, **37**, 991 (1992).

Schwarm, A., and M. Nikolaou, “Chance Constraint formulation of Model Predictive Control”, *AIChE Annual Meeting* (1997).

Scokaert, P. O. M., Mayne, D. Q., and J. B. Rawlings, “Suboptimal model predictive control”, *IEEE Trans. AC*, in press

Söderström, T., I. Gustavsson, and L. Ljung, “Identifiability Conditions for Linear Systems Operating in Closed Loop,” *Int. J. Control*, **21**, 243 (1975).

Stephanopoulos, Geo., and C. Han, “Intelligent Systems in Process Engineering: A Review”, *Proceedings of PSE ’94*, 1339-1366 (1994).

Stephanopoulos, Geo., *Chemical Process Control*, Prentice Hall (1984).

Stephanopoulos, Geo., O. Karsligil, and Matthew Dyer, “A Multi-Scale Systems Theory for Process Estimation and Control”, *preprints, NATO-ASI Series*, Antalya, Turkey (1997).

Thomas, Y. A., “Linear Quadratic Optimal Estimation and Control with Receding Horizon”, *Electronics Letters*, **11**, 19-21 (1975).

Van den Hof, P. M. J., and R. J. P. Schrama, “Identification and Control–Closed Loop Issues”, *10th IFAC Symposium on System Identification*, Copenhagen, Denmark (July, 1994).

Vidyasagar, M., *Nonlinear Systems Analysis*, 2nd. Edition, Prentice Hall (1993).

Williamson, D. *Digital Control and Implementation – Finite Wordlength Considerations*, Prentice Hall (1991).

Zafiriou, E., “On the Effect of Tuning Parameters and Constraints on the Robustness of Model Predictive Controllers”, *Proceedings of Chemical Process Control – CPC IV*, 363-393 (1991).

Zafiriou, E., “Stability of Model Predictive Control with Volterra Series”, *AIChE Annual Meeting*, St. Louis (1993).

Zheng, A., and M. Morari, "Control of Linear Unstable Systems with Constraints", *Proceedings of the American Control Conference*, Seattle, WA, 3704-3708 (1995).

Zheng, Z. Q., and M. Morari, "Robust Stability of Constrained Model Predictive Control", *Proceedings of the American Control Conference*, session WM7, 379-383, San Francisco (1993).

Ziegler, J. G., and N. B. Nichols, "Optimum Settings for Automatic Controllers", *Trans. ASME*, **64**, 759 (1942).